

# ***Initial Architecture for DESEREC (currently foreseen)***

**Maximilian List**  
IABG mbH (DEU)

September 25th, 2006



*Dependability Security by Enhanced Reconfigurability*



# - **Table of Content**

---

## What Do We Have In Mind

- n Design Requirements
- n Goal & Approach

## Details About The Initial Architecture

- n Concept & Overall Architecture
- n Terminology & Components
- n High Level Architecture
- n Internal Structure (*Annex*)
- n Interfaces (*Annex*)
- n Data Exchange (*Annex*)

## The Way Ahead

- n Next Steps



---

# What Do We Have In Mind



## - **Design Requirements & Guidelines (1)**

---

- n Ability to perform **local fast reaction** in order to limit impact of failure or attack into a contained area of the whole information system: it leads to **distributed agents in charge of a sub-system** (e.g. cell)
- n Use existing standard means of monitoring and intrusion detection for information system surveillance
- n Highly secured protocol between agents and centralized component of the DESEREC framework: **DESEREC framework shall not constitute a weakness for the Information System security**



## -Design Requirements & Guidelines (2)

---

- n Ability to **detect severe global incidents** (e.g. distributed attacks such as DDOS: collect information on the overall system status which leads to some **centralized decision modules**
- n Ability to **deploy global (hot) reconfiguration orders** over the whole information system in a **coordinated and fast** way
- n Ability to monitor, control and reconfigure services and its associated resources with minimum knowledge of the information system details in order to **cope with large scalable systems from any domain**



## The Goal

Increase dependability of

- critical,
- open, and
- interconnected

IS (Information System)



# - *DESEREC Initial Approach*

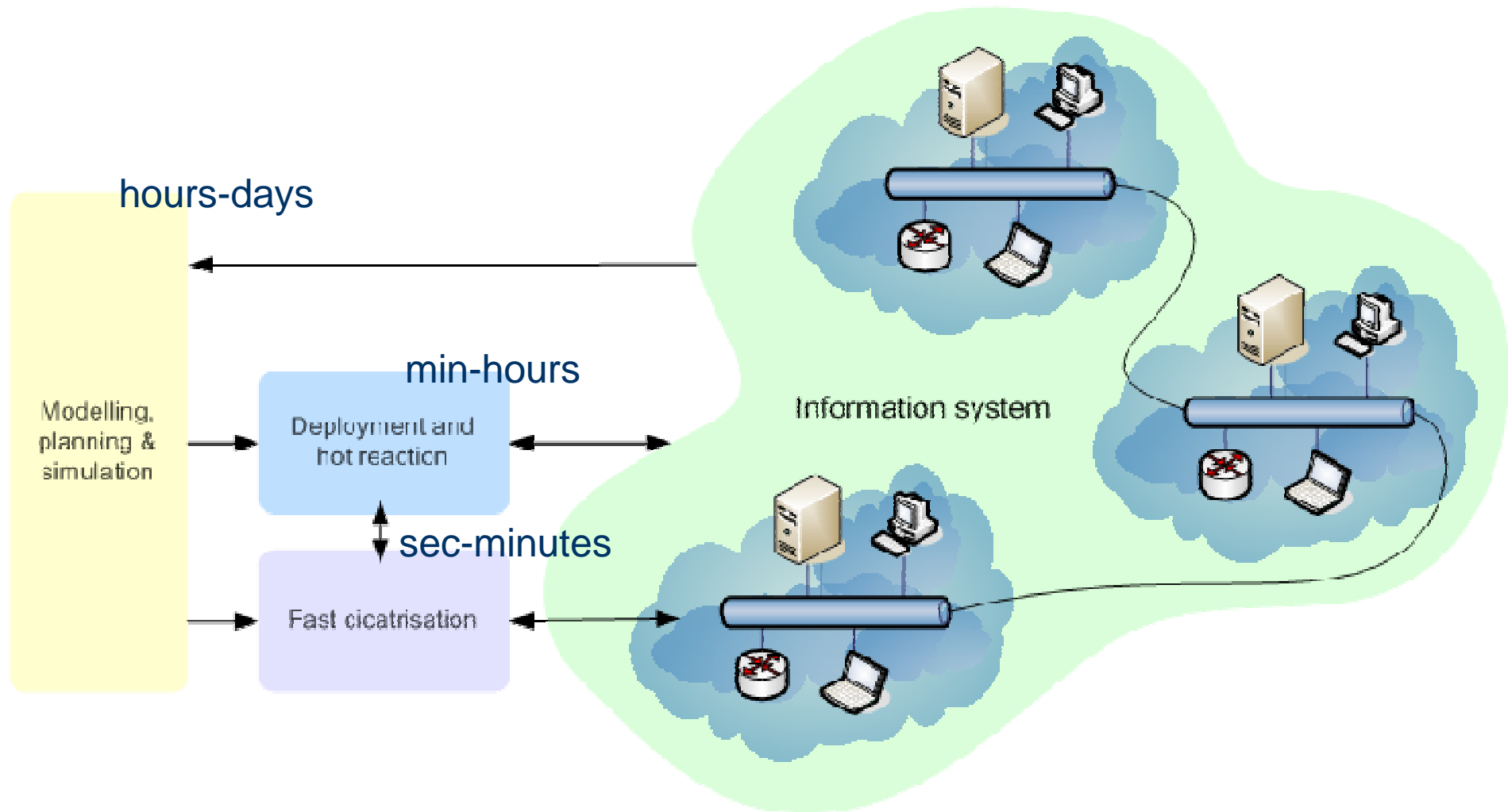
---

## Proposed Solution:

- n Three-tiered response to exceptions and incidents:
  - 4 Modeling, Planning, and Simulation
  - 4 Deployment and Hot Reconfiguration
  - 4 Fast Cicatrisation
- n These three global modules:
  - 4 act at different time scales (e.g. days, hours, minutes)
  - 4 represent different abstraction layers of the DESEREC system
  - 4 support different abstraction layers of the IS
    - | distinguish between services and (sub-)systems layer
    - | but support both
    - | and provide means of translation



# -High Level Architecture – Overview



# -The Three Global Modules

---

## Modeling, Planning & Simulation

- n model monitored IS (HL architecture, vulnerabilities, possible threats)
- n define operational planning for each possible incident (prevent threats)
- n simulate scenario based (on threat configurations)

## Deployment & Hot Reconfiguration

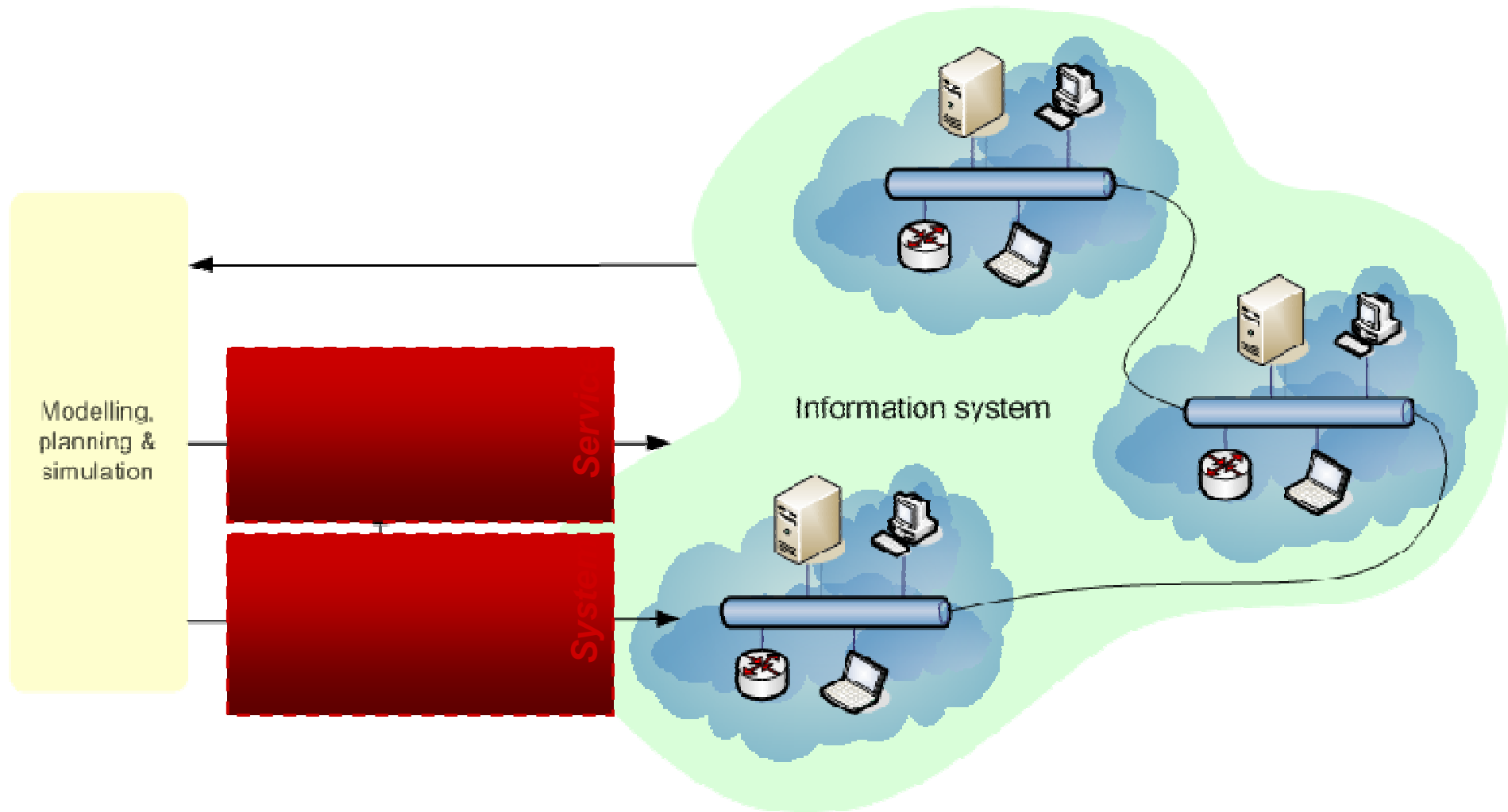
- n identify serious incidents at service level
- n notify operator and select a (pre-)defined reaction scenario
- n ensure the deployment at system level

