

# Introduction



- National Institutes of Informatics
  - National Research Center for Information Science
    - ◆ 80 Research staffs and about 100 PhD students
- Research Interests
  - Software Engineering on Pervasive Computing
    - ◆ Middleware, Design, Modeling
  - Security Patterns, Method using Security Patterns



# **A Survey of Security Patterns and Vulnerability Analysis using Attack Patterns**

Nobukazu Yoshioka<sup>†</sup>, Hironori Washizaki<sup>†,††</sup>,

Katsuhisa Maruyama<sup>‡</sup>

<sup>†</sup>National Institute of Informatics,

<sup>††</sup>The Graduate University for Advanced Studies,

<sup>‡</sup>Ritsumeikan University

# Backgrounds

- Importance of Security
- Difficulty of Development of Secure Systems
  - We should consider many kinds of concerns and situations
- Many Security Patterns have been proposed
  - It is still difficult to use ...
    - ➔ Classification them and propose a new pattern

# Classification of Pattern

## ■ Development Process

- Security Requirements and Analysis
- Security Design
- Security Implementations

→ Discuss Efficiency of Patterns from the security concern view point

→ Propose a new pattern

# Security Concerns

Markus Schumacher: "Security Engineering With Patterns", Springer, 2003:

- **Asset**: Information or resources that have value to an organization or person.
- **Stakeholder**: An organization or person who places a particular value on assets.
- **Security objective**: A statement of intent to counter threats and satisfy identified security needs.
- **Threat**: A potential for a security breach of an asset.
- **Attack**: An action that violates the security of an asset.
- **Attacker**: The entity which carries out attacks.
- **Vulnerability**: A flaw or weakness that could be exploited to breach the security of an asset.
- **Countermeasure**: An action taken in order to protect an asset against threats and attacks.
- **Risk**: The probability that a successful attack occurs.



Discuss Efficiency of Patterns based on these concerns

# Classification of Pattern

## ■ Development Process

- Security Requirements and Analysis
- Security Design
- Security Implementations

→ Discuss Efficiency of Patterns from the security concern view point

→ Propose a new pattern

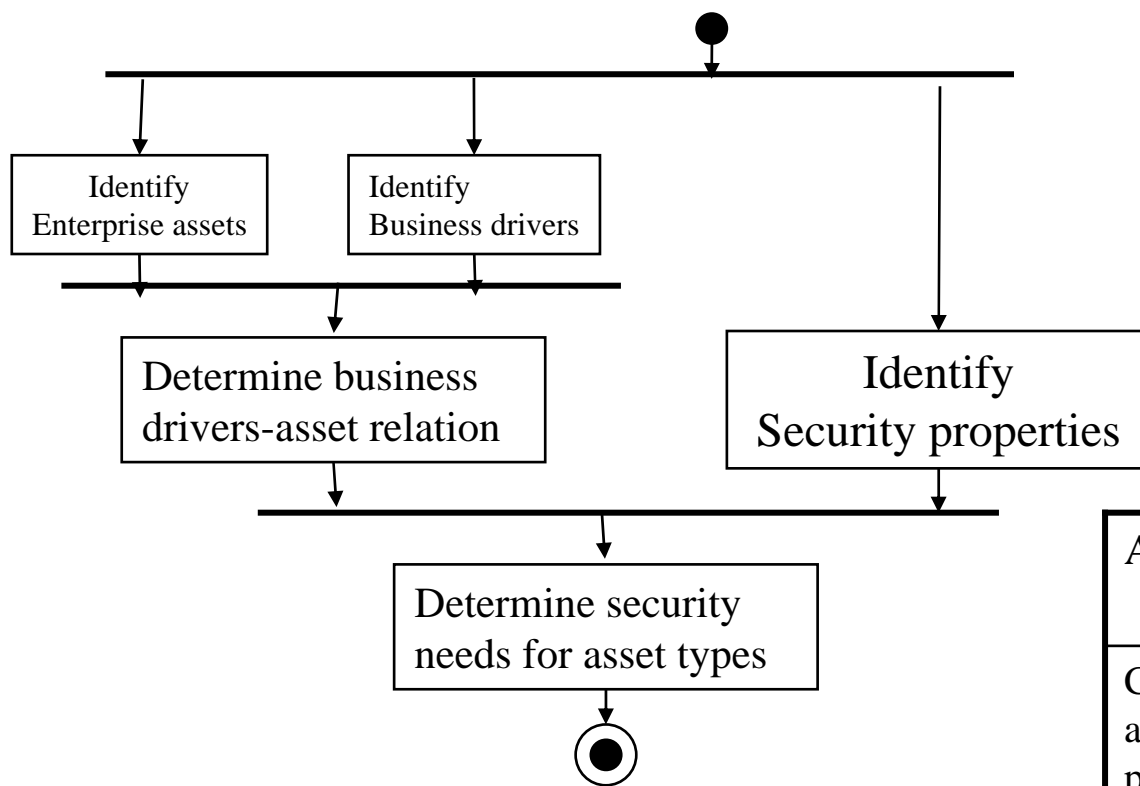
# Patterns for Security Requirements and Analysis

There are Analysis Process Patterns.

- Determine WHAT, assets, we need to protect.
  - Security Needs Identification for Enterprise Assets[SecPat06]
- Determine security needs and HOW FAR protect assets for the requirements?
  - Security needs :
    - Security Types:Confidentiality, Integrity, Availability, Accountability
    - Security Needs Identification for Enterprise Assets[SecPat06])
  - HOW FAR protect assets?: We need take priority.
    - Asset Valuation Pattern[SecPat06]
    - Threat Assessment Pattern[SecPat06]
    - Vulnerability Assessment Pattern[SecPat06]

# Example of Patterns : Security Needs Identification

## Security Needs Identification for Enterprise Assets[SecPat06]



Security needs solution sequence

Common information asset categories

Asset Type	Security Needs	Business Factors
Customer and business partner data	Confidentiality, Integrity, Accountability	<ul style="list-style-type: none"> <li>■ Competitive issues</li> <li>■ Service issues if a public company</li> </ul>

# Efficiency of Security Patterns on Requirements phase

Phase	Requirements and Analysis Phase	
Concept		
Countermeasure	Identified	+
Risk	Identified	+
Threat	Identified	+
Attack	Identified	++
Attacker	Identified	++
Vulnerability	Identified	+
Asset	Defined	+
Stakeholder	Defined	+
Security objective	Defined	+

specification {



Almost all concerns are mentioned

# Classification of Pattern

## ■ Development Process

- Security Requirements and Analysis
- Security Design
- Security Implementations

→ Discuss Efficiency of Patterns from the security concern view point

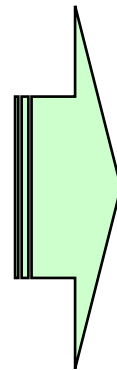
→ Propose a new pattern

# Design Patterns for Security Functions

## ■ Determine HOW TO protect assets

Security type

- Confidentiality
- Integrity
- Availability
- Accountability



Security Functions

- Access Control
- Authentication
- Encryption
- Signature
- Logging, etc.

## ➔ Many Design Patterns Access control, Confidentiality

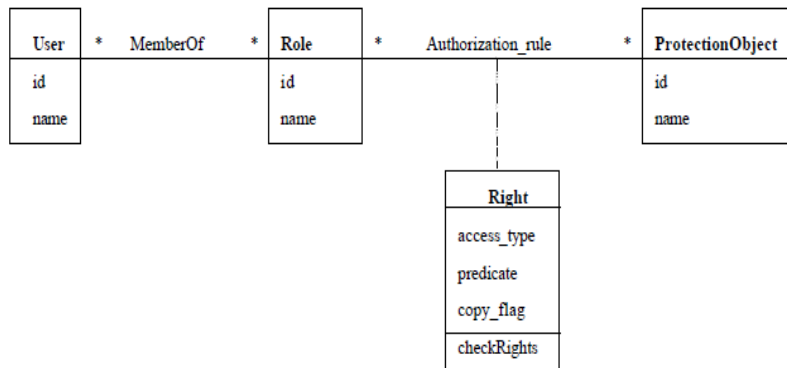
- ◆ Role based Access control Patten
- ◆ Single Access Point, Check point patterns

