# Challenges of Security Risks in Service-Oriented Architectures

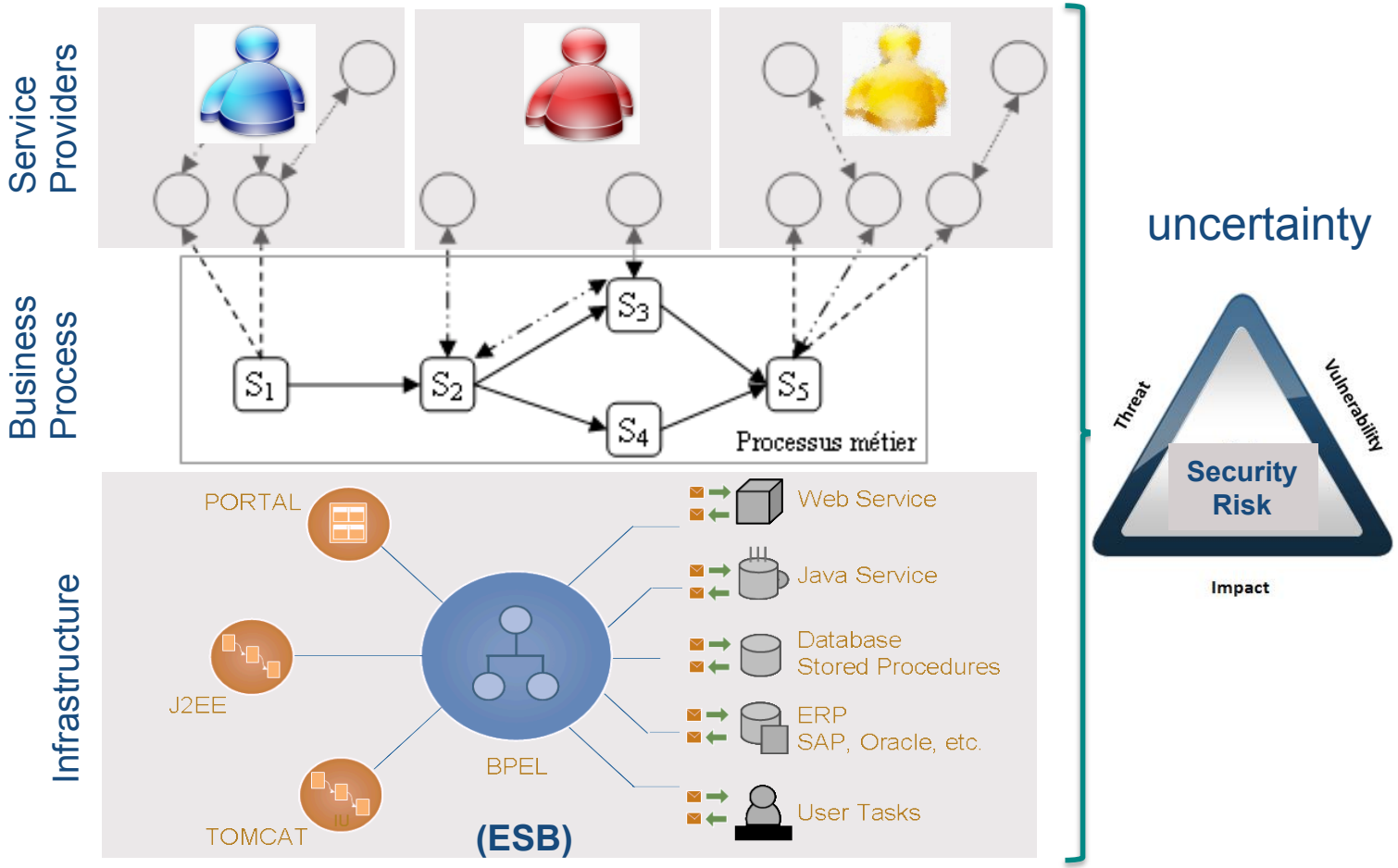
**Youakim Badr**[1], Frederique Biennier[1], Pascal Bou Nassar[3], Soumya Banerjee[2]

[1] LIRIS Lab, INSA-Lyon, France
[2] Agence Universitaire de la Francophonie (AUF)
[3] Birla Institute of Technology, Mesra, India

# Outline

- Motivation Example
- Challenges:
  - Managing security in opened, dynamic, and distributed environments
  - Handling unforeseen threats and deciding on security treatment strategies
- Contributions:
  - Security aware SOA design method
  - Dependency Model and Security Service Reference Model
  - **Design time**: Security Support-decision system
  - **Runtime**: Security Monitor system
- Conclusion and perspectives

