SIXTH FRAMEWORK PROGRAMME
PRIORITY 2
"Information Society Technologies"

Project acronym: DESEREC
Project full title: Dependability and Security by Enhanced Reconfigurability
Proposal/Contract no.: IST-2004-026600-DESEREC

D5.5
Report and Proceedings of the Second Workshop

Project Document Number: DESEREC/D5.5/PU[1]/v1.1 (official version)
Project Document Date: 18/11/2008
Workpackage Contributing to the Project Document: WP5
Deliverable Type and Security: R[2]-PU
Author(s): UMU, IEIIT

Abstract:
This document provides a report on the 2nd DESEREC dissemination workshop.

Keywords: dissemination, dependability, workshop

[1] Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

[2] Type: P - Prototype, R - Report, D - Demonstrator, O - Other

History

Version	Date	Description, Author(s), Reviser(s)
1.1	18/11/2008	PMT approved version

Executive Summary

This document reports about the 2nd DESEREC dissemination workshop, held as part of conference SECURWARE "Second International Conference on Emerging Security Information, Systems and Technologies", 25-31 August, 2008, Cap Esterel, France. Both of them were sponsored by IARIA, which offers the opportunity for partnership in the form of conference sponsorship.

This report mainly focuses on the contributions to the workshop organization and sessions given by DESEREC. A more detailed description and a complete report on the global conference SECURWARE is available at: http://www.iaria.org/conferences2008/SECURWARE08.html

This document is organized as follows: an introductory section briefly depicts the goals, the achievements and the results of the workshop, and then the full detailed program is introduced. At the end of the document, Appendix A contains the material presented at the workshop by the DESEREC consortium and Appendix B collects all the slides presented during the expert panel session organized by DESEREC.

All conference proceedings have been published by IEEE Computer Society Press (CPS), which are available online at IEEE Digital Library.

Contents

Page

1 Introduction 5

2 Workshop organization 7
2.1 Preliminary program 7
2.2 Detailed program for the DEPEND workshop 10

3 Next events related to DEPEND 2008 12
3.1 Special issue of IJCCBS 12
3.2 DEPEND 2009 12

4 Conclusions 14

Appendix A: DEPEND 2008 workshop 15
Design-Time Learning for Operational Planning Improvement 16
Molecular Approach Paves the Way towards High Resilience for Large Mission-Critical Information Systems 23
Heuristics to Perform Molecular Decomposition of Large Mission-Critical Information Systems 29
Generation of Diagnostic Plans for Large ICT Systems 35
Dependability Analysis of Information Systems with Hierarchical Reconfiguration of Services 40
Hierarchical Approach to Dependability Analysis of Information Systems by Modeling and Simulation 45
Towards the Definition of a Web Service Based Management Framework 51
Dependability and Security Metrics in Controlling Infrastructure 59
Advanced Rule-Based Techniques in Mission Critical Systems 67
Event-Driven Architecture for Intrusion Detection Systems Based on Patterns 71
Agent Based Approach to Events Monitoring in Complex Information Systems 78
On System Security Metrics and the Definition Approaches 85
Policy-Driven System Configuration for Dependability 91

Appendix B: Expert panel session 97
Security & Dependability in Complex & Critical Information Systems 98
Future research challenges in dependability 103
Dependability and Security in Complex and Critical Information Systems 107
Towards trustworthy ICT service infrastructures 109

1 Introduction

The 2nd DESEREC dissemination workshop on "Dependability and Security in Complex and Critical Information Systems" took place in Cap Esterel (France) from 25th to 31st of August, 2008. This workshop is part of SECURWARE 2008, both sponsored by IARIA. The official website where the reader can find all the information related to this dissemination workshop is available at:

http://www.iaria.org/conferences2008/DEPEND.html

DEPEND 2008 was conceived as a forum for detailed exchange of ideas, techniques, and experiences with the goal of understanding the academia and the industry trends related to the new challenges in dependability and security in critical and complex information systems.

Figure 1: Some photos taken during the workshop

Researchers, designers and administrators, product and service developers, and other interested people from academia, industry, and the government were encouraged to participate in this workshop with unpublished contributions in several areas, including:

- Building and preserving scalable, secure and resilient architectures.
- Dependability modelling and dynamic management policies.
- Verification and validation (including model checking) of dependable software architectures.
- Real time detection and recovery capabilities against intrusions, malfunctions and failures.
- Redundancy and reconfiguration architectures.

- Integrated response architectures.
- Planning of optimal configurations for anticipated operational modes.
- Modelling of networks and Information Systems.
- Simulation of modelled configurations.
- Fast reconfiguration with priority to critical services.
- Incident (including intrusion) detection and quick containment.

DEPEND 2008 has tried to address new approaches dealing with today's limitations and covering different complementary topics on most of critical activities, which belong to the areas of communications (telephone, Internet), energy and fluids (electricity, gas, water), transportation (railways, airlines, road), health and emergency response, etc. by relying on networked communication and information systems (CIS's).

Along with the workshop, the DESEREC consortium organized an expert panel about "Dependability and Security in Complex and Critical Information Systems" in which, and thanks to the valuable help of well-known experts –such as Dr. Michel Riguidel (Ecole Nationale Supérieure des Télécommunications), Dr. Antonio F. Gómez Skarmeta (University of Murcia), Dr. Aljosa Pasic (Atos Origin) and Yves Paindaveine (Information Society and Media DG, European Commission)–, researchers and managers, coming from academia and industry of several European countries and involved in European projects in the ICT field, found a good and valuable opportunity for exchanging and merging research experiences and drawing the lines for future activities.

All conference proceedings presented in this workshop have been published by IEEE Computer Society Press (CPS). These publications are captured in the online IEEE Digital Library, and professionally indexed through INSPEC and EI Index (Elsevier's Engineering Information Index).

Figure 2: Main frame of the CD material delivered to registered people

These papers are also published online through IEEE Xplore Digital Library. The Online Public Access Catalog (OPAC) for the conference SECURWARE 2008 is available at http://ieeexplore.ieee.org/servlet/opac?punumber=4622544. In this URL the reader can search and download the publications presented in the workshop DEPEND 2008. The conference proceedings have been included and delivered in a CD-ROM (see Figure 2) to every registered people in the conference SECURWARE.

2 Workshop organization

The workshop was organized by the DESEREC consortium, with the coordination action help of the conference SECURWARE. People belonging to the technical program committee come from the below organizations.

Technical Program Committee

André Cotton	Thales Communications, France
Wolfgang Fritsche	IABG, Germany
Antonio F. Gómez Skarmeta	University of Murcia, Spain
Antonio Lioy	Politecnico di Torino, Italy
Javier López	University of Malaga, Spain
Fabio Martinelli	CNR, Italy
Syed Naqvi	CETIC, Belgium
Aljosa Pasic	Atos Origin, Spain
Patrick Radja	Advanced Communication and Security Programs, EADS, France
Michel Riguidel	Ecole Nationale Supérieure des Télécommunications, France
Reijo Savola	VTT Technical Research Centre of Finland, Finland
Neeraj Suri	University of Darmstadt, Germany
Daniel Vladusic	Xlab, Slovenia
Wojciech Zamojski	Wroclaw University of Technology, Poland

Thanks to the Technical Program Committee efforts, around 30 people attended the DEPEND workshop regularly, and almost 75 people attended to the global conference SECURWARE.

2.1 Preliminary program

Besides DEPEND 2008, others conferences and workshops have been held all together under the IARIA sponsorship, namely:

- **SECURWARE**: Second International Conference on Emerging Security Information, Systems and Technologies
- **SENSORCOMM**: Second International Conference on Sensor Technologies and Applications
- **MESH**: International Conference on Advances in Mesh Networks
- **UNWAT**: Second International Workshop on Under Water Sensors and Systems
- **ENOPT**: Second International Workshop on Energy Optimization in Wireless Sensor Networks

Thus, all registered people to the DEPEND 2008 have enjoyed attending to others presentations, focused on different areas like sensors and mesh networks.

Next it is shown the complete program, bringing out the DEPEND sessions in green.

Monday, August 25

12:00	Registration starts
14:00-17:00	**Peer-to-Peer Services in Community Mesh Networks** Andreas J. Kassler and Marcel Cavalcanti de Castro University of Karlstad, Sweden
18:00-19:00	Welcome cocktail

Tuesday, August 26

9:00-9:15	**Opening session** Prof. Dr. Petre Dini, Cisco Systems, Inc., USA / Concordia University, Canada Prof. Dr. Pascal Lorenz, University of Haute Alsace, France Prof. Dr. Jaime Lloret Mauri, Polytechnic University of Valencia, Spain Cisco Distinguished Engineer, Jean Philippe Vasseur, Cisco Systems, Inc., France		
9:15-10:15	Invited Speaker **The Internet of things** Jean Philippe Vasseur, Cisco Systems, Inc., France		
10:15-10:30	Coffee break		
10:30-12:15	SENSORCOMM1	SECURWARE1	MESH1

13:45-15:30	SENSORCOMM2	SECURWARE2	**DEPEND1**
15:30-15:45	Coffee break		
15:45-17:30	SENSORCOMM3	SECURWARE3	**DEPEND2**
17:30-18:30	**Working Group Meeting: Advances on Sensors** **NNN- New Nerve Network for Northern Waters** Jens M. Hovem **Open Discussion: topics, journals**		

The first two-day program started with plenary sessions for introductory and keynote speeches, and then they went on with parallel sessions.

Wednesday, August 27

8:30-10:15	SENSORCOMM4	SECURWARE4	MESH2
10:15-10:30	Coffee break		
10:30-12:15	SENSORCOMM5	SECURWARE5	UNWAT1

13:45-15:30	SENSORCOMM6	SECURWARE6	ENOPT1
15:30-15:45	Coffee break		
15:45-17:30	SENSORCOMM7	SECURWARE7	**DEPEND3**
17:30-19:30	**Working Group Meeting: Security and Trust** Expert Panel **Dependability and Security in Complex and Critical Information Systems** Moderators Andre Cotton, Thales Communications, France Antonio Lioy, Politecnico di Torino, Italy Panelists Michel Riguidel, ENST, France Antonio F. Gómez Skarmeta, University of Murcia, Spain Aljosa Pasic, Atos Origin, Spain Yves Paindaveine, Information Society and Media DG, European Commission **Open Discussion: topics, journals**		

Thursday, August 28

8:30-10:15	SENSORCOMM8	SECURWARE8	**DEPEND4**
10:15-10:30	Coffee break		
10:30-12:15	SENSORCOMM9	SECURWARE9	MESH3

13:45-15:30	SENSORCOMM10	SENSORCOMM17	UNWAT2
15:30-15:45	Coffee break		
15:45-17:30	SENSORCOMM11	SENSORCOMM18	SENSORCOMM22
17:30-17:45	Coffee break		
17:45-19:30	SENSORCOMM12	SENSORCOMM19	SENSORCOMM23

Friday, August 29

8:30-10:15	SENSORCOMM13	SECURWARE10	MESH4
10:15-10:30	Coffee break		
10:30-12:15	SENSORCOMM14	SECURWARE11	SENSORCOMM24

13:45-15:30	SENSORCOMM15	SENSORCOMM20	SENSORCOMM25
15:30-15:45	Coffee break		
15:45-17:30	SENSORCOMM16	SENSORCOMM21	SENSORCOMM26

2.2 Detailed program for the DEPEND workshop

In this section the program of the DEPEND workshop is shown in more detail, by including the presentation titles and the corresponding authors. Note that three of them are out of the DESEREC consortium.

Session	**DEPEND I**
	Tuesday, August 26 (13:45-15:30)
Chair	**André Cotton** Thales Communications, France
Presentations	▪ **Design-Time Learning for Operational Planning Improvement** Daniel J. Martínez, Juan M. Marín, Jorge Bernal, Manuel Gil, and Antonio F. Gómez Skarmeta ▪ **Molecular Approach Paves the Way towards High Resilience for Large Mission-Critical Information Systems** André Cotton, Maurice Israël, and Julien Borgel ▪ **Heuristics to Perform Molecular Decomposition of Large Mission-Critical Information Systems** Maurice Israel, Julien Borgel, and Andre Cotton ▪ **Generation of Diagnostic Plans for Large ICT Systems** Marco D. Aime and Andrea Atzeni

Session	**DEPEND II**
	Tuesday, August 26 (15:45-17:30)
Chair	**Antonio F. Gómez Skarmeta** University of Murcia, Spain
Presentations	▪ **Dependability Analysis of Information Systems with Hierarchical Reconfiguration of Services** Dariusz Caban and Wojciech Zamojski ▪ **Hierarchical Approach to Dependability Analysis of Information Systems by Modelling and Simulation** Katarzyna Michalska and Tomasz Walkowiak ▪ **Towards the Definition of a Web Service Based Management Framework** Juan Manuel Marín Pérez, Jorge Bernal Bernabé, Daniel J. Martínez Manzano, Gregorio Martínez Pérez, and Antonio F. Gómez Skarmeta ▪ **Dependability and Security Metrics in Controlling Infrastructure** Imre Kocsis, György Csertán, Péter László Pásztor, and András Pataricza

Session	**DEPEND III**
	Wednesday, August 27 (15:45-17:30)
Chair	**Manuel Gil Pérez** University of Murcia, Spain
Presentations	▪ **Advanced Rule-Based Techniques in Mission Critical Systems** Maite Avelino and Francisco de la Torre ▪ **Fast Algorithms for Local Inconsistency Detection in Firewall ACL Updates** S. Pozo, R. Ceballos, R. M. Gasca, and A. J. Varela-Vaca Note: *Presentation does not belong to the DESEREC consortium* ▪ **Event-Driven Architecture for Intrusion Detection Systems Based on Patterns** Jesús J. Martínez Molina, Miguel A. Hernández Ruíz, Manuel Gil Pérez, Gregorio Martínez Pérez, and Antonio F. Gómez Skarmeta ▪ **Agent Based Approach to Events Monitoring in Complex Information Systems** Marek Woda and Tomasz Walkowiak

Session	**DEPEND IV**
	Thursday, August 28 (8:30-10:15)
Chair	**Pirkko Kuusela** VTT Technical Research Centre, Finland
Presentations	▪ **A Synchronization Attack and Defense in Energy-Efficient Listen-Sleep Slotted MAC Protocols** Xiaoming Lu, Matt Spear, Karl Levitt, Norman S. Matloff, and S. Felix Wu Note: *Presentation does not belong to the DESEREC consortium* ▪ **On System Security Metrics and the Definition Approaches** Artur Hecker ▪ **Policy-Driven System Configuration for Dependability** Marco D. Aime, Paolo C. Pomi, and Marco Vallini ▪ **A Dependability Case Approach to the Assessment of IP Networks** Ilkka Norros, Pirkko Kuusela, and Pekka Savola Note: *Presentation does not belong to the DESEREC consortium*

3 Next events related to DEPEND 2008

3.1 Special issue of IJCCBS

The Editor-in-Chief for a new journal named International Journal of Critical Computer-Based Systems (IJCCBS, by Inderscience Publishers) has contacted DEPEND chairmen for a possible special issue of this workshop. They have accepted the offer with a great pleasure for DESEREC since this supposes a good dissemination opportunity for the project.

In this special issue will be included extended and revised versions of selected papers from the DEPEND workshop. The list of these papers has been selected by DEPEND chairmen, together with DESEREC partners involved in WP5. The first four papers listed below were selected automatically since they have been selected as "Best Papers" in the conference SECURWARE 2008.

This list is the following one, where five papers of them belong to the DESEREC consortium:

- **Dependability Analysis of Information Systems with Hierarchical Reconfiguration of Services**
  Dariusz Caban and Wojciech Zamojski
  Note: *Paper awarded as "Best Paper" in the conference SECURWARE 2008*
- **A Dependability Case Approach to the Assessment of IP Networks**
  Ilkka Norros, Pirkko Kuusela, and Pekka Savola
  Note: *Paper awarded as "Best Paper" in the conference SECURWARE 2008*
- **Event-Driven Architecture for Intrusion Detection Systems Based on Patterns**
  Jesús J. Martínez Molina, Miguel A. Hernández Ruíz, Manuel Gil Pérez, Gregorio Martínez Pérez, and Antonio F. Gómez Skarmeta
  Note: *Paper awarded as "Best Paper" in the conference SECURWARE 2008*
- **Heuristics to Perform Molecular Decomposition of Large Mission-Critical Information Systems**
  Maurice Israel, Julien Borgel, and Andre Cotton
  Note: *Paper awarded as "Best Paper" in the conference SECURWARE 2008*
- **On System Security Metrics and the Definition Approaches**
  Artur Hecker
- **Policy-driven system configuration for dependability**
  Marco D. Aime, Paolo C. Pomi, Marco Vallini
- **Fast Algorithms for Local Inconsistency Detection in Firewall ACL Updates**
  S. Pozo, R. Ceballos, R. M. Gasca, A. J. Varela-Vaca

Further information about journal topics and editorial board can be found on IJCCBS home page:

http://www.inderscience.com/ijccbs

3.2 DEPEND 2009

Due to DEPEND 2008 has been a very successful event, next year is planned to go on in this way and to celebrate a second conference, although in this occasion won't be linked to the DESEREC project.

The "Second International Conference on Dependability (DEPEND 2009)" will be part of the conference NETWARE 2009, together with other events such as SECURWARE, SENSORCOMM,

MESH and AFIN, will take place in Athens (Greece) from 18th to 23rd of June, 2009. The official website where the reader can find all the information related to this conference is available at:

http://www.iaria.org/conferences2009/DEPEND09.html

The topics suggested by the conference can be discussed in term of concepts, state of the art, standards, implementations, running experiments and applications. Authors are invited to submit complete unpublished papers, which are not under review in any other conference or journal in the following, but not limited to, topic areas:

- Dependability facets.
- Adaptability and (self)adaptability.
- Adaptability and dependability.
- Dependability and security.
- Trust and system dependability.
- Dependability, adaptability, and new technologies.

The complete and specific list of topic areas can be checked online in the section “Call for Papers”.

4 Conclusions

The "First International Workshop on Dependability and Security in Complex and Critical Information Systems (DEPEND 2008)", which corresponds with the 2nd DESEREC dissemination workshop, was held as part of conference SECURWARE "Second International Conference on Emerging Security Information, Systems and Technologies" and took place in Cap Esterel (France) from 25th to 31st of August, 2008.

This workshop was a successful event due to the role of the DESEREC project in the organization of the workshop, just as the number and the scientific and technical quality of the presentations provided by the DESEREC partners. Due to this good experience, "Second International Conference on Dependability (DEPEND 2009)" is planned to be celebrated next year in Athens (Greece) from 18th to 23rd of June, 2009.

On the other hand, and thanks to the appreciated help of well-known experts in the areas of security, dependability and trust (please refer to Appendix B where you can find the slides presented by them); all attendees to the special panel organized by DESEREC had the opportunity of exchanging and sharing research experiences and drawing the lines for future activities.

Appendix A: DEPEND 2008 workshop

This appendix encloses all the presentations done by the DESEREC consortium during the workshop.

Design-Time Learning for Operational Planning Improvement

Daniel J. Martínez, Juan M. Marín, Jorge Bernal, Manuel Gil, and Antonio F. Gómez Skarmeta

University of Murcia, Spain

Paper abstract

This document describes how the DESEREC framework can be enhanced through the usage of an offline learning loop. This mechanism consists basically of computational methods and techniques which enable the framework to gradually improve the effectiveness of its behaviour, by analysing internal data produced by some of the modules, applying some learning algorithms to them and to the information received from the administrator and using then the results for enhancing the framework operation.

The learning loop presented here is strictly related to the improving the quality of the generated operational configurations, which is achieved by adjusting the decisions made by the Planning module while carrying out its task.

Design-time Learning for Operational Planning Improvement

Daniel J. Martinez Manzano
Dept. Information and Communications Engineering, University of Murcia

1st International Workshop on Dependability and Security in Complex and Critical Information Systems (DEPEND 2008)
25th-31st August 2008, Cap Esterel, France

Contents

- Introduction
- The DESEREC Framework
- Proposal of a Learning Loop
- Implementation Considerations
- Conclusion

Introduction

- IST Project: IST-2004-026600
- Target: to define a framework to increase the dependability of existing and new networked Information Systems, responding efficiently to incidents
- Website: **http://www.deserec.eu**

DESEREC
Dependable Security by Enhanced Reconfigurability

Introduction

- Reconfiguration driven by switching across alternative configurations
- Modelling = representation of requirements and resources information with a set of XML-based metamodels
  - Automated processing, extensibility, …
- Planning = generation of operational data:
  - Configurations (based on requirements)
  - Reactions (switching rules, based on possible incidents)

Introduction

- Effectiveness of the whole framework depends heavily on the suitability of the generated operational models
  - Full automation is risky
- Human guided generation is a solution, but should be kept to a minimum

Re-use human expertise by adding learning capabilities to the framework

The DESEREC Framework

- Two major phases:
  - *Design time*: generation of operational models
    - Modelling (system, services, requirements)
    - Planning (configurations, reactions)
  - *Run time*: usage of operational models to keep managed system working as expected, even after incidents occur
    - Configuration
    - Monitoring
    - Reconfiguration

The DESEREC Framework

- The output of the Planning module (configurations + reactions) is analyzed by an Analysis module
  - The Analysis module uses some metrics / indicators to tag the configurations
  - Can also drop some of them
  - And also spot undesirable / incomplete reactions
- Learning capabilities can be most useful at this point, to make Planning learn from the results of Analysis

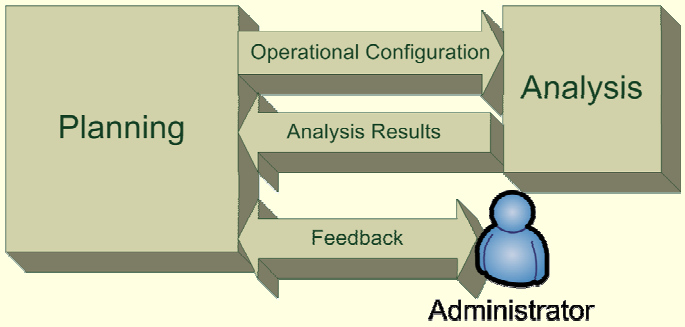
Proposal of a Learning Loop

- Analysis provides a kind of “score” of each operational configuration (OC), by using metrics / indicators
- A low score may indicate that Planning has done something wrong
  - We want it to learn from this mistake
- The score alone does not suffice: Planning needs to know **why** the OC was scored poorly

Proposal of a Learning Loop

- Full automation here is not desired, since the admin might want to keep a poorly scored OC
  - We could be using an overloaded server, because the main one has become down
- The results of the Analysis module should be presented to the admin, and his/her decision should be fed to the learning engine
- With time, the framework would require less and less human guidance
  - "Spam filter"-like behaviour

Proposal of a Learning Loop



Implementation Considerations

- The major implementation steps are:
  - Choose a set of plausible **Planning mistakes**, which Analysis is able to spot
  - Develop a **meta-model for the Analysis results**, able to notify the Planning module about the mistakes made
  - Work out a way to perform a **supervised learning process**, from the Analysis results

Implementation Considerations
Choosing the Planning Mistakes

- Definition of a "working set" of contingencies which will be regarded as mistakes of the Planning process

Type	Description
rejected	Disapproved by administrator
overloaded	Too many allocations on same place
deprecated	Sub-optimal software is used
vulnerable	Risky software is used
incompatible	Incompatible services put together

Implementation Considerations
Modelling the Analysis Results

- These results must notify the Planning module about mistakes spotted in the OCs
- XML a natural choice for same reasons as in the rest of meta-models
  - Extensibility
  - Automation
  - …
- Plus one important addition: **XPath** can be used for pointing to the "bad" part of an OC

Implementation Considerations
Modelling the Analysis Results

- This meta-model allows Analysis to say "what" is wrong in an OC
- The working set of mistakes provides the "why"
  - A severity value can also be added to each mistake type, to be used as a weight factor

Implementation Considerations
Modelling the Analysis Results

```
<xsd:complexType name="mistakeComplexType">
  <xsd:sequence>
    <xsd:element name="location" type="xsd:string" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:attribute name="type" use="required">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="rejected" />
        <xsd:enumeration value="overloaded" />
        <xsd:enumeration value="deprecated" />
        <xsd:enumeration value="vulnerable" />
        <xsd:enumeration value="incompatible" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:attribute>
  <xsd:attribute name="severity" type="xsd:integer" use="required" />
</xsd:complexType>
```

XPath locator expression

Mistake type

Working set of mistakes

Severity value

Implementation Considerations
Learning Process

- Roughly speaking, Planning decisions allocate services (to be provided) onto infrastructure elements (able to provide them)
- This part of an OC is thus a set of ***s*→*e*** items, where:
  - ***s*** is a service to be provided
  - ***e*** is an infrastructure element (host, software…) which can implement it

Implementation Considerations
Learning Process

- If the Analysis module finds any of these items to be bad, it creates a *mistake* structure for it, and adds it to the Analysis results
  - The "location" field points to the offending *s→e* item (via XPath)
  - The "type" field indicates why that item is bad

Implementation Considerations
Learning Process

- Together with the "severity" field (weight factor), these are basic data for a training sample for the learning engine
  - Additional data needed: set of services and set of infrastructure elements

Implementation Considerations
Learning Process

- Complete sample definition: ***{s→e, w, {s_1…s_n}, {e_1…e_m}}***, where:
  - ***s→e*** is the offending allocation
  - ***w*** is the weight factor
  - ***{s_1…s_n}*** is the set of services to be provided
  - ***{e_1…e_m}*** is the set of available infrastructure elements
- Can be used with existing supervised learning algorithms which use weighted samples

Implementation Considerations
Learning Process

- Each allocation generated by Planning may have a holder for a binary tag, stating whether the allocation is good or not
- This value can either come from the human admin, or be predicted by the learning classifier engine
- This allows the classifier to evolve based on the human input, requiring less and less of it with time

Conclusion

- The DESEREC framework has a good potential for enhancement through learning techniques
  - The nature of the workflow advices to rely some times on human input
  - Learning enables the framework to get more and more independent from it

Conclusion

- The workflow loops can be used as bases for other types of looping computing, such as learning as in this case (envisioned as a part of the design time loop)
- The proposed loop does not require a specific learning algorithm
  - Any supervised classifier using weighted samples is a candidate

Design-time Learning for Operational Planning Improvement

Daniel J. Martinez Manzano
Dept. Information and Communications Engineering, University of Murcia

Thank you for your attention!

Molecular Approach Paves the Way towards High Resilience for Large Mission-Critical Information Systems

André Cotton, Maurice Israël, and Julien Borgel

Thales Communications, France

Paper abstract

The molecular approach was introduced in the DESEREC project to describe large Information Systems in a manageable way. This level of abstraction bring to each person running a Business Service the capability to optimize the cost of the level of dependability committed while supporting the Security and Infrastructure administrators duties. The coordination of those management roles under the priority given to the Business-level agreements achieves a maximum of resilience for a minimum of redundant resources.

This molecular approach is presently used by the modelling and simulation tools and is being implemented in the three tiers of the management infrastructure: local reaction (self-healing), global reaction (reconfiguration), and learning loop.

This level of abstraction allows near real-time decisions on reaction and reconfiguration even on large

Information Systems and eases the human understanding of the reasoning used by the decision engines. It also brings the information needed for the evolution of the rule sets governing the self-healing and the reconfiguration.

DEPEND-2008
International Workshop on Dependability and Security
in Complex and Critical Information Systems
25-29 August, Cap Esterel, France

Molecular approach paves the way toward high resilience for large mission-critical Information Systems

André Cotton, Maurice Israël, Julien Borgel
Thales Communications, Colombes, France
Andre.cotton@fr.thalesgroup.com
Maurice.israel@fr.thalesgroup.com
Julien.borgel@fr.thalesgroup.com

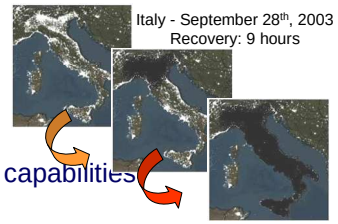
DESEREC
Dependability & Security by Enhanced Reconfigurability

Information Society Technologies

Dependability concerns

- The everyday life of European citizens relies on critical activities supported by networked Information Systems (I.S.):
  - Communications (telephone, Internet)
  - Energy & fluids (electricity, gas, water)
  - Transportation (railways, airlines, road)
  - Health and emergency response

- So far, limited taken actions let these I.S.
  - not failure-proof enough to face:
    - Software & hardware faults
    - Malicious actions: intrusion, virus
  - with poor self-healing capability
    - and therefore sensitive to cascading effects
  - suffering long recovery time



- The DESEREC project aims at leveraging those capabilities
  - in new and existing Information Systems

DESEREC High level objectives

- Ensure the Information system resilience
  - Decreasing over-provisioning of physical resources and avoiding redundancy of equipments.
  - Provide high-level information to business managers: SLA faults, performance of a business service from a provider point of view (and not only from a technical point of view)
  - Give to the managers the capacity to decide of configuration changes or services priority according to business and economical criterion, independently of technical issues.

Endure the technical constraints vs decide your business priority!

