Service dependencies in information systems

Hervé Debar

Professor, Télécom SudParis

Joint work with N.Kheir, N.Cuppens & F.Cuppens

- **Attackers consistently defeating security systems**
  - Need different tools ?

However

- **Many compromises could be discovered with existing logs**
  - Today’s attacks target sensitive information
  - Sensitive (target) information known « a-priori »

Defense trends

- **Intrusion detection/prevention insufficient**
  - Partial perimeter security
  - Alerts largely unusable (feeling)
  - Security Information Management as compliance
- **Other research activities taking of, looking at the attacker**
  - Cyber Situation Awareness (Cyber SA, ~2000)
  - Cyberwar (~2005)
  - Attack attribution (~2008)
  - Advanced persistent threat (APT, ~2010)
- **Objective: better detection**

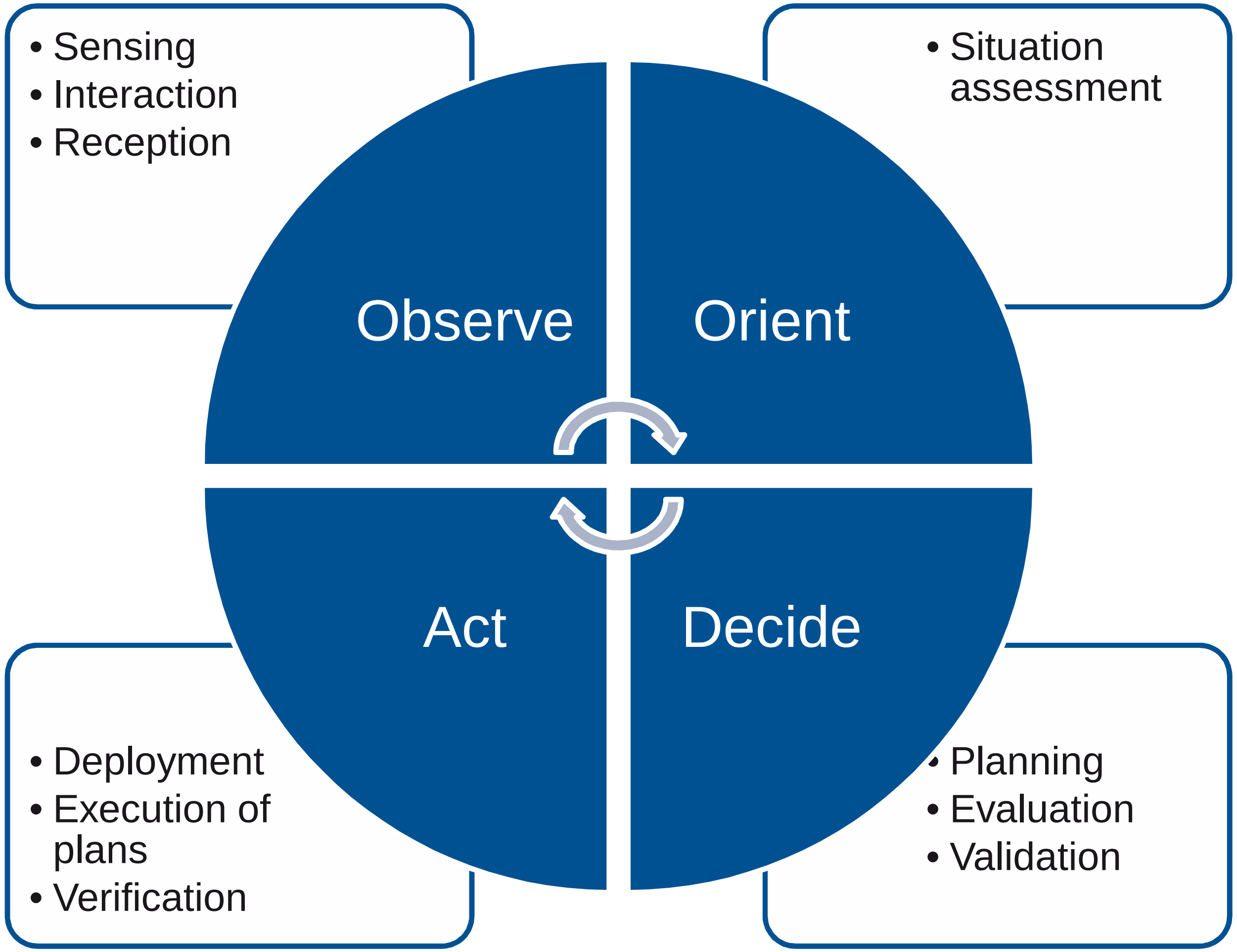
A different objective

- **Security largely statically defined**
  - Design time compromise
  - Monitoring built-in (regulation, etc.)
  - Vulnerabilities & attacks are dynamic
- **What if we could adapt our (limited) resources to the threat**
  - Outside the « security » perimeter
  - Need to process (use) alerts in real-time
- **Move from (cost|security|QoS|useability|…) compromise at design time to compromise at run time**

What is already there ?

- **Dynamic control of networks and services is an established trend:**
  - web service negotiation
  - Cognitive radio
  - Autonomic computing
  - Dynamic firewall rules in VoIP environments
- **Policy-based management**
  - IETF COPS, OPSEC, ...
- **Adaptive *cyber-defense* systems ?**

Background: The OODA Loop (Observe-Orient-Decide-Act)



- Sensing
- Interaction
- Reception

- Situation assessment

- Deployment
- Execution of plans
- Verification

- Planning
- Evaluation
- Validation

Requirements for dynamic security policy management

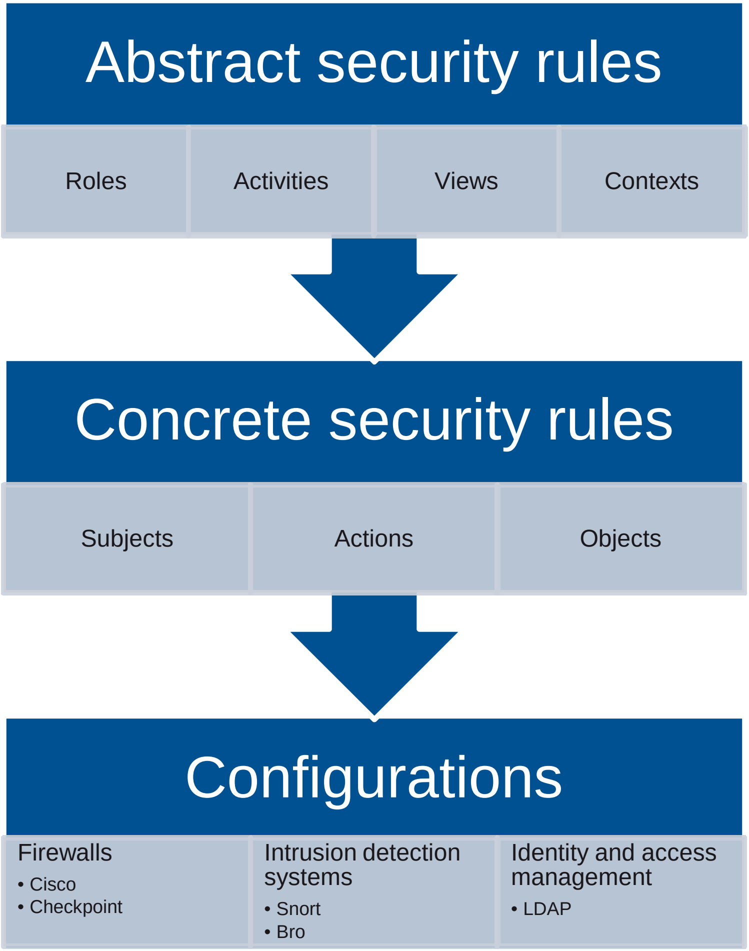
- **Key issue : Assurance that the system behavior is correct**

