SHIELD
PILOT

WP5 – SPD Middleware and Overlay

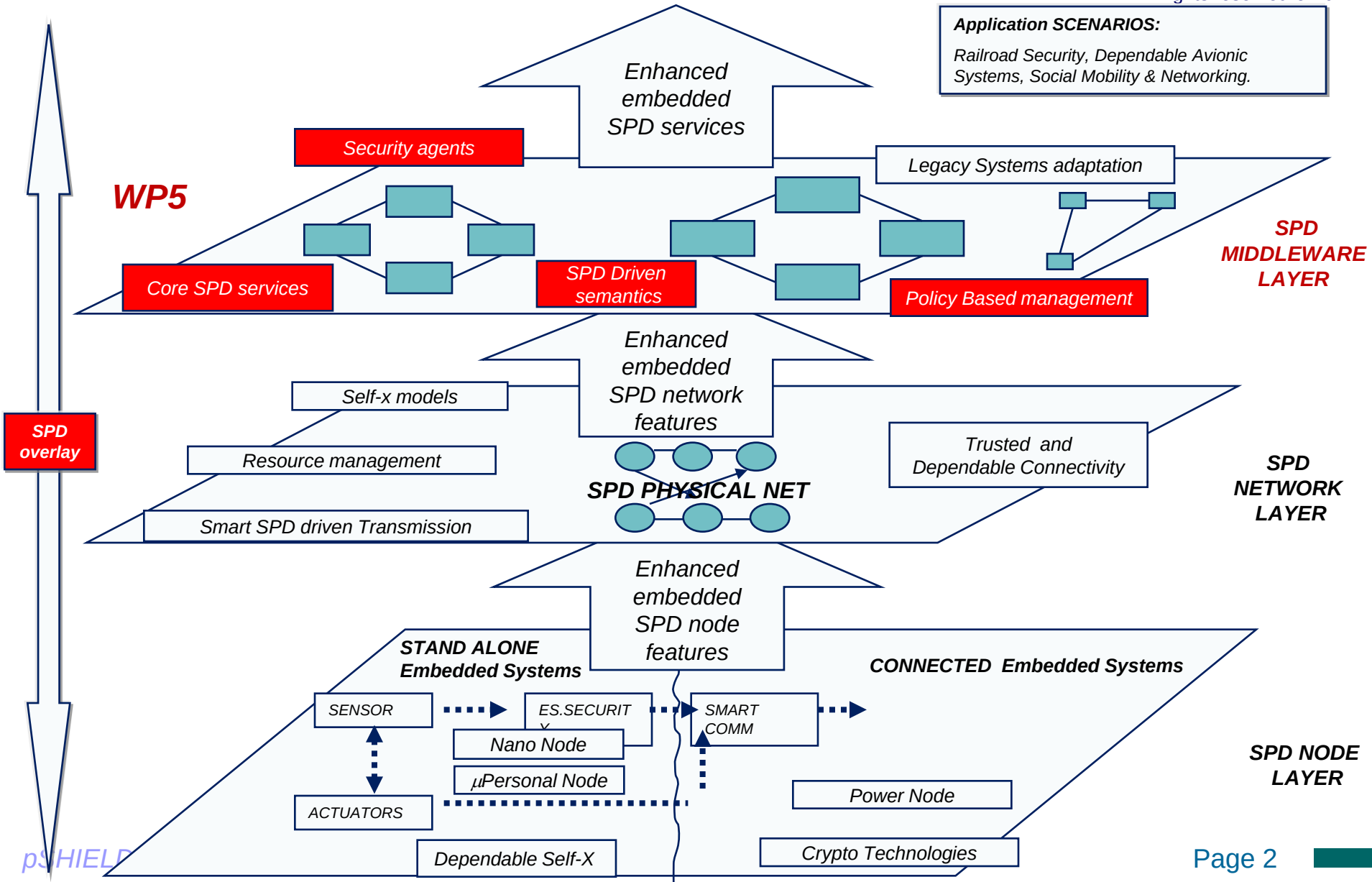
Final Review Meeting

Bruxelles, 14th February 2012

Andrea Fiaschetti – Andrea Morgagni

ARTEMIS Call 2009 – SP6100204

WP5 – Contextualization with respect to pSHIELD



WP5 – Objectives and Information Flow

From pSHIELD Technical Annex, the objectives of WP5 are:

- Define a common **semantic** to describe the SPD interfaces and functionalities

  INVOLVED ACTIVITIES:

  - *Task 5.1 SPD driven Semantics*
  - *Task 5.2 Core SPD Services*
  - *Task 5.3 Policy-based management*

- Introduce the **Overlay** concepts and functionalities

  INVOLVED ACTIVITIES:

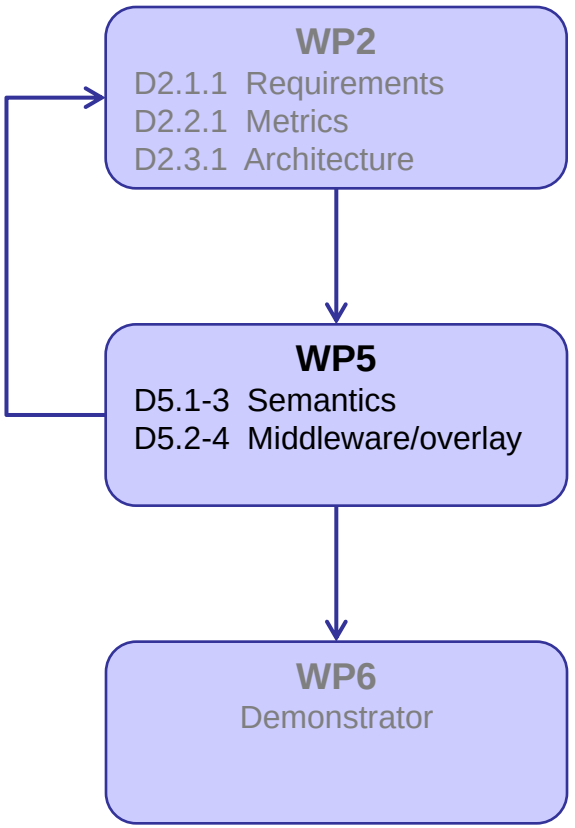
  - *Task 5.4 Overlay monitoring and reacting system by security agents*

