

# Remote Conformance & Interop Testing

**Info Session – IETF96 – Berlin**  
**18 July 2016**

Thomas Watteyne, Remy Leone  
Federico Sismondi, Maria Rita Palattella

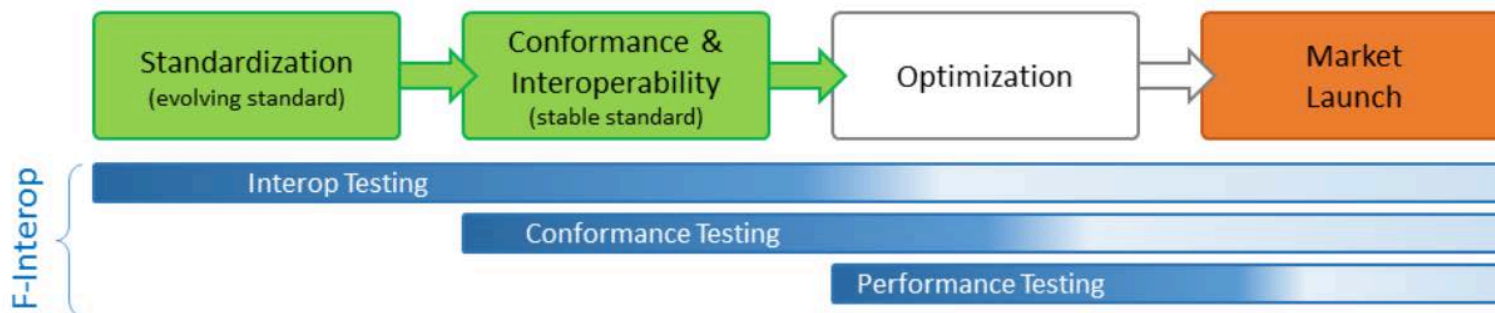
# Goals



1. Describe the F-Interop platform
2. Is this useful for the IETF community?
3. How can the IETF community contribute?



# Why remote conformance & interop?



## ➤ SDOs

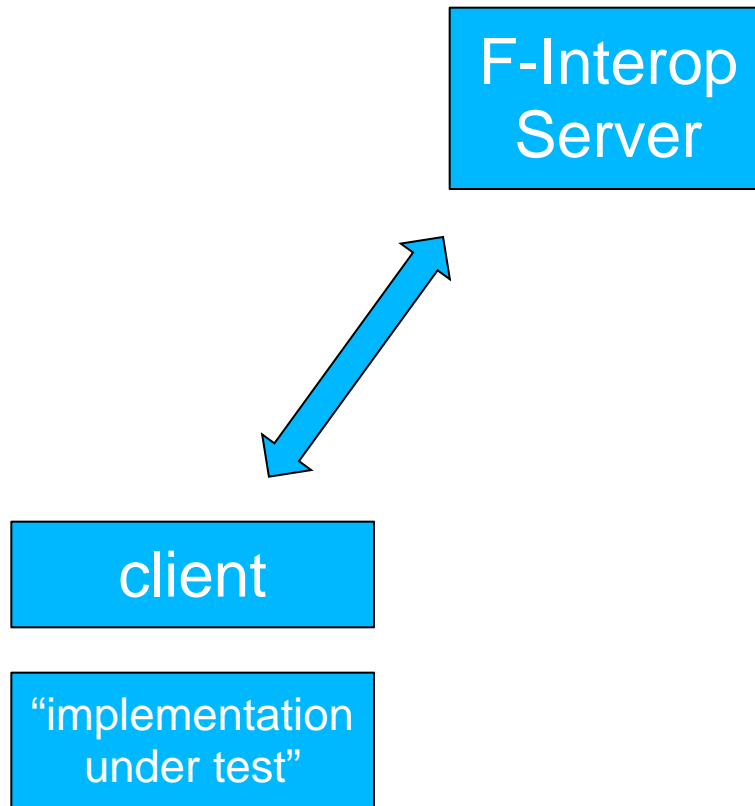
- save time and resources
- running code early
- accelerate standardization process

## ➤ SMEs and companies

- interop tests without needing to travel
- lower development cost
- faster development of standards-based products