Deserec approach

- To improve the resilience of large and complex mission-critical Information Systems
  - By managing their dependability ,
  - By mastering their security,
  - By enabling smart reconfigurations of the resources,
  - By foreseeing and mastering a reconfiguration impact on the IS and more especially on critical Business Services,
  - By simulating the effect of a configuration change and finally,
  - By constantly optimize and tune the framework to stick the IS and BS evolution

A multi-tiered response driven by three objectives

First objective – ***Detect & Prevent***

- Detect proactively incident and potential fault
- Keep as much as possible every failure local
  - Contain the incident: isolate the compromised area

Second objective - ***React***

- Sustain or quickly resume the critical applications
- Reallocate resources used by less critical ones

Third objective – ***Plan***

- Reallocate optimally the resources to recover the full range of services
- Validate the configurations by simulation

The technical objectives as set by DESEREC

- Keep as much as possible every failure local
  - By containment of compromised or failed devices
    - Early detect: distributed monitoring
    - Identify the suspicious area: scope shaping
    - Contain the incident: cut off on the insulation border

Containment

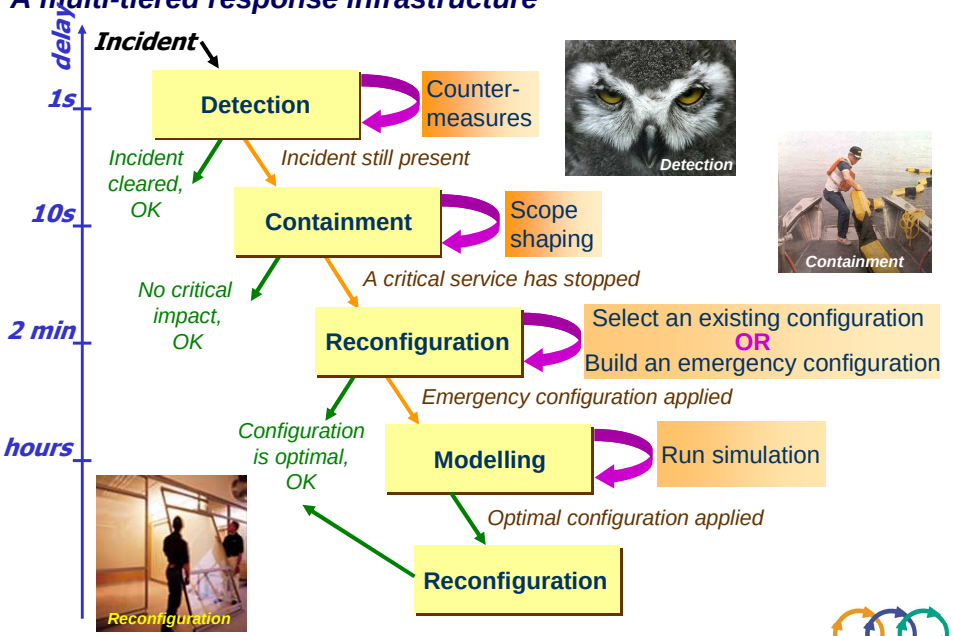
- Enable recovering capabilities
  - To resume quickly the most critical applications
    - By re-allocating the available resources
    - Even in unpredictable situations
  - Planning and simulation tools to restore full services
    - With partial resources

Reconfiguration

- Plan the most probable under-nominal modes
  - With an optimal use of available resources
    - Simulation runs validate modelling
  - Deployment process

Planning

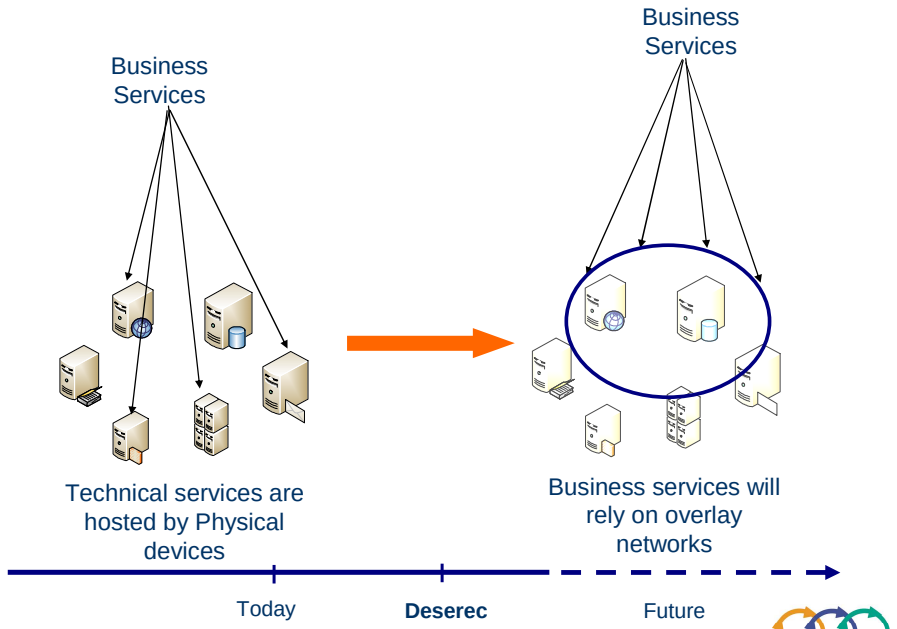
DESEREC expected outcome: A multi-tiered response infrastructure



Information Systems in a nutshell

- IS provide Business Services: the ability for an end-user to increase his productivity, taking into account security, safety and dependability of the service.
- Those BS are relying on Technical Services hosted by physical devices.
- Those physical devices provide 3 key resources, namely:
  - Storage capacities
  - Connectivity
  - Computation capacities
- End-users are interested in Business Services, no matter the underlying technical solutions could be

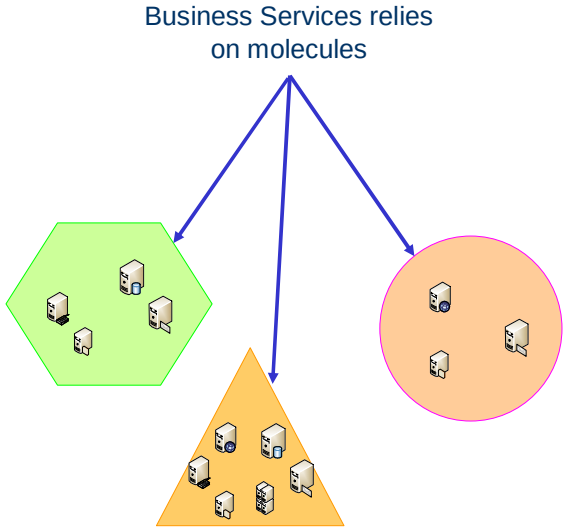
Physical devices evanescence



DESEREC molecular approach

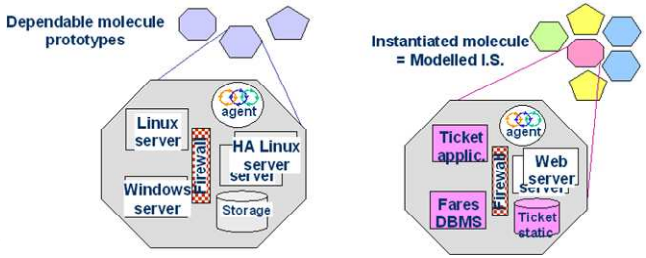
Molecule models are much more simple than detailed ones!

Therefore the time needed to test, validate a new configuration is largely decreased



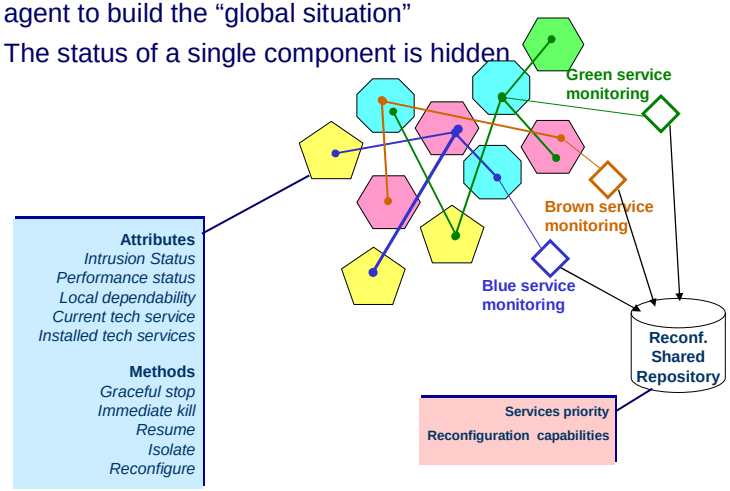
What is a molecule

- A logical system decomposition
- An encapsulated set of devices with some dependability properties hosting a set of technical services
- A molecule is:
  - *self-observable*, i.e. complex enough to provide a reliable status on its own dependability and performance.
  - *self-manageable*, smart enough to implement high-level management requests on its components
  - and capable of autonomous *self-healing* actions in response to certain events.
- The IS is designed from a small set of molecule types, instantiated as needed to host the various TS providing BS.

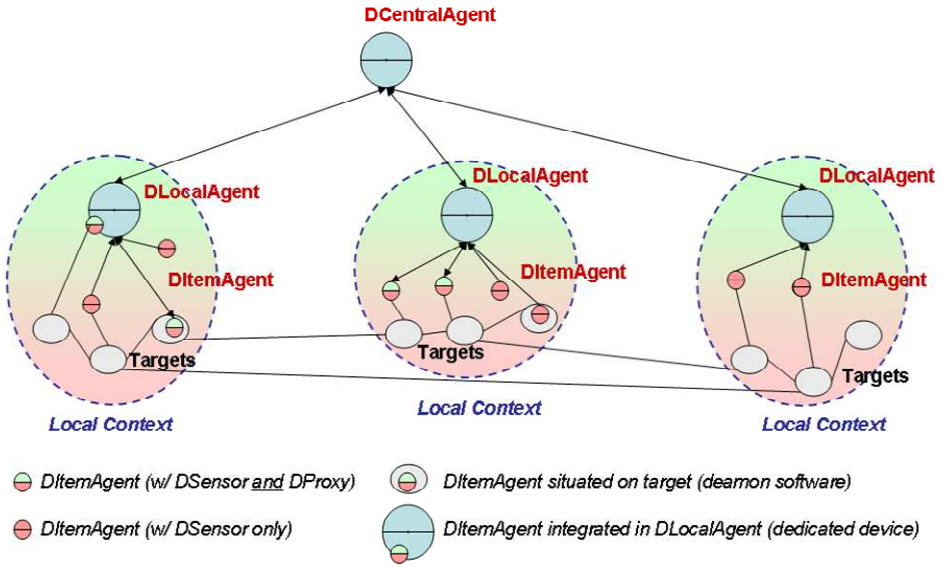


The business service level

- Molecule expose management metrics at the local level to build a so called “local situation”
- Global level metrics are also computed and forwarded to the central agent to build the “global situation”
- The status of a single component is hidden



Global Architecture



Work at the molecule level!

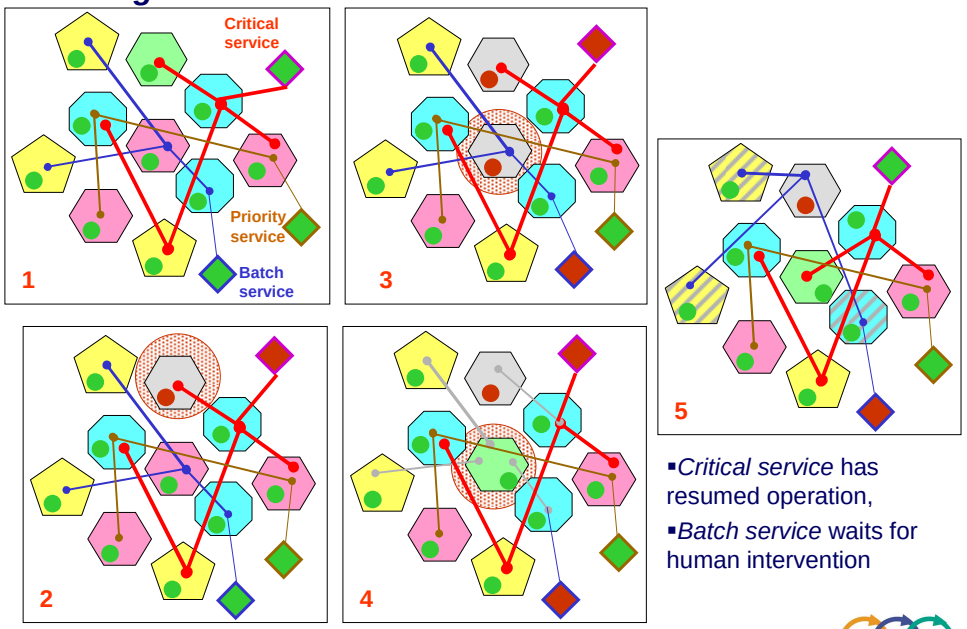
Providing that systems we are addressing are complex, management tasks like:

- Modelling
- Simulation
- Vulnerability analysis
- Reconfiguration impact
- Detection/Reaction scenarios

are computed at the molecule level, even if it lightly damages the accuracy of the results.

The final goal is to reduce time needed to evaluate a configuration and consequently speed up the decision process while mastering its consequences on the business

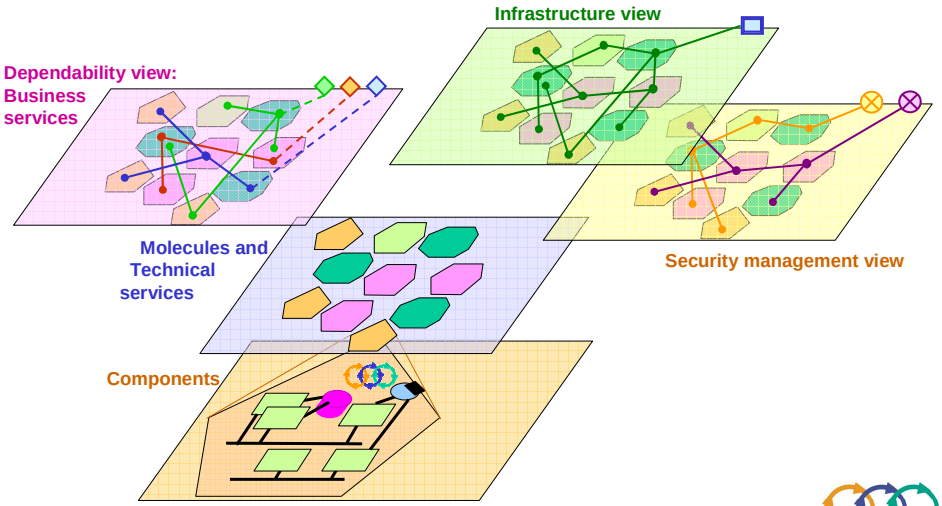
Reconfiguration in action



- *Critical service* has resumed operation,
- *Batch service* waits for human intervention

Multiple views

- Each role benefits of its suited view
- Every business service is managed independently



Progress report

- The consortium has entered the integration phase of the project:
  - Molecular models have been described and used to run both simulation tests and vulnerability analysis
  - Computed configurations are deployed on the target system
  - Local situations are elaborated and high-level metrics are forwarded up to the global level
  - The global situation is computed and updated according to the target system evolution
  - The Deserec framework is validated against 3 industrials test beds
- Next events
  - Validation meetings
  - DESEREC 3rd training workshop, Murcia Spain, 16-17 October 2008, on the results and applications of DESEREC
  - Final demonstration, Athens Greece, December 2008

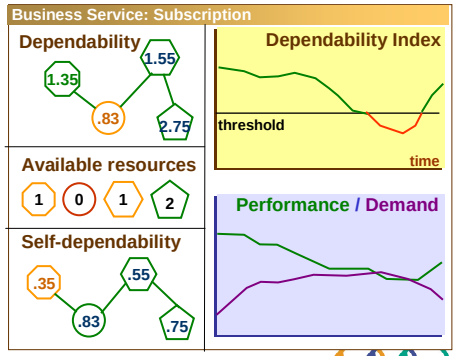
Ultimate goal: dependability-driven management

Instant dependability

- Dependability index
  - The self-dependability index of a molecule is the number of single failures that does not impact the performance
  - The dependability index of a molecule adds the reconfiguration potential
    - And reflect how many single failures could be handled smoothly
  - The Business Service (B.S.) dependability index is a function of the dependability indexes of the Technical Services' molecules.

Demand forecast

- From history and coming events,
  - The demand (#users, transaction rate, …) is forecasted
- Impact on performance
  - Is predicted with simulation tool
  - Performance under threshold could trigger either reconfiguration or provisioning depending on timescale



Heuristics to Perform Molecular Decomposition of Large Mission-Critical Information Systems

Maurice Israël, Julien Borgel, and André Cotton

Thales Communication, France

Paper abstract

The molecular approach has been introduced at the beginning of the DESEREC project to decrease the complexity of managing large mission-critical Information Systems.

Actually it consists in a meta-model based on the fact that gathering relevant Information Systems components together into molecules makes the administration of such complex systems easier. On the one hand, a global and synthetic view of the system is provided. No needs for entering technical details to understand the global situation assess the risks, and decide a reconfiguration. On the other hand, the molecule view simplifies the simulation of the system behaviour and its vulnerability analysis.

Obviously, such a modelling necessarily implies approximations in the computation of performance or dependability metrics. That's the reason why it turns out that it is essential to clearly depict the principles and the rules that lead the system administrator to a given molecule breakdown.

Next to these cutting-up rules we'll try to identify relevant metrics for each type of molecule so that reconfigurations decisions may be taken at the molecule level.

This paper intends to first recall the key principles of the molecule approach and then give rules and principles on how to address the system decomposition in a safe and efficient way.

DEPEND-2008
International Workshop on Dependability and Security
in Complex and Critical Information Systems
August 26th 2008, Cap Esterel, France

Heuristic to perform molecular decomposition of large mission-critical Information systems

André Cotton, Maurice Israël, Julien Borgel
Thales Communications, Colombes, France
Andre.cotton@fr.thalesgroup.com
Maurice.israel@fr.thalesgroup.com
Julien.borgel@fr.thalesgroup.com

DESEREC
Dependability & Security by Enhanced Reconfigurability

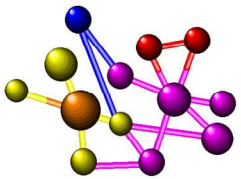
Information Society Technologies

Outline

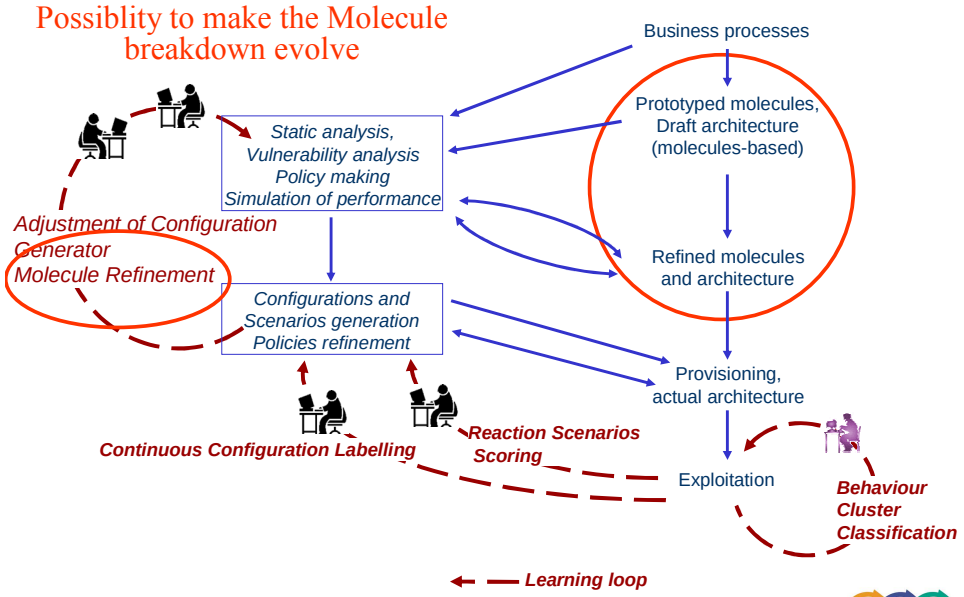
- Molecule concept in a nutshell
- Situation in the Deserec workflow
- Molecule breakdown
- Concrete exemple
- Conclusion

Molecule concept in a nutshell

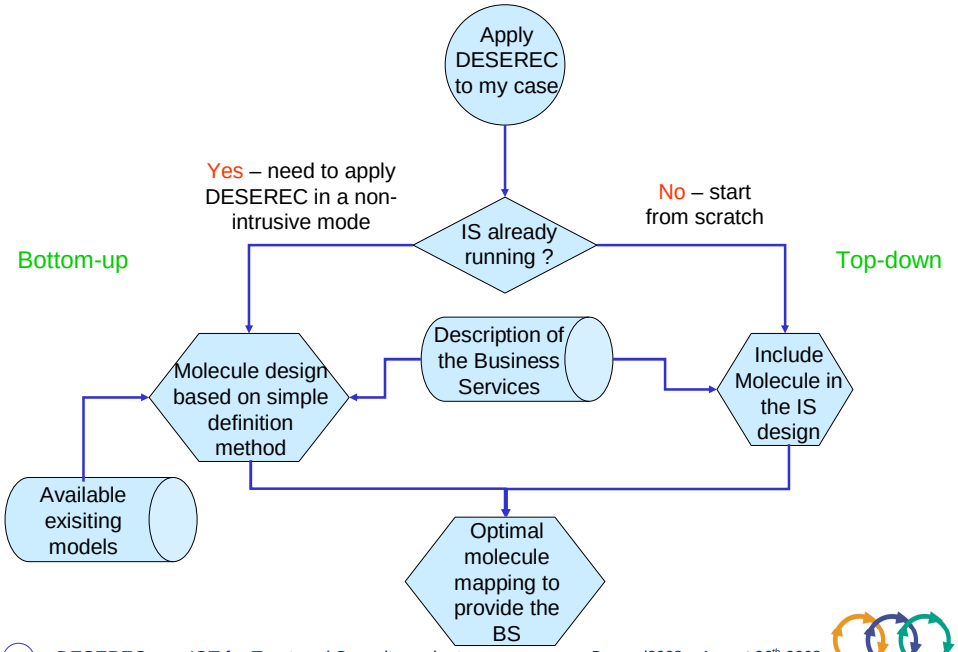
- The Molecule concept is relevant for large and complex mission-critical Information Systems
- Business Services are mapped on Molecules
  - No more directly with Technical Services
- The molecule concept is an abstraction allowing the simplification of the system design
  - “Computer” imply underlying resources like memory, CPU, storage, …
  - “Molecule” imply underlying resources and technical services (servers hosting software, …)



Situation in the Deserec workflow



Molecule breakdown



Molecule design

- In both cases, the molecule design is **guided by indicators** to help the system designer
  - at the design stage to perform the first molecule breakdown to validate a configuration
  - during the DESEREC process thanks to the learning loop that continuously improves the system design to make the molecule definition evolve
- So far, the molecule design is a manual process
- By the experiments performed for the Deserec scenarios on the test beds, a first set of rules to define the Molecule Types has been defined

Molecule design

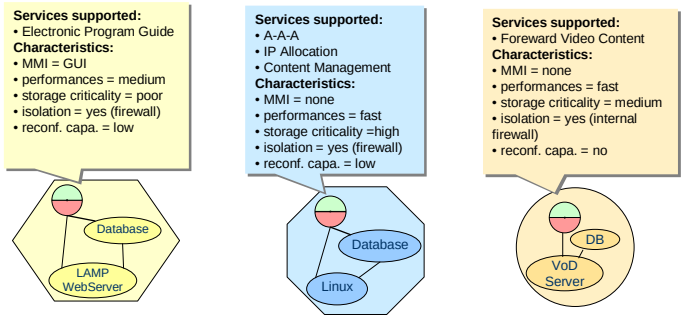
- Initial design
  - Physical limit
    - *"a Molecule Type should be limited to a single physical location"* : in case of a multi-site IS, molecule should be designed in each site
  - Molecule granularity
    - *"number of devices per Molecule Type"* : one single molecule bring no solution, but molecules limited to each and every devices wouldn't reducing the system complexity
  - Molecule type
    - *"gather similar functions into one Molecule type"*
      - limit the number of Molecule Type
      - increase the reconfigurability capabilities
      - balance between the number of different types and the reconfigurability of the system
  - Molecule sensibility
    - *"number of TS per Molecule Type" :* may impact the system in case of reconfiguration

Molecule design

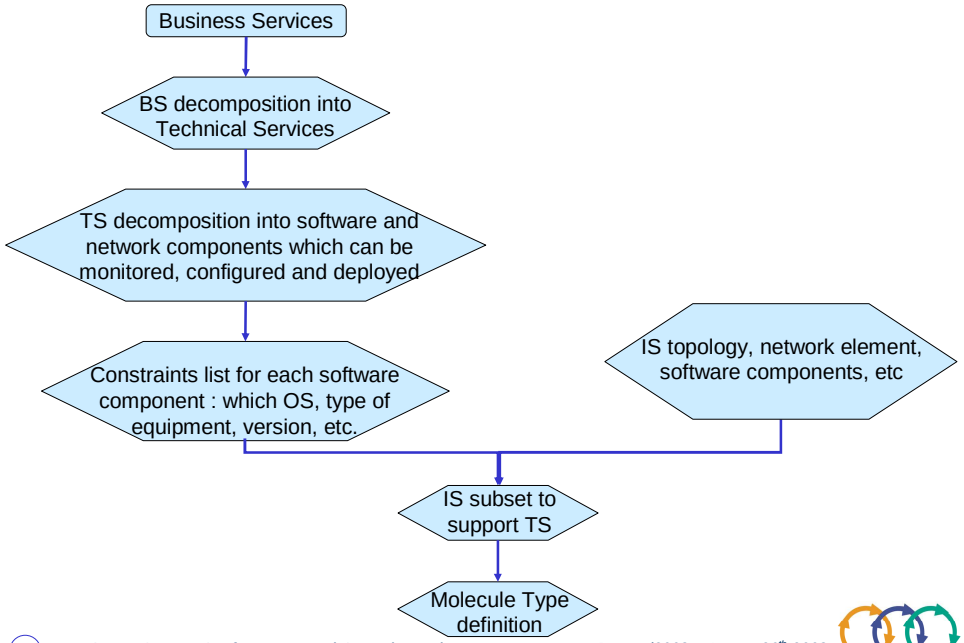
- Incremental design
  - Molecule criticality
    - "*very few instances of a Molecule Type*" : if a Molecule Type is instantiated only once, it introduces a weakness in the reconfigurability capabilities
  - Number of Local reconfiguration scenario
    - "*number of local reconfiguration within the 5 last minutes*" : A molecule involved within too many reconfiguration will probably make the system unstable.
      - If the system is permanently changing, it will of course decrease the Quality of Experience (QoE) of the end users
- To be taken into account after the first configuration generation (end of the design stage)

Top-down approach

- Integration of the molecule concept in the design
- Description of Molecule Types without constraints related to a current implementation
- Possibility to create a molecule Type library to design the BS
  - The system is then built by assembling the molecules instances

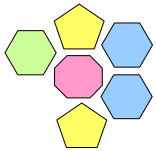


Bottom-up approach



Bottom-up approach

- System description is required
  - fine description of the network equipment, topology, software component, etc.
  - **Pattern identification** : findings of resources shared by the services, then some generic structure, regarding some attributes such as criticality, geographical location, etc…

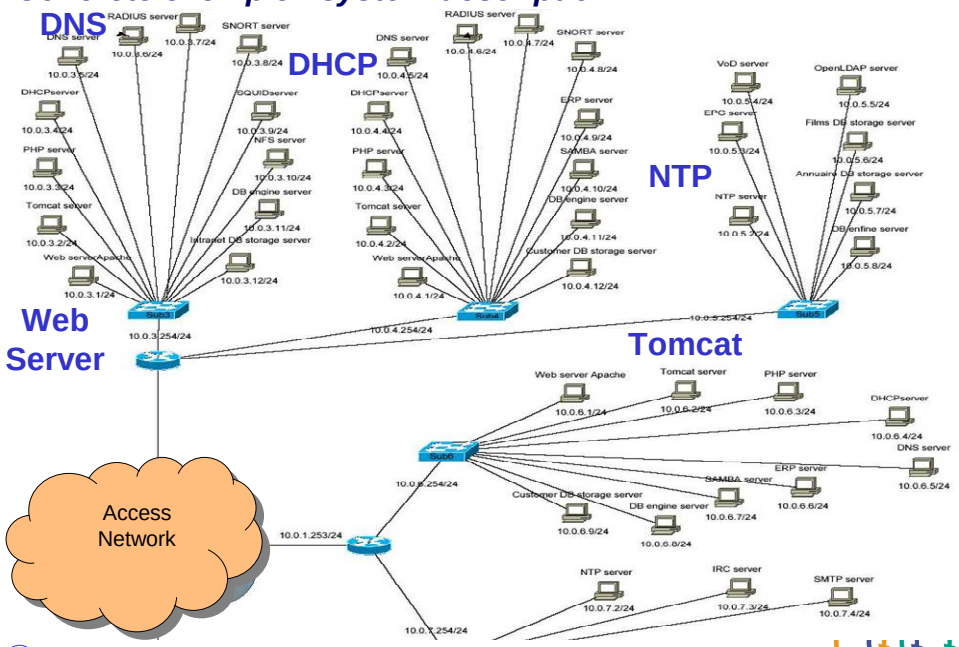


    - Rules are guided by best practices, experience and logic
    - A molecule breakdown is never absolute and definitive :
      - the system is continuously evolving
      - the administrator's view also is changing with time to adjust the IS's representation.
  - As a result, these rules should be seen as principles, heuristic, rather than strict rules that an operator should apply regardless the context.

