HL7® FHIR® Security

Education Event

Testing HL7 SSRAA IG Implementations

Joe Lamy, AEGIS.Net

HL7 International | ASTP Assistant Secretary for Technology Policy | ONC Office of the National Coordinator for Health Information Technology

Agenda

- Introduction and Overview
- Creating Test Cases for SSRAA IG
- Implementing Test Scripts for SSRAA IG
- Demos – Testing SSRAA in Touchstone
- Q&A

Who am I?

Joe Lamy

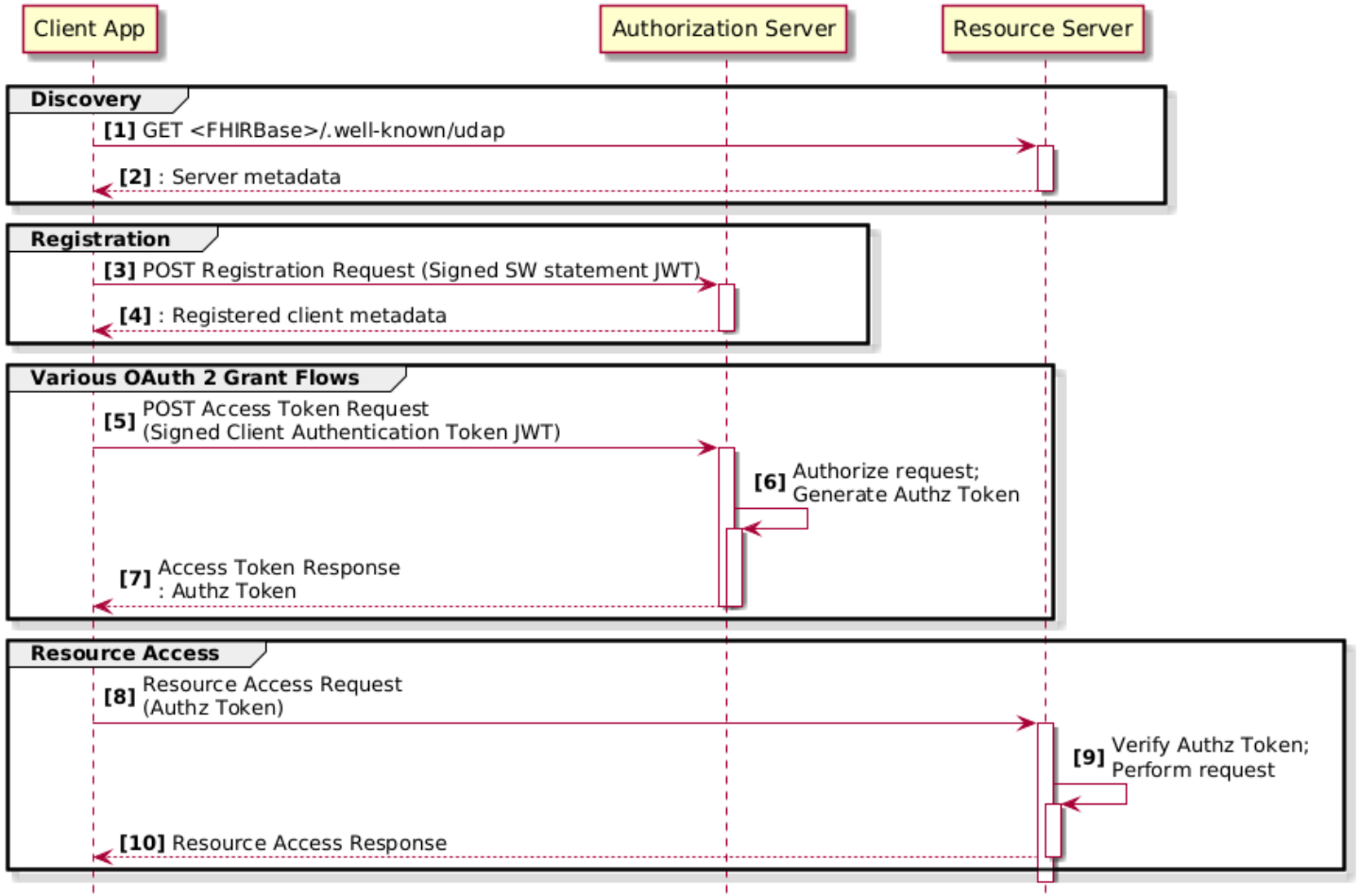
- Healthcare Interoperability Standards and Specifications SME at AEGIS.net
- Test lead for the HL7® Security for Scalable Registration, Authentication, and Authorization IG, aka HL7® SSRAA or HL7® FAST Security

AEGIS.net

- Leader in FHIR testing, with the Touchstone testing platform and test methodology
- Deeply involved in FHIR from the start

HL7® SSRAA overview

- Profile over OAuth 2
- Provides workflow for authenticating and authorizing REST requests
- Plenty to test!



