PLATFORM FOR ONLINE INTEROPERABILITY AND PERFORMANCE TEST

F-INTEROP

Remote Conformance & Interop Testing

TPAC2016 – Web of Things IG Meeting – Lisbon
22nd September 2016

César Viho & Federico Sismondi
INRIA - France

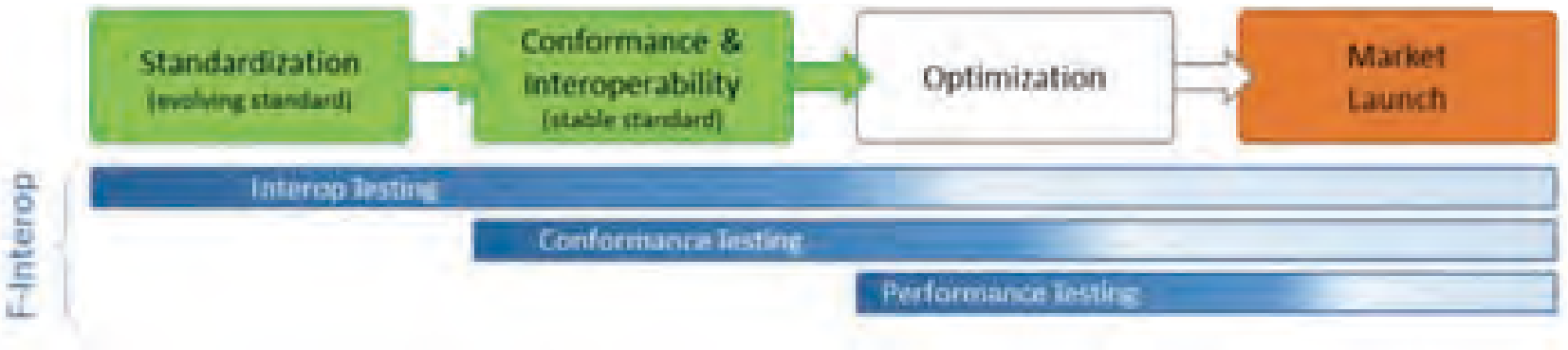
F-Interop H2020 Project

- www.f-interop.eu
- 1 November 2015 – 31 October 2018
- *develop and provide online interoperability and performance test tools to support emerging technologies from research to standardization and market launch*
- 9 partners

Goals

1. Describe the F-Interop platform
2. Is this useful for the WoT community?
3. How the WoT community can help?
   - Introduce the F-Interop open call

Why remote conformance & interop?

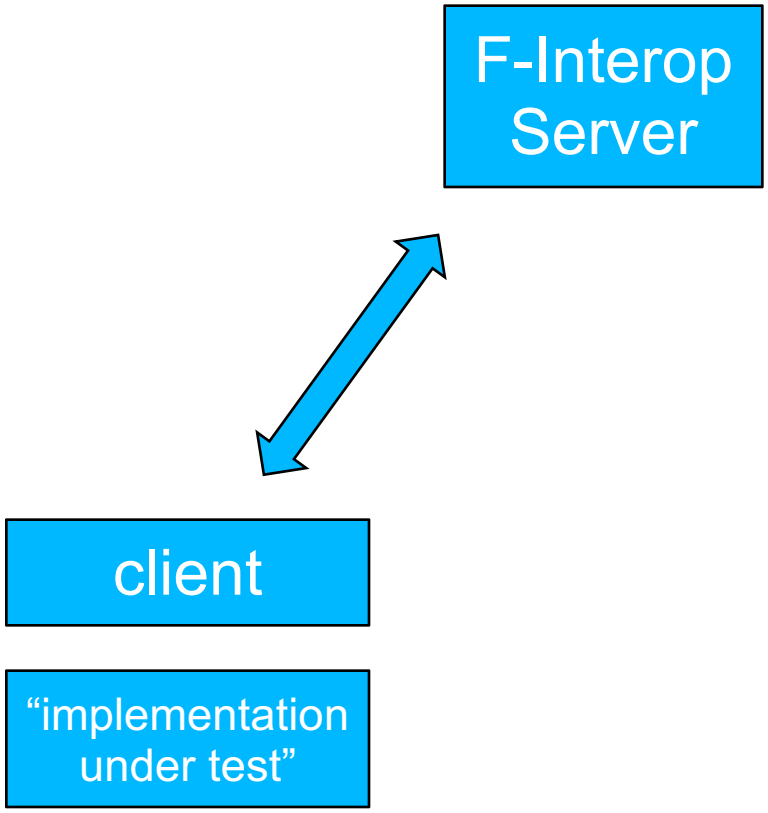


- **SDOs**
  - save time and resources
  - running code early
  - accelerate standardization process

- **SMEs and companies**
  - interop tests without needing to travel
  - lower development cost
  - faster development of standards-based products

