Intégration de la sécurité dans un cadre d'architecture d'entreprise à partir du risque

Zakariya Kamagate

COLLEGES BRETAGNE LOIRE
SCIENCES POUR L'INGENIEUR ET LE NUMERIQUE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

THÈSE DE DOCTORAT DE

L'ÉCOLE NATIONALE SUPÉRIEURE
MINES-TÉLÉCOM ATLANTIQUE BRETAGNE
PAYS DE LA LOIRE – IMT ATLANTIQUE

ÉCOLE DOCTORALE N° 648
Sciences pour l'Ingénieur et le Numérique
Spécialité : *Informatique*

Par
Zakariya KAMAGATE

Intégration de la Sécurité Dans un Cadre d'Architecture d'Entreprise à Partir du Risque

Thèse présentée et soutenue à IMT Atlantique, Campus de Brest, le 14/12/2023
Unité de recherche : LAB-STICC
Thèse N° : 2023IMTA0388

Rapporteurs avant soutenance :

Bouabid EL OUAHIDI	Professeur, Universite Mohamed V-Rabat
Nabil TABBANE	Professeur, SupCom Tunis

Composition du Jury :

Président :	Christophe CLARAMUNT	Professeur ENSAM et Ecole Navale
Examinateurs :	Jamal EL HACHEM	Maitresse de Conférences, Université de Bretagne Sud, Vannes
	Olivier ASSEU	Professeur, INHP Yamoussoukro
	Bouabid EL OUAHIDI	Professeur, Universite Mohamed V-Rabat
	Nabil TABBANE	Professeur, SupCom Tunis
Dir. de thèse :	Yvon KERMARREC	Professeur à IMT Atlantique
Encadrant de thèse :	Jacques SIMONIN	Directeur Des Etudes à IMT Atlantique

A Sindou, mon jumeau-néveu-ami-frère, confident qui vient à peine de me quitter avec une douleur indescriptible- Repose en paix SK Nursi

A mon père et à ma mère

A ma femme

A ma fille

A mes frères et Soeurs

A mon Directeur de thèse

A mon encadrant de thèse

Au Département LUSI

Au Directeur Général de l'ESATIC

Aux DP et DRIT de l'ESATIC

A mes collègues de l'ESATIC

Remerciements

Je tiens à remercier toutes les personnes qui m'ont aidé au cours de cette thèse, oh combien tumultieuse !. Mes pensées vont tout d'abord à Yvon Kermarrec, mon directeur de thèse pour sa disponibilité, son implication entière et sa franche collaboration durant ce long voyage. Au délà de la sphère professionnelle, il a surtout eu une attention sociale et psychologique à mon endroit et a su à chaque moment difficile me donner la force pour reprendre confiance en moi, tenir et continuer l'aventure. Je ne saurais trouver les mots pour décrire mon encadrant de thèse, Jacques Simonin, pour qui j'ai une immense adminration. Sa patience, sa disponibilité, sa hauteur d'esprit face aux situations, m'ont à chaque fois rassuré d'être dans de bonnes mains, malgré les difficultés et les doutes liées à la thèse.
Remerciements particuliers à Laurent Nana et Kamel Karoui, les membres de mon comité de suivi individuel (CSI) qui ont été évalué mon travail chaque année. Merci pour vos remarques et conseils pertinents lors de ces CSI.
Merci à Prof. Adama Konaté, mon Directeur à l'école ESATIC (Abidjan) en Côte d'Ivoire la facilitation du partenariat avec IMT Atlantique à travers lequel nous avions l'opportunité de faire cette Thèse à IMT Atlantique à Brest. Merci également aux Directeurs de la pédagogie de l'ESATIC Prof. Soro ETienne ainsi que celui de la DRIT ProF. Asseu Olivier pour leur soutien énorme. Merci à tous mes collègues de l'ESATIC.
Remercier Madame Delphine LUCAS, assistance de direction à l'ED, est un épheumiste, pour le rôle déterminant qu'elle a joué dans ma thèse. Elle, qui s'est battue avec une détermination et implication désinteressé pour sauver mon projet de thèse à plusieurs reprises. Je ne saurais vous le rendre.
Mes pensées vont également à l'assistante du département LUSSI Ghislaine Le Gall. Merci beaucoup pour votre dévouement à rendre notre séjour facile et pour le soutien dans toutes sortes de tâches administratives.
Je remercie également Philippe Tanguy et son épouse pour leur amitié. Un grand merci à Philip Lenca Directeur de LUSSI et Emmanuel Braux du DISI. Merci à tous les collègues du departement LUSSI, pour votre soutien et vos conseils.
Merci à mes collègues Raogo Kaboré, N'Goran Rodrigue avec qui j'ai effectué

l'aventure de cette Thèse, et partagé le bureau à LUSSI. Merci pour toutes les idées et l'amitié durant toutes ces années. Enfin, je tiens à remercier toutes les personnes impliquées à tous les niveaux dans la réalisation de cette thèse.

Acronymes

C | D | E | G | I | L | M | P | T

C

CICM-L Computation Independent Contextual Model-Logical. 207
CICM-R Computation Independent Contextual Model of Risk. 207
CIM Computation Independent Model. 207

D

DoS Denial-of-Service. 207

E

EA Enterprise Architecture. 207

G

GCD Greatest Common Divisor. 207

I

IDM Ingénierie Dirigée par les Modèles. 207

L

LCM Least Common Multiple. 207

M

MDA Model-Driven Architecture. 207
MDE Model-Driven Engineering. 207

P

PICM-R Platform Independent Contextual Model of Risk. 207
PIM Platform Independent Model. 207

PSM Platform Specific Model. 207

T

TCM-BR Transformation Contextual Model of Business Risk. 207
TCM-LIS Transformation Contextual Model of Logical Information System. 207
TCM-LR Platform Independent Contextual Model of Physical. 207
TCM-LR Transformation Contextual Model of Logical Risk. 207
TCM-PIS Transformation Contextual Model of Physical Information System. 207
TCM-PR Transformation Contextual Model of Physical Risk. 207

Table of Contents

Remerciements xi

Acronymes xiii

Table of Contents xv

Liste des tableaux xvii

Table des figures xix

Introduction Générale 1

I Etat de l'Art 11

1 Cadre Général 13

II Contributions 71

2 Intégration contextuelle du risque dans l'architecture métier du système 73

3 Intégration contextuelle du risque dans l'architecture logique du système 105

4 Intégration contextuelle du risque dans l'architecture physique du système 131

5 Expérimentation et Vérification 149

Conclusion Générale 185

Bibliographie **191**

Table des matières **209**

Liste des tableaux

1.1 Bénéfices d'architecture d'entreprise (extrait de [147]) 18
1.2 Méthodes d'analyse de risque Légendes : (+) Couverture limitée à une seule phase de développement (++) Couverture de deux ou plusieurs phases de développement mais pas la totalité (-) Notion non couverte par la méthode 57
1.3 Méthodes de développement basées sur les modèles, la sécurité et le risque Légende : ++ : couvert complétement + : couvert partiellement - : Non couvert 67

2.1 Modèle de menaces STRIDE et mesure d'atténuation 83
2.2 Echelle de besoins de sécurité 84
2.3 Critères de gestion des risques (extrait de EBIOS, 2022) 85
2.4 Métrique de gravité des événements redoutés 86
2.5 Echelle de vraisemblance des scénarios de menace 86
2.6 Métriques (seuils) pour l'évaluation du risque dans EBIOS . . . 87
2.7 Instanciation du modèle de risque métier (TCM-BR) pour le service métier BSOnlineShopping 100

3.1 Alignement des attributs des donnes métiers et des composants applicatifs logiques du SI 121
3.2 Alignement des risques associés et des composants applicatifs logiques du SI technique 121
3.3 Comparaison entre notre approche et UMLsec 127

4.1 Extrait des services du SI de TCM-PR utiles au traitement du risque dans BSOnlineShopping 143
4.2 Alignement des opérations logiques et de la plateforme technique 143

5.1 Actifs/biens métiers du service métier BOnlineShoping 151
5.2 Etude des scénarios de menaces 155
5.3 Instanciation du modèle de menace STRIDE pour le service métier BSOnlineShopping dans TCM-BR. 157

5.4 Extrait des services du SI de TCM-PR utiles au traitement du risque dans BSOnlineShopping 159
5.5 CICM-R2CICM-L : Alignement des attributs des donnes métiers manipulées dans BSOnlineShopping et des composants applicatifs logiques du SI fonctionnel les supportant 165
5.6 CICM-L2PIM (ET) : Alignement des risques associés à BSOnline-Shopping et des composants applicatifs logiques du SI technique les supportant 165
5.7 PICM-P2PICM-R : Alignement des opérations logiques et de la plateforme technique supportant BSOnlineShopping avec les services réutilisables du SI 173

Table des figures

1.1 TOGAF ADM cycle[62] 22
1.2 Concepts essentiels de sécurité et de risque et leur position dans le TOGAF ADM[130] 28
1.3 Matrice SABSA [149] 29
1.4 Relation entre le cycle de SABSA et TOGAF ADM [140] 30
1.5 Taxonomie d'attributs métiers SABSA [149] 31
1.6 Notions de base en ingénierie des modèles[19] 36
1.7 Couches de specification du MDA [128] 37
1.8 Un processus en Y dirigée par les modèles [33] 39
1.9 La pyramide des niveaux de modélisation[19] 40
1.10 Schéma de base d'une transformation de modèles [19] 42
1.11 Types de transformation et leurs principales utilisations[33] . . 44
1.12 Architecture du standard QVT[129] 46
1.13 Alignement entre MDA et TOGAF[24] 47
1.14 Méta-modèle de SecureUML [100] 59
1.15 Un extrait du profil UMLsec [83] 60
1.16 Méta-modèle ISSRM [104] 65

2.1 Approche de modélisation des menaces et des risques 79
2.2 Diagramme de flux de travail pour la modélisation des menaces 81
2.3 Diagramme de flux de donnée des menaces DFD [133] 81
2.4 Echelle d'évaluation du niveau de risque 87
2.5 Processus de gestion des risques de sécurité 88
2.6 Artefacts de sécurite dans la Phase A TOGAf : Archtecture métier 91
2.7 TCM-BR : Méta modèle du risque métier 93
2.8 Processus de transformation du modèle CIM vers le modèle CICM-R (CIM2CICM-R) 95
2.9 CICM-R : Architecture de Conception du Métier (CIM et TCM-BR) 96
2.10 Spécifications du service métier d'achat en ligne (BSOnlineShopping) : allignement entre tâches métiers et risques. 97
2.11 Analyse du diagramme DFD du service metier 98

2.12 Analyse et évaluation des risques avec EBIOS 99
2.13 Reformulation des spécifications du service métier d'achat en ligne (BSOnlineShopping) 100

3.1 Architecture logique de SABSA 109
3.2 TCM-LR : Méta modèle contextuel logique du risque 110
3.3 Interaction entre Opération Logique du SI et le risque 111
3.4 Modèle d'architecture logique du SI technique supportant le domaine métier de l'achat en ligne. 113
3.5 Transformation du modèle CICM-R vers le modèle PIM (CIM-R2PIM) 114
3.6 Pattern de composant applicatif logique : TCM-LIS) [157] . . . 115
3.7 CICM-L : Méta Modèle Contextuel de Composant Logique) [157] 116
3.8 PIM : Modèle des composants d'application logiques du SI associé à TCM-LIS (cf. [157]) 117
3.9 Méta modèle de dépendance entre composants applicatifs logiques 118
3.10 Arbre de dépendance logique du service SI ISSOnlineShopping createSession opening 119
3.11 Illustration du PIM resultant de la transformation par enrichissement ET. 120
3.12 PIM statique : Modèle de composants applicatifs logiques supportant le service métier BSOnlineShopping. 122
3.13 PIM statique : Données logiques supportant le service métier BSOnlineShopping. 123
3.14 PIM Dynamique : Modèle de composants applicatifs logiques supportant le service métier BSOnlineShopping. 125

4.1 TCM-PR : Modèle contextuel physique du risque 135
4.2 Processus de transformation du modèle PIM vers le modèle PSM (PIM2PSM) 136
4.3 Platform Description Model (PDM) 140
4.4 Architecture de Conception Physique (PSM) 141
4.5 PSM illustration resulting from ST substitution transformation. 142
4.6 Modèle de composants applicatifs physiques réalisant l'arbre de dépendances de sommet LACSessionManagement 145
4.7 Données physiques de création de session 146
4.8 Modèle physique dynamique du service createSessionOpening 147

5.1 Spécifications du service métier d'achat en ligne (BSOnlineShopping) 149
5.2 Actifs systèmes du service métier BOnlineShopping 151

5.3 Détermination des évenements rédoutés-1 152
5.4 Détermination des évenements rédoutés-2 153
5.5 Evaluation des évenements redoutés 153
5.6 Détermination des menaces basée sur STRIDE avec le DFD . . 154
5.7 Evaluation des scénarios de menace 155
5.8 Estimation du risque 156
5.9 Décision de traitement du risque 156
5.10 TCM-LR : Modèle d'architecture logique du SI technique supportant le domaine métier de l'achat en ligne 158
5.11 Intégration du traitement du risque dans l'approche MDA contextualisée 161
5.12 TCM-LIS : Modèle d'architecture logique du SI fonctionnel supportant le domaine métier de l'achat en ligne 162
5.13 CICM-R : Reformulation des spécifications du service métier d'achat en ligne (BSOnlineShopping) conformes au méta-modèle du CIM 164
5.14 PIM statique : Modèle de composants applicatifs logiques supportant le service métier BSOnlineShopping 166
5.15 PIM statique : Données logiques supportant le service métier BSOnlineShopping 167
5.16 PIM dynamique : Service ISSOnlineShopping createSession Opening du SI supportant le service métier BSOnlineShopping . . 169
5.17 PIM dynamique : Service ISSOnlineShopping create ORDER du SI supportant le service métier BSOnlineShopping 170
5.18 PIM dynamique : Service ISSOnlineShopping :updateSession closing du SI supportant le service métier BSOnlineShopping . . . 170
5.19 PIM dynamique : Modèle dynamique d'orchestration des services du SI supportant le service métier BSOnlineShopping. 171
5.20 PSM statique : Modèle de composants applicatifs physiques réalisant l'arbre de dépendances de sommet LACSessionManagement 174
5.21 PSM statique : Modèle de composants applicatifs physiques réalisant l'arbre de dépendances de sommet LACOrderManagement 175
5.22 PSM statique : Données logiques supportant le service métier BSOnlineShopping. 176
5.23 PSM dynamique : Modèle physique dynamique du service ISSOnlineShopping :create Session opening du SI 177
5.24 PSM dynamique : Modèle physique dynamique du service ISSOnlineShopping : create Order date du SI 177
5.25 PSM dynamique : Modèle physique dynamique du service ISSOnlineShopping :updateSession$_{c}losingduSIralisantsonmodlephysiquelogique$ 178

5.26 Script SQL de génération de la base de données physiques relationnelles 179
5.27 Squelette du code Java des services du SI. 180

Introduction Générale

Contexte

Dépendance de la société moderne aux SI

Les sociétés actuelles, l'économie moderne, et autres centres de production dépendent fortement des systèmes d'information et leur interconnexion[83] , qui permettent de gérer et de coordonner des actions complexes. Ces systèmes sont eux-mêmes composés d'autres systèmes [157][173] (comme des systèmes informatiques, des systèmes de traitement de données issues de capteurs) et reliés via internet afin d'assurer leur accessibilité depuis l'extérieur et de permettre ainsi leur suivi et leur maintenance à distance.

Complexité des architectures des systèmes /logiciels actuels

La complexité de ces systèmes s'explique en partie par un assemblage / juxtaposition dans le temps de composants logiciels[173] et autres services afin de satisfaire les évolutions des fonctionnalités attendues par le système lui-même[157]. Cette complexité se traduit principalement par une difficulté à maitriser / connaitre et comprendre cette architecture et d'en donner ne serait-ce qu'une représentation graphique[141][109].

SI sont la cible d'attaques

Ces systèmes complexes sont également au cœur de préoccupations en lien avec la sécurité et peuvent être l'objet d'attaques, afin d'en prendre le contrôle

ou d'obtenir les données et informations qui y sont stockées, voire de les détruire[154][152].

Utilisation inadéquate des mécanismes de sécurité

A ces systèmes, les concepteurs rajoutent d'autres composants assurant les fonctions de sécurité et ceci sans maîtriser l'adéquation de la solution au système cible, en particulier aux exigences métiers associées [83].

L'ingénierie des systèmes et la problématique de sécurité

L'ingénierie des systèmes/logiciels ou Génie Logiciel est une approche interdisciplinaire, dont le but est de formaliser la conception et le développement des systèmes[2][146]. Elle comprend les phases de définition des besoins (exigences), de conception, d'implémentation et la validation du système[39].
Le problème de sécurité lié aux Systèmes d' Information (SI) constitue une préoccupation dans le domaine de l'ingénierie des systèmes[107] [109][39]. L'environnement des SI de nombreuses entreprises sont composés d'un nombre de systèmes connectés pour former un système complexe de systèmes[173][169]. La sécurité est enfin un problème complexe difficile à maîtriser[97][94].

Exigences fonctionnelles et de sécurité indissociées

Souvent, les exigences de sécurité sont mêlées aux exigences fonctionnelles lors des activités de spécification des besoins, ce qui ne permet pas une séparation claire des préoccupations des parties prenantes, permettant de traiter entièrement, chacune d'elle[102][103]. Dans ce cas, il est difficile d'intégrer les exigences de sécurité correctement dans le processus de développement logiciel. D'où le besoin de séparer les exigences fonctionnelles (métiers) des exigences non-fonctionnelles telle que la sécurité.

Passage informel des exigences de sécurité en politique de sécurité

Par ailleurs, le problème du passage de l'expression des besoins à la définition de la politique de sécurité constitue un autre verrou. Ce passage est souvent mis en œuvre de façon informelle, non-structurée, et non-automatique, provoquant des pertes d'information, et donc la génération d'une politique de sécurité incorrecte[39].
De ce qui précède, les méthodes de développement des systèmes logiciels présentent des limites liées à l'intégration de la sécurité dans le processus de développement du système. De plus, en raison de la pression économique, le temps de développement est souvent court et la fréquence des modifications requises est elevée. Cela conduit en pratique à de nombreux défauts de sécurité qui sont exploités par les attaquants. Tous ces problèmes montrent le besoin d'une méthode d'ingénierie adéquate et holistique dans ce domaine, mais aussi adaptée à des évolutions dans le temps.

Motivation et challenge

L'Architecture d'Entreprise ou Enterprise Architecture (EA) en anglais, (y compris l'architecture de sécurité) consiste à aligner le système métier de l'entreprise aux systèmes d'information et physique qui le supportent, pour atteindre les objectifs métiers de manière efficace et efficiente [62]. Ainsi, elle réduit la complexité de l'organisation en fournissant des points de vue spécifiques sur un modèle complet intégré [73].
L'architecture de sécurité d'entreprise cherche à aligner les mesures de sécurité sur les objectifs métiers en définissant les relations entre les composants sur les différents niveaux d'architecture, assurant ainsi traçabilité et justification[134][4]. Cependant, «*une véritable intégration de la sécurité dans l'architecture d'entreprise nécessite une approche d'ingénierie système. Ensuite, la sécurité et le risque sont pris en compte dès que possible dans le cycle de vie du développement de l'ingénierie système* »[130].
Au fil des ans, un objectif important des chercheurs en logiciels est de dévelop-

per des techniques pour modéliser les concepts de domaine en fonction de leur intention (préoccupation) de conception plutôt que l'environnement de mise en œuvre sous-jacent [1]. La gestion efficace de la complexité des techniques joue un rôle très important dans le développement de systèmes d'information précis, fiables et maintenables qui deviennent de plus en plus grands, complexes et distribués. Dans ce contexte, l'Ingénierie Dirigée par les Modèles (IDM), en anglais Model-Driven Engineering (MDE) a été suggérée pour améliorer la qualité des systèmes logiciels complexes [89][33].

L'IDM est une approche qui traite les modèles comme un artefact important pendant le développement de logiciels[20][33]. Elle envisage précisément le problème et le domaine de solution à différents niveaux d'abstractions[146]. L'IDM définit des méthodologies pour chaque niveau d'abstraction et fournit des techniques pour abaisser le niveau d'abstraction en définissant les relations entre les modèles participants. Ainsi elle apporte des bénéfices considérables dans le domaine de l'ingénierie logicielle tels que la productivité, l'interopérabilité, la traçabilité, la réutilisabilité, la maintenabilité... [9] [101][14].

Le framework de conception de logiciel de l'OMG (Object Management Group.)– appelé Model-Driven Architecture (MDA) (Model-Driven Architecture) - est considéré comme une implémentation de l'IDM [146]. L'approche MDA propose trois niveaux d'abstraction : le niveau CIM (Computation Independent Model), le niveau PIM (Platform Independent Model) et le niveau PSM (Platform Specific Model). Pour développer un système, une série de transformation entre modèles est effectuée. En utilisant le framework MDA, la fonctionnalité logicielle est modélisée avec un langage de modélisation standard (UML) en tant que modèle indépendant de plate-forme (PIM), puis transformé en un ou plusieurs modèles spécifiques à la plate-forme (PSM).

Le défi dans notre thèse est formulé dans le sujet : **"Intégration de la Sécurité Dans un Cadre d'Architecture d'Entreprise à partir du Risque"**.

La question est comment combiner les notions d'Architecture d'Entreprise et d'Ingénierie Dirigée par les modèles pour répondre aux préoccupations de la complexité des systèmes logiciels et de la sécurité. Plus précisément, les instances du MDA sont-elles une solution idéale pour l'intégration de la sécurité dans l'EA, en définissant des modèles de sécurité, sous forme de contexte, dédiés

à la sécurité du système ?

En notre connaissance, aucune étude, dans la littérature, basée sur MDA, ne propose une telle approche permettant d'intégrer complètement des aspects de sécurité en prenant en compte les niveaux architecturaux de l'entreprise (métier, IS, physique) et leurs relations. Par conséquent, dans cette thèse nous proposons une méthode de définition et d'intégration de modèles contextuels de sécurité basée sur l'EA, et outillée par l'IDM, notamment par le MDA. Cette méthode fournit un cadre dans lequel les concepts de sécurité sont modélisés en utilisant UML au niveau de l'abstraction CIM et sont fusionnés avec les modèles d'exigences métier pour générer une instanciation au niveau PIM. Ces PIMs améliorés de sécurité sont transformés en spécification standard (PSM) qui à son tour configure notre architecture de référence afin d'obtenir un système d'application sécurisé.

Objectifs de recherche

L'objectif principal de cette recherche est de concevoir une méthode, de développement d'un système logiciel, orientée modèle, cohérente avec une architecture d'entreprise, qui intègre les besoins de sécurité.
La recherche se concentre sur une intégration entre ces éléments séparément et sur la façon dont ces éléments sont liés les uns aux autres pour fournir une approche globale de l'architecture. Notre approche est fondée sur la définition de (méta) modèles contextuels de sécurité et du processus d'intégration des (méta) modèles par transformation.
Les objectifs spécifiques consistent à :

- définir une méthode de construction de méta modèles associés aux points de vue (métier, logique, technologique, physique) de l'architecture d'entreprise. Ces méta modèles sont fondés sur le cadre TOGAF, intégrant les besoins de sécurité, sous forme de contexte ;
- définir une méthode permettant d'intégrer les différents modèles contextuels ;
- expérimenter cette approche à l'aide d'une étude de cas.

Questions de recherche

Les questions de recherche sont dérivées des objectifs de recherche. Trois questions de recherche principales sont au cœur de la recherche menée. Les questions de recherche et leurs sous-questions sont décrites ci-dessous :

QR1 : Comment intégrer la sécurité dans l'architecture métier associée au processus d'ingénierie système, via les risques ?
La première question de recherche est divisée en deux sous-questions :

- QR1.1 : Quels éléments de l'architecture métier peut-on cibler pour une intégration de la sécurité dans le système ?
- QR1.2 : L'approche MDA pour l'intégration de la sécurité au niveau métier est-elle pertinente comme mécanisme ?

QR2 : Quelle méthode de conception de la sécurité, via les risques , pour l'architecture logique supportant une sécurité intégrée dans le métier ?
La deuxième question de recherche est divisée en deux sous-questions :

- QR2.1. Comment intégrer la sécurité à la vue logique d'un système à partir de la vue métier ?
- QR2.2. Comment intégrer les composants logiques de sécurité à la réalisation dynamique des cas d'utilisation d'un système ?

QR3 : Quel impact de la mise en œuvre de la sécurité via les risques dans l'architecture physique et les conséquences sur l'implémentation du système ?

Approche et contributions

La contribution principale de notre thèse est l'intégration des exigences de sécurité basées sur le risque dans le processus de développement de système logiciel. Cela a consisté en la définition de méta-modèles contextuels de sécurité fondés sur l'architecture d'entreprise qui ont été intégrés à l'aide d'outil de modélisation. Il s'agit de modèles contextuels des architectures métier, logique et physique

présentés comme suit :

Contribution 1
Le premier contexte est métier. C'est un contexte à la description du besoin métier d'un système permettant de mieux concevoir l'architecture métier supportée par le système au sens de son exhaustivité. La particularité est la prise en compte des besoins de sécurité du système à un haut niveau d'abstraction, dérivés en exigences de sécurité. Ces exigences sont ensuite héritées et implémentées dans les architectures inférieures (logique, physique), suivant les étapes du processus de développement du système.

Contribution 2
Le deuxième contexte est logique (fonctionnel), intégrant des composants logiques de sécurité à l'architecture cible du SI. Ce contexte est intégré avec la description du besoin métier du système. L'objectif est toujours de mieux concevoir le niveau logique du système en étant cohérent avec la vue logique du SI intégrant la sécurité.

Contribution 3
Le troisième contexte est l'architecture physique ou applicative de l'existant au niveau du système informatique de l'entreprise. Ce contexte est intégré avec l'architecture logique du système afin de réutiliser au mieux les solutions applicatives existantes de sécurité dans le SI. La particularité de notre contribution est de faciliter la réutilisation de solutions outillées de sécurité, plutôt sur étagère . Un cas d'étude basé sur un scénario d'achat d'articles en ligne a permis d'expérimenter notre méthode.

Organisation de la thèse

Le reste de ce manuscrit est structuré comme suit :

Partie I : Etat de l'Art

Elle est composée d'un chapitre (Cadre général) traitant de l'état de l'art.

Chapitre 1 : Cadre général

Ce chapitre fait l'état de l'art des thèmes et concepts clefs de notre étude. Dans un prémier temps, il aborde l'approche d'Architecture d'Entreprise, ses caractéristiques et le choix des éléments requis pour la définition d'une architecture d'entreprise pour le développement d'un système sécurisé.
La seconde partie aborde l'approche de l'IDM, ses caractéristiques et ses composantes.
La troisième partie de ce chapitre est une étude comparative des méthodes et languages de developpement d'un système logiciel satisfaisant les exigences de sécurité et de risque, dépuis les phases de recueil des besoins.

Partie II : Contribution

Chapitre 2 : Intégration contextuelle du risque dans l'architecture métier du système présente notre prémière contribution. C'est la construction d'un modèle contextuel de risque lié au métier qui intègre le processus de développement du système.

Chapitre 3 : Intégration contextuelle du risque dans l'architecture logique du système, présente notre deuxième contribution. C'est la construction d'un modèle contextuel de risque lié à l'architecture logique, qui intègre le processus de développement du système et aligné au métier.

Chapitre 4 : Intégration contextuelle du risque dans l'architecture physique du système, présente notre troisième contribution. C'est la construction d'un modèle contextuel de risque lié à l'architecture physiique du système, qui intègre le processus de développement du système et supporte l'architecture logique.

Chapitre 5 : Expérimentation qui est l'illustration de notre méthode par un cas d'étude de eCommerce.

Conclusion générale et perspectives

Première partie

Etat de l'Art

Chapitre 1

Cadre Général

1.1 Introduction

De nombreuses études existantes ont montré que les causes de la plupart des attaques systèmes ne sont pas liées à des vulnérabilités de codage qui s'appliquent à ces systèmes individuels, ni des problèmes liés à l'environnement d'exécution ou à la technologie en place [14][165][162][42]. Le succès de ces attaques est dû à des problèmes liés à la façon dont les systèmes au sein des organisations sont structurés[42]. Par conséquent, il est nécessaire considérer la sécurité en ce qui concerne tous les composants des systèmes de l'organisation, y compris les données, les processus et même les employés[97][107]. L'approche la plus prometteuse pour appréhender le système/logiciel avec ses différents aspects est l'architecture d'entreprise (Enterprise Architecture-EA)[78]. L'Architecture d'Entreprise est une approche qui vise à gérer des systèmes complexes au sein de l'organisation et de collaborer de la manière la plus efficace [62]. Cependant une intégration correcte de la sécurité dans l'architecture d'entreprise requiert une approche d'ingénierie système permettant d'intégrer de façon cohérente les différents éléments du système logiciel[106]. L'Ingénierie du logiciel s'oriente aujourd'hui vers l'Ingénierie Dirigée par les Modèles (IDM)[20] qui vise à fournir un grand nombre de modèles pour exprimer séparément chacune des préoccupations des utilisateurs, des concepteurs, des architectes[33].
L'objectif principal de ce chapitre est de déterminer les éléments requis pour

le développement d'un système satisfaisant les besoins de sécurité. Pour cela, nous analysons comment un cadre basé sur les concepts de l'EA bien établis tels que TOGAF (The Open Group Architecture Framework)[62], combiné à SABSA (Sherwood Applied Business Security Architecture)[149] peuvent permettre une modélisation du système basée sur l'IDM, plus spécifiquement le MDA (Model-Driven Architecture)[89] en s'appuyant sur des langages d'intégration des différents composants du système.
Ce chapitre détaille le cadre général de notre thèse. Il est organisé comme suit :

— Dans la section 1.2, nous présentons l'architecture d'entreprise pour le développement de systèmes sécurisés. Nous justifions ici le choix des cadres TOGAF et SABSA comme éléments pour notre architecture et la nécessité de développer le système dans un contexte de modèle comme solution d'intégration des différentes préoccupations.

— Dans la section 1.3, nous faisons une présentation de l'IDM ou en anglais, MDE(Model-Driven Engineering), du MDA et des langages de modélisation UML[63] et de transformation (QVT, ATL)[59]. Ensuite nous étudions le lien entre l'IDM et l'EA, notamment l'alignement entre les concepts TOGAF ADM et MDA qui constituent les éléments de base de notre méthode.
— Dans la section 1.4, nous présentons un ensemble de travaux qui se focalisent sur l'ingénierie des exigences de sécurité dans le développement système. Nous distinguons les méthodes qui utilisent des modèles comme la base de leur principe et celles basées sur la gestion des risques. Nous faisons une étude comparative de ces méthodes et expliquons notre position par rapport à l'ensemble de travaux.

1.2 Architecture d'entreprise pour le développement de systèmes sécurisés

Dans cette section, nous nous concentrons sur les éléments requis pour la structuration d'un système répondant aux exigences métiers et supportées par les architectures logique et physique de façon cohérente. L'Architecture d'Entreprise s'efforce de satisfaire ce besoin. Par conséquent, nous allons présenter ce domaine avec ses caractéristiques.
Une initiative importante dans le domaine de l'architecture d'entreprise est la définition de cadres ou Framework[147], qui visent les concepts et activités structurants nécessaires à la conception et à la construction d'un système. Parmi les cadres les plus utilisés on peut citer Zachman[179], TOGAF[62]. TOGAF est un cadre d'architecture d'entreprise approprié pour l'alignement du métier et la technologie et au-delà du Framework, il propose une méthodologie très construite à travers son cycle de developpement qui sert de guidage pour l'architecture des systèmes[30] [54]. En outre, TOGAF est envisagé comme un cadre général qui peut être étendu et complété par des concepts spécifiques à la sécurité, notamment en l'associant à d'autres Frameworks tels que SABSA[109]. SABSA est un cadre d'architecture de sécurité qui complète le cadre TOGAF à travers ses modules relatifs à la sécurité et au risque[109][140].
Dans cette section, nous étudions comment TOGAF et SABSA peuvent constituer un duo pour la construction d'un système de sécurité basé sur le concept d'architecture d'entreprise.
Ensuite, nous présentons les langages de modélisation permettant de modéliser le système avec la prise en compte des spécifications fonctionnelles comme non fonctionnelles telles que la sécurité. Dans le domaine de l'EA, les langages les plus connus sont archimate[72] et UML[127]. Cependant, le critère du développement du système dans une perspective d'ingénierie système, à partir des notions de modèles, présente UML comme langage approprié à cet effet[63].
Dans ce qui suit, nous donnons une définition et caractéristiques d'une architecture d'entreprise, ensuite nous présentons des cadres d'une architecture d'entreprise, notamment nous donnons un détail du cadre TOGAF et sa relation

avec le cadre SABSA avant de terminer avec le langage de modélisation choisi pour notre étude qui est UML.

1.2.1 Définition et caractéristiques d'une architecture d'entreprise

Une entreprise est une ou plusieurs organisations qui ont une mission, des buts et des objectifs d'offrir un extrant tel qu'un produit ou un service (ISO 2000)[74]. Cela inclut l'entreprise étendue (intégration des fournisseurs et des clients) et l'entreprise virtuelle (orientée à l'interopérabilité des entreprises dynamiques en réseau). Les entreprises sont diverses, constituées de multiples éléments interconnectés, à la fois techniques et sociaux. Le développement de grandes organisations ou systèmes complexes impliquent de nombreuses personnes, parties prenantes, chacune avec leur propre point de vue[141].
Les architectures sont un moyen important pour documenter, comprendre et maîtriser la complexité des composants d'une entreprise[169].
Comme décrit dans American National Standards Institute/Institute of Electrical and Electronics Engineers (ANSI/IEEE) Std 1471-2000, une architecture est " *l'organisation fondamentale d'un système, incarné dans ses composants, leurs relations les uns avec les autres et avec l'environnement, et le principe guidant sa conception et son évolution*" (IEEE Comp. Soc. 2000)[74].
L'Architecture d'Entreprise en anglais EA (Enterprise Architecture) est *"un ensemble cohérent de principes, de méthodes et de modèles qui sont utilisés dans la conception et la réalisation de l'organisation de la structure de l'entreprise, les processus métiers, les systèmes d'information et l'infrastructure."* [80].
L'EA est une approche qui vise à gérer des systèmes complexes au sein de l'organisation et de collaborer de la manière la plus efficace pour l'alignement entre l'activité métier de l'entreprise et ses systèmes techniques [62].
En outre, elle consiste à séparer les préoccupations des parties prénantes dans l'elaboration d'un système, et à spécifier plusieurs points de vue[155].
En d'autres termes, l'EA définit comment les systèmes peuvent être utilisés pour répondre aux besoins de l'entreprise de manière plus collaborative[148].
Une vue est définie comme un "*produit de travail exprimant l'architecture d'un*

système du point de vue des préoccupations spécifiques du système".
Dans (ISO/IEC 2007), **un point de vue** est défini comme un *"produit de travail établissant les conventions pour la construction, l'interprétation et l'utilisation de vues d'architecture pour encadrer des problèmes de systèmes spécifiques"*.
Une préoccupation est un « *intérêt pour un système pertinent pour une ou plusieurs parties prenantes* ».
Une partie prenante est un « *individu, une équipe, une organisation ou des catégories de celles-ci, ayant un intérêt dans un système* ».

Les avantages de l'architecture d'entreprise [31] [80][79] ont été examinés et classés dans [122] en avantage stratégique, avantage indirect, avantage métier.
Les avantages stratégiques visent à augmenter la perspicacité et la vue d'ensemble nécessaires pour aligner avec succès l'entreprise et les plateformes technologiques, à donner une meilleure gestion du changement, à assurer la stabilité et l'agilité de l'entreprise, à promouvoir une compréhension commune dans l'ensemble de l'entreprise.
Les avantages indirects consistent : à comprendre la richesse des relations entre une entreprise et ses clients et d'autres partenaires, à donner une meilleure gestion des risques, à la gestion de la complexité du système.
Les avantages immatériels consistent : à fournir une vision globale de l'entreprise, à apporter une aide à la décision.
Les avantages métiers : visent à l'amélioration de l'interopérabilité et de l'intégration des systèmes constituant l'entreprise, à réduire les coûts, à raccourcir le temps de cycle de développement.
Par ailleurs Shah et al.[147], décrivent les bénéfices de l'architecture d'entreprise liés au métier et au SI, résumés dans le tableau 1.1.

Dans [32], les auteurs relèvent que selon IFAC-IFIP Task Force [52] et ISO 15704 il existe deux types d'architectures d'entreprise : Les architectures système (aussi appelées "Type 1") qui traitent de la conception d'un système, et celles des projets de référence d'entreprise (également appelées "Type 2") qui traitent de l'organisation du développement et de la mise en œuvre d'un projet tel qu'une intégration d'entreprise ou un autre programme de développement d'entreprise. Les architectures représentant un système (par exemple un système du Système

Bénéfice	Description
Relatif au SI	
Gestion de la complexité	Faciliter le cadrage et la coordination des programmes projets de gestion et de systèmes d'information. Gérer la complexité et décrire les interdépendances de manière exploitable.
Supervision des ressources techniques	Identifiez et supprimez la redondance
Gestion de la connaissance	Gérer et partager les connaissances de manière modulaire afin qu'elles puisse
Visibilité du SI (IT)	Les ressources et les systèmes informatiques (IT) sont davantage alignés sur s
Relatif au métier	
Réduction de l'impact de la rotation du personnel	Capturez les connaissances des employés et des consultants. Fournir des solu
Adaptabilité plus rapide	Faciliter l'acquisition des connaissances nécessaires pour changer les système
Amélioration des procédures opérationnelles	Comprendre et modéliser les processus métiers. Réviser et réorganiser
La prise de décision	Représenter les couches et les composants d'une entreprise de manière modu

Tableau 1.1 – Bénéfices d'architecture d'entreprise (extrait de [147])

d'Information d'une entreprise) en termes de structure et de comportement[157]. Les architectures de type (2) sont en fait des frameworks qui visent à structurer les concepts et les activités/tâches nécessaires pour concevoir et construire un système d'entreprise. La conception du système (Type 1) doit être cohérente avec celle des autres systèmes de l'entreprise et surtout être alignée sur la stratégie de l'entreprise (Type 2).
Selon Jonkers et al. [79], une approche globale pour l'architecture d'entreprise se compose d'un cadre (Framework) et d'un language de modélisation.
Un Framework ou cadre d'Architecture d'Entreprise est une structure exprimée en termes de diagrammes, texte et règles formelles qui relient les composants d'une entité conceptuelle les uns aux autres [147]. Un Framework fournit les différents points de vue existants sur une architecture, et subdivise l'architecture dans différents domaines, parfois complétée par les relations entre ces domaines[155]. L'objectif principal du framework est de fournir un mécanisme d'organisation afin que les concepts, les problèmes et les connaissances sur l'interopérabilité des entreprises (à la fois inter et intra) puissent être représentés de manière plus structurés[148][97].
Comme exemples de frameworks on peut citer TOGAF, le Framework de Zachman, l'Architecture d'Entreprise de NIST (National Institute of Standards and Technology), le cadre fédéral d'architecture d'entreprise (FEAF – Federal Enterprise Architecture Framework).
Un langage de modélisation définit les concepts pour décrire une architecture[79]. Cela peut être à la fois en langage naturel ou graphique. Les exemples incluent UML[127] et ArchiMate[72].
Une approche globale, où la sécurité et l'architecture d'entreprise sont intégrées, se compose également d'un cadre et d'un langage de modélisation[106][79][78].
Dans ce qui suit nous présentons quelques frameworks d'EA et quelques langages de modélisation des plus utilisés dans l'EA.

1.2.2 Cadres (Framework) d'architecture d'entreprise

En raison de la demande croissante de gestion de systèmes complexes et en utilisant ces systèmes pour collaborer de la manière la plus efficace, le terme «*ar-*

chitecture d'entreprise» a émergé[18][31]. Dans les années récentes, la technologie de l'information a changé les métiers, mais en de nombreux cas, ce changement n'est pas aligné avec la stratégie métier de l'organisation[34]. Cela a influencé les organisations. L'architecture d'entreprise fournit la structure et le contrôle requis pour aligner les opérations d'une entreprise et les technologies de l'information pour soutenir ses objectifs et ses stratégies de l'entreprise [166]. Plusieurs cadres d'EA ont vu le jour.

L'un de ces cadres est le Cadre d'architecture d'entreprise Oracle (**OEAF : Oracle Enterprise Architecture Framework**)[166]. Son objectif principal est de pouvoir travailler en collaboration avec les clients d'Oracle pour développer des feuilles de route stratégiques qui permettent l'alignement entre le métier et les technologies de l'information. Le cadre Oracle est connu pour être un cadre hybride et il est principalement influencé par TOGAF, FEAF et Gartner [166].

Le Cadre Fédérale d'Architecture d'Entreprise : FEAF[123] est un autre exemple d'EA. Publié en mai 2012 dans le cadre de la politique fédérale américaine des directeurs de système d'information. Il permet d'accroître la pratique de l'architecture d'entreprise dans le Gouvernement fédéral américain en définissantt plusieurs principes d'utilisation de l'architecture d'entreprise. Ces principes sont une aide pour les entités fédérales au sein du gouvernement américain qui tirent le meilleur parti de l'utilisation du cadre en éliminant les doublons et en augmentant les ressources partagées[123].

Le Cadre Architectural du Ministère de la Défense : MoDAF[65] est un autre cadre d'évaluation environnementale. Ce cadre a été défini à l'origine par le ministère de la défense du Royaume-Uni de la Défense pour structurer l'intégration des systèmes au sein du ministère. Il a ensuite été modifié pour aider à acquérir les informations nécessaires sur les ressources et les processus de l'entreprise pour accomplir la stratégie prévue de l'organisation [65].

Le Cadre Architectural du Ministère de la Défense des États-Unis : DoDAF est un autre cadre qui a été initialement développé pour être applicable aux systèmes de défense par le Département américain de la Défense [153]. C'est l'original du cadre architectural précédemment développé de commandement, de contrôle, de communication, d'informatique et de renseignement, de surveillance et de reconnaissance connu sous le nom C4ISR [3]. Il organise des

architectures basées sur quatre vues : La vue d'ensemble, la vue opérationnelle, la vue système et la vue technique [153].
Un autre cadre d'EA est **Gartner**. Ce Framework considère l'EA comme une discipline qui devrait toujours être descendante. Par conséquent, pour la consolidation d'une EA, les métiers devraient venir en premier, ensuite l'information et la technologie. Une des stratégies clés de Gartner est de développer l'état futur de l'architecture avant que l'actuelle ne soit documentée. Cette étape est suivie d'autres résultats, y compris l'écart d'analyse et une feuille de route exploitable.
John Zachman définit son cadre d'EA comme une "*structure logique pour classer et organiser les représentations descriptives d'une entreprise qui sont importantes pour la gestion de l'organisation, ainsi que le développement des systèmes de l'entreprise*" [179]. Ainsi, le cadre de Zachman fournit une structure logique pour organiser la conception de l'entreprise, ses artefacts, qui peuvent aider les gestionnaires de l'entreprise dans le processus décisionnel [179].
Le cadre d'architecture de groupe ouvert (TOGAF : The Open Group Architecture Framework) a pour principal objectif de fournir aux organisations une méthodologie qui leur permet d'améliorer leur efficacité métier [62]. Cette amélioration peut être obtenue en utilisant les ressources de manière efficiente et efficace pour avoir un impact plus important sur le retour sur investissement. TOGAF fournit non seulement la mise en œuvre et la convivialité d'utilisation, mais fournit également un bon alignement entre le SI (Système d'Information) et le métier.

En résumé, on peut dire que tous les cadres d'Architecture d'Entreprise existants partagent le même objectif, qui est de créer une architecture d'entreprise qui maximise l'alignement du métier et du SI. Cet objectif vise finalement à réduire la complexité du système et partager des ressources au sein de l'organisation. Les études comparatives [176] [90][27] ont mis la lumière sur le fait que chacun des cadres est toujours activement présenté comme une forme de «meilleure pratique» par ses vendeurs : le cadre de Zachman a "une importance profonde dans la définition de l'architecture d'entreprise", FEAF et DoDAF sont "prouvés pour avoir une applicabilité immédiate". Cependant la norme largement adoptée dans le développement d'architecture d'entreprise est

le TOGAF[90][27]. A travers TOGAF ADM, sa méthode de développement de l'architecture, conçue par l'Open Modeling Group (OMG)[60], TOGAF est le standard de facto dans l'industrie car il fournit une approche structurée pour la conception d'architectures d'entreprise. Puisqu'il est également un standard ouvert, il est bien adapté aux fins de la méthode.
Ainsi, TOGAF semble identifier un processus clair de développement pour toute organisation, ce qui justifie le choix de TOGAF dans le cadre de notre thèse.

1.2.3 Le cadre TOGAF

TOGAF est une méthode détaillée et un ensemble de supports d'outils de développement d'architecture d'entreprise[62].
Dans sa version 9.2 [62], TOGAF est d'abord plus orienté métier. La définition du SI dans TOGAF 9.2 est la gestion du cycle de vie des informations et de la technologie au sein d'une organisation.
Au cœur de TOGAF se trouve la méthode de développement d'architecture (ADM). ADM fournit un processus itératif d'architecture continue de développement. Ce processus est un guidage étape par étape pour la création d'une architecture d'entreprise. Comme l'illustre la figure **??**, l'ADM est continuellement guidé par un processus de «**gestion des exigences**» au centre du diagramme. De par sa nature même, l'architecture traite de l'incertitude et du changement. Il est donc crucial que les exigences et leurs modifications soient identifiées, stockées et introduites dans les phases ADM pertinentes, par un processus dynamique plutôt que par l'utilisation d'un ensemble statique d'exigences. ADM (cf. Figure **??**) se compose de huit phases :

Graphique supprimé pour respecter le droit d'auteur
FIGURE 1.1 – TOGAF ADM cycle[62]

— **La phase préliminaire** décrit les activités de préparation et d'initiation requises afin de répondre à la directive métier pour une nouvelle architecture d'entreprise.
— **Phase A : Vision de l'Architecture** comprend des informations concernant la définition de la portée, l'identification des parties prenantes, la

création de la vision de l'architecture et l'obtention de l'approbation.

- **Phase B : Architecture Métier** a pour objectif, le développement d'une architecture métier cible, décrivant la stratégie les produits/services de l'organisation, les aspects fonctionnels, les processus métiers, sur la base des principes métiers, des objectifs métiers et des moteurs stratégiques.
- **Phase C : Architectures des Systèmes d'Information** se concentre sur l'identification et la définition des considérations relatives aux applications et aux données qui prennent en charge l'architecture métier d'une entreprise. Il comporte en fait deux sous-phases :
  - **Architecture de Données** a pour objectif la définition des principaux types et sources de données nécessaires pour soutenir l'entreprise. Cela doit être fait d'une manière compréhensible, complète, cohérente et stable par les parties prenantes.
  - – **L'architecture d'application** a pour objectif la définition des principaux types de systèmes d'application nécessaires pour traiter les données et soutenir l'entreprise.
- **Phase D : L'Architecture Technologique** cherche à cartographier les composants d'application définis dans la phase d'architecture d'application en un ensemble de composants technologiques. Ceux-ci représentent des composants logiciels et matériels, disponibles sur le marché ou configurés au sein de l'organisation en plateformes technologiques.
- **Phase E : Opportunités et Solutions** éffectue la planification initiale de la mise en œuvre et l'identification des moyens de livraison pour l'architecture définie au précédentes phases.
- **Phase F : Planification de la Migration** aborde la formulation d'un ensemble de séquences détaillées d'architectures de transition accompagnées d'un plan de mise en œuvre et de migration.
- **Phase G : la Gouvernance de la Mise en œuvre** fournit une supervision architecturale de la mise en œuvre.
- **Phase H :Gestion du Changement Architecturale** établit des procédures de gestion du changement apporté à la nouvelle architecture.
- **Gestion des Exigences** : examine le processus de gestion des exigences d'architecture dans tout l'ADM. Il est au cœur de l'ADM et montre com-

ment les exigences doivent être tracées tout au long du processus.
TOGAF ADM est itératif, sur l'ensemble du processus, entre les phases et au sein des phases. Cela permet l'évolution des architectures d'entreprise modélisées avec TOGAF ADM.

En outre TOGAF permet l'extension de son cadre d'architecture d'entreprise à des notions telles que la sécurité. Cette approche est désignée par l'architecture de sécurité d'entreprise[61], qui est traitée dans la section suivante.