Concrete exemple - Starting point : Business Services

- Example based on OTE triple play Information System (National Telecom Company in Greece)
- What are the Business services we want to provide :
  - For custumers :
    - VoD
    - Fast Internet
    - IPTV
    - Content for mobiles
  - For team members
    - Login accounting
    - Intranet
    - Hotline

Concrete exemple : system description



Technical Services and Molecule's type

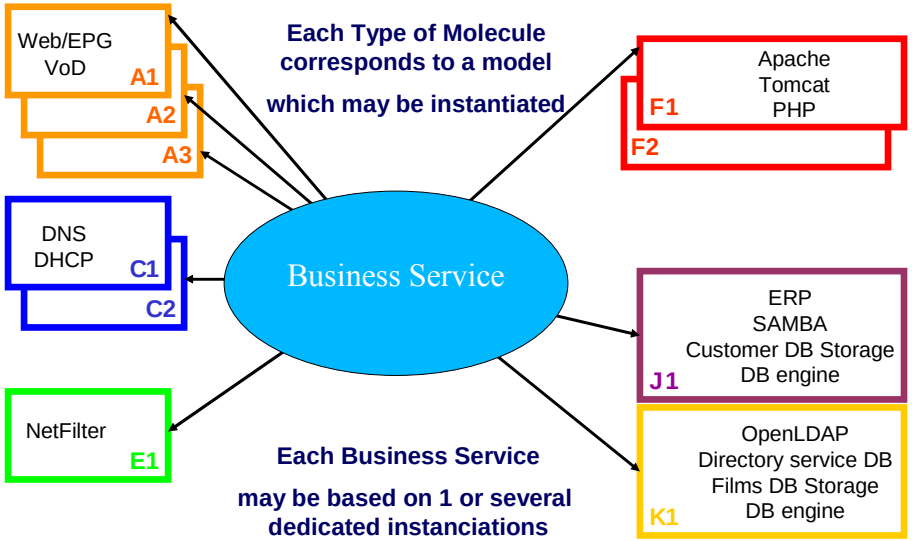
Web/EPG VoD	A
NTP	B
DNS DHCP	C
RADIUS SNORT LNS	D
NetFilter	E
IRC SMTP	G

The Technical Services

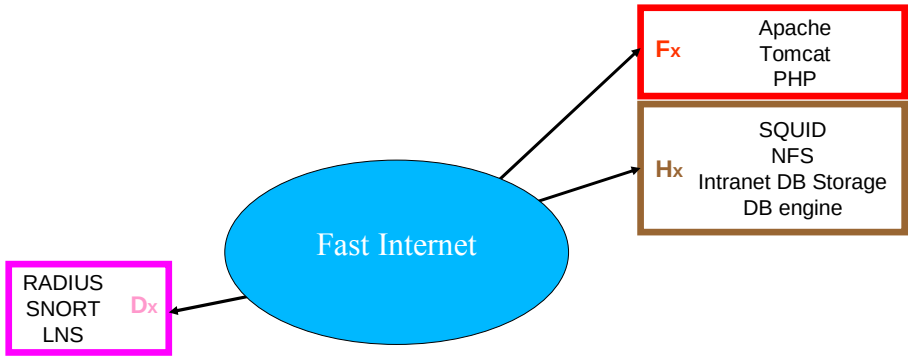
The Molecule types

F	Apache Tomcat PHP
H	SQUID NFS Intranet DB Storage DB engine
J	ERP SAMBA Customer DB Storage DB engine
K	OpenLDAP Directory service DB Films DB Storage DB engine

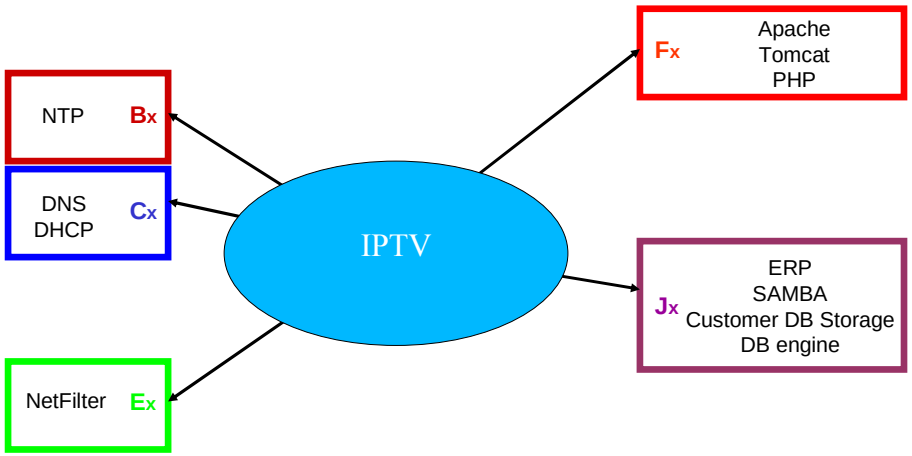
Business Service and Molecule's instantiations principle



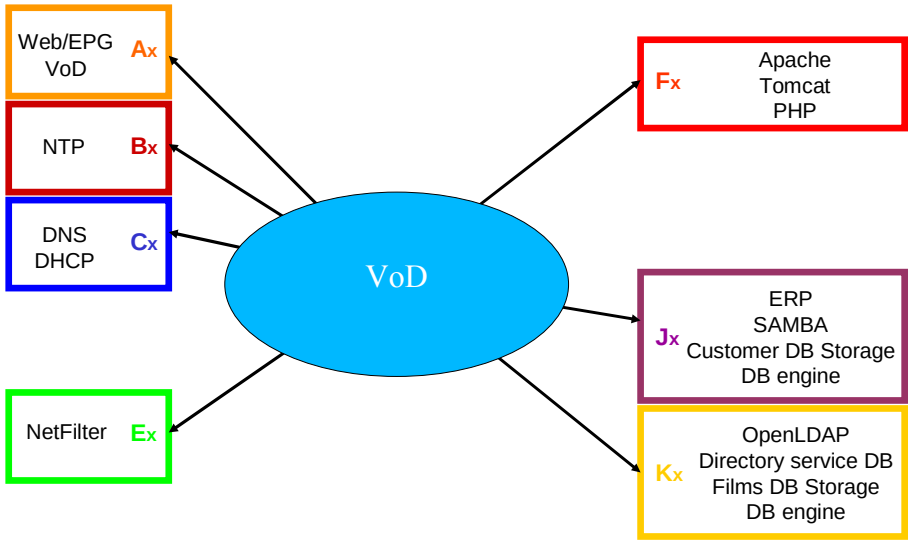
Fast Internet Service



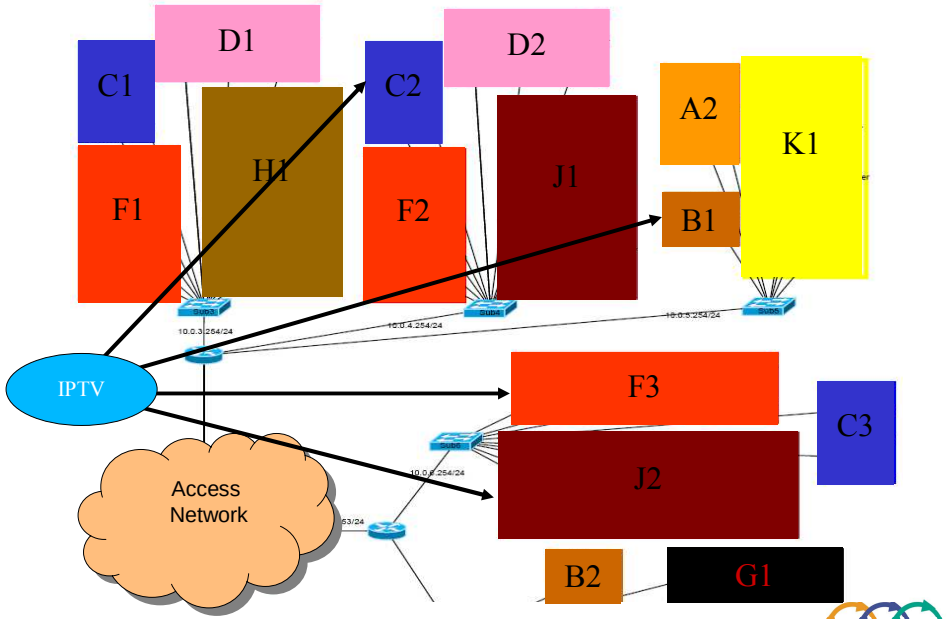
IPTV Service



VoD Service



Concrete exemple



Conclusion

- Expected results
  - an harmonious molecule breakdown that preserves the system reconfigurability, while hiding its complexity
  - Give a simple local and global situation
- Main benefits
  - Increase both the reconfigurability and dependability of complex systems
  - Allow system administrators to keep a global understanding of even complex and large systems.
  - Allow a quicker and safer reaction in case of failure.
- DESEREC experimentations have been performed on realistic test beds
  - Results will be shown during the Deserec Training Workshop in Murcia

Generation of Diagnostic Plans for Large ICT Systems

Marco D. Aime and Andrea Atzeni

Politecnico di Torino, Italy

Paper abstract

This paper proposes a framework for automatic generation of rules to diagnose dependability problems. The generation process starts from formal models of the target system and its configuration, and exploits concepts and techniques from the risk analysis and management field.

Generation of diagnostic plans for large ICT systems

DEPEND'08

Andrea Atzeni (POLITO)
Marco Aime (POLITO)

Tuesday, March 13th

Cap Esterel, France

DESEREC
Dependability & Security by Enhanced Reconfigurability

Information Society Technologies

Risk analysis of ICT systems

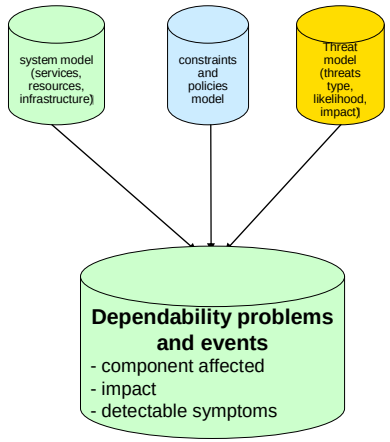
- different methodologies
- value of the assets is defined at high level (business view) - most of the risk is on the low levels
  - e.g. software vulnerability, physical location of the machine
  - difficult understand the real effect of a threat in the whole environment
- some tasks are too costly (resource, time) without an automatic tool
  - **lack of integration at runtime level**

Goals

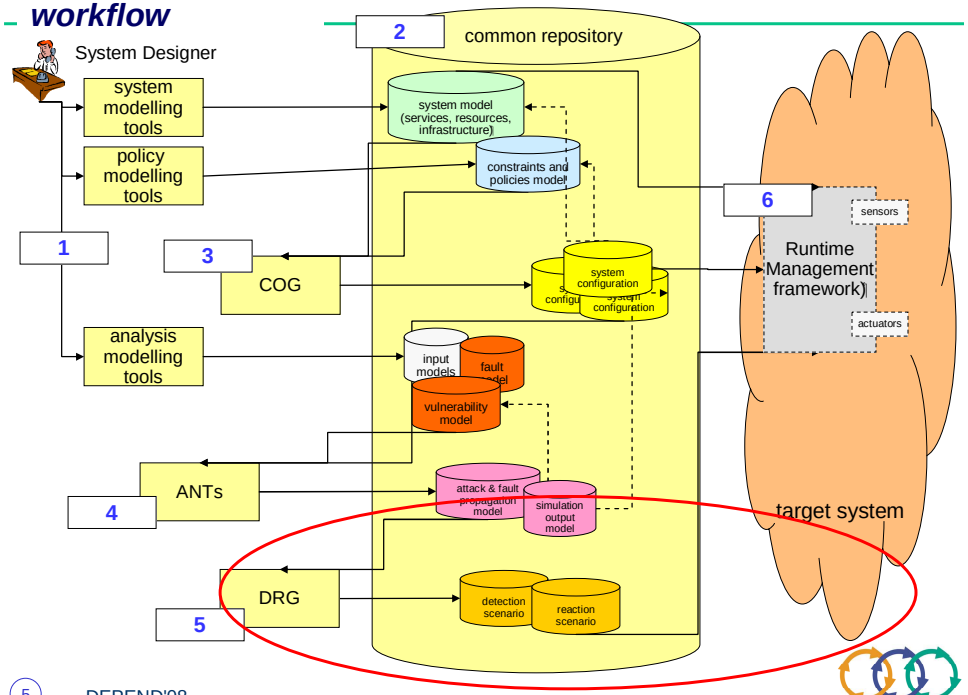
- risk-based intelligence has clear runtime potential benefit
  - prevention
    - to evaluate effectiveness of security controls against an evolving environment
  - detection
    - to better focus the target of sensors and filter their measurement
  - reaction
    - to enforce more flexible reaction policies

our proposal

- by the formal description of
  - system (services, resources)
  - policies (constraints)
  - threats (type, impact, likelihood)
- creation of a knowledge base of correlated dependability problems and events, with specification of detectable signs and possible causes

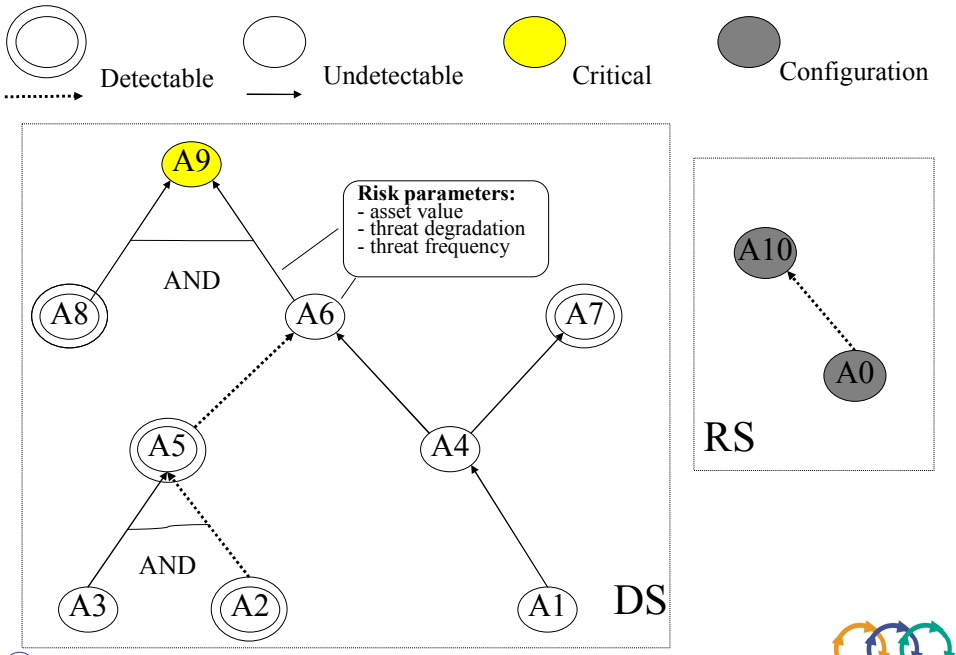


workflow



Detection scenario (DS)

- detection scenario describes diagnostic rules used to identify problems
  - rules computed analyzing current configuration and sensor infrastructure
- they are based on the concept of prognosis-diagnosis-symptoms (PDS) paradigm
- DS is organized in graph of events, where edges are possible causal relationships
- events are classified in two level of abstraction
  - *local* if detection or reaction can be achieved on a local base
  - *global* if there is the need of central (global) management
- to DSs are associated reactions in terms of configuration changes (Reaction Scenario – RS)
- DSs (and RSs) are computed by the DRG (Detection and reaction generation) tool



PDS paradigm

- prognosis is *the prediction of how a patient's disease will progress*
  - in our framework, an event that is identified as important to react/prevent
- diagnosis is *the analysis of the underlying physiological/biochemical cause(s)*
  - in our framework, non-observable events with the employed sensor infrastructure
- symptoms are *the manifestation of a disease*
  - in our framework, observable events

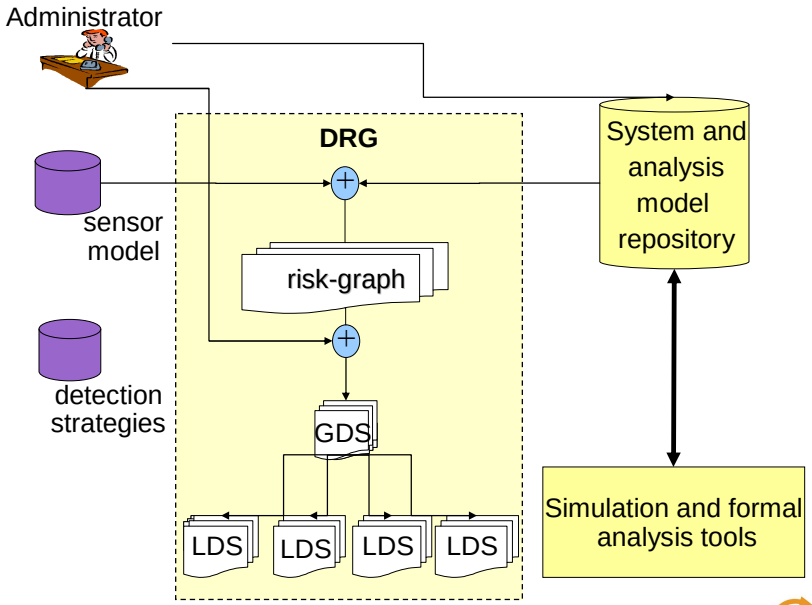
event types

- four macro classes of events
  - **security:** violation of security and dependability properties
    - e.g. availability, intregrity, confidentiality violation
  - **exploitation:** attempted or successful vulnerability exploitation or security policy violation
    - e.g. vulnerability exploitation, vulnerability exploitation attempt
  - **QoS:** performance parameters, focused on service requests
    - e.g. service demand, service latency, service request loss
  - **resource consumption:** consumption of system resources
    - e.g. cpu consumption, bandwidth consumption, memory consumption
  - **configuration:** change of configuration (for reaction)

event parameters

- each event is characterized by
  - the affected system element
    - e.g. sdl, wscdl, ...
  - some numeric parameters, specific for the event
    - e.g. the threshold value trespassed, in terms of packets/s or bytes/s or requests/s
  - the frequency of the event
    - e.g. if it is periodic or transient (like a vulnerability exploitation) or permanent (like a broken component)
  - optionally, which sensors are able to detect it
    - e.g. a snort probe
  - optionally, external references to provide additional information
    - e.g. an IDS signature

DRG high level architecture



input models

- system models identifies service, resource, constraints, and defines Operative Configurations (OC)
- analysis models provides results of formal and simulation analysis on a specific OC
- sensor model describes the employed sensor system, and identifies
  - type of detectable events, where events are detectable, who is able to detect, and the level (local or global) of detectability
  - are organized in pattern referencing the other input models
- detection strategies identify critical events.
  - basically are patterns of events and action required for events (e.g. *prevent* the accident or *detect* the accident)

risk graph

- the risk graph is a graph with associated risk and detection parameters, particularly
  - if the event is detectable, and the level (local or global) of detectability
  - the risk associated to the event, which is extracted exploiting risk-analysis techniques (e.g. MAGERITv2)
    - from the asset value (input of the formal description)
    - from the threat degradation (input of the publicly available database sources, e.g. NVD, OSVDB)
    - from threat likelihood (input of expert knowledge/historic data/attacker models)

Global and local detection scenario (GDS and LDS)

- DS is a way to associate critical events (prognosis) to detectable signs (symptoms, useful to runtime detection of problems) and undetectable hypothesis (diagnosis, useful to make inference on possible critical events causes).
- the GDS is a set of PDSs, each PDS is a sub-graph with
  - just a single prognosis associated
  - from zero to N symptoms events, which describes in causal relationship the detectable events which lead to the prognosis
  - from zero to M diagnosis events, which describes in causal relationship undetectable events which lead to the prognosis
- from the global detection scenarios, local detection scenario is derived
  - PDSs which involves only local component are grouped in LDSs
  - conceptually the same for reaction scenario (RS), in case the reaction involves reconfiguration of local component only

detection scenario algorithm

- PDS are generated starting from a critical event (the prognosis) and then aggregating events that are in causal relation with them.
- the algorithm is customizable to select causal relationships up to
  - a parameterizable distance from the prognosis event
  - a parameterizable confidence in the edge selection
  - parameters are supposed to be automatically improved by data-mining techniques on past experience
- from the sensor model events are tagged as detectable or not
- the algorithm can exploit techniques studied in the fault tree analysis (single and multiple fault diagnosis)
  - e.g. to cut not possible/not likely diagnosis

Conclusion and future work

- we have presented a framework to manage the residual risk of a complex system. In this paper we focused on the detection part
- future point to investigate are
  - precise definition of risk indicators
    - to automatically evaluate the best configuration
    - to diagnose, with sufficient confidence, specific dependability problems
  - the structure of the reaction scenario, in patterns able to capture reconfiguration capabilities of components and components sub-set
  - a detailed sensor model
    - to capture capability of modern state-of-the-art sensor system, with emphasys in intrusion detection techniques.

Dependability Analysis of Information Systems with Hierarchical Reconfiguration of Services

Dariusz Caban and Wojciech Zamojski

Wroclaw University of Technology, Poland

Paper abstract

The paper presents an approach to dependability analysis of a class of networked information systems based on distributed services. The system consists of groups of servers that react autonomously to detected events, component failures and security incidents.

These local reactions affect the availability of the services, provided by the system, then restored by reconfiguration. The dependability analysis is based on the construction of a reconfiguration graph, which is transformed to a state-transition dependability model.

Wrocław University of Technology

Dependability analysis of information systems with hierarchical reconfiguration of services

D. Caban and W. Zamojski

Depend 2008 – Cap Esterel

System dependability

- Ability to execute correctly system functions (tasks, jobs)
  - in anticipated time, in assumed work conditions (environment),
  - in presence of threats, hardware failures or software and human faults
- Characteristics
  - Availability
    - readiness for correct service
  - Reliability / survivability
    - continuity of service
  - Safety
    - absence of catastrophic consequences on users and environment
  - Security
    - protection against unauthorized users
  - Confidentiality
    - absence of unauthorized disclosure of information
  - Integrity
    - absence of improper system state alterations
  - Maintainability
    - ability to undergo repairs and modifications

System model

- Infrastructure
  - Network nodes
  - Connections between nodes
  - Node resources
    - Computational (CPU time, memory, storage bandwidth, …)
  - Connection resources
    - Link/interface throughput
- Services
  - Business service components, technical services, …
    - Deployed on a single node
    - Computational requirements (consume node resources)
  - Communication requirements
    - Consume connection resources
- Configuration
  - Deployment of services on nodes
  - Permissible, if
    - Σ(service requirements on a node) ≤ node resources
    - Σ(service comm. requirements between nodes) ≤ connection resources

System reconfiguration

- Reconfiguration
  - Change of system configuration (deployment)
  - To improve dependability (react to an incident)
- Incidents
  - Hardware failures
    - Handled by infrastructure (dynamic routing, load balancing)
    - Affect the available resources
  - Exploitation of software faults
    - Unintentional, exploitation of vulnerabilities, undirected attacks
  - Drainage of resources attacks
    - Denial of service, especially DDOS
- Redeployment of services
  - As a temporary remedy to incidents
  - To preserve continuity of service
  - Until the affected nodes are restored to proper functioning

Hierarchical reconfiguration

- Hierarchical reconfiguration
  - To reduce the complexity
    - The space of permissible configurations
    - The number of possible reconfigurations
  - Nodes are grouped
    - Each group is independently managed
    - Local management can:
      - Handle an incident (preserve continuity of service)
      - Cut-off a group of nodes to contain the consequences of the failure
    - Higher level of management
      - Redeployment of services from the affected lower level group
- Dependability analysis assumptions
  - Incidents are localized to a single group of nodes
  - Groups fail independently (i.e. incidents are independent)
- Hierarchical decomposition
  - Driven by management requirements
  - Well consides with the analysis requirements

Wrocław University of Technology

DESEREC Project

- EU 6th Framework IP FP6/2002-2006
  - Presented system model inspired by the proposed system architecture
- DESEREC architecture
  - Molecules
    - Locally managed groups of nodes
    - Running specific services
    - Managed by autonomous DLocalAgents
      - Realize triggered local reactions
      - Reconfigure services locally
      - Cut-off the managed nodes (delegate reaction to DGlobalAgent)
      - Realize configuration changes required by DGlobalAgent
  - Global reconfigurations
    - Managed by DGlobalAgent (with administrator in the loop)
    - Using pre-analyzed set of global configurations

DESEREC
Dependability & Security by Enhanced Reconfigurability

Wrocław University of Technology

System availability

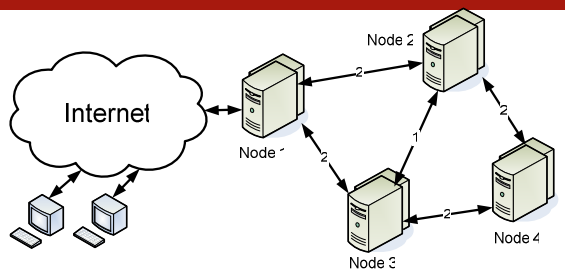
- A measure of system dependability
- Definition
  - Probability *A(t)* that a service is correctly realized at a time instance *t*
- Can be evaluated as:

$$A = \lim_{t \to \infty} \frac{t_{up}}{t}$$

$$A = \lim_{n \to \infty} \frac{n_{up}}{n}$$

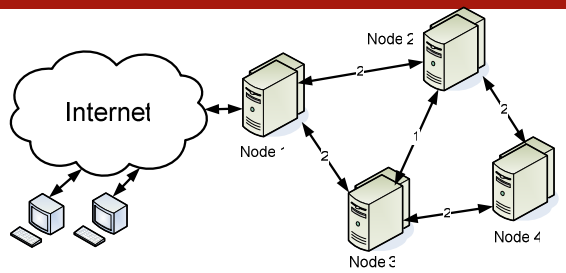
Wrocław University of Technology

Case study



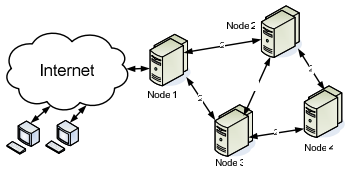
Services
- *w*1 that acts as the entry point for client interactions
- *w*2, *w*3 and *w*4 – applications running on nodes 2 or 3
- *w*5 – the database used by the applications on node 4

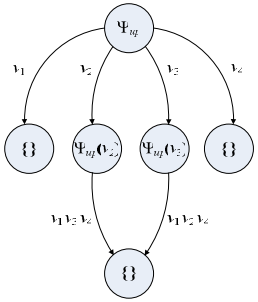
Case study - permissible configurations



$$\Psi_{up} = \begin{Bmatrix} [\{w_1\},\{w_2,w_3,w_4\},\{\},\{w_5\}], \\ [\{w_1\},\{w_2,w_3\},\{w_4\},\{w_5\}], \\ [\{w_1\},\{w_3,w_4\},\{w_2\},\{w_5\}], \\ [\{w_1\},\{\},\{w_2,w_3,w_4\},\{w_5\}], \\ [\{w_1\},\{w_2\},\{w_3,w_4\},\{w_5\}], \\ [\{w_1\},\{w_4\},\{w_2,w_3\},\{w_5\}] \end{Bmatrix}$$

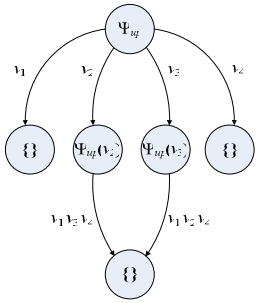
Case study - reconfiguration graph

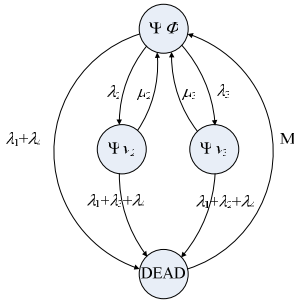




$$\Psi_{up}(v_2) = \{[\{w_1\},\{\},\{w_2,w_3,w_4\},\{w_5\}]\},$$
$$\Psi_{up}(v_3) = \{[\{w_1\},\{w_2,w_3,w_4\},\{\},\{w_5\}]\}.$$

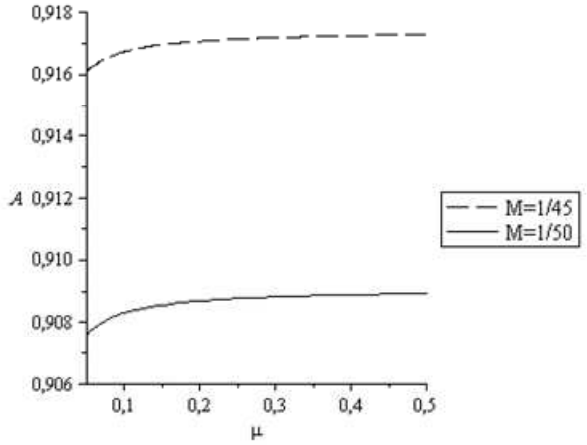
Case study - ST model





$$A = \frac{(5\lambda+\mu)\mathrm{M}}{(3\lambda+\mu)\mathrm{M}+2\lambda(6\lambda+\mu+\mathrm{M})}$$

Case study - results

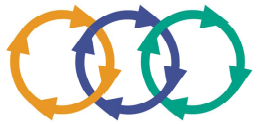


Conclusions

- Proposed dependability analysis
  - extendable
  - scales well in case of large, complex systems
  - commercial and open-source software can be used once the S-T graph is constructed
  - industry grade analysis tools may be achieved with very little effort
- Limited to independent incidents handling assumption
- Multiple reconfiguration policies ensure similar level of dependability improvement
  - Vertices of the reconfiguration graph are labeled with collections of admissible configurations
  -
- Proposed availability metric is not meaningful for systems operating in degraded mode
  - Require a measure of degradation (efficiency of operation)
  - Can be achieved by combining ST model analysis with system simulation

Wrocław University of Technology

Thank you for your attention



DESEREC

Dependability & Security by Enhanced Reconfigurability

Hierarchical Approach to Dependability Analysis of Information Systems by Modeling and Simulation

Katarzyna Michalska and Tomasz Walkowiak

Wroclaw University of Technology, Poland

Paper abstract

The paper presents a method of analyzing functionality aspects of complex information systems. The system analysis is based on integrating different computer system models, the networked systems and applications model, clients' behaviour model and collaboration between services in the analyzed system, into one coherent functionality based view suitable for system simulation. Modelling process incorporates hierarchical approach, i.e. the system is decomposed into groups of hosts and therefore groups of services, called molecules. The model is described in design by authors XML Domain Modelling Language (XDML). The integrated model is automatically transformed into an input file for modified SSFNet simulator. Additional information on input models could be read from other models or the user could play with all parameters of the XDML model. The paper presents developed software which implements described methodology and results of analysis for an exemplar information system.

DEPEND 2008
25-29 August, Cap Esterel, France

Hierarchical approach to dependability analysis of information systems by modeling and simulation

Katarzyna Michalska, **Tomasz Walkowiak**

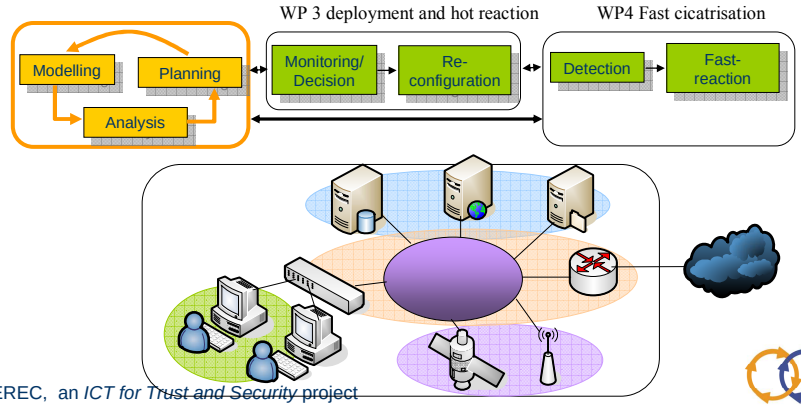
Wroclaw University of Technology

DESEREC
Dependability & Security by Enhanced Reconfigurability

Information Society Technologies

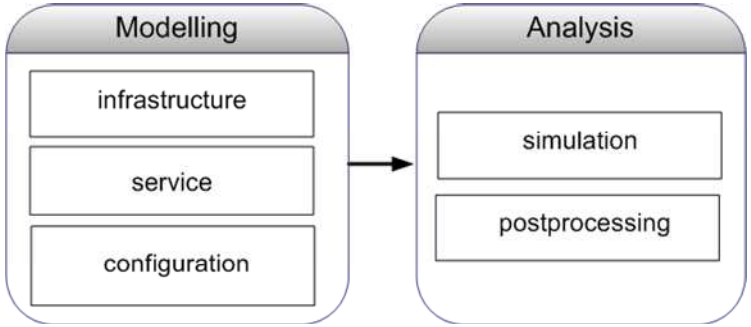
DESEREC project - www.deserec.eu

- **WP2 Operational planning**
- Deals with models of information systems, their intended behaviour (policies), and risks (foreseen faults and attacks also unforeseen failures)
- Checks the expected behaviour of the system when configured in a specific way and it is subject to specific inputs and faults
- To create the needed tools to manage information and scenarios needed to configure the target system and react to faults and attacks



Overview

- Modern CIS usually consist of a large number of hosts, with different software installed it is hard for humans to understand the system behaviour
- Proposed solution:
  - modelling and analysis
    - hierarchical approach – division of the system into smaller subset - molecules
    - functional approach – focusing on business service realized by a system

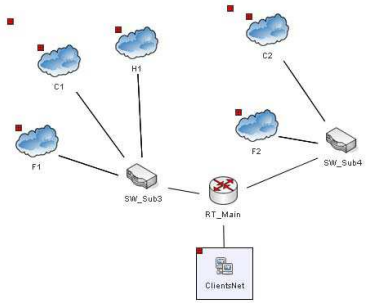


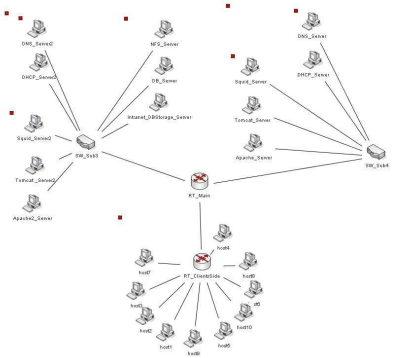
Business service

- Business service (BS)
  - set of telecommunication services based on business logic, that can be loaded and repeatedly used for concrete business handling process (i.e. ticketing service, banking, VoIP, etc)
  - composed of service components and tasks, that are used to provide definite service in accordance with business logic for this process
- Service component (SC)
  - a technical service located on defined host (server) that determined service behaviour, possibilities and requirements (i.e. authentication, data base service, web service, etc.)
  - one host can have more than one service component
- Task (T)
  - the lowest level of business service description
  - It can be seen as a request and response form one service component to another (i.e. lookingNews, getMainPage, logIn)

Molecule

- M =< F,I,C >
  - **F**unctionality (set of business services), **I**nfrastructure (set of hosts and links connecting to hosts), **C**onfiguration (allocation of service components)
  - The decomposition is arbitrary done by grouping hosts in a way that they are realizing related service component with restrictions:
    - Each host belongs to not more then one molecule
    - For any subnet to which any host of a given molecule belongs to there could be only two cases:
      - (a) all subnet elements belongs to the molecule
      - (b) all molecule elements belongs to this subnet



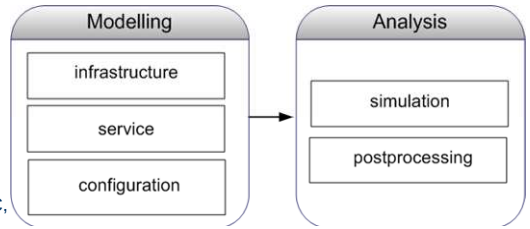


Modelling of information systems

- There exist many organizations involved in creation and standardization of languages used in modelling (OASIS, BPMI, OMG, W3C)
- They realize their aims throu various methods for system description concerning aspect like:
  - network topology, ontology, choreography, etc
- We want to **model** services behaviour and the underlying hardware structure (computers and networks) and then **simulate it**
  - SDL, WS-CDL, ALM, OCL
- System Description Language (**SDL**):
  - defined by the POSITIF Project to provide formal description of networked ICT systems:
    - system topology - physical (hardware) and logical (software) infrastructure;
    - the network configuration and black-box functionality of each node;
    - the security functionality of each node.
- Web Services Choreography Description Language (**WS-CDL**) by W3C:
  - used to describe behaviour of the services inside ICT network
  - focuses on describing the business protocol among different roles
  - static part - description of static relation i.e. role types, communication types, channel types, parameters types;
  - dynamic part - description of cooperation between communication actors and their time correlations.