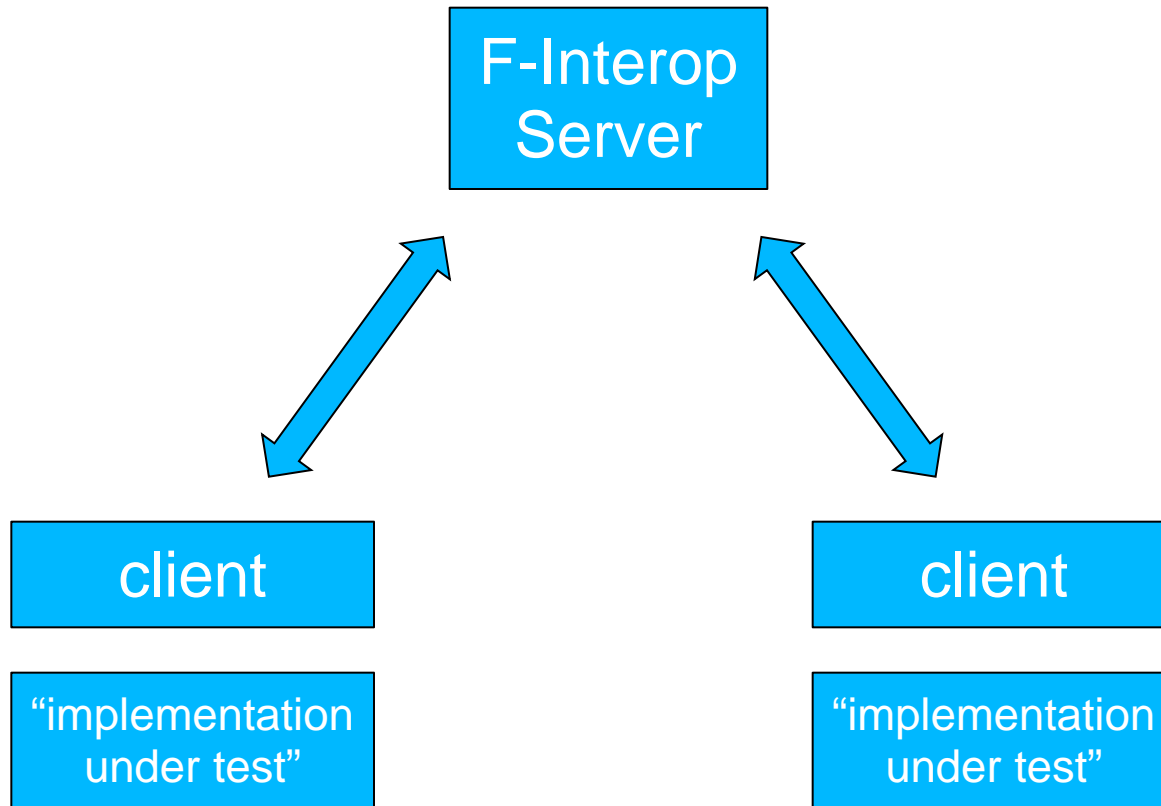
→ more standards-based products

# Core Idea



Conformance Testing

# Core Idea



Interop Testing

# Different Configurations

- A. Tested Device  $\leftrightarrow$  F-Interop test server
- B. Deployed 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

# F-Interop H2020 Project



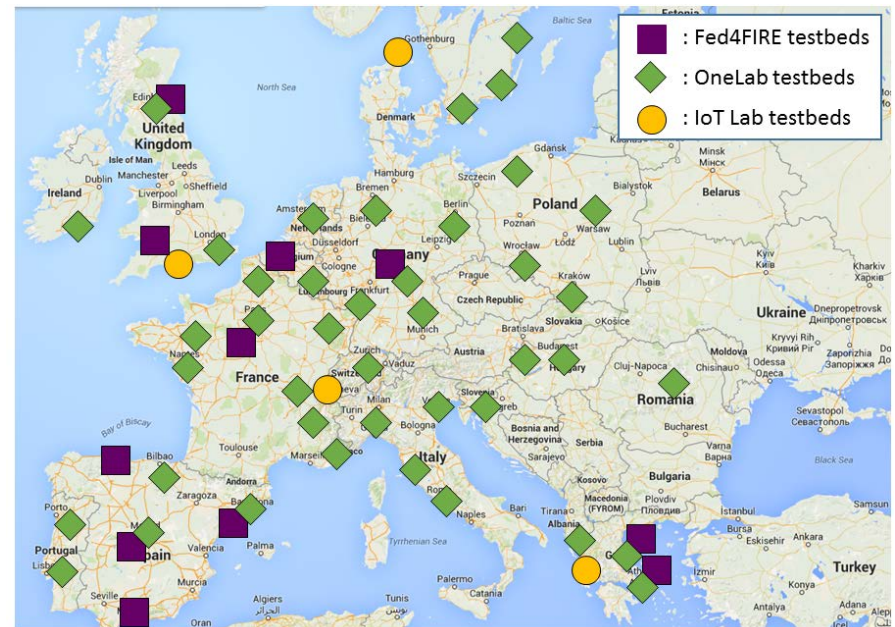
- [www.f-interop.eu](http://www.f-interop.eu)
- 1 November 2015 – 31 October 2018
- *develop and provide online interoperability and performance test tools to support emerging technologies from research to standardization and market launch*
- 9 partners



# Testbeds

32 testbeds, 4755 nodes

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





# Targeted Standards

- Initially standards of the IoT realm
- We take, as a starting point, the ETSI plugtests specifications and build an architecture that allows those to be done remotely (CoAP, 6TiSCH, 6LoWPAN)
- **Contributions/extensions are expected by design**

