



InterUSS Open Source Test Suite

Automated checkouts and ongoing
compliance verification



- **Ben Pelletier** (chairperson)
 - Architect of InterUSS RID automated testing functionality
 - Coordinator for many demonstrations of InterUSS technology
- **Michael Barroco**
 - Architect of InterUSS SCD automated testing functionality
 - Swiss Federal Office of Civil Aviation UAS Flight Authorisation Technical Working Group lead

About InterUSS Platform and Linux Foundation



- Linux Foundation provides governance for collaborative, open-source projects
- *“The mission of the Project is to develop open source, simple, secure and scalable interoperability between service providers. The Project strives to advance safe and efficient drone operations and satisfy needs for UTM and UAM”*

Open-source test suite: Why?



- Authority must verify applicant compliance with requirements before granting authorization to operate
- Manual checkouts are:
 - Difficult
 - Expensive
 - Unreliable
 - Do not effectively detect future regressions
- Automated tests are:
 - Exceptionally easy and inexpensive, after initial development
 - Therefore can be conducted frequently to detect future regressions

Open-source test suite: What?



Test director instructions for RID testing

Manual checkout/test

- “USSP1, start a simulated flight in the test area”
- “USSP2, do you see USSP1’s simulated flight on your display?”

Automated test

- Inject telemetry into USSP1 (as if aircraft were reporting it)
- Observe system state according to USSP2

Open-source test suite: What?



Test director instructions for flight authorisation testing

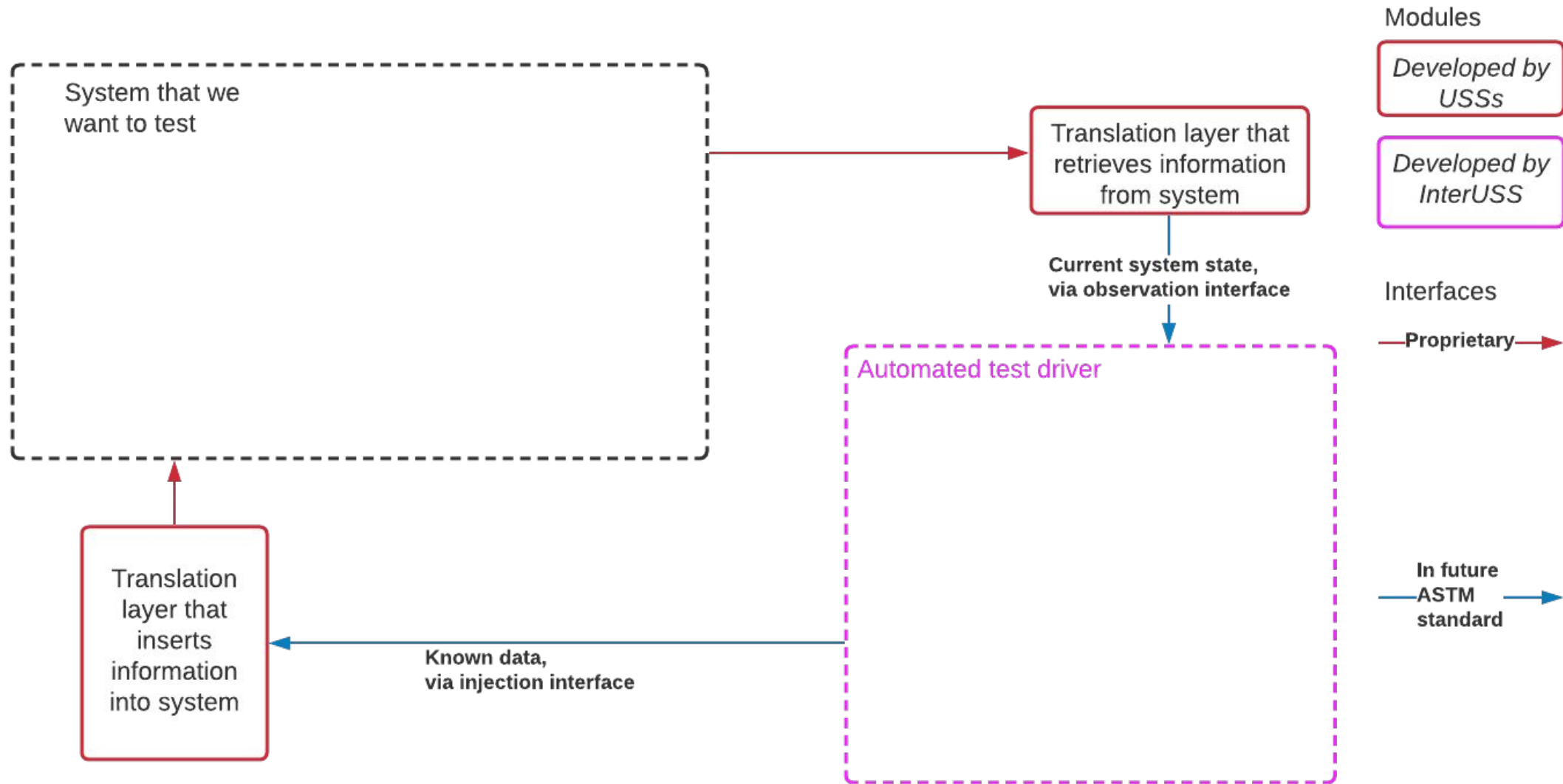
Manual checkout/test

- “USSP1, plan a flight in the test area”
- “USSP2, attempt to plan a flight that conflicts with USSP1’s flight”
- “USSP2, was that planning attempt successful?”