1.2.4 Architecture de sécurité d'entreprise

Une architecture de sécurité est "*une structure organisationnelle, conceptuelle, logique et physique de composants qui interagissent de manière cohérente afin d'atteindre et de maintenir un état de risque géré. C'est un catalyseur/moteur d'un comportement sécurisé, sûr, résilient, fiable et du respect de la vie privée dans les zones à risque de toute l'entreprise*"[61].
L'architecture de sécurité d'entreprise cherche à aligner les mesures de sécurité sur les objectifs métiers (business). Il le fait en définissant les relations entre les composants sur les différents niveaux d'architecture, assurant ainsi la traçabilité et la justification.
Les architectures de sécurité ont un certain nombre de caractéristiques qui doivent être prises en compte par les architectes de sécurité. Ces caractéristiques comprennent la définition d'une méthodologie de sécurité, la composition de vues et de points de vue et l'adressage des flux d'information au sein des systèmes d'architecture [155]. Un cadre de sécurité d'entreprise pose des défis importants et il y a beaucoup de recherches qui soulignent l'importance et les avantages d'une approche holistique[136]. Nous décrivons quelques travaux proposant un cadre de sécurité de l'architecture d'entreprise.
Sommestad et al. [163] ont présenté un cadre d'évaluation de la sécurité utilisant les diagrammes d'influence étendus, basés sur des statistiques bayésiennes pour combiner des graphes d'attaque avec des contre-mesures dans les graphes de défense. Ekstedt et Sommestad ont présenté une approche basée sur un modèle d'EA pour la gestion de la cybersécurité [42]. Franke et al. ont montré comment les cadres d'EA pour l'analyse de dépendance peuvent être étendus dans le

domaine des méthodes quantitatives par l'utilisation de l'analyse arborescente de la faute (FTA) et de réseaux bayésiens (BN) [53]. Zambon et al. ont présenté un modèle et une approche de dépendance temporelle qualitative (QualTD) et pour réaliser l'analyse qualitative d'évaluation des risques de disponibilité dans les architectures de SI [180]. Sommestad et al. ont présenté un outil d'analyse appelé "langage de modélisation de la cybersécurité" (CySeMoL) pour prendre en charge les gestionnaires de la sécurité des systèmes d'entreprise dans l'analyse de la sécurité [162].
Biggs et al. [23] ont décrit une approche basée sur SysML pour la modélisation des préoccupations liées à la sécurité d'un système. Pilipchuk et al.[134] ont suggéré une approche pour dériver les exigences de contrôle d'accès des processus métiers et tester la conformité des logiciels de conception par l'analyse de flux de données, pour répondre à la sécurité et à la confidentialité des exigences dans les organisations à travers les processus métiers.
Barateiro et al. [15] proposent un alignement entre la gestion des risques, la gouvernance et les activités d'architecture d'entreprise (ces concepts se limitant à la cartographie du SI) afin de fournir un soutien systématique pour cartographier et tracer les risques identifiés pour les artefacts modélisés dans une EA. Le document propose un cadre de gestion de risque, y compris un langage spécifique au domaine basé sur XML pour la gestion des risques (Risk-DL) et explique le lien avec la norme ISO 31000 (ISO 31000 :2009).
Dans[73], Innerhofer-Oberperfler et Breu proposent une approche pour l'évaluation et l'analyse systématiques des risques liés au SI dans les organisations et les projets. L'approche est basée sur un modèle utilisant une architecture d'entreprise comme base du processus de gestion de la sécurité. Les auteurs fournissent une description intégrée de la structure, des processus et du paysage informatique sous-jacent d'une organisation à partir des couches métier, applicative, technique et physique. Cela permet d'établir un pont entre les points de vue techniques et métiers sur la sécurité de l'information. La proposition fournit un méta modèle de sécurité de l'information et un processus détaillé de gestion de la sécurité et définit les responsabilités nécessaires et les rôles des parties prenantes participantes.
De la même manière, Ertaul et Sudarsanam [47] proposent d'exploiter le cadre de

Zachman [179] pour définir et concevoir des outils pour sécuriser une entreprise. Cela aide à prendre en charge la planification de la sécurité, en particulier pour le SI.
L'architecture de sécurité d'entreprise appliquée Sherwood (SABSA : Sherwood Applied Business Security Architecture) [149] est une méthodologie pour développer l'architecture d'entreprise axée sur le risque de sécurité de l'information et d'assurance de l'information et pour fournir des solutions d'infrastructure de sécurité qui soutiennent les initiatives métiers dans un contexte d'activités critiques. La méthodologie repose sur le modèle SABSA, qui est basé sur le framework de Zachman, quelque peu adapté à une vision de la sécurité.
Le guide Open Enterprise Security Architecture [130] est un guide donnant un aperçu complet des principaux problèmes de sécurité, principes, composants et concepts sous-jacents aux décisions architecturales. Il fournit un cadre qui sert de référence commune pour décrire l'architecture de sécurité d'entreprise et la technologie de sécurité.
Dans[58], Gandry et al. ont proposé une intégration de la gestion de sécurité des risques et de l'architecture d'entreprise. L'intégration est sous la forme d'une cartographie des concepts entre la gestion des risques de sécurité du système d'information (Information System Security and Risk Management : ISSRM) et la gestion de l'architecture d'entreprise (EAM : Enterprise Architecture Modelling). L'approche s'appuie sur la modélisation de l'architecture d'entreprise pour prendre en charge l'identification des actifs métiers et SI. Il propose également de modéliser le traitement des risques, notamment en relation avec la valeur du risque. Cependant, cette approche n'apporte pas un réel soutien dans l'identification des menaces et des vulnérabilités associés aux éléments de l'architecture.
Même si TOGAF considère la sécurité séparément, il indique que la sécurité doit être abordée à toutes les phases de l'architecture d'entreprise[130]. Par conséquent, il a fourni aux architectes de sécurité une méthodologie à cet effet qui peut présenter des conseils avec un processus étape par étape pour développer des architectures de sécurité d'entreprise. Cette méthodologie se compose d'un certain nombre de politiques et de principes qui doivent être abordés dans chaque phase de la méthode de développement d'architecture (ADM)[62]. Ils sont liés aux propriétés générales de sécurité : authentification, autorisation,

audit, assurance, disponibilité, protection des actifs, administration et gestion des risques.
Dans [176] [90][27][109], les auteurs ont effectué un examen de cadres de sécurité pour déterminer si une méthodologie de sécurité complète et holistique fournirait mieux aux organisations des avantages en matière de sécurité. Ils ont déterminé des critères de comparaison tels que : le soutien de la vision de l'organisation, le domaine supporté, le langage de modélisation, la disponibilité de l'information, l'utilisation de norme internationalement reconnue, le développement du cadre basé sur EA, le développement d'un cadre d'architecture de sécurité d'entreprise comme priorité pour l'ensemble de l'organisation. Il ressort qu'il y a très peu de cadres tels que (TOGAF, SABSA, NIST)[109] [90][27] qui satisfont plusieurs critères, mais aucun cadre ne couvre les critères de test holistique.
Le besoin de developper une approche holistique de sécurité d'EA a conduit à l'adoption d'approches hybrides, combinant plusieurs approches de cadre de sécurité[90]. Certains des facteurs qui motivent les organisations à adopter cette approche sont : l'interopérabilité / la flexibilité, le meilleur ajustement du modèle, aide à l'alignement métier/SI, et la conformité aux normes de l'industrie EA.
Dans ce qui suit nous étudions comment TOGAF peut être combiné à SABSA pour une architecture de sécurité d'entreprise.

1.2.5 Cadre d'architecture de sécurité d'entreprise : TOGAF ADM et SABSA

Dans son livre blanc [61] lié à la sécurité de TOGAF, The Open Group, en collaboration avec SABSA Institute [140], adresse la question de l'intégration des notions de sécurité et de risque dans l'Architecture d'Entreprise. Le document fournit un guidage pour les praticiens de la sécurité et aux architectes d'entreprise souhaitant utiliser le standard TOGAF pour développer une Architecture d'Enterprise. Un aperçu complet de tous les artefacts de sécurités sélectionnés est donné dans la figure 1.2.

Graphique supprimé pour respecter le droit d'auteur

FIGURE 1.2 – Concepts essentiels de sécurité et de risque et leur position dans le TOGAF ADM[130]

Le TOGAF ADM contient le concept d'«artefacts » (produits du travail) qui sont utilisés ou produits par chaque phase. Les concepts de base de l'architecture de sécurité d'entreprise sont exprimés dans la terminologie TOGAF et liés aux concepts TOGAF, ce qui garantira une bonne intégration des concepts de risque et de sécurité pertinents aux phases ADM appropriées.
Nous présentons dans la section suivante, les caractéristiques de SABSA et son lien avec TOGAF.

Cadre de sécurité d'entreprise SABSA

L'approche de développement d'une architecture de sécurité d'entreprise proposée par SABSA est basée sur un modèle à six couches. Ce modèle est utilisé comme base d'un processus de développement d'architecture [140]. En suivant l'évolution de l'architecture d'entreprise en fonction des couches de modèle, la méthodologie devient quelque peu évidente.
Le modèle comprend six couches et suit de près le travail effectué par John A. Zachman dans l'élaboration d'un modèle d'architecture d'entreprise, bien qu'il ait été quelque peu adapté à une vision sécuritaire du monde. Chaque couche représente la vue d'un acteur (vue du métier, vue de l'architecte, vue du concepteur, vue du constructeur, vue du technicien, vue du manager) dans le processus de spécification, de conception, de construction et d'utilisation du bâtiment[140].
La matrice SABSA (cadre) est une structure très similaire au cadre Zachman, mais spécifiquement axée sur les aspects de risque et de sécurité[149]. Cette matrice définit des points de vue pertinents pour la modélisation des risques et de la sécurité, de manière systématique.
L'interprétation détaillée de chacune des six couches horizontales de la matrice SABSA s'effectue à travers l'analyse verticale de chaque couche horizontale en appliquant le six questions cruciales : Quoi ? Pourquoi ? Comment ? Qui ? Où ?

Quand ?[148][149].

Graphique supprimé pour respecter le droit d'auteur
FIGURE 1.3 – Matrice SABSA [149]

(What) Quoi ? Qu'essayez-vous de faire ? – Les actifs à protéger par votre architecture de sécurité ;
(Why) Pourquoi ? Pourquoi le faites-vous ? – La motivation à vouloir appliquer la sécurité, exprimée dans les termes de cette couche ;
(How) Comment ? Comment essayez-vous de le faire ? – Les fonctions nécessaires pour assurer la sécurité à cette couche ;
(Who) Quoi ? Qui est impliqué ? – Les aspects humains et organisationnels de la sécurité à cette couche ;
(Where) Où ? Où le faites-vous ? – Les emplacements où vous appliquez votre sécurité, pertinents pour cette couche ;
(When) Quand ? Quand le faites-vous ? – Les aspects temporels de la sécurité relatifs à cette couche.
Cette matrice donne un ensemble de 6 x 6 cellules, qui représentent l'ensemble du modèle pour l'architecture de sécurité d'entreprise. Le processus du développement d'une architecture de sécurité d'entreprise est un processus continu, consistant à remplir toutes ces 36 cellules[149].
SABSA ne définit pas de concepts de modélisation spécifiques, mais la matrice décrit une grande variété d'aspects qu'un modélisateur devrait être capable d'exprimer.
Par ailleurs, SABSA propose un cycle de développement de manière à garantir la sécurité de l'information d'entreprise à travers un processus continu que nous présentons dans la section suivante.

Développement d'architecture de sécurité SABSA

Le cycle de vie de SABSA[149][140] est proche du cycle planifier-faire-vérifier-agir (PDCA : Plan-Do-Check-Act). Sa particularité est qu'il s'intègre au modèle TOGAF par une correspondance avec ses différentes phases comme décrit dans la figure 1.4.

Graphique supprimé pour respecter le droit d'auteur
FIGURE 1.4 – Relation entre le cycle de SABSA et TOGAF ADM [140]

Les activités menées dans la phase de stratégie et de planification du cycle de vie de SABSA sont considéré comme appartenant à la même catégorie que la phase préliminaire et l'architecture de TOGAF ADM vision. Dans ces phases TOGAF, le cadre et les principes sont choisis et une stratégie ou la vision se développe.
Pour la conception de l'architecture, TOGAF distingue trois niveaux : métier, système d'information et architecture technologique. Dans cette phase, l'architecture est conçue et alignée sur la stratégie/la vision choisie.
Dans la phase de mise en œuvre, l'architecture développée est implémentée dans l'organisation. TOGAF ADM distingue deux phases : Opportunités et Solutions, et Planification des migrations.
Enfin, l'évaluation de la réussite doit être complétée afin de déterminer si ou non, la mise en œuvre a réussi. Les résultats doivent être mesurés par rapport aux stratégies et vision initiales. Ceci est exécuté dans la phase de gouvernance de la mise en œuvre. Cette étape pourrait conduire à de nouvelles idées pour changer la vision ou à de nouveaux changements dans l'architecture et une nouvelle itération du cycle pourrait être lancée. Dans la dernière étape, la gestion du changement d'architecture, les changements internes et externes de l'architecture, de l'organisation ou de son environnement sont observés. Sur la base de cette observation, une nouvelle itération de la méthode du développement de l'architecture peut être effectuée.
La conception d'une architecture commence par la compréhension du métier de l'entreprise, des opérations qu'elle mène, et définit ses moteurs et attributs métiers spécifiques[58][73][149].Les moteurs sont déterminés par les stratégies de l'organisation, les plans opérationnels et les éléments clés à son succès[149].
Le profil d'attribut métier (Business Attribute Profiling) SABSA est au cœur de sa méthodologie [149]. C'est sa technique d'ingénierie des exigences, qui fait la particularité de SABSA et fournit le lien entre les exigences de l'entreprise et la conception de la technologie/des processus. Elle traduit les objectifs et les moteurs (drivers) de l'entreprise en exigences en utilisant une approche basée

sur les risques.
Dans la méthodologie SABSA, ces vertus sont appelées attributs métiers "Business Attribute"[149]. Chaque attribut métier est un condensé d'expérience pratique de l'évaluation des risques métiers et de la classification de ces risques sous des rubriques utiles. La figure 1.5 décrit la taxonomie (non exhaustive) de (85) de ces attributs identifiés.

Graphique supprimé pour respecter le droit d'auteur
FIGURE 1.5 – Taxonomie d'attributs métiers SABSA [149]

SABSA présente une définition générique détaillée et quelques lignes directrices suggérées pour appliquer des mesures à ces attributs[149], que nous n'aborderons dans cette thèse.
Un profil d'attribut d'entreprise est construit par les architectes, en utilisant la taxonomie comme ligne directrice. L'objectif est de documenter les attributs pertinents pour le cas métier en question, redéfinissant chaque attribut sélectionné en termes de cas métier. Ensuite, développer une approche de mesures spécifiques encore une fois liés à l'analyse du cas métier. Le modèle est flexible et adaptatif. Si nécessaire, de nouveaux attributs et de nouvelles définitions devraient être ajoutés pour répondre aux exigences opérationnelles.

En guise de synthèse, nous notons que SABSA et TOGAF constituent des approches complémentaires pour la définition d'un cadre architecture de sécurité [109]. Les deux approches présentent un alignement à travers le cycle de développement SABSA avec TOGAF ADM. En plus la taxonomie d'attributs métiers SABSA est une base pour pour la définition de la phase des exigences métiers et de sécurité de TOGAF. TOGAF propose un cadre d'ingénierie des exigences métier, également connu sous le nom de profilage d'attributs "Attributes Profiling"[130]. AInsi TOGAF et SABSA constituent un duo de cadres et de méthodologie d'architecture d'entreprise dans notre thèse.
La section suivante discutera des langages de modélisation d'EA.

1.2.6 Langage de modélisation d'architecture d'entreprise

Les langages les plus connus dans le domaine de l'architecture d'entreprise sont Archimate et UML[78].
ArchiMate [72] est de plus en plus adopté comme la norme de facto pour la spécification de modèles et de vues d'architecture d'entreprise. On distingue trois couches : la couche métier, la couche application et la couche infrastructure. De plus, le langage prend en compte les aspects structurels, comportementaux et informationnels au sein de chaque couche. En outre, archimate est une norme ouverte qui fournit les constructions de modélisation pour décrire et interconnecter les architectures métiers et techniques[14]. L'application du langage ArchiMate pour représenter les concepts de risque et de sécurité est le véhicule idéal pour considérer ces aspects de manière intégrale. Le langage ArchiMate s'intègre parfaitement aux autres cadres et normes d'architecture d'entreprise, tels que le standard TOGAF et le cadre Zachman, ainsi qu'aux cadres de gestion de la sécurité d'entreprise tels que SABSA[14].
UML[63], dans la version 2.0, est le langage de modélisation du standard OMG (Object Management Group) le plus utilisé dans l'industrie du logiciel. UML permet aux utilisateurs de modéliser le processus métier, la structure de l'application, le comportement de l'application, la structure des données et l'architecture. C'est un langage visuel pour spécifier, construire et documenter les artefacts des systèmes. Il s'agit d'un langage de modélisation à usage général qui peut être utilisé avec toutes les principales méthodes d'objets et de composants, et qui peut être appliqué à tous les domaines d'application (santé, finance, télécommunications, aérospatiale, ...) ainsi que sur plusieurs plateformes de mise en œuvre [147].
Plusieurs initiatives d'extension d'UML pour les besoins de sécurité d'entreprise [83][100][67][42] ont vus le jour. L'inconvénient de ces langages de modélisation de sécurité est qu'ils ont une portée limitée où le développement du système ne couvre pas toutes les disciplines d'architecture d'entreprise [109]. Par ailleurs, une difficulté majeure de la mise en œuvre des politiques (exigences) de sécurité par ces approches, réside dans le fait que ces politiques sont exprimées à un niveau d'abstraction élevé. Pour ce faire l'architecture d'entreprise a besoin

d'être abordée sous une approche de modèle[42]. En effet, un modèle transforme une partie du monde réel, par exemple une organisation, en modèle abstrait en simplifiant sa représentation [42] [68] [94].
L'une des disciplines modernes qui traite des modèles dans le domaine du developpement système est l'Inégnierie Dirigée par les Modèles (IDM)[88], ou en anglais Model Driven Engineering (MDE). L'IDM vise à fournir un grand nombre de modèles pour exprimer séparément chacune des préoccupations des utilisateurs, des concepteurs, des architectes à travers son approche architecturale MDA [60]. MDA permet de représenter le système à travers trois niveaux d'abstraction (CIM : niveau recueil de besoins, PIM : niveau de conception, PSM : lié à la technologie d'implémentation).
Dans le contexte du processus métier d'une entreprise, UML, ainsi que Meta Object Facility (MOF), fournissent également une base pour l'architecture dirigée par les modèles, qui unifie les étapes de développement et d'intégration de la modélisation d'entreprise, en passant par la modélisation architecturale et applicative, jusqu'au développement, au déploiement, la maintenance et l'évolution [53].
En définitive, l'IDM semble constituer une solution adaptée pour la modélisation de l'architecture de sécurité d'entreprise, notamment à travers ses normes MDA et UML.

1.2.7 Synthèse et discussion

Le développement d'un système informatique se traduit à travers la conception de son architecture, qui le représente, destinée à être implémentée. L'architecture représente un système (par exemple un système du Système d'Information d'une entreprise) en termes de structures et de comportement. L'architecture d'entreprise traduit mieux ce concept à travers l'interaction du système avec les différentes structures de l'entreprise. L'architecture d'entreprise utilise la notion de Framework qui vise à structurer les concepts et les activités/tâches nécessaires pour concevoir et construire un système d'entreprise.
TOGAF est le Framework standard de facto dans l'industrie de développement d'architecture d'entreprise. TOGAF ADM est sa méthode de développement

de l'architecture, conçue par l'Open Group, qui permet de mieux aligner les systèmes d'information et les services avec les objectifs opérationnels d'une organisation. TOGAF est donc un cadre idéal pour le développement d'une EA aussi bien pour la prise en compte des aspects de sécurité. Cependant, la description des structures de sécurité d'entreprise ne sont pas décrites dans toutes les phases de TOGAF, d'où le besoin de le combiner avec un autre cadre.
SABSA est l'une des tentatives les plus remarquables de cadre d'architecture de sécurité. SABSA propose un cadre et une méthodologie de manière à garantir la sécurité de l'information d'entreprise à travers un processus continu. Il présente un alignement à travers son cycle de développement avec TOGAF. En plus, SABSA inclut sa propre méthode spécifique pour réaliser l'ingénierie des exigences qui s'aligne avec la phase des exigences de TOGAF [28][148]. Ainsi TOGAF ADM et SABSA constitue un duo de cadres et de méthodologie d'architecture de sécurité d'entreprise choisi dans notre thèse.
Les langages de modélisation de l'EA les plus connus sont Archimate et UML. Des initiatives d'intégration de la sécurité dans le système basées sur ces langages ont vu le jour avec des points forts tels que la prise en compte des besoins de sécurité des les phases précoces du développement (niveau métier). Cependant le gap lors du passage des exigences de sécurité (dû à leur nature abstraite) du métier aux niveaux ultérieures constitue une des limites de ces méthodes. Comme solution, nous nous proposons d'explorer comment l'approche basée sur les modèles, notamment l'IDM avec ses outils MDA et UML, peut être une alternative à ce problème.
Dans la section suivante, nous abordons la notion de l'IDM et son alignement avec TOGAF.

1.3 Ingénierie Dirigée par les Modèles (IDM) et Architecture d'Entreprise

L'ingénierie dirigée par les modèles (IDM)[117] vise à fournir un grand nombre de modèles pour exprimer séparément chacune des préoccupations des utilisateurs, des concepteurs, des architectes, etc. C'est par ce principe de

base fondamentalement différent que l'IDM peut être considérée en rupture par rapport aux travaux de l'approche objet. Dans l'IDM, le point clé (crucial) est l'utilisation des modèles comme des entités primaires afin de les traiter de manière automatique ou (semi)automatique. Ces modèles « formels », sont des représentations abstraites d'une réalité (préoccupation). Cette formalisation est atteinte grâce à la définition des méta-modèles. Fort de ces caractéristiques et avantages, nous allons explorer dans nos travaux en quoi l'IDM peut être un outil pour le développement de système dans un cadre d'EA, représentée sous forme de méta modèle. Pour cela, dans cette partie, nous définissons les concepts clefs de l'IDM tels que les modèles, les méta-modèles les méta-méta-modèles ainsi que les relations qui existent entre ces concepts. Ensuite nous présentons quelques standards de l'OMG (Object Management Group) liés à l'IDM tels que MDA, MOF et la notion de transformations de modèles avec quelques langages de transformation. Nous examinons ensuite comment MDA peut s'appliquer dans le domaine de l'Architecture d'Entreprise avec une analyse d'alignement avec les approches du domaine tel que le cadre TOGAF.

1.3.1 IDM : définition, principes et caractéristiques

L'Ingénierie Dirigée par les Modèles (IDM)[19][117][77], ou en anglais MDE (Model Driven Engineering) , est la discipline qui place les modèles au centre des processus d'ingénierie logicielle. Elle permet de simplifier les représentations et niveaux d'abstraction d'un système pour résoudre les problèmes complexes du monde réel.
L'IDM est une ingénierie générative dans laquelle tout ou partie d'une application est générée par les modèles, contrairement à l'approche traditionnelle de développement ou le code final exécutable est considéré comme l'artefact central. Elle offre un cadre référentiel pour l'étude de différentes problématiques, permettant à différents acteurs de s'intéresser aux divers aspects du système (sécurité, performance, qualité...)[76].
L'artefact central de l'IDM est le modèle.

Modèle : "*un modèle est défini comme une représentation (d'une partie) d'un*

système construit pour un objectif précis. Le modèle doit répondre aux questions que l'utilisateur se pose sur le système qu'il représente" [22].
Un modèle est une simplification d'un processus que l'on souhaite capturer ou automatiser [101]. La simplification est telle qu'elle ne tient pas compte des détails qui peuvent être supervisés à un stade donné du cycle d'ingénierie logicielle. Le but est de se concentrer sur les concepts pertinents à portée de main. On déduit de cette définition, dans [20][145][11], une première relation majeure de l'IDM, entre le modèle et le système qu'il représente, appelée «**représentationDe**». Un modèle est une instance du méta-modèle qui le décrit.

Méta-modèle : c'est *un modèle qui définit le langage d'expression d'un modèle* [128], c.-à-d. le langage de modélisation. La notion de méta-modèle conduit à l'identification d'une deuxième relation liant le modèle et le langage utilisé pour le construire appelé «**conformeA**». Il faut noter que dans l'IDM, tout est modèle. Même un méta-modèle est un modèle, et doit donc être conforme à son méta-modèle. On dit qu'un modèle est conforme à un méta modèle si chacun de ses éléments est conforme à un élément (objet, relation) du méta modèle, et s'il respecte l'ensemble de ses propriétés.

Un langage de modélisation est un langage de spécification, défini généralement par une syntaxe et une sémantique pour l'expression de modèles. Il peut être graphique ou textuel. Un langage graphique tel qu'UML (Unified Modeling Language) [127] utilise des diagrammes pour la représentation des concepts et les relations entre eux. Un langage textuel utilise des mots clefs réservés associés à des paramètres par exemple QVTo [129].

Ces deux relations permettent ainsi de bien distinguer le langage qui joue le rôle de système, du (ou des) méta-modèle(s). La figure 1.6 illustre ces notions et les relations de base de l'IDM.

Graphique supprimé pour respecter le droit d'auteur
images/chap1-2/Relation-syst-mondeReel.png
FIGURE 1.6 – Notions de base en ingénierie des modèles[19]

C'est sur ces principes de base que l'Object Management Group (OMG)[60] s'appuie pour définir ses standards. L'OMG est un consortium regroupant des industriels, des fournisseurs d'outils, et des académiques dont l'objectif principal est de développer des standards pour normaliser les différentes approches d'un domaine mais aussi pour contrôler la prolifération des technolo-gies. Il définit plusieurs standards sur lesquels repose l'approche MDA [128] tels MOF [128], UML, QVT [129].

Graphique supprimé pour respecter le droit d'auteur
FIGURE 1.7 – Couches de specification du MDA [128]

La figure 1.7 montre les couches de spécification du MDA. Dans le noyau se trouvent les standards de base (MOF, UML et CWM). La première couche représente les plates-formes supportées (Java, Corba, Web Service , ect.). La deuxième couche concerne les services sys-tème et enfin l'extérieur montre les domaines pour lesquels des composants métiers doivent être définis (Transport, Télécommunication, Finance, ...).

1.3.2 Approche MDA (Model Driven Architecture)

L'architecture dirigée par les modèles ou MDA (Model Driven Architecture) [146][161] est une proposition de l'OMG initiée en 2000. Elle vise à promulguer de bonnes pratiques de modélisation par la séparation des différentes préoccupations durant le processus de développement de système logiciel. Le MDA est à la fois une architecture et une démarche de développement qui a pour but de séparer les spécifications fonctionnelles (partie métier) d'un système de leur mise en œuvre sur une plateforme spécifique d'exécution.
Ainsi le MDA propose un processus de développement fondé sur trois (3) niveaux d'abstraction et identifie quatre types de modèles : CIM, PIM, PSM, PDM.

— **CIM (Computation Independent Model)** : appelé aussi modèle métier, modélise les exigences du système et spécifie la fonctionnalité (ou le comportement extérieur) d'un système sans montrer de détail de construction.

— **PIM (Platform Independent Model)** : appelé aussi modèle d'analyse et de conception abstraite du système, le modèle PIM décrit les détails spécifiques du système sans montrer les détails de son utilisation sur une plate-forme cible. Il est possible d'élaborer plusieurs modèles PIM qui peuvent être utilisés pour dériver des modèles liés aux différents types de plateforme où la base conceptuelle est la même.

— **PSM (Platform Specific Model)** : appelé aussi modèle d'implémentation ou modèle spécifique des plateformes d'exécution, le modèle PSM est le résultat de la transformation d'un PIM pour prendre en compte les spécifications techniques de la plateforme cible. Notons qu'un modèle PSM peut être dérivé à partir d'un ou plusieurs modèles PIM (chaque modèle PIM a un but conceptuel différent).

Le passage de PIM à PSM fait intervenir des mécanismes de composition et de transformation de modèle avec un modèle de description de la plate-forme (Platform Description Model – PDM), un modèle de description de la qualité de service (Quality of Service – QoS). Cette démarche s'organise donc selon un cycle de développement en «Y» propre au MDD (Model Driven Dévelopment) (Figure 1.8).

Les principaux avantages et bénéfices du MDA sont liés à la portabilité, la productivité, la traçabilité, l'interopérabilité et la maintenance [89] [146].
La nécessité de mettre en place un langage commun de définition de méta modèle a conduit l'OMG à la définition du standard MOF se situant au nieau supérieur de l'architecture de méta modélisation, composée de quatre (4) niveaux.

1.3.3 Architecture de Méta Modélisation

Dans le cadre de l'IDM, l'OMG a défini une architecture de méta-modélisation à quatre niveaux. L'architecture proposée est présentée par la figure 1.9 sous la forme d'une pyramide.

— **Le niveau M0** (ou instance) correspond au monde réel. Ce sont les infor-

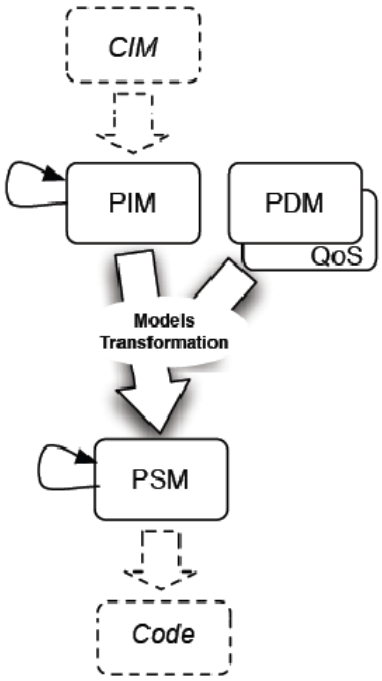

FIGURE 1.8 – Un processus en Y dirigée par les modèles [33]

mations correspondant au système réel que l'on désire modéliser.

— **Le niveau M1** (ou modèle) est composé de modèles représentant la réalité de M0. C'est au niveau M1 que les modèles sont édités. Ces modèles sont conformes aux méta modèles définis au niveau M2. Un modèle UML (comme le diagramme de classes ou le diagramme d'état/transition) est considéré comme appartenant au niveau M1.

— **Le niveau M2** (ou méta-modèle), définit le langage de modélisation et la grammaire de représentation des modèles de niveau M1. Le niveau M2 permet de définir des méta modèles. Les métamodèles contenus dans le niveau M2 sont tous des instances du niveau M3 (notons qu'au niveau M3, il ne peut y avoir qu'un seul méta-méta modèle).

— **Le niveau M3** (ou méta-méta-modèle) est composé de l'entité unique MOF. Le MOF permet de décrire la structure des méta-modèles (niveau M2), et des modèles (M1), d'étendre ou de modifier les méta-modèles

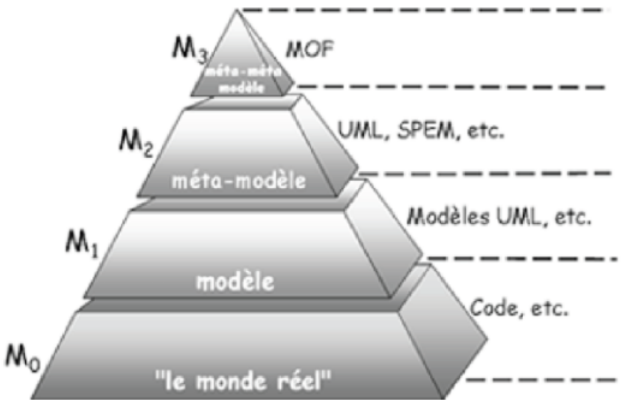

FIGURE 1.9 – La pyramide des niveaux de modélisation[19]

existants. Le MOF est réflexif, il se décrit lui-même. Ce qui permet de dire que le niveau M3 est le dernier niveau de la hiérarchie.

1.3.4 MOF : Meta Object Facility

Le MOF [128] est le langage standard de plus haut niveau d'abstraction, se situant au sommet de l'architecture à quatre niveaux de l'OMG. Entant que modèle, le méta-modèle doit être défini à partir d'un langage de modélisation appelé méta-méta-modèle, qui lui doit avoir la propriété de méta-circularité (être capable de s'auto décrire) pour éviter le nombre illimité de niveaux d'abstraction. Le standard MOF apporte le support de définition des formalismes de modélisation sous forme de méta modèle comme UML.

1.3.5 Le langage UML

Le langage de modélisation unifiée (UML) [127] est un langage de modélisation à usage général dans le domaine de l'ingénierie logicielle. Il a été créé et standardisé par L'OMG en 1997 et est à sa version 2.5.1 .
L'objectif d'UML est de fournir aux architectes système, aux ingénieurs logiciels et aux développeurs de logiciel, des outils pour spécifier, construire, visualiser et documenter des modèles de logiciels.
UML permet de décrire les aspects fonctionnels, structurels et comportementaux

d'un système en définissant douze types de diagrammes.
Bien que UML ait été conçu pour pouvoir modéliser une grande variété de systèmes, il ne peut couvrir tous les domaines. C'est pour cela qu'ont été ajoutés des mécanismes d'extension permettant de personnaliser et d'étendre le langage UML à un domaine ou à un besoin particulier. Ce concept désigne le profil UML.

Le profil UML

Les profils UML [127] permettent d'étendre les métamodèles UML existants pour les adapter à différentes fins. Cela inclut la possibilité d'adapter le métamodèle UML pour différentes plates-formes (telles que J2EE ou .NET) ou domaines (tels que la modélisation en temps réel ou de processus métier). Le profil UML permet aux développeurs d'exprimer des besoins spécifiques à leur domaine. C'est un ensemble d'extensions permettant d'adapter le méta-modèle UML à différentes plateformes techniques (J2EE, .Net, etc.).
Le profil UML est composé des définitions de stéréotypes, des contraintes et des valeurs marquées sur les éléments du modèle. Les stéréotypes sont des annotations que l'on applique à des éléments de modélisation pour les spécialiser. Notons que le stéréotype peut être enrichi par un ensemble d'attributs (tagged values). Cependant, il ne peut pas être instancié dans un modèle de l'utilisateur. Les valeurs marquées sont des paires "nom-valeur" qui ajoutent de l'information supplémentaire aux éléments stéréotypés. Ces informations sont ajoutées aux éléments lors de l'application d'un stéréotype à un élément.

Par ailleurs, la construction des modèles avec les langages de modélisation et la transformation de modèles constituent les fondements principaux de l'IDM. Cette transformation se base sur un principe des langages de transformation et des outils de transformation permettant de garantir les propriétés de réutilisabilité, de portabilité, de traçabilité et d'indépendance des modèles à un environnement spécifique.

Graphique supprimé pour respecter le droit d'auteur
FIGURE 1.10 – Schéma de base d'une transformation de modèles [19]

1.3.6 Transformation de modèles

Le principe fondamental de l'IDM est basé sur les modèles et leur transformation. « *La transformation des modèles est la génération d'un ou de plusieurs modèles cibles à partir d'un ou de plusieurs modèles sources.* »[19].
Cette transformation qui peut être (semi-)automatique), se fait suivant une définition de transformation qui est un ensemble de règles de transformation qui décrivent comment un modèle décrit avec un langage source peut être transformé en un modèle décrit avec le langage cible [20].
La définition et l'automatisation a pour but de rendre les modèles plus opérationnels et d'augmenter la productivité du développement dans une approche IDM.
Une transformation des entités du modèle source permet en un premier temps d'identifier les correspondances entre les concepts des modèles source et cible au niveau de leurs méta-modèles par l'intervention d'une fonction de transformation. Ensuite un programme appelé moteur de transformation ou d'exécution permet de transformer le modèle source en générant automatiquement le modèle cible. La figure 1.10 présente ces étapes de transformation de modèles.

Une transformation de modèle définie par un modèle Mt (cf. Figure 1.10) est l'opération qui permet de transformer un modèle Ma (conforme à son méta-modèle MMa) en un modèle Mb (conforme à son méta-modèle MMb). D'une manière plus formelle, on peut définir cette opération par la fonction suivante : **Mb : MMb = Trans (Ma : MMa, Mt : MMt)**.
Dans la partie suivante nous donnons une description générale des différents types de transformation et les langages de transformation.

Différents types de transformation

Le principe d'une transformation est un processus qui prend en entrée (input) un ou plusieurs modèles conformes à un méta-modèle spécifique et produit en

sortie (output) un ou plusieurs modèles conformes à un méta-modèle donné. Dans la classification proposée dans [35] basée sur la technique utilisée et sur la nature de l'artefact logiciel produit par la transformation, on distingue (2) deux grandes catégories de transformation :

— **Modèle vers modèle** : dans cette catégorie, chaque modèle est exprimé sous la forme d'une instance de son méta-modèle à l'aide d'une structure de donnée appropriée. Les règles de transformation sont exprimées en fonction des entités et structures définies par ces derniers.
— **Modèle vers texte** : aussi connu sous l'appellation de transformation de modèle vers code, ici le modèle source est décrit en fonction de son méta-modèle, mais le modèle résultat est produit de manière non structurée sous forme de texte (code source, fichier XML, etc.). Les règles de transformation sont donc écrites en fonction du méta-modèle du modèle source, et de la syntaxe des fichiers produits.

Par ailleurs on distingue les transformations dites endogènes et exogènes[113] , et les transformations parallèles et horizontales :

— **transformation endogène** : les modèles cible et source sont issus du même méta-modèle.
— **transformation exogène** : les modèles cible et source sont issus de méta-modèles différents.
Une transformation est dite horizontale lorsque les modèles source et cible impliqués dans la transformation sont au même niveau d'abstraction tandis que dans la transformation verticale ils sont de niveaux d'abstraction différents. Il est important de noter que les modèles source et cible peuvent appartenir à des espaces technologiques différents. Ainsi plusieurs combinaisons peuvent se produire telles que : les transformations endogène (exogène) - verticale (horizontale). La figure 1.11 résume les combinaisons possibles entre transformations de modèles.

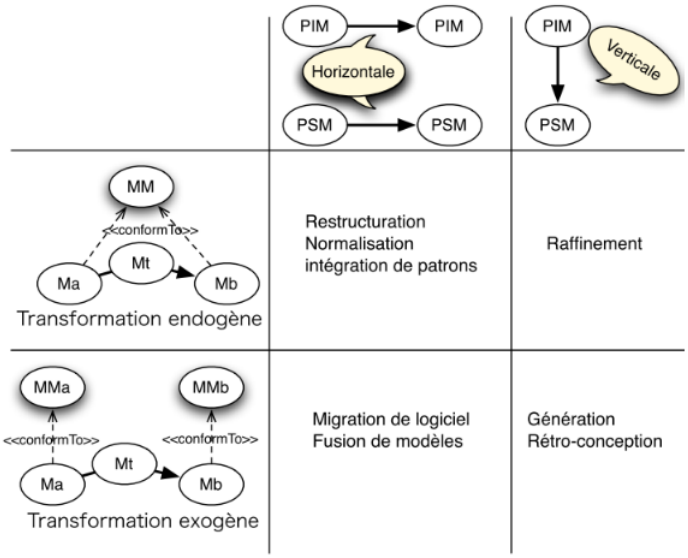

FIGURE 1.11 – Types de transformation et leurs principales utilisations[33]

1.3.7 Langages de transformations

Plusieurs langages de transformation existent et appliquent les techniques de l'IDM aux transformations elles-mêmes. Le principe est d'offrir un méta-modèle permettant de construire des modèles de transformation.
Dans le cadre de nos travaux, nous nous limiterons à présenter les langages ATL [82] et QVT [59] qui sont deux des langages de transformation des plus utilisés dans la communauté de l'IDM :

Langage ATL

Atlas Transformation Langage (ATL) [82], en français Langage de Transformation Atlas, est un langage hybride qui combine des composantes déclaratives et impératives. Il regroupe trois composants : Atlas Model Weaver (AMW) [37], ATL et ATL Virtual Machine. AMW crée des liens entre des éléments de modèle et les enregistre dans un modèle séparé, communément appelé modèle de tissage. ATL est le langage de transformation ; il prend en charge les transformations unidirectionnelles et il est utilisé pour écrire des programmes ATL, qui sont exécutés par l'ATL machine virtuelle. ATL n'est pas conforme à QVT, bien qu'il implémente des concepts et fonctionnalités similaires. ATL est défini par un modèle MOF pour sa syntaxe abstraite et possède une syntaxe concrète textuelle.

Pour accéder aux éléments d'un modèle, ATL utilise des requêtes sous forme d'expressions OCL. Les outils de transformation liés à ATL sont intégrés sous forme de plug-in ADT (ATL Development Tools) pour l'environnement de développement Eclipse.

Langage QVT

QVT (Query, View, Transformation) [129] est un langage du standard OMG pour la transformation de modèles. «**Query**» est une requête qui prend en entrée un modèle et sélectionne des éléments spécifiques de ce modèle. «**View**» est un modèle qui dérive d'autres modèles. « **Transformation**» prend un modèle en entrée pour le modifier ou en créer un autre. QVT est une spécification couplée à celle du MOF dédiée à la manipulation de modèles.
QVT est constitué de trois composantes (cf. Figure 1.12) : une partie déclarative, définie par deux langages de niveaux d'abstraction différents (*QVT-Relations et QVT-Core*) et une partie impérative (*QVT-Operational*).

Le langage *QVT-Relations* est un langage orienté utilisateur permettant de définir des transformations à un niveau d'abstraction élevé. Il implémente la transformation en fournissant des liens qui identifient les relations entre les éléments du modèle source et les éléments du modèle cible. Les traces entre les éléments impliqués dans une transformation étant créées implicitement. QVT-Relations à une syntaxe textuelle et graphique.

Le langage *QVT-Core* est un langage technique de bas niveau servant à spécifier la sémantique du langage QVT-Relations, donnée sous la forme d'une transformation RelationsToCore (cf. Figure 1.12). Ce langage défini par une syntaxe textuelle ne prend en compte que la correspondance des modèles. QVT-Core n'a pas une mise en œuvre complète et n'est pas aussi expressif que QVT-Relations.

Le langage *QVT-Operational* est la composante impérative qui étend les deux langages déclaratifs de QVT en ajoutant des constructions impératives (séquence, sélection, répétition, etc.). QVT-Operational est supporté par le langage Operational Mappings et construit pour écrire des transformations unidirectionnelles. Enfin, QVT propose un mécanisme d'extension nommé Boîte noire (Black Box) pour spécifier des transformations ; il s'agit d'invoquer des fonctionnalités de transformations implémentées dans un langage externe.

QVT-Relation et QVT-Core sont appropriés aux transformations simples où le modèle source et le modèle cible ont une structure similaire. Toutefois, lorsqu'il s'agit de transformations plus sophistiquées où les éléments du modèle cible sont intégrés à la correspondance indirecte avec des éléments dans le processeur source, les langues déclaratifs peuvent être une limitation. Ainsi, la nécessité d'un langage impératif (QVT-Operational) devient évidente. QVT intègre également le langage OCL (Object Constraint Language) qui étend des fonctionnalités impératives.

L'environnement Eclipse Modeling Framework (EMF) fournit une implémentation de QVT-Operational à travers son projet open source M2M (Model To Model). Contrairement à d'autres outils et langages qui ne prennent en charge que certains concepts de la norme QVT, Eclipse QVT-Operational (QVTO) met en œuvre la spécification finale adoptée.

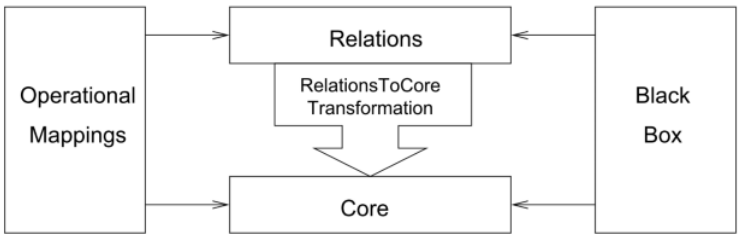

FIGURE 1.12 – Architecture du standard QVT[129]

Le choix de QVT dans notre thèse est fondé sur ses caractéristiques décrites précedemment associé au fait que QVT se base sur les standards existants à savoir le MOF et OCL. Cela permet de développer des outils simples pour un grand nombre d'utilisateurs et est compatibles avec un grand nombre d'outils de modélisation déjà existants.

1.3.8 Complémentarité de TOGAF et MDA

Cette section explore comment l'IDM peut être utilisé comme approche pour le développement d'un système dans un cadre d'Architecture d'Entreprise. Nous examinons d'abord l'alignement entre les frameworks MDA et TOGAF et comment les composants (modélisés) du système peuvent être intégrés avec la notion de transformation de modèles.

Les frameworks prescrivent l'ensemble des livrables qu'une description complète de l'architecture d'entreprise devrait avoir[24]. TOGAF ADM fournit des directives sur la récupération de l'activité d'architecture, et il fournit une méthode étape par étape pour découvrir et comprendre ce que les livrables devraient contenir[62]. TOGAF propose également la création de modèles pour capturer cette compréhension. Cependant TOGAF ne prescrit pas les moyens à utiliser pour capturer les modèles[24]. Ce qu'il faut, c'est choisir une approche de modélisation adaptée à chacun des modèles à capturer.
MDA fournit les moyens de capturer des modèles, de gérer ces modèles, de traduire entre les modèles, et de s'occuper de la génération de code en aval. "*Le but de MDA est de créer une capacité de modélisation d'architecture d'entreprise[24]*". Pour de nombreux modèles proposés dans TOGAF, MDA fournit un très bon ajustement comme indiqué dans la figure 1.13[24].

Graphique supprimé pour respecter le droit d'auteur
FIGURE 1.13 – Alignement entre MDA et TOGAF[24]

A travers ses phases A, B, C et D (cf. Figure 1.13), TOGAF ADM permet la création de modèles en phases sans spécifier d'approche de modélisation ou d'outil à utiliser, et n'entre pas dans les détails des modèles. Il apparait clairement que certains modèles MDA CIM et PIM pourraient être utilisés pour certaines parties des phases A, B, C, et D dans TOGAF[24]. Cependant l'approche MDA de transformation des modèles CIM en modèles PIM et des modèles PIM aux modèles PSM qui amélioreraient l'intégrité du passage de l'architecture à la mise en œuvre, ne fournit pas de moyens de son applicabilité dans les phases A, B et C de TOGAF.
Dans les travaux [156][157], Simonin et al. proposent une approhce d'une double intégration basée sur l'intégration d'un contexte dans un modèle de transformation MDA.
Le contexte définit un modèle/pattern encapsulant les artefacts d'un domaine spécifique (Variabilité de Données[156], Service du Système d'Information[157]) et intégrant les niveaux architecturaux du MDA (CIM et PIM) par une série de transformation.

En effet, les auteurs intègrent un modèle contextuel nommé TCM (Transformation of Contextuel Model) par enrichissement (Enhancement Transformation : ET), puis par substitution (Substitution Transformation : ST) qui produit l'application résultant en un PSM sous la contrainte technique d'un PDM.
La transformation TCM pour amélioration ou CTe (Contextuel Transformation by Enhancement) permet d'enrichir le CIM par les modèles métiers contextuels.
La transformation ET du chaînage des transformations intègre en entrée le modèle amélioré résultant.
Une transformation par substitution ST permet la réutilisation de services externes au SI existant. Cette réutilisation signifie une transformation contextuelle par substitution (CTs : Contextual Transformation by Substitution).
La substitution porte sur un extrait des opérations logiques instanciées par un service physique (externe) du SI, qui correspond aux opérations physiques définissant l'architecture applicative du service SI existant.
Ainsi, Simonin et al. [156][157] proposent une solution pour intégrer TOGAF et MDA à partir de modèles contextuels en enchaînant les transformations, en utilisant le langage de modélisation UML et utilisant le langage de transformation QVT.
En somme, la définition de modèles contextuels est une solution qui prévaut pour définir l'architecture du système en appliquant des contraintes graduelles, et en affinant les spécifications initiales du système. Ainsi dans un système orienté modèle, la définition de l'architecture peut être basée sur le raffinement conformément à MDA. Le modèle contextuel doit donc être étudié aux premiers stades (CIM : exigences des besoins métiers) de l'analyse du système avant les activités d'architecture du système, pour être intégré dans un processus conforme à MDA.

1.3.9 Discussion et conclusion

Nous avons discuté des modèles à différents niveaux abstraits et nous avons montré comment MDA peut prendre en charge tous les modèles abstraits et comment ces modèles peuvent s'intégrer et inter-opérer grâce à la notion de transformation.

Les frameworks (cadres) d'EA sont utilisés pour catégoriser les informations nécessaires afin de décrire une entreprise et stocker ses informations, généralement avec l'aide d'un outil approprié. TOGAF ADM est utilisé pour développer la description d'une architecture d'entreprise qui répond aux besoins métiers de l'entreprise, en remplissant le framework avec les modèles associés aux différentes architectures.
MDA offre des avantages considérables dans la conception et la mise en œuvre de logiciels qui répondent aux exigences métiers de l'entreprise.
Ainsi l'alignement des approches TOGAF ADM et MDA a montré que MDA peut être utilisée dans l'analyse, la conception et le developpement du système, à travers ses niveaux architecturaux décrits à partir de modèles contextuels fondés sur le cadre TOGAF. Et ces modèles peuvent être ensuite intégrés par raffinement et par transformation graduelle d'amélioration et par substitution comme proposé dans [156]. Ce qui met en évidence les améliorations qu'on peut obtenir grâce à l'utilisation de MDA dans L'architecture d'entreprise.

Dans cette partie, nous avons fait un état de l'art de l'IDM. Nous avons présenté les fondements et principes relatifs à l'IDM tels que les modèles, les métas modèles, les langages de modélisation et les relations qui les lient. Ensuite nous avons décrit quelques standards liés à l'IDM, proposés par l'OMG tels que le MDA, le MOF et UML. En outre, nous avons présenté le principe de transformation de modèles, les types de transformation et donné un aperçu de deux langages de transformation de modèles qui sont ATL et QVT. QVT est le langage de transformation de modèle choisi dans le cadre de notre thèse implémenté. L'environnement Eclipse Modeling Framework (EMF) qui est notre environnement de développement fournit une implémentation de QVT-Operational.
D'autre part nous avons étudier comment le système peut être développé dans un cadre d'Architecture d'Entreprise et avec la notion d'IDM. Notamment, comment MDA peut constituer un outil pour la modélisation et l'intégration du système dans le cadre du Framework TOGAF. Cette étude nous a permis de mettre en lumière, l'alignement entre les approches TOGAF et MDA et leur intégration.
Ainsi nous, adoptons dans le cadre de cette thèse les approches TOGAF, MDA, UML et QVT comme élements pour le developpement du système.

Dans la section suivante nous étudions comment l'IDM, à travers son approche MDA, permet la prise en compte des exigences de sécurité destinées à être intégrées dans le développement du système.

1.4 Exigences de sécurité dans le cycle développement de système logiciel

La prise en compte de la notion de sécurité dans l'ingénierie logicielle est difficile en raison de son caractère non fonctionnel et de l'insuffisance ou manque d'outils et de méthodes de développement en matière de sécurité [39]. Cependant les exigences de sécurité tout comme les exigences fonctionnelles sont l'étape fondamentale du cycle de développement logiciel.
La spécification d'une exigence de sécurité utilise des langages et des outils d'expression et de modélisation des exigences de sorte que celles-ci soient compréhensibles, utilisables par les parties prenantes, et en particulier, par les développeurs qui doivent concevoir et construire le système[48].
Dans cette partie nous présentons quelques travaux relatifs à l'ingénierie des exigences de sécurité. Nous avons regroupé ces méthodes selon le principe de modélisation sur lequel elles se basent en référence aux travaux [121] [48][39] [119][9] [104][95] [108] [41], proposés dans la littérature. Ainsi nous pouvons distinguer la catégorie de développement orientée sécurité utilisant les modèles, désignée par Model-Driven Security (MDS) [100][83] ou sécurité dirigée par les modèles et la catégorie basée sur la gestion des risques dans les systèmes d'information[10][164]. Ces catégories sont composées de sous catégories qui peuvent appartenir à plusieurs groupes à la fois selon les paradigmes considérés. Dans cette section , nous aborderons en premier les approches d'ingénierie de sécurité basées sur la gestion des risques, ensuite l'ingénierie de la sécurité dirigée par les modèles, avec une étude comparative de chacune des catégories afin de dégager les caractéristiques permettant de faire un choix pour la suite de nos travaux.

1.4.1 Approche d'exigence de sécurité basée sur la gestion des risques

Nous donnons d'abord une définition des concepts liés à la sécurité et au risque.
Dans ISO (ISO/CEI 27000, 2016)[74], la sécurité des SI est décrite en terme de :
Intégrité : ce principe est de veiller à la sauvegarde de l'exactitude et l'exhaustivité de l'information et de la façon dont il est traité.
Confidentialité : cette condition est de veiller à ce que les informations ne soient pas divulguées ou communiquées à des personnes ou entités qui ne possédent pas les autorisations appropriées.
Disponibilité : cette notion est la propriété d'être accessible et utilisable sur demande par une entité autorisée.
Traçabilité : cette notion c'est l'assurance que les éléments considérés sont tracés et que ces traces sont conservées pour leur exploitation par les personnes autorisées.

Le NIST (National Institute of Standards and Technology) définit la gestion des risques de sécurité comme "*le programme et les processus de support pour gérer les risques de sécurité de l'information pour les opérations organisationnelles (y compris la mission, les fonctions, l'image, la réputation), les actifs organisationnels, les individus, d'autres organisations et de la Nation, et comprend : (i) l'établissement du contexte pour les activités à risque ; (ii) évaluer les risques ; (iii) répondre au risque une fois déterminé ; et (iv) le suivi des risques dans le temps* ". [139].
La norme ISO définit le risque comme : « L'effet de l'incertitude sur l'atteinte des objectifs » [107].
Le risque est souvent caractérisé par référence à des événements et à des conséquences potentielles, ou une combinaison de ceux-ci. Il peut être quantifié en tenant compte des trois éléments : la menace, la vulnérabilité et l'impact.
Une menace peut avoir plusieurs sources, c'est la cause potentielle d'incident, qui peut résulter en un dommage au système ou à l'organisation [168]. Une menace exploite une vulnérabilité pour déclencher un événement d'attaque entraînant un risque.

La vulnérabilité correspond à une faiblesse du système.
L'impact est la conséquence directe ou indirecte de l'insatisfaction des besoins de sécurité sur l'organisme et/ou sur son environnement [92]. Les trois concepts cités supra peuvent modifier les critères de sécurité (confidentialité, intégrité, disponibilité,...).
Le but de la gestion des risques est l'amélioration et la crédibilité de la sécurité des SI. Dans la section suivante nous abordons quelques méthodes de gestion de risque.

