Resolution of deducibility constraint systems for the composition of security-aware Web services

PhD Thesis by

Tigran Avanesov

supervised by

Michaël Rusinowitch

lots of help by

Yannick Chevalier and Mathieu Turuani

Loria, INRIA Nancy - Grand Est, UHP Nancy - 1, IAEM Lorraine

September 19, 2011

Outline

Preserve security property: need cryptography

Preserve security property: need cryptography , but not straightforward

A secure protocol may be not so secure...

We need to verify secure communication schemes

Even if you think they work well...

Symbolic model

Real encryption, e.g. AES

- Message: `'messagetoencrypt'` — `0x7468 6576 6572 7973 6563 7265 746b 6579`
- Key: `'theverysecretkey'` — `0x6d65 7373 6167 6574 6f65 6e63 7279 7074`
- Encrypted message: `0xcd54 381e 3b8f 5981 f108 76e9 4e64 b4b6` (no good ASCII representation)

Abstraction

- Message: m
- Key: k
- Encrypted message: $\{m\}_k^s$

Messages may be complex, e.g. $\{\{m.n\}_k^s\}_{a.n}^s$
Abstract away algorithms.

Symbolic analysis

Symbolic representation

$\{t_1\}^a_{t_2}$	t_1 **encrypt**ed with **public** key t_2
$\{t_1\}^s_{t_2}$	t_1 **encrypt**ed with **symmetric** key t_2
$t_1.t_2$	t_1 **concatenate**d with t_2
$\mathrm{priv}(t_2)$	private key for public key t_2
$[t_1]_{\mathrm{priv}(t_2)}$	**signature** of message t_1 with $\mathrm{priv}(t_2)$
$t_1(t_2)$	apply **hash** function t_1 on message t_2

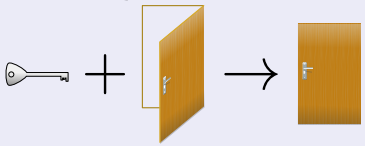
Public-key encryption: Two types of keys

- **Public** key (to encrypt),

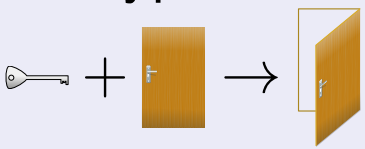
- **Private** key (to decrypt),

Symmetric encryption: **one** shared key

- Encryption:



- Decryption:



Messaging and Protocol participants

Terms

- $a, b, c, \ldots$ — atomic data. They are terms.
- $X, Y, \ldots$ — variables. They are terms.
- p, q — terms $\Longrightarrow$ $\{p\}^s_q, p.q, \ldots$ are terms.
- Ground terms (messages) are terms without variables.

E.g. $\{\{a.b\}^s_k\}^s_{c.b}$ $\quad X.[X]_{\mathrm{priv}(pk)}$ $\quad \{a\}^a_{pk}$

Protocol participants (agents, protocol roles instances)

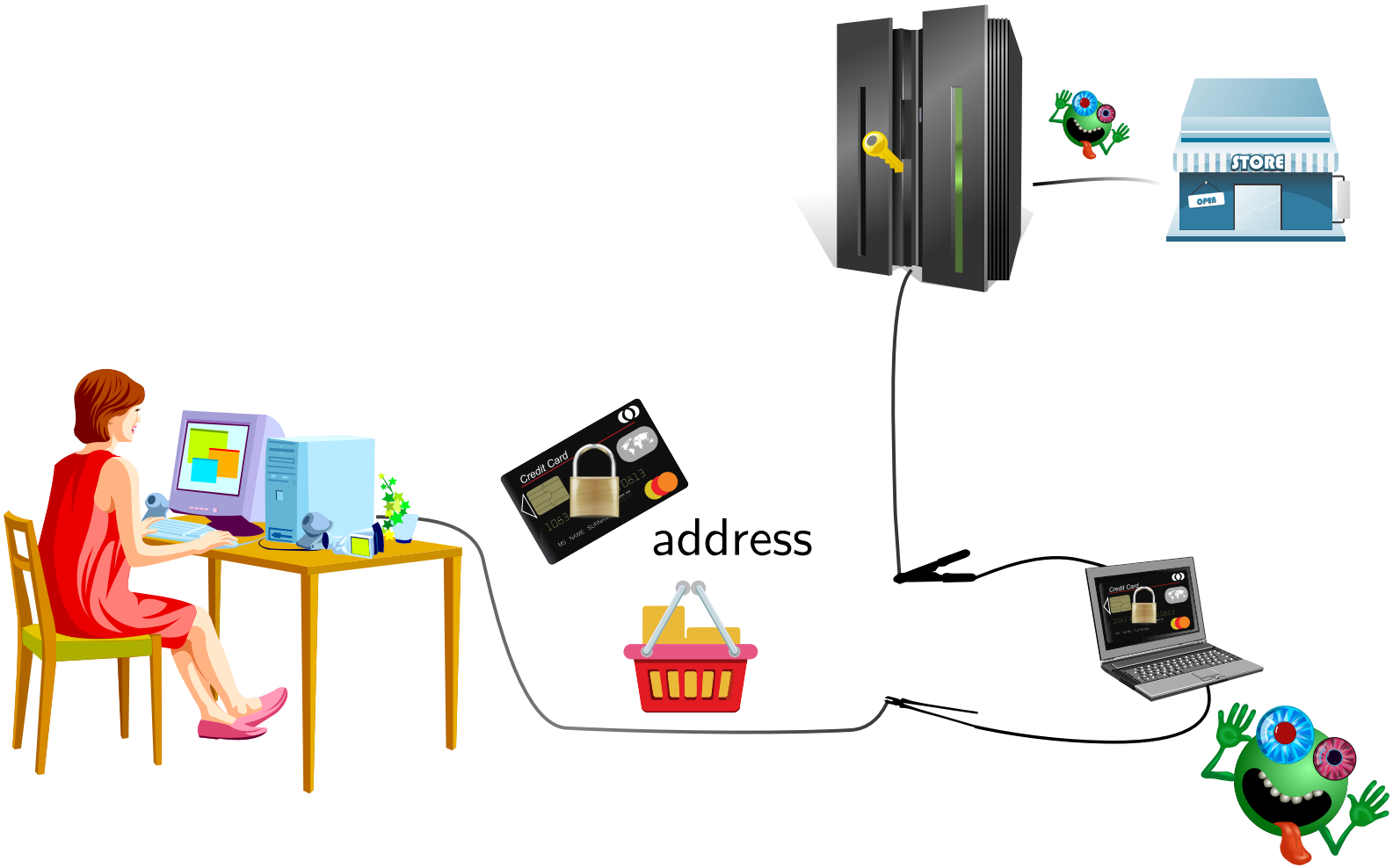
Agent's behavior is defined as $\frac{?}{!}t_1;\ \frac{?}{!}t_2;\ \ldots;\ \frac{?}{!}t_{k-1};\ \frac{?}{!}t_k$
? is "receive", ! is "send"

Example of an agent with four actions

$?\{X\}^s_k;\ !md5(X);\ ?X.Y;\ !\{token\}^s_Y$

Dolev-Yao intruder

— can intercept (read and block) messages

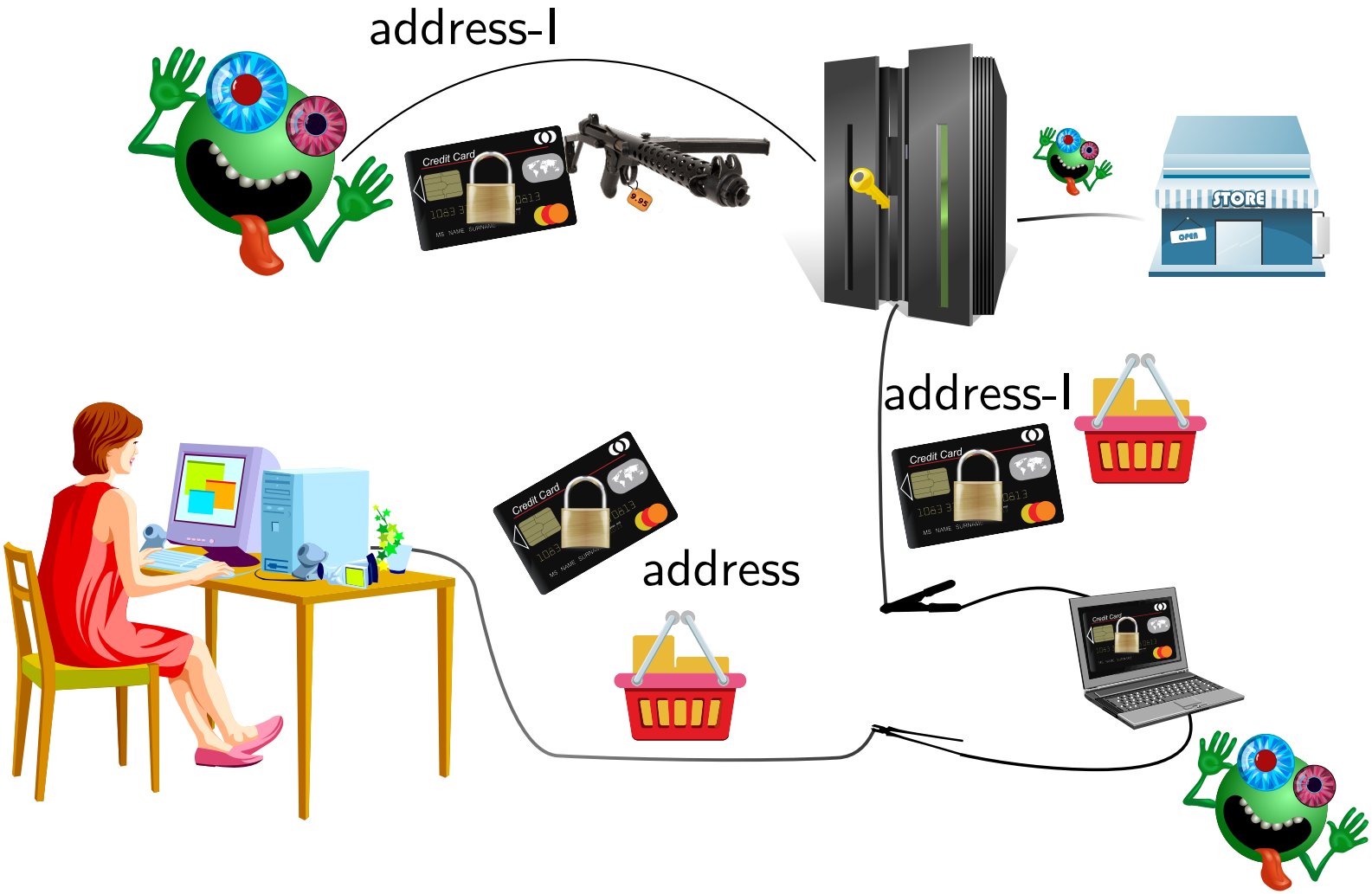


Dolev-Yao intruder

— can generate and send messages (on behalf of honest users)

Dolev-Yao intruder

— can be a legitimate user



Modeling intruder actions

Deduction rules for the intruder

Composition rules	Decomposition rules
$t_1, t_2 \rightarrow \{t_1\}^s_{t_2}$	$\{t_1\}^s_{t_2}, t_2 \rightarrow t_1$
$t_1, t_2 \rightarrow \{t_1\}^a_{t_2}$	$\{t_1\}^a_{t_2}, \mathrm{priv}(t_2) \rightarrow t_1$
$t_1, t_2 \rightarrow t_1.t_2$	$t_1.t_2 \rightarrow t_1$
$t_1, \mathrm{priv}(t_2) \rightarrow [t_1]_{\mathrm{priv}(t_2)}$	$t_1.t_2 \rightarrow t_2$
$t_1, t_2 \rightarrow t_1(t_2)$	

Perfect cryptography

Encryption is a black box.