# Motivating Example:
## SOA and information security in opened and dynamic environments




**o - Information security** : Confidentiality, Integrity, Availability, Accountability, Assurance, Non-repudiation, … *h*

# Web Service Security

## Web service Security Standards

- Application layer: SAML, ebXML, XACML, XML Firewall, …
- Messaging layer : SOAP, WS-Security, XML-Signature, XML Encryption..
- Transport layer: TLS/SSL, HTTP. FTP, SMTP, TCP/IP, …

## XML specific attacks

- oversize payload, coercive parsing, XML injection, WSDL scanning indirect flooding, SOAPAction spoofing, BPEL state deviation, middleware hijacking, …

## Security aware SOA infrastructures?

# Challenges

- Existing SOA design methods
  - provide little attention to integrate security concerns in reference models, guiding each stage of the **service lifecycle** (i.e., design and runtime)
    - **Reference Models:** (OASIS) reference architecture, (Open Group) SOA Ontology, …
    - **SOA Design Methods**: SOMA, SOAD, CBM, SOAF, SODM, …
- SOA security solutions
  - often limited to services, composition mechanisms and technical implementation
  - underestimate the **(opened & dynamic) environment** by which SOA-based applications collaborate and exchange information (=>end-to-end security)
- Need for ***security risk management***
  - **Security Management :** define global and coherent security policies
  - **Risk Management :** OCTAVE, EBIOS, CORAS, SNA,…

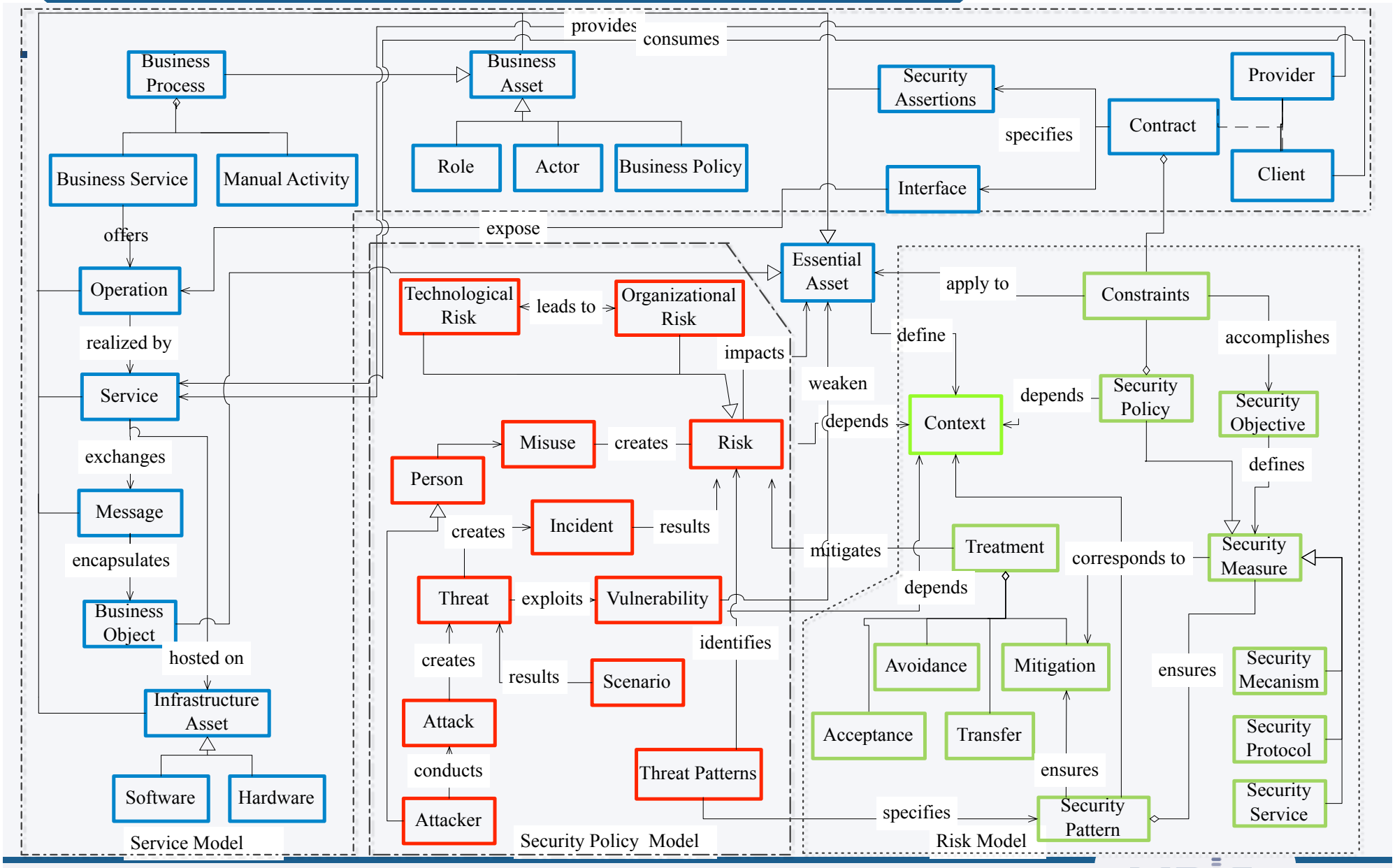
# Contribution: Security aware SOA Design