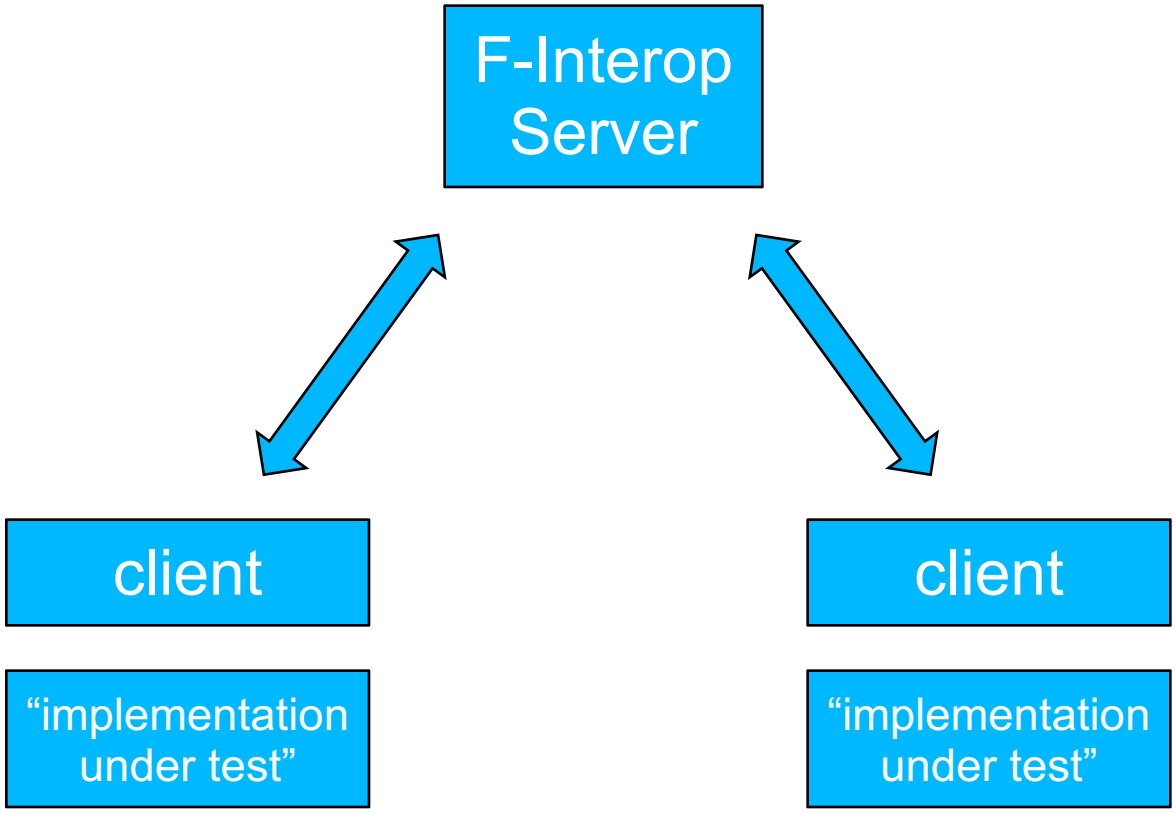
→ more standards-based products

Core Idea



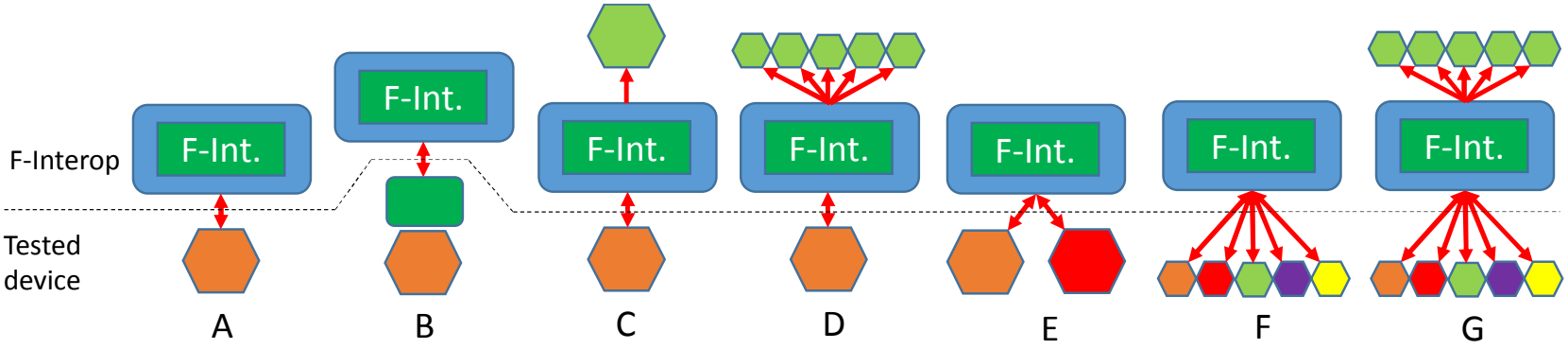
Conformance Testing

Core Idea



Interop Testing

Different Configurations



A. Tested Device ←→ F-Interop test server
B. Deported test with downloaded resource
C. Remote interop with 2 participants
D. Interop against testbed
E. Local interop
F. Remote interop with N participants
G. Remote interop with N participants and testbeds

Testbeds

32 testbeds, 4755 nodes

- **Fed4FIRE** (www.fed4fire.eu/testbeds)
  - 24 testbeds
  - ~1000 nodes
- **OneLab** (onelab.eu)
  - Includes 6 IoT-lab deployments (including 2728 IoT nodes)
- **IoT lab** (www.iotlab.eu)

Targeted Standards

- Initially standards of the IoT realm
  - CoAP
  - 6TiSCH
  - 6LoWPAN

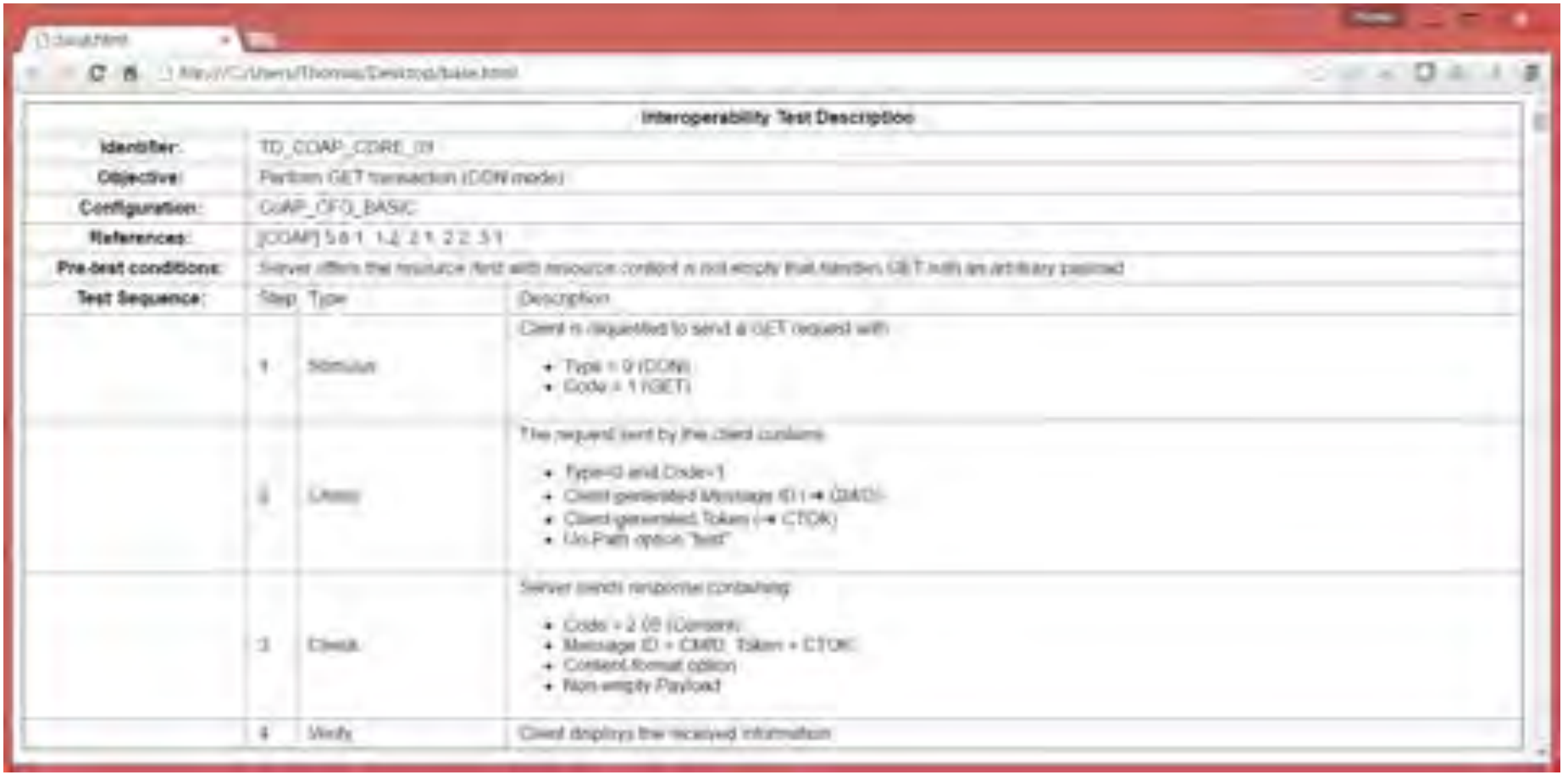
- We take, as a starting point, the ETSI plugtests specifications and build an architecture that allows those to be done remotely