## Fast Cicatrisation

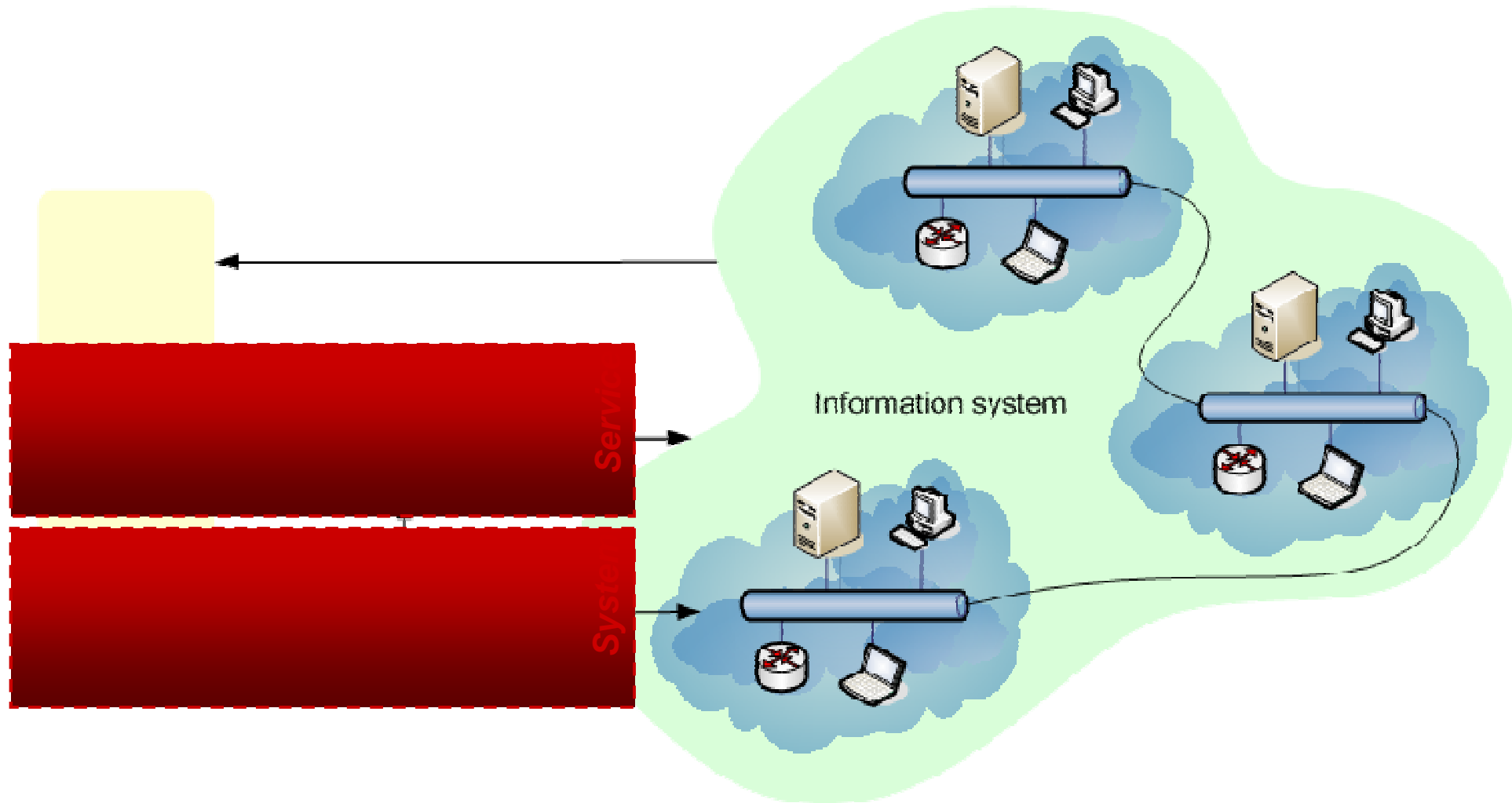
- n collect and normalize raw events from legacy devices
- n identify serious incidents in order to be able triggering an automatic reaction
- n trigger fast isolation reaction on serious incidents