Modelling of information systems

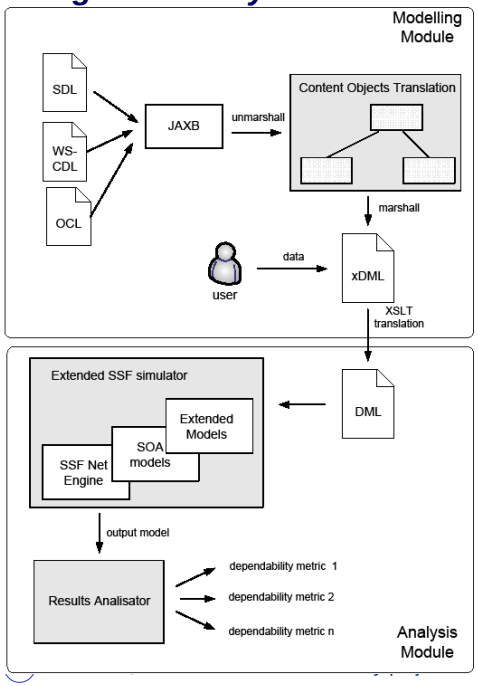
- Operational Configuration Language (**OCL**) -
  - DESEREC project by POLITO
  - system configuration
  - an indirect model (mapping pairs, XML)
  - maps WS-CDL components into SDL elements
- Analysis Load Model (**ALM**)
  - DESEREC project by PWR
  - user description for analysis proposes
  - SDL language give us only host description of a client,
  - client model for simulation:
    - clients number
    - size of their request,
    - description of their behaviour in case of using business service, etc.

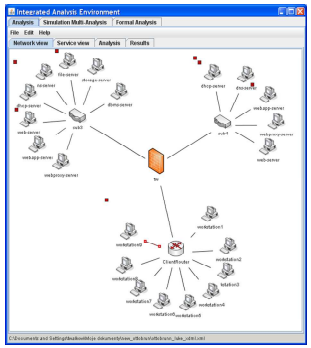


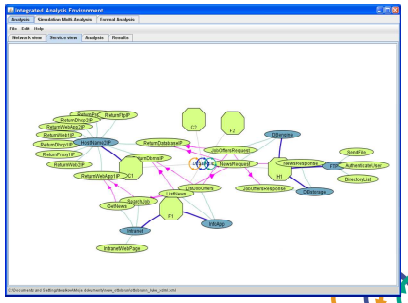
Simulation

- Simulation aim:
  - mimicking the behavior of real life processes
  - by a message interchange between business level services provided by the system
  - on the top of the protocol stack of a network simulator.
- Modified SSFNet simulator:
  - support for traffic generation (models of user behavior with randomness),
  - simulation of business level services,
  - implementation of resource consumption for requests processing,
  - errors (for dependability analysis) in different levels:
    - link failures, network adapter failures, software component failures;
  - Monte-Carlo approach
  - results of simulation (recorded in XML based output file)
- SSFNet uses the Domain Modeling Language (DML) as input
  - text format
  - extended DML (modification of simulator)
  - XDML - XML Domain Modeling Language

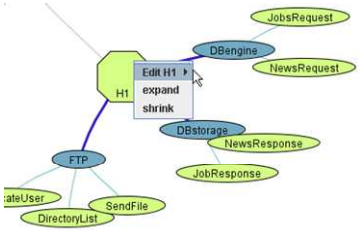
Integrated Analysis Environment framework

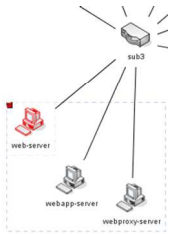


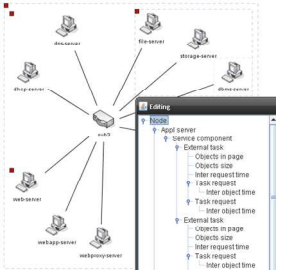


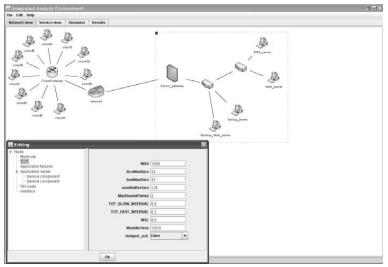


Integrated Analysis Environment framework









Dependability metrics

- The simulation allows observing different parameters of the analyzed information system
- Dependability analysis requires a method of calculating metrics based on observed parameters of the simulated system
- For the needs of this research we provided two metrics of information system dependability
  - availability (service, molecule)
  - response time (task)
- The molecule availability which is defined as a weighted average of availabilities of service components belonging to a given molecule
- The weighting factors are normalized (sums to one) and are selected by a system operator based on importance of a given service.
- Due to a randomness of a user behaviour the calculation of these metrics was done based on Monte-Carlo approach by repeating simulation of the same system *N* times over analyzed period *T*.

Dependability metrics

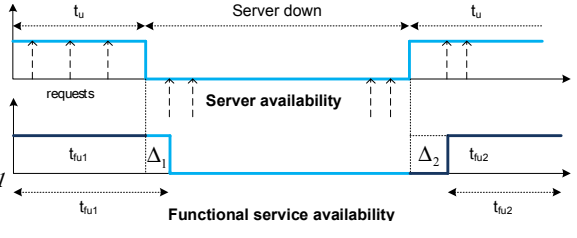
- ***SERVICE AVAILABILITY***
  - Observeability issue
    - either based on observing internal service diagnostics (technical service)
    - or based on ability to service requests (technical / business service)

$$A_s = \frac{\sum t_u}{T}$$

$$A_{fs} = \frac{\sum t_{fu}}{T}$$

$$t_{fu1} = t_{u1} + \Delta_1$$

$$t_{fu2} = t_{u2} - \Delta_2$$



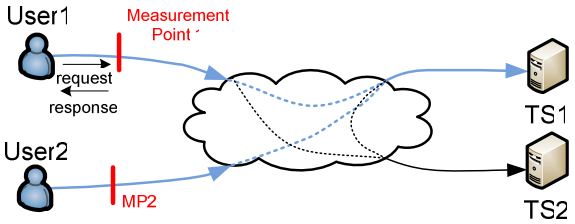
  - Functional approach – SERVICE is **UP** only when is able to deal with service requests

Dependability metrics

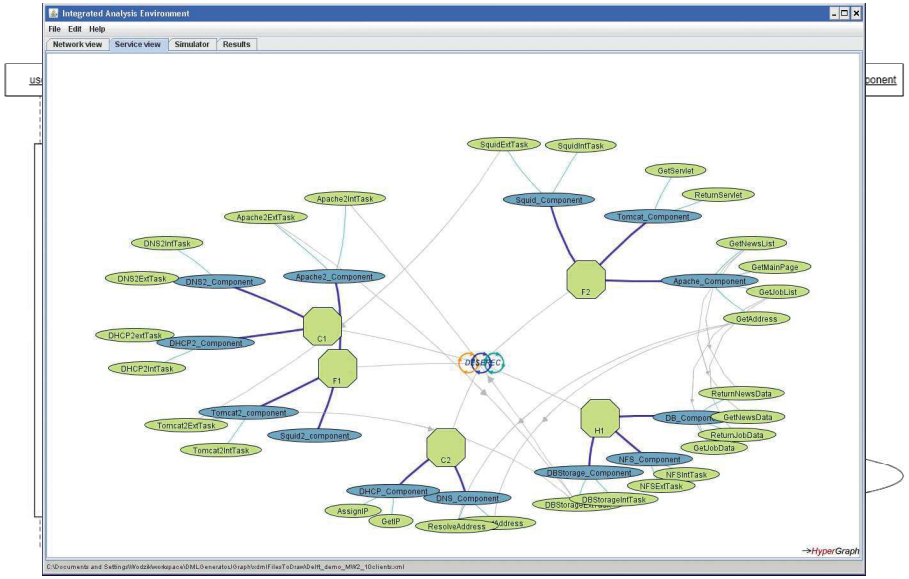
- ***SERVICE AVAILABILITY***
  - Based on reactions to Service queries
    - NORMAL response >> SERVICE **UP**
    - TIMEOUT >> SERVICE **DOWN**
    - ERROR (on page) >> SERVICE either **UP** or **DOWN** (depending on definition)
  - Noticed limitations
    - SERVICE is up, but temporary timeout occurs – it renders SERVICE unavailable (down)
    - What state if no queries present ? - Last state is the current state (it may lead to erroneous conclusions)

Dependability metrics

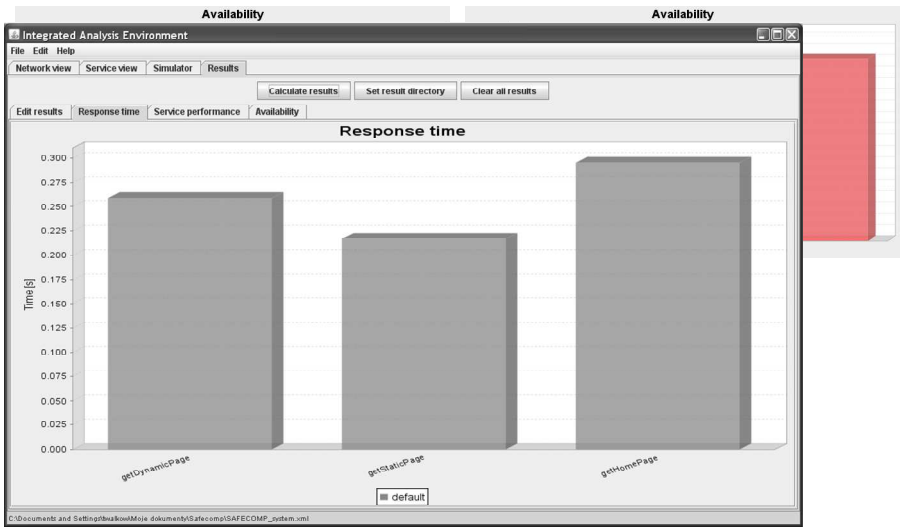
- ***TASK RESPONSE TIME***
  - **Task** level
    - Response of a specific task
    - For various groups of users (not necessarily the same categories as BS)
    - Useful for ***DETECTION / REACTION SCENARIOS***



Case study



Case study - results



Summary

- The proposed method of analyzing service based information systems shows its effectiveness
- Implemented Integrated Analysis Environment with SSFNet simulator gives a tool that allow to model and analyze different service based information system.
- A usage of standard languages for model description (WS-CDL,SDL) and proposed configuration languages (ALM, OCL) with respect to graphical interface (IAE) integrating whole modelling and analysis process makes the tool useful for information system designers and administrators.

- The proposed work is still in progress:
  - Model failures by statistical process (by putting changes in XDML and SSFNet simulator) what could allow to compare dierent configuration in more realistic way taken into consideration for example economical aspects
  - More metrics
  - More case studies

Towards the Definition of a Web Service Based Management Framework

Juan Manuel Marín Pérez, Jorge Bernal Bernabé, Daniel J. Martínez Manzano, Gregorio Martínez Pérez, and Antonio F. Gómez Skarmeta

University of Murcia, Spain

Paper abstract

The definition of advanced and distributed management frameworks is one of the key research issues that still need to be addressed in the network and service management research field. As part of this effort, the use of web service technologies provides some advanced features such as improved interoperability and flexibility. This paper describes the main requirements and key components of a web service-based management framework, how it works (at the level of the designed workflow) and how it has been mapped to a real management framework.

The First International Workshop on Dependability and Security
in Complex and Critical Information Systems
DEPEND 2008
August 25-31, 2008 - Cap Esterel, France

Towards the Definition of a Web Service Based Management Framework

Juan Manuel Marín Pérez, Jorge Bernal Bernabé, Daniel J. Martínez Manzano,
Gregorio Martínez Pérez, Antonio F. Gómez Skarmeta

Department of Communications and
Information Engineering

University of Murcia (Spain)

Outline

- Introduction
- Framework description
- Framework workflow
- Conclusion and Future work

Introduction

- The definition of a distributed management framework is a key research issue that still needs to be addressed
- Some efforts done in this sense: WBEM, WSDM, …
- However, these frameworks lack of some advanced mechanisms like:
  - conflict detection
  - monitoring
  - reconfiguration capabilities…
- The goal is to define a generic management framework that resolves these limitations

Introduction

- Requirements of an IT system management framework:
  - Well defined model
  - Flexibility and definition of abstractions to manage a wide variety of device types
  - Interoperability with other architectures
  - Conflict Detection
  - Integrating the Business and Networking worlds
  - Scalability
  - Reconfigurability

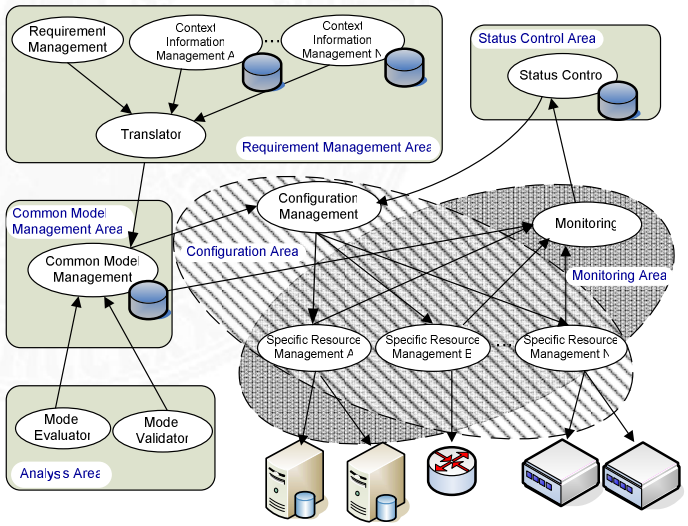
Outline

- Introduction
- Framework description
- Framework workflow
- Conclusion and Future work

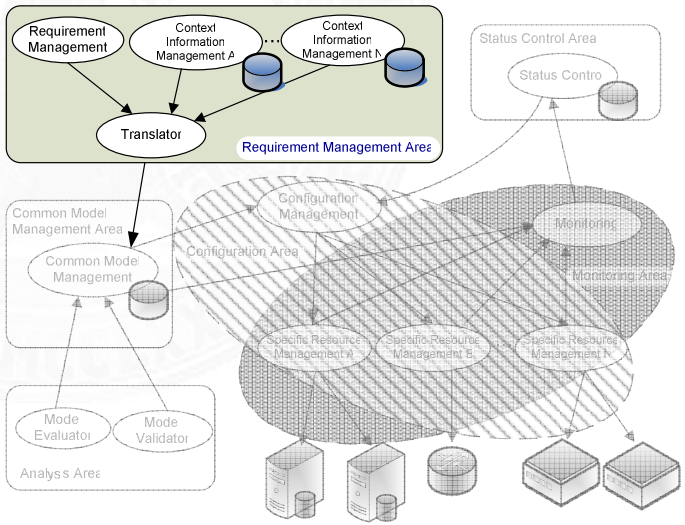
Framework description

- Framework areas:
  - Requirement Management Area
  - Common Model Management Area
  - Analysis Area
  - Configuration Area
  - Monitoring Area
  - Status Control Area

Framework description



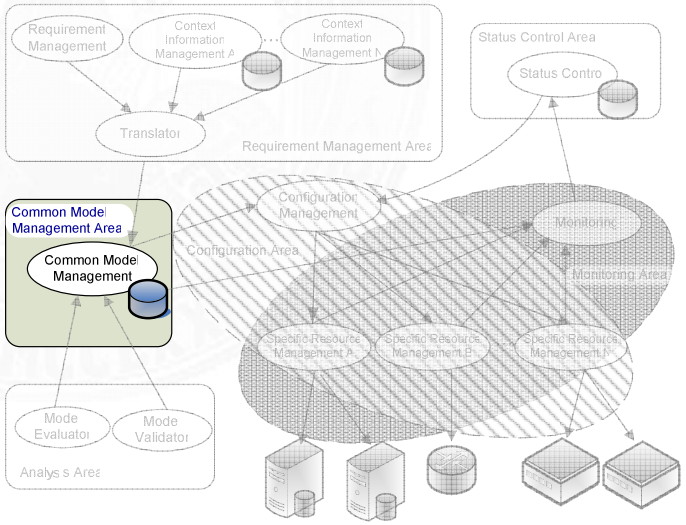
Framework description
Requirement Management Area



Framework description
Requirement management Area

- Manages the requirements defined by the administrator making use of a high level policy language (usually with a graphical console)
- Provides methods to manage administrators' requirements
- A set of *Context Information Management* modules are in charge of the context Information
- The *Translator* takes the requirements and the context information specified in a high level language as input and generates the Common Model

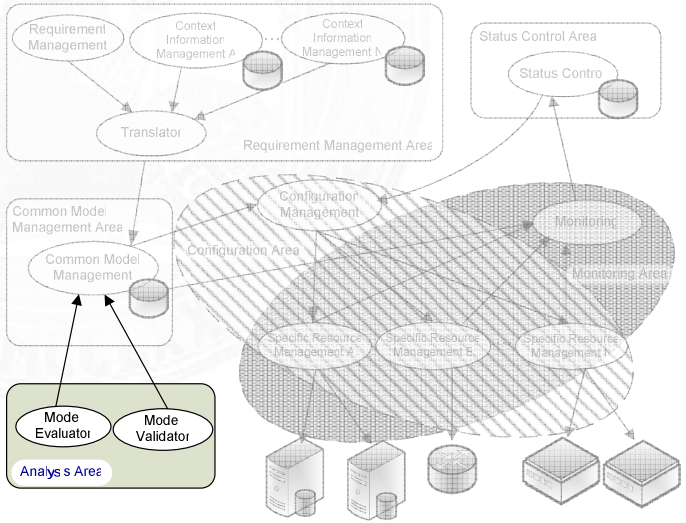
Framework description
Common Model Management Area



Framework description
Common Model Management Area

- Once the model is generated by the *Translator*, it is sent to this area to be stored and managed
- This area uses a standard common model such as CIM (Common Information Model)
- The common model is used by the different framework components in order to:
  - generate final configurations
  - monitor the requirement defined by the administrator
  - …

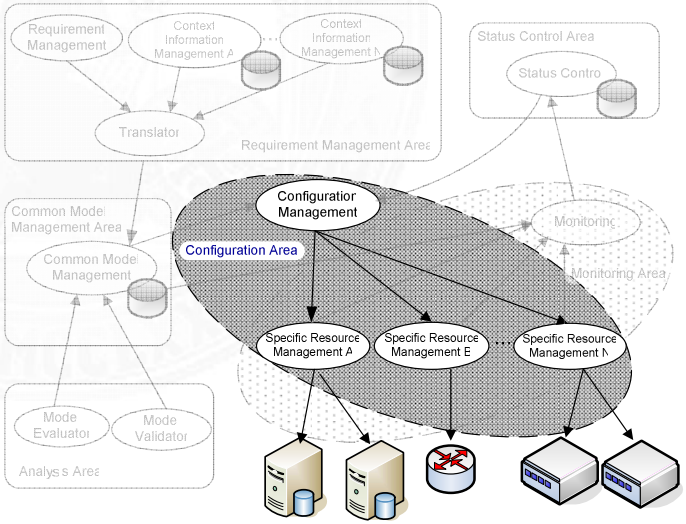
Framework description
Analysis Area



Framework description
Analysis Area

- This area takes the generated common model and validates the requirements specified on it
  - Checks if there is some inconsistence
  - Takes also into account reconfiguration policies
- The *Model Evaluator* gives some metrics measuring the quality of the model
  - This measurement deals with different model aspects like the measurement of the security level

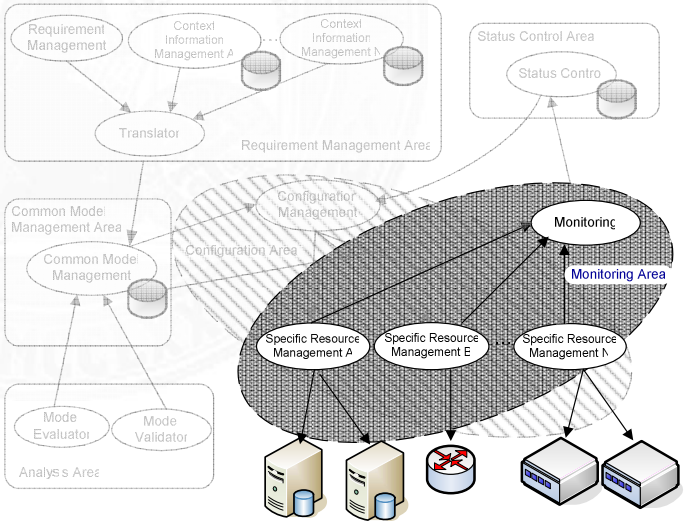
Framework description
Configuration Area



Framework description
Configuration Area

- In charge of producing and enforcing the specific configurations
- Uses the information defined in the Common model for this task
- There is an *Specific Resource Management* module for each management resource type
  - Each module implements a Web interface
  - The *Configuration Management* uses this interface to interact with the modules in order to generate and enforce the configuration in the devices

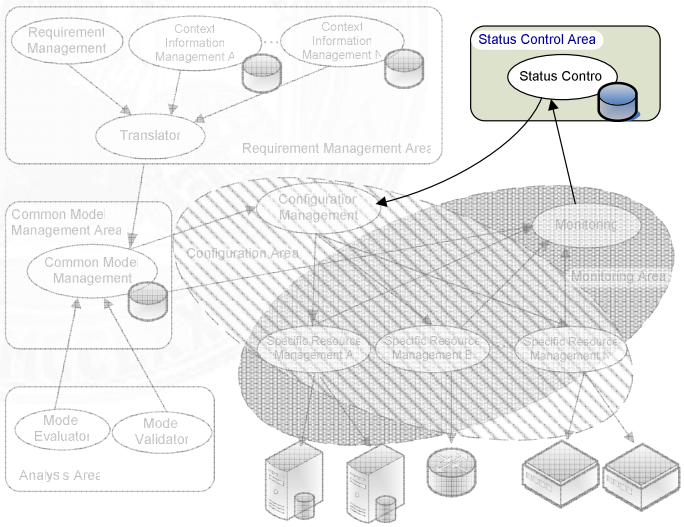
Framework description
Monitoring Area



Framework description
Monitoring Area

- Checking the system to detect any behaviour which may violate the requirements defined by the administrator
- Contains a vulnerability DB
- Recollects the events reported by the sensors
- More than a simple IDS
  - Can access to the common model which has the knowledge of the original requirements
  - Uses this information to compare the monitored data with the administrator desired behaviour

Framework description
Status Control Area



Framework description
Status control area

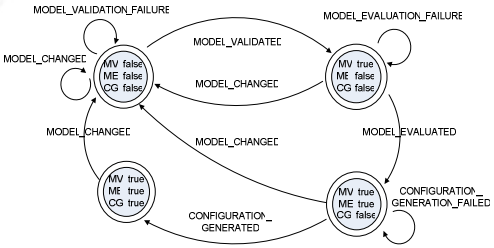
- Controls de system status processing the different alerts which may arise from the monitoring system
- Applies the proper configuration according to the reconfiguration policy
  - Administrador previously defines different requirement sets for different system situations
- Example: increasing or reducing the system's *Security Level*
  - Policy sets are defined for different security levels
  - Attack or intrusion detected by the monitoring module → Apply a higher security level policy set

Outline

- Introduction
- Framework description
- Framework workflow
- Conclusion and Future work

Framework workflow

- A workflow has been defined to allow a proper coordination between different components an achieve a whole management system
- Web Service based Eventing System
  - Standards: WS-Eventing, WS-Notification…
  - Used by the different components to communicate changes in their status asynchronously
- Statechart with the different statuses of the framework and the events that produce the transactions between them



Outline

- Introduction
- Framework description
- Framework workflow
- Conclusion and Future work

Conclusions

- Usage of a Common model and Web Service technologies provide quite interoperability
- The refinement done by the *Translator* allows the administrator to use abstract high level language
- The monitoring system with the status control gives the framework reconfiguration capabilities
- The evaluation and validation of the model allows preventing conflicts or bad quality configurations
- The framework described in this paper has been implemented in the scope of the POSITIF Project

Future Work

- Agent based distributed system → to cooperate and improve the scalability of the framework in large systems
- Semantic approach is being taken trying to apply the concept of ontology and semantic web in the common model
  - validate the model
  - resolve conflicts

Acknowledgment

- POSITIF Project: EU-IST Project: IST-2002-002314
  - *"Policy-based Security Tools and Framework"*
  - Website → http://www.positif.org/
- DESEREC Project: EU-IST Project: IST-2004-026600
  - *"DEpendability and Security by Enhanced REConfigurability"*
  - Website → http://www.deserec.eu

The First International Workshop on Dependability and Security
in Complex and Critical Information Systems
DEPEND 2008
August 25-31, 2008 - Cap Esterel, France

Towards the Definition of a Web Service Based Management Framework

Thank you for your attention!!

Department of Communications and Information Engineering

University of Murcia (Spain)

Dependability and Security Metrics in Controlling Infrastructure

Imre Kocsis, György Csertán, Péter László Pásztor, and András Pataricza

Budapest University of Technology and Economics, Hungary

Paper abstract

Large, distributed IT infrastructures providing business-critical services have to protect themselves against internal and external threats and adapt to changing environmental parameters, as workload. Most widely applied, structural resilience mechanisms use some form of local static redundancy deployed to each critical resource for failover. However, recently both large-scale interconnected distributed systems and virtualization enable on-line structural reconfiguration exploiting a globally managed spare capacity as on-demand failover resource.

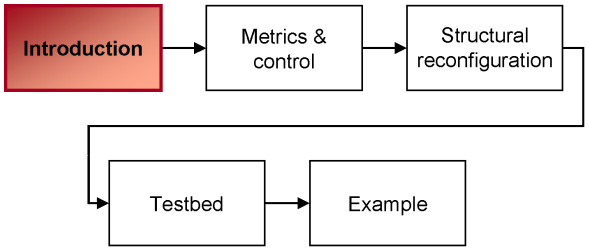
In this paper, we present system and service resilience as a control problem and briefly describe how classes of the widely used, but vague notion of 'IT metrics' map to the concepts of generic control with a special emphasis on the control aspects of structural reconfiguration as a generic resilience mechanism. Most importantly, we introduce some initial metrics that aim at measuring the self-healing capability of systems employing structural reconfiguration.

Dependability and Security Metrics in Controlling Infrastructure

I. Kocsis, Gy. Csertán, L.P. Pásztor, A. Pataricza

ikocsis,csertan,pasztor,pataric@mit.bme.hu

Talk Overview

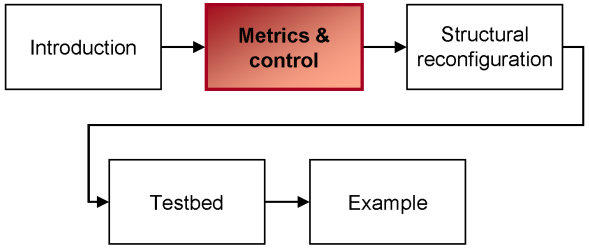


Introduction

- Dependability & security in heterogenous, large IT infrastructures
  - more and more important
  - hard to solve without new methods
- DESEREC
  - EU FP6 IST project
  - enhanced reconfiguration
  - molecule concept
- WG2
  - define metrics to
    - characterize
    - evaluate
    - compare

    the IT ifrastructure
  - support the reconfiguration
  - system management
    - control problem

Talk Overview

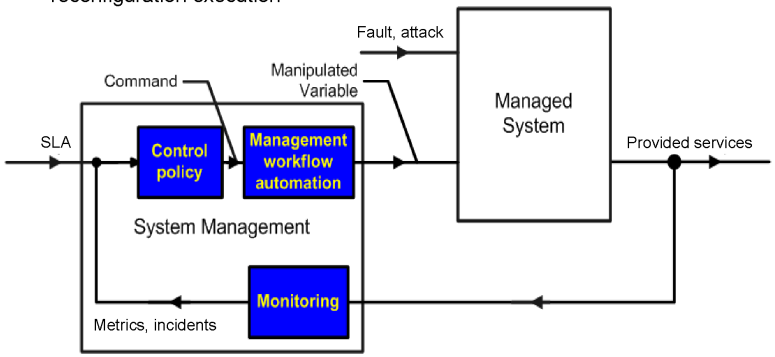


Metrics for Control

- Traditional system management
  - system monitoring
    - well developed sensors & actuators
      - basic hardware components (CPU, disc, NIC, etc.)
      - basic software components (DB, web server, OS, etc.)
    - limited intelligence for control
    - missing points for extension
  - control
    - heuristic approach
    - simple set of reactions (based on low-level sensor values)
- Metric driven service management
  - metrics help to measure and compare in order to
    - assure quality and control the system
  - composability (system & service metrics)
  - goal-based management (SLA as the goal)
  - communication (within the organization between IT and business)

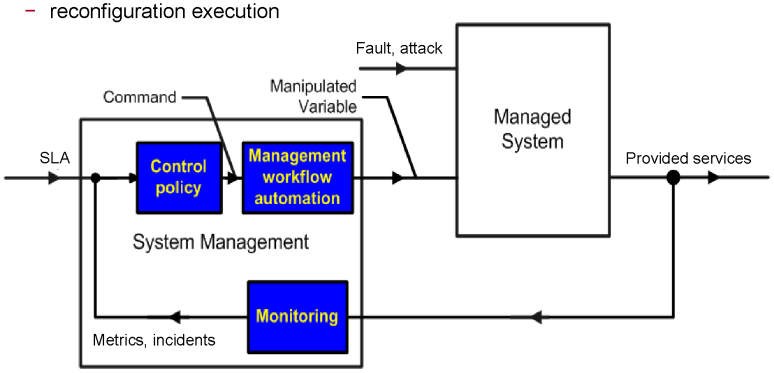
Control loop

- Managed system
- Managing system (system management)
  - monitoring
  - control logic
  - reconfiguration execution



Control loop

- Managed system
- Managing system (system management)
  - monitoring
  - control logic
  - reconfiguration execution

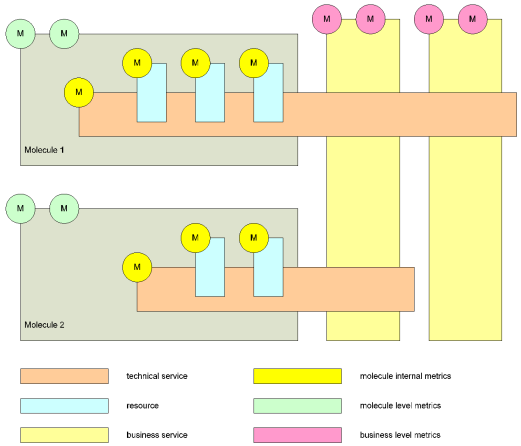


DESEREC metrics

- Metrics identified by the GQM approach
  - method
    - goal (goal (e.g. provide on-line impression for the user)
    - question (e.g. how long a user has to wait for a reaction from the system)
    - metric (e.g. what is the current response time of the service)
  - advantage
    - good for large systems
  - disadvantage
    - it has to be executed for each system case by case
- Three levels for metrics
  - business service level
    - composed
    - SAP measured
  - molecule level
    - molecule type agnostic metrics
  - resource level
    - molecule type dependent metrics

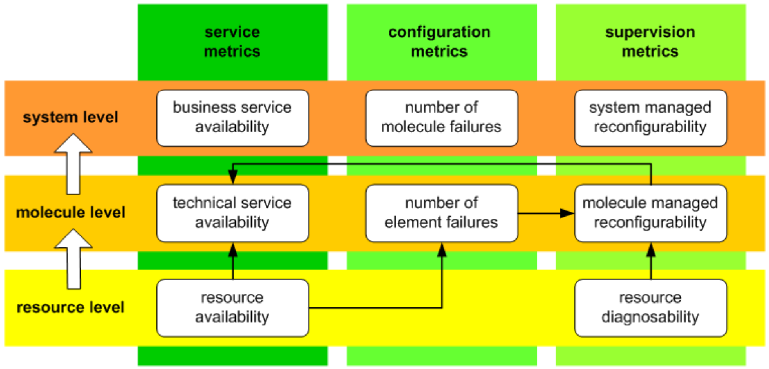
DESEREC metrics cntd.

