Initial Architecture for DESEREC
(currently foreseen)

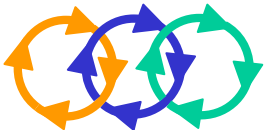
Maximilian List
IABG mbH (DEU)

September 25th, 2006

DESEREC
Dependability Security by Enhanced Reconfigurability

Information Society Technologies

Table of Content



What Do We Have In Mind

Design Requirements & Guidelines (1)

- 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)

- Use existing standard means of monitoring and intrusion detection for information system surveillance

- Highly secured protocol between agents and centralized component of the DESEREC framework: DESEREC framework shall not constitute a weakness for the Information System security

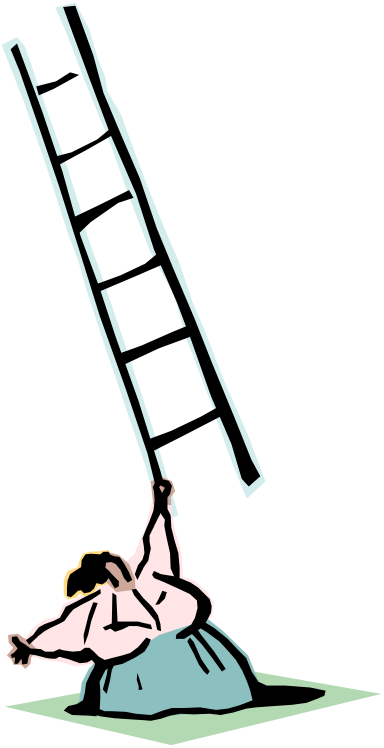
Design Requirements & Guidelines (2)

- 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
- Ability to deploy global (hot) reconfiguration orders over the whole information system in a coordinated and fast way
- 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 of DESEREC

The Goal

Increase dependability of
• critical,
• open, and
• interconnected
IS (Information System)

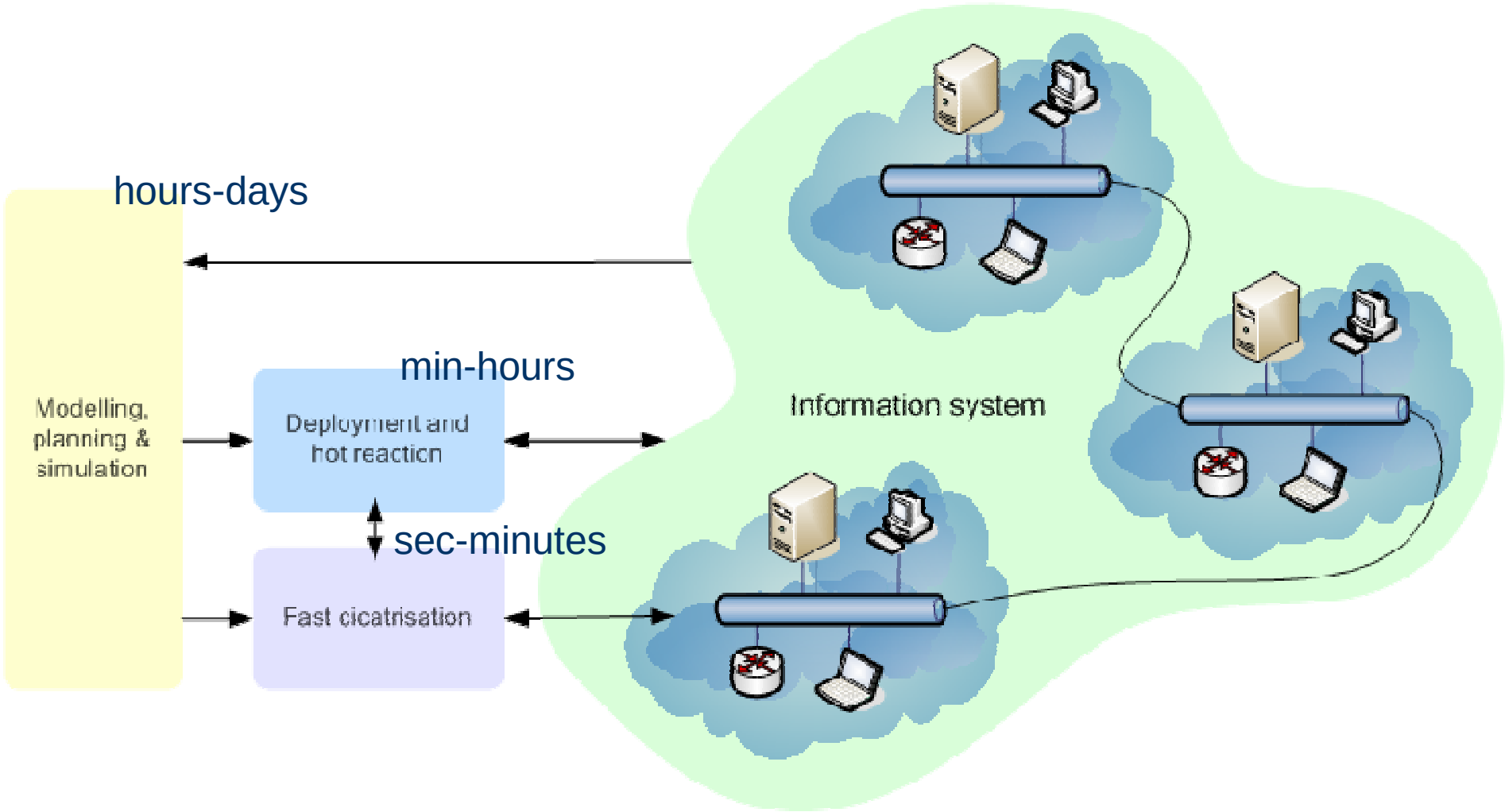


DESEREC Initial Approach

Proposed Solution:

- Three-tiered response to exceptions and incidents:
  - Modeling, Planning, and Simulation
  - Deployment and Hot Reconfiguration
  - Fast Cicatrisation
- These three global modules:
  - act at different time scales (e.g. days, hours, minutes)
  - represent different abstraction layers of the DESEREC system
  - 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