- Develop a **prototype** to be integrated in the demonstrators

  INVOLVED ACTIVITIES:

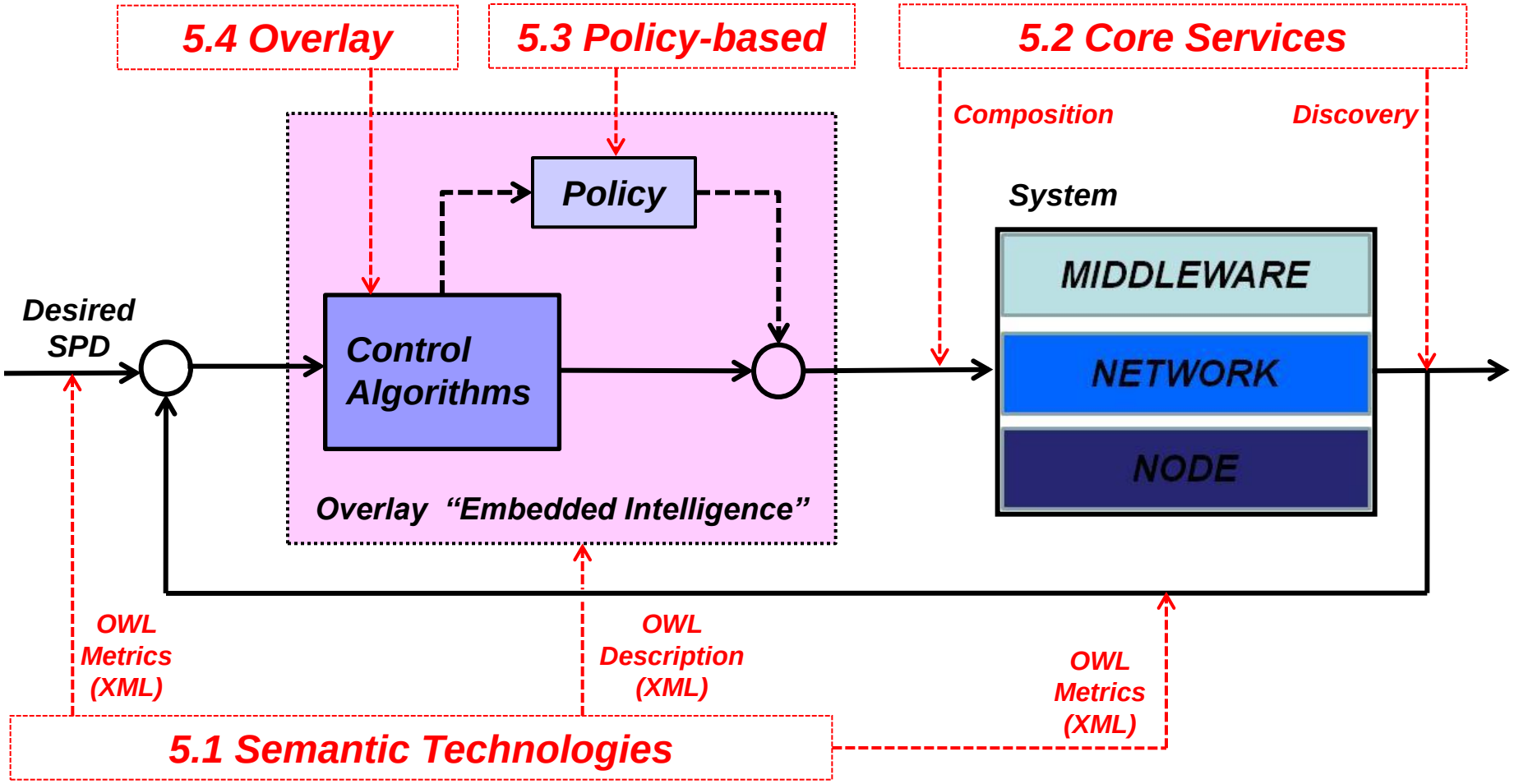
  - *All Tasks*

WP5 Information Flow



WP5 – Synoptic Representation

In order to realize the pSHIELD key concepts, the tasks are mapped and justified in this way:



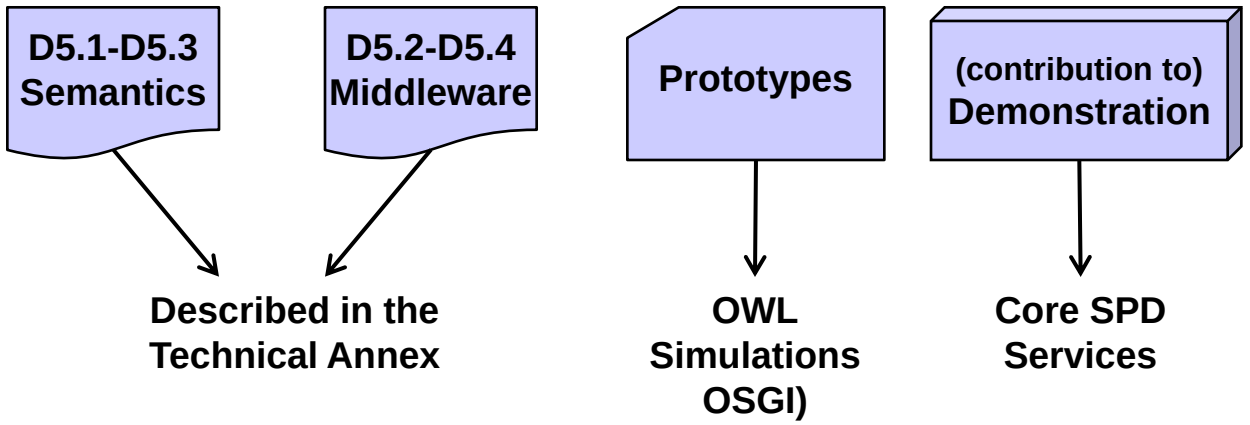
WP5 – Delivered Outputs

From pSHIELD Technical Annex, the outputs of WP5 are:

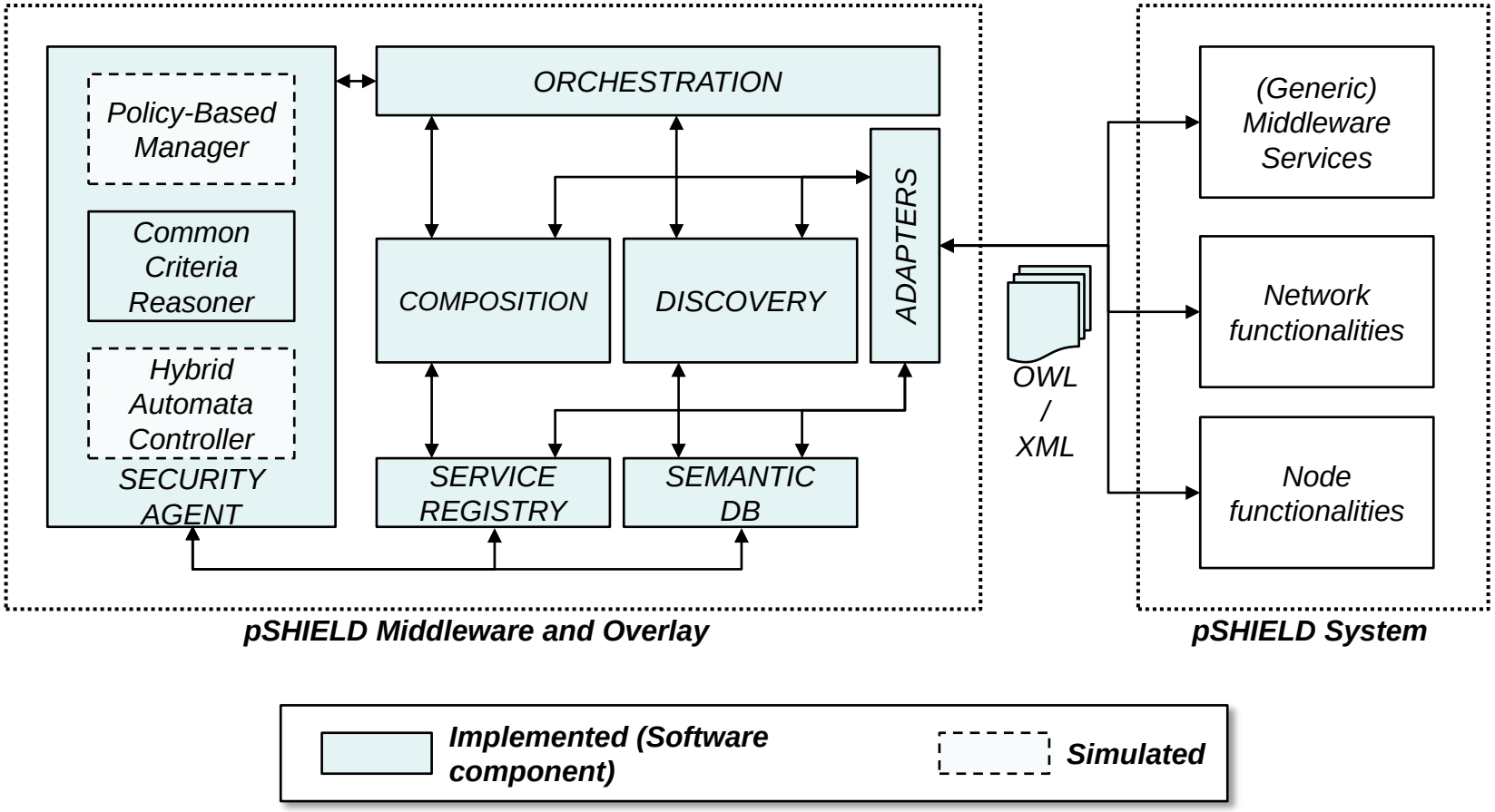
TASKS vs OUTPUT →	D5.1-D5.3 Semantics	D5.2-D5.4 Middleware	Prototypes	Demonstration
Task 5.1 Semantics	X		X	
Task 5.2 Core Services		X	X	X
Task 5.3 Policy-based		X		
Task 5.4 Overlay		X	X	

The nature of prototypes is "heterogeneous", i.e. they could be hardware or software modules, simulation results, high level design documents and so on.

