SES MDI using SDC/SDPi+FHIR
Framework & Considerations

Establishing a framework for
Trusted Interoperable Product Decoupling

ADD UML MODEL
UPDATE Gardens
\Diagram
SES MDI Using SDC/SDPi+FHIR – The Big Idea

A Framework for **Trusted Interoperable Product Decoupling**

Addressing the SES MDI Ecosystem “Trust Gap” ...

Full presentation available @ Gemini SDPi+FHIR “The SES MDI Interoperability Trust Gap” @ https://confluence.hl7.org/x/IQ7xB
Questions to Contemplate …

How to minimize duplication of information & effort?

How to maximize use of work and resources already created?

What is the requirements starting point & ending point?

How to truly integrate SES & MDI … seamlessly?!

How to craft a framework for SES “PnT” MDI …

… With an achievable starting point & compelling future state?
“Hanging Gardens” Model

✓ Perfect? Not!
✓ Useful? Absolutely!!!
✓ Provides a common reference point to consider the questions and identify a workable framework
✓ Cuts across ...
  • Standards
  • Organizations
  • Subject Areas
  • Past/Present/Future
  • ....
“Hanging Gardens” Model – *Layer Integration*

Each layer can be characterized by:

- Unique Subject Concepts & Components (Terms)
- Layer-specific Information & Knowledge (UML models)
- Layer API & Capabilities & Requirements (Inter-layer Spec’s)
- Requirements Formalization (Gherkin & ReqIF Spec’s)
- Implementation Trust Logic (SES Assurance Case Spec’s)
- Implementation Technology Logic (MDI Spec’s)

*Integrating each “hanging garden” can be achieved by “threading” …*  
*horizontally* – across layer-specific characterization dimensions

+  
*vertically* – mapping from scenarios to Plug-and-Trust decoupled product interfaces
Layer Characterization: *Horizontal & Vertical*

**Horizontal**
(Intra-Garden Walkways)
- Unique Subject Concepts & Components (Terms)
- Layerspecific Information & Knowledge (UML models)
- Layer API & Capabilities & Requirements (Inter-layer Spec's)
- Requirements Formalization (Gherkin & ReqIF Spec's)
- Implementation Trust Logic (SES Assurance Case Spec's)
- Implementation Technology Logic (MDI Spec's)

**Vertical**
(Inter-Garden Stairways)
- User Narratives / Use Cases / Requirements
- Reference Architectures / Frameworks
- Device / Health Software Specializations
- Key Interoperability Properties & Controls (PRACTical SES)
- SDPi+FHIR Profiles / IGs
- IEEE 11073-1072x-10799 ModSpecs
- Key Interoperability Properties & Controls (PRACTical SES)
- IEEE 11073-1070x PKP (Safety)
- 20701 – Architecture and Protocol
- 10207 – DiM and Service Model
- 20702 - MDPWS
- Devices on FHIR
- HL7 FHIR
- Physical Layers (Ethernet, Wi-Fi, BT, etc.)

Each Layer Characterized Horizontally & Integrated Vertically
Garden Design: *Horizontal Integration*

**Unique Subject Concepts & Components (Semantics/Terms)**
- Concepts, terms, definitions, algorithms, frameworks, ...

**Layer-specific Information & Knowledge (UML models)**
- Use Case / Activity Diagrams, Sequence Models, Object Models, etc.

**Layer API & Capabilities & Requirements (Inter-layer Spec's)**
- “API” Specifications, Implementation “Platform” Requirements, etc.

**Requirements Formalization (Gherkin & ReqIF Spec's)**
- Feature, Rule, Given/When/Then + Specification, SpecRelation (Source, Target), etc.

**Implementation Trust Logic (SES Assurance Case Spec's)**
- Profiled application of 81001-1, 80001-1, 62304, assurance cases, etc.

**Implementation Technology Logic (MDI Spec's)**
- System Actors, Transactions, Messages, Events, Terminology, Value Sets, etc.

INCLUDE MAPPING COMPONENT / ARTIFACTS + standards in scope

Create a UML model showing relationships
Garden Design: **Vertical Integration**

- **Unique Subject Concepts & Components (Semantics/Terms)**
  
  Concepts, terms, definitions, algorithms, frameworks, ...

- **Layer-specific Information & Knowledge (UML models)**
  
  Use Case / Activity Diagrams, Sequence Models, Object Models, etc.

- **Layer API & Capabilities & Requirements (Inter-layer Spec's)**
  
  “API” Specifications, Implementation “Platform” Requirements, etc.

- **Requirements Formalization (Gherkin & ReqIF Spec’s)**
  
  Feature, Rule, Given/When/Then + Specification, SpecRelation (Source, Target), etc.

- **Implementation Trust Logic (SES Assurance Case Spec’s)**
  
  Profiled application of 81001-1, 80001-1, 62304, assurance cases, etc.

- **Implementation Technology Logic (MDI Spec’s)**
  
  System Actors, Transactions, Messages, Events, Terminology, Value Sets, etc.