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

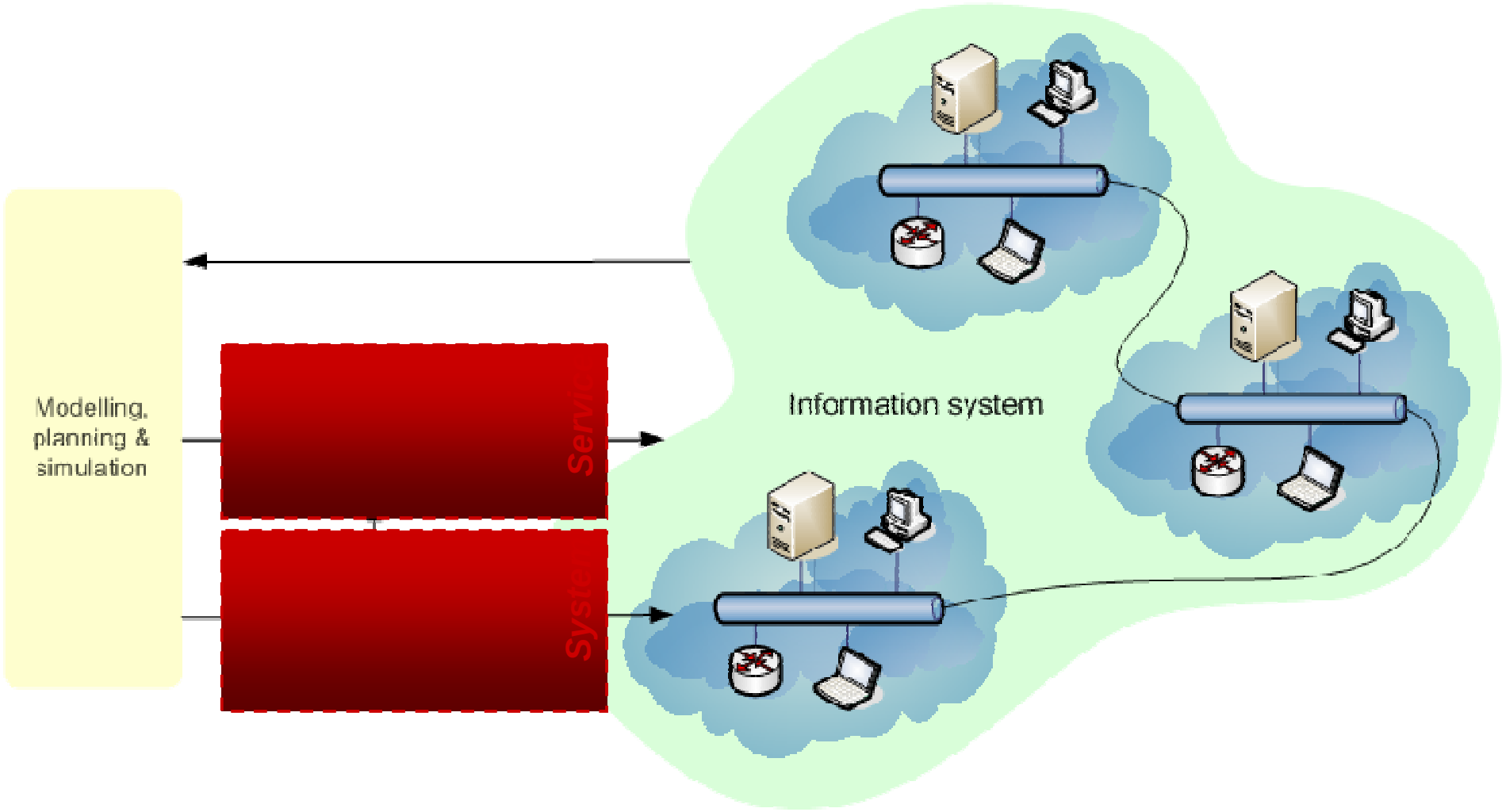
Deployment & Hot Reconfiguration

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

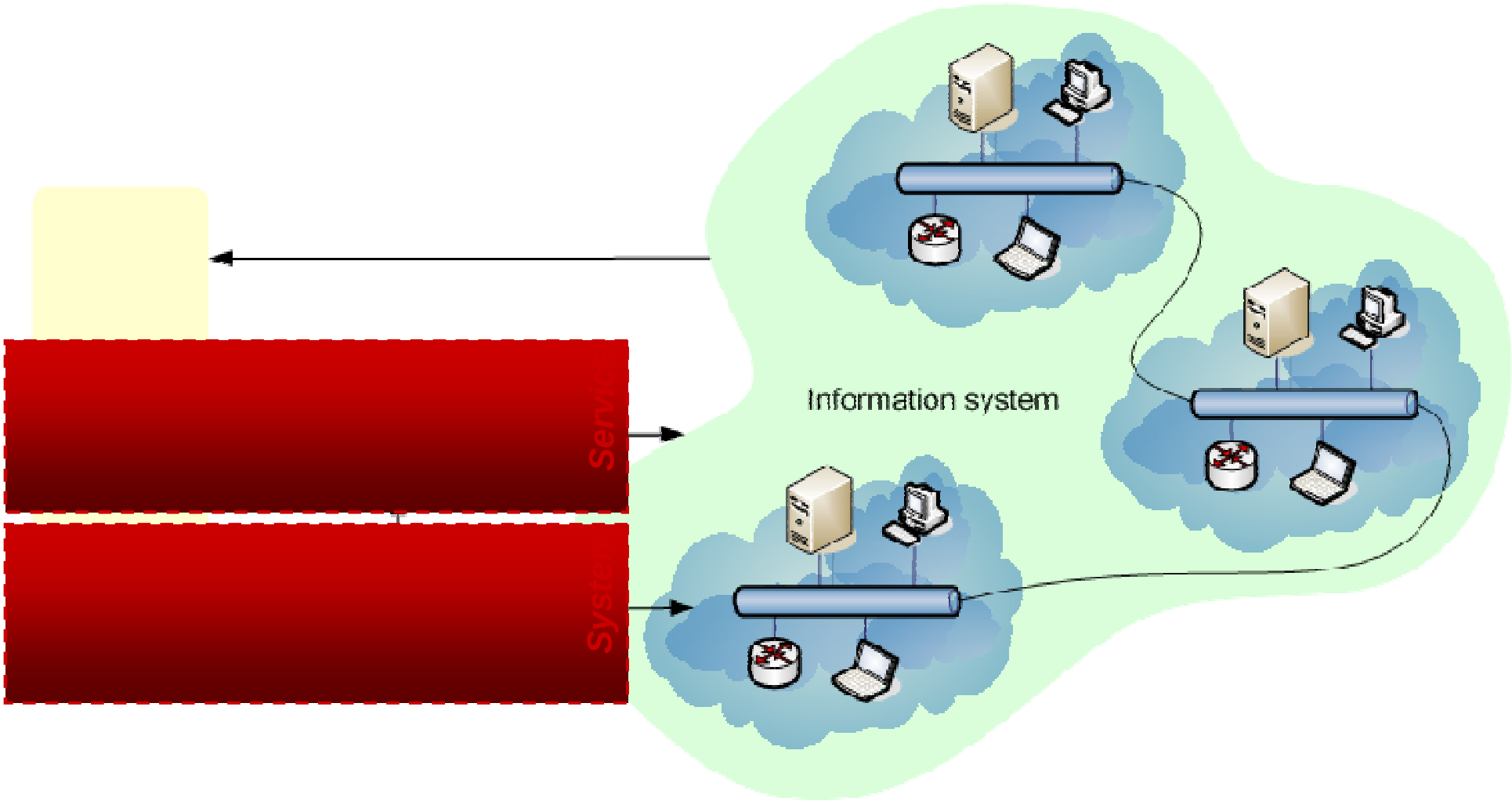
Fast Cicatrisation

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

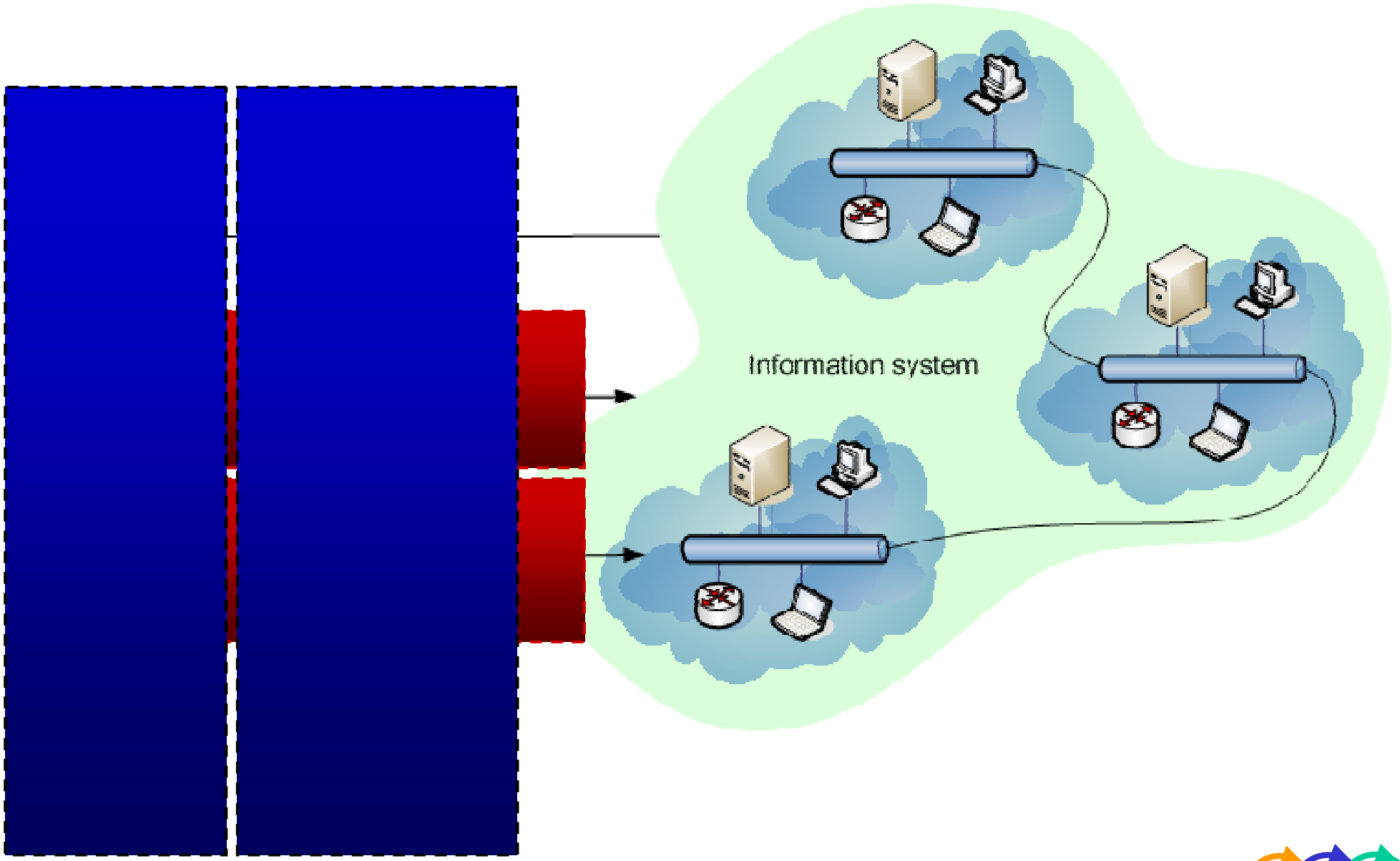
High Level Architecture – Overview



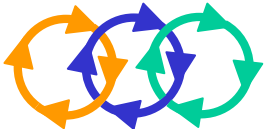
High Level Architecture – Overview



High Level Architecture – Overview



Details About The Initial Architecture



Conceptual Thoughts

Some basic concept thoughts

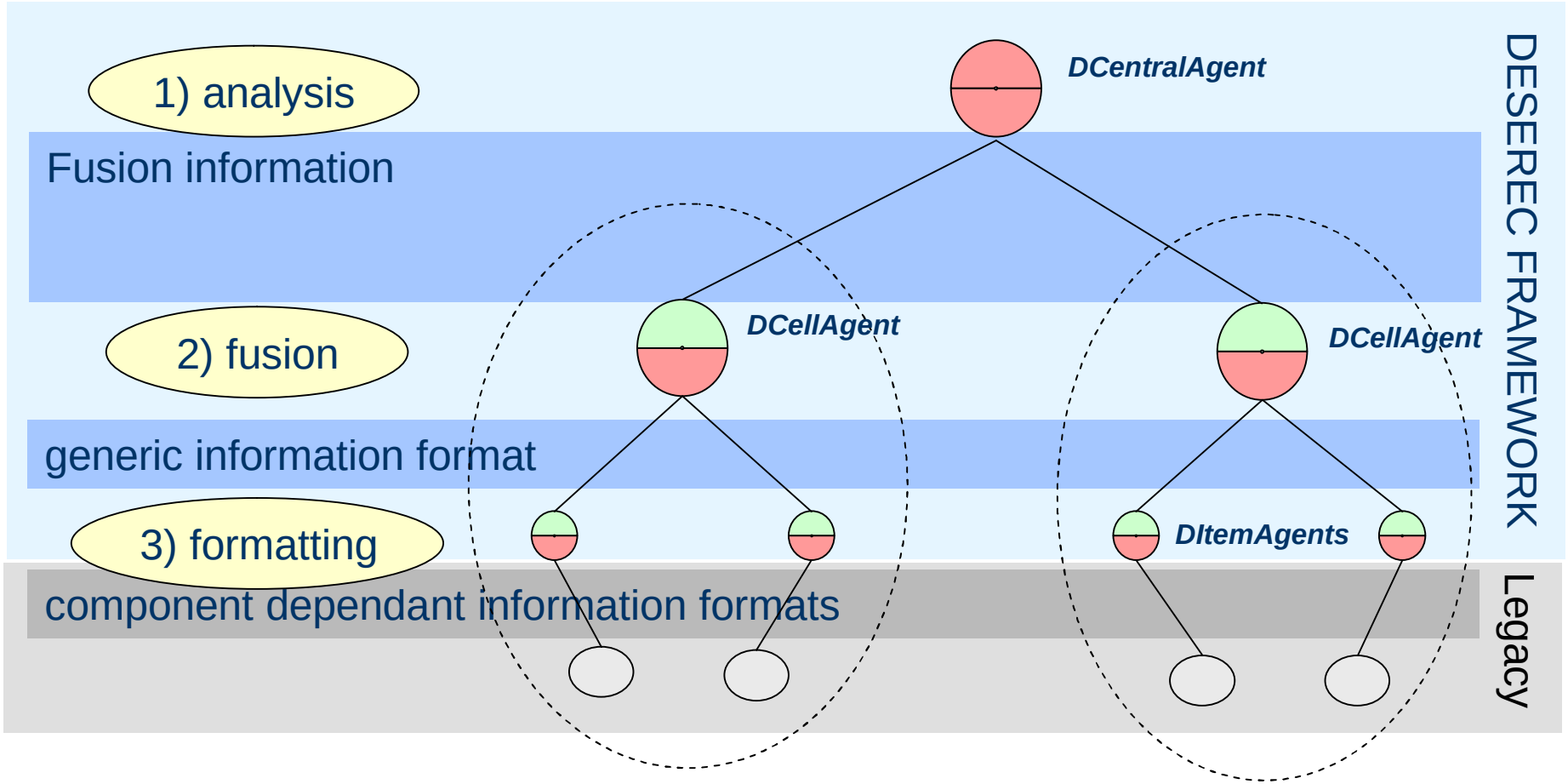
- normalization and standardization
  - of Events, Actions and Responses
  - common protocol and format

- hierarchical vs distributed
  - base structure is hierarchical (3-level architecture : Central - Cell - Item)
  - low-level interaction on a P2P base (fast exchange bw Cells)

- re-use
  - of functions at different DESEREC layers and modules (e.g. Translation)