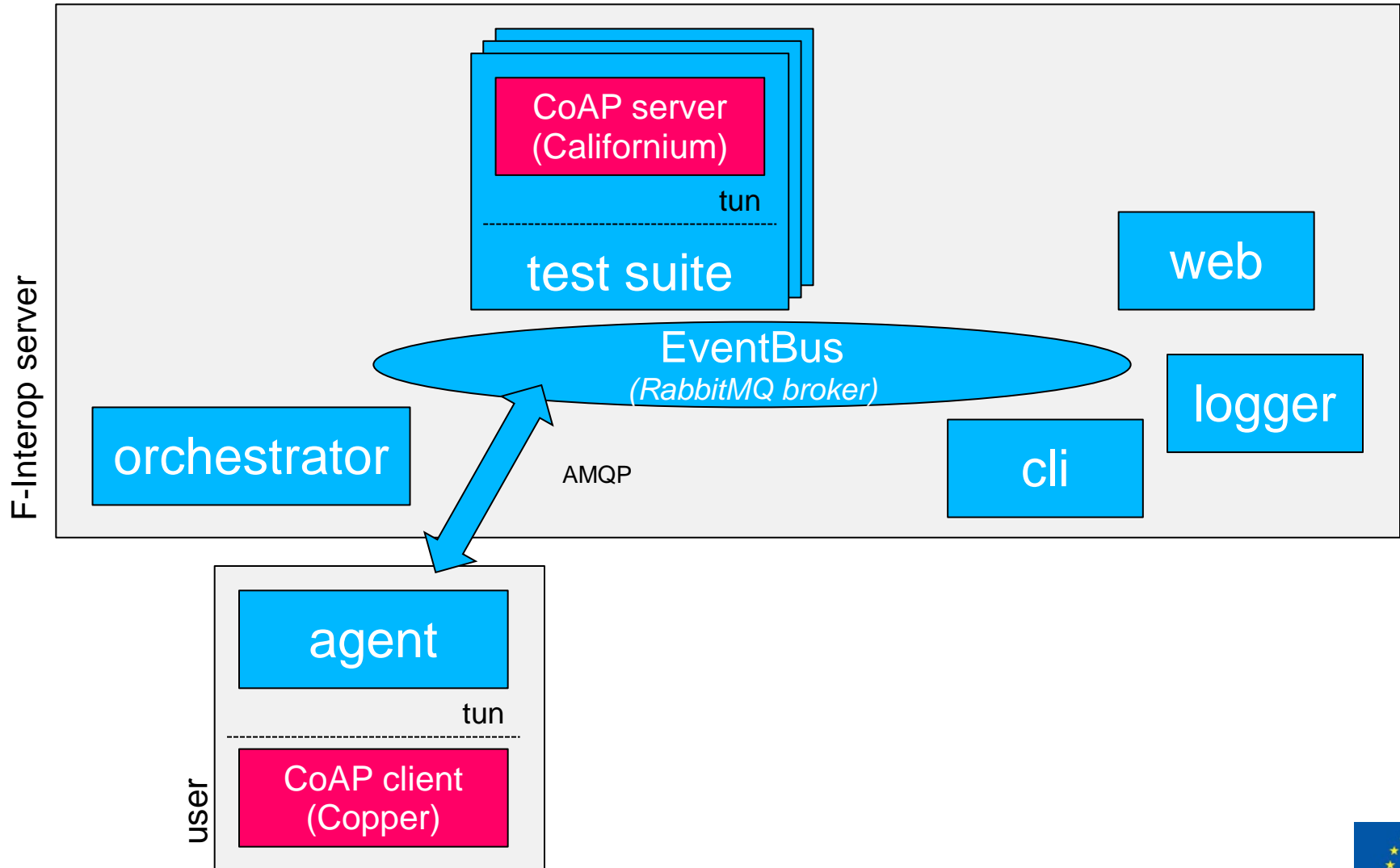
## Demo!