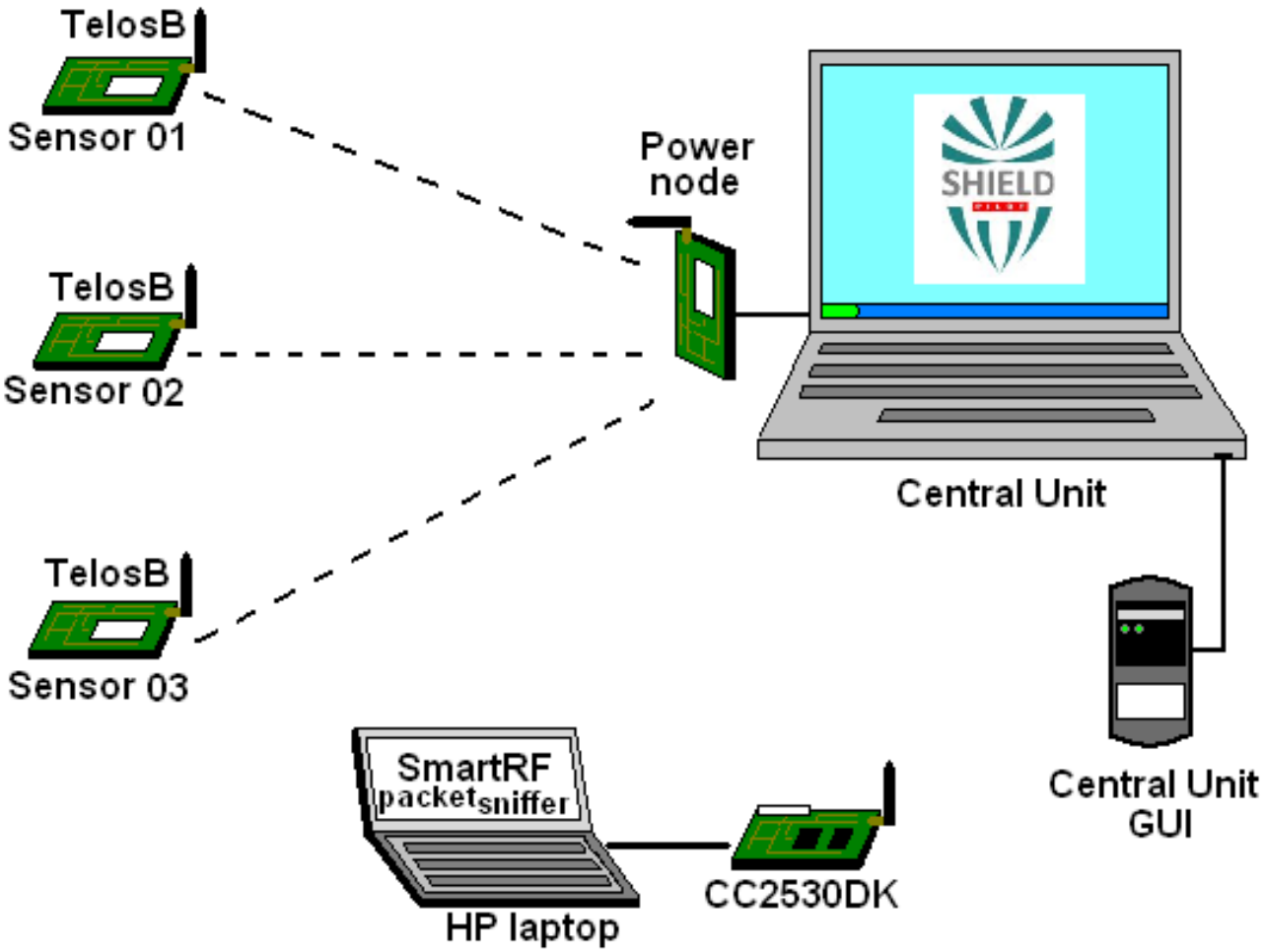
They are included as Annex in D5.x deliverables



WP5 – Rationale of the outputs



WP5 – Prototypal Demonstrator



WP5 – Achievements

- Drawing of an original ontological model of ESs, including the semantic characterization of the system and inferential engine features (based on specific metrics) to face the SPD composability problem

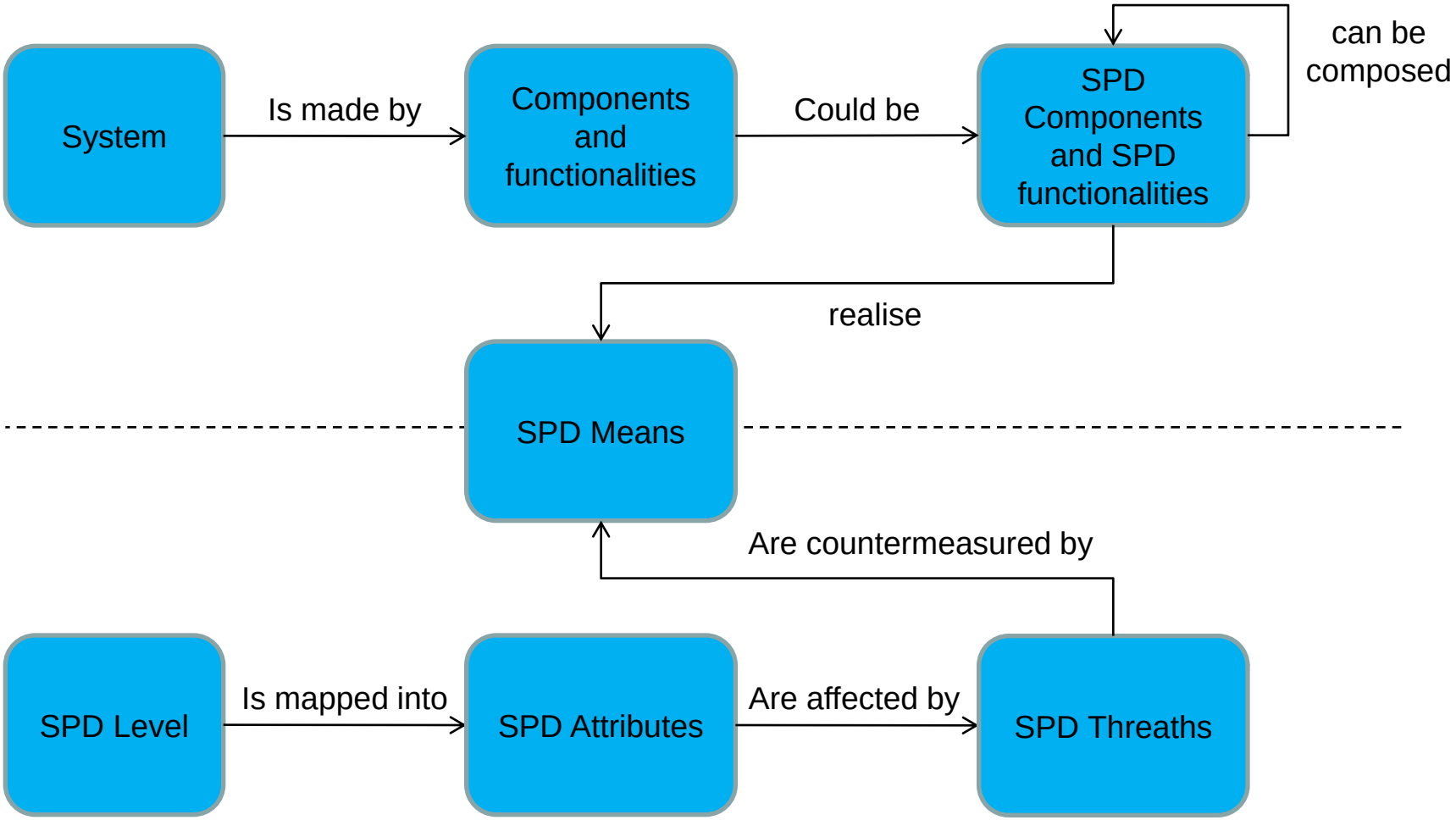
- Design and implementation of a reduced but significant “working” example of the pSHIELD Middleware and Overlay. This Middleware is able to discover and compose SPD functionalities to achieve the desired SPD level.