CREATING TEST CASES FOR SSRAA IG

Task: create a testing suite for SSRAA IG

- FHIR and non-FHIR (OAuth 2 and OIDC RESTful JSON) transactions
- Client and server conformance and behavior requirements
  - Example: Client must validate signed metadata
  - Client/server must negotiate authorization through workflow
- Multiple step, multiple actor workflows
  - Servers: FHIR, OAuth Authorization, IdP, IdP OAuth Authorization
  - Clients: RESTful FHIR, RESTful OAuth, user agent, OAuth Authorization server as client to IdP
- Rigorous requirements analysis and test development process
  - Tests target client, server or both as Systems Under Test (SUTs)
  - Other systems needed are considered Test Tools (TTs)
- Test cases at this step are tool-neutral

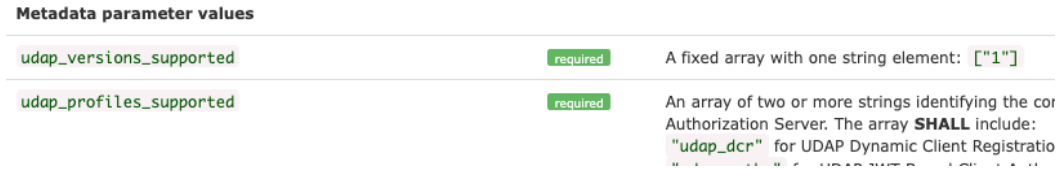
Patterns for getting from requirements to tests

Each actor in a test is a System Under Test (SUT) or a Test Tool (mock, simulator, RI, etc.). We can use patterns to derive tests:

- Client SUT, server SUT: Happy, common paths through a transaction or workflow, conformance and message content checking
  - E.g. Connectathon scenarios
  - You can substitute a TT for a SUT if you only have one system to test
- Client TT, server SUT: Legal variations on the request, invalid requests, requests to drive distinct/conditional server behavior
- Client SUT, server TT: Legal/conditional variations on the request, variant or invalid responses, responses to drive distinct/conditional client behavior

All requirements are not equal

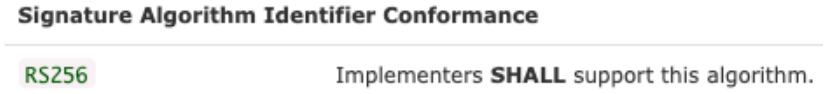
- Many format requirements…



Metadata parameter values

`udap_versions_supported`	required	A fixed array with one string element: `["1"]`
`udap_profiles_supported`	required	An array of two or more strings identifying the cor Authorization Server. The array **SHALL** include: `"udap_dcr"` for UDAP Dynamic Client Registratio

- One complex requirement…



Signature Algorithm Identifier Conformance

`RS256`	Implementers **SHALL** support this algorithm.

- trace to test of one transaction with many checks

Verify UDAP metadata is conformant.

Precondition: Have executed or are executing any test that requests UDAP metadata from a server, and have captured the response.

Steps (performed by test monitor and/or test engine):
1. Verify against each linked conformance requirement.
2. Verify the signature on the signed_metadata JWT.
3. Verify the JWT was signed with a key appropriate for the requested trust community.

- traces to test of full workflow, checking each step

Full workflow - RS256

Preconditions:
1. Tester knows URL for a protected resource request that will fail unless a valid access token is passed.

Steps:
1. Client discovers UDAP metadata from the server. Verify server returns metadata signed using RS256 algorithm.
2. Client registers at the server. Verify Software Statement signed using RS256 algorithm.
3. Client requests an access token from the server. Verify Client Authentication JWT signed using RS256 algorithm.
4. Client accesses protected resources on the server.

Where are we so far?

- Tool-neutral test suite
  - Connectathon scenarios
  - Modular test cases based around features/transactions
- Today: separate artifacts from the IG
  - Word docs, spreadsheets for tracing between requirements and tests
- Future: incorporate testing into the IG or a companion IG
  - Using new FHIR resources TestPlan, Requirements, etc.