- The Security Risk-driven SOA Design Method addresses information security in the SOA from a risk management perspective (...) at design time and runtime

- Cycle de vie
  - The Preparatory Stage
  - The Design Stage
  - The Execution Stage

- Outcome:
  - key models, tools and deliverables in each step to progressively identify business goals, essential assets, and services

# Security Service Reference Model



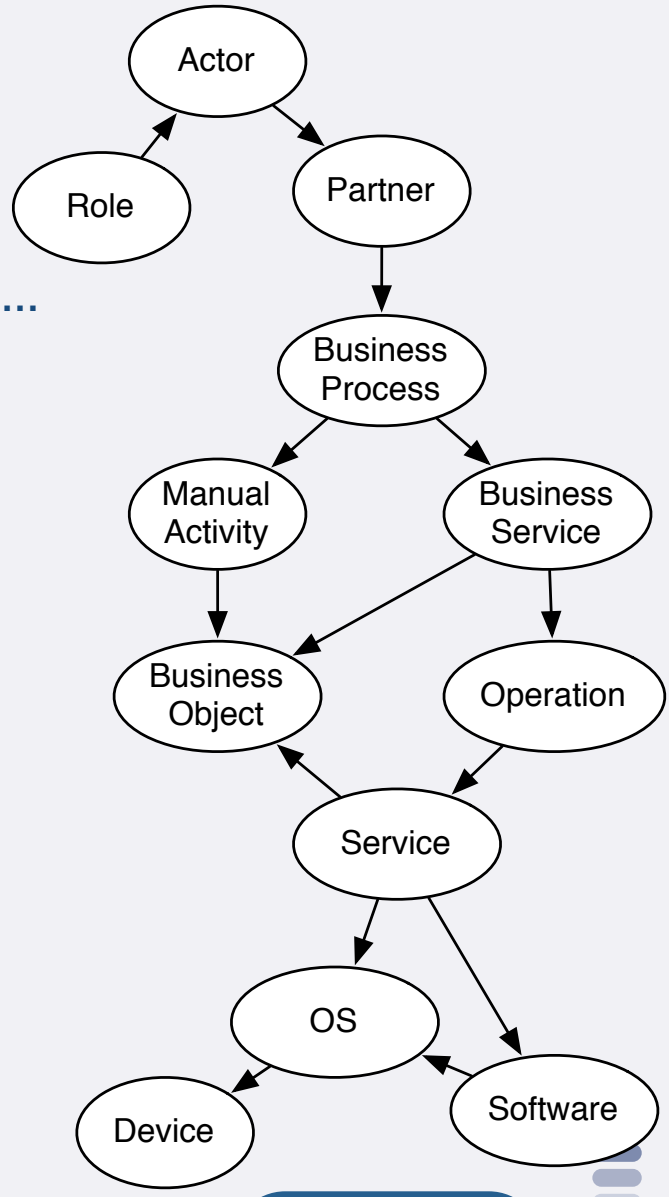
# Dependency Model

## Essential Assets for the SOA design context

- **Business Assets**
  - business processes, documents, partners, actors, roles, …
- **Service Assets**
  - atomic & composite services, operations, messages, …
- **Infrastructure Assets**
  - hardware, software, network protocols, …

