Confidentialité, bases de données et réseaux d'ordinateurs

Hélène Richy

THESE

présentée à

Institut National Polytechnique de Grenoble

pour obtenir le grade de
DOCTEUR INGENIEUR

par

Hélène RICHY

CONFIDENTIALITE, BASES DE DONNEES ET RESEAUX D'ORDINATEURS.

Thèse soutenue le 6 février 1978 devant la Commission d'Examen :

Président : L. BOLLIET

Examinateurs : J.C. CHUPIN
M. DESVERGNES
J. LE BIHAN
G. SAUCIER

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Monsieur Philippe TRAYNARD : Président

Monsieur Pierre-Jean LAURENT : Vice Président

PROFESSEURS TITULAIRES

MM.	BENOIT Jean	Radioélectricité
	BESSON Jean	Electrochimie
	BLOCH Daniel	Physique du solide
	BONNETAIN Lucien	Chimie minérale
	BONNIER Etienne	Electrochimie et électrométallurgie
	BOUDOURIS Georges	Radioélectricité
	BRISSONNEAU Pierre	Physique du solide
	BUYLE-BODIN Maurice	Electronique
	COUMES André	Radioélectricité
	DURAND Francis	Métallurgie
	FELICI Noël	Electrostatique
	FOULARD Claude	Automatique
	LESPINARD Georges	Mécanique
	MOREAU René	Mécanique
	PARIAUD Jean-Charles	Chimie-Physique
	PAUTHENET René	Physique du solide
	PERRET René	Servomécanismes
	POLOUJADOFF Michel	Electrotechnique
	SILBER Robert	Mécanique des fluides

PROFESSEUR ASSOCIE

M.	ROUXEL Roland	Automatique

PROFESSEURS SANS CHAIRE

MM.	BLIMAN Samuel	Electronique
	BOUVARD Maurice	Génie mécanique
	COHEN Joseph	Electrotechnique
	LACOUME Jean-Louis	Géophysique
	LANCIA Roland	Electronique
	ROBERT François	Analyse Numérique
	VEILLON Gérard	Informatique fondamentale et appliquée
	ZADWORNY François	Electronique

MAITRES DE CONFERENCES

MM.	ANCEAU François	Mathématiques appliquées
	CHARTIER Germain	Electronique
	GUYOT Pierre	Chimie minérale
	IVANES Marcel	Electrotechnique
	JOUBERT Jean-Claude	Physique du solide
	MORET Roger	Electrotechnique nucléaire
	PIERRARD Jean-Marie	Mécanique
	SABONNADIERE Jean-Claude	Informatique fondamentale et appliquée
Mme.	SAUCIER Gabrièle	Informatique fondamentale et appliquée

MAITRE DE CONFERENCES ASSOCIE

M.	LANDAU Ioan	Automatique

CHERCHEURS DU C.N.R.S. (Directeur et Maîtres de Recherche)

MM.	FRUCHART Robert	Directeur de Recherche
	ANSARA Ibrahim	Maître de Recherche
	CARRE René	Maître de Recherche
	DRIOLE Jean	Maître de Recherche
	MATHIEU Jean-Claude	Maître de Recherche
	MUNIER Jacques	Maître de Recherche

Je remercie les membres de mon jury, Monsieur L. BOLLIET qui a dirigé cette thèse et a bien voulu présider le jury, ainsi que Madame SAUCIER et Monsieur DESVERGNES qui ont accepté d'y participer ;

je suis particulièrement reconnaissante à J.C. CHUPIN qui a accepté le travail long et fastidieux de lecture et critique détaillée du manuscrit ;

je remercie J. LE BIHAN, Directeur du Projet Pilote SIRIUS, d'avoir bien voulu accepter le rôle ingrat de rapporteur extérieur ;

je tiens également à remercier tous les membres des équipes réseaux de l'ENSIMAG et du Centre Scientifique CII-HB, qui m'ont toujours apporté aide et collaboration ;

j'exprime enfin ma gratitude à Madame C. CHALAND à qui revient tout le mérite de la préparation matérielle de ce document.

TABLE DES MATIÈRES

INTRODUCTION

0. INTRODUCTION

Le sujet

Comment est-il et doit-il être tenu compte du caractère "confidentiel" de certaines informations ou de certains ensembles d'informations dans des applications impliquant à la fois des bases de données et un réseau d'ordinateurs ?

Telle est la question autour de laquelle est centrée notre étude.

Nous nous limitons ici à un environnement supposé fiable et sans erreurs et supposons que les systèmes (exploitation, transport, bases de données) savent protéger efficacement une donnée élémentaire de tout accès non autorisé et maintenir son intégrité.

Motivations

Devant l'impossibilité d'arrêter totalement une concentration d'informations déjà bien avancée, il paraît urgent :

. d'informer les individus sur leurs responsabilités lorsqu'ils communiquent une quantité d'informations (éventuellement superflues) qu'ils estiment confidentielles à un organisme/système auquel ils ne peuvent accorder de confiance ;

. de développer au plan législatif, des moyens de contrôle des individus sur l'utilisation des informations qu'ils confient ;

. de fournir des moyens informatiques minima qui permettront de réaliser des mécanismes de protection efficaces, prenant en compte des contraintes "personnelles" (à définir).

Les objectifs de confidentialité sont laissés, jusqu'à présent, au second plan ; leur intégration vient généralement après la conception du reste. Il nous a donc paru intéressant d'étudier leur intégration dès la conception d'applications.

Plan de l'étude

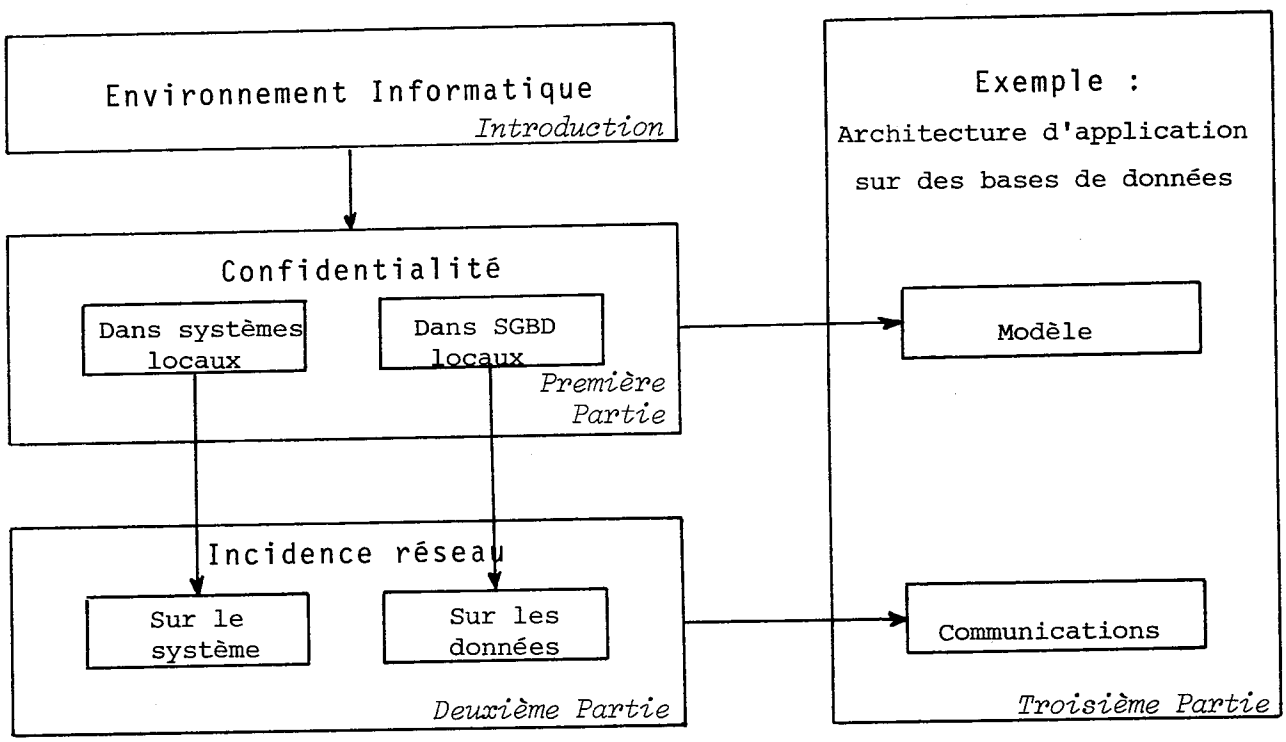
La confidentialité est un concept encore peu connu dans l'environnement informatique. Nous allons donc présenter cet environnement et isoler les problèmes de confidentialité des problèmes d'intégrité et de sécurité, entre autres, avec lesquels ils sont souvent confondus. C'est l'objet de la première partie.

Après avoir limité ce problème à une vérification de concordance entre le statut des données et leurs conditions d'utilisation, nous présenterons les principales solutions qui ont été développées ou proposées, c'est-à-dire sous les deux aspects système et base de données.

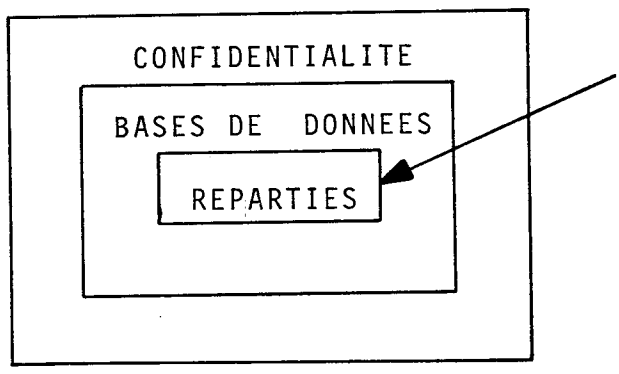
La deuxième partie est consacrée à l'étude de l'incidence de l'introduction des réseaux sur la confidentialité. Les solutions proposées permettent d'aborder la conception d'applications réparties sur un support mieux protégé.

La troisième partie présente deux exemples d'architecture d'applications réparties : l'un développé à l'Université de Rennes (depuis 1974-1975), le projet d'interrogation de fichiers répartis ; l'autre, POLYPHEME, dont les études sont entreprises depuis 1976 à l'Université de Grenoble.

Articulation des chapitres



En coupe :



0.1. Présentation

Le souci de protection des informations confidentielles s'est, depuis longtemps déjà, concrétisé par la mise en oeuvre de *codes* plus ou moins élaborés : du signal de fumée au codage multi-alphabet.

Des algorithmes complexes, envisagés en l'absence d'outils informatiques, peuvent maintenant être réalisés pour coder la masse d'informations concentrées sur les supports informatiques.

Les supports de transmission sont eux-mêmes l'objet de protection, souvent systématique, par l'emploi *d'encrypteurs* spécialisés (pour lignes téléphoniques).

En fait, quel que soit le support retenu pour stockage ou transmission, différentes méthodes ont été mises au point sans qu'aucune cohérence entre ces méthodes n'ait été recherchée.

Dès lors que l'on s'est préoccupé de contrôler la manipulation de ces informations, s'est développée une nécessaire synthèse de ces diverses méthodes ; les systèmes d'exploitation et de gestion de bases de données se sont heurtées à différents types de contrôle :

. identification et authentification,

. contrôle d'accès aux ressources avec prévention ou détection de l'interblocage,

. mesures de reprise (sur panne ou erreur détectée),

. certification de portions de codes, ...

Il nous a donc paru intéressant d'essayer de relier les méthodes existantes entre elles et de proposer leur intégration dans un modèle global de protection.

Les solutions qui sont et seront apportées à ces différentes questions en particulier, participeront à la mise en place de mécanismes dont les besoins sont exprimés maintenant par les législations nouvelles, tant nationales qu'internationales, concernant la protection de la confidentialité de données en environnement informatique. [Rapport de la Commission Informatique et Libertés].

La nécessité de législations nouvelles est d'autant plus justifiée, que de nombreuses administrations, de nombreuses entreprises, ont actuellement recours aux services de l'informatique pour la gestion et le stockage des informations. De plus, *l'accès partagé à des bases de données*, par exemple, n'est plus réservé à des spécialistes. Mais les informaticiens ont-ils les moyens d'assurer une sécurité que les individus, les législateurs, sont en droit de réclamer ?

La carence législative actuelle peut être considérée comme un frein au développement de l'informatique pour certains secteurs.

0.2. Exemples de fraudes

Classification

Les mises en échec de la confidentialité peuvent être classées en plusieurs catégories [SALTZER-SCHROEDER 75] :

. *la divulgation* des objets confidentiels : hors du champ des usages/usagers autorisés,

. *la détérioration* des objets confidentiels : préliminaire à une divulgation ou sa conséquence,

. *l'empêchement total* d'utilisation, intentionnel ou accidentel, justifie la réalisation de nombreuses mesures de reprise (la non fiabilité).

Les attaques

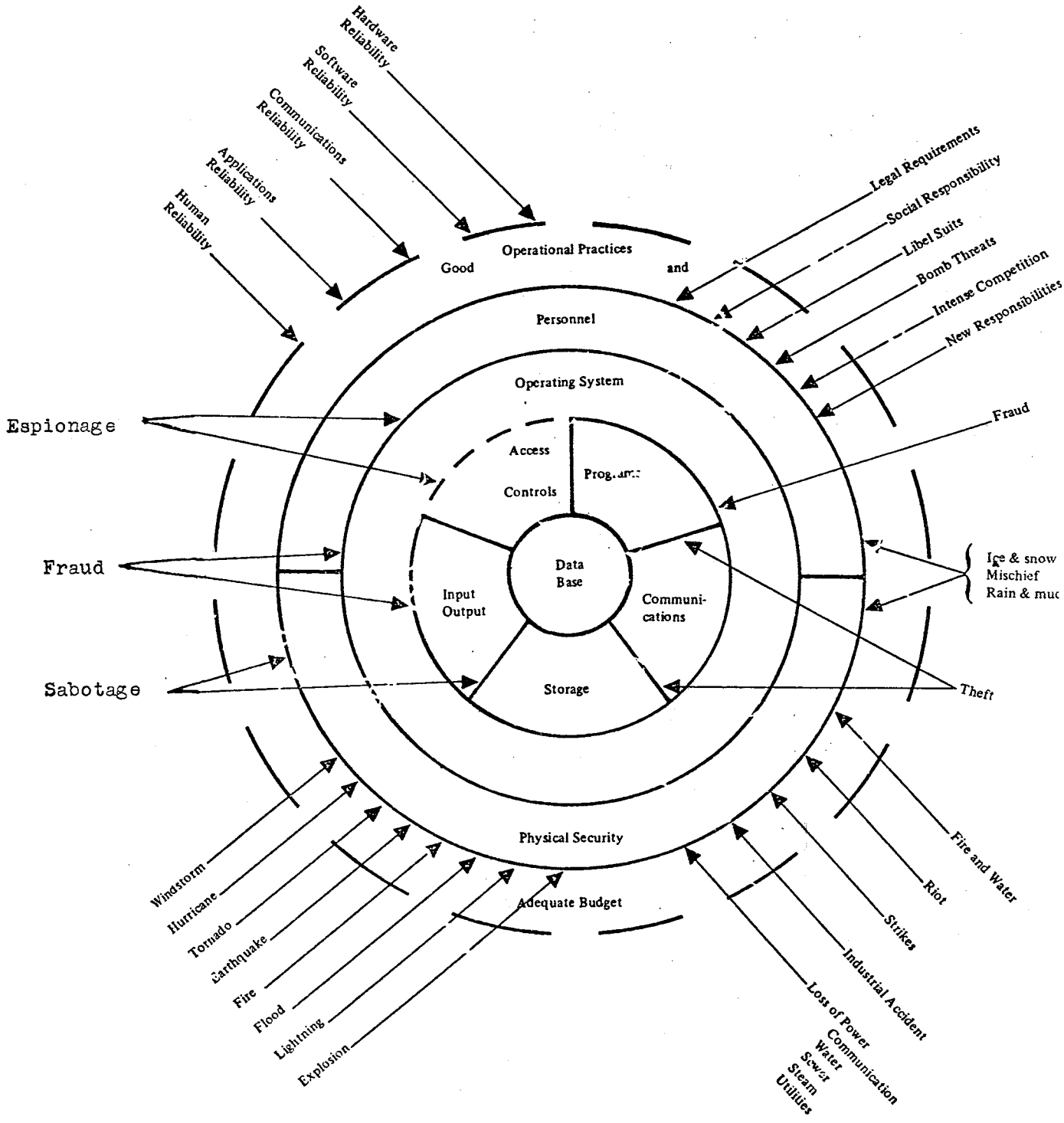
C'est moins une insuffisance des contrôles tant logiciels que matériels, qui est à l'origine des fraudes, que la non vérification des programmes mis en oeuvre. Des méthodes de certification élaborées devraient permettre de venir à bout de ces problèmes que nous n'aborderons pas ici.

Les attaques du logiciel sont actuellement peu répandues, ou plutôt peu connues. Ce qui peut être expliqué par les faits suivants :

. relative liberté d'accès aux informations : copie aisée de fichiers sans contrôle informatique,

. insuffisance des systèmes d'exploitation usuels : leur contournement n'est pas détecté,

. détection a posteriori.



THREATS AND THEIR USUAL DEFENSE (AFIPS 1974)

Figure 0.1.

Le "tout ou rien"

Un système de protection ne peut être considéré comme efficace que s'il est TOTALEMENT INATTAQUABLE. Les systèmes de protection tendent donc vers cette perfection, assez utopique il est vrai.

En général, on sera satisfait d'une protection partielle dont la probabilité de défaillance est assez faible. Peu de modèles d'étude probabiliste ont encore été développés, mais ils permettraient de juger le niveau actuel des protections.

0.3. Vers un modèle global de protection

Le maintien de la confidentialité ne peut être correctement assuré que si l'on a su répondre à un certain nombre de questions préalables concernant *les objets* à protéger et *les sujets* autorisés. Ce qui peut se résumer ainsi :

QUI doit avoir accès ?
A QUELLES informations ?
POUR QUEL usage ?

Il faut donc se donner les moyens de prévenir :
. les malveillances des usagers (*),
. le monopole des ressources (disponibilité),
. la communication à des usagers non autorisés (confinement),
. les accès ou modifications non autorisées (intégrité),
et de valider les mécanismes de protection envisagés.

(*) employés dans un sens large, représentent aussi bien un utilisateur de terminal, le terminal lui-même, un programme ou une procédure quelconque.

Modèle multi-niveaux

Quelques solutions à ces différents problèmes sont effectivement réalisées par des systèmes ; d'autres n'en sont qu'à l'état de proposition. Leur étude est l'objet de la 1ère partie. Elles se situent à différents niveaux :
- . langage évolué,
- . système de gestion de bases de données,
- . système d'exploitation,
- . codage, encryptage des transmissions,
- . logiciel réseau.

Modèle global

Une approche plus intégrée de ces problèmes permet de définir un système de contrôle cohérent, qui prend en compte l'acquis des différentes méthodes existantes. Telle est notre approche dans cette étude.

Il reste néanmoins vrai qu'un *modèle global de protection* peut être conçu comme un système entièrement nouveau. Mais, il nous paraît cependant plus réaliste de rechercher ces spécifications à partir de cette intégration du réel, considérée comme une ébauche pour un nouveau système.

Les principales fonctions de ce modèle sont donc décrites dans la première partie, hormis la fonction de communication qui sera détaillée dans la deuxième partie qui traite plus particulièrement de l'incidence de l'introduction des réseaux sur la confidentialité.

PREMIÈRE PARTIE

DÉFINITION ET PRINCIPES DE LA CONFIDENTIALITÉ

1.1. Définition des concepts

Dans le présent chapitre sont exposés les principaux concepts utilisés dans les études de la confidentialité.

Les définitions, ou plutôt les explications, présentées ici, peuvent être trouvées dans les ouvrages mentionnés en référence et principalement dans ces deux articles : [HOFFMAN 77], [ACM 76].

Sujets abordés :

Accès, contrôle d'accès,
authentification, mot de passe,
autorisation, matrice d'autorisation
certification,
confidentialité, secret,
confinement,
cryptographie, codage, chiffrement,
fraude, attaque, pénétration,
identification,
intégrité, cohérence
intimité
protection
sécurité

ACCES,
CONTROLE D'ACCES

L'accès à une ressource, gérée par un système, est la possibilité qui est donnée . d'approcher cette ressource,

. de communiquer avec elle,

. et en général de l'utiliser et de la modifier éventuellement.

Le contrôle d'accès est réalisé par une procédure de limitation des accès (aux ressoures gérées par le système) aux seules personnes autorisées, ou à d'autres systèmes (sur le réseau). Une telle procédure peut être effectuée :

. par hardware, ou firmware,

. par logiciel,

. par d'autres procédures

. par des personnes.

AUTHENTIFICATION, MOT DE PASSE

Il s'agit de vérifier l'identité ou l'autorisation d'un usager (ou processus). La phase d'identification est ainsi complétée afin d'améliorer la sécurité.

Différentes procédures sont utilisées :

. mot de passe : chaîne de caractères secrète dont l'ordinateur est chargé de vérifier la validité ;

. procédure de la poignée de main (anglais : hand shaking procedure).

Un dialogue est établi entre l'usager et le système de contrôle pour authentifier l'identité de cet usager par une série de questions-réponses basées sur des informations connues de ce seul usager.

AUTORISATION

(anglais "authorization")

C'est la permission donnée par le système de sécurité d'accéder à une donnée, à un fichier, à un enregistrement, .. Cette permission dépend :

. des privilèges de l'utilisateur,

. des privilèges du terminal,

. de l'action demandée,

. de la donnée elle-même,

. de la valeur de la donnée,

. d'autres éléments.

Les privilèges peuvent être représentés par une *matrice d'autorisation* : chaque élément a_{ij} détermine les droits d'accès de la $i^{ème}$ ressource à la $j^{ème}$ ressource.

Il est possible également d'associer aux ressources des *niveaux d'autorisation* sur lesquels sera basée l'autorisation.

L'autorisation consiste donc à accorder des droits d'accès à un usager (un programme ou une procédure ...).

CERTIFICATION

Elle consiste en une validation des programmes, de leur justesse, leur exactitude et de celle des mécanismes de sécurité ou de protection.

Elle est donc indispensable à la sécurité : pour vérifier que la totalité des cas est envisagée, afin de réaliser une protection intégrale.

CONFIDENTIALITE, SECRET

(anglais : "confidentiality")

Elle décrit *le statut accordé aux données* (tel que : secret, ultra-confidentiel, ...) et le degré de protection qui peut être assuré. Le statut est établi après accord entre les fournisseurs et les utilisateurs de ces données.

Maintenir la confidentialité des données consiste donc à les protéger contre toute manipulation non autorisée.

"Le secret consiste à réserver la connaissance de certaines données à certaines personnes, celles-ci n'étant pas autorisées à les révéler en dehors du cercle et des buts prévus" (*)

(*) Extrait de "Le Secret des Fichiers", Editions Cujas, Cahier n° 13, 1976, Institut Français des Sciences Administratives, F. GALLOUADEC-GENUYS, H. MAISL.

CONFINEMENT

Assurer le confinement c'est pouvoir contrôler qu'un processus qui a accès à une information n'est pas capable de communiquer cette information à des processus non autorisés : limiter la transmission des informations. Par exemple, l'accès aux données peut être autorisé à un programme, mais pas la communication de ces données par ce programme.

CODER, DECODER (anglais : to encode
CRYPTER, DECRYPTER to encrypt
CHIFFRER, DECHIFFRER to encipher)

Modification réversible d'une information, de telle sorte que la sécurité soit assurée durant la transmission ou le stockage dans un environnement non protégé.

Deux méthodes :

1) coder : le texte d'origine (plain text) est remplacé par un texte codé (code text) par substitution effectuée à partir d'une *table de codage* (code book) ;

2) chiffrer : le texte d'origine est transformé en un texte chiffré en fonction d'une certaine *clé de transformation*, utilisée aussi lors de la transformation inverse : texte chiffré → texte d'origine.

Le terme "cryptage" qui devrait être utilisé pour représenter indifféremment une de ces deux méthodes, est souvent négligé au profit du terme codage.

Mise en oeuvre :

1) point à point (link encryption) : à chaque étape de la communication dans le réseau, l'information est codée puis recodée, en utilisant généralement des codages différents :

2) de bout en bout (end to end encryption) : le codage est réalisé à l'origine et ne subit aucune modification sur le réseau. Le décodage sera effectué seulement lorsque l'information sera parvenue à sa destination finale.

FRAUDE
ATTAQUE
PENETRATION

L'attaque est la formulation et l'exécution d'un plan de fraude. La pénétration est un accès non autorisé, réussi dans un système.

Les différents types de fraudes sont représentés dans le tableau de Winkler & Danner (*) reproduit ici :

Catégorie		Définition
Divulgation non autorisée	Exposition	Divulgation accidentelle
	Interruption	Saisie intentionnelle
Modification non autorisée	Altération	Modification accidentelle
	Fraude	Modification intentionnelle
Restriction non autorisée	Interruption	Refus accidentel d'un accès correct
	Rupture	Refus intentionnel
Destruction non autorisée	Suppression	Destruction accidentelle
	Elimination	Destruction intentionnelle

(*) S. WINKLER & L. DANNER, "Data security in the computer communication environment", Computer, February 1974, pp. 23-31.

IDENTIFICATION

C'est la reconnaissance d'un individu (programme, terminal ou ressource en général) comme étant le même que celui dont les caractéristiques ont été décrites antérieurement.

Cette reconnaissance s'effectue généralement à l'aide d'un nom unique, fixé à l'avance, ou d'un code d'identification. Il ne s'agit donc pas de l'identité exacte (éventuellement partielle ou usurpée), mais d'une prétention d'identité.

Seules des mesures d'authentification permettront de vérifier cette identité.

INTEGRITE, COHERENCE

Le concept d'intégrité a été ainsi analysé par G.C. EVEREST (*)

. maintenir l'existence des données : sécurité physique, mesures de reprise,

. maintenir la qualité des données : conformément aux déclarations de compatibilité,

. protéger l'information : pas d'altération parasite, confidentialité,

. partager l'information,

et englobe ainsi la notion de confidentialité.

Le plus souvent, on ne considère que *deux types d'intégrité des données* :

1) cohérence des données, en tant que copies d'un original, l'intégrité existe alors lorsque les données ne diffèrent pas de leur origine et qu'elles n'ont pas été modifiées, divulguées ou détruites accidentellement ou intentionnellement. Dans le cas de copies multiples, la cohérence peut être alors assurée de différentes manières :

. soit spatiale : les copies distantes sont identiques,

. soit temporelle : après répercution différée des mises à jour sur les copies.

2) Cohérence des données au sein d'une base ou d'un fichier :

. cohérence interne : entre éléments de cette base (donc testable sur la base elle-même),

. cohérence externe : entre des éléments de la base et des éléments du monde extérieur, tels que des valeurs statistiques ou temporelles.

(*) G.C. EVEREST, "Concurrent update control and data base integrity", Proc. IFIP 1974, pp. 241-270.

Le terme d'intégrité est aussi employé pour qualifier *un système* dont les caractéristiques suivantes sont vérifiées quelles que soient les conditions de fonctionnement :

a) régularité de fonctionnement : tolérance aux pannes,

b) mécanisme de protection totale (pas de cas oublié),

c) exactitude des structures et des données stockées.

INTIMITE

(anglais : "privacy")

C'est un concept qui s'applique à un individu (ou organisation) : c'est son droit de décider quelle information le concernant il souhaite partager avec les autres et aussi quelle information il est prêt à accepter des autres. Il peut alors désirer contrôler le regroupement, l'usage et la dispersion de telles informations et à quelles personnes ou organisations elles sont communiquées. Cela consiste donc à savoir :

qui doit avoir accès ?

à *quelles* informations ?

et pour *quel* usage ?

PROTECTION

Elle recouvre à la fois la sécurité et les mécanismes de contrôle d'accès aux ressources gérées par le système.

SECURITE

C'est la réalisation de la protection

. des données,

. des mécanismes et ressources utilisés pour les manipuler,

. des mécanismes de sécurité eux-mêmes.

On peut distinguer les problèmes de sécurité selon l'objet à protéger. Les mesures prises sont alors fondamentalement différentes selon les cas.

Sécurité des données :

C'est la protection des données contre la destruction, la modification, la divulgation non autorisées, accidentelles ou intentionnelles, par des moyens techniques (ou physiques !).

Sécurité des communications :

C'est protéger les informations qui sont transmises par les lignes de communication et autres équipements de telle sorte que les informations sensibles ne soient pas communiquées à des personnes non autorisées : le codage fait partie de ces mesures.

Sécurité des installations (machines ..) :

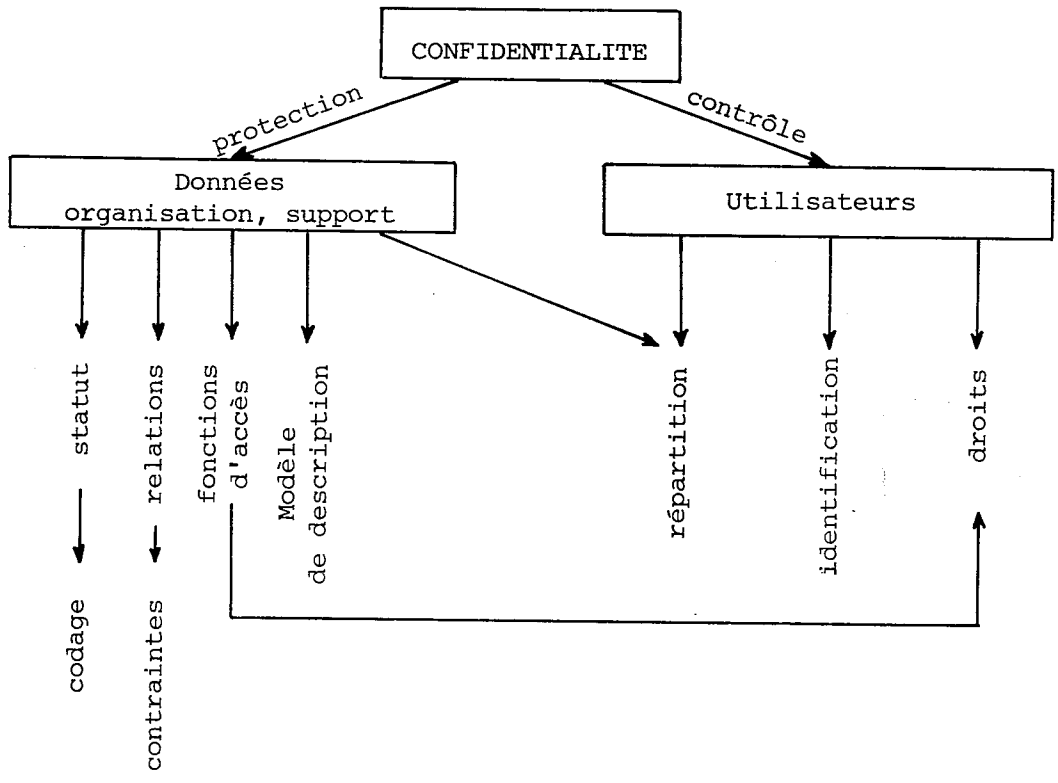
Les fonctions matérielles et logicielles réalisées sur les machines doivent présenter les caractéristiques permettant de fournir un niveau de protection suffisant.

Parmi les définitions citées ci-dessus, nous nous intéressons plus particulièrement ici à celles qui concernent :

. les systèmes de gestion de bases de données (confidentialité, statut des données, contrôle d'accès) ;

. les communications sur un réseau (codage, sécurité du réseau) et liées à la répartition des données ;

comme cela est représenté sur le schéma suivant :



La suite de cette partie est consacrée à l'étude de solutions adoptées d'une part au niveau des systèmes locaux, d'autre part au niveau des SGBD locaux.

1.2. Les solutions adoptées au niveau des systèmes locaux

1.2.1. Le cadre de l'étude

Trois types de méthodes

Les moyens utilisés pour assurer les différentes fonctions de protection visent à rendre les fraudes plus difficiles et plus coûteuses. Par ailleurs, le coût d'exploitation et de mise en oeuvre de ces méthodes est un facteur de choix aussi important que l'efficacité du mécanisme. Un classement en trois catégories peut donc être proposé :

. *le matériel :*

- disponibilité accrue par redondance des installations,
- contrôle d'accès aux lignes, terminaux, machines,
- encrypteurs de lignes ..

. *le logiciel :*

- méthode d'aide à la mise au point de programmes,
- certification de programmes,
- dispersion des informations sensibles,
- mesures d'identification, authentification.

. *Autres méthodes :*

peu formalisées actuellement, elles sont encore difficilement contrôlables (mesures administratives, juridiques) : comment éviter, par exemple, la collusion entre usagers de droits différents ou la communication d'informations hors du contexte informatique.

L'objet de cette étude ne concerne que les moyens logiciels de maintien de la confidentialité des données, ou plus généralement des objets dans les systèmes locaux.