- Technological Assessment of the Policy Based Management for Security applications and preliminary feasibility analysis with respect to pSHIELD

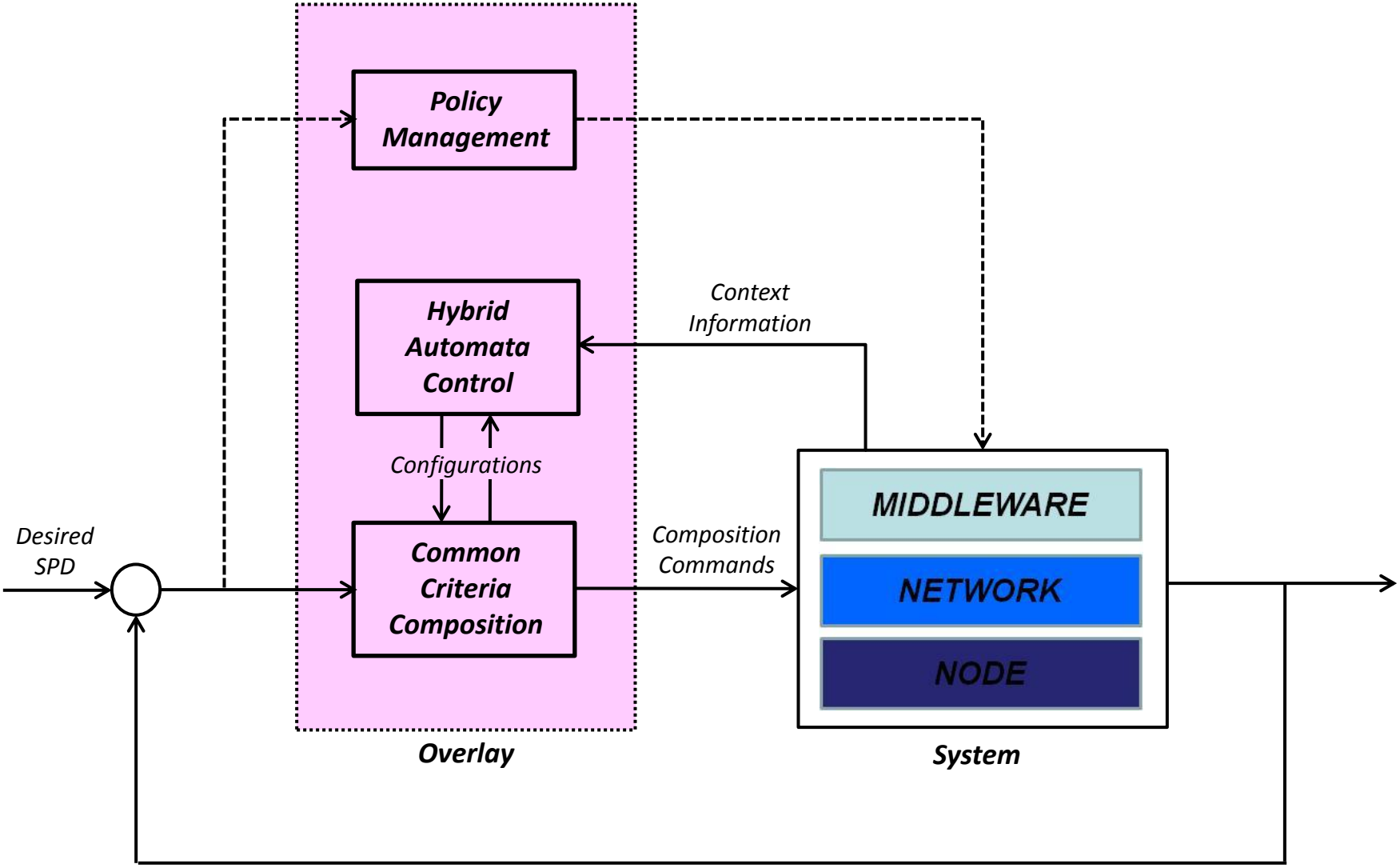
- Formulation of an innovative model to represent (composable) Embedded Systems based on the theory of Hybrid Automata. Thanks to this formulation it has possible to apply some closed-loop control algorithms (like MPC) to optimize the SPD composability in a context-aware way.

