



ARTEMIS JOINT UNDERTAKING
The public private partnership for R&D in the field of Artemis

nSHIELD System Architecture



Budapest, 11-12 September 2012

Nikolaos Pringouris (HAI)



ARTEMIS Call 2010 – ASP6, No: 269317



nSHIELD architecture

Session Overview & Objectives



The nSHIELD architecture session will be split in two parts

A **presentation** that will briefly highlight:

- General status of System architecture deliverables (D2.3, D2.4)
- Overview of System Architecture
- Missing elements to be addressed until D2.4 finalization

A **workshop** with the aim to discuss and agree on the following:

- Discussion on overall architecture and especially on views for each functional layer (in order to address missing material)
- Discussion and definition of interfaces (at various levels)
- Adaptation/Mapping of architecture to application scenarios (at least one)
- Discuss any other unforeseen issue??

System Architecture Deliverables Overview & Needed Actions



D2.3 – Preliminary System Architecture

- deliverable finalized (with delay) on August 3.
- Focus was given:
 - Definition of design methodology approach for each nSHIELD functional layer (based on 4 views)
 - Description of overall architecture and basic elements of the nSHIELD network of ESDs
 - Logical view addressing services and capabilities of each functional layer
 - Categorization of interfaces

D2.4 – Reference System Architecture

- Builds on the grounds of D2.3 and refines it
- Should incorporate comments-suggestions after D2.3 review
- Should address preliminary-incomplete work, basically including:
 - The 3 remaining views for each nSHIELD layer (partially defined)
 - The interfaces describing data and information flows within the system
 - Mapping/Realization of architecture to (a) particular application scenario(s)
 - A better justification – traceability of architecture choices to System requirements (if possible)

D2.4 ToC and Missing Elements

Contents

An overview of what remains to be done in D2.4 based on defined ToC and methodology chosen:

- Development & deployment views must be fully addressed
- Process view can be left for next iteration

1	Executive Summary.....	7
2	Introduction.....	9
3	Terms and Definitions	10
4	Design methodology	11
4.1	Architecture Design Process	11
4.2	Design Considerations.....	13
4.2.1	Distributed vs. Centralized Approach.....	14
4.2.2	Service oriented architecture.....	14
4.2.3	Middleware considerations.....	15
4.2.4	Evolving from interfaces to contracts	15
4.2.5	Interconnectivity of embedded devices.....	16
4.3	Requirements on Architecture.....	17
5	From pSHIELD to nSHIELD	19
6	nSHIELD System Architecture.....	20
6.1	nSHIELD Overall Architecture.....	20
6.2	Node	24
6.2.1	Logical View and Services Description.....	25
6.2.2	Development & Deployment View.....	29
6.3	Network.....	29
6.3.1	Logical View and Services Description.....	29
6.3.2	Development & Deployment View.....	34
6.4	Middleware.....	36
6.4.1	Logical View and Services Description.....	37
6.4.2	Development and Deployment view.....	39
6.5	Overlay.....	41
6.5.1	Logical View and Services Description.....	41
6.5.2	Development and Deployment view.....	42
7	Interfaces	44
7.1	Internal.....	44
7.2	External.....	45
7.3	Components.....	46
8	Application Scenarios Realization.....	46
9	Conclusions	46
10	References	47

Section 4.3 contains a very preliminary list of requirements. Traceability on requirements of D2.2 cannot be extracted

Need refinement for D2.4

Not addressed in D2.3 – need development in D2.4

D2.4 Important dates

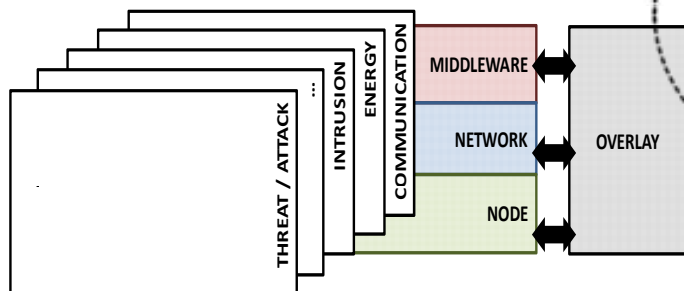


- Current Deliverable completion status: ~ 75%
- Completion date (prior to internal review) → 25th September
(approximately 2 weeks after Budapest meeting)
- Deliverable ready for JU review (after internal review) → 1st October
(that makes 3-4 full days for reviewing and incorporating changes)

