

# Developing Secure Networked Web-based Systems Using MBRA and UMLsec

Siv Hilde Houmb and Jan Jürjens

Department of Computer and Information Science, NTNU, Norway



sivhoumb@idi.ntnu.no

and

Software & Systems Engineering  
Informatics, TU Munich, Germany



juerjens@in.tum.de

<http://www.jurjens.de/jan>



# Outline

---

- Motivation
- UMLsec
- MBRA
- Example



# Motivation

---

- High quality development of critical systems (dependable, security-critical, real-time,...) is **difficult**.
- Many systems developed and deployed do **not** satisfy their quality/non-functional requirements, sometimes with spectacular failures.

# Quality vs. Cost

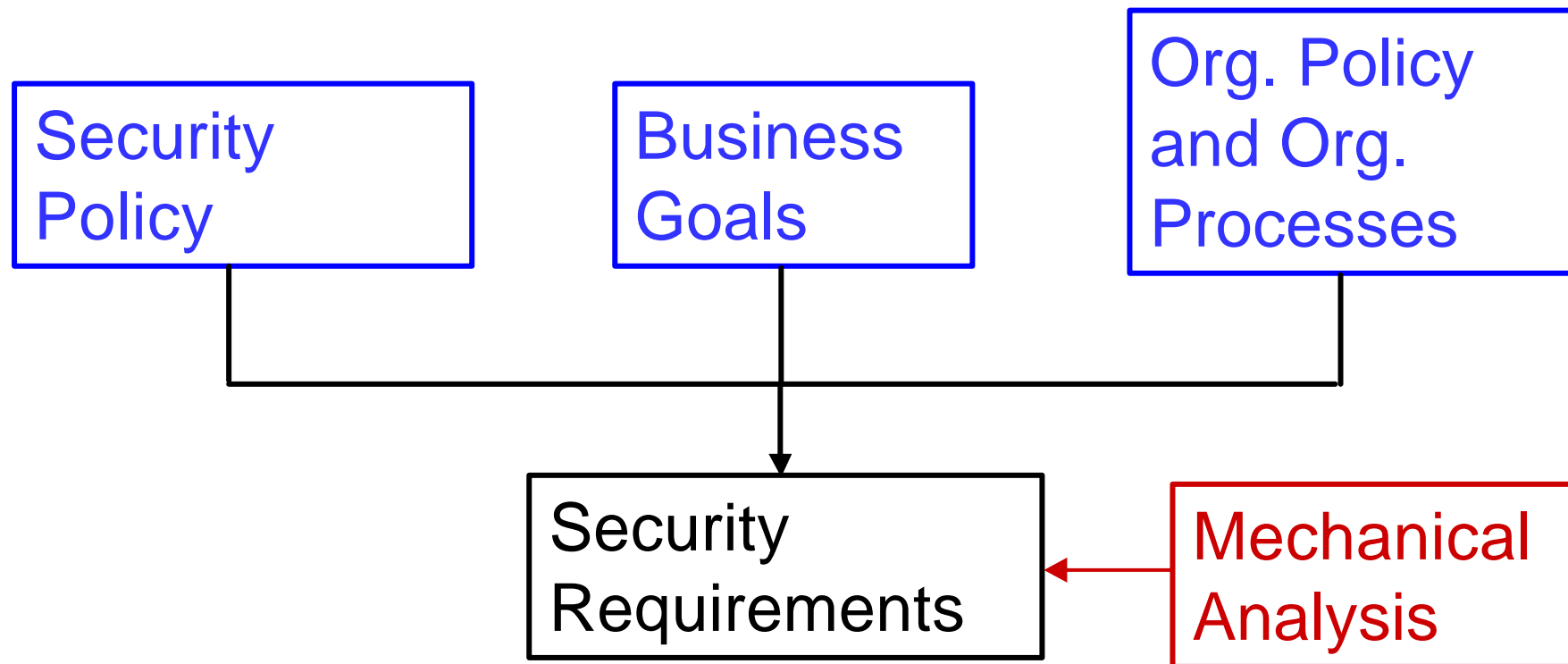
---

- **Correct level** of security
  - Quality vs. cost as a trade-off
- **Security** in conflict with **cost**
- Thorough methods of system design not used if too **expensive** or not efficient enough (time is money and one needs to deliver)
- Security as an **integrated** part of the development (trade-off)
  - Security not as an afterthought
  - Consider both functional and **security requirements**

# Overview of the Approach

---

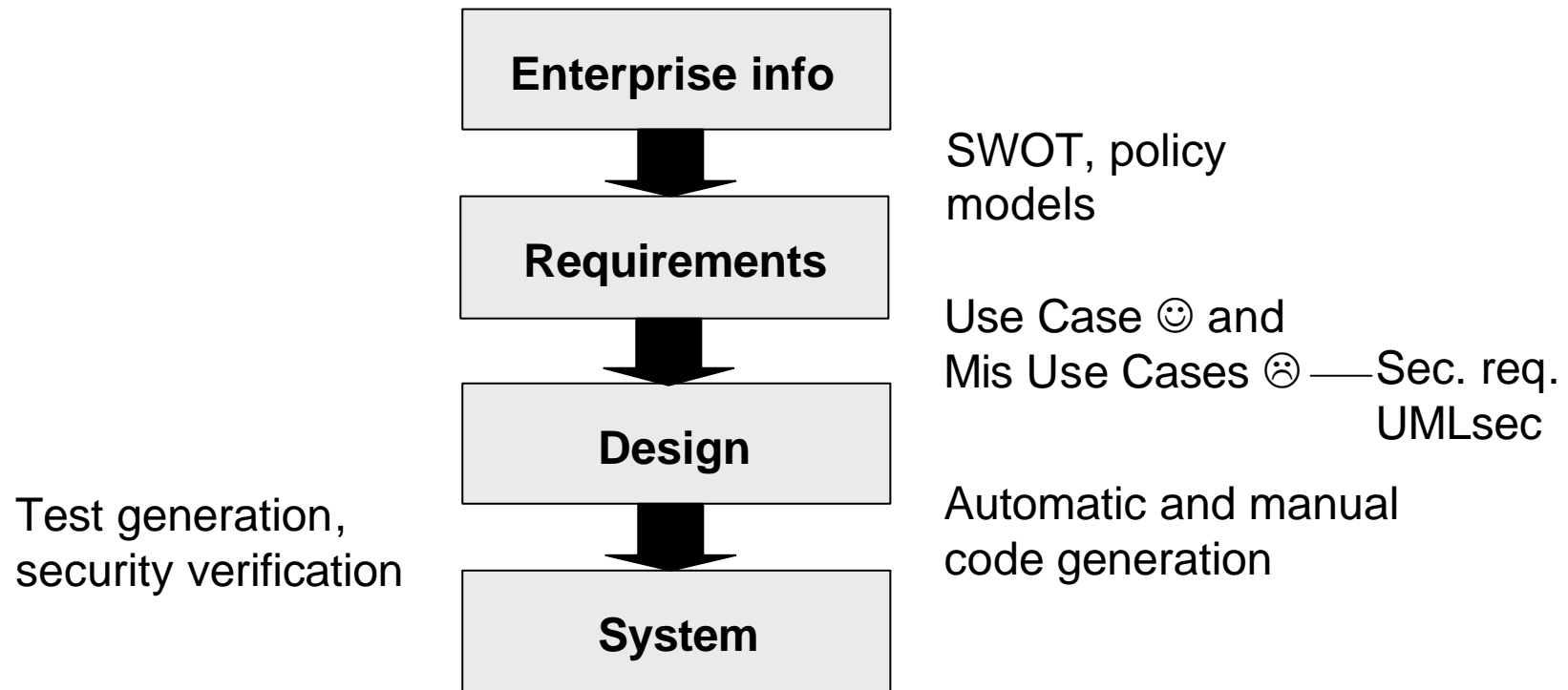
- Two levels:
  - Enterprise level ➡ CORAS
  - Technical level ➡ UMLsec



# Model Driven Development (MDD)

---

Goal: ease **transition** from human **ideas** to executable **systems**



The layer bellow is a realisation of the layer above

---

# Using UML

---

Unified Modeling Language (UML):

- **visual** modeling language
- different **views** on a system
- allows for a high degree of **abstraction**
- de-facto industry **standard** (OMG)
- standard **extension** mechanisms

# UML Extension Mechanisms

---

- Stereotype: **specialised** model element using `<<label>>`
- Tagged value: **attach** `{tag=value}` pair to stereotyped element
- Constraint: **refine** semantics of stereotyped element
- Profile: **a particular set of stereotypes, tagged values and constraints**

# UMLsec

---

UMLsec: extension for **secure systems** development.