Les préoccupations principales

Les fonctions de contrôle correspondent à deux types de préoccupations :

. la protection des informations : maintien de la confidentialité, de la sécurité,

. le partage des informations : intégrité, concurrence, interblocage, cloisonnement entre usagers.

C'est essentiellement la première qui nous intéresse ici. Mais le second point doit être évoqué dès que l'on veut s'intéresser à la répartition ou à l'implantation de ces fonctions.

Le partage

Nous nous intéressons ici à des mécanismes de *protection partagée* des informations : ces mécanismes ne sont le monopole d'aucun système central, mais de sous-systèmes ou même d'applications spécifiques. Le choix d'un tel niveau fonctionnel est seulement guidé par une constatation d'efficacité des mécanismes existants.

Une solution exigeante

Une solution radicale consiste à réserver toutes les ressources à un usage confidentiel. Les inconvénients d'exploitation qui en résultent sont cependant difficilement supportables (rapport efficacité/coût). Mais sa rapidité de mise en oeuvre en fait une solution de premier choix lorsqu'un souci de secret surgit pour un ensemble d'informations. Une telle solution ne permet pas de prendre en compte plusieurs niveaux de sécurité sur des données. En effet, toutes les informations à protéger ont ainsi le même statut. Par niveau de sécurité on entend donc ici uniquement les deux niveaux de traitement.

Classification

Différents critères peuvent être choisis pour classer les mécanismes de protection [HOFFMAN 71] :

. l'évolutivité :

il existe des mécanismes *statiques*, tels que l'identification, l'authentification, les systèmes à capacités ... et des mécanismes plus *dynamiques*, tels que certaines méthodes nouvelles d'authentification ..

. la granularité :

les objets protégés peuvent être :

- soit des données au niveau le plus fin (bit),
- soit des données structurées (articles, fichiers),
- soit des relations entre données,
- soit des catalogues, fonctions d'accès, procédures ..

La présentation qui suit correspond à ce second type de classement. Elle repose sur une étude essentiellement bibliographique dont les principales références sont regroupées en annexe. D'autres peuvent être trouvées dans le rapport (*) et dans [HOFFMAN 69].

Le maintien de la confidentialité passe par le *contrôle d'accès* : il ne peut éviter ces phases obligatoires que sont l'identification, l'authentification et le codage/décodage. Des études générales du problème ont en particulier été développées dans les articles : [WINKLER-DANNER 74], [TURN-WARE 75], [SALTZER-SCHROEDER 75].

(*) Etude de la confidentialité pour les systèmes de bases de données fonctionnant dans le contexte d'un réseau général d'ordinateurs. Rapport bibliographique. Contrat DRME 76/200. H. RICHY. Janvier 1977.

1.2.2. Cryptage et codage

Principe

Des circuits de codage peuvent être ajoutés aux extrêmités de lignes de transmission ou même directement sur un terminal, afin de transformer tous les signaux qui circulent sur cette ligne ou sont émis par ce terminal.

La reconstitution du signal reçu n'est possible que si le receveur connaît la clé utilisée pour la transformation ou si le circuit de décryptage est permanent. La circulation en clair sur les lignes de transmission de la clé de transformation compromet le secret.

La connaissance du principe de cryptage, excepté pour la machine de cryptage IBM, compromet également la sécurité.

Différentes méthodes de négociation préalable sont envisageables en pareil cas, afin de choisir un algorithme et une clé efficaces. Les techniques de transformation par transposition ou substitution sont bien connues et font l'objet de nombreuses études [KAHN 67]. Nous n'y reviendrons donc pas ici.

Dans le cas d'application répartie, se pose en outre le problème de la répartition des clés et de la localisation des codages.

Utilisation

Les techniques de codage utilisées pour protéger des informations stockées sur un support informatique (ou non), servent comme ultime protection : les conséquences de l'intrusion d'un fraudeur dans un système sont ainsi atténuées lorsque ce dernier ignore les clés et algorithmes de codage ; la divulgation n'est cependant que retardée, car après copie des informations et une étude rapide (coûteuse aussi), il est souvent possible de découvrir le principe du codage (hors système).

De nombreuses informations considérées comme *précieuses* sont ainsi protégées dans un environnement informatique : il s'agit soit *d'objets directement confidentiels* (préservation de l'intimité des personnes), soit d'informations permettant d'y avoir accès (mot de passe, catalogues), ou *indirectement confidentiels*.

Efficacité

La connaissance de la clé et de l'algorithme constitue aussi une information d'identification : identification bidirectionnelle (émetteur ↔ récepteur).

Une telle méthode, qui rend incompréhensible pour tout autre usager les informations échangées, ne constitue pas en elle-même une protection suffisante contre les perturbations : après écoute d'une ligne, un fraudeur peut reproduire les mêmes caractères que ceux qui ont permis l'initialisation des échanges et recevoir ensuite des informations (codées) qui ne lui étaient pas destinées. Il se substitue ainsi à un service du réseau par exemple, ou à un usager et peut donc gravement perturber le fonctionnement des échanges. L'introduction d'informations temporelles (dynamiques) constitue un moyen de contourner ce problème (voir mot de passe).

Cependant, l'incapacité actuelle à traiter des informations codées impose que les *données* figurent *en clair* au cours des *traitements*. Tant que l'on ne saura pas crypter la mémoire, l'utilisation des codages sera donc insuffisante à assurer la sécurité.

1.2.3. Identification

Identification et contrôle d'accès

Avant de décider d'autoriser ou de refuser l'accès à un objet, ce dernier doit auparavant avoir été convenablement identifié.

Selon le type d'objets, différentes techniques ont été mises au point : accès à un terminal (clé de contact ..), à un système, à un programme, à un fichier ou à une donnée.

Mais dans tous les cas il s'agit de trouver un moyen d'identifier de manière *unique et difficilement reproductible* ces différents objets. Les solutions logicielles répondent assez mal à cette exigence : toute chaîne de bits étant répétable. Par contre, des contrôles matériels élaborés sont en cours de développement (reconnaissance des empreintes, de la parole, d'une carte magnétique).

Principe

L'information d'identification est communiquée au processeur chargé d'effectuer le contrôle d'accès. Cette transmission pose deux types de problèmes :

. risque de divulgation en cours de transmission,

. erreur d'acheminement : si le processeur auquel elle est destinée a été mal identifié ou a usurpé délibérément une identité qui lui est étrangère.

Les risques sont donc réduits si le nombre des intermédiaires est limité et si l'identification est réciproque. C'est-à-dire, si émetteur et récepteur fournissent chacun leur identification et sont également capables d'effectuer les contrôles sur celle-ci.

1.2.4. Authentification

Les mots de passe

La plupart des systèmes qui se soucient de la protection logicielle des informations confidentielles utilisent des *mots de passe* : à chaque objet à protéger est associé un mot de passe. L'autorisation d'accès à un objet n'est donnée que si le mot de passe correct est fourni. Cette méthode présente certains inconvénients :

. la divulgation du mot de passe est aisée,

. le niveau de contrôle est souvent figé : cette méthode ne permet pas la protection d'éléments d'information très fins ; elle est plus adaptée aux "gros objets" tels que les fichiers.

(cf. Figure 1.1).

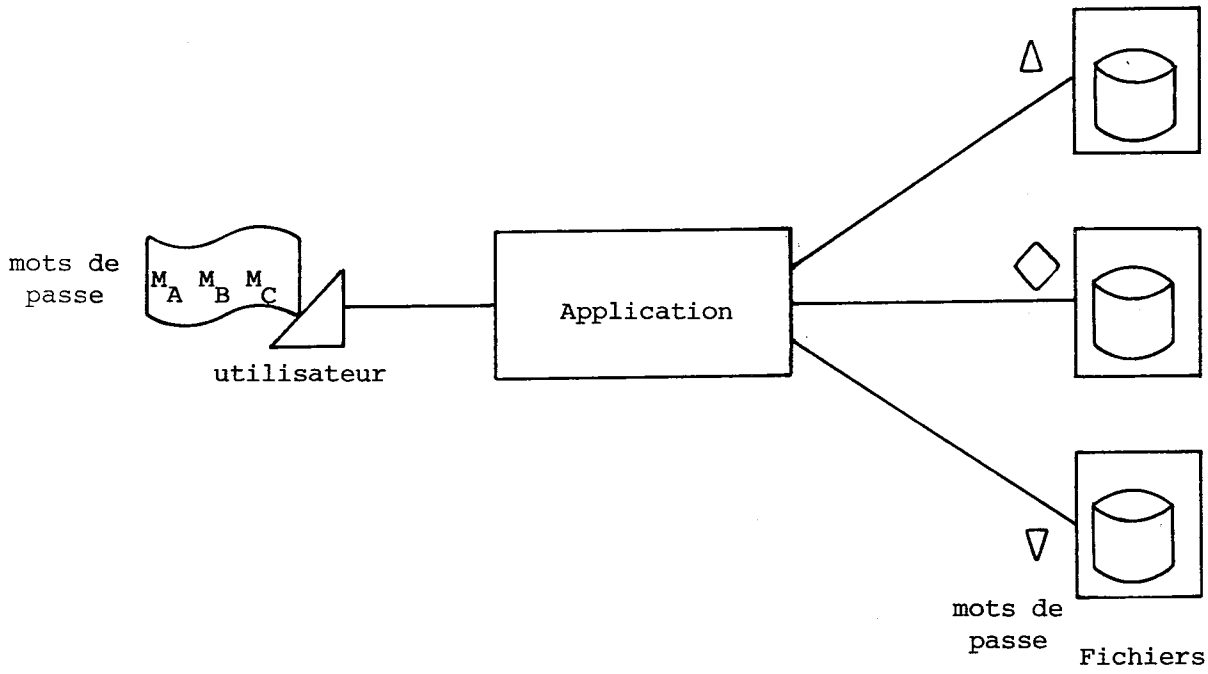
Dans le système MULTICS [SALTZER 74] les informations retenues comme confidentielles doivent être isolées des autres et concentrées dans un fichier séparé bénéficiant d'une protection accrue. Les seuls objets pouvant être manipulés sont des segments de données, des procédures ou des catalogues.

D'autres systèmes [HSIAO 77] proposent un contrôle d'accès au niveau de l'enregistrement. Cependant, une protection vraiment efficace ne peut être assurée que si elle peut, le cas échéant, protéger des granules d'information de faible dimension.

Le mot de passe est une chaîne de caractères, en général choisie par l'usager lui-même que nul autre usager ne doit connaître. Du fait de ces nombreux inconvénients, des améliorations ont souvent été proposées :

. mot de passe attribué par un programme : souvent difficile à mémoriser et donc fragile (car il en existe des copies), excepté dans le cas cité par [SALTZER 74] ;

a) Utilisation de mots de passe



b) Utilisation de capacités

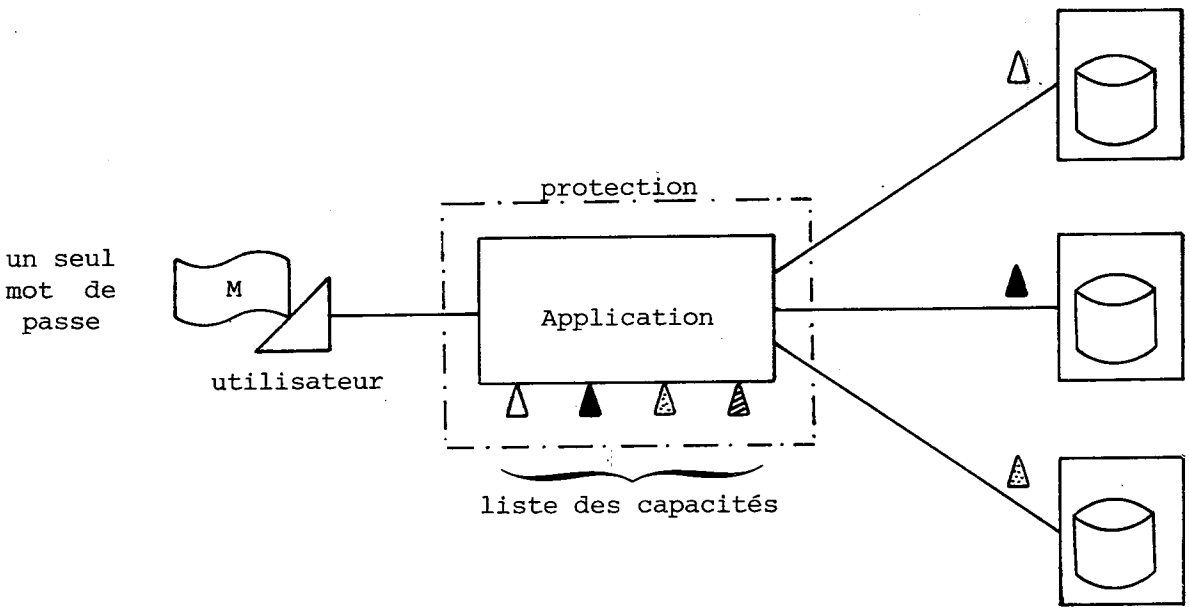


Figure 1.1 - Protection d'accès à des fichiers

. la chaîne de caractères comprend des caractères de positionnement (recul, effacement) ;

. la longueur de la chaîne est ajustée en fonction
- du degré de protection fixé,
- du nombre de tentatives isolées ;

. essai de détection des utilisations frauduleuses des mots de passe (enregistrement des dates de login) ;

. multiplicité des mots de passe : choisis dans une liste ordonnée ou fonction du temps ;

. système de question - réponse pré-enregistrées : éventuellement modifiables. Il peut alors être demandé à l'usager d'exécuter correctement un algorithme ou procédure de la poignée de main (hand shaking procedure) paramétrée ou même temporelle (fonction de l'horloge du système).

Certaines précautions doivent être retenues, quelle que soit la méthode d'authentification utilisée (mot de passe, question-réponse, calcul de fonction temporelle ..) :

. non lisibilité du mot de passe, de la fonction,

. non accessibilité du support : pas de carte,

. évolutivité : dans le cas d'une valeur statique, par exemple, il est nécessaire de pouvoir en changer souvent ; prise en compte d'événements extérieurs (date).

L'authentification représente un complément nécessaire à l'identification et concerne une grande variété d'objets. Il est intéressant de concevoir une telle fonction au niveau global si l'on désire définir un mécanisme complet et dynamique en environnement réseau (cf. 2ème Partie).

La fonction d'authentification (logicielle)

Elle peut être représentée par le schéma suivant :

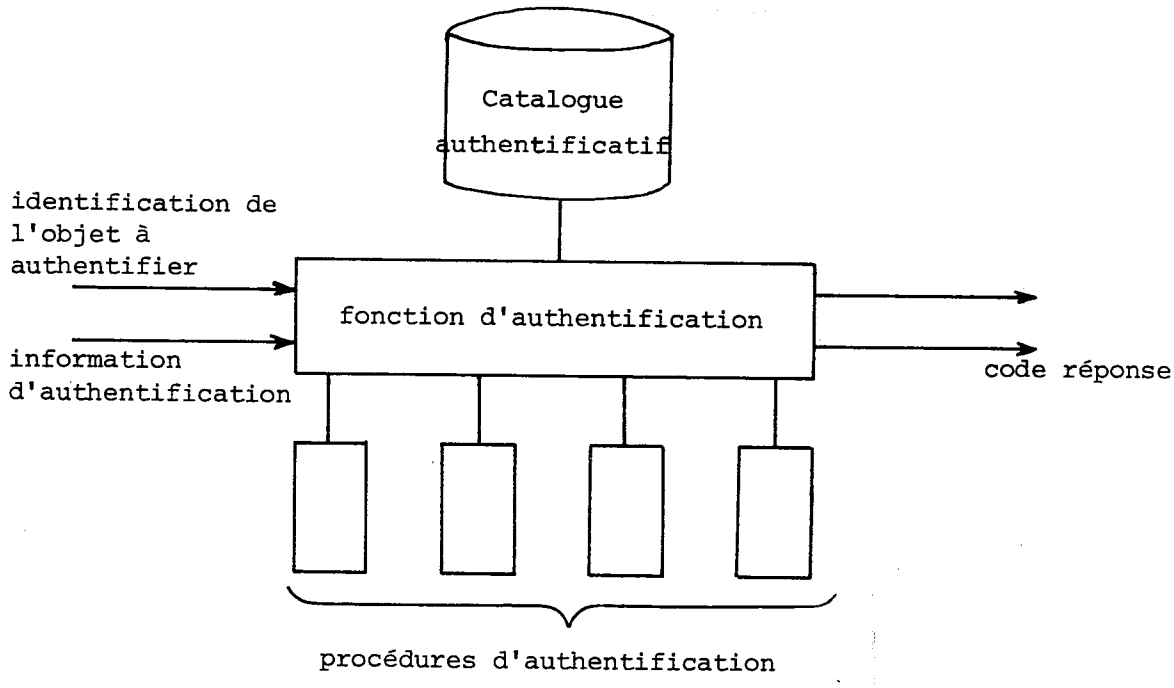


Figure 1.2.

Différentes procédures de contrôle sont associées aux niveaux et aux types d'objets contrôlés ; elles ont un accès privilégié au catalogue authentificatif, dont la protection est fondamentale pour la sécurité de l'ensemble du système.

1.2.5. Langages

La protection peut être introduite à différents niveaux dans les langages. Les langages de programmation usuels sont également concernés par ces problèmes [MORRIS 73].

1.2.5.1. Langage de protection

Exemple du système INGRES

Dans un contexte de bases de données, le système INGRES [STONEBRAKER 74, 75, 76] propose un langage de protection, mis à la disposition du seul administrateur de la base, à l'aide duquel il peut :

. créer des relations partagées,

. cataloguer les données des bases qui nécessitent une protection complémentaire (aucun usager n'a accès au catalogue).

Le propriétaire d'une donnée/relation étant seul autorisé à y accéder (implicitement), peut donner à d'autres utilisateurs des dérogations, sous certaines conditions, afin de les autoriser à interagir sur cette donnée/relation. L'accès à toute relation est donc ainsi protégé par le créateur lui-même.

Les bases considérées sont constituées d'ensembles de relations. La protection entre donc dans ce système comme une nouvelle relation, dite *relation de protection*, dans laquelle sont stockés le nom de la relation à protéger et les conditions d'accès.

Exemple de commande

PERMIT <nom de relation à protéger>
TO <objet à contrôler>
FOR <type d'accès autorisé>
WHERE <qualification>

. Un objet à contrôler peut être soit un usager, soit un terminal, soit l'ensemble de tous ;

. les accès sont classés par type : lecture, écriture, suppression, modification .. et peuvent être complétés par une liste de domaines (n-uplets) accessibles mais non modifiables ;

. la qualification est une expression logique (≃ langage de requête).

Remarques sur INGRES

A partir de l'exemple précédent, on peut faire les remarques suivantes, déjà mises en évidence par les concepteurs de INGRES :

. les contrôles d'accès sont effectués automatiquement par le système INGRES,

. l'administrateur de la base, ou "super-usager" a un rôle très privilégié (concentration du pouvoir de décision de partage),

. l'authentification des usagers est confiée au système UNIX [RITCHIE 74],

. le volume des informations de service,

. les types d'objets protégés sont limités (relation, requête),

. l'encodage des données par des transformations privées n'est pas autorisé.

Une protection globale

La définition d'un langage de protection facilite la manipulation des mécanismes de protection ; elle n'est cependant pas fondamentale à leur fonctionnement.

Le niveau choisi ici se limite à un problème de bases de données et ne présente pas de solution globale. Le rôle centralisateur du super-usager doit également être remis en cause. Il paraît pourtant envisageable de l'étendre à de nouveaux objets.

1.2.5.2. Modification de requête

Par le système INGRES

C'est une *technique de contrôle d'accès* développée dans le système de bases de données INGRES qui présente les avantages suivants : facilité d'implémentation, généralité, simplicité conceptuelle, flexibilité, grande puissance.

A chaque demande d'accès à une relation, le système ajoute les restrictions correspondant à la "qualification" en insérant dans la requête les critères de protection d'accès propres à l'usager propriétaire de la relation.

Le niveau de contrôle est donc plus élevé et confère au mécanisme toute sa puissance. Cependant, les attaques éventuelles à des niveaux intermédiaires compromettent gravement la cohérence et la sécurité des données.

De plus, les contrôles s'effectuent de manière statique : *à la compilation* ; les requêtes précompilées ont donc une durée de validité limitée à celle des relations de protection concernées.

Le coût de cette méthode se décompose comme suit :

. temps de pré-compilation,

. espace de mémorisation des relations de protection.

Afin qu'une telle méthode soit totalement efficace, il faut au préalable s'assurer que les requêtes modifiées fournissent bien les résultats escomptés, c'est-à-dire qu'à tous les niveaux de traduction les modules soient certifiés et contrôlés. Ce qui, vue la complexité et la taille de la plupart des SGBD, présente encore une certaine difficulté.

1.2.5.3. Le confinement

Position du problème (cf. figure 1.3)

Il s'agit de "confiner" un programme en cours d'exécution, afin qu'il ne puisse transmettre d'informations à un autre programme (excepté celui qui l'a appelé).

Un tel objectif participe au maintien du secret de certaines informations et nous intéresse donc ici à ce titre. Les règles qui ont été développées à son sujet [LAMPSON 73] contribuent à la sécurité en général :

. *programme sans mémoire,*

. *transitivité* : l'isolement total d'un programme est peu réaliste, les appels à d'autres programmes ne pouvant être évités. Il faut donc viser à ce que les programmes appelés soient eux-mêmes confinés ;

. *protection des entrées* : le programme à confiner doit permettre au programme appelant de déterminer toutes les entrées autorisées au service qu'il demande ; le superviseur doit pouvoir *garantir* que le programme appelé est conforme aux spécifications du demandeur.

Les solutions

Ce sont des compilateurs de langage évolué qui sont actuellement choisis pour réaliser ce type de protection. Les techniques de contrôle d'accès standard y sont utilisées au maximum [MINSKY 76].
Certains systèmes de conception plus récente, tel HYDRA, présentent de nombreux avantages sur ce point (son approche sera présentée au § 1.2.6).

Un langage de très haut niveau comme ALGOL 68 permet de définir de nouveaux types d'objets auxquels sont associées des procédures d'accès par exemple : les contrôles dynamiques y sont remplacés par des contrôles statiques.

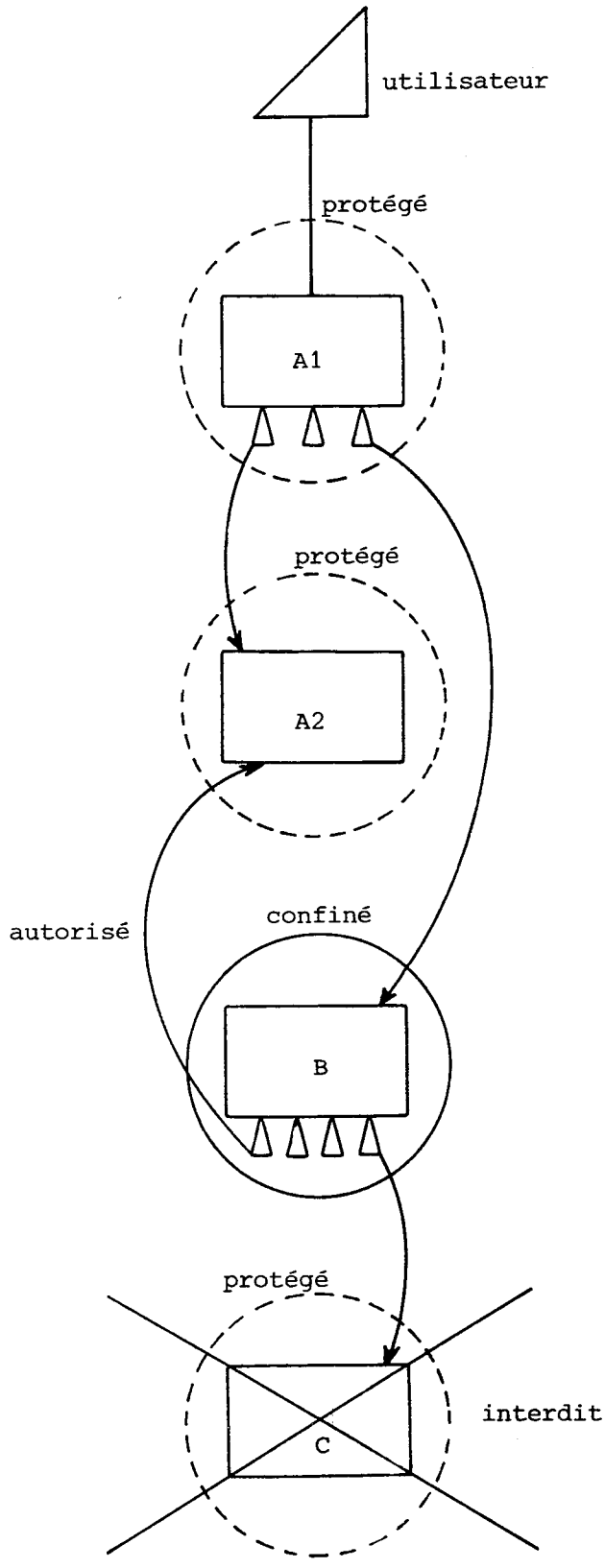


Figure 1.3 - Exemple de confinement

1.2.6. Systèmes d'exploitation

Notre objectif n'est pas de revenir en détail sur les différents types de protection des systèmes existants. D'excellentes analyses en ont déjà été faites [CROCUS 75].Il s'agit de faire ici un simple rappel de quelques mécanismes intéressants, qu'il serait envisageable de développer dans des systèmes répartis pour lesquels se posent de façon accrue les problèmes de synchronisation, parallélisme, allocation de ressources et contrôles d'accès.

Les systèmes d'exploitation usuels visent à une protection *centralisée* (unicité du système) *d'objets simples* (non structurés). Les systèmes de gestion de bases de données, qui ne présentent pas une telle restriction, sont encore le plus souvent incapables de protéger correctement leurs données. Cependant, ils respectent souvent les principes communs suivants :

. pas d'accès direct à l'information,

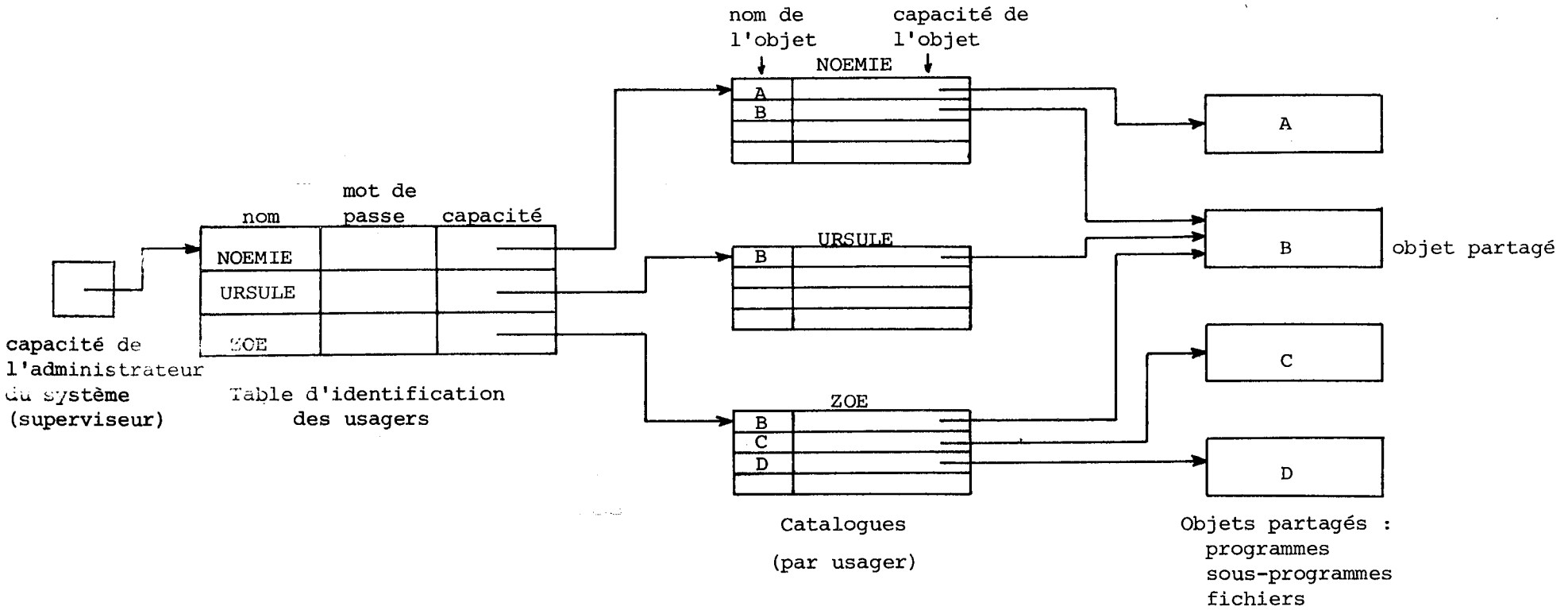
. regroupement des informations précieuses par niveau de protection (notion de noyau protégé et de niveau de secret).

L'utilisation de *sous-systèmes protégés* comme dans MULTICS permet de réaliser des contrôles de protection basés sur le contenu des données [MULTICS].

Le contrôle des accès aux objets gérés par ces systèmes est réalisé de différentes manières : capacités, listes de contrôle d'accès ..

Les systèmes à capacités (reconnus par hardware dans MULTICS) permettent d'associer aux objets des contrôles. Ils sont alors combinés avec des méthodes d'identification-authentification comme l'indique la figure suivante (1.4) :

Figure 1.4 - Un exemple simple d'identification et de contrôle d'accès dans un système à capacités



La présentation rapide du système HYDRA qui suit est donnée à titre d'exemple de système prenant en compte la sécurité. D'autres systèmes non décrits ici, présentent aussi de tels avantages.

Le système HYDRA [WULF 74]

Principe

Les principes retenus pour la conception de HYDRA contribuent au maintien de la sécurité dans ce système. Il s'agit essentiellement de mesures de protection concernant l'ensemble des ressources et contrôlant le fonctionnement des processus : protection, extensibilité, construction modulaire.

Protection

Des opérations primitives constituant un *noyau* permettent de réaliser le contrôle de l'accès aux ressources et d'en créer de nouvelles. Comme dans le cas des langages ou des systèmes de bases de données, l'entité à protéger est définie par les opérations que l'on peut lui appliquer. Les objets qui présentent la même structure et les mêmes nécessités de protection sont regroupés par types. Le seul moyen d'accès à un objet est une "capacité" décrivant les droits de son détenteur sur l'objet considéré.

Performances

Le principe fondamental est donc qu'aucune ressource n'est "manipulée" directement : des mécanismes d'adressage et de protection s'interposent obligatoirement. Il est intéressant ici de connaître les performances et l'efficacité réelle d'un tel système dont les objectifs de protection sont très développés. Les premiers résultats sont à son désavantage : un ralentissement notable des travaux effectués par ce système.

D'autres exemples auraient pu être décrits ici. Mais les références sont suffisamment connues pour que nous y fassions allusion (systèmes MULTICS, PLESSEY, CAP).

1.3. Solutions dans les SGBD locaux

1.3.1. Présentation du problème de la confidentialité des données

Le problème de la protection des informations se pose avec une grande acuité lorsque des données sensibles sont regroupées en quantité importante dans un même centre de traitement : ce qui est donc le cas dans la plupart des bases de données (gérées par des Systèmes de Gestion de Bases de Données). C'est pourquoi nous revenons ici sur la notion de confidentialité appliquée aux données.

Comme il a été dit dans les définitions précédentes, la confidentialité décrit le statut accordé aux données : c'est reconnaître à certains individus, possesseurs des données, le droit d'exiger des protections de leurs informations contre des abus d'utilisation, c'est-à-dire une limitation à certains utilisateurs pour certaines utilisations.

Prenons tout d'abord quelques exemples :

1) 'VICTOR habite VINCENNES' est confidentiel.

Cela signifie qu'un utilisateur de la base de données contenant cette information, ne doit pas connaître à la fois le NOM de cette personne et son DOMICILE, ou bien, dans le cas où le modèle de description de la base est relationnel, on peut dire que la relation HABITE est protégée. En généralisant à l'ensemble des personnes, il peut s'agir alors de protéger la totalité des caractéristiques NOM et DOMICILE (ou toutes les relations HABITE).

Cependant, pour assurer qu'effectivement il n'y aura pas violation de la confidentialité, il faut aussi contrôler toutes les sorties d'information de la base contenant soit NOM soit DOMICILE, en association avec d'autres caractéristiques de la base qui serait discriminante.

Soit la base B des personnes :

nom	cycliste	domicile
ALEX	oui	VINCENNES
IDA	non	FONTENAY
NESTOR	oui	FONTENAY
OSCAR	non	NOGENT
SAMSON	non	NOGENT
VICTOR	oui	VINCENNES

et les requêtes suivantes :

Q1 : nom de tous les cyclistes ?
R1 : ALEX, NESTOR, VICTOR.