WP5 – Semantic in a nutshell

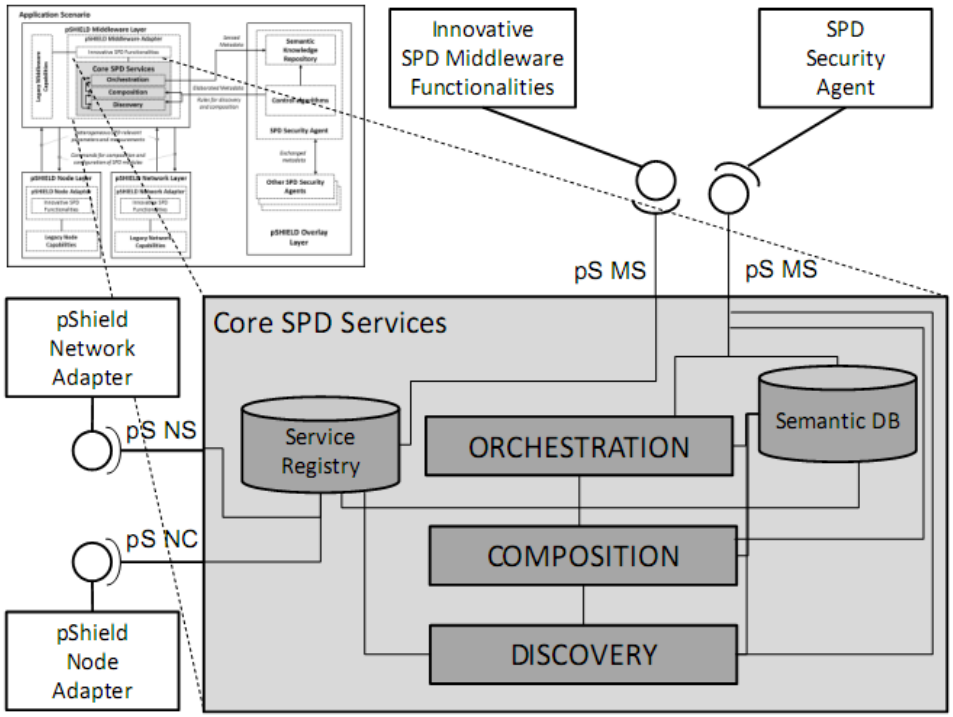
Ontology logical representation: each concept is modelled and the relations are identified in order to have the logical chains that enables the SPD-aware composability

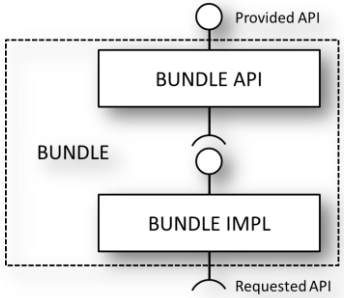


WP5 – Middleware and Overlay in a nutshell



WP5 – Core SPD Services



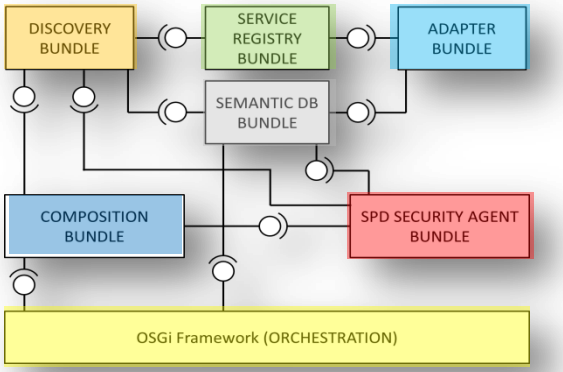


Each service as been implemented as a ***BUNDLE***.

Each bundle is composed by a:

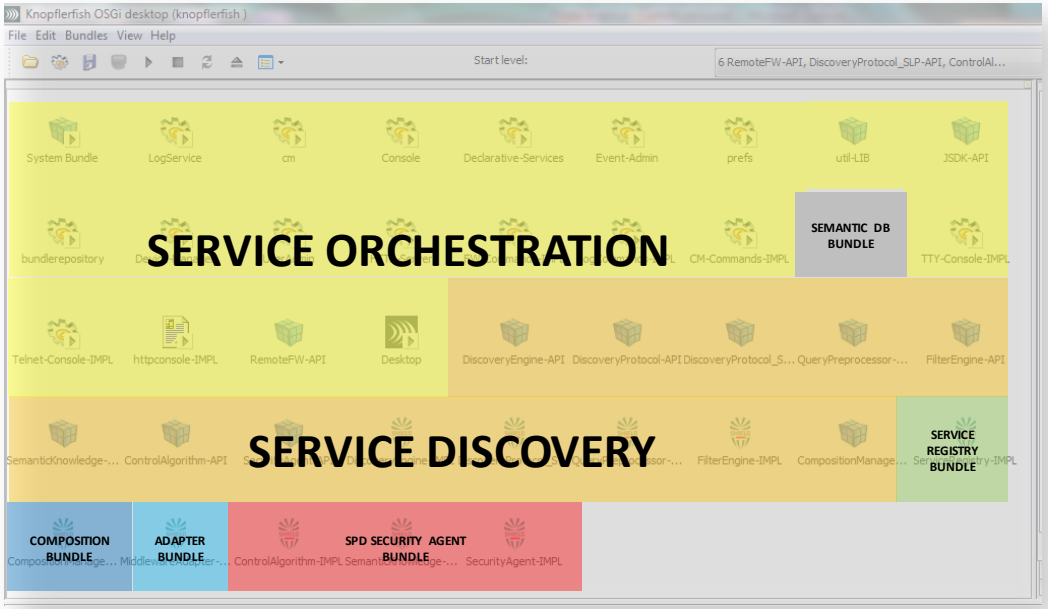
- ***Interface*** (BUNDLE API) that exports APIs
- ***Implementation*** (BUNDLE IMPL) that imports APIs