# Example CoAP Test

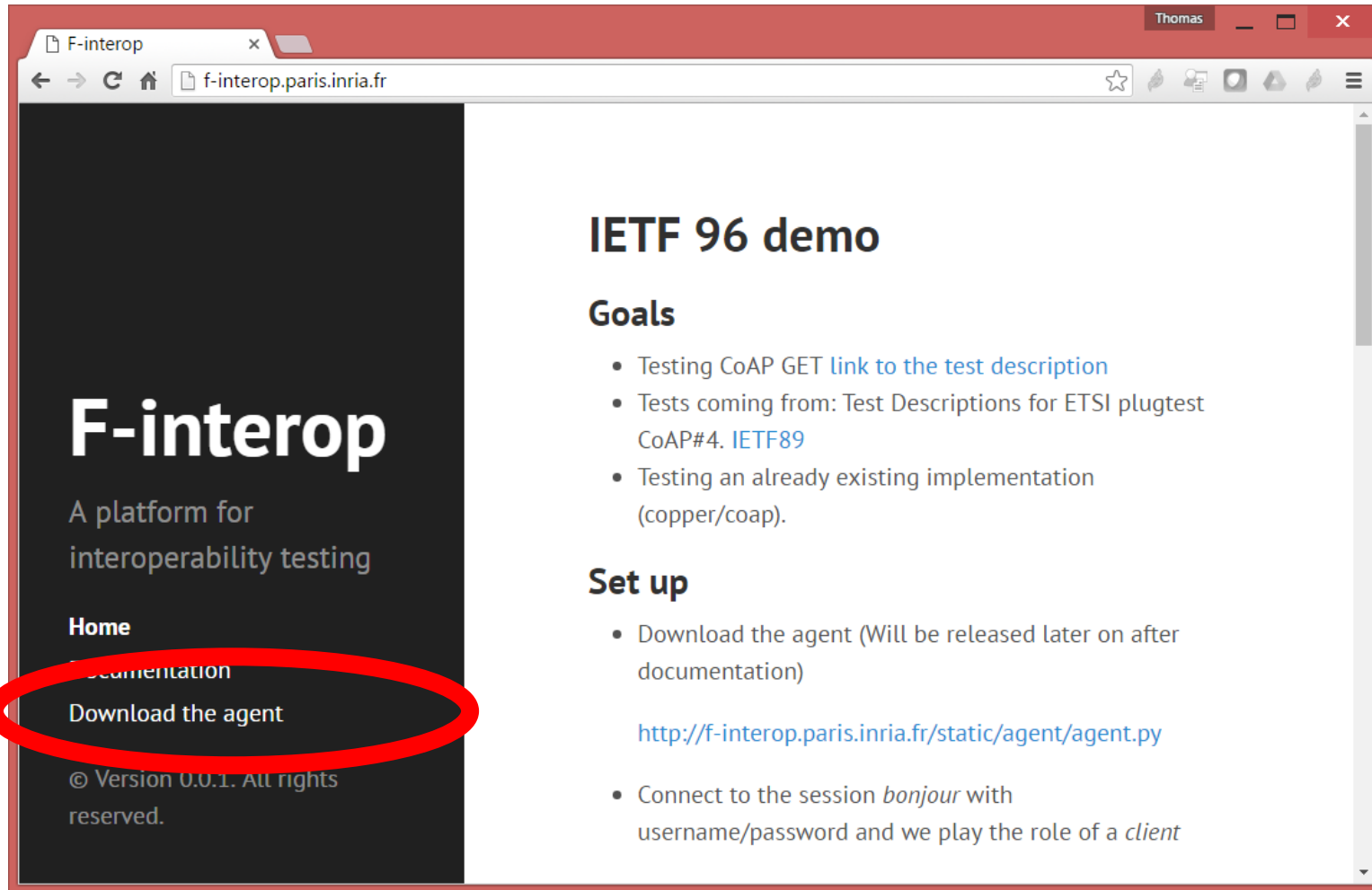
- From ETSI plugtest CoAP#4, IETF89 (London)

Interoperability Test Description		
Identifier:	TD_COAP_CORE_01	
Objective:	Perform GET transaction (CON mode)	
Configuration:	CoAP_CFG_BASIC	
References:	[COAP] 5.8.1, 1.2, 2.1, 2.2, 3.1	
Pre-test conditions:	Server offers the resource /test with resource content is not empty that handles GET with an arbitrary payload	
Test Sequence:	Step	Type Description
	1	Stimulus Client is requested to send a GET request with: <ul style="list-style-type: none"><li>Type = 0 (CON)</li><li>Code = 1 (GET)</li></ul>
	2	Check The request sent by the client contains: <ul style="list-style-type: none"><li>Type=0 and Code=1</li><li>Client-generated Message ID (→ CMID)</li><li>Client-generated Token (→ CTOK)</li><li>Uri-Path option "test"</li></ul>
	3	Check Server sends response containing: <ul style="list-style-type: none"><li>Code = 2.05 (Content)</li><li>Message ID = CMID, Token = CTOK</li><li>Content-format option</li><li>Non-empty Payload</li></ul>
	4	Verify Client displays the received information

# Base Architecture (CoAP example)



# Download the Agent



F-interop

A platform for interoperability testing

Home

Documentation

**Download the agent**

© Version 0.0.1. All rights reserved.

## IETF 96 demo

### Goals

- Testing CoAP GET [link to the test description](#)
- Tests coming from: Test Descriptions for ETSI plugtest CoAP#4. [IETF89](#)
- Testing an already existing implementation (copper/coap).

### Set up

- Download the agent (Will be released later on after documentation)  
  
<http://f-interop.paris.inria.fr/static/agent/agent.py>
- Connect to the session *bonjour* with username/password and we play the role of a *client*

# Connect to the F-Interop Server



```
# sieben @ sieben-lincs in ~/Dropbox/workspace/f-interop_ietf on git:develop x [14:29:58] C:1
$ sudo python -m finterop.agent.agent connect --user bonjour --session bonjour --name client
Password: █
```






# Select and Start the Test Case

Finterop client - Google Chrome

Inbox - remy.leone x IETF Proceedings x https://rawgit.com/ x F-Interop\_IETFBerlin x F-interop x Finterop client x Rémy

f-interop.paris.inria.fr/session/bonjour/coap



## Test cases

### Test case references

- 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)

## Console

**Start Test Case**

28 test cases loaded

**CoAP server URL:**  
coap://[bbbb::2]/test

## No Frame Selected

No Frame

No frame selected for the moment

### Frame list

No test case selected for the moment

# Send CoAP Packets

[bbbb::2]/test - Mozilla Firefox

coap://[bbbb::2]:5683/test

Discover Ping GET POST PUT DELETE Observe Payload Text Behavior Plug

[bbbb::2]:5683 (RTT: 115ms)

## 2.05 Content

☐ Debug Control

Token

use hex (0x..) or string

Request Options

Accept

Content-Format

Block1 (Req.) Block2 (Res.) A

block no. x block no. x

Size1 Size2

total size x total size x

Observe

use integer

**Value**

T... Acknowledgment
C... 2.05 Content
... 63915
T... empty

**Option**

Content-F...	0
Max-Age	...