## Building the Dependency Graph

- **Bayesian Networks learned from surveys**

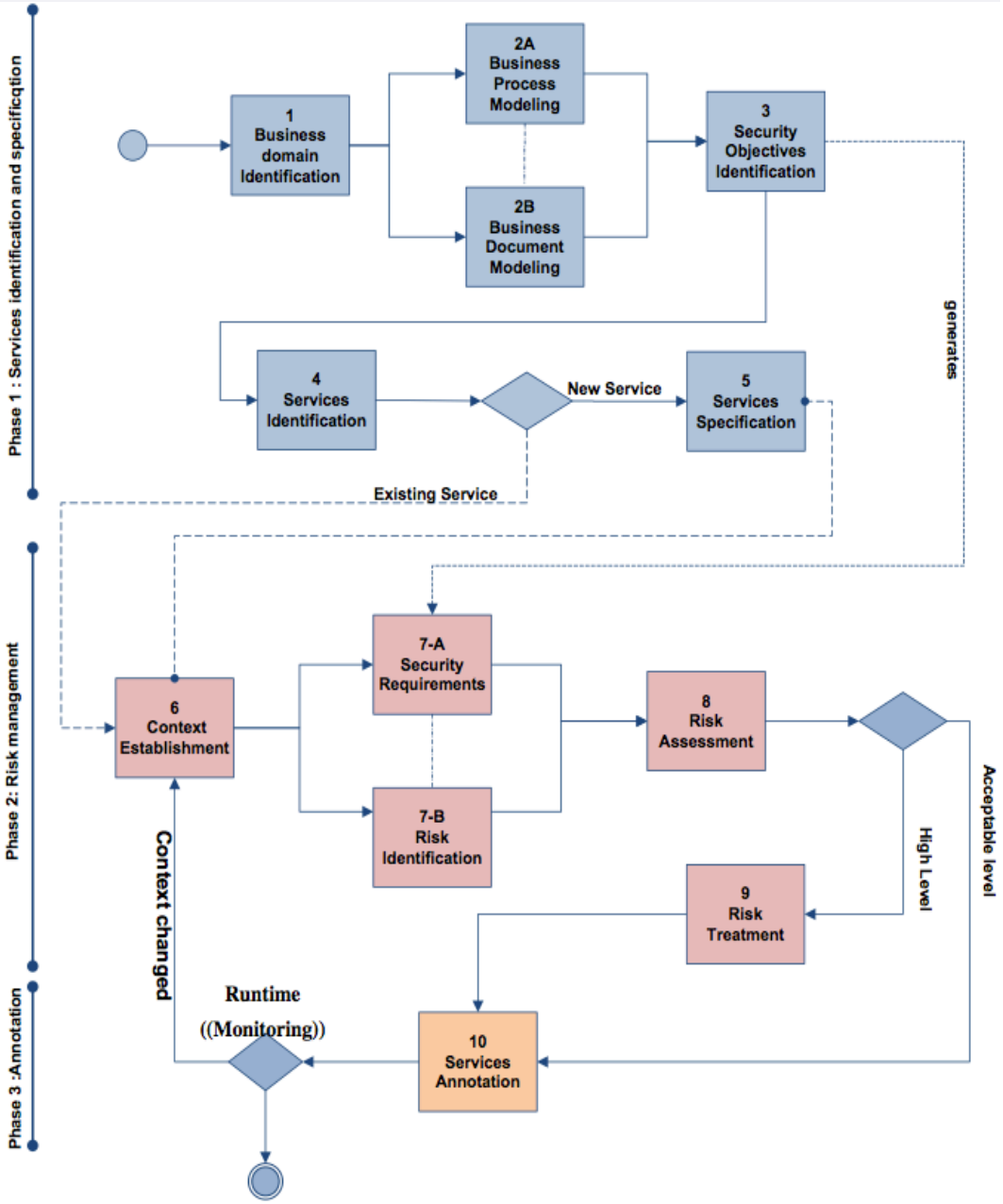


# The SOA Design Method Lifecycle

1- The Service Identification and Specification Phase

2- The Risk Management Phase

3- The Annotation Phase



# The Service Identification and Specification Phase

- 1: Business Domain Identification
- 2A: Business Process Modeling
- 2B: Business Document Modeling
- 3: Security Objectives Identification
- 4: Service Identification
- 5: Service Specification

| Steps | Tasks | Deliverables |
|---|---|---|
| 1 | Identify domain business assets:<br>- what (missions), how (activities)<br>- who (actors), why (motivations) | - Business goals (OMG Business motivation Model)<br>- Business objects and activities<br>- Actors-system interactions |
| 2A/2B | Establish use cases and business processes | - UML use cases<br>- BPMN Business process |
| 3 | Identify business needs and security goals | - Business security goals (EBIOS, OCTAVE)<br>(Confidentiality, Integrity, Availability, ...), |
| 4 | Apply an *outside-in* approach to identify services based on business objects and processes and use cases | **Top-down approach**: manual activities, automated activities (atomic services, composite services, ...)<br>**Bottom-up approach**: legacy and technical services, |
| 5 | Specify service profiles:<br>-Business capabilities,<br>-Functional / non-functional properties | -Service specifications |