Q2 : domicile d'ALEX ?
R2 : VINCENNES

Q3 : domicile de NESTOR ?
R3 : FONTENAY

Q4 : nombre de cyclistes habitant à VINCENNES ?
R4 : deux

Par définition, on sait donc que VICTOR habite VINCENNES. Cette information a été utilisée pour établir la réponse à la question Q4 (question *statistique*).

Le nom de VICTOR a été associé à d'autres noms (ALEX et NESTOR) moins protégés : il y a donc eu dégradation du niveau de secret.

Q0 : domicile de VICTOR ?
R0 : refus de réponse (car confidentiel)

} requête supposée interdite ici pour respecter la confidentialité.

Il paraît donc important ici de prendre des précautions avant d'autoriser les questions du type statistique en particulier.

2) 'La moyenne des salaires' est confidentielle.

Il s'agit de faire en sorte qu'aucun usager ne puisse réunir suffisamment d'éléments pour calculer cette moyenne : le seuil des réponses concernant les salaires est donc limité. L'établissement d'un *historique* des informations précédemment extraites de la base, devrait fournir les moyens d'un tel contrôle.

3) 'La situation familiale' d'une personne ne peut être communiquée que pour l'établissement de son bulletin de paie.

Le contrôle est donc ici plus précis :

. soit limiter au seul usager associé à cette manipulation l'accès à l'information,

. soit limiter à un *programme standard* de calcul de paie,

. soit demander une autorisation explicite (administration, opérateur) au moment de l'utilisation.

Ces exemples appellent quelques remarques plus générales :

. La plupart des SGBD ne permettent évidemment pas d'effectuer de tels contrôles : ils se limitent souvent à interdire totalement l'accès à une information considérée comme absente momentanément de la base.

. Il est rarement possible de distinguer une *valeur* de donnée, en particulier dans une base (exemple : la relation HABITE peut être protégée pour toutes les personnes et non seulement pour un seul individu).

. Le *modèle de description* des données intervient donc beaucoup sur le mode de protection envisageable. Il doit permettre, pour être efficace, au minimum de représenter :

- le niveau de secret d'une donnée,
- le type d'accès (ou programme) autorisé,
- le contexte d'autorisation (relations avec d'autres données, d'autres opérations).

Un modèle relationnel pose évidemment plus de problèmes que d'autres, dans la mesure où à partir d'une seule information, il est possible de retrouver les liens avec d'autres informations.

. Un mécanisme de protection doit permettre de réaliser des contrôles à la fois sur des données élémentaires et sur des ensembles de données reliées sémantiquement : c'est un *changement de dimension* non négligeable. De plus, le volume des informations regroupées dans une base de données, accroît la complexité des mécanismes à mettre en oeuvre : degré de partage, *volume des informations de contrôle*.

. Toute méthode de protection doit tenir compte du fait qu'un usager des bases de données a déjà *une certaine connaissance de la base*. La difficulté de la protection réside donc aussi dans le fait qu'il n'est jamais possible de savoir exactement quel est ce degré de connaissance.

Exemple :

Supposons qu'un usager sache que, dans la base B des personnes présentée plus haut, VICTOR et ALEX habitent la même ville ; il lui suffira de poser la question Q2 : domicile d'ALEX ? pour connaître le domicile de VICTOR.

Aucune analyse des requêtes ne permettra de savoir ici qu'il y a eu divulgation d'information confidentielle (ici : le domicile de VICTOR).

Une solution est de supposer qu'il y a une connaissance "moyenne" (à définir) préalable de certaines informations de la base :

. avantage : limiter les risques de divulgation (sans les éviter totalement, de la part de ceux qui dépassent cette connaissance moyenne) ;

. inconvénient : empêcher des usagers n'ayant pas cette connaissance dite "moyenne", d'accéder à des informations non confidentielles et rendre ainsi la base difficilement exploitable pour eux.

Types de confidentialité

Différents types de confidentialité peuvent cependant être relevés selon que l'objet à protéger et les manipulations qui en sont faites sont plus ou moins complexes.

a) *Confidentialité élémentaire*

l'information simple est en elle-même confidentielle. Exemple : le salaire maximum. Ce cas est cependant très rare et consiste en fait en un simple cas particulier du type b).

b) *Confidentialité multiple*

c'est l'association de certaines informations ensemble qui est confidentielle. Exemple : DUPOND gagne 7 000 F/mois.

Nous considérons donc indifféremment les informations confidentielles de types a) ou b).

La protection d'informations de type a) pose peu de problèmes : un simple verrou d'accès suffit, ou sa suppression dans la description de la base.

Les éléments d'information dits confidentiels sont souvent *dispersés* dans les bases de données : il ne s'agit pas le plus souvent de protéger toutes les valeurs d'une même caractéristique, mais seulement certaines occurrences ; c'est en cela que réside la difficulté, car on suppose en général que les valeurs sont plus fluctuantes (mises à jour) que les structures et donc le mécanisme de protection doit être *évolutif*. Pour obtenir une meilleure protection, il faudrait donc que chaque valeur (= donnée élémentaire) soit individuellement protégée. Une telle méthode ajoute une forte *charge indue* aux fonctions d'accès, difficilement supportable en exploitation courante.

Violations

En outre, quel que soit le type de confidentialité, les attaques - c'est-à-dire tentatives de violation de la confidentialité - présentent un degré de gravité variable ; on peut les classer en deux catégories :

1) *Violation directe*

l'information confidentielle est divulguée en valeur exacte (donc par une seule requête).

Exemple : Q : combien gagne DUPOND ?

R : DUPOND gagne 7 000 F/mois

2) *Violation indirecte, ou partielle*

l'information confidentielle est divulguée en valeur approchée,

- soit par simple comparaison de valeur (directe)

exemple : Q1 : DUPOND gagne-t-il autant que DURAND ?

R1 : Oui

Q2 : combien gagne DURAND ?

R2 : DURAND gagne 7 000 F/mois.

Un nombre limité de requêtes fournira un ensemble de solutions de plus en plus restreint jusqu'à la violation finale de la confidentialité (divulgation de la valeur).

- Soit par rapprochement entre ensembles plus complexes. Le recoupement des différentes réponses obtenues fournit la solution finale : des valeurs *pivots* plus ou moins complexes permettent d'atteindre une caractéristique discriminante non protégée.

Ce type d'attaque est possible dès qu'une valeur à protéger est communiquée en association avec d'autres valeurs non protégées, c'est-à-dire de niveau de secret différent. *Cette dégradation du niveau de secret* entraîne la divulgation de l'information confidentielle.

Exemple : Q1 : quels sont les salaires de tous les employés et combien ont-ils d'enfants ?

R1 : sal1 nb1

sal2 nb2

: :

Q2 : combien DUPOND a-t-il d'enfants ?

R2 : n

Si n ne figure qu'une seule fois dans R1, alors le salaire de DUPOND est divulgué. Ici c'est "nombre d'enfants" qui est utilisé comme information *pivot*, son niveau de secret étant inférieur à celui du "salaire".

- Soit par intersection de résultats statistiques ou de moyennes (totaux ..).

Ce qui vient d'être dit ici concerne la *confidentialité des données*. Mais on peut se demander aussi s'il n'existe pas un autre type de confidentialité à maintenir, à savoir : y a-t-il des utilisations de données qui doivent être tenues secretes ou des utilisateurs par exemple?

L'environnement de fraude doit aussi être considéré ici : des accès parasites aux informations confidentielles peuvent être tentés sans que les contrôleurs de régularité des requêtes n'aient détecté de fraude.

Exemples de fraudes :

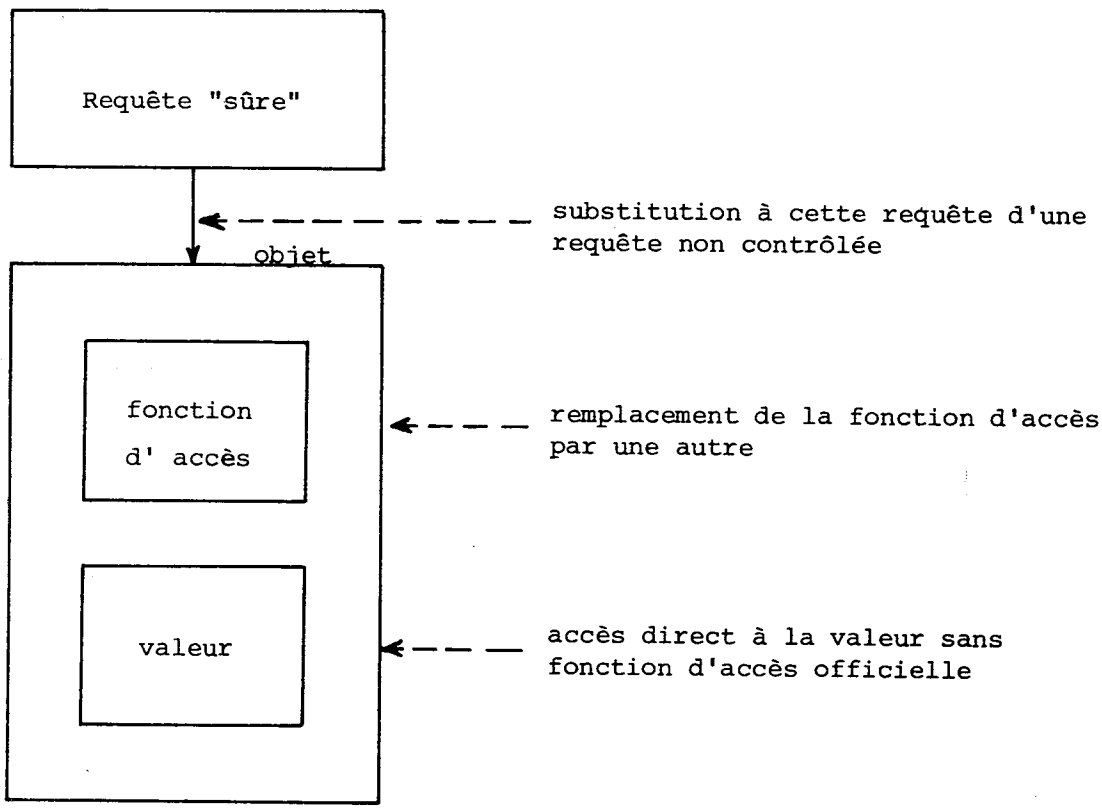


Figure 1.5.

1.3.2. Les méthodes

Les mesures de maintien de la confidentialité des données consistent d'une part à limiter les utilisations, les utilisateurs et les objets accessibles, d'autre part à contrôler les demandes et les accès.

La confidentialité est assurée :

. soit au niveau de la description (dans le PSB en IMS figurent les droits d'accès et les possibilités de traitement. Le PSB n'est pas sous la responsabilité du programmeur) ;

. soit au niveau du langage : par des verrous ou des mots de passe introduits dans les macros (comme dans SOCRATE).

1.3.2.1. La limitation des requêtes

Il est très difficile de limiter les requêtes qui comportent un risque de violation indirecte. A notre connaissance aucun SGBD ne le fait. Plusieurs possibilités ont été envisagées :

. interdire totalement les requêtes portant sur des données confidentielles ou ayant pour résultat certaines de ces données,

. modifier les requêtes de telle sorte qu'elles ne violent pas la confidentialité :

- soit par une extension de l'ensemble réponse (résultat statistique étendu par exemple),
- soit par une restriction de l'ensemble réponse.

La modification doit tenir compte des contraintes de confidentialité, du type d'utilisation, des droits de l'utilisateur.

Exemple :

. Soit une question Q sur la base B :

B : ensemble de N personnes (nom, profession, salaire)

Q : "nombre de personnes gagnant entre 7000 et 8000 F/mois ?"

. Les salaires de certaines personnes sont confidentiels.

. Supposons que n salaires répondent à la sélection Q. Quelle méthode adopter qui ne compromette pas la confidentialité des salaires ?

- refuser de répondre signifie que des personnes dont le salaire est confidentiel, appartiennent à cette catégorie ;

- étendre l'ensemble des réponses aux personnes gagnant entre 6000 et 8 000 F, de telle sorte que le nombre de personnes satisfaisant à cette condition soit plus important et concerne une grande quantité d'informations non confidentielles ;

- limiter la réponse aux informations non confidentielles ; donc la réponse à la question Q est élaborée sur B' = B - (données confidentielles

Les insuffisances sont alors évidentes. Un enregistrement de l'historique des questions et réponses évite de compromettre trop rapidement la confidentialité, dans la mesure où l'on sait déterminer le seuil critique des réponses autorisées successivement. Des études visant à limiter les réponses permettront sans doute bientôt de diminuer les risques [DOBKIN 76].

1.3.2.2. Limitation des informations accessibles

La limitation peut se faire au niveau de la description des données ou des fonctions d'accès aux données.

SOCRATE permet de définir une sous-structure adaptée aux droits de l'usager : c'est sa vue de la base dont la structure a été fixée à la création. Mais il n'autorise pas la différenciation des réalisations d'une même entité ou caractéristique (au point de vue des droits d'accès).

Exemple : 1. l'usager U1 a accès à tous les SALAIRES de la base,
2. l'usager U2 n'a accès qu'aux SALAIRES des employés du service des ventes,
3. l'usager U3 n'a accès à aucun SALAIRE.

La contrainte 2 ne peut être prise en compte dans la plupart des descriptions.

1.3.2.3. Classement des usagers

La notion de super-usager ou administrateur est maintenant acquise dans la plupart des SGBD.

Dans SOCRATE, une base des bases (ou répertoire des bases), accessible seulement par l'administrateur de la base, répertorie l'ensemble des bases, le nombre des utilisateurs et leurs droits d'accès à certains processeurs. Les autres usagers peuvent être classés en deux catégories : ceux qui peuvent utiliser le langage de requête et ceux qui ne peuvent utiliser qu'un ensemble de macro-instructions.

Les informations peuvent également posséder un statut (secret, top secret ..) qui correspond à certaines classes d'usagers.

1.3.2.4. Contrôles

1) des utilisateurs :

par les méthodes d'identification, authentification (login ..)

2) des utilisations :

les types d'utilisations reconnues par les systèmes se limitent le plus souvent à lecture/écriture. Dans le cas des bases de données, il paraît important d'y ajouter d'autres types, tels que copie, mise à jour, comparaison avec niveau de secret ; les sorties d'information ne sont pas contrôlées (les programmes ne sont pas confinés, les demandes antérieures sont oubliées) ;

3) des accès aux informations :

. dispersion des informations afin de rendre très difficile la reconstitution d'informations confidentielles,

. interposition de filtre, d'adressage indirect sur les données,

. modification des informations (brouillage, codage),

. association d'une fonction d'accès, non contournable, tenant compte du contexte de la demande.

On peut donc en conclure ici que la possibilité de définir des fonctions d'accès sur des éléments d'information précis et d'accorder un statut de secret aux données, présente une garantie minimum de protection des données pour des applications bien contrôlées.

Un modèle de maintien de la confidentialité doit donc permettre :

. de décrire les contraintes et de les vérifier,

. de contrôler les usages des informations (dans le temps),

. de maintenir le niveau de secret.

1.3.3. Exemple de SGBD : SOCRATE

Avant d'aborder le problème spécifique des bases réparties, il nous a paru intéressant d'étudier un SGBD particulier sous l'angle de la confidentialité.

Le choix s'est porté sur le système SOCRATE dans la mesure où nous disposons à Grenoble d'une documentation complète sur ce système et surtout parce qu'il présente des possibilités de définition de structure et de protection assez élaborées comparativement à d'autres systèmes.

1.3.3.1. Présentation générale du système SOCRATE

SOCRATE est un système général de gestion de bases de données conversationnel. Il offre les possibilités suivantes :

- définition d'une *structure* des données à manipuler : utilisant les notions d'entité, de caractéristique et de relation, avec possibilité de chaînage des entités et de définition d'inverses. Un langage adapté permet de la décrire, c'est-à-dire :
  - . la structure elle-même,
  - . les fonctions d'accès à certaines données (clés)
  - . l'implantation des données (tableau, liste de valeurs ..).

Les contrôles des types ainsi définis sont effectués à la création de chaque caractéristique d'entité. Les entités sont numérotées par le système dans l'ordre de leur création.

- *Création* de bases de données respectant une structure préalablement définie. Le système met à jour les différents pointeurs, chaînages entre les entités. La création s'effectue au niveau de l'entité, après contrôle des droits des usagers.

• Interrogation, mise à jour des bases. Un langage (+ macro-langage) permet de formuler des *requêtes* "compréhensibles", constituées de filtres et de commandes (CREER, GENERER, RETIRER, INTERROGER, METTRE A JOUR, LIRE, ECRIRE).

Le système SOCRATE contient donc (figure 1.6) :

. son système d'accès aux données,
. son système d'allocation de ressources (mémoire),
. un langage de requête et son compilateur,
. son système de gestion des terminaux,
. un éditeur de texte,
. un langage de structure,
. un système de reprise et d'archivage (avec CHECK POINT,RESTART).

1.3.3.2. Les protections du système SOCRATE (figure 1.7)

1) Un "REPERTOIRE DES BASES", considéré comme une base particulière, géré par un "ADMINISTRATEUR DES BASES", protège l'accès à toutes les bases SOCRATE. Il contient :

. les noms des utilisateurs,
. les droits d'accès à certains processeurs,
. la liste des bases gérées par le système.

2) Différents droits d'accès sont pris en compte par le système :

• au niveau des structures :

possibilité de déclarer une ou plusieurs sous-structures destinées à différentes catégories d'utilisateurs. La sous-structure contient alors un sous-ensemble des données de la structure et les droits d'accès à ces données ;

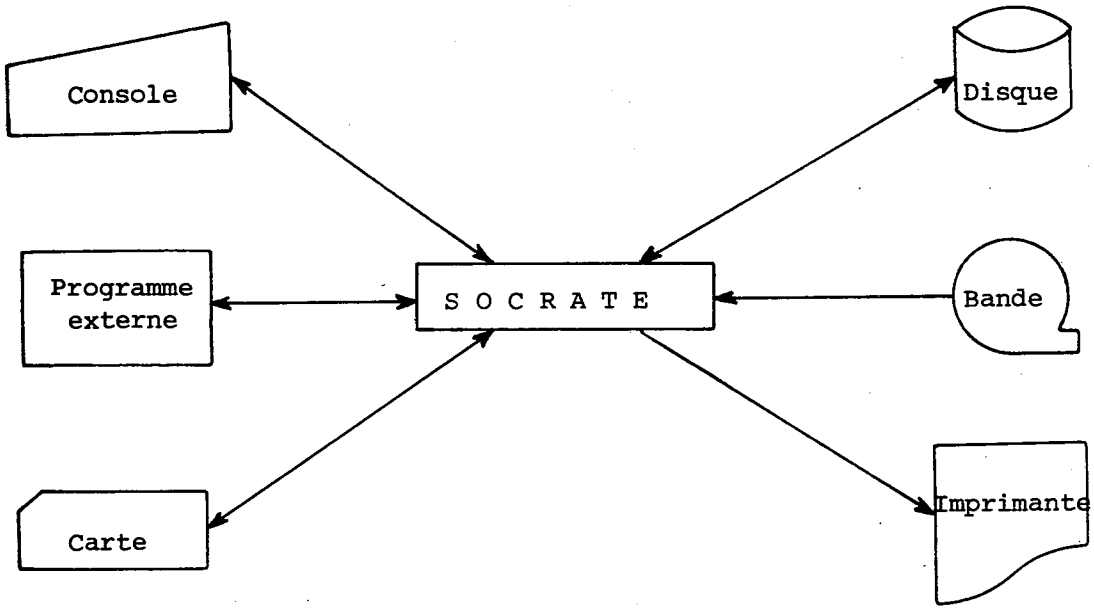


Figure 1.6

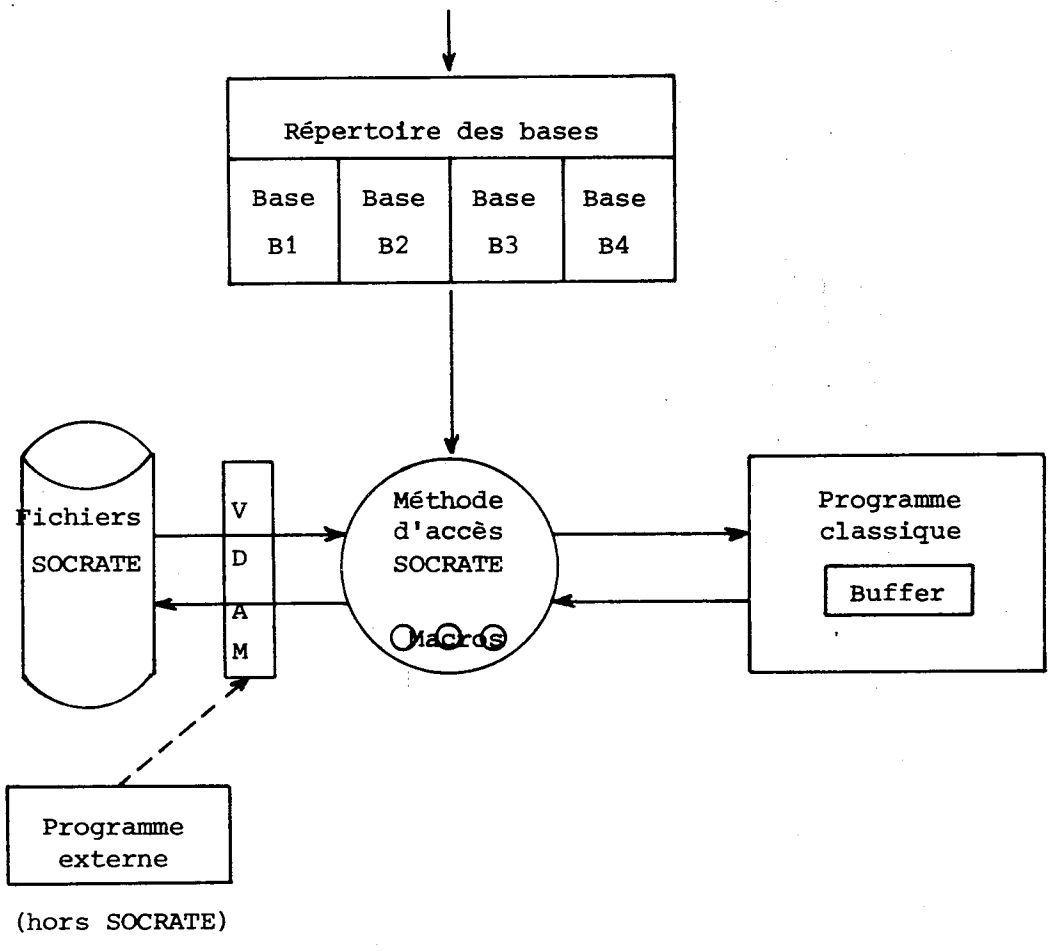


Figure 1.7

• au niveau du langage :

on distingue deux catégories d'utilisateurs :

. ceux qui ont le droit d'utiliser le langage de requête,

. ceux qui ont le droit d'utiliser seulement un ensemble de macro-instructions ;

• au niveau de l'accès aux bases :

une liste des utilisateurs autorisés et de mots de passe est associée à chaque base de données et vérifiée à chaque nouvelle demande d'accès à une base. Des mécanismes de protection évitent le déroulement simultané d'actions incompatibles (blocage, déblocage du fichier ..).

3) Les accès aux données (fichiers) sont à la charge du système SOCRATE : le langage de requête ne contient pas d'instructions d'accès aux fichiers.

Cependant, la méthode d'accès physique utilisée par SOCRATE, peut être utilisée par des usagers extérieurs : les fichiers physiques qui constituent la base SOCRATE sont donc accessibles hors du contrôle de SOCRATE.

1.3.3.3. Le problème de protection des bases de données

Le maintien de la confidentialité dans les systèmes de gestion de bases de données n'est pas actuellement encore résolu de façon efficace. On ne peut en effet se contenter ici d'étendre les mécanismes de protection des systèmes d'exploitation aux bases de données. Certains caractères spécifiques sont à mentionner :

1) grand nombre d'objets à protéger (chaque donnée de la base),

2) différents statuts de protection en fonction des usagers, mais aussi des valeurs des données elles-mêmes ou de sous-ensembles d'entre elles,

3) le niveau de la protection : fonction des concepts manipulés par le système de gestion de la base (entités, relations, caractéristiques ou catégories, ..),

4) la multiplicité des fonctions d'accès due à la puissance du langage de manipulation, en particulier.

1.3.3.4. SOCRATE et le secret

Le niveau de protection est ici multiple : celui des entités, des caractéristiques et des relations. Les structures définissables en SOCRATE peuvent être très complexes : multitude de chaînages, pointeurs, inverses. Leur protection en est donc assez difficile. Le langage de requête lui-même permet d'énoncer des critères très fins et de choisir une méthode d'accès adaptée à l'application (le SUIVANT, par CLE, par NUMERO d'ENTITE ..).

Il paraît donc à ce niveau extrêmement difficile de diagnostiquer quelles sont les questions qui portent atteinte à la confidentialité afin de les refuser globalement. Les solutions proposées permettent seulement de limiter la structure accessible à une sous-structure associée à un groupe d'usagers.

De plus, les fonctions d'accès élémentaires, bien que gérées par le système SOCRATE, peuvent être combinées de telle sorte qu'elles mettent en péril l'intégrité et la sécurité de la base : la possibilité d'accéder à toutes les caractéristiques de la base suivant n'importe quel chemin de parcours en est un exemple.

Le "répertoire des bases" qui devrait contrôler l'ensemble des accès aux bases, n'est en fait qu'une base "ordinaire" et donc à ce titre facilement "accessible" (la connaissance du mot de passe de l'administrateur suffit).

1.3.3.5. Solutions envisageables

Pour limiter au maximum les questions indiscrètes, chaque usager devrait être considéré en particulier et chaque donnée de même. Dans une telle prespective, on atteint rapidement un encombrement limite dont la charge n'est plus supportable par le système. De plus, on ne sait pas actuellement exprimer certaines contraintes.

On peut donc envisager différentes solutions partielles :

1) chaque usager n'utilise qu'une sous-structure décrivant ses droits sur un ensemble limité d'objets. Cela ne permet cependant pas de distinguer les occurrences de valeur différente. Une sous-structure est un sous-ensemble des identificateurs de la structure. La sous-structure permet (cf. figure 1.8) :

. au niveau de chaque identificateur d'entité, de définir un droit :
  - de création (C)
  - de suppression (S)
  - de suppression et création (SC),

. au niveau de chaque identificateur de mot, texte, valeur numérique ou référence, de définir un droit de traitement :
  - d'interrogation seule (I),
  - d'interrogation et de mise à jour (IM),

. de réordonner les identificateurs de même niveau,

. d'ignorer certains identificateurs de la structure.

On remarque donc ici que toutes les occurrences d'une entité seront traitées de la même façon, car les droits de traitement sont globaux à l'ensemble des réalisations d'entités.

Pour une structure donnée, il est donc possible de définir un ensemble de sous-structures.

2) Chaque occurrence d'une donnée est marquée par un droit d'accès intrinsèque (ou dépendant d'autres droits associés à d'autres données). Dès que la base est volumineuse, un tel marquage est complexe. Une telle modification implique que :

. d'une part les fonctions d'accès du système SOCRATE soient complétées par un contrôle du marquage,

. d'autre part la représentation des objets soit plus volumineuse (processeur de gestion de mémoire..).

(Cf. Figure 1.9).

Toute mise à jour remet en cause ce marquage et doit donc être réservée à une certaine catégorie d'usagers.

Cette méthode comporte évidemment certaines redondances s'il n'est pas possible de déclarer globalement des contrôles concernant des sous-ensembles d'informations, contrairement à ce qui est envisagé dans la méthode suivante.

3) Un jeu de catalogues vient s'interposer entre la structure et les données afin de permettre les contrôles. Chaque catalogue concerne une vue limitée de la structure et les filtres (contraintes) associés à chaque caractéristique y sont mentionnés. Les requêtes (ou macros) émises par un usager doivent donc, avant la compilation, être sélectionnées (filtrées) :

. soit refus complet,

. soit aménagement de la requête afin qu'elle ne viole pas la confidentialité.

(Cf. Figure 1.10).

Cette troisième solution est très complexe à mettre en oeuvre dans la mesure où il est le plus souvent difficile de filtrer efficacement des questions : par des informations parallèles, non confidentielles, il est possible de dégager des informations confidentielles sans que le système ne détecte la fraude (exemple : par comparaisons successives d'un ensemble de valeurs ..).

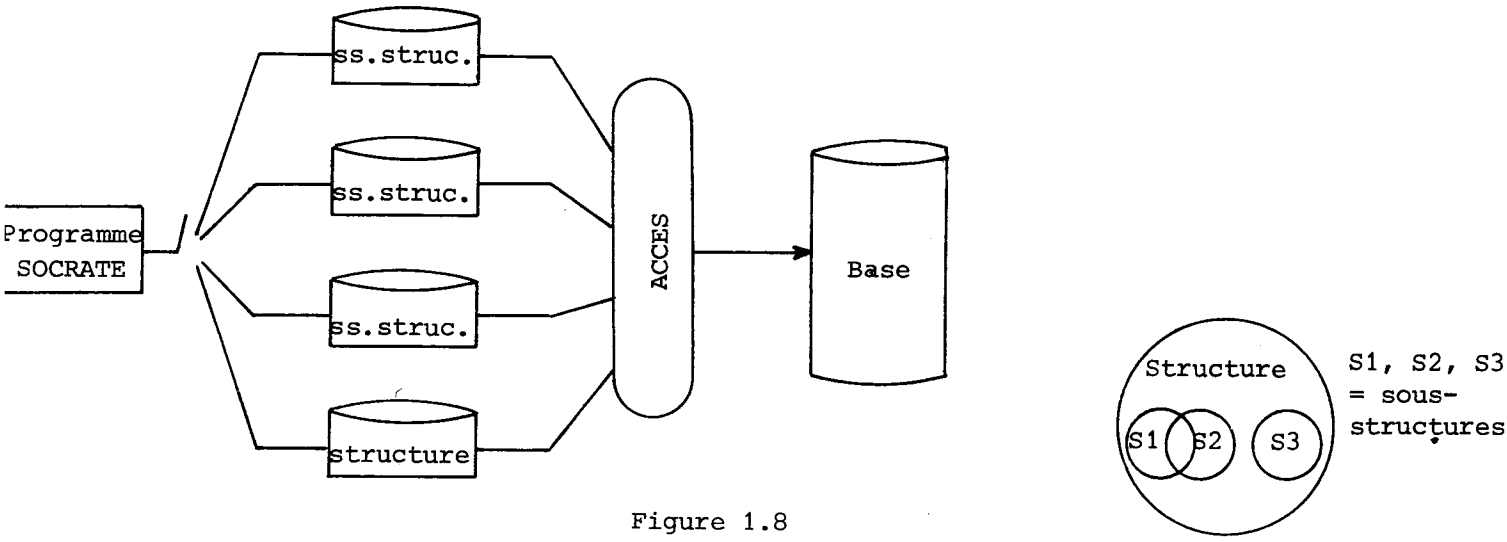


Figure 1.8

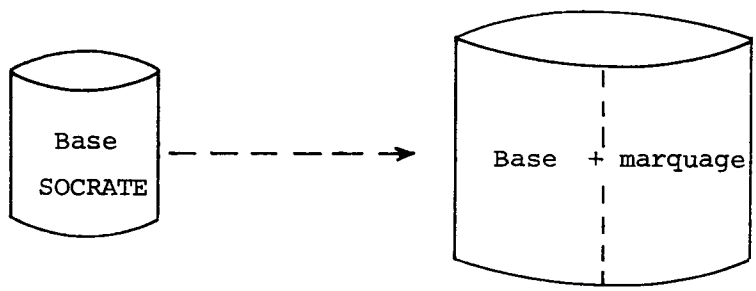


Figure 1.9

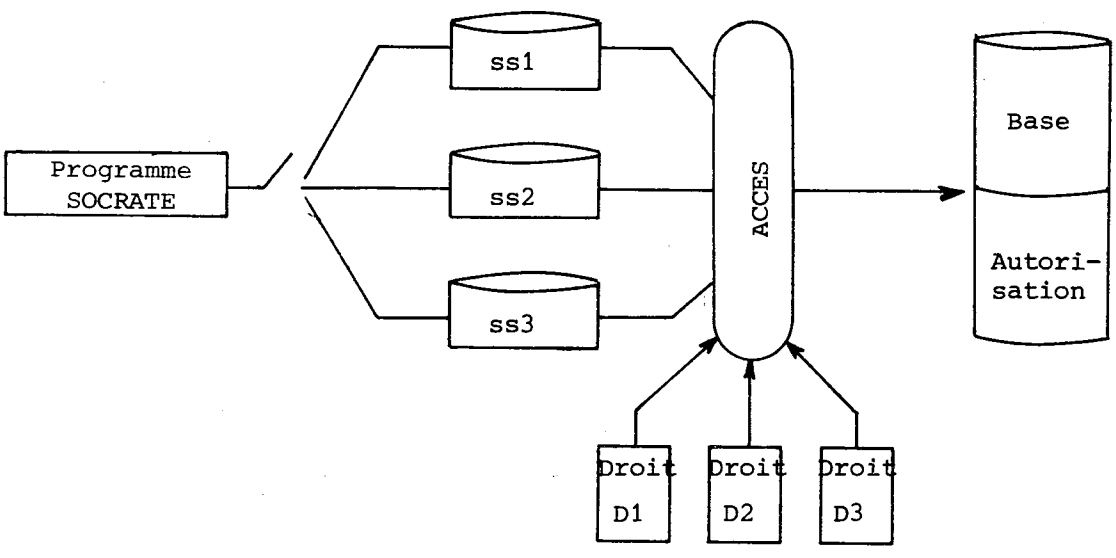


Figure 1.10

Les solutions proposées ici au problème de la confidentialité, tant dans les systèmes locaux que dans la gestion des données locales, n'ont pas d'efficacité totale. Cependant, elles participent, à leur niveau, à une meilleure protection des données dans un milieu homogène (un seul système d'exploitation, un type de SGBD). Avec une efficacité moindre, elles peuvent être étendues à un environnement plus hétérogène, dans lequel la communication revêt une particulière importance, comme celui du réseau.