Automated test

- Inject user intent into USSP1 to conduct a flight with this specific telemetry
- Inject a conflicting user intent into USSP2.
- Verify that user intent from USSP2 was not successful.

Open-source test suite: How?





- Scenario based testing
- Scenarios cover requirement sets
- Scenarios/test cases on per-country basis¹
 - Supports regulator/per country review process as needed
- Community can propose updates
 - Building on aviation's history of sharing learnings to improve safety

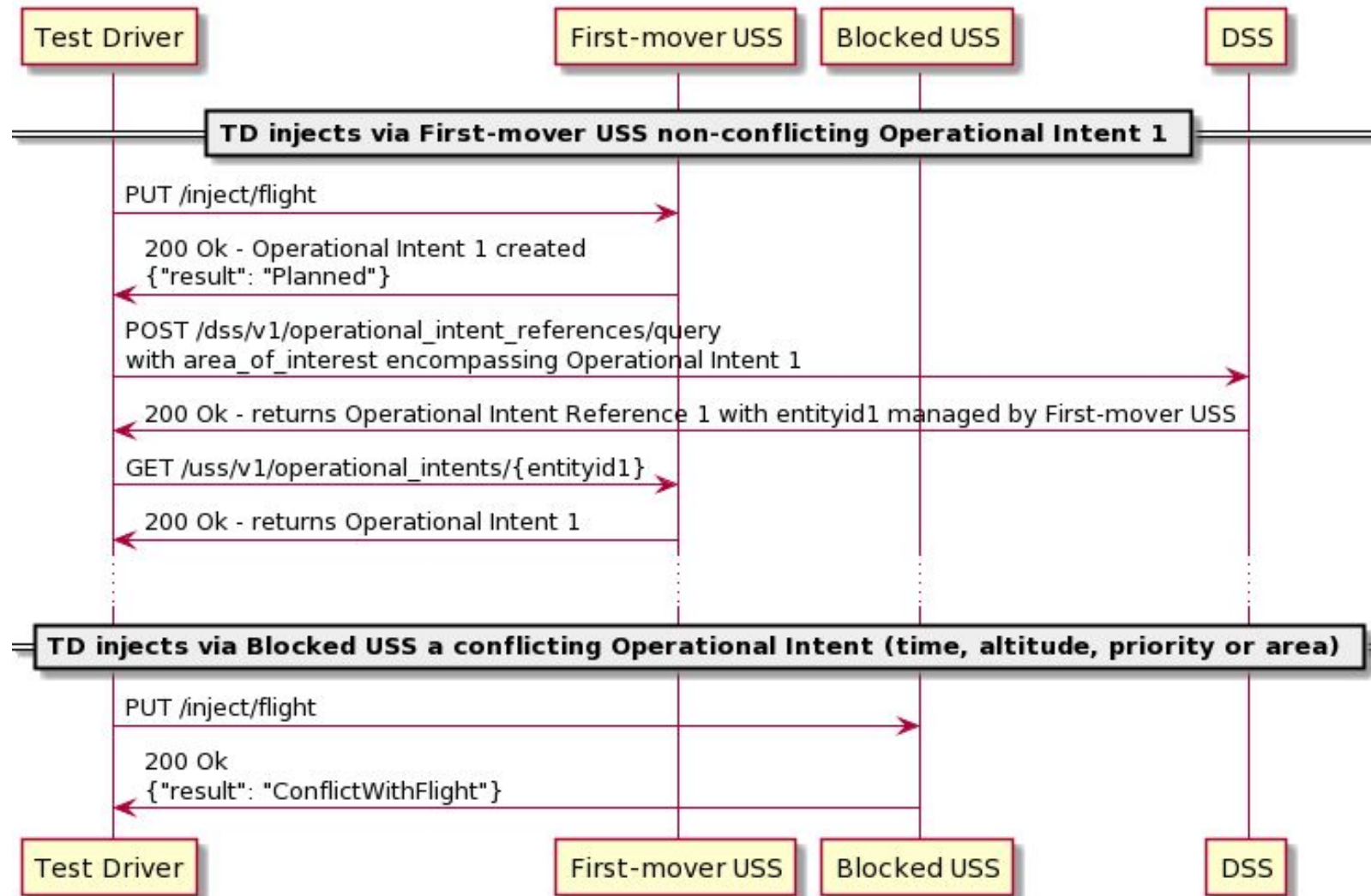
¹e.g.: https://github.com/interuss/dss/tree/master/monitoring/uss_qualifier/scd/test_definitions/CHE