# The Risk Management Phase

- 6: Context Establishment
- 7A: Security Requirements
- 7B: Risk Identification
- 8: Risk Assessment
- 9: Risk Treatment

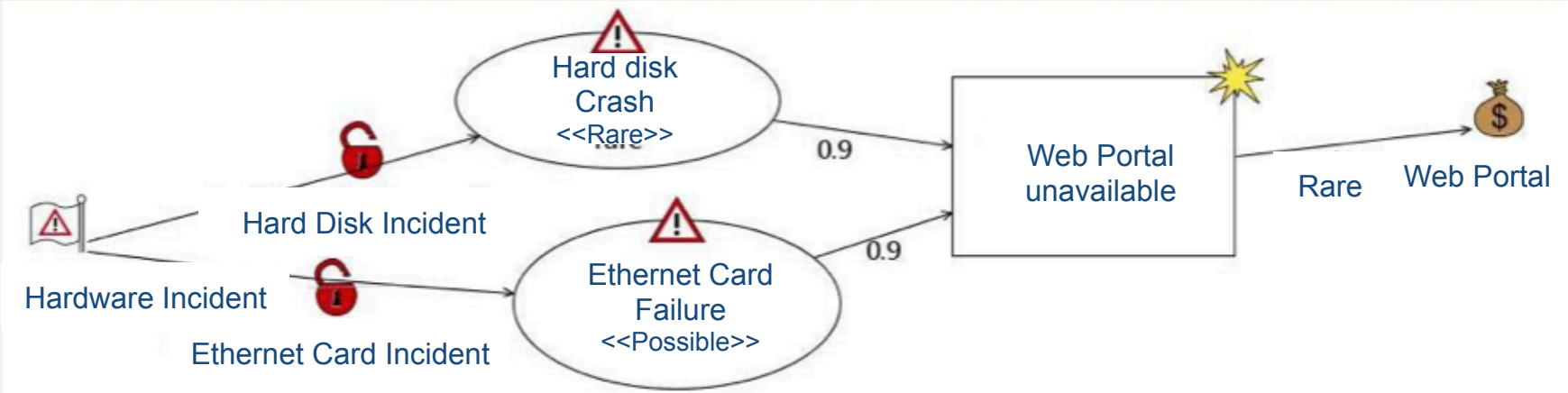
| Steps | Tasks | Deliverables |
|---|---|---|
| 6 | Identify essential assets at business, service and infrastructure | -Essential Assets<br>-Asset contexts: Dependency Graph |
| 7A/7B | - Identify security requirements for each asset based on business security goals<br>-Identify risks related to assets | - Vulnerability list (CERT/MITRE)<br>- Threats list (EBIOS/ OCTAVE)<br>- Security Policy Model |
| 8 | Evaluate risks<br>- Severity of impact<br>- Rate of occurrences | - Risk list<br>- Risk Model |
| 9 | - Prioritize risks<br>- Evaluate security costs<br>- Choose a risk treatment strategy | - Security Policy Model<br>- Treatment strategies:<br>Avoidance, reduction, sharing, retention |
| 10 | Annotate asset security levels with weighted values | - Secure Design ontology<br>- Security annotations (confidentiality, availability, ...) |

# Example: Risk Levels

| | | Severity of Impact | | | |
|---|---|---|---|---|---|
| | | **Insignificant** [0 minute, 30 seconds] | **Minor** [30 seconds, 5 minutes] | **Major** [5 minutes, 2 hours] | **Catastrophic** [2 hours, ∞[ |
| **Rate of Occurrences** | **Rare** [2, 9] :10 hours | **Low Risk** | **Low Risk** | **Low Risk** | *Medium Risk* |
| | **Possible** [10, 19] : 10 hours | **Low Risk** | *Medium Risk* | *Medium Risk* | *Medium Risk* |
| | **Probable** [20, 49] :10 hours | **Low Risk** | *Medium Risk* | **High Risk** | **High Risk** |
| | **Certain** [50, ∞[ : 10 hours | *Medium Risk* | *Medium Risk* | **High Risk** | **High Risk** |

# Example: Availability Threat Scenario

## Web Portal Availability




| | Threat Scenario (1): Web Container Crash | Threat Scenario (2): Router Crash |
|---|---|---|
| **Incident** | Hard Disk Crash | Ethernet Card Failure |
| **Rate of Occurence** | Rare : [0, 1] : 5 years | Possible : [2, 5] : 5 years |
| **Scenario Probability** | 0.9 | 0.9 |
| **Combine Value** | [0, 1] : 5 x 0.9 = [0, 0.9] :5 | [2, 5] : 5 x 0.9 = [1.8, 4.5] :5 |
| **Global Occurrence Probability** | [0, 0.9] :5 + [1.8, 4.5] :5= [1.8, 5.4] :5<br>[1.8, 5.4] : 5 = Rare | |

