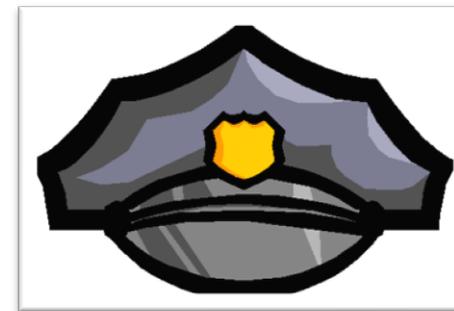

Developing Modular Specifications for Transport Standards: Update and Discussion

How do we achieve interoperable healthcare information systems?

- **Enable stakeholders** to come up with simple, shared solutions to common information exchange challenges
- **Curate a portfolio** of standards, services, and policies that accelerate information exchange
- **Enforce compliance** with validated information exchange standards, services and policies to assure interoperability between validated systems

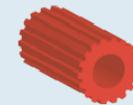


What is the Nationwide Health Information Network (NwHIN)?

- A set of *policies, standards and services* that enable the Internet to be used for secure and meaningful exchange of health information to improve health and health care.
- Enables a *variety of health information exchange scenarios* – from less complex to very robust

Definitions:

- **NwHIN Specifications** define the ways in which information is exchanged
- **NwHIN Exchange** is a community of organizations that use the specifications and the software under a legal and policy agreement
- **CONNECT Software** conforms to the specifications to enable interoperability



Services



Standards



Policies



Trust Fabric

Feedback from States HIE Program:

- Need to talk about what NwHIN is not – i.e., it's not software.
- Relationship to CONNECT (currently States think NwHIN and CONNECT are interchangeable terms)
- Emphasize we are talking about the specs and standards, not about specific software implementations like CONNECT

Direct Specifications

Applicability Statement for Secure Health Transport

XDR and XDM for Direct Messaging

Exchange Specifications

Service Specs

Patient Discovery, Query for Documents, Retrieve Documents

Administrative Distribution

Document Submission

Foundational Specs

Authorization Framework

Messaging Platform

Mod Spec Process

SME Input

Public Feedback

Development Sprints

Internal Feedback

Artifacts Produced:

- Specification
- Test Implementation
- Product Neutral Test Cases

Modular Specification: Secure Transport

- Direct Based Secure Transport
 - SMTP and S/MIME
 - XDR and XDM Conversions

- Exchange Based Secure Transport
 - SOAP over HTTP

- SOAP Based Secure Transport (Completed)
 - Specifications used: Exchange Authorization Framework and Messaging Platform – transport and security infrastructure
- Direct Transport Specifications (In Progress)
 - Specifications used: Direct Applicability Statement for Secure Health Transport and XDR and XDM for Direct Messaging Specifications
 - Expected completion by Dec 15, 2011
- In the future, additional modules will be included based on HITSC/NwHIN Power team criteria

- **Developed a Requirements Traceability Matrix (RTM) in Excel**
 - Reformatted conversational text of the source production specifications into singular requirement statements
 - Non-requirement text (examples, implementation guidance, etc) were moved to appendices
 - Included optionality for each requirement
 - Provided traceability to underlying specifications for each requirement statement (HL7, OASIS, etc.)
 - Provided traceability to associated test implementations and Test artifacts for each requirement

3.2.2.6 Assertion Signature

- Req # 141: The <ds:Signature> element SHALL contain a <ds:SignedInfo> element. I [R] R [O] [Underlying Specs](#) [Example And Guidance](#) [Arch. / Test Document Ref](#)
- Req # 142: The <ds:Signature> element SHALL contain a <ds:SignatureValue> element. I [R] R [O] [Underlying Specs](#) [Example And Guidance](#) [Arch. / Test Document Ref](#)
- Req # 143: The <ds:Signature> element SHALL contain a <ds:KeyInfo> element. I [R] R [O] [Underlying Specs](#) [Example And Guidance](#) [Arch. / Test Document Ref](#)

Internal hyperlinks ease navigation within document

3.2.2.6.1 SignedInfo

- Req # 144: The <ds:SionedInfo> element SHALL speciv the <ds:CanonicalizationMethod>. the <ds:SignatureMethod>. and a <ds:Reference>. I [R] R [O] [Underliva Soecs](#) |