WP5 – Core SPD Services



WP5 Prototype Architecture

OSGI Platform

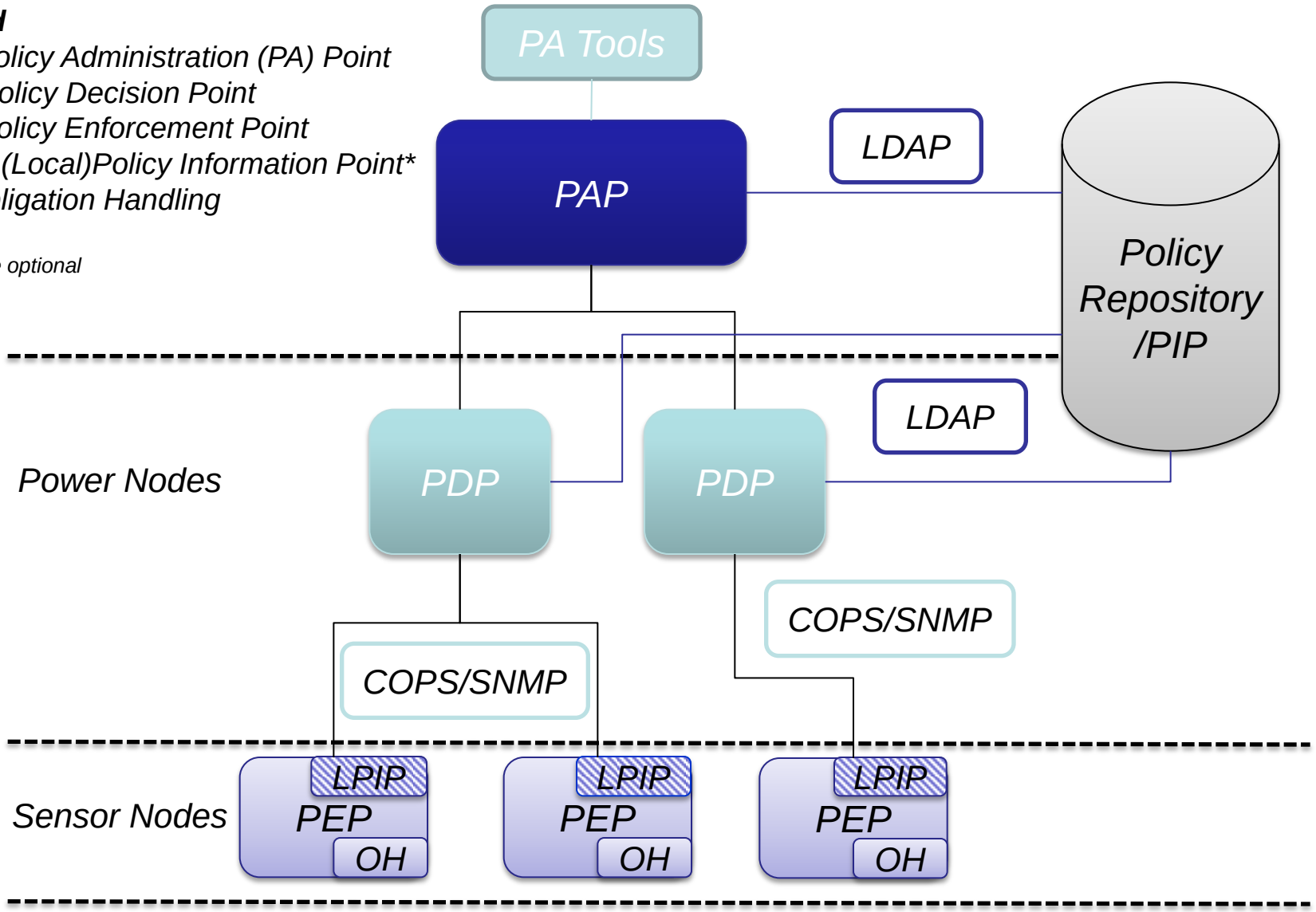


WP5 – Policy Based Management

Legend

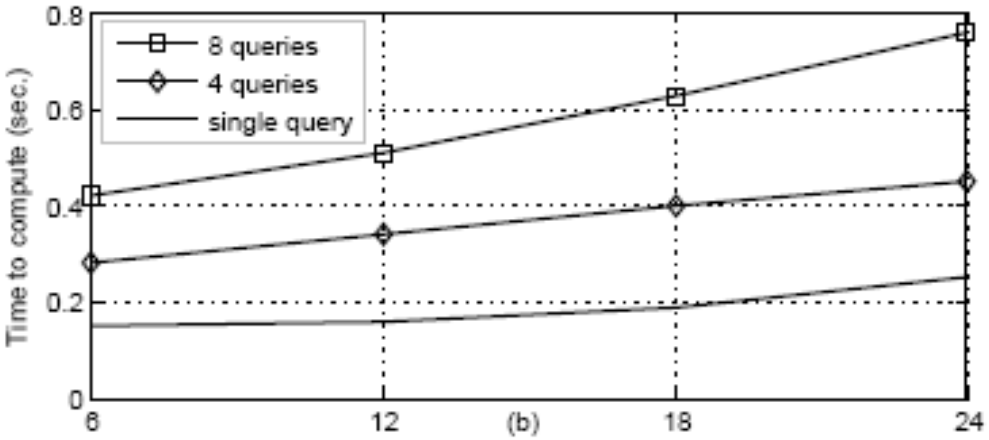
PAP: Policy Administration (PA) Point
PDP: Policy Decision Point
PEP: Policy Enforcement Point
*(L)PIP: (Local)Policy Information Point**
OH: Obligation Handling

