System modelling

Marco Domenico Aime

POLITECNICO DI TORINO

26th September 2006
Wroclaw

DESEREC

Dependability and Security by Enhanced Reconfigurability

Information Society Technologies

PDF created with pdfFactory Pro trial version www.pdffactory.com

Agenda

n models for network simulation

n models to describe complex systems

n UML-based models for security and QoS

n models for web service architectures

PDF created with pdfFactory Pro trial version www.pdffactory.com

Network Simulator (NS)

- discrete event simulator targeted at networking research
- provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks
- languages used with ns2
  - C++ : to implement protocols
  - OTcl: to write simulation scripts
- system description based on a basic graph structure
- no addresses and interface concepts
- no environment definition

Jae Chung and Mark Claypool, NS by Example available at http://nile.wpi.edu/NS/

PDF created with pdfFactory Pro trial version www.pdffactory.com

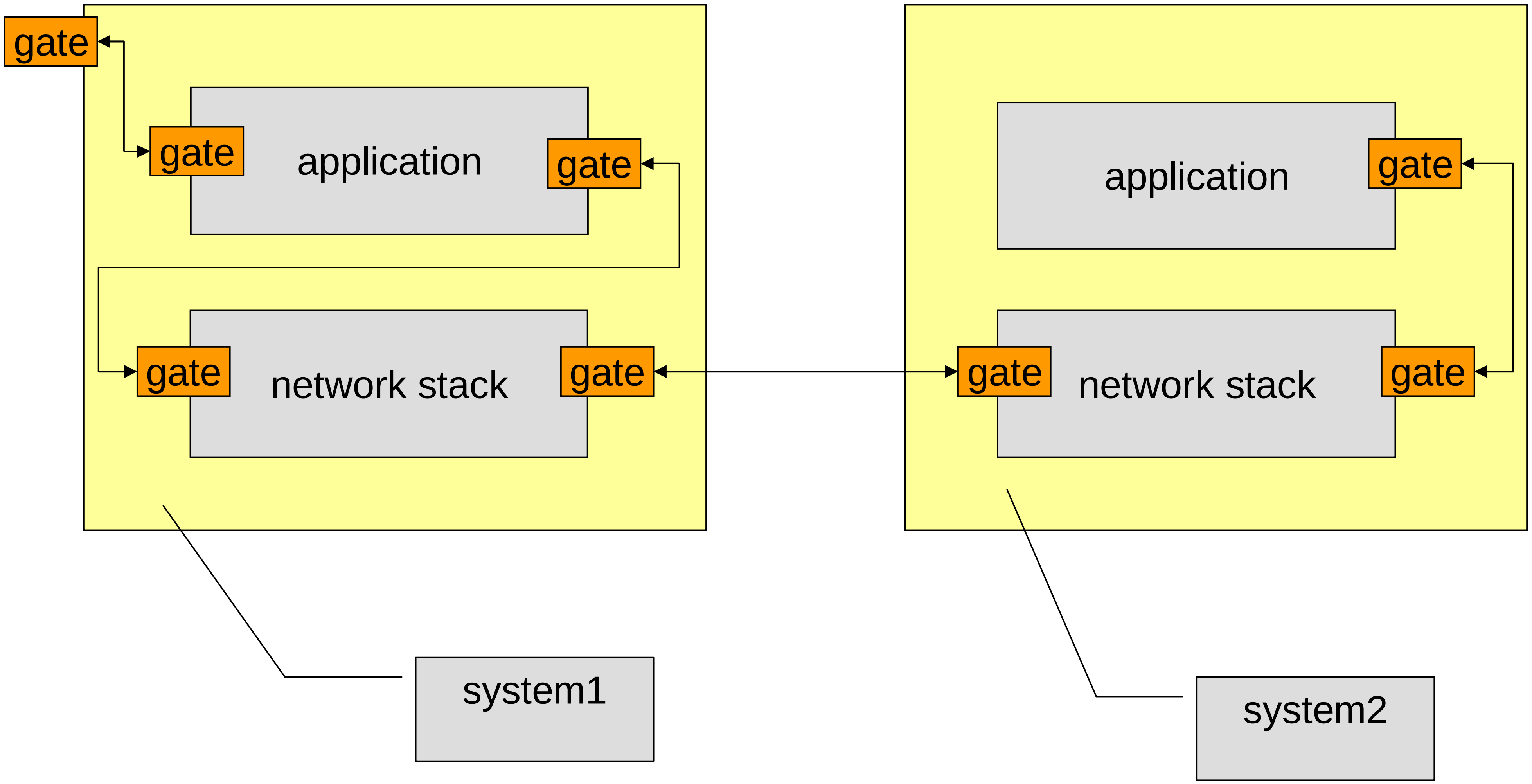
Network Description (NED) by Omnet ++

- n developed for Omnet++ network simulator
- n object oriented description
- n based on
  - 4 components (simple or nested)
  - 4 gates used as external point of access
  - 4 connections to exchange messages between components via gates
- n structured and extensible architecture
- n no services
- n no environment

OMNeT++ homepage available at: http://www.omnetpp.org

PDF created with pdfFactory Pro trial version www.pdffactory.com

NED (Example)



PDF created with pdfFactory Pro trial version www.pdffactory.com

Domain Modeling Language

- n developed for the SSFNet network simulator
- n represents networks using
  - 4 host: id and interface
  - 4 router: like hosts but with more than one interface
  - 4 links: connections between nodes or interfaces
  - 4 traffic: defines actors of a data stream through the network
- n no addresses
- n no environment
- n no software features

DML specifications available at:
http://www.ssfnet.org/ssfdocs/dmlreference.html

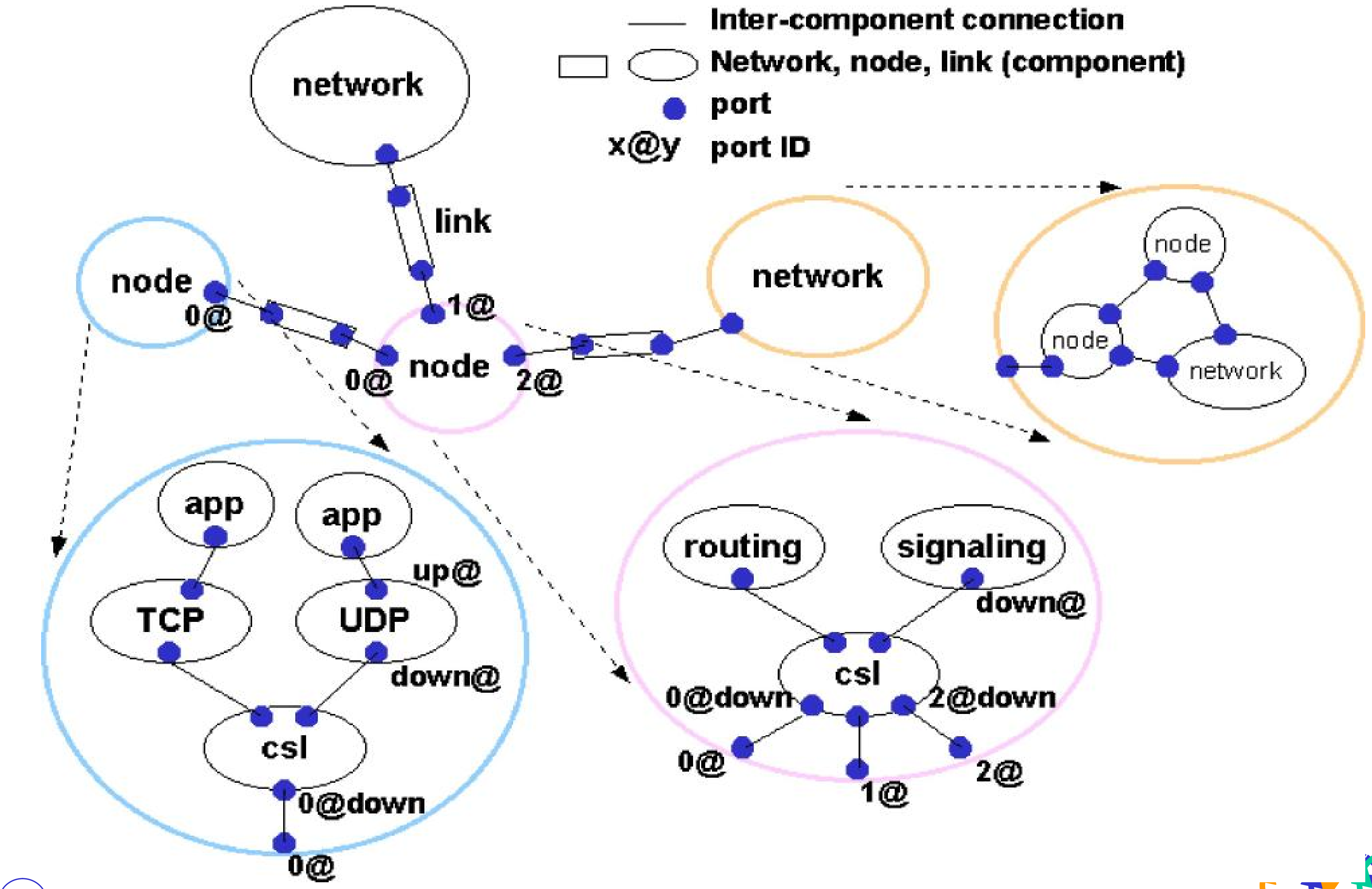
PDF created with pdfFactory Pro trial version www.pdffactory.com

JSim

- Java network simulator
- based on a component architecture: Autonomous Component Architecture (ACA)
- networks are composed by hosts
- every host is created with
  - applications and protocol modules
  - static Core Service Layer (CLS) representing network, data link and physical layers
- modular, easily extensible
- hard-coded (it’s a java library)
- no details on lower network levels

J-Sim homepage, available at http://www.j-sim.org

PDF created with pdfFactory Pro trial version www.pdffactory.com



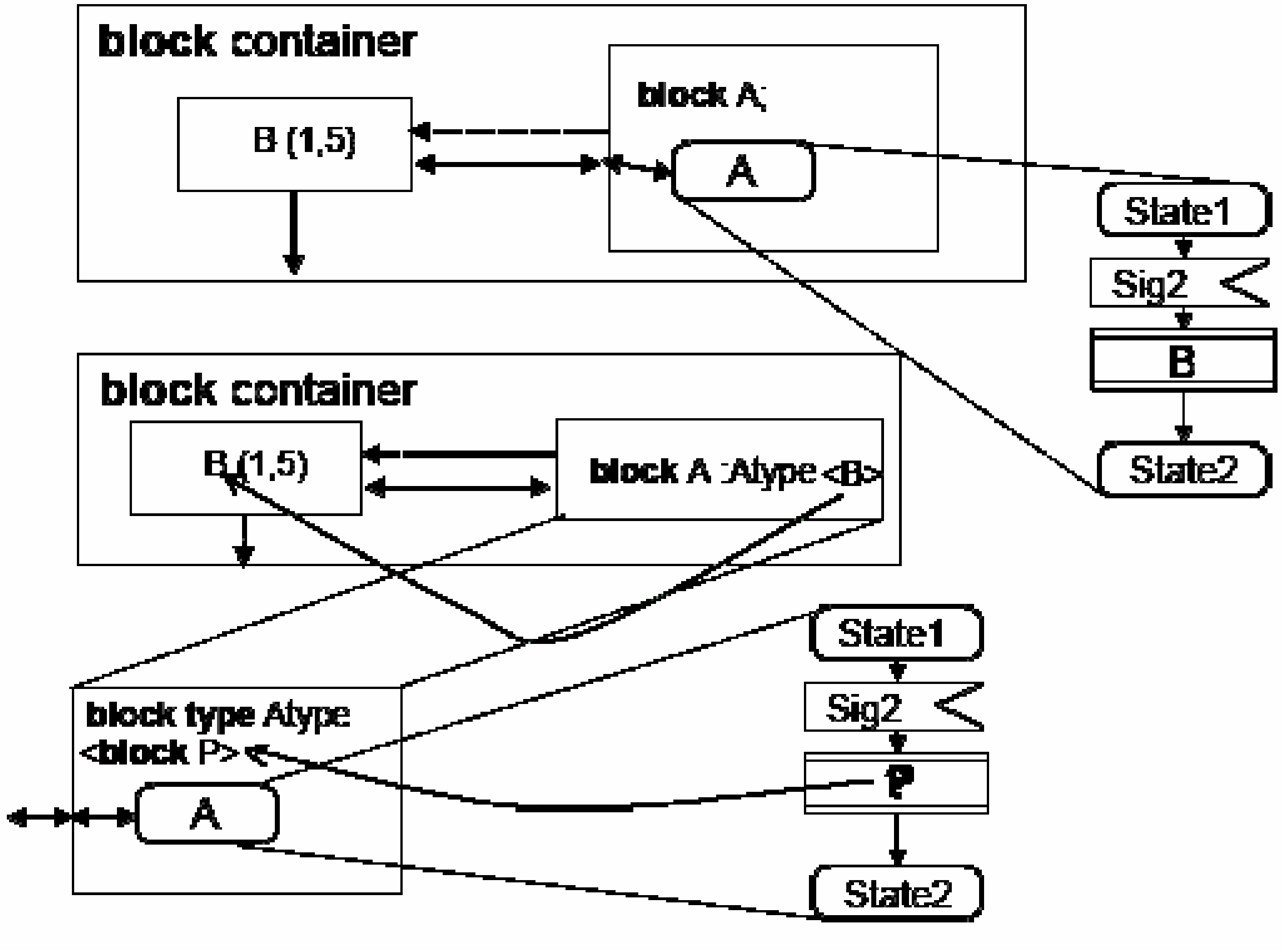
PDF created with pdfFactory Pro trial version www.pdffactory.com

Specification and Design Language

- defined by ITU-T
- graphical specification language, designed for
  - real-time applications
  - process control
  - telecommunication systems
- describes behaviours, using communicating finite state machines
  - uses processes and signals exchanged between them and/or the environment
- suitable for the description of real-time, stimulus-response systems
- not focused on ICT / network description

PDF created with pdfFactory Pro trial version www.pdffactory.com

Example



PDF created with pdfFactory Pro trial version www.pdffactory.com

SysML

- subset of UML 2.0 with extensions
- provides general purpose modeling language to support specification, analysis, design and verification of complex systems
- provides
  - semantics
  - notation
- doesn’t provide methodology
- too flexible: data model cannot be easily standardised

Laurent Balmelli, An overview of the Systems Modeling Language for product and systems development available at http://www-128.ibm.com/developerworks/rational/library/aug06/balmelli/

PDF created with pdfFactory Pro trial version www.pdffactory.com

SysML – describing systems

n blocks

- 4 provides general-purpose capability to model systems as tree of modular components
- 4 modular units
- 4 may contain both structural and behavioral features

n ports

- 4 interaction point between a block or a part and its environment
- 4 connection between ports via connectors

n flows: represent that something can enter or leaving the block

PDF created with pdfFactory Pro trial version www.pdffactory.com

SysML – systems behavior

- activities
  - emphasize the inputs, outputs, sequences, and conditions for coordinating other behaviors
  - provide a flexible link to blocks owning those behaviours
- interactions
  - using UML 2 sequence diagram
  - describe the flow of control between actors and systems

PDF created with pdfFactory Pro trial version www.pdffactory.com

SysML – describing requirements

- n constraints
    - 4 provide a mechanism for integrating engineering analysis (e.g.: performance and reliability models)
    - 4 identifies and names constraint blocks, but does not specify a computer interpretable language for them
- n requirements
    - 4 provides modeling constructs to represent text based requirements and relate them to other modeling elements
    - 4 in graphical, tabular, or tree structure format