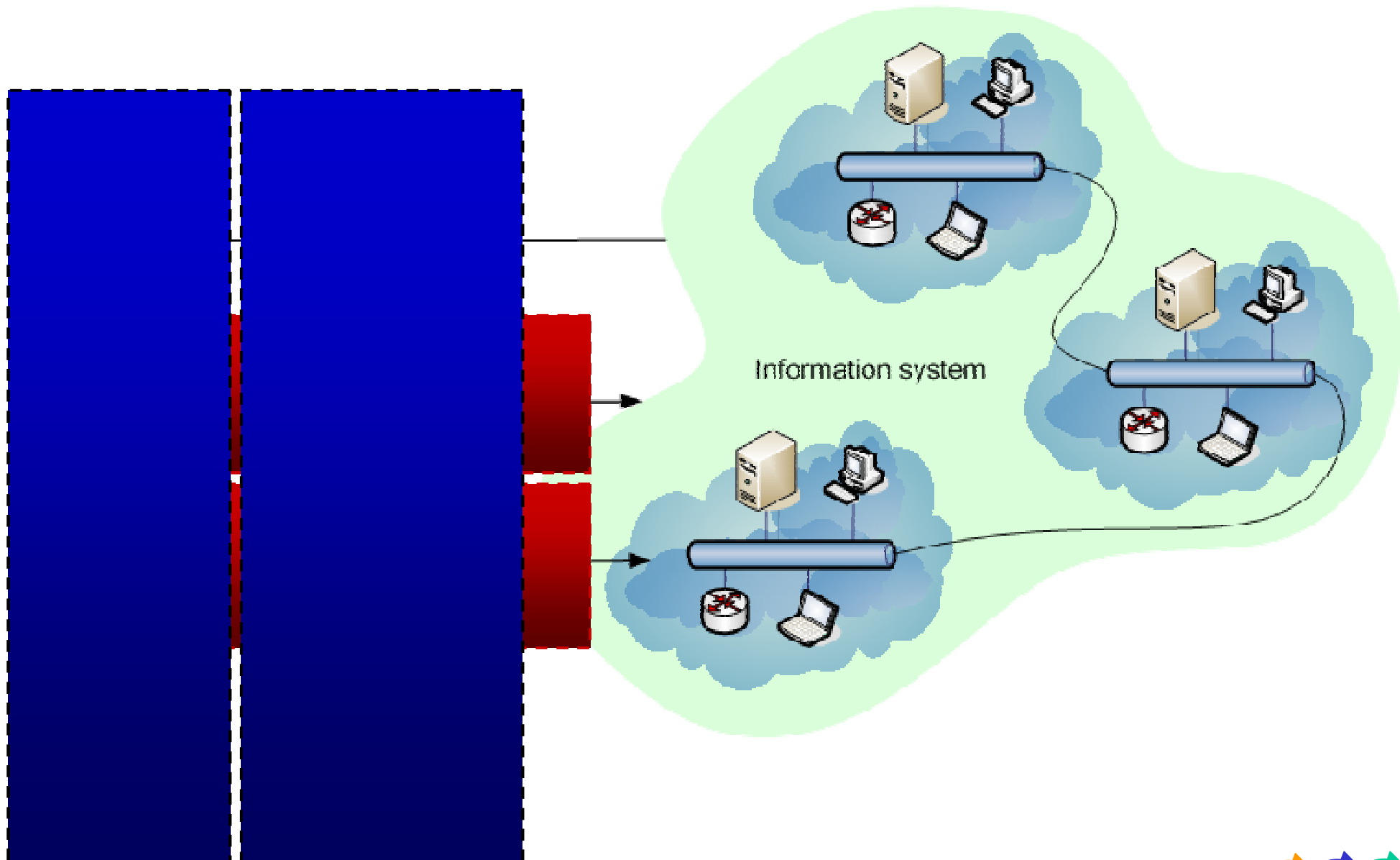
# -High Level Architecture – Overview



# -High Level Architecture – Overview



# -High Level Architecture – Overview



---

# Details About The Initial Architecture



# - **Conceptual Thoughts**

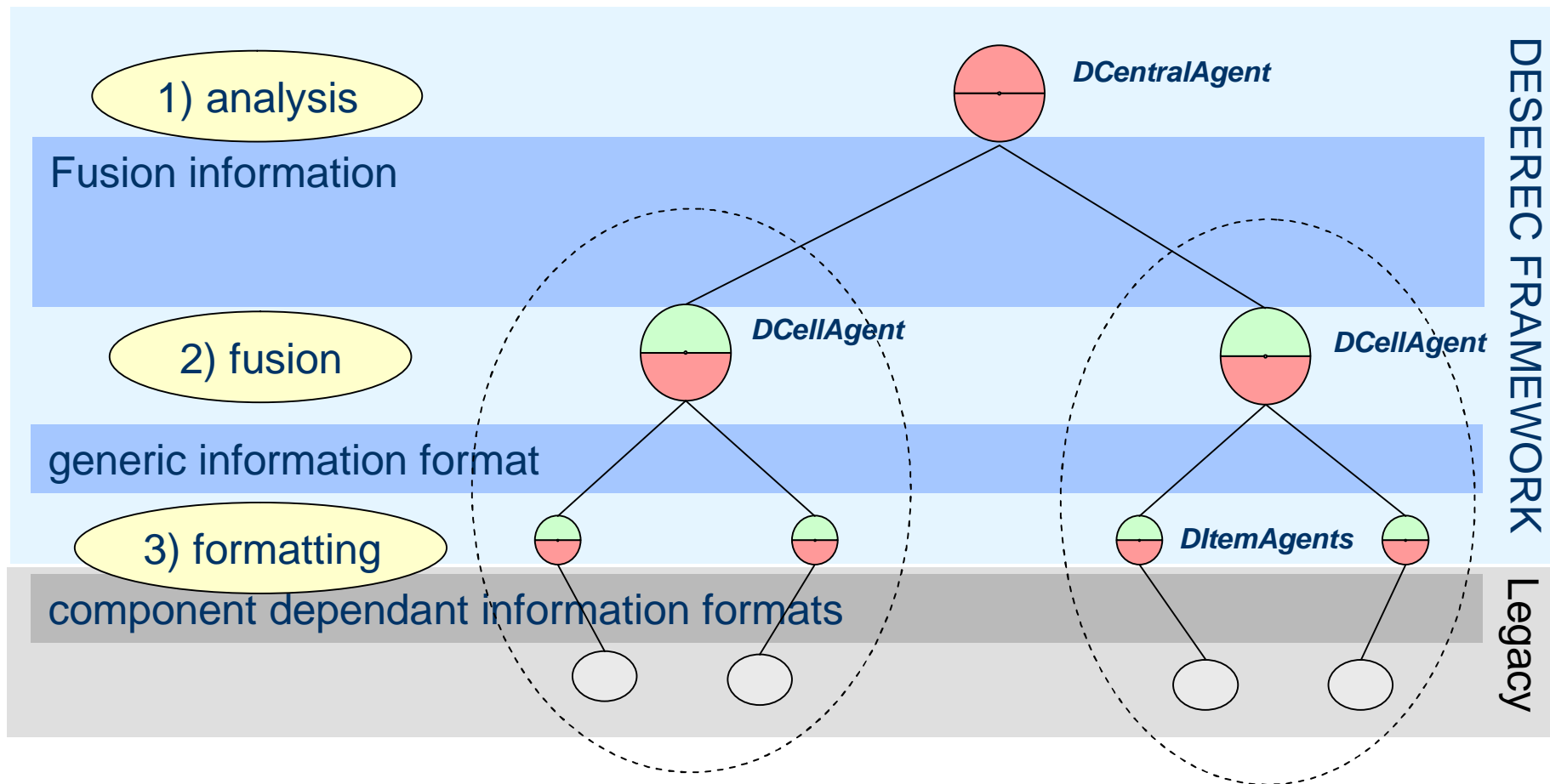
---

## Some basic concept thoughts

- n normalization and standardization
  - 4 of Events, Actions and Responses
  - 4 common protocol and format
  
- n hierarchical vs distributed
  - 4 base structure is hierarchical (3-level architecture : Central - Cell - Item)
  - 4 low-level interaction on a P2P base (fast exchange bw Cells)
  
- n re-use
  - 4 of functions at different DESEREC layers and modules (e.g. Translation)
  
- n overlapping of cells (to be discussed)
  - 4 sensor information collected across cell boundaries
  - 4 !!! atomic reactions limited to one cell (avoid blocking or concurrent device access)



# - Three Stages Logical Topology



- 4 central agent è large vision (i.e. detection of DDOS)
- 4 cell agent è scalability (reduce supervision bandwidth)
- 4 item agent è adaptation to legacy interfaces



# -Terminology & Components (1)

---

## DCentralAgent

- n Decision module
  - 4 (Global) System level detection algorithms
  - 4 Reaction (Service based)
  - 4 Dependability view
- n (Global) System level deployment
  - 4 Policy provision to lower layer
  - 4 interface with vulnerabilities database
- n Able to aggregate logs / alarms and process them



## -Terminology & Components (2)

---

### DCellAgent

- n Sub-system detection algorithms
  - 4 Able to perform information fusion
  - 4 Able to aggregate logs / alarms and process them
- n Fast reaction
  - 4 Able only to decide “dumbly” (detect triggered rules) *à No AI*
  - 4 Able to produce and send configurations to DItemAgents
  - 4 Able to request for policy information, and cache it *à scalability*
- n Able to communicate with other cell agents (to be discussed)
- n Internal substructure
  - 4 LCA (Local Control Agent) : deployment of (re-)configurations to DItemAgents
  - 4 LRA (Local Reaction Agent) : monitoring and detection