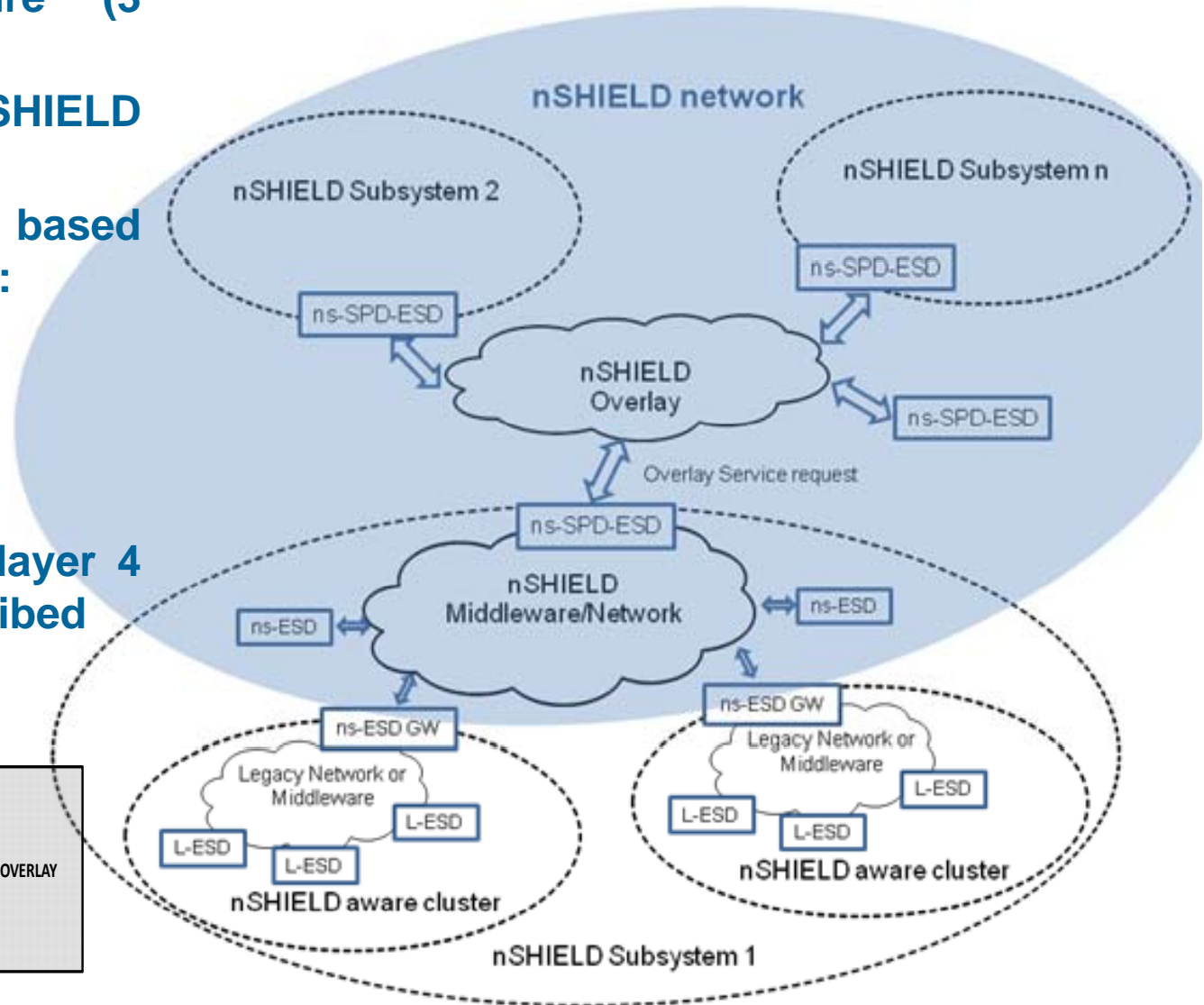
Overview of System Architecture (1/3)