- **Modern security policy expression**
  - Role-based access control (RBAC)

- **Operational model including enforcement and data acquisition**

The OrBAC model

- **Components**
  - Roles (subjects)
  - Activities (actions)
  - Views (objects)
- **Security rules**
  - Prohibitions
  - Permissions
  - Obligations
  - (priorities)
- **Contexts**
  - Temporal
  - Threat
- **Rule management**
  - Conflict resolution



Abstract security rules

Roles | Activities | Views | Contexts

Concrete security rules

Subjects | Actions | Objects

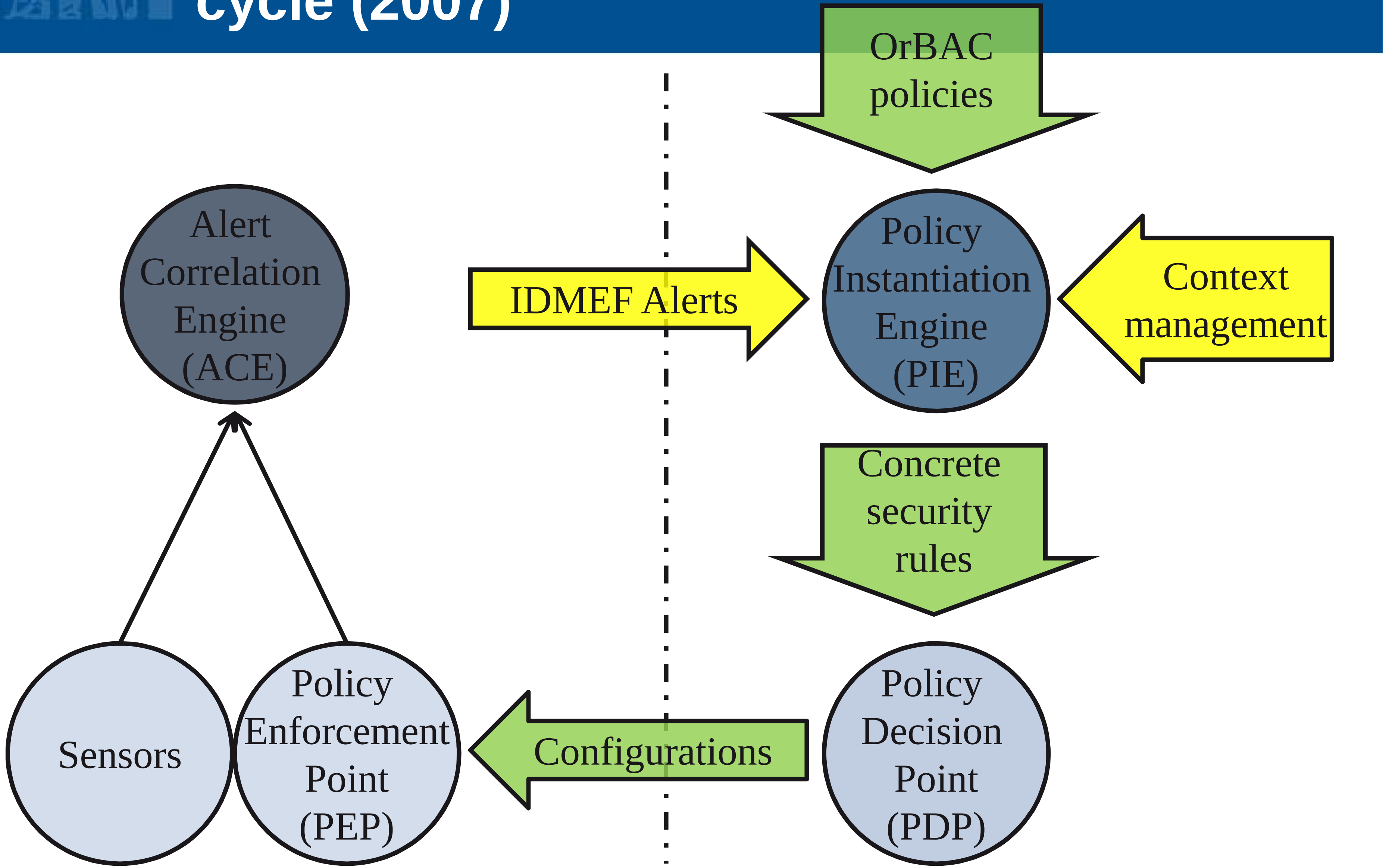
Configurations

Firewalls
- Cisco
- Checkpoint

Intrusion detection systems
- Snort
- Bro

Identity and access management
- LDAP

Operational security cycle (2007)


OrBAC policies
Alert Correlation Engine (ACE)
IDMEF Alerts
Policy Instantiation Engine (PIE)
Context management
Concrete security rules
Sensors
Policy Enforcement Point (PEP)
Configurations
Policy Decision Point (PDP)

Key functions

Threat contexts

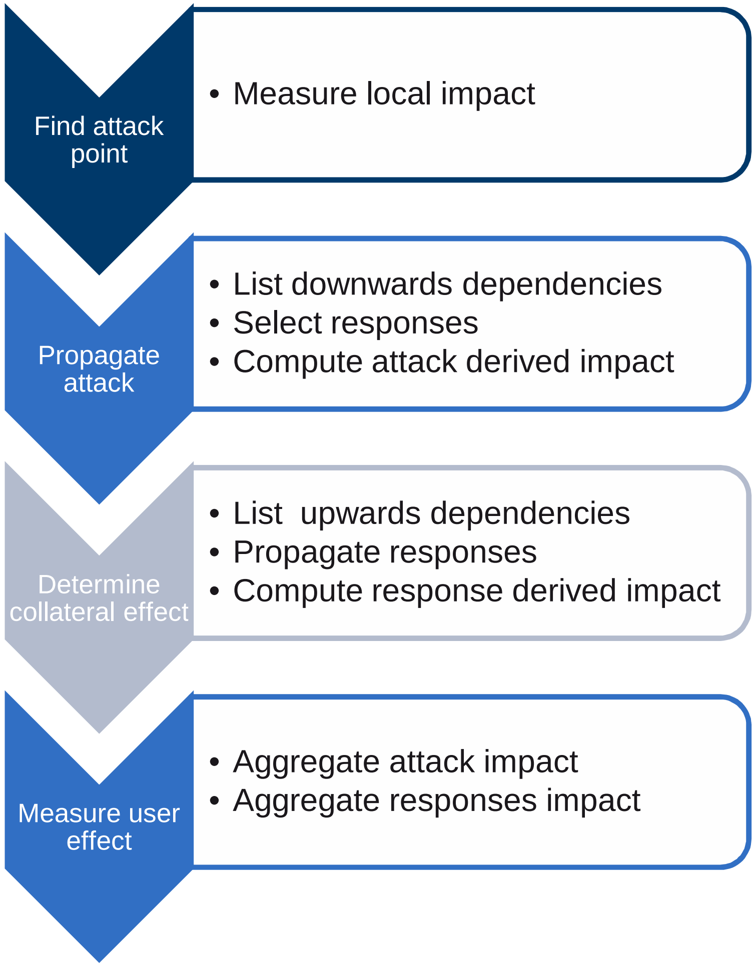
- Labelled through CVE (relationship w. alerts)
- Extensions required (generic attacks)
- Management of rule priorities (conflict resolution)