IMPLEMENTING TEST SCRIPTS FOR SSRAA IG

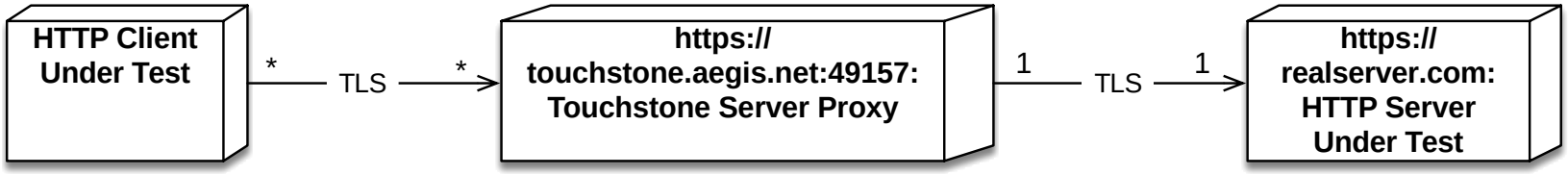
Tools for testing SSRAA IG

- UDAP Test Tool
- Inferno UDAP Security Test Kit
- Touchstone FAST Security
  - Implementation of tool-neutral test cases developed with SSRAA IG
  - Modular test cases based around features/transactions (official, provisional)
  - Connectathon FAST Infrastructure scenarios (official, provisional)

Touchstone SSRAA Test Architecture

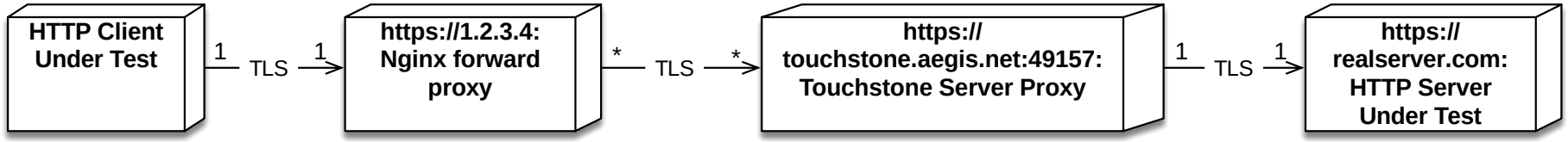
- Test Engine: AEGIS Touchstone
  - Loaded with FHIR TestScripts developed for SSRAA
  - Passive evaluating intermediary (proxy) captures message traffic
  - Test engine adjudicates results
- Client TTs: Touchstone, Postman, CMS Security RI
- Server TTs: CMS Security RI, WireMock

Evaluating intermediary challenges



- Challenge: Proxy endpoints != real endpoints
  - Can point client to proxy for a single request, but any use of returned URLs has to map between real and proxy URLs before making subsequent requests
  - In HL7 UDAP, client and server have explicit requirements (for example JWT “aud” claim) for URLs in messages to correspond to URLs used
- We explored multiple ways of addressing this
  - The problem never fully goes away – there is always something different about some SUT from its production environment
  - But you can move it around and minimize it
  - We found one size didn’t fit all, but **all could find a size**

Evaluating intermediary solutions



- Solution: Transparent forwarding proxy
  - For clients with control over routing (e.g. /etc/hosts) we created dedicated forwarding proxies (cloud-hosted or installed local on client) using nginx
  - Clients could then behave normally, using real server URLs in all messages and HTTP requests
  - This was very useful for capturing all browser traffic
    - We could then decide which requests needed to be part of the test definition or ignored
  - All client requests to the servers they tested with were routed to the single proxy
  - The proxy impersonated each server for TLS, so the client needed to add a special root and intermediate cert to their trust store
  - Some browsers needed to reset HSTS caches

