Online & remote interoperability testing for IoT

Presented by Federico SISMONDI and César VIHO
Agenda

• Why the F-Interop project?
• Online remote interoperability testing requirements
• F-Interop platform architecture and components
• Current status of F-Interop platform
• Remote interoperability testing in practice with F-Interop
• F-Interop achievements and next steps
F-Interop: the context, the needs from SDOs and SMEs
Interoperability testing

“The purpose of interoperability testing is to prove that end-to-end functionality between (at least) two communicating systems is as required by the standard(s) on which those systems are based.”

State of the Art: Face-to-Face Events

Goals of these events:
- Make better standards
- Reduced time-to-market
- Increase adoption

Similar requirements from all SDOs:
- ETSI
- IETF
- ITU
- oneM2M
- W3C

Face-to-Face interop events:
- 6TiSCH: Jul’15, Feb’16, Jul’16
- oneM2M: Sep’15, May’16, Nov’16
- CoAP: Mar’12, Nov’12, Nov’13, Mar’14

Example: ETSI plugtests
- 6TiSCH: Jul’15, Feb’16, Jul’16
- oneM2M: Sep’15, May’16, Nov’16
- CoAP: Mar’12, Nov’12, Nov’13, Mar’14
<table>
<thead>
<tr>
<th>Time</th>
<th>Friday 15</th>
<th>Saturday 16</th>
<th>Sunday 17</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:30</td>
<td>Room Opening</td>
<td>Room Opening</td>
<td></td>
</tr>
<tr>
<td>09:00</td>
<td>TEST SESSION #3</td>
<td>TEST SESSION #7</td>
<td></td>
</tr>
<tr>
<td>11:00</td>
<td>SET-UP</td>
<td>TEST SESSION #4</td>
<td>TEST SESSION #8</td>
</tr>
<tr>
<td>13:00</td>
<td>LUNCH 12:30 to 13:30</td>
<td>LUNCH</td>
<td>LUNCH</td>
</tr>
<tr>
<td>14:00</td>
<td>TEST SESSION #1</td>
<td>TEST SESSION #6</td>
<td>TEAR-DOWN 14:00 to 15:00</td>
</tr>
<tr>
<td>14:00</td>
<td>TEST SESSION #2</td>
<td>TEST SESSION #4</td>
<td></td>
</tr>
<tr>
<td>18:00</td>
<td>WRAP UP</td>
<td>WRAP UP</td>
<td></td>
</tr>
</tbody>
</table>
In practice, attended by large companies working on standardization 😊

Product development

Standardization

Face-to-Face events

Online/remote events

F-Interop

Online

Remote

Inclusive of SMEs, more standards-based products on the market, faster 😊
IoT standards and technologies evolve fast

<table>
<thead>
<tr>
<th>Layer 4</th>
<th>COMPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems Integration</td>
<td>IFTTT</td>
</tr>
<tr>
<td>WoT-a-Mashup</td>
<td>Node-RED</td>
</tr>
<tr>
<td>Physical Mashups</td>
<td>Automated UI Generation</td>
</tr>
<tr>
<td>Web Applications</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer 3</th>
<th>SHARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social Networks</td>
<td>API Tokens</td>
</tr>
<tr>
<td>Delegated Authentication</td>
<td>TLS</td>
</tr>
<tr>
<td>Social WoT</td>
<td>DTLS</td>
</tr>
<tr>
<td>Encryption</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer 2</th>
<th>FIND</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST Crawler</td>
<td>Web Thing Model</td>
</tr>
<tr>
<td>HATEOAS</td>
<td>Search engines</td>
</tr>
<tr>
<td>Link Header</td>
<td>Schema.org</td>
</tr>
<tr>
<td>Semantic Web</td>
<td>Linked Data</td>
</tr>
<tr>
<td>mDNS</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Layer 1</th>
<th>ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>JSON</td>
</tr>
<tr>
<td>Web Hooks</td>
<td>WebSockets</td>
</tr>
<tr>
<td>REST API</td>
<td>HTTP</td>
</tr>
<tr>
<td>URI / URL</td>
<td>Gateway</td>
</tr>
<tr>
<td>MQTT</td>
<td>CoAP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Networked Things</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFC</td>
</tr>
<tr>
<td>Thread</td>
</tr>
<tr>
<td>QR</td>
</tr>
<tr>
<td>Beacons</td>
</tr>
<tr>
<td>Wi-Fi</td>
</tr>
</tbody>
</table>