Méthodes de gestion du risques

Il existe une multitude de méthodes de gestion de risque permettant de supporter l'activité de developpement logiciel [10]. Le rôlr de la méthode consiste à proposer une approche composée d'étapes, consistant généralement à inventorier le système, à le représenter, puis à identifer les risques à partir de listes génériques et à les évaluer selon une échelle bien définie. La plupart de ces méthodes suivent les étapes proposées dans la norme ISO 27005 et présentent donc des similitudes.
Nous sélectionnons un sous-groupe représentatif de l'ensemble des méthodes de gestions des risques basées sur des études récentes[95][121] :

NIST(National Institute of Standards and Technology)[164] documente une approche globale d'analyse des risques. Elle offre un framework et un plan décrivant étape par étape la mise en œuvre de la gestion des risques des systèmes d'information. Il fournit quatre publications (800-30, 800-37, 800-39 et 800-53) couvrant les activités de gestion des risques.
L'objectif de la publication spéciale NIST 800-30 est "*de fournir des orientations pour l'évaluation des risques des systèmes et des organismes d'information fédéraux*". Ce processus d'orientation permet l'identification des facteurs de risques spécifiques, permettant aux organisations de déterminer le niveau inacceptable des risques. NIST Special Publication 800-37 est lié au cadre de gestion des risques (RMF), qui fournit un processus structuré intégrant la sécurité de l'information et des activités de gestion des risques dans le cycle de vie du développement du système.

NIST Spécial Publication 800-39 «*fournit, une approche structurée et flexible pour gérer le risque, avec les détails spécifiques de l'évaluation, intervention et de suivi du risque sur une base continue fournie par l'appui d'autres normes et directives de sécurité NIST*».
Enfin, NIST Spécial Publication 800-53 (Révision 4)« fournit une approche plus holistique de la sécurité de l'information et la gestion des risques, en fournissant aux organisations des contrôles de sécurité nécessaires pour renforcer leurs systèmes d'information »[164].
Pour cela, il propose de conduire une analyse de risque sous 4 etapes : 1 Preparation de l'evaluation, 2- Eectuer une evaluation, 3 - Communiquer les resultats, 4 Maintenir l'évaluation.

CRAMM (Central Computer and Telecommunications Agency Risk Analysis and Management Method) [46] CRAMM est une méthode d'analyse et de maîtrise des risques du SI de l'entreprise, créée en 1986 par Siemens en Angleterre. Elle est utilisée pour justifier des investissements de sécurité. CRAMM met l'accent sur les dimensions techniques de la sécurité et utilise une matrice de risques combinant les menaces sur les actifs et les informations de vulnérabilité. CRAMM couvre les différentes menaces et vulnérabilités auxquelles le système d'information est exposé, qu'elles soient délibérées ou accidentelles. Le processus de CRAMM suit les étapes suivantes : (1) Identification et évaluation des actifs par rapport au coût et l'impact en cas de compromission des éléments qui constituent le système d'information de l'entreprise ; (2) Évaluation de la criticité des menaces et des vulnérabilités du système d'information ; (3) Choix de contre-mesures à mettre en place.
CRAMM permet ainsi de cibler les menaces à surveiller et de savoir quand mettre en place des dispositifs de sécurité supplémentaires..

OCTAVE (Operationally Critical Threat, Actif and Vulnerability Evaluation) [125]est une méthode, développée et publiée par le Software Engineering Institute (SEI) de la Carnegie Mellon University à travers son programme CERT (computer Emergency and Response Team).
La méthode OCTAVE comprend trois versions : OCTAVE, OCTAVE-S et OCTAVE-Allegro [125].
La version OCTAVE est utilisée dans les grandes entreprises de plus de 300 em-

ployés et fournit les lignes directrices pour la conduite de la sécurité intérieure. Quant à OCTAVE-S, elle a été développée pour les petites entreprises de moins de 100 employés. OCTAVE-S suppose que les personnes chargées de l'évaluation des risques sont connues, ainsi que les exigences de sécurité, les menaces et les pratiques de sécurité de l'organisation. Entrevues, sondages et ateliers ne sont pas nécessaire pour mener les activités de cette version.
Enfin, la méthode OCTAVE-Allegro est orientée vers l'évaluation des risques de sécurité de l'information. Elle décrit les étapes et fournit des feuilles de calcul des risques et des questionnaires, comme guides et modèles pour évaluer les risques de l'organisation. OCTAVE se décline en trois principales phases (phase 1 : vue organisationnelle, phase 2 : vue technique et phase 3 : le développement de la stratégie de sécurité et sa planification).

EBIOS (Expression des Besoins et Identification des Objectifs de Sécurité) [92], Créée en 1995 par la DCSSI (Direction Centrale de la Sécurité des SI). La démarche de EBIOS est orientée processus. Son principe général est d'identifier l'actif à partir de l'étude du contexte de l'organisation cible (notamment au niveau du périmètre SI), ensuite exprimer les besoins de sécurité à partir de critères de sécurité : "Confidentialité, Intégrité, Disponibilité", d'évaluer l'impact d'un événement de sécurité sur l'actif, de rechercher les menaces et les vulnérabilités et finalement, à partir de cette information, d'en déduire les besoins de sécurité les plus appropriés relativement au contexte de l'organisation. Une des caractéristiques d'EBIOS est l'utilisation d'une base de connaissances donnant accès à la liste des vulnérabilités (techniques) connues, des contraintes de sécurité et des méthodes d'attaques (menaces). Ci-dessous la description des différentes phases de EBIOS.

- Module 1 – Étude du contexte : À l'issue du premier module, qui s'inscrit dans l'établissement du contexte, le cadre de la gestion des risques, les métriques et le périmètre de l'étude sont parfaitement connus ; les biens essentiels, les biens supports sur lesquels ils reposent et les paramètres à prendre en compte dans le traitement des risques sont identifiés.
- Module 2 – Étude des événements redoutés : Le second module contribue à l'appréciation des risques. Il permet d'identifier et d'estimer les besoins de sécurité des biens essentiels (en termes de disponibilité, d'intégrité,

de confidentialité...), ainsi que tous les impacts (sur les missions, sur la sécurité des personnes, financiers, juridiques, sur l'image, sur l'environnement, sur les tiers et autres...) en cas de non respect de ces besoins et les sources de menaces (humaines, environnementales, internes, externes, accidentelles, délibérées...) susceptibles d'en être à l'origine, ce qui permet de formuler les événements redoutés.

— Module 3 – Étude des scénarios de menaces : Le troisième module s'inscrit aussi dans le cadre de l'appréciation des risques. Il consiste à identifier et estimer les scénarios qui peuvent engendrer les événements redoutés, et ainsi composer des risques. Pour ce faire, sont étudiées les menaces que les sources de menaces peuvent générer et les vulnérabilités exploitables.

— Module 4 – Étude des risques : Le quatrième module met en évidence les risques pesant sur l'organisme en confrontant les événements redoutés aux scénarios de menaces. Il décrit également comment estimer et évaluer ces risques, et enfin comment identifier les objectifs de sécurité qu'il faudra atteindre pour les traiter.

MEHARI (Méthode Harmonisée d'Analyse des Risques Informatiques) [114] : Mehari est une méthodologie française développée par une organisation de sécurité de l'information (CLUSIF). MEHARI fait usage d'une méthode fondée sur la connaissance des procédures de support pour l'évaluation des risques. Contrairement à EBIOS, MEHARI est une approche orientée scénarios. Elle consiste en l'analyse des scénarios qui peuvent affecter la sécurité de l'information. Son principe part de l'analyse des enjeux de la sécurité par rapport aux critères (Disponibilité, Intégrité, Confidentialité). Ces scénarios expriment les dysfonctionnements potentiels de l'entreprise. Ensuite auditer les services de sécurité afin d'identifier les vulnérabilités. Auditer et analyser les situations de risques, permettant d'évaluer les potentialités et les impacts intrinsèques, ainsi que les facteurs d'atténuation de risque. Enfin, de déduire un indicateur de gravité de risque qui permettra de prendre les mesures appropriées dans la phase de gestion de risque. La démarche MEHARI comprend trois phases :

— La phase préparatoire : Cette phase consiste à étudier le périmètre de l'étude et le contexte, de classer l'ensemble des actifs du SI, d'identifier les événements pouvant impacter le bon déroulement du SI et d'évaluer

la gravité de cet impact pour l'entreprise. Cette phase permet de générer le Plan Stratégique de Sécurité (PSS) permettant de fixes les objectifs de sécurité et les métriques permettant d'évaluer le niveau de gravité d'un risque. Il définit la politique de sécurité ainsi que la charte d'utilisation du SI pour ses utilisateurs.

— La phase d'analyse des risques : Cette phase permet de détecter les scénarios de risques qui peuvent remettre en cause un des objectifs de sécurité de l'organisation. Cette phase évalue les risques (probabilité, impact) et exprime les besoins de sécurité, et les mesures nécessaires au traitement du risque. Elle permet de générer le Plan Opérationnel de Sécurité (POS) qui définit les mesures de sécurité qui doivent être mises en œuvre.

— La phase de planification du traitement des risques : Cette phase consiste à analyser les scénarios de risque afin d'identifier et décider du traitement à adopter. Cette phase permet de générerle Plan Opérationnel d'Entreprise (POE) qui assure le suivi de la sécurité par l'élaboration d'indicateurs sur les risques identifiés et le choix des scénarios de risque contre lesquels il faut se prémunir.

Comparaison

Une comparaison générale des méthodes de gestion de risque présentées dans cette section nous permet de mettre en exergue les avantages et les limites des méthodes présentées.

Pour cela nous nous appuyons sur les travaux de [95] [41] pour établir une comparaison des méthodes de gestion du type d'évaluation, de la base de connaissances utilisée par les méthodes pour l'identification des risques, du fondement sur lequel se basent ces méthodes, de la couverture du cycle de développement logiciel et de l'utilisation d'approche de modélisation (Cf. Tableau 1.2).

Méthodes d'analyse de risque	Type d'évaluation du risque	Base de connaissance	Fondement	Cycle de développement	Approche de modélisation
NIST	Qualitative	-	-	+	-
CRAMM	Qualitative	Actifs, vulnérabilités, menaces	BS7799 ISO/CEI 27001,27002	+	-
OCTAVE	Quantitative	Type d'actifs vulnérabilités menaces	ISO 31010	++	-
EBIOS	Quantitative	Type de sources de menaces, Type d'impacts, Type d'actifs supports Menaces et vulnérabilités génériques, Mesures de sécurité génériques	ISO/CEI 15408, 27005 et ISO 31000	++	-
MEHARI	Quantitative	Actifs secondaires Scénarios de Risques Mesures de sécurité	ISO/CEI 27001, 27002,27005	++	-

Tableau 1.2 – Méthodes d'analyse de risque
Légendes :
(+) Couverture limitée à une seule phase de développement
(++) Couverture de deux ou plusieurs phases de développement mais pas la totalité
(-) Notion non couverte par la méthode

En résumé, le tableau de comparaison permet de constater que les approches étudiées ont une diversité quand aux bases de connaissances pour identifier les risques notamment avec la méthode EBIOS. Elles ont aussi une diversité des fondements quand aux standards et normes d'analyse de risque. A l'exception de NIST et CRAMM (évaluation qualitative), le type d'évaluation du risque des différentes approches est quantitatif.
Cependant ces approches ne couvrent pas le cycle de développement logiciel en totalité et aucune de ses méthodes n'est basée sur un langage de modélisation.
Ainsi la prochaine section traite de la question des approches d'exigence de sécurité basées sur les modèles.

1.4.2 Méthodes d'ingénierie de la sécurité orientées modèles

Vu le succès de l'ingénieurie orientée modèle, la communauté scientifique a suggéré d'appliquer cette approche au domaine de la sécurité pour améliorer la qualité logicielle en respectant spécifiquement les exigences de sécurité. Cette approche désignée sous l'appellation MDS (Model-Driven Security) [16] ou la sécurité dirigée par les modèles, propose des approches qui prennent en compte la sécurité dans les différentes phases du développement logiciel basées sur les modèles. A partir des classifications réalisées dans [69], [48], [39], [171], [119][95][104], dans le cadre de nos travaux , nous considérons les approches principales de sécurité dirigée par les modèles : les approches basées sur UML, les approches orientées buts, les approches basées sur les problem-frames et les approches basées sur la gestion du risque.

Approches basées sur UML

Vu l'utilisation de facto du standard UML dans l'ingénierie logicielle, plusieurs approches étendent UML pour intégrer les aspects de sécurité.

SecureUML : SecureUML [100] est basé sur l'extension du modèle UML pour l'expression du contrôle d'accès, Role Based Access Control (RBAC) [1]. RBAC

1. https ://www.sailpoint.com/fr/identity-library/what-is-role-based-access-control/

est l'approche de contrôle d'accès la plus utilisé en raison de ses nombreux avantages reconnus à cet effet. SecureUML définit donc, un vocabulaire pour l'expression de différents aspects du contrôle d'accès tels que l'utilisateur (User), le rôle (Role), la permission (Permission) ainsi que les relations qui existent entre ces aspects. SecureUML introduit le concept de contrainte d'autorisation, définie comme une pré-condition pour accorder l'accès pour une opération.
Ainsi SecureUML à l'avantage d'offrir une extensibilité significative parce qu'il combine la simplicité de la notation graphique de RBAC avec la puissance des contraintes logiques sur les modèles [6]. Les politiques simples de sécurité pouvant être exprimées à partir des permissions basées sur la notion de rôle et les plus compliquées pouvant être définies en ajoutant des contraintes d'autorisation.
Le méta modèle de SecureUML, présenté à la figure 1.14 exprime les différentes caractéristiques mentionnées précédemment.

Graphique supprimé pour respecter le droit d'auteur
FIGURE 1.14 – Méta-modèle de SecureUML [100]

UmlSec UMLsec [83] permet d'exprimer des informations relatives à la sécurité dans les diagrammes UML. Il encapsule les exigences de sécurité dans des stéréotypes UML et des balises dans le profil UMLsec. La figure 1.15 présente une description détaillée des différents stéréotypes proposés. La sécurité est définie en utilisant des valeurs marquées et des contraintes associées aux éléments du méta-modèle. Un certain nombre de risques associés aux actions (delete, read, insert, access pour les liens) et de types d'éléments (un lien de type Internet supporte les risques delete, read, insert ; un lien de type encrypted ne supporte que le risque delete) sont prédéfinis. Les risques "Read" et "access" concernent la vilotaion de la confidentialité, l'authentification, l'autorisation tandis que les risques "delete" et "insert" concernent la vilotaion de l'intégrité. Ces risques ciblent les liens de sécurité.
En outre, UMLsec définit des contraintes sur les éléments exprimant les règles de fonctionnement et la politique de sécurité, qui permettent l'évaluation des aspects sécurité d'une conception du système.

Graphique supprimé pour respecter le droit d'auteur
FIGURE 1.15 – Un extrait du profil UMLsec [83]

La spécification des menaces correspondent aux actions prises par l'adversaire. Ainsi différents scénarios de menaces peuvent être envisagés en se basant sur les capacités de l'adversaire. UMLSec ne dispose pas de méthode explicite pour la modélisation de la sécurité, mais se base sur les méthodes standards de gestion de la sécurité pour satisfaire entre autre la non-répudiation, l'intégrité, l'authenticité, la sécurité du lien de communication, la sécurité des flux de communication .
L'extension de UMLsec est donnée sous la forme d'un profil UML utilisant les mécanismes d'extension UML standard[83]. Les mécanismes d'extensions sont les stéréotypes, les valeurs étiquetées et les contraintes. Les stéréotypes(stereotypes), entre chevrons doubles "« »", définissent de nouveaux types d'éléments de modélisation prolongeant la sémantique des types ou classes existants dans le méta modèle UML. Les stéréotypes sont utilisés pour formuler des exigences de sécurité et des hypothèses sur l'environnement du système. Une valeur étiquetée est une paire nom-valeur entre accolades associant des données à des éléments de modèle, où le nom est appelé balise. La notation correspondante est tag=value avec le nom de la balise (tag) et une valeur (value) correspondante à attribuer à la balise. Les contraintes (constraints) donnent des critères qui déterminent si les exigences sont satisfaites par la conception du système.
Misuse case[158] MisUse Case ou cas de mauvaise utilisation est une autre extension de la spécification des cas d'utilisation du diagramme UML pour représenter le comportement non désiré dans le système. Sindre et Opdahl [159] ont développé cette extension pour mettre en évidence les potentielles attaques contre un système. Ces mauvais cas d'utilisation sont ajoutés au diagramme de cas d'utilisation normal. Cependant en raison du niveau d'abstraction très élevé, il est impossible de fournir un outil de génération de code à partir de ce diagramme de cas d'utilisation. Originalement les cas d'utilisation représentent les besoins métiers, les cas de sécurité représentent les besoins de sécurité du système et les cas de mauvaise utilisation, représentent les menaces de sécurité.

Approches orientées buts

Dans cette approche, les méthodes sont basées sur l'identification et la modélisation des buts en ajoutant des concepts propres aux préoccupations de sécurité abordées. Ces méthodes sont généralement appliquées à l'ingénierie des besoins. Toutefois, elles peuvent s'appliquer aux phases de conception et de définition de l'architecture.

KAOS : est désigné à la fois comme Knowledge Acquisition in autOmated Specification [172] et comme Keep All Objects Satisfied. [171]. Le modèle KAOS tire son origine du domaine de l'ingénierie des besoins et fut désigné par les chercheurs de l'université de Louvain et de l'université d'Oregon. La méthodologie décrit un Framework pour modéliser et raffiner les buts aussi bien que la sélection des alternatives. Le modèle KAOS commence à un niveau élevé qui décrit des besoins abstraits pour le système. Ces besoins abstraits sont séparés en besoins fonctionnels et besoins non-fonctionnels, tandis que les besoins de sécurité relève de la section du non-fonctionnel. Ces modèles de buts peuvent par la suite être utilisés pour générer des modèles objet, des modèles d'opération ou des modèles de responsabilité pour dériver les besoins et restrictions du développement logiciel.

Secure Tropos[118] : Tropos [26] est une méthodologie qui soutient le processus de développement logiciel en décrivant l'environnement du système et le système lui-même. Tropos est utilisé pour modéliser les dépendances entre différents acteurs du système voulant accomplir différents buts en exécutant des plans. C'est une méthode d'ingénierie logicielle qui s'applique à l'ingénierie des besoins. Secure Tropos [118] étend Tropos en ajoutant de nouveaux concepts pour atteindre la modélisation des besoins de sécurité des systèmes à développer. On retrouve les concepts tels que contraintes de sécurité, dépendances de sécurité, entités de sécurité (buts, tâches, ressources de sécurité), caractéristiques de sécurité, objectifs de sécurité, mécanisme de sécurité, menace, etc.

Secure i* [43] : étend le méta-modèle i* avec la modélisation et l'analyse des aspects de sécurité. Elle se concentre sur l'alignement des exigences de sécurité

avec d'autres exigences fonctionnelles et non fonctionnelles. Cet alignement permet d'assurer une cohérence entre les besoins de sécurité et les besoins métiers du système. Cela évite ainsi les conflits entre ces différents besoins. L'approche est fondée sur un méta-modèle de concepts de sécurité contenant quelques notions importantes et leurs relations. Les notions importantes sont les acteurs, les actifs, les menaces et les vulnérabilités qui ne peuvent pas être représentées par les notations de modélisation de i*. Les acteurs sont des entités qui ont (ou cherchent) des objectifs de sécurité. Ces derniers sont l'expression de la décision de traiter les menaces selon des modalités prescrites. Les acteurs peuvent posséder ou déléguer l'autorisation d'utilisation des actifs à d'autres acteurs.
GBRAM (Goal-Based Requirements Engineering Analysis Method)[5] : est une méthode d'analyse des besoins fondée sur des objectifs en utilisant les buts (Goals) et les scénarios pour formuler des politiques de sécurité grâce à un ensemble de questions standard. La méthode GBRAM est composée de deux activités qui sont l'analyse des buts et le raffinement des buts. La prémière (analyse des buts) consiste à explorer les sources d'information pour identifier les buts, les organiser et les classifier. La deuxième activité (raffinement de buts) concerne l'évolution des buts à partir du moment où ils sont identifiés jusqu'au moment où ils sont traduits en exigences opérationnelles. Tous les concepts de GBRAM (buts, agents, parties prenantes...) sont spécifiés seulement sous une forme textuelle dans des schémas de buts, sans qu'ils ne fournissent aucune notation graphique.

Approches basées sur les "Problem frames"

Les approches groupées dans cette catégorie utilisent les idées sous-jacentes aux « problem frames » de Jackson [75]. Les «problem frames» sont vus comme des «patterns» permettant de classifier les problèmes dans l'ingénierie des systèmes.
SEPP (Security Engineering Process using Patterns) [67] [17] : est un processus d'ingénierie de sécurité fondé sur les Frameworks de problèmes de sécurité «problem frames» et associé aux approches de solution. Ces frameworks sont disposés dans un système de modèle, ce qui permet de structurer, de caractériser,

d'analyser et de résoudre les problèmes de sécurité et logiciels du système.
SREF (Security Requirements Engineering Framework)[66] : est une approche basée sur la construction d'un contexte pour le système en utilisant une notation orientée problème afin de représenter les besoins de sécurité, et pour développer et évaluer les arguments de satisfaction pour les besoins de sécurité. SREF est un processus comptant quatre étapes intégrant les exigences ordinaires et les exigences de sécurité.
Abuse Frames [99] : est une méthode qui correspond aux anti-exigences. C'est une approche basée sur les «problem frames» pour définir les anti-besoins, qui représentent les utilisateurs malveillants pour analyser les menaces de sécurité encourues.

Approches basées sur la gestion des risques

Ces approches d'ingénierie de développement des exigences de sécurité sont basées sur les modèles et l'analyse des risques et des menaces.
SQUARE (Security quality requirements engineering methodology) [111] : fondée sur les meilleures pratiques dans l'ingénierie des exigences. La méthodologie globale de SQUARE est composée de neuf étapes. Son objectif est d'intégrer l'ingénierie des besoins de sécurité dans le processus de développement des systèmes. Elle met en évidence sept différents types de besoins de sécurité tels que : contrôle d'accès, protection physique, politique de sécurité, non-répudiation, récupération du système, détection des attaques et des dispositifs de protection.
Tropos Goal-Risk Framework[8] : évalue les risques en tenant compte des relations de confiance qui peuvent exister entre les acteurs. La confiance est définie comme " *la probabilité subjective qui définit l'attente d'un acteur sur le comportement rentable d'un autre acteur*". L'objectif de Tropos Goal-Risk Framework est d'évaluer le risque d'événements incertains sur les stratégies d'organisation et d'évaluer l'efficacité des traitements de ces risques. A cette fin, elle a étendu la méthode Tropos[26] en rajoutant trois (03) couches à l'existant, et qui sont : l'objectif, l'événement et le traitement.
SREP (Security Requirements Engineering Process)[112] : met l'accent sur le développement des dimensions de la sécurité à la phase initiale du cycle

de vie de développement du SI. SREP est dirigé par les modèles et est basée sur les principes de CC (Common Criteria) pour supporter la réutilisation des besoins de sécurité ainsi que l'inclusion des notions d'actif, de menace et de contre-mesure.

La méthode CORAS[38] : est définie comme un cadre fondé sur un modèle pratique pour l'évaluation des risques sans ambiguïté pour les systèmes critiques. Elle fournit un langage graphique pour la modélisation du risque. Elle est fondée sur le standard australo-néo-zélandais(AS / NZS 4360 : 2004). L'analyse des risques de sécurité de la méthode CORAS se compose de huit étapes différentes où les quatre premières étapes se concentrent sur la création du contexte et les quatre dernières étapes sont sur le risque (identification, estimation, évaluation et traitements possibles du risque).

ISSRM (Information Risk Management Security System)[104] : est un modèle qui est défini à partir des approches de gestion des risques, des normes et standards de sécurité, des méthodes de gestion des risques de sécurité et des cadres d'ingénierie logicielle. Le modèle de domaine supporte l'alignement (compatibilité) des langages de modélisation de sécurité. Le modèle décrit trois catégories conceptuelles différentes (cf. Figure 1.16).

Concepts liés à l'actif ou «Asset» (en jaune) : décrivent les actifs de l'organisation regroupés en actifs métier «business asset» et actifs informatiques «IS asset». Ainsi les objectifs de sécurité basés sur les critères de Confidentialité, Intégrité, Disponibilité sont définis pour la protection des actifs «Asset». Les concepts relatifs au risque (en rouge) : définissent le risque, la menace, la vulnerabilité, l'impact, l'événement, la méthode d'attaque.

- Les concepts relatifs au traitement du risque (en vert) : définissent une décision de traitement du risque, les exigences de sécurité et les contrôles de sécurité à implémenter.

ISSRM est supporté par un processus en six étapes, basé sur les méthodologies et les normes d'analyse de risque existantes.

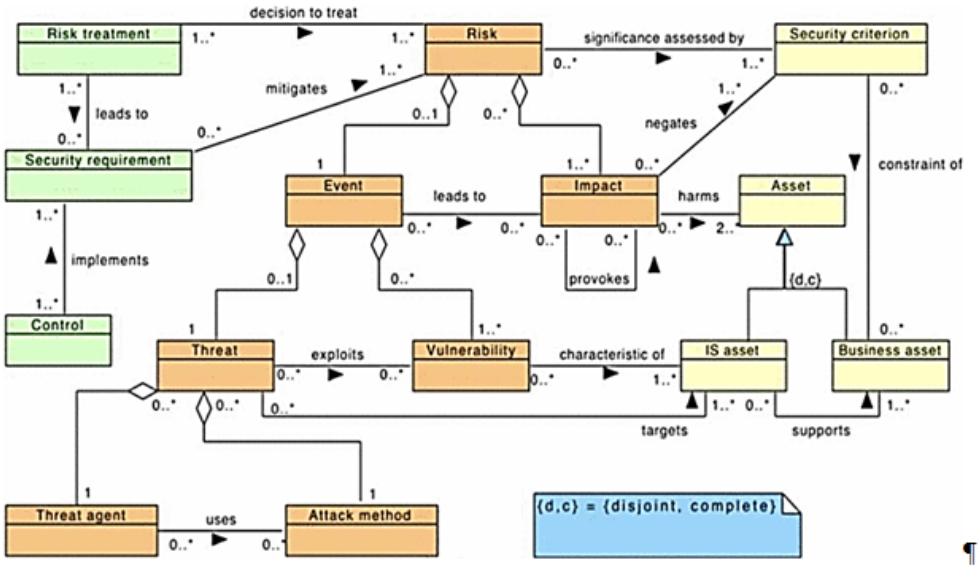

FIGURE 1.16 – Méta-modèle ISSRM [104]

Comparaison

Nous avons établi différents critères, à partir des travaux [48][119][104], pour évaluer les caractéristiques et contributions des différentes approches de l'ingénierie de sécurité orientée modèle à partir de critères basés sur :
- les propriétés de sécurité abordées : telles que CID (Confidentialité, Intégrité, Disponibilité), AC (Contrôle d'Accès), Authentification, non répudiation, responsabilité, ...
- le risque, les ménaces et la vulnérabilité
-les principes de modélisation (le paradigme de modélisation) : type d'approche de modélisation correspondant à la méthode. Le MPM (Multi-Paradigm Modeling) par exemple, utilisé par UMLsec, permet d'utiliser divers langages de modélisation spécialisés au niveau approprié d'abstraction pendant le développement de logiciel) ; MDA (Model-Driven Architecture propose une méthodologie composée par un ensemble des modèles de transformation afin de construire les différents niveaux d'abstraction dans le développement de logiciel) + DSM (Domain Specific Modeling : utilise systématiquement des modèles spécialisés pour spécifier des parties concrètes d'un système), méthodes orientées modèles, etc.

- le langage de modélisation basé sur le profile UML (extension d'UML pour des besoins de sécurité), extension du méta modèle i* et l'utilisation ou non d'un otuil de modélisation.

Enfin nous avons étudié l'étape de développement (ingénierie des exigences, conception, architecture) logiciel abordée par ces méthodes.

Les différentes approches sont consacrées à certains aspects précis de la sécurité et généralement orientées que vers l'étape de l'ingénierie des exigences du système. La plupart des approches disposent d'un langage de modélisation, certaines s'appient sur un otuil de modelisation mais, très peu de ces approches sont basées sur MDA.

Plusieurs de ces approches utilisent les modèles et le risque pour définir les politiques de sécurité, mais aucune de ces approches ne s'appuie complètement sur l'IDM et le MDA afin de formaliser leurs modèles et leurs transitions.

Catégorie	Méthode/ approche	Propriétés de sécurité	Risque	Menace/ Vulnérabilité	Paradigme de modélisation	Language de modélisation	Stade de développement	Outil
Basée sur UML	SecureUML	Contrôle d'accès (AC)	-	-	MDA + DSM	profile UML	Conception	++
	UMLsec	CID (confidentialité, Intégrité, Disponibilité)	-	++	MPM	profile UML	Conception/ Architecture	++
	Misuse cases	CID	+	++	MDA + DSM	profile UML	Conception	-
Orientée But	KAOS	CID, non répudiation, confidentialité, authentification	-	++	orienté modèle	Sécurité (adapté DSL)	Exigences	++
	Secure Tropos	CID, non répudiation, confidentialité	-	-		Extension du méta modèle i* (adapté DSL)	Exigences	++
	Secure i*	CID, récupération, responsabilité,	-	++		Extension du méta modèle i* (adapté DSL)	Exigences	++
	GBRAM	AC, confidentialité, vie privée	++	++	Séquence d'étapes basée sur l'heuristique	-	Exigences	++
Basée sur les "Problem Frames"	SEPP	Confid., Intégrité	-	-	MDA+DSM	Profile UML	Exigences	++
	SREF	CID	++	+	Orienté modèle	ESR(Evolving Security Requirement)	Exigences	++
	Abuse Frames	Analyse de ménace	-	++	Etape séquentielle	-	Exigences	-
Basée sur la gestion du risque	SQUARE		++	++	Etape séquentielle	-	Exigences	++
	Tropos Goal RF	CID	++	++	Orienté modèle	GR6Model	Exigences	++
	SREP	CID	++	++	Unified Process	Méta modèle SREP	Exigences	-
	CORAS	Analyse de risque	++	++	orienté modèle	QFTP	Exigences	++
	ISSRM	CID	++	++	Orienté but	Extension du méta modèle i*	Exigences	-

Tableau 1.3 – Méthodes de développement basées sur les modèles, la sécurité et le risque
Légende :
++ : couvert complétement
+ : couvert partiellement
- : Non couvert

1.4.3 Discussion et conclusion

Les exigences de sécurité sont connues pour être difficiles à identifier, à exprimer et à gérer.
L'analyse des risques est souvent considérée comme un aspect important de l'ingénierie des exigences de sécurité. Dans cette catégorie on distingue les méthodologies de gestion de risque telles que (EBIOS, OCTAVE, CRAMM...) qui proposent un processus de guidage, constitué d'étapes successives, permettant l'analyse, la spécification et l'évaluation du risque. Cependant la plupart des approches de cette catégorie ne couvrent pas toutes les phases de processus théorique de gestion de risque tel que proposé par l'ISO 31000. En plus ces approches ne disposent pas de langage formel de modélisation du risque et le passage de l'analyse à la conception du système est le plus souvent manuel s'il n'existe pas.
L'étude des approches des exigences de sécurité basées sur les modèles nous a permis d'en voir les points forts et les limites pour le développement logiciel.
Une partie de ces approches (basées sur UML : UMLSec et SecureUML) prennent en compte la sécurité à un niveau conceptuel du système, ne traitent pas du risque et ne supportent pas la modélisation et l'analyse de la sécurité à un niveau des exigences (métier). Elles ont pour but de modéliser les systèmes informatiques et les mécanismes de contrôle d'accès associés.
Les différentes approches étudiées, tendent parfois à être partielles et ne modélisent pas tous les aspects de la sécurité. D'autres approches (Secure i* et Secure Tropos) traitent les problèmes de sécurité de façon générale et non pas pour un domaine donné. Bon nombre de ces méthodes (Tropos Goal-Risk Framework, CORAS, ISSRM, SREP) considèrent le risque et les menaces.
En définitive les approches présentées sont consacrées à certains aspects précis de la sécurité et généralement orientées que vers l'étape de l'ingénierie des systèmes. De plus, plusieurs de ces approches utilisent les modèles pour définir les politiques de sécurité mais aucune de ces approches ne s'appuie complètement sur l'IDM et le MDA afin de formaliser leurs modèles et leurs transitions.
Cette problématique constitue l'un des challenges de notre thèse : comment l'approche MDA de l'IDM peut permettre la prise en charge des différentes étapes

de la modélisation du système en intégrant les aspects de sécurité et assurant la formalisation des modèles conformément aux méta modèles de référence.
Dans les prochains chapitres nous proposons une méthode (en guise de contribution) basée sur les modèles, qui intègre l'ingénierie des systèmes et la sécurité ; dans un cadre de (EA), fondée sur le (MDA) de l'IDM. Ainsi, nous proposons des modèles contextuels de risques des architectures métier, logique et physique (impactées par la technologique) permettant la prise en compte de la sécurité au plus tôt et les dériver/transformer (sémi) automatiquement d'une étape vers une autre dans l'ingénierie des systèmes.

Dans ce chapitre (cadre Général), nous avons fait l'état de l'art des approches d'architecture d'entreprise, de l'IDM et d'ingénierie des exigences de sécurité durant le processus d'ingénierie logicielle. A partir des travaux présents dans la littérature , nous avons analysé et/ou classifié les différentes approches des domaines respectifs et discuté du choix pour notre thèse.
Dans le chapitre suivant, nous présentons la méthode d'intégration du modèle contextuel du risque métier qui permet la prise en compte des besoins de sécurité au niveau métier durant le développement du système.

Deuxième partie

Contributions

Chapitre 2

Intégration contextuelle du risque dans l'architecture métier du système

2.1 Introduction

Dans ce chapitre nous présentons notre première contribution qui est une intégration du modèle contextuel du risque métier, libellé TCM-BR (Transformation Contextual Model of Business Risk), dans le processus de développement du système/logiciel. Ce modèle de concepts est un méta-modèle fondé, dans notre étude, sur les Frameworks d'Architecture d'Entreprise TOGAF (The Open Group Architecture Framework)[62] et SABSA(Sherwood Applied Business Security Architecture) [140].
L'intégration du modèle contextuel représentant le risque de sécurité du service métier, permet d'enrichir le modèle métier CIM (Computation Independent Model) de l'architecture du système d'un SI par transformation au sens de l'IDM (Ingénieurie Dirigée par les Modèles), notamment basé sur l'approche MDA (Model-Driven Architecture) [146].

Ce chapitre traite de la question de recherche **QR1 : Comment intégrer la sécurité dans l'architecture métier associée au processus d'ingénierie système, via les risques**?. Plus précisement les sous questions **QR1.1 : Quels éléments de l'architecture métier peut-on cibler pour une intégration de la sécurité dans le système ?** et **QR1.2 : L'approche MDA pour l'intégration de la sécurité au**

niveau métier est-elle pertinente comme mécanisme ?.
Le chapitre est organisé comme suit : dans un premier temps, nous présentons l'étude du contexte lié à la sécurité du système et des travaux connexes, avant de proposer une méthode d'analyse des menaces et des risques de sécurité SI ainsi qu'un processus de gestion des risques de sécurité. Ensuite nous présentons l'approche de construction du modèle contextuel de risque de sécurité du service métier (pour répondre à QR1.1) qui nous permet de décrire le méta modèle de sécurité du risque métier TCM-BR. Le processus d'intégration du modèle du risque métier(pour répondre à QR1 et QR1.1) est présenté avant de terminer par une illustration.

2.2 Contexte et travaux connexes

La sécurité est un aspect crucial des systèmes logiciels actuels. Ainsi, les exigences de sécurité doivent être adéquatement prises en compte dans toutes les phases du développement logiciel, à partir de la collecte des exigences métiers jusqu'à la mise en exploitation du système[66].
Pour répondre à la problématique complexe du développement de systèmes sécurisés répondant aux exigences de sécurité souhaitées, plusieurs approches de développement basées sur les modèles ont récemment apparu[16][83][94].
Le modèle de domaine pour la gestion des risques de sécurité du SI (ISSRM) [104] a déjà été utilisé pour aligner les concepts de langages d'ingénierie des exigences [102][103]. Il définit les concepts de gestion des risques de sécurité à trois différents niveaux conceptuels, ce qui aide les développeurs à identifier des concepts spécifiques de gestion des risques de sécurité du SI.
Barateiro et al. [15] proposent un alignement entre la gestion des risques, la gouvernance et les activités d'architecture d'entreprise (ces concepts se limitant à la cartographie du SI) afin de fournir un soutien systématique pour cartographier et tracer les risques identifiés pour les artefacts modélisés dans une EA.
Innerhofer-Oberperfler et Breu [73] proposent une approche pour l'évaluation et l'analyse systématiques des risques liés au SI dans les organisations et les projets. L'approche est basée sur un modèle utilisant une architecture d'entreprise comme base du processus de gestion de la sécurité. Les auteurs fournissent

une description intégrée de la structure, des processus et du paysage informatique sous-jacent d'une organisation à partir des couches métier, applicative, technique et physique. Cela permet d'établir un pont entre les points de vue techniques et métiers sur la sécurité de l'information. La proposition fournit un méta modèle de sécurité de l'information et un processus détaillé de gestion de la sécurité et définit les responsabilités nécessaires et les rôles des parties prenantes participantes.
De la même manière, Ertaul et Sudarsanam [47] proposent d'exploiter le cadre (framework) de Zachman [179] pour définir et concevoir des outils pour sécuriser une entreprise. Cela aide à prendre en charge la planification de la sécurité, en particulier pour le SI. Dans[58], Gandry et al. ont proposé une intégration de la gestion de sécurité des risques et de l'architecture d'entreprise. L'intégration est sous la forme d'une cartographie des concepts entre la gestion des risques de sécurité du système d'information (Information System Security and Risk Management : ISSRM) et la gestion de l'architecture d'entreprise (EAM : Enterprise Architecture Modelling). L'approche s'appuie sur la modélisation de l'architecture d'entreprise pour prendre en charge l'identification des actifs métiers et SI. Il propose également de modéliser le traitement des risques, notamment en relation avec la valeur du risque. Cependant, cette approche n'apporte pas un réel soutien dans l'identification des menaces et des vulnérabilités associés aux éléments de l'architecture.
Les références citées précédemment développent des avancées conceptuelles ou méthodologiques en reliant la modélisation de l'architecture d'entreprise à la gestion des risques des SI, mais aucune d'entre elles ne propose un modèle intégré et complet pour les deux domaines à partir d'une approche d'ingénierie système/logicielllle telle que l'IDM(Ingénierie Dirigée par les modèles)[146][89].
En plus, au meilleur de nos connaissances, il n'existe pas de vastes et matures travaux de recherche essayant de tirer profit de la recherche en EA et IDM pour améliorer la prise en compte de la sécurité, à partir du risque dans le domaine spécifique de la sécurité de l'information et proposer un modèle conceptuel complet et totalement intégré des trois domaines.
L'IDM ou MDE (Model-Driven Engineering) en anglais, est apparue comme un nouveau domaine du génie logiciel, dont le but est la définition de théories, mé-

thodes, techniques et des outils pour appliquer ce paradigme de développement basé sur des modèles. MDE traite des modèles en tant qu'artefacts primaires lors du développement de logiciels.
L'initiative d'architecture dirigée par les modèles (Model-Driven Architecture : MDA) [89] utilise différents niveaux d'abstraction (CIM, PIM, PSM) pour traiter le problème et le domaine de la solution et définit des méthodologies pour chaque niveau d'abstraction. MDE fournit également des techniques pour abaisser le niveau d'abstraction en définissant des relations entre les modèles participants.
La sécurité pilotée par les modèles (MDS : Model Driven Security) [101] tire partie des techniques d'ingénierie dirigée par les modèles (MDE) pour la modélisation et la mise en œuvre des soucis de sécurité.
Les approches de MDS existentes, basées sur UML se sont enrichies de capacités de modélisation de la sécurité. Dans Misuse Cases[159] et Abuse Cases [110], qui sont des extensions de diagrammes de "cas d'utilisation", l'accent est mis sur l'élucidation de nouvelles menaces et vulnérabilités qui pourraient être exploitées par des acteurs malveillants.
Cependant, ces méthodes manquent de mécanismes de formalité et de transition pour passer des exigences de sécurité aux politiques de sécurité, ce qui entraîne une lacune dans la mise en œuvre de la sécurité.
Notre approche est orientée modèle dans un cadre d'architecture métier de l'EA. Elle s'appuie sur une combinaison des cadres TOGAF et SABSA pour fournir un ensemble fourni des artéfacts (structures) de l'EA pour la modélisation du système.
Néanmoins, dans le but de soutenir une approche fondée sur le métier pour la gestion d'entreprise en matière de sécurité, les informations relatives à la sécurité doivent être fournies au bon niveau d'abstraction. L'approche choisie est de voir comment la sécurité pilotée par les modèles peut répondre à ce besoin en modélisant les exigences de sécurité comme une préoccupation dès l'étape des exigences métiers. Ensuite, comment la sécurité peut être intégrée dans le processus de développement du système, par le biais du mécanisme de transformation de modèles utiles pour leur intégration dans l'architecture globale du système, et ce, tirant ainsi profit de la séparation des préoccupations du concept

de l'IDM.
Notre approche utilise les diagrammes UML pour la modélisation du système et les spécifications de sécurité. Notamment nous utilisons le digramme de classe pour la description du système du point de vue métier et la prise en compte de la sécurité qui décrit les menaces des attaquants à l'endroit du système.
Nous utilisons les avantages de l'approche MDA pour modéliser le système du point de vue métier et résoudre le défi de la transition entre les différentes étapes avec des concepts de transformation de modèle, et nous spécifions la sécurité par le biais de l'analyse des menaces et du risque basée sur STRIDE [154][120] et la méthode d'analyse des risques EBIOS Risk Management[92].
La section suivante décrit notre approche de construction du modèle contextuel de risque métier.

2.3 Approche d'analyse des menaces et du risque

L'objectif de cette section est de proposer une méthode de l'évaluation du risque à partir de d'analyse des menaces de sécurité. Elle est un support pour l'activité des exigences de sécurité pendant l'expression des besoins métiers.
La modélisation des menaces est une forme avancée de gestion des risques, appliquée vers diverses applications et composants associés qui gèrent les sources d'informations critiques. Cette forme de gestion des risques suit des méthodologies plus avancées pour collecter les informations ainsi que ses exercices d'analyse [120] [85].
La modélisation des menaces [133] est une procédure visant à optimiser la sécurité en identifiant les objectifs et les vulnérabilités, puis en définissant des contre-mesures pour prévenir ou atténuer les effets des menaces présentes dans le système. Plusieurs méthodes de modélisation des menaces ont été développées. Mais toutes ne sont pas exhaustives. Quelques unes se concentrent sur l'abstraction et encouragent la granularité tandis que d'autres sont plus centrées sur les personnes. Certaines méthodes se concentrent spécifiquement sur les risques ou les problèmes de confidentialité.
Les méthodes de modélisation des menaces peuvent être combinées pour créer

une vue plus robuste et complète des menaces potentielles. Dans [152], les auteurs proposent une étude de classification de (12) méthodes de menaces selon plusieurs critères. Parmi les méthodes, nous avons TRIKE, Attak Tree, STRIDE, Lindun...
STRIDE est actuellement la méthode de modélisation des menaces la plus aboutie. Inventé par Loren Kohnfelder et Praerit Garg en 1999 et adopté par Microsoft en 2002, STRIDE a évolué au fil du temps pour inclure de nouveaux tableaux spécifiques aux menaces et les variantes STRIDE-per-Element et STRIDE-per-Interaction[85][152]. STRIDE se concentre sur l'identification des menaces potentielles dans chaque sous-composante du système.
Dans notre thèse, nous effectuons une modélisation des menaces à l'aide de STRIDE qui est une approche relativement légère. Le choix de STRIDE est motivé pour plusieurs raisons [85] : (i) c'est une approche systématique et qui permet l'analyse des cyber menaces contre chaque composant du système sur la base de ses connaissances techniques, (ii) il est complet et analyse les propriétés de sécurité telles que : authentification, autorisation, confidentialité, intégrité, non-répudiation et disponibilité contre chaque composant du système, et (iii) il fournit une compréhension claire de l'impact d'une vulnérabilité des composants sur l'ensemble du système et permet de garantir la sécurité du système au niveau des composants.
Ensuite, l'analyse et l'évaluation des risques est réalisée à partir des menaces résultantes de l'approche STRIDE. L'analyse des risques est l'analyse quantitative des risques présents dans un système [92]. Elle est effectuée sur la base des résultats de la modélisation des menaces. L'analyse des risques est effectuée pour trouver les événements redoutés et les scénarios de menace.
Le choix de EBIOS pour l'analyse des risques est dû au fait que c'est une boîte à outils à usage variable. EBIOS permet d'analyser les risques, de les évaluer et de les traiter dans le cadre d'une amélioration continue. La spécificité d'EBIOS réside dans sa souplesse d'utilisation[92]. Il s'agit d'une véritable boîte à outils, dont les activités à réaliser, leur niveau de détail et leur séquencement seront adaptés à l'usage désiré.
Notre approche est illustrée par la figure 2.1 qui présente les deux étapes que nous présentons ci-après.

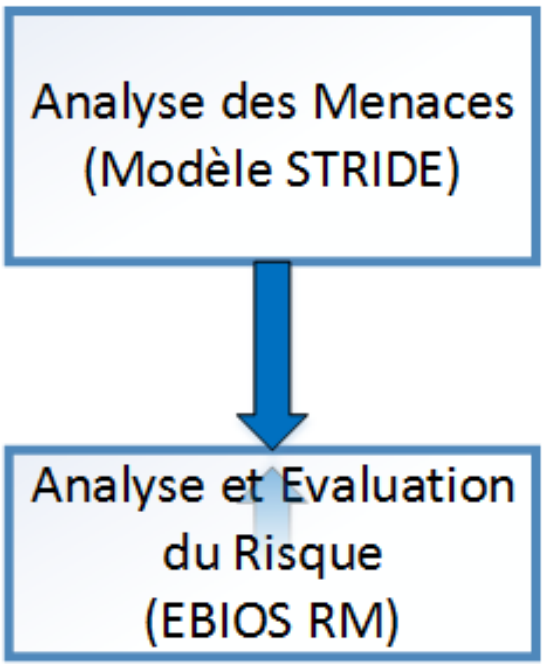

FIGURE 2.1 – Approche de modélisation des menaces et des risques

2.3.1 Modélisation et analyse des menaces : la méthode Stride

STRIDE[120] signifie :

— **Spoofing** (Usurpation d'identité) : un exemple d'usurpation d'identité consiste à accéder illégalement aux informations d'authentification d'un autre utilisateur, telles que le nom d'utilisateur et le mot de passe, puis à les utiliser.

— **Tampering** (Falsification des données) : la falsification des données implique la modification malveillante des données. Les exemples incluent les modifications non autorisées apportées aux données persistantes, telles que celles contenues dans une base de données, et la modification des données lorsqu'elles circulent entre deux ordinateurs sur un réseau ouvert, tel qu'Internet.

— **Repudiation** (Les menaces de répudiation) : Les menaces de répudiation sont associées aux utilisateurs qui nient avoir effectué une action sans que les autres parties n'aient aucun moyen de prouver le contraire. Par exemple, un utilisateur effectue une opération illégale dans un système qui n'a pas la capacité de retracer les opérations interdites. La non-répudiation fait référence à la capacité d'un système à contrer les menaces de répudiation. Par exemple, un utilisateur qui achète un article peut avoir à signer pour l'article lors de sa réception. Le vendeur peut alors utiliser le reçu signé comme preuve que l'utilisateur a bien reçu le colis.

— **Information Disclosure** (Divulgation d'informations) : les menaces de divulgation d'informations impliquent l'exposition d'informations à des personnes qui ne sont pas censées y avoir accès, par exemple, la capacité des utilisateurs à lire un fichier auquel ils n'ont pas été autorisés à accéder, ou la capacité d'un intrus à lire des données en transit entre deux ordinateurs.

— **Denied Of Service (DoS)** - (Déni de service) : les attaques par déni de service (DoS) refusent le service aux utilisateurs valides, par exemple en rendant un serveur Web temporairement indisponible ou inutilisable. Vous devez vous protéger contre certains types de menaces DoS simplement pour améliorer la disponibilité et la fiabilité du système.

— **Elevation of privilege**(Élévation de privilège) : Dans ce type de menace, un utilisateur non privilégié obtient un accès privilégié et dispose ainsi d'un accès suffisant pour compromettre ou détruire l'ensemble du système. Les menaces d'élévation de privilèges incluent les situations dans lesquelles un attaquant a effectivement pénétré toutes les défenses du système et fait partie du système de confiance lui-même, une situation dangereuse en effet.

La figure 2.2 illustre notre méthode d'analyse des menaces.
Dans le modèle de système (modèle métier et SI) où tous les actifs critiques sont

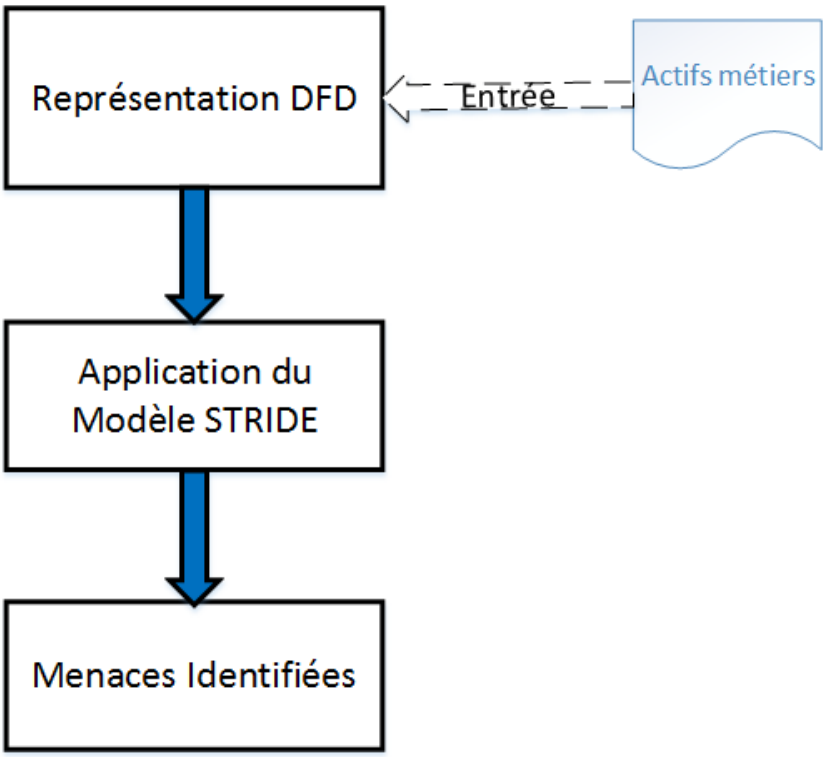

FIGURE 2.2 – Diagramme de flux de travail pour la modélisation des menaces

connus, le diagramme de flux de données (DFD) basé sur les actifs (tâches) du service métier est dessiné, et illustré à la figure 2.2.