- Three levels for metrics
  - business service level
    - composed
    - SAP measured
  - molecule level
    - molecule type agnostic metrics
  - resource level
    - molecule type dependent metrics

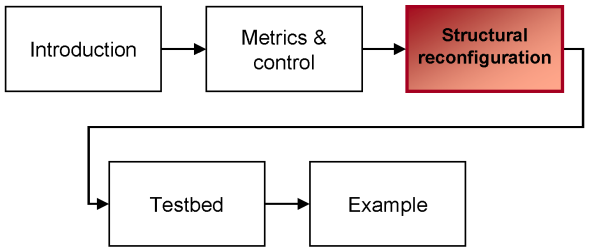


DESEREC metrics cntd.

- Three types of metrics
  - QoS (performance, dependability, security)
  - configuration (healthiness, abilities)
  - supervision (reconfiguration)



Talk Overview

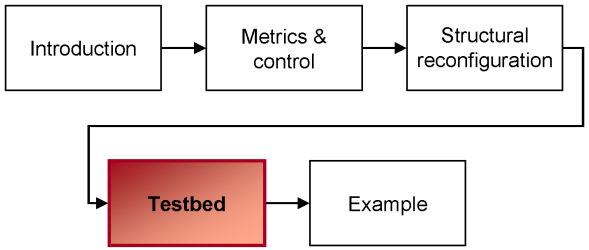


Structural Reconfiguration

„Reallocation of computation, storage or communication that is not only a local, parametric reconfiguration of a system component, but one that changes the service delivery structure and the dependencies between the components in runtime."

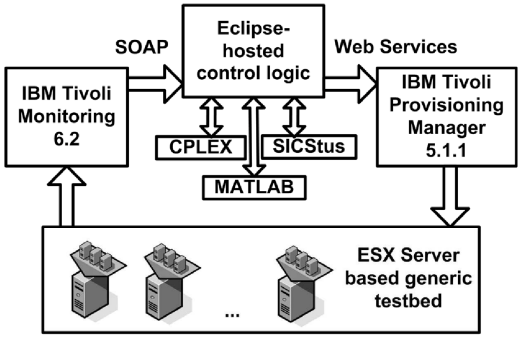
- Widely used methods in industry
  - server node failover
  - hot/cold spares
  - workload driven task migration (load balancing)
  - clustering
  - rerouting of communication
- Academic research
  - systematic handling of reconfiguration for IT system resilience
  - configuration metamodeling
  - reconfiguration description
  - metrics for reconfigurability
    - reconfiguration count of a configuration
    - one step reconfigurability
    - min./max. system reconfigurability

Talk Overview



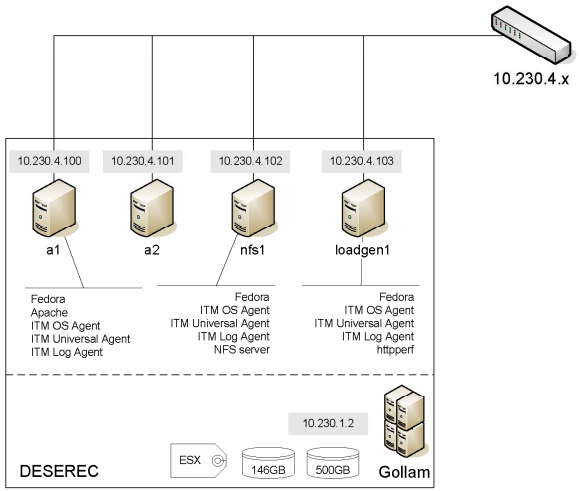
Testbed

- Set up at BUTE DMIS
- For control development
  - metrics validation
  - design and test various control solutions
  - validate reconfiguration workflows
- MAPE-K pattern
  - defined by IBM
  - control loop
    - monitoring
    - analizing
    - planning
    - execution



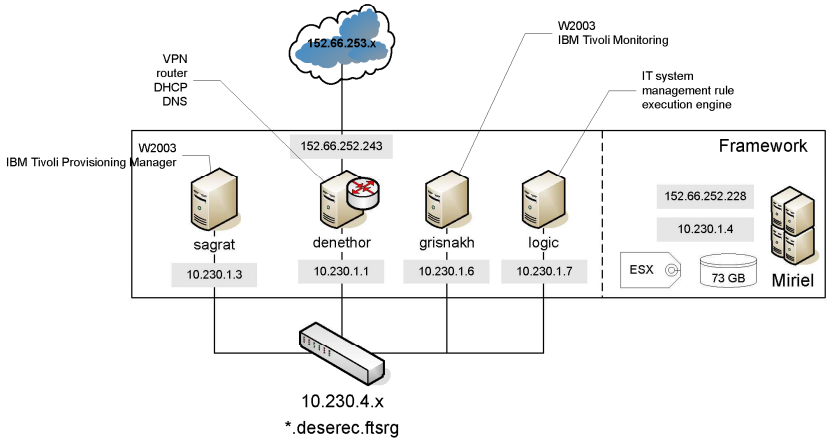
Testbed topology I.

- Managed system



Testbed topology II.

- Managing system

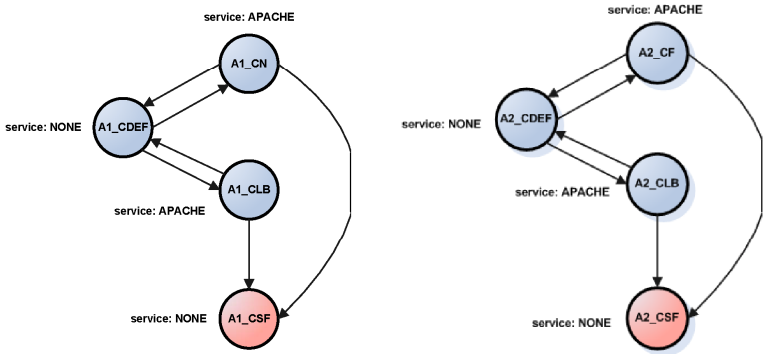


Testbed components

- Managed system
  - Apache1
  - Apache2
  - Load balancer
  - NFS server
  - Name server
  - Firewall
- Managing system (system management)
  - Monitoring
    - IBM Tivoli Monitoring 6.2
  - Control logic
    - Java / simple SOAP + Web Services
  - Reconfiguration execution
    - IBM Tivoli Provisioning Manager 5.1.1
  - Load generator
    - WGET
    - httperf

Metrics, services and system status

- Metrics
  - Low-level metrics: Apache availability (**LL11**)
- Technical service: static web service
  - two apache instances with diagnostic status
- System configuration status

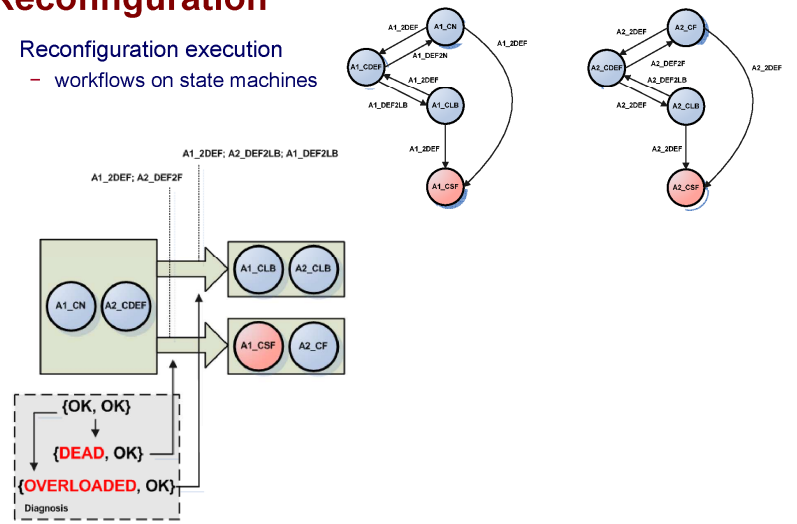


Testbed scenarios

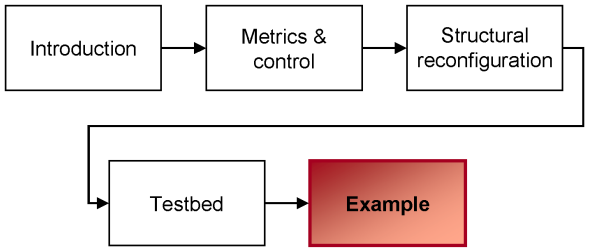
- Metrics
  - Low-level metrics: Apache availability (LL11)
- Scenarios
  - A1 NORMAL
    - nothing to do ☺
  - A1 DEAD
    - switch to A2
  - A1 OVERLOADED
    - start-up the load balancer

Reconfiguration

- Reconfiguration execution
  - workflows on state machines

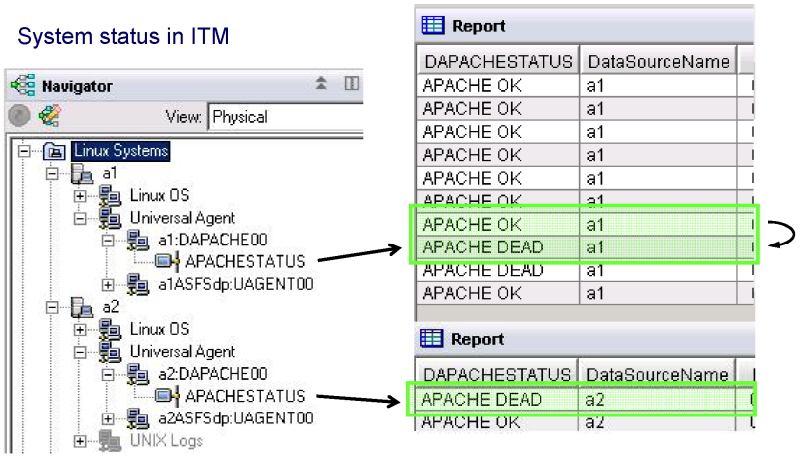


Talk Overview



Scenario No.2 – I.

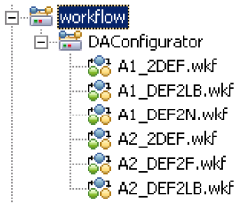
- System status in ITM



Scenario No.2 – II.

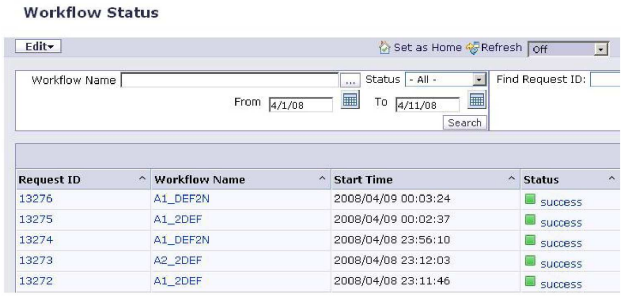
- Inventory in TPM

- Workflow in TPM



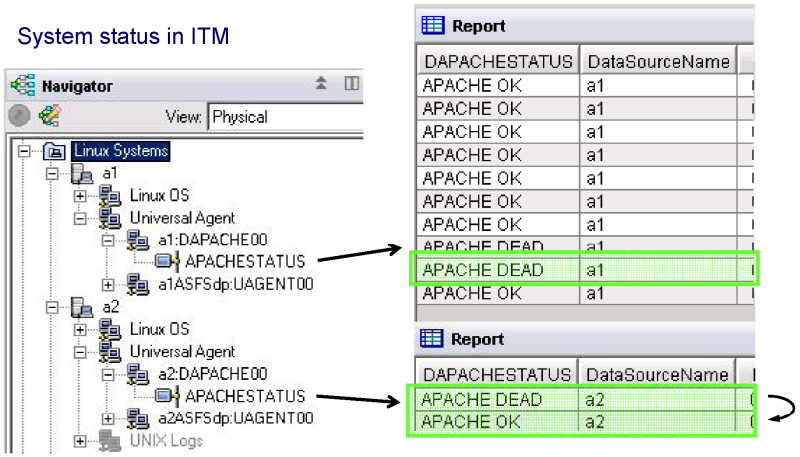
Scenario No.2 – III.

- Workflow status in TPM



Scenario No.2 – IV.

- System status in ITM



Advanced Rule-Based Techniques in Mission Critical Systems

Maite Avelino and Francisco de la Torre

Soluciones Globales Internet, Spain

Paper abstract

This paper deals with the current techniques most efficient for the detection of incidents in Critical Systems. Here it is explained how detection is implemented through rules and algorithms and how correlation can be done in the most powerful way.

Advanced Rule-Based Techniques in Mission Critical Systems

Maite Avelino, GMV Soluciones Globales Internet S.A.

DEPEND III 2008
August 26, 2008, Nice

DESEREC
Dependable Security by Enhanced Reconfigurability

Information Society Technologies

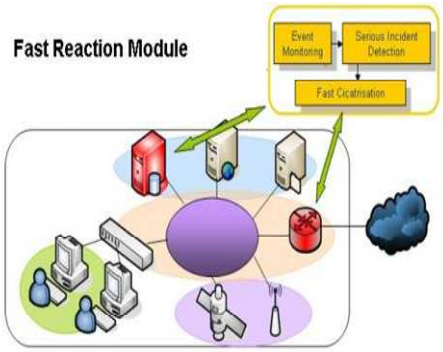
Overview of the Paper

1. Context in DESEREC project.
2. From raw events to correlated incidents.
3. The RETE algorithm.
4. Drools and the rule-based implementation.
5. Conclusions and future work

Context in DESEREC project

Serious Incident Detection

- The Fast Reaction architecture in DESEREC project provides response to Serious Incidents within seconds, so Detection should be done in Real Time.
- It is necessary to continously monitor Events in the target Systems. Trails of Real Time information such as Syslog are required.

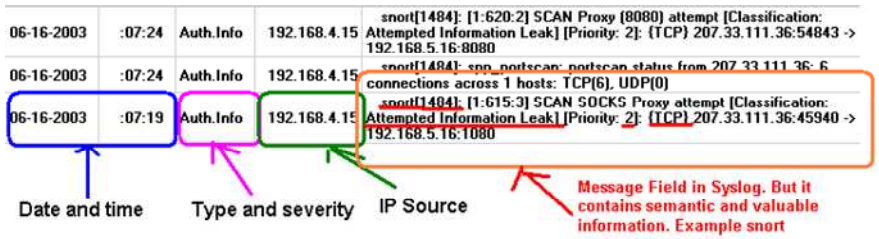


- Incidents need to be well charaterized and expressed in terms of easy raw events relations.
- Parsing from raw event formats such as Syslog to the adapted IODEF (Incident Object Description Exchange Format) has to be done in order to catch the attributes that charaterized the well-known attacks.

From Raw Events to Incidents

Extracting reliable information from raw Events

- Raw Events come from different loggers or trails of information: Syslog, log files from applications, EventViewer, Nagios, etc It is quite vendor dependable. Only standard syslog follows an RFC but anyhow, information has to be analyzed from a semantic point of view.
- It is also possible to calculate and produce events according to certain properties of the systems involved : Metrics and thresholds for them.
- As log sources have heterogeneous format it is required to translate them to a common format to compare and gather information.



From Raw Events to Incidents II

Extracting reliable information from raw Events

- There are several standards for incident description: IDMEF, IODEF, BEEP,etc.
- It was chosen IODEF as it allows customized extensions to the definition. In addition, it is flexible enough for majority of log sources and offers semantic understanding of entities involved in the incidents.
- Once events have been filter to leave out those without interest and accordingly translated, they can be logically compared, and calculations made, this is the Correlation work.
- Basic incidents are compared to logical propositions. Each basic incident is a basic fact.

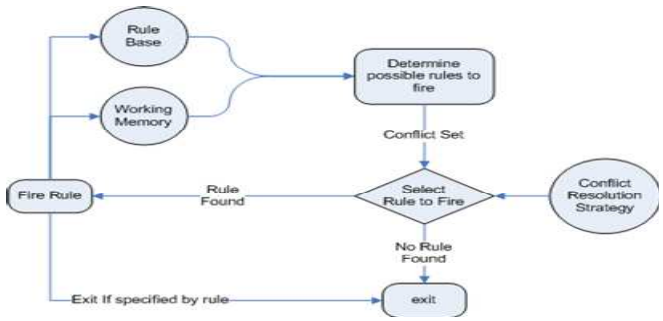
The RETE Algorithm

Inference Process: Algorithms and Engines

- Well-Known Serious attacks are mostly signature based, this mean they can be decomposed in terms of basic incidents and logical relations among them.
- Once a Serious attack has been detected, an alarm to the fast reaction module is triggered, containing the type of attack and the parameter required for taking action in the response.
- First part of the rule contains the facts and relations observed that will trigger the alarm
- Second part of the rule (consequence) will tell what type of attack or serious incident we have then discovered and some other information (severity, for example).
- It is important to store partial information ready to be used not to calculate all the relations every time a new part of the premises appears in the facts.

The RETE Algorithm (II)

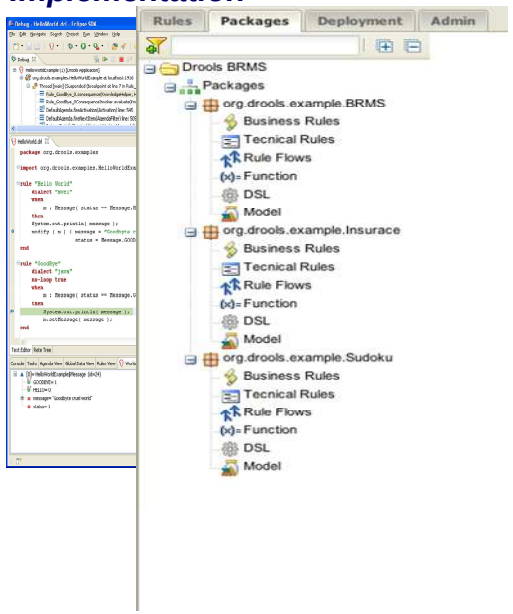
- The RETE algorithm is optimized in time for forward chaining deduction, this is: To determine what type of problem happened once the premises and their relations are given.
- This algorithm stores in memory part of the premises already occurred to fasten next future conclusions triggering and provides conflict resolution solution anytime two or more rule apply at the same time.



Drools and the Rule-Based Implementation

Java Drools

- Drools come as the a Java Object Oriented implementation of a Rule Production System.
- It can efficiently implement the RETE algorithm (but not only this one).
- Recently, new packages such as BRMS – Business Rule Management System allows interaction with the set of rules to be defined. So it is possible not only for designers but for every system architect or security administrator, to design and test and rebuild if required the rules for incident describing.



Conclusions and future work

- **Event collection and normalization.** There is still much work to do when it comes of parsing into normalized information easy to work with. Currently there are working groups for IODEF and IDMEF devoted to normalization and semantic behind common log description.
- **Description of Serious Incident in terms of basic incidents.**

  It is not an easy task to describe some Serious Incidents in terms of some logical relation of basic incidents from raw events. Some work with ontology like Oval, Orbac, lambda and so is into being to help this task.
- **Customization of rule-designing and Rules Engines.**

  Although BRMS and the graphical interface has helped the security administrator to design and try their own rules, it is still required to have standard sets of rules for general purpose like most IDS have, such as Snort for example.

Thanks for your Attention!

mtavelino@gmv.com

Any Question?

Event-Driven Architecture for Intrusion Detection Systems Based on Patterns

Jesús J. Martínez Molina, Miguel A. Hernández Ruíz, Manuel Gil Pérez, Gregorio Martínez Pérez, and Antonio F. Gómez Skarmeta

University of Murcia, Spain

Paper abstract

Intrusion Detection Systems (IDS) are usually one of the basic mechanisms in use when defining security measures in one organization. However, there are a few active research lines still to be addressed regarding these complex systems. This is the case of event aggregation and correlation when dealing with complex attacks, or the improvement in flexibility when dealing with different versions (mutations) of a given attack. These are two of the main objectives of the research work done so far in our group. As part of this effort, this paper presents an event-driven and multi-layer architecture based on the concept of pattern and where concepts such as similarity and credibility degrees are presented as part of a probabilistic approach for dealing with possible variations of a given attack.

1st International Workshop on
Dependability and Security in Complex and Critical Information Systems
(DEPEND 2008)

August 25th-31th, 2008. Cap Esterel, France

Event-Driven Architecture for Intrusion Detection Systems Based on Patterns

Manuel Gil Pérez

Department of Communications and Information Engineering

University of Murcia (Spain)

Outline

- Introduction
- Event-driven architecture
  - Pre-processing layer
  - Event detection layer
  - Decision layer
- Illustrative example
- Conclusions and future work

Introduction

- Existing IDS solutions are only able to discover atomic attacks
  - These atomic attacks do not have any meaning individually
  - Current attacks are much more complex
    - A set of atomic attacks could conform a complex attack
- Complex attack → set of steps that an attacker can carry out in the system to reach his final goal
- New solutions are based on the search of patterns to detect complex attacks

Provide a multi-layer architecture based on patterns, able to react stopping the attack

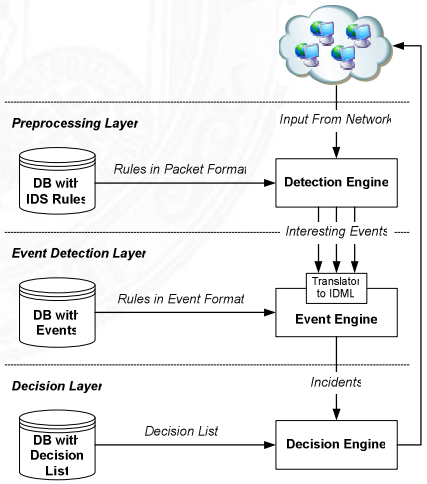
Introduction

- IDS architectures are based on fixed patterns
  - The steps done by the attacker must fit in exactly with the expected ones by the architecture
- Example: Netsky worm virus can mutate in 36 different ways
- Situation that could complicate the management of large systems
  - Define for every kind of attack all its possible variations?

This architecture must support some flexibility, indicating the trust level the system has when detects an attack

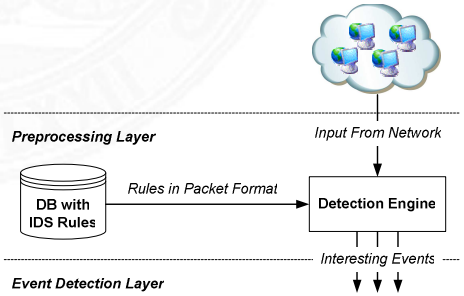
Event-driven architecture

- Architecture composed by three well-defined layers



Pre-processing Layer

- It examines in real-time the packets of interest transmitted through the protected network
- Network packets are considered relevant on the basis of the information contained in a rule database
- This layer is composed by a complete NIDS

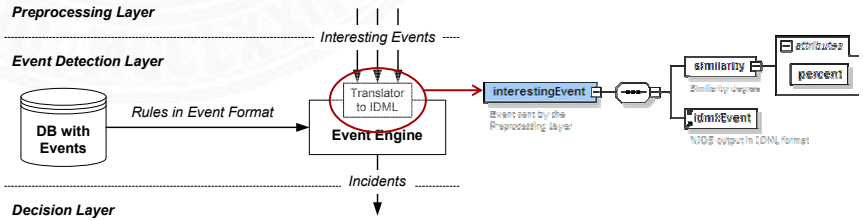


Pre-processing Layer (Similarity Degree)

- Probabilistic value that represents how much of similar is a packet network to the signature stored
  - Provide some flexibility to the signatures defined
  - Avoid to define all the possible variations of an atomic attack
- A value of 100% represents the total certainty that the attack has been produced
  - The *Detection Engine* could discard events detected whose similarity degree is below a certain threshold
- The output of this layer is an XML document with:
  - *Similarity degree* calculated
  - Output itself produced by the NIDS with the event detected

Event Detection Layer

- This layer determines if the *interesting events* represent a complex attack or not
- Provides a high level vision of the raw data correlating the lower events (atomic attacks)
- IDML is the standard format used to define attack graphs
  - Define events, states and transitions between them



Event Detection Layer

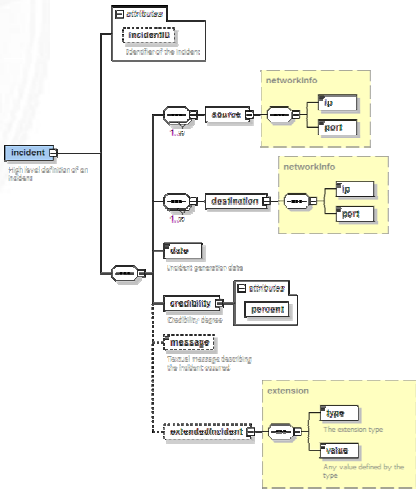
- Each attack graph is labelled with an identifier
  - It will be used later to react properly
  - Several attack graphs can group with the same ID to assign the same reaction afterwards
- The pattern-based *Event Engine* maintains in memory:
  - IP address of the attacker
  - IP address of the victim
  - Current states in which the attacker is:
    - Attack graph identifier
    - List of next states to which attacker might move
    - Accumulated probabilistic value → *credibility degree*

Event Detection Layer (Credibility Degree)

- Same concept to the *similarity degree*
- Probabilistic value that represents the accuracy the attacker has had in each of the steps carried out in the system
  - Provide some flexibility to the attack graphs defined
  - Avoid to define all the possible variations of an complex attack

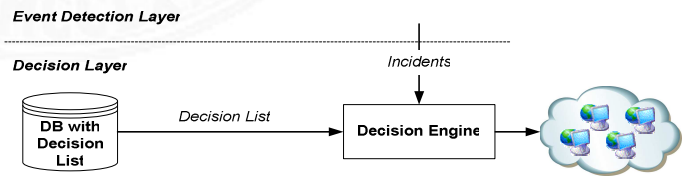
Event Detection Layer

- The output of this layer is an XML document with:
  - Attackers' network info
  - Victims' network info
  - Incident date
  - *Credibility degree* calculated
  - …

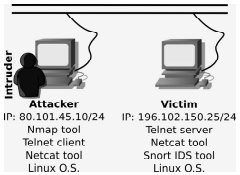


Decision Layer

- This layer executes the appropriate action to stop the attack
- The *Database with Decision List* maintains the information related to how the system should react:
  - Block attacker's IP, report to the administrator, …
- The *Decision Engine* matches the incoming incidents against the list of actions stored in the database
  - Decision made using the elements *incidentID* and *credibility*
  - The decision could vary depending on the *credibility* degree

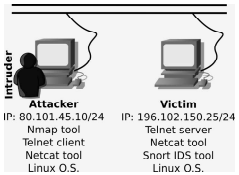


Illustrative Example



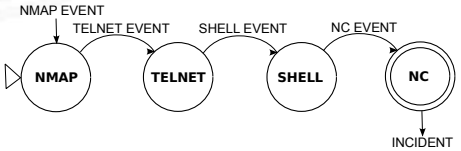
- **Attacker**
  - Linux O.S.
  - Nmap tool to execute a portscan
  - Telnet client
  - Netcat tool
- **Victim**
  - Linux O.S.
  - Telnet server running on port 23
  - Netcat tool
  - Snort to process all packets addressed to Victim

Illustrative Example



Attacker behaviour:

- Attacker does a portscan against Victim
  - Attacker finds a Telnet server (port 23)
- Attacker connects to Victim by means of Telnet
  - Let's suppose Attacker obtained the user/password sniffing the network previously
- Attacker executes a command, leaving a backdoor in Victim using netcat tool
- Attacker connects to Victim using the backdoor



Illustrative Example

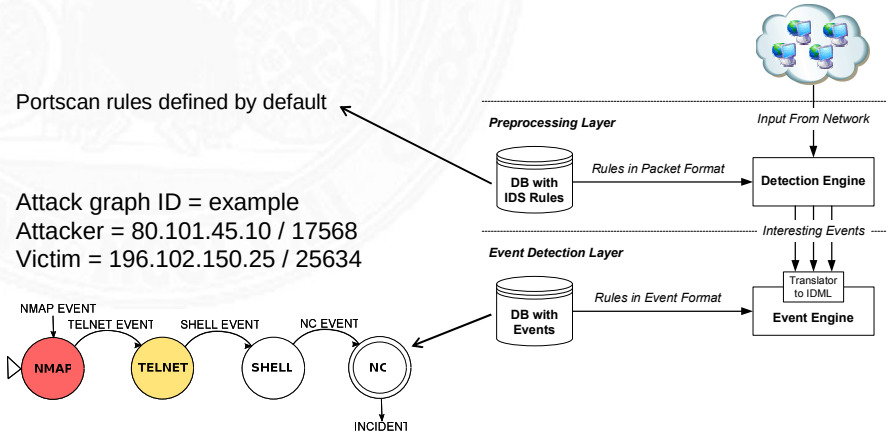
Snort rules defined for the example

alert tcp any any → $TELNET_SERVERS 23 (msg:"TELNET connection established"; flags:S,12; sid:9000465;)
alert tcp any any → $TELNET_SERVERS 23 (msg:"Remote shell execution (/bin/sh)"; flow:to_server,established; content:"/bin/sh"; classtype:remote-shell-execution; sid:9000001; rev:1;)
alert tcp any any → $TELNET_SERVERS 23 (msg:"Remote shell execution (/bin/bash)"; flow:to_server,established; content:"/bin/bash"; classtype:remote-shell-execution; sid:9000002; rev:1;)
alert tcp any any → $TELNET_SERVERS 23 (msg:"Command execution attempted"; flow:to_server,established; content:"nc"; classtype:dangerous-command; sid:9000003; rev:1;)

- Rules for detecting portscan operations are not included here
  - Snort defines them by default

Illustrative Example

- [Attacker]# nmap -sS -sV 196.102.150.25
  - Attacker gets a list of available services → Telnet on port 23