- evaluate UML specifications of **vulnerabilities**
- encapsulate security engineering **patterns**
- also for developers **not specialised** in security
- security from **early** design phases, in system **context**
- make certification **cost-effective** (e.g. **Common Criteria**)

# The UMLsec profile

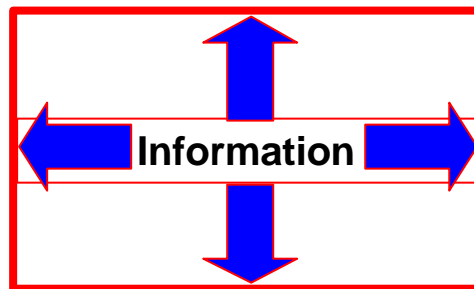
---

- **Recurring** security requirements as stereotypes with tags (secrecy, integrity,...)
- Associated constraints to **evaluate** model, indicate possible **vulnerabilities**
- Ensures that stated security requirements **enforce** given security policy
- Ensures that UML specification **provides** requirements

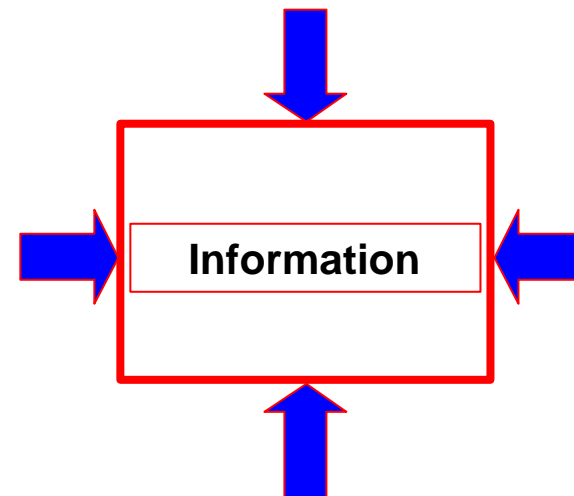
# Basic Security Requirements

---

## Secrecy



## Integrity



# UMLsec Stereotypes, Tags and Constraints

Stereotype	Base Class	Tags	Constraints	Description
secrecy	dependency			assumes secrecy
integrity	dependency			assumes integrity
critical	object	secrecy, integrity		critical object
secure links	subsystem		dependency security matched by links	enforces secure communication links
secure dependency	subsystem		<<call>, <<send>>	structural interaction
data security	subsystem		provides data security	data security
fair exchange	subsystem	start, stop	after start eventually reach stop	enforce fair exchange

<<Internet>>, <<encrypted>>, ...

---

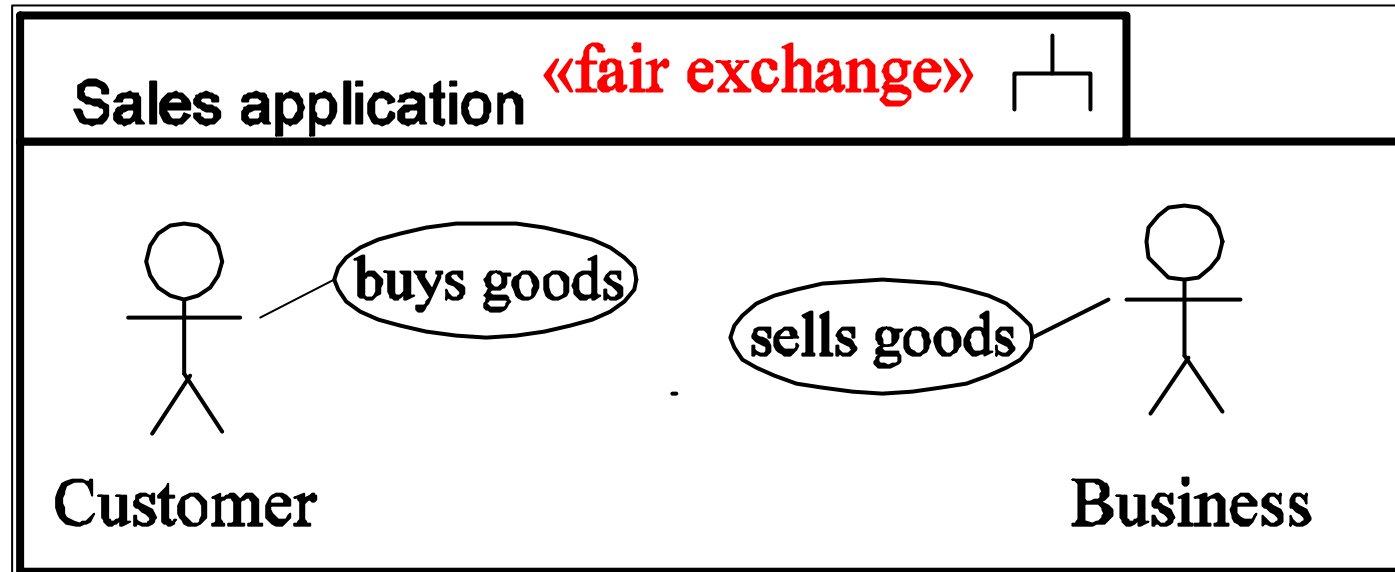
Kinds of communication **links** between respective system **nodes**.

For adversary type  $A$ , stereotype  $s$ , have set  $\text{Threats}_A(s) ? \{\text{delete, read, insert, access}\}$  of actions that adversaries are capable of.

Default attacker:

Stereotype	$\text{Threats}_{\text{default}}()$
Internet	{delete, read, insert}
encrypted	{delete}
LAN	$\emptyset$
smart card	$\emptyset$

# Requirements with use case diagrams



Capture security requirements  
in use case diagrams.

Constraint: need to appear in corresponding  
activity or sequence diagram.

# <<fair exchange>>

---

Ensures generic **fair exchange** condition.

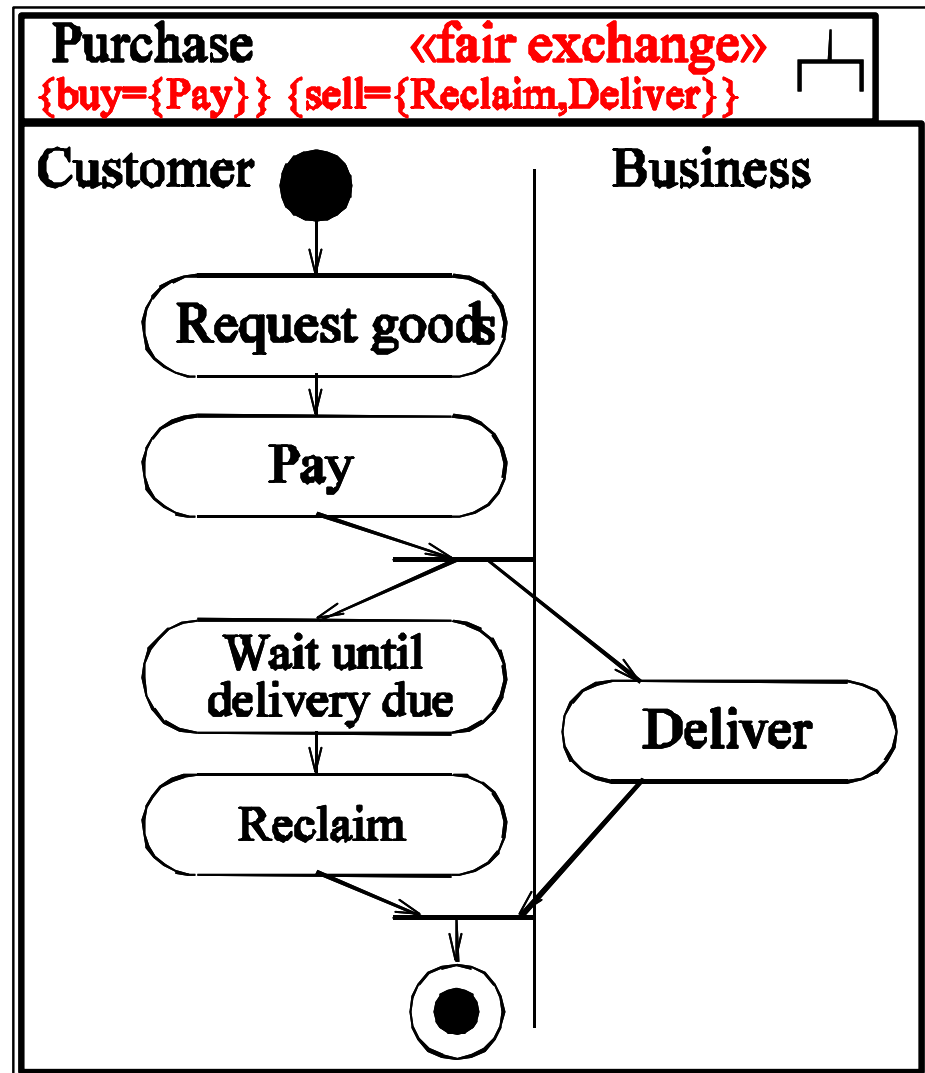
Constraint: after a **{buy}** state in activity diagram is reached, one will eventually reach **{sell}** state.