Source: Building the Web of Things; book.webofthings.io
Creative Commons Attribution 4.0
Leveraging from European testbeds

32 testbeds, 4755 nodes

- Fed4FIRE
  ([www.fed4fire.eu/testbeds](www.fed4fire.eu/testbeds))
  - 24 testbeds
  - ~1000 nodes
- OneLab
  ([onelab.eu](onelab.eu))
  - Includes 6 IoT-lab deployments
    (including 2728 IoT nodes)
- IoT lab
  ([www.iotlab.eu](www.iotlab.eu))
F-Interop: remote interop testing requirements
F-Interop actors and components

F-Interop-Contributor

Provides resources to the platform

F-Interop-Platform

Testing Tools

Test Descriptions

Test Implementations

IUTs

Testbeds

IoT devices

F-Interop-User

Uses platform testing services

Testbeds

IOT devices

...
Gathering requirements

- **Internet**
  - **IUT 2 - CoAP client**
  - **F-Interop User 1**
  - **IUT 1 - CoAP server**
  - **F-Interop User 2**
  
  a) Testbed interoperability session.

- **b) Remote IUT vs testsbed IUT Interoperability session.**

- **Internet**
  - **IUT 1 - CoAP server**
  - **IUT 2 - CoAP client**
  - **F-Interop User 1**
  - **F-Interop User 2**

  b) Remote interop over the internet.

**ETSI Plugtests tests specification**

**Interoperability tests best practices documents**

**Testing tools used during plugtests**
Methodology for deriving F-Interop-Platform requirements

1. **First stage (Online Conformance Testing)**
   - CoAP/OneM2M interop/conformance testing
   - CoAP plugtests tests specification
   - CoAP testing tools used for plugtests

2. **Second stage (6TiSCH Interop/Conformance Testing)**
   - 6TiSCH interop/conformance testing
   - 6TiSCH plugtests tests specification
   - 6TiSCH testing tools used for plugtests

3. **Third stage (PERF & Scalability Testing)**
   - FI-User and FI-Contributor needs for perf & scalability testing

4. **Fourth stage (Energy Efficiency Testing)**
   - FI-User and FI-Contributor needs for energy efficiency testing

**F-Interop User/Contributor needs by test types**

- FI-User and FI-Contributor needs for CoAP/OneM2M tests
- FI-User and FI-Contributor needs for 6TiSCH remote tests
- FI-User and FI-Contributor needs for perf & scalability testing
- FI-User and FI-Contributor needs for energy efficiency testing

**F-Interop Platform Requirements**

- F-Interop Platform Requirement
- Priority
- References to FI-UN and FI-CN
- Scope
- Description

**Final F-Interop Platform Requirements**
## Platform requirements

<table>
<thead>
<tr>
<th>Req. Id</th>
<th>Field/Step</th>
<th>Requirement Scope</th>
<th>Requirement description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>FI-PR.05</td>
<td>FI-Session.1</td>
<td>Test suites discovery and selection.</td>
<td>FI-Platform MAY provide networking compatibility and reachability tests adapted to the protocol testing solution and the location model chosen. FI-Platform MAY provide tools for overcoming most common compatibility and reachability problems — e.g. port forwarding via ssh tunneling for UDP-based test suites must be provided if FI-User is running his/her IUT behind a firewall that is filtering UDP traffic-</td>
<td>FI-UN.18, FI-UN.19</td>
</tr>
<tr>
<td>FI-PR.06</td>
<td>FI-Session.1</td>
<td>Test suit discovery and selection</td>
<td>FI-Platform SHOULD allow active and passive traffic analysis i.e. the use of Agents and probes installed in the IUT(s).</td>
<td></td>
</tr>
<tr>
<td>FI-PR.07</td>
<td>FI-Session.1</td>
<td>Test suit discovery and selection</td>
<td>FI-Platform SHOULD support different tools to gather network traffic and statistics in real-time.</td>
<td></td>
</tr>
<tr>
<td>FI-PR.08</td>
<td>FI-Session.1</td>
<td>Test suites discovery and selection</td>
<td>FI-Platform MUST provide the user with timeline control to allow the user to set scalability parameters for points in time relative to the beginning of test when using active protocol emulators for scalability tests.</td>
<td>FI-UN.22, FI-UN.23, FI-UN.24</td>
</tr>
</tbody>
</table>