- overlapping of cells (to be discussed)
  - sensor information collected across cell boundaries
  - !!! atomic reactions limited to one cell (avoid blocking or concurrent device access)

Three Stages Logical Topology

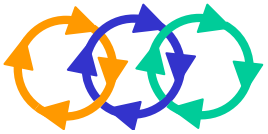


- central agent **è** large vision (i.e. detection of DDOS)
- cell agent **è** scalability (reduce supervision bandwidth)
- item agent **è** adaptation to legacy interfaces

Terminology & Components (1)

DCentralAgent

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



Terminology & Components (2)

DCellAgent

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

Terminology & Components (3)

DItemAgent

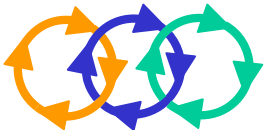
- Comprises DSensor and DProxy

DSensor

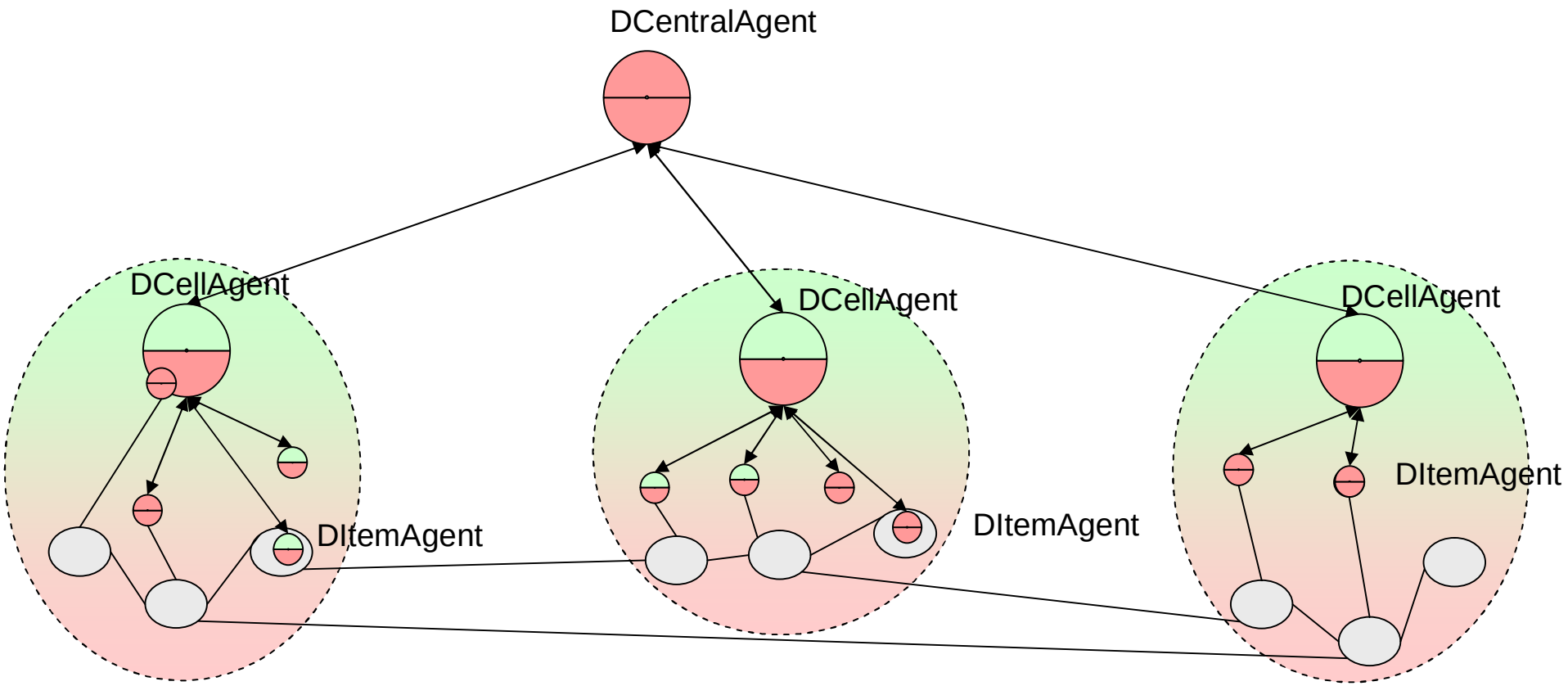
- Monitors target devices/applications
- Event collector
  - Able to filter incidents / legacy events
  - Able to translate to a common event format **à** *CEP / IODEF*
- First Event aggregation