« guaranteed operational states »

- Normal context
- Minimal context
- Convergence (Datalog)

Issues with OSC

- **Selection of enforcement points**
  - Capabilities
  - Limit number of components (reuse)
- **Effect of response**
  - Negative ?
- **Proposed solution: dependencies modeling**



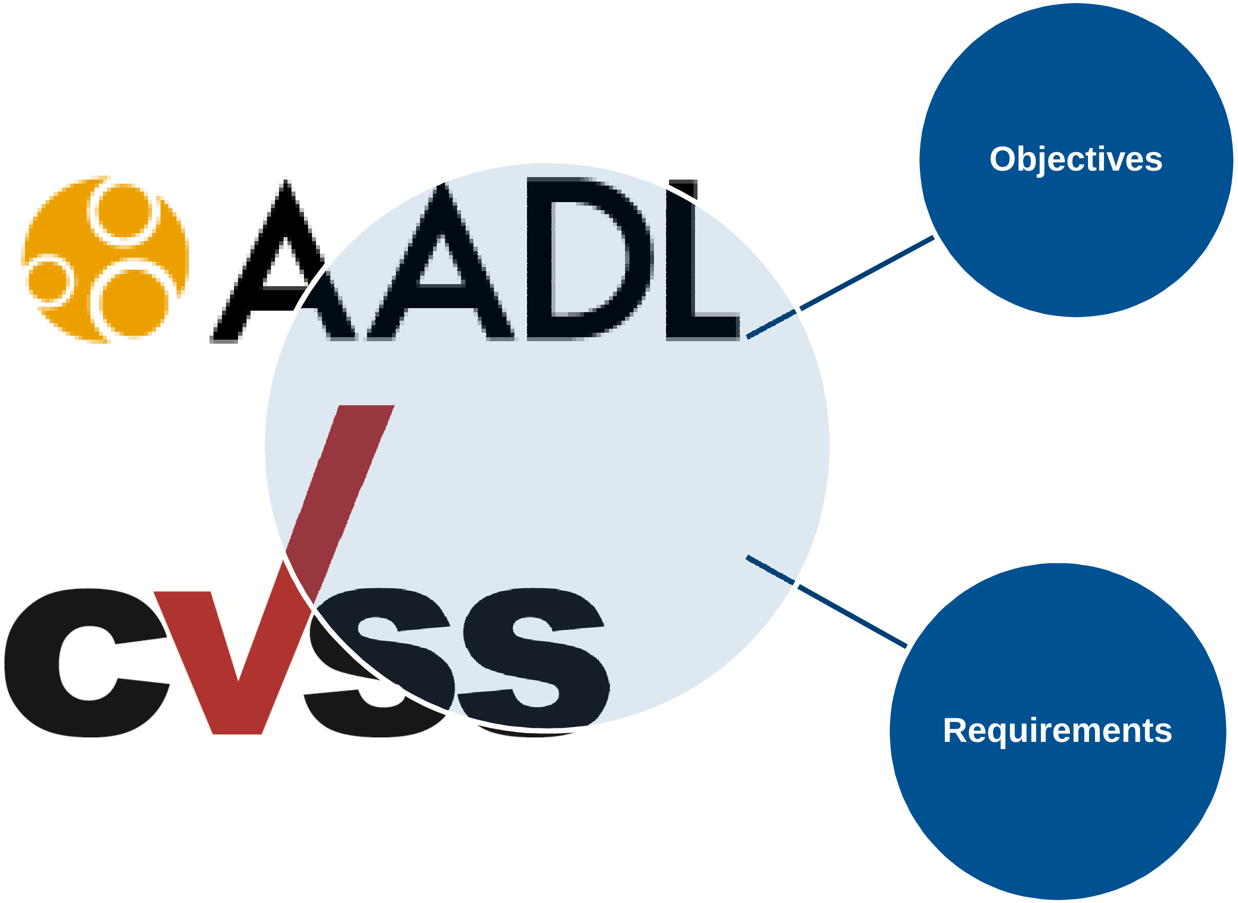
- Measure local impact

- List downwards dependencies
- Select responses
- Compute attack derived impact

- List upwards dependencies
- Propagate responses
- Compute response derived impact

- Aggregate attack impact
- Aggregate responses impact

How do we model and leverage dependencies



- Resolution of PEPs
- *Quantitative* impact assessment

- Formal model
- Capabilities of components
- Intrusion costs
- Modeling tools

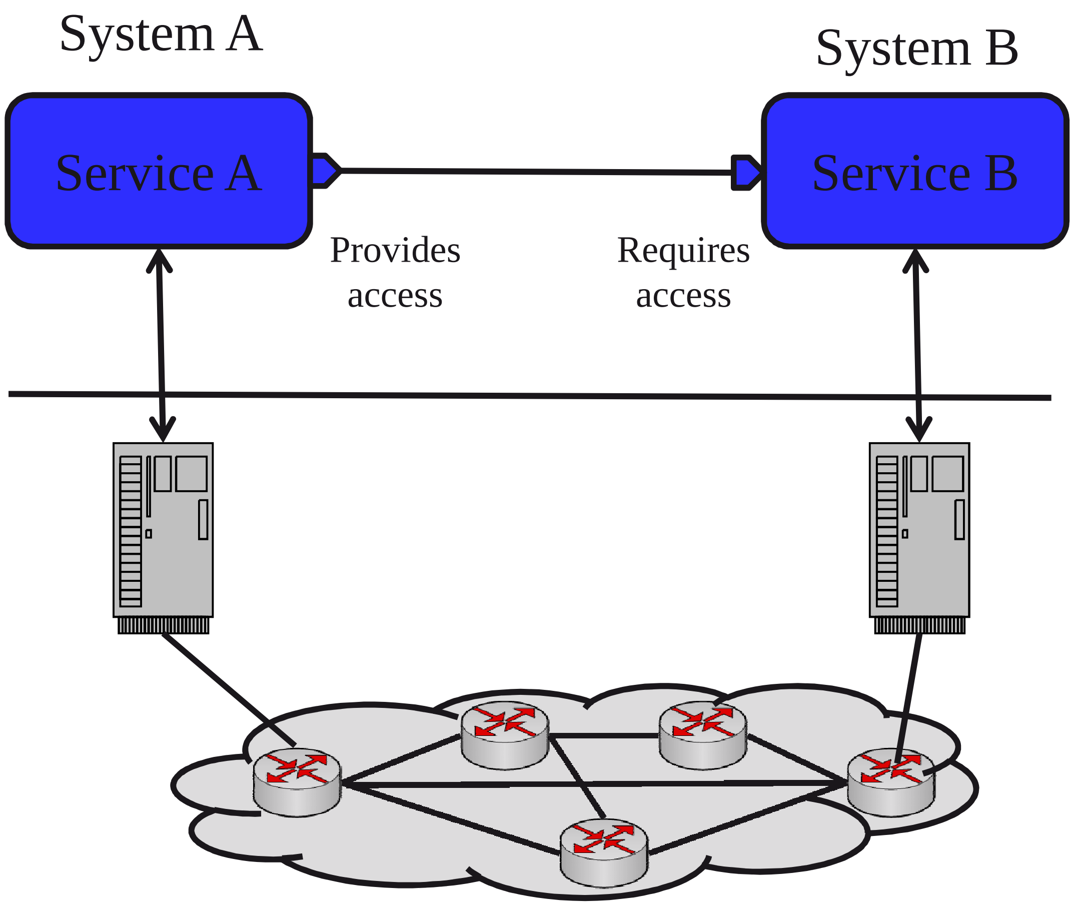
The SAE Architecture Analysis and Design Language (AADL) standard