Analysis of protocol's sessions

Sample protocol ("password restore")

$$A \to B: \quad \{A.K\}^a_{K_B}$$
$$B \to A: \quad \{pwd(A)\}^s_K$$

K — fresh symmetric key generated by A
K_B — public key of B
$pwd(A)$ — forgotten password

Protocol should guarantee

- $pwd(A)$ is known only by A and B.

Protocol insecurity problem

Given a finite set of protocol instances (or their number), **find out** whether the security properties guaranteed by the protocol are not preserved in the presence of a DY intruder.

Reducing insecurity problem to satisfiability of deducibility constraints

$a(A)$ asks to restore his password on $b(B)$

Initial intruder's knowledge: i, a, b, k_b

Constraint system

$$\begin{cases} i, a, b, k_b, \{a.k\}^a_{k_b} \rhd \{a.X\}^a_{k_b} \\ i, a, b, k_b, \{a.k\}^a_{k_b}, \{pwd(a)\}^s_X \rhd pwd(a) \end{cases}$$

One of the solutions: $\{X \mapsto i\}$

Reducing insecurity problem to satisfiability of deducibility constraints

$a(A)$ asks to restore his password on $b(B)$

$$a \xrightarrow{\{a.k\}^a_{k_b}} (1) \quad i \quad (2) \xrightarrow{\{a.X\}^a_{k_b}} \quad b$$
$$a \xleftarrow{\{Y\}^s_k} \quad i \quad (3) \xleftarrow{\{pwd(a)\}^s_X} \quad b$$

Initial intruder's knowledge: i, a, b, k_b

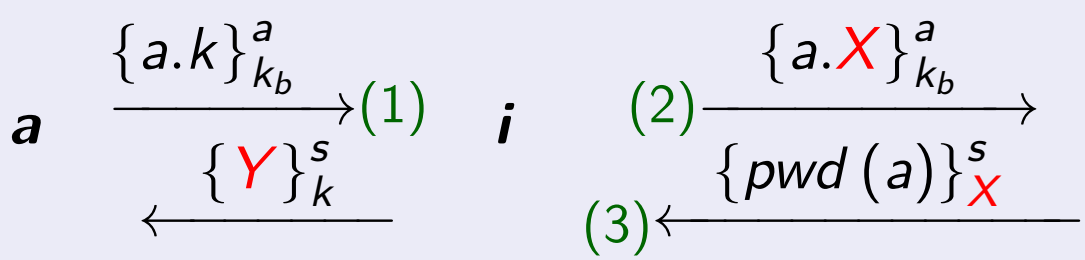
Constraint system

$$\begin{cases} i, a, b, k_b, \{a.k\}^a_{k_b} \rhd & \{a.X\}^a_{k_b} \\ i, a, b, k_b, \{a.k\}^a_{k_b}, \{pwd(a)\}^s_X \rhd & pwd(a) \end{cases}$$

One of the solutions: $\{X \mapsto i\}$

Reducing insecurity problem to satisfiability of deducibility constraints

$\boldsymbol{a}(A)$ asks to restore his password on $\boldsymbol{b}(B)$



Initial intruder's knowledge: i, a, b, k_b

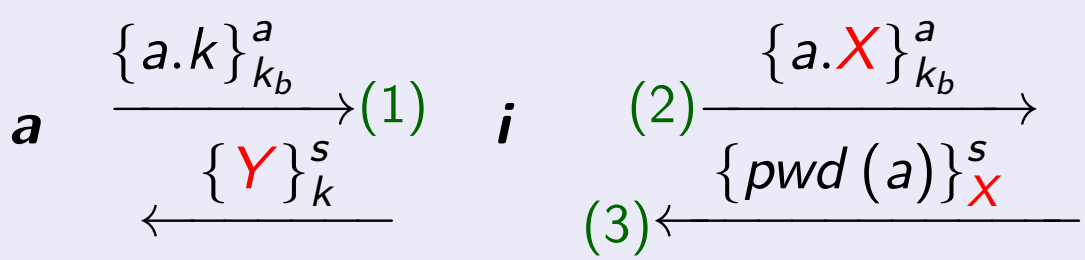
Constraint system

$$\begin{cases} i, a, b, k_b, \{a.k\}^a_{k_b} \rhd & \{a.X\}^a_{k_b} \\ i, a, b, k_b, \{a.k\}^a_{k_b}, \{pwd\,(a)\}^s_X \rhd & pwd\,(a) \end{cases}$$

One of the solutions: $\{X \mapsto i\}$

Reducing insecurity problem to satisfiability of deducibility constraints

$\boldsymbol{a}(A)$ asks to restore his password on $\boldsymbol{b}(B)$



Initial intruder's knowledge: i, a, b, k_b

Constraint system

$$\begin{cases} i, a, b, k_b, \{a.k\}^a_{k_b} \rhd & \{a.X\}^a_{k_b} \\ i, a, b, k_b, \{a.k\}^a_{k_b}, \{pwd\,(a)\}^s_X \rhd & pwd\,(a) \end{cases}$$

One of the solutions: $\{X \mapsto i\}$

Some formal definitions

Deducibility constraint system

$\{E_i \rhd t_i\}_{i=1,\ldots,n}$, where
E_i — finite set of terms, t_i — term for all i.

Derivability

Term t is derivable from set of terms E, if $t \in \mathrm{Der}(E)$, where $\mathrm{Der}(E)$ is a closure of E w.r.t. deduction rules.

Model/solution of $E \rhd t$

Ground substitution σ — model of $\{E_i \rhd t_i\}_{i=1,\ldots,n}$ iff $t_i\sigma \in \mathrm{Der}(E_i\sigma)$ for all i.

Well-formed constraint systems

$\{E_i \rhd t_i\}_{i=1,\ldots,n}$ is well-formed, iff it satisfies

- **Knowledge monotonicity**: $i < j \implies E_i \subseteq E_j$
- **Variable origination**:
  $x \in \mathrm{Vars}(E_i) \implies x$ occurs in some t_j, $j < i$.

Decidable, NP-complete (2001)

Later: algebraic properties

- XOR (2003) i.e. $((a \oplus b) \oplus c) = (a \oplus (b \oplus c))$, $a \oplus b = b \oplus a$ and $a \oplus a = 0$
- Modular exponentiation (2003/4) $a^1 = a$, $(a^b)^c = a^{b \times c}$, $\times$ is AC...
- Prefix rules (2005) $\{X.Y\}^s_K \to \{X\}^s_K$.
- Commutativity of public-key encryption (2003/4)
  $\left\{\{m\}^a_{k_1}\right\}^a_{k_2} = \left\{\{m\}^a_{k_2}\right\}^a_{k_1}$ (RSA)
- Combination of theories (2005)

  disjoint signatures OR bounded message depth, etc.

Contribution 1: Decidability modulo ACI

ACI: set behaviour

- Associative: $(a \bullet b) \bullet c = a \bullet (b \bullet c) = a \bullet b \bullet c$
- Commutative: $a \bullet b = b \bullet a$
- Idempotent: $(a \bullet a) = a$

Intruder's additional rules

$m_1, m_2, \ldots, m_n \rightarrow m_1 \bullet m_2 \bullet \ldots \bullet m_n$

$m_1 \bullet m_2 \bullet \ldots \bullet m_n \rightarrow m_i$ for all i

Modeling set of nodes in XML messaging

```
<data>
 <i><id>a</id><n>2</n></i>
 <i><id>b</id><n>1</n></i>
 <i><id>c</id><n>1</n></i>
 <i><id>d</id><n>3</n></i>
</data>
```

As a term:

$a.2 \bullet b.1 \bullet$
$c.1 \bullet d.3$

In protocol we can write:

$?c.N \bullet X$

Dolev Yao intruder is too powerful

Example

Peer-to-peer communications. Dolev-Yao intruder should be able to intercept messages between any two peers!

We present another intruder model
(which has DY intruder as a special case)

Multiple intruders. Example.

Some spy

uses weak places
in some network

Multiple intruders. Example.