- **Contributions/extensions are expected by design**
  - Including:
    - oneM2M
    - **Web of Things (WoT)**

CoAP remote online interop testing
A proof of concept

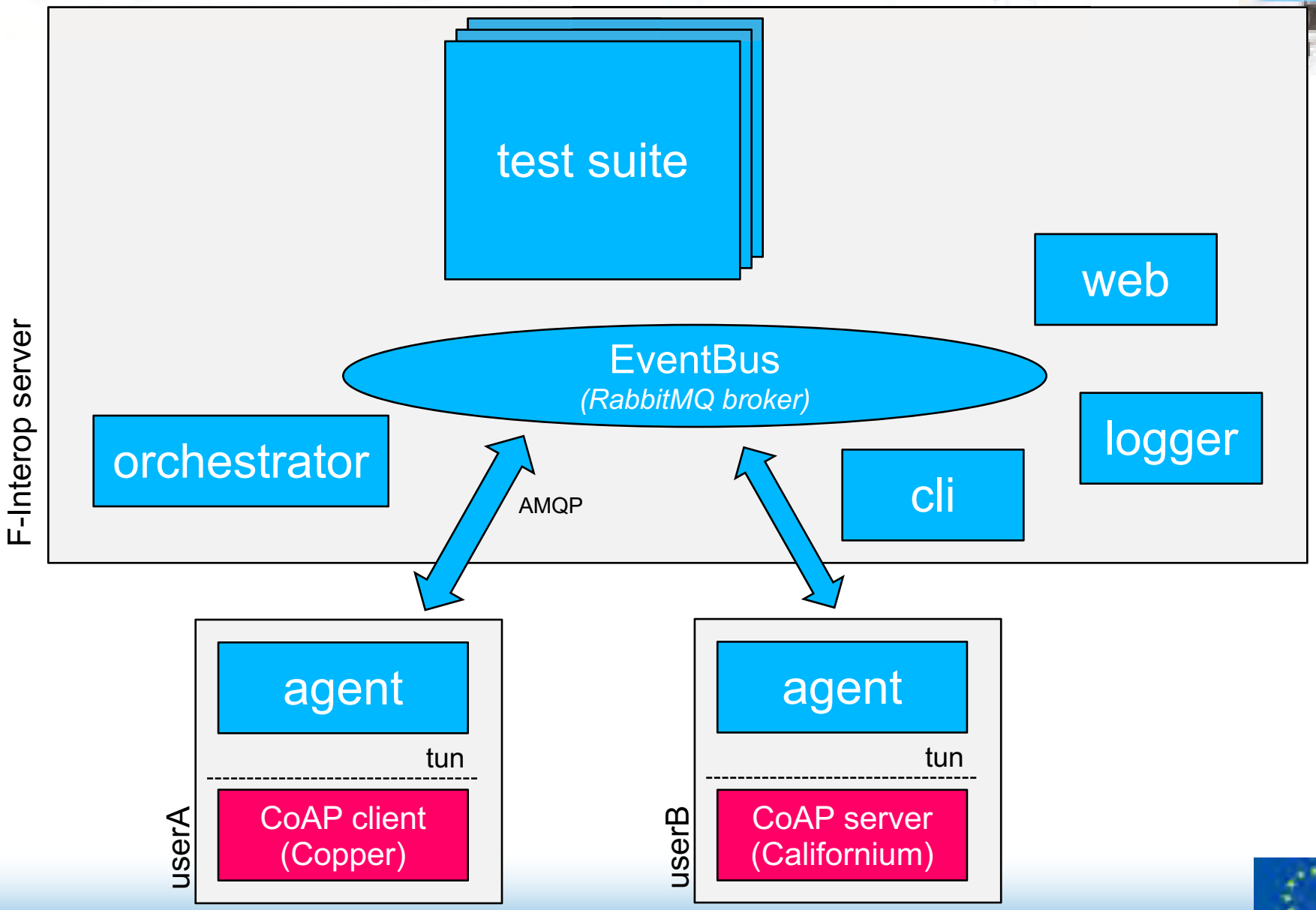
Example CoAP Test

- From ETSI plugtest CoAP#4, IETF89 (London)

