IETF Hackathon

IETF 122
15–16 March 2025
Bangkok, Thailand

WE HAD 2 PROJECTS!

- Each testing different adopted documents of the WG
- Transitioning SCRAPI to HTTP-based asynchronous registration
  – Jon Geater
- YANG modelling of the SCITT architecture
  – Nobuo Aoki

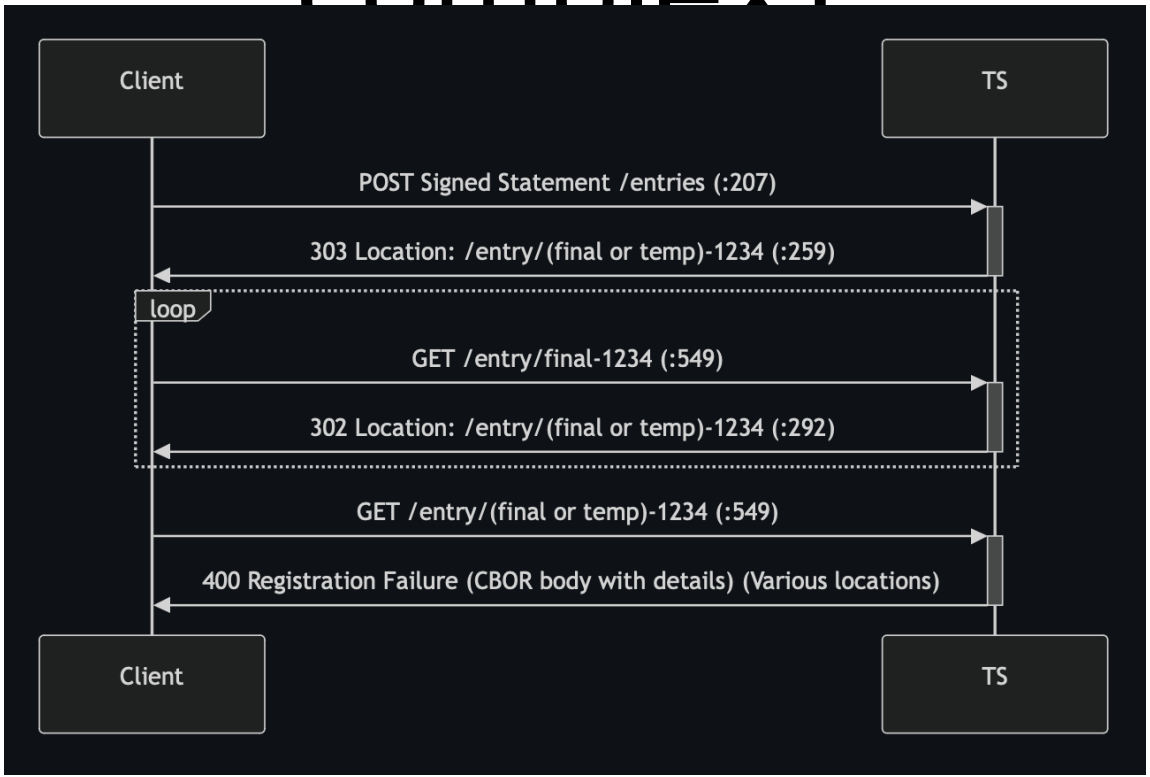
Both continue work from last time and reflect improvements in the specs since IETF 121