Par ailleurs, un mécanisme central de protection peut être défini, s'appuyant sur les systèmes locaux, avec lesquels une coopération fructueuse peut être établie : ce milieu est alors considéré comme étant sûr, fiable et protégé (c'est-à-dire ne propageant pas les erreurs, assurant des reprises confidentielles et sans fuite).

C'est donc dans cette direction que le problème de la confidentialité doit maintenant être abordé afin de permettre le développement d'applications réparties gérant des données confidentielles.

DEUXIÈME PARTIE

INCIDENCE DE L'INTRODUCTION DES RÉSEAUX

L'incidence de l'introduction des réseaux sur la confidentialité, peut être envisagée sous deux aspects :

. d'une part au niveau du système, c'est-à-dire de la communication entre systèmes d'exploitation hétérogènes, de l'accès à ce système de communication et de la répartition fonctionnelle des applications sur un réseau ;

. d'autre part, sous un aspect spécifiquement données : duplication des informations (copies multiples), gestion des catalogues, modèle de coopération, intégrité de données réparties.

Certaines solutions présentées dans la première partie ne peuvent être utilisées dans ce nouvel environnement qu'en respectant certaines réserves que nous commencerons par présenter ici.

Les aspects système sont illustrés essentiellement par des remarques concernant le réseau CYCLADES : fonctionnement, utilisation et améliorations envisageables.

Par ailleurs, l'utilisation de systèmes de gestion de bases de données hétérogènes répartis sur les sites d'un réseau, pose de nouveaux problèmes d'accès, résolus soit par l'intermédiaire de logiciel de connexion rudimentaires (tels le concentrateur multi-connexion), soit par de nouvelles architectures de coopération implantées sur les réseaux. De nouvelles relations sémantiques entre données peuvent alors être concrétisées et contrôlées glogalement (intégrité).

2.1. Extension des solutions locales

Les méthodes *d'identification* et leur véritable complément que sont les procédures *d'authentification*, demeurent très utilisées par les systèmes de communication. Cependant, pour tenir compte

. de la variation rapide du nombre d'utilisateurs simultanés (faible temps de connexion),

. de la stratification des interfaces entre usagers et programme d'application,

de tels mécanismes doivent être concis afin d'éviter une surcharge importante sur les communications dans le réseau.

La méthode d'authentification par mot de passe est très répandue dans les systèmes à temps partagé, pour contrôler l'accès aux terminaux ou aux fichiers. Son utilisation sur un réseau suppose que :

. les protocoles y soient préparés : c'est-à-dire tiennent compte du caractère de secret de certaines informations (voir le protocole appareil virtuel CYCLADES) ;

. les systèmes garantissent que les mots de passe sont protégés (les systèmes étant eux mêmes authentifiés au préalable) ;

. la structure d'implantation choisie ne provoque pas une charge critique du réseau : selon la topologie du réseau entre autres, le mécanisme d'authentification peut donc être centralisé ou réparti.

La fonction d'authentification représentée au § 1.2.4 n'est donc pas remise en cause par la répartition. Cependant, toute vulnérabilité au niveau de la circulation des informations compromet cette fonction dans son ensemble. Le plus sûr est déjà de limiter cette circulation en choisissant une implémentation sur le réseau.

Conséquences de la répartition

L'utilisation d'un langage de protection par un *administrateur central* n'est pas remise en cause. C'est seulement la fonction nouvelle de cet administrateur qui reste à définir : ses privilèges doivent être partagés entre différentes autorités qui coopèrent pour en garantir l'efficacité. Cette fonction de protection ne peut être isolée du contexte général des applications réparties et doit être intégrée à un système global de protection qui ne serait pas nécessairement basé sur un modèle de description relationnel (comme INGRES).

La méthode de modification de requêtes est utilisable dans un environnement réparti ; les contraintes sont alors issues des différentes bases au lieu d'une seule ; l'algorithme proposé par INGRES s'adapte uniquement à un modèle relationnel de description de données, c'est donc une telle description qui devrait être retenue pour l'ensemble des bases, si l'on veut pouvoir appliquer directement les conclusions de cette étude. Nous reviendrons sur ce choix au cours de la troisième partie.

La mise en oeuvre peut être envisagée en respectant l'un des schémas suivants :

a) contrôle dans chaque base des contraintes locales sur les portions de requêtes à traiter localement : ce qui suppose qu'aucune contrainte nouvelle n'est introduite au niveau global ;

b) contrôle global de l'ensemble des contraintes : la formulation à ce niveau des contraintes issues des bases présente certaines difficultés dans le cas de duplication, par exemple. La cohérence y est cependant plus facile à assurer.

Par ailleurs, le problème du *confinement* devrait être abordé pour des systèmes répartis, dans lesquels les processus sont eux-mêmes répartis et éventuellement confinés. Cela reste pour l'instant une question ouverte.

Les études en cours concernant *la communication entre systèmes* et plus précisément entre systèmes protégés, ouvrent de nouveaux horizons pour l'exploitation des réseaux. Les noyaux protégés réalisés actuellement sur une machine unique pourraient ainsi être répartis. Les communications entre ces noyaux sont alors protégées et gérées par un système appelé système réseau.

La conception d'une machine réseau logique (MRL) [DU MASLE 74] participerait alors à la mise en oeuvre d'applications réparties dont elle serait le support. Une telle perspective sort du cadre de notre étude, mais présente de réels avantages en ce qui concerne les possibilités de prise en compte de la sécurité sur l'ensemble du réseau, lorsqu'une telle machine l'autorise.

2.2. Incidence sur le système

Le but de cette seconde partie n'est pas de donner un historique complet des réseaux, mais de permettre, à partir des expériences acquises en ce domaine et des formalisations en cours, d'analyser les outils existants et leur adéquation à résoudre les problèmes de secret et de contrôle auxquels les applications, concernant les bases de données en particulier, se heurtent actuellement.

Le terme de confidentialité ou de secret utilisé ici pour qualifier certains éléments de réseau, n'est pas une caractéristique généralement retenue dans les études de réseaux. Nous espérons que son introduction contribuera à une meilleure description des composants d'un réseau.

L'exemple du réseau CYCLADES nous étant le plus familier, sera étudié plus en détail dans son implémentation sous SIRIS 8.

Les critiques formulées à son sujet n'ont évidemment pas de portée générale, mais permettent de dégager quelques principes d'étude afin de connaître quel degré de confiance peut être accordé aux outils existants et quels critères de base sont à prendre en compte.

2.2.1. Généralités

Mise en commun des ressources

Le développement actuel de l'informatique est particulièrement lié à celui des télécommunications.

Dans le domaine des réseaux d'ordinateurs, cet essor permet de bénéficier des facilités d'installations éloignées pour réaliser des applications informatiques de grande ampleur. D'un point de vue commercial, les logiciels et matériels déjà existants sont ainsi rentabilisés par une utilisation plus intensive. Cependant, seules les entreprises puissantes peuvent se permettre de développer leur propre réseau à partir d'installations antérieures ou en créant de toute pièce un nouveau réseau (SITA par exemple). Des réseaux administratifs se répandent donc assez rapidement. Pour les entreprises dont les moyens sont plus limités, il reste le recours à des sociétés de service ou de conseil ; mais il est souvent difficile d'adapter un réseau existant à de nouvelles applications dont la localisation ne correspond pas à l'architecture du réseau envisagé. Seul un réseau au maillage très fin peut permettre de résoudre ce genre de problème : c'est le cas du réseau téléphonique.

Les applications qui y sont développées sont encore très disparates ; elles concernent surtout des manipulations de fichiers (transfert, mise à jour) ou de bases de données, l'aide à la décision, la saisie des informations sur place ou la décentralisation de certaines gestions.

Cependant, quel que soit le type de réseau envisagé (excepté le cas d'un réseau mono-utilisateur fiable et sûr) et son mode d'utilisation, *le partage du réseau* entre des utilisateurs ou des organisations à vocation ou contraintes différentes, pose le problème de la protection des informations pour le maintien de la confidentialité.

Dans ce chapitre, nous allons tout d'abord faire le point sur les différents types de réseaux existants et sur leur aptitude à maintenir la confidentialité.

Types de réseaux

Les réseaux de terminaux dont l'utilisation était déjà bien connue des militaires, se sont répandus pour permettre le développement des applications commerciales, telles que réservation de places d'avion ou contrôles bancaires. Les terminaux, initialement rapprochés, sont alors situés à quelques kilomètres et souvent utilisés en dessous de leur capacité. Tous les échanges ont lieu entre *les terminaux* et *l'ordinateur central*. Le contrôle est donc nécessairement *centralisé* et c'est à l'unique système central de résoudre les problèmes de confidentialité. L'accès aux terminaux et aux lignes relève alors d'autres autorités. Le contrôle est d'autant plus facile à assurer qu'il n'y a aucun échange entre les terminaux, alors que l'ordinateur central lui, est utilisé simultanément à partir des terminaux. Les lignes de transmission ont donc une *structure étoilée* que l'on retrouve encore dans d'autres types de réseaux. [CYCLADES 1-2].

En résumé, les caractéristiques d'un réseau de terminaux sont :

. un seul centre de traitement où est centralisé le pouvoir de décision,

. des concentrateurs,

. des terminaux rattachés à ces concentrateurs.

Cependant, le terme de réseau est généralement associé à des configurations *multi-ordinateurs* .

. soit le centre de traitement est constitué de plusieurs machines inter-connectées localement,

. soit ces machines sont dispersées géographiquement.

Dans le second cas, la structure du réseau peut être plus complexe : plusieurs chemins sont possibles entre les centres de traitement et les terminaux, plusieurs systèmes d'exploitation, pas de contrôle central.

Certains réseaux sont spécialisés pour des applications commerciales, mais souvent dans ce cas le pouvoir de décision reste centralisé.

Un réseau non spécialisé est maintenant proposé dans de nombreux pays par les PTT qui ont souvent le monopole de la transmission. Des lignes à haute vitesse, de meilleur qualité que celles des lignes téléphoniques classiques, peuvent alors être utilisées. La sécurité des informations est cependant dans ce cas primordiale étant donnée la quantité d'usagers concernés et le type d'application généralement mise en oeuvre. Ce type de réseau et les réseaux utilisés par les administrations supposent donc qu'une certaine sécurité peut être assurée, afin que *le partage des lignes* soit "sûr".

Les transmissions entre les ordinateurs inter-connectés peuvent se faire suivant différents systèmes de commutation (ligne ou message). Le principe de la commutation de messages est le suivant : chaque message émis vers un destinataire du réseau emprunte un chemin qui passe par les centres intermédiaires. Des messages émis consécutivement à même destination, peuvent donc emprunter des chemins intermédiaires différents.

Ainsi, l'écoute des lignes (en d'autres points que celui d'émission ou de réception) ne permettrait pas nécessairement de reconstituer la totalité de la conversation échangée entre deux usagers. Une telle structure procure aussi une plus grande fiabilité et sécurité et une meilleure utilisation des lignes.

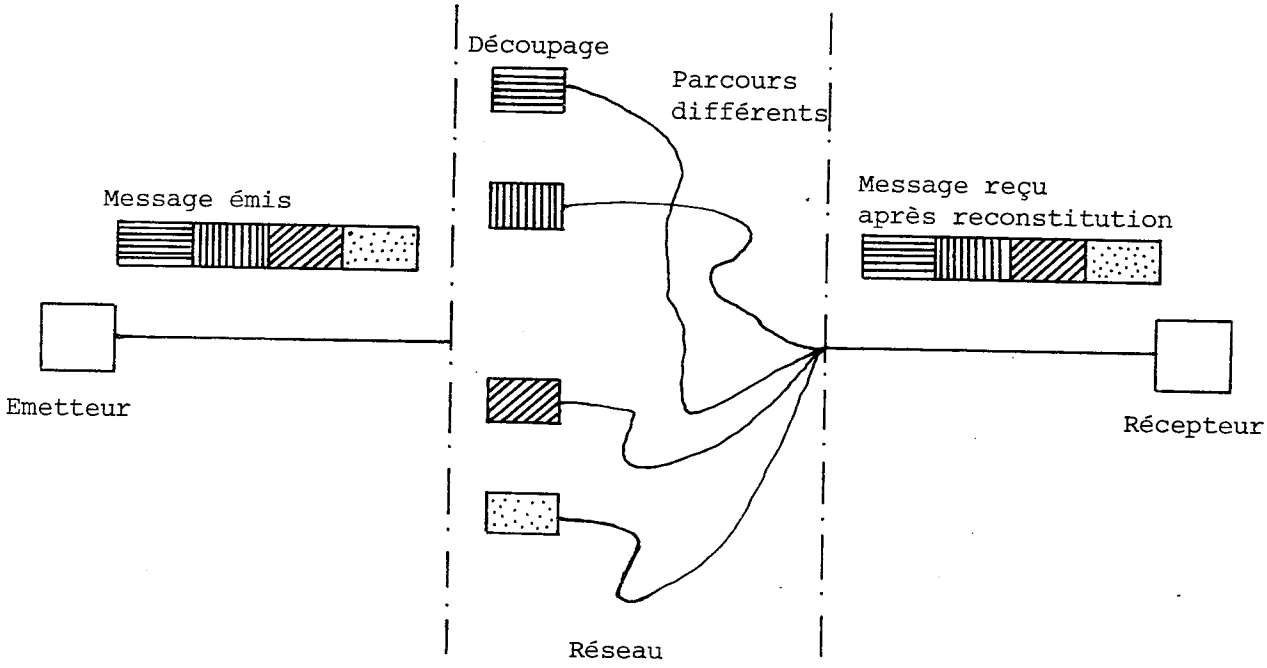


Figure 2.1

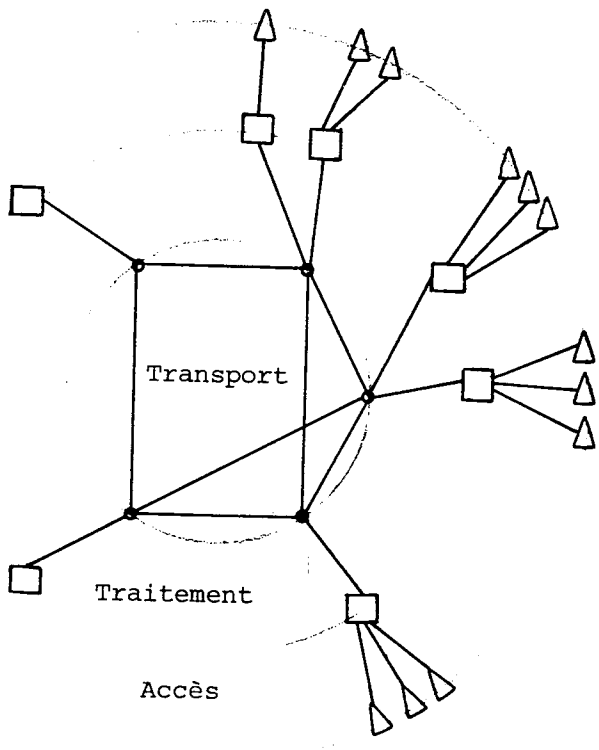


Figure 2.2 - Réseau à trois niveaux

Les fonctions de contrôle doivent donc s'adapter aussi au système de commutation.

La topologie des réseaux fait souvent intervenir un sous-réseau de communication pour isoler les fonctions de communication des problèmes de traitement (voir CYCLADES, figure 2.2).

Un réseau peut donc être considéré comme un ensemble de *sites* sur lesquels sont répartis les matériels et les logiciels. Grâce au réseau de transport, les *utilisateurs* du réseau peuvent échanger des informations de taille quelconque.

Contrôle

Quel que soit le type de réseau envisagé, de nombreux problèmes de mise en oeuvre et d'exploitation sont semblables. Les choix effectués dès la conception des réseaux permettent cependant d'en résoudre certains d'une façon plus élégante et souvent plus efficace aussi. Parmi ces problèmes on peut citer ici :

- la localisation des contrôles,
- l'hétérogénéité,
- la sécurité des lignes,
- l'adjonction de nouveaux centres de traitement.

Dans les applications usuelles, les objectifs de centralisation ou de décentralisation sont en général bien définis : par exemple, un magasin central contrôlant l'ensemble des succursales, ou un réseau de grandes banques autonomes les unes par rapport aux autres.

Dans le cas des réseaux spécialisés, le degré d'autonomie accordé aux centres de traitement peut donc être très variable.

En outre, les réseaux dits généraux doivent permettre que des choix éventuellement contradictoires soient réalisés suivant les applications. Il s'agit donc de prévoir différentes possibilités de fonctionnement au cours desquelles l'usager de définit lui-même en particulier *la localisation de ses contrôles et le degré de sécurité exigé.*

Du fait de *l'hétérogénéité* du matériel, en système d'exploitation, de tels réseaux se comportent comme une machine dont le fonctionnement d'ensemble n'est régi par aucun système global.

Dans tous les cas, les contraintes de sécurité doivent être respectées : en particulier, l'adjonction d'un nouveau centre ne doit pas la compromettre, les supports de transmission doivent être sûrs et disponibles. (Les considérations de fiabilité ne seront pas prises en compte ici).

Des règles précises doivent donc être définies en ce qui concerne ces conditions de raccordement et le degré de sécurité.

Les contrôles à répartir ou centraliser peuvent être classés en différentes catégories :

. le contrôle des communications et synchronisations : selon le type de commutation, les problèmes de séquencement et de réassemblage s'y ajoutent. Cette fonction peut être réalisée par les processus assurant la communication ou par les systèmes locaux ;

. le contrôle des interconnexions entre processus : est soit centralisé dans un noeud, soit réparti sur tous les noeuds, indépendamment de la topologie du réseau. Le plus souvent, chaque processus peut établir directement une connexion avec un autre processus quelconque du réseau. Le choix de l'itinéraire à suivre est également contrôlé par certains mécanismes permettant d'éviter la surcharge. De tels mécanismes sont efficacement incorporés au logiciel réparti sur chaque noeud.

. Le contrôle du trafic : c'est à la fois un contrôle de flux et un système de mesure réalisé par matériel ou logiciel réparti sur le réseau.

Les fonctions de contrôle du secret s'ajoutent donc à ces mécanismes et leur répartition en est aussi discutable. Il est cependant intéressant de noter que la topologie du réseau n'impose pas a priori une solution (centralisée/répartie).

Une solution de première urgence est donc d'introduire des contrôles aux différents niveaux défaillants actuellement, qui permettraient de remédier aux insuffisances :

. des systèmes non fiables,
. des lignes vulnérables,
. des contrôles d'accès aux installations (terminaux,..),
. des protocoles.

De simples codages au niveau des lignes, des protocoles de bout en bout des applications amélioreraient notablement le degré de sécurité du réseau.

Certaines questions restent donc posées :

. y a-t-il intégration ou séparation des mécanismes de protection et de communication ?

. comment un mécanisme de protection peut-il être lui-même protégé ?

Notion de protocole

La coopération de systèmes soulève, entre autres, les problèmes déjà connus par ailleurs de la communication et des reprises d'erreurs. L'architecture actuelle des réseaux ne comportant pas de système propre et la variété des applications envisagées reposent à nouveau ces questions.

Pour rendre un service efficace, le réseau doit donc assurer une bonne *communication des informations* entre ses différents composants. Leur hétérogénéité et leur disparité rendent nécessaire la définition de *standards* et de *protocoles* permettant l'adaptation aisée de tout nouveau composant (ce terme est employé ici au sens le plus général, il désigne le programme, l'utilisateur, le terminal, les procédures système, les sous-systèmes).

La spécification des protocoles doit être claire et permettre une réalisation aisée, quasi automatique. Ils se répartissent sur plusieurs niveaux, correspondant à des degrés d'abstraction différents. Des procédures de reprise, de contrôle, y sont associées. Ces protocoles gèrent le dialogue logique entre des composants. Dans le fonctionnement usuel des réseaux, leurs relations sont *hiérarchiques* (nous verrons ultérieurement la conséquence sur la confidentialité d'une telle structure) (figure 2.3).

Les protocoles permettent donc de définir les conditions dans lesquelles les échanges entre certains composants seront réalisés, comment ils seront établis et sous quelles conditions ils seront achevés : c'est déjà un début de standardisation (voir le protocole appareil virtuel de CYCLADES).

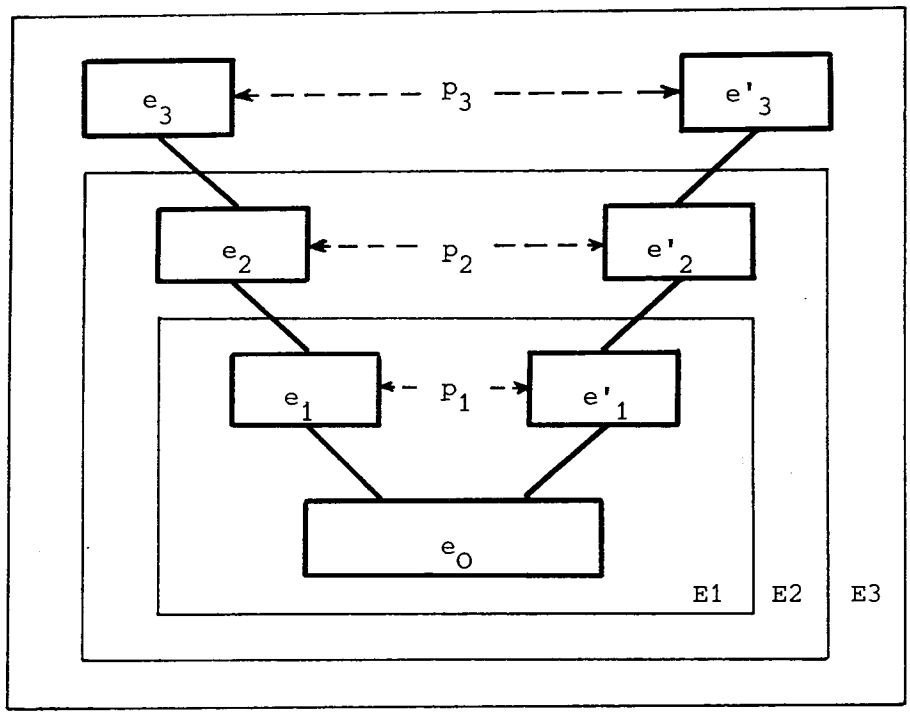


Figure 2.3 - Hiérarchie de protocoles

$E_i = \{e_i, e'_i, p_i\}$ = "machine" abstraite répartie

p_i = protocole

e_i, e'_i sont des composants sémantiquement identiques

On peut donc déjà dire ici que les choix des caractéristiques des protocoles influent sur la confidentialité, en ce sens que :

. le mode d'identification est plus ou moins élaboré,

. l'algorithme de contrôle prévoit une redondance ou un contrôle temporel,

. le mode de fonctionnement dépend d'un type de gestion des messages particulier, permet un accès multiusager ...

Quelques exemples de réseaux

Quelques exemples de réseaux sont présentés ici pour leur spécificité de structure de contrôle, ou simplement à cause des problèmes particuliers de secret auxquels ils se heurtent.

On peut aussi se reporter au réseau SATIN [LIPNER 75] qui aborde le problème de la sécurité tant dans l'architecture que dans l'implémentation des processus.

. Un réseau homogène : le réseau SOC (Système d'Ordinateurs Connectés)

c'est un projet conjoint d'IBM France, du CIRCE, du CEA, de l'IMAG, de l'Ecole des Mines de Fontainebleau, réalisé pour des systèmes IBM 360/370 sous O.S. [SOC 1-2].

La réalisation d'un *système réseau* facilite le contrôle du réseau, dans la mesure où celui-ci est concentré dans un seul système, éventuellement réparti. Il est cependant dépendant des systèmes de contrôle des ordinateurs du réseau sur lesquels il réside et dont il utilise les mécanismes. De plus, dans SOC, aucun privilège n'est accordé à un ordinateur particulier du réseau.

. Un réseau décentralisé international : celui de la SITA (fig. 2.4)
(Société Internationale de Télécommunication Aéronautique)

Ce réseau gère au niveau international les réservations de places d'avion. Il se heurte au manque de législation internationale concernant la confidentialité : chaque pays est maître des informations qui y transitent, mais rien n'assure que celles-ci seront acheminées sans erreur ou qu'elles parviendront à leur destinataire initialement prévu.

. Un réseau postal : TRANSPAC (figure 2.5)

Le réseau TRANSPAC doit bientôt desservir l'ensemble du territoire français (voir schéma). Il utilise la technique de la commutation par paquets expérimentée par l'ARPA (USA) et CIGALE (France) entre autres.

A chaque ensemble de commutateurs est associé un *point de contrôle local* qui s'occupe de la gestion des commutateurs (abonnés, routage). L'ensemble du réseau est lui-même muni d'un *centre de gestion* chargé de la surveillance et de l'exploitation générales.

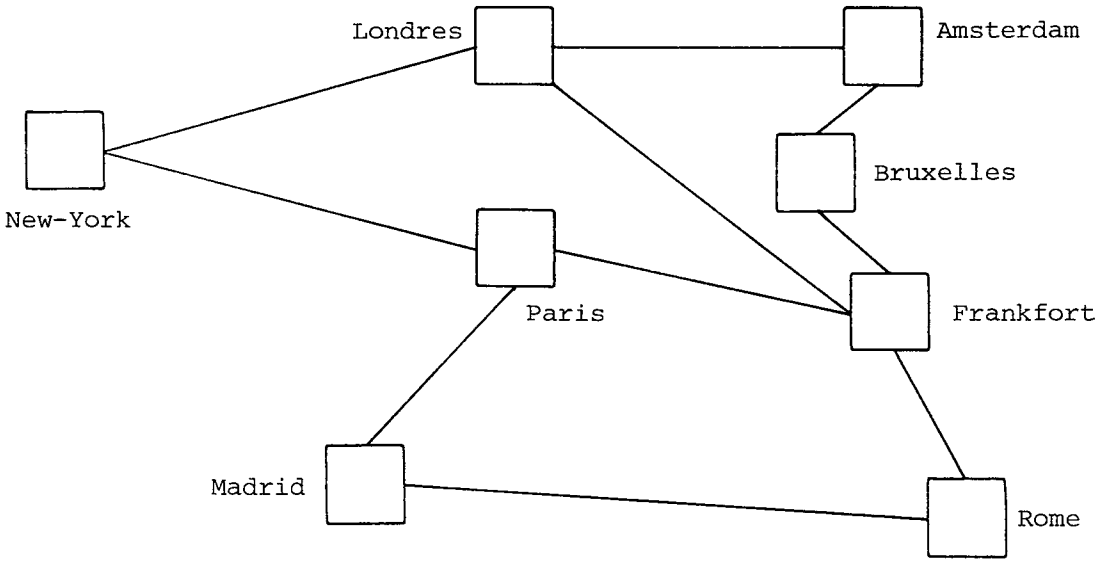


Figure 2.4 - Le réseau de la SITA

transpac en 1978 et 1980

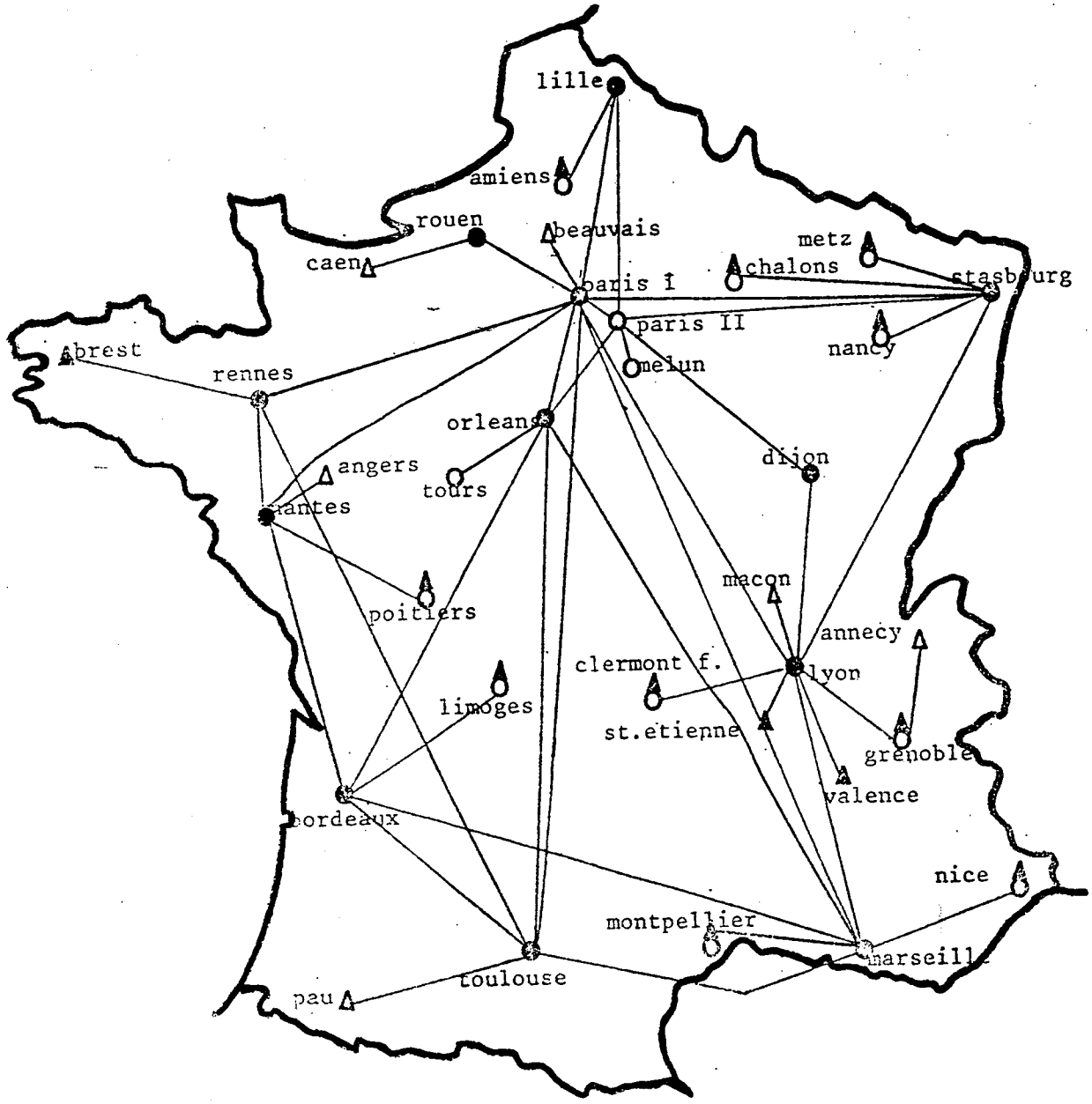


Figure 2.5.

78	80	
●	○	Commutateurs
▲	△	autres points d'accès pour terminaux asynchrones

Extrait de "Caractéristiques techniques d'utilisation des services TRANSPAC"
Direction Générale des Télécommunications, Novembre 1976

En résumé, on peut dire que les préoccupations de confidentialité n'ont pas été prises en compte lors de la conception des réseaux et de leur logiciel. Elles n'apparaissent qu'au moment des applications.

De même, les systèmes n'ont pas tenu compte des réseaux jusqu'à maintenant, mais on s'oriente actuellement vers une attitude plus conciliante qui consiste à les adapter aux fonctions des réseaux. Les réseaux y gagneraient beaucoup si un tel comportement prévalait en ce qui concerne le secret. En particulier, il est à souhaiter :

. des protocoles adaptés,

. des systèmes de contrôle réparti efficaces,

. des mécanismes de synchronisation et de communication propres.

La structure des réseaux à plusieurs niveaux, dissociant le transport des autres fonctions, peut être considérée comme un facteur positif de protection. Des dispositifs indépendants peuvent être aisément associés à ces niveaux.