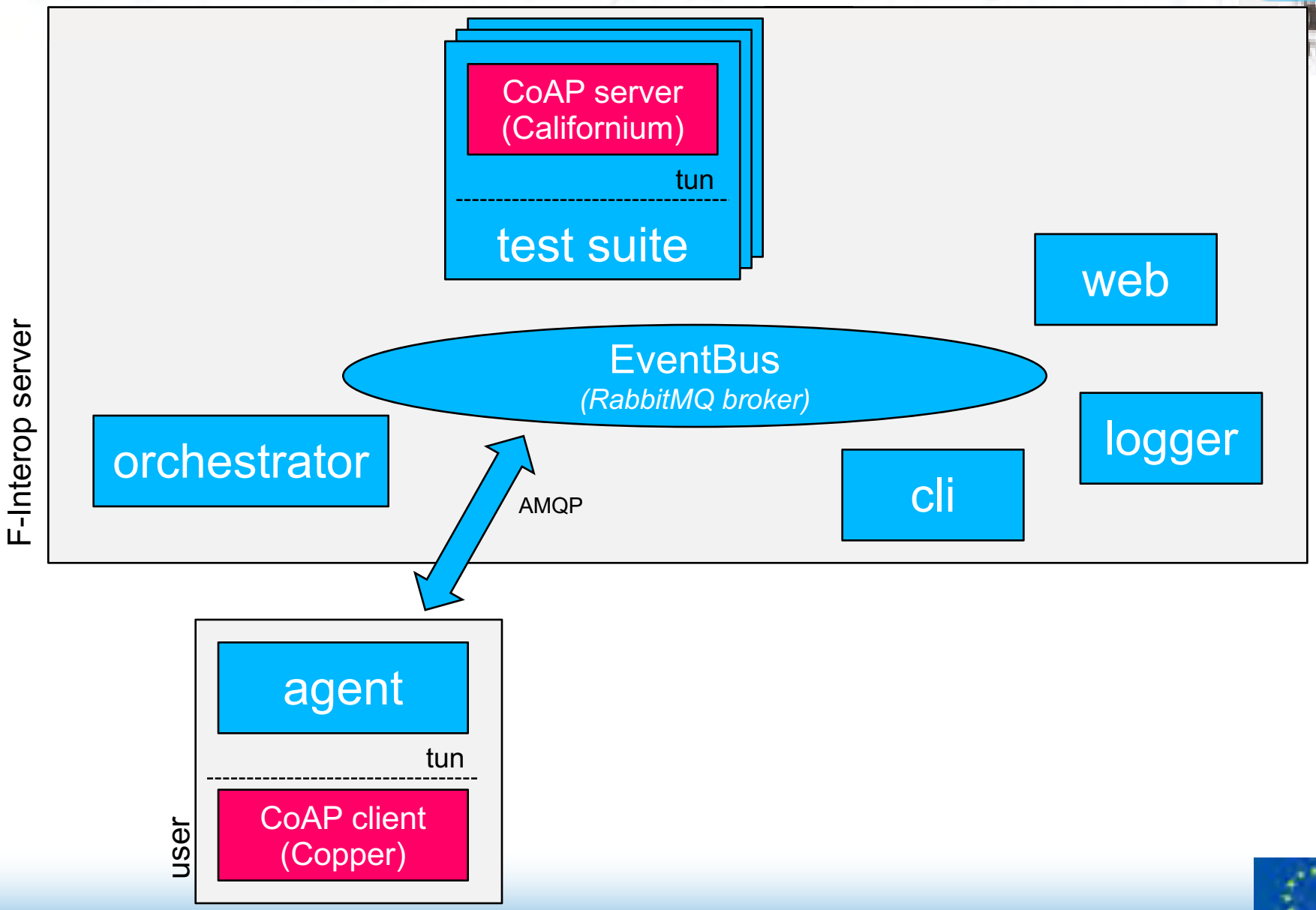


Interoperability Test Description			
Identifier:	TD_COAP_CORE_09		
Objective:	Perform GET transaction (CON mode)		
Configuration:	CoAP_CFG_BASIC		
References:	[CoAP] 5.8.1, 1.2, 2.1, 2.2, 5.1		
Pre-test conditions:	Server offers the resource /test with resource content is not empty (Mandatory), GET (with an arbitrary payload)		
Test Sequence:	Step	Type	Description
	1	Stimulus	Client is requested to send a GET request with • Type = 0 (CON) • Code = 1 (GET)
	2	Check	The request sent by the client contains • Type=0 and Code=1 • Client generated Message ID (→ CMID) • Client generated Token (→ CTOK) • Uri-Path option "test"
	3	Check	Server sends response containing • Code = 2.05 (Content) • Message ID = CMID, Token = CTOK • Content-format option • Non-empty Payload
	4	Verify	Client displays the received information

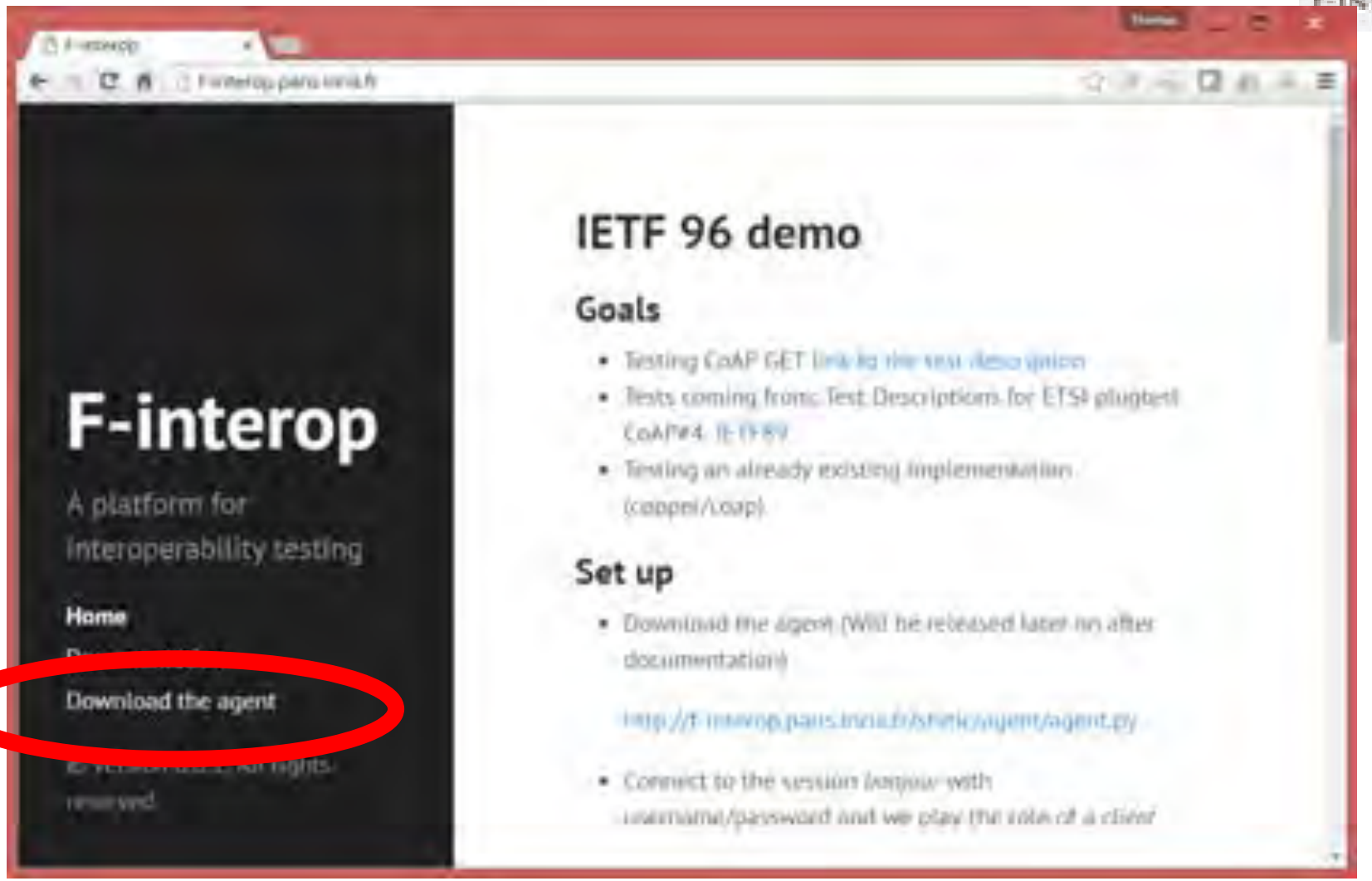
Base Architecture (CoAP interop)