(Cannot be ensured for systems that an attacker can stop completely.)

# Example <<fair exchange>>

Customer buys goods from a business.

Fair exchange means: after payment, customer is eventually either **delivered** good or able to **reclaim** payment.



# Security Analysis

---

- Model classes of **adversaries**
- May **attack** different parts of the system according to threat scenarios
- Example: **insider** attacker may intercept communication links in LAN
- To evaluate security of specification, simulate jointly with adversary model

# Applications

---

- Common Electronic Purse Specifications
- Analysis of multi-layer security protocol for web application of major German bank
- Analysis of SAP access control configurations for major German bank
- Risk analysis of critical business processes (for Basel II / KontraG)
- ...

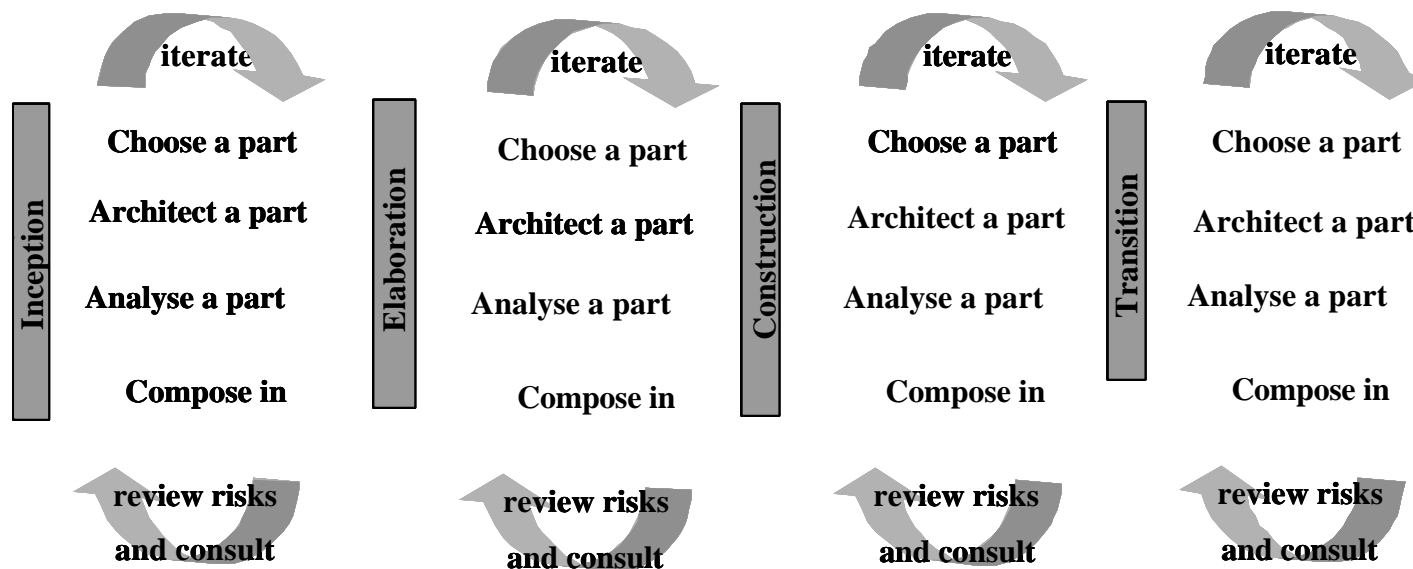
# Model Based Risk Assessment

---

- **Integrated** process
  - Make use of system models both as system description and as input to risk assessment
- Support **reuse** (and ease the updating of system description during and after assessment)
- **Cost efficient** (at the right time for the right cost)
- Precise specification of security requirements using UMLsec

# Integrate SD and RM process (1)

- Emphasise on handling risk at the **right time** for the **right price** (as early as possible)



- Make use of MBRA and use models both to **describe the system** and **as input to RA**

# MBRA for Networked Enterprises

---

## Sub-process 1: Identify Enterprises

- Activity 1.1: Identify and describe each enterprise
- Activity 1.2: Identify and describe the security policy for each enterprise
- Activity 1.3: Identify and describe security requirements and existing security mechanism for each enterprise using UMLsec
- Activity 1.4: Describe the set of enterprises contained in the networked enterprise
- Activity 1.5: Describe the common security policy of the networked enterprise
- Activity 1.6: Specify the set of common security requirements based on the common security policy using UMLsec

## Sub-process 2: Identify Relationship between enterprises

- Activity 2.1: Identify and describe relationship between each enterprise in the networked enterprise
- Activity 2.2: Identify and describe the distribution of responsibility among the enterprises
- Activity 2.3: Identify stakeholders and specify in which enterprise they belong

## Sub-process 3: Identify Context

- Activity 3.1: Identify areas of relevance
- Activity 3.2: Identify and value assets
- Activity 3.3: Identify policies and evaluation criteria
- Activity 3.4: Approval

## Sub-process 4: Identify Risks

- Activity 4.1: Identify threats to assets
- Activity 4.2: Identify vulnerabilities of assets
- Activity 4.3: Document unwanted incidents

## Sub-process 5: Analyse Risks

- Activity 5.1: Consequence evaluation
- Activity 5.2: Frequency evaluation

## Sub-process 6: Risk Evaluation

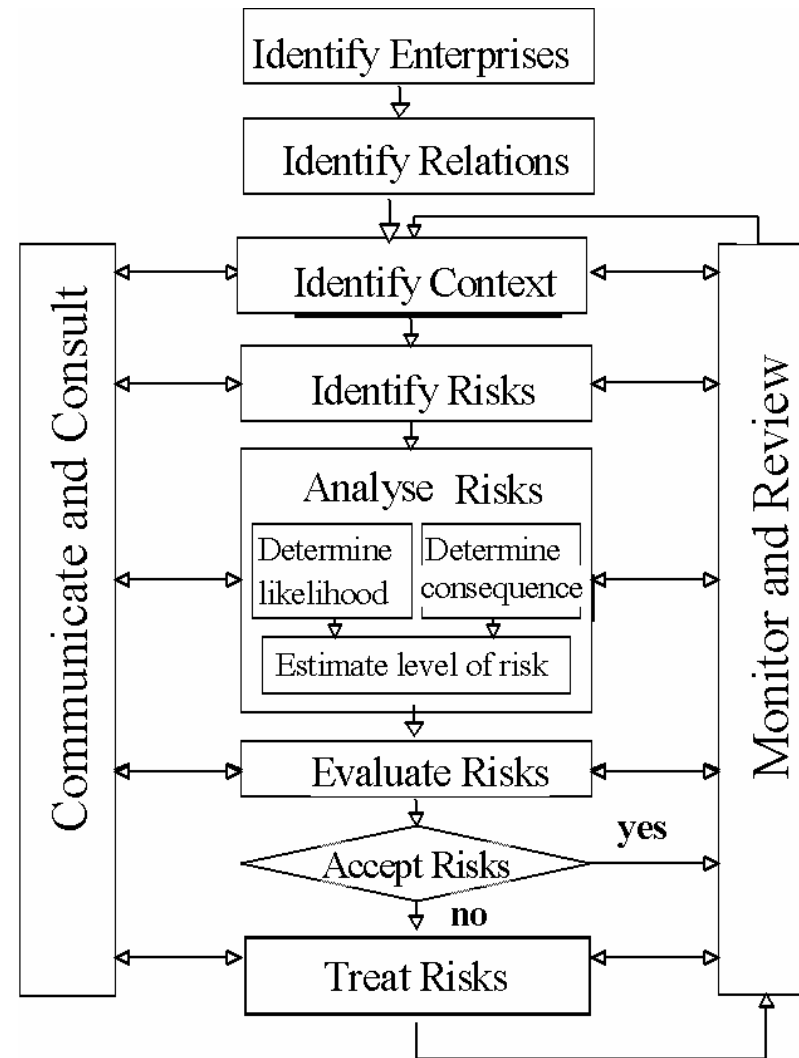
- Activity 6.1: Determine level of risk
- Activity 6.2: Prioritise risks
- Activity 6.3: Categorise risks
- Activity 6.4: Determine interrelationships among risk themes
- Activity 6.5: Prioritise the resulting risk themes and risks

## Sub-process 7: Risk Treatment

- Activity 7.1: Identify treatment options
- Activity 7.2: Assess alternative treatment approaches