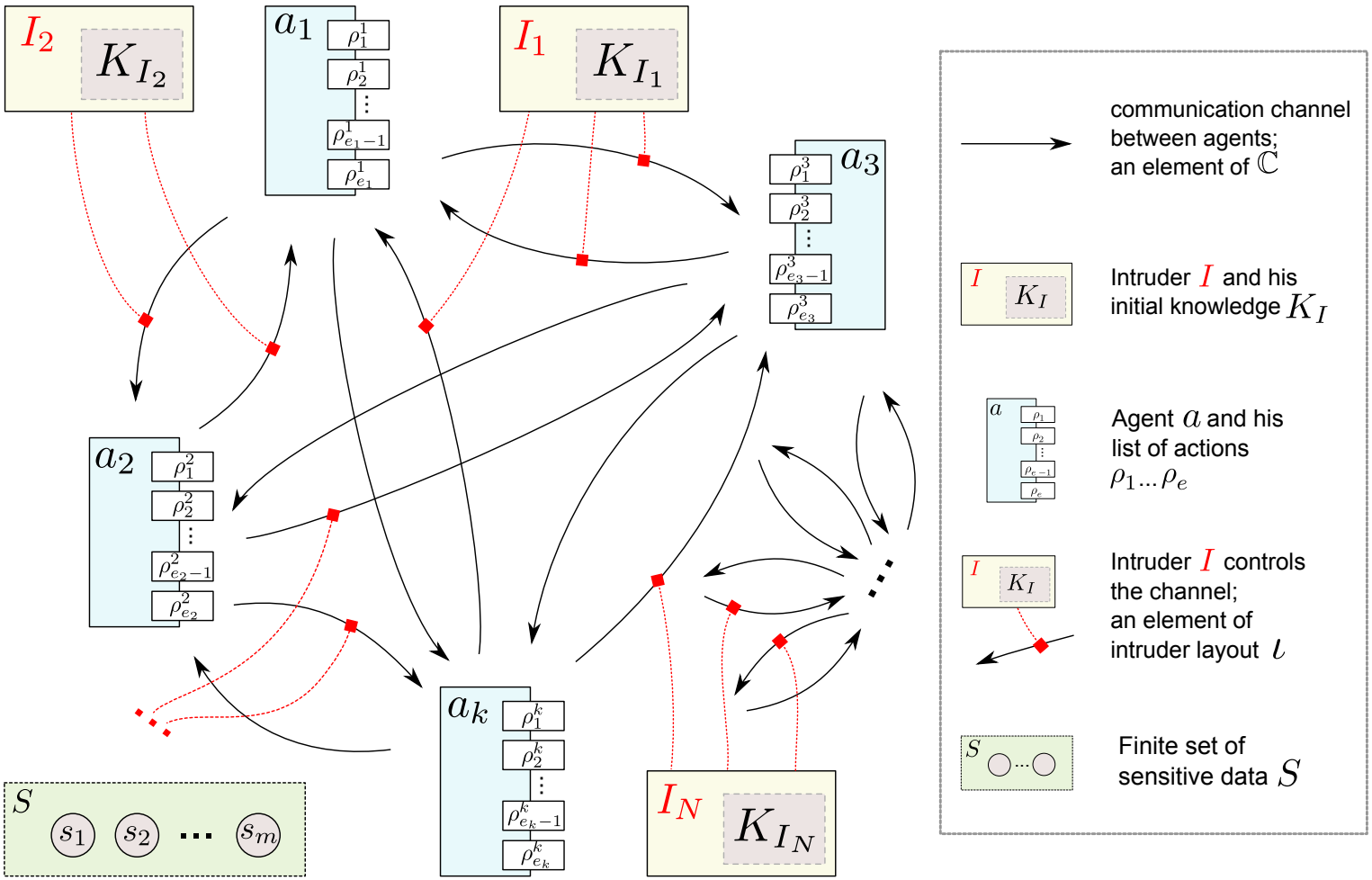
Some spy

uses weak places
in some network

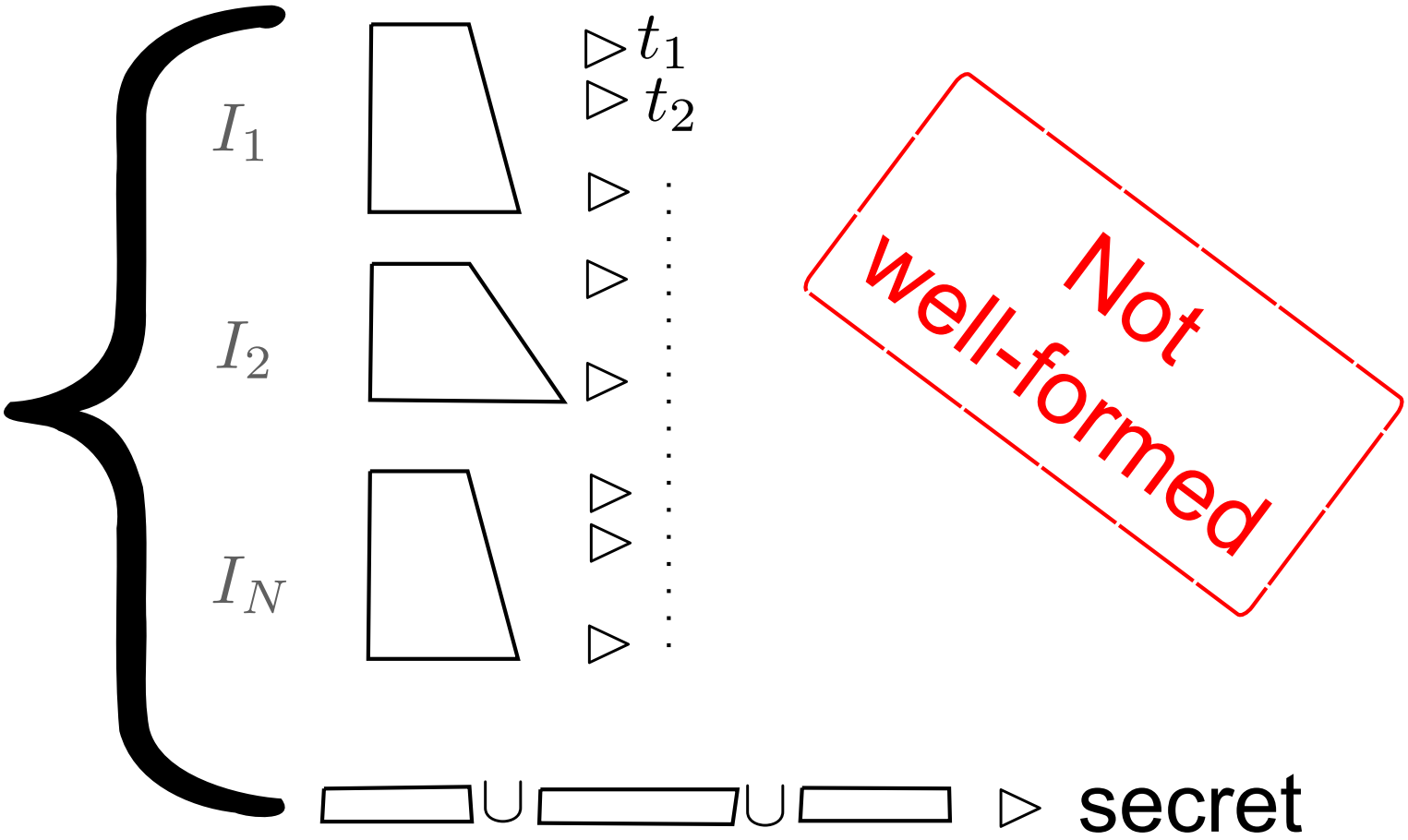
to implant his devices

Contribution 2: Multiple intruders model

- Multiple "local" intruders, different control domains
- Cannot communicate during attack, only after
- Secrecy is decidable, for protocol sessions analysis



Multiple intruders require general constraints



Moreover, variables produced by I_1 can appear in knowledge of I_2...

Contribution 3: Deciding general constraint systems

Satisfiability of "general" deducibility constraint systems

- Decidable for both DY and DY+ACI
- *NP*-complete.

Closest work

L.Mazaré (PhD thesis, 2006):
atomic keys, decidable for DY

Here:

- Complex symmetric keys
- ACI symbol

Decidability of general constraint systems

Key property

If $\mathcal{S}$ is satisfiable **then** there exists a solution σ that maps each variable to a set of non-variable subterms of $\mathcal{S}$ (and private keys...) instantiated with σ.

For the case of DY constraints (w/o ACI), instead of "ACI sets" we may use "pair of pairs".

Using this property we may find a bound on the size of such solution. Thus, bound a space for searching a solution.

General constraints are more complex to solve

Subterm deduction system

- composition rules: $x_1, \ldots, x_k \to f(x_1, \ldots, x_k)$
- decomposition rules pattern: $f(t_1, \ldots, t_m) \to s$, where s is a subterm of t_i for some i.

Subterm (convergent) deduction system

Input: Subterm (convergent) deduction system D and a constraint system C.

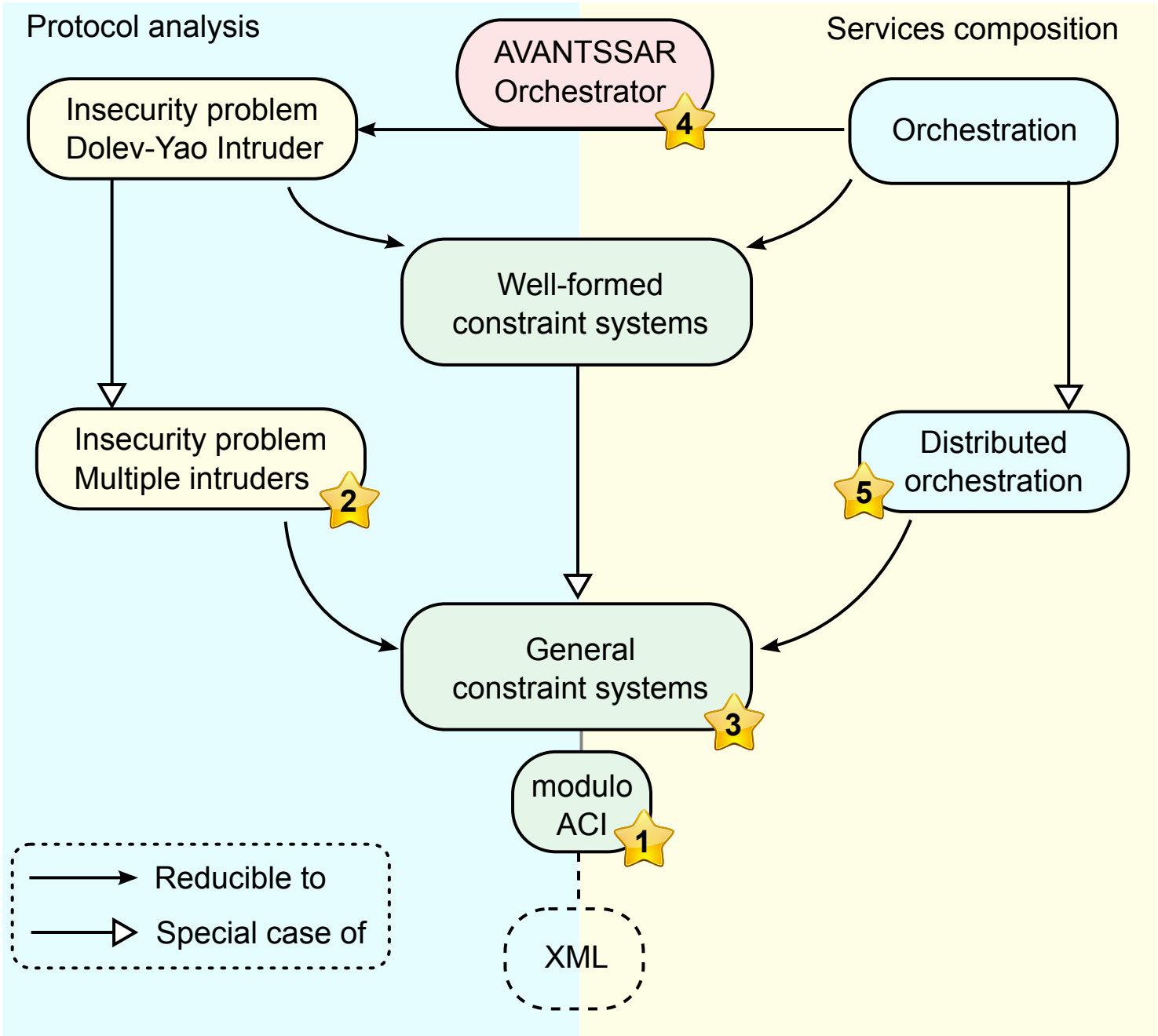
Question: is C satisfiable?

Undecidable
if either knowledge monotonicity or variable origination is not satisfied.

Decidable for well-formed constraints

- Subterm-convergent equational theories [Baudet '05]
- Later, Subterm-convergent deduction systems

We are here...


Protocol analysis
Services composition
AVANTSSAR Orchestrator
4
Insecurity problem Dolev-Yao Intruder
Orchestration
Well-formed constraint systems
Insecurity problem Multiple intruders
2
Distributed orchestration
5
General constraint systems
3
modulo ACI
1
XML
Reducible to
Special case of

Web service

is a software system with machine-processable interface.

Web Service is a black box with...

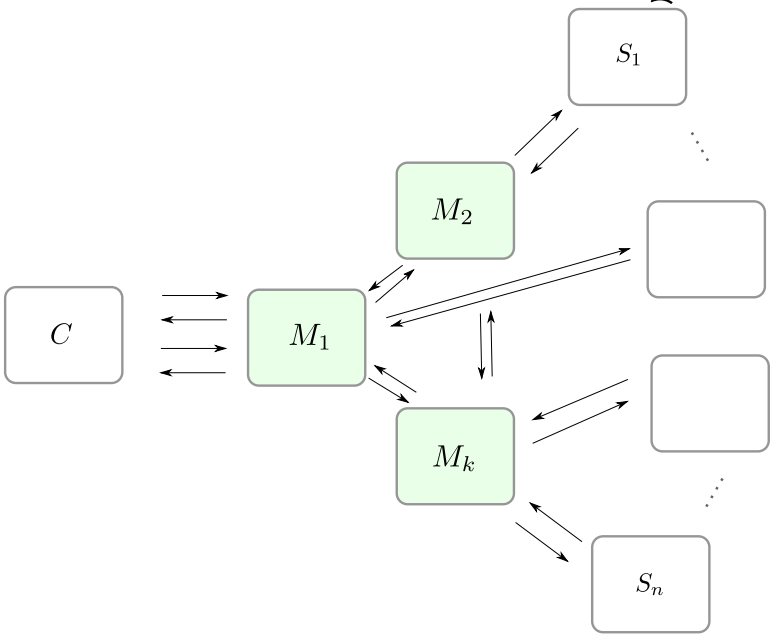
- **Interface** (WSDL): set of operations, operation is a pair "receive-send".
- **Usage scenario** (e.g. WS-BPEL): sequence to follow, Moreover, one may need to invoke second operation with a specific value used in the first one.
- **Security policies** (WS-SecurityPolicy). E.g. a given part of the input of a given operation must be encrypted/signed with given key...