# Execution Stage

## A Continuous Security Improvement Process

1) From *risk management phase* to *service specification phase*

- Risk high => choose a risk treatment strategy

2) From *runtime* to *risk management phase*

- Context changes => establish the context

## Security Decision-Making System

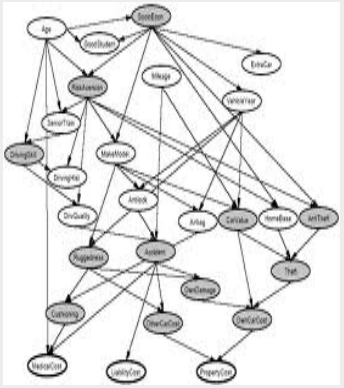
## Service Monitoring System

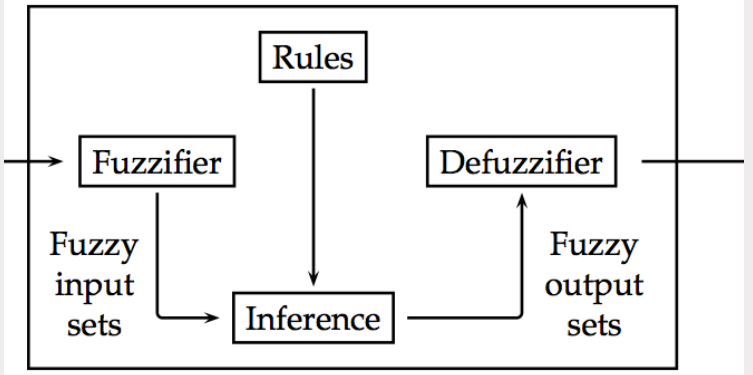
# A Decision-making System for Security Risk Treatments

**Problem**: Deciding on the best risk treatment strategy to deal with threats often relies on *rules of thumb* and often incorporates security analyst's *intuition* and judgment.

**Risk Treatment Decision Process**:
[Threats] cause [Risks] handled by [Security Objectives] resulting in [Security Treatment]

**Fuzzy Logic:**
- Simulating analogy and approximation
- Handling imprecision measures conveyed by the natural language




- Avoidance
- Reduction
- Sharing
- Retention

| SOA Ecosystem | Security Threats | Dependency Graph | Fuzzy Inference System | Treatment Strategies |
|---|---|---|---|---|

# The Decision-making System for Security Risk Treatments
## Fuzzy Variables and Memberships

### 1- Fuzzy Linguistic Variables

T(Essential Assets) = {*Service, Operation, Message, Business Process*}

T(*Vulnerability*) = {*Low, Medium, High*}

T(*Incident*) = {*Random, Regular, Intentional*}

T(*Threat*) = {*Malicious, Accidental, Failure, Natural*}

T(*Security Objective*) = {*Confidentiality, Integrity, Availability, Accountability, Assurance*}

T(Security Measure)={*Encryption, Authentication, SecureTransmission*}

T(*Rate of Occurrence*) = {*Certain, Possible, Probable, Rare*}

T(*Severity of Impact*) = {*Insignificant, Major Impact, Loss*}

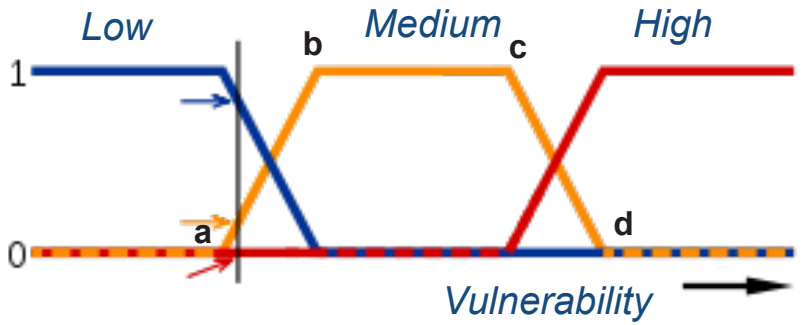
T(*Risk*) = { Low, Medium, High}

T(*Risk Treatment*) = {*Reduction, Sharing, Avoidance, Retention*}

### 2- Membership Functions

$$T(u) = \begin{cases} 0 & u \le a \\ (u\text{-}a)/(b\text{-}a) & a < u \le b \\ 1 & b < u \le c \\ (d\text{-}u)/(d\text{-}c) & c < u \le d \\ 0 & d < u \end{cases}$$

$$0 \le a \le b \le c \le d \le 1$$



# The Decision-making System for Security Risk Treatments:
## *Fuzzy Production Rules*

### 3- Fuzzy rules

$R_1$ IF [Essential Assets] AND [Vulnerability] AND [Incident] THEN [Threat]