Advantages

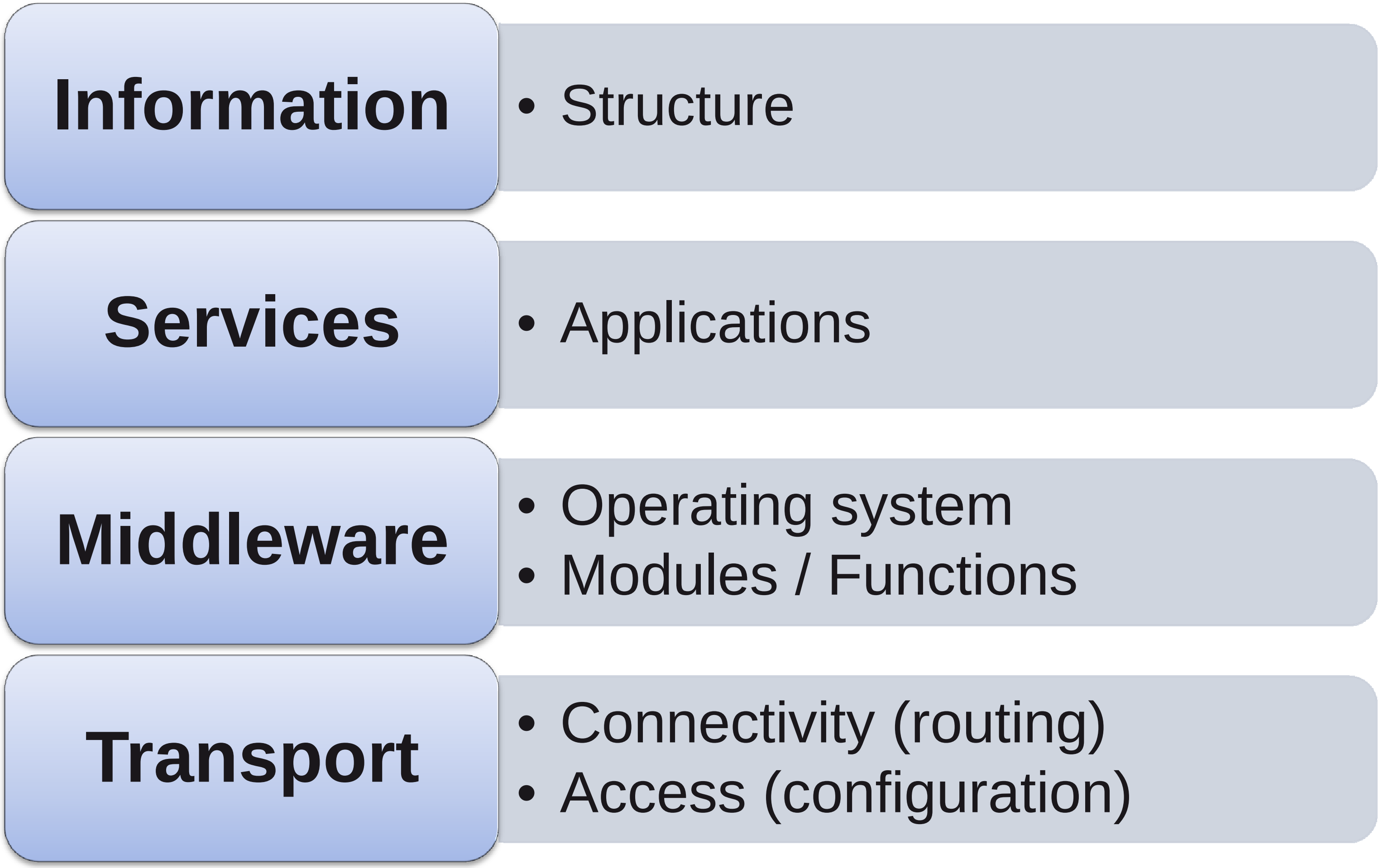
- Separation between interfaces and internal behavior
- Scalability by aggregation
- Operational modes
- Separation between topology and workflow
- Fault model

