

Security Patterns: A Method for Constructing Secure and Efficient Inter- Company Coordination Systems

EDOC 2004@Montrey

Nobukazu Yoshioka
Shinichi Honiden
Anthony Finkelstein

2004/9/23

1

EDOC 2004@Montrey

Security Patterns

Difficulties of Inter-company coordination systems

- Efficient implementation
 - Patterns: Abstract Templates of System Behavior
 - Costs: Performance associate with patterns
 - Method: Abstract Design + Patterns → Detailed Design
- Security issues: security hole, adverse impact on performance
 - Security Patterns
 - Security Costs

2004/9/23

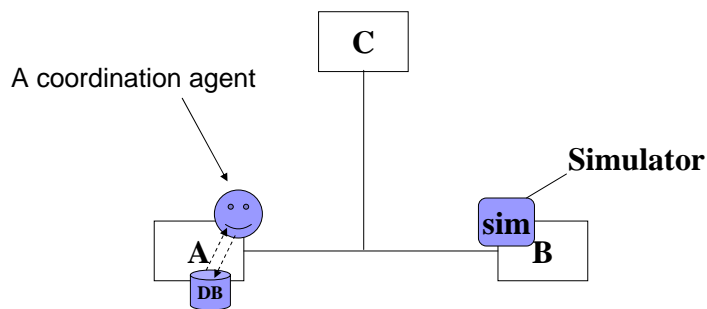
2

Outline

- Simple Example
- Problems & Motivations
- Our Solutions
- Application Example
- Conclusions & Future Work

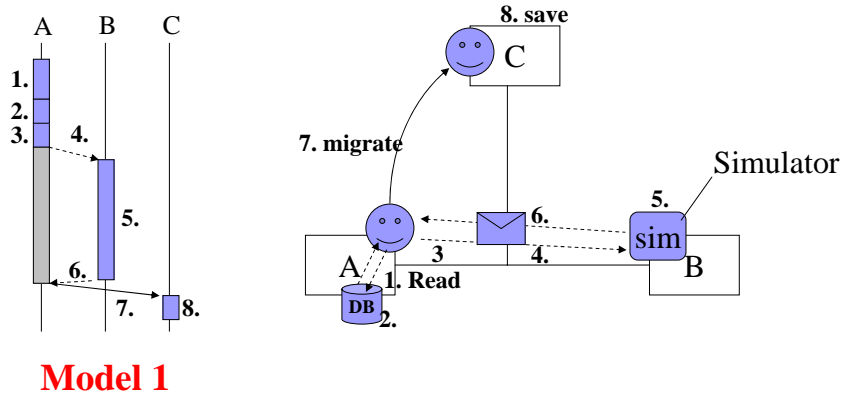
Simple Example

- Coordination:**
1. At first, an agent is in Host A
 2. Reads database in Host A
 3. Uses simulator in Host B
 4. Finally, saves the result in Host C.



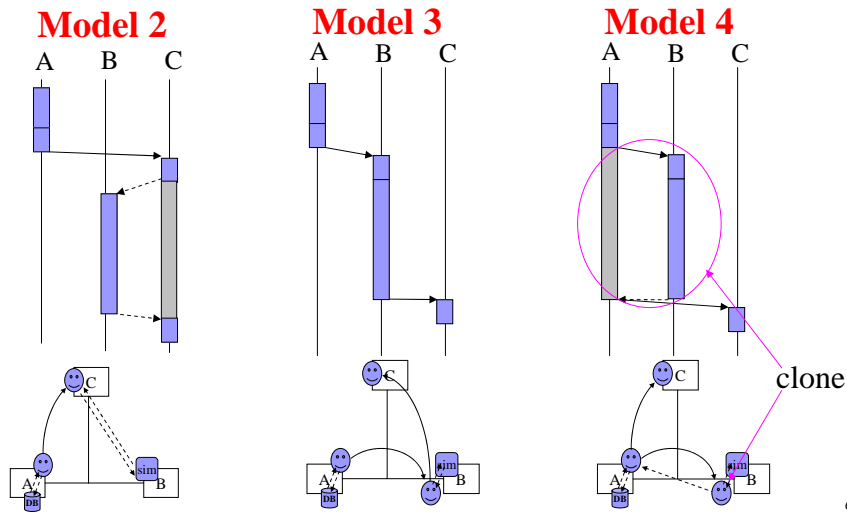
Behavior example: Model 1

Reads database in host A and uses simulator by message passing then migrates to host B, finally saves the result.

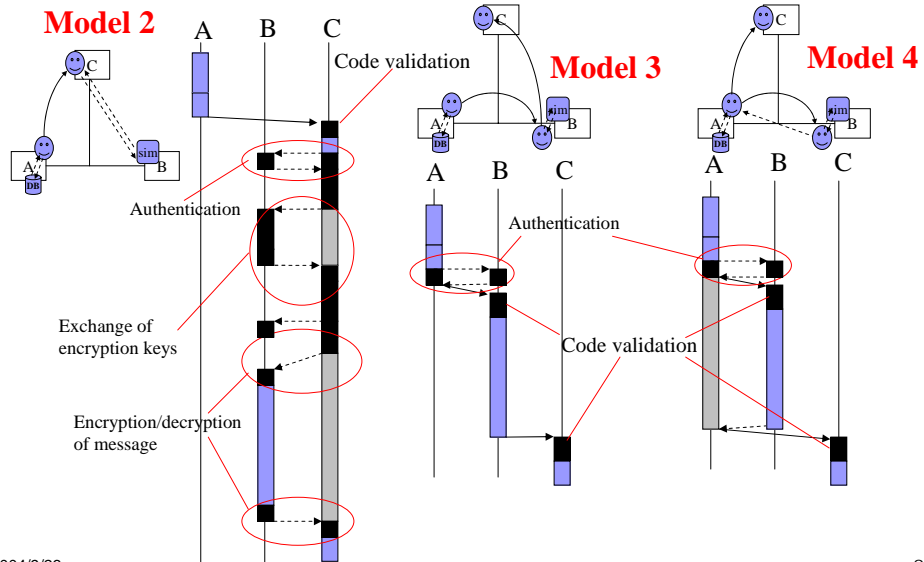


Other Behavior models

We have many candidates even in the very simple example.