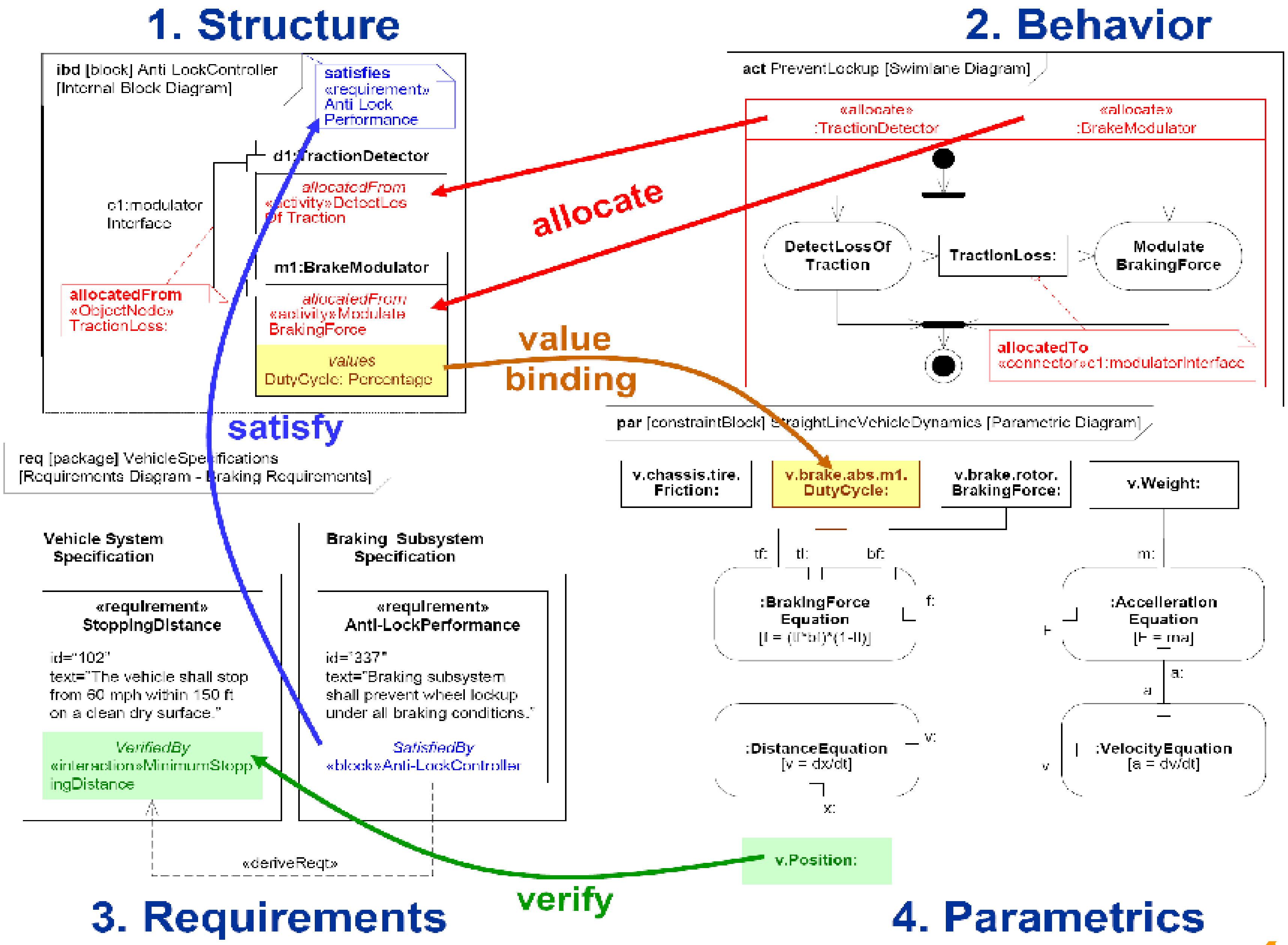
PDF created with pdfFactory Pro trial version www.pdffactory.com

SysML – general view

- structure defined using blocks + ports
- behavior can be define using state machines or sequence diagrams
- policy may be defined using requirements diagrams

PDF created with pdfFactory Pro trial version www.pdffactory.com

SysML – general view



PDF created with pdfFactory Pro trial version www.pdffactory.com

UML for Quality of Service modeling

definition of QoS using:

n characteristics

 4 a named parameter for evaluation to which assign a value

n constraints

 4 limitation imposed on characteristics by system and project specifications

n levels of execution

 4 to define different quality levels via aggregation of constraints

can be associated to any element of the description

n UML Profile for Modeling QoS and FT Characteristics and Mechanisms, v1.0

 4 http://www.omg.org/cgi-bin/apps/doc?formal/06-05-02.pdf

PDF created with pdfFactory Pro trial version www.pdffactory.com

Catalog of QoS characteristics

n coherence

n performance

- 4 throughput
- 4 latency
- 4 efficiency
- 4 demand

n dependability

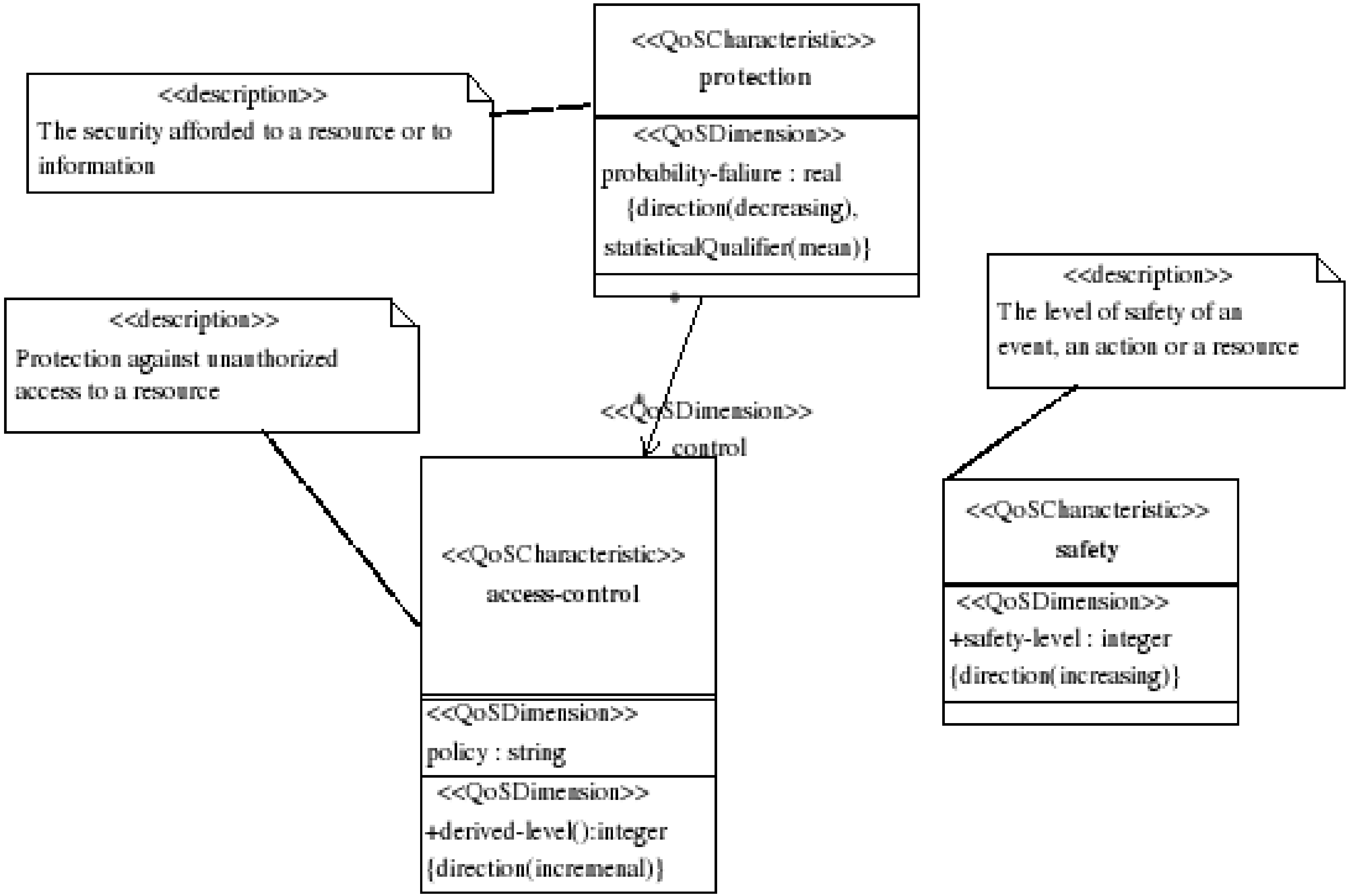
- 4 reliability
- 4 availability

n functionality

- 4 security
- 4 integrity

PDF created with pdfFactory Pro trial version www.pdffactory.com

Example - security



<<description>>
The security afforded to a resource or to information

<<QoSCharacteristic>>
protection

<<QoSDimension>>
probability-faliure : real
{direction(decreasing),
statisticalQualifier(mean)}

<<description>>
Protection against unauthorized access to a resource

<<QoSDimension>>
control

<<description>>
The level of safety of an event, an action or a resource

<<QoSCharacteristic>>
access-control

<<QoSDimension>>
policy : string

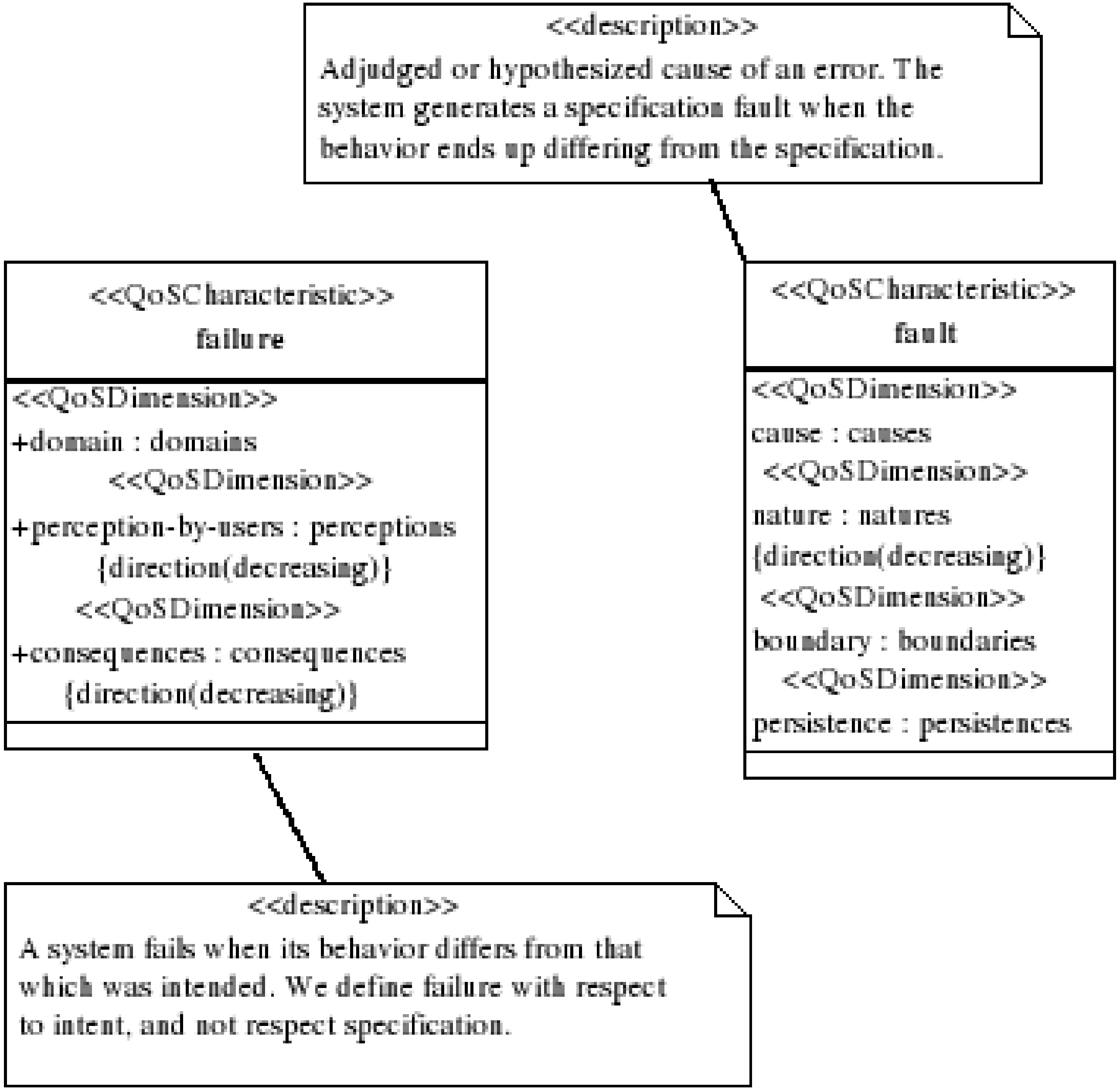
<<QoSDimension>>
+derived-level():integer
{direction(incremenal)}

<<QoSCharacteristic>>
safety

<<QoSDimension>>
+safety-level : integer
{direction(increasing)}

PDF created with pdfFactory Pro trial version www.pdffactory.com

Example – fault and failure



<<description>>
Adjudged or hypothesized cause of an error. The system generates a specification fault when the behavior ends up differing from the specification.

<<QoSCharacteristic>>
failure

<<QoSDimension>>
+domain : domains
<<QoSDimension>>
+perception-by-users : perceptions
{direction(decreasing)}
<<QoSDimension>>
+consequences : consequences
{direction(decreasing)}

<<QoSCharacteristic>>
fault

<<QoSDimension>>
cause : causes
<<QoSDimension>>
nature : natures
{direction(decreasing)}
<<QoSDimension>>
boundary : boundaries
<<QoSDimension>>
persistence : persistences

<<description>>
A system fails when its behavior differs from that which was intended. We define failure with respect to intent, and not respect specification.