Base Architecture (CoAP interop demo)

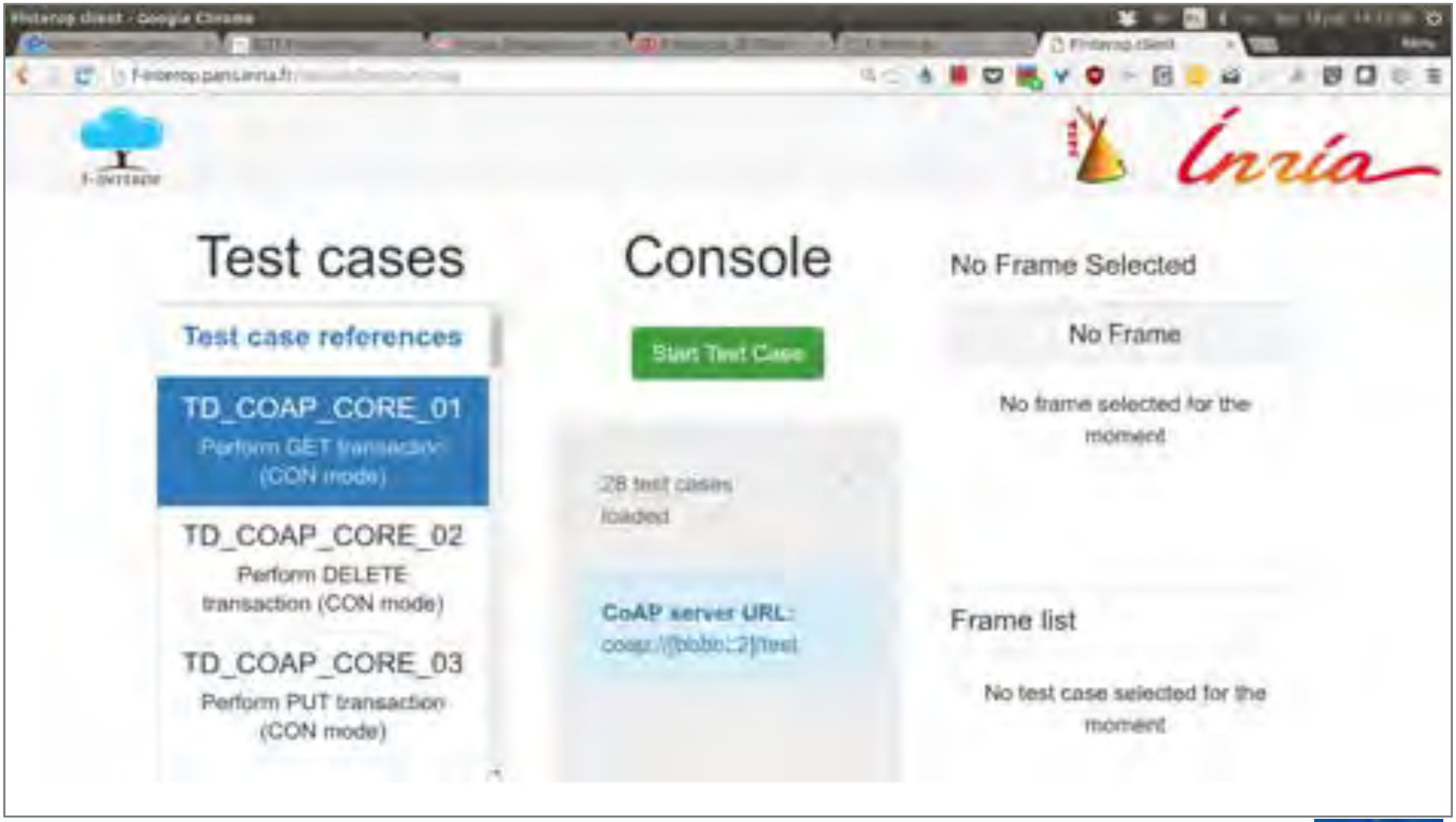


Download the Agent

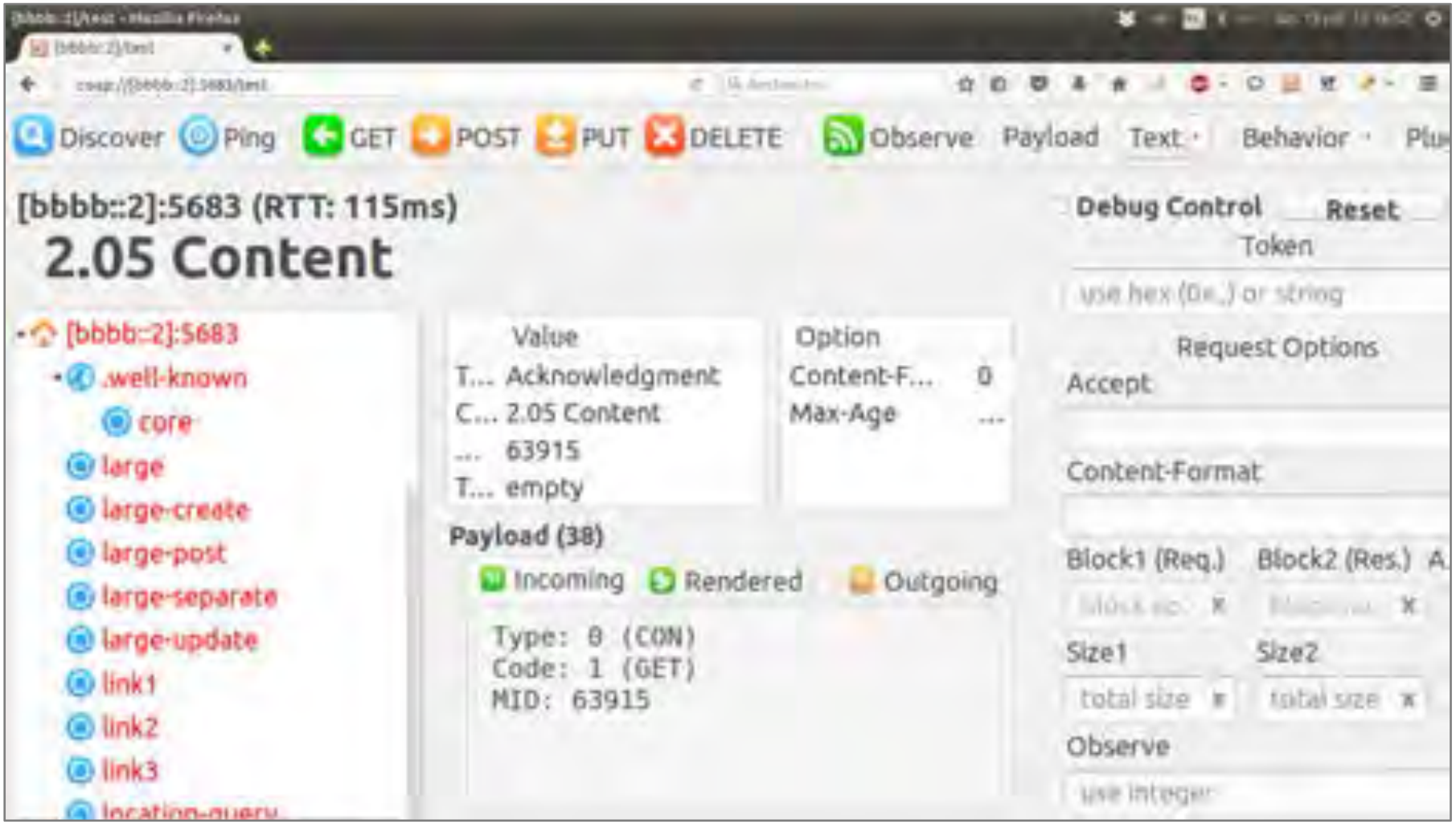