For complete list see https://www.f-interop.eu/index.php/documents/public-deliverables
F-Interop: Location Models and Architecture
Location models

Both IUT and users in same location

Remote user-to-user

Remote single-user testing against automated IUT (hosted by F-Interop)

Both IUT and users in the same location

Remote user-to-user testing

Remote single-user testing against automated IUT (hosted by F-Interop)

Data plane
Control plane
Unspecified interface
Platform architecture overview

Graphical User Interface
Session Orchestrator
Resource Repository
Result Repository
F-Interop central server

Event Bus (RabbitMQ Broker)

Testing Tool

F-Interop core component
Protocol specific component
User’s component

User 1
Agent
IUT 1
environment

Testing tool API

User 2
Agent
IUT 2
environment

Testing tool API
Testing Tool architecture overview

- **Testing Tool**
  - Test Coordinator
  - Traces Dissector
  - Traces Analyser
  - .pcap DB
  - Packet Router & Sniffer
  - GUI adaptor

- **GUI**
  - User A
  - IUT 1
  - PC
  - User B
  - IUT 2
  - PC

- **Agent**
  - IP tun

- **Remote location**

- **F-Interop core**
- **Testing Tool specific**
- **User’s component**
- **Data plane**
- **Control plane**
- **Not-specified interface**

- **GUI adaptor**
- **.pcap**
- **DB**

- **Remote location**

© All rights reserved
A F-Interop session

Paris, 16-18 October 2018
F-Interop session: Only 8 steps to get your IoT device tested

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>User registration, authentication and authorization &amp; IUT registration / identification</td>
</tr>
<tr>
<td>1</td>
<td>Test suites discovery and selection</td>
</tr>
<tr>
<td>2</td>
<td>Session resources description</td>
</tr>
<tr>
<td>3</td>
<td>Resource reservation</td>
</tr>
<tr>
<td>4</td>
<td>Resource provisioning, configuration and session setup</td>
</tr>
<tr>
<td>5</td>
<td>Test execution</td>
</tr>
<tr>
<td>6</td>
<td>Results analysis and reporting</td>
</tr>
<tr>
<td>7</td>
<td>Results and session configuration storage</td>
</tr>
</tbody>
</table>
The F-Interop platform

go.f-interop.eu – go! check it out!
The F-Interop session – step 0

User registration, authentication and authorization
The F-Interop session – step 1

Test suites discovery and selection

- Test suite
- Configuration
- Start

Filter by type:
- Conformance
- Interoperability
- Performance
- Privacy

1. 6TISCH Testing Tool (public)
2. 6LoWPAN test suite (single user) (public)
3. 6LoWPAN test suite (user to user) (public)
4. CoAP test suite (single user) (public)
5. CoAP test suite (user to user) (public)
6. CoMI test suite (single user) (public)
7. CoMI test suite (user to user) (public)
The F-Interop session – step 2

Test Suite Setup

- Testing tool version (optional)
  - testingtool.version

- Test cases selection (optional, all testcases are executed by default)
  - testsuite.testcases

- Automated-IUT selection
  - Californium CoAP Server
  - Californium CoAP Server
  - August Cellars CoAP Server
  - CoAPthon CoAP Server
  - libcoap CoAP Client
  - aiocoap CoAP Client
  - Californium CoAP Client
  - CoAPthon CoAP Client
The F-Interop session – step 3 and step4
Resource reservation and resource provisioning
F-Interop session - Step 5

Test execution (1/3)

How does my implementation will reach other implementations?