$R_2$ IF [Threat] AND [Rate of Occurrence] AND [Severity of Impact] THEN [Risk]

$R_3$ IF [Risk] AND [Security Objective] THEN [Securiy Measure]

$R_4$ IF [Security Measure] THEN [Risk Treatment]

.

### Examples of rules in stage $R_i$, $R_2$, R3 and $R_4$:

$R_{11}$ IF Essential Assets is *Service* AND Vulnerability is *High* AND Incident is *Intentional* THEN Threat is *Malicious*

$R_{21}$ IF Threat is *Malicious* AND Rate of Occurrence is *Possible* AND Severity of Impact is *Loss* THEN Risk is *High*

$R_{31}$ IF Risk is AND Security Objective is *Confidentiality* THEN Security Measure is *Encryption*

$R_{41}$ IF Security Measure is *Encryption* THEN Risk Treatment is *Reduction*

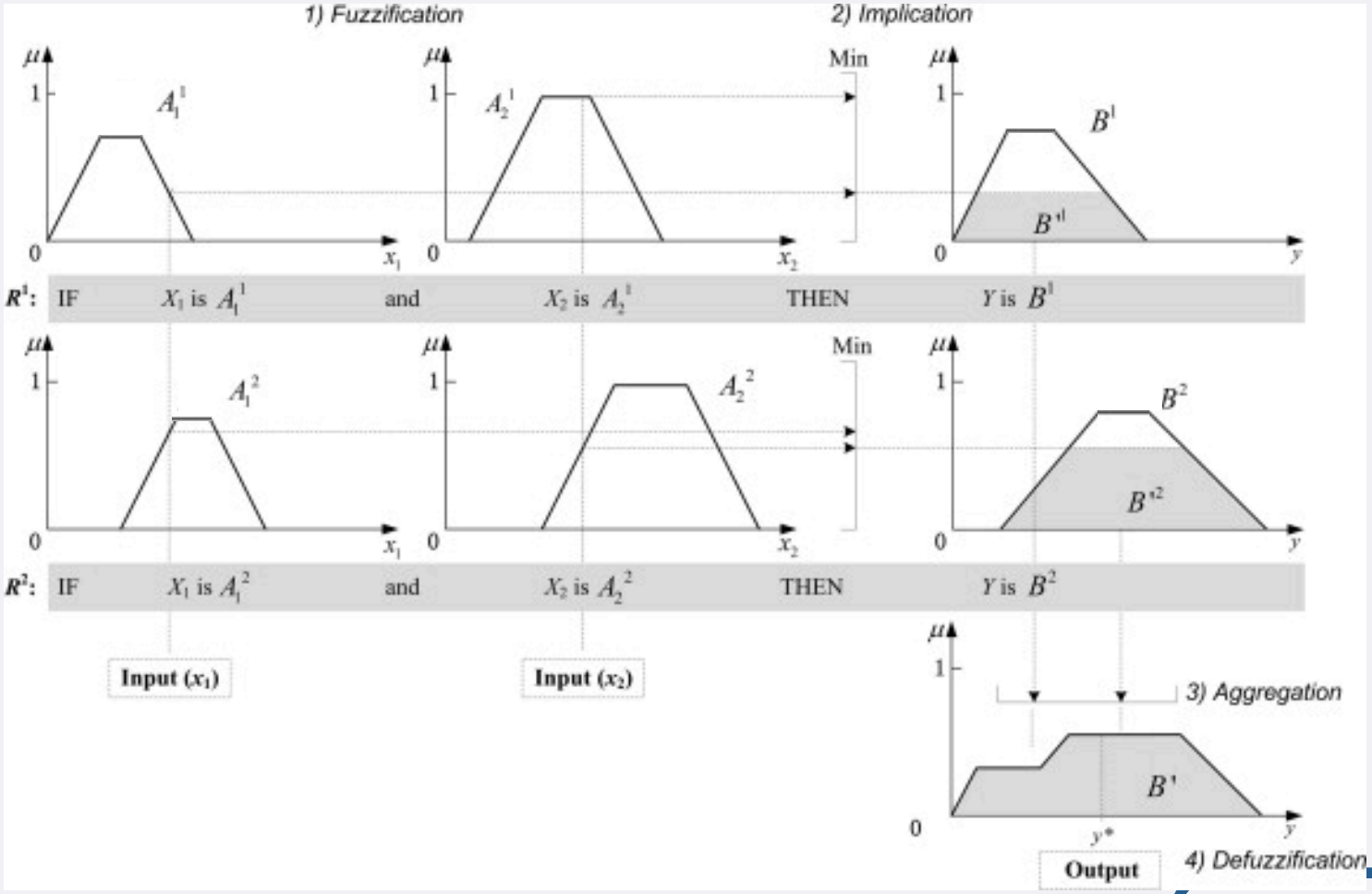
.