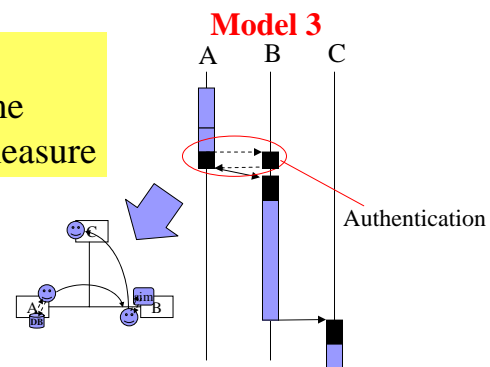


Other Behavior models



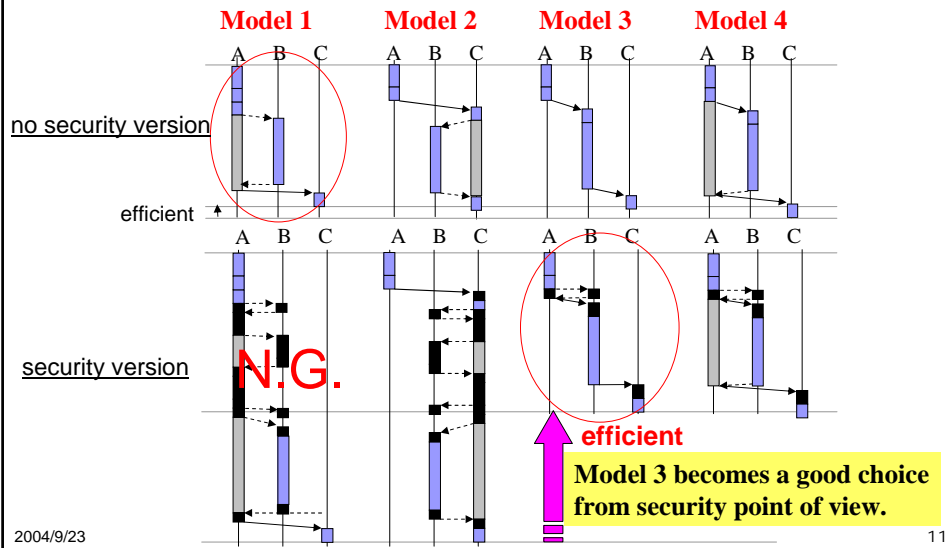
Evaluation of models by Cost

Cost: computation time
 + communication time
 + time for security measure



	Model 1	Model 2	Model 3	Model 4
Cost (sec)	64.32	68.52	34.52	34.54

Evaluation of models



Problem 1:

Security version \neq No security version as for the most efficient model

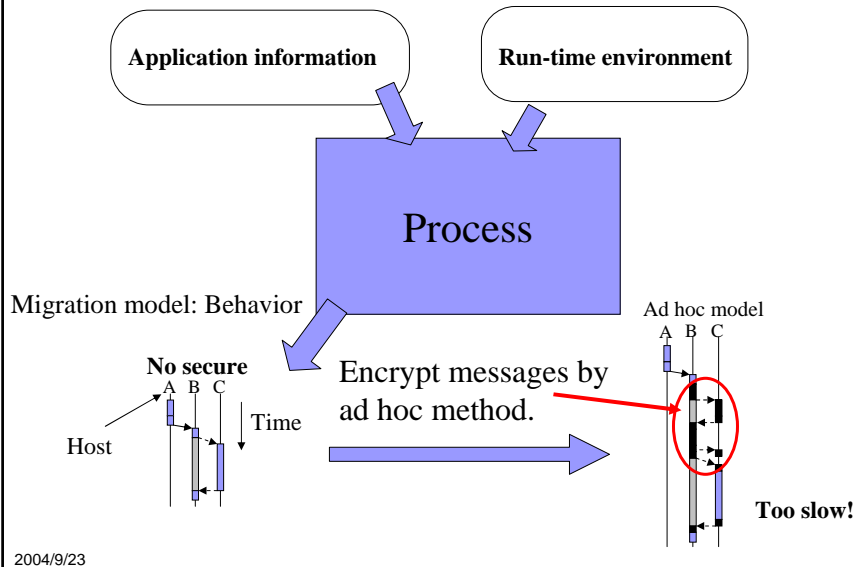
- The most efficient model in security version is not necessarily the same as that in no security version.

- no security version : **Model 1** (RPC)
- security version: **Model 3** (Migration)

- We must not design security version based on non-security version.

→ **We have to consider behavior model with security measure.**

Previous solution



Problem 2

- Extra security measures tend to be considered which may make behavior model too slow.
- Difficulties in selecting the implementation details
 - The appropriate implementation details depend on the situation (run-time environment).
- Difficulties in realizing security
 - It is not easy to realize security efficiently.