Use Case(s) to SOA to DS to KIP(SES) to SDPi to SDC to SFC
Layer Characterization: *Horizontal & Vertical Model*

**Horizontal** (Intra-Garden Walkways)
- Unique Subject Concepts & Components (Terms)
- Layer-specific Information & Knowledge (UML models)
- Layer API & Capabilities & Requirements (Inter-layer Spec’s)
- Requirements Formalization (Gherkin & ReqIF Spec’s)
- Implementation Trust Logic (SES Assurance Case Spec’s)
- Implementation Technology Logic (MDI Spec’s)

**Vertical** (Inter-Garden Stairways)
- User Narratives / Use Cases / Requirements
- Reference Architectures / Frameworks
- Device / Health Software Specializations
- Key Interoperability Properties & Controls (PRACTical SES)
- SDPi+FHIR Profiles / IGs
- IEEE 11073-1072x-10799 ModSpecs
- IEEE 11073-1070x PKP (Safety)
- ISO/IEC 11073-10101x, Nomenclature & 11073-10201x DIM
- ISO/IEC 11073-10101x Architecture and Protocol
- ISO/IEC 11073-10702x MDPWS
- 10207 – DiM and Service Model
- 20701 – HTTP/2, REST, IoT, DDS, ...

**CREATE A UML MODEL SHOWING ALL THE BITS THAT GET INTEGRATED HORIZONTALLY & VERTICALLY**

**Each Layer Characterized Horizontally & Integrated Vertically**

**Physical Layers (Ethernet, Wi-Fi, BT, etc.)**
Garden: **Use Case Scenarios**

- **Unique Subject Concepts & Components (Terms)**
  - Narrative Stories, Actors (People & Systems), Scenarios, Activities, ...

- **Layer-specific Information & Knowledge (UML models)**
  - Use Case / Activity Diagrams, Sequence Models, Object Models, etc.

- **Layer API & Capabilities & Requirements (Inter-layer Spec's)**
  - "API" Specifications, Implementation "Platform" Requirements, etc.

- **Requirements Formalization (Gherkin & ReqIF Spec's)**
  - Feature, Rule, Given/When/Then + Specification, SpecRelation (Source, Target), etc.

- **Implementation Trust Logic (SES Assurance Case Spec’s)**
  - , etc.

- **Implementation Technology Logic (MDI Spec's)**
  - Actors, Transactions, Messages, Events, Terminology, Value Sets, etc.

Update layout to focus on Use Case Stuff – Minimize horizontal / vertical
Add layer Scope / Garden Theme
Garden: **Reference Architectures**

- Architecture layer
- Include: MDIRA/ICE and SDC/SOA examples
Garden: **Key Interoperability Purposes (SES)**

- SES KIP
- Includes:
  - SES profiles @ MDI standards
  - Assurance Case Templates
  - Interoperability model(s)
  - 4 Key Purpose Areas for MDI
  - Profiles for applying SES standards to SDC/SDPi+FHIR (incl. 81001-1, 80001-1, 82304, etc.)
  - INCLUDE 81001-1 / 80001-1 REQUIREMENTs that are aligned with LIFECYCLE PHASES including RIGHT SIDE STAKEHOLDERS and filling in “open” requirements => provisioned to device (via PKI?) and including arch requirements etc. [baseline PnT + “value added” for enhanced SES] *** Usability of Device Coupling Strategies, PnT, …

- SFC spec includes SES requirements that COULD / MUST be completed in the Deployment RM ... + how formalized in the PKI/EKU provisioned at implementation?
Garden: **IHE SDPi**

- IHE SDPi
- IHE TF Constructs:
  - ✓ Use Cases (actors, transactions)
Garden:  *ISO/IEEE 11073 SDC*

- SDC – ModSpecs, PKP, BICEPS, SOMDA, MDPWS
- <? 5 slides or one or ... ???>
- <include mapping to TF volumes>
- Integrated Rxxxx
- BICEPS model
- SOMDA Model
- WS-*
Hanging Gardens: *Pulling it all together ...*

- SES MDI (TF-1 Appendix A)
  - SFC – Specification / formalization / organization / ...
  - Assurance Cases – composable / computable / V&V’able / CA’able
  - SES Requirements – per 81001-1, 80001-1, 60601-x-y, 62304, etc.
  - KIP(Regulatory) – sans implementation tech (see earlier slide)
  - CA @ SFC ...

- ACID TEST:
  - ✓ Requirement Test: Verify that PKP OID is in a device’s security certificate
  - ✓ Challenge: Do we know WHY it has to be there? Can we trace that requirement to the initial layer (Scenario?) that mandated its inclusion?
Additional Information
OMG Requirements Interchange Format (ReqIF) “Mapping” Background
OMG ReqIF: Base Model

Figure 10.3 - Specification (Specification), requirement (SpecObject), requirement relation (SpecRelation), relation group (RelationGroup) and associated attributes (AttributeDefinition, AttributeValue)
ReqlF: Requirement Hierarchies & Relationships

Two requirements may have a relation to each other, for example to establish traceability between a Customer Requirements Specification and a System Requirements Specification. Having a relation is represented by an association of one SpecRelation element to two SpecObject elements, one being the source, one the target of the relation.

The two specifications that are related to each other (in the above example: a Customer Requirements Specification and a System Requirements Specification) are referred to by the sourceSpecification and targetSpecification association of a RelationGroup instance.

The hierarchical structure of a requirement specification is represented by SpecHierarchy elements.

"mapping" between Source & Target Requirements