Graphique supprimé pour respecter le droit d'auteur
FIGURE 2.3 – Diagramme de flux de donnée des menaces DFD [133]

Pour appliquer la modélisation des menaces STRIDE à un système, la première étape consiste à décomposer le système en sa logique ou composants structuraux. Les composants peuvent être des processus/éléments communiquant en interne au sein du système ou des éléments externes communiquant avec le système. L'étape suivante consiste à tracer un diagramme de flux de données (DFD) pour chaque composant système qui visualise ses fonctionnalités internes ou externe au système. Le DFD utilise quatre symboles standard :
(i) Flux de données (Data Flow) c'est-à-dire, données de communication, (ii) Données stockées (Data Store) c'est-à-dire base de données, journaux ou fichier, (iii)

Processus (Process) c'est-à-dire, unités de fonctionnalité et (iv) Entité externe/ inter acteur (Interactor), c'est-à-dire les extrémités du système.
Le DFD est utilisé pour identifier les menaces présentes dans le système. Le modèle de menace STRIDE identifie six attaques, à savoir l'usurpation d'identité, la falsification, la répudiation, la divulgation d'informations, le déni de service et l'élévation de privilège. Ce modèle vérifie parmi ces attaques laquelle sera la plus sujette au système.
Les actifs et les menaces sont étroitement corrélés. Une menace ne peut pas exister sans un actif cible. Les menaces sont généralement prévenues en appliquant une sorte de protection aux actifs. Dans la modélisation des menaces, le modèle d'un système montre toutes les entités critiques pour la sécurité telles que les actifs, les points d'accès et les canaux de communication. Les menaces peuvent être identifiées en parcourant chacun de ces éléments critiques de sécurité et en créant des hypothèses de menace qui violent la confidentialité, l'intégrité ou la disponibilité de l'entité.

Modèle de menace	Propriété de sécurité violée	Stratégie d'atténuation	Technique d'atténuation
Spoofing	Authentication	Authentication	Passwords, Tokens, Biometrics, Https, Ipsec, Crypto tunnel, Digital signatures, Authenticators
Tampering	Integrity	Integrity, permission	ACLs/permissions, Https, Ipsec, Digital Signatures, Keyed Mac, Crypto Tunnels
Repudiation	Non repudiation	Fraud prevention, logging, signatures	Digital Signatures, Logging
Information Disclosure	Confidentiality	Permissions, encryption	SSL, Ipsec, Https, File encryption (PGP), Disk encryption (FileVault, ItLocker)
Denial of Service	Availability	Availability	Fail over, load balancing, Elastic cloud design, More capacity
Elevation of privilege	Authorization	Authorization, isolation	Roles, Privileges, Fuzzing, Input validation, Firewalls, Sandboxes

Tableau 2.1 – Modèle de menaces STRIDE et mesure d'atténuation

Le résultat du processus d'identification des menaces est un profil de menace pour un système, décrivant toutes les attaques potentielles, chacune des menaces qui doit être atténué ou accepté. Le tableau 2.1 (extrait et adapté de [154]) résume la méthode STRIDE : les propriétés de sécurité violées par chacune des menaces (colonne 2), la stratégie de défense (atténuation) suggérée (colonne 3) et les mécanismes de sécurité appropriés (colonnes 4).

2.3.2 Méthode d'analyse et d'évaluation des risques basée sur EBIOS

L'analyse des risques est effectuée pour chaque actif en fonction des menaces et de leurs valeurs de risque (besoin de sécurité). Nous définissons les éléments extraits de la méthode EBIOS, pris en compte pour cette activité.
Le tableau 2.2 présente les propriété de sécurité définies avec les echelles correspondantes comme besoins de sécurité.

Niveau de l'échelle	**Description détaillée de l'échelle**
Confidentialité	
Public	La ressource est public
Limité	La ressource ne doit être accessible qu'au personnel et aux partenaires
Réservé	La ressource ne doit être accessible qu'au personnel interne impliqué
Privé	La ressource ne doit être accessible qu'à des personnes identifiées et ayant le besoin d'en connaitre
Intégrité	
Détectable	La ressource peut ne pas être intègre si l'altération est identifiée
Maitrisé	La ressource peut ne pas être intègre, si l'altération est identifiée et l'intégrité de la ressource retrouvée
Intègre	La ressource doit être rigoureusement intègre
Disponibilité	
Plus de 72H	Le service peut être indisponible plus de 72 heures
Entre 24 et 72h	Le service doit être disponible dans les 72 heures
Entre 4 et 24h	Le service doit être disponible dans les 24 heures
Moins de 4h	Le service doit être disponible dans les 4 heures

Tableau 2.2 – Echelle de besoins de sécurité

Les sources de menaces peuvent être une source humaine interne, externe (employé, partenaire, partie tierce, hacker, hacktiviste...) malveillante, avec des capacités importantes de nuisance, un virus ciblé ou non, évenement interne (panne électrique ou incendie menant à l'indisponibilité des serveurs hébergeant le service métier).
Le tableau 2.3, résume les critères retenus des modules EBIOS, pour l'analyse des risques, adaptés à notre étude.

Action	**Critère de gestion des risques (règle choisie pour réaliser l'action)**
Estimation des événements redoutés (module 2)	Les événements redoutés sont estimés en termes de gravité à l'aide de l'échelle définie à cet effet
Evaluation des événements redoutés (module 2)	Les événements redoutés sont classés par ordre décroissant de vraisemblance
Estimation des scénarios des menaces (module 3)	Les scénarios de menaces sont estimés en termes de vraisemblance à l'aide de l'échelle définie à cet effet
Evaluation des scénarios de menace (module 3)	Les scénarios de menaces sont classés par ordre décroissant de vraisemblance
Estimation des risques (module 4)	La gravité d'un risque est égale à celle de l'évenement rédouté considéré. La vraisemblance d'un risque est égale à la vraisemblance maximale de tous les scénarios de menaces liés à l'événement redouté considéré
Evaluation des risques (module 4)	Les risques dont la gravité est critique, et ceux dont la gravité est importante et la vraisemblance forte ou maximale, sont jugés comme intolérables
Evaluation des risques (module 4)	Les risques dont la gravité est importante et la vraisemblance significative, et ceux dont la gravité est limitée et la vraisemblance forte ou maximale, sont jugés comme significatifs. Les autres risques sont jugés comme négligeables.

Tableau 2.3 – Critères de gestion des risques (extrait de EBIOS, 2022)

Les métriques rétenus pour les activités d'analyse et évaluation du risque sont : la gravité des évenements rédoutés, la gravité des scénarios de menace, la vrasembalance et le niveau du risque. Le tableau 2.4 présente l'échelle d'évaluation de la gravité des évenements rédoutés.

Niveaux de l'échelle	**Description détaillée de l'échelle**
1. **Négligeable**	Impacts surmontés par le service sans aucune difficulté
2. **Limitéé**	Impacts surmontés par le service avec quelques difficultés
3. **Importante**	Impacts surmontés par le service avec de sérieuses difficultés
4. **Critique**	Impacts non surmontés par le service (son bon fonctionnement est menacé)

Tableau 2.4 – Métrique de gravité des événements redoutés

Le tableau présente 2.5 l'échelle pour estimer la vraisemblance des scénarios de menace et des risques.

Niveaux de l'échelle	Description détaillée de l'échelle
1. Minime	Cela ne devrait pas se (re)produire
2. Significative	Cela pourrait se (re)produire
3. Forte	Cela devrait se (re)produire un jour ou l'autre
4. Maximale	Cela va certainement se (re)produire prochainement

Tableau 2.5 – Echelle de vraisemblance des scénarios de menace

Les risques associés à diverses menaces sont identifiés à l'aide de paramètres de risque tels que la vraisemblance (possibilité) du risque, le seuil de risque et l'impact du risque.
La **vraisemblance du risque** fait référence à la vulnérabilité du système. Elle indique si le système sera attaqué par une menace ou non.
L'impact du risque fait référence à l'effet du risque sur le système. Cela dépend de la matrice de dépendance qui indique dans quelle mesure un certain risque affectera le système.
La valeur du risque fait référence au produit de la vraisemblance du risque et de son impact. Le tableau 2.6 indique les métriques (seuils) pour l'évaluation du risque dans EBIOS.

Sur la base de la valeur, le risque sera comparé au seuil de risque et cet état se verra attribué des risques intolérables (élevés), Significatifs (moyens) et

Gravité	1.Négligeable	2.Limitée	3.Importante	4.Critique
Vraisemblance	1.Minime	2.Significative	3.Forte	4.Maximale

Tableau 2.6 – Métriques (seuils) pour l'évaluation du risque dans EBIOS

négligeables (faibles) (Cf. Figure 2.4).

Niveau du risque	Acceptabilité du risque	Intitulé des décision et des actions
Negligeable	Acceptable en l'état	Aucune action n'est à entreprendre
Significatif	Tolérable sous controle	Un suivi en terme de gestion de risque est à mener et des actions sont à mettre en place, dans le cadre d'une amélioration continue sur le moyen et long terme
Intolérable	Innacceptable	Des mesures de réduction du risque doivent impérativement être prises à court terme. Dans le cas contraire, tout ou partie de l'activité sera réfusé.

Figure 2.4 – Echelle d'évaluation du niveau de risque

La section suivante présente le processus de gestion de sécurité basé sur le risque qui se nourrit de l'approche d'analyse des menaces et d'évaluation des risques présenté dans la section courante.

2.4 Processus de gestion des risques de sécurité

Cette section propose un processus pour la prise en compte de la sécurité durant le cycle de développement du système logiciel. Il constitue un guide permettant de supporter, étape par étape, les activités relatives à la définition de la sécurité pendant les différentes phases du cycle de développement logiciel.
La méthode de gestion des risques de sécurité proposée ici (cf. Figure 2.5) répond à la définition ci-dessus. Elle est conforme à la norme ISO 27005 [74] et les normes de gestion des risques [10] définies par l'ISO[74].
Cette méthode est un guide pour la construction du méta modèle du risque métier en exécutant chaque action du processus avec la base des concepts utiles à sa représentation du méta modèle.

Le processus met l'accent sur la façon dont l'architecture de sécurité d'entreprise TOGAF peut prendre en charge chaque action du processus des scénarios de service métier et le SI supportant le métier. Voici la description de chaque action :

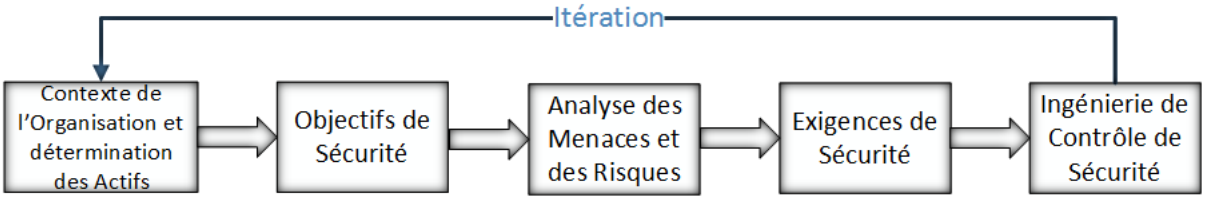

FIGURE 2.5 – Processus de gestion des risques de sécurité

a. Contexte de l'organisation et Identification des actifs : Le "Module 1 : Définition du Contexte" de Ebios est utilisée dans cette étape. Elle consiste à connaître le domaine de l'organisation, son environnement, en déterminant précisément ses limites et en identifiant ses ressources, ses actifs et ses services. Un "actif" est considéré comme tout ce qui a valeur à l'organisation et contribue à la réalisation de ses objectifs. Les actifs sont distingués en actifs métiers (processus métiers, informations, services, compétences et capacités) et les actifs du SI résument une composante du SI qui soutiennent les actifs métiers [58].
Cette étape se nourrit des phases préliminaire et phase A (Vison) de TOGAF associées au concept de sécurité. La phase préliminaire de (TOGAF) établit le contexte de sécurité requis pour guider la conception architecturale. Pour créer le contexte de sécurité, les artefacts de sécurité suivants doivent être déterminés au cours de cette phase : Moteurs métiers/Objectifs métiers, principes de sécurité, appétit pour le risque, domaines de risque clés/analyse de l'impact sur l'entreprise , plan de ressources de sécurité [130]. Ces artefacts peuvent être intégrés dans une documentation architecture existante.

b. Objectifs de sécurité : les préoccupations de sécurité des parties prenantes seront généralement exprimées sous la forme d'« objectifs de sécurité » abstraits qui font partie des objectifs non fonctionnels. Les objectifs de sécurité, également appelés critères ou propriétés de sécurité, sont des critères qui servent d'indicateurs pour évaluer l'importance d'un risque [104]. Les critères sont là

pour garantir que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu. Chaque actif peut avoir un ou plusieurs critères de sécurité selon le contexte d'utilisation et un niveau de gravité selon le scénario de risque.
Les objectifs sont généralement définis en terme de confidentialité , intégrité, disponibilité (confère EBIOS).
Aux fins de cette présentation, nous considérons que des concepts tels que la responsabilité et la non-répudiation peuvent être classées dans la catégorie intégrité, tandis que l'authentification est nécessaire pour garantir la confidentialité et l'intégrité, et enfin l'anonymat ou l'inobservabilité font partie de la confidentialité (Cf. méthode STRIDE).
Dans le contexte de l'architecture de l'entreprise, les objectifs de sécurité d'un SI sont définis à l'aide critères sur les actifs de l'entreprise (par exemple, la confidentialité et l'intégrité des données dans paiement des achats en ligne). Le résultat de cette étape est l'enrichissement du méta modèle de sécurité avec les critères de sécurité des actifs de l'entreprise en précisant le poids de réalisation de chacun.

c. Analyse des menaces et des risques : La menace décrit tout événement dans le contexte des actifs spécifiques de l'entreprise (actifs métiers, actifs SI), pouvant violer les objectifs de sécurité (propriétés) définis. Dans notre approche, nous utilisons la méthode STRIDE pour déterminer les menaces ciblant le système, Ebios pour évaluer les évènements rédoutés, les scénarios de menace et estimer le niveau de risque. .
d. Exigences de sécurité : les exigences de sécurité sont les besoins de sécurité pour traiter les risques identifiés. Elle est définie par la décision sur la manière de traiter le risque conçue comme décision de traitement du risque.
Il existe quatre types de mesures (décisions) pour traiter le risque : atténuation (mitigation) ou réduction du risque, évitement (avoidance) du risque, transfert (transfert) de risque et acceptation (acceptation) du risque. Les décisions d'atténuation (réduction) des risques conduisent à des stratégies et mécanismes de sécurité. Éviter les risques ou accepter les risques n'ont pas besoin d'exigence de sécurité alors que la décision de transfert de risque nécessite parfois, certaines

exigences de sécurité concernant les parties tiers.
Le (tableau 2.1) présente un guide pour l'activité des exigences de sécurité basée sur la méthode des menaces STRIDE.
e. Ingénierie du contrôle de sécurité : La dernière étape du processus de gestion de la sécurité est la définition de contre-mesures appropriées pour faire face aux risques identifiés. Le contrôle (également appelé contre-mesure ou sauvegarde) est un moyen conçu pour améliorer la sécurité, spécifié par une exigence, et mis en œuvre pour s'y conformer. Les contrôles de sécurité peuvent être un processus, des politiques, des appareils, des pratiques, des mécanismes ou autres moyens, utilisés pour atténuer, transférer, accepter ou éviter les risques. Les contrôles de sécurité possibles peuvent être dérivés des meilleures pratiques ou de listes de contrôle de la sécurité et doivent être évalués pour déterminer leur contribution à la réduction du risque global. L'ingénierie du contrôle de sécurité concerne la phase d'implémentation du système au niveau de l'architecture physique et utilise les mécanismes de sécurité correspondants aux services de sécurité préconisés au niveau logiques.
Le processus est itératif : Il est clair que l'introduction de contre-mesures peut introduire de nouveaux risques qui nécessitent une nouvelle analyse et évaluation. Les contre-mesures peuvent également introduire de nouveaux éléments de modèle dans l'architecture d'entreprise et modifier les dépendances existantes ou en créer de nouvelles.
Pour évaluer les effets des nouvelles contre-mesures une instance de l'architecture d'entreprise actuelle peut être créée et utilisée pour modéliser les changements architecturaux introduits par la nouvelle contre-mesure. Cette approche consiste à proposer une nouvelle solution de sécurité [73].
Une solution de sécurité est composée de plusieurs contrôles de sécurité atomiques. Chaque contrôle de sécurité atomique est directement lié aux menaces qui sont adressées.
La solution proposée est modélisée comme un scénario qui ne modifie pas l' architecture d'entreprise existante mais fonctionne sur un modèle temporaire dupliqué de l'architecture d'entreprise qui peut être utilisé pour dériver le delta des changements architecturaux introduits par la contre-mesure.
Il est possible de définir plusieurs solutions de sécurité différentes qui s'attaquent

aux mêmes menaces pour fournir des alternatives pour le processus de prise de décision. A cet effet, le coût des contrôles de sécurité est à estimer.
Si une solution de sécurité est approuvée pour la mise en œuvre, les modifications modélisées dans le scénario sont appliquées à l'architecture d'entreprise. Cela nécessite de vérifier les conflits potentiels si différents scénarios de contre-mesures fonctionnent sur les mêmes pièces de modèle. Ces conflits doivent être résolus en un effort coordonné des parties prenantes concernées.

2.5 Approche de construction du modèle contextuel de risque métier

Notre approche, fondée sur TOGAF, sa méthode ADM, et SABSA, cible la construction d'un méta modèle contextuel de risque métier enrichissant le modèle CIM du MDA.
L'architecture métier de TOGAF, correspondant à la (phase B) de l'ADM, modélise les processus/services métiers, les rôles, les responsabilités et les structures. Il reflète la sécurité du métier et la façon dont la sécurité de l'information est liée au fonctionnement de l'organisation.
Le modèle métier de sécurité de TOGAF est l'extension du modèle métier TOGAF ADM qui décrit les aspects de sécurité du domaine métier de l'EA [130]. La phase B aide à localiser les acteurs légitimes qui interagissent avec le produit/service/-processus. Les artefacts d'architecture métier liés à la sécurité sont enrichis par les architectures contextuelles et conceptuelles du modèle SABSA[149], correspondant à la phase de stratégie et planification dans son cycle de développement.

Graphique supprimé pour respecter le droit d'auteur
FIGURE 2.6 – Artefacts de sécurite dans la Phase A TOGAf : Archtecture métier

Les livrables (cf. Figure 2.6) de ce modèle métier de sécurité comprennent l'architecture de la politique de sécurité, le modèle de domaine de sécurité, le cadre de confiance, l'évaluation des risques, le registre des lois et règlements applicables, le registre du cadre de contrôle applicable [130].

Le modèle du métier concerné par nos travaux est le service métier tel que défini dans le méta-modèle d'architecture métier TOGAF.
Ce modèle est spécifié par l'analyste de risques métiers et l'expert en sécurité. L'analyste métier spécifie les services métiers et les informations qu'ils utilisent. Une collaboration avec l'expert en sécurité permet ainsi d'obtenir le modèle de service métier enrichi par un contexte de sécurité.
Le langage UML permet de décrire divers aspects du domaine métier associés aux risques de sécurité. Ainsi les aspects métiers de sécurité peuvent être intégrés aux modèles CIM dans MDA.
La section suivante consacre la construction du méta modèle métier de sécurité .

2.6 Méta modèle de sécurité du risque métier : TCM-BR

L'application de contraintes pour définir des modèles est bien connue dans la littérature. Une telle approche peut, par exemple être liée au contexte pour concevoir des modèles de services informatiques [2] [157][156].
Néanmoins, l'approche pour définir les modèles contextuels doit être étendu lorsqu'un processus, et non le comportement de l'utilisateur, définit les contraintes. MDE convient à mettre en œuvre une telle extension, compte tenu de ses possibilités de généralisation [167].
L'extension des mécanismes de l'approche MDA que nous proposons est donc une méta-modélisation du contexte, représentant des concepts utiles à la description de contexte, tel que celui de sécurité pour l'architecture métier, et permettant d'automatiser l'intégration du contexte dans un modèle via une transformation.
Le méta modèle appelé TCM-BR (Transformation Contextuel Model - Business Risk) (voir Figure 2.7) est défini par un expert en sécurité du risque pour influencer le processus de développement du système logiciel dès le début (CIM : besoins métiers). TCM-BR est composé des éléments suivants :
- (*OperatingCondition*) : la condition des opérations métiers, contrainte par une contrainte métier (*businessConstraint*). La condition des opérations est contrôlée

par le cas de contrôle (*ControlCase*) qui définit le contrôle lié à l'opération métier qui est une contrainte de sécurité. Il est caractérisé par le risque associé à l'opération ; la mesure d'atténuation du risque et le mode synchrone ou asynchrone du traitement du risque.

Figure 2.7 – TCM-BR : Méta modèle du risque métier

"Risk" est un risque identifié à partir de l'analyse des menaces dans l'activité de l'organisation dans le contexte de scénarios métiers. Le risque est la combinaison d'une menace avec une ou plusieurs capacités de vulnérabilité conduisant à un impact négatif nuisant à un ou plusieurs actifs. Des menaces et les vulnérabilités composent le risque qui est un événement et l'impact est la conséquence.
La menace : Threat fait référence à la menace et est identifiée à partir du modèle de menace STRIDE définissant le type de menace.
Le traitement asynchrone ou pas du risque est la conséquence de l'évaluation du risque et des mesures de traitement requises. Une action de réduction (mitigation) comme mesure est prise ici et le mode de traitement Synchronuous est décidé selon que l'opération de traitement du risque soit en mode synchrone ou asynchrone.
Le modèle métier du risque est soutenu par un processus de gestion de risque qui est un guide pour son instanciation.

2.7 Processus d'intégration du métier du risque dans l'architecture du système : CIM2CICM-R

Dans l'approche Model-Driven Engineering (MDE), l'un des principaux bénéfices attendus est la possibilité de réutiliser facilement les modèles déjà créés

pour concevoir de nouvelles applications. Le fait qu'un modèle contextuel de sécurité impacte le cycle de vie de différents systèmes et que ce modèle contextuel évolue, un problème à résoudre est de faire évoluer les modèles de manière cohérente dans le temps conformément aux méta modèles les décrivant.
Le contexte doit être intégré dans une application lors de la phase de modélisation pour permettre une intégration fluide à l'intérieur de l'application et faciliter l'évolution dans le temps.
Une solution pour intégrer un modèle contextuel est d'enchaîner les transformations, en utilisant des langages de modélisation pour saisir les modèles pour le service métier [173] , ou en adoptant plusieurs langages de transformation [45]. C'est une solution qui prévaut pour définir l'architecture du système en appliquant des contraintes graduelles, en affinant les spécifications initiales du système [36]. Ainsi dans un système orienté modèle, la définition de l'architecture peut être basée sur le raffinement conformément à MDA [177].
Le modèle contextuel doit donc être étudié aux premiers stades (CIM : exigences des besoins métiers) de l'analyse du système avant les activités d'architecture du système, pour être intégré dans un processus de développement conforme à MDA.
La connaissance correspondante des éléments d'entrée de la transformation permet à un expert de formuler des règles associant un contexte à un processus de développement système. La figure 2.8 décrit le processus de transformation du modèle CIM vers le modèle CICM-R nommé (CIM2CICM-R). Il implique les modèles CIM, TCM-BR, le CICM-R contextuel et le CICM-R.

La Transformation Contextuelle des Modèles par amélioration (Contextuel Transformation by Enhancement : CTe), comme définie dans [156] [157] est celle utilisée pour l'intégration des modèles au niveau métier (cf. Figure 2.8).
CTe permet d'enrichir le CIM avec le modèle métier du risque de sécurité TCM-BR. Le résultat de cet allignement (enrichissement) du CIM est le "Modèle Contextuel du Risque Indépendant de la Conception" libellé CICM-R (CIM-contextualisé par le risque métier).
L'ensemble des règles de transformation est défini par un expert du métier. Une description des modèles impliqués est donnée ci-après.

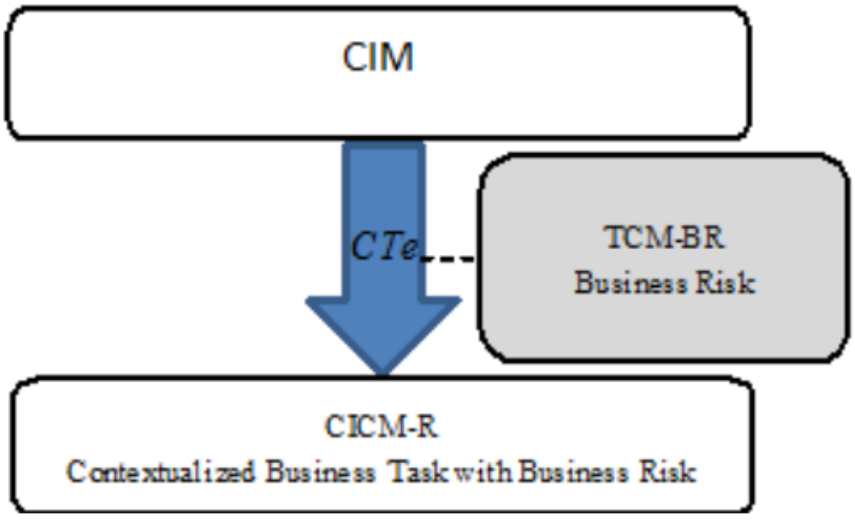

FIGURE 2.8 – Processus de transformation du modèle CIM vers le modèle CICM-R (CIM2CICM-R)

Méta modèle CIM Le CIM est un modèle de service métier «*BusinessService*» (le même concept que dans le méta-modèle TOGAF) qui décrit un service aux entreprises (*BSOnlineShopping*). Dans le méta modèle CIM, les concepts ciblent la description d'une tâche métier composant un service métier. Le service métier *BusinessService* est composé de tâches métiers désignées par "*BusinessTask*". «*BusinessTask*» spécifie le numéro d'ordre d'une tâche composant le service métier (1 pour la tâche métier Lire identifiant et mot de passe, de l'utilisateur, (*BSReadCustomerCredentials*) du service d'achat en ligne et (2 pour la tâche Créer la session pour l'utilisateur avec la date d'ouverture de la session (*BSCustomerCreateSession*). Ces concepts peuvent prendret en compte certaines contraintes facilitant la génération du code des services.

Transformation CIM vers CICM- R (CIM2CICM- R) Le méta-modèle contextuel de Sécurité est le résultat de contextualisation de la transformation d'alignement par enrichissement CTe du modèle CIM décrit précédemment et du modèle du risque TCM-BR. La spécification d'un service métier du CIM est réalisée par un expert métier et un expert en risques de sécurité avec l'instanciation du modèle contextuel de sécurité métier, qui relie une instance de tâche métier "Business Task" et une instance de risque métier «*BusinessRisk*» (Ex :

Lire les informations de connexion et mot de passe (*BSReadCustomerCredentials→BSCustomerAuthentification*) lié à l'authentification de l'utilisateur).
Le méta-modèle CICM-R (cf. Figure 2.9 illustre cette transformation.

Le méta-modèle CICM-R (CICM – Risk) C'est le résultat de la transformation CTe qui montre une relation de mappage entre une tâche d'un service métier et les risques métiers associés. Il est composé des modèles CIM et TCM-BR et décrit l'architecture de conception du métier (cf. fig. 2.9).

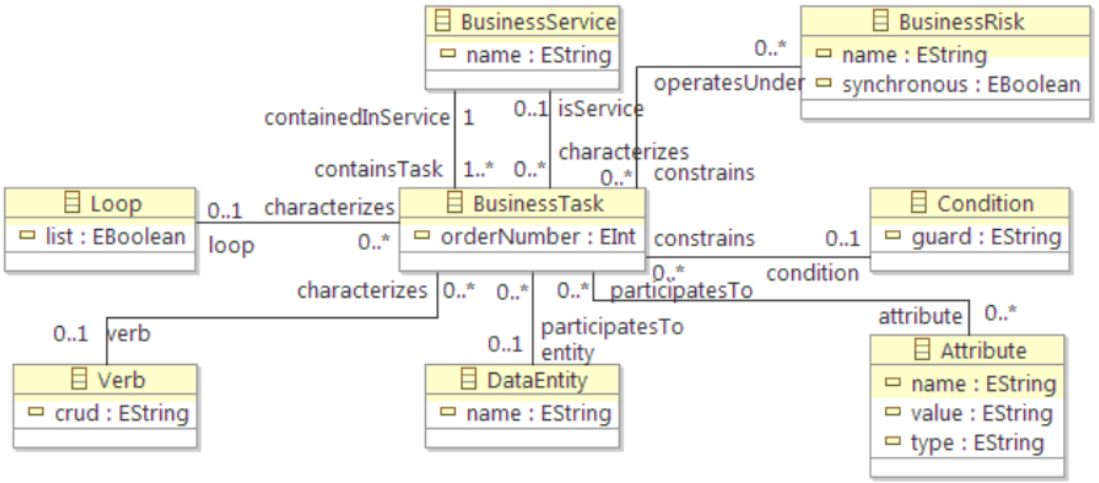

FIGURE 2.9 – CICM-R : Architecture de Conception du Métier (CIM et TCM-BR)

Le méta modèle est composé d'un service métier (*Business Service*) composé de tâches métiers (**Business Task**) identifiées par un numéro d'ordre (*orderNumber*). une boucle (*Loop*) caractérise la liste des différentes tâches où chaque tâche manipule un attribut (**Attribute**) typé (**type**) ayant un nom (*name*) et une valeur(*value*). Une tâche est associée à un objet métier (**Data Entity**) et peut être contrainte par une condition (**Condition**) de protection (*guard*). La tâche est enfin associée au concept de **BusinessRisk** qui fait le lien entre une tâche métier et le risque représenté par son nom (*name*) et le mode synchrone ou asynchrone *synchronous* qui sont dans le concept ControlCase du méta-modèle de TCM-BR (cf. Figure 2.7). La spécification d'un service métier du risque CIM-R est réalisée par un expert métier et un expert en sécurité.

2.8 Illustration de l'intégration de la sécurité au niveau du modèle métier par transformation de modèles

Tout au long de notre travail, les instances de description des modèles sont illustrées par un scénario de deux tâches d'un service d'achat en ligne effectué par un client (BSOnlineShopping) :
1. Read customer login and password (Lire identifiant et mot de passe, de l'utilisateur, qui existent)
2. Create a customer session with the openning date (Créer la session pour l'utilisateur avec la date d'ouverture de la session)
Ce scénario est une description simplifiée du contexte de l'entreprise et de l'identification des actifs conformément au processus de gestion des risques définis dans le courant chapitre.
Dans le cadre du traitement du risque, les exigences métiers sont complétées par l'association de propriété de sécurité à certaines tâches. Cette association d'exigences métiers de nature technique nécessite une collaboration entre expert métier et expert sécurité.

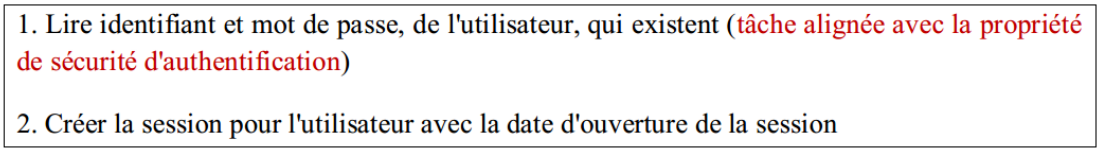

FIGURE 2.10 – Spécifications du service métier d'achat en ligne (BSOnlineShopping) : allignement entre tâches métiers et risques.

La propriété de sécurité d'authentification est associée à la lecture de l'identifiant et du mot de passe de l'utilisateur du portail de commerce en ligne(Ye et al., 2015). Cette propriété est requise pour assurer le bon fonctionnement de cette tâche. Cela est la deuxième phase de notre processus de gestion de la sécurité "Objectifs de sécurité". L'analyse des menaces à partir de STRIDE consiste à donner la correspondances de la menace à chaque propriété associée aux actifs comme valeur métier.

Elements	Spoofing	Tampering	Repudiation	Information Disclosure	DoS	Elevation of privelege
Data Flow		**x**		**x**		
Data Store						
Process	**x**					
Interactor	**x**					

FIGURE 2.11 – Analyse du diagramme DFD du service metier

A partir du diagramme des flux de données (DFD) (cf. Figure 2.11) nous effectuons une analyse des menaces relatives à la tâche (1) dont l'exécution requiert une attention particulière à cause de l'interaction du système avec une entité (externe ici -le client). Car la lecture des identifiants et mot de passe implique un flux de données "Data Flow", une interaction d'entités "Interactor" et fait partir d'un processus de réalisation d'une activité (tache) métier. L'analyse consiste dans un premier temps à identifier les différentes menaces, puis les classifier et prioriser en fonction de probable impact sur l'entreprise. Ainsi une sélection des menaces est effectuée sur la base des objectifs métiers et de l'appétit au risque. Cette tâche est réalisée par un expert métier et un expert en sécurité.
Les menaces ciblant le service ici sont : "spoofing" violant l'authentification, "tampering" violant l'intégrité et "Information Disclosure" violant la confidentialité. Pour rappel l'authentification est nécessaire pour préserver l'intégrité et la confidentialité.
Ainsi la menace "spoofing" viole l'authentification et cible le bon fonctionnement de la tâche 1 (lecture des identifiants et mot de passe) a été analysée avec la méthode EBIOS comme le montre la figure 2.12.

Comme décrit dans la figure 2.12 le niveau de gravité du risque est à l'échelle 3/4 (Importante) avec une vraissemblance (2/4 Significative). L'évaluation du risque de "spoofing" porte à (3) le seuil du risque qui est 'important". Donc nécessite une mesure de traitement.
Le traitement du vol de données (Credentials spoofing) ciblant l'identifiant et le mot de passe de l'utilisateur nécessite une stratégie d'authentification comme mesure d'atténuation et doit être traité de façon synchrone avec leur lecture,

Evenement redouté	**Besoin de sécurité**	**Sources de menaces**	**Impacts**	**Gravité**
(Spoofing) Vol de données	Privé (Aut	Hacker	Impossibilité d'accéder à son portail Perte d'argent Poursuite judiciaire Perte de clients	3.Importante

Scénarios de menaces	**Sources de menaces**	**Vraisemblance**
Spoofing des identifiants utilisateurs	Hacker Virus ciblé ou non ciblé Employé malveillant	2. Signicative
Spoofing des données bancaires d'un utilisateur	Hacker Virus ciblé ou non ciblé Employé malveillant	3.Forte

Niveau de risque				
Gravité	1. Négligeable	2.Limitée	**3.Importante**	4.Critique
Vraisemblance	1.Minime	**2.Significative**	~~**3.Forte**~~	4.Maximale

FIGURE 2.12 – Analyse et évaluation des risques avec EBIOS

puisqu'elle en dépend.
Le tableau 2.7 présente une instanciation du modèle de risque métier (TCM-BR.) pour le service métier *BSOnlineShopping*.

L'approche MDA contextualisée permet d'intégrer par enrichissement les modèles représentant un contexte lié au métier au niveau CIM. Un modèle contextuel de transformation par amélioration CTe est utile pour cela. Afin d'intégrer des exigences métiers de nature technique telles que celles ciblant la sécurité du système à développer, le modèle de transformations TCM associés au risque sont pris en compte. A chaque étape du chaînage de la transformation, un nouveau modèle contextuel est créé par une intégration TCM pendant le processus d'amélioration, en tenant compte des modèle précédent.
Le résultat de la transformation contextuelle d'enrichissement CIM2CICM-R d'un modèle de service métier offre un lien entre certaines tâches métier du

Tâche du Service métier	Modèle de menace	Propriété de sécurité	Risque pour service métier	Synchronisation du traitement du risque
1.	Spoofing	Authentication	Credentials spoofing	Synchronous
			Unauthorized	Asynchronous
2.	Tampering	Integrity	Disruptive device	Synchronous

TABLEAU 2.7 – Instanciation du modèle de risque métier (TCM-BR) pour le service métier BSOnlineShopping

service et des risques établis au niveau métier.
La spécification textuelle du service métier BSOnlineShopping est reformulée à l'aide de balises. La reformulation est sous la responsabilité d'un analyste des exigences métiers. Dans l'étude de cas, la spécification de BSOnlineShopping (cf. Figure 2.13).

```
<Business Service><name>BSOnlineShopping</name>
<Business Task><order number>1</order number>
<Verb><crud>read</crud></Verb>
<Data Entity><name>User</name></Data Entity>
<Attribute><name>login</name><type>String</type></Attribute>
<Attribute><name>password</name><type>String</type></Attribute>
<Condition><guard><>null</guard></Condition>
<Operating Condition>Authentication</Operating Condition>
</Business Task>
<Business Task><order number>2</order number>
<Verb><crud>create</crud></Verb>
<Data Entity><name>Session</name></Data Entity>
<Attribute><name>opening</name><type>String</type></Attribute>
</Business Task>
</Business Service>
```

FIGURE 2.13 – Reformulation des spécifications du service métier d'achat en ligne (BSOnlineShopping)

La reformulation traduit la transformation des modèles CIM et TCM-BR

instanciés. Elle assure ainsi une traçabilité des spécifications métiers afin d'être conforme au méta-modèle du CIM décrivant le service métier (BSOnlineShopping) composé de tâches (1-2) et où chaque tâche manipule un attribut typé (type) d'un objet métier (Data Entity) et peut être contrainte par une condition (Condition), une boucle (Loop) ou une condition opérationnelle (Operating Condition) décrivant les spécification de sécurité.

Synthèse

Dans ce chapitre, nous avons présenté notre approche d'intégration de la sécurité au niveau métier basée sur l'architecture métier de TOGAF enrichie par SABSA et outillé par le MDA au niveau CIM.
Notre approche est similaire aux approches telles que [73], [58] [107] qui se fondent sur l'architecture métier de l'EA pour l'analyse et la conception du système. Comme [58] [107], nous avons défini un processus de gestion de risque de sécurité qui nous sert de guidage dans notre démarche. Mais la particularité de notre approche, est qu'elle est basée sur des méthodes pratiques combinées d'analyse de menace (STRIDE) et d'évaluation du risque avec EBIOS. Cela permet d'élargir la base de connaissance et une meilleure prise en compte du risque. Si [73] établi un graphe de dépendance entre les architecture métier et logique pour la prise en compte des besoins métiers (fonctionnel et de sécurité), nous utilisons l'approche de transformation de modèle promu par l'IDM. Pour cela nous avons défini un modèle contextuel de sécurité basé sur le risque qui intègre le modèle métier par transformation contextuel CTe pour enrichir le modèle métier cible conforme au CIM de l'architecture MDA. C'est là, toute la nouveauté de notre méthode par rapport à celles citées.

2.9 Conclusion

Dans ce chapitre nous avons présenté notre modèle d'intégration des exigences de sécurité basée sur l'analyse du risque métier. L'approche basée sur le service métier décrit par le framework TOGAF intègre les exigences de sécurité à l'aide de l'approche MDA au niveau d'abstraction CIM. Le résultat est un modèle contextuel de sécurité dirigé par un processus de sécurité, conforme aux normes

et méthodes de gestion du risque.
Le mécanisme d'intégration basé sur la transformation des modèles par enrichissement permet un alignement entre les artefacts métiers et de sécurité du modèle résultant. Une assistance automatisée de la transformation assure la dérivation des spécification d'exigences métiers (y compris la sécurité) en exigences au niveau de l'architecture logique supportant le métier.
Intégrer les exigences de sécurité dans les premières étapes du développement logiciel peut améliorer considérablement les aspects importants de "Sécurité" des SI à partir du métier. Cela requiert une collaboration entre l'expert métier d'EA et expert en sécurité afin de résoudre les problèmes de spécifications d'exigences fonctionnelles et techniques (sécurité) et leur intégration. Cette collaboration permet de résoudre les conflits liés aux choix et décisions dans cette étape, qui peuvent impacter négativement la conception architecturale.
Notre effort est une contribution à la contrainte d'intégrer les exigences de sécurité pendant le processus de modélisation du service métier pour les applications. Elle apporte une solution (RQ : Reponse à la Question) aux questions de recherches (QR : Question de Recherche) suivantes de notre thèse (cf. Introduction Générale) :
.QR1 : Comment intégrer la sécurité au niveau métier dans le processus d'ingénierie système, via les risques ?

RQ-1 : En fournissant les éléments requis pour la construction d'un système de sécurité avec le cadre TOGAF et son extension pour la sécurité enrichi par le cadre de risque métier d'entreprise SABSA, et outillé par l'approche MDA.

• QR1.1 : Quels éléments de l'architecture métier peut-on cibler pour une intégration de la sécurité dans le système ?

RQ-1.1 : la prise en compte de la sécurité au début du développement logiciel (niveau métier) avec une activité de collaboration entre experts du métier et de la sécurité, pour un alignement des exigences métiers et de sécurité, guidé par un processus de gestion des risques.

• QR1.2 : L'approche MDA pour l'intégration de la sécurité au niveau métier est-elle pertinente comme mécanisme ?

RQ. 1.2 : l'approche MDA a permis d'exprimer les exigences de sécurité sous forme de contexte, avec le modèle abstrait CIM de son architecture. Cela résout la difficulté de traiter le caractère abstrait des exigences non fonctionnelles (sécurité) au niveau métier. Enfin MDA a permis d'enrichir un modèle de service métier par le contexte de sécurité défini par des risques. Et l'outillage avec XML a permis la réformulation des spécifications métiers, destinée à la modélisation et l'architecture logique du système.
Dans le chapitre suivant (chapitre 3) nous présentons le modèle contextuel de risque logique ciblant l'architecture logique du SI.

Chapitre 3

Intégration contextuelle du risque dans l'architecture logique du système

3.1 Introduction

Ce chapitre est consacré à notre seconde contribution qui consiste en l'intégration par un modèle contextuel du risque logique libellé TCM-LR (Transformation Contextual Model of Logical Risk), dans le processus de développement du système /logiciel. Le méta modèle représentant ce contexte étend l'approche UMLsec en ciblant l'architecture logique de l'EA, notamment utilisant les Framework TOGAF et SABSA au niveau d'abstraction PIM (Platform Independent Model) du MDA.
Ce chapitre traite des questions de recherche **QR2 : Quelle méthode de conception de la sécurité, via les risques, pour l'architecture logique supportant une sécurité intégrée dans le métier ? ;**
QR2.1. Comment intégrer la sécurité à la vue logique d'un système à partir de la vue métier ? ; et **QR2.2. Comment intégrer les composants logiques de sécurité à la réalisation dynamique des cas d'utilisation d'un système** ?.
Le chapitre est organisé comme suit : dans un premier temps, nous présentons l'étude du contexte et des travaux connexes, avant de proposer une approche de conception du méta modèle de risque logique que nous présentons par la suite. Ensuite nous décrivons le schéma du processus d'intégration du méta

modèle obtenu dans la phase de conception du système PIM par une série de transformation. Cette approche est illustrée enfin par le cas d'étude d'achat en ligne.

3.2 Contexte et travaux connexes

Les approches d'ingénierie de sécurité existantes, basées sur les modèles, au niveau de la conception du système, telles que Mal-activity diagram [159], ModelSec [142], UMLsec [83], secureUML [100], SECTET [1], UACML[160], SecureMDD [116], se concentrent sur la façon d'identifier, de capturer et de modéliser les objectifs et les exigences de sécurité du système qui doivent être appliquées dans le logiciel en cours de développement. Ces approches se concentrent sur la cartographie des exigences de sécurité identifiées au stade de l'ingénierie des exigences, sur les entités de conception du système (composants, classes, méthodes et interactions). Bien que bon nombre de ces approches soient basées sur l'approche MDA, la plupart sont dédiées à la modélisation d'un domaine spécifique (DSM : Domain Specific Modeling) et ne couvre pas un large spectre de préoccupations de sécurité, si ce n'est uniquement le domaine du contrôle d'accès (AC : Access Control) [50]. Cependant certains de ces efforts, comme UMLsec, sont multi-paradigmes (MPM : Multi-Paradigm Modeling)[56], permettent d'utiliser divers langages de modélisation spécialisés au niveau approprié d'abstraction pendant le développement du logiciel. Aussi UMLsec prend en charge l'analyse de sécurité formelle pour vérifier la satisfaction des propriétés de sécurité spécifiées.
En outre, peu d'entre ces approches disposent d'un ensemble d'outils qui aident, à générer du code de sécurité ou des configurations avec le code source du système. Et cela, en s'appuyant sur l'utilisation de techniques de l'IDM. La plupart de ces efforts, ne traitent pas la manière dont ces exigences de sécurité sont conçues et mises en œuvre dans ces systèmes. Ainsi, les développeurs de logiciels devront généralement créer ces exigences de sécurité avec les implémentations des fonctions métiers du système. Ce qui conduit rapidement à des systèmes avec des capacités de sécurité intégrées (codées en dur) qui sont souvent difficiles à modifier. Le codage en dur de cette sécurité dans les systèmes logiciels, com-

plique son intégration ainsi que celle de contrôles de sécurité tiers. Ainsi cette approche limite la flexibilité d'adapter la sécurité appliquée afin de répondre aux nouveaux défis de sécurité.
En somme, bien que les approches susmentionnées présentent des limites, elles disposent de concepts utiles et avantageux pour répondre au défi de l'intégration de la sécurité au niveau de la conception du système. En effet, l'approche de modèle, supporté par l'IDM avec son standard MDA, permet d'assurer la productivité, la traçabilité et la portabilité des modèles lors du développement du système, ainsi que leur interopérabilité et leur maintenance [89]. Elle permet de résoudre les problèmes de conception de solutions de sécurité (séparément des préoccupations fonctionnelles), destinées à être implémentées plus tard en les intégrant aux fonctionnalités métiers du système.
Cependant une intégration des besoins du métier (y compris la sécurité) avec l'architecture de conception du SI requiert une vue holistique du système à travers l'approche de l'EA, et notamment permettant, comme dans notre étude, d'aligner le métier et le SI.
Les approches étudiées y compris UMLsec n'offrent pas une méthode de développement du système de sécurité fondée sur un cadre de EA et l'IDM.
Plusieurs études d'intégration de la sécurité dans un cadre d'architecture d'entreprise (notamment au niveau de l'architecture logique du SI existant), telles que [12] [44] [73] [106][104][107][162][15][109], ont été menées.
Bien que couvrant, plus ou moins, tous les aspects de l'organisation, ces cadres ne disposent pas d'outils et/ou méthodes de développement du système dans le cadre d'EA, notamment basés sur les modèles.
Les travaux tels que ceux de Barateiro et al. [15], Oberperfler et Breu ([73], Gandry et al. [58], proposent des approches pour l'évaluation et l'analyse systématique des risques liés au SI dans l'entreprise et l'intégration des risques sous la forme d'une cartographie des concepts. Ces approches sont basées sur un modèle utilisant une architecture d'entreprise comme base. Ces approches proposent un alignement entre la gestion des risques et les activités d'architecture d'entreprise (métier, logique, physique et technologique) afin de fournir un soutien systématique pour cartographier et tracer les risques identifiés pour les artefacts modélisés dans une EA. Cet alignement des concepts est renforcé, notamment

par un arbre de dépendances des composants de l'architecture comme proposé par [73]. Cela permet d'établir un pont entre les points de vue logiques et métiers sur la sécurité de l'information.
Les trois approches précédentes sont similaires à la nôtre. Cependant notre approche utilise les concepts de l'IDM, notamment le MDA pour modéliser les entités métiers de l'entreprise basées plutôt sur le service métier (composé de tâches métiers), supporté par le SI.
Nous nous proposons de concevoir le modèle de sécurité logique sous forme de contexte, puis de transformer le modèle métier enrichi par le risque CICM-R en un modèle indépendant de la plateforme PIM enrichi par l'architecture logique du SI. Cette transformation inclu les composants logiques liés à la sécurité. Ce modèle est conçu avec le langage UMLsec, grace à une méta modélisation des Framework de l'EA (TOGAF et SABSA), permettant une extension de UML aux exigences de sécurité basées sur les menaces et les risques comme proposé au niveau du métier (Cf. chap. 2). Le méta modèle résultant, est un modèle contextuel par transformation du risque logique, dont la méthode de conception est décrite dans ce qui suit.

3.3 Processus de construction du méta modèle de risque logique

Le méta modèle contextuel du risque de sécurité logique est construit à partir de l'architecture logique d'entreprise TOGAF enrichie par la sécurité de risque. Le méta modèle étend les concepts d'UMLsec qui permet d'instancier les modèles de sécurité encapsulant les artefacts de l'architecture logique du système en développement.
L'architecture de sécurité logique correspond à l'architecture logique du SI au niveau de TOGAF relative aux risques de sécurité. Une description des exigences de sécurité est donnée dans la phase C de TOGAF[61]. Cette description est développée avec plus de détails dans le cadre SABSA [150], à travers son architecture de sécurité logique. La couche logique est largement concernée par la vision fonctionnelle de la sécurité, définissant un ensemble d'exigences fonction-

Graphique supprimé pour respecter le droit d'auteur
FIGURE 3.1 – Architecture logique de SABSA

nelles. Il ne prête pas attention à ce stade aux mécanismes de sécurité qui seront utilisés pour fournir ces fonctions. Elles font partie de l'architecture de sécurité physique(cf. Chapitre 4).
Dans le cycle de vie SABSA, pour le développement de l'architecture de sécurité, la phase logique concerne les activités intitulées "Design"(conception), qui englobent le design des architectures de gestion logique, physique, des composants et des services [149]. Cette activité de conception, exécutée sous la responsabilité d'un concepteur, produit un modèle cohérent et déterminé par lequel tous les éléments du système métier sécurisé s'assemblent. L'architecture logique de SABSA servant de base pour cette conception est décrite dans la figure 3.1.

L'architecture de sécurité logique décrit les relations et l'interdépendance entre les divers éléments du système. Elle traite du flux logique (composant logique utilisant les données logiques) d'une étape à l'autre dans le traitement sécurisé des informations du métier. Elle décrit également l'architecture de sécurité au niveau des entités logiques(données logiques) qui ont une identité, un sens, une structure, mais aucune représentation physique. Ces concepts sont encapsulés dans des composants logiques de sécurité, représentés graphiquement, sous forme de méta modèle étendant l'approche UMLsec.
UMLsec [83][83] définit des contraintes sur les éléments exprimant les règles de fonctionnement et la politique de sécurité, qui permettent l'évaluation des aspects de sécurité d'une conception du système. La spécification des menaces correspondent aux actions prises par l'adversaire. Ainsi différents scénarios de menaces peuvent être envisagés en se basant sur les capacités de l'adversaire.
UMLsec ne dispose pas de méthode explicite pour la modélisation de la sécurité, mais se base sur les méthodes standards de gestion de la sécurité pour satisfaire entre autre la non-répudiation, l'intégrité, l'authenticité, la sécurité du lien de communication, la sécurité des flux de communication .
L'extension de UMLsec est donnée sous la forme d'un profil UML utilisant les mécanismes d'extension UML standard[83]. Sur la base de ces concepts, nous

présentons dans la section suivante le méta modèle logique du risque.