DProxy

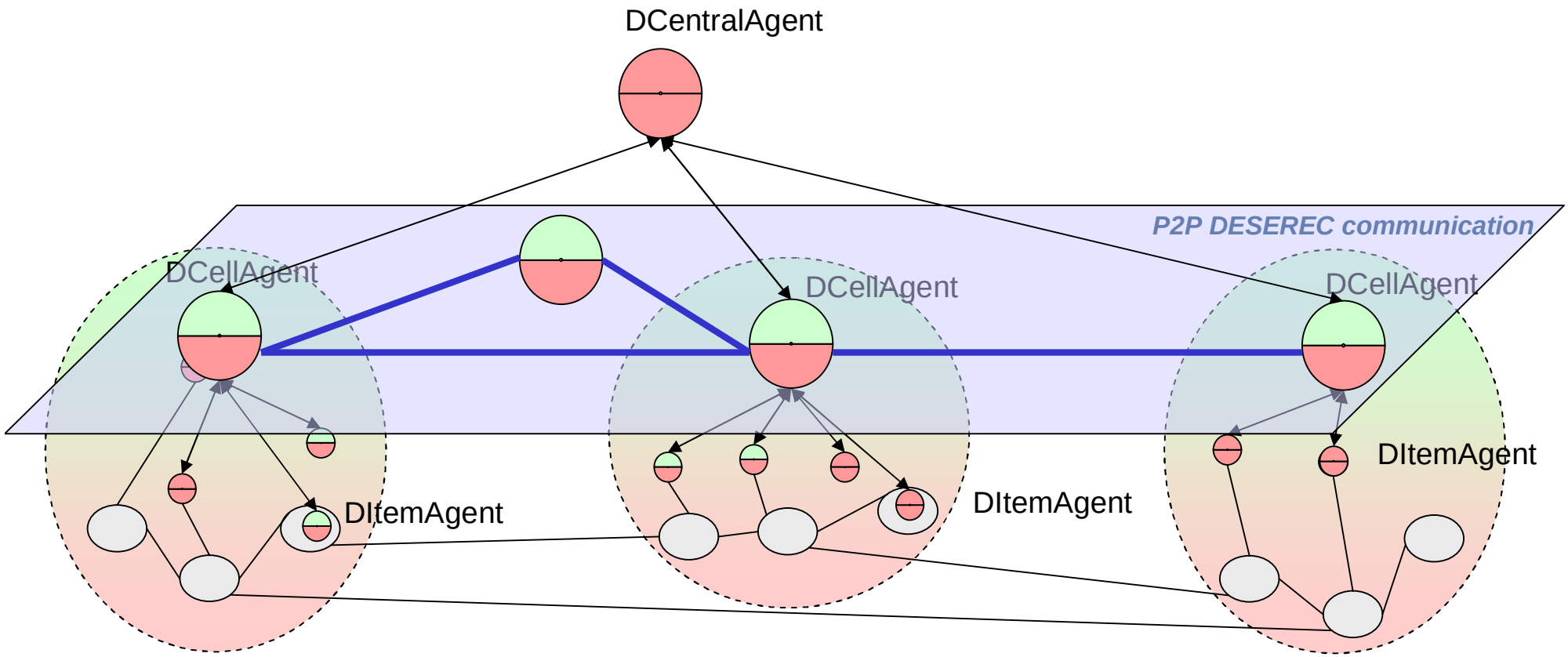
- Translator
  - Able to enforce configurations on target device/application
  - Able to manage low level translation conflicts
- Deployment on target device/application level



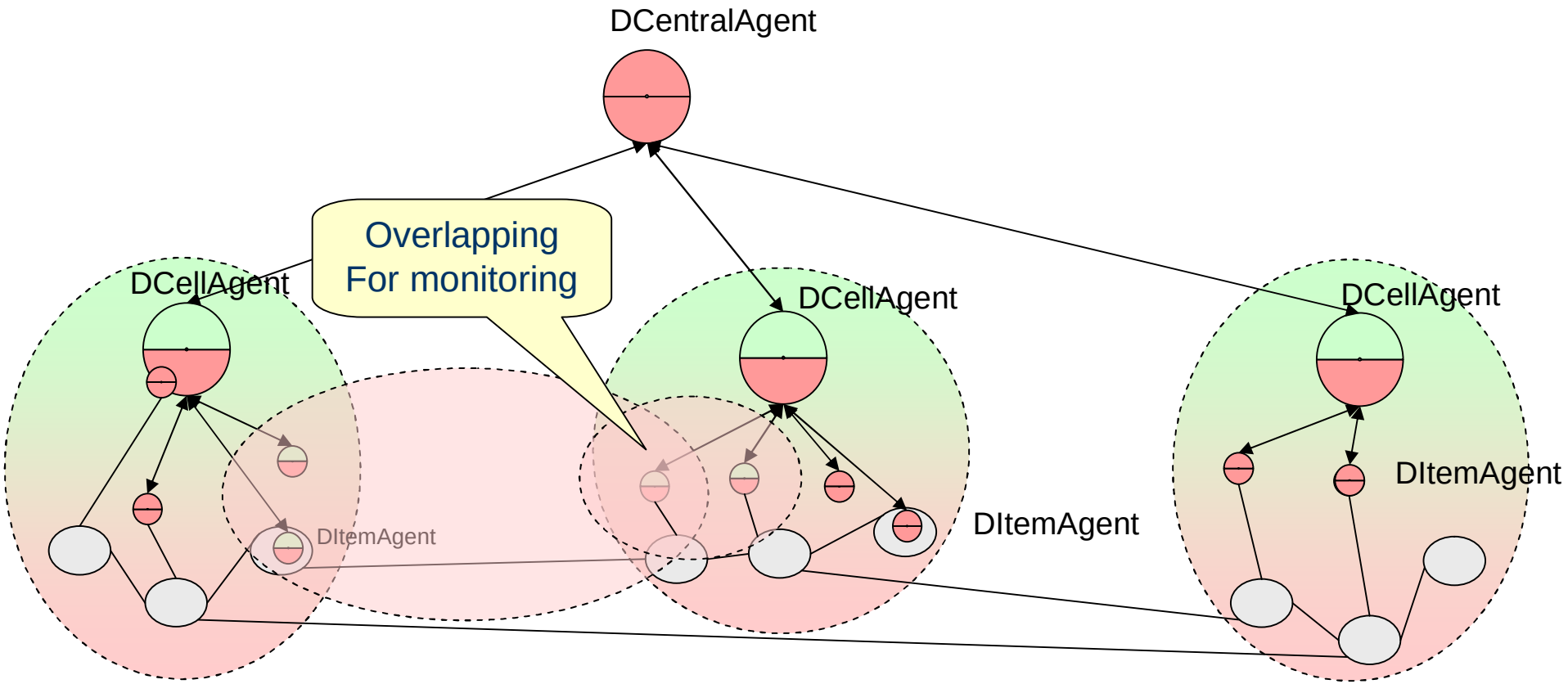
Overall Architecture – Hierarchical View