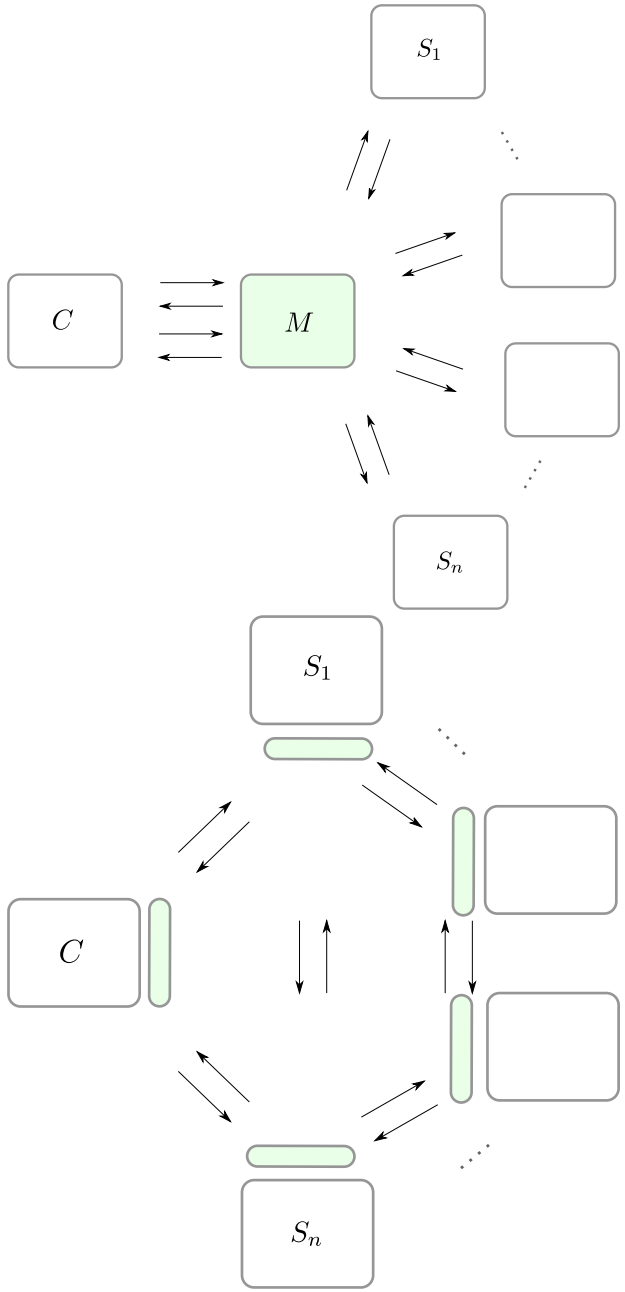
... Exactly as an instance of cryptographic protocol role ...

Web Services use XML as basis for the interface and communications.

Web Services composition options

- Orchestration: a central entity, mediator (orchestrator) ⟶
- Choreography: services communicate directly with each other following a global strategy ↘
- Distributed orchestration: mediator is decentralized ↙





Web Services orchestration problem

Given a finite set of available services

Each in a form of a sequence of operations on which the security policies are already applied.

Client

- Sequence of requests with expected responses.

To build a *mediator*

- Is a new "executable" Web Service, white box
- Satisfying the client's requests
- Reusing existing Web Services
- Adapting messages
- Have initial knowledge
  (e.g. account information, public keys)

Automata-based approaches (e.g. Roman model)

Available services

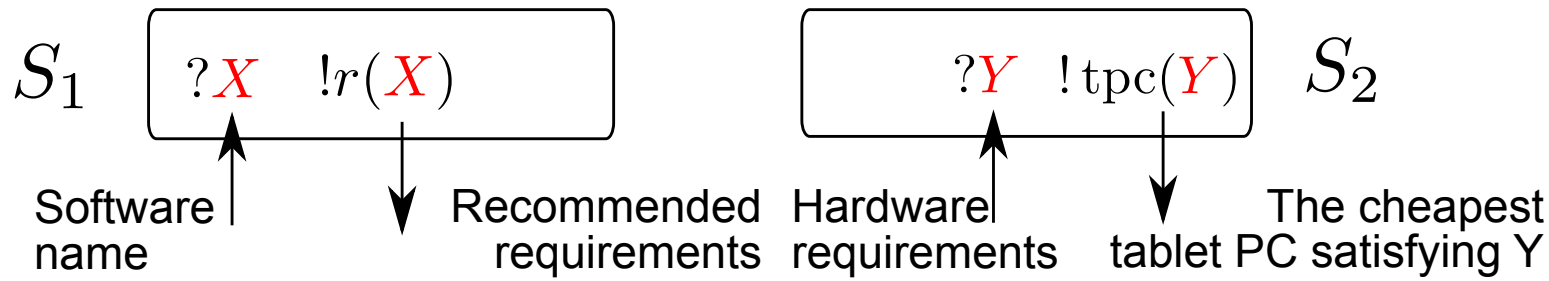
- State machines
- Transitions are labeled with service's operations

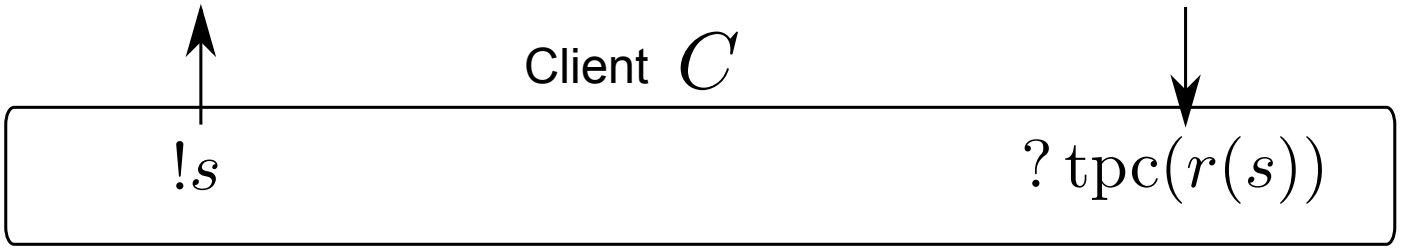
Orchestration problem

Simulate the behaviour of a given target service by delegating operation invocations to the community of available services

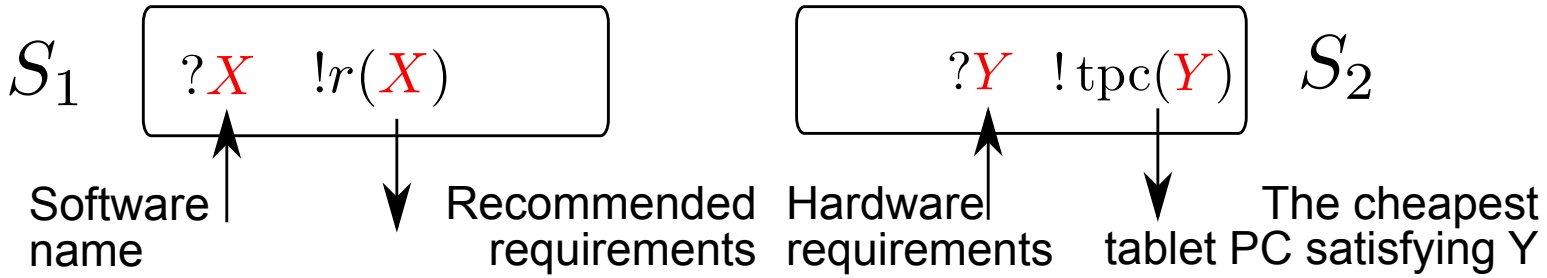
- Usually message structure is not considered (working on the level of operations)
- Security policies are not taken into account