Incidence du réseau sur l'environnement

L'environnement réseau ne pose pas de question typiquement nouvelle face au problème du secret. On peut cependant relever différents aspects, accentués par une telle configuration :

1) des réseaux, tels CYCLADES, présentent une séparation entre les deux niveaux : *traitement* ~~et~~ *communication*. A ces deux niveaux sont attachées des préoccupations de secret résolues par des méthodes souvent fort différentes. Les diverses solutions présentées dans la première partie restent applicables dans cet environnement. (*)

2) L'extension des systèmes *multi-usagers* (temps partagé ..) avait déjà préparé les systèmes à ce type de problème : importance accrue du contrôle d'accès aux ressources, prévention de l'interblocage, maintien de la cohérence, méthodes d'authentification, identification des usagers ..

3) La connexion de machines par un réseau offre un *nouveau point d'accès au système*, non plus direct, mais par l'intermédiaire d'un autre système, et étend donc sa vulnérabilité.

4) La multiplicité des *lignes de transmission* dont la surveillance globale ne peut être assurée, nécessite que l'utilisation de procédures de codage (ou encryptage) soit généralisée.

5) Les *défaillances* locales (noeud, système) risquent donc d'être plus répandues dans un environnement dont la modularité de fonctionnement n'est pas encore une réalité : altération du fonctionnement du reste du réseau à partir d'un simple incident local.

(*) Les solutions apportées au problème de secret peuvent être introduites
- au niveau de la communication, c'est-à-dire indépendamment du traitement, comme cela a été décrit dans les pages qui précèdent,
- au niveau du traitement : c'est ce qui sera décrit dans les exemples d'applications présentés ici.

Le maintien de la sécurité dans un environnement réparti peut être facilité lorsque quelques principes de base sont respectés. Il est intéressant de les rappeler ici, tout en notant bien que certains concernent essentiellement la fiabilité [POPEK 75].

1) *modularité de fonctionnement :*

une défaillance locale, qui affecte les utilisateurs locaux ne doit pas compromettre la sécurité de l'ensemble et conditionne alors les choix de répartition des catalogues, des copies ...

2) *adaptabilité*

des systèmes de contrôle à d'autres réseaux, afin d'éviter une remise en cause qui risque d'être périlleuse ;

3) *niveau de protection*

adapté au type de contrôle souhaité : plus le niveau est haut, plus il y a d'entités à contrôler ; un niveau basique (tel le codage) présente une efficacité maximale, mais peut allourdir l'ensemble des traitements ;

4) *sûreté de fonctionnement*

des systèmes participants, base de coopération entre systèmes.

2.2.2. Exemple : le réseau CYCLADES

2.2.2.1. Principe de la communication

CYCLADES est un réseau d'expérimentation des techniques d'exploitation d'un *réseau général*. Reliant de nombreux centres de recherche et universités, il leur permet de développer un logiciel de transport et des applications, sur un matériel et des systèmes *hétérogènes*. (Figure 2.6).

Quelques principes de base, déjà expérimentés par ARPA, ont été retenus :. commutation de messages,

. indépendance du réseau de communication et des centres de traitement,

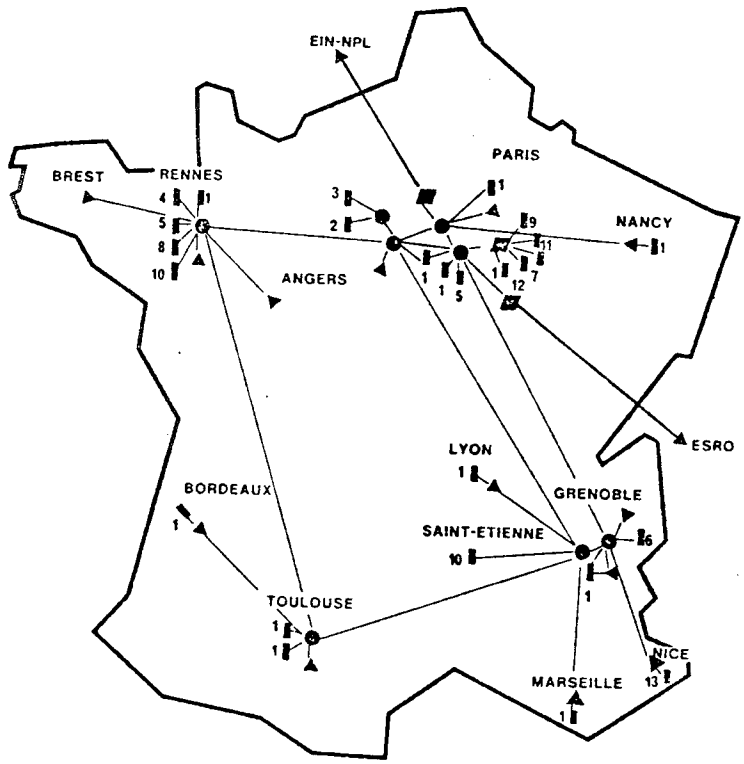
. hiérarchisation des protocoles.

- Le mode de commutation est la *commutation par paquets*.

- Le réseau de communication assure la transmission des datagrammes, mais ne se soucie d'aucun contrôle d'erreur, ni de la fragmentation ou réassemblage, il peut même refuser des paquets, les détruire en cas de saturation. Il est constitué de MITRA 15 CII reliés entre eux par des lignes téléphoniques louées.

- Le réseau de transport est constitué de stations de transport réparties sur chaque site, qui échangent entre elles des paquets confiés au réseau de communication ; ce sont elles qui s'occupent de fragmenter éventuellement les lettres pour constituer les paquets de taille limitée. Elles assurent aussi les contrôles : réémission éventuelle, numérotation des paquets, réassemblage, ..

cyclades



CONFIGURATION FIN 77

→ : *Liaison avec réseau externe*
— : *Liaison téléphonique*
▶ : *Concentrateur*
● : *Nœud*
▮ : *Serveur*
▰ : *Convertisseur d'interfaces*

1 - CII IRIS 80
2 - CII IRIS 45
3 - CII IRIS 55
4 - CII 10070
5 - CII MITRA 15
6 - IBM 360/67
7 - SIEMENS 7730
8 - HP 1200
9 - UNIVAC 1108
10 - T 1600
11 - PDP 10
12 - IBM 370/158
13 - CII IRIS 50

Figure 2.6.

• Les usagers, appelés abonnés, échangent des lettres ou des télégrammes, avec ou sans contrôle, par l'intermédiaire des stations de transport, selon un mécanisme d'adressage de leur choix ; la lettre est une unité d'information de taille quelconque, dont le contenu est ignoré des stations de transport.

Avant d'étudier plus en détail les protocoles du réseau CYCLADES, il est intéressant de rappeler quelques définitions du vocabulaire utilisé pour décrire le réseau CYCLADES :

calculateur participant : ordinateur relié au réseau CYCLADES,

réseau de communication ou sous réseau : ensemble de calculateurs (noeuds) reliés par des lignes de communication, assurant le transport des paquets,

station de transport : interface entre *le calculateur participant* et *le réseau de communication* ; assure le service de transport,

service : une ressource et ses moyens d'accès,

serveur : interface logicielle *service* - réseau,

abonné : programme utilisateur du service de transport,

utilisateur : un demandeur de *service* via le réseau (programme ou humain),

client : interface logicielle utilisateur - réseau,

porte : point d'accès au service de transport (pour un abonné),

flot : voie logique entre deux *portes*.

2.2.2.1.1. La station de transport

Implémentation

Chaque système hôte du réseau CYCLADES dispose d'une station de transport (S.T.) qui gère les communications entre les abonnés et le réseau de transport.

Chaque station doit assurer un service minimum, appelé "service de base" qui permet aux abonnés d'envoyer et de recevoir des lettres et des télégrammes sur des flots. Pour réaliser un tel service, la ST effectue :

. un multiplexage entre les flots,

. fragmentation et réassemblage de lettres à l'intérieur d'un flot.

D'autres mécanismes optionnels peuvent être proposés sur certains sites. Ils permettent :

. le contrôle d'erreur sur les lettres,

. le contrôle de flux sur les lettres,

. l'adressage réduit.

Logiciel [CYCLADES 5]

Les échanges d'informations avec la ST

Différentes solutions, fonction des propriétés de la machine sur laquelle elle est implémentée, sont proposées. Mais toutes supposent que l'usager est conscient de la valeur accordée à ses informations : c'est à lui de choisir *les méthodes de codage ou de dispersion* qui lui paraissent les plus efficaces.

Exemple :

l'abonné peut fournir un ou plusieurs tampons à la ST pour chaque flot. Il y recevra alors les informations qui lui sont destinées. En outre, il a la possibilité de demander que ces tampons soient partagés entre plusieurs flots. La gestion de ces tampons, moins nombreux dans ce second cas, est à la charge de la ST.

Les contextes

Aux portes et flots sont associés des contextes qui les représentent. Ils sont détruits à la fermeture. Une clé de protection est fournie à l'abonné à la création du contexte.

Les interfaces

C'est à ce niveau que les contrôles devraient être les plus importants. En fait, en ce qui concerne *l'interface abonné-station de transport*, seule une vérification de *clé de protection* est effectuée. *Les blocs de demande* doivent posséder cette clé pour être traités. N'importe quel utilisateur peut donc demander accès au transport. Les demandes impossibles à traiter sont *droppées* et actuellement aucune exploitation du fichier, où elles sont stockées, n'est envisagée. Il serait intéressant d'étudier ici les causes d'anomalies (erreurs/fraudes, conséquences).

Logiciel non protégé

D'importants catalogages sont construits et manipulés par la ST. De nombreuses tables décrivant les usagers, les connexions ou les services demandés, sont gérées par une tâche "ordinaire". Les protections sont donc celles du sous-système SYNCOP sous lequel elle est programmée (l'étude de SYNCOP est détaillée plus loin).

Faible contrôle d'accès

Les interfaces sont peu raffinés. Des fraudes, ou même simplement des erreurs, peuvent compromettre gravement la sécurité des données confiées à la ST. Les *boîtes aux lettres* sont en particulier très vulnérables (cf. § 2.3.3.).

Pas de protection en cours de transport

Les communications avec le réseau se font par l'intermédiaire d'un gérant de la ligne. Les cas de panne sont traités . Les mécanismes du système assurent la surveillance sur l'attribution des lignes. Sur les lignes, seuls les contrôles d'erreurs sont effectués (parités, ..). Une simple modification des tables de routage associées aux noeuds ou des paramètres d'une connexion, permettrait à un usager "espion" de recevoir des informations qui ne lui sont pas destinées. Une déformation intentionnelle au niveau du réseau de transport est réalisable par simple modification de logiciel, les logiciels du réseau n'étant pas vérifiés/certifiés régulièrement.

Proposition : utilisation optimale d'un logiciel peu sûr

Des remarques concernant quelques choix de principe de CYCLADES peuvent cependant être formulées :

. à la notion de PORTE, qui est en fait une "porte ouverte", accessible à tous, peut s'ajouter celle de PORTE ETROITE. Cela évitera d'encombrer inutilement un service par des demandes qui seront rejetées ultérieurement. Seul un nombre limité de demandes ayant des caractéristiques précises pourront être envoyées à la PORTE ETROITE. Cette porte bénéficie d'une surveillance accrue et le fonctionnement du service associé n'est pas perturbé par des demandes souvent injustifiées.

. Un FLOT PRIVE peut être défini entre deux usagers très exigeants. Un tel flot ne serait pas multiplexé. Une telle notion, exigeante en ressources, ne serait utilisable que dans des cas limités d'applications.

. Les deux notions de PORTE ETROITE et FLOT PRIVE reviennent en fait à la définition d'un *mode d'exploitation privilégié* de réseau, permettant la mise en oeuvre d'applications à haut degré de sécurité. D'autres dispositifs de protection (codage ..) s'y adjoignent.

Il nous semble cependant que les principes choisis dans CYCLADES ne sont pas en contraciction fondamentale avec les objectifs de la confidentialité. Il faudrait, pour que cela demeure vrai au niveau de la réalisation, que les méthodes de programmation, les choix d'implémentation, les techniques d'exploitation, soient cohérents avec ces principes.

2.2.2.1.2. Boîtes à lettres

Principe

Le système de boîtes à lettres (bal) permet à des tâches de se communiquer des informations. Une tâche ne peut recevoir de messages provenant d'autres tâches que si elle a *déclaré* auparavant une bal désignée par un identificateur (= 4 caract.). Toute tâche, qui connaît cet identificateur, peut être émettrice de messages. Une bal se comporte comme une file d'attente, gérée suivant le principe FIFO [BAL].

Protection

Seule la tâche propriétaire peut *supprimer* une bal. Les messages destinés à une bal sont situés dans une zone d'espace virtuel commune à toutes les tâches (accès en mode moniteur), c'est-à-dire qu'aucun usager ne peut y avoir accès (en lecture/écriture).

Hormis quelques noms réservés aux systèmes ou au temps partagé, tous les autres identificateurs sont autorisés sans règle d'emploi.

Pas de contrôle d'accès aux bal

Tout usager peut envoyer des messages :

. en cas d'absence de la bal destinataire, l'émetteur est prévenu par un code anomalie, mais peut effectuer autant de tentatives qu'il le désire ;

. si cette bal existe, le message y est déposé et sera reçu par le propriétaire de la bal (si celui-ci la relève). Le propriétaire ne peut éliminer des messages qui ne l'intéressent pas. Ainsi, un perturbateur peut-il saturer les boîtes à lettres des autres usagers dès qu'il connaît les identificateurs de bal.

Insuffisance des fonctions associées à la relève des bal

L'usager peut uniquement se mettre en attente de messages. Il ne dispose d'aucune fonction complémentaire, telle que par exemple :

. attente sur critère,

. attente multiple,

. accès conditionnel à une bal.

Protection

Les deux remarques précédentes contribuent à une critique de la protection de ces bal. De telles procédures de communication entre tâches ne sont donc utilisables que pour des utilisations très particulières (voire privilégiées). Leur emploi devra être contrôlé, ce qui n'est pas le cas actuellement (voir logiciel ST).

2.2.2.1.3. Propositions d'amélioration

Sans changer fondamentalement le fonctionnement de la station de transport, il est cependant envisageable de proposer quelques améliorations qui accroîtraient la sécurité de la communication sur le réseau.

La principale faille est la trop grande disponibilité des portes. Pour y remédier, on peut choisir une des méthodes suivantes (non exclusives) :

a) Définir un *allocateur de porte* chargé de choisir le nom de la porte que l'usager demandeur peut déclarer. Les demandes sont donc centralisées par l'intermédiaire de cet allocateur : les doubles déclarations sont ainsi évitées, un même numéro de porte qui est nouvellement libéré ne peut être immédiatement utilisé par un autre usager.

Outre ce fonctionnement *statique*, permettant d'allouer une porte pour la durée d'une communication donnée, on peut envisager un fonctionnement plus *dynamique* et réparti tel que :

. l'allocateur ne choisit plus un seul numéro de porte, mais plusieurs afin d'éviter de regrouper sous une même identification (l'en-tête des lettres contenant les numéros de portes émettrice et destinataire) l'ensemble des informations en circulation sur le réseau ;

. l'allocateur est capable de se synchroniser avec son homologue distant afin de répercuter localement les modifications des numéros de portes distants ;

. selon une fréquence à définir, l'allocateur peut choisir un nouvel ensemble de portes.

Dans le cas de ce second type de fonctionnement, les communications avec l'allocateur doivent être plus fréquentes et synchronisées, afin d'en accroître la sécurité.

La connaissance du ou des numéros de porte ainsi attribués n'est pas nécessaire à l'usager. Il lui suffit de connaître une simple référence, ou un numéro de porte virtuelle, qui sera ensuite traduit à l'aide de l'allocateur au niveau de la station de transport, lorsque les messages seront effectivement émis. Réciproquement, l'usager distant utilisera aussi une référence à cette porte (qui peut être identique à celle utilisée localement, ou non).

b) Contrôler les accès aux portes dans la station de transport. Pour cela, il suffit d'introduire un contrôle supplémentaire dans le logiciel ST existant. Une zone est déjà prévue pour communiquer une information de contrôle, de type clé. A chaque porte est donc associé un verrou. Seul l'usager qui en connaît la clé peut l'utiliser. De plus, il est possible de limiter le nombre d'usagers autorisés et de les identifier au préalable. La séquence de contrôle est donc la suivante :

. demande d'accès à la porte p_i, avec la clé c_k par l'usager u_j,

. contrôle de l'identité de u_j (appartenance à une liste des usagers autorisés, par exemple),

. contrôle de la clé ouvrant le verrou associé à p_i.

c) Contrôler les flots : en limitant le nombre des flots autorisés par une table contenant les seules combinaisons valables, par exemple. Chaque demande d'émission ou de réception sur un flot donne alors lieu à un contrôle élémentaire.

2.2.2.2. Facilités offertes pour le traitement

2.2.2.2.1. Protocoles

Parmi les protocoles, le protocole de transport joue le rôle de *protocole de base*, c'est-à-dire minimum pour toute application sur le réseau CYCLADES. D'autres protocoles ont cependant été définis pour permettre de nouvelles applications et le choix de règles de dialogue entre utilisateurs.

- Le protocole de transport [CYCLADES 3] :

Il fixe les règles de communication entre stations de transport.

. Chaque *paquet* qui circule sur le réseau contient, en plus des informations confiées par les abonnés dans les lettres ou télégrammes, une *commande* (avec ou sans paramètres) et l'identification des stations émettrices et destinataires. Les commandes existantes (dans la version 2) permettent essentiellement d'établir ou de supprimer des connexions.

. Pour réaliser un *système de contrôle*, il serait intéressant de disposer de nouvelles commandes, avec ou sans paramètres, pour surveiller les échanges entre stations.

. La réception d'une lettre est contrôlée par l'émission systématique d'une *lettre d'acquittement* du receveur concernant cette lettre, une fois que la totalité de la lettre est reconstituée (après réassemblage éventuel si la lettre était répartie dans plusieurs paquets).

. Un *télégramme*, dont le contenu occupe un seul paquet, n'est pas acquitté. En cas de perte, il n'y a aucun moyen de contrôle.

. L'espace de stockage associé à chaque noeud est géré par le logiciel de base du réseau. Seulement, pour une raison de contrôle de flux, une information de "*crédit*" circule en paramètre des lettres et des acquittements (nécessaire en l'absence de primitives simples de synchronisation). Elle permet de connaître la capacité restant disponible pour la réception. Aucun usager ne dispose donc d'une commande pour s'accaparer l'ensemble des ressources disponibles et interdire l'accès aux autres usagers.

. La *taille des lettres*, dont le maximum est fixé au moment de l'ouverture de la session, peut excéder la taille des paquets : un dispositif standard de *fragmentation* et *réassemblage* fonctionne donc au niveau de la station de transport (numérotation des paquets, référence de la lettre dans chaque fragment, temps d'attente limité, réémission automatique, ..). L'usager n'est donc pas maître de ce découpage. C'est à lui de fournir éventuellement des lettres de taille plus petite, après avoir effectué le découpage de son choix et de reconstituer ensuite son message, au cas où l'algorithme de fragmentation/réassemblage lui paraît insuffisant. Mais en ce cas, le contrôle doit être effectué au niveau de son application (problème de redondance avec le contrôle de la ST).

- Le protocole de connexion :

L'établissement d'une connexion entre deux usagers A et B suppose qu'au préalable ceux-ci se soient entendus sur les caractéristiques de fonctionnement de leur dialogue, à savoir :

. qui a l'initiative du dialogue,

. quel débit maximum d'information peut être accepté par chacun d'eux,

. quelle est la taille maximale des messages recevables.

En fait, les usagers A et B n'ont pas un rôle symétrique : A est un client du réseau et en ce sens sert d'interface entre l'utilisateur et le réseau. B est à l'écoute du réseau et est identifié par une porte b dont le nom est supposé connu des utilisateurs.

La connexion est donc réalisée à l'initiative de A (par l'intermédiaire de la ST) qui indique à B le nom de la porte sur laquelle il veut établir une connexion. Cependant, il demeure possible de modifier le nom de la porte une fois la connexion établie par une commande de basculement.

On peut donc remarquer que ce protocole utilisateur ne tient pas compte d'informations concernant le secret : aucune possibilité de contrôler les autres flots ouverts sur la même porte, aucune information sur le niveau de secret à maintenir, ...

• Le protocole appareil virtuel [CYCLADES 4] :

Il permet de caractériser d'une manière standard la plupart des terminaux existant actuellement (terminaux conversationnels ou lourds).

Il suffit donc de substituer à l'appareil réel à connecter un *appareil virtuel* ayant ses caractéristiques pour que la communication soit possible (figure 2.7).

Inversement, tout service mis à la disposition des utilisateurs doit donc savoir gérer un type unique d'appareil : l'appareil virtuel.

Le protocole peut être décomposé en plusieurs parties, qui ne seront pas détaillées ici :

. les règles de connexion (voir protocole de connexion),

. le contrôle du dialogue : fonction du mode de dialogue, il vise à autoriser les changements de sens de transmission, à échanger des fonctions et des états,

. le contrôle d'application : fixe le format des textes et leur découpage (en lignes, pages, ..),

. la négociation d'options : permet de définir les caractéristiques du terminal virtuel.

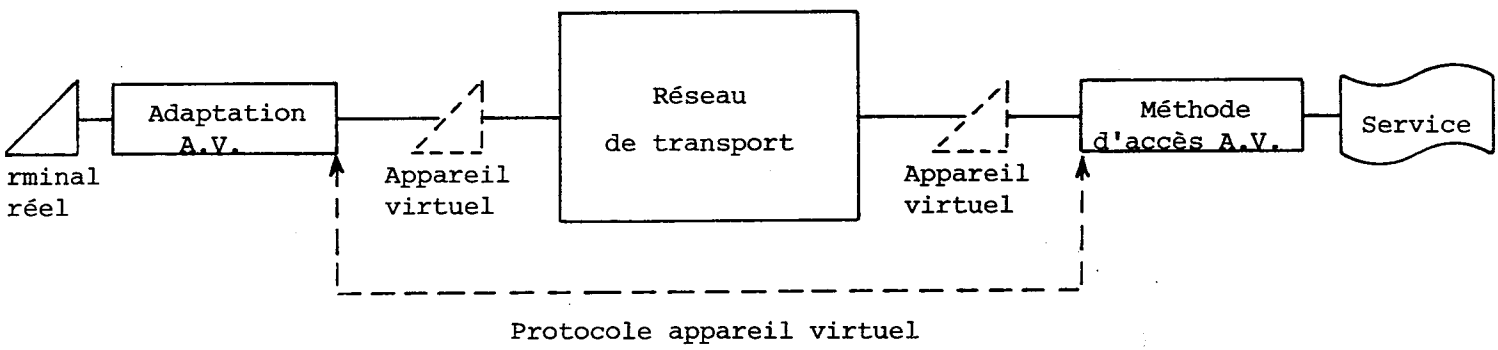

rminal
réel
Adaptation
A.V.
Appareil
virtuel
Réseau
de transport
Appareil
virtuel
Méthode
d'accès A.V.
Service
Protocole appareil virtuel

Figure 2.7.

2.2.2.2.2. L'accès au réseau

Tout usager du réseau peut être considéré soit comme un *client*, soit comme un *serveur*. Le dialogue est établi entre un client et un serveur lorsque ceux-ci respectent le même protocole.

L'accès au serveur est réalisé par la station de transport du même site ; le client doit utiliser lui-même une interface avec sa station de transport pour que la communication soit possible. Il a pour cela le choix entre plusieurs éventualités :

1) écrire sa propre interface : communication par boîtes à lettres par exemple, avec gestion des demandes et des réponses ;

2) utiliser une interface existante de macro-instructions (exemple : en langage FANNY, ou directement METASYMBOL ou PL/1)pour proposer un programme d'application ;

3) utiliser un logiciel adapté à cette station de transport : tel qu'un concentrateur. Il doit alors connaître à la fois les commandes de connexion au concentrateur et celles de connexion au serveur. En général, les communications avec les serveurs respectent le protocole d'appareil virtuel.

Le choix d'une de ces solutions est fonction aussi du degré d'interaction souhaité entre le client et le serveur (donc aussi fonction du serveur).

L'usager d'un service doit se faire connaître à une station de transport, c'est-à-dire choisir un numéro de porte sur laquelle il établira éventuellement un flot avec un ou plusieurs serveurs.

La gestion des flots est effectuée par la station de transport. Le choix du numéro de porte n'est pas contrôlé.

Le client doit pouvoir utiliser à tout moment les services du réseau. Ces services sont matérialisés par des serveurs identifiés par une adresse comprenant :

. un numéro de porte,

. un numéro de station de transport.

Certains serveurs multiusagers rendent simultanément un service analogue à plusieurs usagers en regroupant leur demande sur une même porte.

A des portes différentes correspondent en général des services différents.

Chaque client peut donc ouvrir un ou plusieurs flots sur une même porte, si le serveur le supporte, pour utiliser un même service de plusieurs manières simultanément. Le cas le plus simple reste celui pour lequel un serveur ne sert qu'un client à la fois pendant toute la durée du service.

Selon le service, différents types de problèmes peuvent se poser si plusieurs usagers sont autorisés simultanément à utiliser ce service. Les serveurs bases de données doivent assurer *l'intégrité* des bases gérées et la *cohérence* entre les différents fragments de ces bases. Dans le cas général, un serveur doit être capable d'assurer la *confidentialité* des informations mises en cause. A l'intérieur d'un même service, le cloisonnement entre usagers doit être suffisamment étanche pour que l'on puisse s'en assurer.

Il est à noter cependant que pour des applications bien particulières, un degré de protection supplémentaire serait assuré si un usager avait les moyens de travailler en exclusion mutuelle sur le réseau, c'est-à-dire un ou plusieurs serveurs et stations de transport, par exemple. Un tel dispositif aurait des conséquences de blocage/synchronisation que nous n'envisagerons pas ici : cela reviendrait alors à considérer le réseau comme une ressource préemptive ; mais vue la taille de son environnement, une telle solution est peu réaliste.

Les services offerts par le réseau de transport :

La station de transport assure l'établissement de flots entre abonnés distants. Cet établissement est conditionné par la réservation de ressources auprès de la station de transport. Elle assure aussi la suppression d'un flot et l'échange des informations. Elle prend en compte ces demandes pour obtenir l'accord des deux participants au flot.

Par contre, aucun service de *gestion des abonnés* n'existe : les numéros de porte sont choisis à l'amiable et il n'y a en général pas de conflit. Mais une telle situation n'est pas acceptable dans un réseau en exploitation permanente.

2.2.2.3. Un exemple d'outil réseau facilitant la programmation d'applications réseau : SYNCOP

Objectifs

SYNCOP est présenté comme un Sous-Système Normalisé de Commutation de Processus pour la télé-informatique et les réseaux d'ordinateurs [SYNCOP].

Ces deux types d'applications mettent en oeuvre de nombreux utilisateurs ou abonnés désirant contrôler simultanément l'exécution de certains programmes ou processus. Les systèmes d'exploitation existants n'apportent en général pas de solution à ces problèmes. Ils s'avèrent surtout inefficaces dans les domaines suivants :

- . la synchronisation,
- . la commutation,
- . le degré de multiprogrammation.

En particulier sur les grosses machines, une lourdeur de ces mécanismes (lent + coûteux) est ressentie.

La solution immédiate qui consisterait à modifier ces systèmes offre de sérieuses difficultés en général :

- . pour la création de mécanismes de synchronisation,
- . pour le transfert de zones de mémoire entre tâches,
- . pour le nombre de tâches.

SYNCOP est donc conçu comme un *sous-système* sur un système existant, appelé système hôte, et permet la réalisation des principales applications télé-informatiques (ou réseau), ou au moins donne à l'utilisateur des moyens simples pour adapter le sous-système à ses besoins précis.

Il nous intéresse particulièrement ici en tant qu'outil d'implémentation du logiciel du réseau CYCLADES (sur IRIS 80 sous SIRIS 8 et IBM 360/67 sous CP 67).

Principes de base

Ces principes répondent aux objectifs posés dans le paragraphe précédent, à savoir :

- réaliser la commutation d'un grande nombre de processus,
- prévoir la communication entre ces processus,
- compléter les mécanismes de synchronisation,
- améliorer la gestion des horloges pour la synchronisation.

Le sous-système réalisé peut, en outre, s'adapter aux applications de la téléinformatique et propose à l'utilisateur une interface standardisée. Différentes versions sont disponibles : multimachine, extensible.

Il reste à regretter que l'objectif de protection n'ait pas été suffisamment pris en compte, en particulier face aux risques spécifiques dûs à la communication d'informations entre processus.

Les contrôles existants, quoiqu'encore insuffisants, pénalisent tant les utilisateurs qu'elle les conforte dans leur attitude de laisser pour compte ces mécanismes.

Ce système suppose donc que les usagers soient respectueux des règles de programmation et n'outrepassent pas leurs droits (sauf dans quelques cas limites où les processus osés ou trop gourmands sont purement et simplement supprimés par le système !).

La plupart des remarques d'utilisation concernent des choix d'implémentation.

L'insuffisance des mécansimes de contrôle demeure l'objection fondamentale à son utilisation pour des applications réseau confidentielles :

. le contrôle de propriété des informations est trop souple,

. le mécanisme d'allocation de zones de mémoire de grande taille repose sur le système hôte (cf. version SIRIS 8) et ne comporte qu'une vérification sommaire (une table contient l'adresse de la zone, sa taille et le nom du propriétaire),

. les contrôles sont facultatifs,

. aucune protection particulière des tables du sous-système (utilisées pour gérer les files, la mémoire centrale, la synchronisation ou les processus),

. pas de nettoyage des zones en fin d'utilisation.

Une contrainte importante persiste pour la réalisation de ce sous-système : *tous les systèmes hôtes doivent posséder une possibilité d'attente multiple et d'horloge en tant que primitive utilisateur.*

En fait donc, SYNCOP n'apporte pas de mécanismes nouveaux, mais propose une facilité d'utilisation : c'est une interface "intelligente" entre le système et un utilisateur désireux d'exploiter les possibilités de la téléinformatique et des réseaux.

2.2.2.2.4. Exemple d'accès : le concentrateur multiconnexions

Présentation

Un concentrateur de terminaux est un *logiciel capable de gérer plusieurs terminaux légers* (télétype, écran de visualisation ..) et leur connexion à des serveurs interactifs présents dans le réseau (ici CYCLADES). (figure 2.8).

Il regroupe ainsi plusieurs abonnés-clients et leur permet d'établir, d'exploiter et de supprimer des liaisons avec un ou plusieurs serveurs. La particularité de ce concentrateur est de permettre à un usager de terminal quelconque, travaillant sous le système de temps partagé SIRIS 8, d'exploiter *simultanément* plusieurs connexions et non pas successivement comme c'est le cas dans les concentrateurs usuels. [CMC]

Il présente donc les avantages suivants :

. pas de monopole d'un terminal pour une connexion,

. exploitation immédiate des résultats des échanges avec plusieurs serveurs (synthèse rapide) à partir d'un seul terminal,

. établissement de connexions multiples à un même serveur, à partir d'un seul terminal (si ce serveur le supporte),

. parallélisme du travail avec des serveurs très interactifs (sans blocage).