➔ **We also have to construct systematic approach instead of ad hoc approach.**

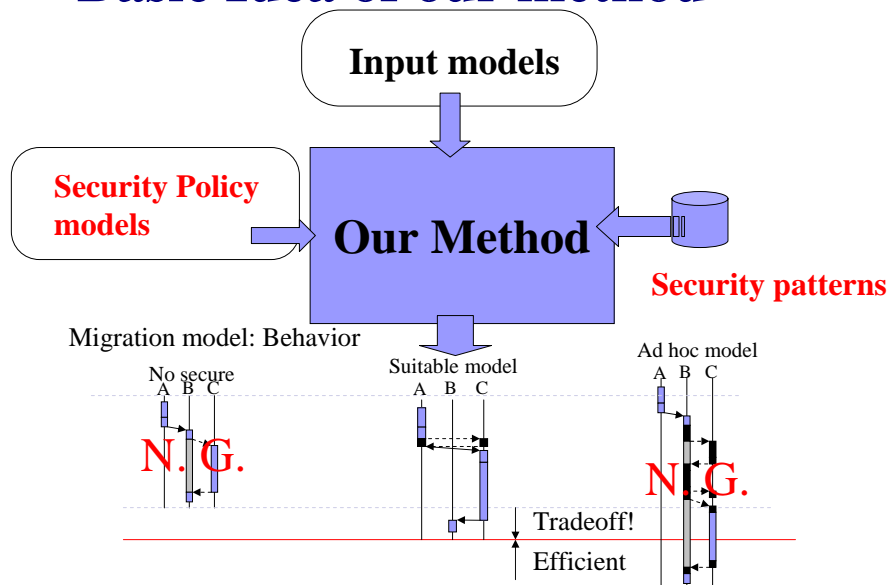
Systematic approach should

- Detect danger part in the application on the target run-time environment.
- Consider security issue in design phase
 - Design appropriate model which doesn't include security holes
 - Design efficient model which doesn't include extra security measure
- Select the most efficient model

2004/9/23

15

Basic Idea of our method



2004/9/23

16

Our Solutions

- How to construct secure systems?

Abstract design models: Static deployments + Application logic



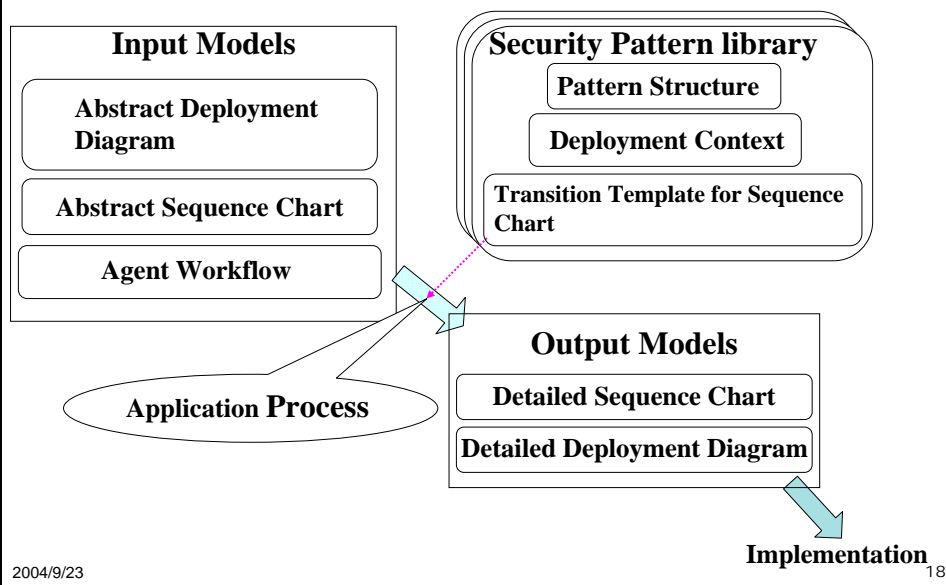
Security Patterns:

add Dynamic deployments + Security measures

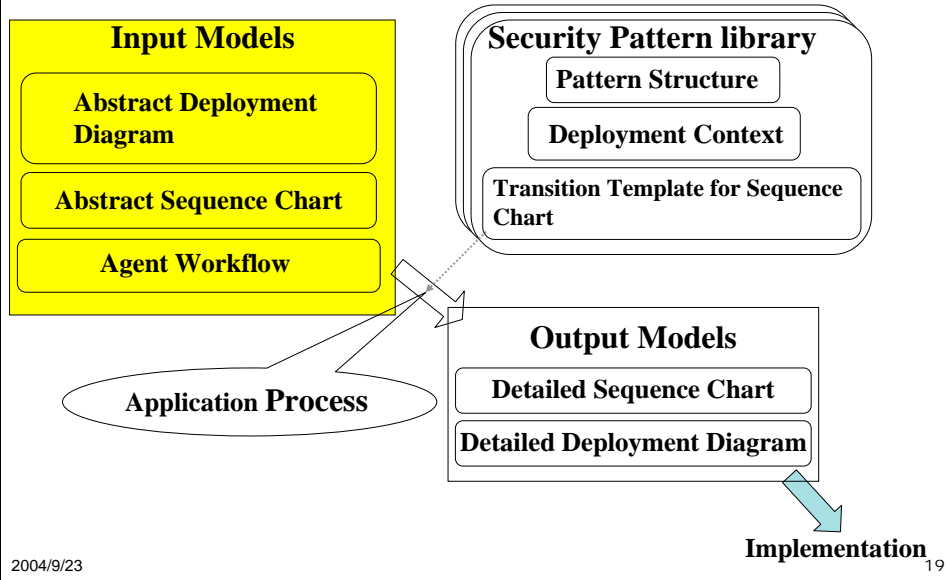
Detailed design models

- How to specify security policy?
 - ➔ Security patterns: Run-time deployment context + Templates
- How to select efficient behavior?
 - ➔ Costs: Performance data of patterns