PDF created with pdfFactory Pro trial version www.pdffactory.com

UML for Failure Tolerance modeling

- aggregation that associates one or more replicas that are physically located
- clients invoke aggregation and group manager decides which replica executes
- fault detection to get status of replicas and decide
- cooperation between replicas is defined by replication styles
  - only one / more than one running
  - persistent/ transient

PDF created with pdfFactory Pro trial version www.pdffactory.com

UML for security

- UMLsec
  - Developed by Jan Jürjens (TU München, Germany)
  - Towards development of secure systems using UMLsec, Jan Jürjens, 2001
  - http://www4.in.tum.de/~umlsec/

- Secure UML
  - SecureUML: A UML-Based Modeling Language for Model-Driven Security, Torsten Lodderstedt, David Basin, and Jürgen Doser, 2002

- NEW!
  - A Formal Framework for Secure Design and Constraint Checking in UML, Thuong Doan, Laurent Michel, and Steven Demurjian, 2005

PDF created with pdfFactory Pro trial version www.pdffactory.com

UMLsec

- UMLsec is an UML extension (profile) for *secure* systems development
- developed by Jan Jürjens (TU München, Germany)
- goals:
  - evaluate UML specifications for *vulnerabilities* in design
  - encapsulate *established rules* of prudent security engineering
  - make available to developers *not specialized* in security
  - consider security from *early* design phases, in system *context*
  - make verification *cost-effective*

PDF created with pdfFactory Pro trial version www.pdffactory.com

UMLsec

- define labels (*stereotypes*) for UML model elements which, when attached, add security-relevant information to these model elements

- this security-relevant information can be:
  - security assumptions on the physical level (e.g. `«Internet»` stereotype)
  - security requirements on the logical structure of the system (e.g. `«secrecy»` stereotype) or on the specific data values (e.g. `«critical»` stereotype)
  - security policies that sub-systems are supposed to obey (e.g. `«fair exchange»`, `«secure links»`, `«data security»`, or `«no down-flow»` stereotypes)

PDF created with pdfFactory Pro trial version www.pdffactory.com

UMLsec

applies to:

- Use case diagrams
  - Used to described typical interaction between a user and a computer system in requirement elicitation
  - Can be used to capture *security requirements*
- Activity diagrams
  - Used to model workflow and to explain use cases in more details
  - Can be used to make *security requirements more precise*
  - Constraint: after a {buy} state in activity diagram is reached, eventually reach {sell} state.
- Deployment diagrams
  - Used to describe the physical layer of a system
  - Can be used to check the security requirements on the logical level of the system are enforced by level of physical security, or whether additional security mechanisms (such as encryption) have to be employed

PDF created with pdfFactory Pro trial version www.pdffactory.com

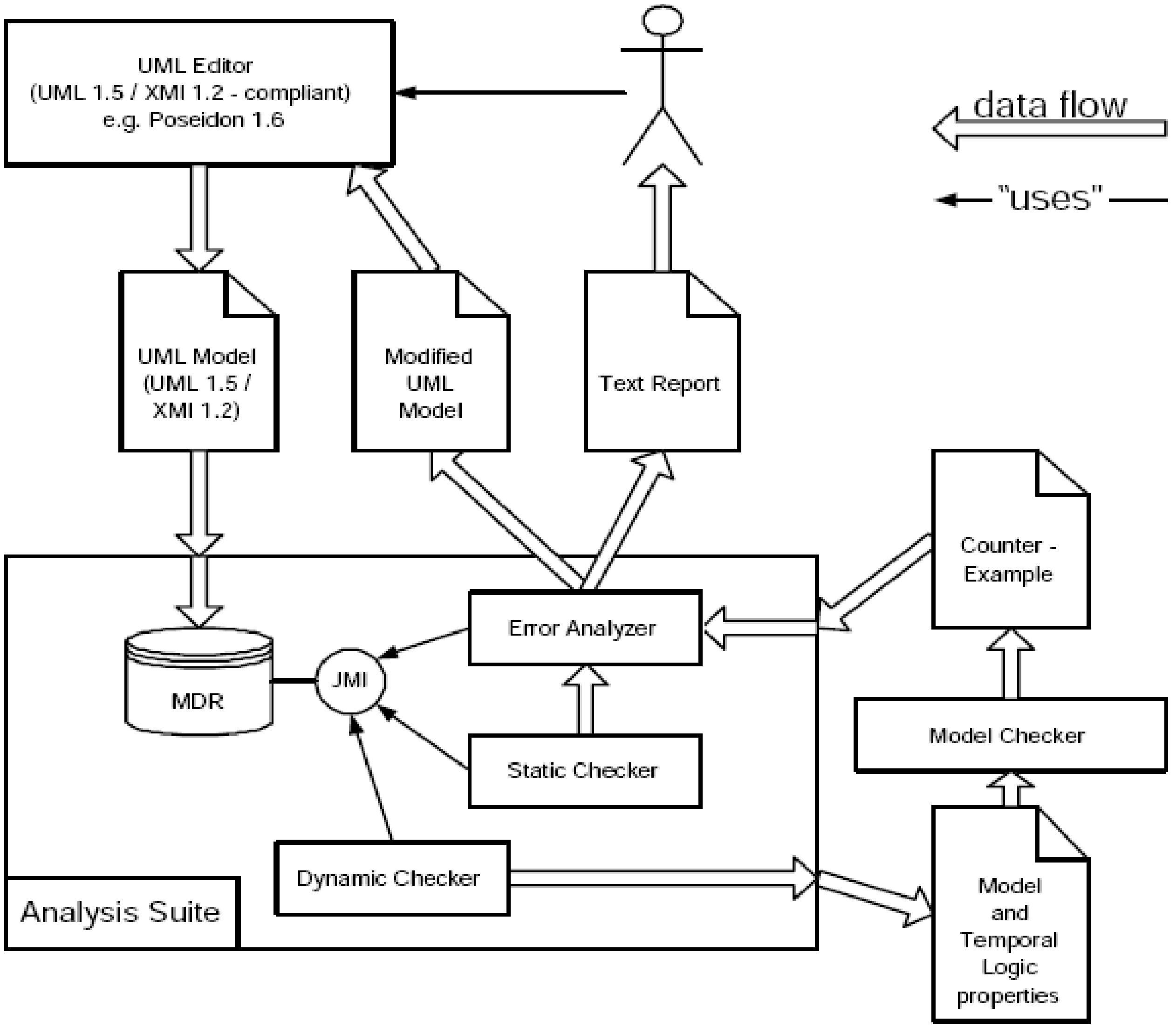
UMLsec

applies to:

- Sequence diagrams
  - Used to specify interaction between different parts of a system
  - Using UMLsec stereotypes, can be extended with information giving the security requirements relevant to that interaction
  - UMLsec defines a notation that can be used to model cryptographic data
- Statechart diagrams
  - Show the changes in state throughout an object's life
  - Can be used to specify security requirements on the resulting sequences of state and the interaction with the object's environment

PDF created with pdfFactory Pro trial version www.pdffactory.com

UMLsec: tool support



PDF created with pdfFactory Pro trial version www.pdffactory.com

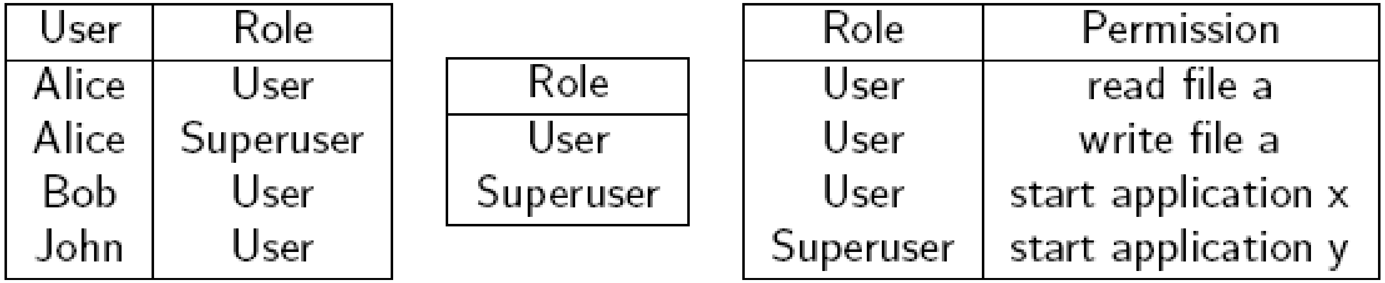
Secure UML

- UML extension for modelling language for the model-driven development of secure, distributed systems
- approach: Role-Based Access Control (RBAC) with additional support for specifying authorization constraints

PDF created with pdfFactory Pro trial version www.pdffactory.com

Secure UML

- Role-Based Access Control decouples users and permissions by roles representing jobs or functions



User	Role
Alice	User
Alice	Superuser
Bob	User
John	User

Role
User
Superuser

Role	Permission
User	read file a
User	write file a
User	start application x
Superuser	start application y

- extensions:
  - Hierarchies on roles, users and permissions
  - Authorization Constraints: formulae used to make stateful acces control decisions
    - Example. role customer may withdraw money when he is the owner and the amount is less than 1,000 $.

PDF created with pdfFactory Pro trial version www.pdffactory.com

Secure UML

Syntax

- abstract syntax defined by a MOF metamodel
- concrete syntax based on UML and defined with a UML profile
- syntax and semantics based on an extension of RBAC
- the key idea:
  - Access Control formalizes the permissions to perform actions on (protected) resources
  - we leave these open as types whose elements are not fixed
  - elements specified during combination with design language (via subtyping from existing types)

PDF created with pdfFactory Pro trial version www.pdffactory.com

Secure UML - tool

- SecureUML Visio for Microsoft Office Visio Professional 2003
  - the SecureUML Visio template defines a custom UML dialect to help system architects build roles based access control systems (RBAC)
  - http://www.foundstone.com

PDF created with pdfFactory Pro trial version www.pdffactory.com

WS- family*

- BPEL4WS
  - Business Process Execution Language for Web Services version 1.1 http://www-128.ibm.com/developerworks/library/specification/ws-bpel/
- WS-CDL
  - WS Choreography Model Overview http://www.w3.org/TR/ws-chor-model/
  - Web Services Choreography Description Language Version 1.0 http://www.w3.org/TR/ws-cdl-10/
- Security related WS-*
  - Web Service protocol stack http://roadmap.cbdiforum.com/reports/protocols/summary.php

PDF created with pdfFactory Pro trial version www.pdffactory.com

BPEL4WS

- Business Process Execution Language for Web Services
- defines the behavior (**orchestration**) of a business process based on interactions between the process and its partners
- enables the composition of multiple synchronous and asynchronous Web services

- a BPEL script is an XML document that conforms to the BPEL schema
- BPEL scripts are interpreted at runtime by a BPEL engine that identifies keywords or activities and executes them as defined in the BPEL script

PDF created with pdfFactory Pro trial version www.pdffactory.com

BPEL4WS

a BPEL process can:

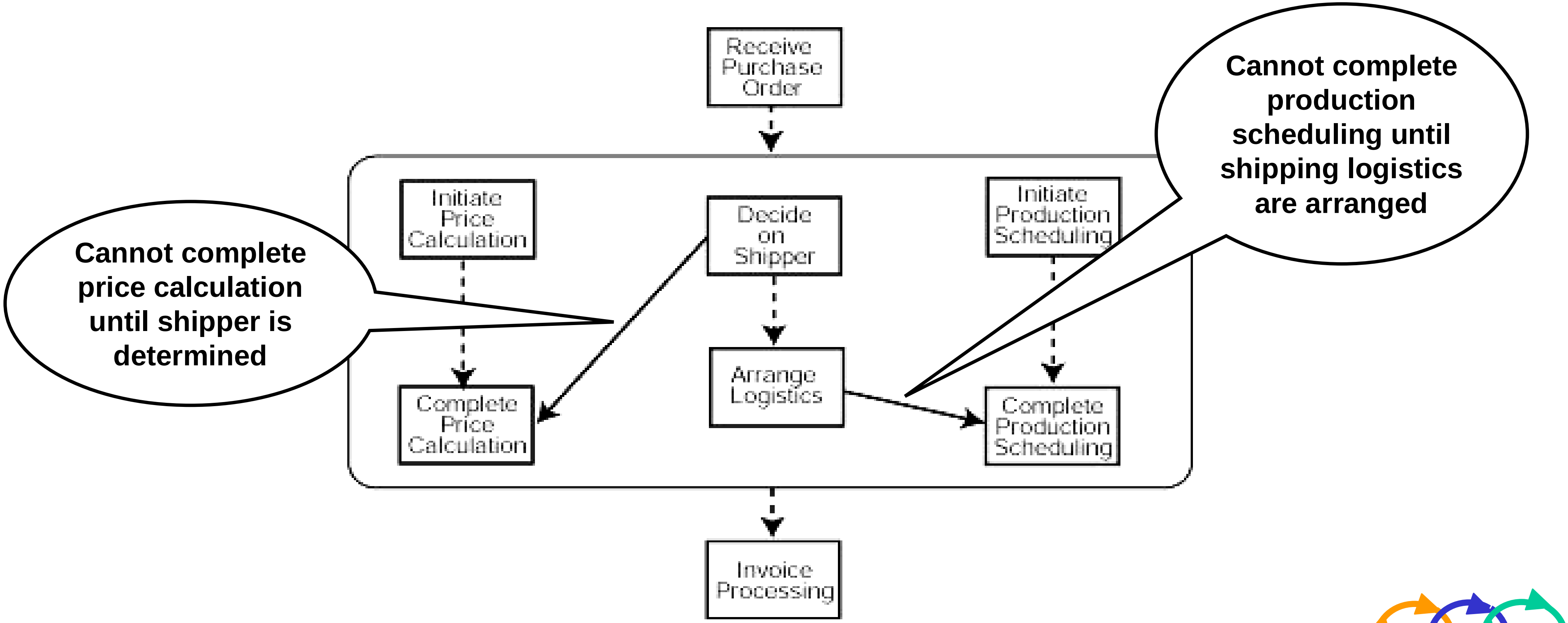
- communicate by exchanging messages with other web services using ***receive, reply,*** and ***invoke*** activities
- control execution flow using ***while, switch, sequence, pick, flow,*** and ***wait*** activities
- handle faults that can occur during processing using ***catch*** and ***catchall*** activities
- model event-driven programming using ***onMessage*** and ***onAlarm*** event handlers
- roll back transactions using compensation handlers

PDF created with pdfFactory Pro trial version www.pdffactory.com

BPEL4WS

dependencies:

- BPEL supports dependencies between activities in complex business processes
  - uses the structured activity **flow** and its property **link** to express the dependency


Receive Purchase Order
Initiate Price Calculation
Decide on Shipper
Initiate Production Scheduling
Arrange Logistics
Complete Price Calculation
Complete Production Scheduling
Invoice Processing

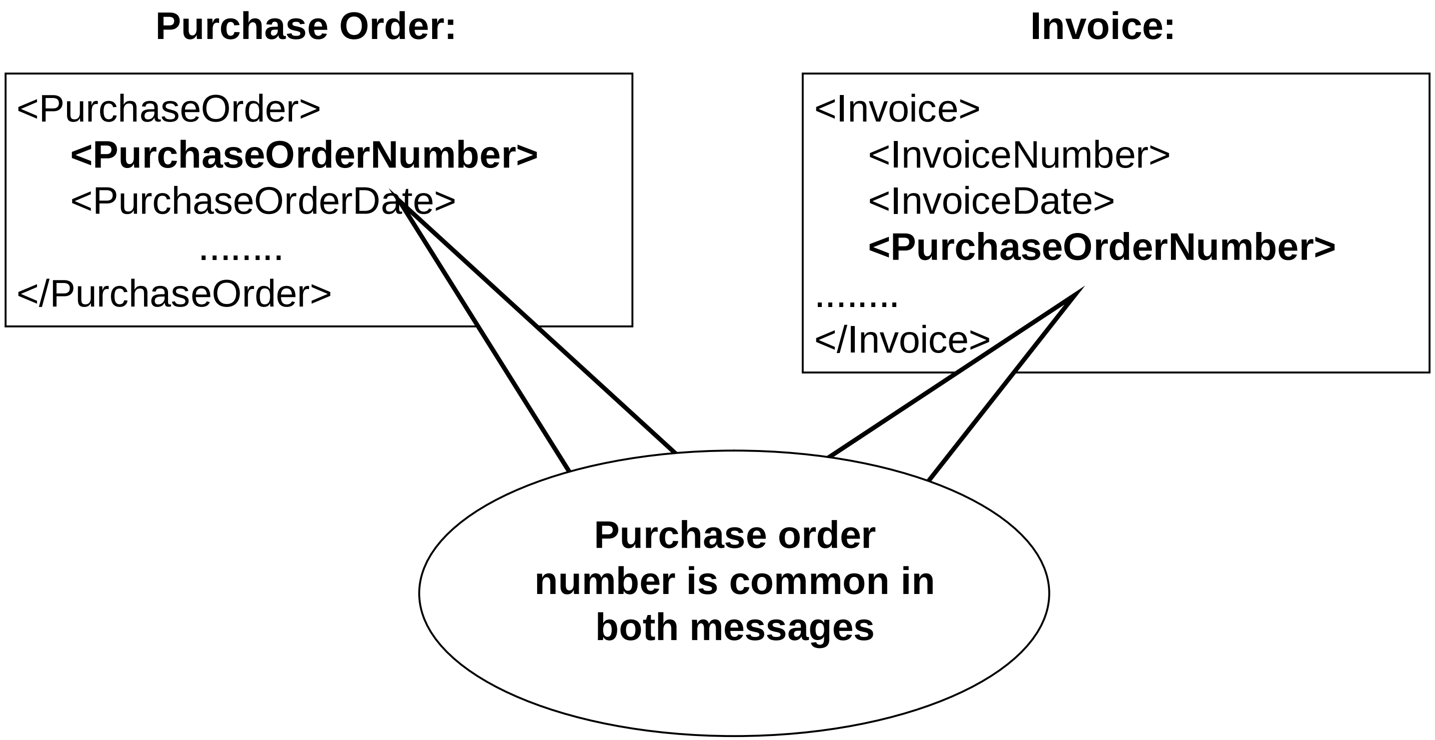
Cannot complete price calculation until shipper is determined