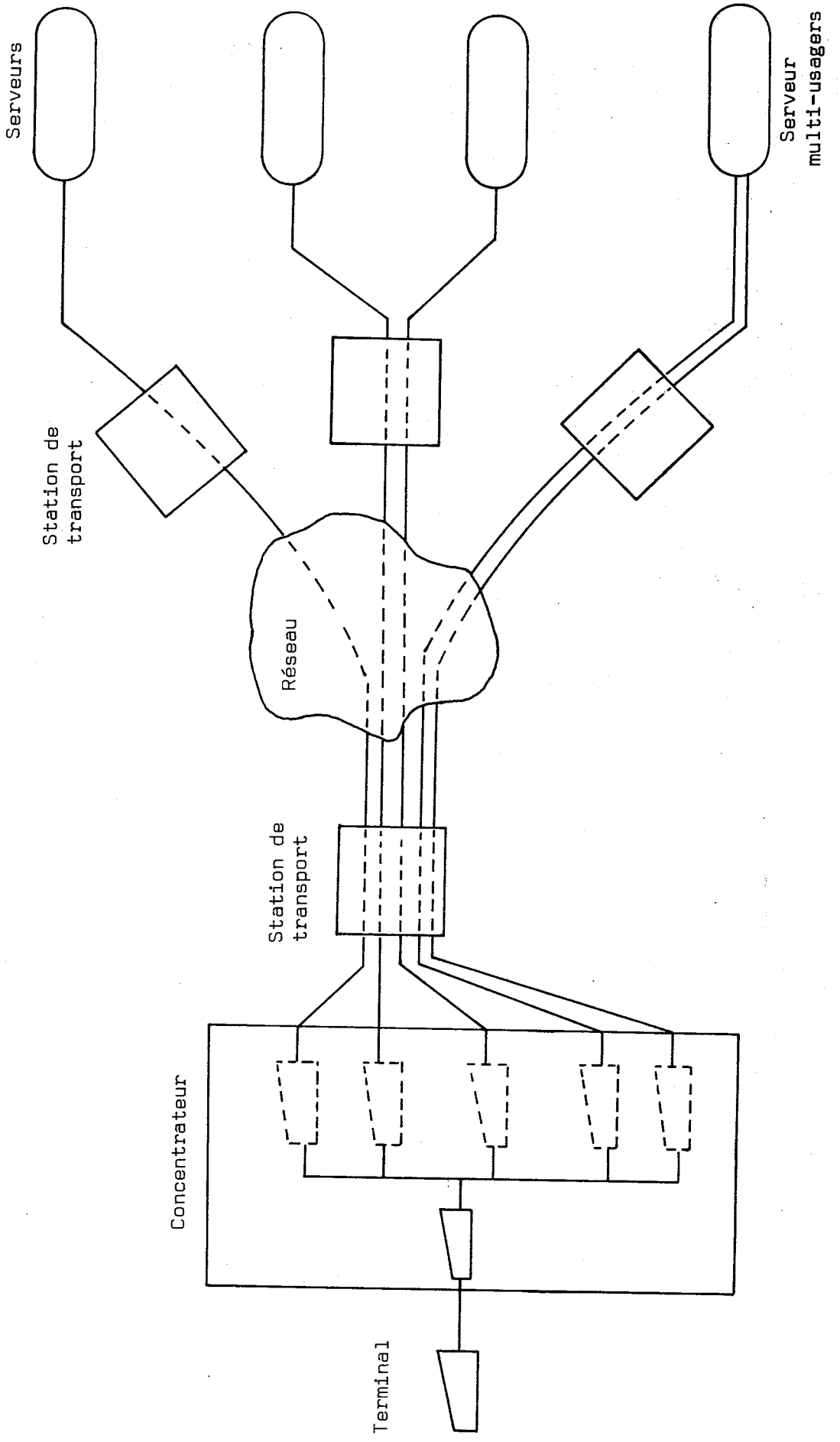


Figure 2.8 Multi-connexions d'un terminal par le concentrateur

Spécificité des connexions multiples

L'exploitation simultanée des connexions avec plusieurs serveurs élargissant le champ d'utilisation des serveurs du réseau, demande une extension des commandes de manipulation des connexions, à savoir :

. connaissance à tout instant du contexte de travail, c'est-à-dire de l'ensemble des connexions établies avec les serveurs (actives ou non actives) ;

. spécification d'un contexte actif : passage du mode connecté local au mode connecté réseau avec exploitation de plusieurs connexions simultanées ;

. identification des transactions ;

. éventuellement, orientation de certaines éditions vers un support (fichier) secondaire, afin de bénéficier du fonctionnement interactif d'autres serveurs moins bavards ;

. coopération des serveurs.

Identification des connexions

A chaque serveur est attribué un *nom de serveur* (abréviation discriminante). Il permet de différencier les messages échangés par différentes connexions. Pour éviter les confusions, dans le cas de connexions multiples à un même serveur, à ce nom est associé un *rang de connexion*. Cette identification, appelée *nom de connexion*, est communiquée à l'usager du terminal dès que la demande de connexion est satisfaite. Elle lui servira par la suite à référencer cette connexion dans certaines commandes : en particulier la commande d'activation (RESTART) et de déconnexion (DISCONNECT).

Selon le mode de fonctionnement, cette identification peut figurer au début de chaque message reçu par l'usager du terminal pour le compte de cette connexion. Une autre possibilité de reconnaissance des transactions propres à chaque connexion, dans le cas où deux connexions sont actives, consiste à partager le support (écran ou papier) en deux parties : la partie gauche étant réservée à la première connexion, la partie droite à la seconde. D'autres solutions sont également intéressantes, mais il est important que cette différenciation des transactions soit claire pour que ce concentrateur multiconnexions soit facile d'emploi. La solution actuelle qui consiste à identifier par *trois* caractères en tête de ligne n'apporte pas la clarté suffisante.

Principes de fonctionnement

Le terminal est représenté par un processus "appareil réel". Pour chaque connexion terminal ↔ serveur, le concentrateur simule le terminal par un processus *"appareil virtuel"*. Ces processus sont multiplexés sur le processus "appareil réel". Un processus "logger" contrôle l'ensemble. Ces trois types de processus sont coordonnés par le système normalisé de commutation de processus SYNCOP [SYNCOP].

Une console opérateur, gérée comme un terminal ordinaire, bénéficie de deux commandes supplémentaires permettant

. de supprimer les autres connexions au concentrateur,

. d'arrêter le fonctionnement du concentrateur.

La gestion des terminaux est laissée au système de temps partagé : l'accès au concentrateur est donc autorisé à des terminaux banalisés (indépendamment des caractéristiques propres de chaque type de terminal léger). (Fig.9) Il n'a donc aucun souci d'adaptation. La communication entre tâches temps partagé et tâches batch utilise l'interface boîte à lettres SIRIS 8 [BAL].

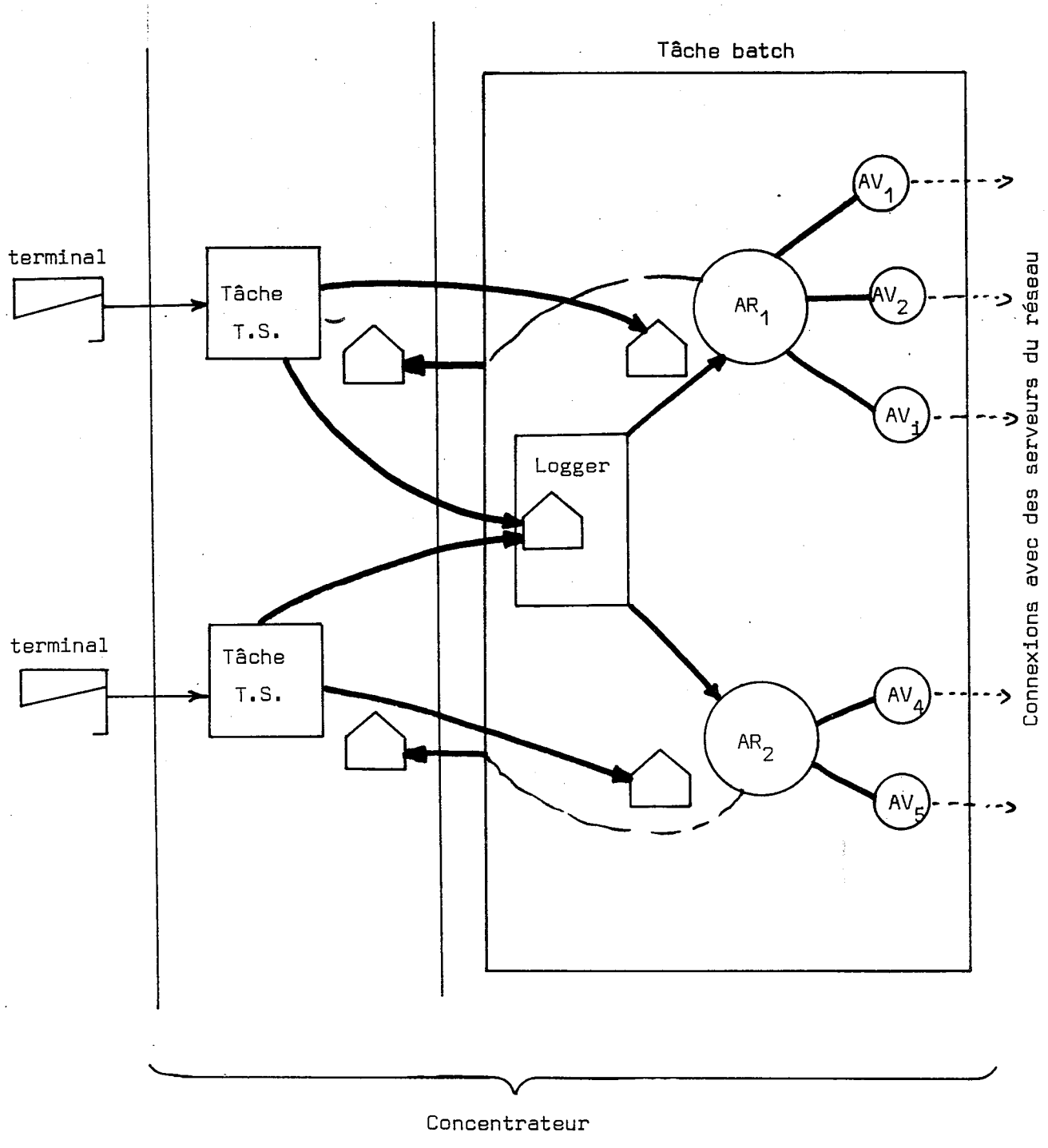


Figure 2.9 - Architecture du concentrateur

Gestion des connexions avec les serveurs

Traitement des demandes de connexion

Réponse négative dans les cas suivants :

. le nombre maximum de connexions autorisées est atteint,

. le serveur n'est pas décrit dans la "table des serveurs" associée au concentrateur,

. la demande de connexion échoue (serveur absent, ST indisponible ..)

Réponse positive :

la demande de connexion avec le serveur a réussi. Un processus "appareil virtuel" est créé et son contexte est initialisé (caractéristiques du serveur, nom de connexion). La procédure AVTRANS est appelée pour gérer les transmissions.

Exploitation d'une ou plusieurs connexions

La procédure AVTRANS associée à "l'appareil virtuel" (A.V) gère la liaison suivant le protocole appareil virtuel CYCLADES pour une connexion à l'alternat. Les seuls messages de contrôle traités dans certaines commandes sont les suivants :

your turn : une fois la connexion établie, c'est le serveur qui "a la main" qui est autorisé à émettre des messages. La commande YTRN permet"d'envoyer la main" à un serveur (autorisé à émettre)

attention : envoi d'une attention de niveau 1 ou 2

nego : envoi d'une priorité.

Le texte des messages est conservé en attente dans une file associée à l'AV. En fonction du contexte de l'activation, ces files sont vidées sur le terminal utilisateur (avec en en-tête le nom de connexion ou sur une portion de support).

Rupture accidentelle ou intentionnelle d'une connexion

Toute rupture est signalée au terminal utilisateur, qu'elle soit accidentelle ou intentionnelle.

Dans le cas où l'unique connexion active est rompue accidentellement, l'utilisateur doit se remettre en mode connecté local pour demander éventuellement une nouvelle connexion.

Les tables descriptives des connexions sont mises à jour. Un rang de connexion n'est pas réutilisable après une déconnexion.

Utilisation

Le concentrateur multiconnexions constitue une première approche du problème de coopération entre serveurs. Ses mécanismes permettent la confrontation simultanée des résultats des échanges effectués pour le compte de différentes connexions.

Dans le cas où le serveur la supporte, la multiconnexion à un même service offre des perspectives nouvelles, en particulier pour l'exploitation rapide de serveurs type base de données.

Par ailleurs, il peut évidemment être utilisé pour la mise au point de serveurs : il permet de demander plusieurs connexions à partir d'un seul terminal.

Un tel outil peut s'insérer dans une architecture d'application répartie base de données. Il est alors intéressant d'utiliser une méthode d'accès programme : c'est-à-dire qu'il faut définir une interface standard permettant :

. de formuler les commandes standard du concentrateur,

. de manipuler des éléments d'information en vue de la coopération,

. de condenser certaines commandes en véritables fonctions d'accès.

Les commandes spécifiques de manipulation d'informations font actuellement défaut pour qu'une telle coopération soit possible. Le stockage temporaire de quelques informations devrait y être associé.

2.2.2.2.5. Proposition d'utilisation plus sûre de la station de transport

Soit donc le cas usuel d'un *transfert d'informations* sur le réseau CYCLADES. La protection peut être assurée à différents niveaux :

a) modelage des informations à transférer : par codage, restructuration de fichiers, dispersion sur plusieurs supports, découpage en plusieurs fichiers ;

b) un service de transfert de fichier protégé : partage des émissions entre plusieurs flots, utilisation dynamique des portes ;

c) la station de transport effectue les contrôles (clés d'accès aux portes, table des flots autorisés) ;

d) CIGALE : éviter que de fausses identifications ne puissent circuler sur les lignes de transmission en ajoutant un mécanisme d'identification des lignes interne à CIGALE.

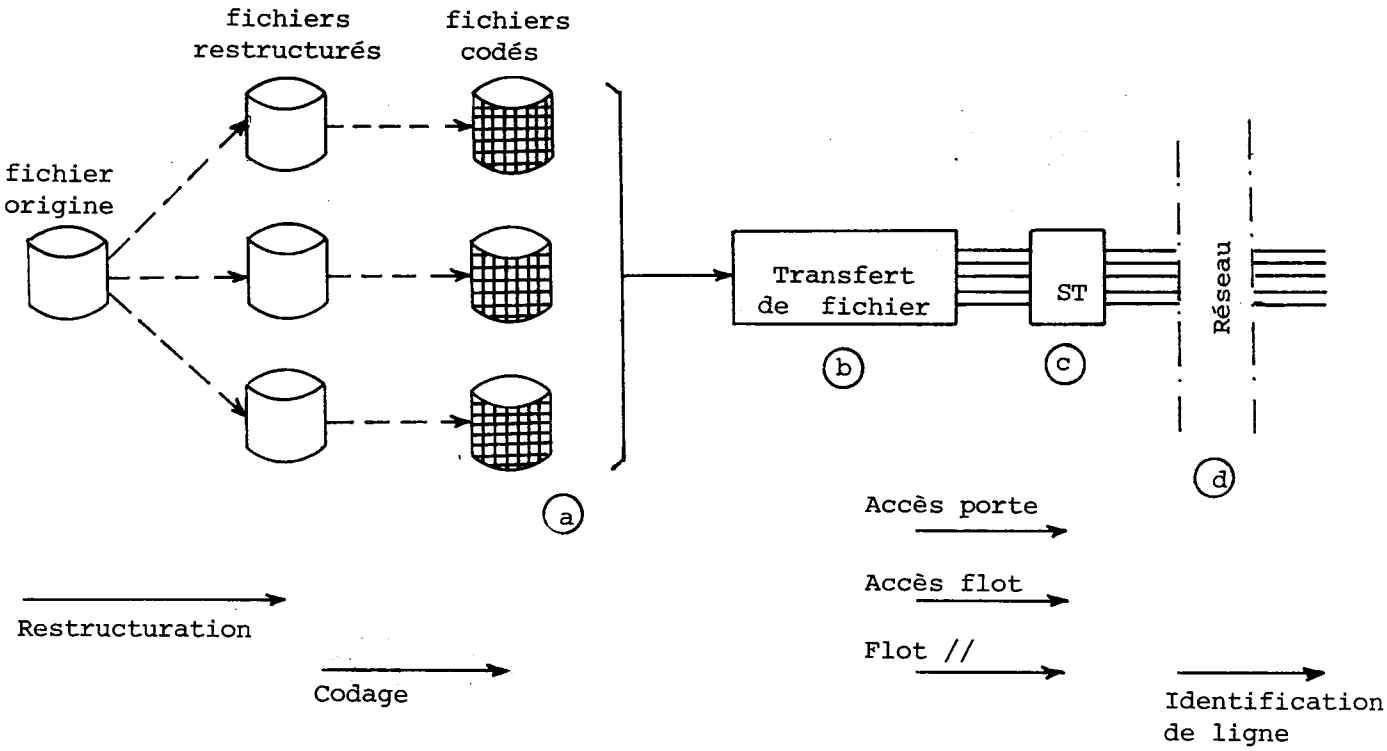


Figure 2.10.

Dans cet exemple, nous supposons que ni CIGALE ni la ST ne sont modifiées dans le sens d'un accroissement de la sécurité. C'est donc à l'usager de gérer lui-même ses informations afin d'obtenir un maximum de protection.

Les principes élémentaires d'utilisation de la ST à respecter dans la perspective d'un accroissement de la confidentialité, sont donc les suivants :

1) disperser les informations sensibles sur plusieurs flots,
2) éviter le partage d'une porte entre plusieurs flots,
3) ne pas faire circuler les informations de sécurité et de contrôle.

Plus généralement, quel que soit le service envisagé, le comportement du serveur le représentant devrait respecter ces principes ; ce qui correspond au fonctionnement suivant :

. un serveur doit associer une porte différente à chaque nouvel usager. Le numéro de cette porte étant déterminé au préalable (hors réseau pour respecter le principe 3). La prise de contact se fait sur une porte spéciale, dite *"porte de contact"* partagée entre tous les demandeurs. C'est sur cette porte que sont filtrées les demandes selon un protocole propre au serveur (identification, authentification).

. Pour transmettre des informations sensibles, il faut adapter le nombre de flots et la durée de vie de chaque flot à la quantité d'information à transmettre et à la rapidité du transfert (en fonction du trafic et de la distance à parcourir).

. Quel que soit l'algorithme choisi pour le mode de numérotation des portes et le rythme de basculement d'une porte à une autre, à aucun moment il ne doit en être fait mention sur le réseau. Les informations de sécurité sont supposées avoir fait l'objet d'une convention préalable (hors réseau).

. La protection des informations sensibles ne peut être totalement garantie sur le réseau. Il importe donc de les "dessensibiliser" avant de les transmettre (par codage, restructuration, dispersion, par exemple).

2.2.3. Une nouvelle conception des applications

Une application dite répartie concerne des informations physiquement éloignées et gérées éventuellement par des logiciels hétérogènes, eux-mêmes répartis. On a donc à prendre en compte deux caractères spécifiques de la répartition :

. le délai de transmission des informations n'est pas négligeable,

. les informations sont gérées par des autorités différentes.

2.2.3.1. Critère de choix d'une architecture

L'architecture choisie doit permettre un fonctionnement optimal, c'est-à-dire :

. prise en compte de la totalité des informations concernées,

. temps de réponse acceptable,

. limitation de la charge indue : encombrement du système, autres applications possibles et non perturbées,

. sécurité.

Il ne s'agit pas de chercher à satisfaire uniquement ce dernier point, au détriment peut-être de tous les autres, mais d'estimer quels sont les aspects privilégiés sur lesquels la sécurité a une incidence.

Nous allons essayer ici d'envisager la sécurité de l'application répartie à partir des deux objectifs de contrôle et localisation, qui semblent fondamentaux et spécifiques de toute architecture d'application répartie.

2.2.3.2. Le contrôle

Les contrôles existant, tant sur les SGBD que sur les systèmes, restent insuffisants, par exemple, pour ce qui est du contrôle du parallélisme d'exécution ou de la cohérence entre bases réparties. Cette absence de contrôle "d'ensemble" peut être comblée par la mise en place sur chaque centre de répartition d'un contrôleur ayant tout pouvoir sur les modules locaux de l'application. Cependant, quelle que soit l'emprise des contrôleurs, l'autorité risque d'être pesante pour l'ensemble des usagers. Des solutions intermédiaires plus supportables doivent donc être proposées.

L'intégrité

Il s'agit d'un contrôle très particulier concernant le contenu de certaines informations : toute modification d'information ne peut être réalisée que si les contraintes d'intégrité sont respectées.

Question : le caractère réparti de l'application fait-il intervenir de nouvelles contraintes ?

La coopération de plusieurs bases permet de regrouper les traitements d'informations gérées initialement par des SGBD différents (et appartenant à des bases différentes).

Les redondances d'information en particulier sont alors plus fréquentes. Si l'on désire que le contenu de l'ensemble des bases qui veulent coopérer soit cohérent, il faut :

. au moment de la mise en commun des bases, vérifier la cohérence : rejet des données incohérentes ou modification après accord de l'administrateur de la base,

. au cours de la coopération, maintenir la cohérence.

La répartition des bases sur un réseau rend plus difficile cette seconde procédure : les délais de vérification doivent être pris en compte.

Cependant, comme par hypothèse les bases sont supposées exister indépendamment les unes des autres avec une administration différente, le nombre de telles contraintes est limité (ce qui n'est pas le cas de bases initialement réparties). De plus, le risque de fraude y est plus important (sur un réseau) : les vérifications sont donc plus urgentes qu'ailleurs.

Exemple de contrôles :

ils sont associés aux différentes contraintes :

a. le numéro de sécurité sociale comporte 13 chiffres, dont le premier est soit 1, soit 2 ;

b. lorsque sexe = masculin, le premier chiffre du numéro de sécurité sociale est 1 ;

lorsque sexe = féminin, le premier chiffre du numéro de sécurité sociale est 2 ;

c. lorsqu'il y a changement de situation d'un employé (création d'un nouvel employé ou modification de son emploi), il faut modifier la valeur du nombre d'employés correspondant à son ancien employeur et à son nouvel employeur

SALARIES SEXE NUMERO-SS EMPLOI	dans la même base (B1)
EMPLOYEUR NB-EMPLOYE ADRESSE	autre base (B2)

Toutes les modifications de EMPLOI dans B1 doivent être répercutées sur NB-EMPLOYE dans B2 le plus rapidement possible.

Si l'on veut garantir qu'à tout moment B1 et B2 sont cohérentes, il faut bloquer les accès à B2 pendant les modifications de EMPLOI dans B1 jusqu'à ce qu'elles soient répercutées sur NB-EMPLOYE dans B2.

Des commandes paramétrées de blocage/déblocage sur un champ donné doivent donc être prévues dans une application multibases afin d'éviter que des blocages prolongés systématiques de l'ensemble des bases ne soient demandés par tout usager du modèle de coopération. Il est donc dommage que certains SGBD ne supportent pas de telles commandes : l'accès à de telles bases est alors bloqué pendant tout le travail multibase.

Sur un réseau

Certains contrôles ne sont pas réalisables à distance, dans la mesure où les systèmes du réseau ne disposent pas tous des mêmes possibilités. Ils sont donc relégués à des contrôleurs locaux entre lesquels ont lieu des échanges privilégiés. Les autorisations sonc donc délivrées, soit par *l'autorité locale*, soit par *l'autorité la plus proche*.

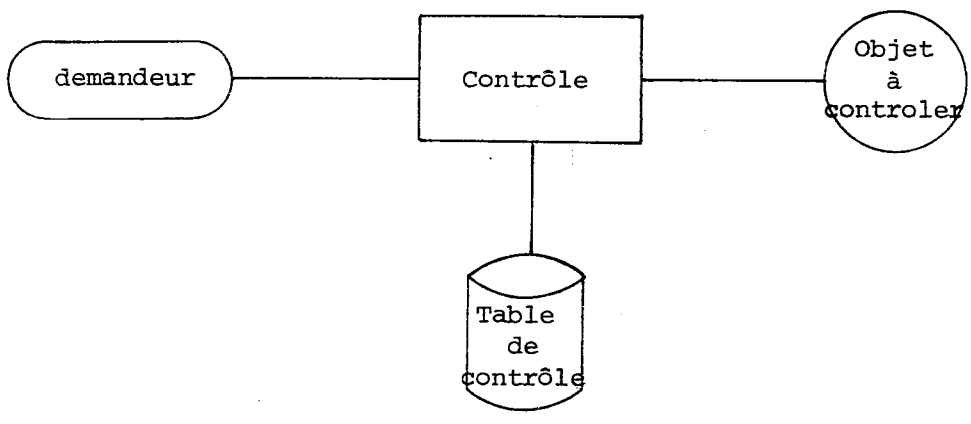


Figure 2.11

Dans le cas où les responsabilités sont partagées entre plusieurs contrôleurs *la communication des informations* nécessaires au contrôle doit être sûre (fiabilité + sécurité). Si le réseau ne peut l'assurer, tout contrôle sera inefficace.

Il peut alors être intéressant de disposer sur le réseau de plusieurs canaux d'échange, ou de plusieurs réseaux logiques ; les informations de contrôle y bénéficient alors d'une sorte de priorité sur les autres.

Dans le cas de CYCLADES, une telle solution ne peut être envisagée : lettres et télégrammes utilisent les mêmes lignes de transmission et aucun message ne peut être acheminé en priorité.

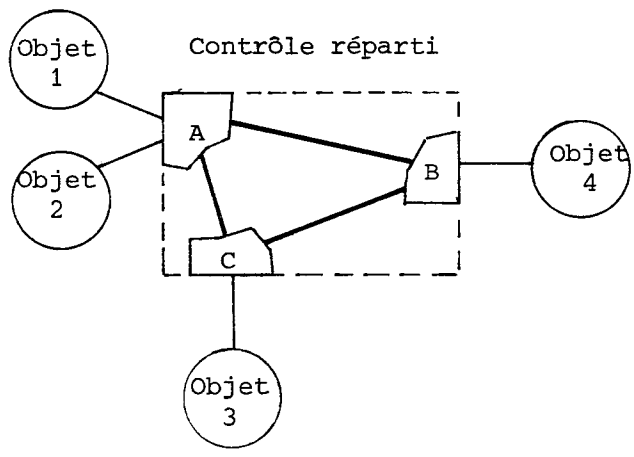


Figure 2.12

Si des moyens matériels importants peuvent y être consacrés, *un centre de contrôle* indépendant peut être créé : il centralise toutes les fonctions de contrôle du réseau, mais ne permet pas d'adjonction de fonctions nouvelles mieux adaptées à des applications particulières. La duplication du matériel permet d'augmenter aussi la fiabilité d'un tel centre avec lequel les communications sont privilégiées. Une telle solution remet cependant en cause la topologie du réseau et sort du cadre de cette étude.

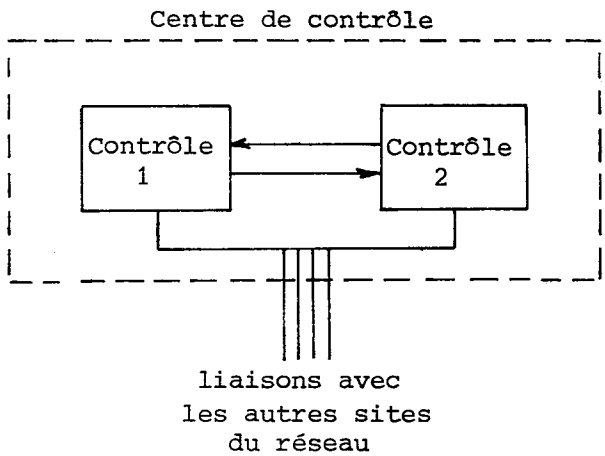


Figure 2.13

Dans le réseau CYCLADES, les applications réparties doivent donc être contrôlées sur un site quelconque.

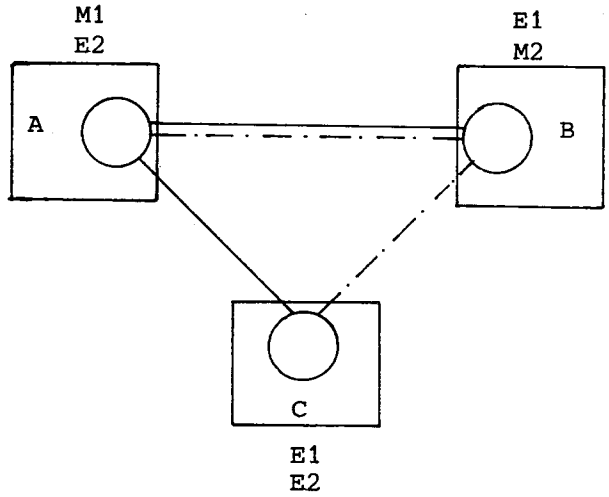
L'application sur les fichiers répartis présentée dans la troisième partie a retenu les principes suivants :

. répartition des contrôles (entre "esclaves"),

. un contrôleur "maître" en communication permanente avec les autres (bien que la topologie du réseau ne s'y prête pas),

. échange des rôles de "maître" et "esclave" pour chaque application (le maître est le plus proche par exemple).

On a donc le schéma suivant avec trois sites :



Pour l'application 1 :

. A est maître

. B et C sont esclaves

Pour l'application 2 :

. B est mâître

. A et C sont esclaves

Figure 2.14

Dans une architecture

Les contrôles à répartir sont trop liés à l'application envisagée pour que l'on puisse ici proposer une règle générale. Seuls certains contrôles, détaillés dans les pages précédentes, présentent des caractéristiques communes.

La solution la plus générale est de concevoir un *noyau de contrôle réparti*. Les possibilités d'implémentation en sont actuellement limitées. Leur intégration à des systèmes réseau permettrait aux nouveaux usagers du réseau de choisir parmi ces modules ceux qui correspondent à leur application. A défaut, c'est chaque usager qui devra définir lui-même la répartition et les contrôles associés :

. la centralisation entraîne une certaine lourdeur des traitements et facilite les abus de pouvoir ; toutes les demandes doivent être transmises à ce contrôleur central ;

. la dispersion des pouvoirs nécessite une bonne synchronisation et un algorithme de prise de décision suffisamment élaboré pour qu'aucun centre ne soit privilégié ou pénalisé par rapport aux autres.

De plus, un découpage judicieux des principales fonctions à réaliser, aide à la conception de modules suffisamment simples pour qu'ils puissent être vérifiables et contrôlés plus sûrement (voir exemple § 4.4).

On peut cependant remarquer que, vu l'état actuel des réseaux, les options d'architecture prises à ce niveau n'ont que peu d'incidence sur la sécurité. Ce sont essentiellement les fonctions de base telles que synchronisation et communication, qui doivent être sûres. Sans cela, aucune architecture, aussi simple soit-elle, n'offre de sécurité. C'est donc le dosage de ces fonctions dans une application qui donne le degré de la protection.

2.2.3.3. Localisation

Les schémas types

Certaines données sont initialement réparties, ainsi que le logiciel associé (SGBD). Le logiciel de l'application multibases utilisant ces données réparties peut être lui-même réparti sur les différents sites. Cependant, certains objets (modules, programmes, données) peuvent avoir plusieurs usages et doivent donc être maintenus disponibles sur le réseau.

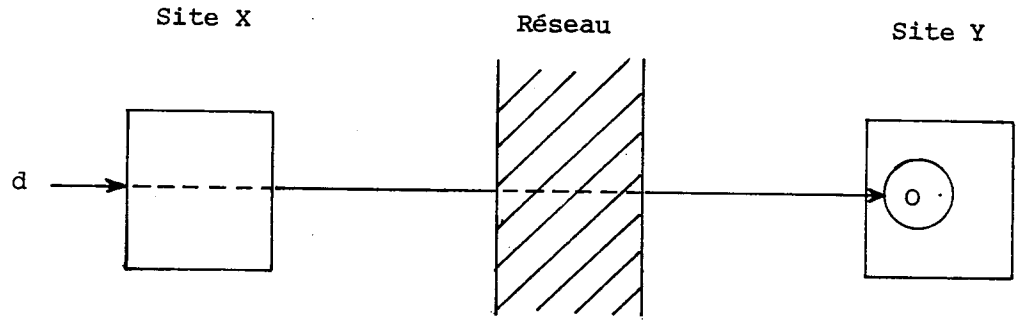
Plusieurs cas sont envisageables (voir schéma) :

1) l'objet n'est pas transportable (volume, disponibilité) et doit donc être accessible par des usagers distants ;

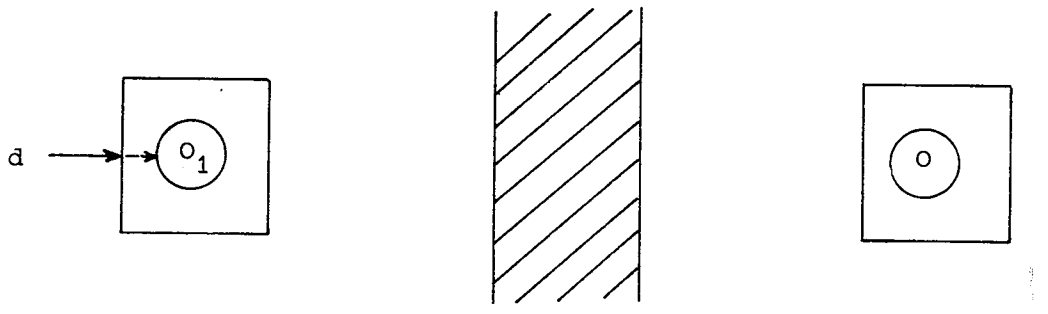
2) différentes copies sont disponibles (cohérence) sur chaque site ;

3) une demande de transfert de l'objet (original) peut être satisfaite sur simple requête de l'usager.

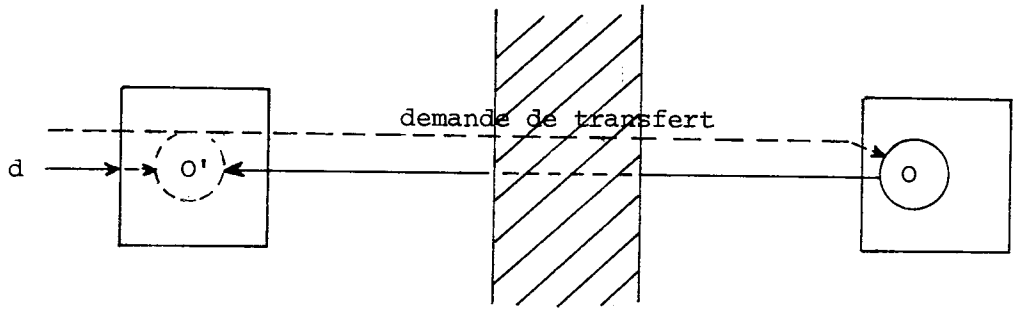
Comment satisfaire une demande d d'accès à un objet O :



1. Accès par le réseau à l'unique version de l'objet O



2. Accès à l'objet O_1, copie conforme de l'objet O (pré-existant)



3. Accès à l'objet O' après transfert de l'objet O sur le site du demandeur.

Figure 2.15

En fonction des caractéristiques de l'objet, de son utilisation et de sa sécurité, une de ces trois possibilités est à envisager. Les critères pris en compte sont détaillés dans ce qui suit.