## ➔ Patterns for Availability

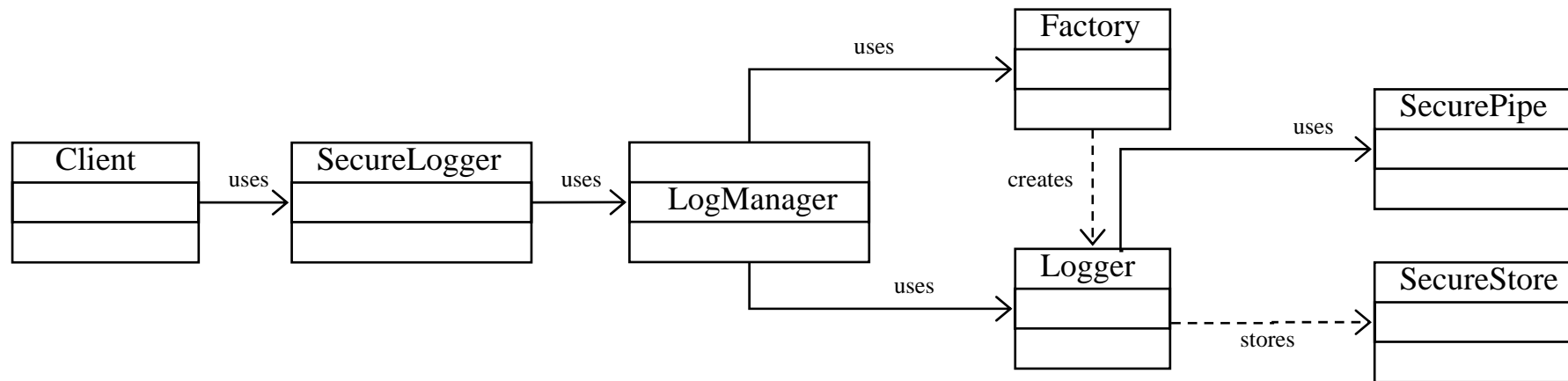
- ◆ Firewall Pattern: IP level, Transportation Level, Service Level

## ➔ Security Patterns on OS Level for Web applications

# Security Design Pattern Examples



Role based Access Control Pattern [Fernandez01]



Secure Logger Pattern with Secure Log Store Strategy [CoreSecurityPatterns05]

# Efficiency of Security Patterns on Design phase

Phase	Design Phase	
Concept		
Countermeasure	Feasibility	++
Risk	Estimated	
Threat	Feasibility	
Attack	Feasibility	
Attacker	Feasibility	
Vulnerability	Feasibility	
Asset	Designed with security	+
Stakeholder	Reviews	
Security objective	Reviewed	+

specification



➡ Mainly focused on Security Countermeasure pattern

# Classification of Pattern

## ■ Development Process

- Security Requirements and Analysis
- Security Design
- Security Implementations

→ Discuss Efficiency of Patterns from the security concern view point

→ Propose a new pattern

# Patterns for Security Implementation

## 【Implementation Phase】

### ■ Secure Programming

- Guidelines for secure program to avoid security flaws
  - Input validation, buffer overflow, etc.

### ■ Secure Refactoring

- Remove security vulnerability
  - ◆ Change public field to private one
  - ◆ Remove setting method and declare final
  - ◆ Hiding classes which do not need to be publicly visible

### ■ Attack Patterns

- How to break software: ⇒ useful for improving the implementation
  - Attack Patterns for Web application, MediaPlayer, Web Browser, etc.

# Efficiency of Security Patterns on Implementation phase

Phase	Implementation Phase	
Concept		
Countermeasure	Implemented	++
Risk	Measured	
Threat	Realized	+
Attack	Tested	++
Attacker	Tested	
Vulnerability	Realize	++
Asset	Implemented with security	+
Stakeholder	Tests	
Security objective	Reviewed	

specification {



Mainly focused on attack and vulnerability concerns

# Classification of Pattern

## ■ Development Process

- Security Requirements and Analysis
- Security Design
- Security Implementations

→ Discuss Efficiency of Patterns from the security concern view point

→ Propose a new pattern

# Discussion: Efficiency of Security Patterns

Phase	Requirements Phase	Design Phase	Implementation Phase
Concept			
Countermeasure	Feasibility +	Feasibility ++	Implemented ++
Risk	Estimated +	Estimated	Measured
Threat	Feasibility +	Feasibility	Realized +
Attack	Feasibility ++	Feasibility	Tested ++
Attacker	Feasibility ++	Feasibility	Tested
Vulnerability	Feasibility +	Feasibility	Realize ++
Asset	Designed with security +	Designed with security +	Implemented with security +
Stakeholder	Reviews +	Reviews	Tests
Security objective	Reviewed +	Reviewed +	Reviewed

Specification

Overleaped Area

# Discussion: Efficiency of Security Patterns

Phase	Requirements Phase	Design Phase	Implementation Phase
Concept			
Countermeasure	Feasibility +	Feasibility ++	Implemented ++
Risk	Estimated +	Estimated	Measured
Threat	Feasibility +	Feasibility	Realized +
Attack	Feasibility ++	Feasibility	Tested ++
Attacker	Feasibility ++	Feasibility	Tested
Vulnerability	Feasibility +	Feasibility	Realize ++
Asset	Designed with security +	Designed with security +	Implemented with security +
Stakeholder	Reviews +	Reviews	Tests
Security objective	Reviewed +	Reviewed +	Reviewed

Specification

Lacked Relation

Lacked Area

# Classification of Pattern

## ■ Development Process

- Security Requirements and Analysis
- Security Design
- Security Implementations

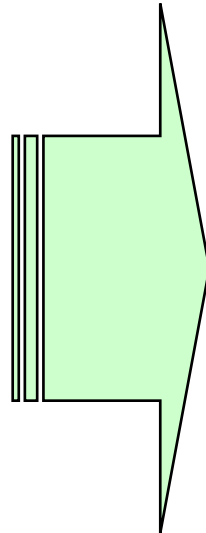
→ Discuss Efficiency of Patterns from the security concern view point

→ Propose a new pattern

# Security Requirements and Design

## 【Security Requirements】

- WHAT we protect?
- Security Needs
  - Confidentiality
  - Integrity
  - Availability
  - Accountability
- HOW FAR protect assets?:  
We need take priority.
  - ➔ Threat Analysis
  - ➔ Vulnerability Analysis



## 【Security Design】

- WHERE we need to protect?
  - Which Object? Classes?
  - Messages? Protocols?
- HOW TO protect?
  - Access Control
  - Authentication
  - Encryption
  - Signature
  - Logging, etc

# Gap between Security Requirements and Design

## 【Security Requirements】

- WHAT we protect?
- Security Needs
  - Confidentiality
  - Integrity
  - Availability
  - Accountability
- HOW FAR protect assets?:  
We need take priority.
- ➔ Threat Analysis
- ➔ Vulnerability Analysis

## 【Security Design】

WHERE we need to protect?

- Which Object? Classes?
- Messages? Protocols?

HOW TO protect?

- Access Control
- Authentication
- Encryption
- Signature
- Logging, etc

Which Security Level is proper?

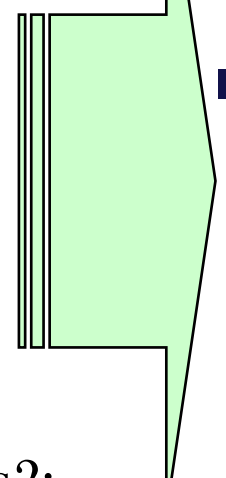
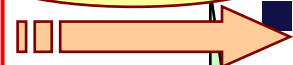
•What is detailed threat?  
•Which parts are vulnerability?