# Overview of RMP (iterative)



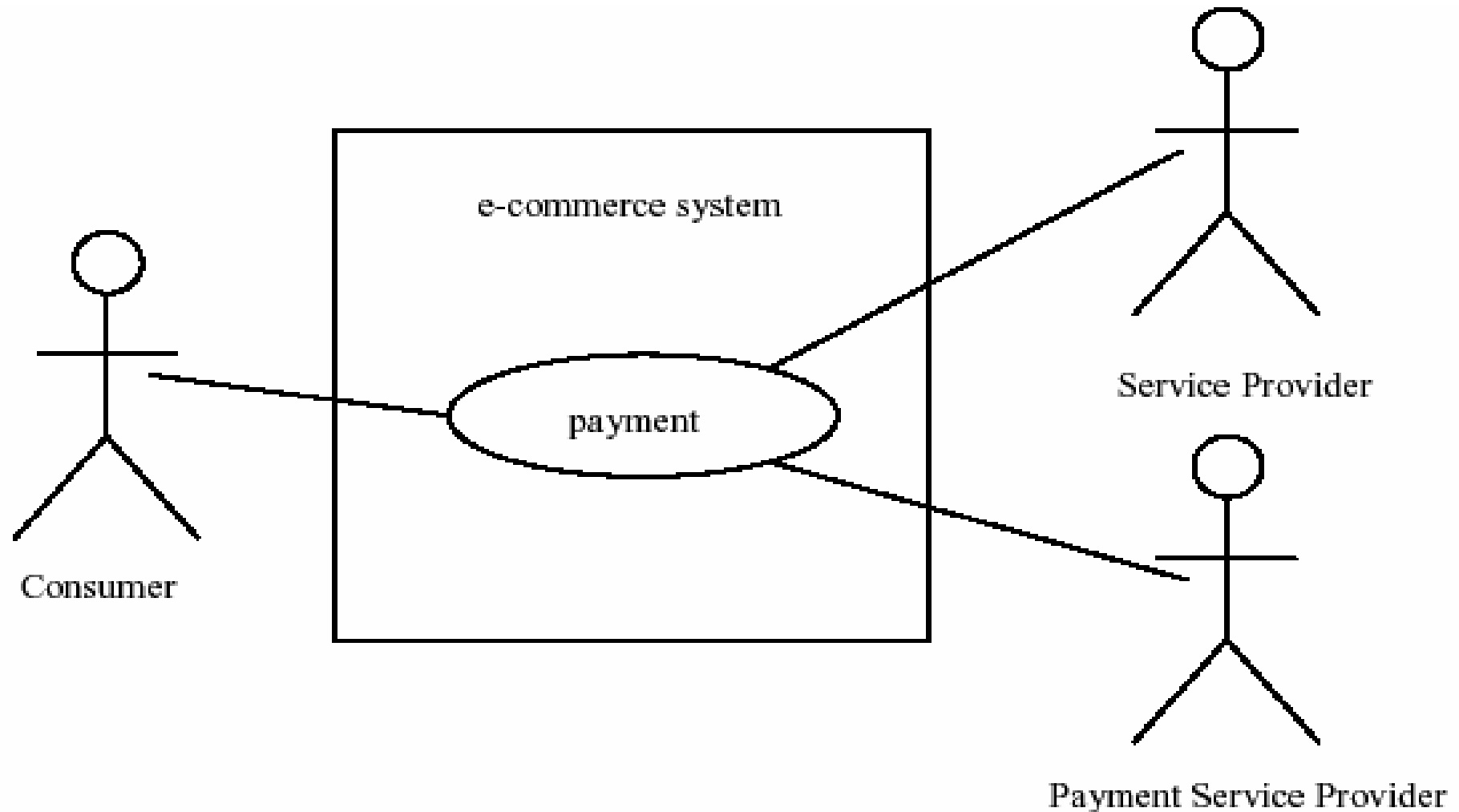
# Small Example: E-Commerce

---

- System: E-Commerce (procure travel packets)
- ToA: Payment mechanism
- Stakeholders:
  - Consumer
  - Service provider
  - Payment service provider
- Sub process 1, 2, 7 and activity 4.1

# Overview Payment Mechanism

---



# SP 1: Identify Enterprises

---

- Activity 1.1: Identify and describe each enterprise
  - Service provider
    - Develop the interface and the interaction between the system and the payment service
  - Payment service provider
    - Develop the payment service mechanism (maintenance is not considered in this assessment)

## A.1.2: Security Policy for each Enterprise

---

- Security policies:
  - **Service provider:** Identity and payment information to consumer should never be revealed to an unauthorised party (never needs to be more clearly specified)
  - **Payment service provider:** Contract with service provider that guarantees that identify and payment information of customer is never disclosed to a unauthorised party

# A. 1.3: Sec. Req. and Sec. Mech.

---

- For each enterprise
- Use UMLsec stereotypes, tags and constraints

# A. 1.4: Set of Enterprises

---

- Describe the set of enterprises contained in the networked enterprise

# A. 1.5: Common security policy

---

- Describe the common security policy of the networked enterprise

# A. 1.6: Common Sec. Req. UMLsec

---

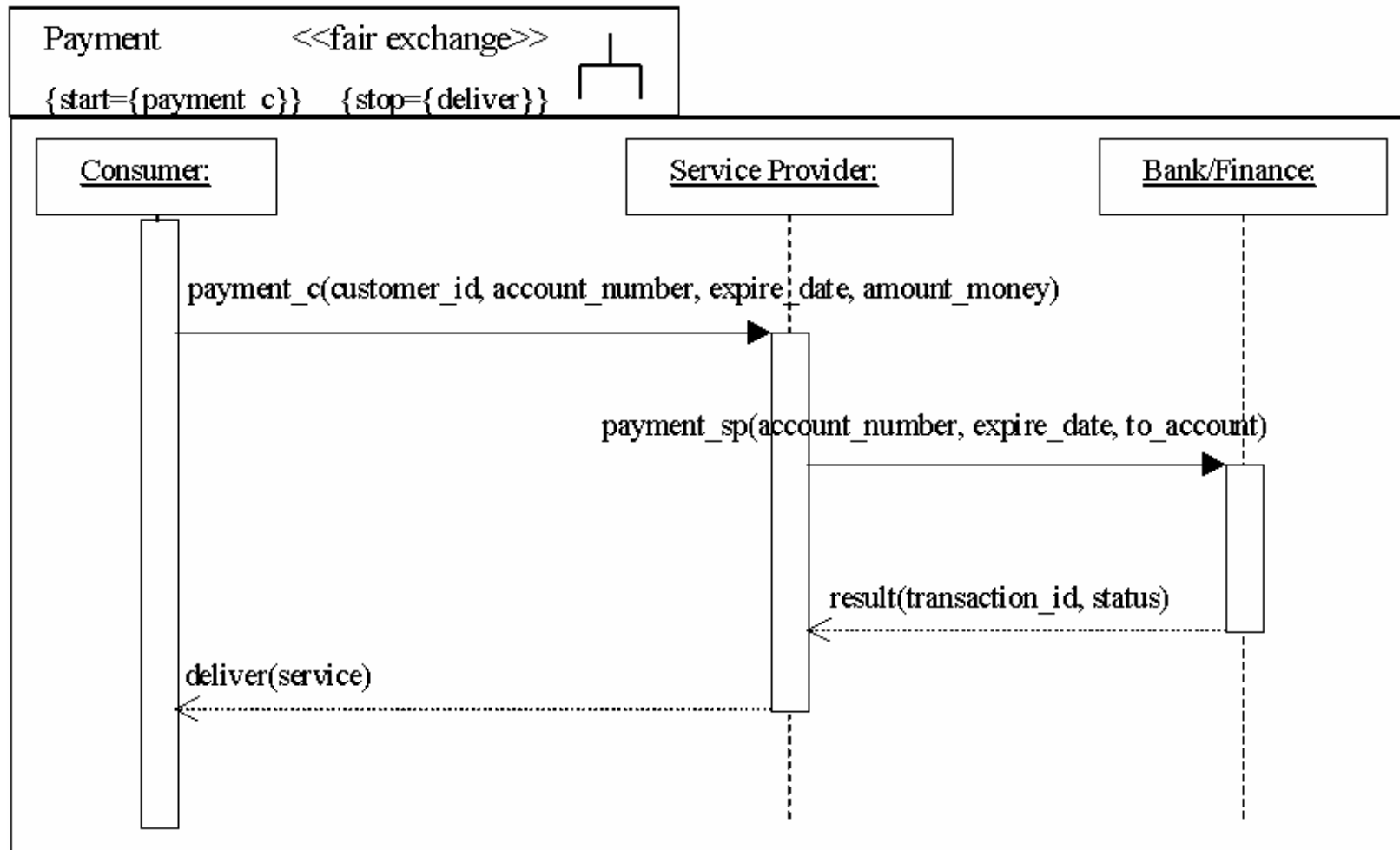
- Confidentiality
  - customer id, consumer account number, service provider account number, expire date and the amount of money to be transferred
- Integrity
  - Amount of money to be transferred
- Non-repudiation
  - Of transaction to prevent denial of send/receive
- Accountability
  - Customer and service provider to be able to trace