Requirements reformatted to be singular, testable statements

141	OASIS Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0	5 SAML and XML Signature Syntax and Processing	W3C XML Signature Syntax and Processing (Second Edition)	4.3 The SignedInfo Element
142	OASIS Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0	5 SAML and XML Signature Syntax and Processing	W3C XML Signature Syntax and Processing (Second Edition)	4.2 The SignatureValue Element
143	OASIS Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0	5 SAML and XML Signature Syntax and Processing	W3C XML Signature Syntax and Processing (Second Edition)	4.4 The KeyInfo Element
144	OASIS Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0	5 SAML and XML Signature Syntax and Processing	W3C XML Signature Syntax and Processing (Second Edition)	4.3 The SignedInfo Element
147	OASIS Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0	5 SAML and XML Signature Syntax and Processing	W3C XML Signature Syntax and Processing (Second Edition)	4.3.3.1 The URI Attribute
148	OASIS Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0	5.4.2 References	W3C XML Signature Syntax and Processing (Second Edition)	4.3.3 The Reference Element
149	OASIS Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0	5.4.4 Transforms	W3C XML Signature Syntax and Processing (Second Edition)	6.6 Transform Algorithms
150	OASIS Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0	5 SAML and XML Signature Syntax and Processing	W3C XML Signature Syntax and Processing (Second Edition)	4.3.3.5 The DigestMethod Element

Links to underlying specifications provided where appropriate

Developed Clear and verifiable Conformance Criteria

- Develop vendor neutral test cases to ensure conformance of implementations to specifications
- Test implementation – that conforms to the specification and can be used for validation testing (an outside source for sending and receiving of messages)

Test Package

A test case typically guides the transfer of a type of message between the system under test and a control.

Each test case clearly traces back to both the RTM and the underlying spec, as well as to test data if applicable.

Test Focus	Initiator/Receiver Test Case ID	System Implementation Type	Testing Tool Implementation Type	Purpose/Description	Test Steps
Cert paths and trust	R-2-R	Any	Any	Testing Tool sends a message to the System, using an organization-bound cert which is mutually trusted	Preconditions: Both the Testing Tool and the System are bound to the same organization certificate. Test Steps: 1. The Testing Tool sends the message to the System. 2. The System successfully processes the message and returns an MDN to the Testing Tool. 3. The Reviewer verifies conformance of the MDN and verifies that the Audit Log has been created. The reviewer verifies conformance of the Certificate using the Certificate Checklist.

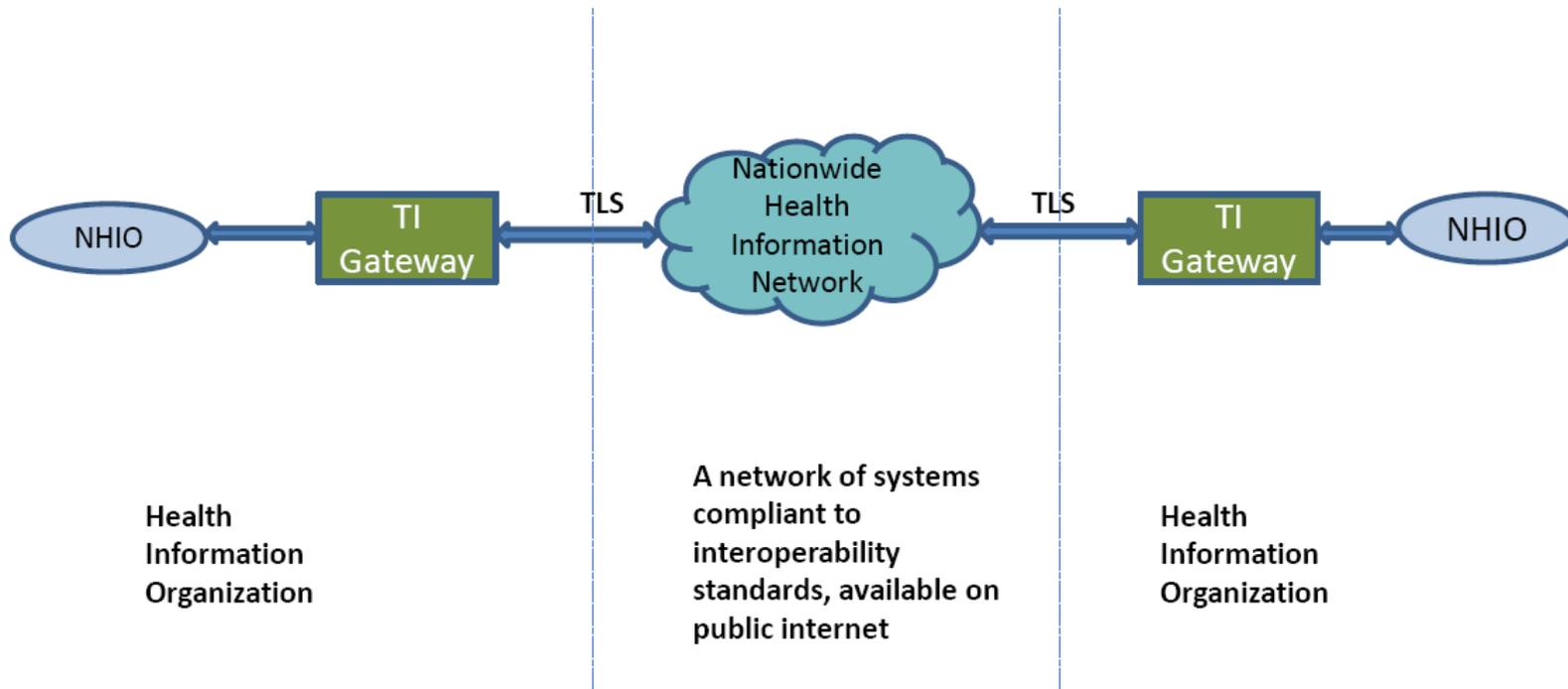
RTM	Underlying Spec	Test Data IDs
63	RFC 5280	201
64		
65		
66		
68		
72		
73		
74		
75		
76		