**Payload (38)**

☒ Incoming ☐ Rendered ☐ Outgoing

Type: 0 (CON)  
Code: 1 (GET)  
MID: 63915

**[bbbb::2]:5683**




- .well-known
  - core
- large
- large-create
- large-post
- large-separate
- large-update
- link1
- link2
- link3
- location-query

# Finish Test Case

Finterop client - Google Chrome

https://rawgit.com/ x F-Interop\_IETFBerlin x F-interop x Finterop client x Rémy

f-interop.paris.inria.fr/session/bonjour/coap

## Test cases

**Test case references**

- 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)

## Console

**Finish Test Case**

28 test cases loaded

**CoAP server URL:**  
coap://[bbbb::2]/test

## No Frame Selected

No Frame

No frame selected for the moment

### Frame list

No test case selected for the moment



## Test cases

TD_COAP_CORE_01	pass
Perform GET transaction (CON mode)	
TD_COAP_CORE_02	pass
Perform DELETE transaction (CON mode)	
TD_COAP_CORE_03	pass
Perform PUT transaction (CON mode)	
TD_COAP_CORE_04	pass
Perform POST transaction (CON mode)	
TD_COAP_CORE_05	inconc
Perform GET transaction (NON mode)	
TD_COAP_CORE_06	pass
Perform DELETE transaction (NON mode)	
TD_COAP_CORE_07	fail
Perform PUT transaction (NON mode)	
TD_COAP_CORE_08	
Perform POST transaction (NON mode)	
TD_COAP_CORE_09	
Perform GET transaction with separate response (CON mode, no piggyback)	
TD_COAP_CORE_10	
Perform GET transaction containing non-empty Token (CON mode)	
TD_COAP_CORE_11	
Perform GET transaction containing non-empty Token with a separate response (CON mode)	
TD_COAP_CORE_12	
Perform GET transaction using empty Token (CON mode)	
TD_COAP_CORE_13	
Perform GET transaction containing several URI-Path options (CON mode)	
TD_COAP_CORE_14	

## Console

Start Test Case

## TD\_COAP\_CORE\_07

Gave the verdict **fail**

Review frames:

4, 5

## More informations

127.0.0.1 ] CoAP [NON 13185] PUT /test> [ pass ] match: CoAP(type=1, code=3) [ fail ] mismatch:  
CoAP(opt=Opt(CoAPOptionContentFormat()), pl=Not(b'')) CoAP.opt: CoAPOptMismatch got: expected: CoAPOptionContentFormat()  
127.0.0.1 ] CoAP [NON 59898] 2.04 Changed > [ pass ] match: CoAP(type=1, code=Any(65,68), tok=b'b\alda')

Test case TD\_COAP\_CORE\_07 started, press the Finish button when completed

## TD\_COAP\_CORE\_06

Gave the verdict **pass**

Review frames:

2

## More informations

## TD\_COAP\_CORE\_05

Gave the verdict **inconc**

Review frames:

1, 2

## More informations

## TD\_COAP\_CORE\_04

Gave the verdict **pass**

Review frames:

2

## More informations

Analyse TC - TD\_COAP\_CORE\_07

## Frame n°4

## CoAP

Version: 1  
Type: 1  
TokenLength: 2  
Code: 3  
MessageID: 0x3381  
Token: b'b\alda'  
Options:  
CoAPOptionUriPath:  
Delta: 11  
Length: 4  
Value: test  
Payload: b'98'

## UDP

## IPv4

## NullLoopback

## Frame list

1. [127.0.0.1 -> 127.0.0.1 ] UDP 50845 -> 50845
2. [127.0.0.1 -> 127.0.0.1 ] UDP 49374 -> 5684
3. [127.0.0.1 -> 127.0.0.1 ] Internet Control Message
4. [127.0.0.1 -> 127.0.0.1 ] CoAP [NON 13185] PUT /test
5. [127.0.0.1 -> 127.0.0.1 ] CoAP [NON 59898] 2.04 Changed