Critères de localisation

Les caractéristiques de l'objet :

. son volume : taille du fichier,

. sa structure : morcelable ou non,

. son environnement : les autres objets constituant le contexte minimal acceptable, tels que bibliothèque ou compilateur,

. sa disponibilité : contraintes liées à l'environnement, au partage, à la protection ou à la portabilité,

. son niveau de secret : très confidentiel, à protéger, pour tous.

Les caractéristiques de l'utilisation :

. le contexte : autres objets concernés par la même application,

. l'accès : accès à une partition, modification de structure, ..

. la fréquence des demandes d'accès : lecture article par article, par exemple,

. le niveau de protection assuré,

. identification de l'utilisateur (y compris son éventuel niveau d'autorisation)

Remarques

La figure 2.15 pose un certain nombre de questions relatives au maintien de la confidentialité :

. les copies peuvent-elles être protégées aussi efficacement sur tous les sites ?

. comment peut-on estimer le coût des différentes méthodes ?

. le choix de l'unité de transfert importe-t-il ?

. y a-t-il plusieurs types de copies : l'original, les copies à jour ?

. les accès aux copies doivent-ils être enregistrés ?

. les copies de catalogues sont-elles à étudier en particulier ?

Il est hasardeux d'essayer de donner une réponse générale concernant ces différents points. Il s'agit plutôt ici de remarques :

la mise en oeuvre d'une procédure de transfert de fichier pourra bientôt être estimée avec précision pour certains types de fichiers (sur CYCLADES : "Virtual Transfer Protocol") et permettra de choisir entre les schémas 1 et 3. De plus, son intégration à un éventuel système réseau la rendra disponible à un ensemble plus large d'utilisateurs. On ne peut envisager de choisir une unité de transfert que si les fonctions d'accès réseau sont suffisamment précises : par exemple, si pour une application limitée, il est intéressant d'interroger à distance une base de données sans la transférer en entier, seules les caractéristiques permettant d'évaluer le critère d'interrogation devront être accédées. Ainsi, *minimiser la quantité d'informations* échangées sur le réseau va dans le sens d'une meilleure protection de la confidentialité des informations sur le réseau. Cependant, dans le cas où des échanges plus importants doivent être effectués, un excès de découpage allourdit l'ensemble de l'application, sans apporter de sécurité totale. Un compromis doit donc être trouvé pour de tels échanges.

Une des fonctions d'un système réseau

Un système réseau permet en particulier une gestion globale des ressources sur le réseau en regroupant l'ensemble des demandes d'accès. La localisation des objets à accéder peut donc être envisagée de plusieurs manières :

. l'usager connaît (explicitement) le site où se trouve l'objet qui l'intéresse ; le nom du site figure en paramètre de sa demande,

. l'objet demandé est dupliqué sur le réseau :

- soit l'usager choisit lui-même la copie qui l'intéresse,
- soit il fournit au système du réseau une liste de sites (au choix) ;

. l'usager ne connaît pas le site où se trouve l'objet,

. les caractéristiques de cet objet lui permettent de connaître sa localisation dans un catalogue.

Dès que le nombre d'objets (fichiers, procédures, ..) disponibles sur le réseau devient important, chaque usager se pose ces questions de localisation. Il paraît donc judicieux d'intégrer au système réseau, s'il existe, une fonction effectuant un tel service, afin de le mettre à la disposition de l'ensemble des usagers.

2.3. Incidence sur le traitement des données

2.3.1. Spécificité de la protection des données

La sécurité des objets à protéger dans un réseau concerne aussi bien les SGBD que les systèmes :

. les SGBD parce que l'accès à une donnée, inclue dans une base, est contrôlée par le système de gestion de cette base (exemple : SOCRATE, IMS) ;

. les systèmes parce que l'information qui est communiquée sur le réseau passe successivement sous le contrôle de différents systèmes, en l'absence de système propre au réseau.

Les problèmes de sécurité qui se posent à ces deux niveaux ne sont que partiellement semblables [DOWNS 77]. Relevons ici quelques uns de ces aspects :

1) *Les usagers leur comportement :*

dans un système, les usagers sont indépendants et s'intéressent à des objets différents : le problème de partage des informations se pose donc peu.

Par contre, dans les bases de données (BD), le partage constant d'une même base pose des problèmes nouveaux de maintien de *l'intégrité* en particulier (blocage). De plus, ces usagers jouissent de droits de types différents sur les mêmes objets.

2) *Les objets protégés :*

On considère actuellement plus volontiers que les systèmes protègent des *objets physiques* (fichiers, segments). Les BD concernent des enregistrements ou des champs de valeurs, considérés comme des *objets logiques*. La protection des éléments de base est alors plus fine (à cause de la taille des éléments à protéger) et plus complexe (à cause du nombre d'éléments et du nombre d'accès).

3) *L'authentification des usagers :*

Dans les deux cas, des informations décrivant les droits des usagers doivent être regroupées. Des procédures de LOGIN les utilisent :

. soit directement (système),

. soit à partir de l'identification fournie par le système (SGBD).

Le mode de stockage et de manipulation diffère alors selon le type de procédure.

4) *Les contrôles d'accès :*

Les systèmes présentent un cas simple, comparativement aux SGBD : *une matrice d'accès* suffit à regrouper les informations servant à autoriser/interdire un accès d'un demandeur à un objet géré par le système. Des noyaux de système ont donc pu être conçus sans trop de complexité à la base.

Les accès aux données contenues dans une base sont par contre contrôlés en fonction d'autres critères (que les caractéristiques du demandeur et de l'objet) : la valeur de la donnée, son historique et d'autres informations même doivent être prises en compte. Les contrôles y sont donc plus difficiles à mettre en oeuvre.

Les objectifs de la protection sont donc souvent les mêmes, mais les méthodes de réalisation diffèrent. Pour accroître l'efficacité des mécanismes et permettre la certification des opérations effectuées, l'objectif demeure de simplifier au maximum ces mécanismes, limiter les interfaces ou du moins les formaliser et les structurer, et minimiser les procédures de protection.

2.3.2. Sûreté du catalogue et des copies

La répartition d'informations, de procédures, ou plus généralement d'objets sur un réseau, pose rapidement un problème de *localisation*. Le catalogage des objets permet de résoudre de manière commode un tel problème.

Une procédure chargée de la gestion du catalogue doit en assurer le contrôle. Ses fonctions principales sont les suivantes :

. localisation d'un objet à partir de son nom,

. maintenance du catalogue (création, destruction de nouvelles descriptions d'objets),

. définition de nouveaux types d'objets (extensibilité).

La confidentialité ne peut être assurée sur le réseau que si l'accès à ces objets est suffisamment contrôlé et donc si l'accès aux descriptions de ces objets dans le catalogue est suffisamment protégé.

Parmi les caractéristiques des objets on peut citer : le nom, la localisation, le droit d'accès, le type. On peut distinguer aussi différents types de demandeurs d'accès au catalogue.

La multiplicité des copies de certains catalogues complique encore ce problème : aucun système cohérent de protection de copies n'est actuellement développé (pas seulement dans le cas de catalogues dupliqués). Il serait intéressant dans ce sens de définir avec précision

. la description d'une copie, son type de protection, ses accès,

. les contrôles d'accès à une version (avec maintien de l'intégrité,

. l'algorithme de choix d'une copie, tenant compte des contraintes d'intégrité et de protection (c'est-à-dire différencier aussi les copies en fonction de leur protection et non plus les banaliser pour l'ensemble des usagers : accroître ainsi la sécurité par rapport au cas où il existe une seule version de l'information).

2.3.3. Maintien de l'intégrité

L'intensification de la coopération entre bases distantes risque de compromettre l'intégrité des données si des messures efficaces de contrôle ne sont pas développées rapidement ; ce danger est accentué d'une part par l'accroissement des possibilités d'accès, d'autre part par la difficulté de mettre en place un mécanisme capable d'assurer la cohérence entre des informations physiquement éloignées (réparties sur différents sites et gérées éventuellement par des systèmes différents). Les contraintes à respecter peuvent être formulées sous différentes formes. Le modèle Data Semantics (DS) propose une expression élégante de ces contraintes. D'autres modèles tels que INGRES [STONEBRAKER 74, 75], considèrent plus les contraintes au sein de l'ensemble des données.

Programme de vérification d'intégrité (D.S)

A chaque demande de modification (insertion, suppression), les contraintes d'intégrité concernées sont vérifiées. L'opération demandée n'est effectuée que si ces contraintes sont satisfaites.

Soit par exemple la contrainte suivante : deux époux sont de sexe différent. Le programme de mise à jour associé à un époux aura la forme suivante :

```
UPDATER (epoux) ← PROG (x, y)
                    if sexe (x) = sexe (y)
                       then failure
                       else STD (x, y) end
```

C'est-à-dire que au cas où sexe (x) est différent de sexe (y), il n'y a pas de mise à jour autorisée, sinon le programme standard de mise à jour est effectué (STD).

Dans le cas de contraintes plus complexes, une telle formulation est difficilement acceptable. Exemple : soit la contrainte : deux futurs époux sont de sexe différent et leur situation familiale devient = marié. Elle porte à la fois sur l'objet sexe et sur l'objet situation familiale. Pour être autorisées, ces mises à jour devraient être simultanées (sur sexe et situation familiale). Les contraintes sont ici exprimées localement par catégorie et non globalement pour un ensemble de catégories.

Afin d'améliorer cette insuffisance, il faudrait agir à deux niveaux :

. au niveau du langage : pouvoir spécifier conjointement les mises à jour,

. au niveau des contraintes : les exprimer globalement.

Cette formulation, quoique encore insuffisante, peut être développée sur un réseau dans la mesure où le modèle de description choisi est Data Semantics. Cependant, il ne fournit pas de méthode pour la mise en oeuvre sur le réseau.

Modification de requêtes (INGRES)

Une autre approche, qui est plus facilement applicable sur un réseau, a été choisie pour le système INGRES.

Les techniques de maintien de l'intégrité développées dans le système INGRES, concernent le contrôle des erreurs de mise à jour risquant de compromettre la cohérence de la base [STONEBRAKER 75].

Les contraintes d'intégrité sont exprimées de manière prédicative dans un langage de haut niveau, comme cela a été préconisé par [FLORENTIN 74] et [CHAMBERLIN 73]. De plus ici des algorithmes d'implémentation sont indiqués.

Le langage de référence : QUEL (QUEry Language) présente une certaine indépendance des données (accès, structure) et adopte en partie une description relationnelle (relations et tuples).

Les contraintes d'intégrité sont stockées sous forme *d'assertions* concernant des variables de la base. Selon la complexité de ces variables, différents algorithmes permettent de vérifier ces assertions au cours des requêtes : les mises à jour qui violent ces propositions ne seront pas autorisées.

Quatre types d'algorithmes :

1) quand la contrainte ne porte que sur une variable simple indépendante. Exemple : les salaires des employés sont supérieurs à zéro ;

2) quand des variables multiples indépendantes sont concernées par la mise à jour d'une variable tuple. Exemple : les employés doivent gagner moins de 10 fois la valeur des ventes de leur département, si celui-ci a des ventes supérieures à zéro ;

3) comme 2), mais avec deux ou plusieurs tuples dans la mise à jour. Exemple : aucun employé ne peut gagner plus que son chef.

4) lorsque les variables appartiennent à un ensemble (un total, une moyenne). Exemple : le chef doit gagner plus de deux fois le salaire moyen des employés de son département.

Les contrôles d'intégrité sont donc effectués au même titre que les contrôles du langage : au plus haut niveau. Il reste donc encore des possibilités de notation frauduleuse de cette intégrité lorsque l'accès est situé à un plus bas niveau.

Cette solution présente par contre l'avantage de faciliter la déclaration de ces contraintes. Mais elle est évidemment très liée au modèle de description. Cependant, une telle méthode peut être facilement envisagée pour résoudre les problèmes de confidentialité à ce niveau. Les types de contraintes et d'algorithmes doivent alors être redéfinis dans une telle perspective.

Mise en oeuvre sur un réseau

L'intégrité est assurée par une ou plusieurs procédures de contrôle associées aux données. Il y a donc principalement deux solutions :

1) *unicité*

de la procédure de validation : elle gère tous les accès aux informations. Toutes les demandes circulent entre le demandeur et cette procédure, pour chaque accès. C'est au point de stockage qu'il paraît le plus simple de situer une telle procédure, associée ainsi au gérant local. Mais, même avec une telle configuration, les transmissions sont nombreuses et sujettes à divulgation.

2) *répartition :*

des processus distants effectuent la validation et seules les demandes autorisées parviennent au point de stockage. Le gérant local des données accepte donc les décisions prises par ces processus (éventuellement avec un consentement préalable) : il peut donc y avoir en ce cas un véritable *partage des responsabilités*. Un tel partage n'est justifié que si ses membres offrent une réelle sûreté de fonctionnement. Le gérant local se trouve alors dans une situation de *dépendance* vis à vis des autres.

Autres cas

Au-dela du problème d'intégrité, une telle alternative se présente plus ou moins clairement dès que l'on cherche à concevoir une fonction générale mise à la disposition d'utilisateurs de réseau général.

Des discussions de ces différentes approches peuvent aussi être trouvées dans le rapport INFOTECH State of The Art Report "Distributed Systems".

2.3.4. Exemple de traitement simple de données sur un réseau

Utilisation d'un concentrateur multiconnexions

Cet exemple présente un type d'architecture réseau extrêmement simple : les systèmes de gestion de bases de données sont complétés par des serveurs permettant l'accès à une base à partir du réseau. Le concentrateur multiconnexions décrit au § 2.2.2.2.4, permet d'accéder simultanément à plusieurs serveurs. Le mélange des informations, hors de l'environnement protégé que constitue le SGBD, pose donc des problèmes nouveaux de confidentialité : il s'agit là d'un début de coopération. Les traitements effectués sur les bases sont contrôlés par les SGBD locaux.

L'utilisateur est seul responsable de la coopération entre les bases. Il se présente comme un utilisateur local auprès des SGBD, c'est-à-dire qu'il est identifié et autorisé à ces deux niveaux. A l'utilisateur (terminal) peut être substitué un programme utilisant une interface adaptée au concentrateur. Cette application est réalisable sur le réseau CYCLADES.

Les contrôles d'accès au réseau ont donc lieu comme l'indique la figure 2.16 :

1) en entrée du concentrateur : identification de l'utilisateur, du terminal,

2) sur l'ensemble des connexions aux serveurs du réseau : cette limitation éventuelle du nombre de connexions simultanées, du type des connexions, .., est la partie la plus spécifique de l'application et aussi la plus complexe. C'est à ce niveau que devraient être introduites les contraintes de confidentialité globales ;

3) les contrôles propres au réseau : effectuées par le logiciel CYCLADES (contrôle d'erreur, contrôle d'acheminement sur les paquets) ;

4) accès réseau au serveur : identification, authentification ;

5) accès au SGBD : fonction de type SGBD (mot de pase, droit d'accès).

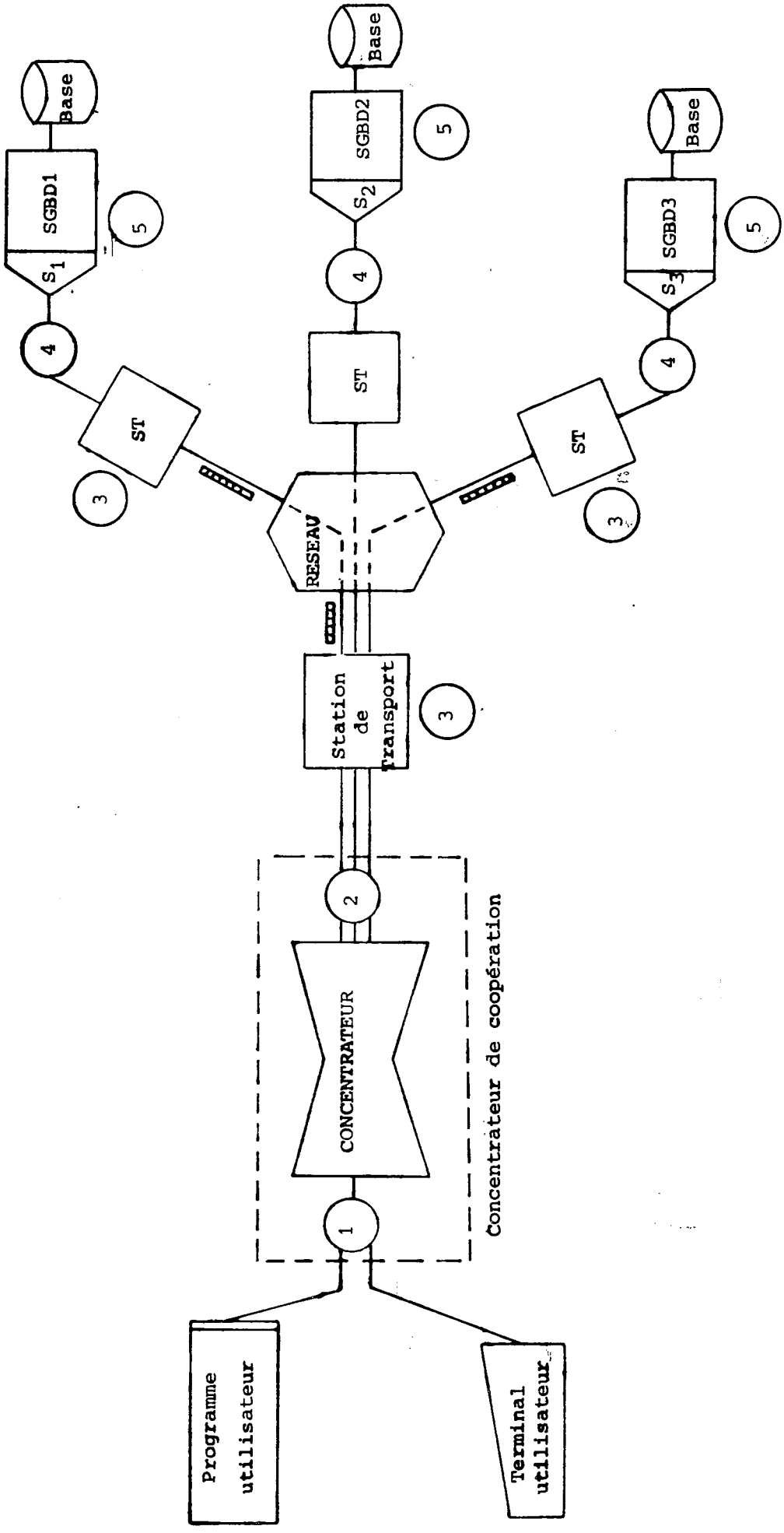

Base
SGBD1
S1
5
4
ST
3
Base
SGBD2
S2
5
4
ST
Base
SGBD3
S3
5
4
ST
3
RESEAU
Station de Transport
3
2
CONCENTRATEUR
1
Concentrateur de coopération
Programme utilisateur
Terminal utilisateur

Figure 2.16

La synchronisation et les échanges d'information sont à la charge de l'utilisateur et donc sous sa responsabilité.

La répartition des contrôles pendant les différentes étapes de travail est montrée par le tableau suivant :

Contrôle / étape	1 accès au CMC	3 multi-connexion	3 Réseau	4 Serveur du SGBD	5 SGBD
Login CMC	X				
Connexion à un serveur		Nbre connexions	X	X	
Lecture sur une base Ecriture		Secret de coopération	paquet	accès	accès
Déconnexion d'un serveur		journal	X	journal	
Logout CMC	X				

Figure 2.17

Aux différents niveaux (1 à 5) s'effectuent les nécessaires identifications et autorisations. Cette redondance (apparente) est cependant fondamentale à la sécurité de l'application sur le réseau, car entre ces niveaux, l'information est sujette à des modifications intempestives (circulation de l'information).

Le maintien de la confidentialité n'est pas assuré ici de manière automatique : c'est à l'usager du terminal de ne pas outre-passer ses droits d'accès à chaque base. Aucune contrainte globale d'accès aux bases ne peut être formulée car les logiciels et les connexions aux serveurs sont indépendants (on pourrait uniquement contrôler l'accès à plusieurs bases gérées par un même SGBD).

Cet exemple, un peu simple, ne comporte ni catalogue (les serveurs sont supposés connus de l'usager), ni copie, ce qui évite les problèmes spécifiques de leur protection.

2.3.5. Hétérogénéité des SGBD : choix d'un modèle

Objectifs des modèles

Les modèles permettent de représenter la connaissance que l'on a d'une certaine réalité. Ils sont plus ou moins complets, selon la façon dont ils approchent cette réalité. Les données figurant le réel sont donc structurées par ces modèles, plus ou moins bien adaptés aux applications et éventuellement spécialisés pour certaines manipulations (exemple : bases de données documentaires).

Les caractéristiques des modèles existants, tels que IMS, IDS, CODASYL, TDMS, AOV, 'Data Semantics', modèle relationnel de Codd, sont rappelées dans le cours de C. DELOBEL [DELOBEL 75], ainsi que de nombreuses références.

Les caractéristiques générales requises d'un modèle sont son efficacité et son évolutivité. Certains modèles, tels que 'Data Semantics', préconisent de plus l'indépendance sémantique des données : c'est-à-dire la dissociation entre la description et la manipulation des données.

Un modèle de bases de données est donc conçu pour permettre d'extraire, de mettre à jour, d'introduire des données et de les interroger, dans des conditions satisfaisantes (du point de vue de la protection : intégrité et confidentialité). La confrontation dans un réseau de SGBD hétérogènes demande donc qu'un modèle commun de description soit défini.

Lorsqu'une application multi-bases est envisagée, le modèle choisi doit permettre d'étendre les possibilités offertes par les systèmes locaux pour développer des applications nouvelles. Dans ce but, seuls les modèles proposés par Codd (relationnel) et Abrial (Data Semantics) sont retenus ici. Les autres systèmes, couramment utilisés, peuvent en fait se déduire des modèles relationnels.

Un modèle de coopération soulève évidemment de nouveaux problèmes de description qui ne sont pas envisagés ici : nouvelles catégories, fonction d'accès multibase .. [POLYPHEME 75] [ADIBA 76].

Avant d'aborder le problème spécifique de la confidentialité, il est intéressant d'analyser le comportement de ces modèles face au problème d'intégrité.

Un exemple de modèle de description : Data Semantics

La sémantique des données est présentée par J.R. Abrial [ABRIAL 74]. Il distingue des *objets concrets* (physical objects) qui représentent une entité concrète du monde réel et des *objets abstraits* (abstract objects) qui désignent une caractéristique d'un objet concret. La description d'un objet au sein de ce modèle est donnée par ses *connexions* avec d'autres objets (concrets ou abstraits). A chaque connexion sont associés les noms de deux *fonctions d'accès* (il peut s'agir éventuellement de fonctions multivaluées).

Un tel niveau élémentaire de description (relation binaire et fonction d'accès) semble le mieux adapté à la définition sémantique d'une base de données. Les fonctions d'accès présentent certaines propriétés intéressantes (cardinalité, minimum) et peuvent être décrites, le cas échéant, par des *programmes* (pour les contrôles de validité ou d'intégrité, par exemple).

Les objets des bases peuvent être classés en différentes *catégories* (exemple : les entiers, les personnes). Des opérations standard sont définies sur les objets et les relations permettant leur création/suppression. A chaque opérateur peut alors être associé un programme (cf. § 2.3.3.).

ANNEXE

Afin de donner une idée plus précise des problèmes de secret de la coopération de bases de données, un tableau regroupant les caractéristiques des bases existantes en vue d'une éventuelle interaction (au niveau des informations enregistrées, des usagers, des destinataires), est présenté ici. Il a été réalisé à partir du rapport de la Commission "Informatique et Libertés" (La Documentation Française)

Volontairement limitée aux fichiers des personnes, cette étude ne prétend pas à l'exhaustivité. Elle présente les charnières existant entre les domaines de l'emploi, de la Sécurité Sociale ou de la police en particulier Des interconnexions automatisées n'y sont pas encore développées, mais certains échanges de renseignements ont déjà lieu à titre officieux entre services publics. Un tel rapport montre bien l'urgence du problème du secret, tant dans le domaine privé que public.

Détenteur du fichier	Caise Primaire d'Assurance Maladie	Caisse d'Allocation Familiale	Protection Maternelle et Infantile	Direction de la Gendarmerie
Caractéristiques du fichier	. décrit l'assuré et les ayant-droit . 10 millions d'assurés (bientôt 30 millions)	. utilisé pour le calcul des prestations . 4 millions de familles	'GAMIN' . pour dépistage précoce des enfants . à terme 250 000 enfants	. 90 fichiers identiques . description des personnes recherchées . 400 000 personnes (+ archives de 1 850 000 personnes)
Informations enregistrées	. n° INSEE nom adresse date divorce.. . + fiche manuelle sur les prestations des 3 dernières années	. n° INSEE nom adresse date de naissance . + identification du conjoint/concubin . + identification des enfants . informations sur - ressources - logement	. identification des parents . données médicales sur enfant	. description des personnes recherchées ?
Origine des informations	. les assurés	. les allocataires	. parents + médecin	. la gendarmerie et les fiches de police (aucune contestation possible)
Destination des résultats du traitement	. la caisse	. statistiques . communiqués aussi à la police ou à l'inspection du travail	. medecin . mais aussi accessible aux assistantes sociales	. fichier de recherche et de documentation pour les affaires en cours

TROISIÈME PARTIE

EXEMPLES D'ARCHITECTURE D'APPLICATIONS SUR DES BASES DE DONNÉES RÉPARTIES

Les installations existantes sont gérées par des systèmes dont les caractéristiques ne sont pas comparables. Sur ces différents systèmes sont développés des Systèmes de Gestion de Bases de Données (SGBD) pour lesquels aucun standard n'est actuellement défini. Il est donc intéressant, au vu de l'état actuel des bases de données, de disposer pour la coopération de bases réparties d'un *modèle de description* commun représentant la configuration des bases, c'est-à-dire prenant en compte des structures hétérogènes (entités, hiérarchies, relations, ..).

Le modèle relationnel répond assez bien à ces objectifs et en général tout modèle permettant une analyse complète du réel fournit une description suffisante des bases existantes. [CODD 70].

En outre, l'environnement réparti requiert que des *méthodes d'accès* communes et un mécanisme de *catalogage* soient développés afin de faciliter, pour les usagers, l'accès aux ressources et la gestion des données dispersées (*localisation*). *Un système de contrôle* (centralisé ou réparti) permettant de maintenir la cohérence des bases et de vérifier les autorisations d'accès, doit y être adjoint.

La mise en oeuvre d'une telle application suppose donc que des mécanismes, plus ou moins élémentaires, permettent au minimum de réaliser les fonctions suivantes :

. contrôle (par modules indépendants),
. localisation (à partir des catalogues),
. exécution parallèle, asynchrone,
. gestion des ressources.

Les différents éléments qui constituent le corps de l'application répartie proposent donc des degrés de confidentialité variables. L'objet de ce chapitre est d'analyser les choix effectués dans certains modèles et certaines architectures (en cours de réalisation) selon leur efficacité pour maintenir le secret.

3.1. Exemple 1 : interrogation de fichiers et de bases de données répartis

3.1.1. Présentation de l'application [FICHIERS REPARTIS]

- Le travail effectué au sein d'une équipe de l'IRISA à Rennes depuis 1973, a consisté en la réalisation d'une maquette sur le réseau CYCLADES permettant l'interrogation de fichiers répartis (fichier Cobol et base SOCRATE décrivant le personnel). Les fichiers interrogés ont un contenu sémantiquement proche. Le problème d'hétérogénéité des informations est résolu par un catalogage adapté : l'ensemble des informations interrogeables est décrit virtuellement dans un macro-descriptif ; les caractéristiques réelles des informations réparties sur les bases sont contenues dans un sous-descriptif. Les requêtes générales, conformes à la macro-description, sont alors traduites en langage pivot, puis en termes de chaque sous-descriptif et complétées par des fonctions d'accès aux bases. Les modules fonctionnels de l'application sont répartis sur le réseau et contrôlés par un module privilégié appelé noyau maître.

3.1.2. Le modèle

Aucun modèle usuel de description n'a été choisi ici. Les informations sont caractérisées par leur type, nombre d'occurrences, longueur et champ de valeurs éventuellement.

La structure virtuelle des informations est, dans cette première maquette, hiérarchisée : toute information, excepté la racine, a un père utilisé ensuite pour la définition du chemin d'accès.

Ce macro-descriptif contient, outre cette description virtuelle des informations, un caractère de présence de l'information dans chacune des bases, afin d'optimiser l'analyse des requêtes.

Dans une seconde partie, cette macro-description présente les caractéristiques des bases décrites (ou partiellement décrites pour cette application), à savoir : verrou d'accès, blocage temporaire, nom virtuel de la base et surtout identification du sous-descriptif associé dans lequel figure la description plus complète de la base ou du fichier.

Des applications qui mettent en jeu des fichiers différents utilisent un macro-descriptif différent. Pour des interrogations de mêmes données, vues sous une structure différente, un macro-descriptif plus adapté peut être constitué, sans que les fichiers ou bases ne soient modifiés.

Ainsi, un usager dont les droits sont limités à une application précise n'aura-t-il pas les moyens d'outrepasser ses droits en exploitant le macro-descriptif.

Un tel choix de descriptif permet de limiter les fraudes et de cerner ainsi les utilisateurs concernés.

Un macro-descriptif est donc réservé à :

. un ensemble d'utilisateurs,

. un ensemble d'applications (ici il s'agit d'interrogation).

Les chemins d'accès aux informations qui y sont décrits sont le plus souvent directs (efficacité). Cependant, au cours d'une interrogation, il n'est pas possible de considérer les mêmes données avec des structures différentes (changement de hiérarchie). En fait, une description relationnelle des données ne poserait pas un tel problème, alors qu'ici un changement de macro-descriptif est nécessaire.

Prenons un exemple :

soit B1 une base décrivant les COMMANDES-CLIENTS
et les PRODUITS

soit B2 une base décrivant d'autres COMMANDES-CLIENTS
et d'autres PRODUITS

• Une première application sur B1 et B2 consisterait à interroger l'état du stock, c'est-à-dire à faire l'inventaire des différents produits, et pour chaque produit à lister les commandes en cours. Les données peuvent donc être décrites par la structure hiérarchisée suivante :

PRODUIT
  caractéristiques du produit
  COMMANDE (n)
    caractéristiques de la commande

• Une deuxième application sur B1 et B2 permettrait de connaître pour toute commande la quantité de produit disponible. La structure adoptée à une telle interrogation serait la suivante :

COMMANDE
  caractéristiques de la commande
  PRODUIT (p)
    caractéristiques du produit

La correspondance entre cette description virtuelle et la description réelle de l'information est effectuée au moment de la traduction (voir l'architecture). Une même information, qui figure dans n bases, sera décrite dans n sous-descriptifs différents pour une même application. S'il existe exceptionnellement plusieurs fonctions d'accès dans une même base à une information, celle-ci sera décrite plusieurs fois (sous un nom différent) dans le même sous-descriptif.

3.1.3. L'architecture

Les étapes de l'interrogation

L'utilisateur dispose d'un Langage Général d'Interrogation (LGI) simple, lui permettant essentiellement :

. de choisir une macro-description, c'est-à-dire une vue globale des bases qu'il désire interroger,

. de réserver le sous-ensemble des bases locales à interroger,

. d'énoncer un critère de sélection : expression logique portant sur les informations (virtuelles) de la vue globale,

. de définir les résultats de sa sélection : édition de certaines caractéristiques sur les objets sélectionnés dans les bases locales.

D'autres informations concernant le mode de travail choisi, le contrôle d'accès (mot de passe ..), figurent aussi dans cette requête.

La requête est alors analysée, sur place si possible, éventuellement remodelée en cas de refus pour non conformité à la macro-description, puis traduite en langage intermédiaire (Langage Pivot) et transmise aux différents contrôleurs associés aux bases réservées. Le noyau maître de l'application garde cependant le contrôle (ce qui facilite la reprise en cas de panne sur le réseau par exemple). La requête intermédiaire (en LP) est alors interprétée en termes des bases locales à l'aide des sous-descriptifs ; les fonctions d'accès nécessaires y sont ajointes. Le programme résultant après compilation éventuelle des séquences externes, est directement exécutable par le SGBD(ou un interface); il fournit les résultats de la sélection (selon le mode de travail demandé), qui sont alors traduits, conformément à la macro-description. Une dernière étape, après synchronisation sur les différentes réponses reçues, consiste en une fusion des résultats, coordonnée par le noyau maître.

Le découpage du logiciel

Les principales fonctions du logiciel sont donc :

1) analyse syntaxique → un arbre syntaxique + une table des valeurs,

2) traduction en langage pivot (LP) → programme en LP,

3) génération du programme d'interrogation → programme exécutable par le SGBD,

4) traduction des résultats → résultats "globaux",

5) synchronisation

6) contrôle : lancement à distance, liaison entre systèmes locaux.