For running the tests both implementations need to be reachable. For this we will set up a 2D tunnel (agent only) between both implementations under test (CIT).

The software component for setting this up is called the agent. It plays a role similar to a VPP client.

The following demo will describe how to install and launch the agent. This component will create a tun interface in your PC which allows you to communicate with other implementations. The solution goes more or less like this:

Messages

More about the agent component:
F-Interop session - Step 5
Test execution (2/3)

[Image of F-Interop session interface]

Session

- Session ID
- Pending action
- Test case ID: TD_C0AP_CORE_01
- Test case URL: http://doc.f-interop.eu/testa/TD_C0AP_Core_01
- Test case objective: Perform GET (transaction/C0M model)
- Configuration ID: C0AP_C0M_01
- Configuration URL: http://doc.f-interop.eu/testa/TD_C0AP_C0M_01
- Test case pre-conditions:
  - Server offers the resource /test with resource content in a response
  - Handles GET with an arbitrary payload
- Nodes:
  - c0ap_client
  - c0ap_servec
- C0AP серверs are requested to offer the following resources:
  - /test
  - Should not exceed 1kbytes
  - Should not exceed 253bytes
  - Should not exceed 1kbytes
  - Should not exceed 253bytes
  - Resource not containing query parameters
  - Resource not containing parameters
  - Resource which cannot be served immediately and which cannot be acknowledged

Messages

- Next test case to be executed:
- Test case ID: TD_C0AP_C0M_01
- Test case URL: http://doc.f-interop.eu/testa/TD_C0AP_Core_01
- Test case objective: Perform GET (transaction/C0M model)
- Configuration ID: C0AP_C0M_01
- Configuration URL: http://doc.f-interop.eu/testa/TD_C0AP_C0M_01
- Test case pre-conditions:
  - Server offers the resource /test with resource content in a response
  - Handles GET with an arbitrary payload
- Nodes:
  - c0ap_client
  - c0ap_server
- C0AP серверs are requested to offer the following resources:
  - /test
  - Should not exceed 1kbytes
  - Should not exceed 253bytes
  - Should not exceed 1kbytes
  - Should not exceed 253bytes
  - Resource not containing query parameters
  - Resource not containing parameters
  - Resource which cannot be served immediately and which cannot be acknowledged

[Image of F-Interop session interface]
F-Interop session - Step 5
Test execution (3/3)
F-Interop session - Step 6
Analysis of traces, verdicts and reporting (1/2)
F-Interop session - Step 6
Analysis of traces, verdicts and reporting (2/2)
F-Interop session - Step 7
Session results and configuration storage

<table>
<thead>
<tr>
<th>Sessions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>id:</strong> 4fb09e12-0451-4f7f-aa07-5e63c745d23a</td>
<td><strong>Result</strong></td>
</tr>
<tr>
<td><strong>status:</strong> terminated</td>
<td></td>
</tr>
<tr>
<td><strong>event:</strong> 092bb585-a6a6-486c-8661-6f8f0104682</td>
<td></td>
</tr>
<tr>
<td><strong>password:</strong> EZZPM2RV</td>
<td></td>
</tr>
<tr>
<td><strong>start_date:</strong> 29/08/2018 16:19</td>
<td></td>
</tr>
<tr>
<td><strong>testSuiteType:</strong> interoperability</td>
<td></td>
</tr>
<tr>
<td><strong>amqp_url:</strong> amqp://7B0WK0VB:<a href="mailto:EZZPM2RV@mq.dev.f-interop.eu">EZZPM2RV@mq.dev.f-interop.eu</a>:443/4fb09e12-0451-4f7f-aa07-5e63c745d23a</td>
<td></td>
</tr>
<tr>
<td><strong>login:</strong> 7B0WK0VB</td>
<td></td>
</tr>
<tr>
<td><strong>users:</strong> federico_sismondioujx, myslice</td>
<td></td>
</tr>
<tr>
<td><strong>end_date:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>testSuite:</strong></td>
<td></td>
</tr>
<tr>
<td><strong>slice_id:</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sessions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>id:</strong> 55a63a9d-98dd-4973-9101-c18fafb9c730</td>
<td><strong>Result</strong></td>
</tr>
<tr>
<td><strong>status:</strong> terminated</td>
<td></td>
</tr>
<tr>
<td><strong>password:</strong> DF34VT04</td>
<td></td>
</tr>
<tr>
<td><strong>start_date:</strong> 29/08/2018 11:58</td>
<td></td>
</tr>
<tr>
<td><strong>testSuiteType:</strong> Interoperability</td>
<td></td>
</tr>
<tr>
<td><strong>amqp_url:</strong> amqp://ORSX725P-DF34VT04@mq.dev.f-interop.eu:443/55a63a9d-98dd-4973-9101-c18fafb9c730</td>
<td></td>
</tr>
<tr>
<td><strong>login:</strong> ORSX725P</td>
<td></td>
</tr>
<tr>
<td><strong>users:</strong> federico_sismondioujx, myslice</td>
<td></td>
</tr>
</tbody>
</table>
F-Interop: interoperability testing in practice