Overall Architecture – Distributed P2P Cell Communication

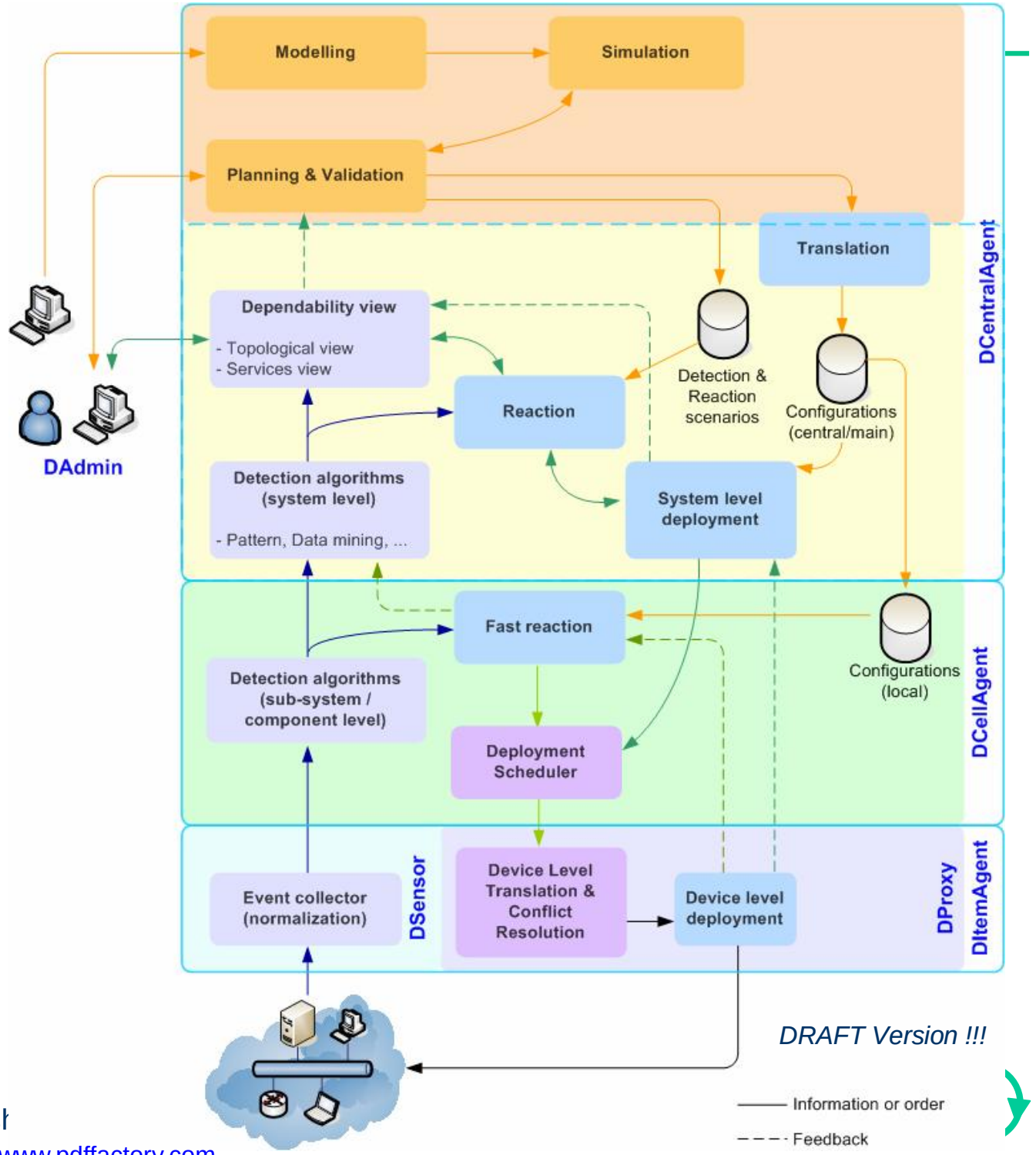


Overall Architecture – Cell Overlapping

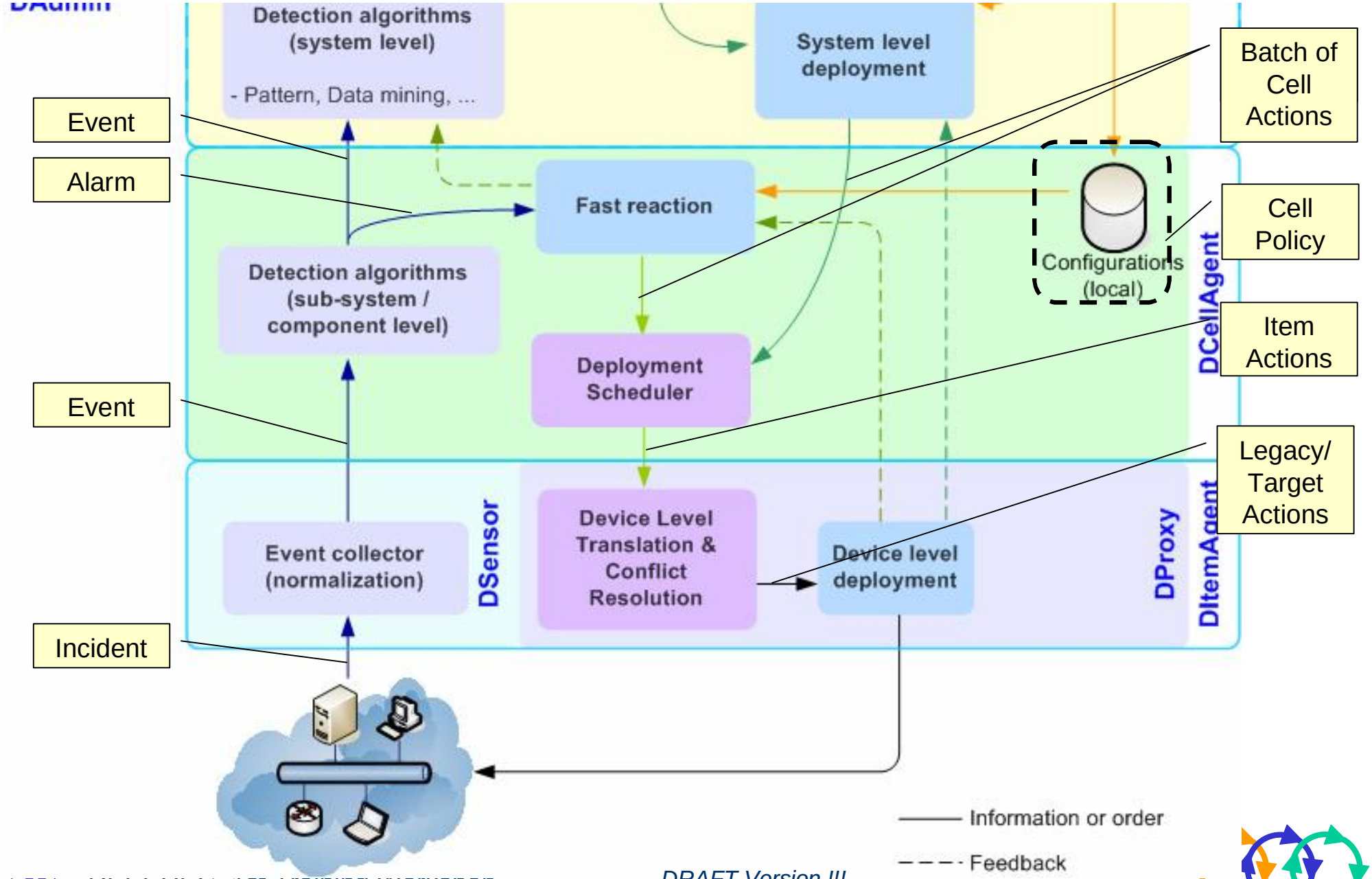


Draft : to be discussed!!!