- Hierarchical structure (3 tiers)
- 3 main types of nSHIELD ESDs + the L-ESDs.
- ESDs are described based on 4 functional layers:
 - Node layer
 - Network layer
 - Middleware layer
 - Overlay layer
- For Each functional layer 4 viewpoints are prescribed



nSHIELD



Types of Embedded System Devices (ESDs) in an nSHIELD system network

- **Legacy Embedded System Device** (L-ESD)
 - ✓ Physical Embedded System device characterized by three subsets of Functionalities at Node layer (processors, memory, battery), Network layer (communication, protocol stacks), Middleware (services, functionalities)
 - ✓ Not able to run or deploy any additional functionality (prescribed by nSHIELD middleware or other layers)
- **nSHIELD Embedded System Device** (nS-ESD)
 - ✓ Can be considered as an L-ESD equipped with the minimum SPD capabilities that relate to the 3 first layers of the nSHIELD functional architecture (node, network and middleware)
- **nSHIELD Embedded System Device Gateway** (nS-ESD-GW)
 - ✓ An nS-ESD that provides extra functionality (**technology dependant**) to allow L-ESDs to interact with the nSHIELD middleware (proxy or translator behaviour)
- **nSHIELD SPD Embedded System Device** (nS-SPD-ESD)
 - ✓ An enhanced nS-ESD that provides a full implementation of the core services required by the overlay layer

Types of view used for description of each nSHIELD functional layer

- **Logical view:** The logical view is concerned with the functionality that the layer provides. (In most cases UML class diagram should be used).
- **Development view:** The development view illustrates a system from a programmer's perspective and is concerned with software management. (combination of the UML Component diagram and Package diagram can apply).
- **Process view:** The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the runtime behaviour of the system (UML behaviour diagrams like Activity diagram, Sequence diagram or State diagram)
- **Physical view (deployment view):** The physical view depicts the system from a system engineer's point-of-view. It is concerned with the topology of software components on the physical layer, as well as the physical connections between these components. This view is also known as the deployment view. UML Deployment diagram and/or component or structure diagrams can be utilized).

For **D2.4** at least views with green highlight should be addressed.

Process view is more complicated and may be left until some work in WP3, WP4, WP5 is provided as feedback.

Missing Elements to be addressed prior to review



Essential elements that are partially developed or missing and must be addressed prior to deliverable D2.4 finalization include:

- **Development & Deployment views** for all nSHIELD functional layers (partially defined)
- **Interfaces** describing data and information flows as prescribed in section 7 of D2.4
- A possible **Mapping/Realization** of architecture to (a) particular application scenario(s)
- Linking of architecture to System requirements

Main Problems-Issues:

- **Development & Deployment views definition** → assigned to different teams of partners, responsiveness to requests quite low, consistency-uniformity of descriptions across layers must be improved to have a good quality deliverable
- **Interfaces** → proposed tables are incomplete, some definitions require that internal structure of a functional layer is known (i.e. through a deployment view). Collaboration between teams working in different layers is needed
- **Scenario Mapping/Realization** → some scenarios seem irrelevant to nSHIELD concept, some partners that defined scenarios are not involved in D2.4. Work in this area is very minimal yet
- **Linking System requirements** → status of D2.2 and final list of reqs is unknown.

Workshop Session

General guidelines

- Present Open points regarding overall architecture (HAI):
 - Mainly regarding further analysis of some logical nodes (check support slides later on)
- For each functional layer (node, network, middleware, overlay)
 - Check list of services prescribed in logical view
 - A deployment - development view must exist (including one or multiple diagrams).*
 - Each team leader (see below) should in brief present the status of each functional layer analysis, what remains to be done and risky points:
 - Node layer → ACORDE
 - Network layer → HAI
 - Middleware layer → SE
 - Overlay layer → SE (UNIROMA1)

A discussion should be initiated for each layer to come to an agreement

- * Although Development view is different we consider that in some cases it can possible be assimilated in the deployment one. That is why a single subsection is prescribed for both of them.