# A New Development Method

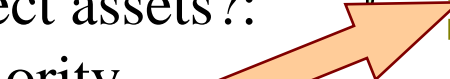
## 【Security Requirement】

- WHAT we protect?
- Security Needs
  - Confidentiality
  - Integrity
  - Availability
  - Accountability
- HOW FAR protect assets?:  
We need take priority.
  - ➔ Threat Analysis
  - ➔ Vulnerability Analysis

Asset Design



Attack Design



## 【Security Design】

- WHERE we need to protect?
  - Which Object? Classes?
  - Messages? Protocols?
- HOW TO protect?
  - Access Control
  - Authentication
  - Encryption
  - Signature

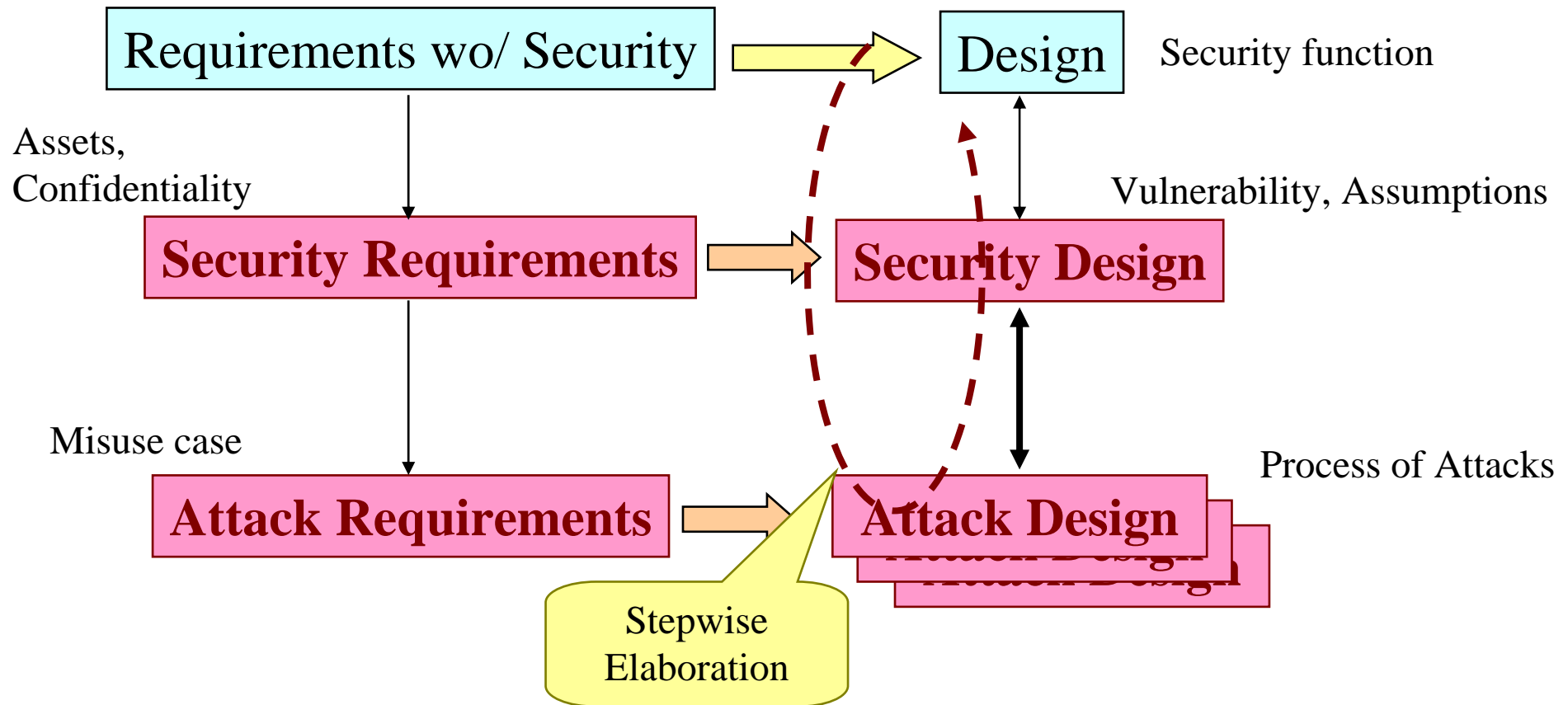
HOW TO attacked?

WHERE is vulnerability?

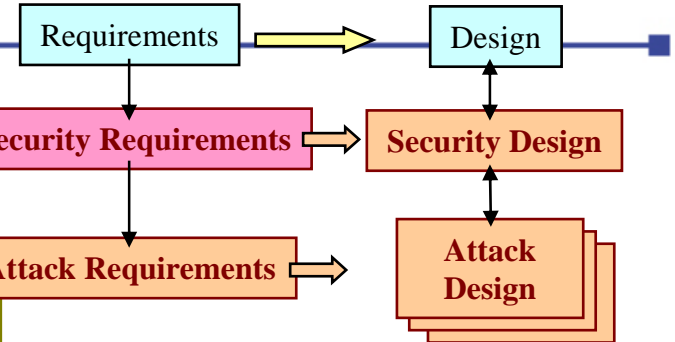


# Overview of Our Development Method

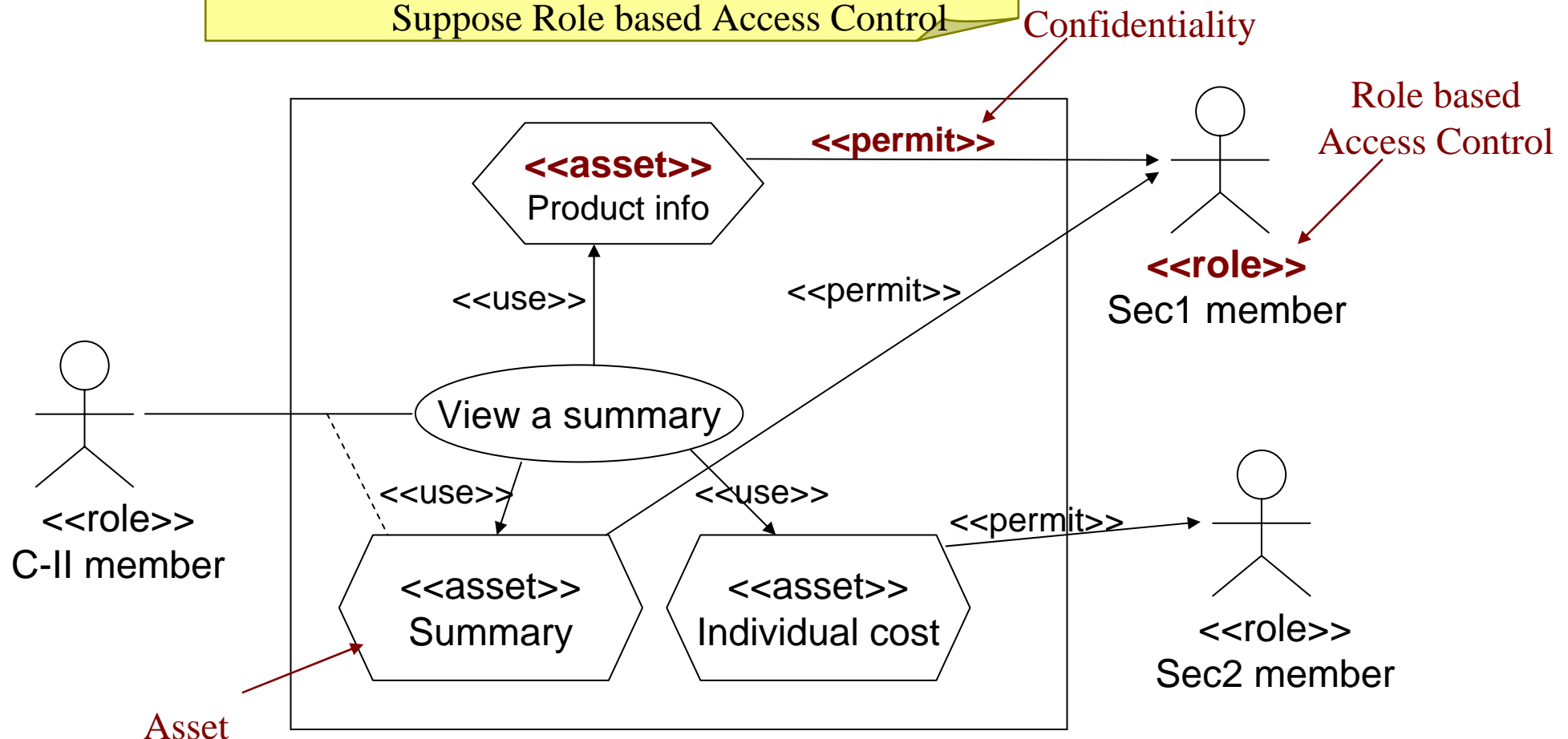
Refine security concerns step by step and design security functions