Overview of Our Method

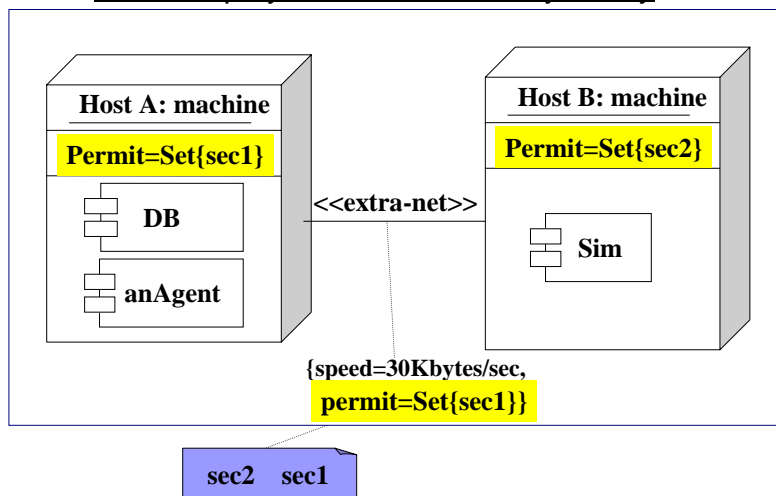


Overview of Our Method



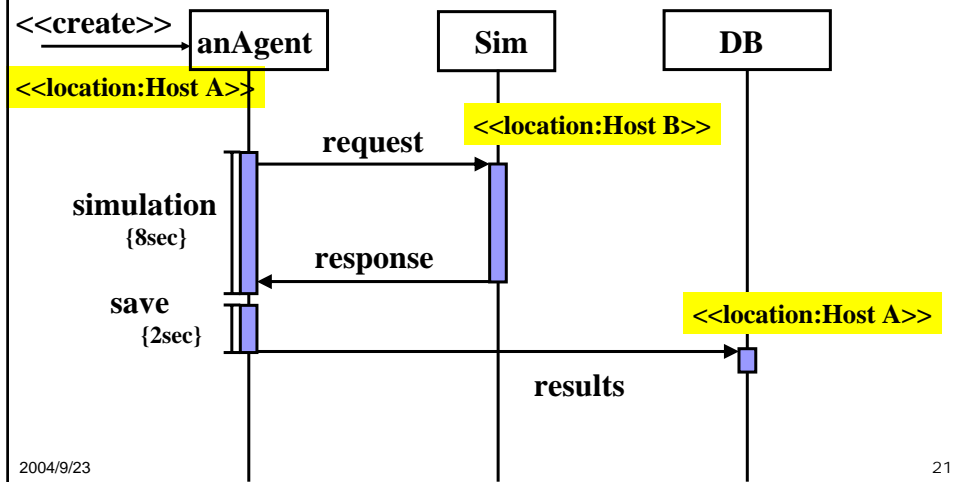
An Example of Abstract Deployment Diagram