High Level Architecture & Modules – Functions

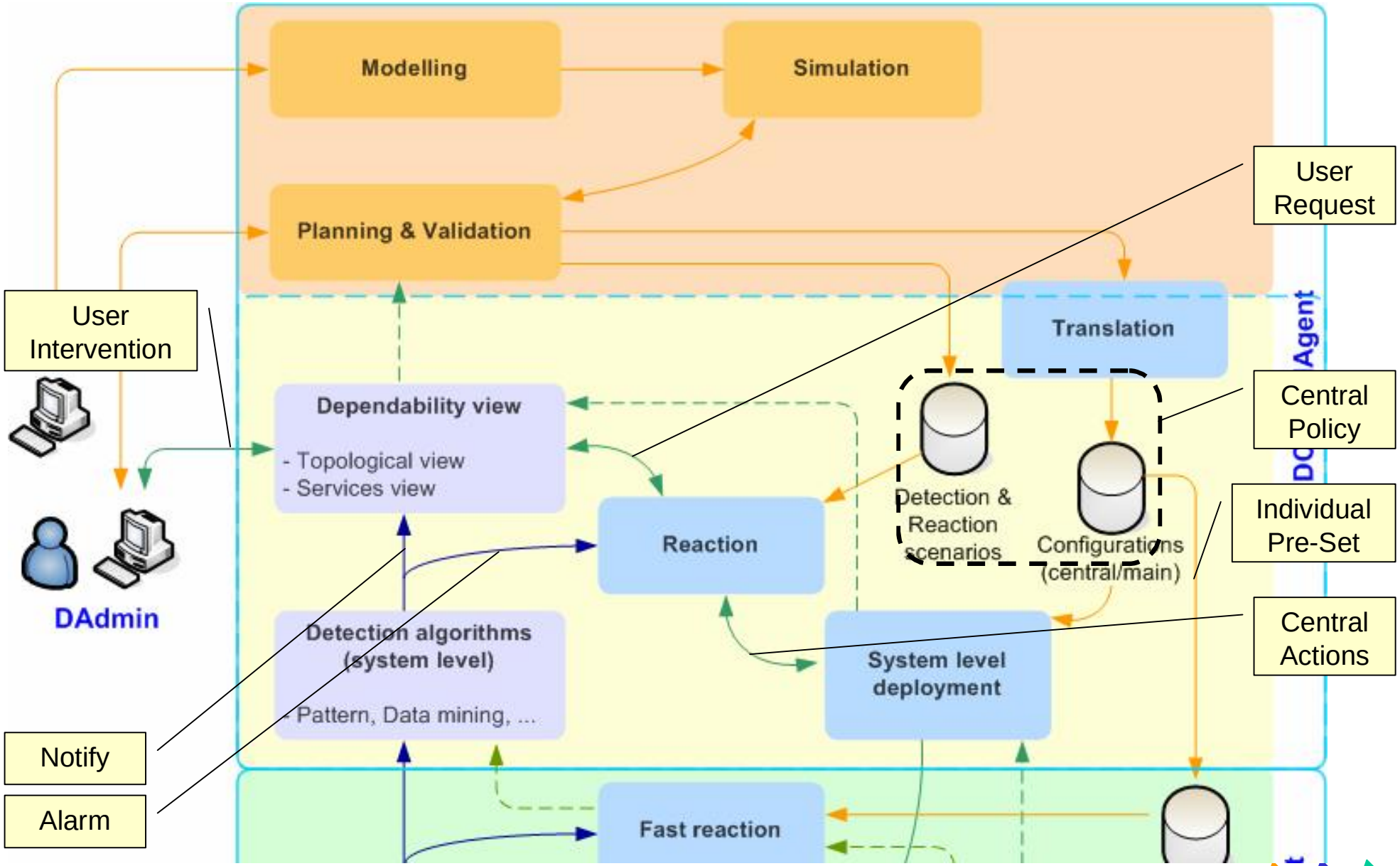


High Level Architecture & Modules – Functions



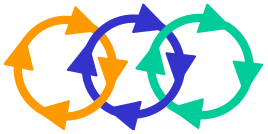
DRAFT Version !!!

High Level Architecture & Modules – Functions

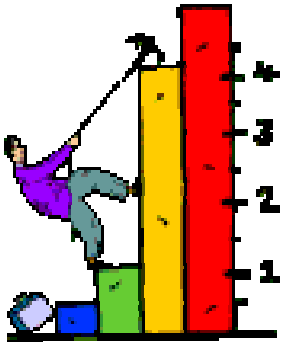


DRAFT Version !!!

The Way Ahead



Next Steps



Finalize

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

Future Steps

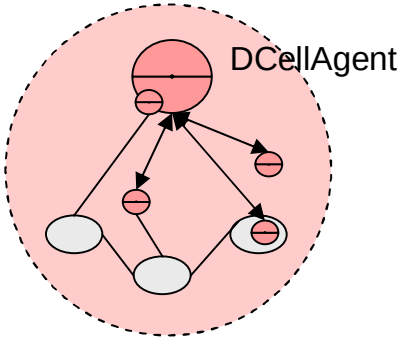
- Self-Learning and -Healing mechanisms (intelligence)
- …

Questions ???

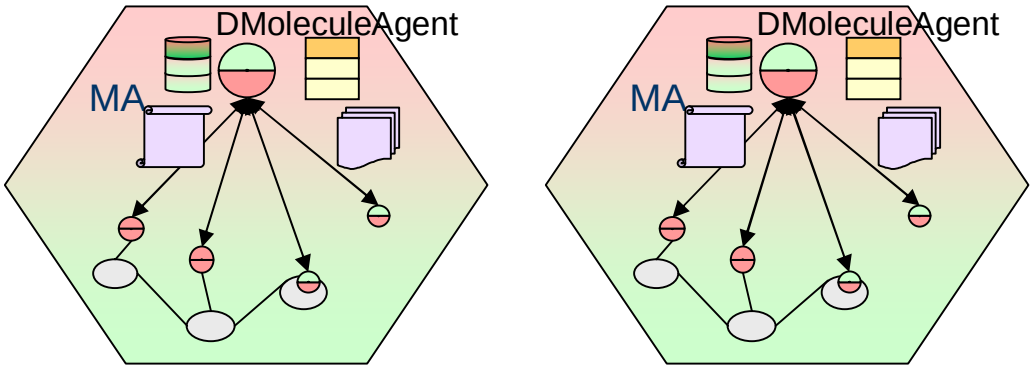
Annex

Cell and molecule

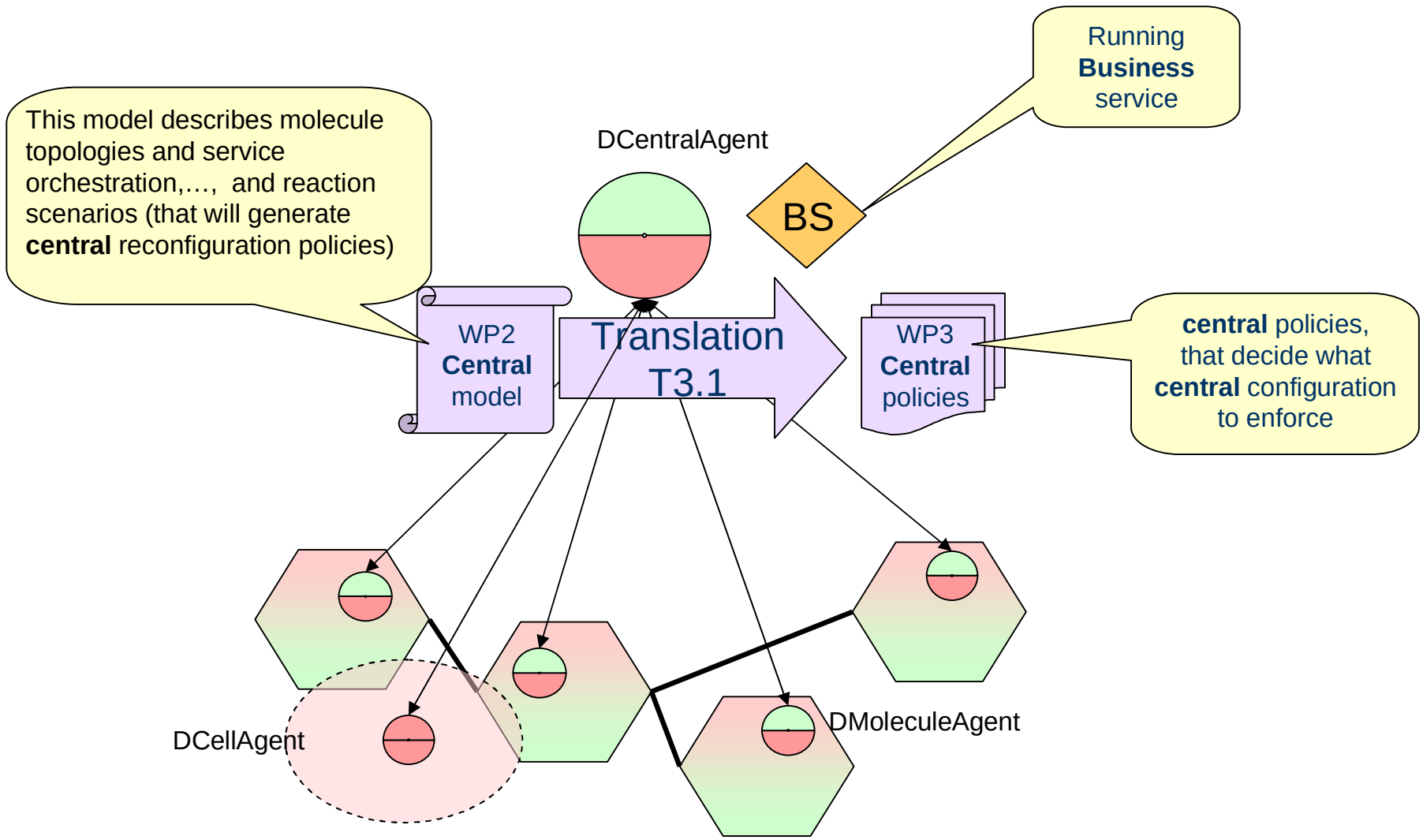
- A **cell** is a aggregation of all system elements (hardware and software) participating to a specific global service monitoring.
  - Cells may overlap molecule for monitoring purpose
  - It can be managed by a Cell Agent (which is centralize or not)