# Security Requirements

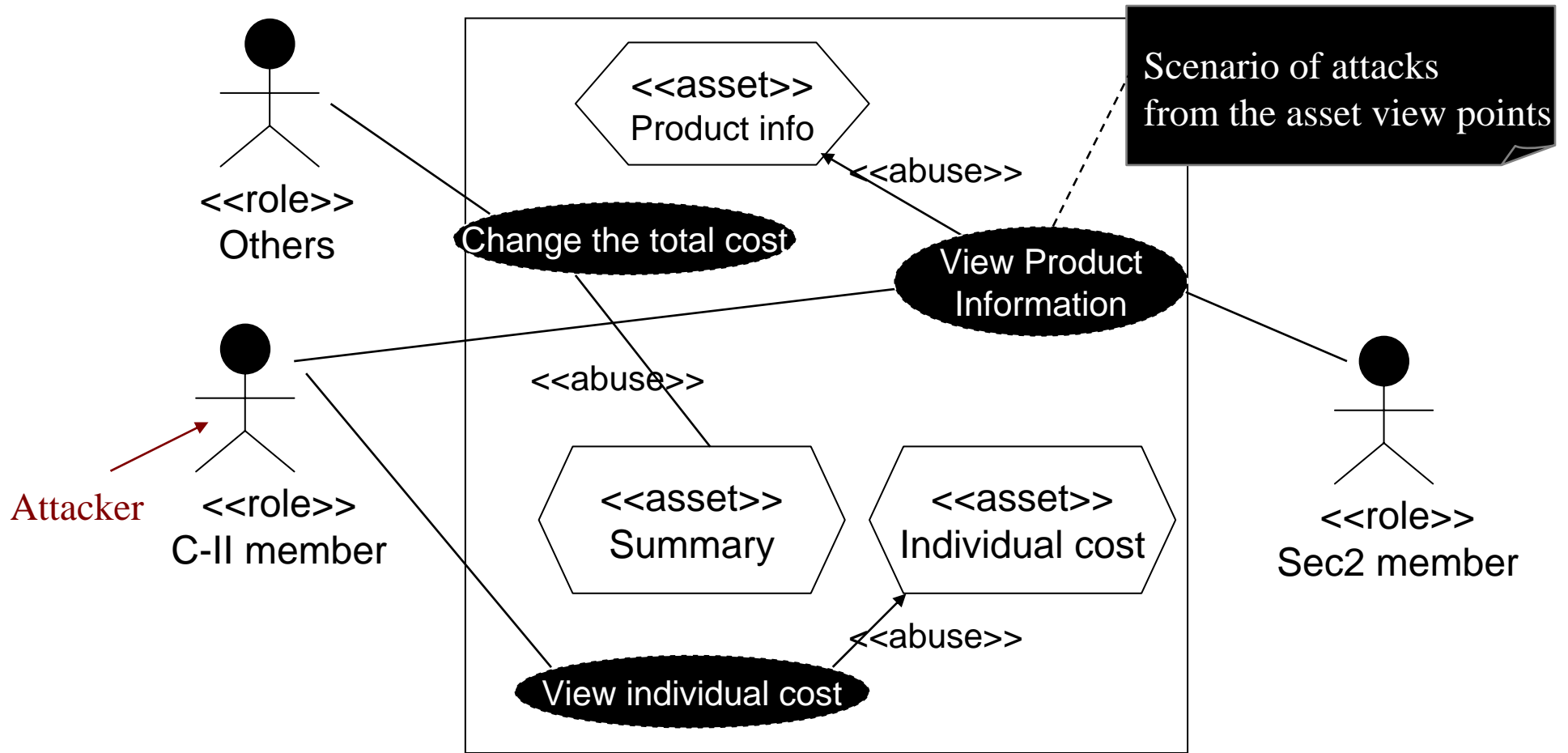
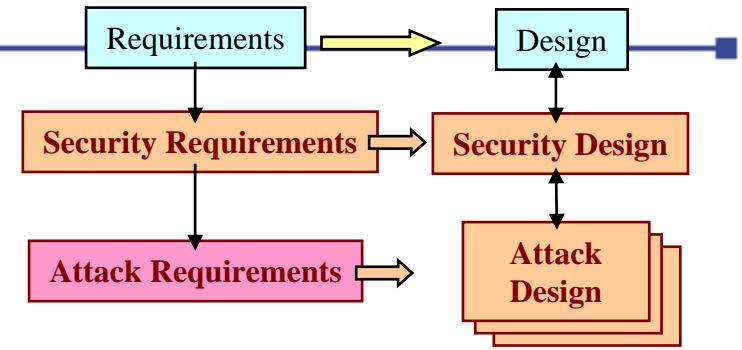


- use case including
    - Assets definition with **<<asset>>**
    - Confidentiality definition with **<<permit>>**
- Suppose Role based Access Control



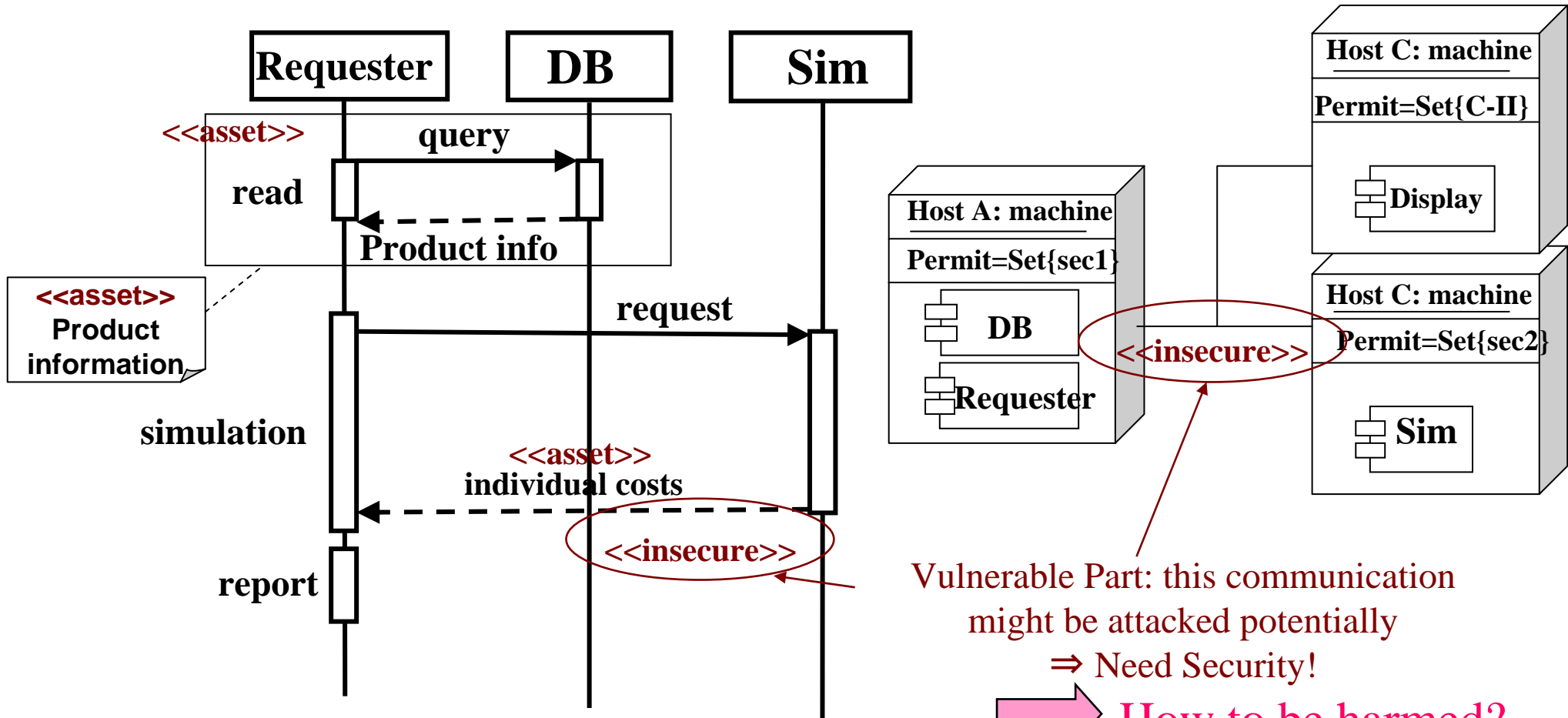
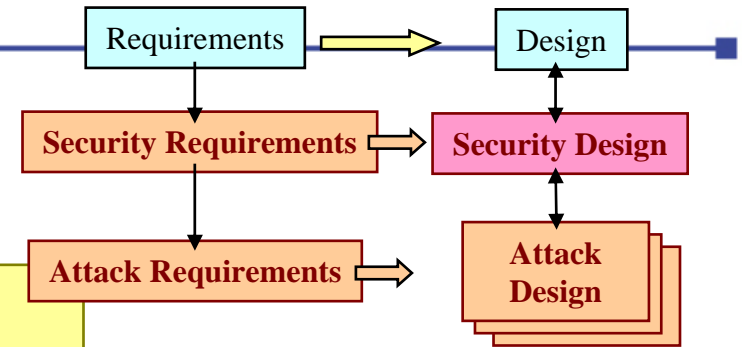
# Attacker's Requirements