3.3.1 Méta modèle Logique de sécurité du risque (TCM-LR)

Le méta modèle TCM-LR (Transformation Contextuel Model- Logical Risk) est le Modèle Contextuel de Transformation au niveau logique qui correspond aux éléments de sécurité dans l'architecture logique du Système d'Information du cadre d'architecture d'entreprise TOGAF. Les composants applicatifs logiques, les opérations logiques qui les composent, sont concernés par ce méta modèle. Il permet d'enrichir une tâche du service métier avec un composant logique du lié aux risques.
TCM-LR (cf. Figure 3.2) est le méta modèle contextuel de composants applicatifs logiques (Logical Application Component) du SI supportant les exigences métiers de nature fonctionnelle du métier ciblé et le risque (Risk) de sécurité identifié. A chaque composant applicatif logique modélisé, est associé le modèle de risque de sécurité.
La modélisation des interactions entre une opération logique supportant une

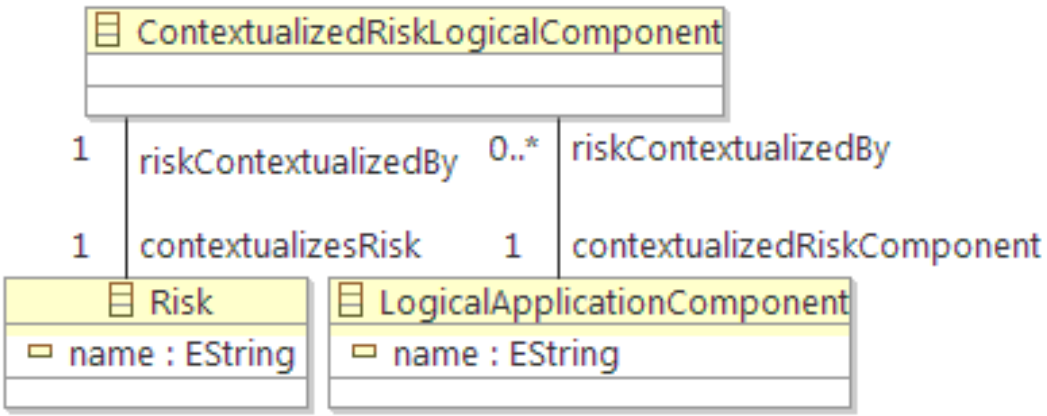

FIGURE 3.2 – TCM-LR : Méta modèle contextuel logique du risque

tâche métier et une opération logique liée au risque supportant un risque associé à une tâche métier avec le rôle de la synchronisation est illustrée à la figure 3.3.
TCM-LR comprend un composant d'application logique (*LAC : Logical Application Component*) qui est composé des opérations d'application logique (*LAO :*

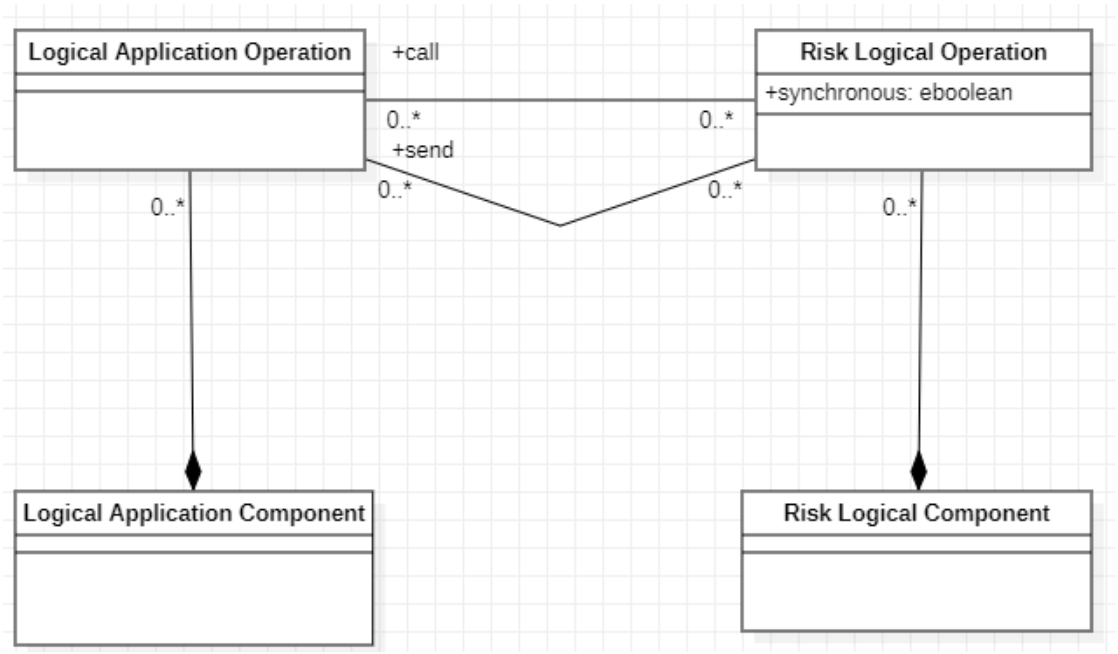

FIGURE 3.3 – Interaction entre Opération Logique du SI et le risque

Logical Application Operations) qu'il gère. Ex : (L'application logique LACSessionManagement est une "Gestion de session" "Session Management" qui gère l'opération logique *LAOReadCredentials* supportant la tâche métier "1. Lire identifiant et mot de passe, de l'utilisateur"). Les opérations d'application logiques (LAO) sont conçues à partir d'une tâche métier d'un attribut de donnée métier produit par cette tâche : (ex : LAOReadCredentials, l'opération logique de "lecture de l'identifiant et du mot de passe" de la tâche (1) est conçue à partir des attributs (login et mot de passe) de la donnée logique du client (user)).
Les Opérations d'Application Logiques sont liées aux Opérations Logiques de Risque (RLO) soit par une relation d'appel "call" ou de retour "send". En d'autre terme une relation de requête (request) et réponse (respond).
(RLO) détermine à travers le processus de gestion des risques, selon le caractère synchrone du traitement du risque, à quelle moment une opération d'application envisagée (LAO) déterminée à risque doit être prise en compte pour la décision de traitement préventive. Exemple :(*LOAReadCredential* est liée à *ProcessCredentialSpoofing* par la relation « call ») signifiant un appel préalable (synchrone) à la prise en compte du risque liée à l'opération avant son exécution.

Les RLO sont encapsulés dans les LAC liées au risque (RLC : Risk Logical Component).
Dans le contexte de l'architecture logique, les contrôles de sécurité sont encapsulés dans les services de sécurité [150]. L'architecture du Système d'Information (SI) comprend les services de sécurité fonctionnels et leur classification de sécurité. Les services de sécurité capturent les exigences de sécurité résultant des mesures de traitement du risque.
Dans TCM-LR les services de sécurité sont représentés par le RLC. Les services logiques de sécurité sont décrits par l'architecture logique de SABSA et présentés sous forme de catalogue. Le catalogue des services de sécurité est une liste de services qui fournissent des fonctionnalités spécifiques à la sécurité dans le cadre de l'architecture globale. La classification de sécurité est une étiquette attachée à un actif, selon un schéma de classification. Il détermine les exigences de sécurité applicables aux actifs. Par exemple, concernant le contrôle d'accès, la confidentialité ou la disponibilité. C'est un moyen de mettre en œuvre la politique de sécurité plus tard, au niveau physique.
Exemple : le service *LACRiskManagement* permet d'atténuer le risque *ProcessCredendetialSpoofing* en protégeant le composant *LACUserManagement*).

Instance du modèle logique de sécurité étendu au stéréotype UMLsec
TCM-LR (Logical Risk), est le méta modèle des composants applicatifs logiques (LAC : Logical Application Component) du SI supportant les exigences métiers liées au risque du métier ciblé. Une instance du méta modèle (TCM-LR) est un composant logique de sécurité stéréotypé «security» (étandant le composant logique «Component») dédié à la gestion des exigences de sécurité.
LACRiskManagement (cf. Figure 3.4), qui instancie le composant «security» contient les propriétés de sécurité (autorisation, authentication, confidentiality, integrity, availability, non repudiation…) associées aux menaces, définies par la méthode STRIDE (voir chapitre 2). Ce composant logique désigné est une fonction logique définissant un service de sécurité logique comme préconisé par SABSA. Ex : La menace spoofing, ciblant la tâche (read login and password of user), est associé au risque processCredentialSpoofing qui est associé à la propriété (exigence) de sécurité «*Authentication*».

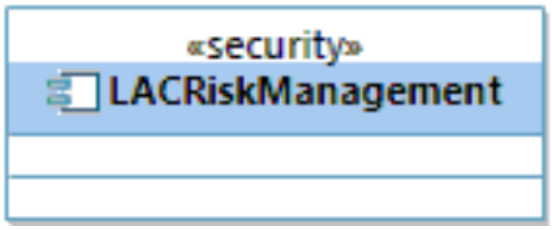

FIGURE 3.4 – Modèle d'architecture logique du SI technique supportant le domaine métier de l'achat en ligne.

Le service logique de gestion de risque **LACRiskManagement**, permettant d'atténuer le risque *processCredentialSpoofing*. Cela permet d'établir le couple (risk, RLC) et donc (*processCredentialSpoofing* , «*Authentication*»), qui à chaque risque, associe le composantc(service) logique du risque dédié à son traitement, fournit sous forme de cataloguqe par SABSA[149], selon la stratégie de défense envisagée.
Il faut noter qu'aucune dépendance de ces composants avec les composants applicatifs logiques supportant les exigences métiers de nature fonctionnelle n'est conçue a priori. Cette dépendance est réalisée par un arbre de dépendance de sécurité. En effet les propriétés de synchronisation de traitement du risque permettent une conception automatique de ces dépendances. Le choix pour le SI dédié au portail de commerce en ligne est la conception d'un seul composant supportant les exigences métiers liées au traitement du risque (cf. Figure 3.4).

3.3.2 Processus d'intégration du modèle de risque logique dans l'architecture du système : CICM-R2PIM

Dans la continuité de notre approche d'intégration des modèles contextuels (cf. chapitre 2), nous utilisons le concept du (MDE) afin d'intégrer le modèle de sécurité au niveau logique pour concevoir l'architecture logique de sécurité du SI.
L'intégration des modèles contextuels doit donc être étudié au stade (CICM-R : Computer Independent Contextuel Model- Risk) qui est le résultat du processus de transformation au niveau métier (architecture des exigences métiers et de sécurité du risque).
Conformément à l'approche MDA, nous dérivons les modèles vers d'autres

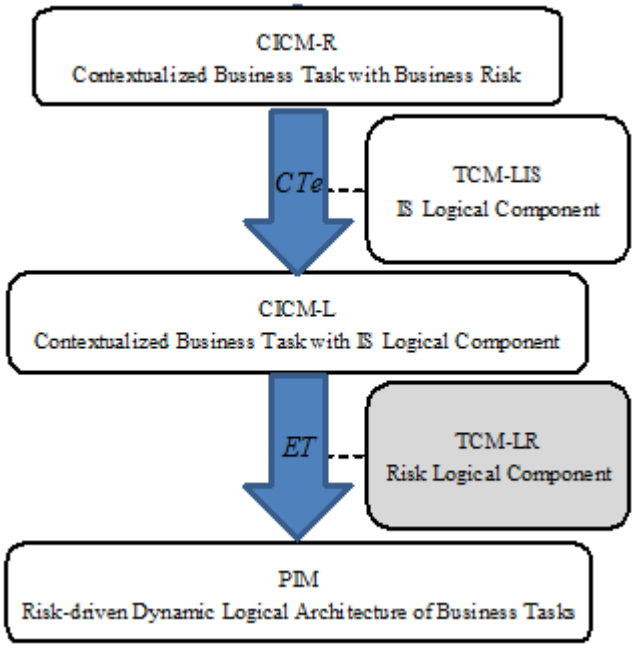

FIGURE 3.5 – Transformation du modèle CICM-R vers le modèle PIM (CIM-R2PIM)

modèles par la transformation appelée (model to model : M2M) [25], d'abord dans un même niveau d'abstraction CIM (transformation horizontale), grâce au processus de transformation contextuel CTe, ensuite nous obtenons le modèle résultant de la phase de conception par transformation des niveaux différents d'abstraction : du modèle CIM vers le modèle PIM (transformation verticale) grâce au processus de transformation par enrichissement « ET : Enhancement Transformation».
Comme nous pouvons le voir dans la figure 3.5, cette étape qui constitue la phase de conception de notre méthode est composée de deux transformations constituant le processus de passage du niveau CICM-R vers le niveau CICM-L, du niveau CICM-L vers le niveau PIM (Plateform Independant Model). Donc CICM-R2PIM = (CICM-R2CICM-L, CICM-L2PIM).

— **Processus de transformation CICM-R vers CICM-L (CICM-R2CICM-L)** CICM-R2CICM-L est la transformation contextuelle par enrichissement CTe (voir chapitre 2) du modèle CICM-R (modèle origine ou source) vers le modèle CICM-L (Computer Independant Contextuel Model- Logical)

ou «Modèle Contextuel du métier avec les composants logiques du SI » (modèle cible ou target).
La nature contextuelle de la transformation est l'intervention du modèle contextuel décrivant les composants logiques du SI représentés par TCM-LIS (Transformation Contextual Model of Logical Information System). TCM-LIS, modélise les composants applicatifs logiques du SI supportant les exigences métiers de nature fonctionnelle du métier ciblé. Le pattern (cf. Figure 3.6) utilisé pour cette conception repose sur une typologie des composants d'application logique fondée sur la durée du cycle de vie des composants [157]. Il indique qu'un composant de type «*activity*» dépend d'un composant de type «*person*», ou d'un composant de type «*document*» ou de type «reference», si cette relation supporte le métier de type «*activity*» et dépend du composant d'application logique *LACUserManagement* de type «*person*», et non l'inverse.

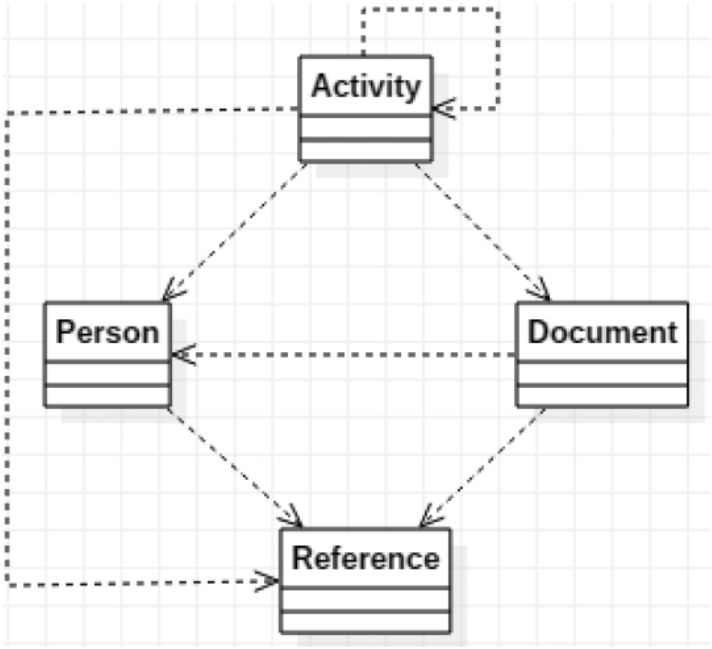

FIGURE 3.6 – Pattern de composant applicatif logique : TCM-LIS) [157]

CICM-L ou Modèle Contextuel du métier avec les composants logiques du SI, décrit le méta modèle résultant de la transformation, Modèle Contetuel de Composant Logique(voir Figure 3.7). C'est un méta modèle

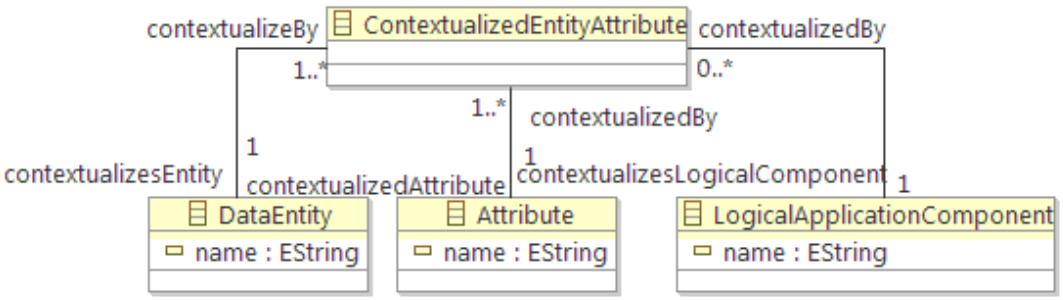

FIGURE 3.7 – CICM-L : Méta Modèle Contextuel de Composant Logique) [157]

contextuel du service métier associant les données métiers avec le modèle de composant d'applicatif logique TCM-LIS supportant les exigences métiers de nature fonctionnelle du métier ciblé.

A chaque composant applicatif logique (LogicalApplicationComponent) modélisé par TCM-LIS, est associé une donnée métier (DataEntity) qui est associée à un attribut (Attribute). Cet alignement est assuré par un expert en architecture logique.
Le descriptif de CICM-R2CICM-L est détaillé dans [157] et est en déhors des questions de recherche abordées dans cette thèse. Donc ce aspect n'est pas traité dans ce mémoire. Les modèles CICM-R, TCM-LIS et CICM-L sont ainsi des extensions du méta modèle CIM. Donc la transformation CTe du processus CICM-R2CICM-L est développé en utilisant ces méta-modèles qui formalisent les modèles origine (source) et cible (target).

— **Transformation CICM-L vers PIM (CICM-L2PIM)**
Dans cette étape, la transformation ET(transformation verticale) se fait du méta modèle CICM-L(modèle origine ou source) et vers le PIM (Modèle dynamique du risque métier avec les composants logiques du SI)(modèle cible ou target).
Le modèle d'entré (source) CICM-L est enrichi par TCM-LR décrit dans la section précédente.
Cette transformation du modèle logique du SI enrichi par le modèle de contexte de sécurité est décrit par le méta modèle PIM de la figure 3.8 .
Ainsi le PIM est le résultat de la transformation ET du modèle du CIM

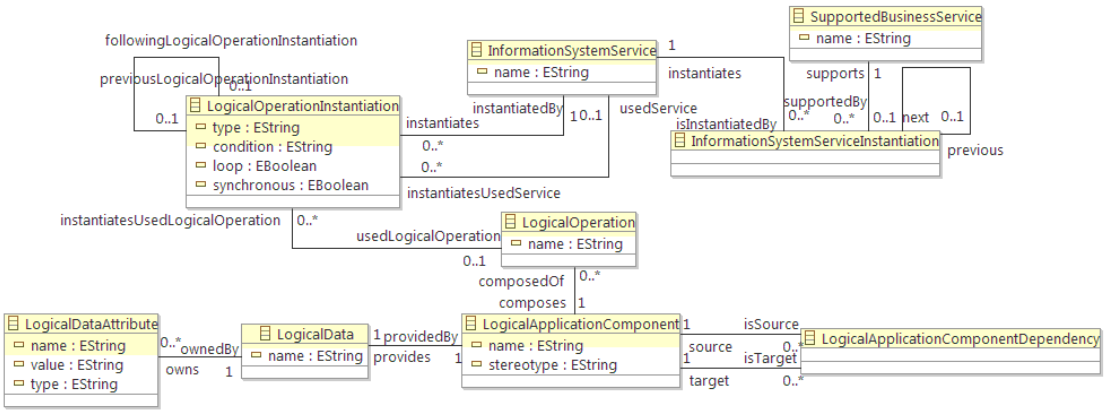

FIGURE 3.8 – PIM : Modèle des composants d'application logiques du SI associé à TCM-LIS (cf. [157])

contextualisé par le risque et par les composants applicatifs logiques du SI fonctionnel. Cette transformation est contextualisée par TCM-LR, de telle façon que chaque risque soit aligné avec le composant applicatif logique du SI technique qui le supporte. L'expert sécurité est le responsable de cet alignement.
Les modèles CICM-L, TCM-LR sont conformes au méta-modèle CIM tout comme le modèle PIM est conformes au méta-modèle PIM. Donc la transformation "ET" du processus CICM-L2PIM est développé en utilisant ces méta-modèles qui formalisent les modèles origine et cible.

— **Le niveau PIM**
Cette phase correspond au résultat de l'analyse et la conception dans le développement des systèmes. Le PIM définit l'architecture des composants applicatifs logiques, les données et les opérations qui satisfont les besoins d'un système ainsi que les exigences de sécurité associées. Le PIM qui est la phase de conception d'un système peut être vu comme l'implémentation abstraite du système sans entrer dans les détails techniques. Par exemple nous pouvons identifier les classes et ses interactions sans indiquer le langage de programmation utilisé pour les implémenter.
Le résultat du processus de transformation CIM-L2PIM est le modèle d'architecture logique dont le méta modèle est décrit dans la figure (3.8).

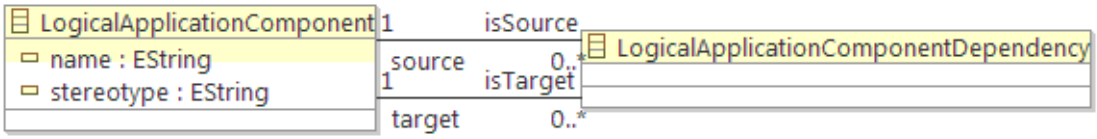

FIGURE 3.9 – Méta modèle de dépendance entre composants applicatifs logiques

L'architecture logique est la conception de composants applicatifs logiques où les composants du SI alignés avec les tâches du service métier sont complétés par les dépendances et les opérations logiques utiles au service métier.

— **Dépendance entre composants logiques du système**
La séquence de transformations CICM-R2CICM-L et CICM-L2PIM doit être capable de dériver la fonctionnalité ou le métier du système, et de conserver toute la sémantique du système, y compris les aspects non fonctionnels comme les aspects de sécurité. Pour obtenir une vue intégrée des modèles des différents niveaux de l'architecture d'entreprise, le service métier et le cycle de vie des composants logiques du SI utilisés sont modélisés en détail. Dans ce but, les activités individuelles (tâches/opérations) d'un service métier sont modélisées, décrivant le flux de contrôle et le flux d'information. De cette façon, nous sommes en mesure de définir des dépendances à travers les limites des niveaux de l'architecture par le biais d'un méta modèle de composants logiques permettant d'assurer la hiérarchie des dépendances entre ces composants. Cette hiérarchie signifie que les dépendances entre composants logiques supportent la séquence de tâches du service métier et respectent la règle de modélisation précédente et le pattern défini dans [157] qui supporte le modèle des composants d'application logiques du SI.

Une instance du méta modèle est un ensemble de services SI composant les LACs.
La définition d'un service SI est basée sur les propriétés d'un arbre de dépendance logique (Logical Dependency Tree : LDT) constitué de LACs et ce conformément à l'algorithme décrit dans[157].

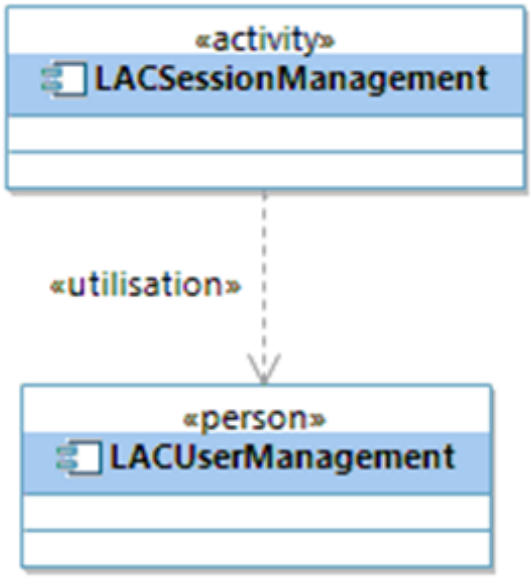

FIGURE 3.10 – Arbre de dépendance logique du service SI ISSOnlineShopping createSession opening

L'arbre de dépendances logiques permet de définir un service de SI tel qu'il existe un sommet LACroot à partir duquel tous les composants de l'arbre LDT sont accessibles par une chemin de dépendances (fonction path) et tel qu'il n'y ait pas de cycle.
La figure 3.10 représente le résultat de TCM-LIS contraignant le service métier BSOnlineShopping (cf. Figure 2.10).

Cette dépendance permet d'assurer la traçabilité des spécifications métiers, mais surtout de sécurité, définies en amont (architecture métier) et dérivées au niveau de l'architecture logique et qui pourront servir plus tard à la technologie.
La transformation ET permet d'intégrer dans l'architecture logique, des composants logiques traitant les risques. Elle assure aussi la sémantique de la dépendance entre les différents niveaux hiérarchiques (architectures métier et logique de l'EA) en implémentant les recommandations définies dans le modèle d'application logique. Ces recommandations sont les suivantes :

— Si une activité A1 est alignée avec un composant logique LAC1 et si

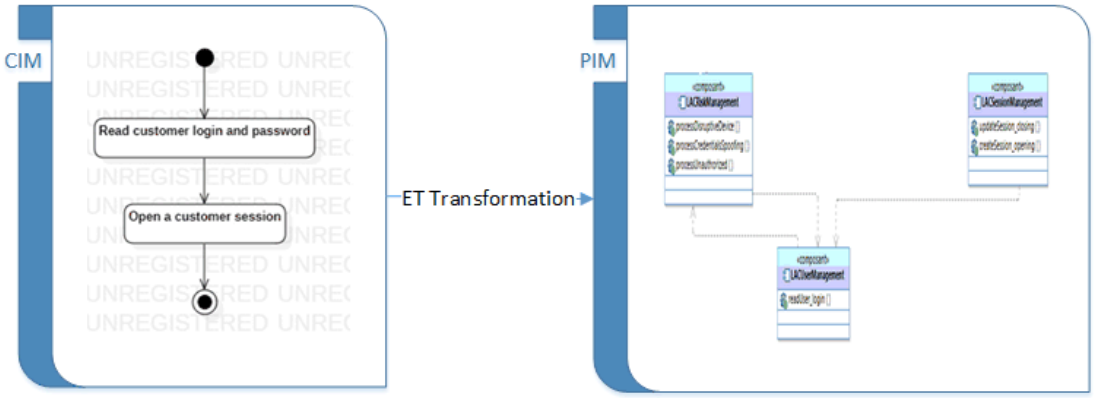

FIGURE 3.11 – Illustration du PIM resultant de la transformation par enrichissement ET.

une activité A2 est alignée avec un composant logique LAC2 ; si A1 précèdes A2, Alors LAC2 dépend de LAC1 dans le modèle PIM.

— Si dans CICM-R, une activité A1 est alignée avec le risque R1 Alors deux (2) cas sont possible :

1. LAC1 depend de LACRiskManagement (e.g. : LAOReadCredentials de LACUserManagement depend de LAOProcessR1 associé à "Credentials spoofing") ;

2. LACRiskManagement dépend de LAC1 (e.g. : LAOProcessR6 associé à la "non-autorisation" dépend de LAOReadCredentials géré par LACUserManagement)

La figure 3.11 présente la transformation "ET", illustrée par le diagramme d'activités (CIM, à gauche), décrivant l'illustration d'un scenario (de deux tâches) et le diagramme des composants (PIM, à droite) décrivant la gestion des composants d'application logiques (incluant le Composant d'Application Logique du risque) et leurs interdépendances. La figure 3.11 illustre les recommandations de l'alignement des activités des composants applicatifs logiques ainsi que des deux cas d'interdépendance des composants d'application logiques.

3.3.3 Illustration

Instance de CICM-L

Le tableau 3.1 est une instance du méta-modèle CICM-L, illustré par notre cas d'étude du service métier d'achat en ligne (cf. Figure 2.10).

Donnée métier (Data Entity)	Attribute	Composant applicatif logique
user	login	LACUserManagement
	password	
session	opening	LACSessionManagement
	closing	

Tableau 3.1 – Alignement des attributs des donnes métiers et des composants applicatifs logiques du SI

Ce modèle est le résultat de l'alignement de chaque attribut de donnée métier avec le composant applicatif logique modélisé dans TCM-LIS (cf. Figure 3.1) qui le supporte. Ex : les attributs *login et password* de la donnée métier *User* sont supportés par le composant applicatif logique **LACUserManagement**, et les attributs *opening* et *closing* de la donnée métier *Session* sont supportés par le composant applicatif logique **LACSessionManagement**.

Instance de CICM-R

De la même manière le tableau suivant décrit une instance du modèle logique de sécurité qui à chaque risque associé à une tâche métier, est associé un composant applicatif logique lié au risque et dédié à sa gestion.

Dans le cadre de notre étude de cas, un seul composant de gestion du risque *LACRiskManagement* permet la gestion des risques "Credentials spoofing", "Unau-

Risque	Composant applicatif logique
Credentials spoofing	LACRiskManagement
Unauthorized	
Disruptive device	

Tableau 3.2 – Alignement des risques associés et des composants applicatifs logiques du SI technique

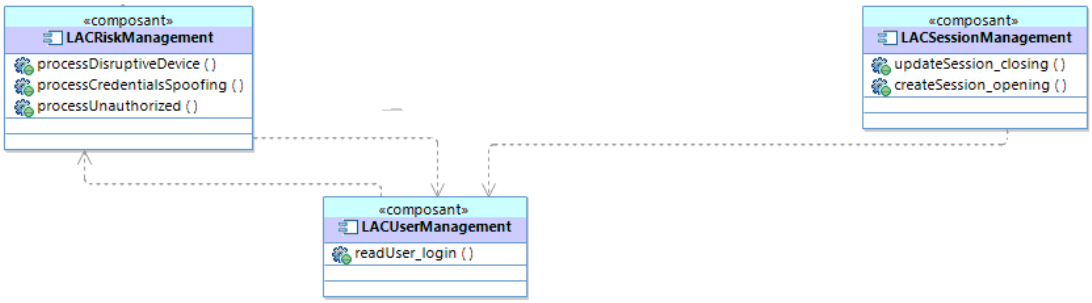

FIGURE 3.12 – PIM statique : Modèle de composants applicatifs logiques supportant le service métier BSOnlineShopping.

thorized" et "Disruptive device".

Instance du PIM

Le PIM est le résultat de la transformation CICM-R2CICM-L (CTe) et CICM-L2PIM (ET) du modèle du CIM avec le contexte de risque et des composants applicatifs logiques du SI fonctionnel. La transformation ET est implémentée avec le langage operational-QVT, recommandé par l'OMG (Object Management Group) [59]. Les modèles suivants sont le résultat de ET chainée avec une transformation implémentée avec operational-QVTv, du PIM vers UML2 afin de représenter le PIM sous sa forme statique et sous sa forme dynamique.

Le modèle statique de composants applicatifs logiques.

Le modèle de composants applicatifs logiques sont complétés par les dépendances et les opérations logiques. Ces opérations de ces composants sont utiles au service métier.

FIGURE 3.13 – PIM statique : Données logiques supportant le service métier BSOnlineShopping.

Le modèle de composants applicatifs logiques supporte le service métier BSOnlineShopping.
Dans l'étude de cas (cf. Figure 3.14), les dépendances concernant le composant LACRiskManagement, soit vers LACUserManagement pour supporter le risque asynchrone Unauthorized, soit à partir de LACUserManagement pour supporter le risque synchrone Credentials spoofing complètent automatiquement le modèle des composants logiques extraits du SI avec leurs dépendances pour supporter le service métier BSOnlineShopping. Chaque opération logique est conçue directement à partir d'une tâche métier et d'un attribut de donnée métier produit par cette tâche, qui permet grâce aux alignements précédents (Tableaux 3.1 et 3.2) de l'intégrer à un composant applicatif logique.

Le modèle de données logiques

La conception automatique des données logiques (cf. Figure 3.13) est fondée sur les données métiers, et leurs attributs, indiqués dans la spécification du service métier. La contrainte de cette conception est l'alignement avec les composants applicatifs logiques (cf. Tab. 3.1) puisqu'une donnée logique ne peut être produite que par un seul composant applicatif logique. Par exemple, la donnée logique LDUser est produite par le composant applicatif logique LACUserManagement. tout comme la donnée logique LDSession est produite par le composant applicatif logique LACSessionManagement.

Le modèle de données logiques où la conception des dépendances n'a pas été implémentée, puisqu'elle est conforme aux dépendances des composants applicatifs logiques qui les produisent.

Ainsi, une représentation des données logiques produites par les opération logiques peut donner des détails sur les opérations métiers et aider à identifier précisément les ressources concernées par les risques. Dans ce cas, le processus de traitement du risque peut être illustrer dynamiquement par un diagramme de séquence pour une meilleure analyse et prise en compte du risque.

La modélisation dynamique de l'architecture logique

La modélisation dynamique de l'architecture logique est représentée par les diagrammes de séquence associés, chacun, à un service du SI. Un service du SI étant défini par un service métier et un arbre de dépendances logiques, conformément à l'algorithme décrit dans [157]. Par souci de concision, dans notre illustration, l'exemple du modèle de composants applicatifs logiques représentés à la figure 3.10, fait apparaître un seul arbre de dépendances logiques avec pour sommet LACSessionManagement. La séquence de tâches du service métier (cf. Figure 3.14) est donc supportée, du fait des alignements (cf. Tableau 3.2), par le service du SI supportant les tâches 1 et 2 (arbre de sommet LACSessionManagement). Ci-après, le diagramme de séquence défini avec la conception d'un service du SI et par le support de tâches du service métier BSOnlineShopping. Le diagramme de séquence (cf. Figure 3.14) représente le service ISSOnlineShopping :createSession opening du SI qui supporte les tâches 1 et 2 du service métier.

La traduction du synchronisme du risque "Credentials spoofing" par rapport à la tâche de lecture des identifiant et mot de passe de l'utilisateur est notée par un ovale de couleur grise dans le diagramme de séquence. Il traduit bien la force du couplage entre la requête (call) de lecture des identifiant et mot de passe et le traitement du risque.

Concernant l'asynchronisme du risque Unauthorized par rapport à cette même tâche, il est souligné par l'ovale de couleur noire. Il traduit le fait que le traitement de ce risque nécessite d'avoir la donnée logique contenant identifiant et mot de passe en entrée. Ces deux représentations de dépendance entre une opération logique (readUser_login) et des traitements de risque (processCredentialsSpoofing et processUnauthorized) soulignent aussi le fait que la requête

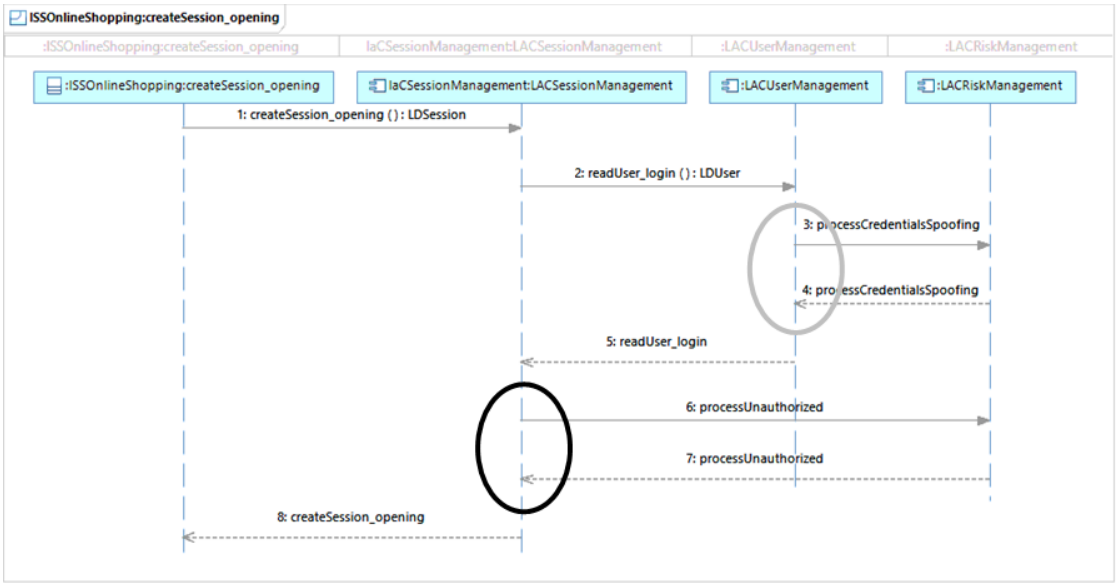

FIGURE 3.14 – PIM Dynamique : Modèle de composants applicatifs logiques supportant le service métier BSOnlineShopping.

d'un traitement de risque ne peut entrainer une requête d'une opération logique de type fonctionnel.

Synthèse

Nous avons présenté l'extension d'UMLsec, étendant le méta modèle TOGAF et SABSA, pour le développement de systèmes sécurisés, au niveau de la phase de conception, conforme au PIM de l'approche MDA. Cette extension, sous la forme d'un profil UML utilisant les mécanismes d'extension UML standard, est un méta modèle de sécurité logique du système, encapsulant les structures logiques de l'entreprise sous forme de composants (composés d'opérations) applicatifs logiques LAC (avec des opérations de nature fonctionnelle ou liée au risque). Les exigences de sécurité définies en amont (architecture métier, cf. Tab. 2.1), en se référant au scénario de menaces STRIDE sont héritées et définies sous forme de stéréotypes. Les contraintes de traitement synchrone ou non du risque sont adressées en association avec les appels "call" et réponses "send" entre composants liés au risque. L'intégration du métier et du SI par transformation des modèles enrichis, permet d'aligner les composants logiques supportant le service métier, aux tâches métiers avec les modèles de sécurités associés. Un arbre de

dépendance, défini par les services du SI composant le service métier, permet d'établir une orchestration (ordre d'exécution) des services concernés, et une traçabilité des spécifications définies dans l'architecture métier. On peut ainsi vérifier que les exigences de sécurité énoncées sont prise en compte pour chaque étape du cycle de développement du système.

Comme démontré, UMLsec peut être utilisé pour encapsuler des règles établies sur une ingénierie de sécurité, également en appliquant des modèles de sécurité, et en les mettant ainsi à la disposition des développeurs non spécialisés en sécurité. Afin de comparer notre approche à celle d'UMLsec, nous avons retenu un ensemble de critères permettant une caractérisation des approches :
- EA framework : l'utilisation de cadre d'architecture d'entreprise pour la modélisation du système.
– Exigences de sécurité traitées : le terme sécurité comporte plusieurs exigences sous jacentes comme la confidentialité, l'intégrité, la disponibilité, l'authentification, le contrôle d'accès.... Ce critère est évidemment essentiel à considérer pour identifier les approches couvrant le maximum d'exigences de sécurité.
- Approche de menaces : la prise en compte des menaces dans la modélisation de la sécurité.
-Approche de modélisation : l'approche utilisée pour la modélisation telle que le MDE.
-Modélisation de la Dépendance : permet d'établir une relation de dépendance entre les éléments du système.
– Support méthodologique : certaines approches proposent une méthodologie pour introduire la sécurité dans les systèmes ou les applications. Cette introduction peut être faite à travers l'approche MDA.
Le tableau (cf. Tab3.3) ci-dessous, résume la comparaison entre notre approche et UMLsec.

Critères	Approches	
	UMLsec	**Notre approche**
EA Framework	Non	Architecture Logique (TOGAF, SAB
Exigences de Securité	Plusieurs (stéréotypes)	Plusieurs (stéréotypes)
Approche de menaces	Générique (basée sur l'attaquant)	STRIDE
Modélisation de la Dépendance	Secure dependency	Arbre de dépendance
Concept de modélisation	Orienté modèle	MDE
Language de Modélisation	Plusieurs diagrammes UML	Diagrammes UML (Activité, classe, seq
Gestion de la sécurité	Sécurité primitive (cryptographie, authentification)	Gestion du risque (Services logiques de s
Support méthodologique	Non défini	MDA (intégration par transformati

Tableau 3.3 – Comparaison entre notre approche et UMLsec

Ainsi les deux approches adressent plusieurs exigences de sécurité sous forme de stéréotypes, permetttent d'établir une relation de dépendance entre les éléments du système, et modélisent le système avec le langage UML.
Cependant contrairement à UMLsec, notre approche modélise le système dans un cadre d'architecture d'entreprise avec l'approche de l'ingénierie Dirigée par les modèles basée sur l'approche méthodolgique MDA. Cela permet une séparation nette des préoccupations fonctionnelle et de sécurité, conduisant à la définition de modèle contextuel du risque dédié à l'architecture logique. Cette méthode à l'avantage de développer le système de sécurité par modules graduels à chaque phase du cycle de développement et les intégrer par transformation successive pour obtenir l'architecture logique de nature à la fois fonctionnelle, mais aussi technique.

3.4 Conclusion

Dans ce chapitre nous avons présenté le modèle logique de sécurité, qui est une méta modélisation du Framework TOGAF enrichi par SABSA avec le contexte de sécurité au niveau de l'architecture logique du SI. Le méta modèle est une extension de l'approche UMLsec, introduisant un nouveau stéréotype de composant "Security" d'application logique lié au risque. Ce composant, qui est un service logique de sécurité, permet de supporter les exigences de sécurité définies dans le processus de gestion de risque. Le challenge dans la phase de conception du système logiciel est comment assurer une vue cohérente du système en permettant une traçabilité des spécifications fonctionnelles et techniques du métier au niveau logique.
Les questions de recherche suivantes traitées dans ce chapitre, traduisesnt clairement ce défi. QR2 : Quelle méthode de conception de la sécurité, via les risques, pour l'architecture logique supportant une sécurité intégrée dans le métier ?
QR2.1. Comment intégrer la sécurité à la vue logique d'un système à partir de la vue métier ?
QR2.2. Comment intégrer les composants logiques de sécurité à la réalisation dynamique des cas d'utilisation d'un système ?
Pour répondre à ces problématiques (QR2 et QR2.1) nous avons utilisé l'ap-

proche MDE pour la représentation des modèles contextuels logiques de TOGAF enrichis par SABSA, conformes au modèle PIM de conception. Cette approche permet de réaliser les structures de l'organisation sous forme de modèles abstraits utiles à la réalisation de l'architecture logique du système dans une vue cohérente (QR2.2). Le mécanisme de transformation des modèles par enrichissement successif permet un alignement entre les artefacts fonctionnels et de sécurité à l'aide de relation de dépendance entre les éléments des architectures métier et logique (QR2 et QR2.1). Cette dépendance est représentée par un graphe de dépendance entre composants du système associant les dépendances de sécurité permettant d'adresser le risque de façon synchrone ou asynchrone. Une collaboration d'un expert en sécurité et en architecture logique d'entreprise fut utile à la réalisation de cette tâche (QR2, QR2.1 et QR2.2).
Le chapitre suivant est dédié à la phase d'implémentation des exigences métiers supportées par les fonctions logiques avec la prise en compte des besoins de sécurité.

Chapitre 4

Intégration contextuelle du risque dans l'architecture physique du système

4.1 Introduction

Comme mentionné dans les chapitres précédents, la modélisation métier et logique sont des modèles clés pour améliorer le développement des exigences de sécurité. C'est un bon moyen d'identifier les structures de l'entreprise, les objectifs et les actifs de l'entreprise et de maintenir la traçabilité à partir des actifs métiers jusqu'aux composants de sécurité du système informatique.
Après s'être focalisé sur l'analyse des actifs métiers et des composants logiques, il convient d'expliquer comment les sécuriser contre les risques auxquels ils sont exposés. La manière dont la sécurité sera mise en œuvre doit toujours être alignée en permanence avec les architectures métier et logique , mais aussi physique, comme préconisé. Cela permet d'apporter une réponse à la question de recherche **QR3 : Quel impact de la mise en œuvre de la sécurité via les risques dans l'architecture physique et les conséquences sur l'implémentation du système ?**.
Au niveau de l'architecture physique de la plupart des organisations, la gestion des actifs métiers repose fortement sur le système d'information et les ressources

(technologiques) informatiques. La plupart des processus/services sont principalement mis en œuvre dans des systèmes électroniques, et les informations sensibles de l'organisation sont stockées dans des bases de données et transmises sur les réseaux informatiques.
Ainsi, au niveau de la phase d'implémentation du cycle de développement du système, la question est de savoir comment assurer une traçabilité des spécifications fonctionnelles et de sécurité définies au niveau métier. Et comment la technologie doit supporter le métier dans une vue cohérente en intégrant les exigences de sécurité.
Le chapitre précédent a examiné les fonctionnalités et les flux logiques du système d'information. Ce chapitre propose une intégration d'un modèle contextuel de risque physique dans le cycle de développement du système logiciel.
Ainsi, une attention est portée sur les composants physiques, leur caractéristique et leur mode d'usage. Egalement, l'on se penche sur la structure physique des données qui sont utilisées pour réaliser les structures d'information logiques et les mécanismes physique de sécurité qui implémentent les services de sécurité logiques.
Du point de vue MDA, les modèles physiques peuvent piloter des outils de déploiement automatisé[146]. Les outils MDA peuvent générer des diagrammes de déploiement à partir de modèles PIM puis implémenter un mappage vers le modèle PSM, contraint par le PDM (la plateforme technique), pour générer un modèle de déploiement squelettique.
Dans la continuité de notre méthode basée sur MDA, le meta modèle physique de sécurité est modélisé dans les cadres TOGAF et SABSA, étendant l'approche UMLsec. L'utilisation de MDA comme outil de développement pour la phase d'implémentation permettant une intégration des modèles par une transformation par substitution des modèles définis avec des services physiques réutilisables.
Ainsi, ce chapitre présente, dans un premier temps l'approche de construction du model contextuel physique du risque de sécurité avant de décrire le méta modèle qui en résulte. Par la suite nous présentons le processus d'intégration du modèle de sécurité dans le processus de développement du système tout en présentant les différents modèles au niveau PIM et PSM du MDA qui y participent. Enfin une instance du méta modèle de sécurité est donné dans la phase

d'expérimentation en guise d'illustration.

4.1.1 Approche de construction du modèle contextuel de risque de sécurité physique

Le méta-modèle contextuel de risque pour le modèle physique recouvre les concepts décrivant un service du SI implémenté et réutilisable, c'est à dire son lien avec les opérations logiques réalisées et la plateforme sur laquelle il est déployé.
La construction du méta modèle est basé sur la méta-modélisation de l'architecture technologique de TOGAF ADM[62], enrichie par SABSA[149].
L'architecture technologique de TOGAF offre les artefaCTs pour l'implémentation des composants applicatifs logiques qui supportent le métier du SI cible, afin d'obtenir le service applicatif physique.
Le méta modèle d'architecture physique de risque est constitué de structures puisées au niveau des architectures de sécurité physique et des composants de SABSA[151]. L'architecture de sécurité physique SABSA décrit la présentation de la sécurité du systèmes de l'entreprise. Elle définit comment le SI au niveau de l'architecture logique est aligné à la structure des données telles que les fichiers et les bases de données. L'alignement des mécanismes de sécurité physiques aux services de sécurité logiques est aussi défini. En outre elle définit comment les mécanismes de sécurité physiques composés de fichiers systèmes et de bases de données peuvent être appliquées aux exigences de sécurité de service de l'architecture logique. De même SABSA définit les règles, les pratiques et les procédures pour fournir les détails de l'implémentation des politiques de sécurité.
Les artéfacts suivants sont concernés par cette couche architecturale : *Data model , Security Mechanisms, Security rules, practices, procedures, Users, Applications et the UI for Security, Platforms and Networks Infrastructure (Layout), Platforms and Networks Infrastructure (Capacity Plan and Resilience model)"*. Au niveau de l'architecture de sécurité des composants, chaque composant est un élément de l'ensemble, et ces éléments sont assemblés selon les conceptions intégrées dans l'architecture de sécurité physique, qui à son tour réalise les modèles de service fonctionnel au niveau de la couche logique.

Les livrables de l'architecture de sécurité des composants incluent un cadre pour les normes de sécurité et une liste de toutes les normes de sécurité requises. Aussi la description des spécifications de toutes les technologies, produits et outils qui ont été sélectionnés, avec des conseils pour les équipes de projet sur comment, pourquoi, où et quand les utiliser, est dressée. La liste inclue un schéma de nommage et un cadre pour définir les rôles, les identités, les profils de privilèges d'accès (permissions ou autorisations). Tout comme une conception détaillée de l'infrastructure de sécurité, y compris les processus d'application à exécuter, les nœuds de plate-forme sur lesquels ils doivent être hébergés, est fournie.
Nous avons vu (cf. Chapitre 3) que UMLsec [83] était un bon candidat à la méta modélisation des structures de TOGAF et SABSA utiles à la conception du système en encapsulant les composants logiques au niveau abstrait PIM. Dans cette même veine UMLsec présente l'avantage de capturer les exigences de sécurité définies au niveau des besoins et supportées par le SI, avec un large spectre de propriétés de sécurité. Il permet de définir une dépendance liée à la sécurité entre les éléments de l'architecture qui peut être cartographiée à l'aide de plusieurs diagrammes UML, présentant le système sous divers angles tels que les diagramme de déploiement pour le passage du PIM niveau PSM.
La section suivante décrit le méta modèle physique du risque.

4.2 Meta modèle contextuel du risque de sécurité physique : TCM PR

Le méta modèle contextuel physique de risque TCM-PR (Transformation Contextuel Model -Physical Risk) (voir Figure4.1) est le modèle de sécurité de l'architecture physique qui intègre les contrôles et mécanismes de traitement de risque de sécurité pertinents, définis dans les phases du processus de gestion de risque (voir chapitre 2 et modélisés dans les modèles TCM-BR et TCM-LR).

Le métamodèle décrivant TCM-BR (cf. Figure4.1), est la conceptualisation du contexte des services du SI *ContextualisedInformationSystemService* définis par l'attribut *name* du nom des services SI utilisés *UsedInformationSystemService* et l'implémentation *ImplementedInformationSystemService* des opération logiques

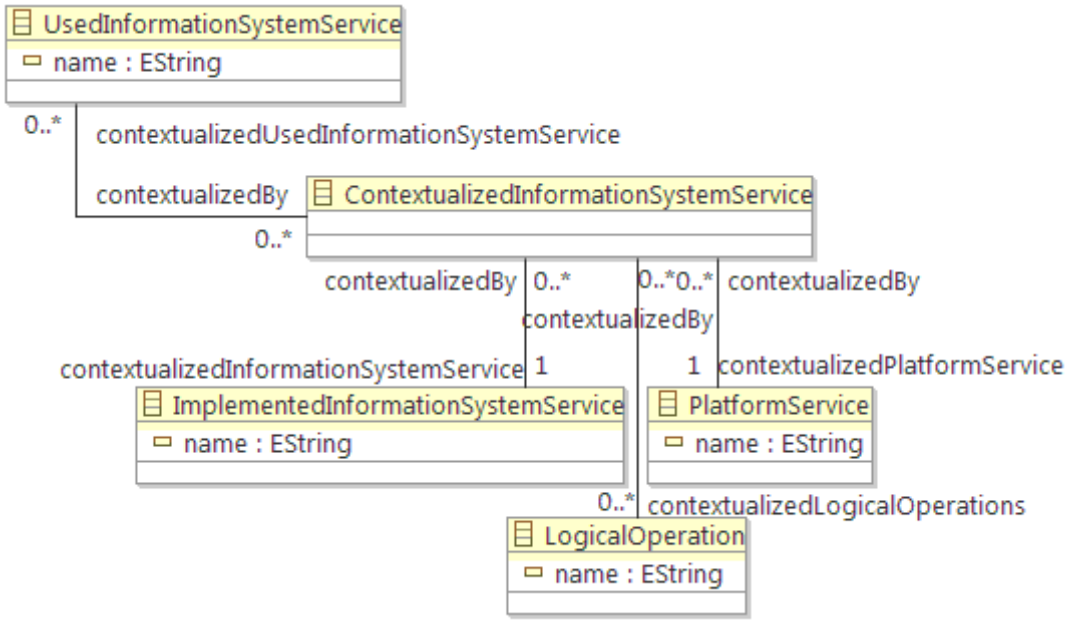

Figure 4.1 – TCM-PR : Modèle contextuel physique du risque

LogicalOperation sur une plateforme de service *PlatformService*. Les opérations logiques incluant les opérations fonctionnelles comme les opérations techniques (ie : sécurité).

Le méta modèle est une vue de la sécurité d'architecture technologique spécifique (combinant TOGAF et SABSA). TCM-PR contient un service spécifique au processus de traitement du risque : Service applicatif de sécurité qui décrit un service physique du SI existant. Une instance de ce concept est généralement un service prêt à l'emploi (off-the-shelf), défini comme synchrone ou asynchrone. Il est défini par une opération logique de risque, effectué sur un nœud de déploiement, en considérant qu'il pourrait être déployé sur différents environnements d'exécution.
TCM-PR offre une substitution du codage d'opérations logiques déployées sur une plateforme technique par un service existant du SI technique.
Par conséquent, les modèles définis précédemment sont intégrés dans l'approche MDA utilisant des méthodes de transformation de substitution. Cela défini l'architecture (applicative) physique de notre approche d'intégration décrite dans

la section suivante.

4.3 Processus d'intégration du modèle physique de sécurité au niveau de l'architecture physique : phase d'implémentation PSM

Une fois l'analyse et la conception du système réalisée, dont le résultat est l'architecture logique de sécurité du service métier, la prochaine étape est la phase d'implémentation (réalisation) de l'application sur une technologie spécifique.