- **Type 1: Internal node interfaces:** address data and interactions that occur between the 4 functional layers within a single ESD
- **Type 2: External node interfaces:** address data and interactions that may occur between the 4 functional layers of different ESDs
- **Type 3: Components' or intra-layer interfaces:** include the interfaces that may exist between the various components implementing an nSHIELD functional layer

General guidelines for discussion

Open points regarding interfaces:

- As a 1st step tables 7.1 and 7.2 must be completed and possibly existing data must also be refined. This will address **Internal node interfaces** and **External node interfaces**
- As a 2nd step **Components' or intra-layer interfaces** must be defined and described. A proper deployment view can greatly assist to that.
- It is very difficult to described all types of interfaces with great detail (especially for **Type 3** interfaces). This should be the responsibility of WP3, WP4, WP5. D2.4 should give more a high level description of data that flows.

Adaptation/Realization of Architecture to application scenarios



- Feasibility to do that at this stage should be discussed.

Other Issues

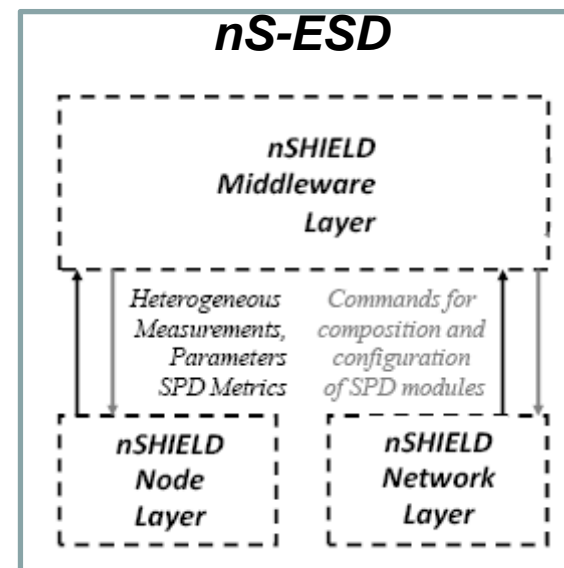
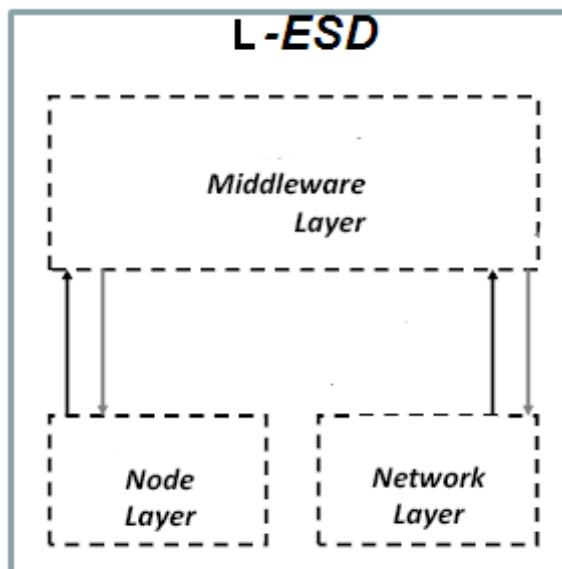


– TBD.