Static Deployments with Security Policy



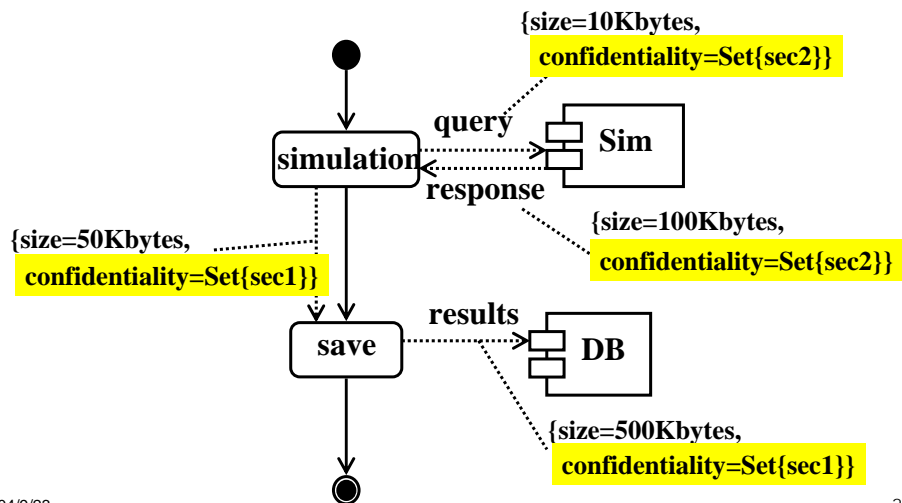
An Example of Abstract Sequence Chart

Application logic without security + Location information

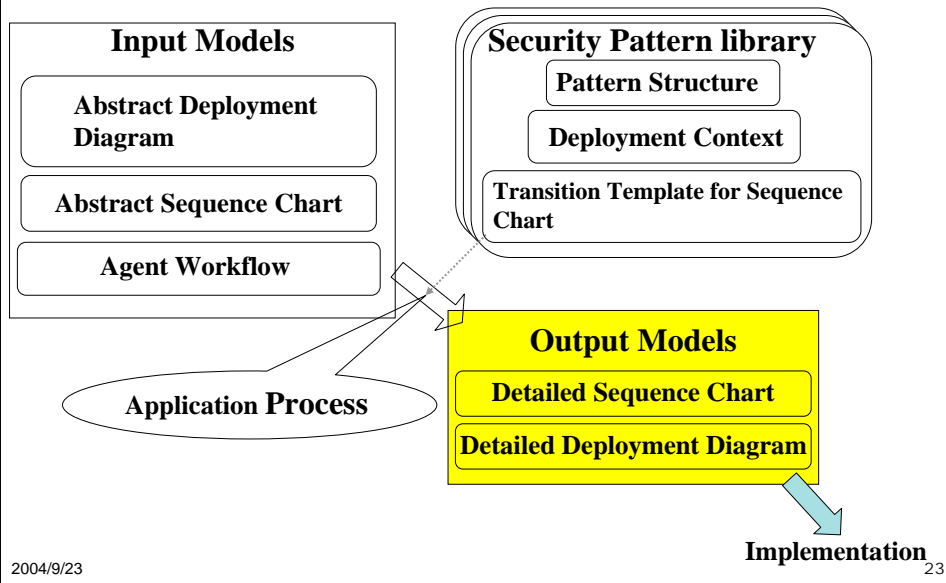


An Example of Agent Workflow

Coordination logic + Security Policy



Overview of Our Method

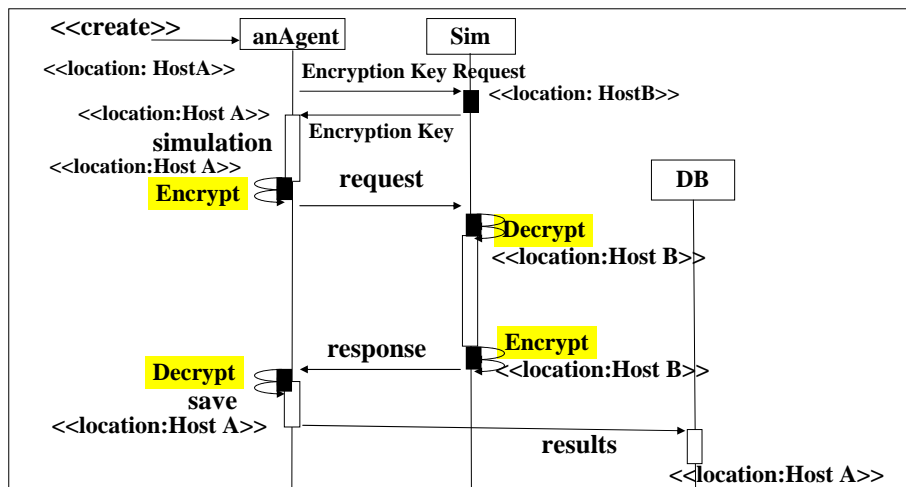


2004/9/23

23

An Example of Detailed Sequence Chart

Application logic + Behaviors + Security Measures

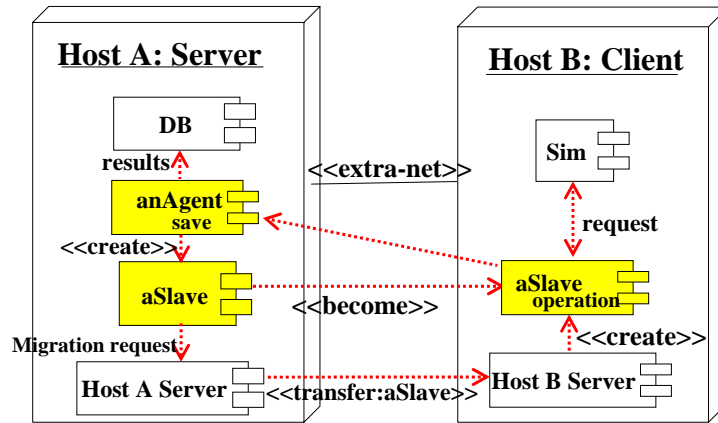


2004/9/23

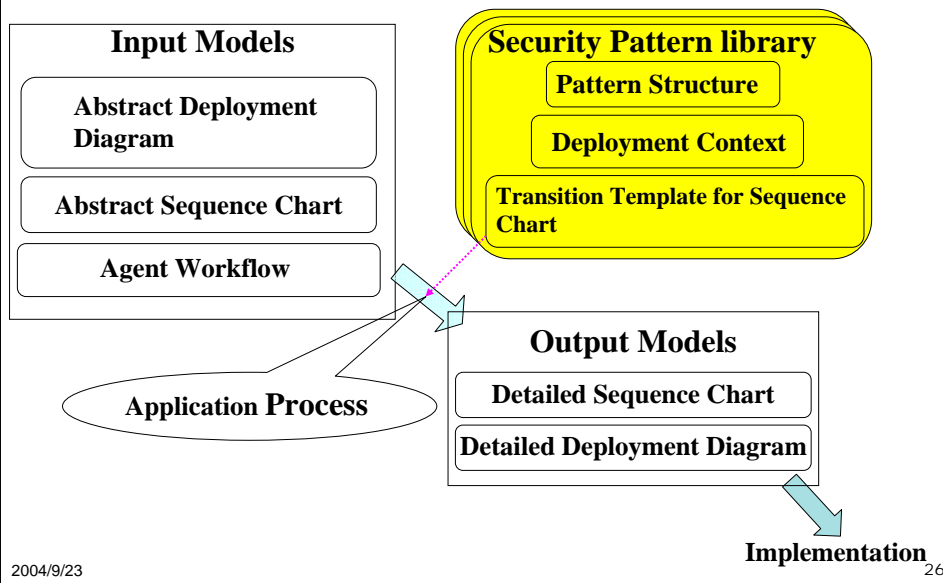
24

An Example of Detailed Deployment Diagram

Dynamic Deployments of Agents

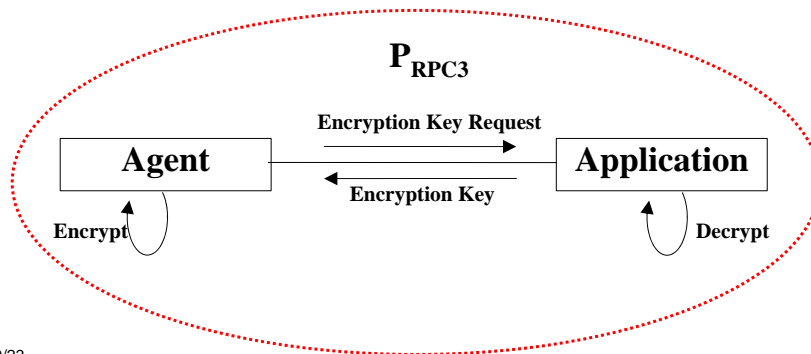


Overview of Our Method



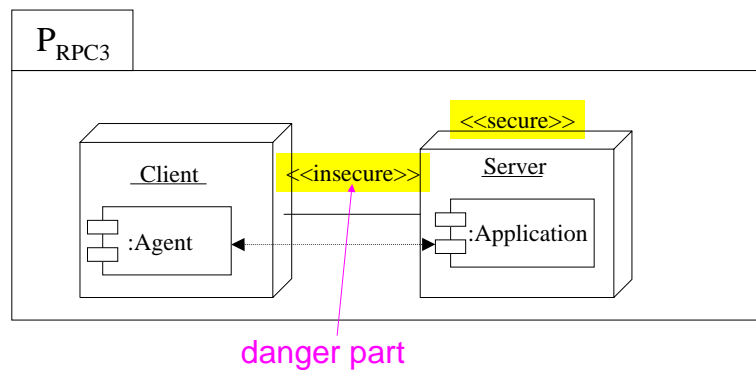
Pattern Structure