## -Terminology & Components (3)

---

### DItemAgent

- n Comprises DSensor and DProxy

### DSensor

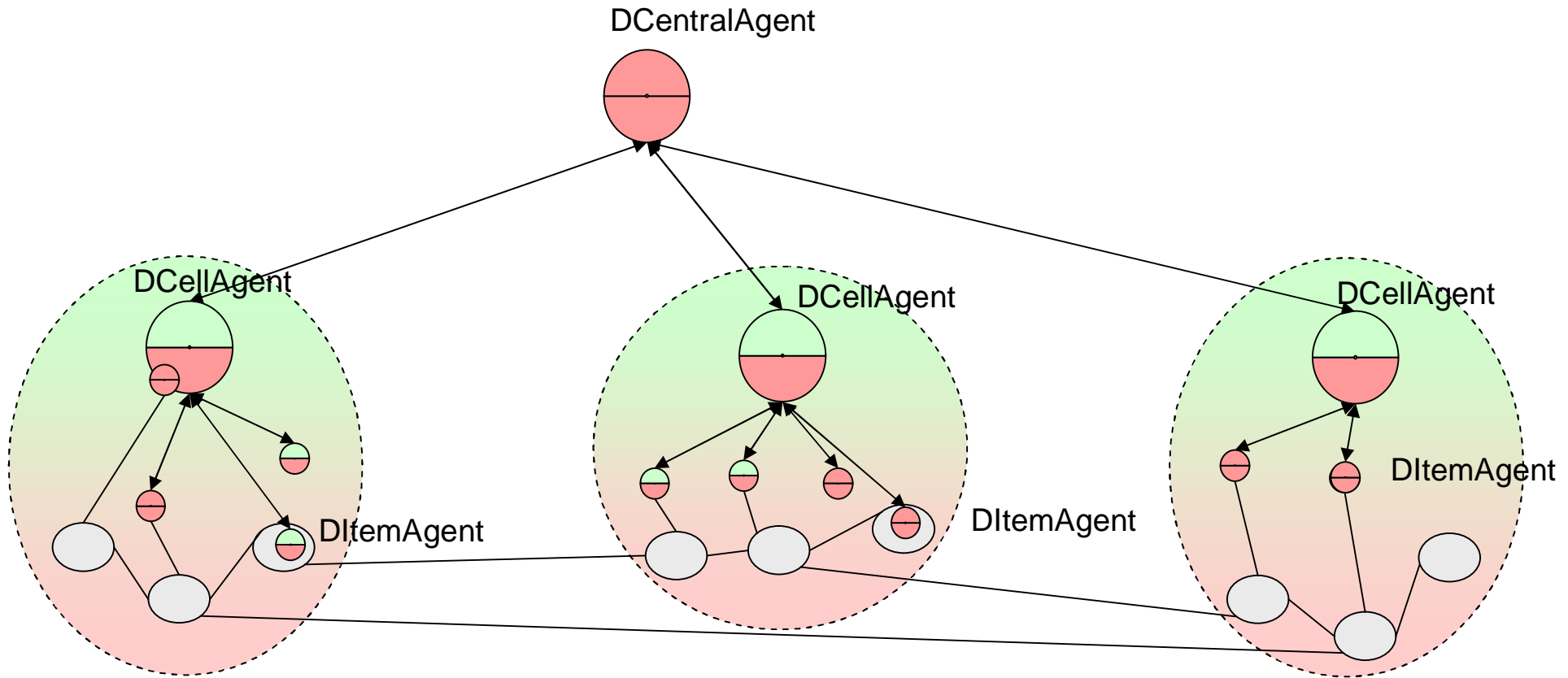
- n Monitors target devices/applications
- n Event collector
  - 4 Able to filter incidents / legacy events
  - 4 Able to translate to a common event format à *CEP / IODEF*
- n First Event aggregation

### DProxy

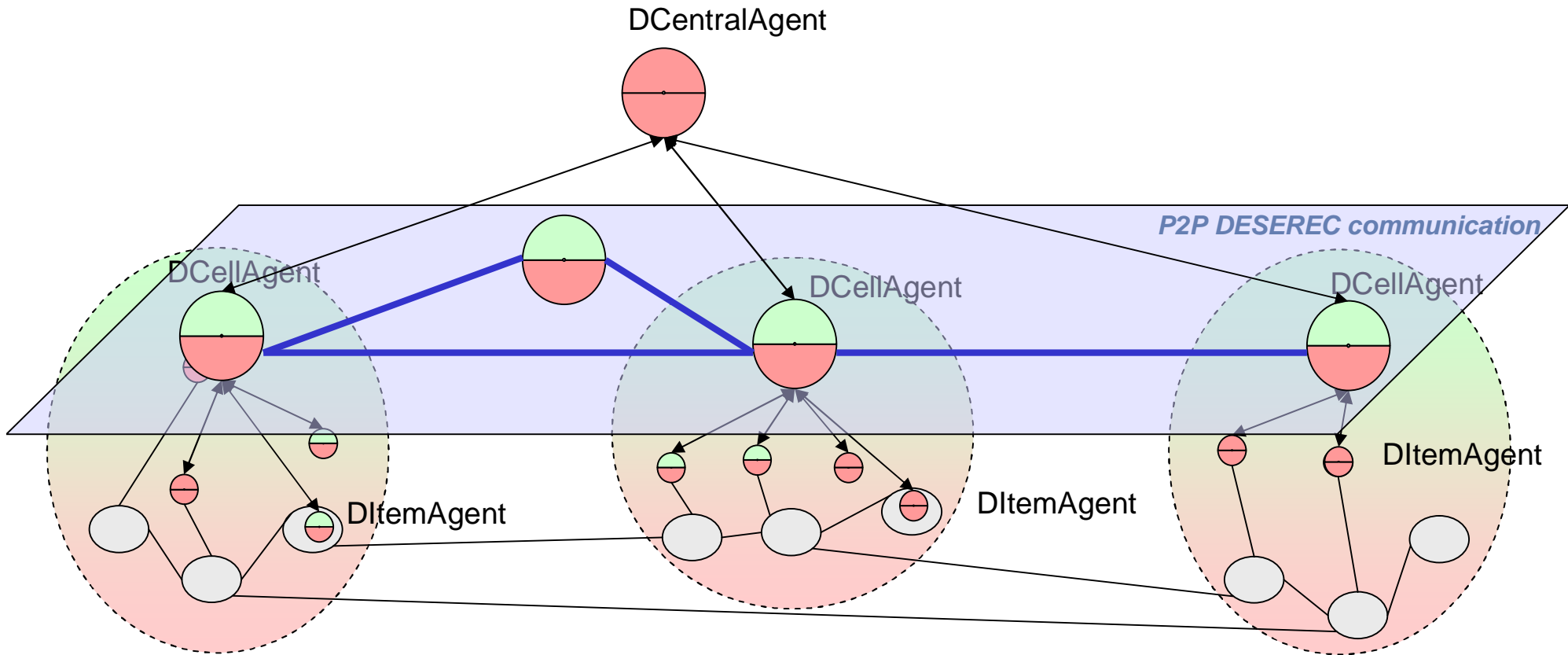
- n Translator
  - 4 Able to enforce configurations on target device/application
  - 4 Able to manage low level translation conflicts
- n Deployment on target device/application level



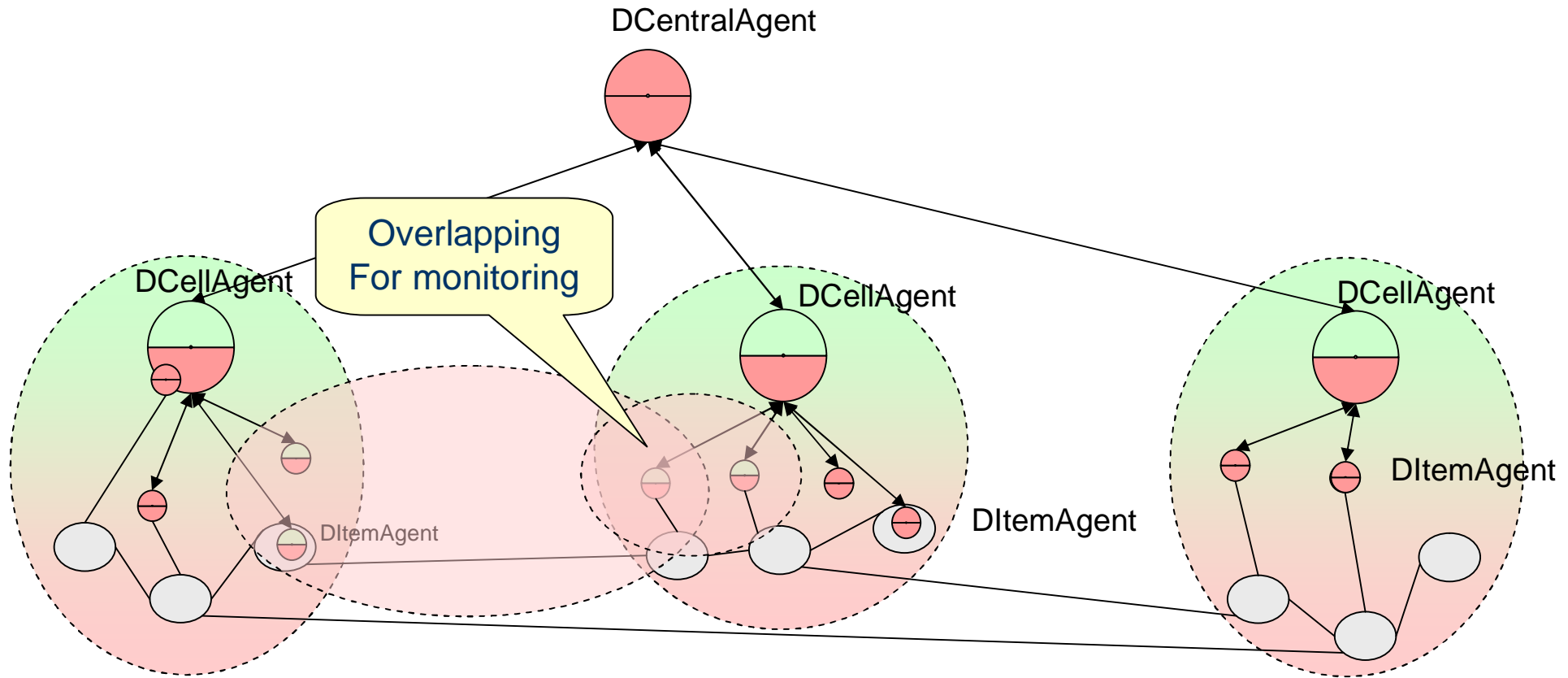
# -Overall Architecture – Hierarchical View