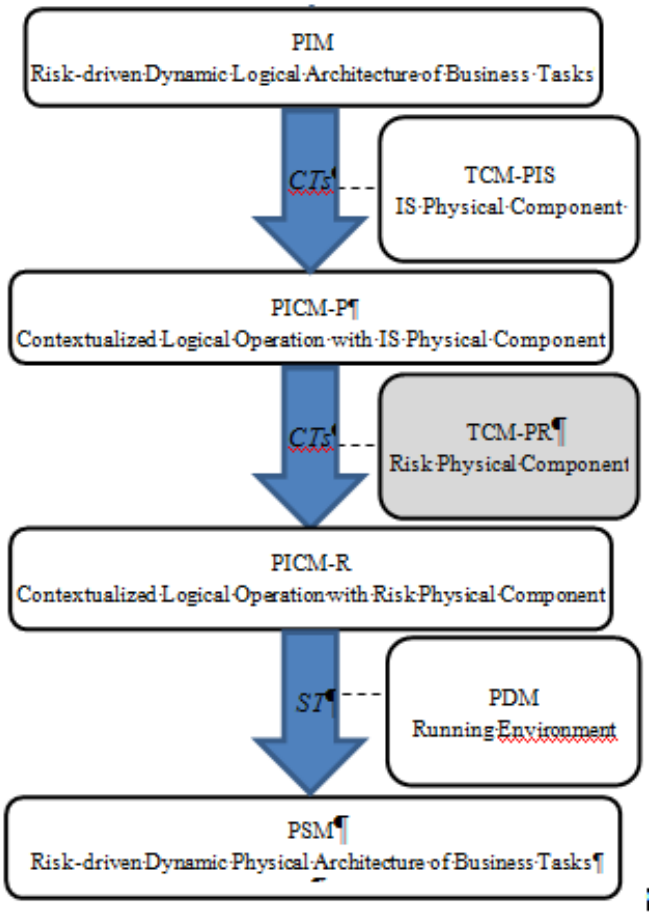

FIGURE 4.2 – Processus de transformation du modèle PIM vers le modèle PSM (PIM2PSM)

Afin de faire face à la transformation du modèle d'architecture logique en un modèle d'architecture (applicative) physique (PIM2PSM), une transformation par substitution (ST : Substitution Transformation) permet la réutilisation de services externes au SI existant comme décrit dans [157]. Cette réutilisation est représentée par une transformation contextuelle par substitution (CTs : Contextual Transformation by Substitution).
La substitution porte sur un extrait des opérations logiques instanciées par un service physique (externe) du SI, qui correspond aux opérations physiques définissant l'architecture applicative du service SI existant.
Comme nous pouvons le voir dans la figure 4.2, la phase d'implémentation de notre approche est composée de six modèles intervenant dans le processus de transformation pour passer du niveau PIM au niveau PSM avec les trois (3) étapes suivantes de passage : du niveau PIM vers le niveau PICM-P (PIM2PICM-P) par CTs, du niveau PICM-P vers le niveau PICM-R (PICM-P2PICM-R) par CTs, et enfin, du niveau PICM-R vers le niveau PSM (PICM-R2PSM) par ST. Et ce, avec l'intervention des modèles contextuels TCM-PIS, TCM-PR et PDM.
La sections suivantes illustrent les différentes transformations.

4.3.1 Transformation du PIM vers PICM-P (PIM2PICM-P)

Dans cette phase, la transformation (PIM2PICM-P) est réalisée par une transformation contextuelle par substitution CTs (transformation horizontale : même niveau d'abstraction) qui part du méta modèle PIM (modèle origine ou source), au méta modèle PICM-P (Platform Independent Contextuel Model- Physical) (modèle cible ou target) avec le méta modèle TCM-PIS comme contexte.
Cette transformation répond à la règle suivante : PICM-P = CTs (PIM, TCM-PIS).
Le méta-modèle TCM-PIS (TCM – Physical Information System), qui est le modèle contextuel de cette intégration avec substitution, contient le concept suivant :
« **Physical Application Service : PAS** » décrit un service physique du SI existant (externe à la transformation), recommandé pour implémenter les opérations logiques encapsulées dans les composants applicatifs logiques du SI supportant

les exigences métiers de nature fonctionnelle, du métier ciblé.
L'instanciation de ce concept signifie que le service est un pattern décrivant un contexte du modèle physique du SI TOGAF [157], qui peut être réutilisé. Il est défini par la ou les opérations d'application logique effectuées et le nœud de déploiement, en considérant qu'il peut être déployé sur différents environnements d'exécution (Ex : le service physique PASAuthentication (Physical Application Service Authentication) réalise le composant logique LAOReadCredentials et est déployé sur un nœud qui est serveur web).
Ainsi le service logique d'authentification de session est supporté par le service physique d'authentification qui contient un mécanisme physique qui peut être la solution "Mutual two-way and three-way", "authentication exchanges", ou "Session context (finite state machine)" comme recommendé dans (SABSA Physical Security Service[151]).
Dans le cadre de notre thèse, TCM-PIS ne contient aucun service du SI utile à la réalisation du cas d'étude de BSOnlineShopping.
PICM-P est le "Modèle Contextuel des Opérations Logiques avec les Composants Physiques du SI". Il montre une relation de correspondance (notion de « Contextualized Logical Application Operation ») entre opération(s) d'un composant applicatif logique et service applicatif physique du SI conçu par l'Architecte d''Entreprise.
Etant donné que TCM-PIS ne contient aucun service, PICM-P n'est pas pertinent pour une illustration.

4.3.2 Transformation de PICM-P vers PICM-R (PICM-P2PICM-R)

La transformation PICM-P2PICM-R est une transformation contextuelle par substitution CTs (transformation horizontale) qui part du méta modèle PICM-P (modèle origine) vers le méta modèle PICM-R (Platform Independent Contextuel Model- Risk) (modèle cible).
La transformation PICM-P2PICM-R fait intervenir le méta modèle TCM-PR décrit plus haut et définissant le contexte de la sécurité au niveau physique.
Le méta-modèle PICM-R montre une relation de cartographie (alignement) entre

une opération logique du risque et un service physique du risque.
Cette cartographie est réalisée par un expert du risque qui instancie le concept de contextualisation qui relie une instance de l'opération logique liée au risque avec une instance du service applicatif physique qui la réalise. Ainsi le service physique du risque implémente l'opération logique du risque.
La transformation (PIM-P2PICM-R) obéit donc à la règle suivante : $PICM-R = CTs(PICM-P, TCM-PR)$
Ceci montre donc une intégration des artefacts des architectures logique et physique de TOGAF par la transformation des modèles contextuels. Cette intégration satisfait la logique de l'EA qui consiste en l' alignement entre les niveaux architecturaux qui y interviennent.

4.3.3 Transformation PICM-R vers PSM (PICM-R2PSM)

La transformation (PICM-R2PSM) est une transformation par substitution ST (transformation verticale) qui part du méta modèle (PICM-R) vers le méta modèle (PSM) (Platform Specific Model) avec le contexte PDM (Plateform Description Model).
PDM décrit l'architecture technique (correspondant au TOGAF) et spécifie les contraintes techniques d'une transformation ST. Il contient des notions spécifiques à la définition du pattern de l'architecture physique proposée par [156] telles que « Application-Server », « Service Layer » et « Data Access Layer ».
Le PDM (cf. Figure 4.3), composée des éléments d'infrastructure technique [137] contient :
SupportedBusinessService : décrit le service métier supporté par la plateforme d'implémentation. Il est décrit par l'attribut "name" et lié à la plateforme de service par la rélation *supportedBy*.
PlatformService : supporte le service métier. C'est la description logique et physique de la plateforme (environnement, nœud, lien de communication) d'exécution de l'application qui réalise le service métier qu'il supporte. La plateforme est caractérisée par une couche logique (contenant les nœuds logiques d'exécution, l'environnement logique d'exécution et les communications logiques d'exécution) qui instancient les composants physiques d'exécution au niveau de

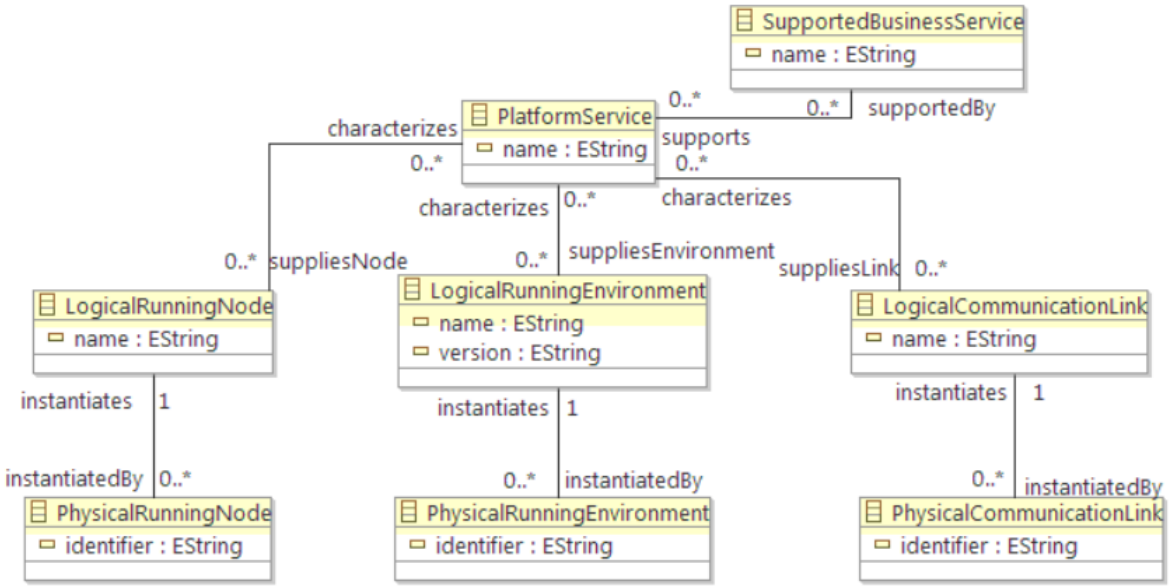

FIGURE 4.3 – Platform Description Model (PDM)

la couche physique.
LogicalRunningNode : décrit un nœud d'exécution au niveau logique et qui instancie un ou plusieurs nœuds physiques d'exécution "**PhysicalRunningNode**". De la même manière l'environnement d'exécution "**LogicalRunningEnvironment**" (respectivement le lien de communication "**LogicalRunningCommunication**"), au niveau logique, instancie un ou plusieurs environnements physiques d'exécution "PhysicalRunningEnvironment" (respectivement le lien de communication "PhysicalRunningCommunication") au niveua physique.
La transformation (PICM-R2PSM), transformation ST qui fournit un PSM résultant d'un PICM-R et d'une contrainte technique spécifiée dans le PDM est définie par : $PSM = ST(PICM - R, PDM)$. PSM est décrit par le méta modèle de l'architecture de conception physique (cf. Figure 4.4).

Le PSM est l'architecture applicative du SI implémentée sur une plateforme physique. Elle regroupe les éléments du service métier du SI supportés par les composants applicatifs logiques et par les composants technologiques. Conformément à l'architecture en 3 couches représentée par PDM, le PSM est composé d'une couche présentation, d'une couche métier, d'une couche de données, et d'une couche technique, chacune décrite par un composant dédié et contenant

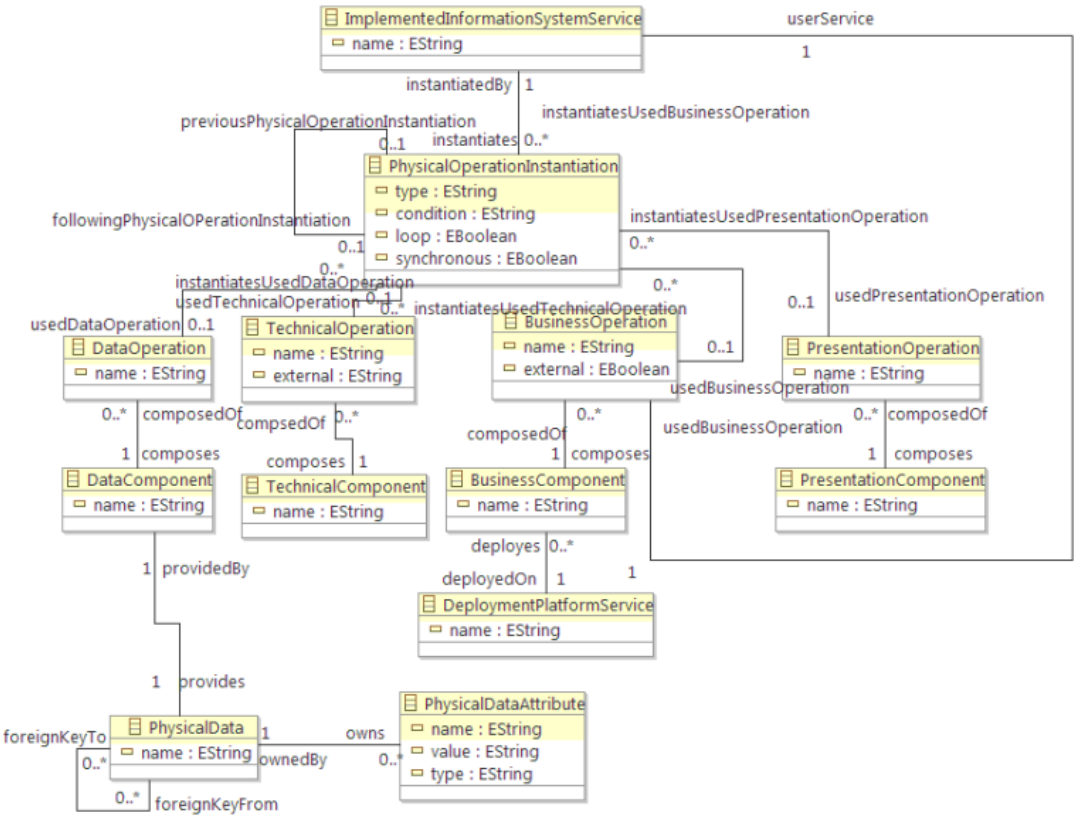

FIGURE 4.4 – Architecture de Conception Physique (PSM)

les opérations implémentées. Une instance de l'architecture physique est décrite à la figure 4.5 avec les dépendances entre les composants.

4.3.4 Le niveau PSM : : Architecture de conception physique

La transformation du modèle PICM-R2PSM est une transformation MDA classique prenant en compte les composants technologiques. Ces composants tels que définis dans le méta-modèle TOGAF et SABSA peuvent être spécialisés dans l'exécution de nœuds, d'environnements d'exécution ou de liens de communication. Le concept résultant du PSM est un service d'application axé sur les risques. Le diagramme de séquence sur la figure 4.5 est le PSM résultant et illustrant cette transformation.

Le diagramme de séquence instancie les concepts du méta modèle PSM (cf.

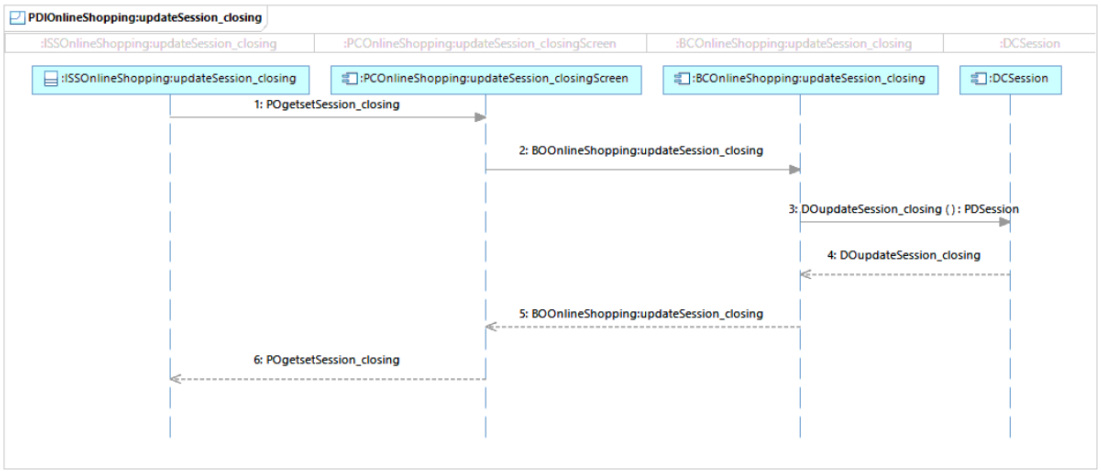

FIGURE 4.5 – PSM illustration resulting from ST substitution transformation.

Figure 4.4) liés à l'IHM (Interface Homme-Machine) : la couche Présentation (Presentation Component/Operation), la couche métier (Business Component/Operation) et la couche de données (Data Component/Operation) de l'application. L'application décrit le service de gestion de session du SI du cas d'étude d'achat en ligne.

4.4 Illustration : Transformation Du PIM au PSM avec un contexte lié aux risque physique

4.4.1 Alignement entre les opérations logiques implémentées et les services SI

(Le tableau 4.1) décrit une instance du méta modèle TCM-PR (Physical Risk) par la description des services du SI recommandés pour implémenter les opérations logiques encapsulées dans les composants applicatifs logiques du SI supportant les exigences métiers de sécurité. Il offre une substitution du codage d'opérations logiques déployées sur une plateforme technique par un service existant du SI technique.

Service du SI	**Opération(s) logique(s) implémentées**	**Plateforme de déploiement**	**Commentaire**
ISSKeePass	processCredentialsSpoofing	EN Web Server	Logiciel libre et open source, gestionnaire de mots de passe totalement gratuit et recommandé par l'Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI)
ISSArchi2Secu	processDisruptiveDevice	EN Web Server	Service de détection de numéro IP à risque développé avec l'approche MDA, contextualisée par le fichier des IP à risque, constitué à partir d'une détection d'un nombre important d'appels en cascade.

Tableau 4.1 – Extrait des services du SI de TCM-PR utiles au traitement du risque dans BSOnlineShopping

BSOnlineShopping		**Services réutilisables du SI**		
Opération logique	**Plateforme technique}**	**Service du SI**	**Opération(s) logique(s) implémentées**	**\textbf{Plateforme de déploiement}**
processCredentialsSpoofing	EN Web Server	ISSKeePass	processCredentialsSpoofing	EN Web Server
processDisruptiveDevice		ISSArchi2Secu	processDisruptiveDevice	

Tableau 4.2 – Alignement des opérations logiques et de la plateforme technique

La réutilisabilité du service du SI au niveau physique est décrit à l'aide de l'ensemble des opérations logiques réalisées et de la plateforme technique sur laquelle le service est déployé. Dans notre cas d'étude BSOnlineShopping, deux services de TCM-PR sont pertinents (cf. Tableau 4.1) : **ISSKeePass et ISSArchi2Secu** qui sont décrits dans la colonne de commentaire du tableau, sont des services de sécurité, réutilisés respectivement en vue de l'implémentation des opérations logiques **processCredentialsSpoofing** et **processDisruptiveDevice**, et tous destinés à être déployés sur un server web.

4.4.2 Le PICM-R (Platform Independent Contextual Model-Risk)

PICM-R est le PIM enrichi des services SI applicatifs liés à la sécurité et alignés avec des opérations logiques (cf. Tableau 4.1). Le Tableau 4.2 représente cet alignement qui peut être réalisé par un architecte physique ou applicatif du système ou de manière automatique si un catalogue de ces services existe.

-Dans l'illustration, une instance du PDM décrit une plateforme avec un environnement JEE (en particulier, des EJB (Enterprise JavaBeans) [9] et un système de gestion de base de données relationnelle SQL. Ce modèle permet de générer automatiquement, en plus de l'architecture physique du système, le script de génération de la base de données et un squelette du code des services du SI dans un cadre EJB.
-Le PSM est le résultat de la transformation PICM-R2PSM, du PIM obtenu précédemment, contrainte par le PDM avec une architecture JEE en couches (Présentation, Métier et Accès aux données) et SQL. Cette transformation est implémentée avec operational-QVT. Les modèles suivants sont le résultat de ST chainée avec une transformation du PSM vers UML2 implémentée aussi avec operational-QVT :

Le modèle de composants applicatifs physiques dédié à la réalisation de l'arbre de dépendances de sommet LACSessionManagement est illustré ci-dessous (cf.

Figure 4.6). Par rapport à l'architecture en 3 couches choisie dans notre illustration : PC (respectivement PO) indique un composant (Component) (respectivement une opération(Opreation)) de la couche Présentation (Presentation), BC (respectivement BO) indique un composant (respectivement une opération) de la couche Métier (Business), DC (respectivement DO) indique un composant (respectivement une opération) de la couche Donnée (Data). Les dépendances sont fixées par le pattern d'architecture physique associé (composant de la couche présentation dépend de composant de la couche métier et composant de la couche métier dépend de composant de la couche d'accès aux données. Une couche Technique est ajoutée afin de réaliser les opérations logiques telles que pour le composant TCRiskManagement.

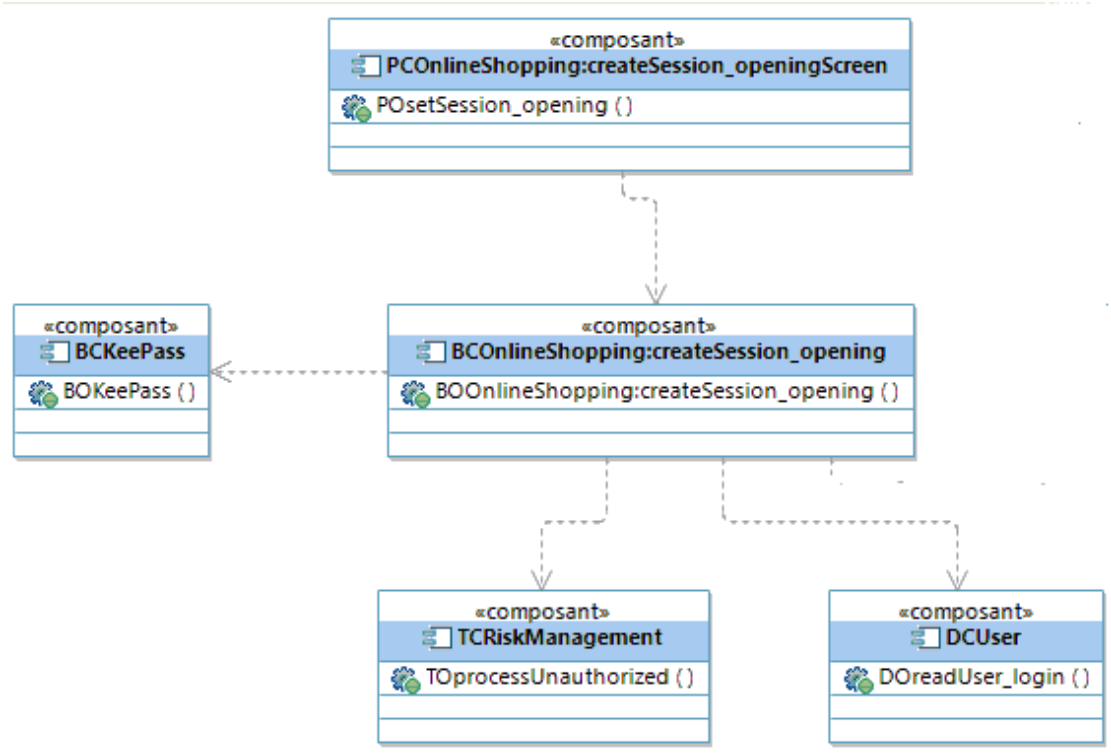

FIGURE 4.6 – Modèle de composants applicatifs physiques réalisant l'arbre de dépendances de sommet LACSessionManagement

Ainsi l'arbre de dépendance dans la figure 4.6 qui part du sommet $PCOnlineShopping$: $createSession_openingScreen$, composant de création de session exécute l'opération de création de session $POsetSession-opening()$ et dépend du composant

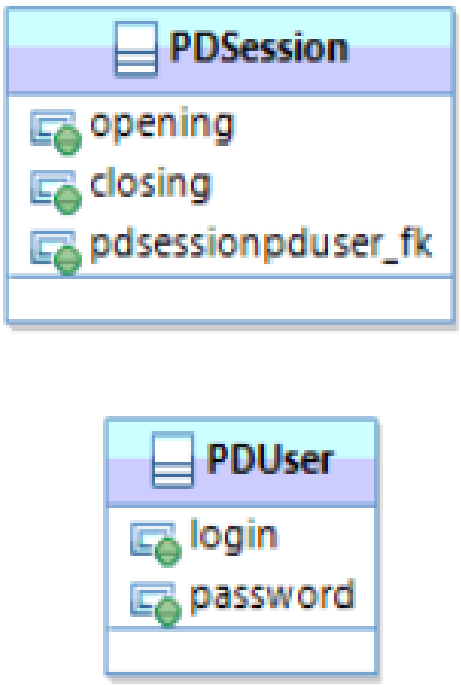

FIGURE 4.7 – Données physiques de création de session

métier (BC) d'ouverture de session qui exécute l'opération équivalente et dont la réalisation dépend du composant de données utilisateur de session et du composant technique.
Dans ce modèle de composants applicatifs, l'appel à des services réutilisables du SI est conçu sur la couche Métier. C'est le cas dans ce modèle avec l'appel à l'opération de **BCKeePass** et l'opération technique **TOprocessUnauthorized()** dédiée au traitement du risque "unauthorized", contenu dans le composant *TCRiskManagement*. De même, les services du SI sont conçus sur cette couche Métier avec les opérations offertes par $BCOnlineShopping : createSession_opening$.

La conception automatique des données physiques (cf. Figure 4.7) est fondée sur la réalisation des données logiques. Dans le modèle de données physiques, les dépendances sont représentées par des clés étrangères (libellée pd_fk) du fait des contraintes SQL modélisées dans le PDM.

FIGURE 4.8 – Modèle physique dynamique du service createSessionOpening

Dans le modèle de données généré, la donnée *PDSession* réalise la dépendance de la donnée logique *LDSession* vers *LDUser* (conforme à la dépendance de LACSessionManagement vers LACUserManagement), grace à la jointure avec la clef étrangère ($pdsessionpduser_fk$).
Le modèle dynamique de l'architecture physique est illustré avec les diagrammes de séquence associés aux services du SI générés et des instances des composants applicatifs physiques (cf. Figure 4.6 et Figure 4.8). Ces diagrammes sont générés à partir du pattern de couches implémenté dans la transformation PICM-R2PSM (cf. 4.3.3).
Le diagramme de séquence Figure 4.8 représente l'architecture physique dynamique du service $ISSOnlineShopping : createSession_opening$ qui réalise son architecture logique (cf. Chapitre 3).
De la même façon, l'ovale gris souligne la représentation d'un appel synchrone pour le service réutilisé BOKeePass alors que l'ovale noir cible la représentation d'un appel asynchrone avec l'opération technique TOprocessUnauthorized.

4.5 Conclusion

Le chapitre a présenté une méthode d'intégration d'un modèle contextuel de risque physique dans le cycle de développement du système logiciel conforme à

la transformation du PIM en PSM dans l'approche MDA.
La méthode propose une approche de construction du méta modèle du risque physique fondé sur l'architecture technologique de l'EA. Le méta modèle étend le concept de UMLsec, encapsulant les composants physiques des architectures TOGAF et SABSA.
Les résultats obténus dans ce chapitre permettent de repondre à la question de recherche **QR3 :-** ***Quel impact de la mise en œuvre de la sécurité via les risques dans l'architecture physique et les conséquences sur l'implémentation du système ?***.
En effet, les instances du méta modèle sont des services physiques de sécurité, "réutilisables", implémentant les services logiques correspondantes, définies au chapitre 3.
Le processus d'intégration du modèle de risque physique au système par MDA consacre un apport fondamental dans nos travaux, avec la prise en compte du contexte de sécurité par une transformation par substitution, s'appuyant sur les travaux de [156][157]. Cette transformation par substitution intègre dans le cycle de développement du système (dans la phase d'implémentation), les modèles physiques (patterns du SI fonctionnel et technique) supportant l'architecture logique qui elle supporte le métier.
La phase d'expérimentation a permit d'obtenir l'architecture du système avec les dépendances des composants de nature fonctionnelle, physique et technique. Le diagramme de séquence est une illustration dynamique de l'application avec l'interaction et dépendance des composants physiques ainsi que la réalisation du traitement synchrone ou asynchrone du risque.
L'implémentation sur l'environnement JEE a permit l'obtention du script SQL permettant de générer la base de données physiques et l'application.
Le chapitre suivant est consacré à l'expérimentation illustrée par un cas d'étude d'achat en ligne.

Chapitre 5

Expérimentation et Vérification

5.1 Etude de cas

L'étude de cas est basé sur un système de commerce électronique (d'achat en ligne), désigné par (BSOnlineShopping) (cf. Figutre5.1). L'étude de cas montre une intégration du traitement du risque associé à une propriété de sécurité dans une démarche de développement d'un système. La solution choisie pour les exigences métiers de nature fonctionnelle est l'approche MDA contextualisée [156] qui a l'intérêt d'être déjà outillée [157]. Cette solution doit donc être étendue pour les exigences métiers de nature technique telles celles définissant la sécurité du système. Le 1 est consacré à l'analyse des menaces et évaluation du risque. Le 2 est dédié à la définition de contextes propres au traitement du risque et introduit la démarche. Le 3 cible la transformation des spécifications métiers en un modèle d'architecture logique et le 4, la transformation de l'architecture logique en une architecture physique, incluant le code. Le 5.2 est consacré à la vérification du cas d'étude et une discussion déduite de l'étude cas est suggérée

1. Lire identifiant et mot de passe, de l'utilisateur, qui existent (tâche alignée avec la propriété de sécurité d'authentification)
2. Créer la session pour l'utilisateur avec la date d'ouverture de la session
3. Sélectionner un ensemble d'articles caractérisés par leur code, leur libellé et leur prix
4. Lire les détails de la carte bancaire avec le numéro, la date d'expiration et le code CVV
5. Créer la transaction avec la banque avec sa date et le montant total (tâche alignée avec la propriété de sécurité d'intégrité)
6. Créer la commande avec sa date et sa référence
7. Modifier la session pour l'utilisateur avec la date de fermeture de la session

FIGURE 5.1 – Spécifications du service métier d'achat en ligne (BSOnlineShopping)

dans le 5.3.

1. **Analyse des menaces et évaluation des risques**

a) **Etablissement du contexte de l'organisation et des actifs**

Un système de commerce électronique (service métier d'achat en ligne) comprend un certain nombre de tâches et d'interactions complexes qui sont difficiles à analyser complètement dans cette expérimentation. Ainsi, nous avons fait une sélection ciblée des tâches au cœur de l'activité du service métier d'achat en ligne (cf. Figure 5.1).
Les services de lecture des identifants clients et de paiement sont particulièrement intéressants dans cette chaîne de valeur où les informations sensibles des clients, des commerçants et des entreprises sont requises pour s'authentifier et effectuer des transactions. Les actifs au sein de ces services nécessitent une haute sécurité en terme de besoin de confidentialité, d'intégrité et de disponibilité. Ces services fournissent une surface d'attaque raisonnablement interessante pour l'analyse des menaces de sécurité et la gestion des risques.

Actifs/biens méties identifiés

Les actifs ou biens sont les actifs métiers essentiels du service métier menant au bon fonctionnement du système d'achat en ligne. Ces actifs sont des fonctions qui traitent des informations essentielles en entrée ou/et en sortie. Le tableau 5.1 présente les actifs métiers.

Les actifs systèmes (biens supports)

On suppose dans le cadre de notre cas d'étude que le service métier est hébergé sur des serveurs dans un environnement cloud redondé chez un prestataire de service spécialisé dans ce domaine (cf. Figure 5.2). Dans la suite du document, le(s) serveur(s) hébergeant notre service sera nommé SHP. Nous considérons de même l'équipement utilisé par l'utilisateur de notre service que nous nommerons EU dans la suite de l'étude.

b) Besoin de sécutité

[HTML]C0C0C0 **Service métier**	**Services (essentiels) du SI**	**Informations (données) essentielles concernées**	**Dépositaire**
Service Achat en ligne	Lecture identifiants Utilisateur	Login utilisateur Mot de passe	Utilisateur Propriétaire service
	Ouverture de session	Session ID	Utilisateur Propriétaire service
	Choix des Article	Date de création/ ouverture de session Code Article Libellé Article Prix ARticle	Utilisateur
	Lecture détails carte bancaire	Numéro Carte bancaire Date d'expiration CVV	Utilisateur Propriétaire service
	Transaction Bancaire	Date de la Transaction Montant de la Transaction	Banque Propriétaire service
	Fin de Commande	Référence de la commande Date de création de la commande Date de fermeture de la session	Propriétaire service

Tableau 5.1 – Actifs/biens métiers du service métier BOnlineShoping

Biens supports	**Biens essentiels**	**Lecture Identifiants Utilisateurs**	**Choix des articles**	**Lecture détails bancaire**	**Transaction bancaire**	**Fin commande**
Biens supports propriétaire service						
SHP - Serveur Hébergeant Partenaire		X	X	X	X	X
DBU - Base de données Informations Utilisateurs		X	X	X	X	X
SA – Service Achat en ligne		X	X	X	X	X
Biens Utilisateurs						
EU – Equipement Utilisateur		X	X	X		X

Figure 5.2 – Actifs systèmes du service métier BOnlineShopping

Dans l'exemple de spécification du service métier d'achat en ligne décrit, la propriété de sécurité d'authentification est associée à la lecture de l'identifiant et du mot de passe de l'utilisateur du portail de commerce en ligne [178]. De même, les propriétés de sécurité de confidentialité et d'intégrité sont liées à la tâche de création d'une transaction avec la banque de l'utilisateur du fait des échanges de données, induits par cette

transaction, sur le réseau [124].

c) Analyse des ménaces et évaluation du risque

Etude des évenements redoutés avec Ebios

Chaque ligne des (figures 5.3 et 5.4) suivantes, représente un événement redouté (non exhaustive) par rapport au service d'achat défini. La gravité de chaque événement redouté est estimée conformément à l'échelle de gravité des évenements redoutés.

Evenement redouté	Besoin de sécurité	Sources de menaces	Impacts	Gravité
Lecture Identifiants Utilisateurs				
Indisponibilité de la base de données	Disponibilité (Moins de 4h)	Incendie chez l'hébergeur Hacker (Attaque DDOS)	Impossibilité d'accéder à son portail Perte d'argent Poursuite judiciaire Perte de clients	3.Importante
Vol de données	Confidentialité (Privé)	Hacker		3.Importante
Suppression de la base de données	Confidentialité (Privé)	Hacker		4.Critique
Choix des articles				
Altération des informations des articles	Intégrité (Intègre)	Hacker Employé malveillant Concurrent	Perte d'argent Effacement des traces en cas de litige Sabotage produits	2.Limitée
Possibilité d'altération des informations liées à la création ou à l'ouverture de session	Intégrité (Maitrisé)	Hacker Employé malveillant		2.Limitée

FIGURE 5.3 – Détermination des évenements rédoutés-1

Evaluation des événements redoutés

La figure 5.5 présente les résultats de l'évaluation de 6 événements redoutés à la gravité critique ou importante.

Analyse de la ménaces avec l'approche STRIDE

Nous déterminons les menaces à partir de la méthode STRIDE (cf. Figure 5.6) , nous utilisons le diagramme de flux de données DFD dont

Lecture détails bancaire				
Vol ou fuite des informations bancaires	Privé	Hacker Employé malveillant	Usurpation carte bancaire Utilisateur Poursuite judiciaire	4.Critique
Transaction bancaire				
Altération des informations de la transaction	Intégrité (Intègre)	Hacker	Perte d'argent	4.Critique
Fin commande				
Altération ou modification de la référence de la commande	Intégrité (Intègre)	Hacker Employé malveillant	Perte d'argent Effacement des traces en cas de litige	3.Importante
Altération de la date de création de la commande	Intégrité (Maitrisé)	Hacker Employé malveillant		2.Limitée
Altération de la date de fermeture de session	Intégrité (Maitrisé)	Hacker Employé malveillant		2.Limitée

FIGURE 5.4 – Détermination des évenements rédoutés-2

Gravité	Evènements redoutés
4. Critique	Suppression de la base de données Vol ou fuite des informations bancaires Altération des informations de la transaction bancaire
3. Importante	Indisponibilité de la base de données Vol de données Altération ou modification de la référence de la commande
2. Limitée	Altération des informations des articles Possibilité d'altération des informations liées à la création ou à l'ouverture de session Altération de la date de création de la commande Altération de la date de fermeture de session
1. Negligeable	

FIGURE 5.5 – Evaluation des évenements redoutés

les initiaux signifient : - DF : Data Flow (Flux de Données), DS : Data Store (Stockage de données avec une Base de données), I : Interacteur (interaction avec une Entité interne ou Externe), P : Process (Processus).

Tâches/ Actifs métiers	Spoofing	Tampering	Repudiation	Information Discolsure	Denied of Service	Elevation of Privilege
Lire identifiants (DF, DS, I)						
Creer Session (DF)						
Selection Article						
Lire Carte Bancaire (DS, I)						
Transaction (DF, I , P)						
Creer Commande						
Modifier/Ferméture Session						

FIGURE 5.6 – Détermination des menaces basée sur STRIDE avec le DFD

Pour une question de concision nous nous limitons à l'étude des tâches de lecture d'identifiants, de création de session, de lecture de la carte bancaire et de la transaction. La tâche de sélection d'article étant sous-jascente à celle de la trasaction. Cette étape permet d'élaborer les scénarios de menaces basés sur EBIOS.

Scénarios de menaces par Ebios : Conformement aux menaces déterminées précedemment, le tableau 5.2 défini les scénarios de menace. Ensuite, nous évaluons les menaces (cf. Figure 5.7) en déterminant leur niveau de vraisemblance.

Estimation du risque :
La figure (cf. Figure 5.8) permet d'estimer les risques avec les niveaux de gravité définis selon les couleurs (rouge : intolérable, orange : significatif et vert : négligeable).
Exigences de sécurité et décision de traitement du risque : Comme le montre la figure 5.9, 4 risques à réduire et/ou à transférer en priorité. Les risques jugés comme prioritaires et significatifs sont essentiellement destinés à réduire (atténuer), et les risques jugés comme non prioritaires,

[HTML]9B9B9B	[HTML]9B9B9B**Scénarios de menaces**	[HTML]9B9B9B**Sources de menaces**	**Sources de menaces**
Spoofing			
	Spoofing des identifiants utilisateurs	Hacker Virus ciblé ou non ciblé Employé malveillant	2.Signicative
	Spoofing des données bancaires d'un utilisateur	Hacker Virus ciblé ou non ciblé Employé malveillant	3.Forte
Tampering			
	Altération des informations articles	Hacker Virus ciblé ou non ciblé Employé malveillant	1.Minime
	Altération du montant de la transaction	Hacker Virus ciblé ou non ciblé Employé malveillant	2.Signicative
	Altération des informations de la commande	Hacker Virus ciblé ou non ciblé Employé malveillant	2.Signicative

Tableau 5.2 – Etude des scénarios de menaces

Vraisemblance	**Scénarios de menaces**
4.Maximale	
3.Forte	Spoofing des données bancaires d'un utilisateur
2.Significative	Spoofing des identifiants utilisateurs Altération du montant de la transaction Altération des informations de la commande
1.Minime	Altération des informations articles

Figure 5.7 – Evaluation des scénarios de menace

destinés à être acceptés.

Le tableau de la figure 5.9 suivante présente les exigences de sécurité identifiés sous (04) possibilités : l'évitement, la réduction, la prise et le transfert. Dans le tableau, les croix correspondent aux premiers choix tandis que celles entre parenthèses correspondent aux autres possibilités acceptées ou acceptables.

Gravité	4. Critique	-Risque de suppression de la base de données			
	3.Importante	-Risque de vol ou fuite des informationst bancaires	-Risque de vol de données -Risque d'altération des informations de la transaction	Risque d'indisponibilité de la base de données	
	2. Limitee	-Risque d'altération des informations des articles -Risque d'altération des informations de création/ouverture de session -Risque d'altération de la référence de la commande			
	1. Négligeable	-Risque d'altération de la date de création de la commande -Risque d'altération de la date de fermeture de session			
		1. Minime	2. Significative	3. Forte	4. Maximale
		Vraisemblance			
Légende		Risques négligeables	Risques significatifs	Risques intolerables	

FIGURE 5.8 – Estimation du risque

Risques	Evitement	Réduction	Prise	Transfert
Risque d'indisponibilité de la base de données	(X)	X	(X)	(X)
Risque de vol de données		(X)	X	(X)
Risque de suppression de la base de données		(X)	X	(X)
Risque d'altération des informations des articles	(X)	X	X	
Risque d'altération des informations liées à la création ou à l'ouverture de session	(X)	X	X	
Risque de vol ou fuite des informations bancaires	(X)	X	X	(X)
Risque d'altération des informations de la transaction	(X)	(X)	X	(X)
Risque d'altération de la référence de la commande	(X)	X	X	
Risque d'altération de la date de création de la commande	(X)	X	X	
Risque d'altération de la date de fermeture de session	(X)	X	X	

FIGURE 5.9 – Décision de traitement du risque

Dans notre cadre d'étude, nous n'avons pas de risques résiduels. Pour rappel, un risque résiduel est un risque qui ne pouvait ni être évité, ni transféré à l'issu de notre étude. Des partages de risques ou des sous-

criptions d'assurance cyber permettront d'assurer la véracité du tableau indiqué ci-dessus.

2. **MDA contextualisé par le traitement du risque.**
Dans le cadre du traitement du risque, les exigences métiers sont complétées par l'association de propriété de sécurité à certaines tâches. Cette association d'exigences métiers de nature technique nécessite une collaboration entre expert métier et expert sécurité.

L'approche MDA contextualisée permet d'intégrer par enrichissement ou par substitution des modèles représentant un contexte [156]. Cette approche est appliquée avec des exigences métiers de nature fonctionnelle dans [157]. Afin d'intégrer des exigences métiers de nature technique telles que celles ciblant la sécurité du système à développer, trois modèles contextuels aux transformations (TCM : Transformation Contextual Model) associés au risque sont pris en compte :

- **TCM-BR (Business Risk)**, qui représente le modèle de menace STRIDE instancié pour les propriétés de sécurité associées aux spécifications du service métier à traiter (cf. Table 5.3). La synchronisation du traitement du risque, en dernière colonne, indique le mode avec lequel le risque doit être traité. Pour le service métier BSOnlineShopping, le traitement de l'identifiant et du mot de passe doit être traité de façon synchrone avec leur lecture, puisqu'elle en dépend, l'autorisation via le profil de l'utilisateur dépend de la lecture de l'identifiant et du mot de passe et est donc en mode asynchrone, la vérification de la nocivité d'un terminal, via son adresse IP, est synchrone avec la transaction bancaire qui en dépend [62].

Modèle de ménacee	**Propriété de sécurité**	**Risque pour service métier**	**Synchronisation du traitement du risque**
Spoofing	Authentication	Credentials spoofing	Synchronous
		Unauthorized	Asynchronous
Tampering	Intégrty	Disruptive Device	Synchronous

Tableau 5.3 – Instanciation du modèle de menace STRIDE pour le service métier BSOnlineShopping dans TCM-BR.

-**TCM-LR (Logical Risk)**, qui est le modèle des composants applicatifs logiques (LAC : Logical Application Component) du SI supportant les exigences métiers de nature technique du métier ciblé. Il est notable qu'aucune dépendance de ces composants avec les composants applicatifs logiques supportant les exigences métiers de nature fonctionnelle n'est conçue a priori. En effet les propriétés de synchronisation de traitement du risque permettent une conception automatique de ces dépendances. Le choix pour le SI dédié au portail de commerce en ligne est la conception d'un seul composant supportant les exigences métiers liées au traitement du risque (cf. Figure 5.10) : *LACRiskManagement*.

FIGURE 5.10 – TCM-LR : Modèle d'architecture logique du SI technique supportant le domaine métier de l'achat en ligne

-**TCM-PR (Physical Risk)**, qui est la description des services du SI recommandés pour implémenter les opérations logiques encapsulées dans les composants applicatifs logiques du SI supportant les exigences métiers de nature technique du métier ciblé. Afin de pouvoir être réutilisé, un service du SI au niveau physique est décrit à l'aide de l'ensemble des opérations logiques réalisées et de la plateforme technique sur laquelle le service est déployé. Dans le cas de BSOnlineShopping, deux services de TCM-PR sont pertinents (cf. Tableau 5.4).

Service du SI	Opération(s) logique(s) implémentées	Plateforme de déploiement	Commentaire
ISSKeePass	processCredentialsSpoofing	EN Web Server	Logiciel libre et open source, gestionnaire de mots de passe totalement gratuit et recommandé par l'Agence Nationale de la Sécurité des Systèmes d'Information (ANSSI)
ISSArchi2Secu	processDisruptiveDevice	EN Web Server	Service de détection de numéro IP à risque développé avec l'approche MDA, contextualisée par le fichier des IP à risque, constitué à partir d'une détection d'un nombre important d'appels en cascade.

Tableau 5.4 – Extrait des services du SI de TCM-PR utiles au traitement du risque dans BSOnlineShopping

Ces trois modèles contextuels de transformation de modèles sont grisés dans le schéma de l'approche MDA contextualisée étendue avec le traitement des risques (cf. Figure 5.11).

Deux types de transformation de modèles permettent d'aligner les modèles contextuels de transformation : une transformation d'alignement par enrichissement (CTe) et une transformation d'alignement par substitution (CTs). Une transformation ET intègre en entrée un modèle résultant d'un alignement par enrichissement et une transformation ST un modèle résultant d'un alignement par substitution.
Comme indiqué dans [157], les concepts sont extraits ou étendus à partir du méta-modèle TOGAF [62]. Conformément à l'approche MDA, les modèles de la Figure 5.11 sont détaillés par phase de développement : du CIM (Computation Independent Model) au PIM (Platform Independent Model), du PIM au PSM (Platform Specific Model). Les résultats sont illustrés avec le langage UML2 [63][à partir de transformations complémentaires vers le méta-modèle UML implémentées pour le PIM et le PSM. L'objectif est de rendre les modèles générés plus lisibles.

3. **Du CIM au PIM avec un contexte lié aux risques**
   Pour l'intégration du traitement du risque dans la transformation des spécifications métier du CIM en un modèle d'architecture logique du PIM, deux modèles contextuels décrits précédemment (cf. 2) sont utiles :
   TCM-BR permet d'enrichir une tâche du service métier avec une propriété de sécurité.

   TCM-LR permet d'enrichir une tâche du service métier avec un composant applicatif logique du SI technique.
   Le modèle contextuel déjà défini dans [157] pour l'approche ciblant les exigences métiers de nature fonctionnelle est :

   TCM-LIS (Logical Information System) qui modélise les composants applicatifs logiques du SI supportant les exigences métiers de nature fonctionnelle du métier ciblé. Par rapport au SI lié à un portail d'achat en ligne, le modèle conçu par les architectes d'entreprise de cette vue logique du SI

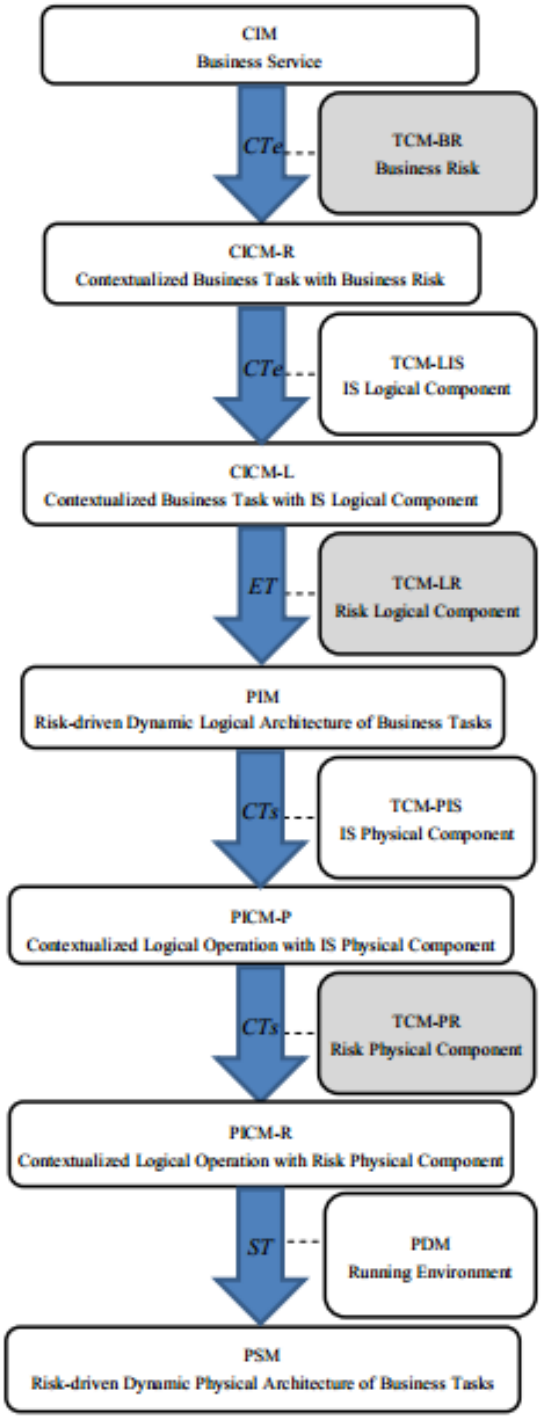

FIGURE 5.11 – Intégration du traitement du risque dans l'approche MDA contextualisée

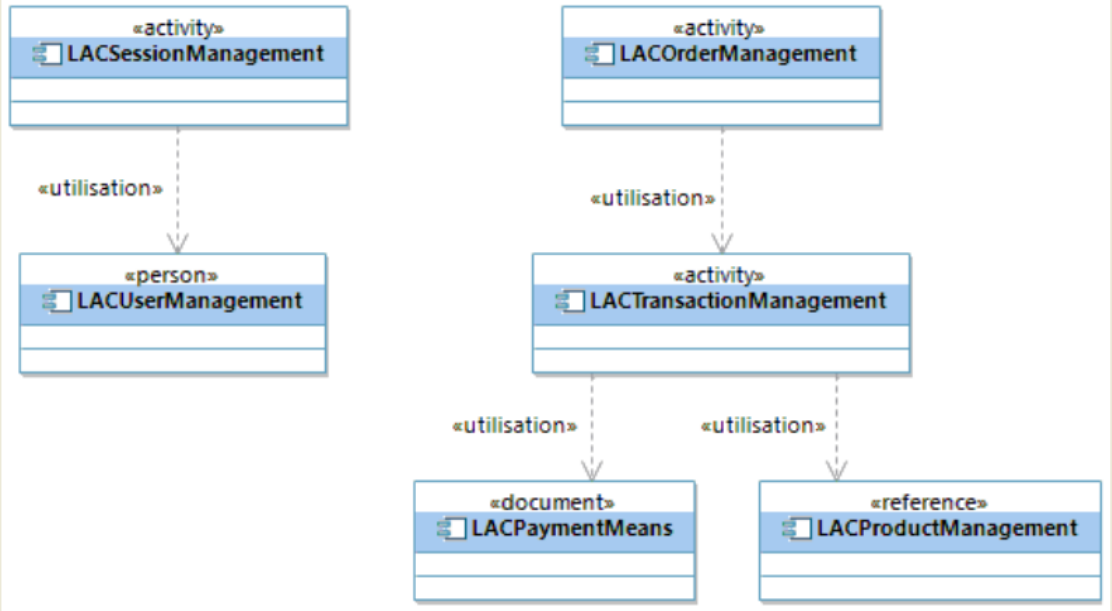

FIGURE 5.12 – TCM-LIS : Modèle d'architecture logique du SI fonctionnel supportant le domaine métier de l'achat en ligne

est représenté dans la Figure 5.12. Le pattern utilisé pour cette conception est fondé sur le cycle de vie des composants applicatifs logiques [157]. Il indique qu'un composant de type « activity » dépend d'un composants de type « person », ou d'un composant de type « document » ou de type « reference », si cette relation supporte le métier. Par exemple *LACSession-Management* de type « activity » dépend de *LACUserManagement* de type « person », et non l'inverse. Pour*LACOrderManagement* et *LACTransaction-Management* qui sont du même type « activity », c'est l'expertise métier liée à un portail qui préconise que c'est la commande qui dépend de la transaction et non l'inverse, puisque la commande est entérinée lorsque la transaction est achevée.