Additional assets

- XML representation
- Standard graphical tools
- Static and dynamic models



Dependencies are sometimes layered



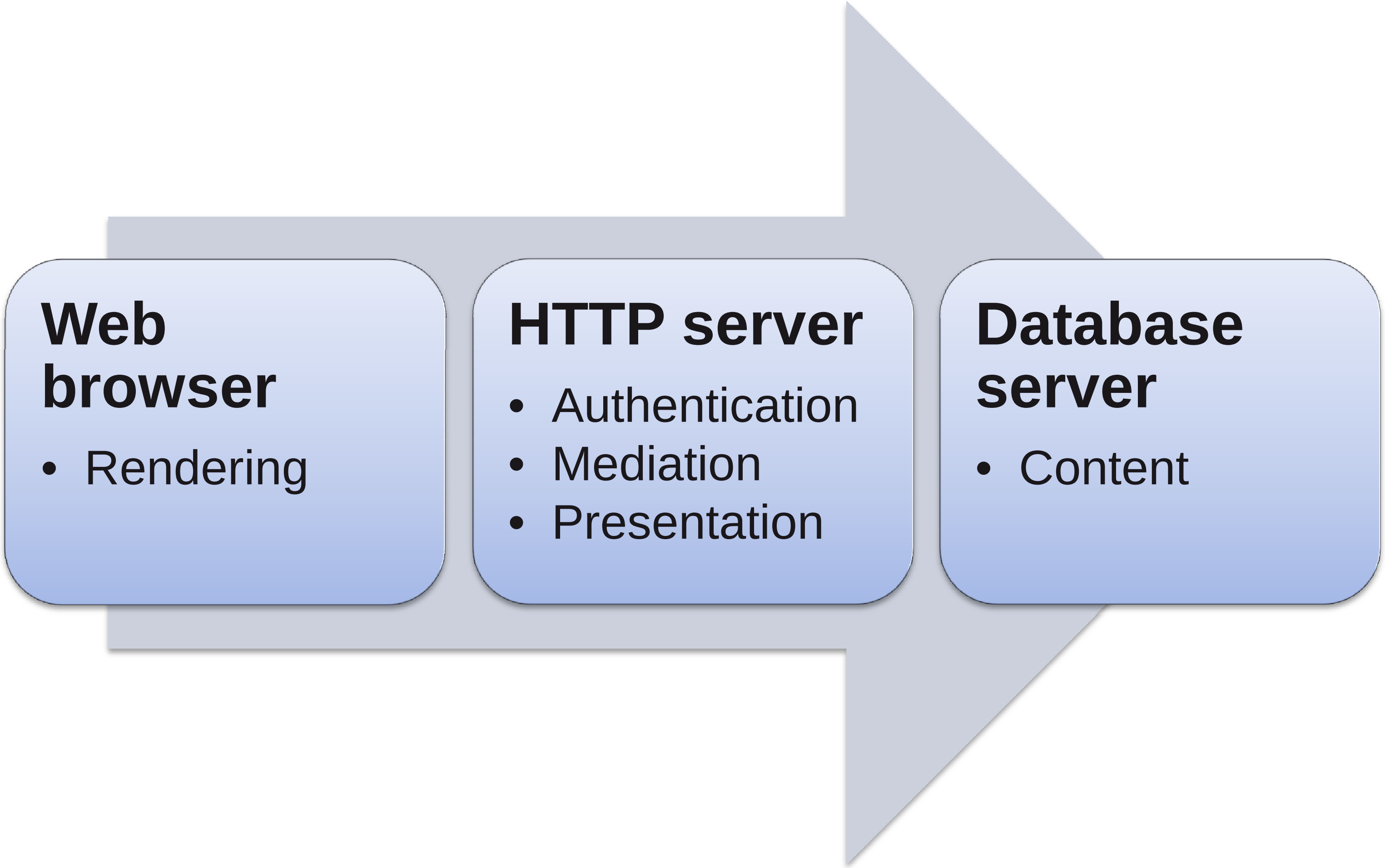
Information
- Structure

Services
- Applications

Middleware
- Operating system
- Modules / Functions

Transport
- Connectivity (routing)
- Access (configuration)

Dependencies are sometimes sequential



Web browser

- Rendering

HTTP server

- Authentication
- Mediation
- Presentation

Database server

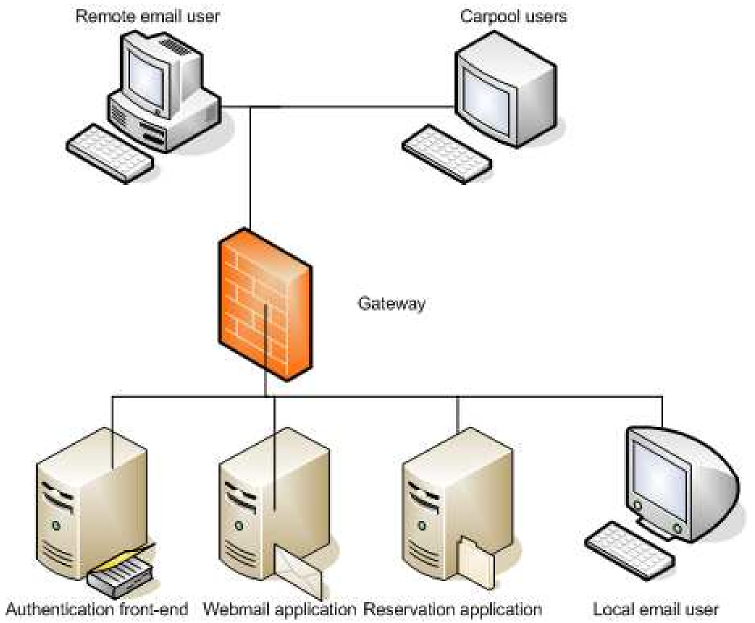
- Content

Dependencies properties

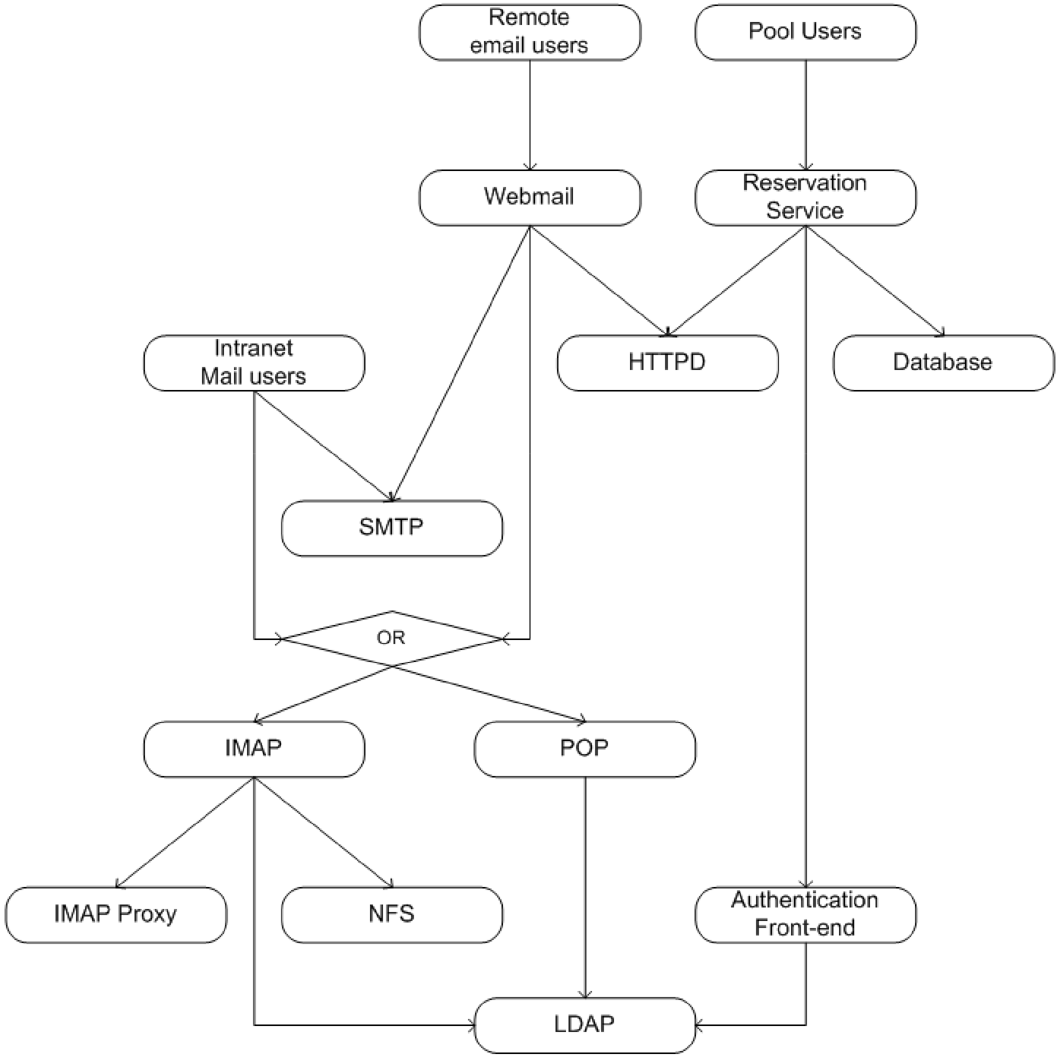
- **Topology**
  - User-side dependency
  - Service-side dependency
  - Proxy dependency
- **Workflow**
  - Start
  - Idle
  - Request
  - Stop
- **Temporality**
- **Failure impact**

Use case: car reservation platform

- **Content**
  - 3 web services
  - 3 user classes
- **Vehicle reservation**
  - Registered users only
  - Check available vehicles
  - Requires reservation
  - Cancel reservation
- **Email**
  - Webmail
  - POP
  - IMAP
- **Hidden services**
  - LDAP
  - NFS
  - MySQL
  - SMTP