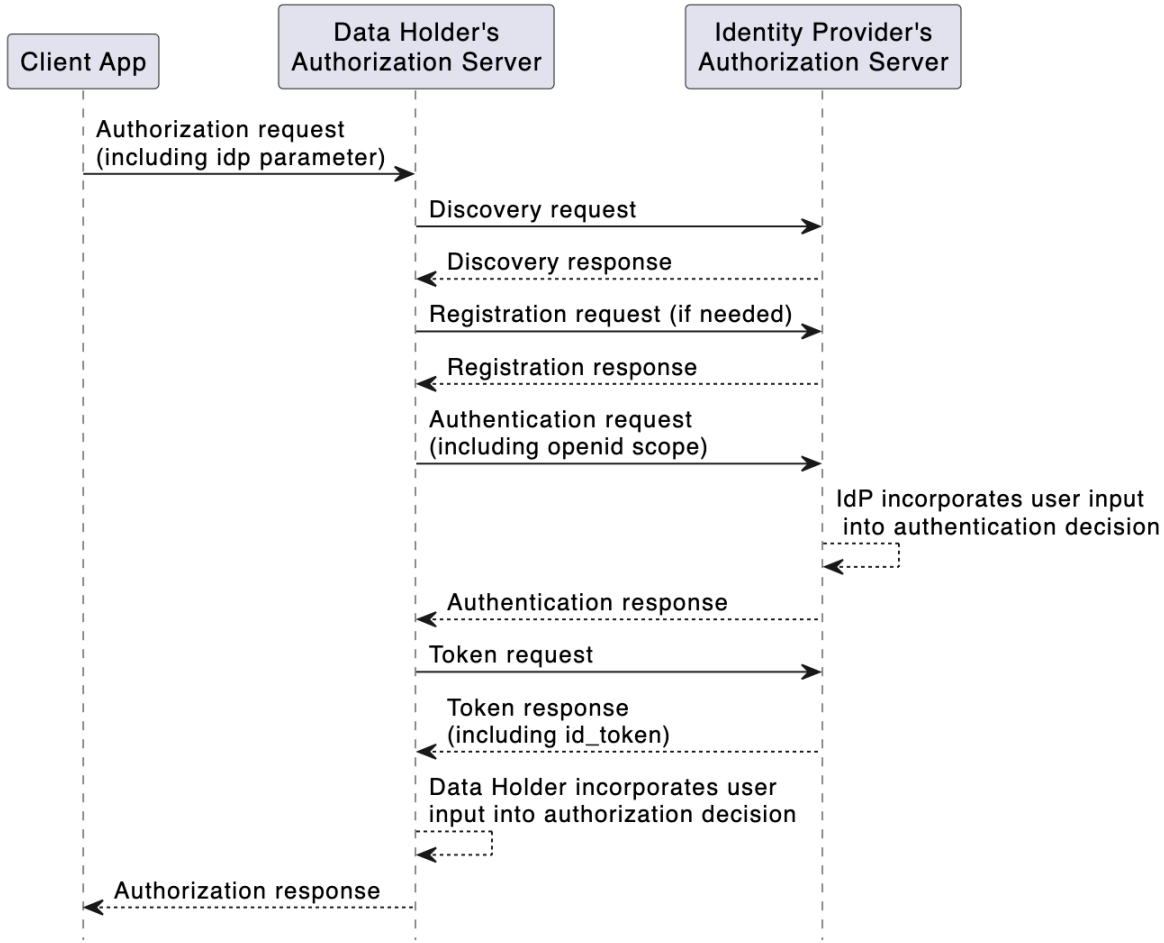
Evaluating intermediary solutions

- Solution: One side or the other performs URL mappings in test mode
  - Create a test mode in code where requests can be intercepted and rerouted
  - Some clients preferred to do this, either because the effort was less, or because their test environment was too locked down for them to control routing
  - They are considering refactoring this into a common SDK for other users of our testing tool

DEMOS

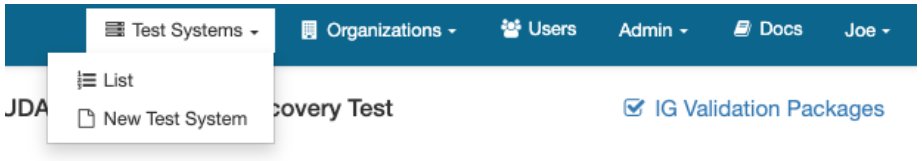
Demo – Testing SSRAA in Touchstone

- TestScript execution covering UDAP Discovery and Registration
- TestScript execution covering the UDAP Tiered OAuth flow (which is itself just one part of the overall UDAP flow), shown to the right



Try it yourself

- Go to Touchstone
- If you don’t already have an account, add one
  - Add yourself to the UDAP Implementers organization (I will approve)
- If you don’t already have a client test system, add one
  - Can put any valid URL in Base URL
  - Check FHIR-Client, uncheck FHIR-Server
  - Enter your IP address (https://whatismyipaddress.com/)
  - Choose “Origin IP of request” for “Match Peer-to-Peer client request to test execution using”
  - Save



Try it yourself (continued)

- Find the UDAP Discovery test script “01-hl7-udap-discovery-happy-path” and create a test setup
  - Select your client for the origin
  - Select CMS - RI-FAST-Identity-Matching-1-0-0-Server for the destination
  - Save and Execute
  - Click on the orange “Waiting for Request” button
- Pop this in your browser: https://touchstone.aegis.net:63220/fhir/.well-known/udap
- In Touchstone, hit the green Refresh button
- Boom! You have tested a FAST Security client and server.

Q&A

More questions or feedback

- joe.lamy@aegis.net
- X: https://x.com/specsherpa
- LinkedIn: https://www.linkedin.com/in/joe-lamy-8756b89