Example test case: ASTM SCD nominal planning



Strategic Coordination Interoperability

Nominal planning test



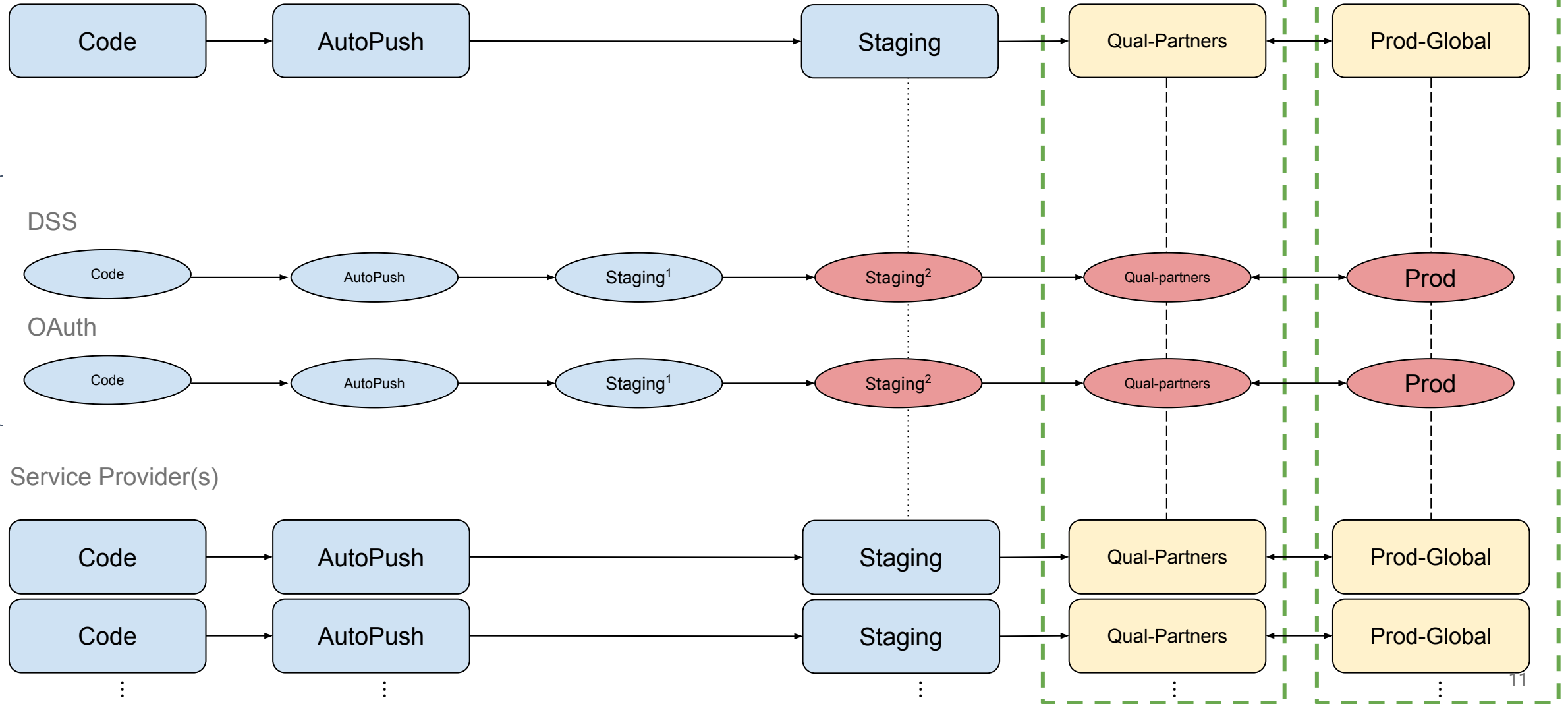


- Output from test suite invocation
- Machine-readable JSON following Report schema
- Includes:
 - Traceable description of system under test
 - Test suite version
 - Test cases version
 - Test targets configuration
 - USSP software versions
 - USSP reported capabilities
 - List of issues detected (if any)
 - Summary and description of issue
 - Which check in which test triggered the issue
 - Which requirement was not met due to the issue
 - Severity of problem
 - Details of interactions relevant to the issue (for USSP debugging)

Deployment process with automated testing



Display Provider



Case studies



- Automated RID testing implemented and demonstrated by Avison
- Automated SCD testing implemented and demonstrated by ANRA, Avison, and Dronetag as observed by Swiss regulator
- Automated DSS testing implemented and used by Wing and Skyguide



Get the details at interuss.org

Roadmap



Expand test cases.



Deeper root cause reporting.



Add geoawareness & traffic information to meet all 4 mandatory services.



Expand accessibility and ease of conducting automated tests.

For more information:
To get involved:

interussplatform.org
github.com/interuss/dss



InterUSS

Platform