Specify attacks against assets with **misuse case**



# Security Design

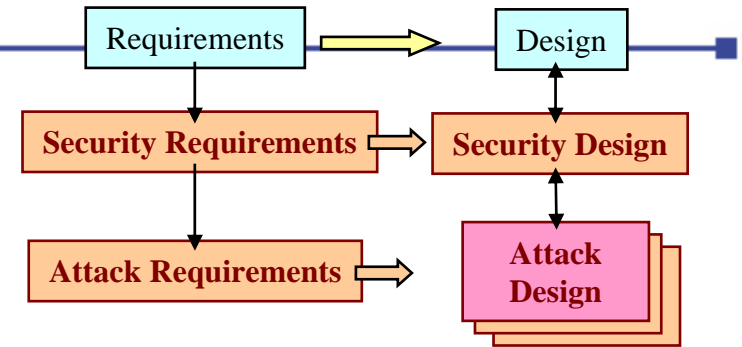
- Reification of assets
- Specify Vulnerability with `<<insecure>>`



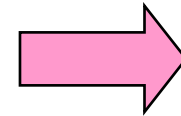
Vulnerable Part: this communication might be attacked potentially  
 ⇒ Need Security!

➔ **How to be harmed?**

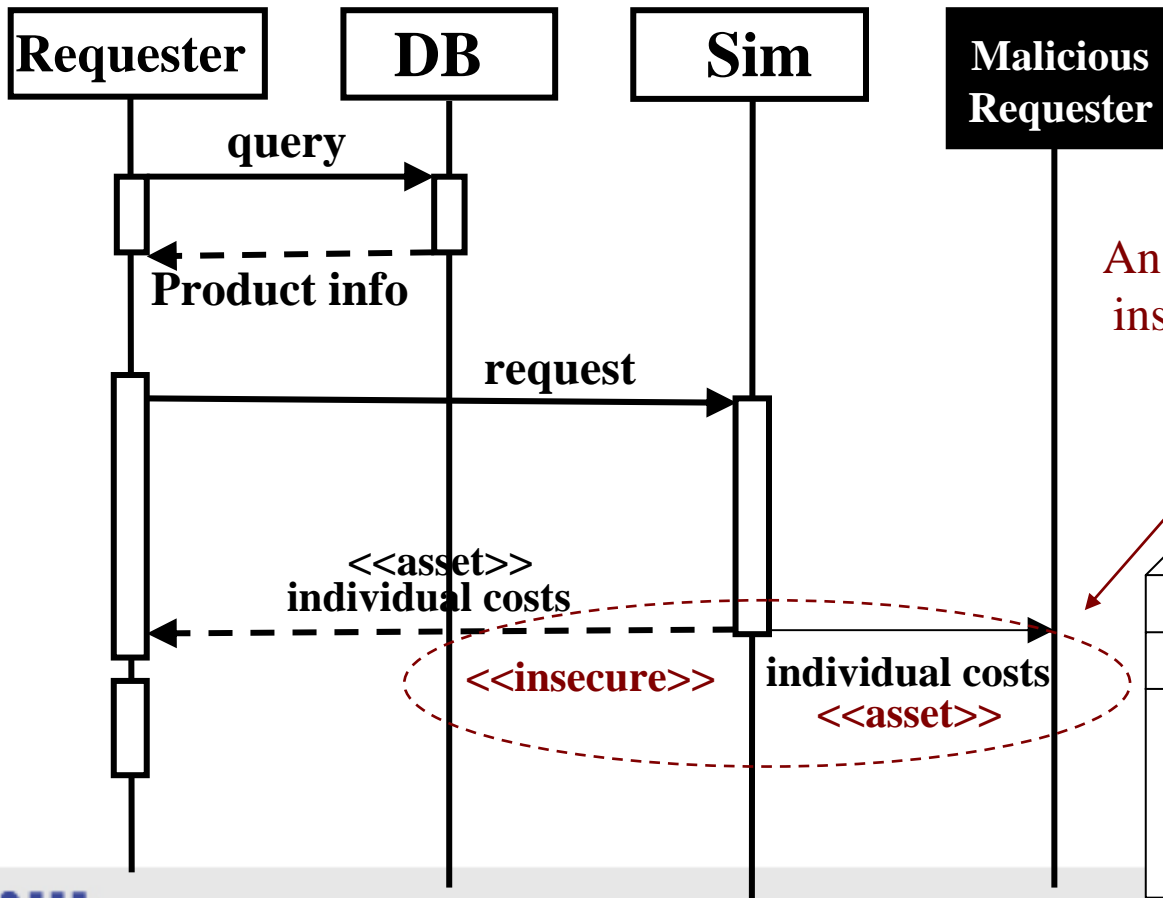
# Attack Design



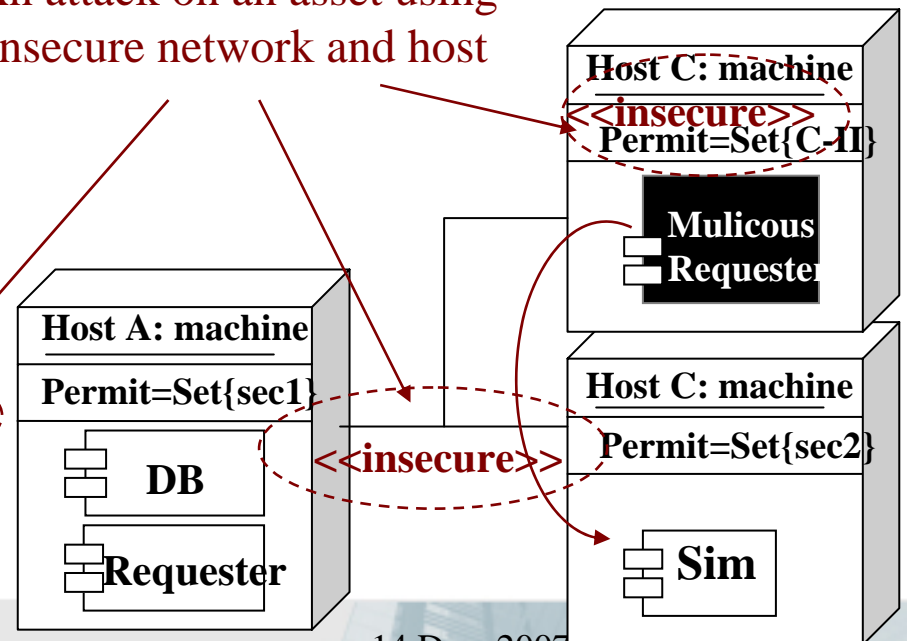
Design of an attack against vulnerable parts



- confirmation and find of vulnerability

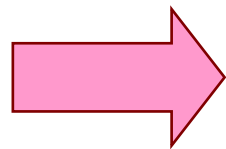


An attack on an asset using insecure network and host



# Security Modeling based on Attack Patterns

- Difficulty of Covering all vulnerability
  - We need to check combinations of insecure parts in data flow and deployments
- Difficulty of Consistency Check between vulnerability, attacks and counter-measures



**Attack Patterns abstracting attacks**

- We can cover all by applying patterns
- Clearly relation between vulnerability, attacks and counter-measures