Use case schematic dependencies description



Remote email users

Pool Users

Webmail

Reservation Service

Intranet Mail users

HTTPD

Database

SMTP

OR

IMAP

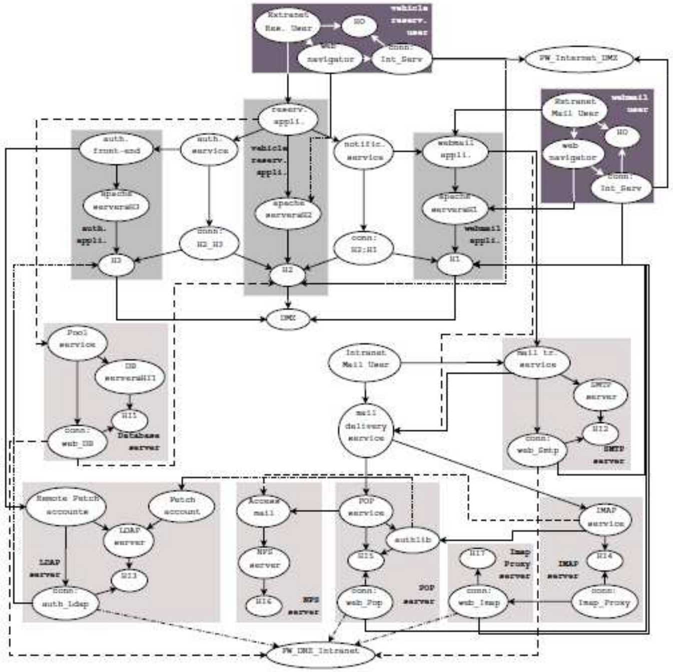
POP

IMAP Proxy

NFS

Authentication Front-end

LDAP

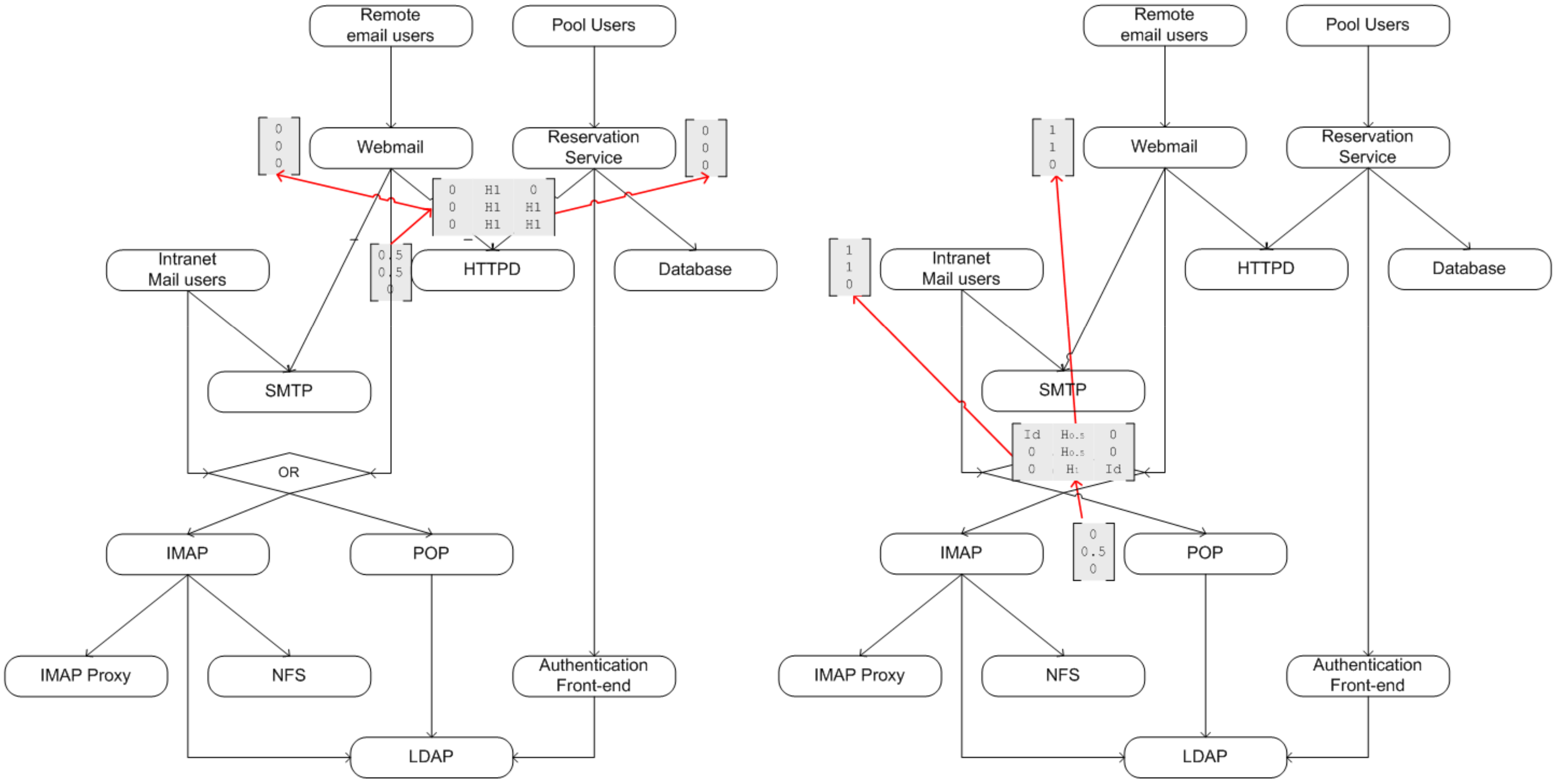


The « Quality of Experience » Index

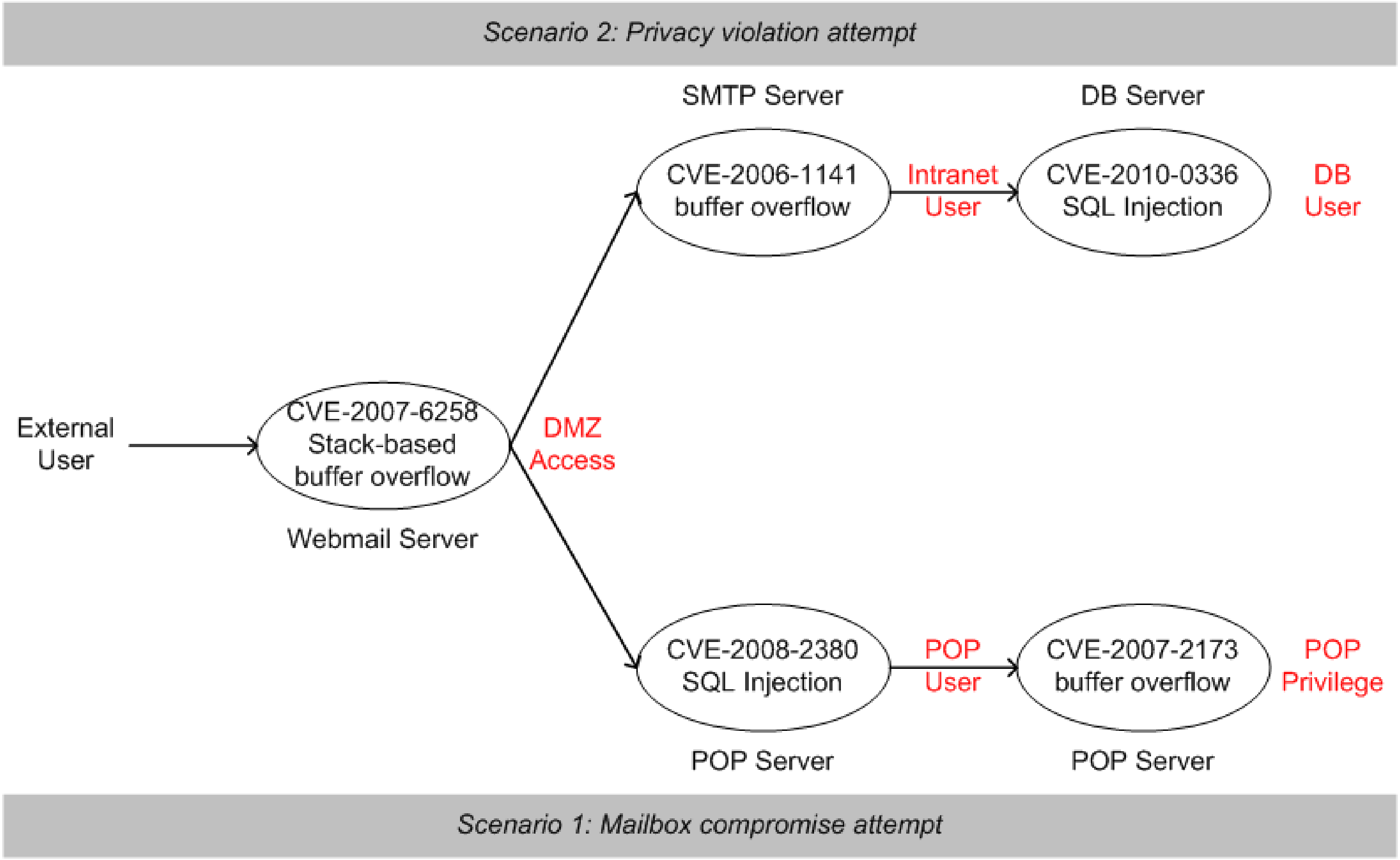
- Qualitative evaluation of attack impact
- Attack evaluated with CVSS vector score
- Impact transfer matrixes attached to each dependency
  - Both upwards and downwards
  - Functions (0, Id, Hx)
  - Sensitive choice