.

.

# The Decision-making System for Security Risk Treatments:
## *Evaluation and Inference*

### 4 - Fuzzy evaluation method to propagate multi-stage analysis



# A Service Monitoring System for Vulnerability Detection

**Problem**: Revealing security profiles disclose service weaknesses to potential threats by providing critical information about essential assets

**Security Annotations**: obfuscate security information and enrich service descriptions with a global security level

**Annotation value:** For a service *s* that depends on *n* assets, $x_1$, .., $x_n$

$$V_C = \frac{\sum_{i=1}^{n} x_i \times w_i}{|A_s|} \qquad x_i = \begin{cases} 0 & \text{if } x_i \text{ is vulnerable} \\ 1 & \text{if } x_i \text{ is invulnerable} \end{cases}$$

**Examples: Confidentiality, Availability, Supervision, …**

*Supervision* ⊆
(∀ hasPertinentEssentialAsset.Message) ∧
(∀ hasPertinentEssentialAsset.BusinessObject) ∧
(∀ hasPertinentEssentialAsset.HostingServer) ∧
(∀ hasPertinentEssentialAsset.OperatingSystem)

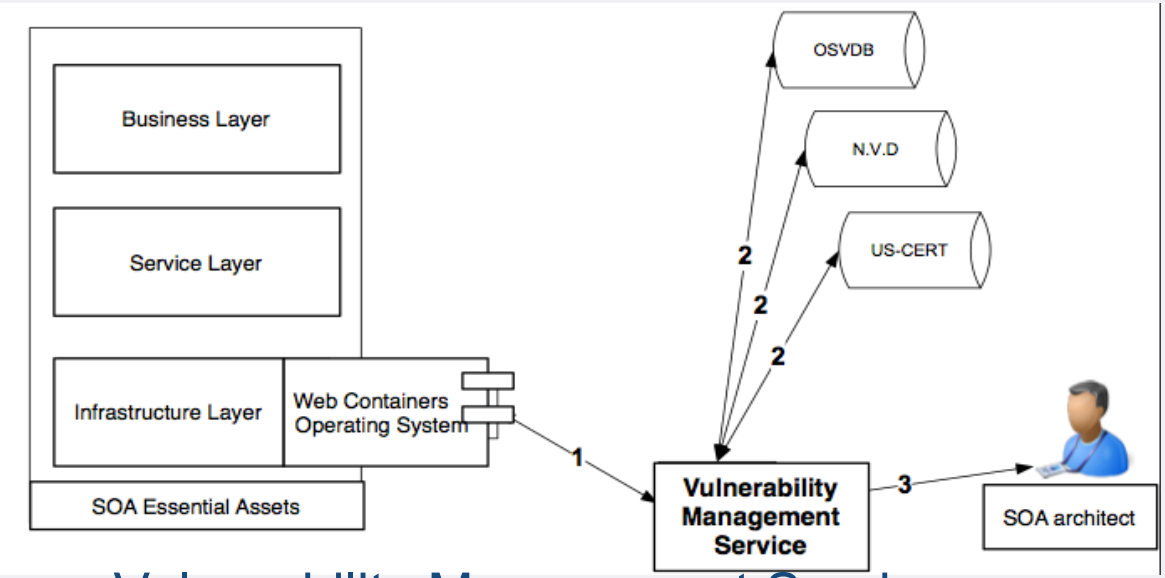
# A Service Monitoring System for Vulnerability Detection

## Public Vulnerability Databases

- National Vulnerability Database (NVD)
- Open Source Vulnerability DataBase (OSVDB)
- United States Computer Emergency Readiness Team (US-CERT)

## The Common Platform Enumeration (CPE)

cpe:/{part}:{vendor}:{product}:{version}:{update}:{edition}:{language}




Vulnerability Management Service

# Thank you

- Questions ?