Checklists are used for each resultant message/log – to check each individual element for conformance. Again, we trace back to both the RTM and the underlying spec.

Message Header:					
mdn-request-header	×	Verify that format is: "Disposition-Notification-To" ":" mailbox *(";" mailbox) - mailbox is the email address of the recipient of the MDN message Example: Disposition-Notification-To: email@domain.com, email2@domain2.com	52	RFC 3798	2.1
Disposition-Notification-Options		Verify that format is: "Disposition-Notification-Options" ":" disposition-notification-parameters = parameter *(";" parameter) WHERE parameter = attribute "=" importance " " value *(";" value)		RFC 3798	2.2

To illustrate the size of the artifact: the phase 2 (DIRECT) test package currently consists of approximately 150 test cases/flows and 7 conformance checklists ranging from 20 to 80 individual checks, along with test data guidance. It is meant to be usable by a wide audience (test tool creators, active pilots, and system builders). We are also pursuing building out automated tools built on the package.

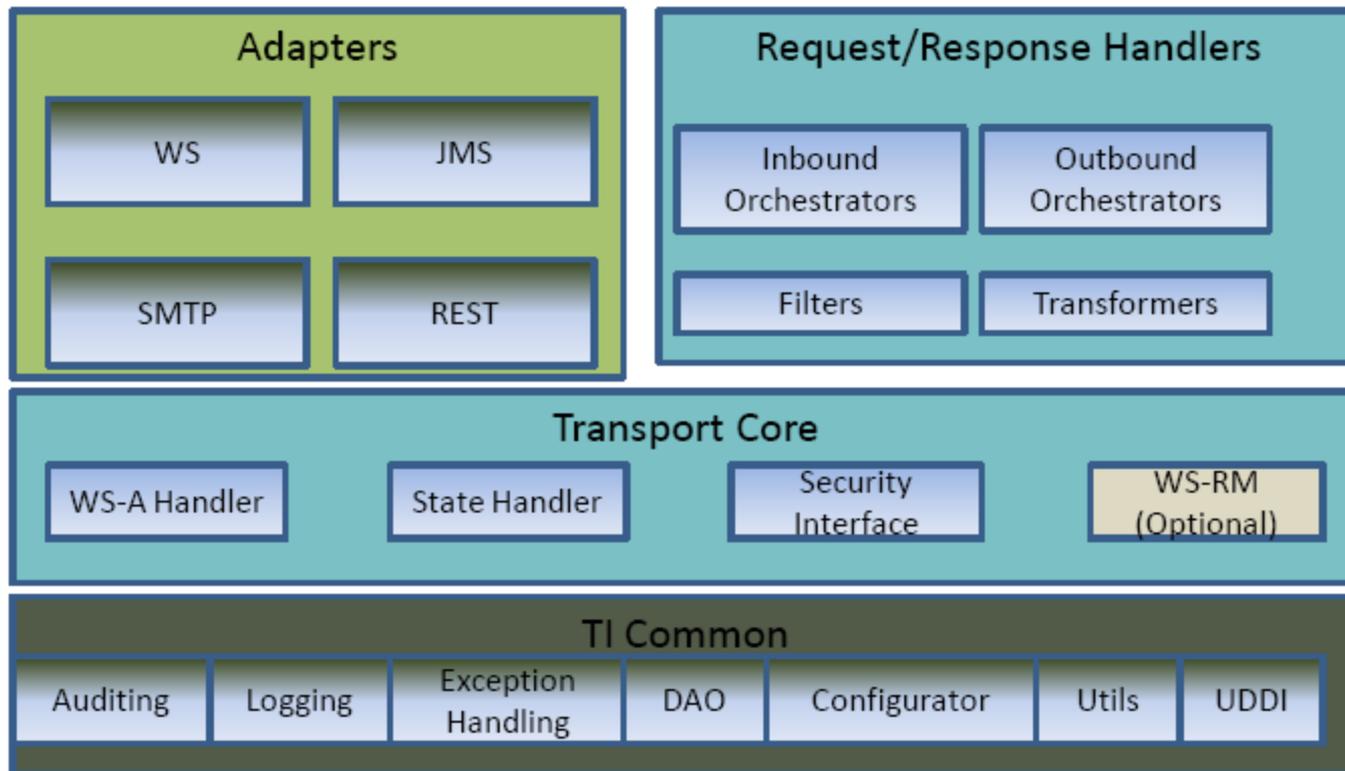
- Contextual Diagram



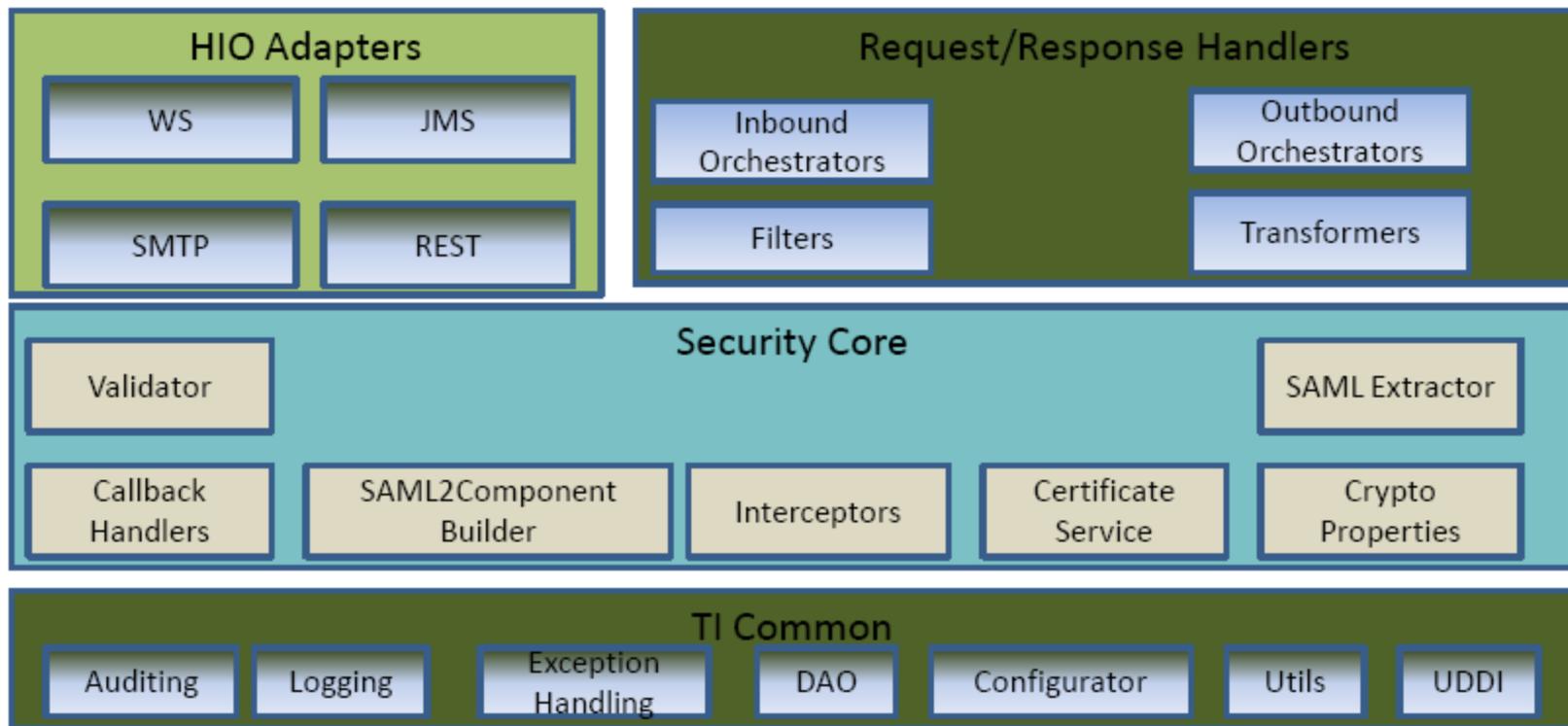
- The deliverables have been available throughout the Project lifecycle at <http://modularspecs.siframework.org/>
- Public calls have been held throughout the process to gather input from the stakeholder community
- There will be a formal review period for 90 days after the conclusion of each phase.

Backup - Details

- Transport Module Architecture Block Diagram



- Security Module Architecture Block Diagram



- Secure Transport Architecture Diagram