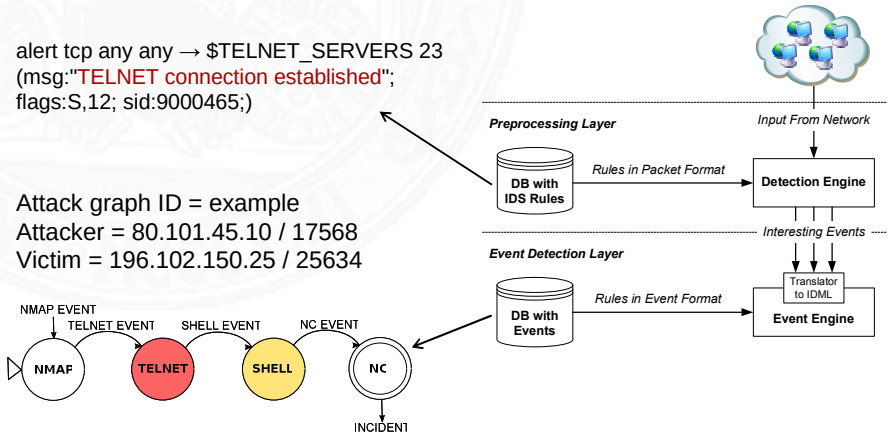


Illustrative Example

- [Attacker]# telnet 196.102.150.25
  - Attacker accesses to Victim through the Telnet server

alert tcp any any → $TELNET_SERVERS 23
(msg:"TELNET connection established";
flags:S,12; sid:9000465;)

Attack graph ID = example
Attacker = 80.101.45.10 / 17568
Victim = 196.102.150.25 / 25634



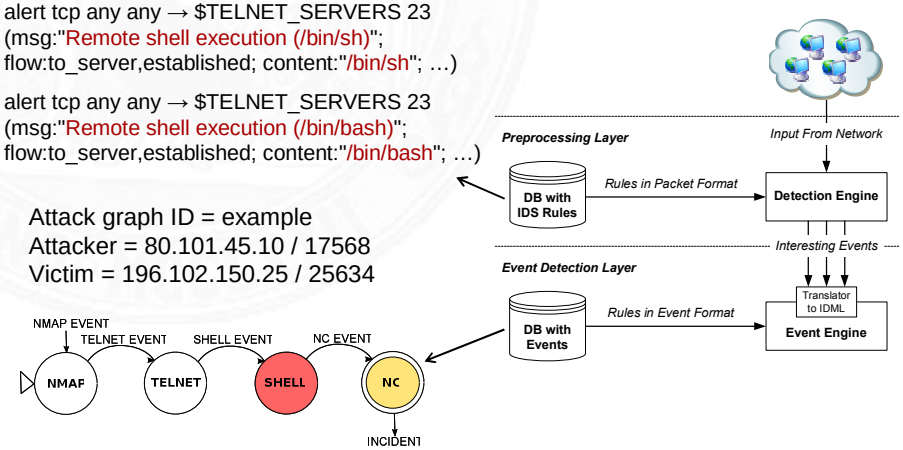
Illustrative Example

- [Victim]# nc -l -e /bin/bash -p 25634
  - Attacker leaves netcat tool listening on port 25634

alert tcp any any → $TELNET_SERVERS 23
(msg:"Remote shell execution (/bin/sh)";
flow:to_server,established; content:"/bin/sh"; …)

alert tcp any any → $TELNET_SERVERS 23
(msg:"Remote shell execution (/bin/bash)";
flow:to_server,established; content:"/bin/bash"; …)

Attack graph ID = example
Attacker = 80.101.45.10 / 17568
Victim = 196.102.150.25 / 25634

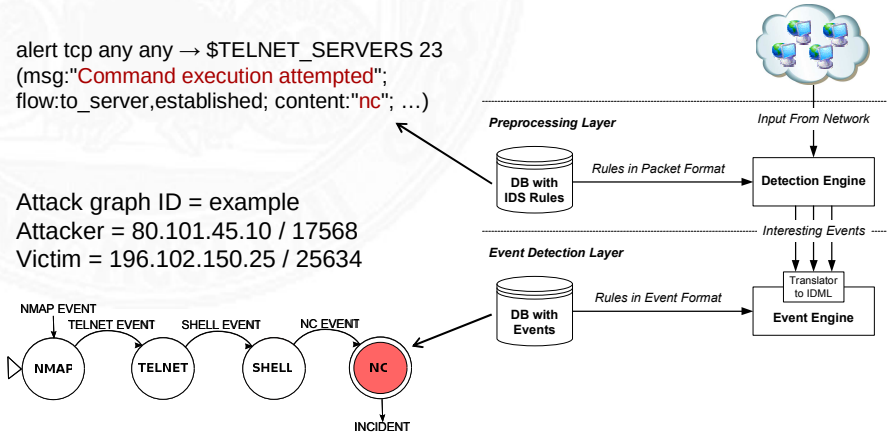


Illustrative Example

- [Attacker]# nc -vv 196.102.150.25 25634
  - Attacker accesses to the remote shell through the netcal tool

alert tcp any any → $TELNET_SERVERS 23
(msg:"Command execution attempted";
flow:to_server,established; content:"nc"; …)

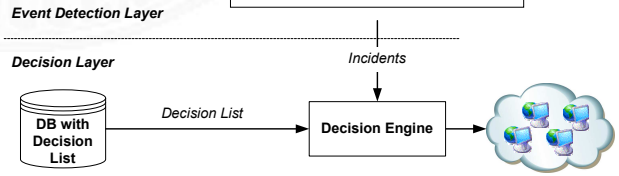
Attack graph ID = example
Attacker = 80.101.45.10 / 17568
Victim = 196.102.150.25 / 25634



Illustrative Example

- Possible reaction: Add filtering rule to block Attacker

```
<incident incidentID="example">
    <source>
        <ip>80.101.45.10</ip>
        <port>17568</port>
    </source>
    <destination>
        <ip>196.102.150.25</ip>
        <port>25634</port>
    </destination>
    <date>27/08/2008 17:00</date>
    <credibility percent="100"/>
</incident>
```



Conclusions

- Three-tier architecture based on patterns to detect complex attacks
  - Lower layer is based on existing solutions to aggregate raw data (network packets) and only report interesting events
  - Higher layer detection engine based on patterns correlates atomic attacks to discover attacks much more complex
  - It is able to respond in front such a complex attack to avoid more security problems
- Two probabilistic values (similarity and credibility) are calculated
  - Trust level the system has in detecting an attack
  - More flexibility for detecting *mutations* of a given attack

Future work

- An initial design is done
  - A prototype implementation is being developed using Snort
- Evaluate the prototype by calculating performance measures
  - The authors hope to publish them in following papers

1st International Workshop on
Dependability and Security in Complex and Critical Information Systems
(DEPEND 2008)

August 25th-31th, 2008. Cap Esterel, France

Event-Driven Architecture for Intrusion Detection Systems Based on Patterns

Thanks for your attention!!

mgilperez@um.es

Department of Communications and Information Engineering

University of Murcia (Spain)

Agent Based Approach to Events Monitoring in Complex Information Systems

Marek Woda and Tomasz Walkowiak

Wroclaw University of Technology, Poland

Paper abstract

This work is devoted to some theoretical aspects of events monitoring in Complex Information Systems. Authors depict some aspects of nowadays monitoring systems that make them ineffective. Next, research areas are presented. It is followed by a proposition of a new technique of monitoring that is being implemented, based on agents and idea of the molecule. Then test system architecture is discussed. Lastly, the advantages of agents based approach are revealed.

DEPEND 2008
25-29 August, Cap Esterel, France

Agent based approach to events monitoring in Complex Information Systems

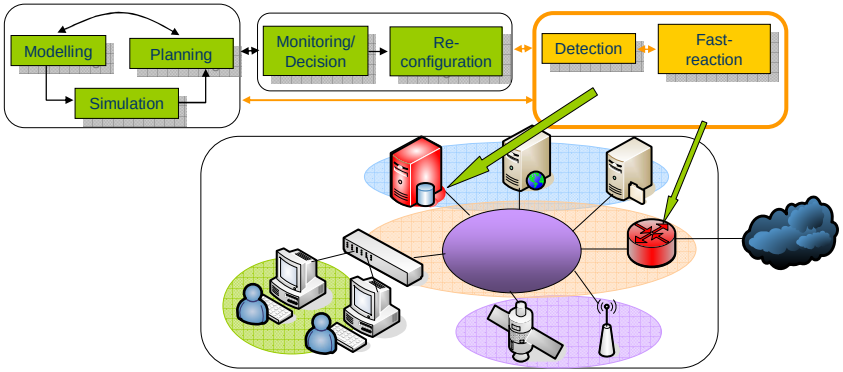
Marek Woda, **Tomasz Walkowiak**
Wroclaw University of Technology

DESEREC
Dependability & Security by Enhanced Reconfigurability

Information Society Technologies

Monitoring in DESEREC project

- http://www.deserec.eu
- **WP4: Fast cicatrisation**
- Provide the basic conceptual and technical tools for implementing incident detection and fast reaction.



Monitoring

Monitoring

- process of collecting,
- analyzing,
- signaling event occurrences,
- a key part of systems security / reliability.

Monitoring - software

Open source

- e.g. OpenNMS, SNMP based solutions

Commercial

- e.g. IBM Tivoli Monitoring, AdRem NetCrunch

Drawbacks of existing systems

- ineffective,
- sparse intelligence,
- massive resource consumption,
- often expensive.

Monitoring systems - requirements

Requirements towards monitoring

- flexibility,
- modularity,
- capability processing large datasets,
- ability to working huge multi-networks,
- produce meaningful information,
- fast response.

Agent based monitoring system

Prototype

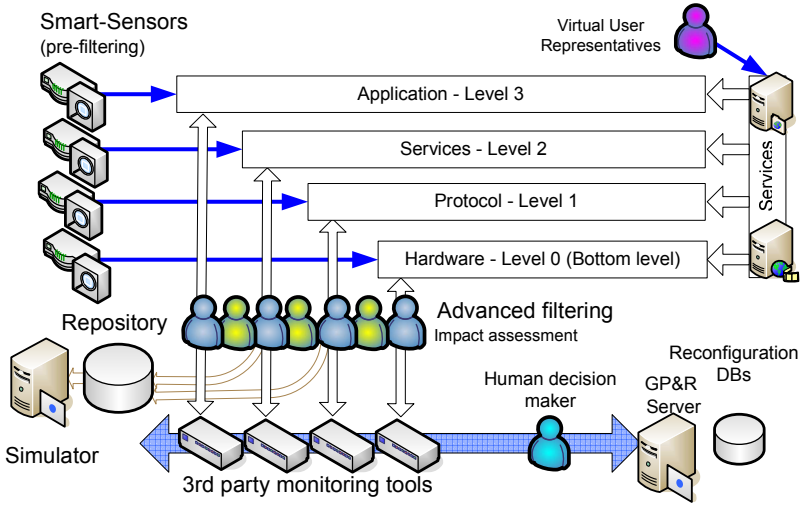
- synthesis of several technologies,
- AI employed (fuzzy logic, data mining, neural networks and/or clustering techniques),
- system is built using 'smart' (intellignet) agents.

Agent based monitoring system

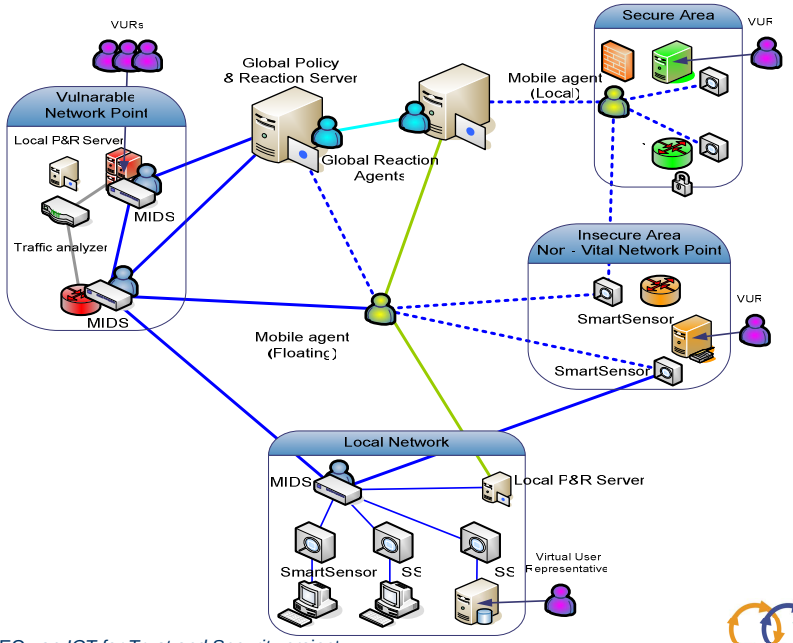
Terms and definitions

- Molecule
- Business Service
- System Tier
  - High level (business service level);
  - Middle level (component service level);
  - Low level:
    - operating system (local),
    - smart agents (remote).

Agent based monitoring system



System overview

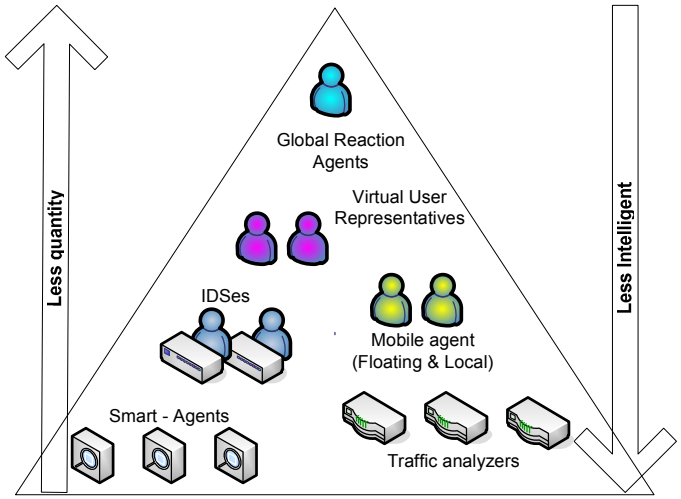


Agent based monitoring system

Types of agents

- molecule, global reaction,
- mobile (communication),
- User Virtual Representatives,
- smart agents.

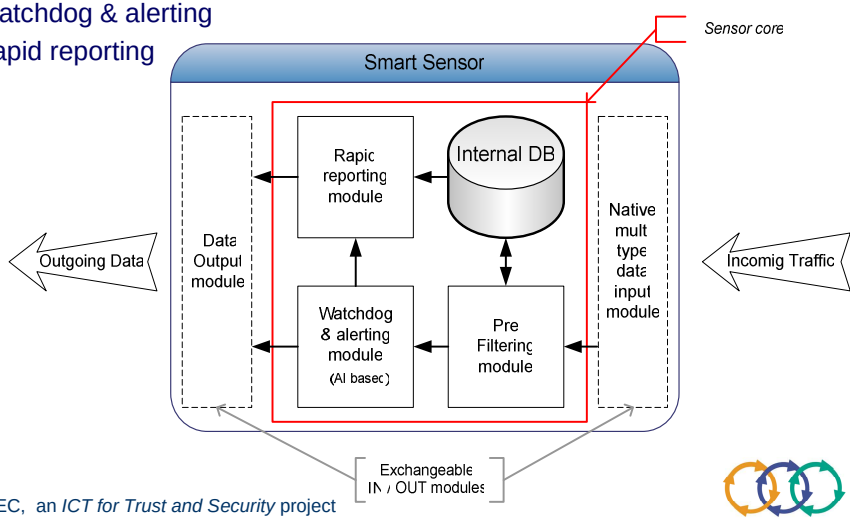
Agent based monitoring system - agents hierarchy



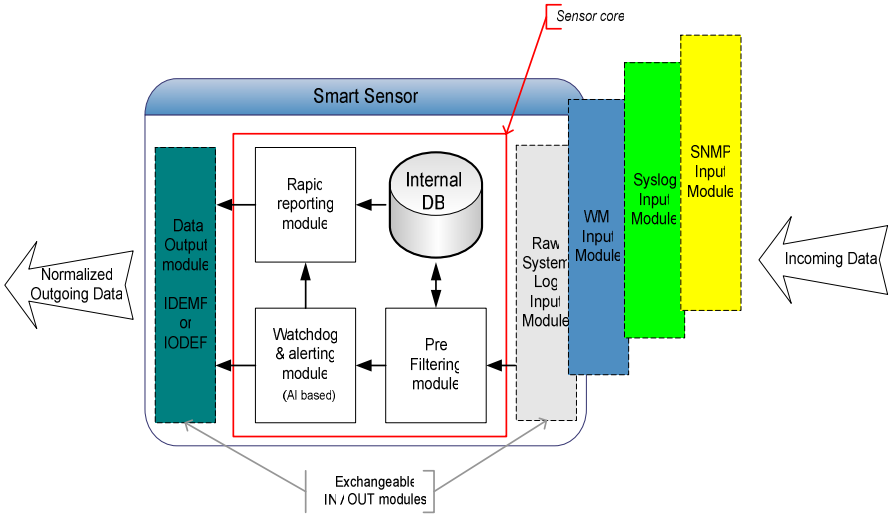
Agent based monitoring system – smart agent components

Modules

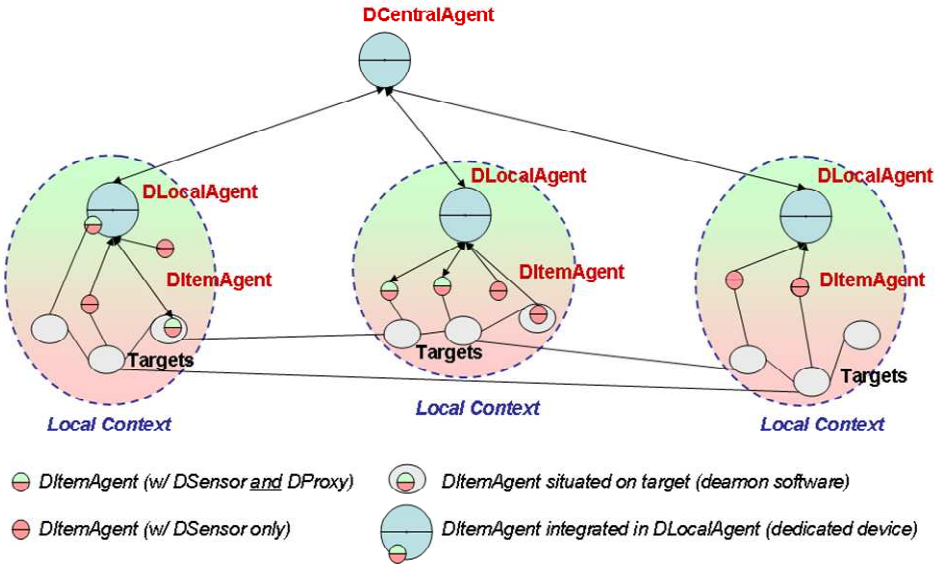
- Input / Output
- Pre-filtering
- Watchdog & alerting
- Rapid reporting



Smart Sensor architecture overview

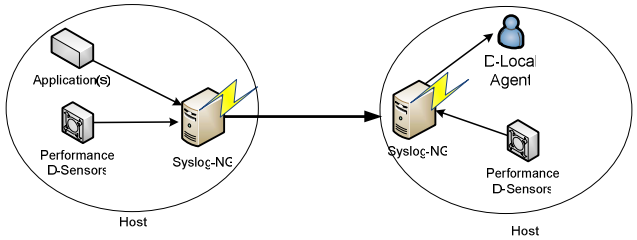


DESEREC Architecture – generic concept

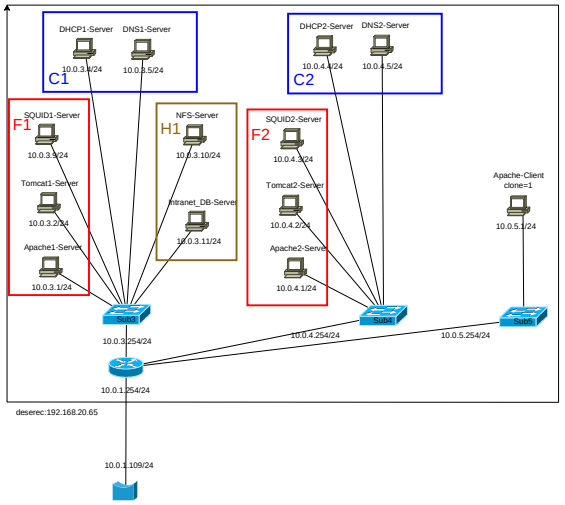


DESEREC monitoring – D-Sensor

- **Performance D-Sensor**
- **Syslog D-Sensor**
  - **Apache/Tomcat,PHP**
  - **Radius, Samba, BEA Weblogic**
  - **MySQL, Squid, Oracle, Darwin**
  - **FTP, SMTP, BIND, DHCP, IPTables**
  - **NAGIOS, SNORT**
  - **TLIntegratorProbe, DOSProt**
  - **….**

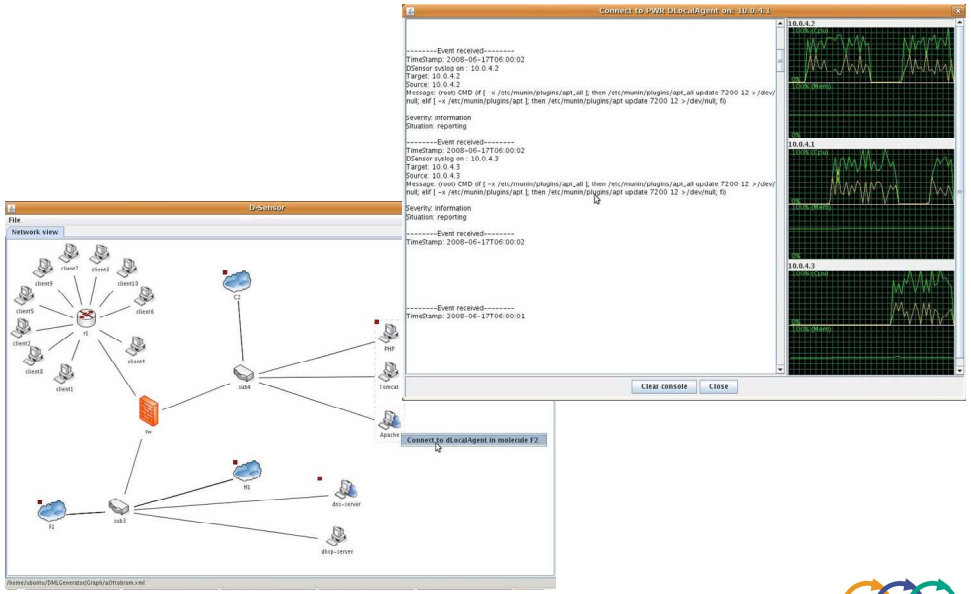


DESEREC monitoring – virtual testbed

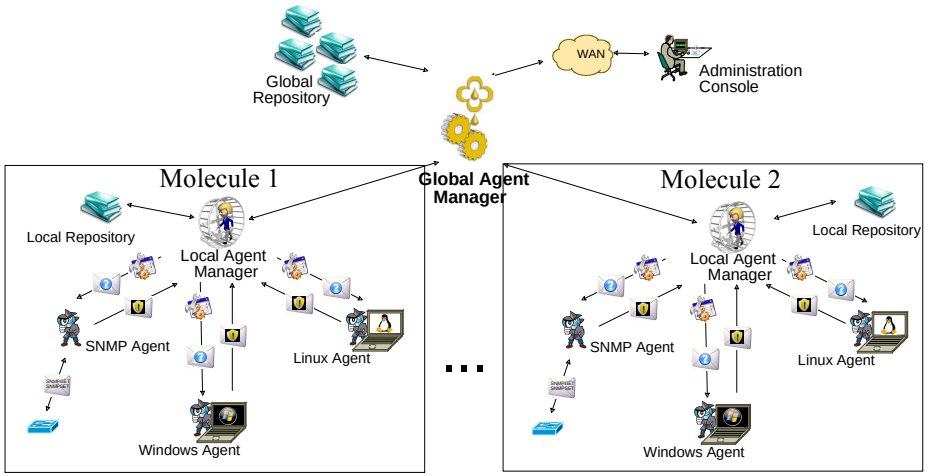


- DLocalAgents at 10.0.4.1 (F2)
- DCentralAgent at 10.0.1.109 (Ubuntu VN-UML host)
- DLocalAgent D-Sensors available (syslog parsers):
  - Apache,
  - DNS,
  - DHCP,
  - NFS,
  - SQUID,
  - Tomcat,
  - Performance D-sensor.

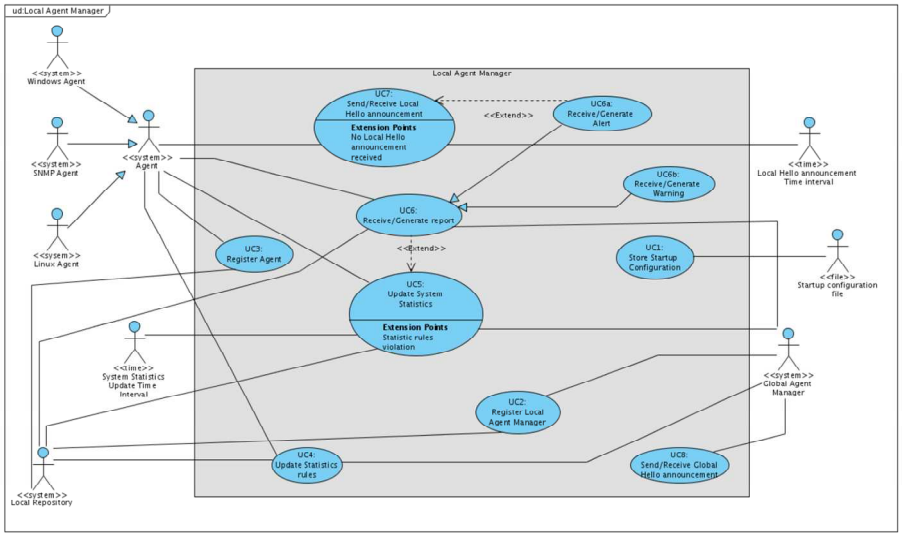
DESEREC monitoring - virtual testbed



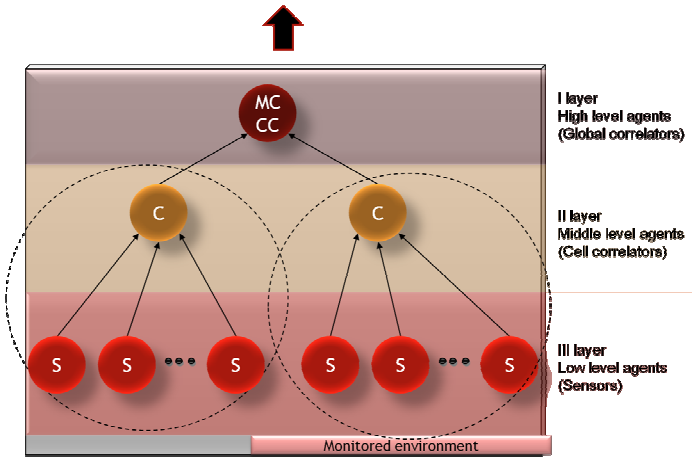
NESMA - application



NESMA - application

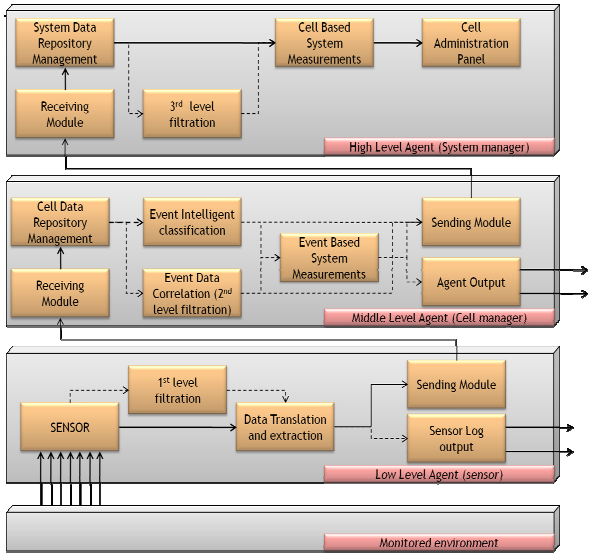


SEA – application

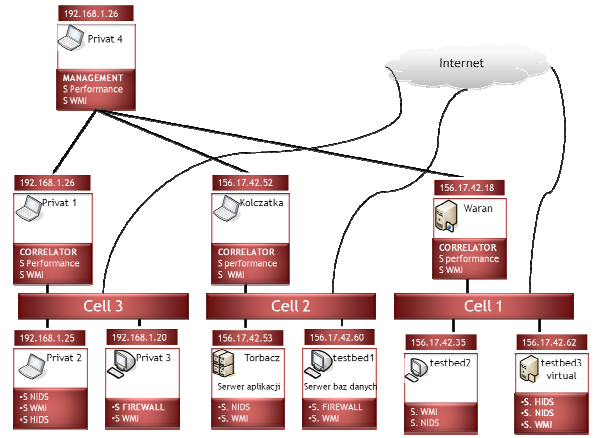


S – sensor performing data collection
C- correlators performing data correlation for agent-cells
MC – Main correlators performing data correlation from independent agent-cells

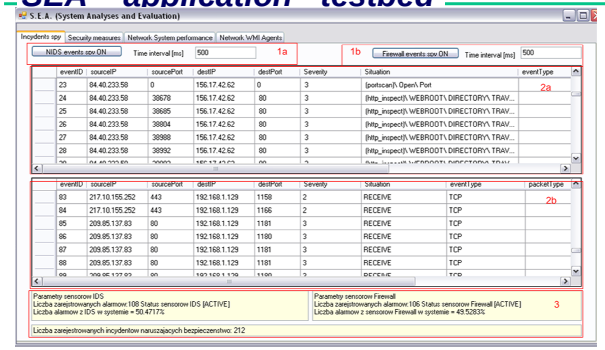
SEA – application



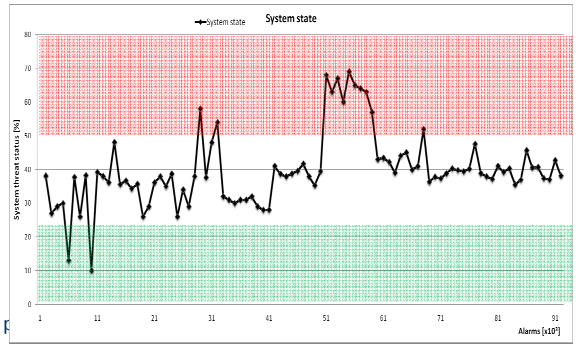
SEA – application - testbed



SEA – application - testbed



Cell (molecule) state is accumulated value of danger incidents discovered in the cell, sophisticated attempts, cell reliability and thread for a cell.



Agent based monitoring system

Prototype advantages

- combines WMI & SNMP mechanisms,
- OS, domain independent,
- automatically notifies about threats,
- accurate recognition of the events,
- pre-filtering mechanisms prevents false alarms,
- analyses high traffic volume due to distributed nature.

Future development

- more AI
- automatic rule parameters setting up
- Virtual User Representatives

On System Security Metrics and the Definition Approaches

Artur Hecker

Ecole Nationale Supérieure des Télécommunications, France

Paper abstract

In this survey paper, we assess existing approaches to security metric definition. We classify proposed definitions and discuss their advantages and problems. We argue that without a more restrictive definition, the apparently common term degenerates to a mere buzzword, which can be dangerous in terms of suggested comparability. We conclude with some guidelines on IS metric definition and sketch an alternative concept for the operational IS security evaluation.

On System Security Metrics and Definition Approaches

Artur Hecker
TELECOM ParisTech (ENST)

Motivation: why IS metrics?