Paris, 16-18 October 2018
CoAP protocol use case

• What’s CoAP?
  • IETF standard for the IoT, with a web approach
  • client-server communications
  • ~ HTTP for IoT but enabling asynchronous transactions

• Testing CoAP interoperability, what do we need?
  • Two implementations, a client and a server
  • Test specification
  • Test Setup
  • Demo using F-Interop (video)
CoAP defines request-response exchange pattern

<table>
<thead>
<tr>
<th>Client</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>+------&gt; Header: GET (T=CON, Code=0.01, MID=0x7d35)</td>
<td></td>
</tr>
<tr>
<td>GET    Token: 0x20</td>
<td></td>
</tr>
<tr>
<td>Uri-Path: &quot;temperature&quot;</td>
<td></td>
</tr>
<tr>
<td>&lt;------+ Header: 2.05 Content (T=ACK, Code=2.05, MID=0x7d35)</td>
<td></td>
</tr>
<tr>
<td>2.05   Token: 0x20</td>
<td></td>
</tr>
<tr>
<td>Payload: &quot;22.3 C&quot;</td>
<td></td>
</tr>
</tbody>
</table>

Also enables SUSBCRIPTION to states

Sources:

<table>
<thead>
<tr>
<th>t</th>
<th>Observed State</th>
<th>CLIENT</th>
<th>SERVER</th>
<th>Actual State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>unknown</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>18.5 Cel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>18.5 Cel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>19.2 Cel</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: A Client Registers and Receives One Notification of the Current State and One of a New State upon a State Change
CoAP remote interop test. What do we need?

F-Interop platform

- F-Interop core
- Testing Tool specific
- User’s component
- Data plane
- Control plane
- Not-specified interface

IUT 1
User A
PC

IUT 2
User B
PC

Remote location
F-Interop platform
F-Interop Server
Remote location
CoAP remote interop test.
Tunneling
Packet sniffer

F-Interop platform

Testing Tool
Packet Router & Sniffer

F-Interop core
Testing Tool specific
User's component
Data plane
Control plane
Not-specified interface

User A
Agent
IUT 1
PC
Remote location

User B
Agent
IUT 2
PC
Remote location
CoAP remote interop test.

Tunneling
Packet sniffing
Network traces logging

F-Interop platform

Testing Tool

Not-specified interface
Agent
IUT 1
IP tun
User A
PC
Remote location

IUT 2
IP tun
User B
PC
Remote location

F-Interop core
Testing Tool specific
User's component
Data plane
Control plane
Not-specified interface
CoAP remote interop test.