Cannot complete production scheduling until shipping logistics are arranged

PDF created with pdfFactory Pro trial version www.pdffactory.com

message correlation:

- involves the association of two or more messages with each other in an asynchronous environment
  - this works by **associating contents** in a given message with its correlating message



Purchase Order:

```
<PurchaseOrder>
    <PurchaseOrderNumber>
    <PurchaseOrderDate>
            ........
</PurchaseOrder>
```

Invoice:

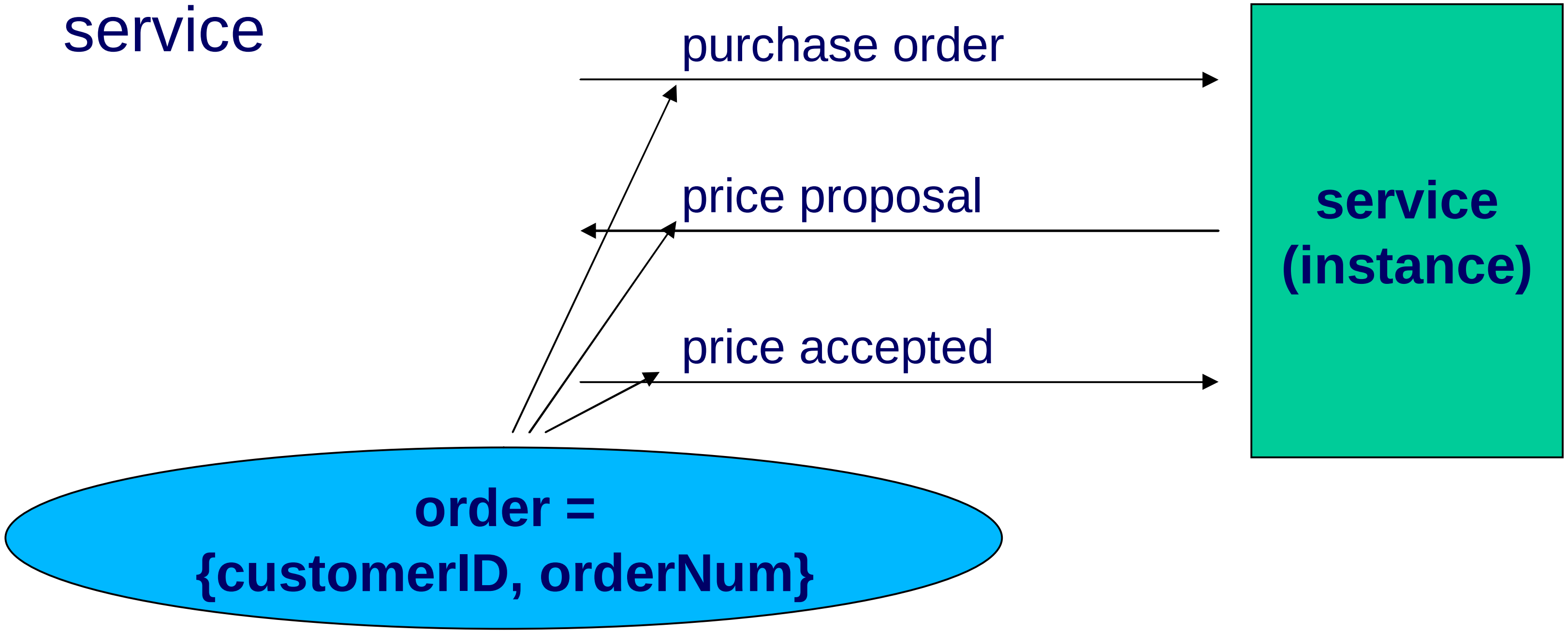
```
<Invoice>
    <InvoiceNumber>
    <InvoiceDate>
    <PurchaseOrderNumber>
........
</Invoice>
```

Purchase order number is common in both messages

PDF created with pdfFactory Pro trial version www.pdffactory.com

message correlation:

- involves the association of two or more messages with each other in an asynchronous environment
- BPEL uses **correlation sets** to explicitly define message correlation
  - correlation sets allow a message to be delivered not only to the correct port, but also to the correct *instance* of the service



PDF created with pdfFactory Pro trial version www.pdffactory.com

endpoint reference:

- use “endpoint references” for dynamic selection of service providers and invocation of their operations
- leverage the **WS-Addressing** specification for this capability
    - WS-Addressing defines a **standard representation for endpoint references** that incorporates information from a WSDL description as well as policy information:

```
<wsa:EndpointReference xmlns:wsa="...">
    <wsa:Address>http://www.someendpoint.com</wsa:Address>
    <wsa:PortType>PurchaseOrderPortType</wsa:PortType>
</wsa:EndpointReference>
```

PDF created with pdfFactory Pro trial version www.pdffactory.com

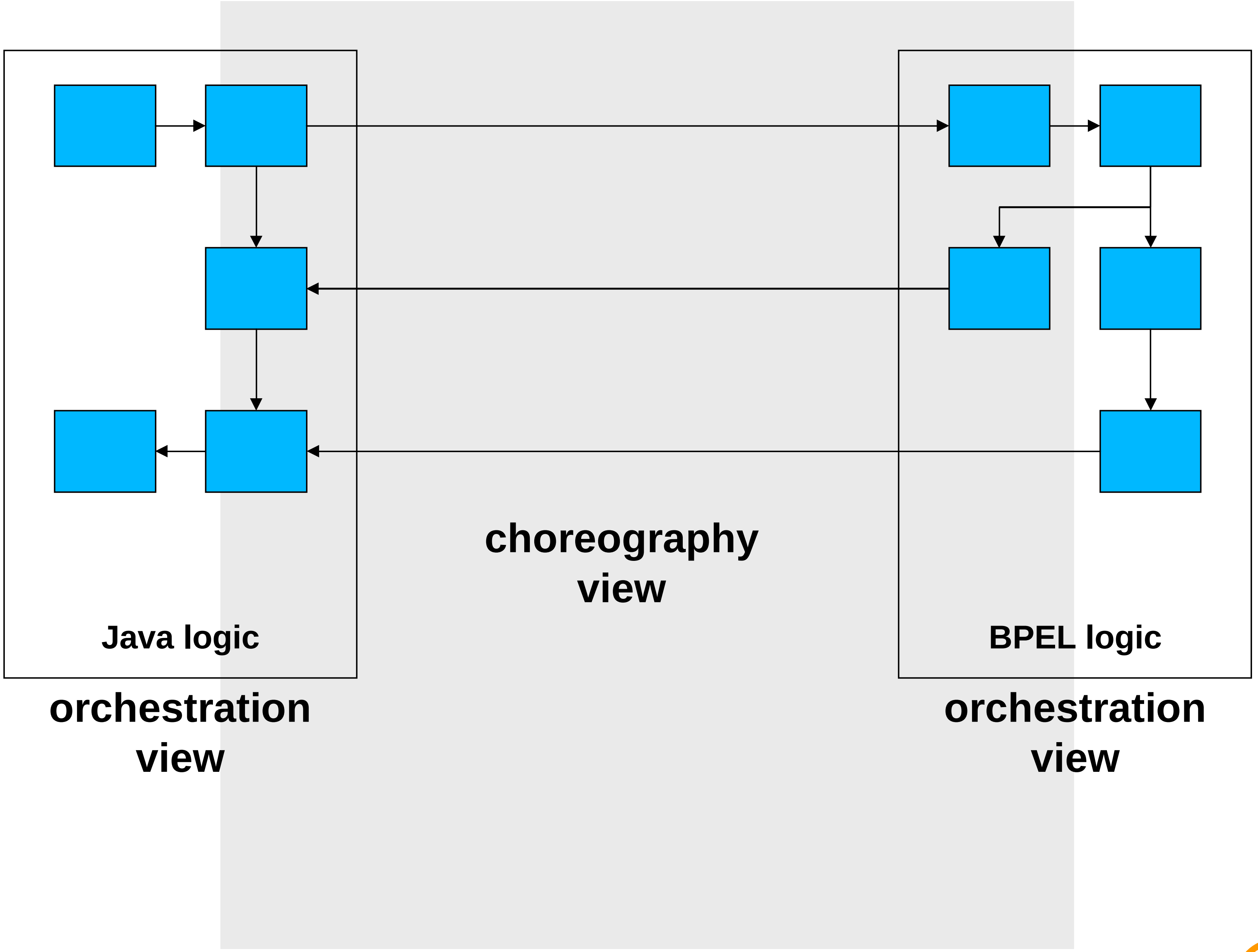
WS-CDL

W3C's Web Services Choreography Description Language

- XML-based language that describes peer-to-peer collaborations of participants by defining, from a global viewpoint, their common and complementary observable behavior; where ordered message exchanges result in accomplishing a common business goal
- targeted for composing interoperable, peer-to-peer collaborations between any type of participant regardless of the supporting platform or programming model used by the implementation of the hosting environment
- http://www.w3.org/TR/ws-cdl-10/

PDF created with pdfFactory Pro trial version www.pdffactory.com

CDL vs BPEL



choreography view

orchestration view

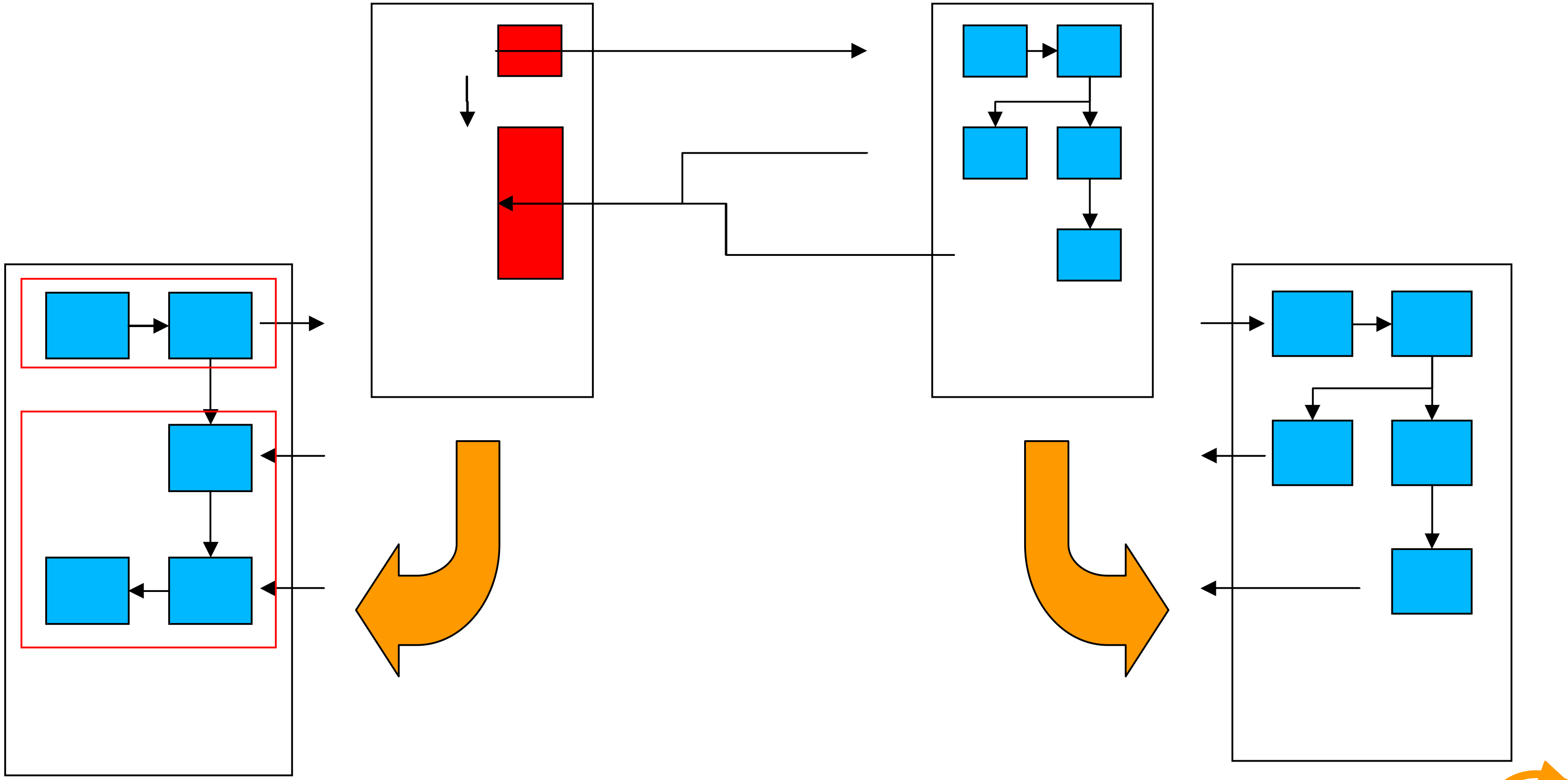
orchestration view

PDF created with pdfFactory Pro trial version www.pdffactory.com

CDL vs BPEL

From WS-CDL we can synthetise (part of?) the internal logic, that is the BPEL description

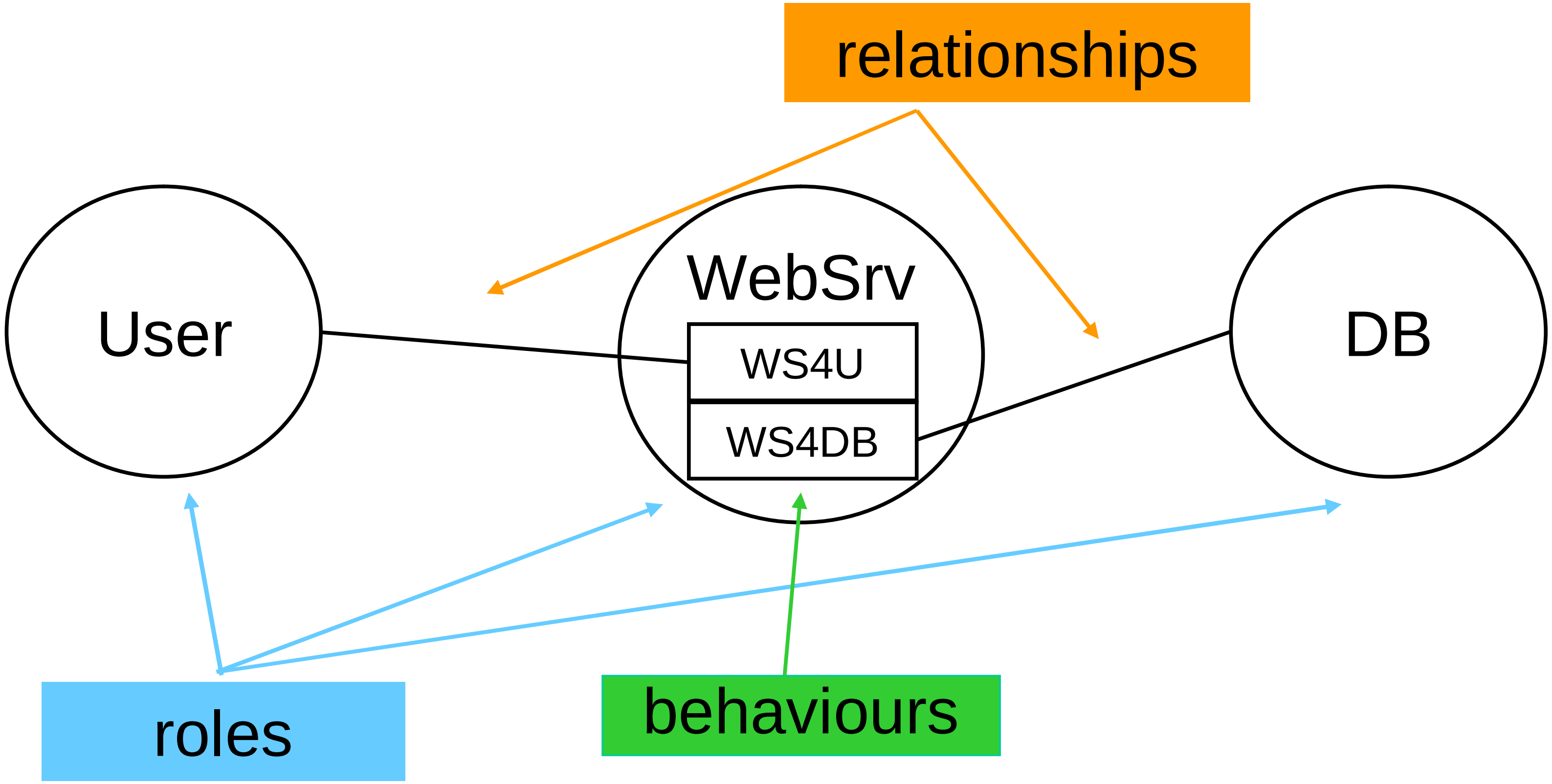
- some details can be hidden in the choreography view, and can be described in the orchestration view