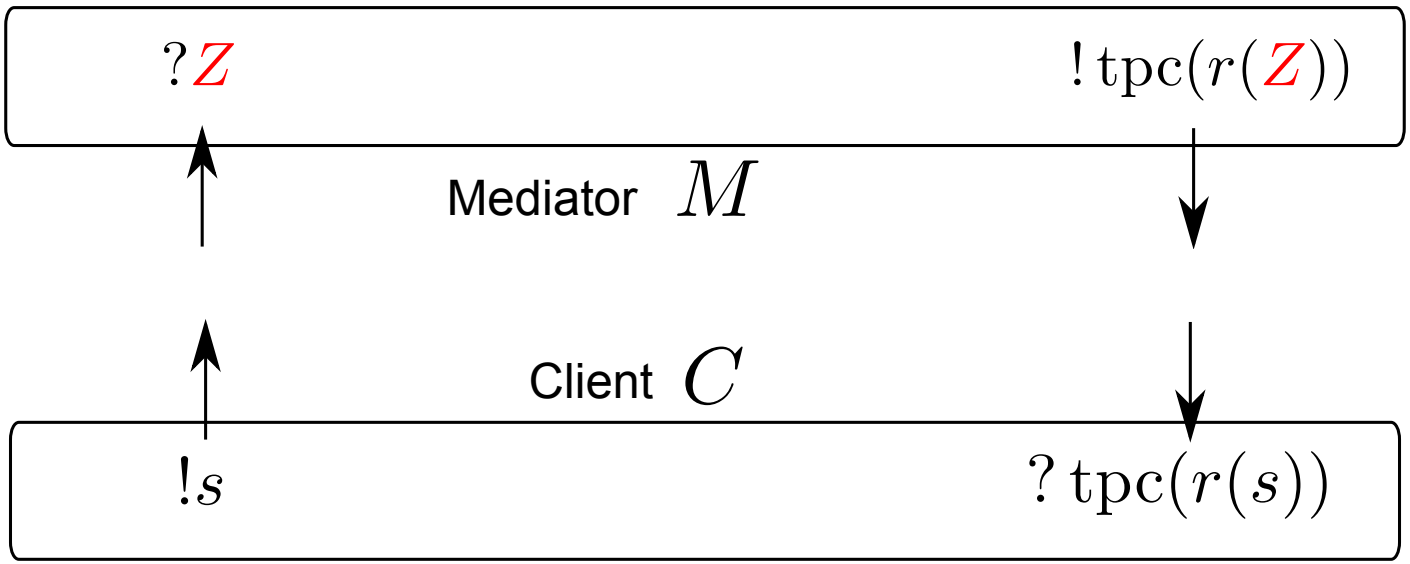
Web Services Example



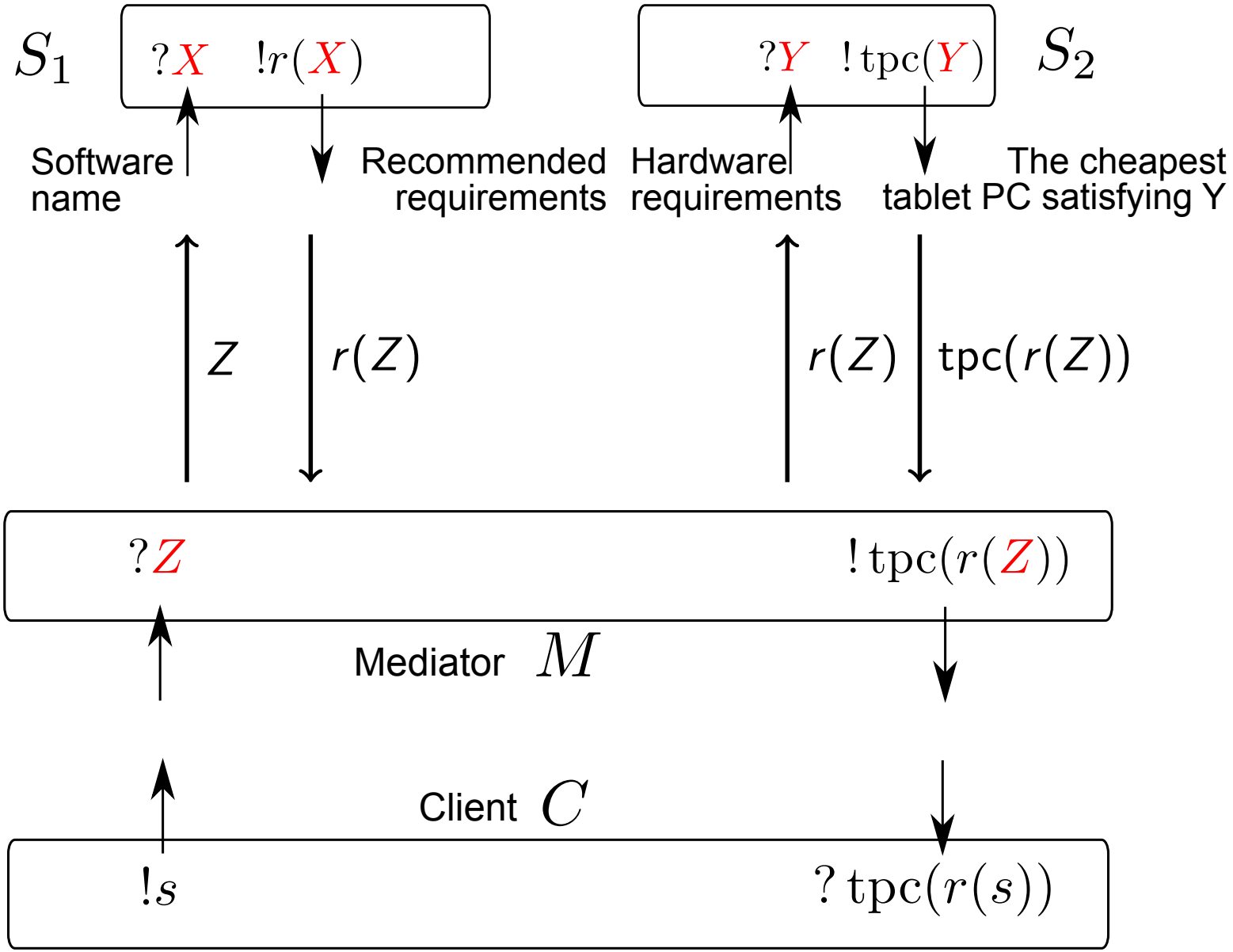


Web Services Example

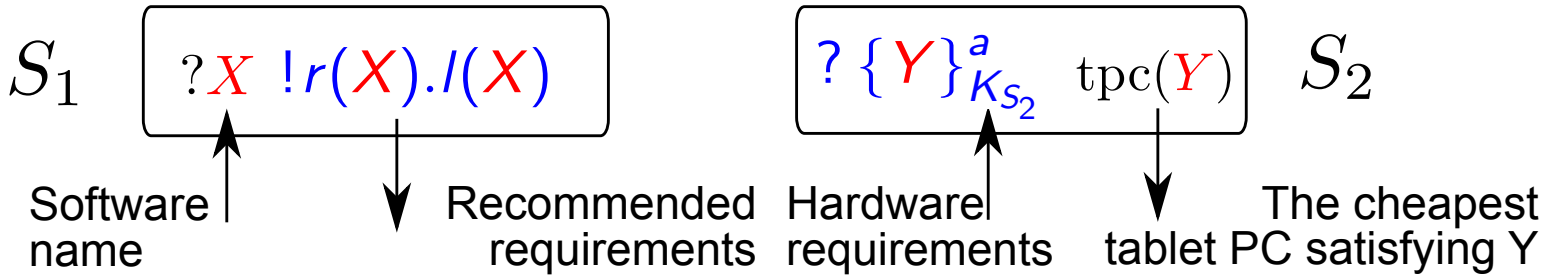


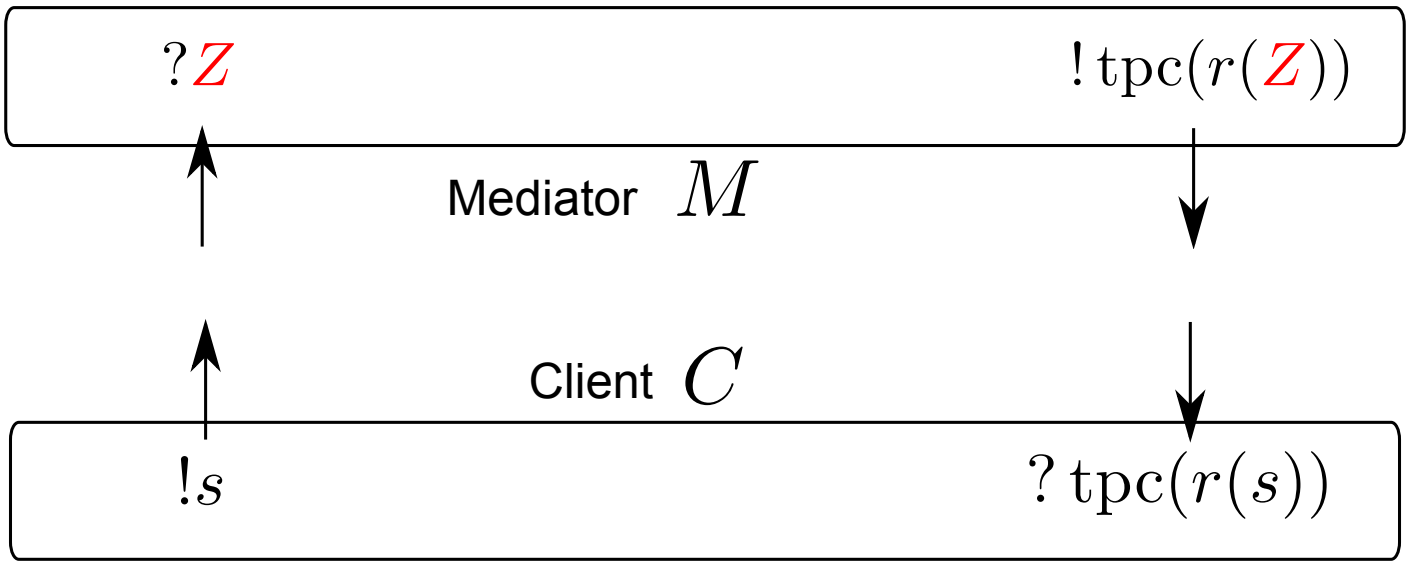


Web Services Example

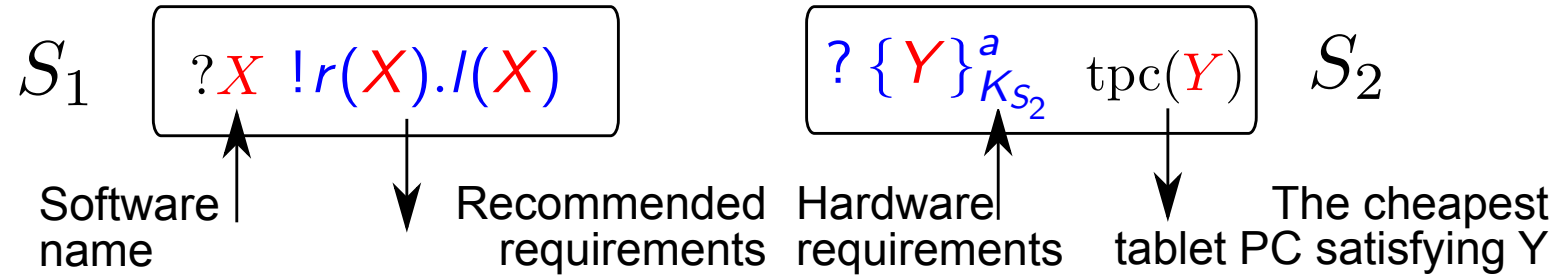


Web Services Example

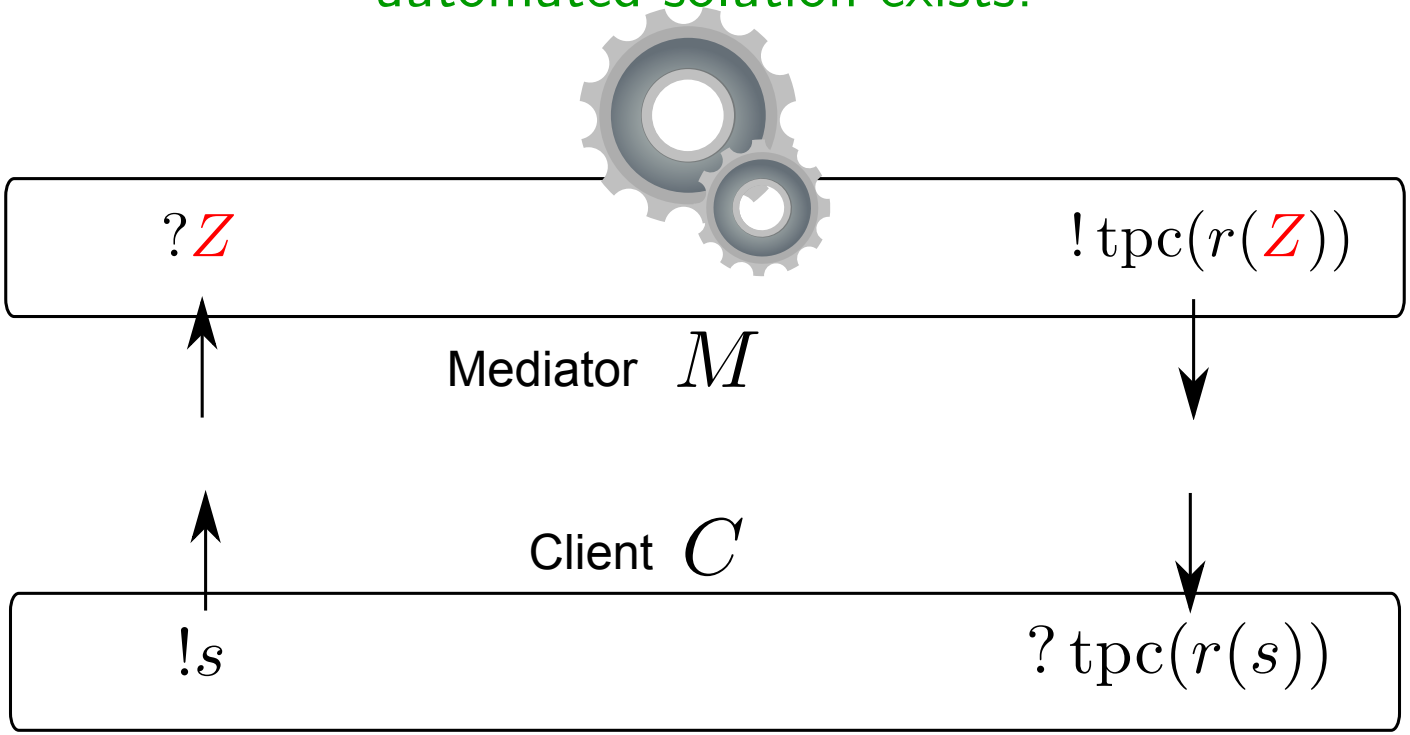




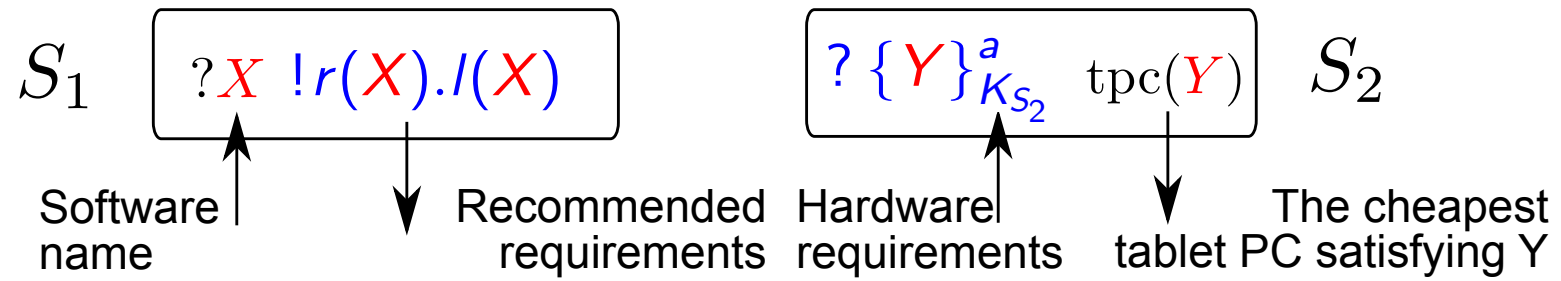
Web Services Example



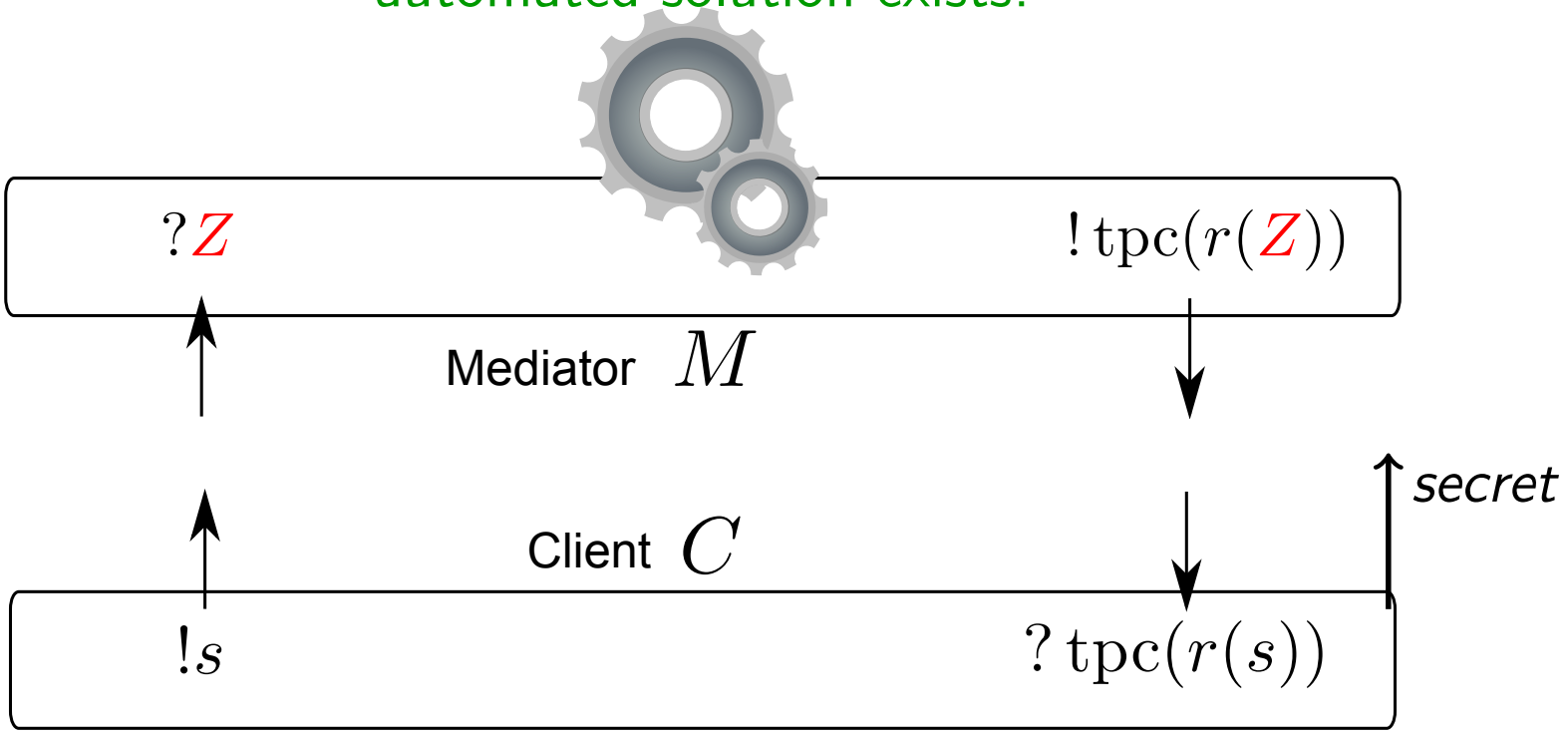
The number of invocations is bounded $\Longrightarrow$
automated solution exists!



Web Services Example



The number of invocations is bounded $\Longrightarrow$
automated solution exists!



Contribution 4: Tool for automatic orchestration

If the number of interactions is bounded:

- Build deducibility constraints for the Mediator M