Describes the collaborations of agents and applications in terms of security measure



Deployment Context

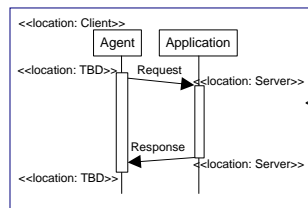
Describes the security context of patterns



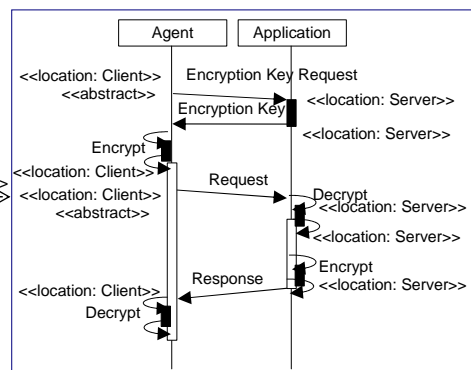
Transition Template for Sequence Chart (P_{RPC3})

Illustrates how to get a Detailed Sequence Chart from an abstract one

Applicable Template (before transition)



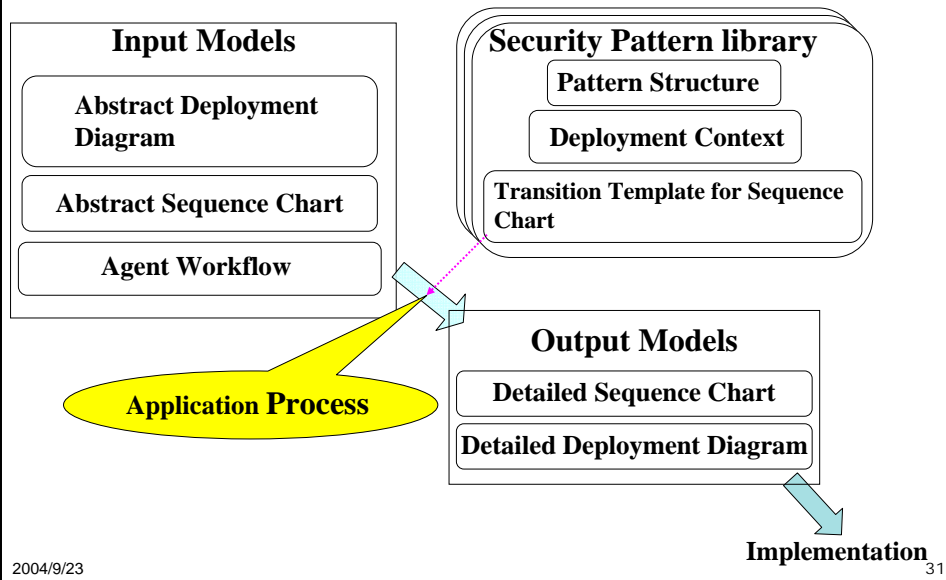
Sequence Chart Template (after transition)



Security Pattern library

Danger Part		Applicable patterns			Security measure
Destination host	Network	Remote message	cloning	Mobile agent	
secure	secure	P _{RPC1}	P _{MS1}	P _{MO1}	signature
insecure	secure	P _{RPC2}	P _{MS2}	P _{MO2}	auth. & signature
secure	insecure	P _{RPC3}	P _{MS3}	P _{MO3}	enc.
insecure	insecure	P _{RPC4}	P _{MS4}	P _{MO4}	auth. & enc.