Le schéma de répartition choisi donne la priorité au contrôleur local le plus proche et ne délègue que temporairement ses pouvoirs aux contrôleurs locaux éloignés (figure 3.1). Il n'y a aucune communication entre les noyaux esclaves ainsi répartis.

3.1.4. La sécurité

Elle intervient à la fois au niveau du choix du modèle et du choix de l'architecture, comme nous venons de le décrire dans les pages précédentes. On peut toutefois faire un premier bilan des choix effectués sur cette maquette et de leurs conséquences sur la confidentialité :

1) Les hypothèses de départ, volontairement limitatrices, ne posent pas vraiment le problème de la coopération entre bases, mais seulement de la mise en commun d'informations sémantiquement identiques. *Aucune nouvelle contrainte de confidentialité* n'est donc introduite du fait de cette juxtaposition de données hétérogènes. Une contrainte visant à limiter la comparaison entre données de bases différentes, reviendrait en fait à en interdire la juxtaposition.

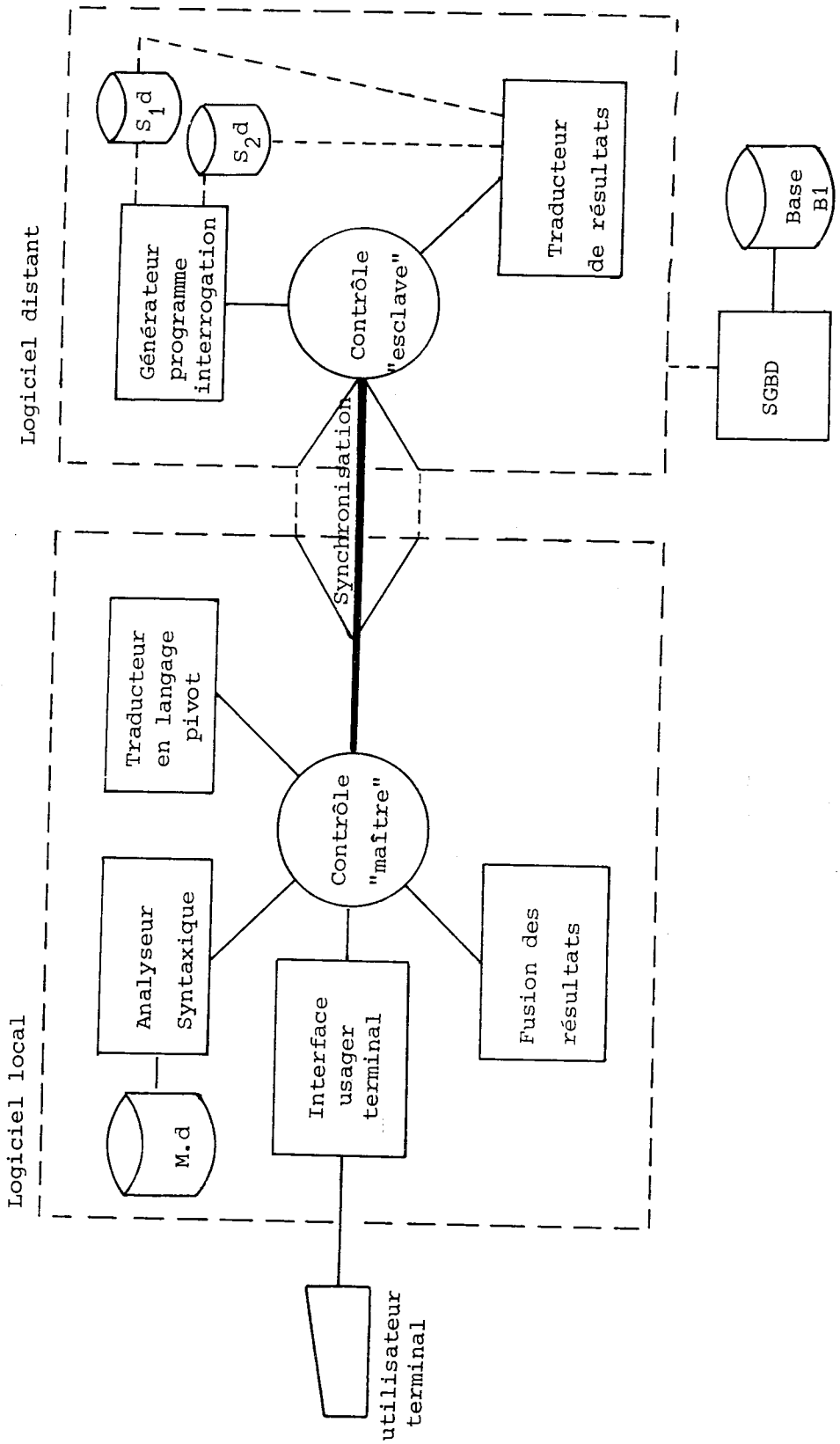

Logiciel distant
S_1^d
S_2^d
Générateur programme interrogation
Contrôle "esclave"
Traducteur de résultats
Base B1
SGBD
Synchronisation
Logiciel local
Traducteur en langage pivot
Contrôle "maître"
Analyseur Syntaxique
M.d
Interface usager terminal
Fusion des résultats
utilisateur terminal

Figure 3.1

2) *La définition statique de la macro-description* permet d'adapter la vue des bases à un type d'utilisateur en particulier (sans aller jusqu'à réserver un descriptif par usager) en fonction à la fois de ses besoins, c'est-à-dire essentiellement du type d'application envisagée, et des impératifs de secret.

3) *La répartition des centres de contrôle* (maître et esclaves) assure qu'une sécurité au moins égale à celle réalisée localement est effectuée au sein de l'application ; cela présente sur le réseau les avantages suivants : amélioration du diagnostic en cas d'incident, multiplication des procédures d'identification, authentification, contrôles parallèles.

3.2. Exemple 2 : Les communications dans POLYPHEME

3.2.1. Présentation

Le projet POLYPHEME [POLYPHEME 75] consiste à permettre la coopération de bases de données hétérogènes sur un réseau d'ordinateurs. Le système de coopération qui sera ainsi développé sur CYCLADES comprend :

. des éléments descriptifs de la coopération : description de la vue globale des bases qui peuvent coopérer, des vues locales de ces bases [ADIBA 77],

. la gestion des transactions : définition de requêtes d'interrogation, méthode de décomposition et d'évaluation sur les bases locales [CALECA 76],

. le contrôle réparti de l'exécution des transactions locales : un ensemble de primitives de communication propres au réseau CYCLADES réalise les échanges d'informations [ANDRE 77].

Le maintien de la confidentialité dans un tel modèle intervient donc à différents niveaux :

1) *dans les SGBD locaux* : les systèmes hétérogènes existant actuellement (IMS, CODASYL DBTG, SOCRATE, S2OO, ..) n'offrent pas les mêmes possibilités. Cependant, comme nous supposons ici que les bases locales ont été développées indépendamment et continuent d'être exploitées localement par l'intermédiaire du SGBD concerné, il n'est pas envisageable d'intervenir à ce niveau sur la confidentialité. On suppose donc ici que les bases de données sont maintenues confidentielles "localement" par les SGBD : tous les accès aux données, lorsqu'elles sont dans une base, sont contrôlés ; si l'utilisateur utilise le système de coopération pour accéder à une base, il subira les mêmes contrôles qu'un utilisateur local.

2) *Dans les systèmes d'exploitation locaux* : les requêtes, les résultats extraits des bases sont, dans des procédures, confiés au système local. Les données figurant dans les résultats sont donc hors de la protection de leur SGBD. Leur confidentialité doit donc être assurée par chaque système d'exploitation, comme cela a été présenté dans la première partie de ce rapport.

3) *Au cours des échanges* (ou entrées-sorties) *sur le réseau* : différentes propositions ont été faites à ce sujet dans la seconde partie : codage, utilisation de station de transport bénéficiant d'un contrôle sur les portes et les flots, gestion de plusieurs flots pour un même transfert. A cela, peuvent s'ajouter des contrôles plus spécifiques à cette application : en effet, les objets manipulés sont ici des primitives ou procédures. Leur contenu est inconnu du système de communication ; mais leur accès fait l'objet de contrôles à développer, comme nous allons le présenter au § 3.2.2.

4) *Dans le modèle de description* : un nouveau type de confidentialité peut être introduit par la coopération. Dans ce cas, aucun SGBD ne peut la garantir. Le modèle doit donc permettre de décrire ces contraintes sur des données réparties et de les vérifier préalablement à toute évaluation locale si possible. Le niveau de la description est donc aussi le niveau limite de la confidentialité.

Exemple 1 :

Soit B1 une base des voitures, immatriculations, adresse du propriétaire. Soit B2 une base des personnes, nom, profession, immatriculation de voiture. Soit la relation de confidentialité "habite". C'est-à-dire que l'association (nom, adresse) est confidentielle.

Une requête globale violant cette confidentialité devrait donc être refusée :

. soit dès l'analyse de la requête si le modèle permet de décrire et de reconnaître une telle relation simple,

. soit au moment de la décomposition,

. soit au moment de la synthèse des réponses, si une évaluation est nécessaire (par exemple, si la relation "habite" est confidentielle pour les personnes exerçant une certaine profession, il faudra évaluer préalablement la profession ou avoir modifié la requête en ce sens).

3.2.2. Communication entre programmes

Le fonctionnement actuel du réseau CYCLADES ne permet pas à un programmeur d'application de demander des entrées-sorties simples sur le réseau. L'interface existant avec la station de transport repose sur les notions de porte, flot, lettre, télégramme, très éloignées encore du "READ", "WRITE" du programmeur de langage évolué. L'absence de système réseau est donc très sensible à ce niveau.

Chaque concepteur d'application essaie donc, en attendant le développement de tels systèmes, de résoudre ce problème à sa façon. L'approche proposée pour mettre en oeuvre POLYPHEME est la suivante :

fournir en langage évolué (PL/1) des possibilités d'appel de procédures locales situées sur d'autres sites du réseau. Cette notion de procédure, en tant qu'objet réparti (portable), est également adoptée dans un système en cours de développement sur le réseau, comme IGOR [DU MASLE 74].

Certaines caractéristiques du réseau ont guidé ce choix :

a) le temps d'entrée-sortie réseau est très importante, comparativement à une entrée-sortie sur le système local ;

b) le parallélisme permet une utilisation optimale des possibilités des machines du réseau (et pallie à l'inconvénient sité en a) ;

c) la limitation des entrées-sorties et le parallélisme sont favorisés par des demandes d'entrées-sorties portant sur des objets dont la "durée" est importante : ce qui peut être le cas pour des procédures en particulier.

L'interface PL/1 proposée présente donc des *asynchronismes*. C'est-à-dire que le programmeur a la possibilité de lancer plusieurs exécutions en parallèle sur les différents sites et de continuer pendant ce temps une exécution locale. Lorsqu'une procédure distante est terminée, il peut alors effectuer un traitement adapté. Les temps dits morts en programmation synchrone, sont utilisés ici soit pour le calcul, soit pour lancer d'autres entrées-sorties réseau, c'est-à-dire d'autres appels de procédures.

Principe de fonctionnement de l'interface PL/1

Chaque site possède un *moniteur* chargé de gérer les appels de procédures locales et de transmettre aux moniteurs concernés les demandes d'appel de procédures locales sur les autres sites ; c'est le moniteur qui lancera la procédure de retour en fin d'exécution de la procédure lancée (distante).

Exemple : chronologie d'un appel de procédure

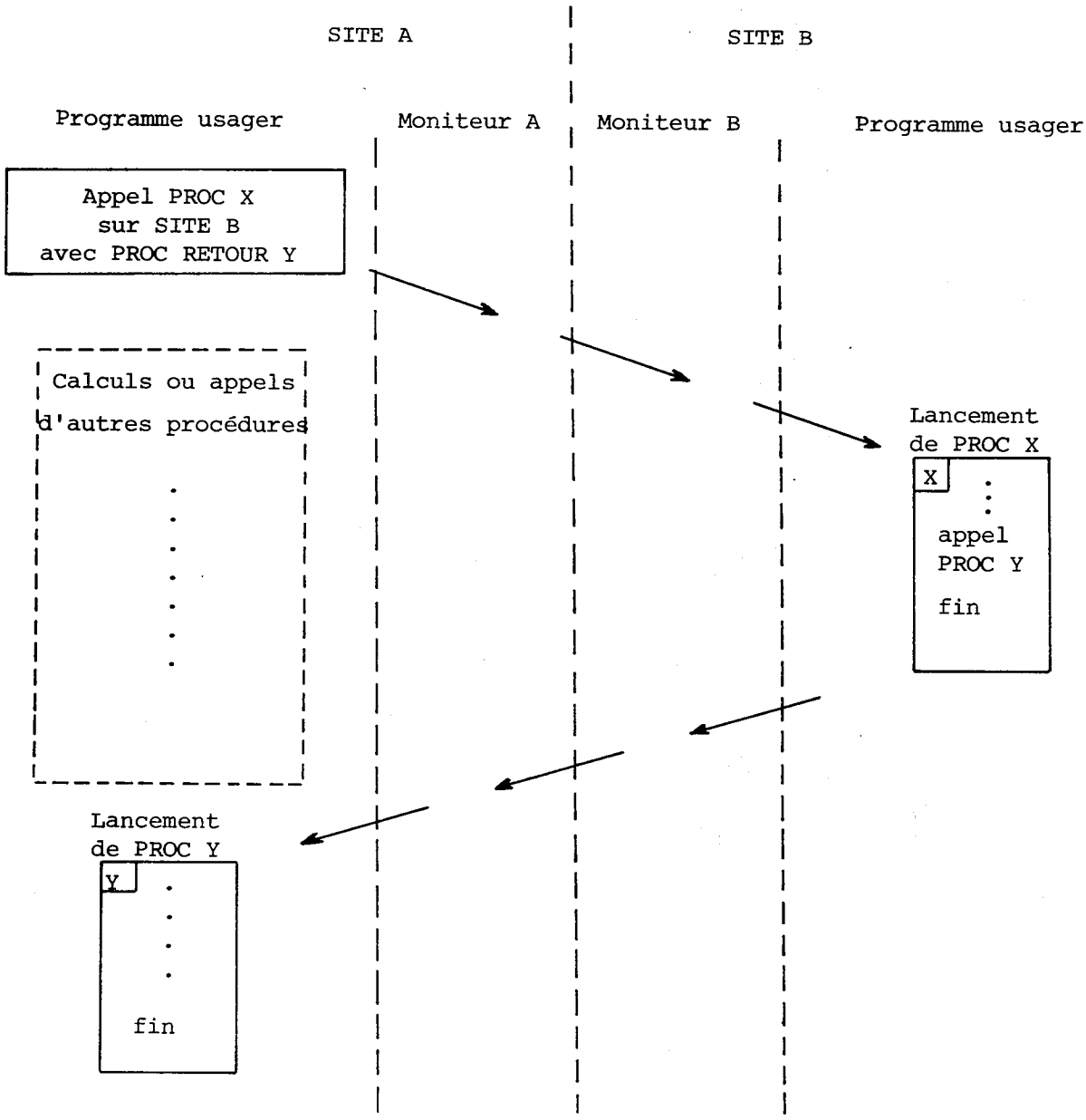


Figure 3.2

Choix du langage PL/1 :

. général : pour être utilisé sur les machines hétérogènes du réseau CYCLADES,

. évolué : pour programmeurs d'applications.

Confidentialité des communications

Le fonctionnement des communications entre programmes dans POLYPHEME, appelle les remarques suivantes :

1) aucune donnée n'est paramètre d'un appel : il s'agit toujours de PROCEDURES ; c'est le contrôle d'accès à ces procédures qui doit être contrôlé (catalogue) ;

2) l'interface n'a aucune action sur le contenu de la procédure ;

3) les procédures sont supposées sans erreurs ;

4) les procédures sont supposées préserver la confidentialité des objets manipulés (surtout lorsqu'il s'agit de données extraites des bases de données) ;

5) l'échange d'informations avec la procédure se limite à un passage de paramètres contrôlé.

Exemple 2 :

Soit PROC X = calcul de paie des employés sur une base de type SOCRATE

PARAMETRES

entrée
- E1 = nom de la base des employés
- E2 = liste des numéros d'agent des employés ou TOUS
- [E3 = classe de l'usager] (*)

sortie
- S1 = montant de la paie et nom de l'employé pour tous les employés figurant dans la liste E2

(*) La classe de l'usager permet d'ajuster les contraintes de confidentialité dans le cas où la procédure X est commune à des usagers ayant des droits différents.

Exemple 3 :

Le calcul de paie est effectué par plusieurs procédures auxquelles les droits d'accès sont contrôlés (fonction de la classe à laquelle appartient l'usager).

PROC P1 : calcul de la paie des employés ne dépassant pas un plafond de salaire S ;

PROC P2 : calcul de la paie d'une catégorie d'employés (les cadres par exemple) ;

PROC P3 : calcul de la paie des employés du service E (du rayon jardinage d'un magasin, par exemple).

Le maintien de la confidentialité des communications entre programmes se limite donc ici à la protection des procédures :

a) la protection la plus efficace est celle qui est utilisée *dès la conception des procédures* : différencier les procédures pour les classes d'usagers différents (comme dans l'exemple 3) ;

b) l'accès aux procédures est donc fondamental : vérification des droits d'accès par le système de *catalogage* des procédures ;

c) au cas où il n'est pas possible de différencier des procédures par classe d'usager (pour raison de place ou de complexité), les sorties d'informations de ces procédures doivent être limitées : *confinement des données*, en particulier.

Le catalogage des procédures

La correspondance entre *nom de procédure* et *adresse de procédure* est effectuée par le moniteur local, après consultation d'une table des procédures. Dans cette table, ou *catalogue*, seront incluses les informations permettant le contrôle d'accès (autre que interdire totalement ou autoriser à tous).

En fait, chaque usager aura à sa disposition un *catalogue personnifié* des procédures auxquelles il a accès et des conditions d'accès. Seules certaines informations de ce catalogue sont modifiables par lui. Seul "l'administrateur" (ou utilisateur privilégié) peut agir sur la totalité des informations contenues dans ces catalogues.

L'usager dispose donc de primitives d'accès au catalogue :

CATAL : pour cataloguer une nouvelle procédure en définissant les droits d'accès,

DECATAL : pour décataloguer une procédure (dont il est déclaré propriétaire).

Elles lui permettent de créer ou de supprimer dynamiquement un point d'entrée dans le catalogue.

Lors d'un appel de procédure locale, la procédure de retour est incluse dans le catalogue de l'appelant. Les moniteurs de l'appelant et de l'appelé doivent effectuer les contrôles sur la procédure appelée (vérification des droits). Une procédure de retour n'est cataloguée que pour une durée limitée, sous le contrôle du moniteur.

Les procédures standard ou "utilitaires"

Pour faciliter la programmation des demandes d'entrées-sorties réseau, il est intéressant de définir des primitives plus globales qui constituent un niveau intermédiaire entre l'entrée-sortie réseau et le programme d'application. Elles peuvent être par exemple pour une application de gestion de copies multiples :

COPIER un fichier F sur le site A
METTRE A JOUR un fichier F sur le site B pour conformité C

Les objets à protéger sont donc les paramètres de ces primitives (si l'on suppose bien sûr qu'elles sont sans erreur et confinées) : en particulier les fichiers. Les mécanismes à introduire ici sont donc du même type que ceux introduits pour protéger les appels de procédures : contrôle d'accès, catalogue des fichiers, adresse, clé d'accès et droits d'accès, identification des usagers.

3.2.3. Confidentialité dans le modèle

1) Les contraintes de confidentialité *internes aux bases locales* sont vérifiées par le SGBD local. En aucun cas le modèle de coopération ne doit supplanter le SGBD défaillant, car son action se révèlerait toujours inefficace pour les accès locaux directs au SGBD local.

2) Par contre, *les contraintes nouvelles* (voir exemple 1, § 3.2.1) sont évidemment à la charge du modèle de coopération. Il doit permettre de décrire ces contraintes, d'entreprendre les contrôles au niveau le plus haut possible (à partir de la requête si possible).

Le traitement d'une contrainte peut être considéré comme *une transaction normale*, semblable à celle qui est effectuée pour satisfaire une requête portant sur des données réparties : décomposition, évaluation sur les bases locales, synthèse des résultats. Il permet de savoir si la condition d'accès est ou non respectée et d'entreprendre les modifications éventuelles de la requête globale (refus de réponse, réponse partielle, ..).

Le mécanisme de communication entre programmes décrit au § 3.2.2. permet donc de réaliser de tels contrôles. L'introduction de procédures "utilitaires" concernant spécialement l'évaluation des contraintes réparties constitue une simple facilité de programmation. On peut envisager de développer les fonctions suivantes, par exemple :

EVALUER contrainte C sur base B
MARQUER les données D suivant la contrainte C.

Les optimisations prennent ici une place prépondérante dans la mesure où l'évaluation d'une contrainte d'accès est en général suivie de l'accès effectif à l'information (si la contrainte est satisfaite). Le plus souvent, les informations ainsi contrôlées appartiennent au même environnement sémantique de la base.

Une étude sémantique de ces contraintes devrait permettre d'insérer rapidement au modèle POLYPHEME les contrôles de confidentialité qui lui font défaut ; les primitives de communication sont donc capables d'assurer le transfert de ces contraintes dans le cadre d'un système de maintien de la confidentialité.

CONCLUSION

La démarche adoptée dans cette étude a consisté, dans un premier temps, à montrer ce qu'il est possible d'attendre des mécanismes de protection existants. Nous avons également essayé de définir les principaux concepts associés à la confidentialité. Notre intérêt s'est alors principalement porté sur les réseaux et, plus spécialement, sur un type d'application sur ces réseaux : celles qui mettent en jeu des bases de données.

Certains points présentés dans cette étude mériteraient des prolongements. Il s'agit de domaines encore peu abordés, tels que :

- cohérence et contrôle d'accès à des bases réparties,
- étude des contraintes de confidentialité,
- communication entre systèmes - noyaux,
- comparaison des méthodes de protection ascendantes et descendantes,
- modélisation des fonctions de protection.

On peut rappeler ici les grandes lignes du point de vue que nous adoptons :

. le développement de systèmes spécifiques aux réseaux et l'éclatement des noyaux de sécurité en noyaux répartis, communiquant entre eux, pour lesquels la sécurité est une préoccupation de premier plan, devront bientôt permettre une mise en oeuvre plus sûre des applications dites "réparties" ;

. l'approche système de la confidentialité a permis de mettre en évidence les carences sémantiques actuelles de la définition de la confidentialité dans une application sur les bases de données réparties ;

. l'adoption rigoureuse des principes de sécurité élémentaires dans les applications réparties concernant des bases de données, permettra de garantir un niveau minimal "d'intimité", mais jamais une protection totale. L'intégration de la protection doit être effectuée de manière cohérente entre le modèle, l'architecture et le système d'interrogation. Les difficultés de la formalisation simple des fonctions de protection montrent bien, s'il en était encore besoin, que de problème est encore mal compris.

Les exemples présentés dans la troisième partie (interrogation de fichiers répartis, communication dans POLYPHEME,) ont permis d'illustrer ce point de vue et dénoncent l'urgence de la mise en oeuvre des mécanismes permettant de maintenir la confidentialité. Ces réalisations ont été le centre de notre réflexion sur la confidentialité.

On peut cependant faire ici une liste des principaux axes de recherche bien développés actuellement, dont les conclusions devraient permettre d'améliorer la sécurité des informations dans un environnement réseau, en particulier :

. méthodes d'authentification "inviolables",

. certification, preuve de programme, vérification des systèmes de protection (à la conception et en fonctionnement),

. sensibilité des programmes/données aux pannes : vulnérabilité,

. étude sur l'utilisation des données : confinement ...

. codage.

B I B L I O G R A P H I E

Les références sont données :

- *soit par nom d'auteur et date,*
- *soit par nom de produit (CYCLADES, fichiers répartis, POLYPHEME, SOC, SYNCOP, TRANSPAC).*

ABRIAL 74 — J.R. ABRIAL, *Data Semantics*, Cargese, Corse, Avril 1974.

ACM 76 — *Privacy, security and the information processing industry*, ACM, Los Angeles Chapter, 1976.

ADIBA 76 — M. ADIBA & C. DELOBEL, *Le problème de la coopération de plusieurs bases de données*, Université de Grenoble, Juin 1976.

ADIBA 77 — M. ADIBA, *MOGADOR, Modèle général de données réparties*, Note Technique LA 7, Grenoble, Mai 1977.

AFIPS 74 — *AFIPS System Review Manual on Security*, American Federation of Informations Processing Societies, Inc., Montvale, N.J. 07645, 1974.

ANDRE 77 — E. ANDRE & P. DECITRE, *On providing distributed application programmers with control over synchronization*, Symposium on Computer Network Protocols, Liège, Belgique, Février 1978.

BAL — *Procédures systèmes SIRIS 7 / SIRIS 8, version B09 / C09*, Manuel d'utilisation CII-HB, Ref 4493 E/FR.

CALECA 76 — J.Y. CALECA & J.M. FORESTIER, *Interrogation simultanée de bases de données*, Projet DEA Génie Informatique, Grenoble, Juin 1976.

CHUPIN 77 — J.C. CHUPIN, H. RICHY, J. SEGUIN, *Data sharing and cooperation between DBMS in heterogeneous computer networks*, AICA 77, Pise, Italie, Octobre 1977.

CMC — D. PORTAL & H. RICHY, *Réalisation d'un concentrateur "intelligent"*, Bibliothèque CYCLADES, UTI 076, Août 1976.

CODD 70 — E. CODD, *A relational model on data for large shared data banks*, Communications ACM, 13,6, Juin 1970, pp. 377-387.

CROCUS 75 — *Systèmes d'exploitation des ordinateurs. Principe de conception*, Editions DUNOD, 1975.

CYCLADES 1 — L. POUZIN, *Présentation du réseau CYCLADES*, Note CYCLADES, GAL 506.

CYCLADES 2 — L. POUZIN, *Rapport d'analyse du projet CYCLADES*, Note CYCLADES, GAL 510.

CYCLADES 3 — *Spécifications fonctionnelles de la station de transport du réseau CYCLADES, Protocole ST - ST*, Note CYCLADES, SCH 502.3, Mai 1973.

CYCLADES 4 — H. ZIMMERMANN, *Protocol for a virtual terminal protocol (VTP)*, Note CYCLADES, TER 533.1, Juillet 1976.

CYCLADES 5 — *Spécifications de réalisation de la station de transport ST2 portable*, Note CYCLADES, SCH 536.1, Février 1975.

DELOBEL 75 — C. DELOBEL, *Les systèmes de bases de données*, Rapport, Université de Grenoble, Juin 1975.

DOBKIN 76 D. DOBKIN, A.K. JONES, R. LIPTON, *Secure data bases : protection against user inference,*
Research Report n° 65, Avril 1976, Bibliothèque SIRIUS, PRO.E.011.

DOWNS 77 D. DOWNS & G.J. POPEK, *Approaches to data management security,*
Proceedings 2nd Workshop on Distributed Data Management and Computer Networks, Berkeley, Mai 1977.

DU MASLE 74 J. DU MASLE, M.N. FARZA, G. SERGEANT, *Proposed organization of an interpreter intended for the implementation of high level procedures in a computer language,*
IFIP Working Conference on Command Language, Suède, Juillet 1974.

FICHIERS REPARTIS 1 P. BOURRET, P. BOSC, H. RICHY, *Rapport de Synthèse,*
Université de Rennes, Janvier 1975.

FICHIERS REPARTIS 2 A. CHAUFFAUT, H. RICHY, J.M. VILLARD, *Définition du langage d'interrogation et analyseur syntaxique,*
Université de Rennes, Février 1975.

FICHIERS REPARTIS 3 P. BOSC, A. CHAUFFAUT, H. RICHY, J.M. VILLARD, *Méthode d'interrogation par langage pivot et aspect réparti de l'application,*
Université de Rennes, Novembre 1975.

HOFFMAN 69 L.J. HOFFMAN, *Computers and privacy : a survey,*
Computing Surveys, 1,2, Juin 1969.

HOFFMAN 71 — L.J. HOFFMAN, *The formulary model for flexible privacy and access controls*, FJCC 1971.

HOFFMAN 77 — L.J. HOFFMAN, *Modern methods for computer security and privacy*, Prentice Hall International, 1977.

HSIAO 74 — D.K. HSIAO, D.S. KEN, C.J. NEE, *Context protection and consistent control in data base systems (part 1)*, Report OSU.CISRC.TR.73.9, Ohio State University, Colombus, Février 1974.

KAHN 67 — D. KAHN, *The codebreakers*, New York, MacMillan, 1967.

LAMPSON 73 — B.W. LAMPSON, *A note on the confinement problems*, ACM vol 16, n° 10, Octobre 1973, pp. 613-614.

LIPNER 72 — S.B. LIPNER, *SATIN Computer security*, Bibliothèque CYCLADES, SEC 003, Septembre 1972, MITRE Corporation MCI 75.2.

MINSKY 76 — N. MINSKY, *International resolution of privacy protection in data base systems*, C.ACM, 19,3, Mars 1976, pp. 148-159.

MORRIS 76 — J.H. MORRIS, *Protection in programming languages*, C.ACM, 16,1, Janvier 1973, pp. 15-21.

NEGARET 76 R. NEGARET, *Etude de l'allocation des ressources dans les systèmes répartis,*
Thèse de Troisième Cycle, Université de Rennes, Décembre 1976.

POLYPHEME 75 Groupe de travail POLYPHEME, *Propositions pour un modèle de répartition et de coopération de bases de données dans un réseau d'ordinateurs,*
RR n° 29, LA 7, Université de Grenoble, Décembre 1975.

POPEK 75 G.J. POPEK, *On data secure computer networks,*
ACM SIGCOMM/SIGOPS, Santa Monica, Mars 1975.

Rapport de la Commission "Informatique et Libertés" (tomes 1 et 2),
La Documentation Française, 1975.

RITCHIE 74 D.M. RITCHIE & K. THOMPSON, *The unix time sharing system,*
C.ACM, 17,7, Juillet 1974, pp. 365-375.

SALTZER 74 J.H. SALTZER, *The protection and the control of informations sharing in Multics,*
C.ACM 17,7, pp. 388-402, Juillet 1974.

SALTZER 75 J.H. SALTZER & M.D. SCHROEDER, *The protection of information in computer systems,*
Proceedings of the IEEE, Septembre 1975.

SOC 1 M. SOMIA BRECHET, *Etude de système de contrôle d'un réseau d'ordinateurs distribués et de ses relations avec les systèmes de contrôle locaux,*
Thèse de Doctorat d'Etat, Université Paris VI, Juin 1975.

SOC 2 S. DANISH, *Système de contrôle du réseau d'ordinateurs SOC*, Thèse de Docteur-Ingénieur, Université Paris VII, Février 1974.

STONEBRAKER 74 M. STONEBRAKER, *High level integrity assurance in relational data base management systems*, ERL M473, Août 1974. Université de Berkeley, Californie.

STONEBRAKER 75 M. STONEBRAKER, *Implementation of integrity constraints and views by query modification*, ERL M514, Mars 1975. Université de Berkeley, Californie.

STONEBRAKER 76 M. STONEBRAKER & P. RUBINSTEM, *The INGRES protection system*, Berkeley ERL M594, Juillet 1976.

SYNCOP Ng.X. DANG, V. QUINT, J. SEGUIN, G. SERGEANT, *Présentation et définition de SYNCOP , un sous-système normalisé de commutation de processus pour la télé-informatique et les réseaux d'ordinateurs*, Rapport de Recherche n° 64, LA7 Grenoble.

TRANSPAC *Caractéristiques techniques d'utilisation des services TRANSPAC*, Direction Générale des Télécommunications, CNET, Issy les Moulineaux, Novembre 1976.

TURN 75 R. TURN & W.H. WARE, *Privacy and security in computer systems*, Protection of Information in Computer Systems, COMPCON, IEEE, Fall 1975, pp. 49-56.

WINKLER 74 S. WINKLER & L. DANNER, *Data security in the computer communication environment*, Computer, Février 1974, pp. 23-31.

WULF 74 W. WULF & al., *HYDRA : the kernel of a multiprocessor operating system*, C.ACM 17,6, Juin 1974, pp. 337-345.

dernière page de la thèse

AUTORISATION DE SOUTENANCE

VU les dispositions de l'article 3 de l'arrêté du 16 Avril 1974,

VU les rapports de présentation de Messieurs :

- L. BOLLIET, Professeur à l'U.S.M.G.
- J. LE BIHAN, Ingénieur de Recherche - I.R.I.A. LE CHESNAY

Mademoiselle Hélène RICHY

est autorisée à présenter une thèse en soutenance pour l'obtention du diplôme de DOCTEUR-INGENIEUR, spécialité "Génie Informatique".

Grenoble, le 24 Janvier 1978

Ph. TRAYNARD
Président
de l'Institut National Polytechnique