# - Overall Architecture – Distributed P2P Cell Communication –



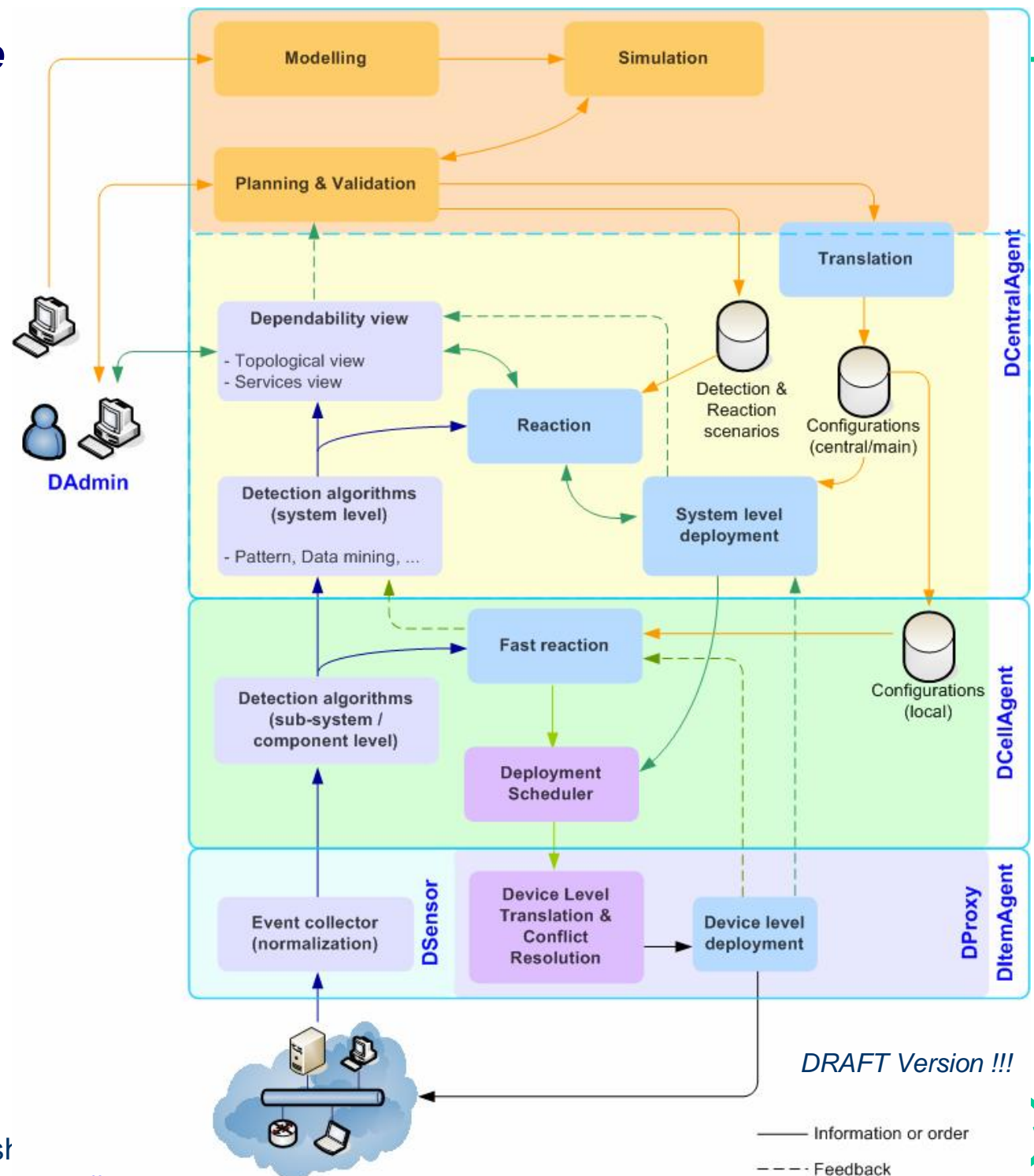
# -Overall Architecture – Cell Overlapping



Draft : to be discussed!!!