PDF created with pdfFactory Pro trial version www.pdffactory.com

WS-CDL

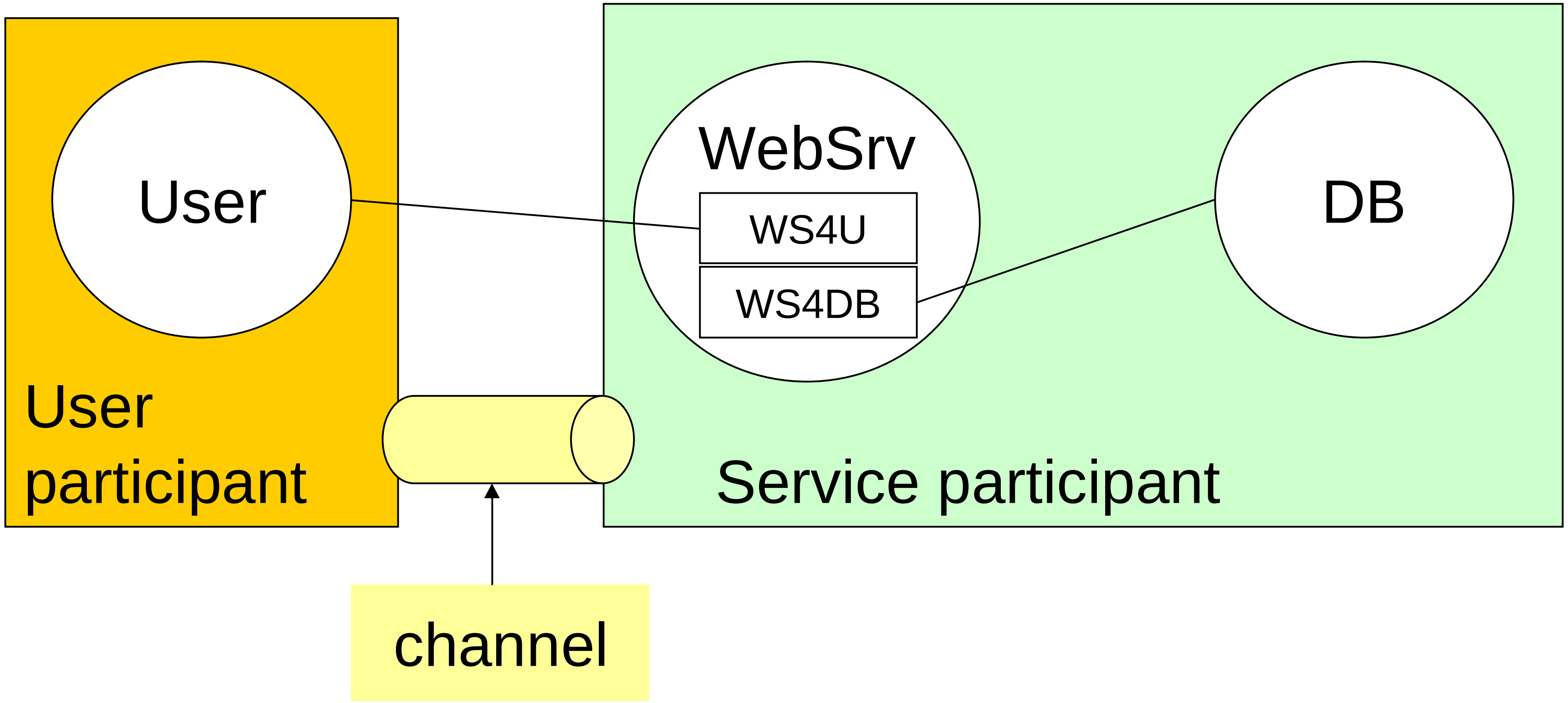
- Roles, Behaviours, Relationships
  - define the logical view of the system
  - relationship between **two** roles/behaviours [limit?]



PDF created with pdfFactory Pro trial version www.pdffactory.com

Participants, Channels

- group some roles, define the way in which participants talk
- channel (instantiated as variables) can be passed through interactions



PDF created with pdfFactory Pro trial version www.pdffactory.com

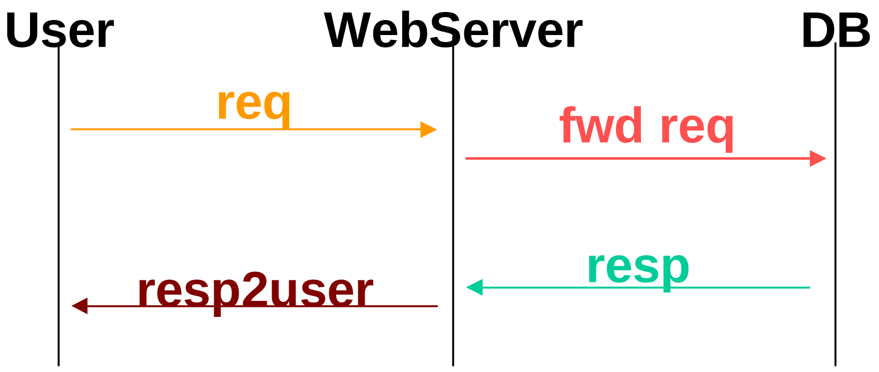
WS-CDL

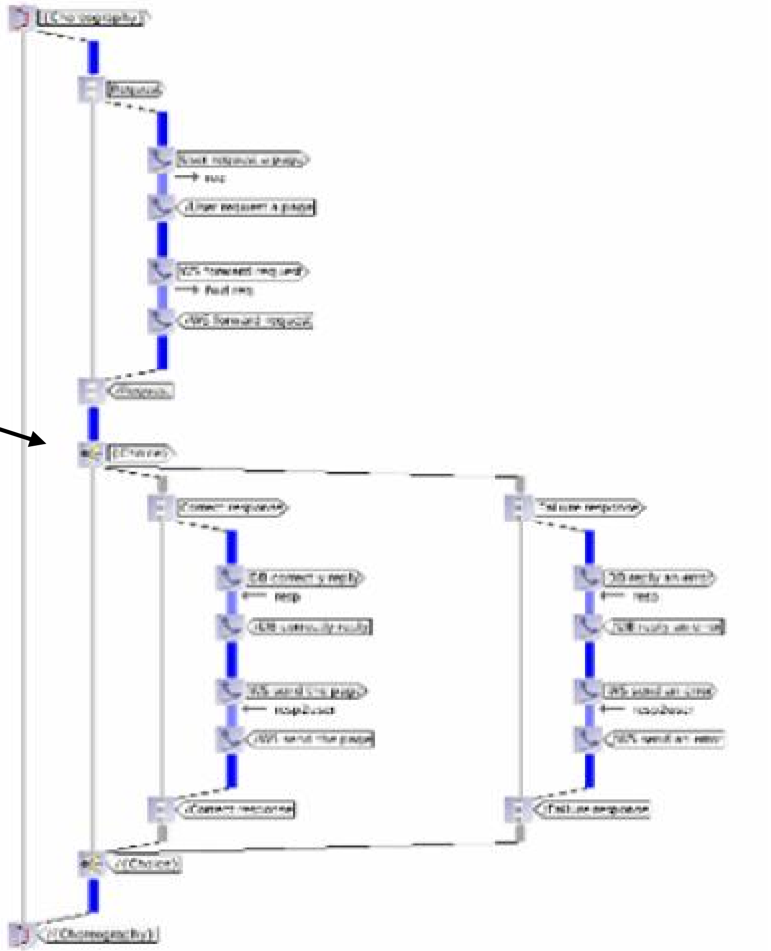
- n Types (Information Types, Channel Types)
  - 4 define the messages/channel structure
- n Tokens, Token locators
  - 4 pieces of variables (locate somewhere), relevant for the system functions

PDF created with pdfFactory Pro trial version www.pdffactory.com

WS-CDL

- Choreographies, Interactions
  - describe a (multi-participant) protocol
  - describe an internal “behaviour”?
  - allow control structures

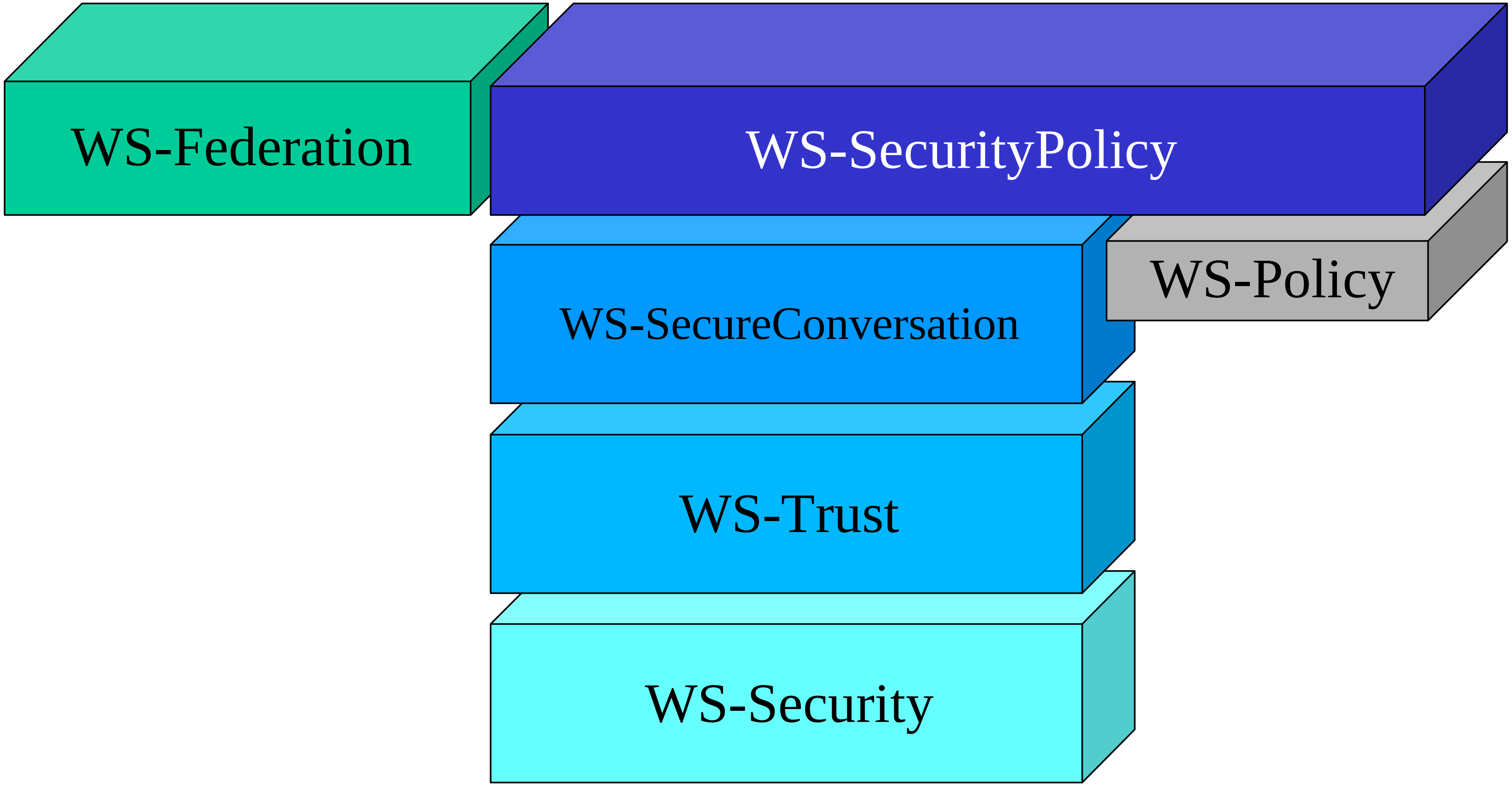




- Variables
  - information exchange (forwarding), states, channels, exceptions

PDF created with pdfFactory Pro trial version www.pdffactory.com

Security related WS-*



PDF created with pdfFactory Pro trial version www.pdffactory.com

Security related WS-*

- WS-Federation
  - This specification defines mechanisms to allow different security realms to federate by allowing and brokering trust of identities, attributes, authentication between participating Web services
- WS-Security
  - Describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. See WS Security Services
- WS-SecureConversation
  - Defines extensions that build on WS-Security to provide secure communication. Specifically, it defines mechanisms for establishing and sharing security contexts, and deriving session keys from security contexts
- WS-SecurityPolicy
  - An addendum to WS-Security. Indicates the policy assertions for WS-Policy which apply to WS-Security
  - **WS-Policy:** provides a general-purpose model and corresponding syntax to describe and communicate the policies of a Web service
- WS-Trust
  - Defines extensions that build on WS-Security to request and issue security tokens and to manage trust relationships

PDF created with pdfFactory Pro trial version www.pdffactory.com

System modelling: P-SDL

Marco Domenico Aime

POLITECNICO DI TORINO

26th September 2006
Wroclaw

DESEREC

Dependability and Security by Enhanced Reconfigurability

Information Society
Technologies

PDF created with pdfFactory Pro trial version www.pdffactory.com

P-SDL – POSITIF System Description Language

- new language, developed by the POSITIF FP6 project
- provides a grammar for the formal description of a network environment
- goals:
  - define a common language for the tools in the POSITIF Framework
  - detailed description of an ICT system, both from the physical and logical point of view
  - aimed to perform various security analysis

PDF created with pdfFactory Pro trial version www.pdffactory.com

P-SDL - cont.