**LPIPs are optional*



WP5 – Policy Based Management: Performance Issues

- Performance issue during policy execution (as PDP)
  - Computation time during policy execution
- The processing time depends on:
  - Policy specification language (high level or low level)
  - Type of policy execution engine
  - Underlying formalisms used to describe attributes (e.g. Knowledgebase in pSHIELD)
  - No. of simulteneous execution



No. of instances/class in Knowledge Base

- Figure shows computation time for simultaneous policy execution
- Rule-based semantic policies (SWRL-SQWRL)
  - *A high level language*
- Underlying formalism: OWL
  - *High level comapare to simple XML*
- Simulteneous queries from Application requires execution of simulteneous policies
- Performance measure in P42.0 GHz,1GB RAM windows machine

Implication:

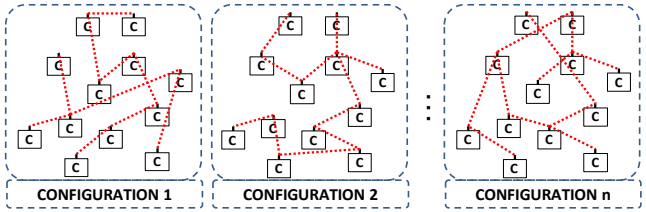
-Latency is one of the QoS requirements for Web services; latency includes request processing time

-Figure shows only with small no. of instances simulteneous policy execution takes increasing amount of time

-Run-time decision support with simultaneous query processing may not be possible with such settings

WP5 –Overlay and Control Algorithms

Select among (already CC-validate and feasible) configurations

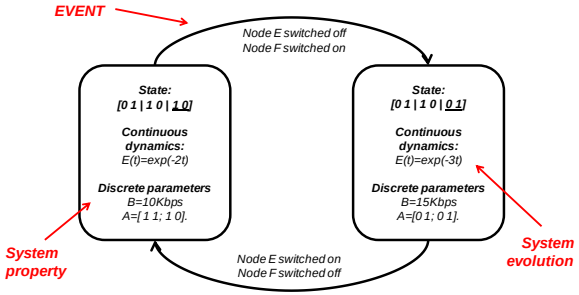


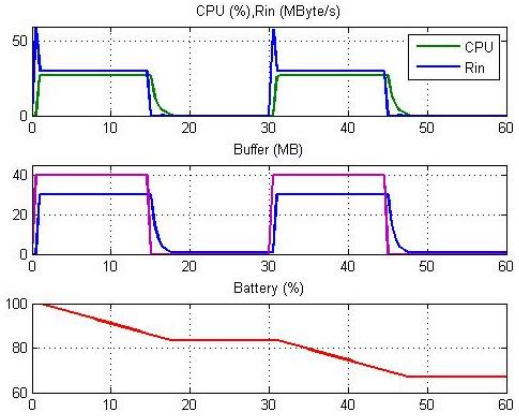
Metrics composition

CONFIGURATION	SPD Level
CONFIGURATION 1	2
CONFIGURATION 2	3
CONFIGURATION 3	2
CONFIGURATION 4	1
CONFIGURATION 5	**4**
CONFIGURATION …	…

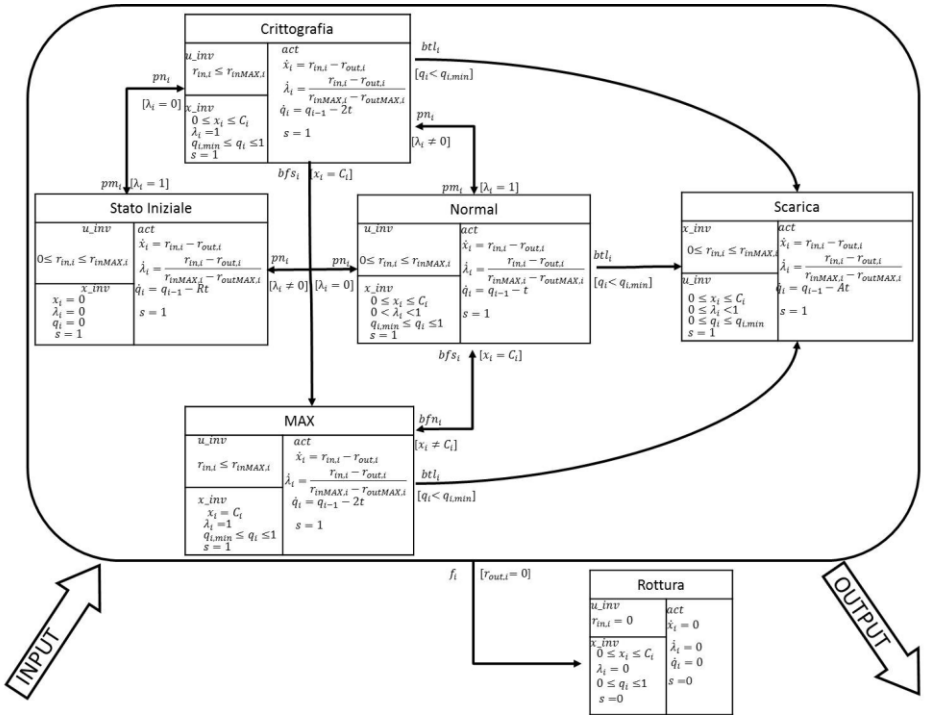
Configudation that matches the desired SPD Level

Hybrid Automata approach -> From configurations -> to Operating Conditions





Control algorithms with context information



WP5 – Conclusions

- Good starting point:
  - Semantic composability enabled
  - OSGI Framework performing SPD composability
  - Overlay technologies harmonization to drive composability in a context-aware way

- Lesson learned:
  - Semantic composability needs to be reinforced
  - Safety Critical (H/S real-time) modelling for validation
  - Standardization: how to bring in the industry?

- Challenging work still to be done with and enriched consortium in the nSHIELD project...

...but for the moment, thanks to the valuable contributors of WP5!