# Definition of Attack Patterns

## ■ Application Context

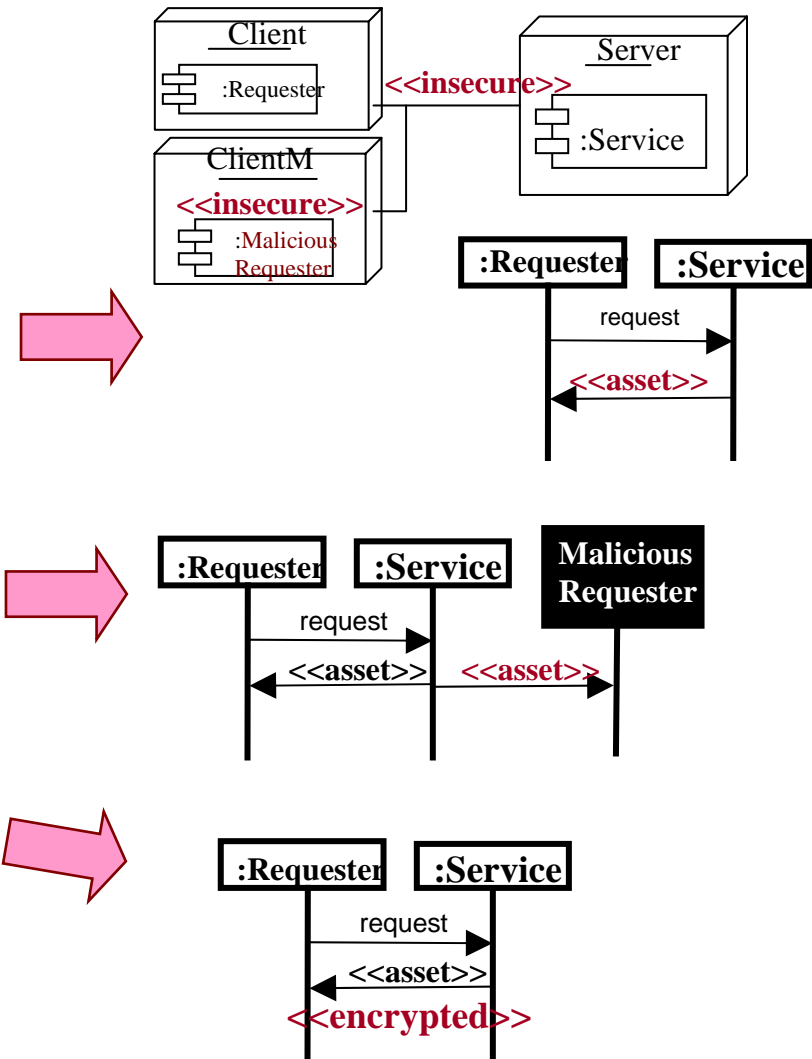
- Specify environments and context to be attacked
- Asset and vulnerability are specified

## ■ Method of Attack: Problem

- Procedure of attacks

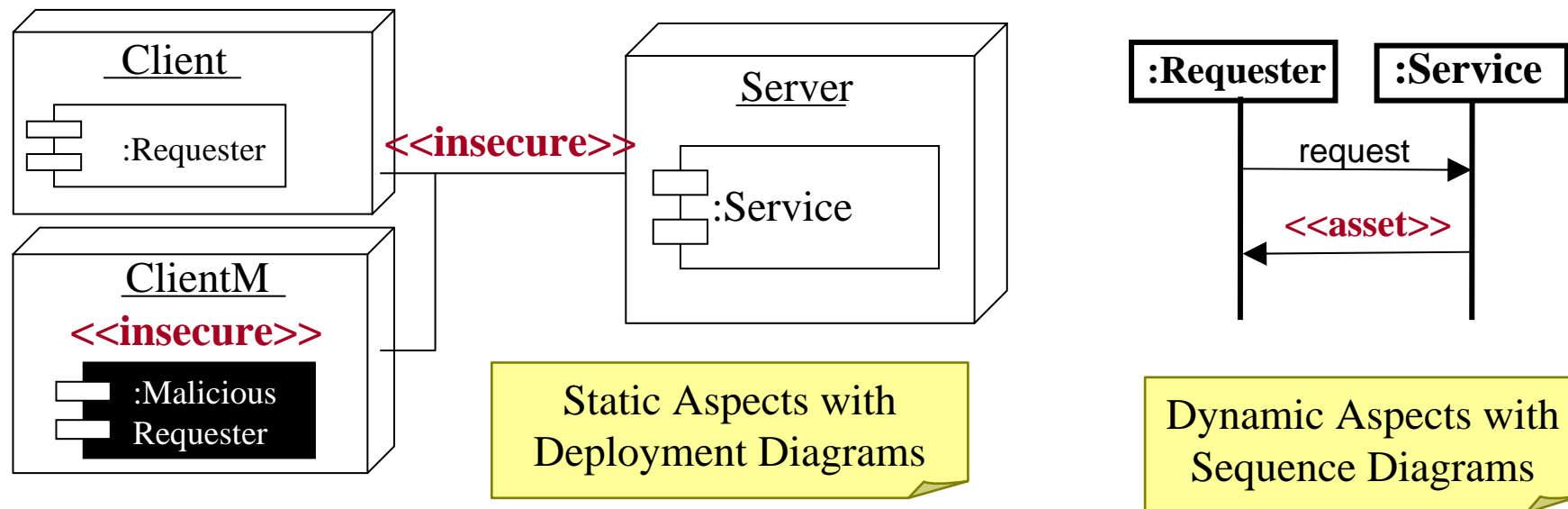
## ■ Solution

- Counter-measures with security functions



# An Example: Wiretapping Pattern (1 / 2)

Application Context Specifies environments and context to be attacked

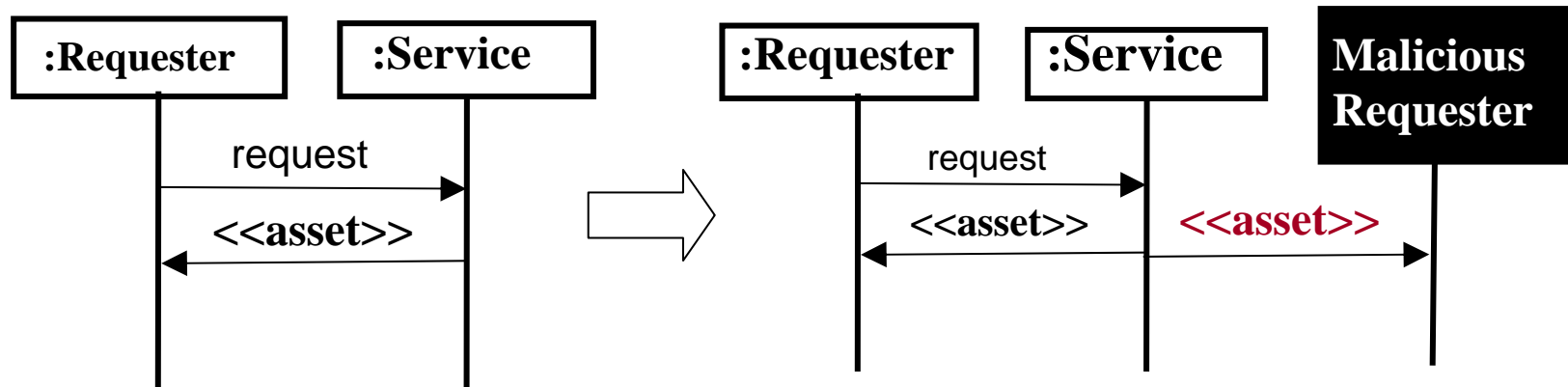


Definition of insecure: members who don't have a permission to access the asset can access to client host

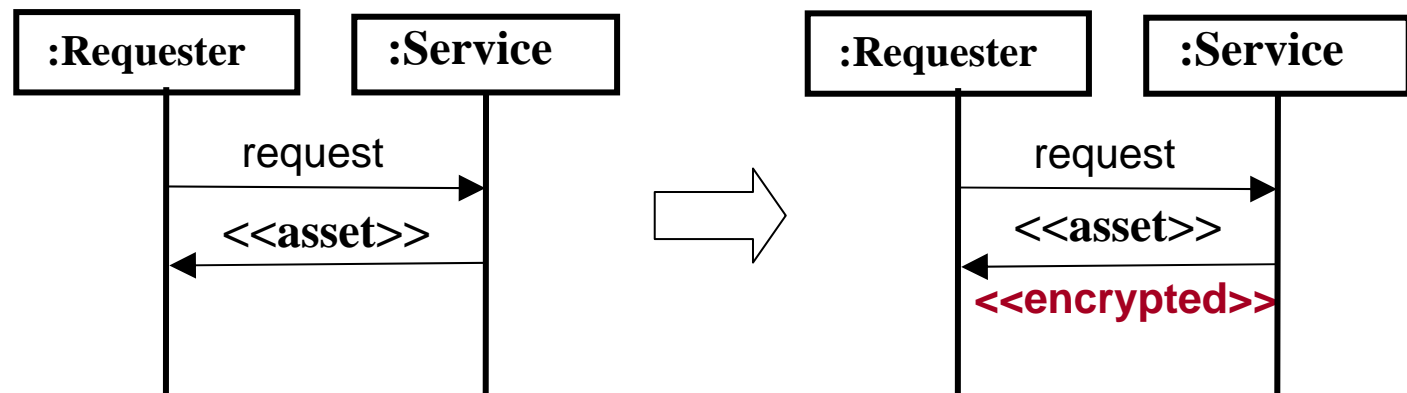
Precondition: a malicious requester can be created in an insecure host