- Can solve the constraint system $\Longrightarrow$ can implement M

WS Orchestration vs Protocol Analysis

Services	Protocols
Available service/Client	Protocol role
Mediator	Intruder
Final state of Client	Attack state (secret emitted)

Implemented as AVANTSSAR Orchestrator

- reused a tool (CL-AtSe) for protocol analysis
- `http://avantssar.eu`; `http://cassis.loria.fr/`, $\rightarrow$ AVANTSSAR Orchestrator

About the approach

Advantages

- Allows fully automatic procedure
- Rich message adaptation abilities
- Take into account security primitives.

Assessment of the tool

- Digital Contract Signing (OpenTrust)
- Public Bidding (OpenTrust)
- Car Registration Process (Siemens AG)

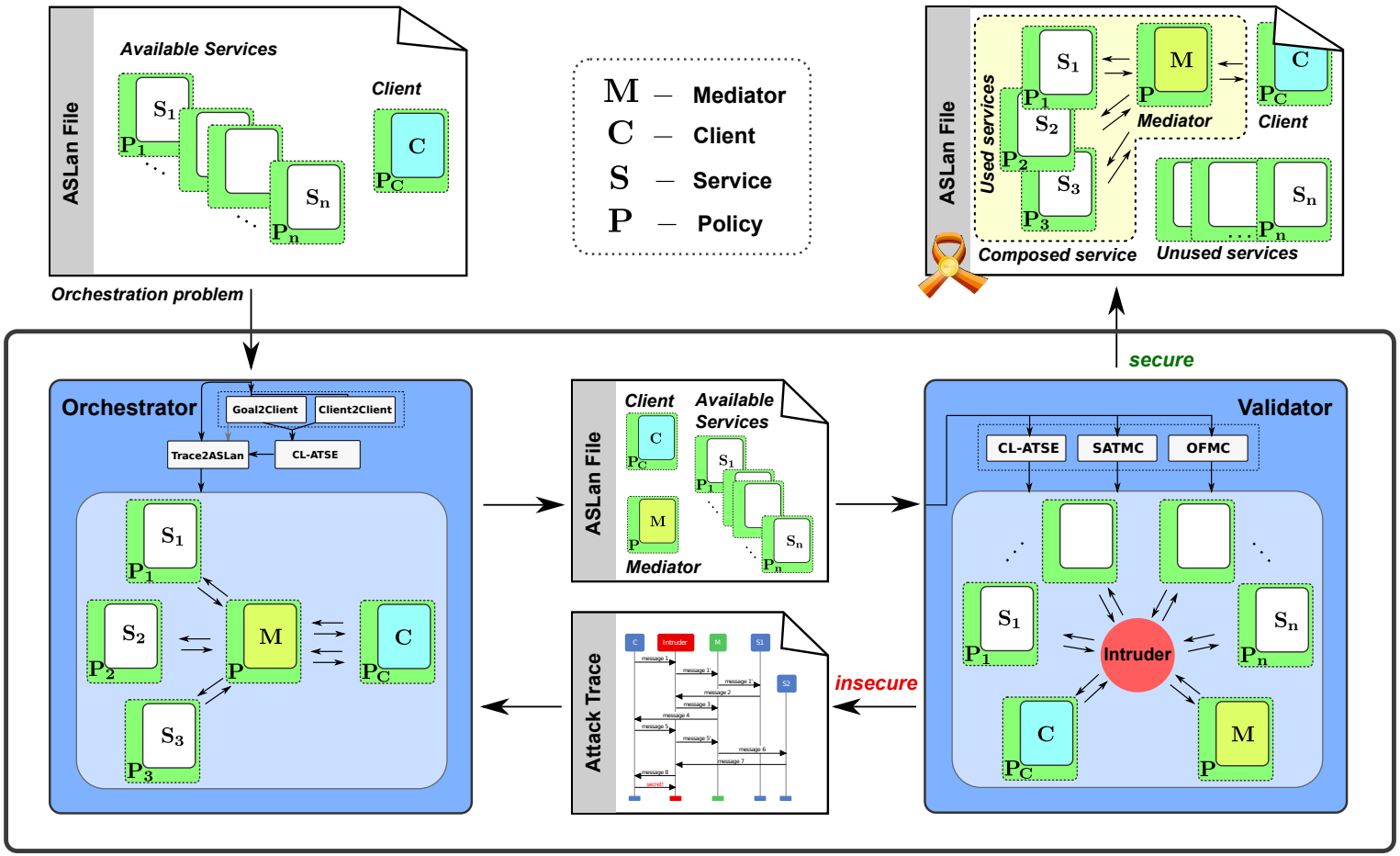
Disadvantages

- Limit on number of invocations (with all consequences)