- Tunneling
- Packet sniffing
- Network traces logging
- Test coordination
- Network traces dissection

F-Interop platform

Testing Tool
- Test Coordinator
- Traces Dissector
- .pcap DB
- Packet Router & Sniffer
- GUI adaptor

F-Interop Server

GUI

Agent

IUT 1

IP tun

User A

PC

Remote location

User's component

F-Interop core

Testing Tool specific

User's component

Data plane

Control plane

Not-specified interface

IUT 2

IP tun

User B

PC

Remote location
CoAP remote interop test.

Tunneling
Packet sniffing
Network traces logging
Test coordination
Network traces dissection
Network traces analysis

F-Interop platform

Testing Tool
- Test Coordinator
- Traces Dissector
- Traces Analyser
- .pcap DB
- Packet Router & Sniffer
- GUI adaptor

F-Interop Server

GUI
Agent
IUT 1
IP tun
PC
User A
Remote location

GUI
Agent
IUT 2
IP tun
PC
User B
Remote location

F-Interop core
Testing Tool specific
User’s component
Data plane
Control plane
Not-specified interface
How do we add new tests?

1. **GUI adaptor**
2. **F-Interop Server**
3. **F-Interop platform**
4. **Test case description (YAML format)**
5. **Test case script for .pcap analysis (python API)**
6. **Testing Tool**
7. **Test case description (YAML format)**
8. **Test case description (YAML format)**
9. **Test case description (YAML format)**

**Remote location**

**F-Interop platform**

**F-Interop Server**

**Testing Tool**

**Traces Dissector**

**Traces Analyser**

**Packet Router & Sniffer**

**.pcap DB**

**IP tun**

**Remote location**

**GUI**

**Agent**

**IUT 1**

**PC**

**Remote location**

**GUI**

**Agent**

**IUT 2**

**PC**

**IP tun**

**User A**

**User B**

**Remote location**

**Remote location**
Under the Hood: What’s a test?

1. Test description in human and machine language

```
--- testcase
 testcase_id: TD_COAP_CORE_01_v01
 url: http://f-interop.paris.inria.fr/tests/TD_COAP_CORE_01_v01
 configuration: CoAP_configuration_BASIC
 objective: Perform GET transaction (CON node)
 pre_conditions: Server offers the resource /test with resource content is not empty that handles GET with an arbitrary payload
 references: '{CoAP} 5.8.1, 1.2, 2.1, 2.2, 3.1'
 sequence:
   - step_id: 'TD_COAP_CORE_01_v01_step_01'
     type: stimuli
     lut: coap_client
     description:
       - Client is requested to send a GET request with
       - Type = 1(CON)
       - Code = 1(GET)
   - step_id: TD_COAP_CORE_01_v01_step_02
     type: check
     description:
       - The request sent by the client contains
       - Type=0 and Code=1
       - Client-generated Message ID (\u2794 CMD)
       - Client-generated Token (\u2794 CTOK)
       - Uri-Path option "test"
   - step_id: TD_COAP_CORE_01_v01_step_03
     type: check
     description:
       - Server sends response containing
       - Code = 2.05(Content)
       - Message ID = CMD, Token = CTOK
       - Content-format option
       - Non-empty Payload
   - step_id: TD_COAP_CORE_01_v01_step_04
     type: verify
     lut: coap_client
     description:
       - Client displays the received information
```

- **STIMULI** either executed manually by USER
- Or by a automated IUT

- **CHECK** executed automatically by Analyzer on the exchanged traces
  (automatically issue verdicts!)

- **VERIFY** executed manually by user! (User sends verify response using GUI)
Under the Hood: What’s a test?

2. Test analysis scripts for CHECK for conformance and interoperability errors

```python
#!/usr/bin/env python3
from ttproto.ts_coap.common import CoAPTestCase
from ttproto.ts_coap.templates import *

class TD_COAP_CORE_01 (CoAPTestCase):
    def run(self):
        # match stimuli
        self.match_coap("client", CoAP(type="con", code="get",
                                      opt = self.uri ("/test")))
        CMID = self.frame.coap["mid"]
        CTOK = self.frame.coap["tok"]

        # match step 2
        self.next()
        if self.match_coap("server", CoAP (code = 2.05,
                                           mid = CMID,
                                           tok =CTOK,
                                           pl = Not(b""),
                                           )):
            # match step 3
            self.match_coap("server", CoAP (opt = Opt (CoAPOptionContentFormat()),
                                             ), "fail")
```
Demo: Single-user interop session use case
( 10 mins approx.)
F-Interop: current status
F-Interop-Platform