- QoE index computed from user perspective sensitivity on confidentiality, availability and integrity

Upwards propagation examples

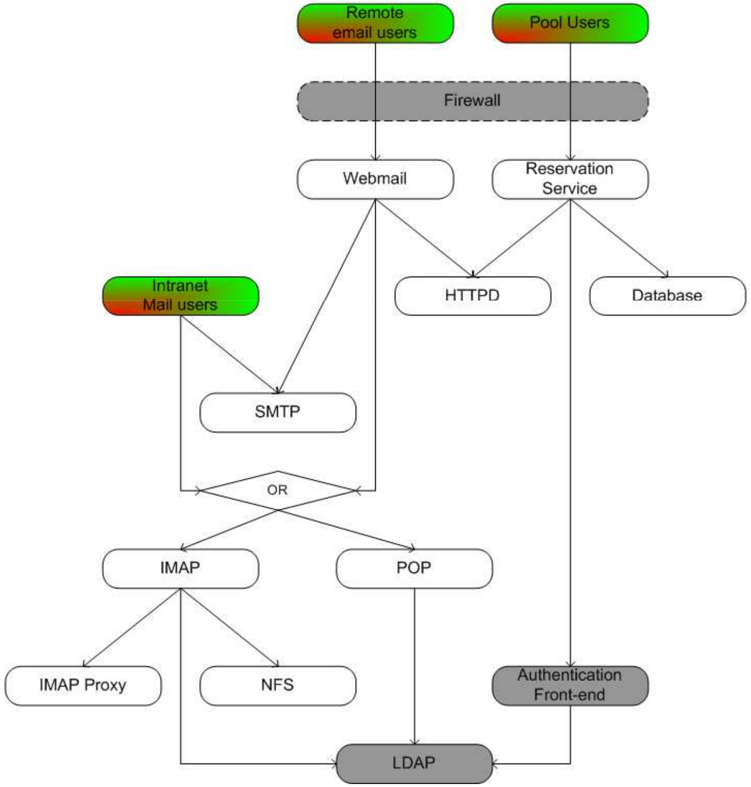


Attack propagations

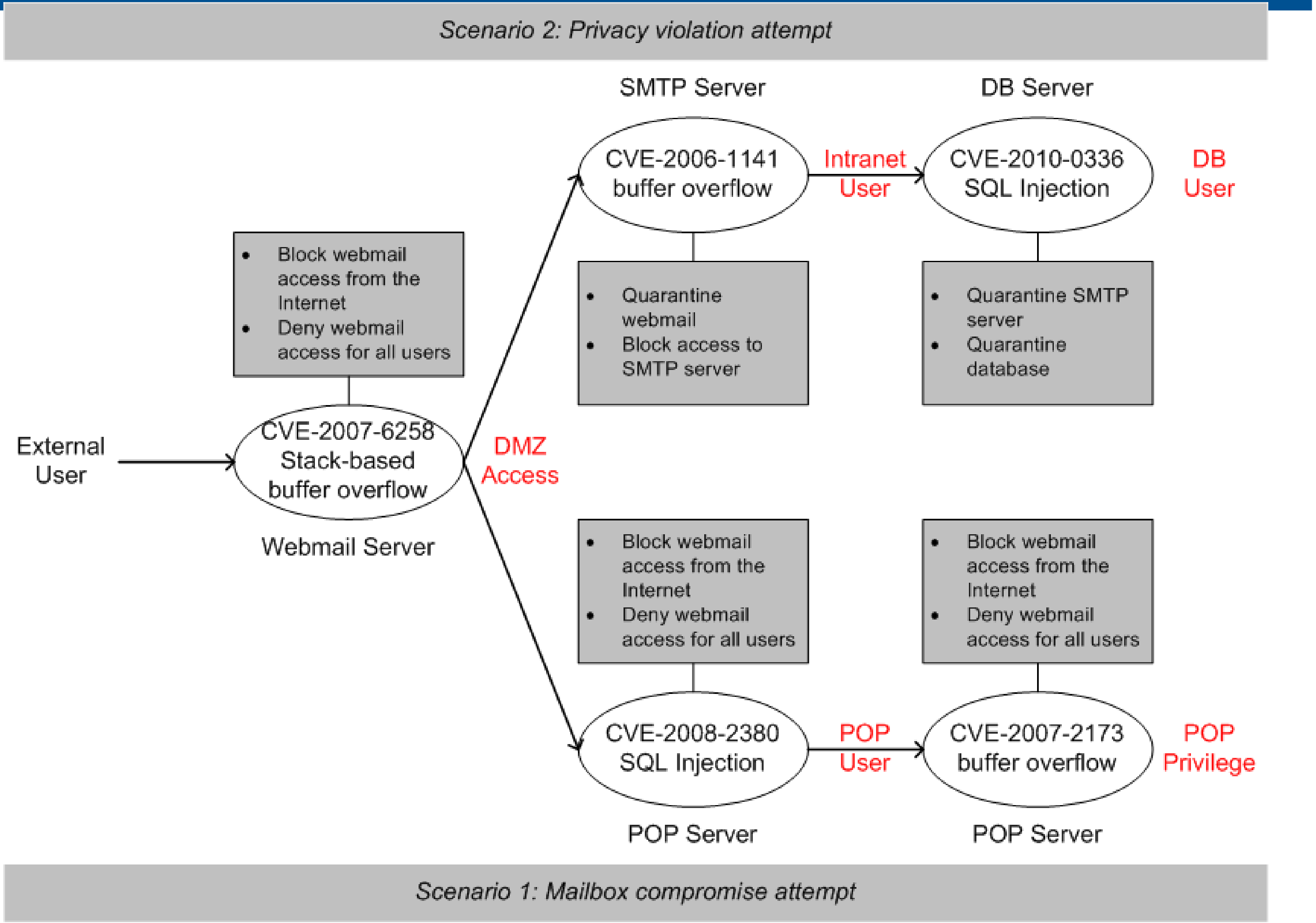

Scenario 2: Privacy violation attempt
SMTP Server
DB Server
CVE-2006-1141 buffer overflow
Intranet User
CVE-2010-0336 SQL Injection
DB User
External User
CVE-2007-6258 Stack-based buffer overflow
Webmail Server
DMZ Access
CVE-2008-2380 SQL Injection
POP User
CVE-2007-2173 buffer overflow
POP Privilege
POP Server
POP Server
Scenario 1: Mailbox compromise attempt