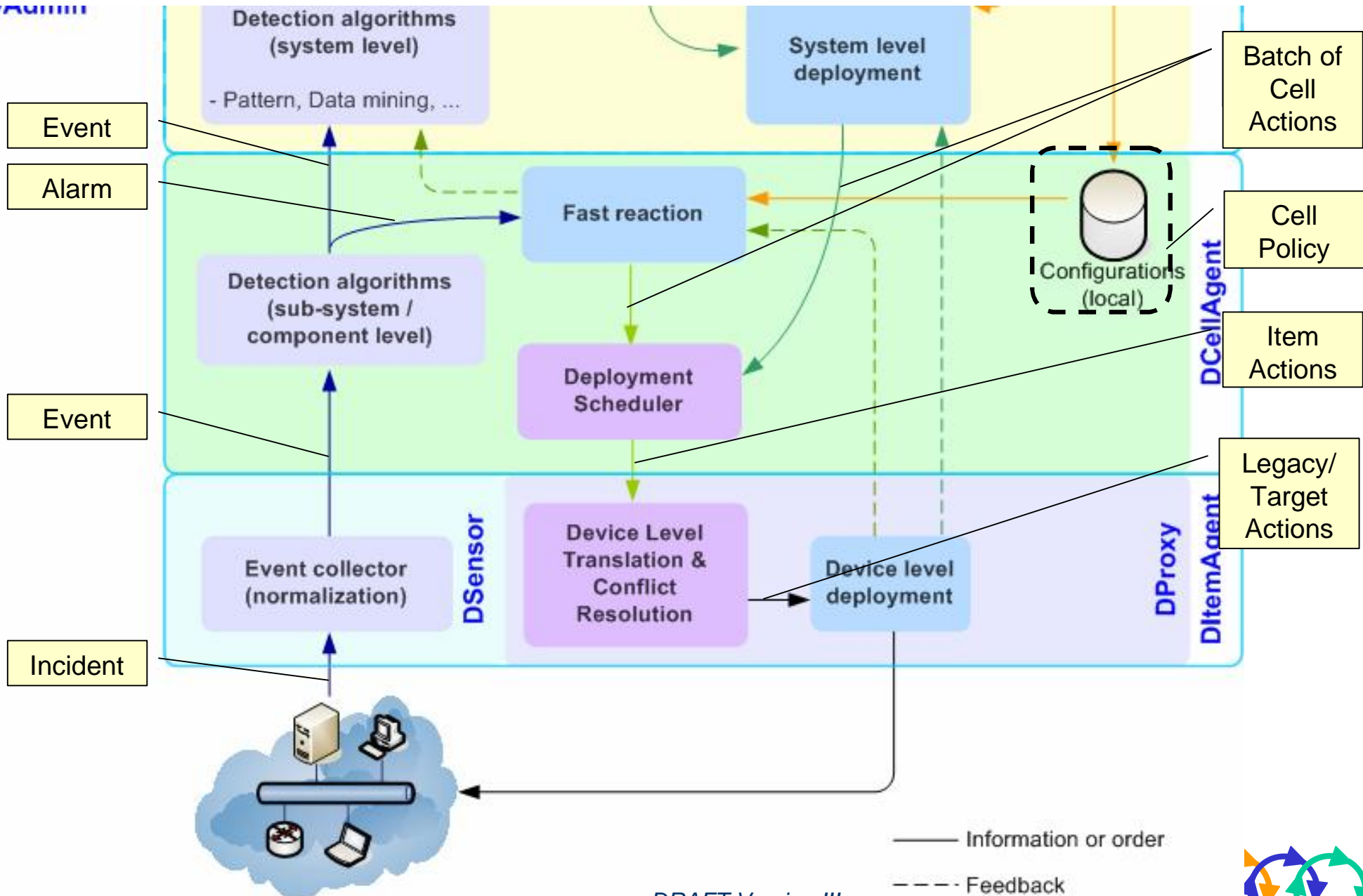


# -High Level Architecture & Modules – Functions

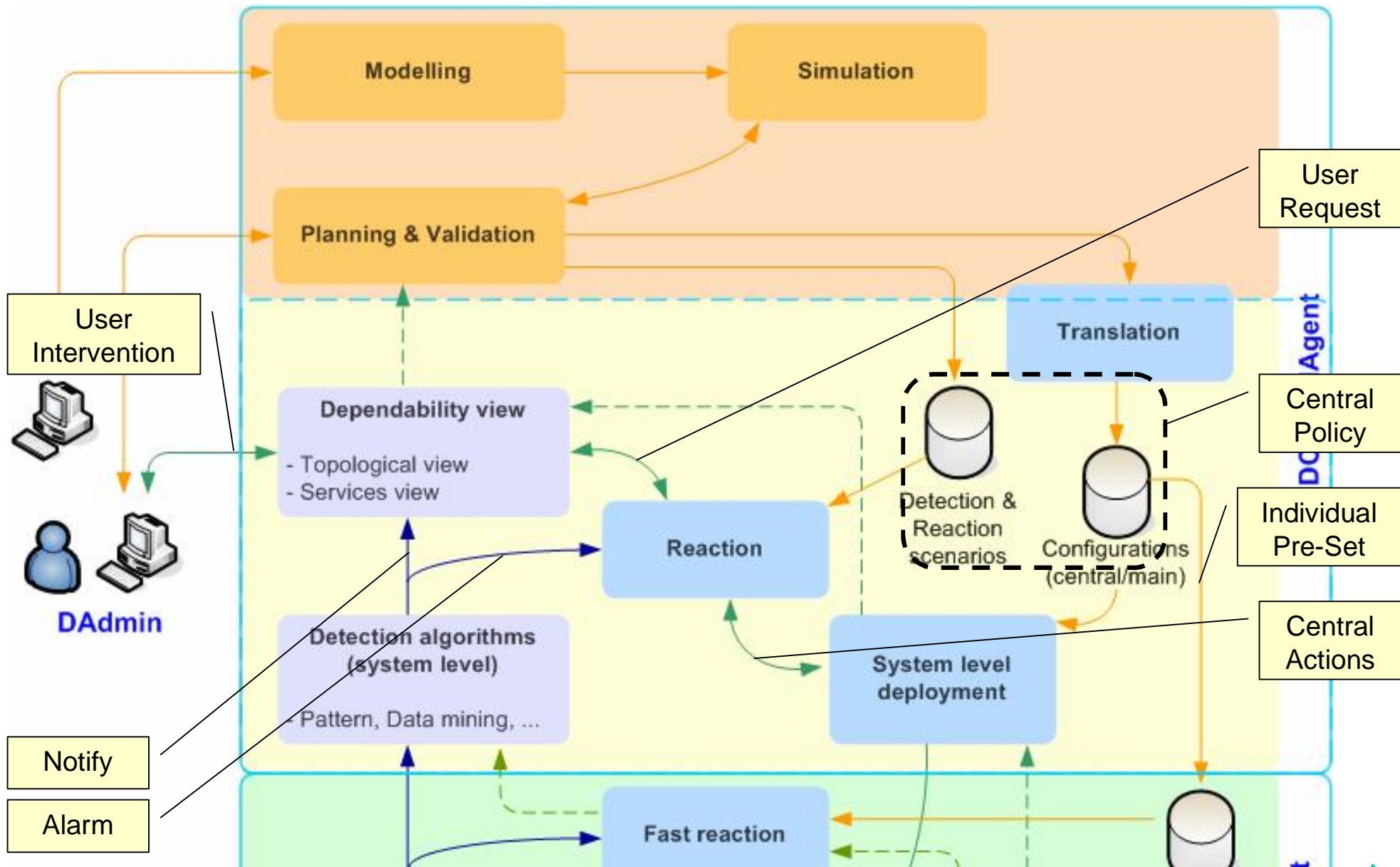


# -High Level Architecture & Modules – Functions

DRAFT



# -High Level Architecture & Modules – Functions



---

# The Way Ahead



## -Next Steps

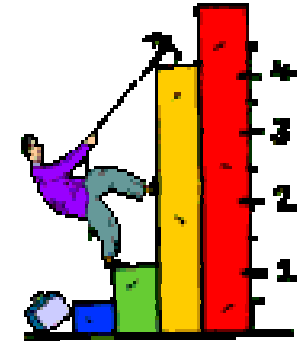
---

### Finalize

- n Initial architecture and components
- n Evaluation and definition of module inter-dependencies
- n Definition of interfaces (protocol, format, data, ...)
  - 4 Modules-Intercommunication Bus
  - 4 Message Semantic
- n ...

### Future Steps

- n Self-Learning and -Healing mechanisms (intelligence)
- n ...



---

# Questions ???



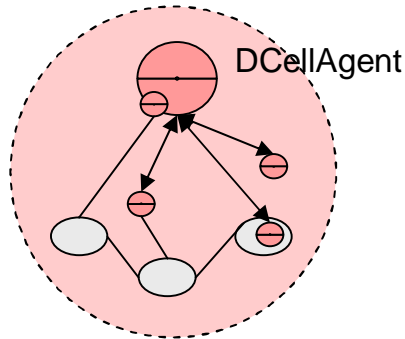
---

# Annex

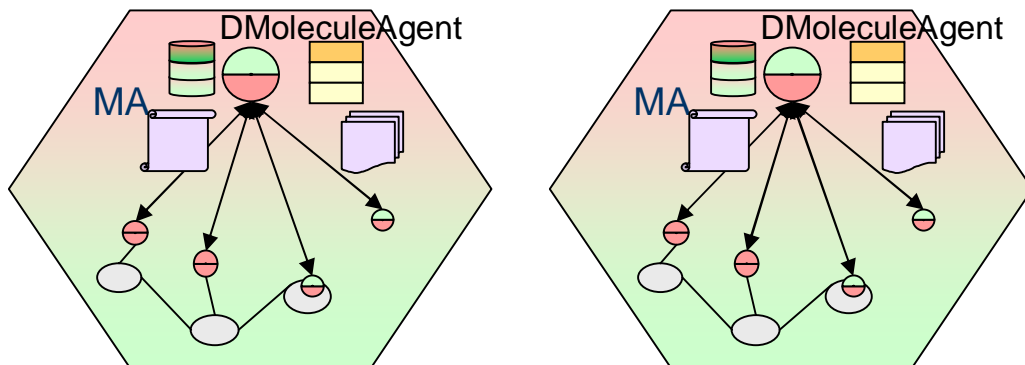


## - Cell and molecule

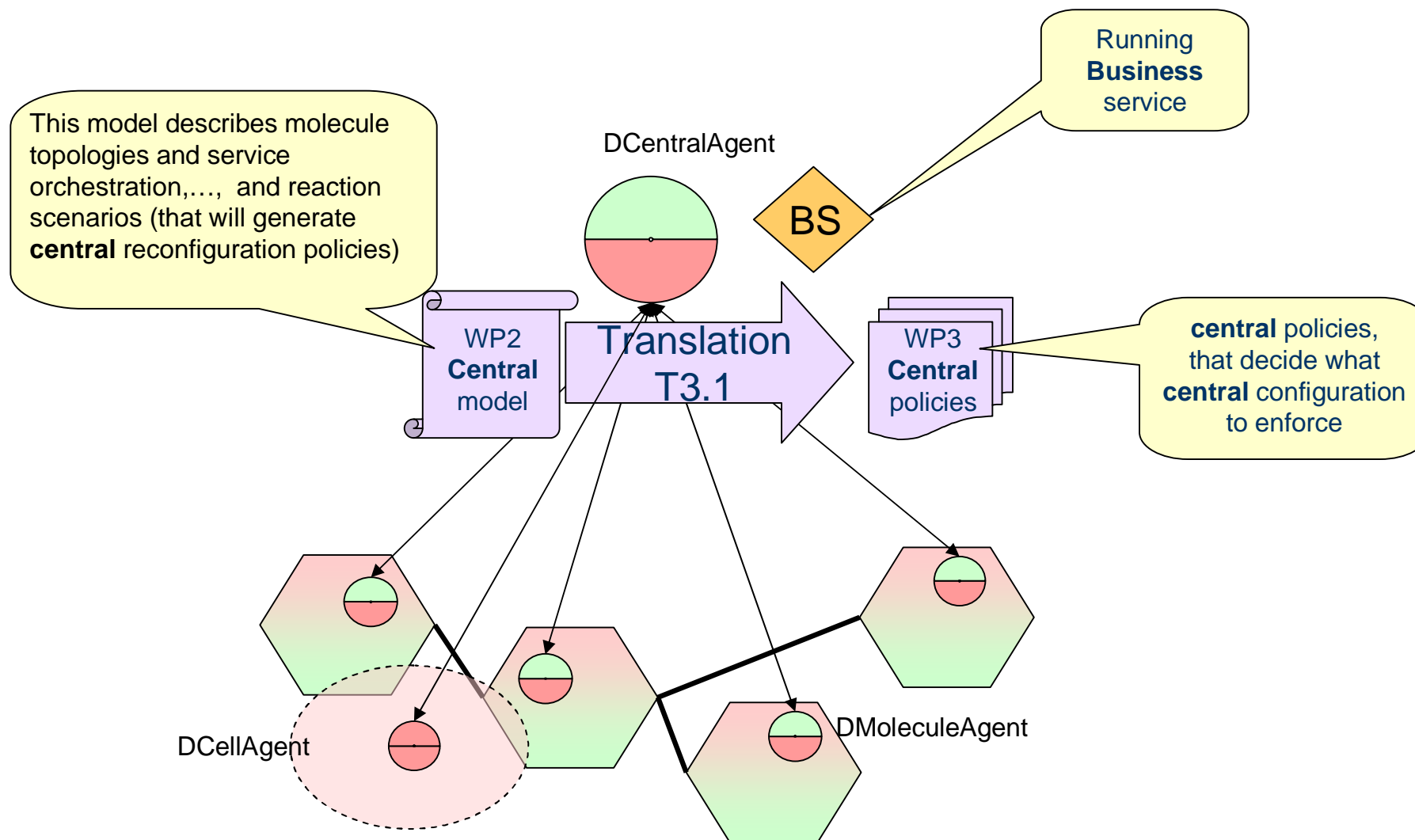
- A **cell** is a aggregation of all system elements (hardware and software) participating to a specific global service monitoring.
  - Cells may overlap molecule for monitoring purpose
  - It can be managed by a Cell Agent (which is centralize or not)