Developed by F-Interop

CoAP interop testing tool
CoAP server performance testing tool
WoT interop testing tool
Privacy testing tool

Developed by F-Interop contributor

oneM2M interop testing tool
6TiSCH conformance testing tool
LoRaWAN conformance testing tool
OSCORE conformance testing tool
CoMI interop testing tool
Semantic data validator testing tool

F-Interop-User
Uses test tools and tests his IUT

F-Interop-Contributor
Provides new tests, updates test suites, etc.

Developed by F-Interop contributor

CoAP interop testing tool
oneM2M interop testing tool
6LoWPAN interop testing tool
OMA Lightweight M2M interop testing tool
6LoWPAN interop testing tool
OSCORE conformance testing tool
CoMI interop testing tool
Semantic data validator testing tool
F-Interop’s face-to-face and remote interoperability test events:
- 14-15 July 2017: ETSI #1 F-Interop 6TiSCH Interoperability Event, Prague (CZ), Inria-UR1
- October 2018: F-Interop CoAP Interoperability Event, Full remote, To be organized by Kereval & Inria-UR1
F-Interop: Achievements and next steps
Achievements and next step

• F-Interop project is officially ending in November 2018
• **F-Interop platform answers to the predefined needs**
  • Platform enables full remote & online interop test execution
  • Tests IoT implementations from companies’ premises
  • Feedbacks to standardization process
  • Shortens time-to-market for IoT implementations
• **Future of the F-Interop platform**
  • Remote online ETSI’s plugtests events?
  • Labelling/certification for IoT protocols?
  • Continuous interoperability testing! Already in the pipes!
Continuous interoperability testing

- Not a just idea, we have running code!!
- Check out [https://fedmon.fed4fire.eu/](https://fedmon.fed4fire.eu/) -> F-Interop IoT tests
Continuous interoperability testing
Continuous interoperability testing
Thank you

Contacts:
federico.sismondi@irisa.fr ; cesar.viho@irisa.fr
F-Interop’s interoperability testing tools

- “The purpose of interoperability testing is to prove that end-to-end functionality between (at least) two communicating systems is as required by the standard(s) on which those systems are based.” (ETSI)
- testing tools create an environment which will help the user(s) executing online and remote standard-based interoperability tests
- communication “tunneling” between IUTs (a VPN-like setup)
  - controlled environment
  - bypass UDP-blocking firewalls and other middle boxes
- coordinating the interop test
  - dispatches commands to users (through GUI) based on test descriptions
  - e.g. ‘user1: CoAP Client is requested to send a GET request with...’
- sniffing the traffic
  - generate PCAP files records
  - Users can download PCAP files at the end of test session.
- dissecting the messages
  - include wireshark-like view of the exchanges between implementations
  - help users find problems in the messages exchanged by the implementations
- analyzing the traffic exchanged
  - automatically issue PASS/FAIL/INCONCLUSIVE verdicts based on the test description
Testing Tool architecture overview:

The diagram illustrates the architecture of a testing tool, including various components and their interactions.

- **GUI**: Graphical User Interface
- **Agent**: Central component for test coordination
- **IUT 1, IUT 2**: Items Under Test
- **PC**: Personal Computer
- **IP tun**: IP tunneling
- **Not-specified interface**: Interface not specified in the diagram
- **GUI adaptor**: Adapter for the GUI

The connections and interactions between these components highlight the testing process, including test coordinator, dissector, analyzer, router, sniffer, and analyzer for specific F-Interop core and Testing Tool functionalities.

The diagram also shows user locations (User A and User B) connected through remote locations (PC and User B's location).
 DEMAND: Survey Results – Remote vs. Face-2-face

Do you think your organization will participate more in testing services when provided remotely?

![Pie chart showing 75% Yes, 25% No]