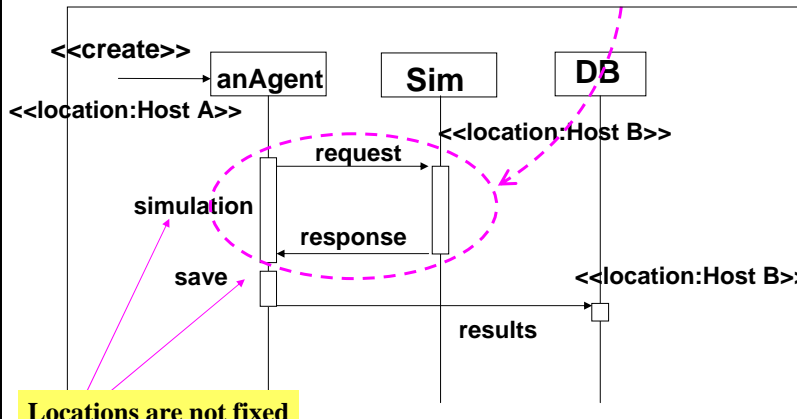
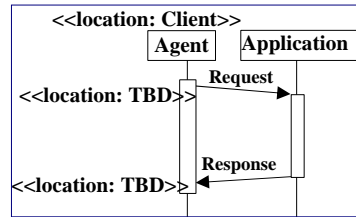
Overview of Our Method



Application Process

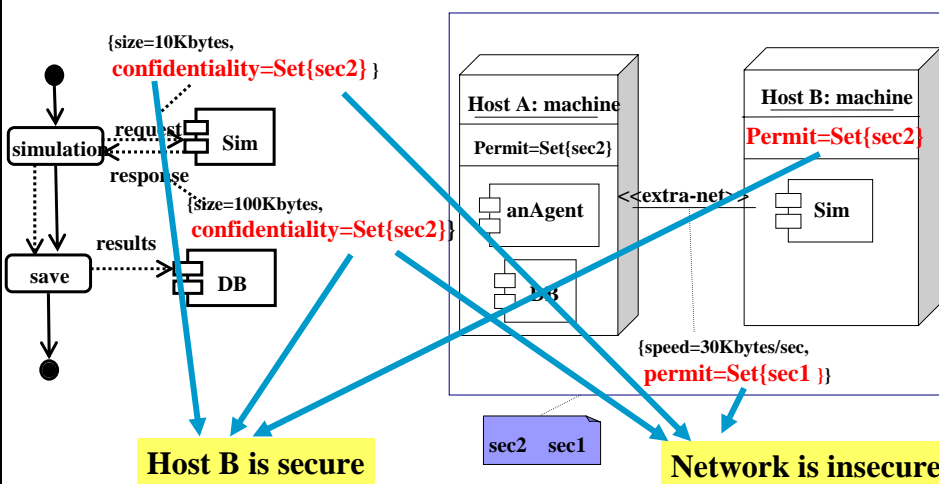
1. while (application parts exist){
 - 1.1 Look for applicable part in an **Abstract Sequence Chart**
 - 1.2 Detect danger parts using an **Agent Workflow** and an **Abstract Deployment Diagram**
 - 1.3 Select applicable patterns
 - 1.4 Apply the patterns to **input models**
2. **Calculate performance** of output models and Select one

1.1 Look for applicable part in an **Abstract Sequence Chart**



Locations are not fixed

1.2 Detect danger parts using an **Agent Workflow** and an **Abstract Deployment Diagram**



Host B is secure

Network is insecure

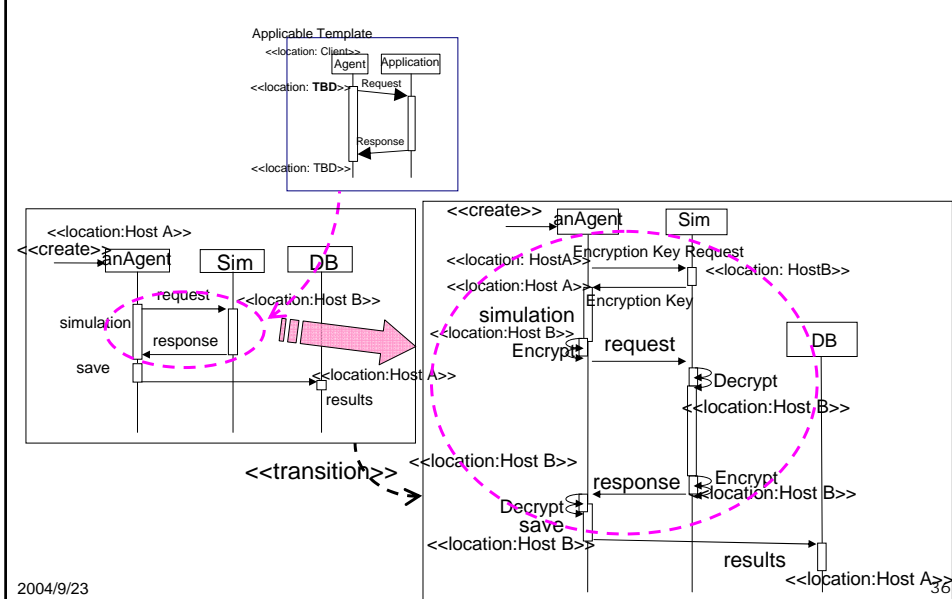
1.3 Select applicable patterns using danger parts information

Danger Part		Applicable patterns			Security measure
Destination host	Network	Remote message	cloning	Mobile agent	
secure	secure	P_{RPC1}	P_{MS1}	P_{MO1}	signature
insecure	secure	P_{RPC2}	P_{MS2}	P_{MO2}	auth. & signature
secure	insecure	P_{RPC3}	P_{MS3}	P_{MO3}	enc.
insecure	insecure	P_{RPC4}	P_{MS4}	P_{MO4}	auth. & enc.