- features:
  - based on XML
  - user-friendly (although verbose)
  - “guides” the user, setting strict limitations on the details available for each element
    - example: network ports on an hub can't have an IP address assigned.

```
<hub id=”hub1” ifaces=”4”>
   <interface id=”if01” protocol=”10-100BaseT” />
   <interface id=”if02” protocol=”10-100BaseT” />
   <interface id=”if03” protocol=”10-100BaseT” />

   <interface id=”if04” protocol=”10-100BaseT”>
      <addr type=”ipv4” netmask=”1.2.3.4”>5.6.7.8</addr>
   </interface>
```

ERROR!

```
</hub>
```

PDF created with pdfFactory Pro trial version www.pdffactory.com

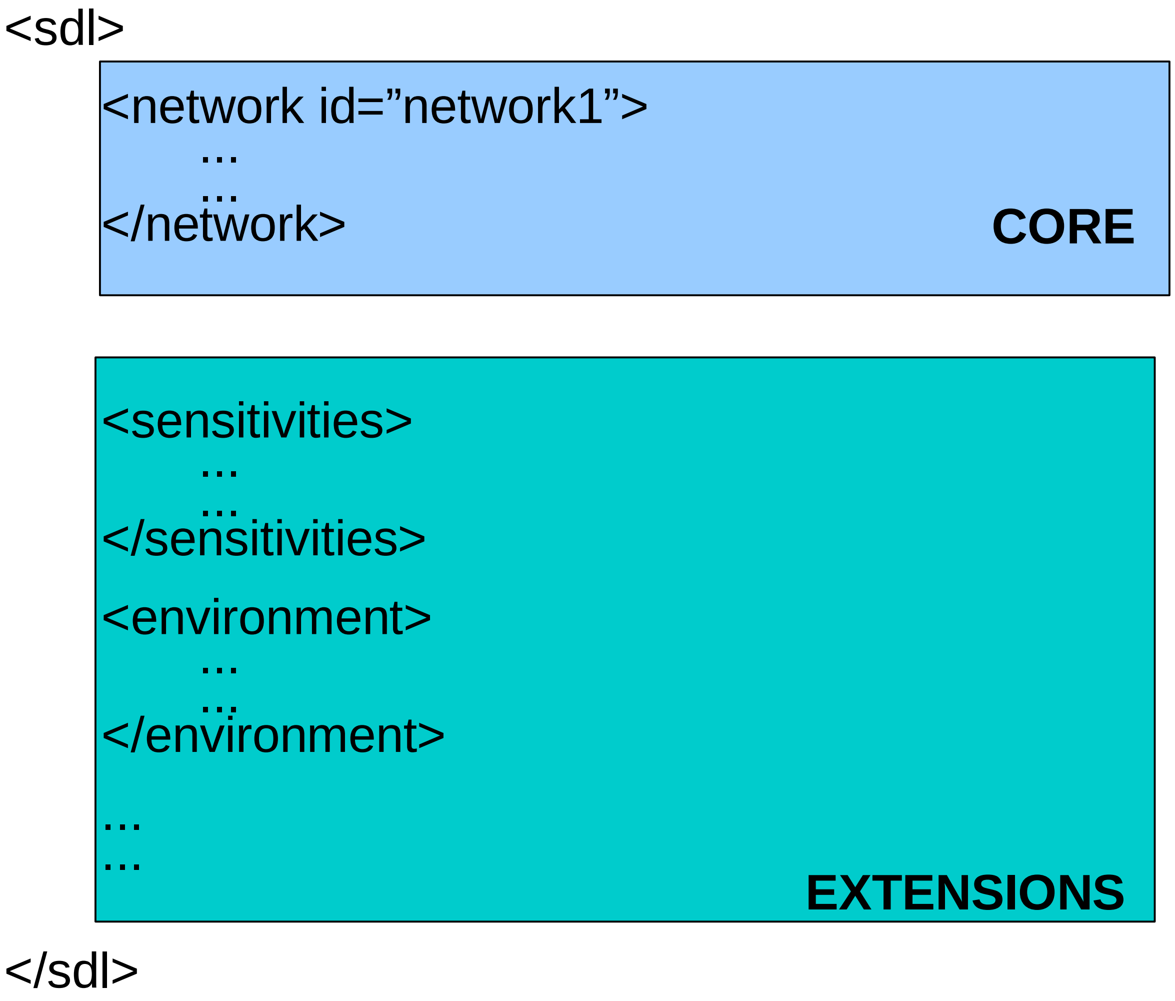
P-SDL parts

two main different parts are currently defined in the P-SDL language

- P-SDL Core
  - the main (mandatory) part, providing description for the network and its components
- P-SDL Extensions
  - additional (optional) part, giving details not strictly related to the network environment
  - depends from the elements defined in the Core

PDF created with pdfFactory Pro trial version www.pdffactory.com

Example – Basic P-SDL structure



```
<sdl>
    <network id=”network1”>
          ...
          ...
    </network>
    <sensitivities>
          ...
          ...
    </sensitivities>
    <environment>
          ...
          ...
    </environment>
    ...
    ...
</sdl>
```

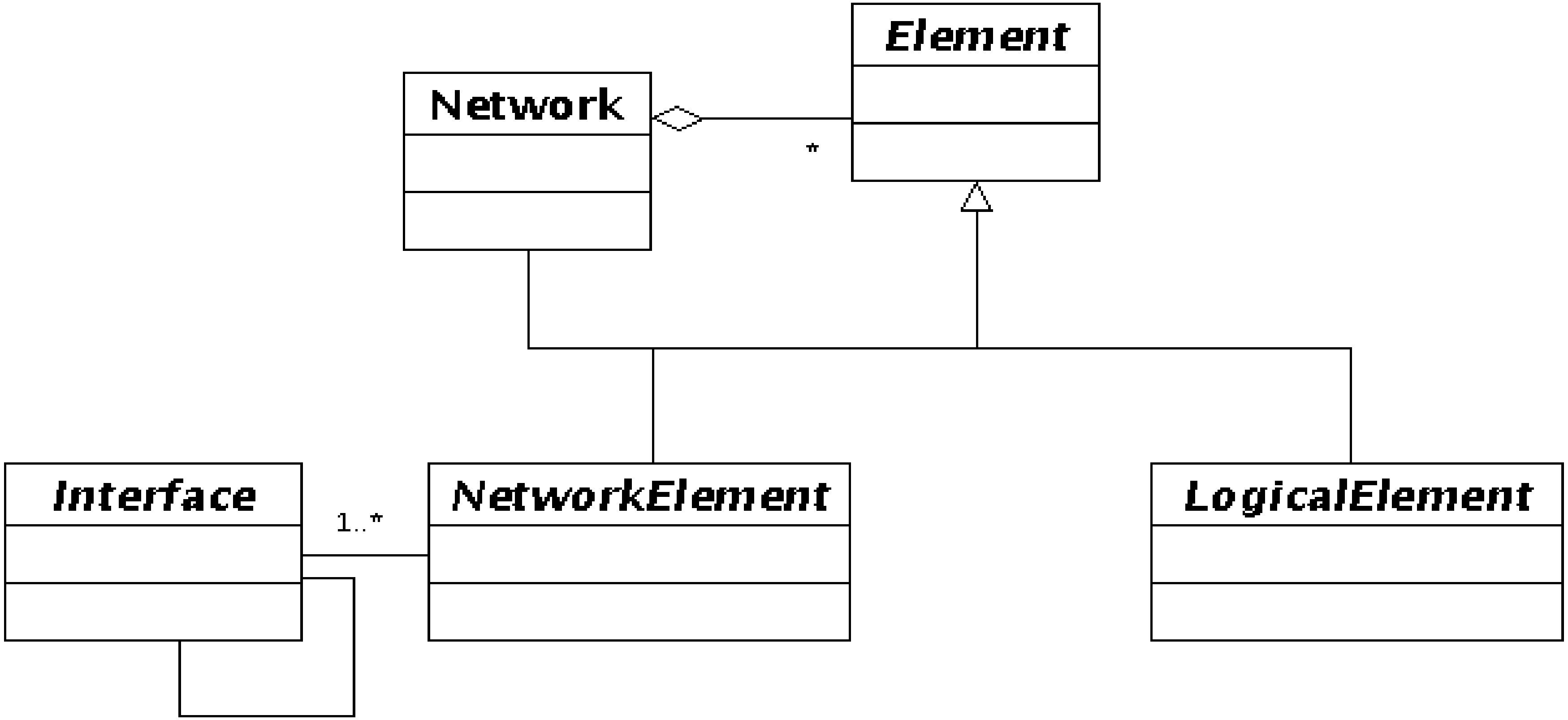
PDF created with pdfFactory Pro trial version www.pdffactory.com

P-SDL Core

- aimed to the description of the various network nodes
- two main subclasses are defined
  - network elements
    - physical nodes (i.e. Computer, Switch, Printer...)
    - node attributes (i.e. Id, number of network interfaces)
    - network interfaces are indicated inside the nodes
  - logical elements
    - services running on a node, and their service access point (SAP)
    - elements like operating system, running software or kernel, and their details (version, maintainer...)
    - security capabilities, like supported security protocol or packet filtering
- the network structure is defined by the connections between network interfaces belonging to different nodes

PDF created with pdfFactory Pro trial version www.pdffactory.com

P-SDL Core – UML Model

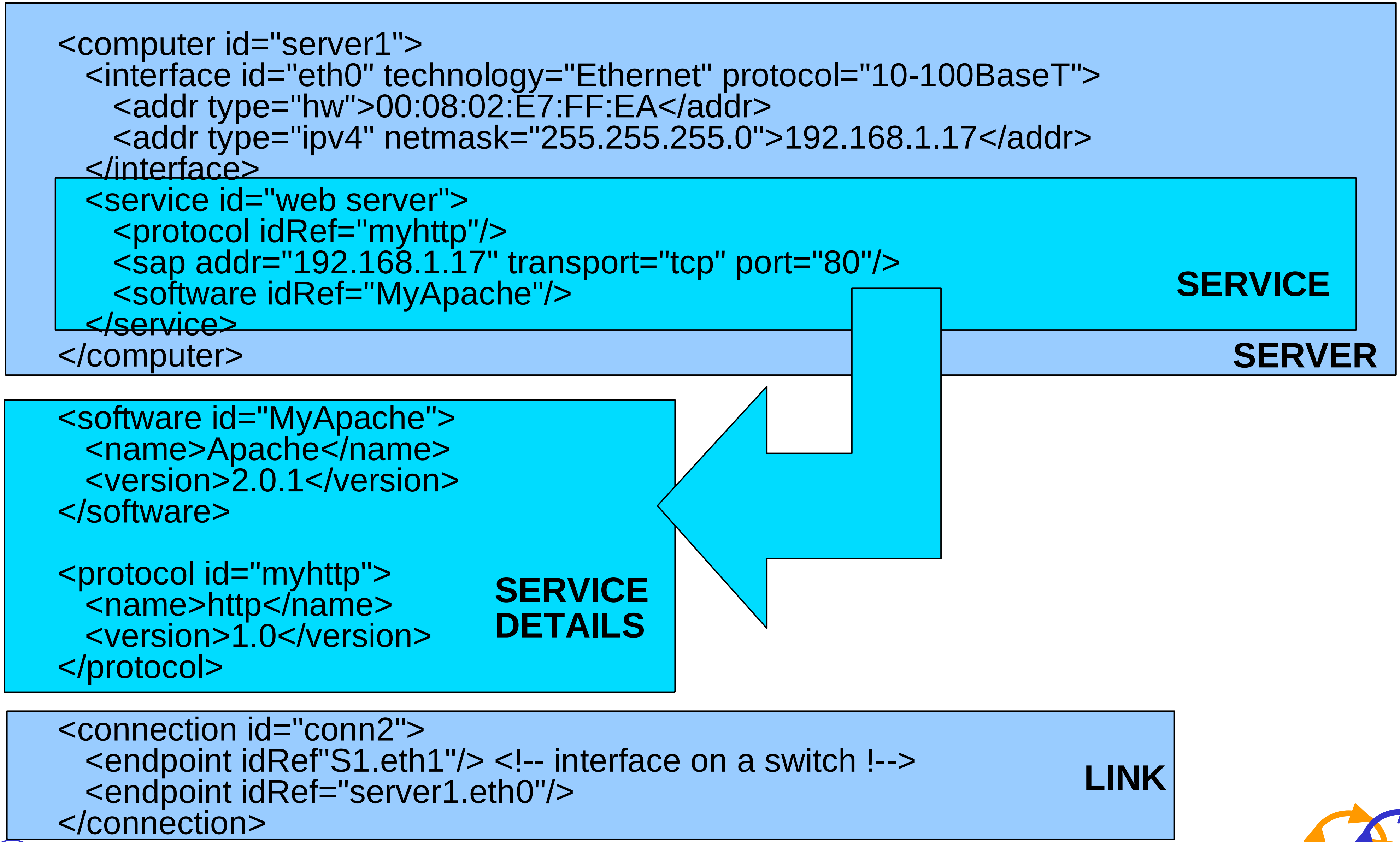


- a Network aggregates both Network and Logical elements
- NetworkElement and LogicalElement **are the main abstract classes for nodes and features of a network**
- each NetworkElement has at least one Interface
- connections are represented as links between different interfaces

PDF created with pdfFactory Pro trial version www.pdffactory.com

P-SDL Core - Example

server running Apache:



```
<computer id="server1">
  <interface id="eth0" technology="Ethernet" protocol="10-100BaseT">
    <addr type="hw">00:08:02:E7:FF:EA</addr>
    <addr type="ipv4" netmask="255.255.255.0">192.168.1.17</addr>
  </interface>
  <service id="web server">
    <protocol idRef="myhttp"/>
    <sap addr="192.168.1.17" transport="tcp" port="80"/>
    <software idRef="MyApache"/>
  </service>
</computer>
```

```
<software id="MyApache">
  <name>Apache</name>
  <version>2.0.1</version>
</software>

<protocol id="myhttp">
  <name>http</name>
  <version>1.0</version>
</protocol>
```

```
<connection id="conn2">
  <endpoint idRef"S1.eth1"/> <!-- interface on a switch !-->
  <endpoint idRef="server1.eth0"/>
</connection>
```

PDF created with pdfFactory Pro trial version www.pdffactory.com

P-SDL Extensions

- sometimes, just describing the network and its nodes isn't enough
  - security analysis
  - design
  - simulation
- P-SDL Extensions permit to express informations related to a given network environment
  - add details to the “core” description
- the P-SDL Extensions currently defined are
  - Environment
  - Sensitivities

PDF created with pdfFactory Pro trial version www.pdffactory.com

P-SDL Extensions - Environment

- describes physical characteristics related to the network description
  - locations : the physical areas where a node is located
    - Area, Building, Floor, Room, Locker (nested)
    - contain network elements
    - may have security features (alarm, access control, EM shielding)
  - power lines : basic description of the power supply system
    - nodes can be connected to different power supplies
    - locations may have more than one power supply
    - definition of UPS

PDF created with pdfFactory Pro trial version www.pdffactory.com

Example – locations and power supply

```
<environment>
    <power_line id="ENEL" default="true"/>
    <power_line id="generator"/>
    <building id="boella">
        <security_feature type="alarm" level="high"/>
        <room id="e-security">
            <security_feature type="access control" level="medium"/>
                <contained idRef="pc1"/>
                <contained idRef="pc2" />
                <contained idRef="server1">
                    <ps idRef="generator" />
                </contained>
            <locker id="rack1">
                <power_line id="UPS" ups="yes">
                    <source idRef="generator"/>
                </power_line>
                <contained idRef="switch1">
                    <ps idRef="UPS" />
            </contained>
            </locker>                                         LOCKER
        </room>                                                  ROOM
    </building>                                              BUILDING
</environment>
```

P-SDL Extensions - Sensitivity

- describe security level (LOW / MEDIUM / HIGH) for elements (physical / logical) in P-SDL description
  - confidentiality
  - integrity
  - availability
- the Sensitivity extension contains a list of references to elements sharing the same security levels

PDF created with pdfFactory Pro trial version www.pdffactory.com

Example

```
<sensitivities>
  <sensitivity>

     <feature type="availability" level="high"/>
     <feature type="confidentiality" level="high"/>

     <target idRef="SVNServer"/>

  </sensitivity>

  <sensitivity>

     <feature type="integrity" level="low"/>

     <target idRef="pc1"/>
     <target idRef="pc2"/>
     <target idRef="pc3"/>
     <target idRef="pc4"/>

  </sensitivity>
</sensitivities>
```

PDF created with pdfFactory Pro trial version www.pdffactory.com

CIMAPI library

- developed within the POSITIF project
- written in Java
- used to query P-SDL descriptions
    - translation from P-SDL to CIM (through XSLT)
        - standard
        - widely used and tested
    - interface between user/developer and data
    - can be used to obtain any information available in an P-SDL file