Hackathon Plan - SCRAPI

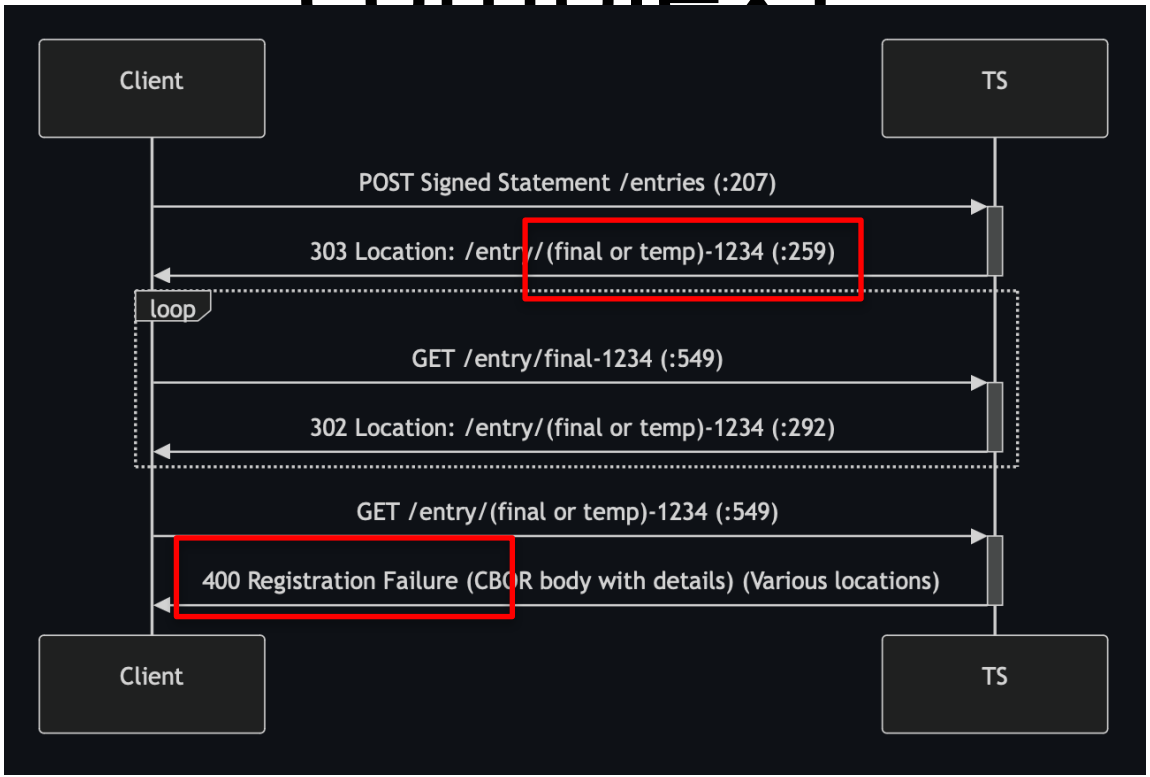
- COSE/CBOR-based SCRAPI was already working well, but there have been changes to remove as many custom messages from the reference API as possible, especially where they included custom CBOR
- Based on this we also have a simple proof-of-concept python wrapper for the reference API which resolves everything to pycose/cbor2 and the RFC9290 objects

- This hack aimed to prove that the new theoretically simpler model is also ergonomic for programmers

Concept – why is this complex?



Concept – why is this complex?



What got done - SCRAPI

- What was achieved
  - Proof of concept transition of the former model to the new model
  - The same test rig passes on both!
- Links:
  - Spec: https://github.com/ietf-wg-scitt
  - Code: https://github.com/scitt-community/py-scrapi
  - Code: https://github.com/JAG-UK/rfc9290
  - Releases: https://pypi.org/project/py-scrapi/
  - Releases: https://pypi.org/project/rfc9290/

What we learned - SCRAPI

- The theory worked in practice!
  - Only a small improvement in protocol complexity, but big improvement in code complexity
  - Better compatibility for retro-fit implementations
  - The REST overload and game-of-cups with locators does make the internal object handling a bit more error-prone, but it’s worth it
  - Probably no new work to feed back, but will consider the opportunity and implications for very long term management of Verifiable Data Structures and whether it reads into SCITT architecture or COSE Receipts

Spin-off SCITT's Hackathon Project

Hackathon Plan - YANG

- Find unresolved issues that will help SCITT to further mature in supply chain security
  - Focus on *RFC 9472*, *draft-ietf-scitt-architecture-11*, and *draft-ietf-scitt-scrapi-04*

- This hackathon plan
  cf.
  https://github.com/aoki-n1/spec-btw-spec-scitt/tree/main/Hackathon-122
  - Finding room for new spec
  - Thoroughly examine the validity of new spec
  - Triage topics that we have come up with
  - Ponder the problem statement, key issues, and solutions

What got done - YANG

- What did we achieve?
  - We identified issues and processed them to be placed on the SCITT discussion
  - YANG modeling of a set of statements that match the SCITT architecture
  - Future SCITT's support for dynamic statement lifecycles
- What did we postpone to the hackathon in IETF Madrid?
  - Further exploration of unsolved problems in Supply Chain Security
  - Ponder a case study that will lead to feedback for SCITT

What we learned - YANG

- Lessons learned from this hackathon
  - Identifying the tasks to be tackled at the Madrid hackathon
  - Report on the problem statements, key issues, and solutions discovered for the SCITT WG
  - Draft writing is a little bit intimidating for newcomers!
- What's next?
  - Breakdown of high-level solutions
  - Comparison of solutions that take implementation and specification descriptions into account
  - Output to some media for reporting to SCITT WG

Wrap Up

Team members:

- Nobuo Aoki
- Jon Geater
- Steve Lasker

<Other links, contacts or notes>