- Management requires measurement
- Build more dependable and robust systems
- In principle: operational System evaluation
  - Capture significant parameters
  - Process the data
  - Make a decision
- In practice:
  - From static judgment on an isolated IS function to a strategic enterprise decision helper

IS Metric research

- Literature
  - Vaughn et al.
  - Seddigh et al.
- European Research
  - RESIST, BUGYO, DESEREC, SERENITY, POSITIF, …
- Standards
  - NIST 800-26, NIST 800-55, Canadian IAM, SSE-CMM, ISO 27004

Typical (ab)use of metrics

- Personal usage
  - Any algorithm that does what I need
  - If not the whole process
- Broad usage
  - Business efficiency indicator
  - Distance in a metric space
  - Security strength comparator
  - Emerging properties evaluator
- Dialectic: the same word suggests the same meaning & understanding

A short guide to discussions … with IS metric experts ;-)

- Do not say what it is
  - Abstain from definitions, metric spaces and/or comparability thoughts
  - Never be trapped in a given problematic
    - Do not try to sketch a metric in any given context
    - Instead come up with your own context
- Say how you want it to be
  - Underline the importance of classifications
    - Properties, taxonomies, etc.
  - Do not spare examples to sound even more serious
    - Do not mind if examples cover nail strength and rocket science efficiency
- Be cool: solve old-standing humanity problems
  - Put the real complexity within the obscure metric constructs
  - Come up with a graph/model/algorithm using these as leafs/arcs/calls
  - If your algorithm does not work (it won't), blame the metrics, the people behind them or the whole IS research

Existing definitions

- Not a valid definition
  - "A function on a set of system parameters"
- Mathematical definition
  - What is the underlying space?
  - Is IS metric a subtype of metric?
  - How can we produce a metric from several metrics? On what space?
  - What are mathematical properties? f(a,a)? f(a,-a)? f(a,0)? f(a,b) ~ f(b,a)?
  - Is not the final goal
  - Example
    - Take algorithmic complexity (Kolmogorov): K(x)
    - Vulnerability = easier than the supposed way to do sth
    - Correct definition, but K(x) incomputable
    - See Evans et al. for calculation of apparent complexity
    - Hardly applicable in practice

Definition approaches

- Pragmatic definitions
  - A posteriori definitions (NIST 800-55)
    - Tools designed to facilitate decision making and improve performance <…> through collection, analysis, and reporting of relevant performance <…> data
    - Ok… but HOW?
      - What are these tools?
        - Typical metrics: syslog logs, telnet sessions?
      - What to measure and where? How to combine different tool outputs?
- WISSR definition
  - An IS metric/measure is a value, selected from a partially ordered set by some assessment process, that represents an IS-related quality of some object of concern. It provides, or is used to create, a description, prediction, or comparison, with some degree of confidence
  - Better: something could be not a metric
  - Important: notion of confidence
    - E.g. repeatable / verifiable

Definition requirements

- Need a clear definition
  - What is a metric?
    - A. What is it?
    - B. What is it required to comply with?
  - What is not a metric according to A and B?
- "Metric" is a very bad work
  - No order relationships on security
    - Is Sec1 better than Sec2? In which sense?
    - How to compare security policies?
  - Metric = distance
    - Distance between a, b: d(a,b) = 0 or infinity

Expected IS metric properties

- Objective/subjective
- Qualitative/quantitative
- Static/dynamic
- Absolute/relative
- Direct/indirect

See Vaughn et al.

Possible approaches
Low level metrics

- Relatively easy to define metrics:
  - Low level metrics
    - Stay close to the physical world
    - In essence, measurements attached to verifiable and objective properties of an object
      - Typically have numerical scales
        - Key length of a cipher, password quality, performance of an IS function on a given platform
      - Total order relationship exists on the parameter
        - A mathematically correct definition
        - Comparable within the same context
  - Problem: assess the same LL metric on different subsystems, how to combine values?
    - Requires system knowledge
    - Aggregation…

Possible Approaches
Ranking based definitions

- Metric definition
  - IS Metric is the distance between the specified state (as should-state) and the real operational state (as is-state).
- Measures conformity of the deployed security
- Relies on precise requirement documents
  - Protection profiles, SPP, etc.
  - Mapping onto a common scale through pre-established matrices/tables
  - Only a weak link to the real impact, enumerations of « wrong » configurations
- Projects requirements on the assessment framework
  - Parameters need to be assessed in respect to the verified requirements
- Inherently incomparable
  - Needs same ranking / same requirements / same assessment system

Possible Approaches
RA-based approach problems

- Easy to give binary answers but numerical answers are difficult
- RA is not a fixed point
  - Subjective
  - Always obsolete
- Even presuming a perfect RA
  - Verifies the conformity by presuming that the actual countermeasure works correctly
    - Probably a good assumption for an isolated, single countermeasure
  - Is not true for distributed countermeasures
    - Distributed countermeasure: several execution environments interconnected by a communication facility
      - The latter is typically shared, with high dynamics
      - Here, the correctness of the security function depends on the underlying network's load & state, the robustness of the protocols, etc.
    - => Mixes security and reliability domains

Possible Approaches
Attack-graph based approach

- Describe all possible attacks
  - Form a tree by defining required preparatory actions as parent nodes
  - Could be extended to a graph
- Define the metric on the graph paths
  - In essence, length of the path plus score every arc according to e.g. CVSS
- Until now only system design - not operational
  - Extension: establish the graph automatically
    - Capture the configuration/versions of elements in runtime
    - Establish the graph with the public vulnerability DBs
      - NIST NVD, public OSVDB, etc.

Possible Approaches
Attack-graph / problems

- Not directly related to local RA… but very dependent on the Vulnerability DBs
  - Usually only include entries of the form element->{vulnerable versions & configurations}
    - E.g. permits to use the most robust Apache
- Cannot discover deviations from the local security policy
  - Still needs conformity checking → RA
  - Resolve conflict: local policy vs. most robust
  - Difficult to treat vulnerabilities on the system level
    - Interdependent configurations
    - Overload/dimensioning
    - Unauthorized usage/DoS/etc.
  - Graphs grow quickly and explode for complex systems
  - Could be applied to subsystems, at a local level, etc.

Possible Approaches
Attack Surface Estimation

- Estimate the number of access points to the subject system
  - Interfaces, protocols, active instances, etc.
  - Main assumption: more = more vulnerable
  - Does not treat threats, no relation to SLA
- Doubtful methodology
  - Does more really *still* mean more vulnerable?
    - Is FTP (2 ports) more vulnerable than HTTP (1 port)? What about Skype?
  - Does not take into account system architecture
    - No consideration of protection measures
    - No consideration of engineering
    - No consideration of values and/or assets
- Difficult to compare different systems
  - For instance, Windows and Linux OS
  - Works best on different versions/shapes of the same well-understood security subject under certain basic assumptions

Metrics & models

- Bidirectional dependency
  - Models describe what will be captured, metrics operate on this data
    - More context in the model allows simpler metrics
  - Intra-systemic dependencies
    - How to compose/aggregate values?
    - Should be a function of the model or treated in the metric
      - To express redundancy, functional dependency, etc.
  - Complexity shifting
    - The more you put into model the less you need in the metrics and vs.
      - Given the unclear metric definition and experience with modeling…
    - Compare RA-based approach and attack-graph approach

Metrics & comparability

- Often sought for
- Dangerous: security is subjective by definition, linked to the confidence and the environment
  - Resulting for two different Ras and/or attack graphs for two different systems and/or environments
- IS metrics: not only technical parameters matter
  - Organizational policies, user education, procedures are difficult to compare and score
  - Examples: “percentage of educated users”, “motivation of workers”, “quality of security agents”, “rigor of the password process”

Lessons learnt

- Parameters to be captured are the most crucial choice regarding IS metrics
  - Express relations between captured parameters within models to simplify the badly understood metrics
- Limite sense and applicability of every IS metric in advance
  - Context, security policy, available knowledge, services
- Drop comparability requirements beyond system boundaries
- Third party evaluations are doubtful
  - Need field experience with the system, topology data, usage data
- Generally applicable IS metrics are nonsense

Idea for an alternative approach

- Motivation: difficulties with RA in modern systems
  - Big virtual part, controlled by software
    - Changing shape, dynamic topology, adaptation
  - RA introduces new, add-on processes
    - Costly and impractical
- Idea: re-instauration of the sysadmin power
  - Identify system-typical, recurrent weaknesses
    - Describe these in knowledge databases (ontologies)
      - Note: understanding/prevention not required at this point and may well be impossible
  - Describe mitigating actions known through experience
  - Assess the probability of the occurrence of a problem through respective measurements (far/close to a problematic situation)
    - Inherently local distance constructs, defined by the local administration
  - Define thresholds as action triggers
- Novelties
  - Acknowledges the complexity, incomparable, informal
  - Describes the systems through its quirks, rather than its shape
  - No separation of robustness and security, accent on detection

Conclusion

- IS metrics are dangerous since they suggest common term and understanding in a purely understood domain
- Classification
  - Low level metrics
    - Based on available quantitative parameters
  - High-level metrics: ranking-based, conformity distance, attack-graph, attack surface
    - Have limitations
    - Link to the risk assessment
- Metrics are linked to the available system knowledge
  - Emphasize and define local metrics through this knowledge, not otherwise

Policy-Driven System Configuration for Dependability

Marco D. Aime, Paolo C. Pomi, and Marco Vallini

Politecnico di Torino, Italy

Paper abstract

When managing a complex information system for critical activities, reaching a high level of dependability is a key requirement. A central challenge is identifying a set of alternative system configurations to be enforced when particular system situations take place. In large systems, this task requires automation as the administrator can hardly cope with the complexity of all components and their interactions. We propose an innovative approach, based on model transformation, to perform automatic generation of alternative configurations taking into account dependability constraints.

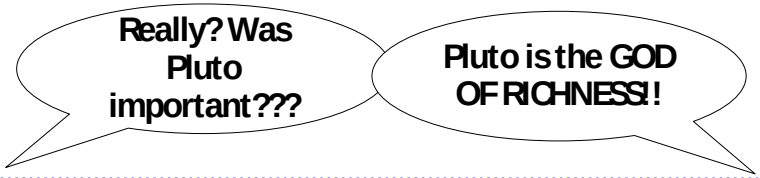
Policy-driven system configuration for dependability

Marco Domenico Aime <m.aime@polito.it>
Paolo Carlo Pomi <paolo.pomi@polito.it>
Marco Vallini <marco.vallini@polito.it>

Introduction

standard high availability techniques
- top class hardware and software
- massive replication of the devices

problems
- does not resolve classical security problem
- hard re-usage of the devices for different purposes
- solution very far from the requirements



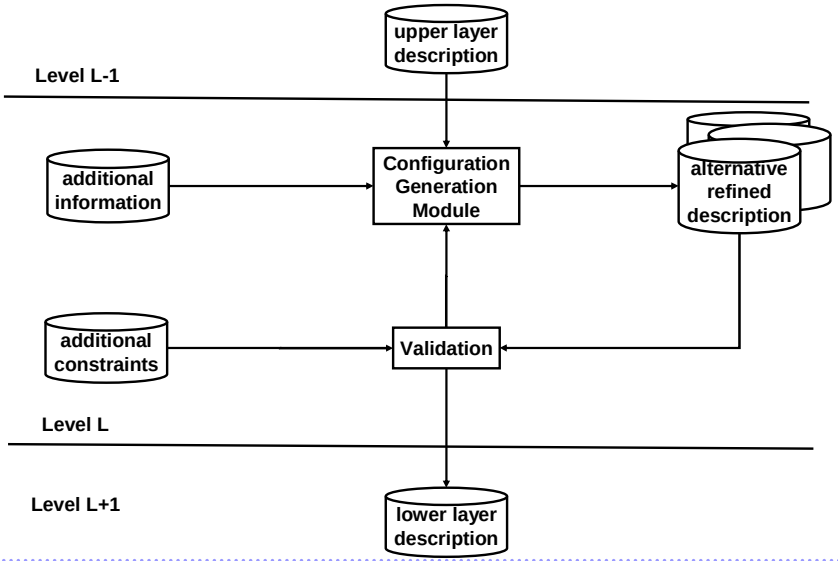
Proposal

generation of alternative configurations considering
- dependability constraints
- security requirements
- Quality of Service levels

automatically using
- model transformation
- model validation

performing a multi-level policy refinement process

General architecture



Models for Business Services

Business Service (BS) composed by

- Business Service Component (BSC): software package to be hosted that has:
  - software dependencies
  - computational power needs
  - storage space requirements
- interaction between different BSC with attached requirements of
  - confidentiality
  - integrity
  - availability

Models for network infrastructure

network graph composed by

- hosts with one or more network devices:
  - purely for networking/connectivity/security purposes
    - e.g. switch, hub, firewalls, IP-SEC aware routers
  - to host BSC with
    - software installed to satisfy sw dependencies
    - computation power and storage space
    - exploitable access points
- links: represents the physical connection of two network devices
- also networks are hosts (to allow multi-level grouping)

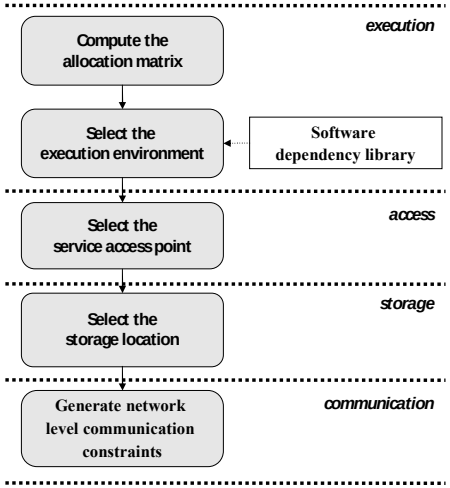
Allocating BSC on hosts

for every BS finds the machine(s) capable to host it considering

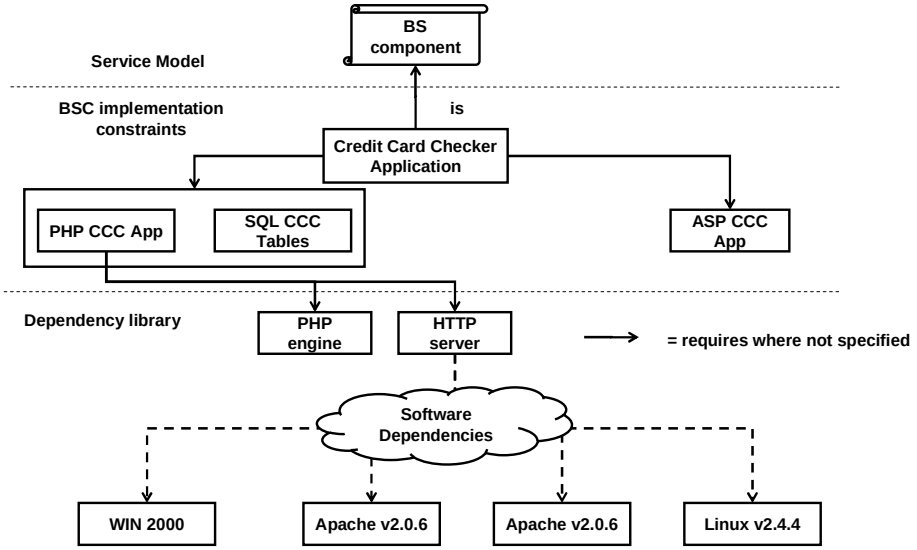
- software dependencies
- computational capabilities
- storage capabilities

every alternative generates a different configuration

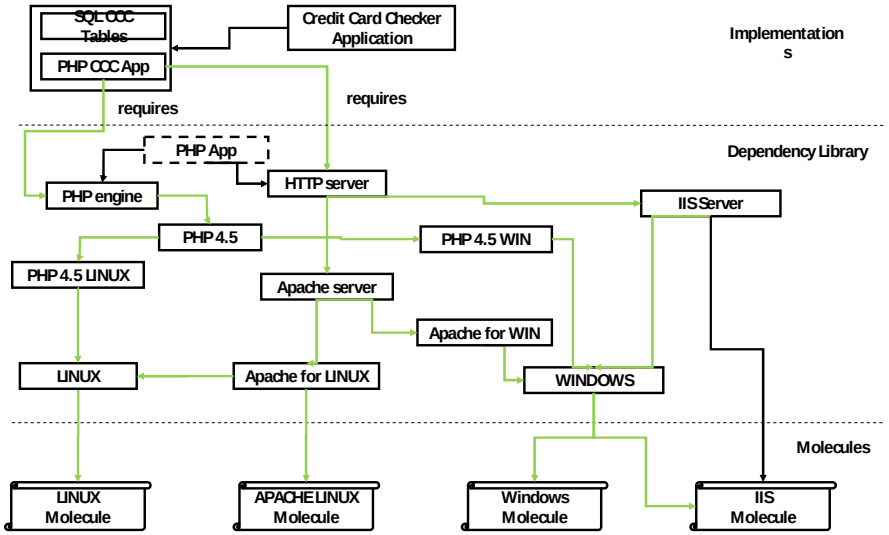
Allocating BSC on hosts - steps



Managing software dependencies



Example of software dependency solution

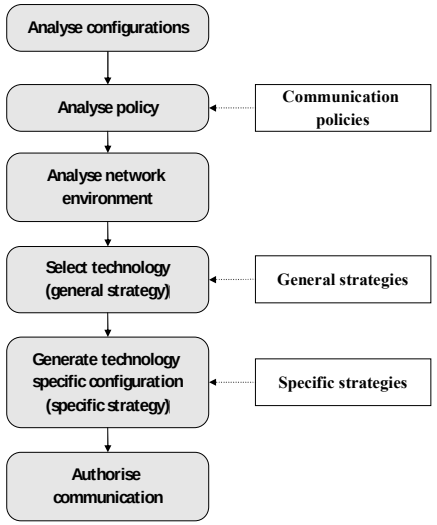


Allow and protect interactions

transforming interaction between BSC in

- interactions between hosts
- rules to permit their inter-communication
  - firewall configuration
- rules to protect their channel
  - using mechanism that does not affect directly applications like IPSec and SSL/TLS

Allow and protect interactions- steps



Case of study: news browsing application

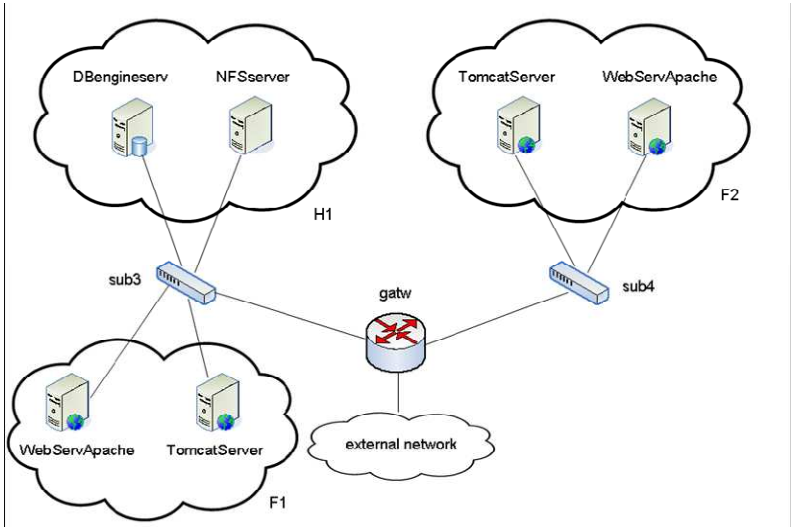
clients connected by intranet using two web apps:

- NEWS that displays up to date news about the company
- JOB that display job opportunities.

Each application relies on its own database.

Interaction between the application and its database is subject to confidentiality and integrity by internal organization policy

Case of study: hosting network



Case of study: alternative allocation

execution:

- NEWS and JOB web applications are provided by the application servers (since)

access:

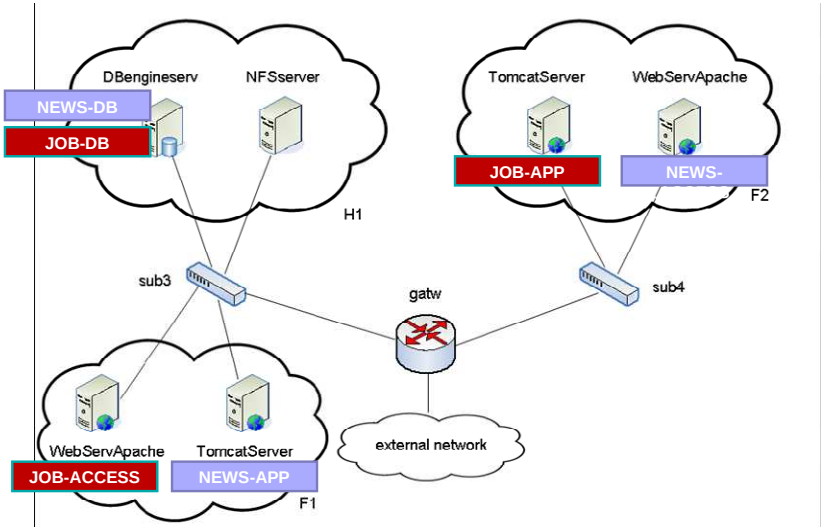
- by both web server and application server

storage:

- software packages implementing applications can be stored on the NFS server or on the execution host
- NEWS and JOB databases are managed by the DB

if we allow the system to install additional technical services the number of alternatives increases

Case of study: a possible allocation



Case of study: communication protection

from requirements

- DBEngine – F1.TomcatServer → confidentiality=medium, integrity= high
- DBEngine – F2.TomcatServer → confidentiality=high, integrity= high
- using rules we decides the technology, considering the available alternatives (SSL/TLS)
- then for the technology decide the configuration
  - confidentiality medium → AES 128
  - confidentiality high → AES 256

from allocation

- allow all the communications necessary (every machine has a personal firewall, router act also as a firewall)

Conclusion and future work

an approach to generate alternative configurations

- to decide which device(s) host different software packages composing a business service
- to guarantee and protect the interactions between these parts

Next steps

- evaluate configurations to choose the best one
- use it in runtime environment as an additional dependability technique
- usage in virtual networks (also hosts can be allocated)

Appendix B: Expert panel session

This appendix collects the material presented in the expert panel session "Dependability and Security in Complex and Critical Information Systems", under the working group meeting "Security and Trust".

Session moderators:

André Cotton is head of TAI laboratory in Thales Communications, Colombes, France. Engineer in physics, he build a large experience in networked mission-critical information systems in several entities of Thales group, through positions of W.P. manager, project manager and technical director in companies addressing ground transportation (metro, bus), energy networks (gas, electricity) and management applications for telecommunication operators.

He contributed in several European research projects, was coordinator of (FP6/IST) SEINIT integrated project and is presently coordinator of (FP6/IST) DESEREC project.

Dr. **Antonio Lioy** holds a M.Sc. degree (summa cum laude) in Electronic Engineering and a Ph.D. in Computer Engineering. Currently he is Full Professor at the Politecnico di Torino, where he leads the TORSEC research group active in information systems security. This group has taken part to several international security projects (such as ICE-TEL, ICE-CAR, GUIDES, AIDA, TESI, NASTEC, SecurIST, POSITIF, DESEREC, and Open-TC).

His current research interests are in the fields of network security, PKI, and policy-based system protection.

Figure 3: Some photos taken during the panel session

Security & Dependability in Complex & Critical Information Systems

Speaker: **Michel Riguidel**

Ecole Nationale Supérieure des Télécommunications, ENST, France

Dr. Michel Riguidel is the Head of the Department of Computer Science and Networks, at ENST (Ecole Nationale Supérieure des Télécommunications) in Paris, where he lectures in security and advanced networks.

His research is oriented towards security of Information Systems and Networks, architecture of communication systems (Grids, Next Generation Internet, Active, Ad hoc and configurable Networks), software radio and protocol engineering. He has headed-up many R&D Projects in security: watermarking, PKI, formal methods, critical infrastructure protection.

Currently, he is working on security of heterogeneous and mobile systems within Ambient Intelligence and Quantum Networks.

Security & Dependability in Complex & Critical Information Systems

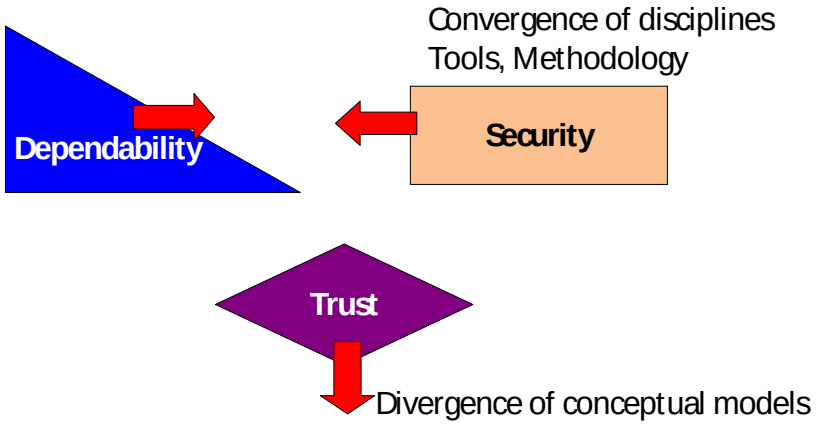
michel.riguidel@telecom-paristech.fr

École nationale supérieure des télécommunications

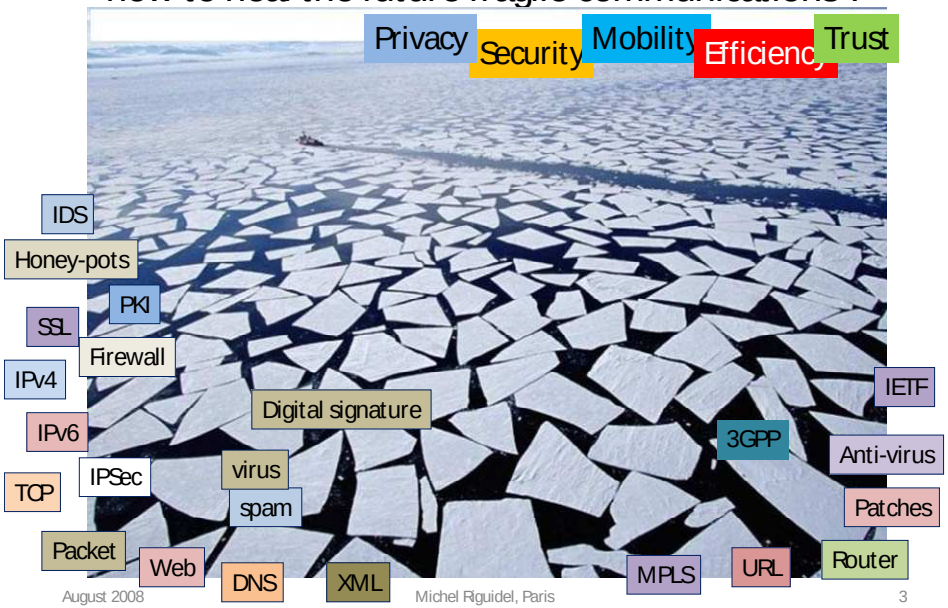
Depend 2008, Cap Esterel, August 27, 2008

Plates tectonics : Continental drift

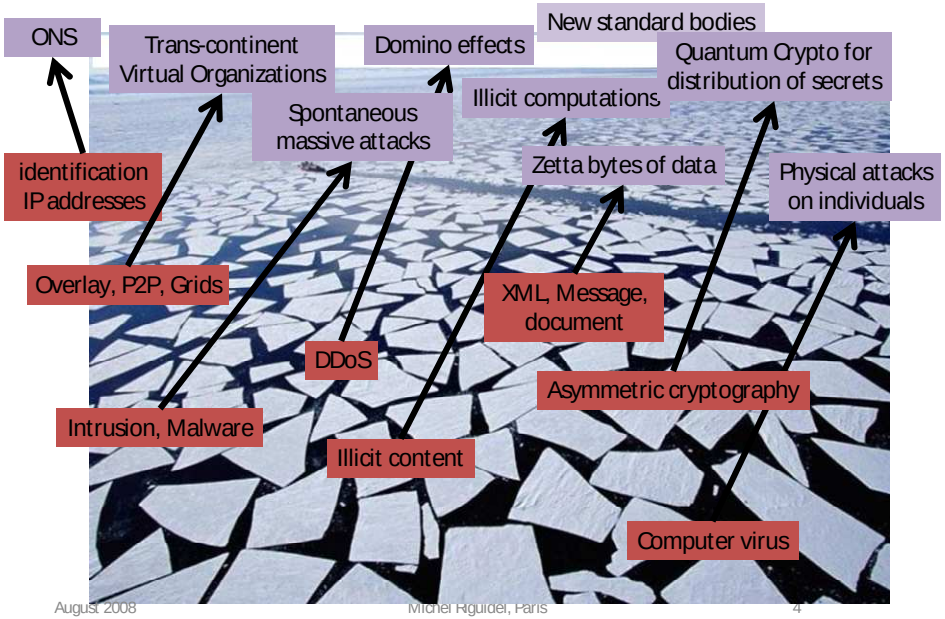
for scientific disciplines



Internet is broken : how to heal the future fragile communications ?