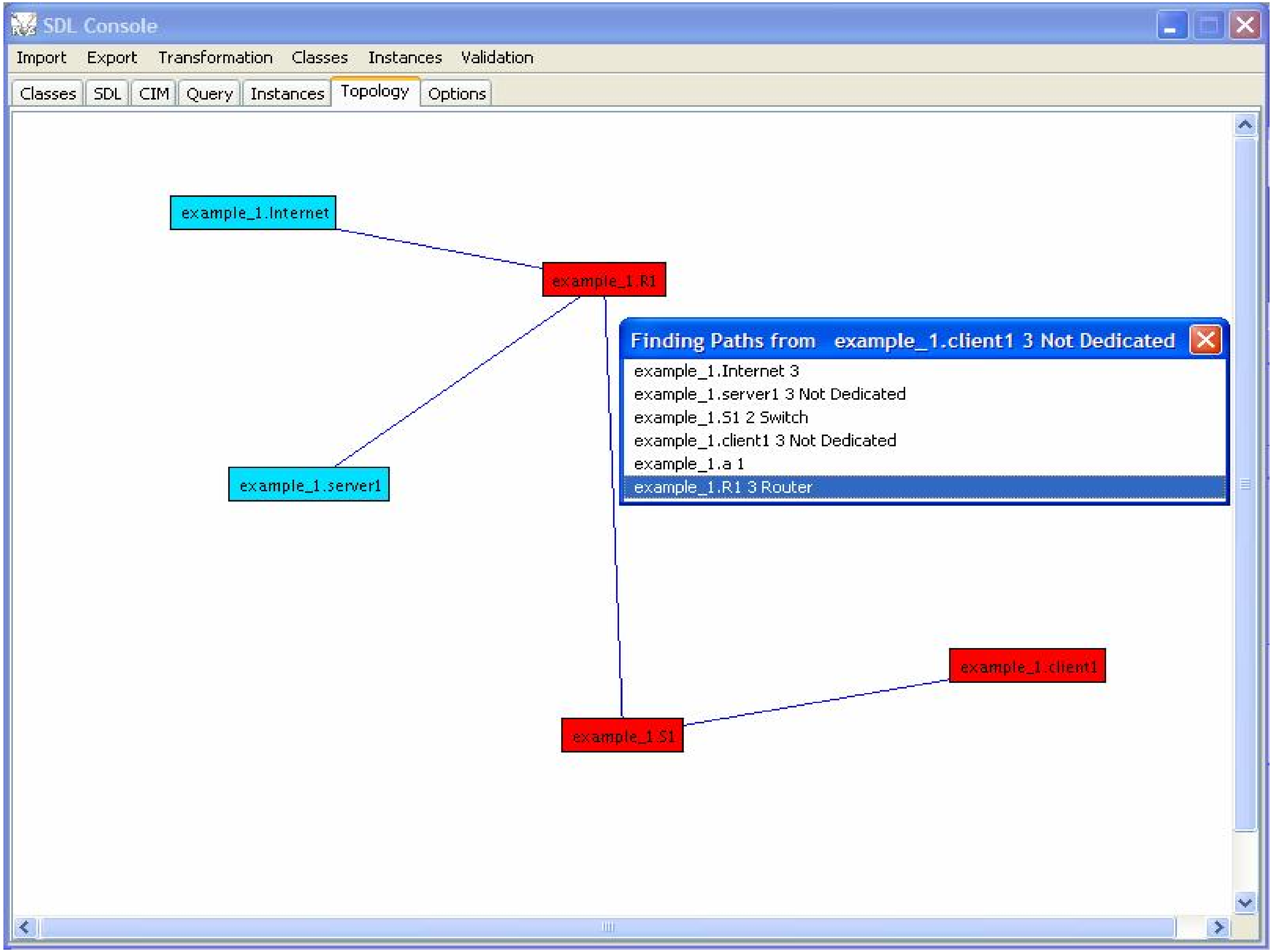
PDF created with pdfFactory Pro trial version www.pdffactory.com

CIMAPI library - cont.

- some tools using CIMAPI
  - P-SDL Console
    - validates P-SDL and generates CIM
    - graphical representation of the network and its details
  - Network Mapper
    - scans a real network and generates P-SDL description
  - Vulnerability Analyser
    - from an P-SDL description, discovers known vulnerabilities
  - Network Designer
    - graphical tool to design a network and obtain its P-SDL description

PDF created with pdfFactory Pro trial version www.pdffactory.com

Example – P-SDL Console


SDL Console
Import Export Transformation Classes Instances Validation
Classes SDL CIM Query Instances Topology Options
example_1.Internet
example_1.R1
example_1.server1
example_1.S1
example_1.client1

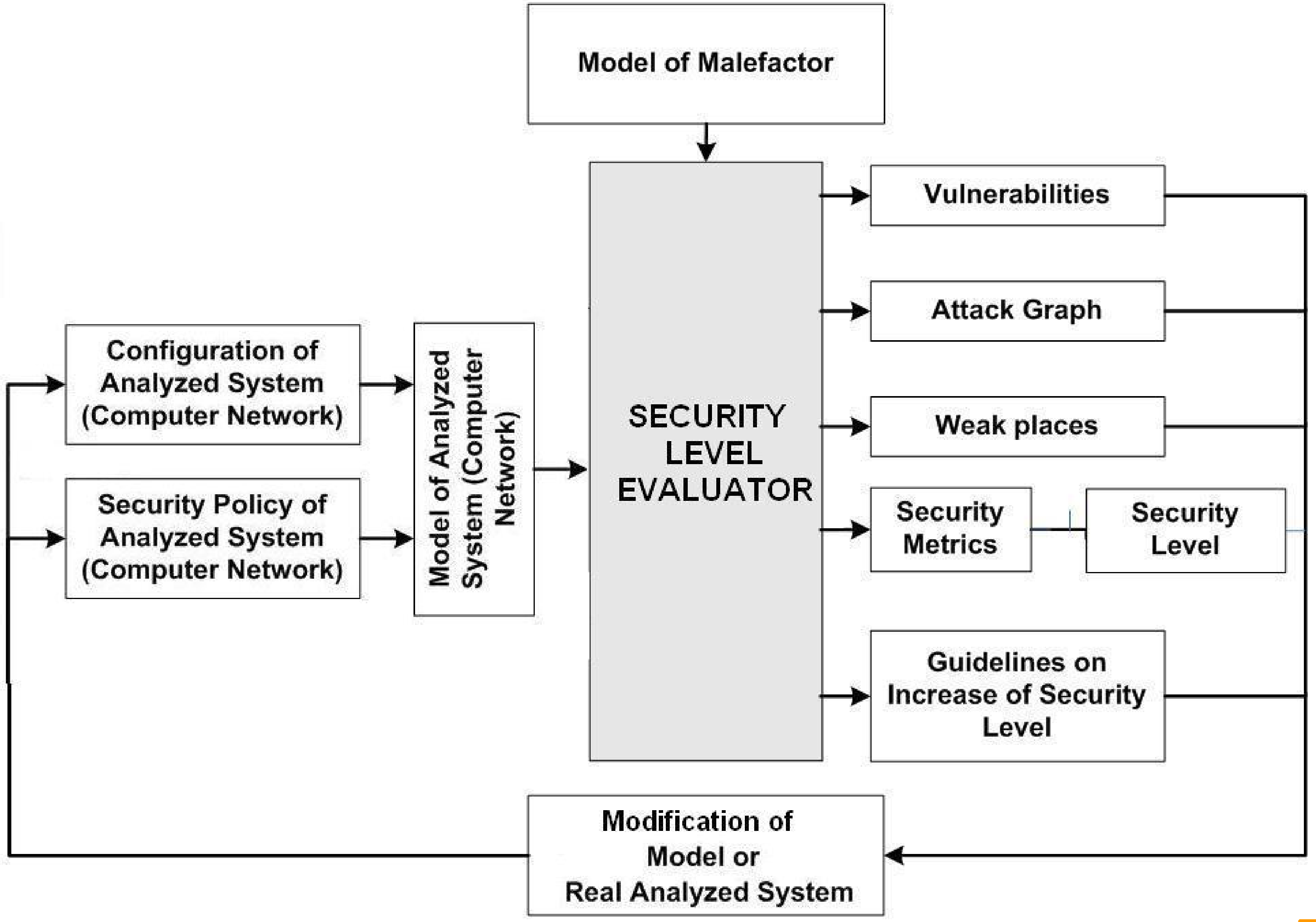
Finding Paths from example_1.client1 3 Not Dedicated

example_1.Internet 3
example_1.server1 3 Not Dedicated
example_1.S1 2 Switch
example_1.client1 3 Not Dedicated
example_1.a 1
example_1.R1 3 Router

PDF created with pdfFactory Pro trial version www.pdffactory.com

Security Level Evaluation

malefactor's action simulation approach



Model of Malefactor
Configuration of Analyzed System (Computer Network)
Security Policy of Analyzed System (Computer Network)
Model of Analyzed System (Computer Network)
SECURITY LEVEL EVALUATOR
Vulnerabilities
Attack Graph
Weak places
Security Metrics
Security Level
Guidelines on Increase of Security Level
Modification of Model or Real Analyzed System

PDF created with pdfFactory Pro trial version www.pdffactory.com

Main features of the Approach 1/2

- Based on malefactor’s action simulation and integrated family of various expert knowledge models
- Two phases:
  - (1) construction of attack graph and
  - (2) computation of different security metrics using combination of qualitative techniques of risk analysis
- Taking into account diversity of malefactor’s positions, intentions and experience;
- Estimating the influence of different configuration and policy data;
- Taking into account not only attack actions (which use vulnerabilities), but the common actions of legitimate users and reconnaissance actions;

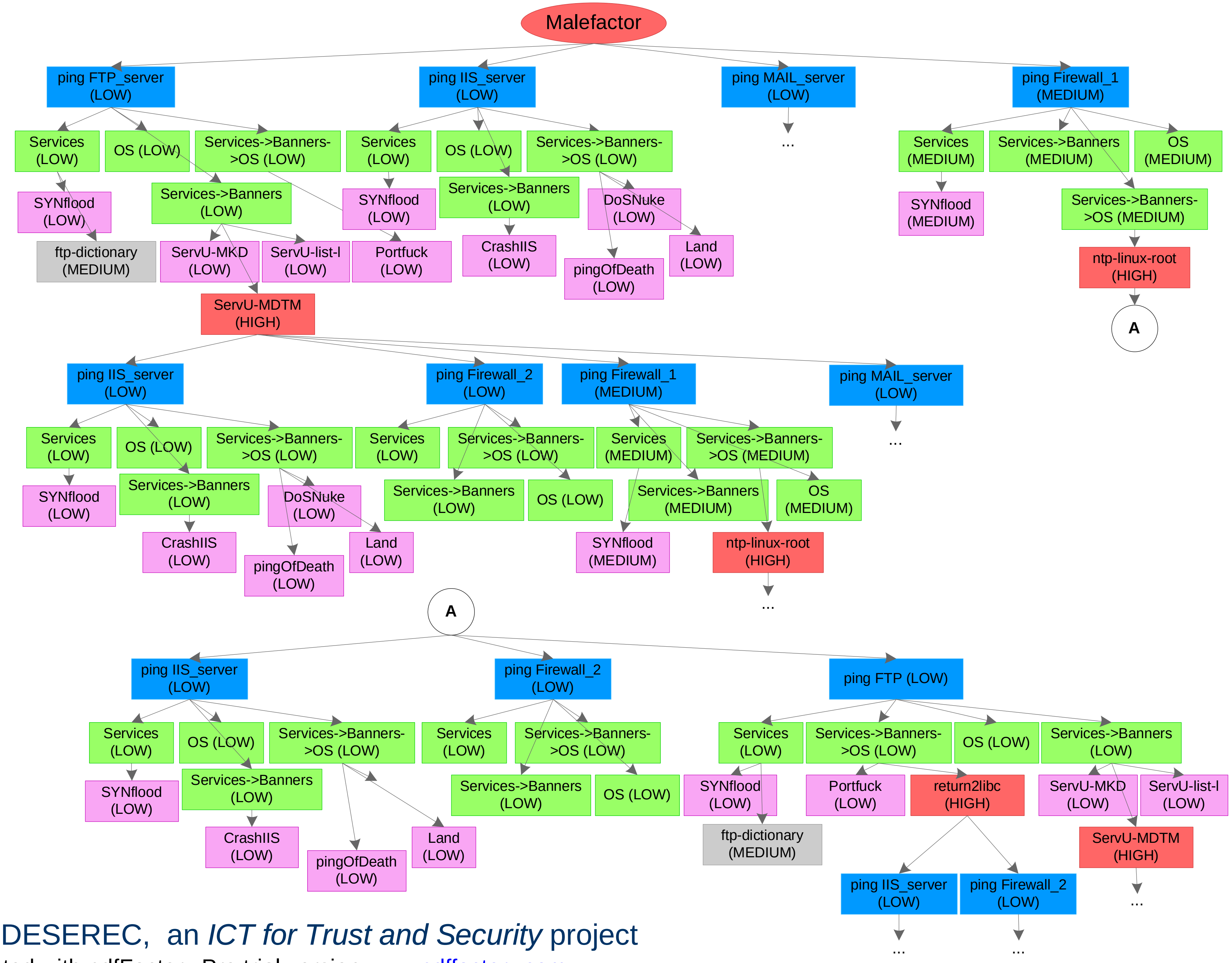
PDF created with pdfFactory Pro trial version www.pdffactory.com

Main features of the Approach 2/2

- Investigation of various threats for different resources;
- Detection of “weak” places;
- Usage of multiple databases of vulnerabilities (OSVDB,NVD, OVAL);
- CVSS (Common Vulnerability Scoring System) approach is used for computation of a part of primary security metrics;
- comparing calculated metrics and user requirements

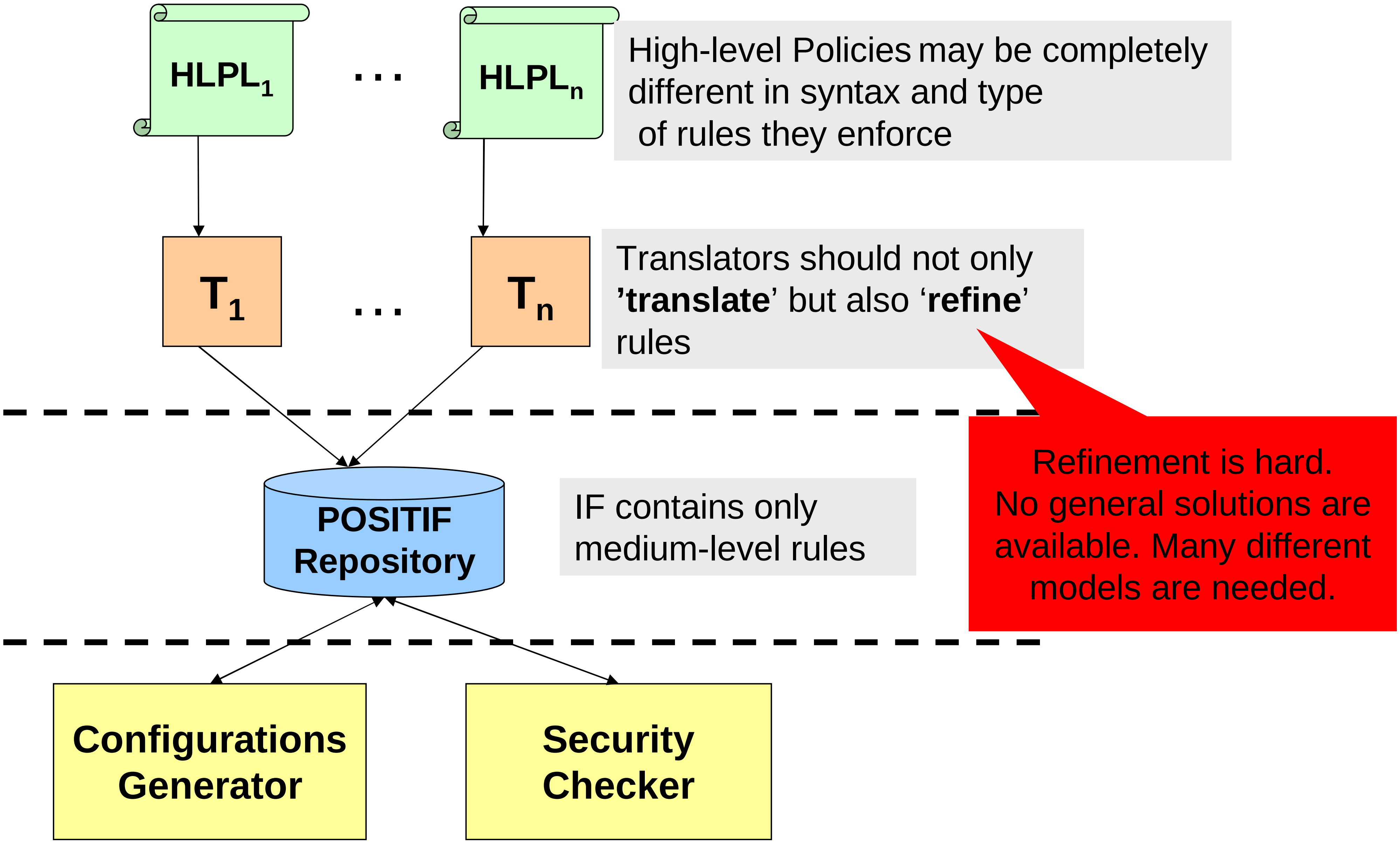
PDF created with pdfFactory Pro trial version www.pdffactory.com

Example: Attack Graph



PDF created with pdfFactory Pro trial version www.pdffactory.com

Policy refinement



$HLPL_1$ . . . $HLPL_n$

High-level Policies may be completely different in syntax and type of rules they enforce

T_1 . . . T_n

Translators should not only ’**translate**’ but also ‘**refine**’ rules

Refinement is hard. No general solutions are available. Many different models are needed.