Overall Architecture Support slides



- Compare an L-ESD with a nS-ESD

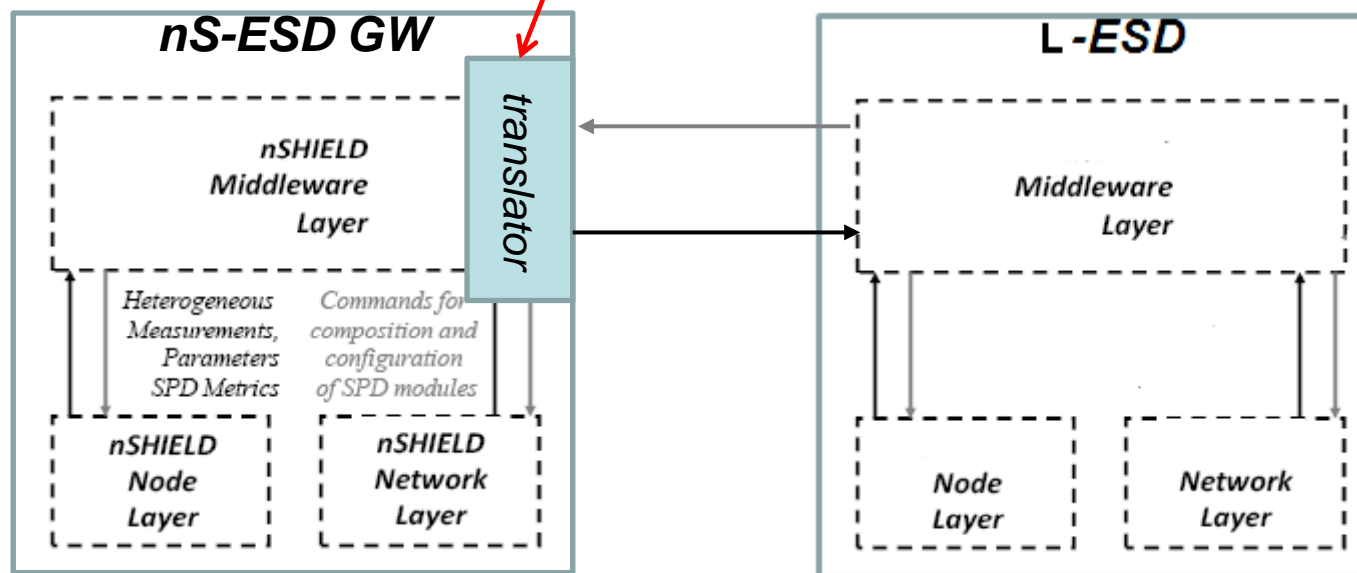


Overall Architecture Support slides



- How do we enable L-ESDs to participate in nSHIELD?
- Currently we have prescribed a nS-ESD-GW but is this the best solution?

Technology dependant component: intercepts service requests and transforms them into a logical format that an L-ESD can understand.