- 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

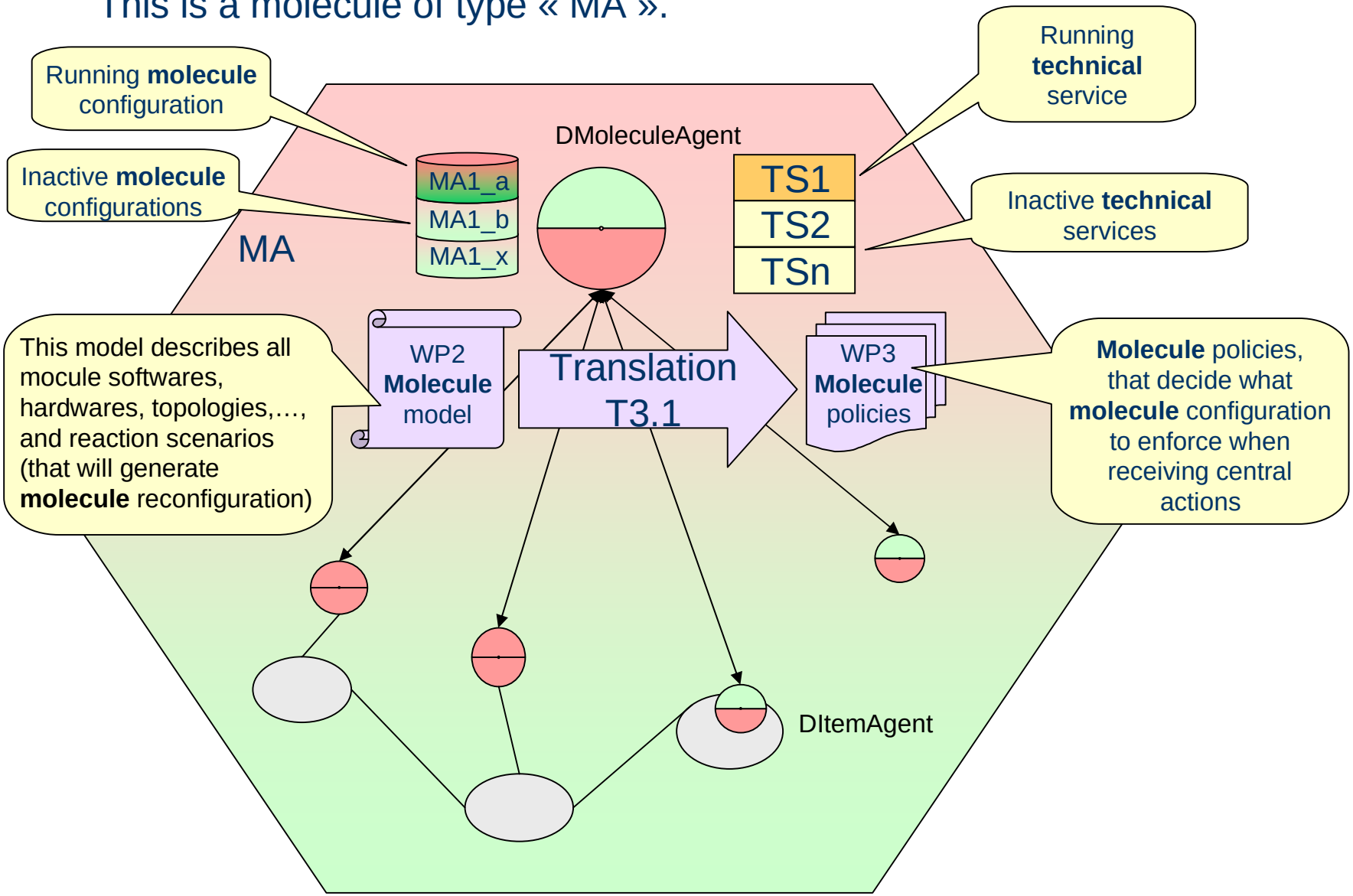


Central description

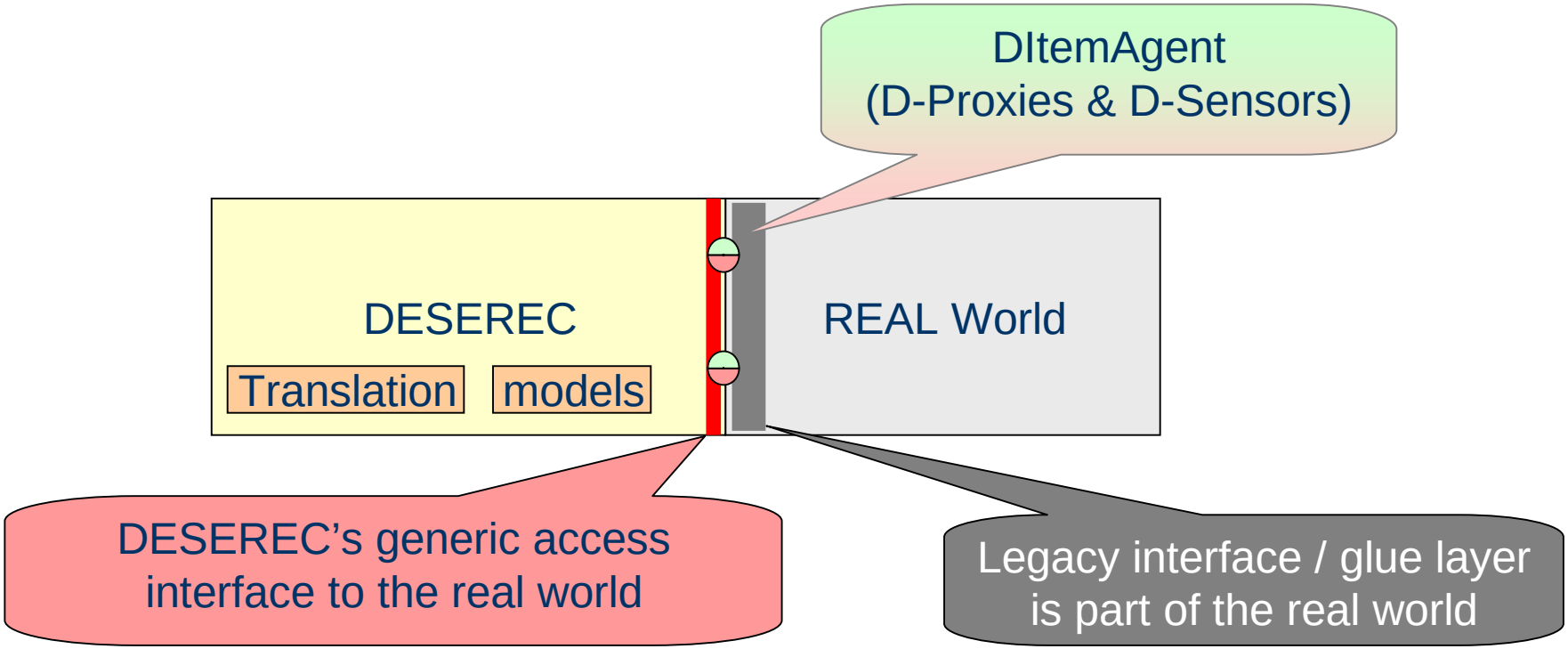


Molecule description

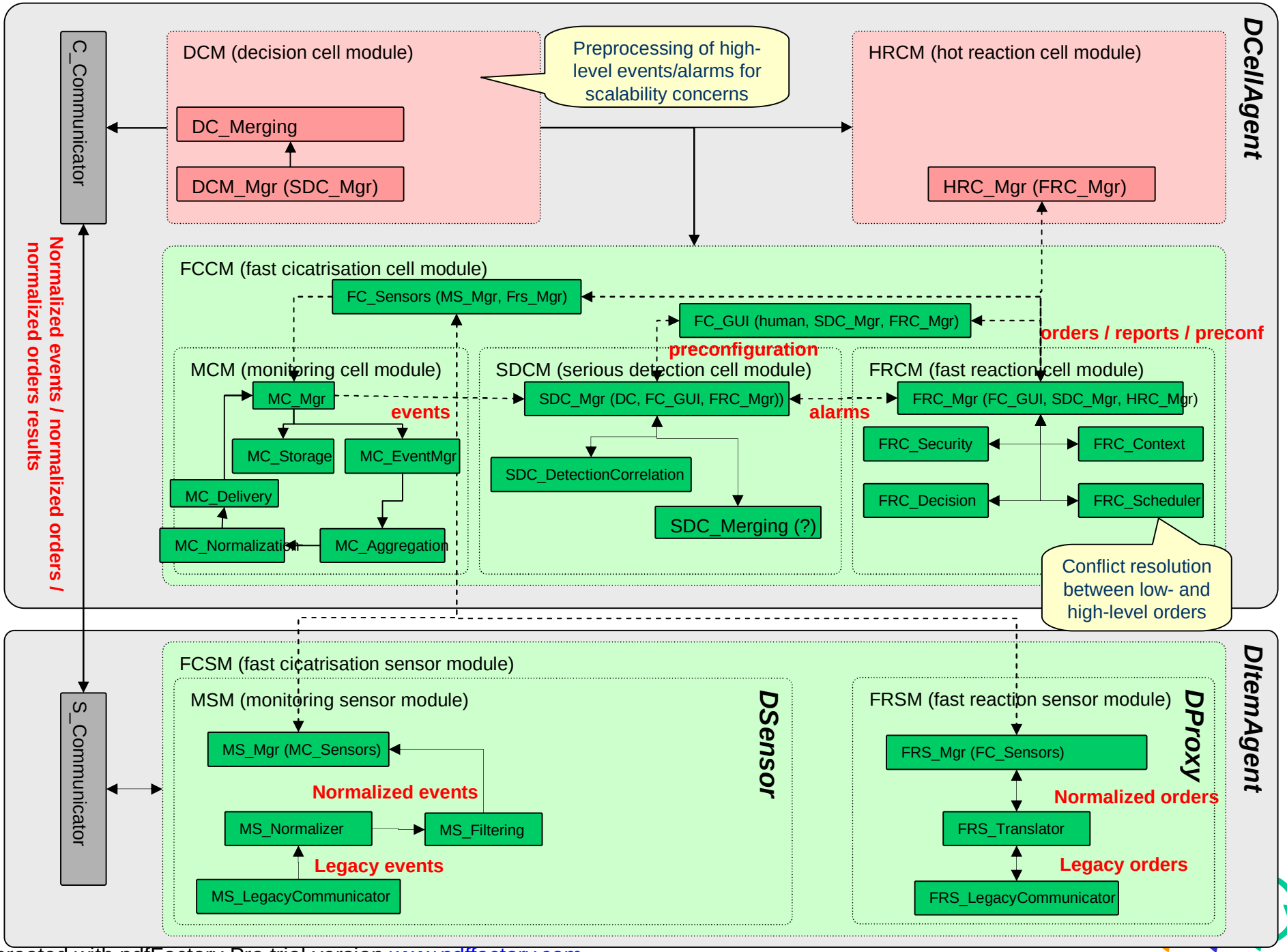
This is a molecule of type « MA ».



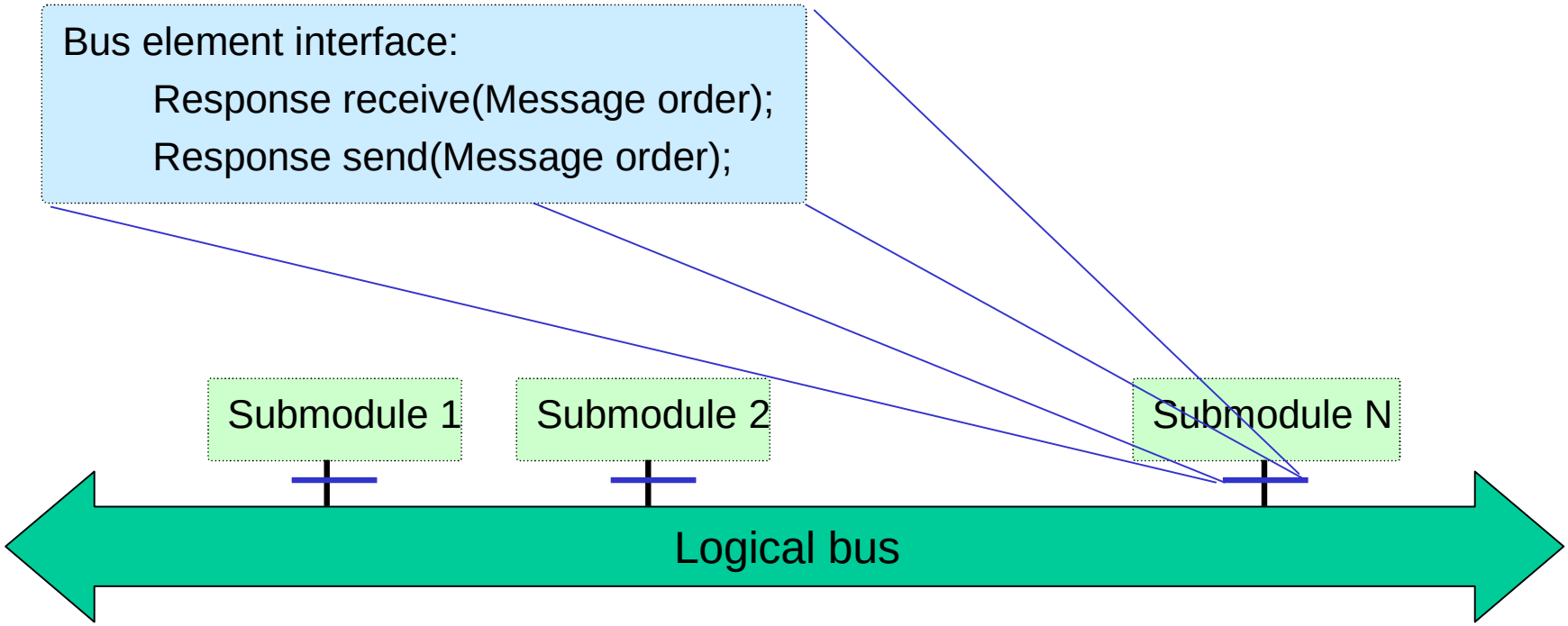
DItemAgent (D-Proxy & D-Sensor)



Internal Structure



Modules Inter-Communication Bus – Overview



Modules Inter-Communication Bus – Message Semantic

Address

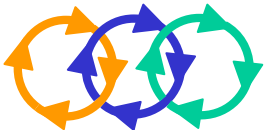
- CellLocation (*None, <ipaddress>*)
- SensorLocation (*None, <ipaddress>*)
- ModuleId (*M, FR, D*)
- SubModuleId(*Mgr, Decision, …*)

MessageType (*Event, Action, ActionResult*)

Message

- Header
  - *SourceAddress*
  - *DestinationAddress*
  - *Type*
- Body
  - *Data1*
  - *Data2*
  - *….*
- Integrity

DRAFT Version !!!



Modules Inter-Connection – Overview