# UMLsec Stereotypes and Tags

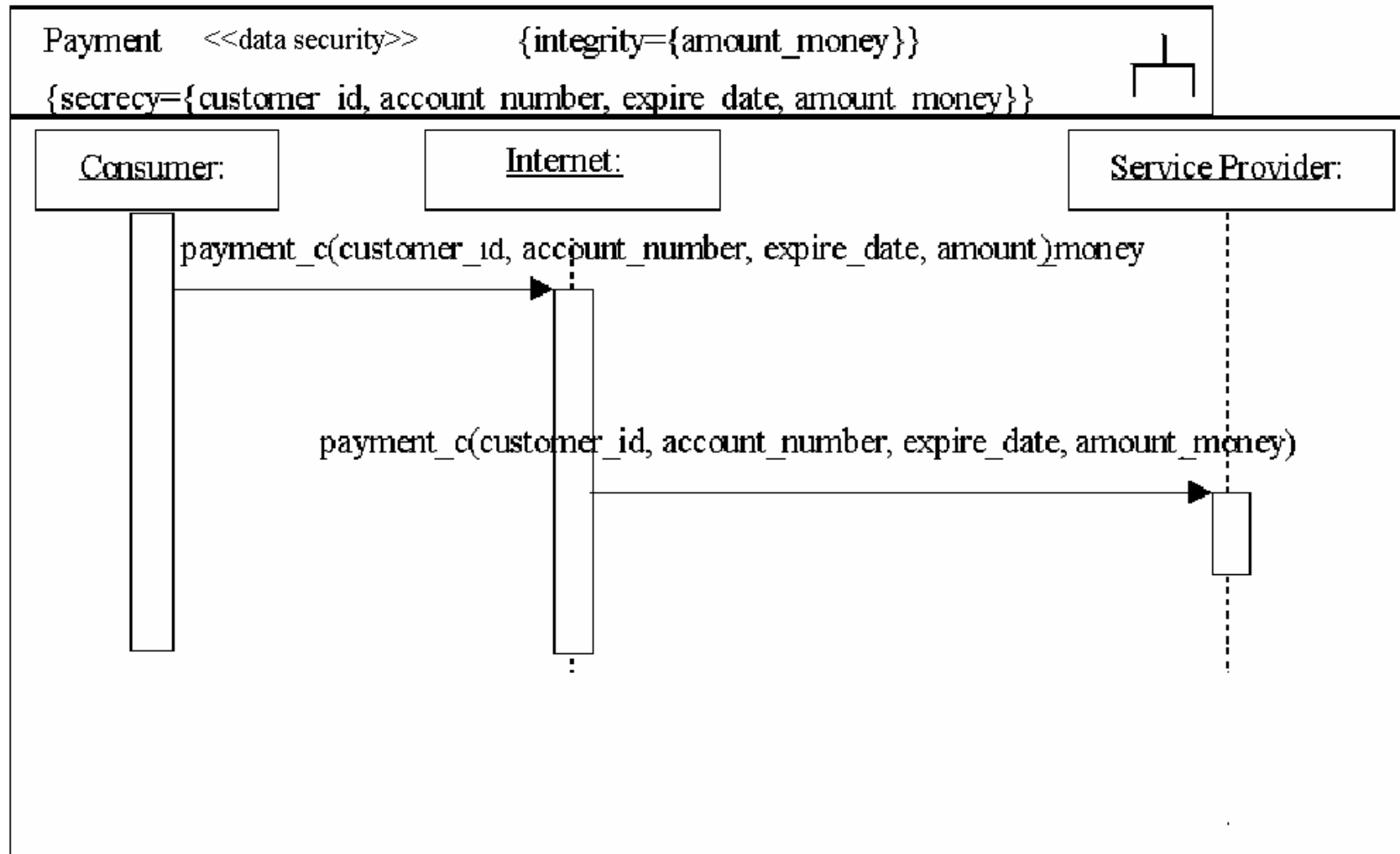
---

- Tool support can be used to test and verify the fulfillment of security requirements using UMLsec stereotypes, tags and constraints
- Confidentiality and integrity
- <<data security>>
  - {secrecy} - confidentiality
  - {integrity} - integrity
- Non-repudiation
- <<fair exchange>>
  - {start} and {stop}

# UMLsec Sec. Req. Diagram (<<fair exchange>>)



# UMLsec Sec. Req. Diagram (`<<data security>>`)



## SP 2: Identify Relationship between Enterprises

---

- Activity 2.1: Identify and describe relationships between each enterprise in the networked enterprise
  - Payment service provider shall deliver a mechanism fulfilling the requirement specification
  - Payment service provider shall prevent the disclosure of consumer id and payment information to unauthorised parties on behalf of the service provider
  - Service provider is responsible for the quality of the requirement specification, which is the contract between the two parties (issues not covered by the specification is the responsibility of the service provider)

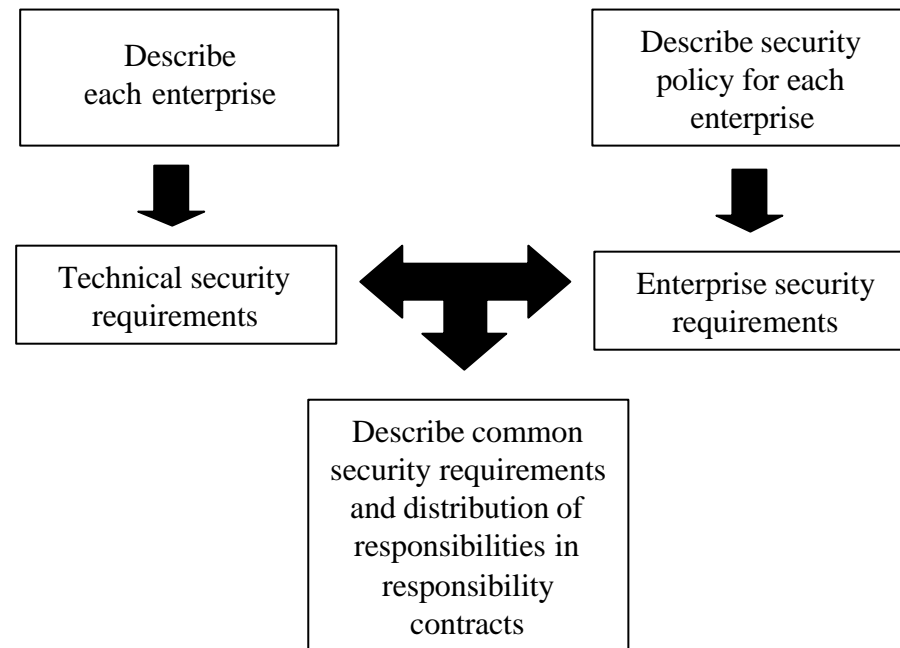
## SP 2: Identify Relationship between Enterprises

---

- Activity 2.2: Identify and describe the distribution of responsibility among the enterprises
  - Confidentiality and integrity of the consumer id and the payment information are within the responsibility of the parties involved in the communication.
  - Service provider is responsible for the communication between the consumer and the service provider (e-commerce application)
  - Payment service provider is responsible for the communication issued by the payment mechanism (by contract through the requirement specification)
- Activity 2.3: Identify stakeholders and specify in which enterprise they belong

# Overview of SP 1 and 2

---



# MBRA for Networked Enterprises

---

## Sub-process 1: Identify Enterprises

- Activity 1.1: Identify and describe each enterprise
- Activity 1.2: Identify and describe the security policy for each enterprise
- Activity 1.3: Identify and describe security requirements and existing security mechanism for each enterprise using UMLsec
- Activity 1.4: Describe the set of enterprises contained in the networked enterprise
- Activity 1.5: Describe the common security policy of the networked enterprise
- Activity 1.6: Specify the set of common security requirements based on the common security policy using UMLsec

## Sub-process 2: Identify Relationship between enterprises

- Activity 2.1: Identify and describe relationship between each enterprise in the networked enterprise
- Activity 2.2: Identify and describe the distribution of responsibility among the enterprises
- Activity 2.3: Identify stakeholders and specify in which enterprise they belong

## Sub-process 3: Identify Context

- Activity 3.1: Identify areas of relevance
- Activity 3.2: Identify and value assets
- Activity 3.3: Identify policies and evaluation criteria
- Activity 3.4: Approval