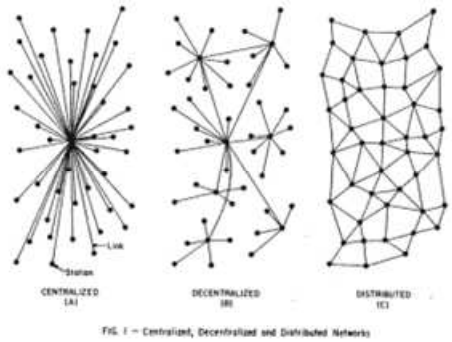


The new landscape in security



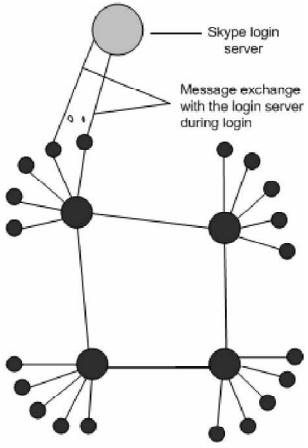
Hidden Web, Deep services : death of the 7 layer model

Old Internet :
flat architecture



First Drawing of Internet (1962)

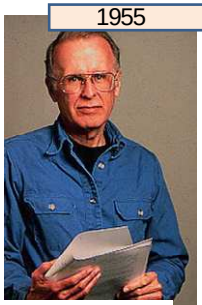
Future Web :
distributed "aggressive" services



Example : Skype architecture

Languages : complexification of abstract typing

1955 1958 1965 1986 1992



Floats, Integers Independent within the memory

John Backus

Fortran

List : organization of the memory with strings

John Mac Carthy

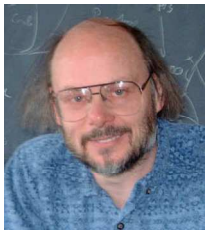
Lisp

Pointers: Static Structures

Dennis M. Ritchie

C

Pointers : objects, dynamic, programs

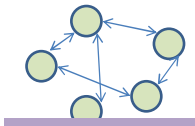


C++ Bjarne Stroustrup

Java

Networks : Complexification of abstract typing of links

1960-2000 | 2000-2010 | 2010-2020 | 2020-2030



Traditional network : nodes and links

Asynchronous
Text
Text Mining
Scalable protocols

Google
Web

ubiquity

Overlay network, P2P : Introduction of topology BitTorrent, Chord

Geography
Multimedia

Ontologies

Virtual tier networks : Static structure of personalized nodes

History
memory

Semantic dynamic networks : Programmable structures

Knowledge

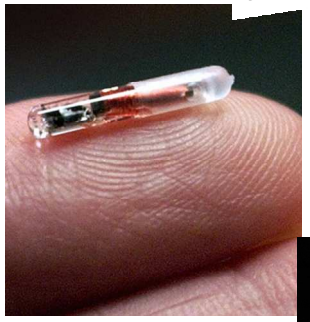
Regular: artefacts for individuals & enterprises

Traditional security to be improved and revisited (architecture & protocols):
Classical Cryptography
Engineering Security

Dwarf : tiny program, simple artifact, scarce resource

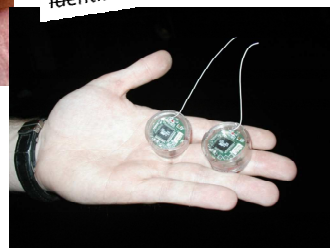
TINYOS

The digital world is
neither fractal
nor scalable …
For tiny objects,
Emergence of
self-* models
at the collection level



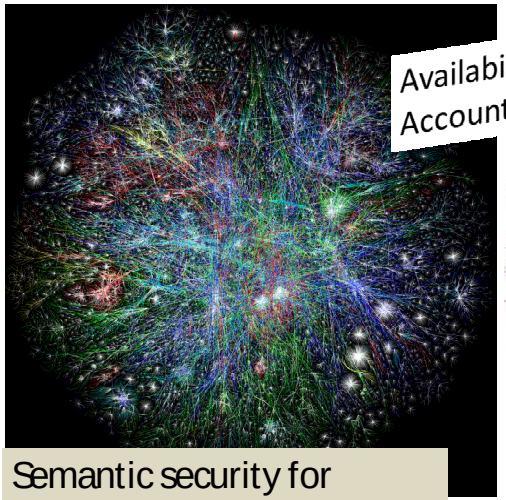
Traceability

~~Identification & Authentication~~

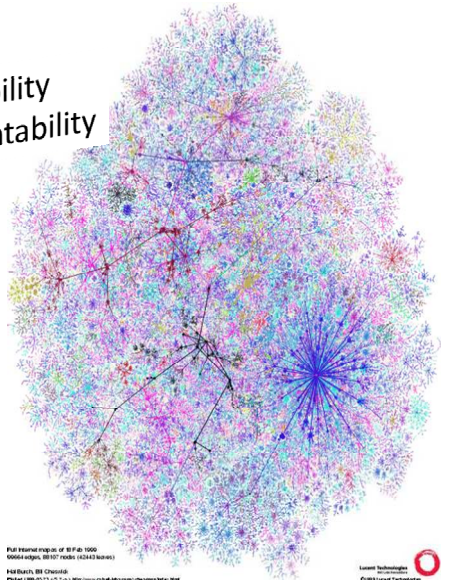


Stochastic Security
Strong security at the collection level (architecture)
Cheap weak security at the individual level
(massive & simple algorithms)

Huge, Giant : Complex systems, inextricable problems



Availability
Accountability



Semantic security for
Complexity & human values
Trust

Digital hybrid urbanization : holistic future Internet

No more monochromic, mono-technology security

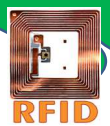
Beyond 3-4G

Current Internet
WDM-IPv4-IPv6-MPLS

Services
Hooked to several
infrastructures

Internet of Things

Galileo-GPS-Glasnos
Clock and Position



RFID

Computational Cryptography

Traditional hierarchical ladder of the current internet

Governmental organizations

amateur Hacker

Re-equilibrium of forces within the future internet (attacks, cryptanalysis)

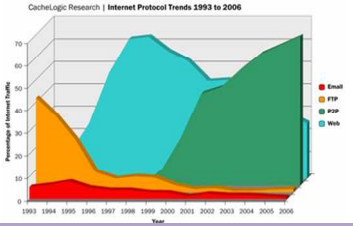
Massive externalized
furtive computer power
running within the
anonymous networks
confidential illicit computations

Standalone end-user

New Crypto with computation, history and geography ?
Alice and Bob are no more alone in this world:
They have witnesses, alibis, trajectories
They leave traces …

A science of the Web : technical challenges

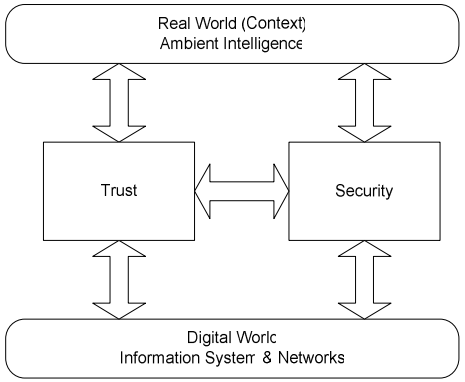
- **Old Web : importance of the underlying protocols**
- computers connected (Web pages, Web sites)
  - 1986 : ancient Web : (Wide Electronic Board)
  - 1991 : Web (Berners-Lee)
    - Web with text
    - 1995 : first success (Java encapsulated)
      - Bandwidth issue (wait-wait-wait)
  - 2000 : high data rate
    - Web with Multimedia
    - Web2 (Multiparties, Virtual), "Semantic" Web, …



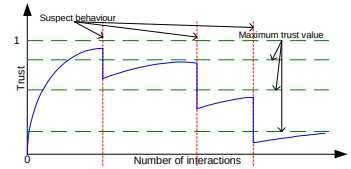
Catastrophic event in the protocol world in 2000 : Web Http decreases, P2P protocols raises drastically

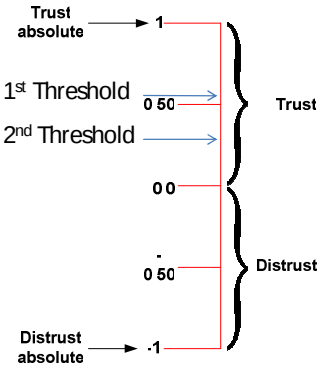
- **Future Web : importance of mobility, context and reinvestment of Humans & Reality**
- Computers (Mobile, Multimedia), networks at large, within physical world
  - profound evolution in parallel with Future Internet
    - Geography : Mobility, Ubiquity
      - Reconciliation with nomadicity (vocal Web)
      - Search engines with locality, smarter search engines : Post-Google engines
    - History : Memory of the web (Next Generation of the Deep Web, Hidden Web)
      - Stochastic XML (see P Senellart PhD Thesis December 2007, Paris)
    - Knowledge
      - Representation, Visualization
      - Search engines, Social computing, Natural language technology
  - Web of intentional Things
    - Things will display their public life cycle, will blog (for maintenance)

Security & Trust



Dissociation between both Infrastructures/Instrumentations of Trust & Security/Dependability





Trust Continuum
1st Threshold to modify behavior
2nd Threshold to stop interacting

Intercontinental Thought : new models

beyond an idyllic, pre-scripted vision of future networks

- Neither unique nor providential solution
  - Model, **counter-model, alter-model**
  - Arrival of China and India on the IT scene
    - change in concerns (demographic, development)
    - change in power
  - Alter-models
    - the pseudo-libertarians ("Naives of the Internet")
    - repression pure players (some governments).
- Cyberspace & Cyber-governance (Κυβερ : rudder)
  - Technology free rein?
    - descriptive of order
      - no control and regulation becomes self-reflexive
    - normative of order
      - governing is defining order
  - In technologies, we talk about
    - often : what is
    - not often : what ought to be or what could be

ICT Security and Trust Research Reflection Group

Advisory Board on Research and Innovation for SEcurity, Privacy and Trustworthiness in the Information Society (RISEPTIS)

- RISEPTIS Advisory Board will be to provide visionary guidance on policy and research challenges in the field of security and trust in the Information Society. It will do so by formulating recommendations on:
- **Policy environment** – The development of coherent legal and administrative
  - frameworks, operational environments, and human behaviour relating to security, privacy and confidence, in view of the technological changes leading to and arising from the future Information Society,
- **Research Agenda** – Future European research and development that can facilitate
  - the creation of an Information Society that will be secure, whilst respecting freedom and privacy of its citizens, with due attention given to the ICT infrastructures, networks, services and applications.

Future research challenges in dependability

Speaker: **Aljosa Pasic**
Atos Origin, Spain

Dr. Aljosa Pasic is the Head of department, Atos Research & Innovation. Aljosa Pasic graduated Information technology at Electro technical Faculty of Technical University Eindhoven, The Netherlands. From 1996 to 1998, he worked for Cap Gemini (Utrecht, The Netherlands) and in January 1999, he joined Sema Group Spain (now part of Atos Origin).

His current position is Head of Department within Atos Research and Innovation, where, during the past ten years, he coordinated and participated in a number of security related EU projects from FP4, FP5, FP6 and e-Ten programmes. He is also the chairman of Trust, Security and Dependability Working Group of NESSI (European Technology Platform for Software & Services).

NETWORKED EUROPEAN SOFTWARE & SERVICES INITIATIVE

Future research challenges in dependability - an industrial perspective from NESSI

Aljosa Pasic
Atos Origin sae

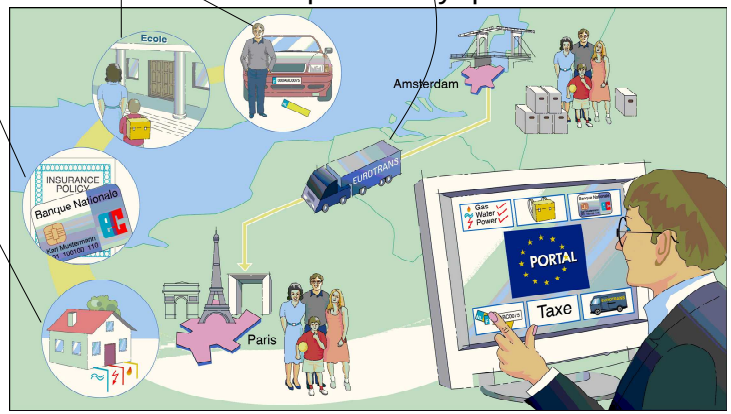
IARIA workshop

Agenda

- The need
- The answer
- The process
- The overview
- The challenges

Service Oriented World

Applications will need to utilise shared and co-owned services out of different domains of control that require to obey separate security policies and ask for diverse security and dependability qualities

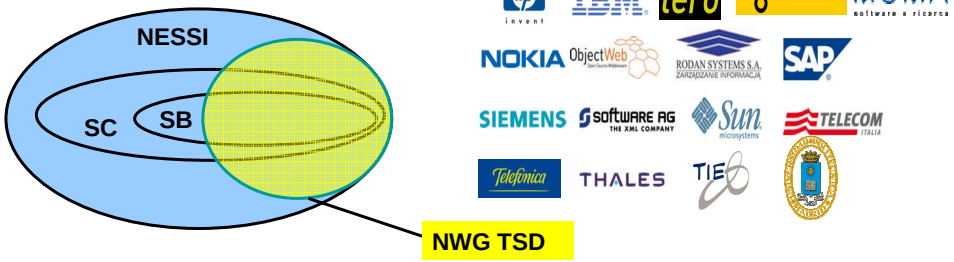


Coming problems

- For industry: Demand for Secure software is much higher than available security expertise
- For research/technology: New complex scenarios (e.g. ambient intelligence) introduce security issues not addressed by conventional engineering processes
- For market consultants: Security properties difficult to measure and it is also difficult to evaluate their “compositional effects”
- For users: Security segmentation and market definitions are blurring: “service infrastructure” covers network infrastructure, perimeter, desktop, server and application security
- For auditors and lawyers: Who is accountable and liable for what?
- For society: Trust becomes a “key enabler” for service provision and use
- For everyone: How much should we spend on security?

NESSI WG TSD

European Technology Platform: Networked European Software & service Initiative , NESSI



Objectives

- Address the security and dependability requirements, challenges and priorities of emerging service oriented software applications
- Bridge two communities: the software engineering (services, GRID) community and the security community
- Support the NESSI vision and respond to security-related NESSI challenges
- Address long-term research on trust, security and dependability in software and services

Mapping challenges, concepts and research topics

	Development environment	Operational environment	Business dynamics
Decrease Gap between TSD need & TSD availability	Security patterns (FP6 – Serenity)	Trusted computing (FP6 - Open TC)	Trusted VO (FP6 – TrustCom)
Handle complexity	Simulation & modeling (FP6-Deserec)	Runtime monitoring (FP6 – Sensoria)	Cross-domain scalability (FP6-GRIDTrust)
Improve TSD Decision Making	Proof based (FP6 – Mobius)	Risk based (FP6 – Fastmatch)	Privacy level based (FP6 – Prime)

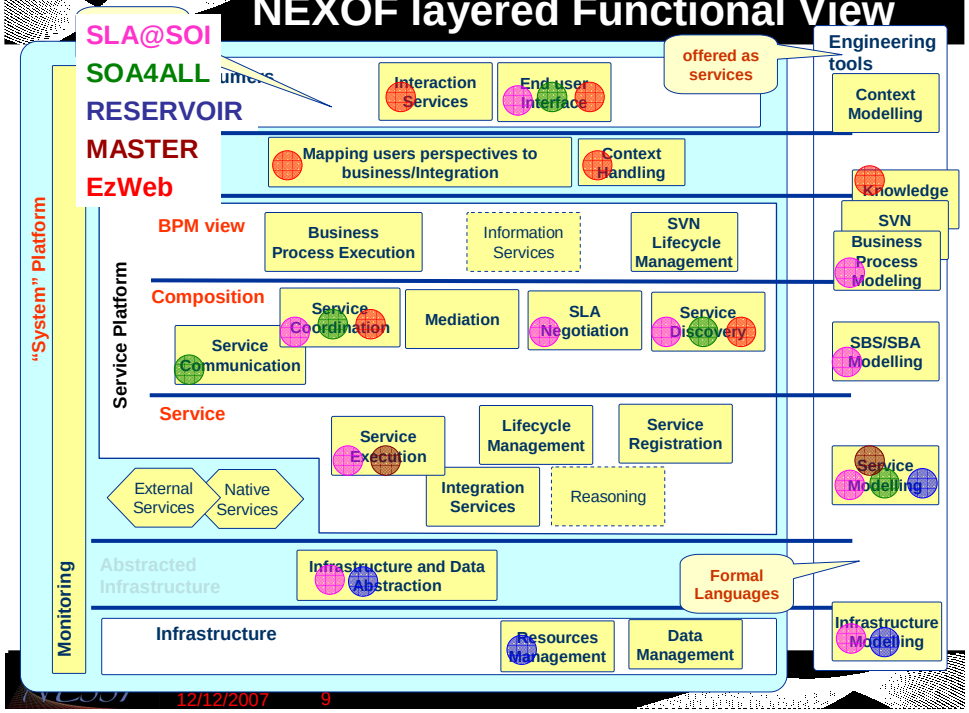
What should it be?

- Dynamic
- Adaptable
- Composable
- Measurable
- Predictive
- Scalable
- Persuasive
- Open
- Trustworthy
- Interoperable

• In TSD engineering and modelling
• In TSD control and management
• In TSD level assurance

Approaches, properties and research challenges

NEXOF layered Functional View



WG subtopics

1. Privacy in services and service oriented architectures
2. Identity manamgement and identity as a service
3. Security policy langauges and mechanisms for services
4. Trust analysis, managment and monitoring
5. End-to-End (E2E) verification of trust, security, and dependability properties
6. Security and resilience engineering for services
7. Security of the Human Computer Interface
8. Dependable architectures
9. Scalable security
10. Security for event-based infrastructures

Enabling the next wave of services

aljosa.pasic@atosorigin.com

Dependability and Security in Complex and Critical Information Systems

Speaker: **Antonio F. Gómez Skarmeta**

University of Murcia, Spain

Dr. Antonio F. Gómez Skarmeta received the M.S. degree in Computer Science from the University of Granada and B.S. (Hons.) and the Ph.D. degrees in Computer Science from the University of Murcia Spain. Since 1993 he is Professor at the same department and University. He has work on different research projects in the area of security and dependability. He was also coordinator of a Socrates CDA (European Master on Soft Computing) and a Leonardo project for Distance and Open Learning.

He is currently collaborating in two IST projects related to dependability and security POSITIF and DESEREC. Additionally he had been collaborating in two IST projects related IPv6 Euro6IX and 6Power where several advanced services like multicast, multihoming, security and adaptative multimedia applications are being deploy in IPv6 networks.

The First International Workshop on Dependability and Security
in Complex and Critical Information Systems
DEPEND 2008
August 25-31, 2008 - Cap Esterel, France

Panel Session: Dependability and Security in Complex and Critical Information Systems

Antonio F. Gómez Skarmeta
University of Murcia
Spain

Department of Communications and Information Engineering

University of Murcia (Spain)

Challenges

- Information System Management increase the complexity of the models to work with in relation to classical network management systems and PBNM
- Need for flexible and powerful models for service provision, data governance and system configuration allowing for affordable management functions
- Trust and dependability can not be built in isolation neither can it rely on technologies of one layer only, there is need to provide integrated view and considered some hierarchy of models
- Interoperable (policy-aware) architectures from trusted communications to the application, including to the data; Web Service technologies to support interoperability.
- Usage of a common models and abstraction of the underlying system providing and overlay view of IT the business objectives behind.
- Architectural design to support the flexibility and the dynamicity of the trust management in a vertical approach from application to the network and data layer.

Challenges

- Dynamic in the trust relations to allow flexible and spontaneous trust relation to be created based on the communications needs. Confederation concepts
- There is a need homogeneous naming of entities (natural and legal persons, objects, virtual entities, devices, content, processes, applications, etc) and trust attribution schemes (identity management, authentication, authorization, accountability, reputation) .
- Semantic security layer need to be considered as an additional component in the trust relation to allow the possibility to enrich the trust models
- Internet of things implies to be able to associate trust and behaviours tags to the digital objects and their relation

Towards trustworthy ICT service infrastructures

Speaker: **Yves Paindaveine**

Information Society and Media DG, European Commission

Yves Paindaveine is a microelectronics and computer science engineer by education. While at the Open Software Foundation Research Institute, he worked, among others, on distributed systems and security. He joined the European Commission in 1998 as a scientific officer for RTD projects dealing with health informatics (IST Programme). Later on, he joined the Unit "security" headed by Jacques Bus, where he is working on network security and dependability.

DEPEND2008

Cap Esterel, August 27, 2008

Towards trustworthy ICT service infrastructures

Yves PAINDAVEINE

Directorate General Information Society and Media

Unit F5 Security

European Commission

Yves.Paindaveine@ec.europa.eu

Content

Security in the European Commission

ICT research on Trust and Security

- Current FP7 activities
- Future priorities

The European Union

27 Member States

23 languages

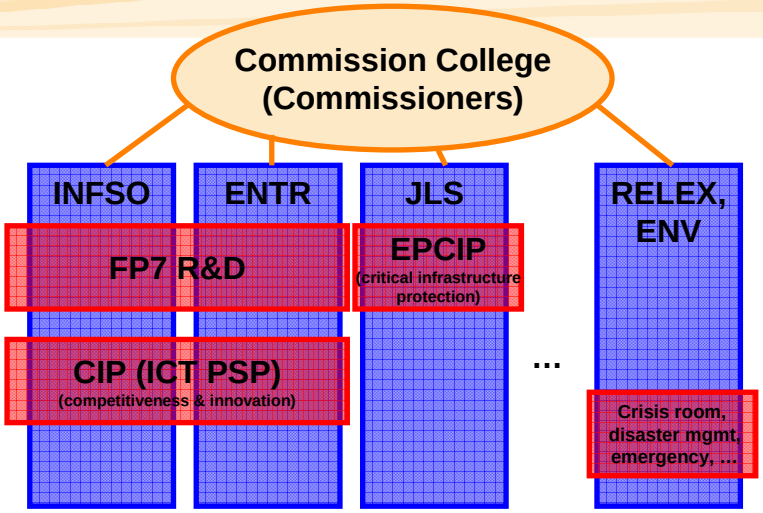
4.3 mio km²

500 mio persons

11 trio EUR GDP
(16.6 trio US$)

USA	Japan	China
50 states	-	-
9.8 mio km²	0.38 km²	9.6 mio km²
302 mio	127 mio	1320 mio
13.8 trio US$	4.4 trio US$	3.4 trio US$

Security in the European Commission

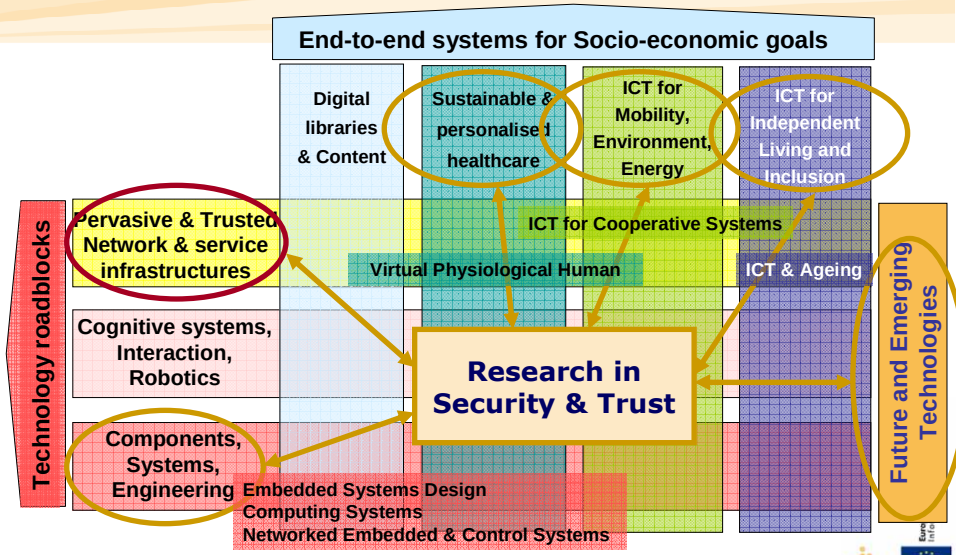


Content

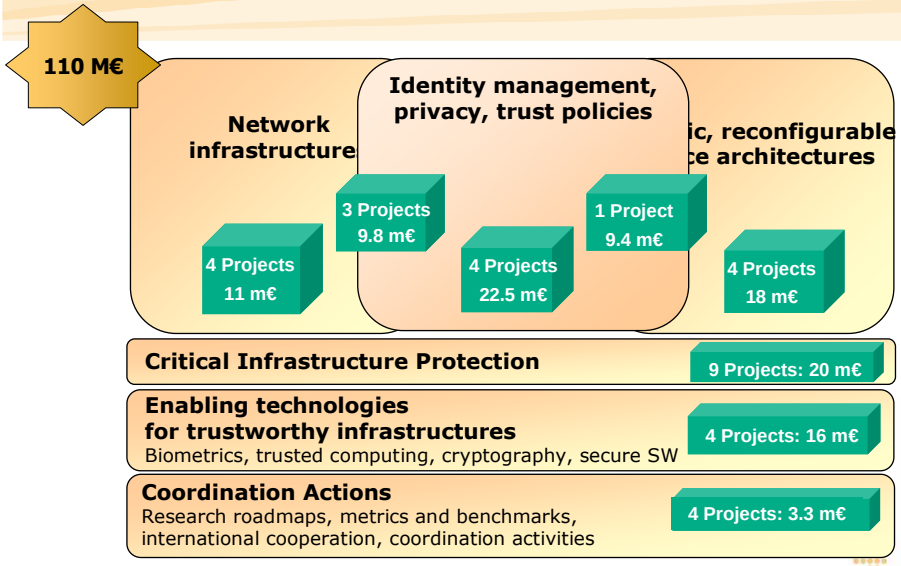
ICT research on Trust and Security

- **Current FP7 activities**

ICT Work Programme 2007-08
Security & Trust in perspective



ICT Work Programme 2007-08
33 new FP7 projects in Security & Trust



Content

ICT research on Trust and Security

- **Future priorities**

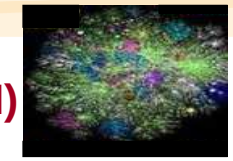
DRAFT

General Principles for the FP7 ICT Work Programme 2009-2010

Principles

- Continuity of Challenges from WP07-08
- Mid-term research: most impact on markets in 2015-2020 timeframe
- Some Priorities
  - Future Internet
  - Beyond CMOS (nano, photonics, organics, quantum, molecular)
  - Two JTIs (ENIAC - nanoelectronics, ARTEMIS – embedded systems)

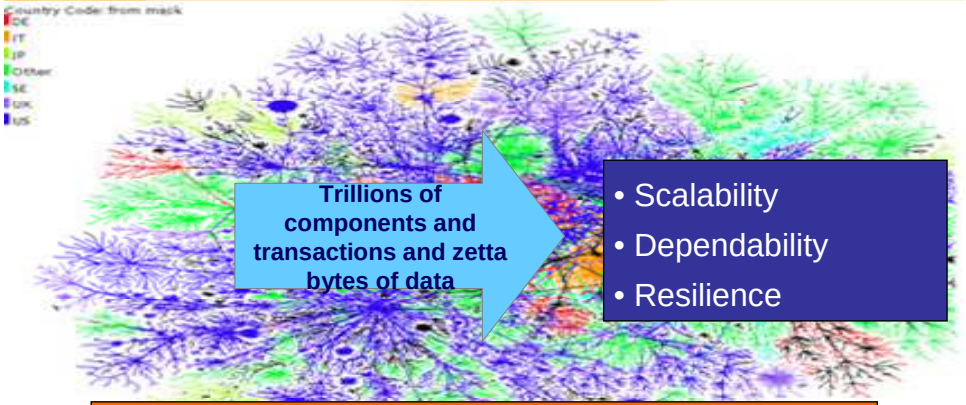
Challenges for RTD for a Trustworthy Information Society



- **A *trustworthy* Future Internet (FI)**
  - Built-in security, dependability and trust
  - Enabling users to understand security, privacy, trust in the FI

- **… and three closely inter-related research perspectives**
  - Securing the architecture of FI
  - Protection against emerging threats
  - Privacy, IDM and Trust in the FI Society

Future Internet: Complexity!



Trillions of components and transactions and zetta bytes of data

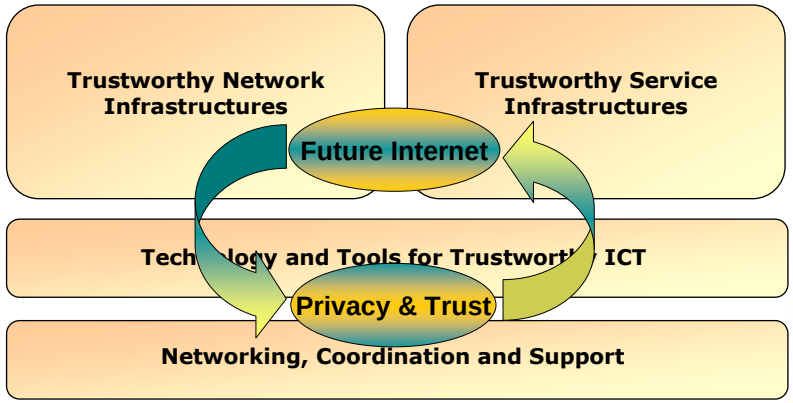
- Scalability
- Dependability
- Resilience

Collaborative Security!
End-to-End security and trust
in highly complex networks and services!
Non-functional requirements (trustworthiness) part of the design and construction

DRAFT

Trustworthy ICT in WP09-10

Trustworthy ICT ➔ Secure, reliable, resilient, surviving attacks, guarantee a desired level of service, protects user data and privacy, providing usable and trusted tools to support the user



DRAFT

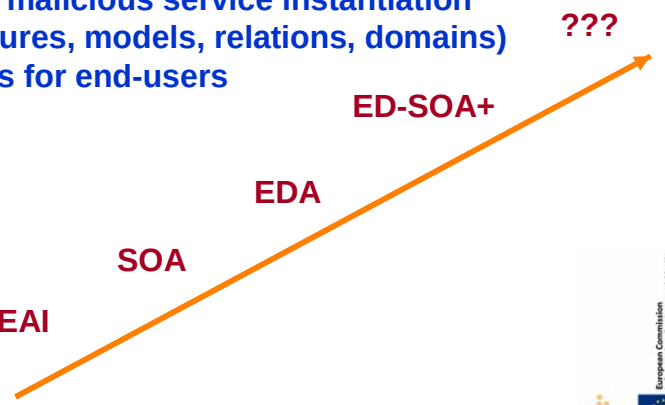
Trustworthy Service Infrastructures ICT in WP09-10

- Privacy protecting interoperable services on the FI
- User-centric, privacy respecting ID for persons, things and virtual entities
- Adaptive frameworks for managing trust throughout life-cycle
- Experiments and demonstration
- Attention to usability, social acceptance, human self-determination and privacy, economic and legal viability

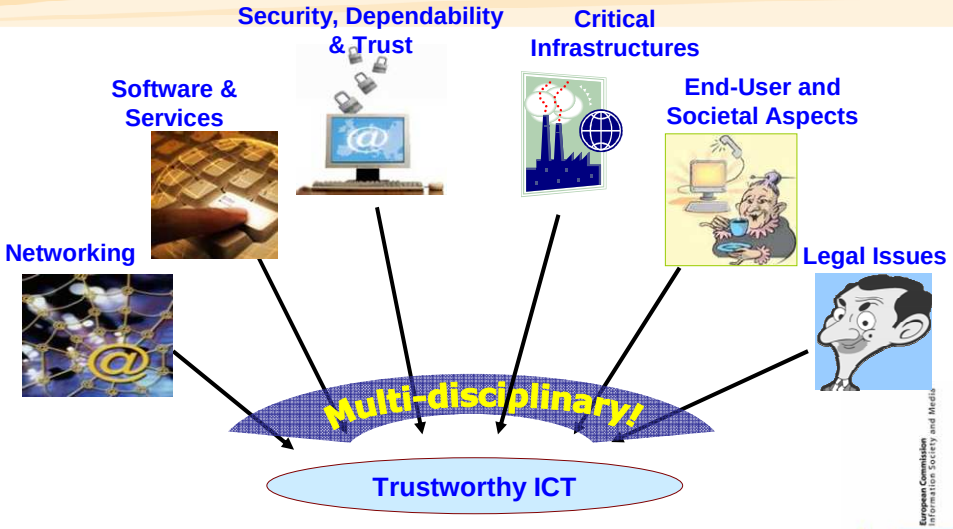
Towards Dynamic Service architectures

How to

- **End-to-end runtime security properties**
- **Legitimate ←→ malicious service instantiation**
- **Trust (architectures, models, relations, domains)**
- **Trustworthiness for end-users**



Trustworthy ICT Research Stakeholders



Activities and initiatives

- **RISEPTIS AB → ThinkTrust WGs**
  - **Security and dependability in future networks and services**
  - **Privacy & Trust**
- **INCO, Incotrust CA**
- **ICT 2008 event in Lyon (25-27 November)**
  - → http://ec.europa.eu/information_society/events/ict/2008/index_en.htm
- **FIA, Future Internet Assembly, 9-10 Dec 2008, Madrid**

IncoTrust: http://www.inco-trust.eu/
ThinkTrust: http://www.think-trust.eu/

DRAFT

Timetable for WP09-10

– Oct	Discussion in ICT-C, other consultations finalisation
25-27 Nov	Presentation in ICT Conference in Lyon (FR)
~ Apr 09	Closure Call 4
~ Oct 09	Closure Call 5 (Trustworthy ICT)
~ Febr 10	Closure Call 6

http://cordis.europa.eu/fp7/ict/security/home_en.html

Becoming an expert

- https://cordis.europa.eu/emmfp7/