Enforcement points (PEP) and responses

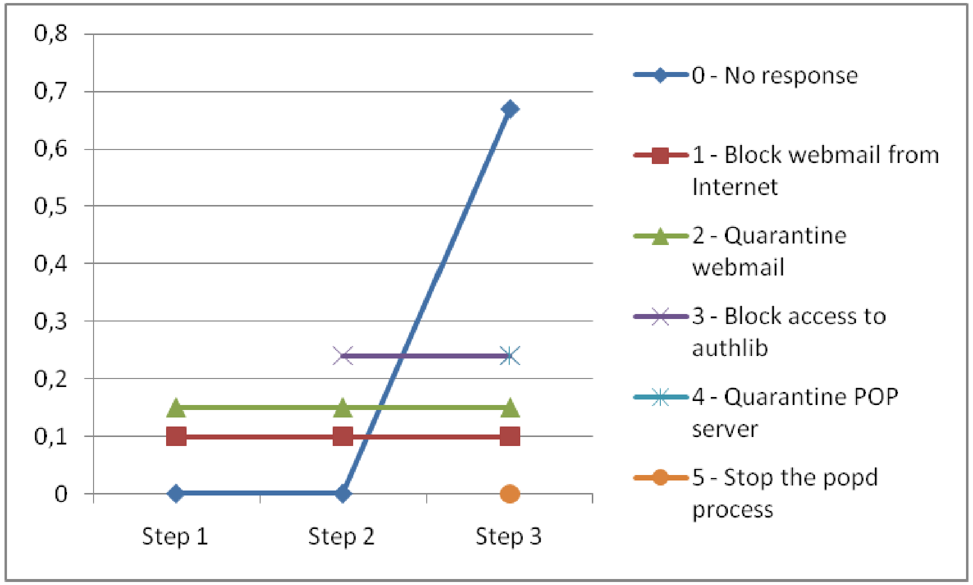
- **Components have at least minimal PEP functions**
  - Shutdown
- **Security components have additional power**
  - Firewall: filtering, quarantine
  - LDAP: user-level access control
- **Finding PEPs : downwards dependency propagation**



Response strategies

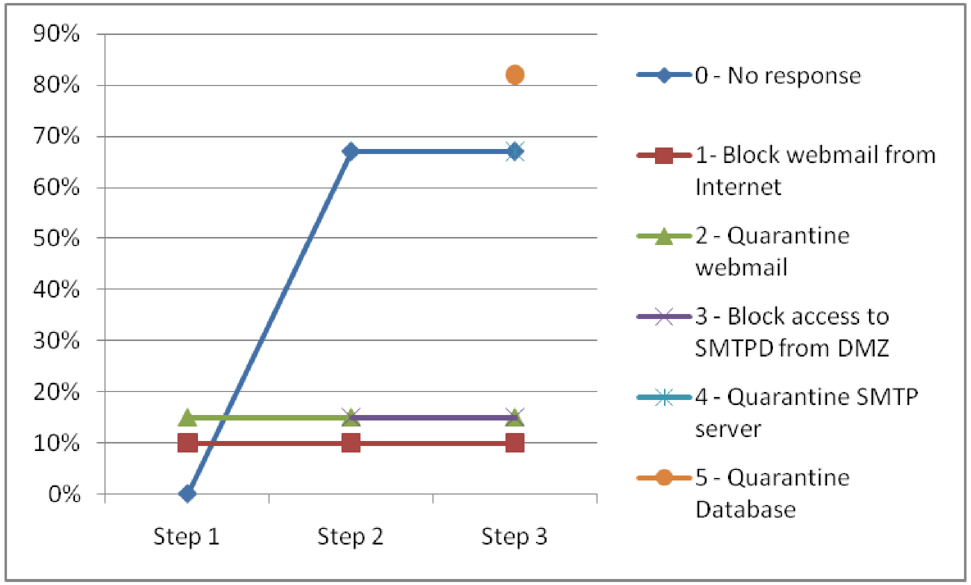

Scenario 2: Privacy violation attempt
SMTP Server
DB Server
CVE-2006-1141 buffer overflow
Intranet User
CVE-2010-0336 SQL Injection
DB User
Block webmail access from the Internet
Deny webmail access for all users
Quarantine webmail
Block access to SMTP server
Quarantine SMTP server
Quarantine database
External User
CVE-2007-6258 Stack-based buffer overflow
DMZ Access
Webmail Server
Block webmail access from the Internet
Deny webmail access for all users
Block webmail access from the Internet
Deny webmail access for all users
CVE-2008-2380 SQL Injection
POP User
CVE-2007-2173 buffer overflow
POP Privilege
POP Server
POP Server
Scenario 1: Mailbox compromise attempt

Scenario 1: mailbox compromise attempt



- **Step 1**
  - HTTP server compromised
  - Response 1 impacts extranet users
  - Response 2 impacts all users
    - Access still possible through POP and IMAP
  - Response 0 allows normal behavior
- **Step 2**
  - Response 3 locks auth for all users
  - Response 4 locks both AUTH and POP
    - Users cannot open new sessions
  - Response 0 allows normal behavior
- **Step 3**
  - Attacker objective met
    - Strong impact
  - Response 5 leaves IMAP open for all users
    - No impact

Scenario 2: Privacy violation attempt



Step 1

- HTTP server compromised
- Response 1 impacts extranet users
- Response 2 impacts all users
  - Access still possible through POP and IMAP
- Response 0 allows normal behavior

Step 2

- Attack impact realized
- Response 1 activated

Step 3

- Additional candidates responses ineffective

Known issues (so far)

- **Scale**
  - Definition of transfer matrixes
  - Modularity of modeling tools
  - Perspective: Patterns ?
- **Model management and maintenance**
  - New vulnerabilities, services
  - New attack classes
- **Model use**
  - Uncertainty of environment
    - Presence/absence of machines
    - Unidentified assets (printers, level 2 switches, …)
  - Differentiation of assets

Aggregation of individual responses

- **Qualitative: conflict resolution mechanisms**
- **Perspective: Quantitative**
  - Combinations
  - Norms
- **Countermeasures over time**
  - Switchover between counter-measures
  - Start from “non-virgin” state
  - Oscillations
  - Deactivation of counter-measures
  - Distribution time versus efficiency time

Conclusions and future work

- **Adaptive security possible**
- **Difficulties to overcome**
  - Definition of dependencies and reaction patterns
  - Qualitative decision support (Simulation)
  - Acceptance