## Sub-process 4: Identify Risks

- Activity 4.1: Identify threats to assets
- Activity 4.2: Identify vulnerabilities of assets
- Activity 4.3: Document unwanted incidents

## Sub-process 5: Analyse Risks

- Activity 5.1: Consequence evaluation
- Activity 5.2: Frequency evaluation

## Sub-process 6: Risk Evaluation

- Activity 6.1: Determine level of risk
- Activity 6.2: Prioritise risks
- Activity 6.3: Categorise risks
- Activity 6.4: Determine interrelationships among risk themes
- Activity 6.5: Prioritise the resulting risk themes and risks

## Sub-process 7: Risk Treatment

- Activity 7.1: Identify treatment options
- Activity 7.2: Assess alternative treatment approaches



# A. 3.2: Identify and Value Assets

---

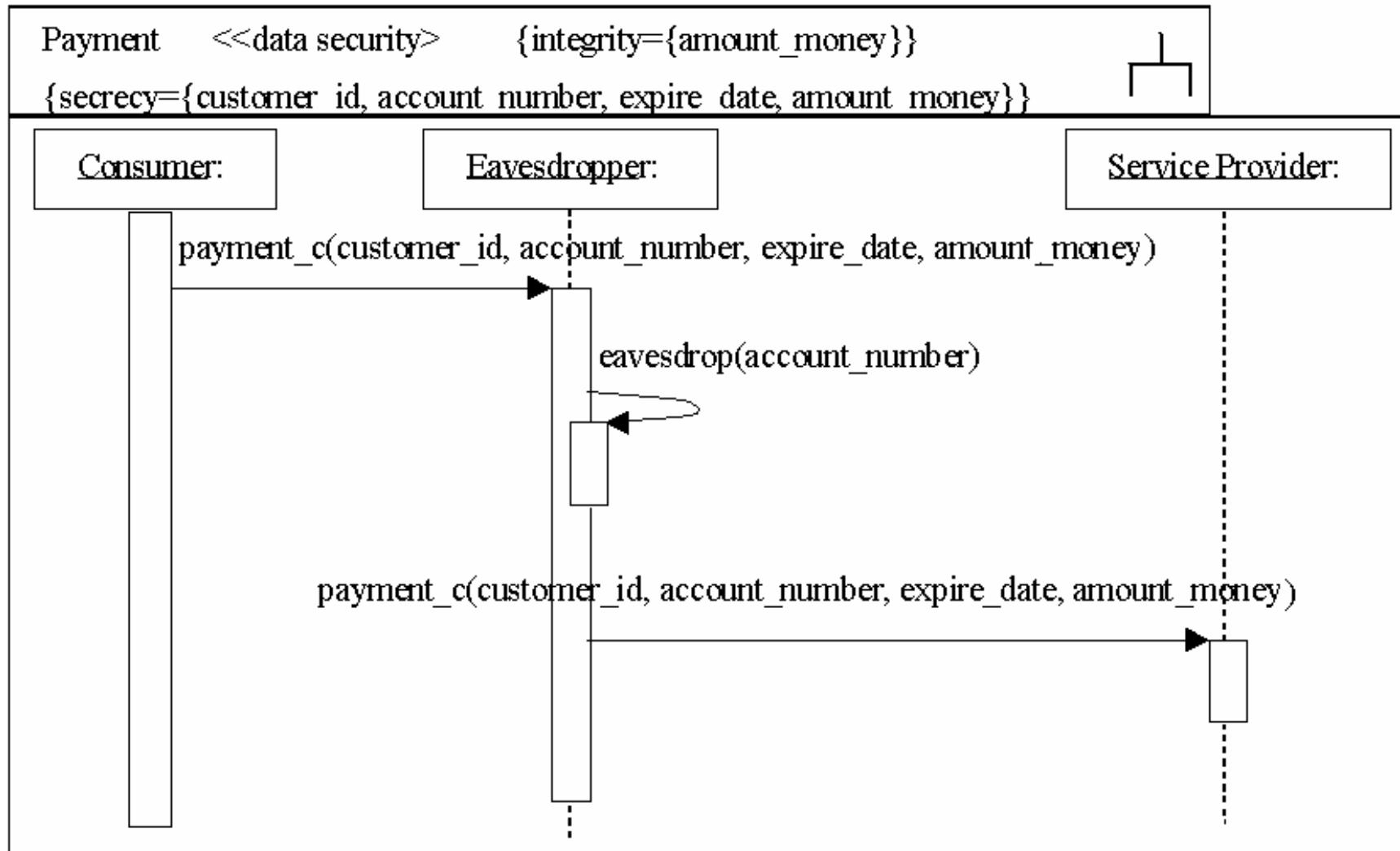
- Information assets (pre-defined asset groups)
  - consumer id
  - service provider id
  - consumer account number
  - consumer expire date
  - service provider account number
  - amount of money to be transferred
- Other asset groups
  - physical, human, law and regulation etc.

# A. 4.1: Identify Threats to Assets

---

- Confidentiality
  - Eavesdropping of consumer id, account number, expire date, amount of money to be transferred
- Integrity
  - Manipulation of amount of money to be transferred
  - (manipulation of service provider account number)

# Threat Scenario (Eavesdrop)

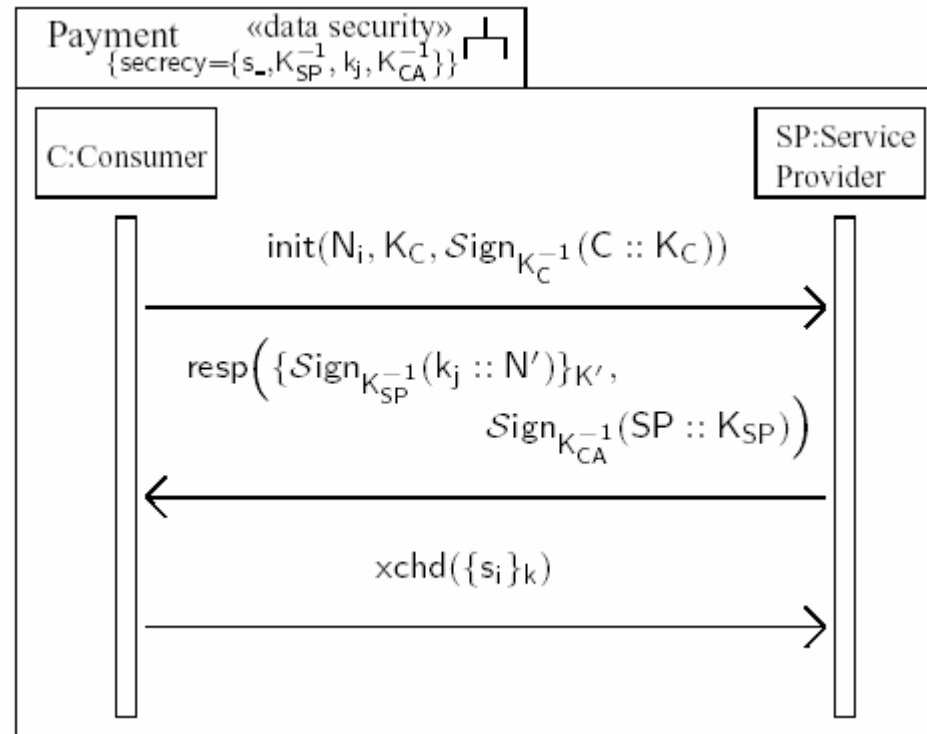


# SP 7: Risk Treatment

---

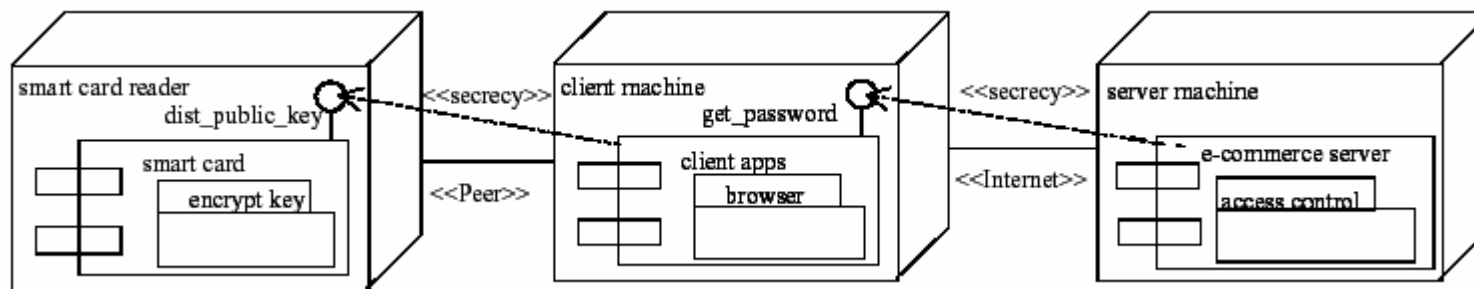
- Reduce consequence and/or frequency
- A. 7.1: Identify treatment options
  - Encrypt link (either on physical layer, network layer, transport layer or application layer)
  - Authentication of consumers using weak authentication (username and password)
  - Authentication of consumers using strong authentication (smart card)
- A.7.2: Assess alternative treatment approaches (cost-benefit)