# 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):

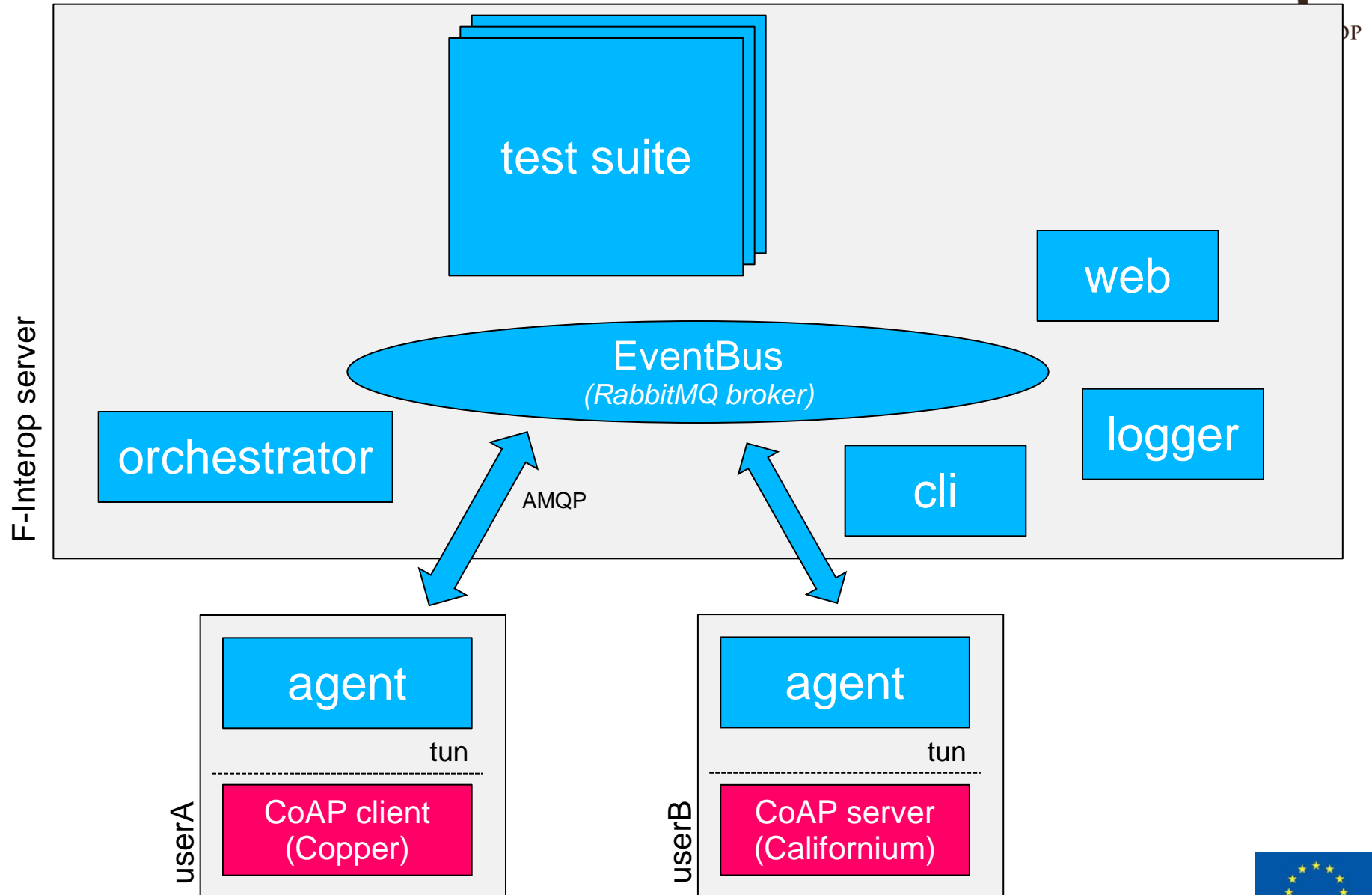
    def run (self):

        # 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"]

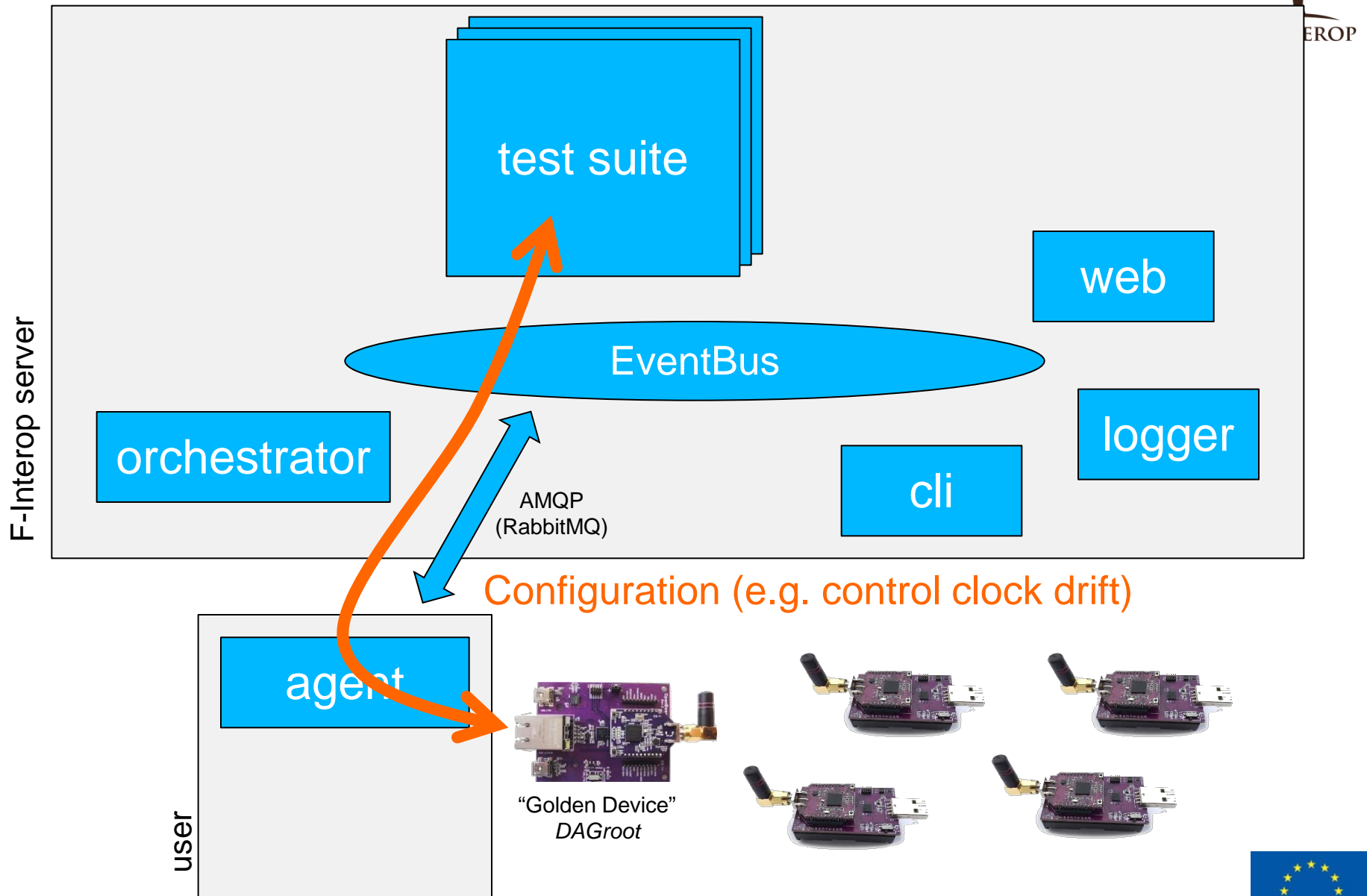
        # match step 2
        self.next()
        if self.match_coap ("server", CoAP (
            code = 2.05,
            mid = CMID,
            tok = CTOK,
            pl = Not(b""),
        )):