AVANTSSAR platform, ASLan

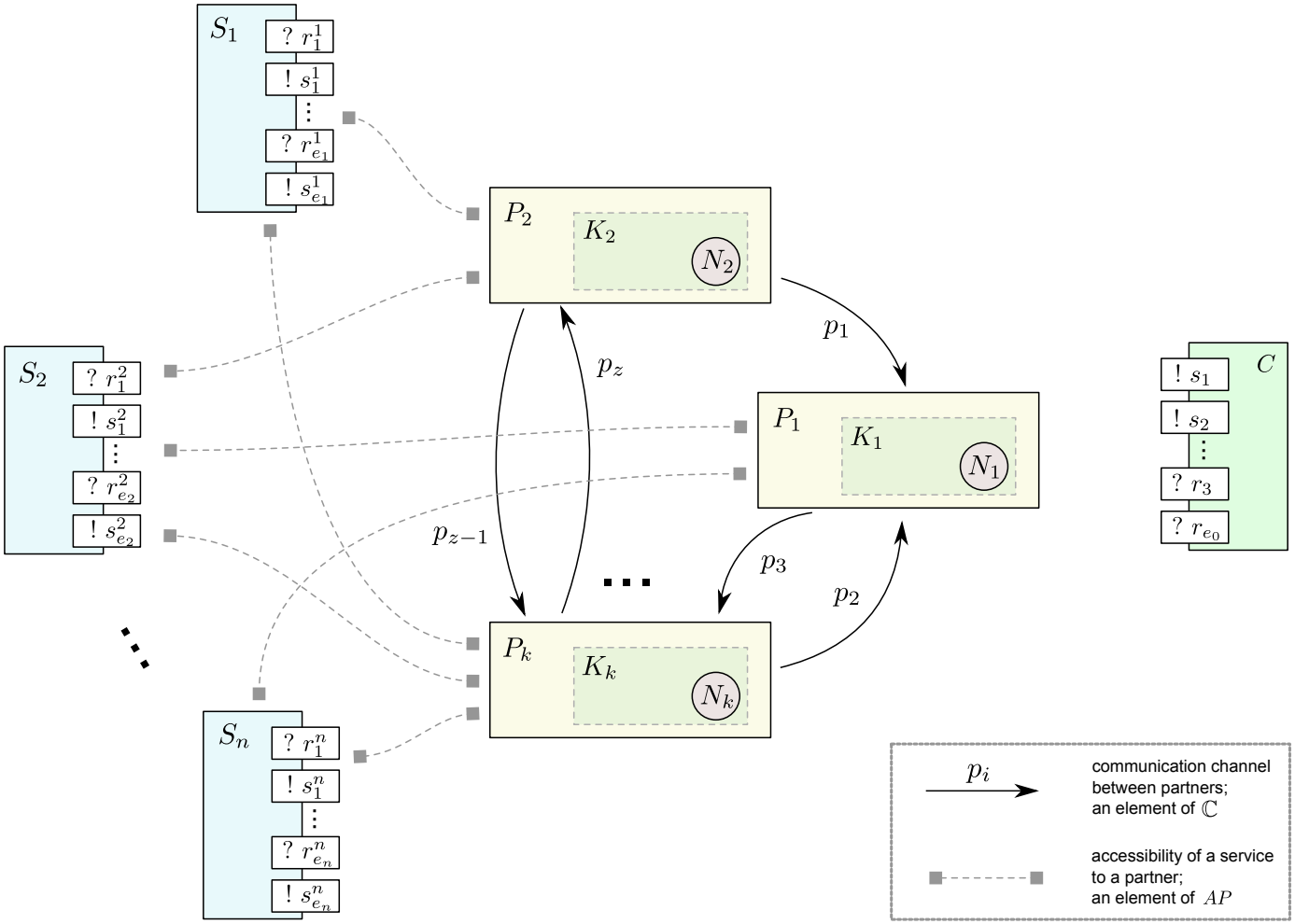
A transition in ASLan language

```
step step_0 (A,I,J) :=
  state_Adder(A,1,dummy_msg,dummy_msg).
  iknows(pair(I,J))
    =>    state_Adder(A,2,I,J).
          iknows(apply(plus,pair(I,J)))
```



AVANTSSAR Validation Platform

Contribution 5: Distributed orchestration model



- Available services S_i with list of actions
- Partners P_j with knowledge K_j, confidential data N_j
- Communication channels with message patterns p_l
- Accessibility of services (e.g. private services for organizations)

Non-disclosure policy

Problem

- Partner P_i represents some organization.
- P_i possesses some data (initial) K_i.
- K_i contains confidential information N_i.
- P_i does not want to send such messages to P_j that would allow P_j to obtain any element of N_i.
- Still, N_i can be used for WS invocations.

Direct approach

Use negative constraints, i.e. $E \not\triangleright n$ (E— knowledge of P_j, n — confidential data of P_i)

Sufficient condition we use

Confidential data n should not appear as a subterm in messages sent to partners.

Deciding existence of partner-mediators

- **Reducing** distributed orchestration to general constraint systems:
  - E.g., *Partner P_i invokes service's operation*:
    Current knowledge $K_i \rhd$ operation input
    $K_i := K_i \cup$ operation output
- **Non-disclosure conditions**: For every step $P_i \to P_j : t$ we must ensure $\mathrm{Sub}(t) \cap N_i = \emptyset$

Extending the general constraints satisfiability procedure

If exists solution for deducibility constraint system that satisfies non-disclosure condition, **then** there exists one with bounded size which satisfies both constraint system and non-disclosure condition.

Summary

Contributions

Deducibility constraints

- ACI symbol in deducibility constraints
- Relaxing "well-formedness"
- Complexity class *NP*-complete

Protocol analysis

- Multiple non-communicating intruders model
- Decidability for secrecy problem
- Modeling sets of XML nodes

Web services composition

- AVANTSSAR Orchestrator tool
- Model for a distributed orchestration with non-disclosure policy
- Automatic decision procedure for mediators synthesis

Summary

Research directions

Theoretical

- Allowing negation in general constraint systems
- More algebraic properties for general constraints (e.g. XOR)
- Remove atomicity for public keys
- Web Services composition with unbounded number of invocations

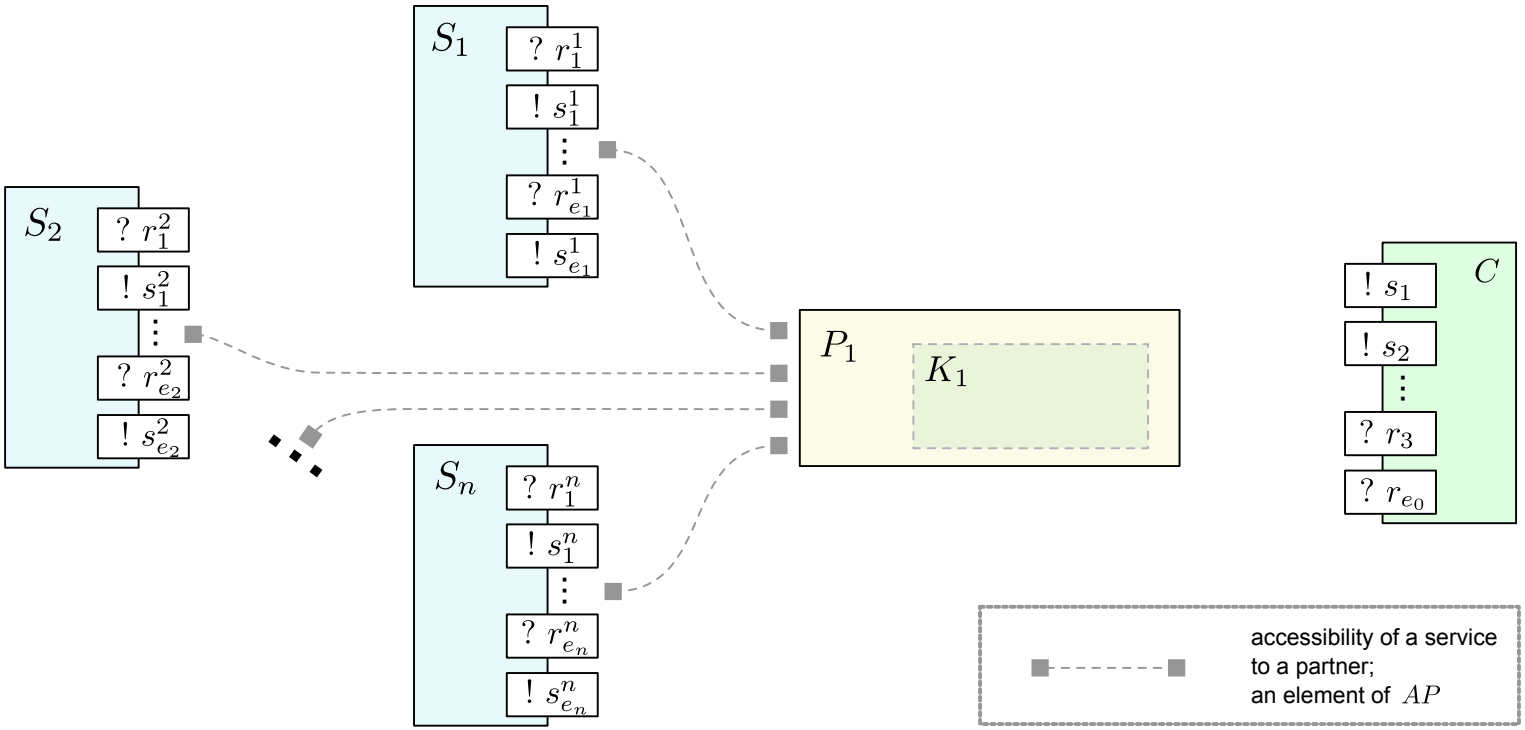
Practical

- Explicit link (with a tool) from standards to model
- Effective implementation of general constraints satisfiability