Le passage du CIM au PIM est donc réalisé par une succession de transformation avec les modèles suivants :

— Le **CIM** qui contient un ensemble de services métiers, composés de tâches. Afin d'être conforme au méta-modèle décrivant un service métier (Business Service) composé de tâches (Business Task) et où

chaque tâche manipule un attribut (Attribute) typé (type) d'un objet métier (Data Entity) et peut être contrainte par une condition (Condition), une boucle (Loop) ou une condition opérationnelle (Operating Condition), la spécification textuelle d'un service métier est reformulée à l'aide de balises. La reformulation est sous la responsabilité d'un analyste des exigences métiers. Dans l'étude de cas, la spécification de BSOnlineShopping (cf. Figure 5.1) est reformulée avec des balises (cf. Figure 5.13).

Cette reformulation met en évidence une condition de non nullité (<>null) pour la 1ère tâche qui spécifie que identifiant et mot de passe de l'utilisateur existent, une boucle pour traduire un ensemble d'articles dans la 3ème tâche, et enfin des conditions opérationnelles d'authentification pour la 1ère tâche et d'intégrité pour la 5ème tâche.

— Le **CICM-L (Computation Independent Contextual Model-Logical)** est le résultat de l'alignement de chaque attribut de donnée métier avec le composant applicatif logique du SI fonctionnel modélisé dans TCM-LIS (cf. Figure 5.12) qui le supporte. Cet alignement est sous la responsabilité d'un architecte logique. Le Tableau 5.5 représente cet alignement pour notre cas d'étude, alignement simplifié du fait de la conception d'un seul composant dans ce SI.

— Le PIM est le résultat de la transformation ET du modèle du CIM contextualisé par le risque et par les composants applicatifs logiques du SI fonctionnel. Cette transformation est contextualisée par (TCM-LR), de telle façon que chaque risque soit aligné avec le composant applicatif logique du SI technique qui le supporte. L'expert sécurité est le responsable de cet alignement. Le Tableau 3.2 représente cet alignement pour notre cas d'étude, alignement simplifié du fait de la conception d'un seul composant dans le SI technique. Tableau 5.7 – Alignement des risques associés à BSOnlineShopping et des composants applicatifs logiques du SI technique les supportant.

La transformation ET est implémentée avec le langage operational-

```
<Business Service><name>BSOnlineShopping</name>
<Business Task><order number>1</order number>
<Verb><crud>read</crud></Verb>
<Data Entity><name>User</name></Data Entity>
<Attribute><name>login</name><type>String</type></Attribute>
<Attribute><name>password</name><type>String</type></Attribute>
<Condition><guard><>null</guard></Condition>
<Operating Condition>Authentication</Operating Condition>
</Business Task>
<Business Task><order number>2</order number>
<Verb><crud>create</crud></Verb>
<Data Entity><name>Session</name></Data Entity>
<Attribute><name>opening</name><type>String</type></Attribute>
</Business Task>
<Business Task><order number>3</order number>
<Verb><crud>read</crud></Verb>
<Data Entity><name>Product</name></Data Entity>
<Attribute><name>identifier</name><type>Integer</type></Attribute>
<Attribute><name>name</name><type>String</type></Attribute>
<Attribute><name>cost</name><type>Real</type></Attribute>
<Loop><list>true</list></Loop>
</Business Task>
<Business Task><order number>4</order number>
<Verb><crud>read</crud></Verb>
<Data Entity><name>BankingCard</name></Data Entity>
<Attribute><name>number</name><type>Integer</type></Attribute>
<Attribute><name>expiration</name><type>String</type></Attribute>
<Attribute><name>code</name><type>Integer</type></Attribute>
</Business Task>
<Business Task><order number>5</order number>
<Verb><crud>create</crud></Verb>
<Data Entity><name>Transaction</name></Data Entity>
<Attribute><name>validation</name><type>String</type></Attribute>
<Attribute><name>amount</name><type>Real</type></Attribute>
<Operating Condition>Integrity</Operating Condition>
</Business Task>
<Business Task><order number>6</order number>
<Verb><crud>create</crud></Verb>
<Data Entity><name>Order</name></Data Entity>
<Attribute><name>date</name><type>String</type></Attribute>
<Attribute><name>reference</name><type>Integer</type></Attribute>
</Business Task>
<Business Task><order number>7</order number>
<Verb><crud>update</crud></Verb>
<Data Entity><name>Session</name></Data Entity>
<Attribute><name>closing</name><type>String</type></Attribute>
</Business Task>
</Business Service>
```

FIGURE 5.13 – CICM-R : Reformulation des spécifications du service métier d'achat en ligne (BSOnlineShopping) conformes au méta-modèle du CIM

Donnée métier (Data ENtity)	**Attribut (Attribute)**	**Composant applicatif logique**
User	login	LACUserManagemen
	password	
Session	opening	LACSessionManagemen
	closing	
Product	identifier	LACProductManagement
	name	
	cost	
BankingCard	number	LACPaymentMeans
	expiration	
	code	
Transaction	valisation	LACTransactionManagement
	amount	
Order	date	LACOrderManagemen
	reference	

Tableau 5.5 – CICM-R2CICM-L : Alignement des attributs des donnes métiers manipulées dans BSOnlineShopping et des composants applicatifs logiques du SI fonctionnel les supportant

Risque	**Composant applicatif logique**
Credentials Spoofing	LACRiskManagemen
Unauthorized	
Disruptive device	

Tableau 5.6 – CICM-L2PIM (ET) : Alignement des risques associés à BSOnlineShopping et des composants applicatifs logiques du SI technique les supportant

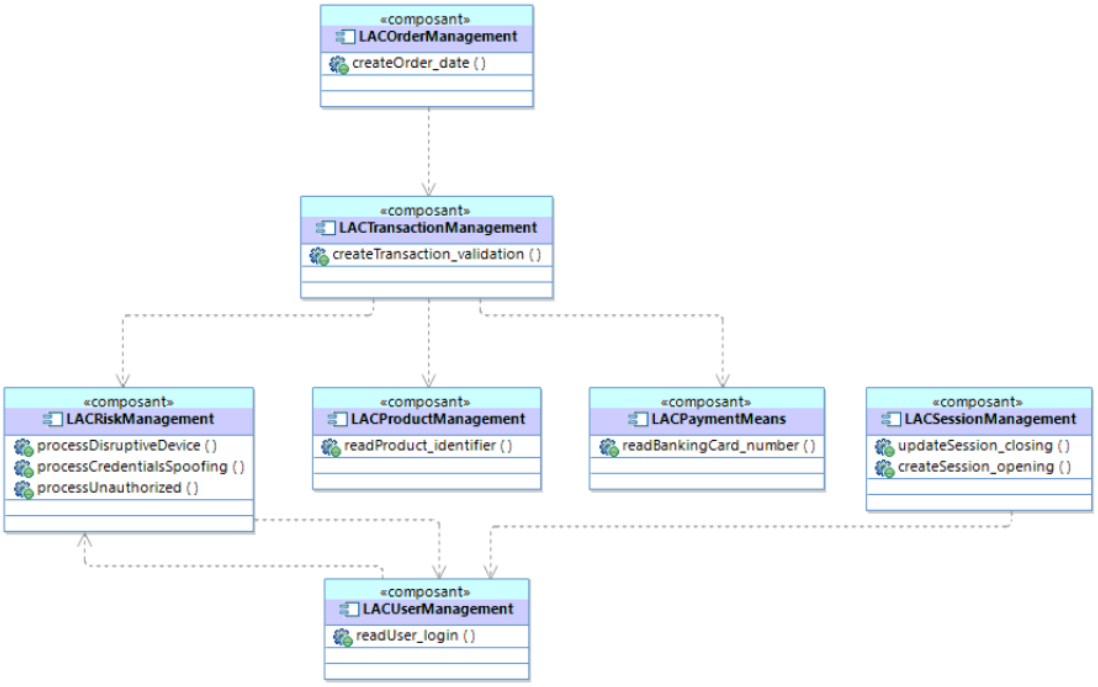

FIGURE 5.14 – PIM statique : Modèle de composants applicatifs logiques supportant le service métier BSOnlineShopping

QVT recommandé par l'OMG (Object Management Group) [59]. Les modèles suivants sont le résultat de ET chainée avec une transformation implémentée avec operational-QVT du PIM vers UML2 :

— Le modèle de composants applicatifs logiques où les composants du SI alignés avec les tâches du service métier sont complétés par les dépendances et les opérations logiques, opérations de ces composants, utiles au service métier.

Dans l'étude de cas (cf. Figure 5.14), les dépendances concernant le composant *LACRiskManagement*, soit vers *LACUserManagement* pour supporter le risque asynchrone Unauthorized, soit à partir de *LACTransactionManagement* pour supporter le risque synchrone Disruptive device ou de *LACUserManagement* pour supporter le risque synchrone Credentials spoofing complètent automatiquement le modèle des composants logiques extraits du SI avec leurs dépendances pour supporter le service métier *BSOni-*

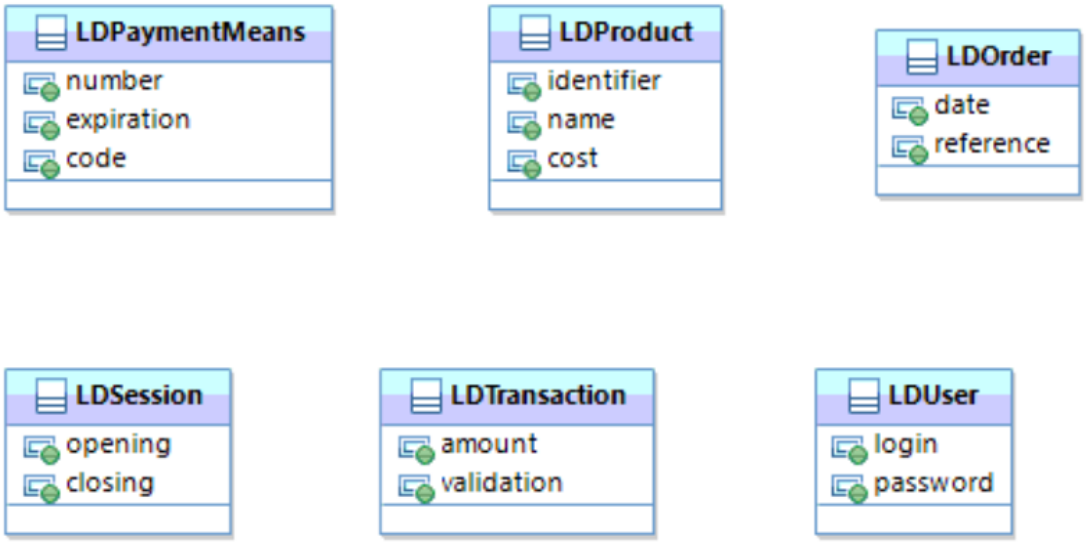

FIGURE 5.15 – PIM statique : Données logiques supportant le service métier BSOnline-Shopping

neShopping. Chaque opération logique est conçue directement à partir d'une tâche métier et d'un attribut de donnée métier produit par cette tâche (cf. Figure 5.13), qui permet grâce aux alignements précédents (Tableaux III et IV) de l'intégrer à un composant applicatif logique.

— Le modèle de données logiques où la conception des dépendances n'a pas été implémentée, puisqu'elle est conforme aux dépendances des composants applicatifs logiques qui les produisent.

La conception automatique des données logiques (cf. Figure 5.15) est fondée sur les données métiers, et leurs attributs, indiqués dans la spécification du service métier (cf. Figure 5.13).
La contrainte de cette conception est l'alignement avec les composants applicatifs logiques (cf. Tableau 3.1) puisqu'une donnée logique ne peut être produite que par un seul composant applicatif logique. Par exemple, la donnée logique LDPaymentMeans est produite par le composant applicatif logique LACPaymentMeans.

— La modélisation dynamique de l'architecture logique est représentée par les diagrammes de séquence associés, chacun, à un service du SI. Conformément à l'algorithme décrit dans [157], un service du SI est défini par un arbre de dépendances logiques.
Par exemple, le modèle de composants applicatifs logiques représenté Figure 5.14 fait clairement apparaitre deux arbres de dépendances logiques avec les deux sommets *LACOrderManagement et LACSessionManagement*.
La séquence de tâches du service métier (cf. Figure 5.13) est donc supportée, du fait des alignements (cf. Tableau 3.1), par un premier service du SI supportant les tâches 1 et 2 (arbre de sommet *LACSessionManagement)*, un deuxième service supportant les tâches 3, 4, 5 et 6 (arbre de sommet *LACOrderManagement)* et un troisième service supportant la tâche 7 (arbre de sommet *LACSessionManagement*). Ci-après, les diagrammes de séquence sont donc définis avec la conception d'un service du SI et par le support de tâches du service métier *BSOnlineShopping*.
— Le diagramme de séquence (Figure 5.16) représente le service $ISSOnlineShopping : createSessionOpening$ du SI qui supporte les tâches 1 et 2 du service métier.

La traduction du synchronisme du risque Credentials spoofing par rapport à la tâche de lecture des identifiant et mot de passe de l'utilisateur est notée par un ovale de couleur grise dans le diagramme de séquence.

Il traduit bien la force du couplage entre la requête de lecture des identifiant et mot de passe et le traitement du risque. Concernant l'asynchronisme du risque Unauthorized par rapport à cette même tâche, il est souligné par l'ovale de couleur noire.
Il traduit le fait que le traitement de ce risque nécessite d'avoir la donnée logique contenant identifiant et mot de passe en entrée. Ces deux représentations de dépendance entre une opération logique

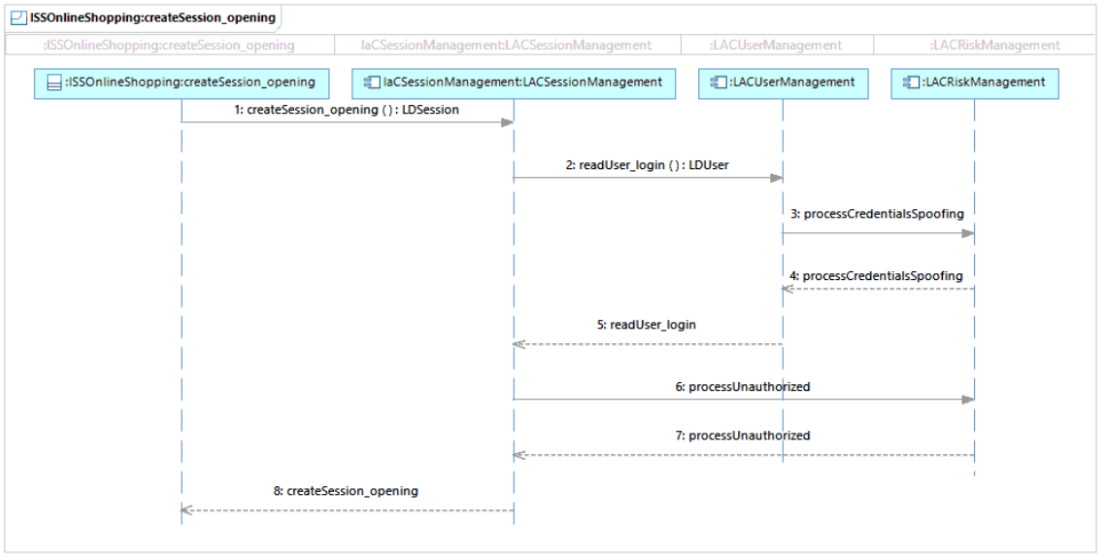

FIGURE 5.16 – PIM dynamique : Service ISSOnlineShopping createSession Opening du SI supportant le service métier BSOnlineShopping

($readUser_login$) et des traitements de risque ($processCredentialsSpoofing$ et $processUnauthorized$) soulignent aussi le fait que la requête d'un traitement de risque ne peut entrainer une requête d'une opération logique de type fonctionnel.

— Le diagramme de séquence Figure 5.17représente le service $ISSOnlineShoppingcreateOrderdate$ du SI qui supporte les tâches 3, 4, 5 et 6 du service BSOnlineShopping.

L'ovale gris souligne le synchronisme entre le risque supporté par l'opération logique $processDisruptiveService$ et la création de la transaction avec la banque supporté par l'opération logique $createTransactionvalidation$.

— Le diagramme de séquence de la Figure 5.18 représente le service *ISSOnlineShopping :updateSession Closing* du SI qui supporte la tâche 7 du service métier $BSOnlineShopping$.

— Les modèles des services du SI sont complétés par leur orchestration dans le diagramme de séquence Figure 5.19 afin de supporter le service métier BSOnlineShopping.

La conception automatique du PIM recouvre ainsi l'architecture

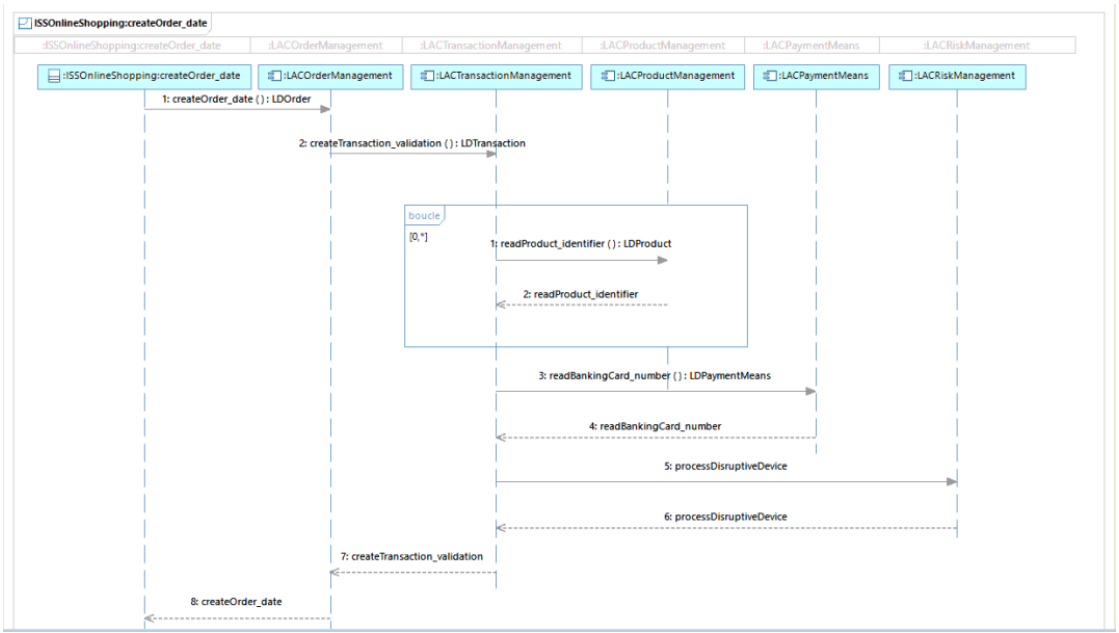

FIGURE 5.17 – PIM dynamique : Service ISSOnlineShopping create ORDER du SI supportant le service métier BSOnlineShopping

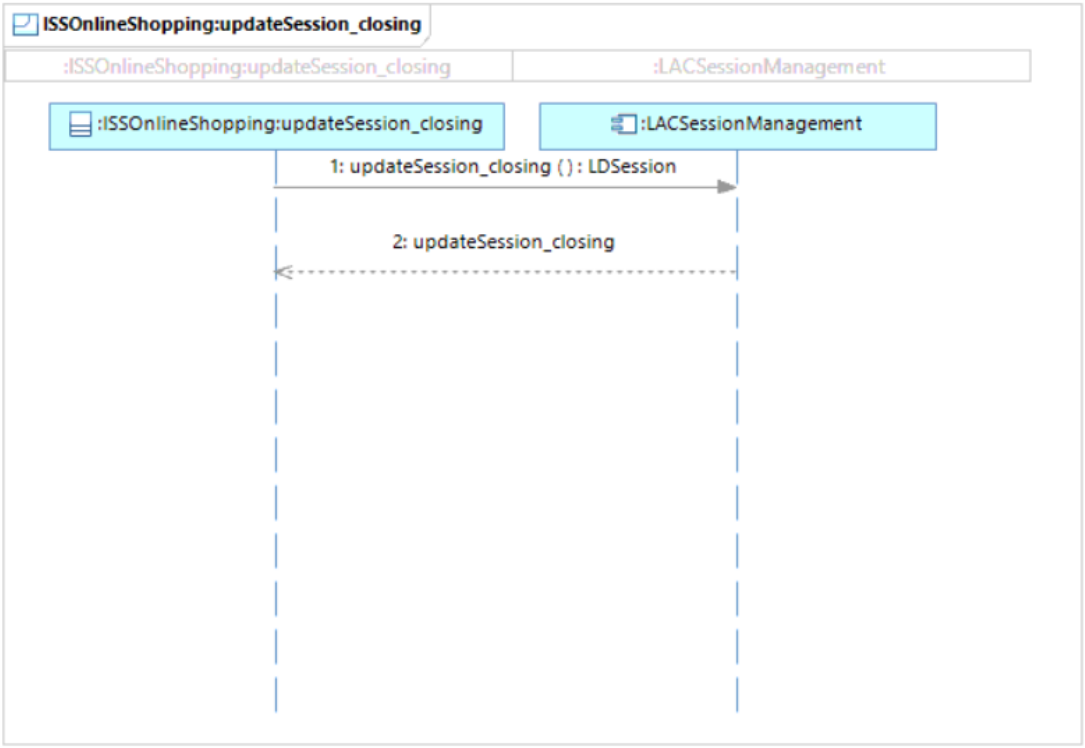

FIGURE 5.18 – PIM dynamique : Service ISSOnlineShopping :updateSession closing du SI supportant le service métier BSOnlineShopping

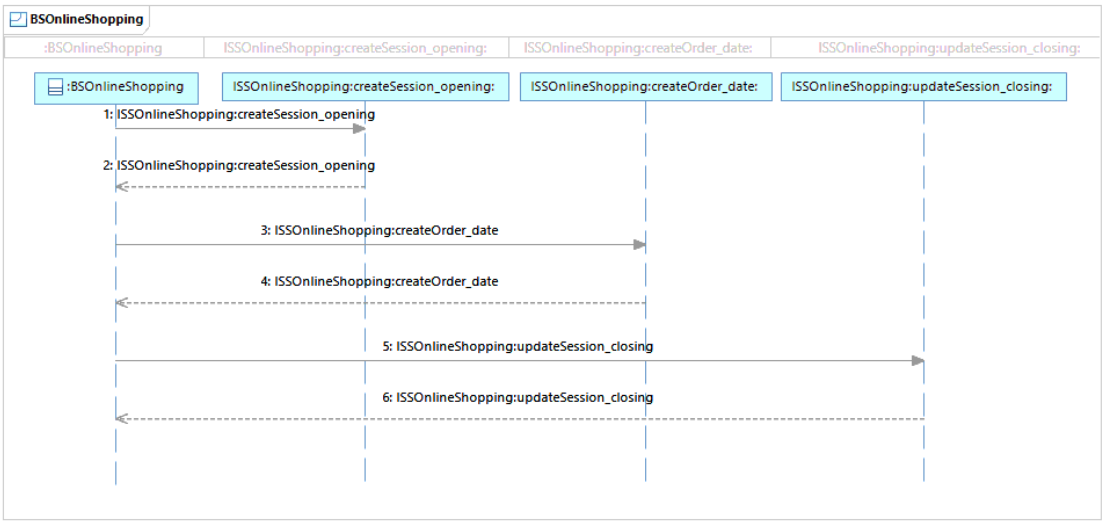

FIGURE 5.19 – PIM dynamique : Modèle dynamique d'orchestration des services du SI supportant le service métier BSOnlineShopping.

logique statique supportant le service métier BSOnlineShopping avec le modèle des composants applicatifs logiques et le modèle de données logiques et son architecture logique dynamique avec les diagrammes de séquence représentant les services du SI supportant le service métier.

Ces deux aspects de l'architecture logique permettent de concevoir le PSM spécifique à une plateforme technique.

4. **Du PIM au PSM avec un contexte lié aux risques**

Pour l'intégration du traitement du risque lors de la transformation du modèle d'architecture logique en un modèle d'architecture physique, code inclus, un seul modèle contextuel est utilisé :

— **TCM-PR** offre une substitution du codage d'opérations logiques déployées sur une plateforme technique par un service existant du SI technique.

Le modèle contextuel recommandé dans [157], et non mis en œuvre dans cette étude de cas, est :

— **TCM-PIS (Physical Information System)**, qui, de manière analogue

à TCM-PR, est la description des services du SI recommandés pour implémenter les opérations logiques encapsulées dans les composants applicatifs logiques du SI supportant les exigences métiers de nature fonctionnelle, cette fois, du métier ciblé. Dans notre étude de cas, TCM-PIS ne contient aucun service du SI utile à la réalisation de BSOnlineShopping.

Plus classique dans l'approche MDA, le PDM (Platform Description Model) décrit une plateforme avec un environnement JEE (en particulier, des EJB (Enterprise JavaBeans) [7]) et un système de gestion de base de données relationnelle SQL. Ce modèle permet de générer automatiquement, en plus de l'architecture physique du système, le script de génération de la base de données et un squelette du code des services du SI dans un cadre EJB. De plus, cette architecture est une architecture en 3 couches (Présentation (HMI : Human-Machine Interface), Métier (service) et Donnée (accès aux données)) [137].

Sachant que PICM-P (Platform Independent Contextual Model-Physical) n'est pas pertinent pour notre étude de cas, le passage du CIM au PIM est réalisé par une succession de transformation avec les modèles suivants :

— **Le PICM-R (Platform Independent Contextual Model-Risk)** est le résultat de l'alignement des opérations logiques du PIM et de la plateforme conçue par l'architecte technique du système avec les services du SI existant et réutilisables décrits dans TCM-PR (cf. Tableau 5.4). Le Tableau 4.2 représente cet alignement qui peut être réalisé par un architecte physique ou applicatif du système ou de manière automatique si un catalogue de ces services existe.

BSOnlineShopping		**Services réutilisables du SI**		
Opération logique	**Plateforme technique}**	**Service du SI**	**Opération(s) logique(s) implémentées**	**\textbf{Plateforme de déploiement}**
processCredentialsSpoofing	EN Web Server	ISSKeePass	processCredentialsSpoofing	EN Web Server
processDisruptiveDevice		ISSArchi2Secu	processDisruptiveDevice	

Tableau 5.7 – PICM-P2PICM-R : Alignement des opérations logiques et de la plateforme technique supportant BSOnline-Shopping avec les services réutilisables du SI

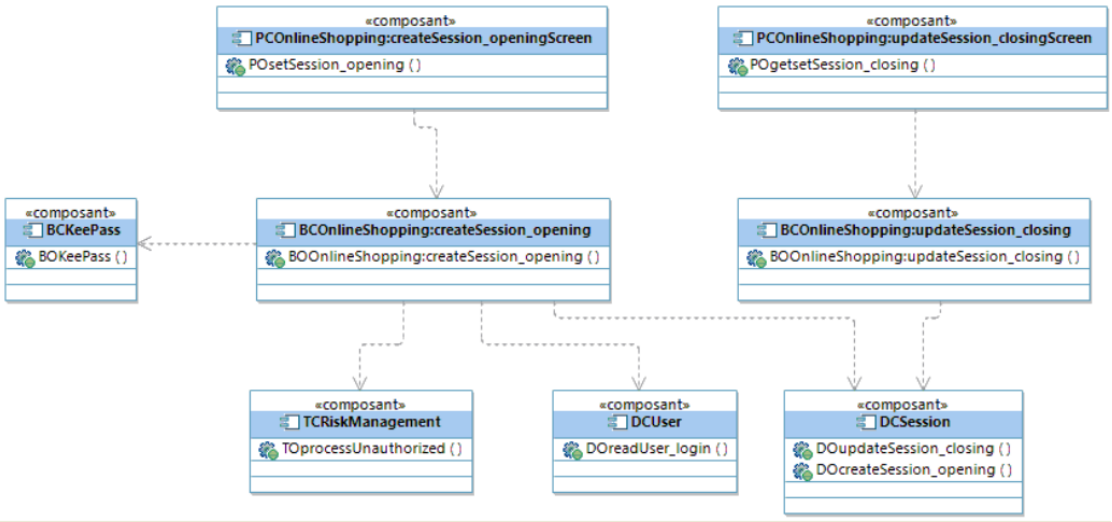

FIGURE 5.20 – PSM statique : Modèle de composants applicatifs physiques réalisant l'arbre de dépendances de sommet LACSessionManagement

— **Le PSM** est le résultat de la transformation ST, du PIM obtenu précédemment, contrainte par le PDM avec un architecture JEE et SQL. Cette transformation est implémentée avec operational-QVT.
Les modèles suivants sont le résultat de ST chainée avec une transformation du PSM vers UML2 implémentée aussi avec operational-QVT :

— Le modèle de composants applicatifs physiques est scindé, afin d'en faciliter la lecture, en une partie dédiée à la réalisation de l'arbre de dépendances de sommet LACSessionManagement (cf. Figure 5.20) et une partie dédiée à la réalisation de l'arbre de dépendances de sommet LACOrderManagement (cf. Figure 5.21). Pour mémoire, par rapport à l'architecture en 3 couches décrite dans le PDM, PC (respectivement PO) indique un composant (respectivement une opération) de la couche Présentation, BC (respectivement BO) indique un composant (respectivement une opération) de la couche Métier, DC (respectivement DO) indique un composant (respectivement une opération) de la couche Donnée. Les dépendances sont fixées par le pattern d'architecture associé. Une couche Technique est ajoutée afin de réaliser les opérations logiques telles que pour le composant TCRiskManagement.

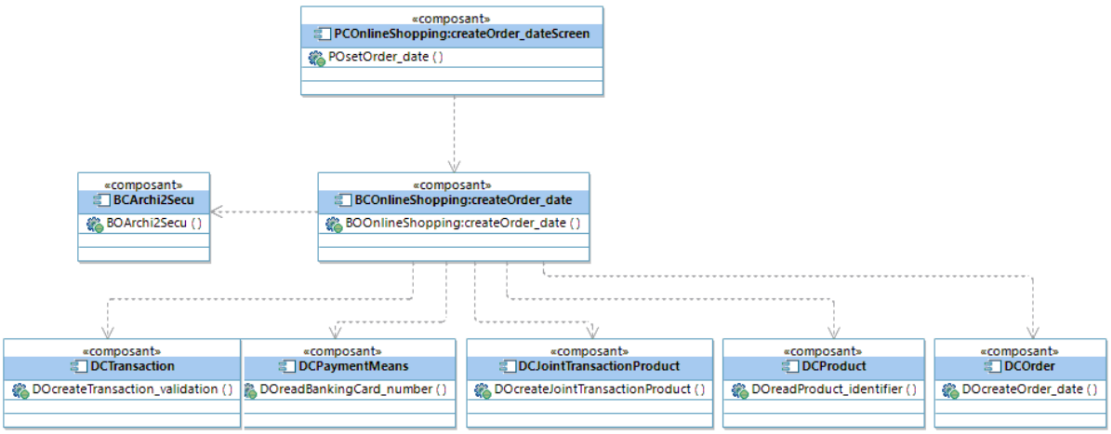

FIGURE 5.21 – PSM statique : Modèle de composants applicatifs physiques réalisant l'arbre de dépendances de sommet LACOrderManagement

Dans ce modèle de composants applicatifs, l'appel à des services réutilisables du SI est conçu sur la couche Métier. C'est le cas dans ce modèle avec l'appel à l'opération de BCKeePass. De même, les deux services du SI sont conçus sur cette couche Métier avec les opérations offertes par $BCOnlineShopping : createSessionopening$ et $BCOnlineShopping : updateSessionclosing$.

Il en est de même ici avec l'appel à l'opération de BCArchi2Secu sur la couche Métier.

— Le modèle de données physiques où les dépendances sont représentées par des clés étrangères ($_fk$) du fait des contraintes SQL modélisées dans le PDM.

La conception automatique des données physiques (cf. Figure 5.22) est fondée sur la réalisation des données logiques, mais elle doit aussi prendre en compte chaque boucle. En effet, chaque boucle induira une jointure pour réaliser la dépendance vers la donnée logique en boucle et à partir de la donnée logique qui en dépend. Dans le modèle généré, la donnée de jointure JointPDTransactionPDProduct réalise la dépendance de la donnée logique LDTransaction vers LDProduct (conforme à la dépendance de LACTransactionManagement vers LACProduct-Management), qui est la cible de la boucle. o Le modèle dynamique

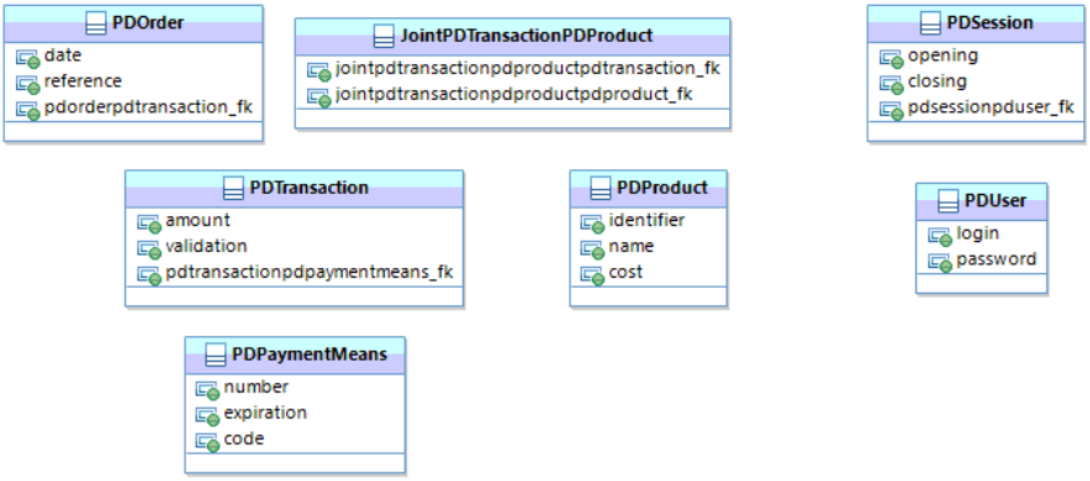

FIGURE 5.22 – PSM statique : Données logiques supportant le service métier BSOnlineShopping.

de l'architecture physique avec les diagrammes de séquence associés aux services du SI générés dans le §2 et des instances des composants applicatifs physiques (cf. Figure 5.20 et Figure 5.21). Ces diagrammes sont générés à partir du pattern de couches modélisé dans le PDM.

— Le diagramme de séquence Figure5.23 représente l'architecture physique dynamique du service $SSOnlineShoppingcreateSessionopening$ qui réalise son architecture logique (cf. Figure 5.16).

De la même façon, l'ovale gris souligne la représentation d'un appel synchrone pour le service réutilisé BOKeePass alors que l'ovale noir cible la représentation d'un appel asynchrone avec l'opération technique TOprocessUnauthorized.

— Le diagramme de séquence Figure 5.24 représente l'architecture physique dynamique du service $ISSOnlineShopping : createOrder_date$ qui réalise son architecture logique (cf. Figure 5.17).

En plus de la signification synchrone de l'ovale gris pour BOArchi2Secu, la génération du modèle UML du diagramme de séquence implémente la conception d'une boucle intégrant non seulement l'opération physique de lecture du produit *DOreadProduct identifier*, mais

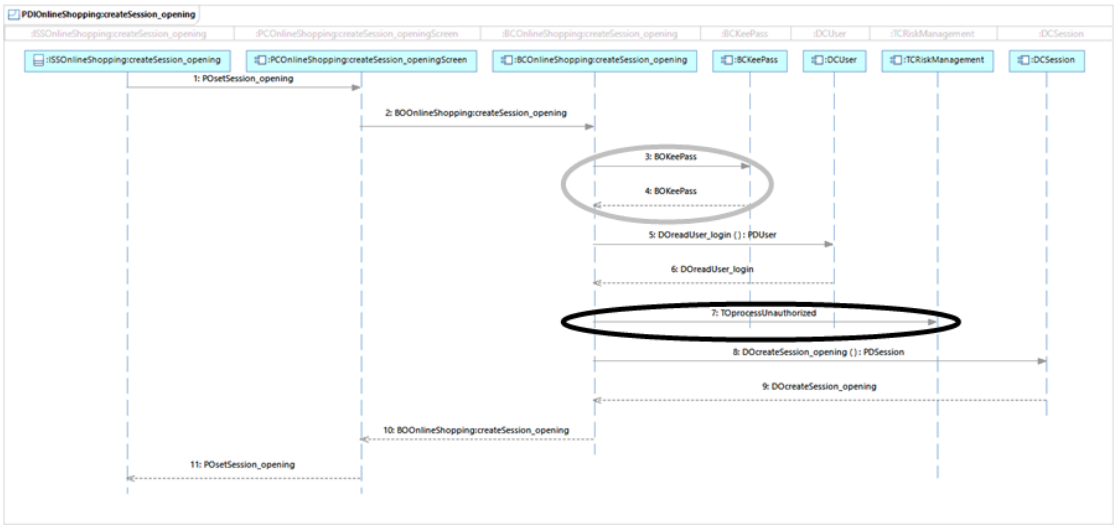

FIGURE 5.23 – PSM dynamique : Modèle physique dynamique du service ISSOnlineShopping :create Session opening du SI

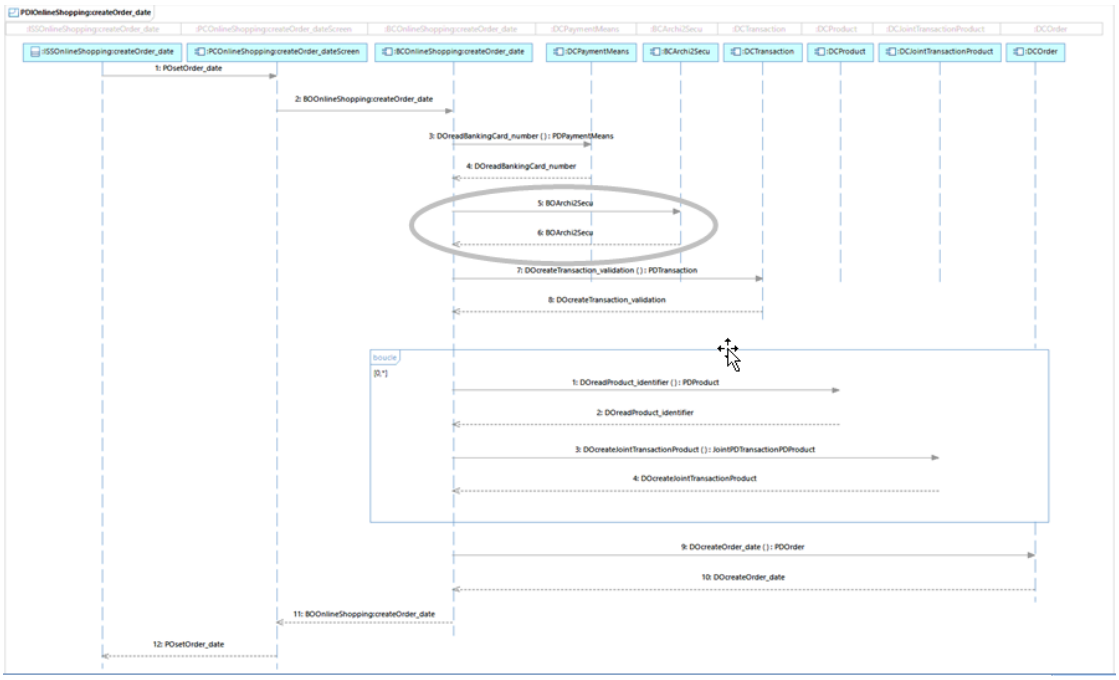

FIGURE 5.24 – PSM dynamique : Modèle physique dynamique du service ISSOnlineShopping : create Order date du SI

aussi l'opération physique de création de la jointure *DOcreateJointPDTransactionPDProduct* grâce à la transaction précédemment créée *DOcreatePDTransaction validation*.

— Le diagramme de séquence Figure 5.25 représente l'architecture physique dynamique du service $ISSOnlineShopping : updateSession_closing$ du SI qui réalise son architecture logique (cf. Figure 5.18).

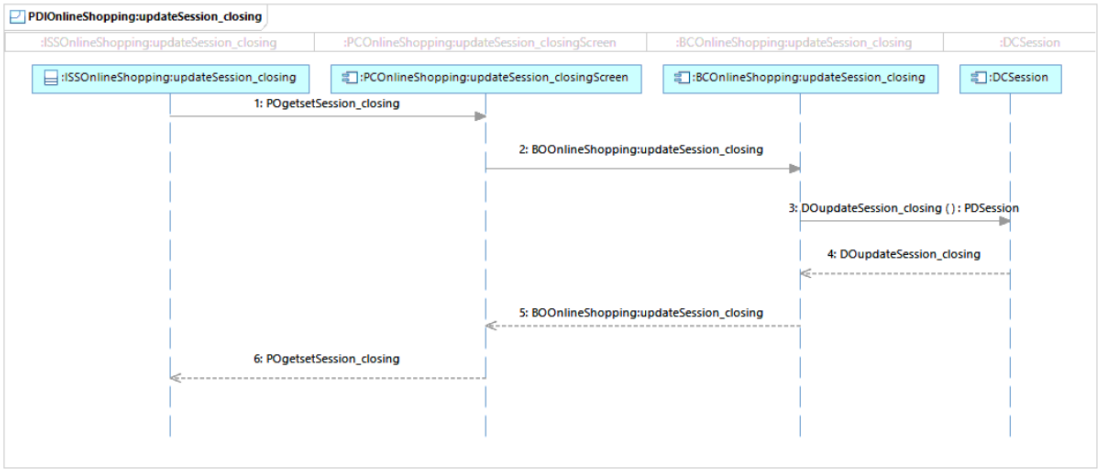

FIGURE 5.25 – PSM dynamique : Modèle physique dynamique du service ISSOnlineShopping :updateSession$_closingduSIralisantsonmodlephysiquelogique$

— Le script de la génération SQL de la base de données relationnelles (cf. Figure 5.26) conforme au modèle de données physiques (cf. Figure 5.15).
— Le squelette du code Java des services du SI (cf. Figure 5.27) cohérent avec le modèle d'architecture physique dynamique et ses trois diagrammes de séquence (cf. Figure 5.23, Figure 5.24 et Figure 5.25).

5.2 Vérification du cas d'étude

La vérification consiste à vérifier l'alignement des séquences d'opérations physiques constituant les trois services orchestrés (cf. Figure 5.19 - PIM dynamique : Modèle dynamique d'orchestration des services du SI supportant le service métier BSOnlineShopping) dans les diagrammes de séquence physiques (cf. Figure 5.23 – PSM dynamique : Modèle physique dynamique du service ISSOnlineShopping :create Session opening du SI ; Figure 5.24 – PSM dynamique : Modèle physique dynamique du service ISSOnlineShopping : create Order date

```
@Override
public void BOOnlineShopping:updateSession_closing(String openingSession, String closingSession)
{
   sessionDAO.DOupdateSession_closing(session);
}

@Override
public void BOOnlineShopping:createOrder_date(int identifierProduct, String nameProduct, float costProduct, int
numberPaymentMeans, String expirationPaymentMeans, int codePaymentMeans, float amountTransaction, String
validationTransaction, String dateOrder, int referenceOrder)
{

   paymentmeansDAO.DOreadBankingCard_number(paymentmeans);

   this.BOArchi2Secu();

   transactionDAO.DOcreateTransaction_validation(transaction);

   /**
    * Loop start
    */

      productDAO.DOreadProduct_identifier(product);

      jointtransactionproductDAO.
                  DOcreateJointTransactionProduct
                  (jointtransactionproduct);

   /**
    * Loop end
    */
   orderDAO.DOcreateOrder_date(order);
}

@Override
public void BOOnlineShopping:createSession_opening(String loginUser, String passwordUser, String openingSession, String
closingSession)
{

   this.BOKeePass();

   userDAO.DOreadUser_login(user);
   /**
    * Exception if not: <>null
    */


   /**
    * Appel asynchrone d'une opération technique
    */
   TOprocessUnauthorized();

   sessionDAO.DOcreateSession_opening(session);
}
```

FIGURE 5.26 – Script SQL de génération de la base de données physiques relationnelles

du SI ; Figure 5.25 – PSM dynamique : Modèle physique dynamique du service ISSOnlineShopping :updateSessionclosingduSI...) avec la séquence de tâches du service métier (cf. Figure 5.1 - Spécifications du service métier d'achat en ligne

```
@Override
public void BOOnlineShopping:updateSession_closing(String openingSession, String closingSession)
{
   sessionDAO.DOupdateSession_closing(session);
}

@Override
public void BOOnlineShopping:createOrder_date(int identifierProduct, String nameProduct, float costProduct, int
numberPaymentMeans, String expirationPaymentMeans, int codePaymentMeans, float amountTransaction, String
validationTransaction, String dateOrder, int referenceOrder)
{

   paymentmeansDAO.DOreadBankingCard_number(paymentmeans);

   this.BOArchi2Secu();

   transactionDAO.DOcreateTransaction_validation(transaction);

   /**
    * Loop start
    */

      productDAO.DOreadProduct_identifier(product);

      jointtransactionproductDAO.
                  DOcreateJointTransactionProduct
                  (jointtransactionproduct);

   /**
    * Loop end
    */
   orderDAO.DOcreateOrder_date(order);
}

@Override
public void BOOnlineShopping:createSession_opening(String loginUser, String passwordUser, String openingSession, String
closingSession)
{

   this.BOKeePass();

   userDAO.DOreadUser_login(user);
   /**
    * Exception if not: <>null
    */


   /**
    * Appel asynchrone d'une opération technique
    */
   TOprocessUnauthorized();

   sessionDAO.DOcreateSession_opening(session);
}
```

FIGURE 5.27 – Squelette du code Java des services du SI.

(BSOnlineShopping)).

Dans l'alignement du service *ISSOnlineShopping : createSession opening* avec

le service métier BSOnlineShopping, la tâche 1 précède bien la tâche 2 : OK.
Dans l'alignement du service *ISSOnlineShopping : updateSession closing*, dernier service de l'orchestration, avec le service métier *BSOnlineShopping*, on retrouve bien la dernière tâche du service métier qui est la tâche 7 : OK.
Par contre, dans l'alignement du service *ISSOnlineShopping : createOrder date* avec le service métier BSOnlineShopping, on a la séquence : tâche 4, puis tâche 5, puis tâche 3, et enfin tâche 6.
Par ailleurs l'opération *DOcreateJointPDTransactionPDProduct* n'est pas alignée avec une tâche métier, puisque créant une donnée de jointure imposée par la base de données relationnelle (choix d'architecture technique) : NOK.

En fait, ce désalignement résulte de deux règles de conception d'une opération d'accès à une donnée de jointure telles qu'implémentées dans la transformation PICM-R2PSM.
Ces deux règles sont les suivantes :

— la clé étrangère commune à un ensemble de jointures (ici celle de PDTransaction) doit être instanciée en amont de celles des clés étrangères multiples associées (ici celles de PDProduct représentant les différents produits commandés).
— la création de la donnée de jointure DOcreateJointPDTransactionPDProduct doit être instanciée dans la même boucle que l'instanciation de la donnée fournissant les clés étrangères multiples (n produits => n jointures).

Pour **ISSOnlineShopping :createOrder_date**, l'application de ces deux règles signifie que la tâche 5 de création de **PDTransaction** doit précéder la boucle réunissant l'opération de sélection de produit et celle de création de donnée de jointure, d'où le désalignement.

L'impact sur la spécification du service métier est intéressant car ce désalignement induit, pour être corrigé, une nouvelle spécification des tâches 3, 4 et 5 du service métier avec une scission de la tâche 5 où sont séparés l'accès au serveur bancaire (sécurité) de l'acquisition au montant de la transaction (commerciale).

Le service métier aligné avec l'architecture physique serait alors :

1. Lire identifiant et mot de passe, de l'utilisateur, qui existent (tâche alignée avec la propriété de sécurité d'authentification)
2. Créer la session pour l'utilisateur avec la date d'ouverture de la session
3. Lire les détails de la carte bancaire avec le numéro, la date d'expiration et le code CVV
4. Créer la transaction avec la banque avec sa date (tâche alignée avec la propriété de sécurité d'intégrité)
5. Sélectionner un ensemble d'articles caractérisés par leur code, leur libellé et leur prix
6. Mettre à jour la transaction avec la banque avec le montant total
7. Créer la commande avec sa date et sa référence
8. Modifier la session pour l'utilisateur avec la date de fermeture de la session

L'intérêt de l'alignement du métier sur le physique (en mode rétro) est de séparer le traitement de la sécurité (tâche 3 et tâche 4) du traitement commercial (tâche 5 et tâche 6) et donc de justifier l'approche liée à la sécurité proposée dans la thèse.

5.3 Conclusion et discussion du cas d'étude

Il ya un apport de l'approche MDA contextualisée pour l'intégration du risque vue comme un ensemble de contextes propres à chacune des vues d'architecture (métier, logique, physique) dans un développement de système.
L'intégration du risque au niveau métier impose d'anticiper la collecte des exigences de sécurité dès la spécification du service métier. Cette intégration se fait via les propriétés de sécurité qui sont des concepts partageables avec les experts du métier.
Seuls les alignements ne sont pas automatisés et la génération des modèles facilitée par la prise en compte du modèle logique du SI et des services réutilisables

du SI a un coût nettement diminué.
Néanmoins, ces alignements nécessitent des doubles expertises :
- Métier + Sécurité pour l'alignement du risque sur les propriétés de sécurité, elles-mêmes alignées sur des tâches du service métier ;
- Métier + Sécurité pour la synchronisation du traitement d'un risque avec une tâche métier ;
- Logique ou Fonctionnel + Sécurité pour la conception des composants applicatifs logiques dédiés à la sécurité dans le SI ;
- Physique ou Applicatif + Sécurité pour la réutilisation de services du SI à vocation de sécurité.

Deux thèmes de discussion semblent découler du cas d'étude, le premier sur les rôles concernés par le développement d'un système, le second sur les modifications impactant les principes d'architectures logique et d'architecture physique.

1. **Rôles autour du développement**
   Ces besoins de double expertise sont intéressants car ils permettent de mieux intégrer les experts techniques, ici pour la sécurité, lors des différentes phases du développement.
   L'utilisation automatisée de l'architecture logique du SI lors du développement d'un service métier modifie le métier de l'architecte dont la responsabilité devient centrée sur les alignements pour l'aspect logique de nature fonctionnelle ou lié à la sécurité. Ceci induit plus d'échanges avec les experts métiers afin de mieux définir les exigences du service métier.

2. **Architecture logique et architecture physique**
   L'intégration de composants logiques techniques introduit la possibilité de cycles qui sont fortement déconseillés pour les composants logiques de nature fonctionnelle. Dans le cas d'étude, il existe en un cycle car LACRiskManagement dépend de LACUserManagement, et inversement. De même, la conception de composants applicatifs physiques de nature technique, tel que TCRiskManagement, met en cause le pattern de l'archi-

tecture en couches choisi dans l'étude de cas. En effet, un composant de sécurité, dont dépend un composant de la couche Métier, ne dépend pas d'un composant d'accès aux données. C'est une perspective de ce travail de définir les données manipulées par des opérations techniques, données souvent qualifiées de paramètres liés à la configuration d'un système.