POSITIF Repository

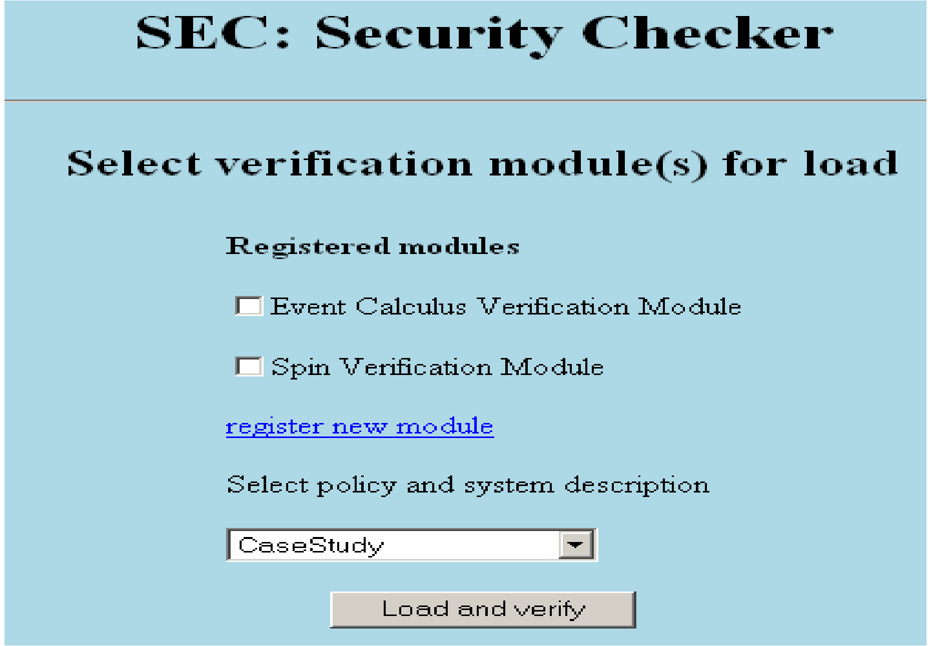
IF contains only medium-level rules

Configurations Generator

Security Checker

PDF created with pdfFactory Pro trial version www.pdffactory.com

Example: Security Checker



SEC: Security Checker

Select verification module(s) for load

Registered modules

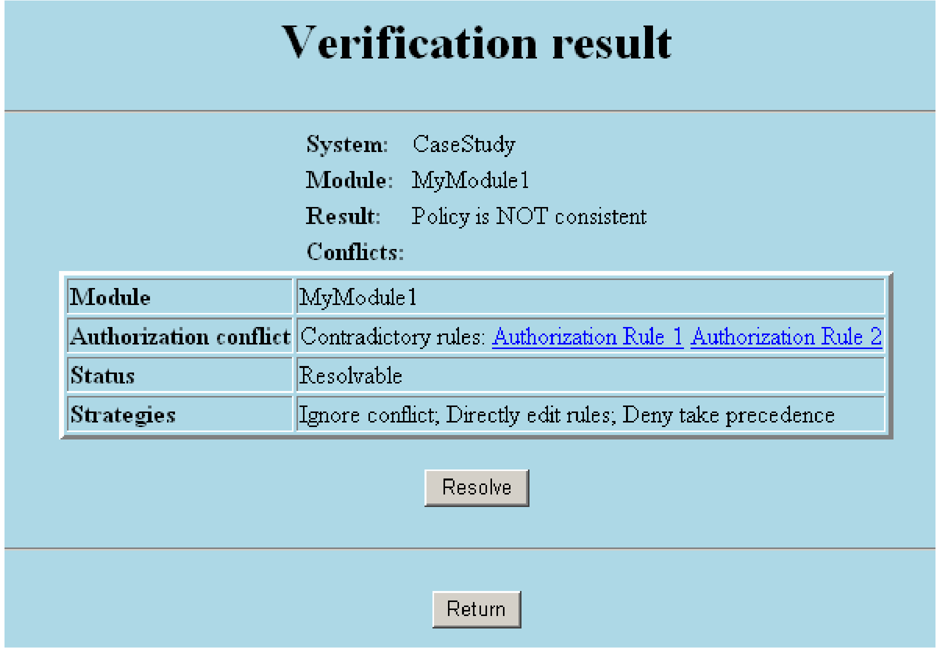
□ Event Calculus Verification Module

□ Spin Verification Module

register new module

Select policy and system description

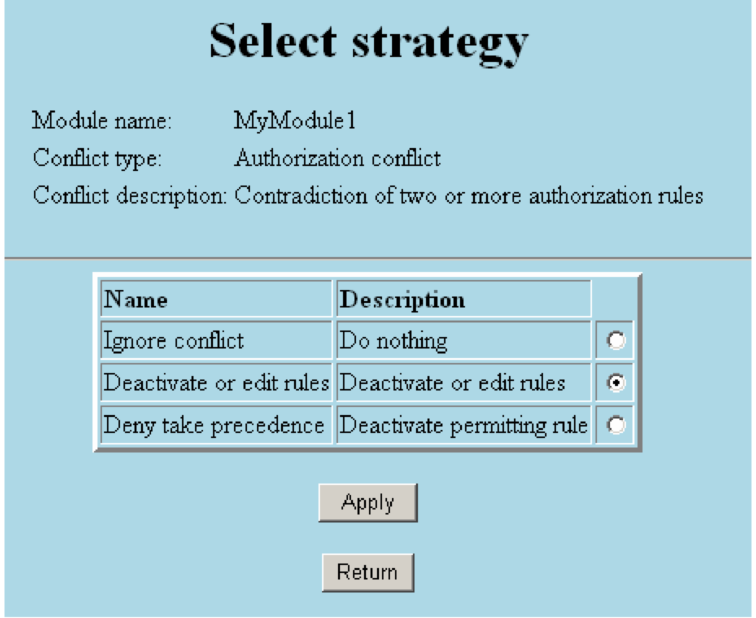
CaseStudy



Verification result

System: CaseStudy
Module: MyModule1
Result: Policy is NOT consistent
Conflicts:

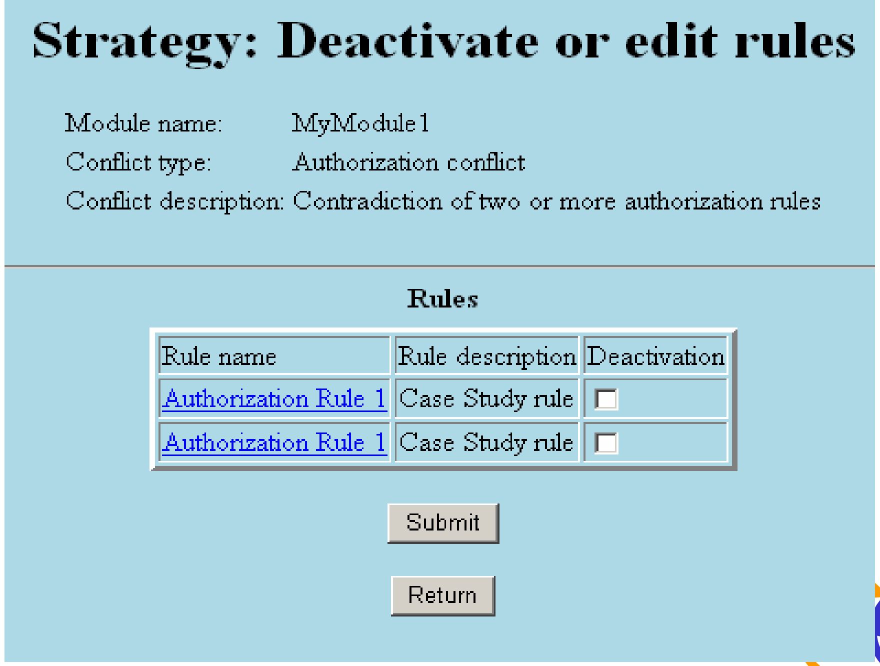
Module	MyModule1
Authorization conflict	Contradictory rules: Authorization Rule 1 Authorization Rule 2
Status	Resolvable
Strategies	Ignore conflict; Directly edit rules; Deny take precedence



Select strategy

Module name: MyModule1
Conflict type: Authorization conflict
Conflict description: Contradiction of two or more authorization rules

Name	**Description**	
Ignore conflict	Do nothing	○
Deactivate or edit rules	Deactivate or edit rules	●
Deny take precedence	Deactivate permitting rule	○



Strategy: Deactivate or edit rules

Module name: MyModule1
Conflict type: Authorization conflict
Conflict description: Contradiction of two or more authorization rules

Rules

Rule name	Rule description	Deactivation
Authorization Rule 1	Case Study rule	□
Authorization Rule 1	Case Study rule	□

PDF created with pdfFactory Pro trial version www.pdffactory.com

Security Checker

- no general techniques valid for verification and policy consistency checking
  - specific field solutions: formal model + formal techniques
- methods selected for POSITIF = many modules
  - Event Calculus => authorization and resource conflicts
  - model checking => authorization and resource conflicts
  - algebraic checking => static conflicts in rules of the same type (filtering, channel protection, etc.)

PDF created with pdfFactory Pro trial version www.pdffactory.com

Verification Modules

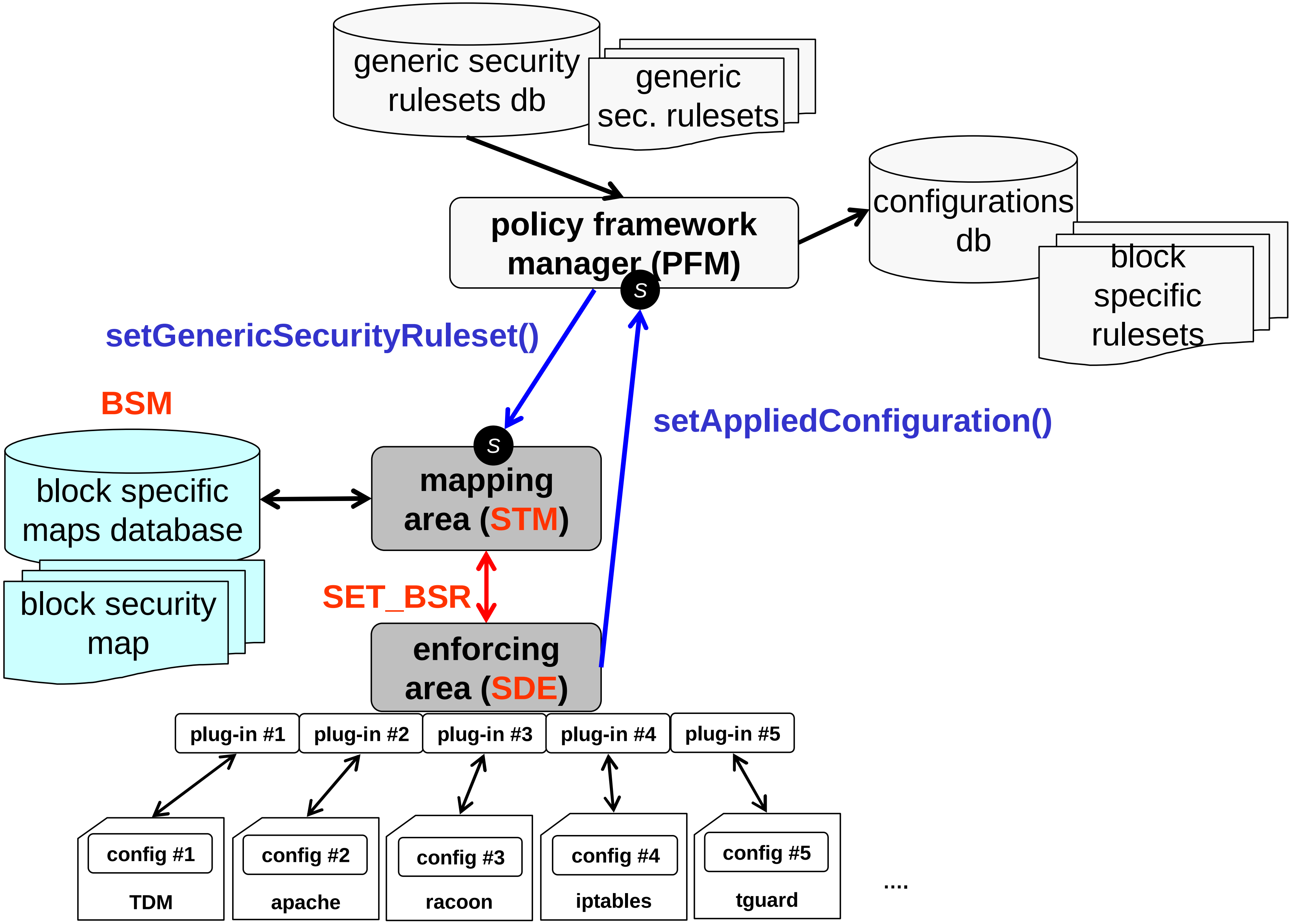
- n Event Calculus verification module
  - 4 security policies and system description are translated into domain-dependent Event Calculus axiomatic
  - 4 conflict predicates are introduced
  - 4 abductive inference is used for conflict detection
  - 4 *Implemented in Jess*

- n SPIN verification module
  - 4 security policies and system description are translated into Promela data structures, processes, and assertions
  - 4 policy conflicts introduced as additional assertions
  - 4 *impemented in SPIN-Promela*

- n algebraic approach
  - 4 static conflicts in rules of the same type (filtering, channel protection, etc.)
  - 4 implementation of conflict detection
  - 4 implementation of semilattice-based conflict resolution
  - 4 extension of Java-based graph library to define the semilattice

PDF created with pdfFactory Pro trial version www.pdffactory.com

Generation of device configurations



PDF created with pdfFactory Pro trial version www.pdffactory.com

System modelling: conclusions

Marco Domenico Aime

POLITECNICO DI TORINO

26th September 2006
Wroclaw

DESEREC

Dependability and Security by Enhanced Reconfigurability

Information Society
Technologies

PDF created with pdfFactory Pro trial version www.pdffactory.com

System modelling languages summary

- n languages focused on simulation are lacking
  - 4 software features, service view, security and dependability features
  - 4 but provides good models to represents network topology
- n SysML is a powerful representing language but
  - 4 is too flexible
  - 4 doesn’t help user in the description
- n UML security and QoS extensions
  - 4 provide some good models, but incomplete and too flexible
- n Web Service languages
  - 4 just good for service composition and workflow
- n P-SDL
  - 4 minimal
  - 4 has a policy language associated (P-SPL)
  - 4 service view extension under construction

PDF created with pdfFactory Pro trial version www.pdffactory.com

System modelling in DESEREC

if you want DESEREC to analyse your system, you must be ready to describe, at least:

- n system structure
  - 4 network topology
  - 4 interface between network and services
  - 4 service internal structure
  - 4 physical locations
- n requirements
  - 4 security, dependability, QoS requirements
  - 4 high-level configuration policies (service priorities, service allocation and orchestration strategies)

PDF created with pdfFactory Pro trial version www.pdffactory.com