Connect to the F-Interop Server

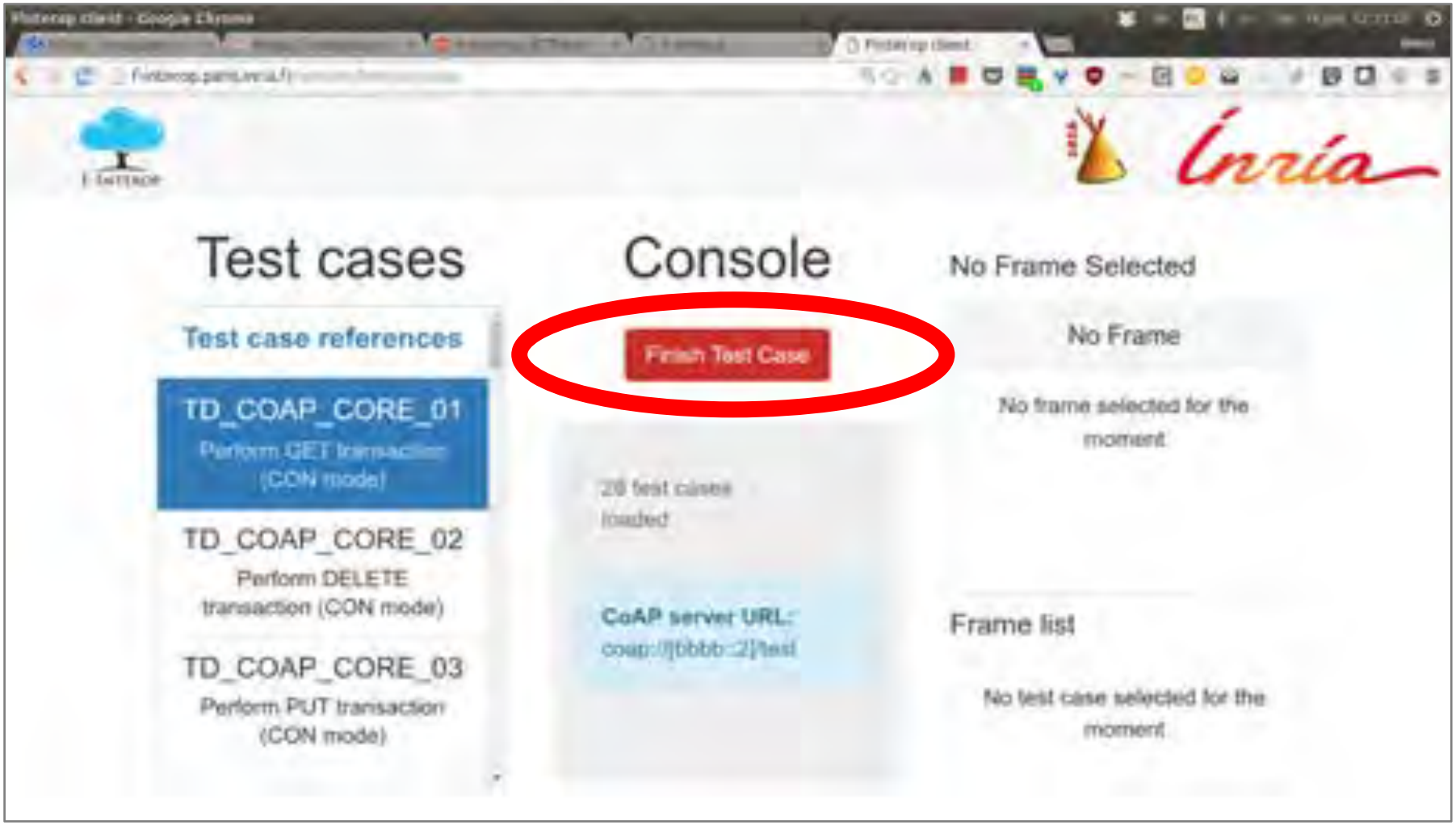
Select and Start the Test Case