2004/9/23

35

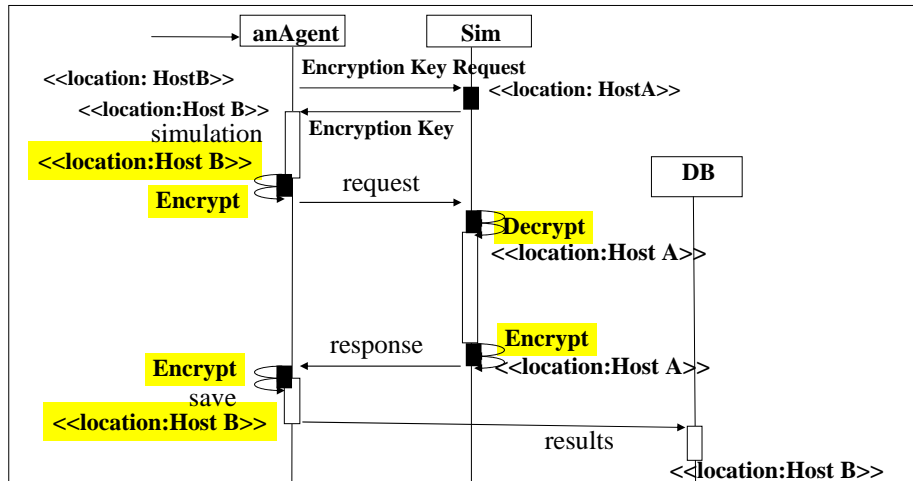
1.4 Apply the patterns to input models



2004/9/23

36

Detailed Sequence Chart applied by Sequence Chart Transition for P_{RPC3}



2004/9/23

37

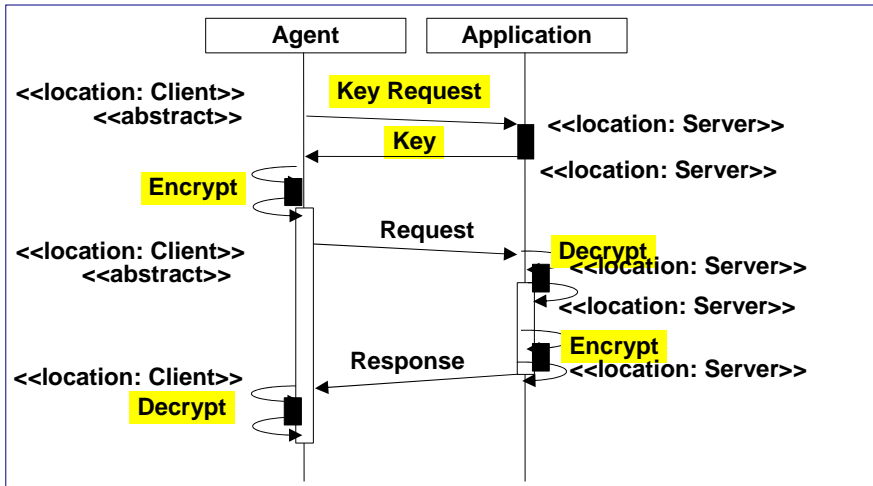
2. Calculate performance of output models and Select one

- Total time = $\sum CT + \sum MT + \sum SM$
- CT: computation time
- MT: message communication time
- SM: time for security measure
 - Time for **authenticating** hosts
 - Time for **obtaining** encryption/signature **key**
 - Encryption and decryption** time
 - Signing and verification** time

2004/9/23

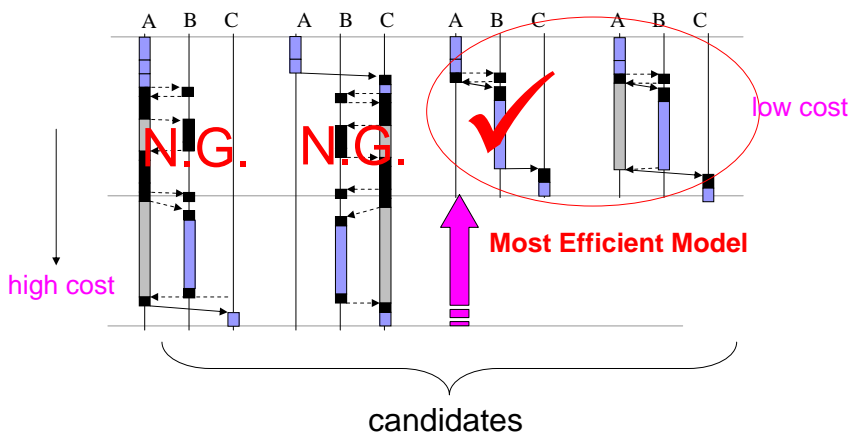
38

Time for Security Measure of RPC3



key-exchange + 2 x (encryption + decryption)

Selection of efficient model



Conclusions

- Design Method:
 - Secure and efficient distributed system:
 - ⇒ Security patterns
 - Guideline for selecting proper behavior: efficiency points of view
 - ⇒ Costs: Performance data of patterns

Future work

- CASE tools
- Dynamic environment/behavior changes
- Customization of patterns: Validation of patterns