Conclusion Générale

Révue de recherche

L'intégration des aspects de sécurité et de risque dans le processus global du cycle de vie de développement des systèmes logiciels est une tâche difficile. Le plus souvent, il existe un écart entre les spécifications de sécurité définies au stade du recueil des besoins du système et leur mise en œuvre au stade de l'implémentation du système.

L'une des causes, est l'inadaptation des méthodes de développement existantes face aux systèmes applicatifs actuels, qui sont eux mêmes composés de plusieurs autres systèmes juxtaposés et interoppérants pour satisfaire des besoins de fonctionnalités de l'entreprise. Cela créé un contexte de complexité des systèmes d'aujourd'hui avec des vulnérabilités résultant de leur développement et de la manière dont ces systèmes interagissent dans l'organisation. Une autre cause est le manque d'expert en sécurité dans le processus de developpement des systèmes logiciels avec pour conséquence la mauvaise expression des besoins de sécurité et l'application de solutions de sécurité inappropriées.

Dans cette thèse, nous avons proposé une approche de développement de systèmes logiciels, en utilisant une architecture d'entreprise comme base pour la définition de modèles contextuels, outillé par l'ingénieurie Dirigée par les Modèles (IDM) et intégrant les préoccupations de sécurité fondées sur le risque.

A cette fin, un certain nombre d'objectifs ont été défini :

Révue des objectifs de recherche

— Définir les artefacts requis pour la conception et le developpement du système : notre approche se fonde sur les structures des frameworks d'EA (TOGAF et SABSA) pour définir des modèles contextuels (fonctionnels et sécurité) propres à chaque niveau architectural (architecture métier, architecture du SI et architecture physique).
— Définir un processus d'ingénierie logicielle basé sur l'approche IDM, permettant de prendre en compte à la fois l'aspect métier (fonctionnel) et l'aspect de sécurité (non-fonctionnel) : notre approche fournit un processus de développement fondé sur MDA afin de guider les concepteurs dans leur démarche. Ce processus est constitué d'un ensemble d'étapes (ingénierie des exigences, conception, implémentation) correspondant respectivement aux niveaux d'abstraction (CIM, PIM et PSM).
— définir une méthode permettant d'intégrer les différents modèles contextuels (dès la phase d'ingénierie des exigences) : notre méthode intègre les aspects de sécurité dans les différentes phases de l'ingénierie des systèmes avec l'intégration par transformation des modèles contextuels conçus.
— Expérimenter cette approche à l'aide d'une étude de cas.

Notre approche a répondu aux questions suivantes :

Questions de recherche

Les questions de recherche sont dérivées des objectifs de recherche. Trois questions de recherche principales sont au cœur de la recherche menée. Les questions de recherche et leurs sous-questions sont décrites ci-dessous.

QR1 : Comment intégrer la sécurité dans l'architecture métier associée au processus d'ingénierie système, via les risques ?
La première question de recherche est divisée en deux sous-questions :

QR1.1 : Quels éléments de l'architecture métier peut-on cibler pour une intégration de la sécurité dans le système ?

QR1.2 : L'approche MDA pour l'intégration de la sécurité au niveau métier est-elle pertinente comme mécanisme ?

QR2 : Quelle méthode de conception de la sécurité, via les risques , pour l'architecture logique supportant une sécurité intégrée dans le métier ?
La deuxième question de recherche est divisée en deux sous-questions :
QR2.1. Comment intégrer la sécurité à la vue logique d'un système à partir de la vue métier ?
QR2.2. Comment intégrer les composants logiques de sécurité à la réalisation dynamique des cas d'utilisation d'un système ?

QR3 : Quel impact de la mise en œuvre de la sécurité via les risques dans l'architecture physique et les conséquences sur l'implémentation du système ?

Révue des contributions

La contribution principale de notre thèse est l'intégration d'exigences de sécurité basées sur le risque dans le processus de développement de système logiciel. Cela a consisté en la définition de méta-modèles contextuels de sécurité basés sur l'architecture d'entreprise qui ont été intégrés à l'aide d'outil de modélisation. Il s'agit de modèles contextuels des architectures métier, logique et physique présentés comme suit :

Contribution 1

- Nous avons défini un processus de gestion de la sécurité basé sur l'analyse des menaces STRIDE et l'évaluation du risque avec EBIOS. Les étapes de ce processus, fondées sur les structures du framework TOGAF, enrichies par SABSA, offrent un guidage pour de la définition des méta modèles contextuels du risque (réponse à QR1).
- Le premier contexte est métier : TCM-BR. C'est un contexte à la description du besoin métier d'un système permettant de mieux concevoir l'architecture métier supportée par le système au sens de son exhaustivité(réponse à QR1.1). La particularité est la prise en compte des besoins de sécurité du système à un

haut niveau d'abstraction (CIM), dérivés en exigences de sécurité. Les besoins de sécurité au niveau métier sont intégrées par transformation contextuelle par enrichissement de modèles du CIM : CIM2CICM-R (réponse à QR1, QR1.2). Le résultat de cette transformation est l'architecture metier du système enrichie par le contexte du risque métier.
- Ces exigences (métier et de sécurité) sont ensuite reformulées grace à l'outil XML, héritées et implémentées dans les architectures inférieures (logique (réponse à QR2), physique) , suivant les étapes du processus de développement.

Contribution 2

- Le deuxième contexte est logique (fonctionnel), intégrant des composants logiques de sécurité à l'architecture cible du SI : TCM-LR. Ce contexte est intégré avec la description du besoin métier du système(réponse à QR2). L'objectif est toujours de mieux concevoir le niveau logique du système en étant cohérent avec la vue logique du SI intégrant la sécurité.
- Nous avons étendu l'approche UMLsec par la définission de profiles UMLs représentant les exigences de sécurité de l'architecture logique fondées sur les menaces et le risque. Un arbre de dépendance logique, fondé sur l'algorithme défini dans [157], nous a permis d'assurer la cohérence et le lien entre les architectures métier et celles inférieures(réponse à QR2, QR2.1). Enfin la transformation de modèles CIC-R2PIM par enrichissement (ET) nous a permis d'intégrer les modèles concernés pour obtenir l'architecture de conception logique du système (réponse à QR2.2).

Contribution 3

le troisième contexte est l'architecture physique ou applicative de l'existant au niveau du système informatique de l'entreprise. Cette étape est consacrée à la définition du modèle contextuel de sécurité TCM-PR conforme au niveau d'abstraction PIM.
Ce contexte est intégré avec l'architecture logique du système afin de réutiliser au mieux les solutions applicatives (telles que les services physiques) existantes de sécurité dans le SI(réponse à QR3).
Le processus de transformation contextuelle par substitution CTs et par substitu-

tion ST du PIM vers le PSM : PIM2PSM nous a permis de produire l'architecture dynamique applicative de sécurité du système avec la génération du script de la base de données et du squelette du code applicatif (réponse à QR3).
La particularité de notre contribution est de faciliter la réutilisation de solutions outillées de sécurité, plutôt sur étagère.
Ces contributions sont soutenues par un cas d'étude (eCommerce) en guise d'illustration de notre méthode.

Perspectives

De façon générale, à cours terme, notre méthode doit être expérimentée, dans un cadre industriel, par des spécialistes en cybersécurité pour une évaluation concrète.

Dans contribution 1 : L'entreprise étant en perpétuelle mutation avec de nouvelles ménaces et vulnérabilités émergeantes, une prise en compte du changement dans notre approche permettra son adaptabilité (mis à jour) aux nouveaux risques. Pour ce faire, une approche basée sur les outils d'intélligence artificielle est appropriée pour l'analyse des cyber-menaces de plus en plus sophistiquées.

Dans Contribution 2 : l'approche UMLsec offre le moyen de vérifier de façon formelle les modèles (stéréotypes de sécurité) conçus. Notre méthode pourrait être vérifiée pour formaliser les modèles de sécurité dédiés à la définition des exigences de sécurité telles que l'authentification, l'intégrité.

Bibliographie

[1] Muhammad Alam, Ruth Breu et Michael Hafner. « Model-driven security engineering for trust management in SECTET. » In : *J. Softw.* 2.1 (2007), p. 47-59.

[2] Unai Alegre, Juan Carlos Augusto et Tony Clark. « Engineering context-aware systems and applications : A survey ». In : *Journal of Systems and Software* 117 (2016), p. 55-83.

[3] Abdullah S Alghamdi. « Evaluating defense architecture frameworks for C4I system using analytic hierarchy process ». In : *Journal of Computer Science* 5.12 (2009), p. 1075.

[4] Bandar Alshammari. « Enterprise Architecture Security Assessment Framework (EASAF). » In : *J. Comput. Sci.* 13.10 (2017), p. 558-571.

[5] Annie I Antón et Julia B Earp. « A Multidisciplinary Electronic Commerce Project Studio for Secure Systems ». In : *4th National Colloquim for Information Systems Security Education (NCISSE)* (2000).

[6] Rudolph Araujo et Shanit Gupta. « Design authorization systems using secureUML ». In : *Foundstone Professional Services* (2005), p. 2-16.

[7] John Arthur et Shiva Azadegan. « Spring Framework for rapid open source J2EE Web Application Development : A case study ». In : *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network*. IEEE. 2005, p. 90-95.

[8] Yudistira Asnar et al. « From trust to dependability through risk analysis ». In : *The Second International Conference on Availability, Reliability and Security (ARES'07)*. IEEE. 2007, p. 19-26.

[9] Yudistira Asnar et al. « Risk as dependability metrics for the evaluation of business solutions : a model-driven approach ». In : *2008 Third International Conference on Availability, Reliability and Security*. IEEE. 2008, p. 1240-1247.

[10] ENISA ad hoc working group on risk assessment et risk management. *Inventory of risk assessment and risk management methods*. ENISA, 2006.

[11] Colin Atkinson et Thomas Kuhne. « Model-driven development : a metamodeling foundation ». In : *IEEE software* 20.5 (2003), p. 36-41.

[12] Issa Atoum, Ahmed Otoom et Amer Abu Ali. « A holistic cyber security implementation framework ». In : *Information Management & Computer Security* (2014).

[13] Mohamed El-Attar et al. « Extending the UML statecharts notation to model security aspects ». In : *IEEE Transactions on Software Engineering* 41.7 (2015), p. 661-690.

[14] Iver Band et al. « Modeling enterprise risk management and security with the ArchiMate language ». In : (2015).

[15] Jose Barateiro, Goncalo Antunes et Jose Borbinha. « Manage risks through the enterprise architecture ». In : *2012 45th Hawaii International Conference on System Sciences*. IEEE. 2012, p. 3297-3306.

[16] David Basin, Jürgen Doser et Torsten Lodderstedt. « Model driven security : From UML models to access control infrastructures ». In : *ACM Transactions on Software Engineering and Methodology (TOSEM)* 15.1 (2006), p. 39-91.

[17] Kristian Beckers, Denis Hatebur et Maritta Heisel. « A problem-based threat analysis in compliance with common criteria ». In : *2013 International Conference on Availability, Reliability and Security*. IEEE. 2013, p. 111-120.

[18] Peter Bernus, Ovidiu Noran et Arturo Molina. « Enterprise architecture : Twenty years of the GERAM framework ». In : *Annual Reviews in Control* 39 (2015), p. 83-93.

[19] Jean Bézivin. « In search of a basic principle for model driven engineering ». In : *Novatica Journal, Special Issue* 5.2 (2004), p. 21-24.

[20] Jean Bézivin. « Model driven engineering : Principles, scope, deployment and applicability ». In : *Summer School on Generative and Transformational Techniquesin Software Engineering*. 2005, p. 1-33.

[21] Jean Bézivin et Jean-Pierre Briot. « Sur les principes de base de l'ingénierie des modèles. » In : *Obj. Logiciel Base données Réseaux* 10.4 (2004), p. 145-157.

[22] Jean Bézivin et Olivier Gerbé. « Towards a precise definition of the OMG/MDA framework ». In : *Proceedings 16th Annual International Conference on Automated Software Engineering (ASE 2001)*. IEEE. 2001, p. 273-280.

[23] Geoffrey Biggs, Takeshi Sakamoto et Tetsuo Kotoku. « A profile and tool for modelling safety information with design information in SysML ». In : *Software & Systems Modeling* 15 (2016), p. 147-178.

[24] Terence J Blevins, John Spencer et Fred Waskiewicz. « TOGAF ADM and MDA ». In : *The Open Group and OMG* (2004).

[25] Lossan Bondé. « Transformations de Modèles et Interopérabilité dans la Conception de Systèmes Hétérogènes sur Puce à Base d'IP ». Thèse de doct. Thèse de doctorat, Université des Sciences et Technologies de Lille, 2006.

[26] Paolo Bresciani et al. « Tropos : An agent-oriented software development methodology ». In : *Autonomous Agents and Multi-Agent Systems* 8 (2004), p. 203-236.

[27] Q Bui. « Evaluating enterprise architecture frameworks using essential elements ». In : *Communications of the association for information systems* 41.1 (2017), p. 6.

[28] Jason S Burkett. « Business Security Architecture : Weaving Information Security into Your Organization's Enterprise Architecture through SABSA® ». In : *Information Security Journal : A Global Perspective* 21.1 (2012), p. 47-54.

[29] Markus Buschle et al. « A Tool for automatic Enterprise Architecture modeling ». In : *International Conference on Advanced Information Systems Engineering*. Springer. 2012, p. 1-15.

[30] Brian H Cameron et Eric McMillan. « Analyzing the current trends in enterprise architecture frameworks ». In : *Journal of Enterprise Architecture* 9.1 (2013), p. 60-71.

[31] David Chen, Guy Doumeingts et François Vernadat. « Architectures for enterprise integration and interoperability : Past, present and future ». In : *Computers in industry* 59.7 (2008), p. 647-659.

[32] Vanea Chiprianov et al. « Extending enterprise architecture modeling languages for domain specificity and collaboration : application to telecommunication service design ». In : *Software & Systems Modeling* 13.3 (2014), p. 963-974.

[33] Benoît Combemale. « Ingénierie Dirigée par les Modèles (IDM)–État de l'art ». In : (2008).

[34] R Covington et Hamza Jahangir. « The oracle enterprise architecture framework ». In : *Oracle, Redwood Shores, CA* (2009).

[35] Krzysztof Czarnecki et Simon Helsen. « Classification of model transformation approaches ». In : *Proceedings of the 2nd OOPSLA Workshop on Generative Techniques in the Context of the Model Driven Architecture*. T. 45. 3. USA. 2003, p. 1-17.

[36] Jim Davies et al. « Compositionality and refinement in model-driven engineering ». In : *Brazilian Symposium on Formal Methods*. Springer. 2012, p. 99-114.

[37] Marcos Didonet Del Fabro, Jean Bézivin et Patrick Valduriez. « Weaving Models with the Eclipse AMW plugin ». In : *Eclipse Modeling Symposium, Eclipse Summit Europe*. T. 2006. Citeseer. 2006, p. 37-44.

[38] Folker Den Braber et al. « Model-based security analysis in seven steps–a guided tour to the CORAS method ». In : *BT Technology Journal* 25.1 (2007), p. 101.

[39] Munante Denisse. « Une approche basée sur l'Ingénierie Dirigée par les Modèles pour identifier, concevoir et évaluer des aspects de sécurité. » Thèse de doct. Pau, 2014.

[40] C Alberts A Dorofee. *Managing information security risks : the OCTAVE (SM) approach*. 2002.

[41] Éric Dubois et al. « A systematic approach to define the domain of information system security risk management ». In : *Intentional Perspectives on Information Systems Engineering* (2010), p. 289-306.

[42] Mathias Ekstedt et Teodor Sommestad. « Enterprise architecture models for cyber security analysis ». In : *2009 IEEE/PES Power Systems Conference and Exposition*. IEEE. 2009, p. 1-6.

[43] Golnaz Elahi et Eric Yu. « A goal oriented approach for modeling and analyzing security trade-offs ». In : *Conceptual Modeling-ER 2007 : 26th International Conference on Conceptual Modeling, Auckland, New Zealand, November 5-9, 2007. Proceedings 26*. Springer. 2007, p. 375-390.

[44] JHP Eloff et MM Eloff. « Information security architecture ». In : *Computer Fraud & Security* 2005.11 (2005), p. 10-16.

[45] Brian Elvesæter et al. « Aligning business and IT models in service-oriented architectures using BPMN and SoaML ». In : *Proceedings of the First International Workshop on Model-Driven Interoperability*. 2010, p. 61-68.

[46] ENISA. *CRAMM (CCTA Risk Analysis and Management Method)*. Sous la dir. de ENISA :European Union Agency for Cybersecurity. url : https://www.enisa.europa.eu/topics/risk-management/current-risk/risk-management-inventory/rm-ra-methods/m_cramm.html.

[47] Levent Ertaul et Raadika Sudarsanam. *Security Planning Using Zachman Framework for Enterprisesy*. 2005.

[48] Benjamin Fabian et al. « A comparison of security requirements engineering methods ». In : *Requirements engineering* 15 (2010), p. 7-40.

[49] Ali Fatolahi et Fereidoon Shams. « An investigation into applying UML to the Zachman framework ». In : *Information Systems Frontiers* 8.2 (2006), p. 133-143.

[50] David F Ferraiolo et al. « Proposed NIST standard for role-based access control ». In : *ACM Transactions on Information and System Security (TISSEC)* 4.3 (2001), p. 224-274.

[51] Joachim Fischer, Toby Neumann et Anders Olsen. « SDL code generation for open systems ». In : *International SDL Forum*. Springer. 2005, p. 313-322.

[52] IFIP-IFAC Task Force. « GERAM : Generalised enterprise reference architecture and methodology ». In : *IFIP-IFAC Task Force on Architectures for Enterprise Integration March Version* 1.3 (1999).

[53] Ulrik Franke, Waldo Flores Rocha et Pontus Johnson. « Enterprise architecture dependency analysis using fault trees and bayesian networks ». In : *IR-EE-ICS_2009 : 004*. 2009, p. 209-216.

[54] Susanne M Glissmann et Jorge Sanz. « An approach to building effective enterprise architectures ». In : *2011 44th Hawaii International Conference on System Sciences*. IEEE. 2011, p. 1-10.

[55] Anat Goldstein et Ulrich Frank. « A language for multi-perspective modelling of IT security : objectives and analysis of requirements ». In : *International Conference on Business Process Management*. Springer. 2012, p. 636-648.

[56] Anat Goldstein et Ulrich Frank. « Components of a multi-perspective modeling method for designing and managing IT security systems ». In : *Information Systems and e-Business Management* 14.1 (2016), p. 101-140.

[57] Rui Gomes. « Resilience and enterprise architecture in SMES ». In : *JISTEM-Journal of Information Systems and Technology Management* 12 (2015), p. 525-540.

[58] Eric Grandry, Christophe Feltus et Eric Dubois. « Conceptual integration of enterprise architecture management and security risk management ». In : *2013 17th IEEE International Enterprise Distributed Object Computing Conference Workshops*. IEEE. 2013, p. 114-123.

[59] Object Management Group. *Meta Object Facility (MOF) 2.0 Query/View/-Transformation Specification*. Sous la dir. de Object Management Group (2016). url : http://www.omg.org/spec/QVT/1.3.

[60] Object Management Group. *OMG-Standards Developmment Organization*. Sous la dir. de Object Management Group 2023. url : https://www.omg.org/.

[61] The Open Group. *Integrating Risk and Security within a TOGAF® Enterprise Architecture*. Sous la dir. de TOGAF® Series Guide. url : https://pubs.opengroup.org/togaf-standard/integrating-risk-and-security/integrating-risk-and-security_0.html.

[62] The Open Group. *The Open Group*. Sous la dir. de The TOGAF® Standard Version 9.2. url : http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html.

[63] The Open Group. *UML 2.2 Unified Modeling Language [Accessed*. Sous la dir. d'Object Management Group (2009). url : http://www.omg.org/spec/UML/2.2/.

[64] Jennifer Guild. *Scripting quality of security service (QoSS) safeguard measures for the suggested INFOCON system*. Rapp. tech. NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 2004.

[65] Manish Gupta. *Strategic and Practical Approaches for Information Security Governance : Technologies and Applied Solutions : Technologies and Applied Solutions*. IGI Global, 2012.

[66] Charles Haley et al. « Security requirements engineering : A framework for representation and analysis ». In : *IEEE Transactions on Software Engineering* 34.1 (2008), p. 133-153.

[67] Denis Hatebur, Maritta Heisel et Holger Schmidt. « A security engineering process based on patterns ». In : *18th International Workshop on Database and Expert Systems Applications (DEXA 2007)*. IEEE. 2007, p. 734-738.

[68] Veselina Hensel et Kerstin Lemke-Rust. « On an integration of an information security management system into an enterprise architecture ». In : *2010 Workshops on Database and Expert Systems Applications*. IEEE. 2010, p. 354-358.

[69] Christoph Hochreiner et al. « Using model driven security approaches in web application development ». In : *Information and Communication Technology : Second IFIP TC5/8 International Conference, ICT-EurAsia 2014, Bali, Indonesia, April 14-17, 2014. Proceedings 2*. Springer. 2014, p. 419-431.

[70] Michael Howard et Steve Lipner. *The security development lifecycle*. T. 8. Microsoft Press Redmond, 2006.

[71] Edward Humphreys. *Implementing the ISO/IEC 27001 information security management system standard*. Artech House, Inc., 2007.

[72] Maria Eugenia Iacob et al. « ArchiMate 2.0 Specification ». In : (2012).

[73] Frank Innerhofer-Oberperfler et Ruth Breu. « Using an Enterprise Architecture for IT Risk Management. » In : *ISSA*. Citeseer. 2006, p. 1-12.

[74] *ISO/IEC 27001 and related standards Information security management*. https://www.iso.org/isoiec-27001-information-security.html/. Accessed : 2023-01-06. 2022.

[75] Michael Jackson. *Problem frames : analysing and structuring software development problems*. Addison-Wesley, 2001.

[76] Jean-Marc Jézéquel, Benoit Combemale et Didier Vojtisek. *Ingénierie Dirigée par les Modèles : des concepts à la pratique...* Ellipses, 2012.

[77] Jean-Marc Jézéquel, Sébastien Gérard et Benoit Baudry. *Le génie logiciel et l'IDM : une approche unificatrice par les modèles*. 2006.

[78] Henk Jonkers et Dick AC Quartel. « Enterprise architecture-based risk and security modelling and analysis ». In : *International Workshop on Graphical Models for Security*. Springer. 2016, p. 94-101.

[79] Henk Jonkers et al. « Concepts for modeling enterprise architectures ». In : *International Journal of Cooperative Information Systems* 13.03 (2004), p. 257-287.

[80] Henk Jonkers et al. « Enterprise architecture : Management tool and blueprint for the organisation ». In : *Information systems frontiers* 8.2 (2006), p. 63.

[81] Andrew Josey. *TOGAF® Version 9.1-A Pocket Guide*. Van Haren, 2016.

[82] Frédéric Jouault et al. « ATL : a QVT-like transformation language ». In : *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*. 2006, p. 719-720.

[83] Jan Jürjens. « UMLsec : Extending UML for secure systems development ». In : *International Conference on The Unified Modeling Language*. Springer. 2002, p. 412-425.

[84] Olaf Kath. « Towards executable models : transforming EDOC behavior models to CORBA and BPEL ». In : *Proceedings. Eighth IEEE International Enterprise Distributed Object Computing Conference, 2004. EDOC 2004.* IEEE. 2004, p. 267-274.

[85] Rafiullah Khan et al. « STRIDE-based threat modeling for cyber-physical systems ». In : *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE. 2017, p. 1-6.

[86] Samir Kherraf, Éric Lefebvre et Witold Suryn. « Transformation from CIM to PIM Using Patterns and Archetypes ». In : *19th Australian Conference on Software Engineering (aswec 2008)*. 2008, p. 338-346. doi : 10.1109/ASWEC.2008.4483222.

[87] Jan Killmeyer. *Information security architecture : an integrated approach to security in the organization*. Auerbach Publications, 2006.

[88] Anneke Kleppe et al. *The model driven architecture : practice and promise*. 2003.

[89] Anneke G Kleppe et al. *MDA explained : the model driven architecture : practice and promise*. Addison-Wesley Professional, 2003.

[90] Svyatoslav Kotusev. « A comparison of the top four enterprise architecture frameworks ». In : *British Computer Society (BCS)* (2021).

[91] Deuk Kyu Kum et Soo Dong Kim. « A Method to Generate C# Code from MDA/PSM for Enterprise Architecture ». In : *5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSAR'06)*. IEEE. 2006, p. 238-243.

[92] *La méthode Ebios Risk Management-Le guide*. https://www.ssi.gouv.fr/guide/la-methode-ebios-risk-manager-le-guide//. Accessed : 2023-01-06. 2022.

[93] Robert Lagerström et al. « Enterprise architecture management's impact on information technology success ». In : *2011 44th Hawaii International Conference on System Sciences*. IEEE. 2011, p. 1-10.

[94] Ulrich Lang et Rudolf Schreiner. « Model Driven Security Management : Making Security Management Manageable in Complex Distributed Systems. » In : *MODSEC@ MoDELS*. 2008.

[95] Nabil Laoufi. « Processus guidé pour l'identification des exigences de sécurité à partir de l'analyse des risques ». Thèse de doct. Conservatoire national des arts et metiers-CNAM, 2017.

[96] Craig Larman. « Object-Oriented Analysis and Design ». In : *Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and the Unified Process* (2002), p. 10-11.

[97] Sara Larno, Ville Seppänen et Jarkko Nurmi. « Method framework for developing enterprise architecture security principles ». In : *Complex systems informatics and modeling quarterly* 117.20 (2019).

[98] Seok-Won Lee, Robin A Gandhi et Siddharth Wagle. « Towards a requirements-driven workbench for supporting software certification and accreditation ». In : *Third International Workshop on Software Engineering for Secure Systems (SESS'07 : ICSE Workshops 2007)*. IEEE. 2007, p. 8-8.

[99] Luncheng Lin et al. « Using abuse frames to bound the scope of security problems ». In : *Proceedings. 12th IEEE International Requirements Engineering Conference, 2004.* IEEE. 2004, p. 354-355.

[100] Torsten Lodderstedt, David Basin et Jürgen Doser. « SecureUML : A UML-based modeling language for model-driven security ». In : *International Conference on the Unified Modeling Language*. Springer. 2002, p. 426-441.

[101] Levi Lucio et al. « Advances in model-driven security ». In : *Advances in Computers*. T. 93. Elsevier, 2014, p. 103-152.

[102] Raimundas Matulevicius, Nicolas Mayer et Patrick Heymans. « Alignment of misuse cases with security risk management ». In : *2008 Third International Conference on Availability, Reliability and Security*. IEEE. 2008, p. 1397-1404.

[103] Raimundas Matulevičius et al. « Adapting secure tropos for security risk management in the early phases of information systems development ». In : *International Conference on Advanced Information Systems Engineering*. Springer. 2008, p. 541-555.

[104] Nicolas Mayer. « Model-based management of information system security risk ». Thèse de doct. University of Namur, 2009.

[105] Nicolas Mayer, André Rifaut, Eric Dubois et al. « Towards a risk-based security requirements engineering framework ». In : *Workshop on Requirements Engineering for Software Quality. In Proc. of REFSQ*. T. 5. 2005.

[106] Nicolas Mayer et al. « An integrated conceptual model for information system security risk management and enterprise architecture management based on togaf ». In : *Ifip working conference on the practice of enterprise modeling*. Springer. 2016, p. 353-361.

[107] Nicolas Mayer et al. « An integrated conceptual model for information system security risk management supported by enterprise architecture management ». In : *Software & Systems Modeling* 18.3 (2019), p. 2285-2312.

[108] Nicolas Mayer et al. « Towards the ENTRI framework : security risk management enhanced by the use of enterprise architectures ». In : *International Conference on Advanced Information Systems Engineering*. Springer. 2015, p. 459-469.

[109] Michelle McClintock et al. « Enterprise security architecture : Mythology or methodology ? » In : *ICEIS (2)*. 2020, p. 679-689.

[110] John McDermott et Chris Fox. « Using abuse case models for security requirements analysis ». In : *Proceedings 15th Annual Computer Security Applications Conference (ACSAC'99)*. IEEE. 1999, p. 55-64.

[111] Nancy R Mead et Ted Stehney. « Security quality requirements engineering (SQUARE) methodology ». In : *ACM SIGSOFT Software Engineering Notes* 30.4 (2005), p. 1-7.

[112] Daniel Mellado, Eduardo Fernández-Medina et Mario Piattini. « Applying a security requirements engineering process ». In : *Computer Security–ESORICS 2006 : 11th European Symposium on Research in Computer Security, Hamburg, Germany, September 18-20, 2006. Proceedings 11*. Springer. 2006, p. 192-206.

[113] Tom Mens et Pieter Van Gorp. « A taxonomy of model transformation ». In : *Electronic notes in theoretical computer science* 152 (2006), p. 125-142.

[114] *Méthode Harmonis ´ee d'Analyse de Risques (MEHARI)*. http://www.clusif.fr//. Accessed : 2023-01-06. 2022.

[115] Daniel Minoli. *Enterprise architecture A to Z : frameworks, business process modeling, SOA, and infrastructure technology*. Auerbach Publications, 2008.

[116] Nina Moebius et al. « SecureMDD : a model-driven development method for secure smart card applications ». In : *2009 International Conference on Availability, Reliability and Security*. IEEE. 2009, p. 841-846.

[117] Djedjiga Mouheb et al. *Aspect-oriented security hardening of UML design models*. T. 2105. Springer, 2015.

[118] Haralambos Mouratidis et Paolo Giorgini. « Secure tropos : a security-oriented extension of the tropos methodology ». In : *International Journal of Software Engineering and Knowledge Engineering* 17.02 (2007), p. 285-309.

[119] Denisse Muñante et al. « A review of security requirements engineering methods with respect to risk analysis and model-driven engineering ». In : *International Conference on Availability, Reliability, and Security*. Springer. 2014, p. 79-93.

[120] Suvda Myagmar, Adam J Lee et William Yurcik. « Threat modeling as a basis for security requirements ». In : (2005).

[121] Douraïd Naouar. « MoRiA Une méthode basée sur les modèles pour l'analyse des risques de cybersécurité : application à un système complexe de défense navale ». Thèse de doct. Ecole nationale supérieure Mines-Télécom Atlantique Bretagne Pays de la Loire, 2022.

[122] Eetu Niemi et Samuli Pekkola. « The benefits of enterprise architecture in organizational transformation ». In : *Business & information systems engineering* 62.6 (2020), p. 585-597.

[123] NIST. *Federal Enterprise Architecture Framework*. Sous la dir. de CSRC. url : https : / / csrc . nist . gov / glossary / term / federal _ enterprise _ architecture.

[124] MK Normalini, T Ramayah et Muhammad Salman Shabbir. « Investigating the impact of security factors in E-business and internet banking usage intention among Malaysians ». In : *Industrial Engineering & Management Systems* 18.3 (2019), p. 501-510.

[125] *Octave Critical Threat, Asset and Vulnerability Evaluation (OCTAVE)*. http://www.cert.org/octave/. Accessed : 2023-01-06. 2015.

[126] Jon Oldevik et al. « Toward standardised model to text transformations ». In : *European Conference on Model Driven Architecture-Foundations and Applications*. Springer. 2005, p. 239-253.

[127] OMG. « https ://www.omg.org/spec/UML/2.5.1 ». Version 2.4.1. In : (2017).

[128] OMG. *OMG Meta Object Facility (MOF) Core Specification*. T. 2.4.1. http ://www.omg.org/, 2013.

[129] OMG. *QVT : Meta object facility (mof) 2.0 query/view/transformation*. T. 1.1. http ://www.omg.org/spec/QVT/1.1/PDF/-Dec2022, Jan2011.

[130] 2016 Open Group Guide. *Integrating Risk and Security within a TOGAF R Enterprise Architecture*. 2016.

[131] S Shervin Ostadzadeh, Fereidoon Shams Aliee et S Arash Ostadzadeh. « An MDA-based generic framework to address various aspects of enterprise architecture ». In : *Advances in Computer and Information Sciences and Engineering*. Springer, 2008, p. 455-460.

[132] Roxane Pagnamenta. « Gouvernance de l'information ». In : (2014).

[133] Maragathavalli Palanivel et Kanmani Selvadurai. « Risk-driven security testing using risk analysis with threat modeling approach ». In : *SpringerPlus* 3.1 (2014), p. 1-14.

[134] Roman Pilipchuk, Stephan Seifermann et Robert Heinrich. « Aligning business process access control policies with enterprise architecture ». In : *Proceedings of the Central European Cybersecurity Conference 2018*. 2018, p. 1-4.

[135] Shaun Posthumus et Rossouw Von Solms. « A framework for the governance of information security ». In : *Computers & security* 23.8 (2004), p. 638-646.

[136] Vijaykumar Rachamadugu et John A Anderson. « Managing security and privacy integration across enterprise business process and infrastructure ». In : *2008 IEEE International Conference on Services Computing*. T. 2. IEEE. 2008, p. 351-358.

[137] Philippe Ramadour et Corine Cauvet. « Approach and model for business components specification ». In : *International Conference on Database and Expert Systems Applications*. Springer. 2002, p. 628-637.

[138] Yosef Rauchwerger, Finn Kristoffersen et Yair Lahav. « Cinderella SLIPPER : An SDL to C-code generator ». In : *International SDL Forum*. Springer. 2005, p. 210-223.

[139] *risk management - Glossary — CSRC*. https://csrc.nist.gov/glossary/term/riskmanagement/. Accessed : 2023-01-06. April 2022.

[140] *SABSA Enterprise Security Architecture*. https://sabsa.org/whitepapers/. Accessed : 2022-12-06. 2022.

[141] Pallab Saha. *A systemic perspective to managing complexity with enterprise architecture*. IGI Global, 2013.

[142] Oscar Sánchez et al. « ModelSec : a generative architecture for model-driven security ». In : *Journal of Universal Computer Science* 15.15 (2009), p. 2957-2980.

[143] SNV Schweizerische. « Information technology-Security techniques-Information security management systems-Requirements ». In : *ISO/IEC International Standards Organization* (2013).

[144] CLUSIF-Club de Sécurité Informatique Française. *MEHARI*. Sous la dir. de CLUSIF. url : https://clusif.fr/wp-content/uploads/2015/10/mehari-2010-risk-analysis-and-treatment-guide.pdf.

[145] Edwin Seidewitz. « What models mean ». In : *IEEE software* 20.5 (2003), p. 26-32.

[146] Bran Selic. « MDA manifestations ». In : *The European Journal for the Informatics Professional (UPGRADE)* 44.10 (2008), p. 29-32.

[147] Hanifa Shah et Mohamed El Kourdi. « Frameworks for enterprise architecture ». In : *It Professional* 9.5 (2007), p. 36-41.

[148] Marzieh Shariati, Faezeh Bahmani et Fereidoon Shams. « Enterprise information security, a review of architectures and frameworks from interoperability perspective ». In : *Procedia computer science* 3 (2011), p. 537-543.

[149] John Sherwood, Andrew Clark et David Lynas. *Enterprise security architecture*. CRC Press, 2005.

[150] John Sherwood, Andrew Clark et David Lynas. *Enterprise security architecture/Chap 11-LogicalSecurity Architecture-§289-330*. CRC Press, 2005.

[151] John Sherwood, Andrew Clark et David Lynas. *Enterprise security architecture/Chap 7-Physical Security Architecture-§111-135*. CRC Press, 2005.

[152] Nataliya Shevchenko et al. *Threat modeling : a summary of available methods*. Rapp. tech. Carnegie Mellon University Software Engineering Institute Pittsburgh United ..., 2018.

[153] Timothy R Shives et Laban M Pelz. *Analyzing the US Marine Corps enterprise information technology framework for IT acquisition and portfolio governance*. Rapp. tech. Acquisition Research Program, 2012.

[154] Adam SHOSTACK. *Threat modeling : Designing for security*. John Wiley Sons, 2014, 2014.

[155] Nuno Silva, Miguel Mira da Silva et Pedro Sousa. « A viewpoint for analyzing enterprise architecture evolution ». In : *2017 IEEE 21st International Enterprise Distributed Object Computing Conference (EDOC)*. IEEE. 2017, p. 20-29.

[156] Jacques Simonin et John Puentes. « Automatized integration of a contextual model into a process with data variability ». In : *Computer Languages, Systems & Structures* 54 (2018), p. 156-182.

[157] Jacques Simonin et al. « Information system services generation of business services specification and based on a system-of-services logical architecture pattern ». In : *International Journal of Cooperative Information Systems* 29.03 (2020), p. 2050002.

[158] Guttorm Sindre et Andreas L Opdahl. « Capturing security requirements through misuse cases ». In : *NIK 2001, Norsk Informatikkonferanse 2001, http ://www. nik. no/2001* 74 (2001).

[159] Guttorm Sindre et Andreas L Opdahl. « Eliciting security requirements with misuse cases ». In : *Requirements engineering* 10.1 (2005), p. 34-44.

[160] Nadera Slimani et al. « UACML : Unified access control modeling language ». In : *2011 4th IFIP International Conference on New Technologies, Mobility and Security*. IEEE. 2011, p. 1-8.

[161] Richard Soley et al. « Model driven architecture ». In : *OMG white paper* 308.308 (2000), p. 5.

[162] Teodor Sommestad, Mathias Ekstedt et Hannes Holm. « The cyber security modeling language : A tool for assessing the vulnerability of enterprise system architectures ». In : *IEEE Systems Journal* 7.3 (2012), p. 363-373.

[163] Teodor Sommestad, Mathias Ekstedt et Pontus Johnson. « Combining defense graphs and enterprise architecture models for security analysis ». In : *2008 12th International IEEE Enterprise Distributed Object Computing Conference*. IEEE. 2008, p. 349-355.

[164] National Institute of Standards et Technology. *Information Technology /Cybersecurity*. Sous la dir. de NIST. url : https://csrc.nist.gov/projects/risk-management/about-rmf.

[165] Ketil Stolen et al. « Model-based risk assessment–the CORAS approach ». In : *iTrust Workshop*. 2002.

[166] Halen Sun, Saen Xu et Paul Silverstein. « Oracle enterprise architecture framework : Information architecture domain ». In : *Oracle White Paper. URL : http ://www. oracle. com/technetwork/topics/entarch/oea-info-archframework-dev-process-513866. pdf, accessed on October* 17 (2012), p. 2014.

[167] Michael Szvetits et Uwe Zdun. « Systematic literature review of the objectives, techniques, kinds, and architectures of models at runtime ». In : *Software & Systems Modeling* 15.1 (2016), p. 31-69.

[168] *Threat -Glossary — CSRC*. https://csrc.nist.gov/glossary/term/threat. Accessed : 2023-01-06. 2022.

[169] Carlos Troche. « Documenting complex systems in the enterprise ». In : *Intl. conf. on Complex Systems*. T. 47. 2006.

[170] Carnegie Mellon University. *Operationally Critical Threat,Asset, and Vulnerability Evaluation (OCTAVE) Framework,Version 1.0*. Sous la dir. de Institute Of. url : https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=13473.

[171] Axel Van Lamsweerde. « Elaborating security requirements by construction of intentional anti-models ». In : *Proceedings. 26th International Conference on Software Engineering*. IEEE. 2004, p. 148-157.

[172] Axel Van Lamsweerde et al. « The KAOS project : Knowledge acquisition in automated specification of software ». In : *Proceedings of the AAAI Spring Symposium Series*. 1991.

[173] Iohan Gonçalves Vargas, Thiago Gottardi et Rosana Teresinha Vaccare Braga. « Approaches for integration in system of systems : a systematic review ». In : *2016 IEEE/ACM 4th International Workshop on Software Engineering for Systems-of-Systems (SESoS)*. IEEE. 2016, p. 32-38.

[174] François Vernadat. « Enterprise Modeling in the context of Enterprise Engineering : State of the art and outlook ». In : *International Journal of Production Management and Engineering* 2.2 (2014), p. 57-73.

[175] Jonathan Whelan et Graham Meaden. *Business architecture : A practical guide*. Routledge, 2016.

[176] Robert Winter et Ronny Fischer. « Essential layers, artifacts, and dependencies of enterprise architecture ». In : *2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06)*. IEEE. 2006, p. 30-30.

[177] A Wayne Wymore. *Model-based systems engineering*. CRC press, 2018.

[178] Quanqi Ye et al. « Formal analysis of a single sign-on protocol implementation for android ». In : *2015 20th International Conference on Engineering of Complex Computer Systems (ICECCS)*. IEEE. 2015, p. 90-99.

[179] John A Zachman. « A framework for information systems architecture ». In : *IBM systems journal* 26.3 (1987), p. 276-292.

[180] Emmanuele Zambon et al. « Model-based qualitative risk assessment for availability of IT infrastructures ». In : *Software & Systems Modeling* 10 (2011), p. 553-580.

[181] Novica Zarvić et Roel Wieringa. « An integrated enterprise architecture framework for business-IT alignment ». In : *Designing Enterprise Architecture Frameworks : Integrating Business Processes with IT Infrastructure* 63.9 (2014).

[182] Zhengshu Zhou et al. « A systematic literature review on enterprise architecture visualization methodologies ». In : *IEEE Access* 8 (2020), p. 96404-96427.

Table des matières

abstract ix

Remerciements **xi**

Acronymes **xiii**

Table of Contents **xv**

Liste des tableaux **xvii**

Table des figures **xix**

Introduction Générale **1**
Contexte 1
Dépendance de la société moderne aux SI 1
Complexité des architectures des systèmes /logiciels actuels . . . 1
SI sont la cible d'attaques 1
Utilisation inadéquate des mécanismes de sécurité 2
L'ingénierie des systèmes et la problématique de sécurité 2
Exigences fonctionnelles et de sécurité indissociées 2
Passage informel des exigences de sécurité en politique de sécurité 3
Motivation et challenge 3
Objectifs de recherche 5
Questions de recherche 6
Approche et contributions 6
Organisation de la thèse 7
Partie I : Etat de l'Art 8
Partie II : Contribution 8
Conclusion générale et perspectives 9

I Etat de l'Art **11**

1 Cadre Général **13**
1.1 Introduction 13
1.2 Architecture d'entreprise pour le développement de systèmes sécurisés 15
1.2.1 Définition et caractéristiques d'une architecture d'entreprise 16
1.2.2 Cadres (Framework) d'architecture d'entreprise 19
1.2.3 Le cadre TOGAF 22
1.2.4 Architecture de sécurité d'entreprise 24
1.2.5 Cadre d'architecture de sécurité d'entreprise : TOGAF ADM et SABSA 27
1.2.6 Langage de modélisation d'architecture d'entreprise 32
1.2.7 Synthèse et discussion 33
1.3 Ingénierie Dirigée par les Modèles (IDM) et Architecture d'Entreprise 34
1.3.1 IDM : définition, principes et caractéristiques 35
1.3.2 Approche MDA (Model Driven Architecture) 37
1.3.3 Architecture de Méta Modélisation 38
1.3.4 MOF : Meta Object Facility 40
1.3.5 Le langage UML 40
1.3.6 Transformation de modèles 42
1.3.7 Langages de transformations 44
1.3.8 Complémentarité de TOGAF et MDA 46
1.3.9 Discussion et conclusion 48
1.4 Exigences de sécurité dans le cycle développement de système logiciel 50
1.4.1 Approche d'exigence de sécurité basée sur la gestion des risques 51
1.4.2 Méthodes d'ingénierie de la sécurité orientées modèles . . 58
1.4.3 Discussion et conclusion 68

II Contributions **71**

2 Intégration contextuelle du risque dans l'architecture métier du système **73**
2.1 Introduction 73
2.2 Contexte et travaux connexes 74
2.3 Approche d'analyse des menaces et du risque 77
2.3.1 Modélisation et analyse des menaces : la méthode Stride . 79
2.3.2 Méthode d'analyse et d'évaluation des risques basée sur EBIOS 84
2.4 Processus de gestion des risques de sécurité 87

2.5 Approche de construction du modèle contextuel de risque métier 91
2.6 Méta modèle de sécurité du risque métier : TCM-BR 92
2.7 Processus d'intégration du métier du risque dans l'architecture du système : CIM2CICM-R 93
2.8 Illustration de l'intégration de la sécurité au niveau du modèle métier par transformation de modèles 97
2.9 Conclusion 101

3 Intégration contextuelle du risque dans l'architecture logique du système 105
3.1 Introduction 105
3.2 Contexte et travaux connexes 106
3.3 Processus de construction du méta modèle de risque logique . . 108
3.3.1 Méta modèle Logique de sécurité du risque (TCM-LR) . . . 110
3.3.2 Processus d'intégration du modèle de risque logique dans l'architecture du système : CICM-R2PIM 113
3.3.3 Illustration 121
3.4 Discussion 128

4 Intégration contextuelle du risque dans l'architecture physique du système 131
4.1 Introduction 131
4.1.1 Approche de construction du modèle contextuel de risque de sécurité physique 133
4.2 Meta modèle contextuel du risque de sécurité physique : TCM PR 134
4.3 Processus d'intégration du modèle physique de sécurité au niveau de l'architecture physique : phase d'implémentation PSM 136
4.3.1 Transformation du PIM vers PICM-P (PIM2PICM-P) 137
4.3.2 Transformation de PICM-P vers PICM-R (PICM-P2PICM-R) 138
4.3.3 Transformation PICM-R vers PSM (PICM-R2PSM) 139
4.3.4 Le niveau PSM : : Architecture de conception physique . . 141
4.4 Illustration : Transformation Du PIM au PSM avec un contexte lié aux risque physique 142
4.4.1 Alignement entre les opérations logiques implémentées et les services SI 142
4.4.2 Le PICM-R (Platform Independent Contextual Model-Risk) 144
4.5 Conclusion 147

5 Expérimentation et Vérification 149
5.1 Etude de cas 149
5.2 Vérification du cas d'étude 178

5.3 Conclusion et discussion du cas d'étude 182

Conclusion Générale **185**
Révue de recherche 185
Révue des objectifs de recherche 186
Questions de recherche 186
Révue des contributions 187
Perspectives 189

Bibliographie **191**

Table des matières **209**
Resumé 213

COLLEGES SCIENCES
BRETAGNE POUR L'INGENIEUR
LOIRE ET LE NUMERIQUE

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Titre : Intégration de la Sécurité Dans un Cadre d'Architecture d'Entreprise à Partir du Risque

Mot clés : Architecture Logicielle, Ingénierie Dirigée par les Modèles, Architecture d'Entreprise, Sécurité SI, Gestion de Risque

Résumé : Les sociétés actuelles et l'économie moderne, dépendent fortement des systèmes logiciels et leur interconnexion, qui permettent de gérer et de coordonner des actions complexes en vue de satisfaire entre autres, les besoins de performance, de productivité de l'entreprise.
Cependant, ces systèmes sont devenus des éléments critiques de ces organisations, et sont au cœur de préoccupations en lien avec la sécurité. En effet, ces systèmes sont le plus souvent la cible de nombreuses cyberattaques visant à en prendre le contrôle, à obtenir des données et informations sensibles ou à les détruire.
Adresser les problèmes de sécurité et de risque concernant le processus global du cycle de vie de développement des systèmes logiciels est une tâche difficile. Le plus souvent, il existe un écart entre les spécifications de sécurité définies lors de la phase des besoins du système et la mise en œuvre de la sécurité lors de la phase d'implémentation. Ce fait est dû aux méthodes de développement traditionnelles qui sont inefficaces dans un contexte de complexité des systèmes d'aujourd'hui avec des vulnérabilités résultant de leur développement et de la manière dont ces systèmes interagissent dans l'organisation.

Cette thèse se concentre sur une nouvelle approche de l'intégration de la sécurité en tant que composante clé de l'architecture du système logiciel, en se basant sur l'évaluation des risques et en utilisant le cadre d'Architecture d'Entreprise (EA) TOGAF comme fondement pour définir les modèles du système ainsi que le Model Driven Engineering (MDA) de l'approche d'Ingénierie Dirigée par les Modèles (IDM) comme outil de développement. Cette thèse propose une méthodologie, combinant les méthodes STRIDE et EBIOS, pour l'identification des risques spécifiques à l'entreprise, en tenant compte des facteurs contextuels et des menaces. En utilisant le MDA, elle modélise ces risques comme contexte lié aux architectures métier, logique et physique de l'EA et ce conformement aux niveaux abstraits CIM, PIM et PSM du MDA. Ce qui permet une meilleure compréhension des interdépendances entre les composants fonctionnels et de sécurité. L'intégration de la sécurité est envisagée dès la phase des exigences et de la conception de l'architecture du système en utilisant des modèles de sécurité et des patrons architecturaux fondés sur l'EA. Le MDA permet d'automatiser le processus d'intégration en améliorant les modèles respectifs et substituant des composants de sécurité, garantissant ainsi une cohérence entre les modèles et les implémentations. La proposition vise ainsi, à rapprocher les points de vue techniques et métier sur la sécurité de l'information.

Pour expérimenter le cadre proposé, une étude de cas basée sur le e-Commerce a été menée pour évaluer sa pertinence et vérifier son applicabilité en utilisant des outils MDA pour la génération de code.

En définitive, cette thèse contribue ainsi à la convergence de l'Architecture d'Entreprise, de la sécurité informatique et du Model Driven Engineering (MDA) en fournissant un cadre méthodologique novateur. L'intégration de la sécurité basée sur l'évaluation des risques devient un impératif pour les organisations cherchant à préserver la confidentialité, l'intégrité et la disponibilité de leurs systèmes informatiques tout en restant agiles face aux menaces émergentes, le tout soutenu par des modèles contextuels.

Title: Integrating Security Into A Risk-Based Enterprise Architecture Framework

Keywords: Software Architecture, Model Driven Engineering, Enterprise Architecture, IS Security, Risk Management

Abstract: Current societies and the modern economy depend heavily on software systems and their interconnection, which make it possible to manage and coordinate complex actions in order to satisfy, among other things, the performance and productivity needs of the company.
However, these systems have become critical elements of these organizations, and are at the heart of security concerns. Indeed, these systems are most often the target of many cyberattacks aimed at taking control of them, obtaining sensitive data and information or destroying them.
Addressing security and risk issues regarding the overall software systems development lifecycle process is a difficult task. Most often, there is a gap between the security specifications defined during the system requirements phase and the security implementation during the implementation phase. This fact is due to traditional development methods which are ineffective in the context of the complexity of today's systems with vulnerabilities resulting from their development and the way these systems interact in the organization.

This thesis focuses on a new approach to integrating security as a key component of software system architecture, based on risk assessment and using the Enterprise Architecture (EA) framework TOGAF as a basis for defining the system models as well as the Model Driven Architecture (MDA) of the Model-Driven Engineering (MDE) approach as a development tool.
This thesis proposes a methodology, combining the STRIDE and EBIOS methods, for the identification of company-specific risks, taking into account contextual factors and threats. Using the MDA allows modeling these risks as a context linked to the business, logical and physical architectures of the EA in accordance with the abstract CIM, PIM and PSM levels of the MDA. This allows a better understanding of the interdependencies between functional and security components.
Security integration is considered from the requirements and system architecture design phase using EA-based security models and architectural patterns. MDA helps automate the integration process by improving the respective models and substituting security components, thus ensuring consistency between models and implementations. The proposal thus aims to bring together technical and business points of view on information security.

To experiment with the proposed framework, an e-Commerce based case study was conducted to evaluate its relevance and verify its applicability using MDA tools for code generation.

Ultimately, this thesis contributes to the convergence of Enterprise Architecture, IT security and Model Driven Engineering (MDA) by providing an innovative methodological framework. Integrating security based on risk assessment is becoming an imperative for organizations seeking to preserve the confidentiality, integrity and availability of their IT systems while remaining agile by facing emerging threats, all supported by contextual models.