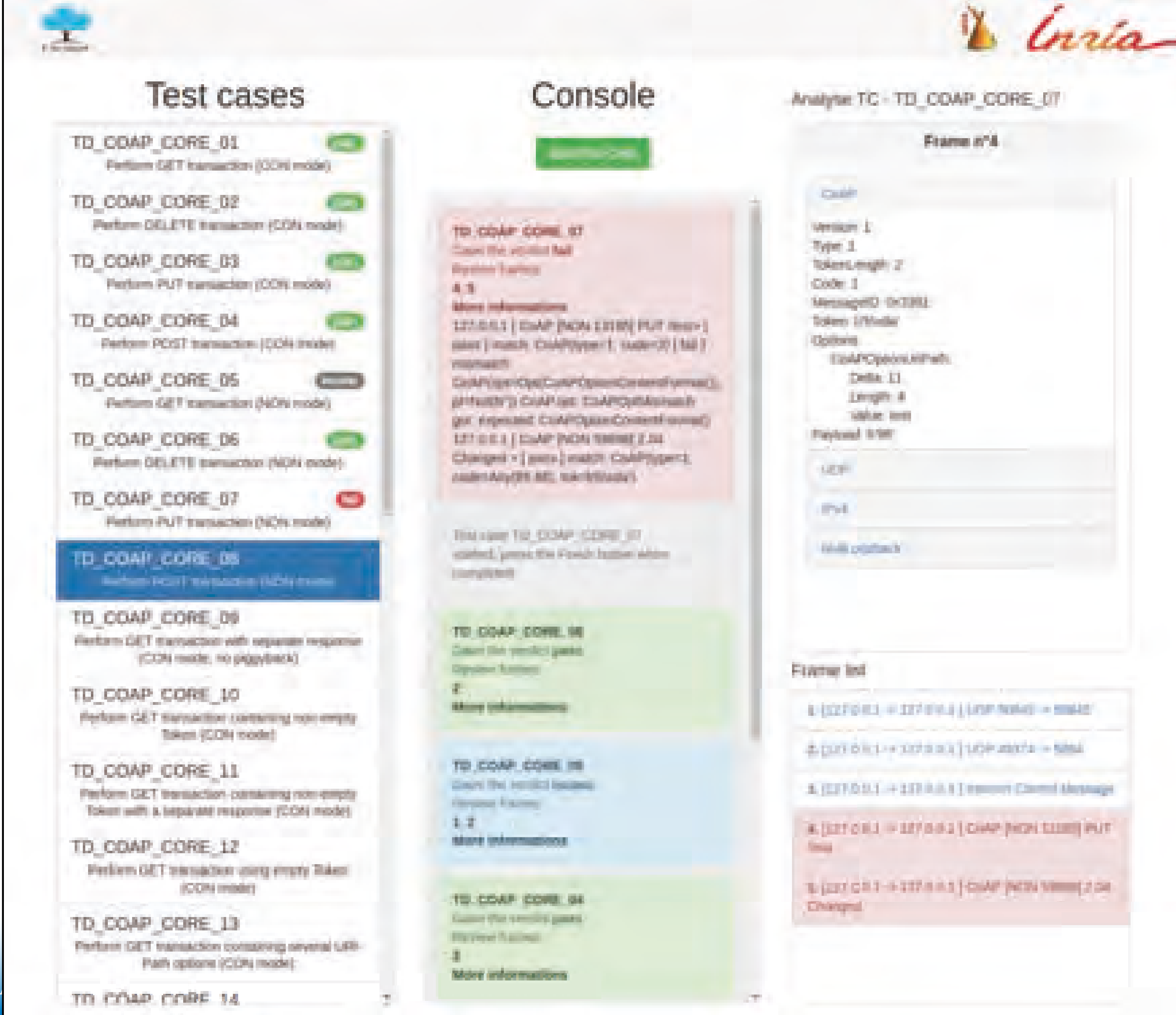
Send CoAP Packets



Finish Test Case

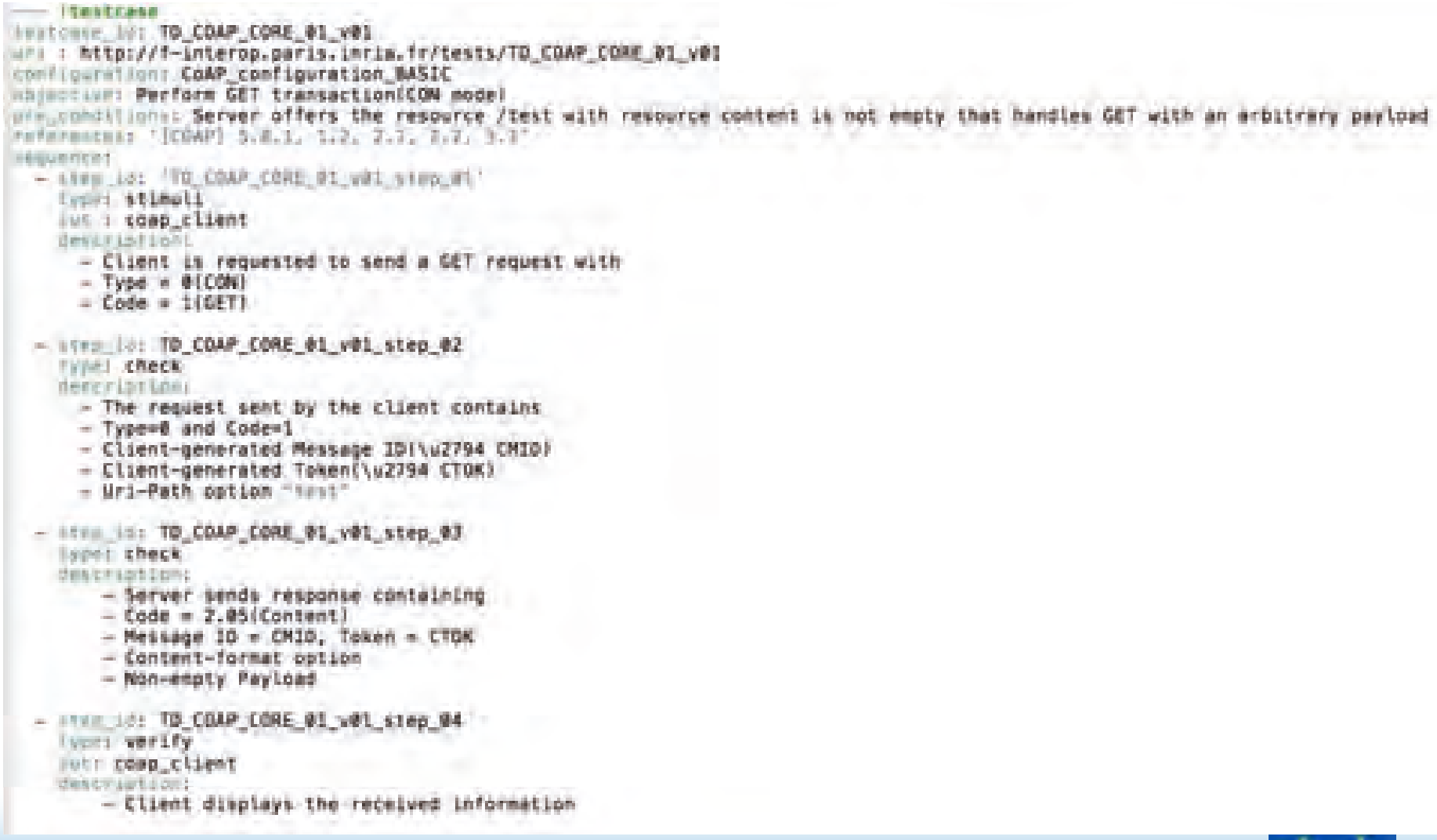


Verdict!