# An Example: Wiretapping Pattern (2/2)

Method of Attack specifies attack sequence against systems

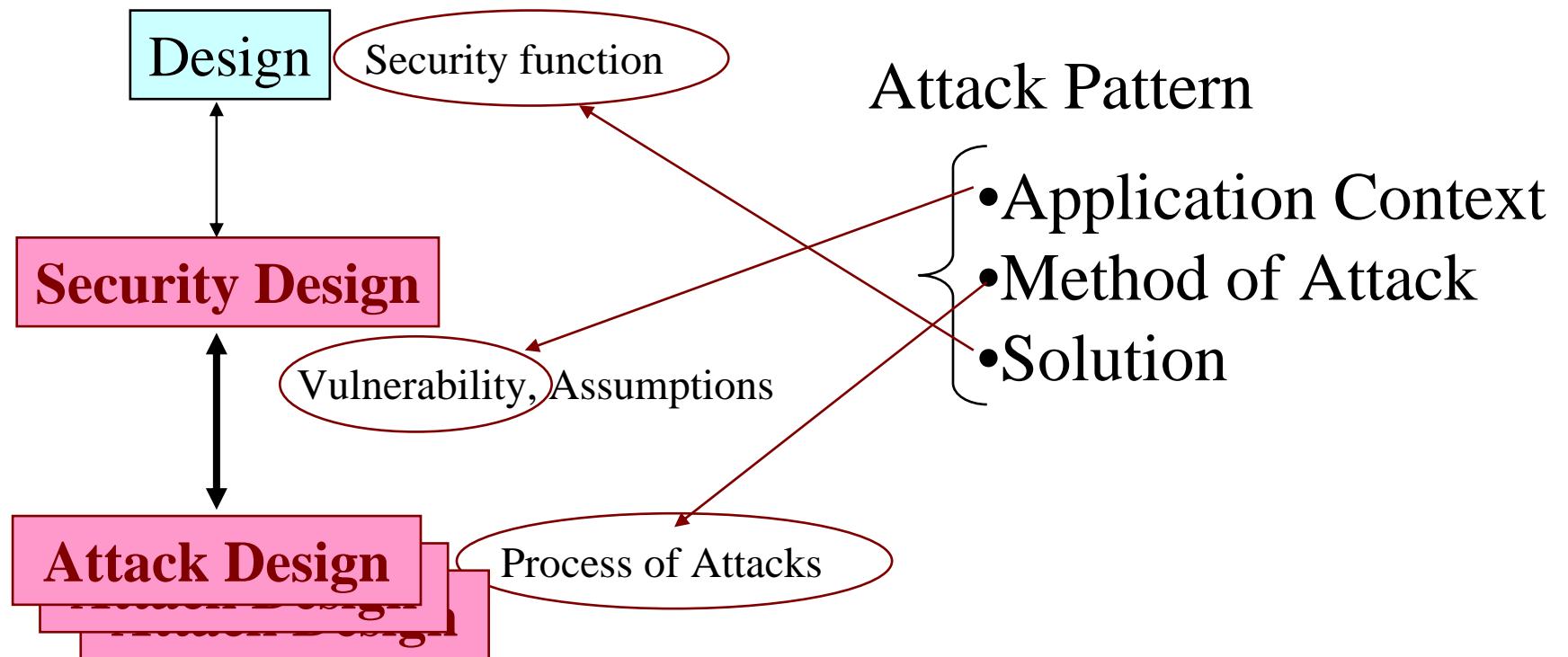


Solution includes security functionalities



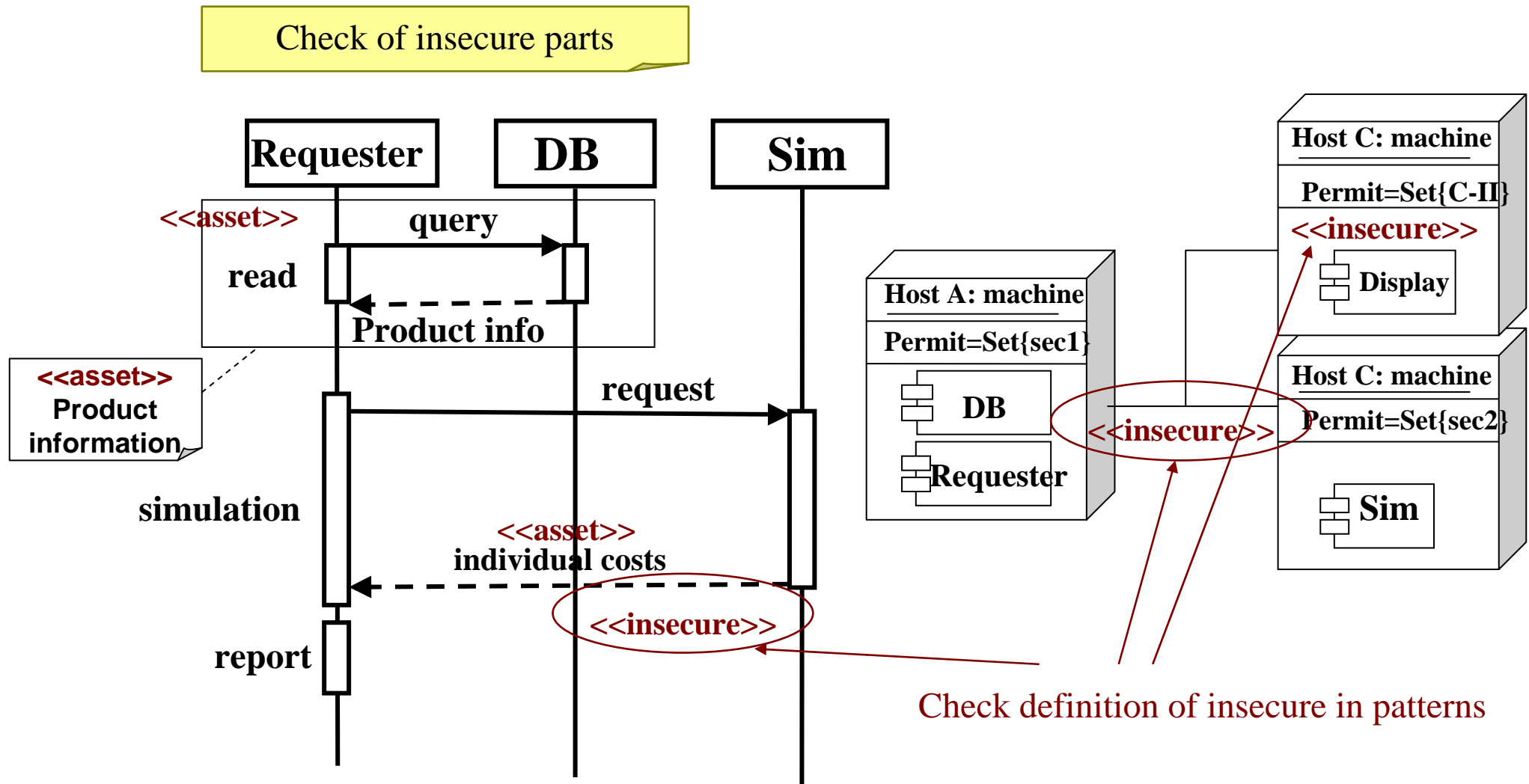
# Modeling Support with Attack Patterns

Attack Patterns bridge the gap between models



# Application of Pattern (1 / 3)

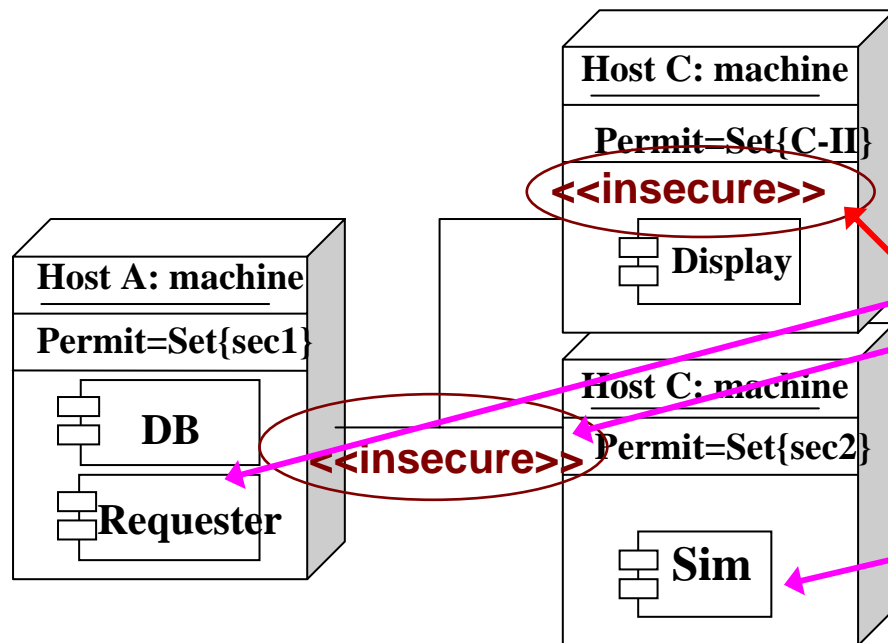
Check of insecure parts



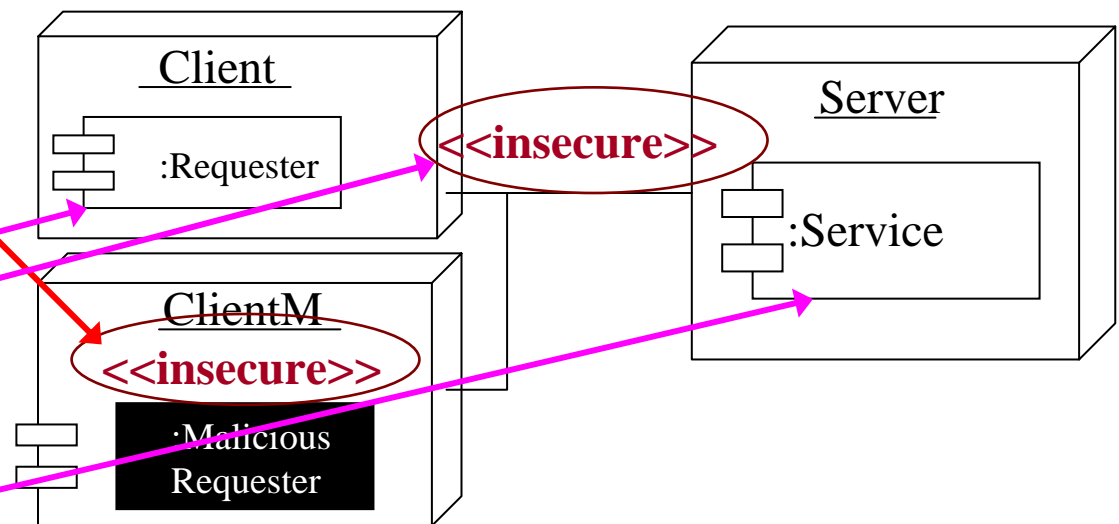
# Application of Pattern (2 / 3)

Check structure of application context using deployment diagrams

Deployment Diagram of a target system



Application Context in Pattern

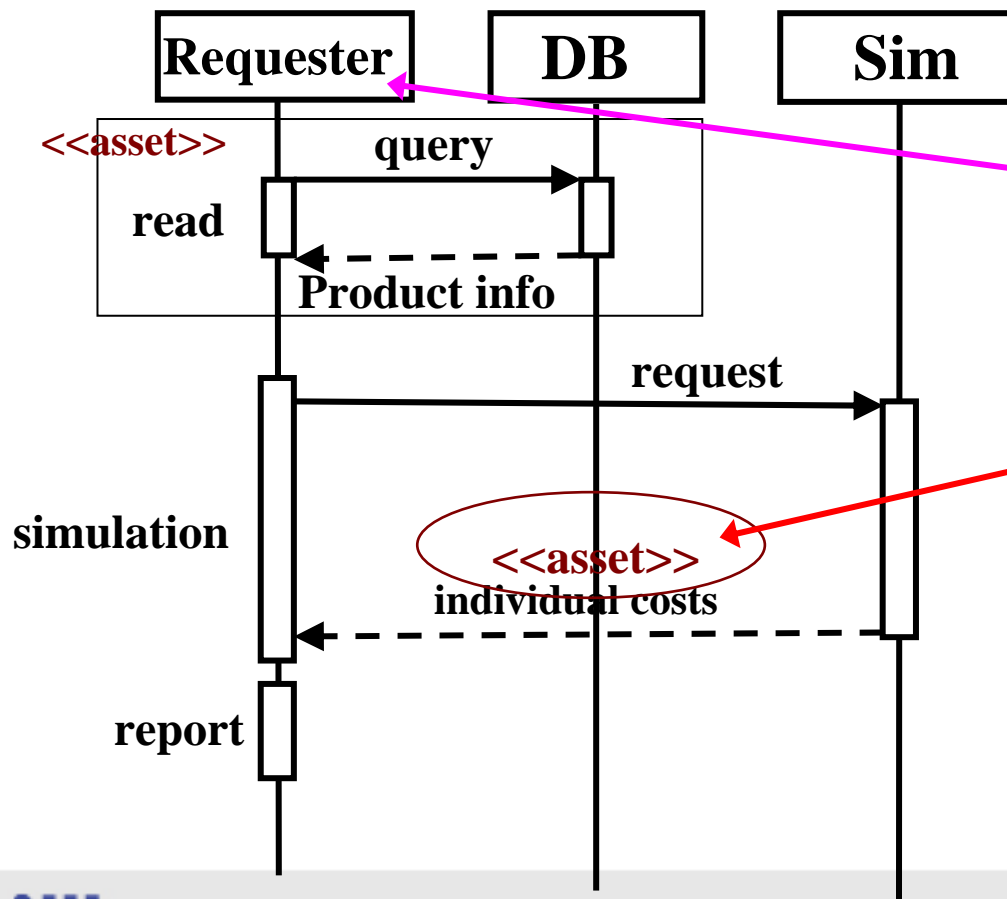


Check correspondence of insecure and components between system model and pattern

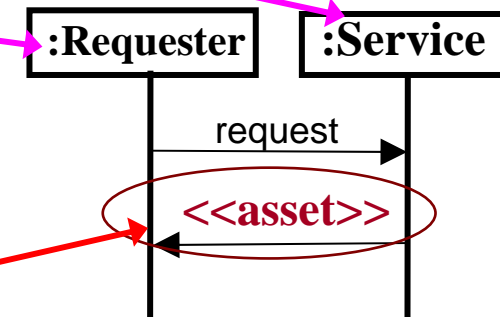
# Application of Pattern (3 / 3)

Check data flows in application context using sequence diagrams

Sequence Diagram of a target system



Application Context in Pattern



Check correspondence of asset between models

# Conclusion

- Categorize Security Patterns from the development process view point
  - Discussion: Efficiency of Security Patterns
  - ➔ Need Attack Design and Relation between requirements and Implementation
- ➔ Propose: A New Security Design Method
  - Stepwise development including Attack Design
  - Attack Patterns support Security Modeling
    - ◆ **Support vulnerability analysis**
    - ◆ **Relation between vulnerability, attacks and counter-measures**

# Future works

- Definition of Models: Syntax and Semantics
- Evaluation based on Example
- Provide Many Patterns
- Methodologies using Security Patterns
- Tool Support
  - Auto-detection of
    - ◆ vulnerability, insecure parts
    - ◆ Application Context
  - Semi-auto instantiation of attacks and counter-measures