- A **molecule** is the aggregation of pre-defined system elements (hardware and software), managed by a DESEREC molecule agent, that includes predefined potentials configurations
  - Molecule can be instantiated more than once in a system (pattern/template)
  - Molecule cannot overlap

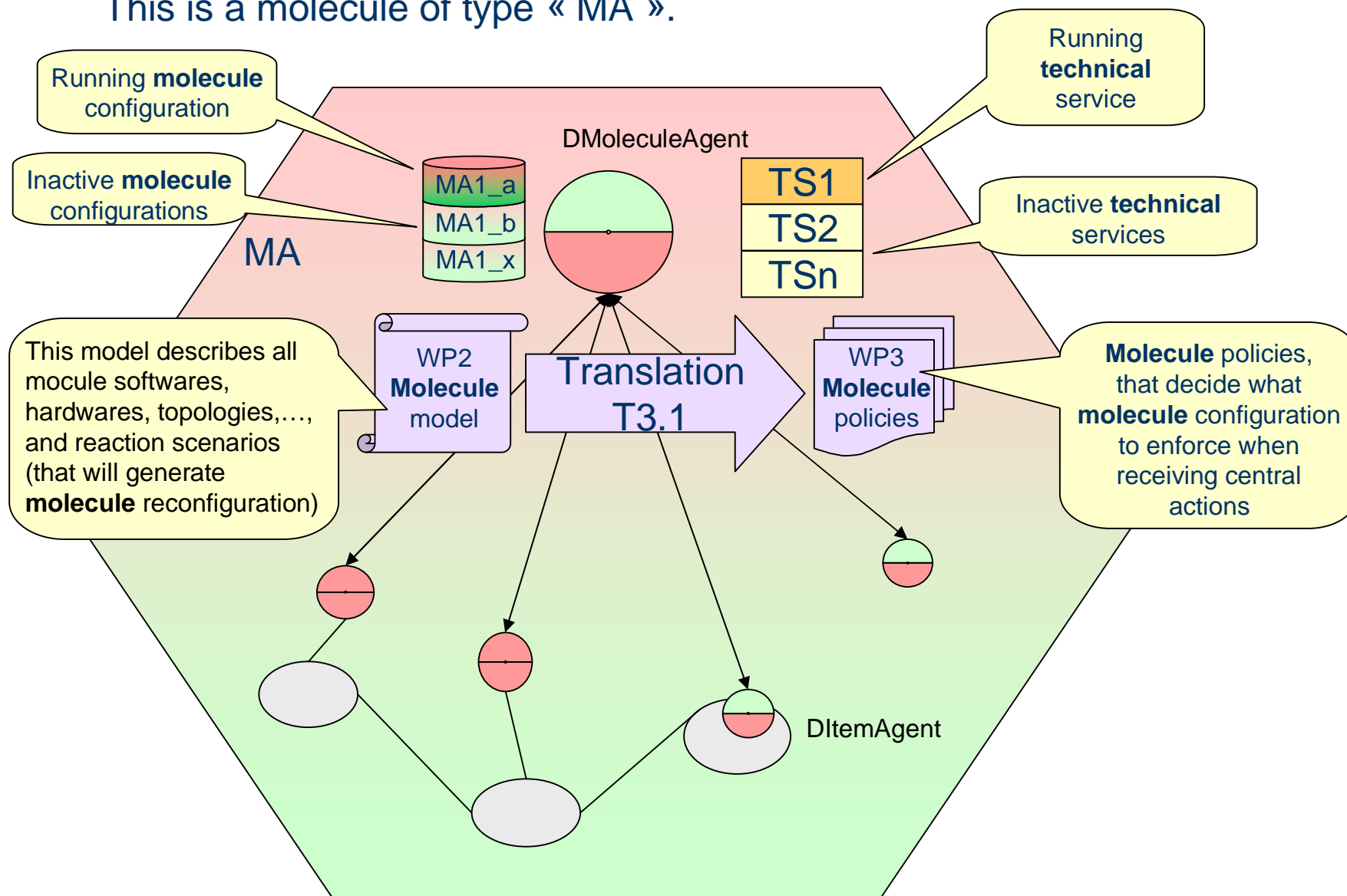


# -Central description

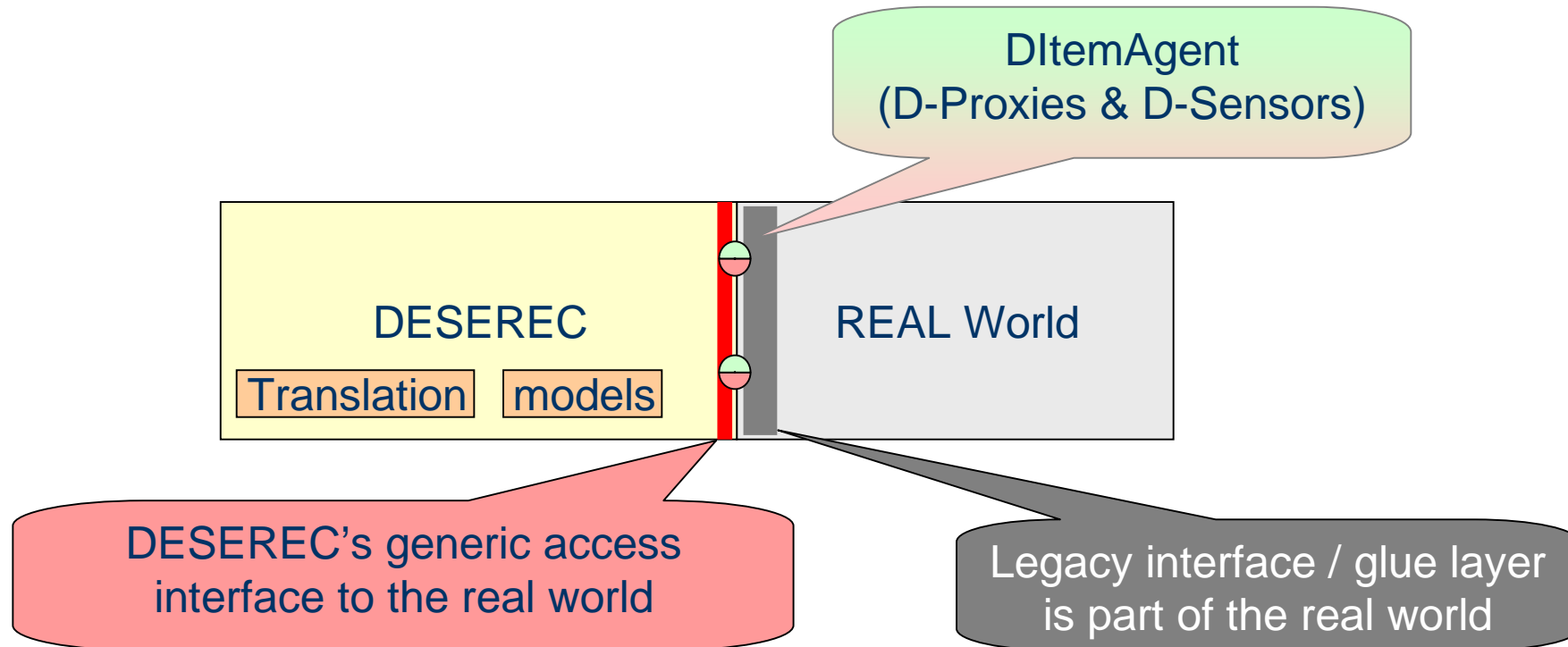


# - Molecule description

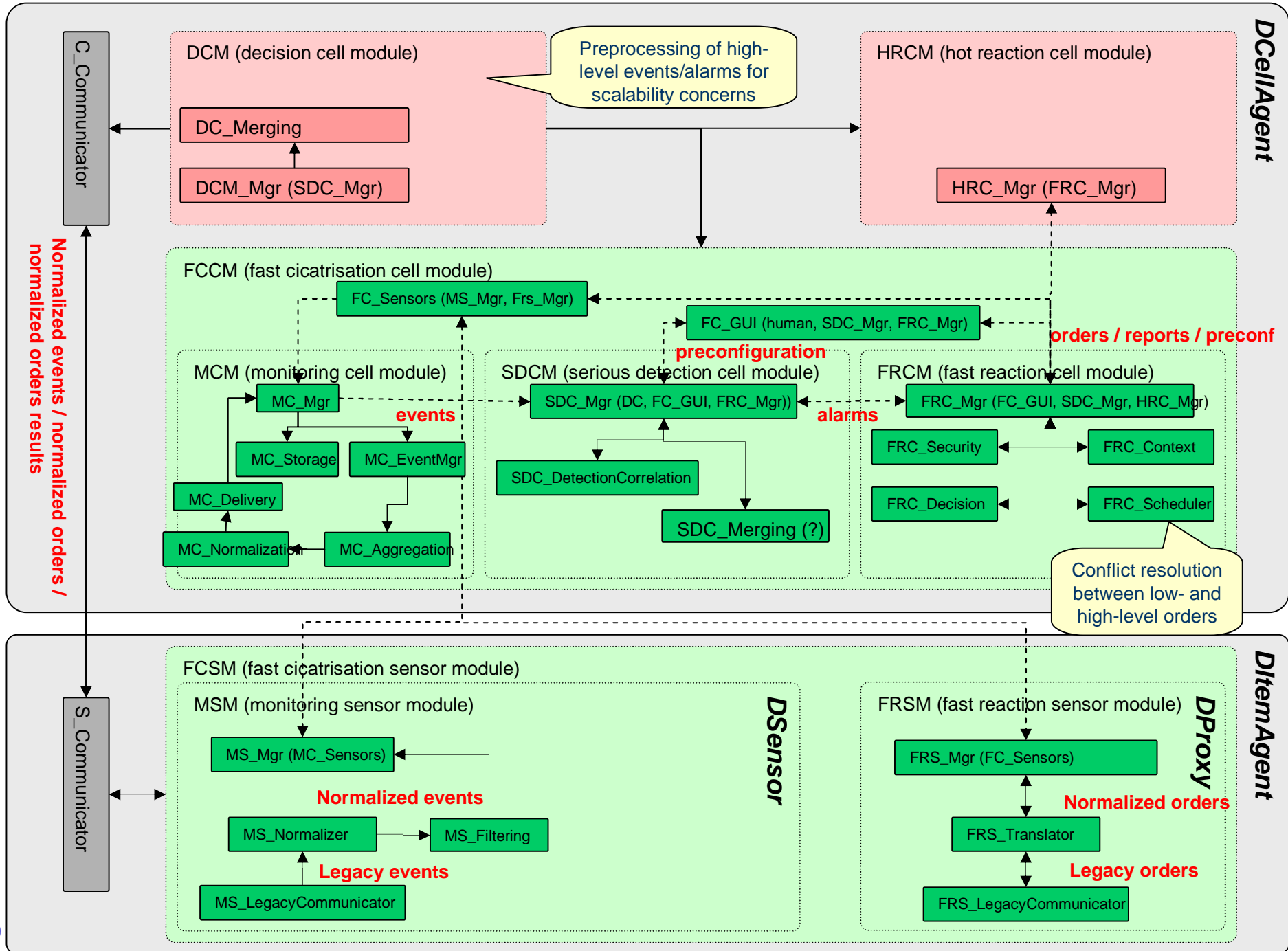
This is a molecule of type « MA ».



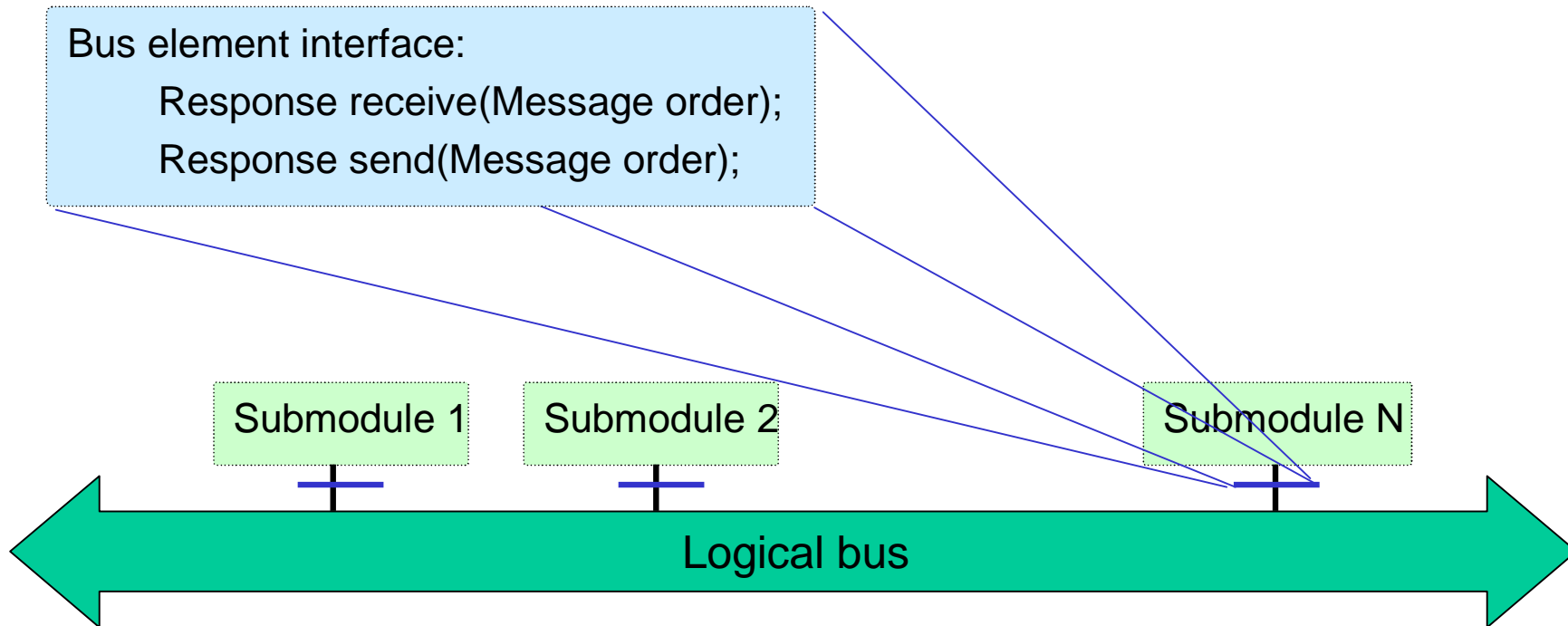
## - DItemAgent (D-Proxy & D-Sensor)



# -Internal Structure



# - Modules Inter-Communication Bus – Overview



# - Modules Inter-Communication Bus – Message Semantic

## Address

CellLocation (*None, <ipaddress>*)

SensorLocation (*None, <ipaddress>*)

ModuleId (*M, FR, D*)

SubModuleId(*Mgr, Decision, ...*)

MessageType (*Event, Action, ActionResult*)

## Message

### Header

*SourceAddress*

*DestinationAddress*

*Type*

### Body

*Data1*

*Data2*

*....*

### Integrity

*DRAFT Version !!!*



# - Modules Inter-Connection – Overview