# Treatment Option Diagram 1



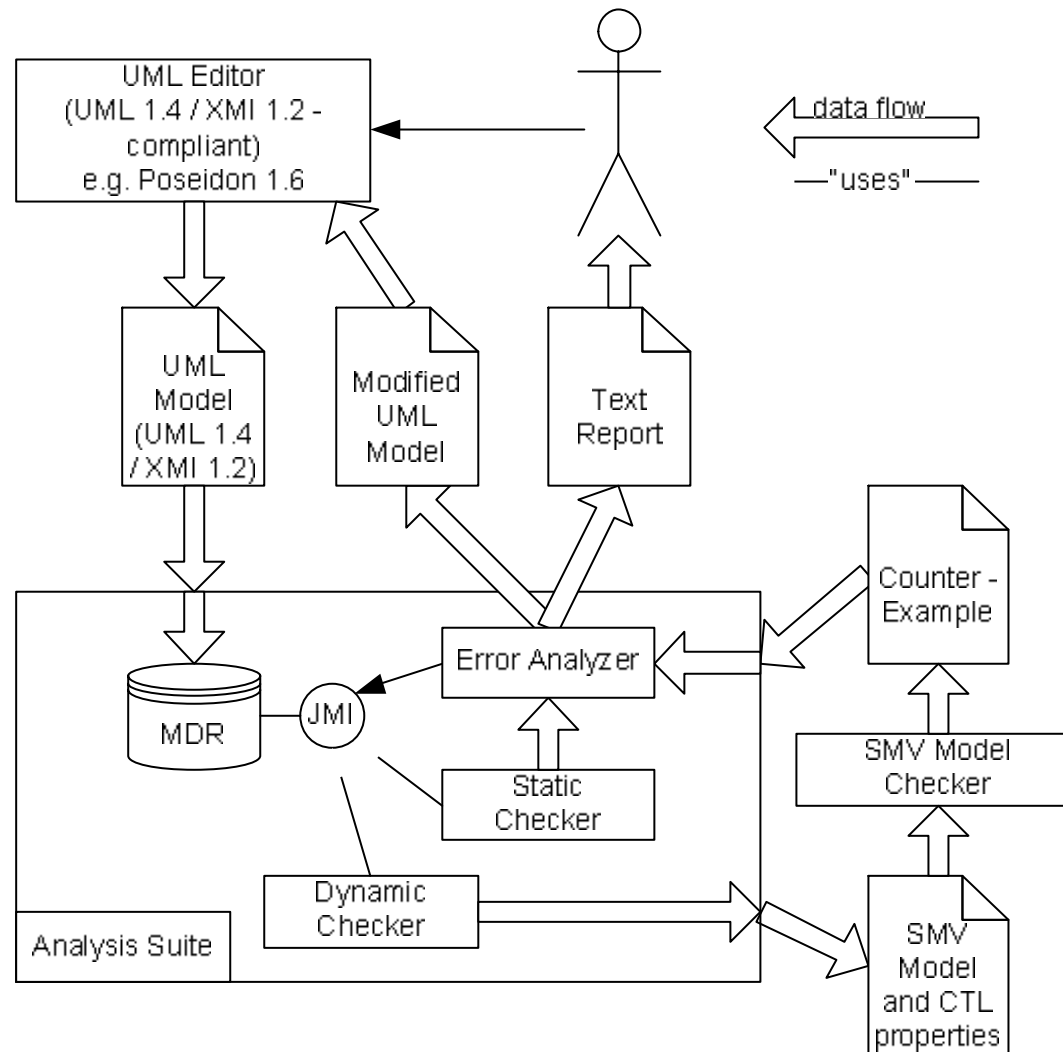
## Encryption of information

# Treatment Option Diagram 2

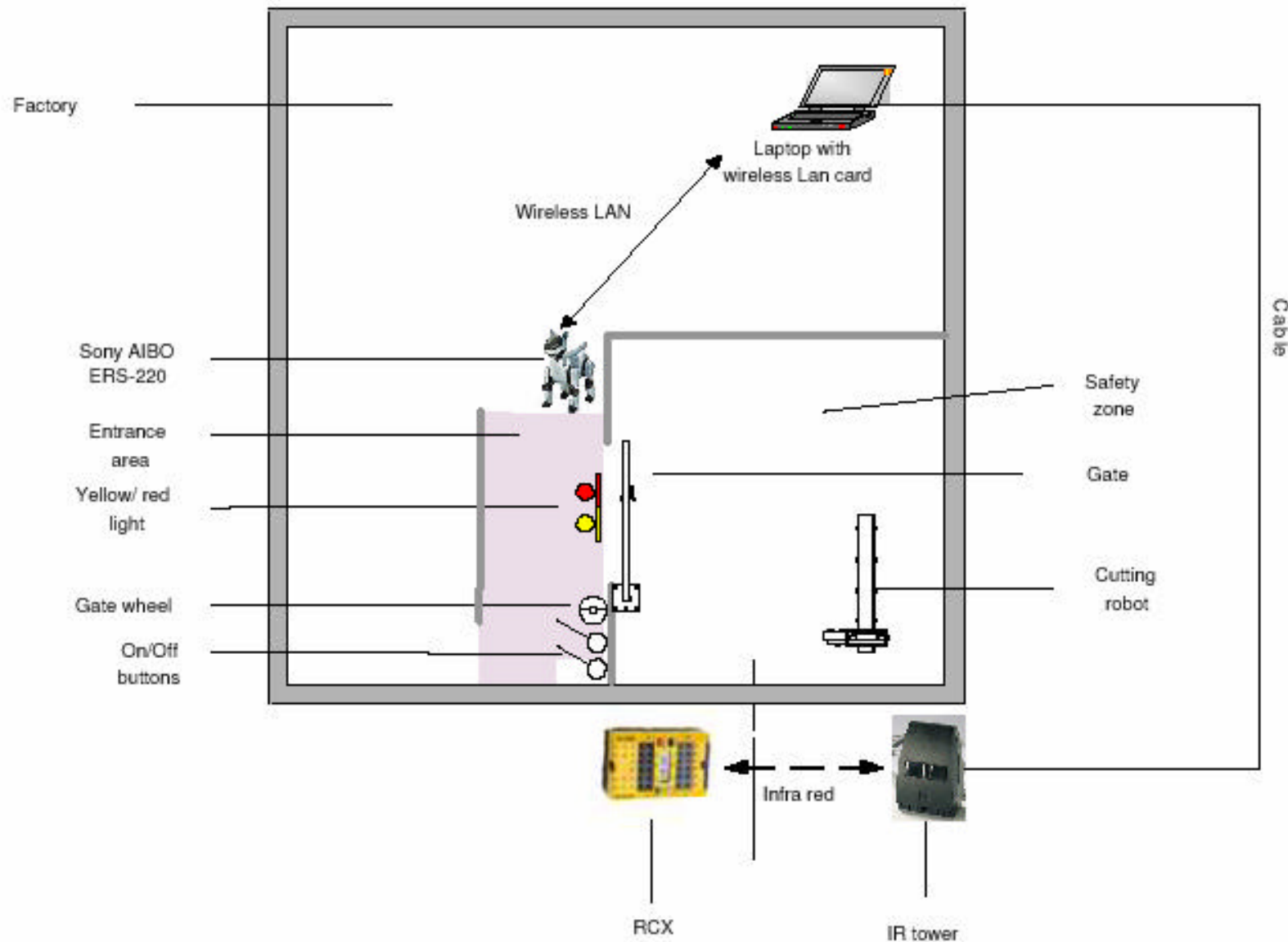


Authenticate consumer (strong authentication)

# Tool Support for UMLsec



# Prototype System of a Production Cell



# Some Resources

---

Book: Jan Jürjens, Secure Systems Development with UML, Springer-Verlag, due 2003.