            # match step 3
            self.match_coap ("server", CoAP (
                opt = Opt (CoAPOptionContentFormat()),
            ), "fail")
```

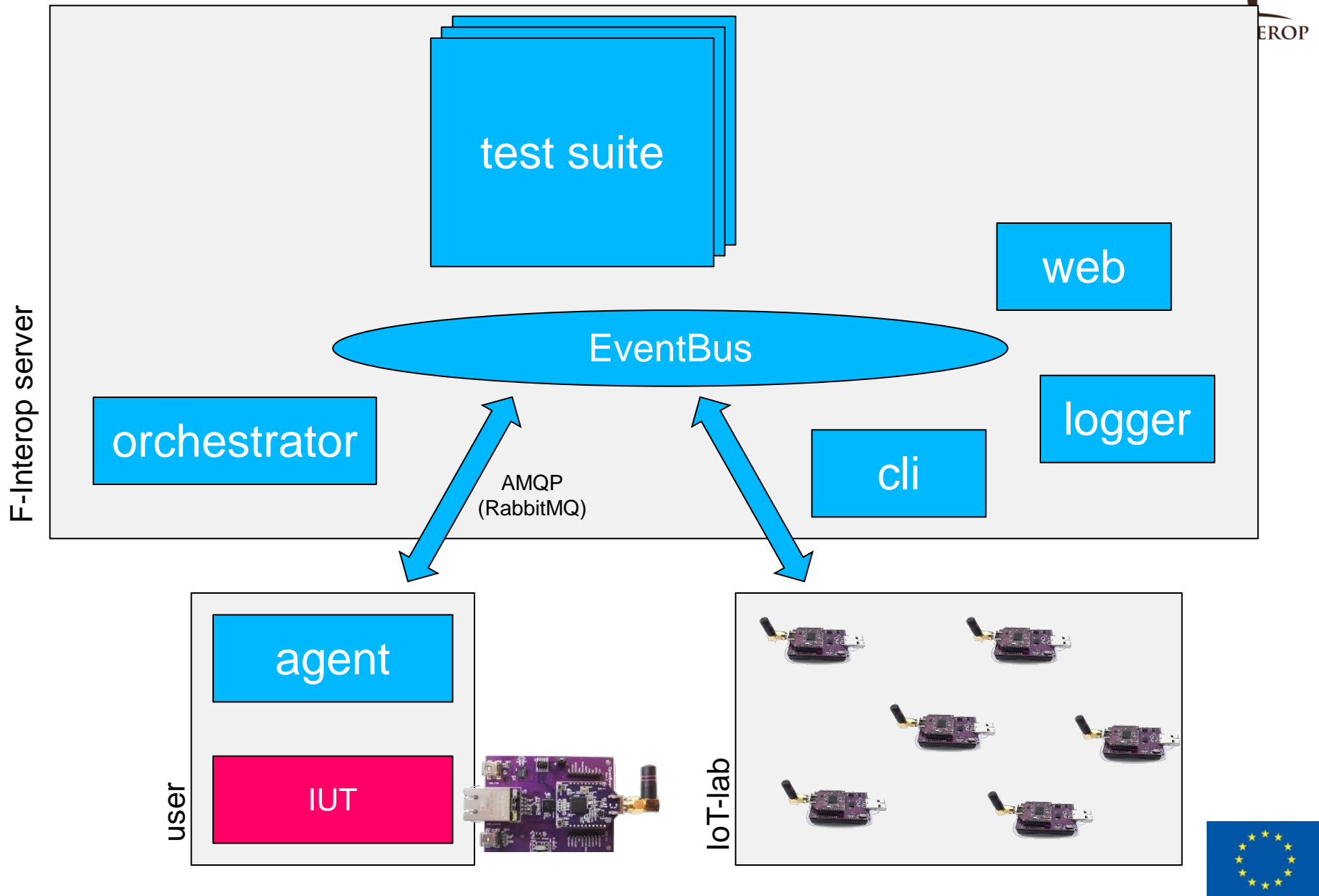
# Base Architecture (CoAP interop)



# Advanced Architecture (6TiSCH example)



# Advanced Architecture (testbed example)



# Advanced Architecture



- Integration into continuous integration
- On a testbed, loading imageA on 50 nodes, imageB on 50 nodes, verify interoperability
- Build virtual networks with sniffers replaying packets heard at userA at userB
- ...



# Next Milestones

- July 2016
  - minimal CoAP testing
- November 2016
  - Functional platform
- March 2017