Test cases
Console
Analyse TC - TD_COAP_CORE_07
Frame n°4
TD_COAP_CORE_01
Perform GET transaction (CON mode)
TD_COAP_CORE_02
Perform DELETE transaction (CON mode)
TD_COAP_CORE_03
Perform PUT transaction (CON mode)
TD_COAP_CORE_04
Perform POST transaction (CON mode)
TD_COAP_CORE_05
Perform GET transaction (NON mode)
TD_COAP_CORE_06
Perform DELETE transaction (NON mode)
TD_COAP_CORE_07
Perform PUT transaction (NON mode)
TD_COAP_CORE_08
TD_COAP_CORE_09
TD_COAP_CORE_10
TD_COAP_CORE_11
TD_COAP_CORE_12
TD_COAP_CORE_13
Frame list

Under the Hood: What's a test?



```
--- !testcase
testcase_id: TD_COAP_CORE_01_v01
uri : http://f-interop.paris.inria.fr/tests/TD_COAP_CORE_01_v01
configuration: CoAP_configuration_BASIC
objective: Perform GET transaction(CON mode)
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
    ref : coap_client
    description:
      - Client is requested to send a GET request with
      - Type = 0(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 CMID)
      - 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 = CMID, Token = CTOK
        - Content-format option
        - Non-empty Payload

  - step_id: TD_COAP_CORE_01_v01_step_04
    type: verify
    ref: coap_client
    description:
        - Client displays the received information
```

Under the Hood: What's a test?

```
#!/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")
```

Next Milestones

- July 2016
  - minimal CoAP interop testing (done) -> see demo
- November 2016
  - Functional platform available
  - CoAP CORE interop tests
- March 2017
  - 6TiSCH support, update at IETF98
  - CoAP interop test (advanced version)
- July 2017
  - Use at 6TiSCH/6lo plugtests
  - **minimal WoT interop testing**

WoT interop test case example

Properties

Identifier	**TC_WOT_BASE_01**
Objective	Read Boolean Property
References	3.2.3.1 Property, 3.2.4.1 Simple Data
Pre-test conditions	Exposing Thing provides boolean Property
Test sequence	
1. Stimulus	Consuming Thing sends `Retrieve` to Property
2. Check	Consuming Thing sends - protocol operation bound to `Retrieve` - no payload - to Property URI
3. Check	Exposing Thing sends - positive response code - payload formatted according to TD
4. Verify	Consuming Thing displays read value

Source: https://github.com/w3c/wot/blob/master/plugfest/2016-beijing/plugfest-test-cases-beijing-2016.md

How the WoT community can help?

- **Contributors:**
  - Help us extending F-Interop for interop in WoT context
  - List requirements, identify key priority WoT standards
  - Develop test suites for (new) standards
  - Provide feedback on architecture and choices
- **Users:**
  - Use F-Interop for remote interop events/plugtests

Open Call

Open Call Categories

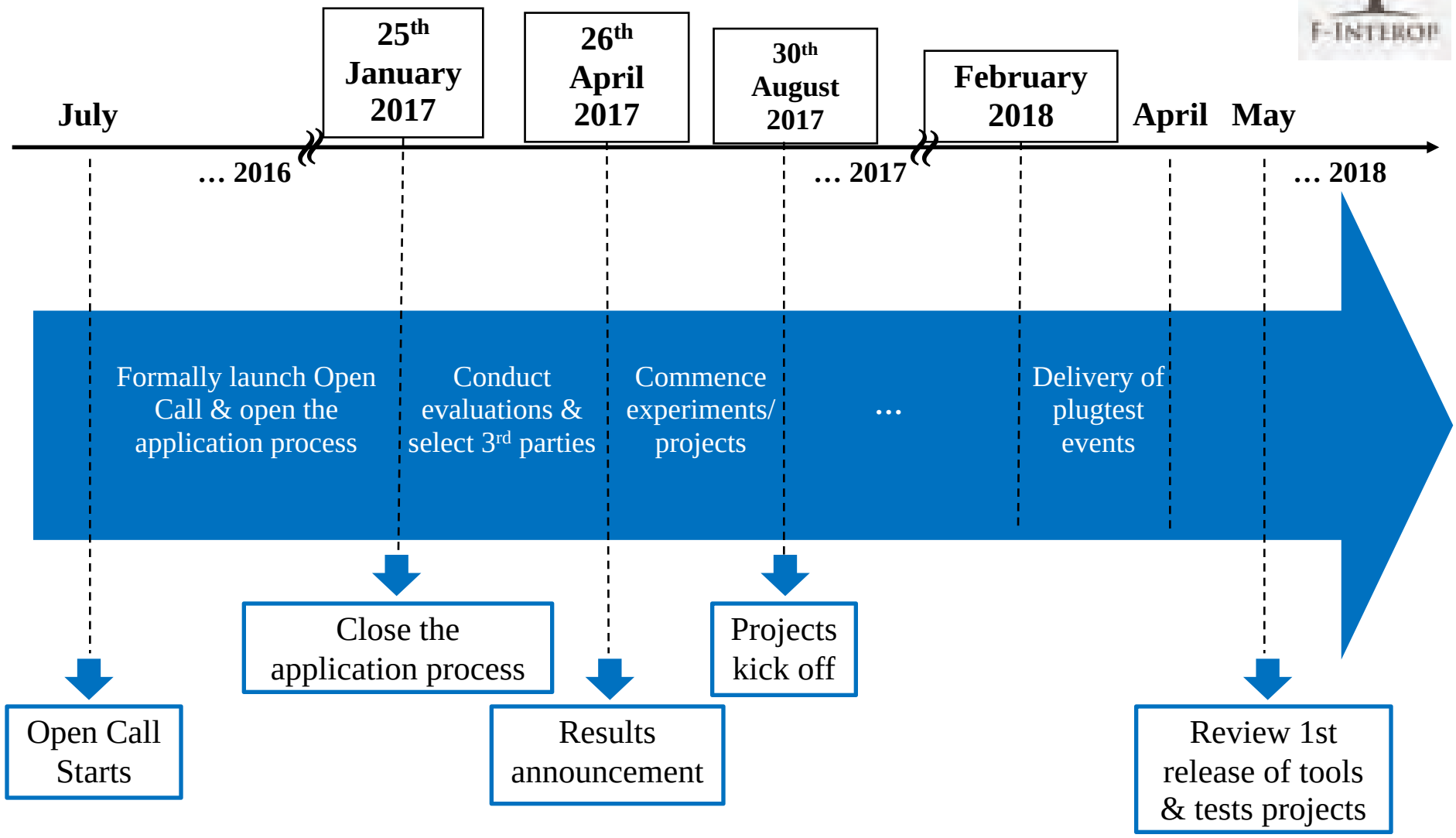
- **New testing tools** to extend capabilities of F-Interop
- **New test descriptions** to test conformance and interoperability of other standards
- **SME F-Interop assessment reports:** SME device Interop tests to test F-Interop platform
- **Plugtest Events:** Third parties selected to conduct 3 remote online plugtest events

Supported Activities & Budget

610k for 19 projects

List of Categories	Grants	Award
New F-Interop tools extensions	3	100 000
New interop test descriptions	3	60 000
SME devices F-Interop tests and report	10	10 000
Plugtest Events	**3**	**10 000**

Important Dates



How to apply?

- Template for the proposal
- Guide for Applicants
- Standard Industrial Experiment Contract
- Open Call Terms and Conditions
- **Submission Portal**

http://www.f-interop.eu/index.php/open-call

Thank you for your attention

Open-call: http://www.f-interop.eu/index.php/open-call

Please, feel free to contact us directly or later via:
Federico.Sismondi@inria.fr, Cesar.Viho@irisa.fr