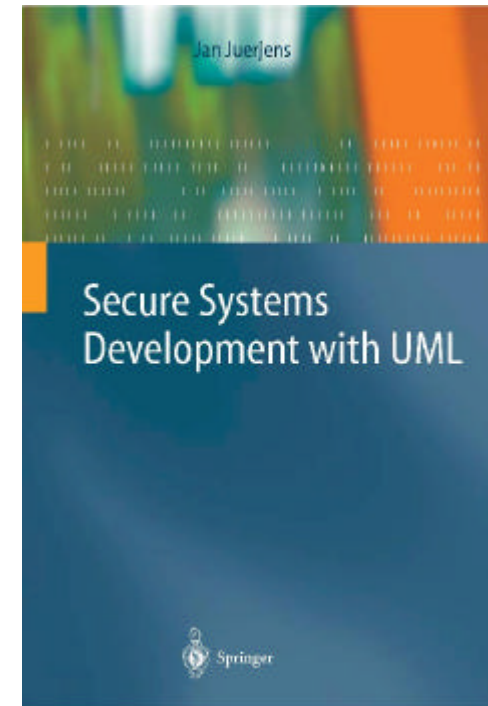
Tutorials: June: UMLws (SFO); Sept: FME (Pisa), FDL (Frankfurt), SAFECOMP (Edinburgh), Nov: LADC2003(Sao Paulo)

Special SoSyM issue on Critical Systems Development with UML

More information (slides etc.):

<http://www4.in.tum.de/~juerjens/csdumltut>

(user: Participant, password: Iwasthere)



# <<secure links>>

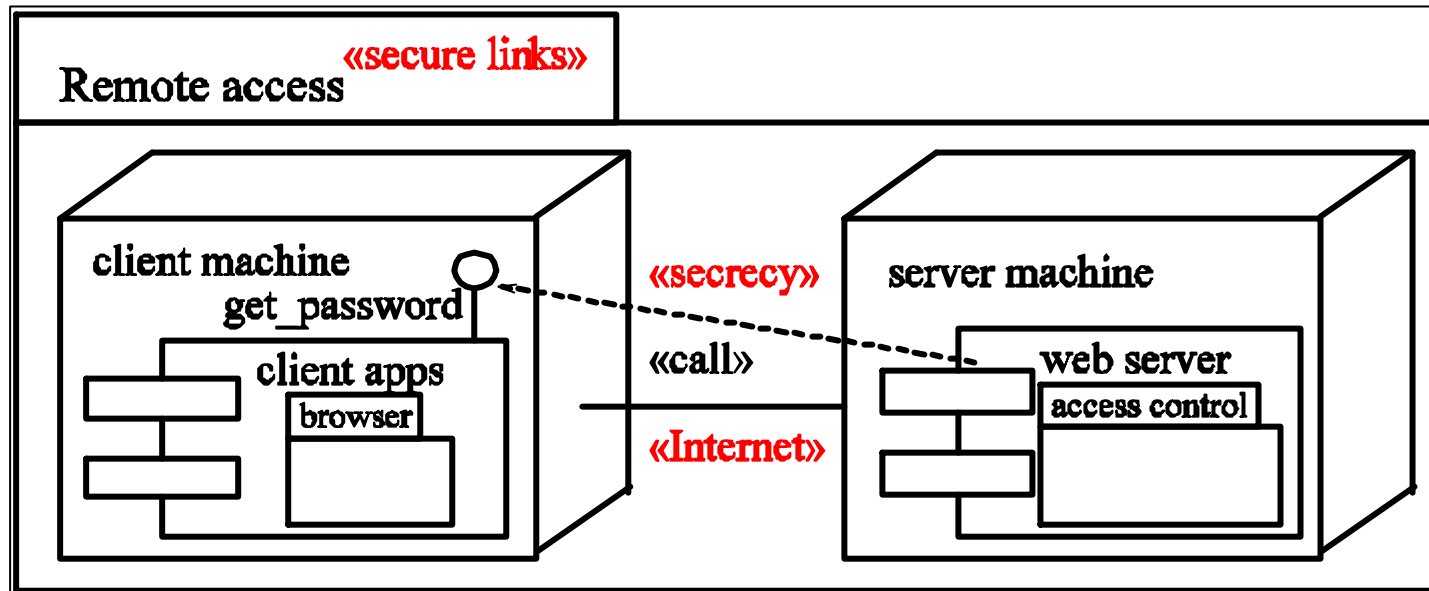
---

Ensures that physical layer meets security requirements on **communication**.

Constraint: for each dependency  $d$  with stereotype  $s \in \{\ll\text{secrecy}\gg, \ll\text{integrity}\gg\}$  between components on nodes  $n$  and  $m$ , have a communication link  $l$  between  $n$  and  $m$  with stereotype  $t$  such that

- if  $s = \ll\text{secrecy}\gg$ : have **read ? Threats<sub>A</sub> (t)**.
- if  $s = \ll\text{integrity}\gg$ : have **insert ? Threats<sub>A</sub> (t)**.

# Example <<secure links>>



- Given **default** adversary type, constraint for stereotype <<secure links>> **violated**:
  - According to the **Threats<sub>default</sub>(Internet)** scenario, <<Internet>> link does not provide secrecy against **default** adversary.

# <<secure dependency>>

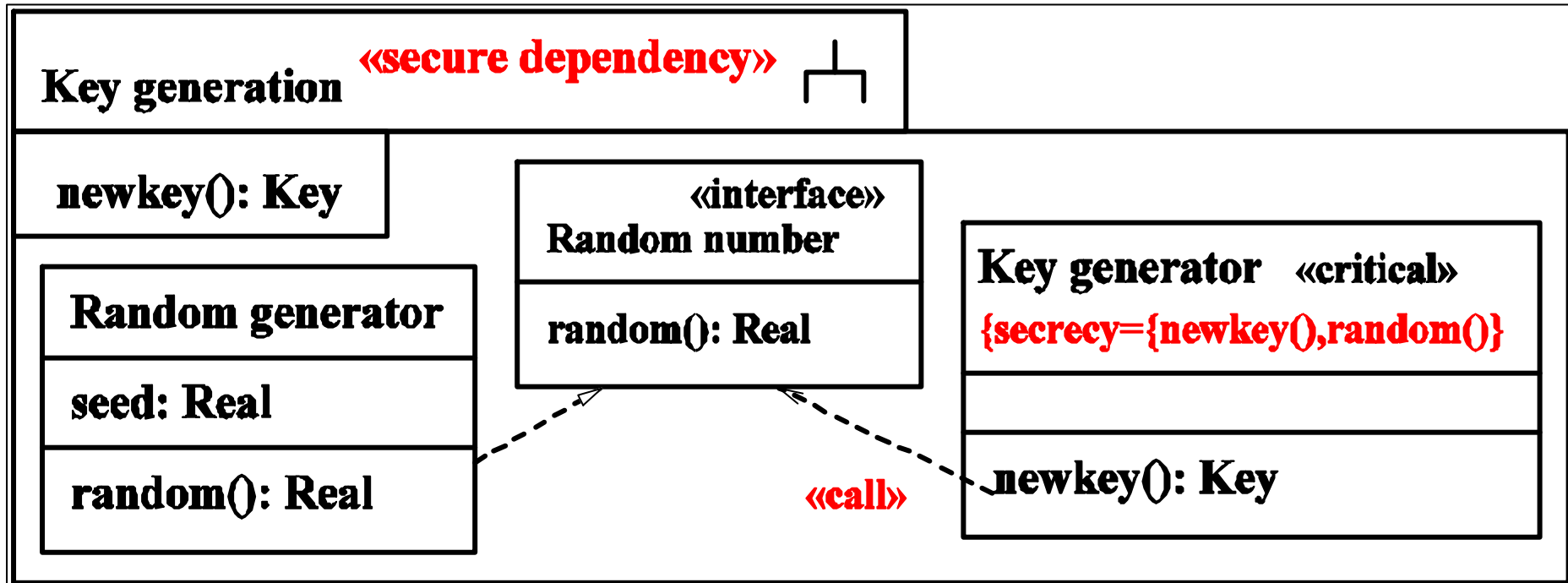
---

Ensure that <<call>> and <<send>> dependencies between components **respect** security requirements on communicated data given by tags {**secrecy**}, {**integrity**}.

Constraint: for <<call>> or <<send>> dependency from *C* to *D* (and similarly for {**secrecy**}):

- Msg in *D* is {**secrecy**} in *C* if and only if also in *D*.
- If msg in *D* is {**secrecy**} in *C*, dependency stereotyped <<secure dependency>>.

# Example <<secure dependency>>



- **Violates** <<secure dependency>>:
  - **Random generator** and <<call>> dependency do not give security level for **random()** to **key generator**.

# <<data security>>

---

Security requirements of data marked <<critical>> **enforced** against threat scenario from deployment diagram.

Constraints:

- **Secrecy** of {secrecy} data preserved.
- **Integrity** of {integrity} data preserved.

# Example <<data security>>

Variant of TLS  
(INFOCOM'99).  
Violates {secrecy}  
of  $s$  against  
default  
adversary.

{secrecy = { $s, K_C^{-1}$ }}

«data security»

«critical»

«Internet»