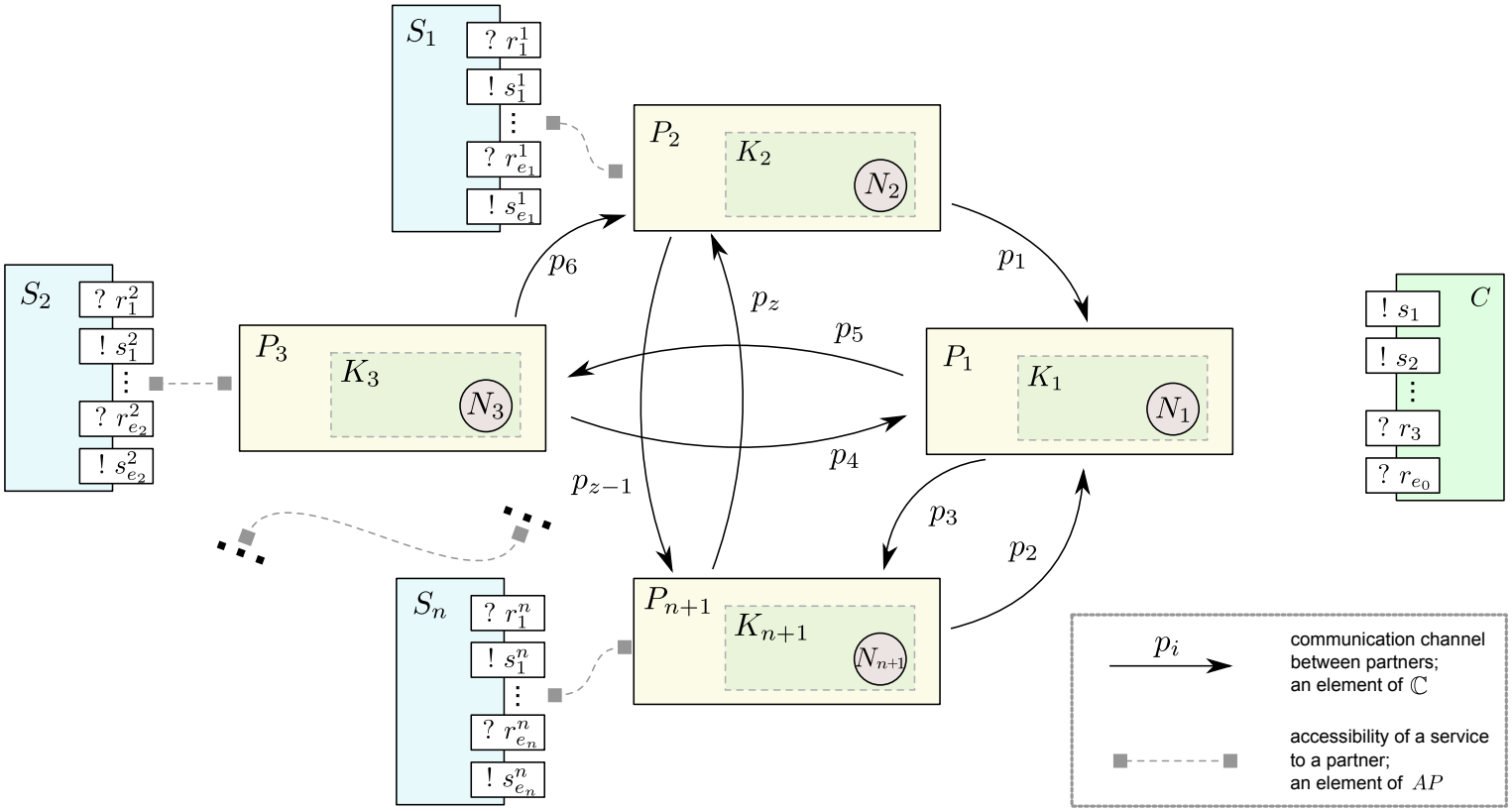
Thank you for your attention

Might be useful

Special case: orchestration



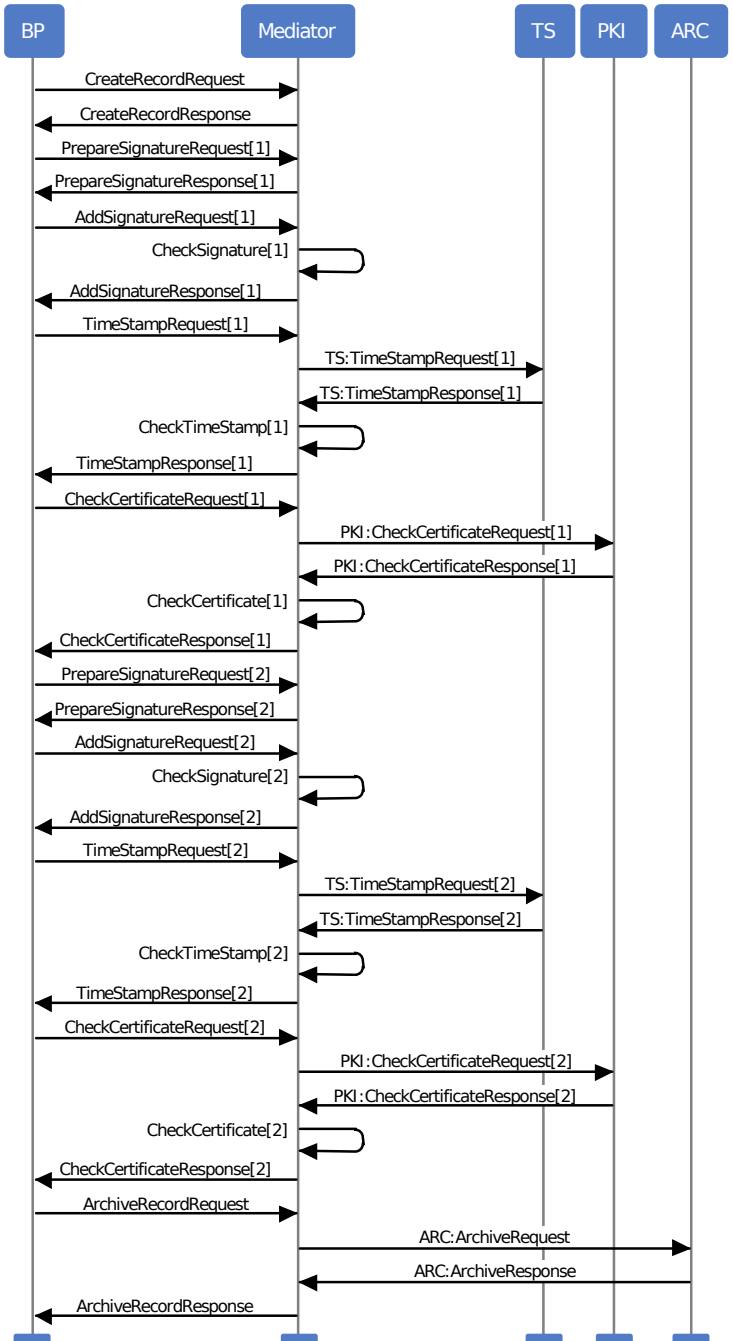
Special case: choreography



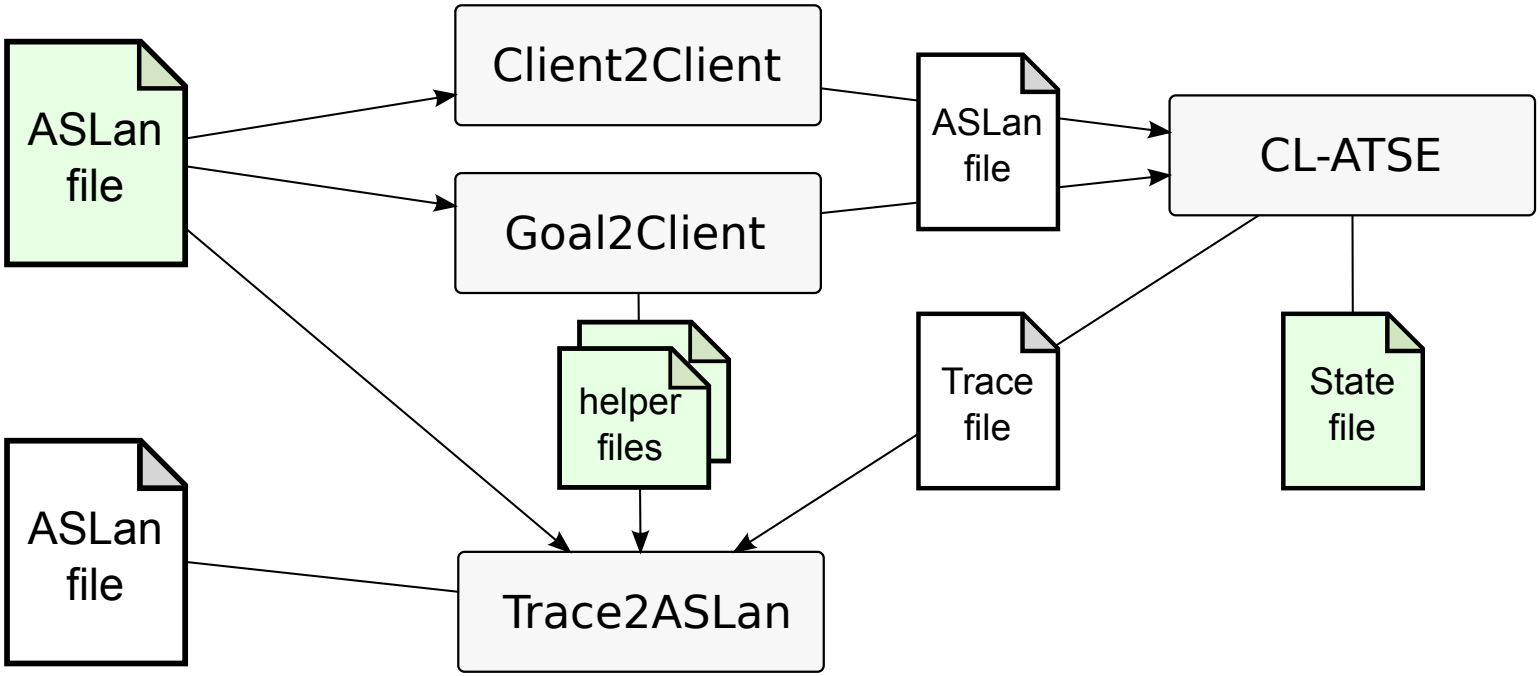
Benefits of considering multiple intruders

- Communications resistant to DY intruder require more exigent protocols.
  Need a **compromise** between resources (including responsiveness) and security.
- An organization that knows the weak links (easy for physical access) **can verify** whether such multiple intruders may damage the confidentiality of their data.

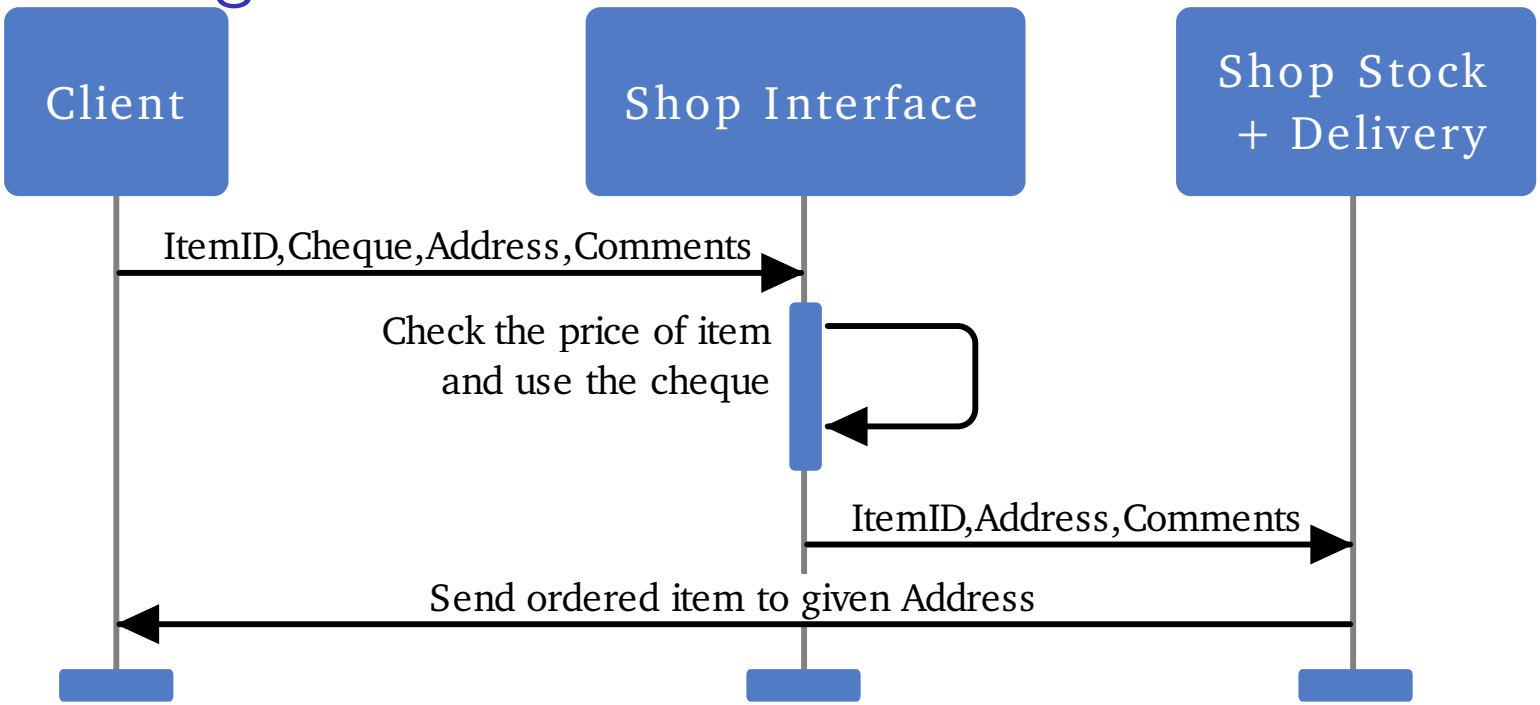
Digital Contract Signing



AVANTSSAR Orchestrator scheme



Ordering scenario



Bad request

```
<ItemID>simple</ItemID>
<Cheque>cheque5</Cheque>
<Address>addr</Address>
<Comments>cmnts</Comments>
<ItemID>gilded</ItemID>
```

```
cmnts =
    </Comments>
<ItemID>
    gilded
</ItemID>
<Comments>
```