  - 6TiSCH support, update at IETF98
- July 2017
  - Use at 6TiSCH/6lo plugtests

# What about the IETF?



- Contributors:
  - Develop test suites for (new) standards
  - Provide feedback on architecture and choices
  - List requirements, identify standards
- Users:
  - Use F-Interop for remote interop events





# 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
- **Plugtests** to conduct 3 remote online plugtest events
- **SME device Interop tests** to test F-Interop platform



# Supported Activities & Budget

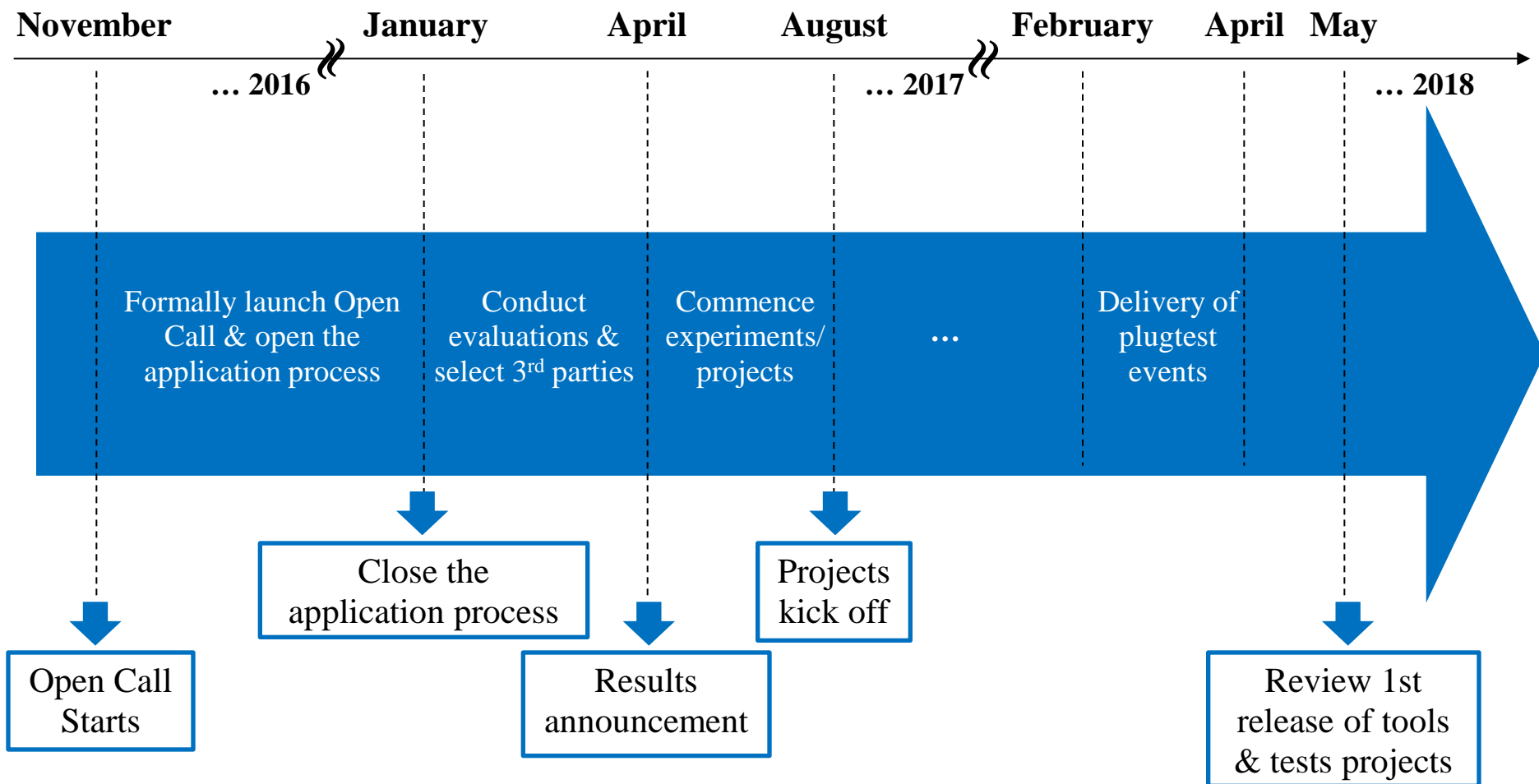


610k for 19 projects

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

**[www.f-interop.eu](http://www.f-interop.eu)**