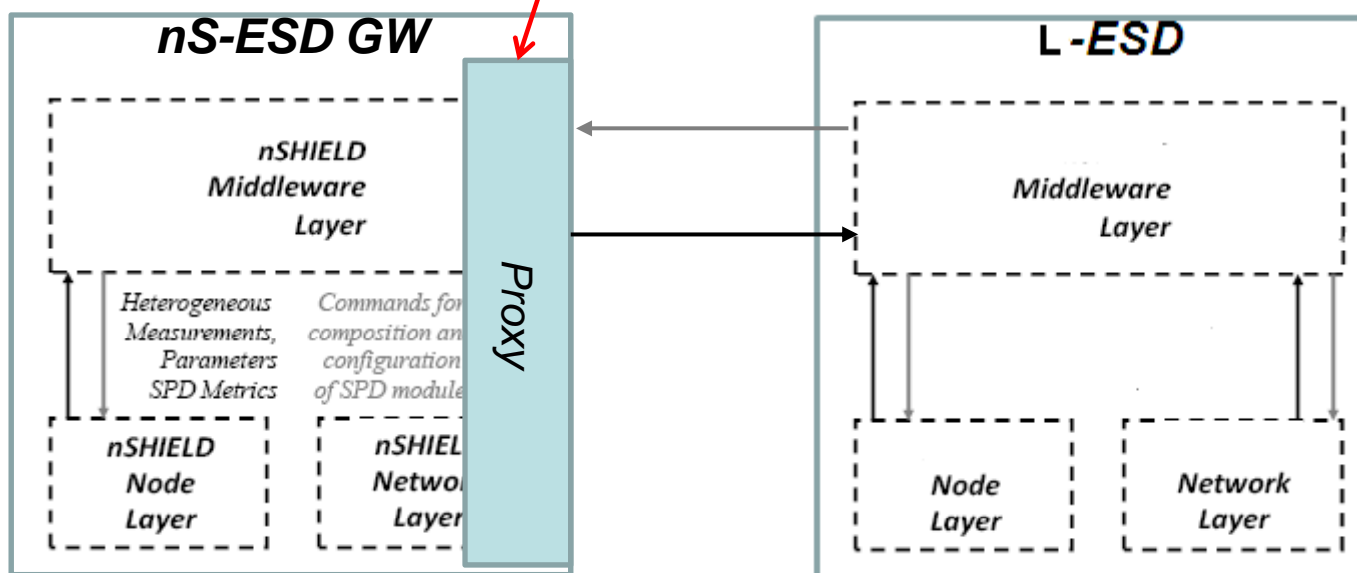


1st case

Overall Architecture Support slides

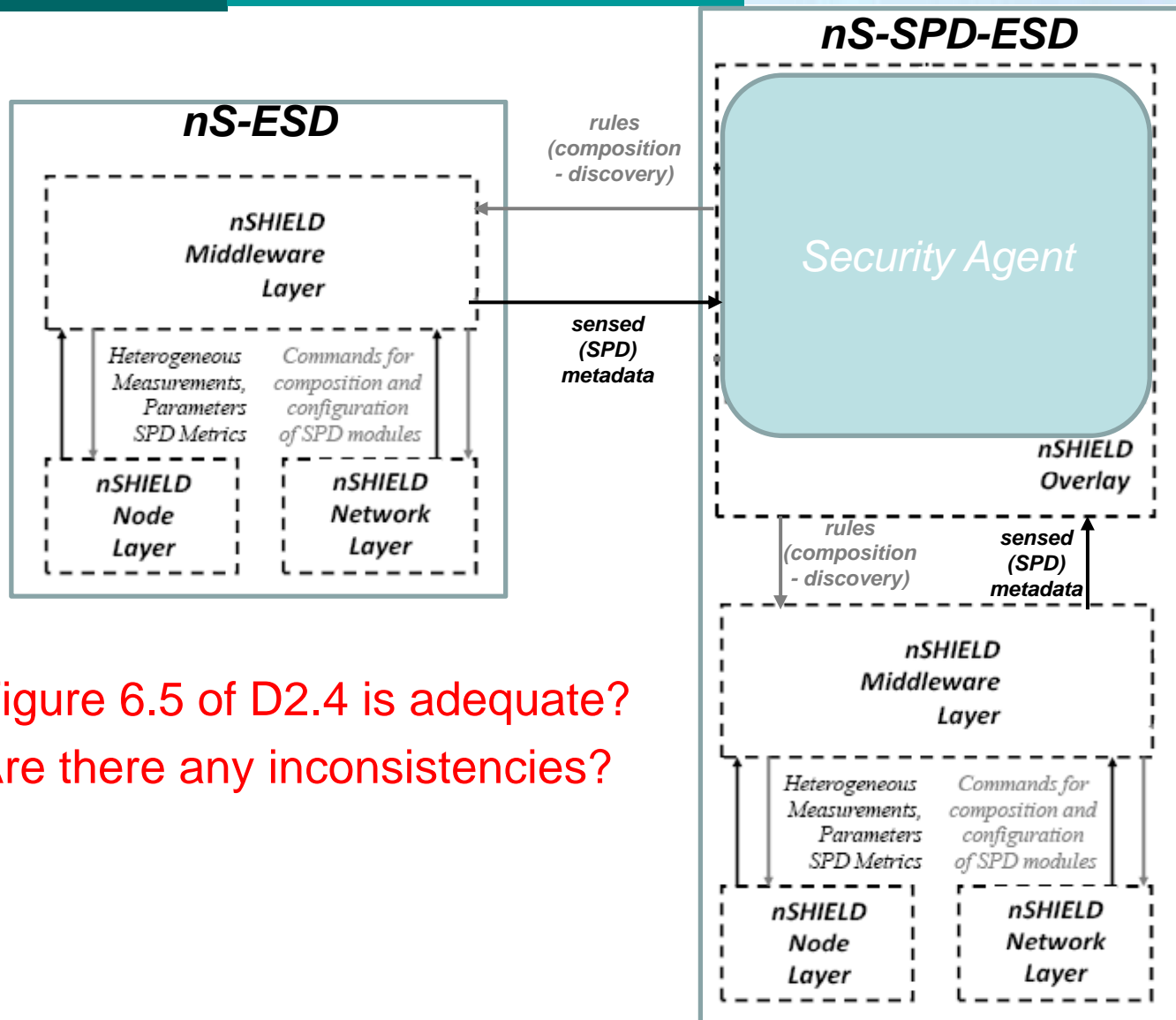


*Technology dependant component:
Adapts service requests into the physical and logical format
that an L-ESD can understand and vice versa.
This mainly provides support for legacy devices with non standard
physical communications*



2nd case

Overall Architecture Support slides



- Figure 6.5 of D2.4 is adequate?
- Are there any inconsistencies?

Figure 6-5: Internal architecture of nSHIELD ESDs with respect to the 4 functional layers