

PLATFORM FOR ONLINE  
INTEROPERABILITY AND  
PERFORMANCE TEST



# Remote Conformance & Interop Testing

**TPAC2016 – Web of Things IG Meeting – Lisbon**

**22<sup>nd</sup> September 2016**

**César Viho & Federico Sismondi**

**INRIA - France**



# 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



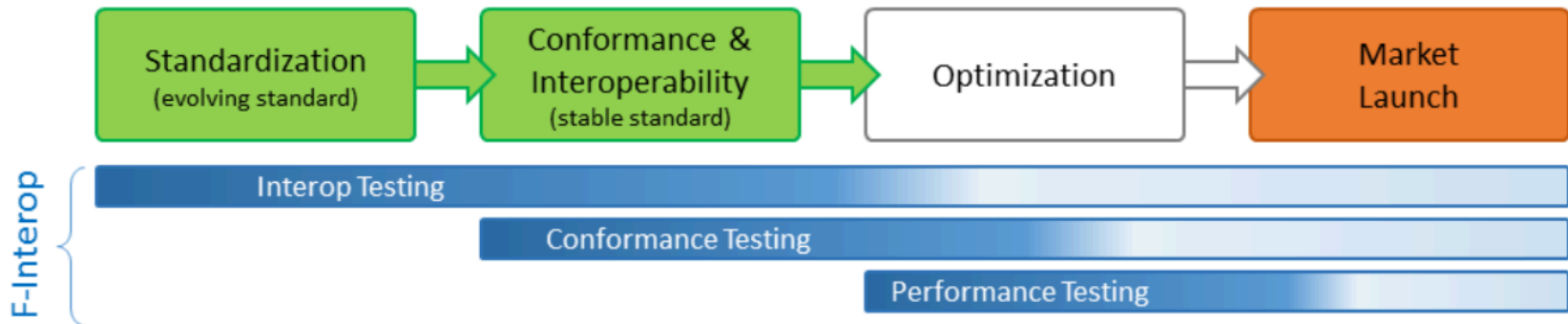
# Goals



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

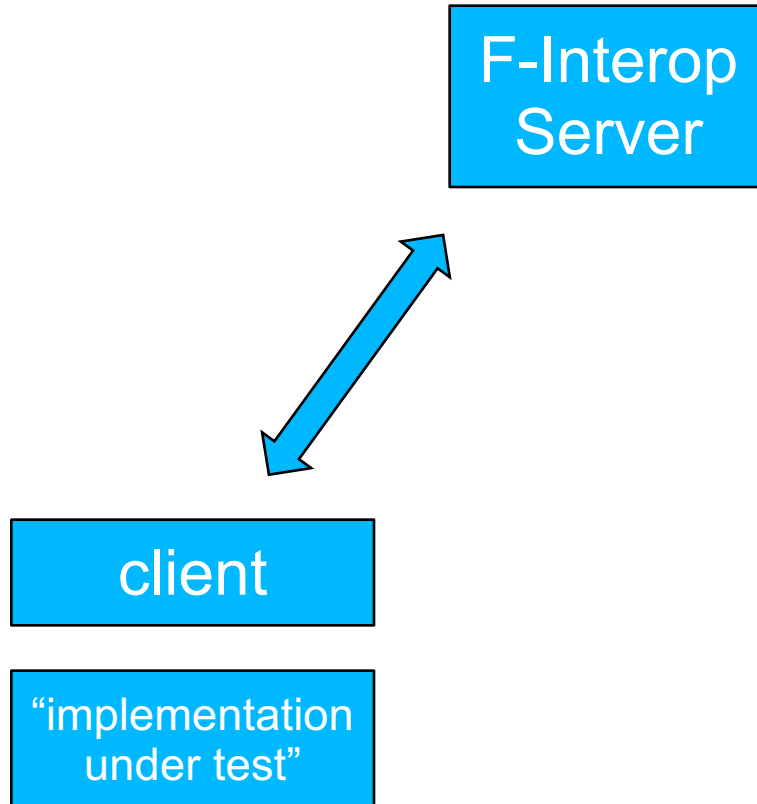
## ➤ SMEs and companies

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

→ more standards-based products



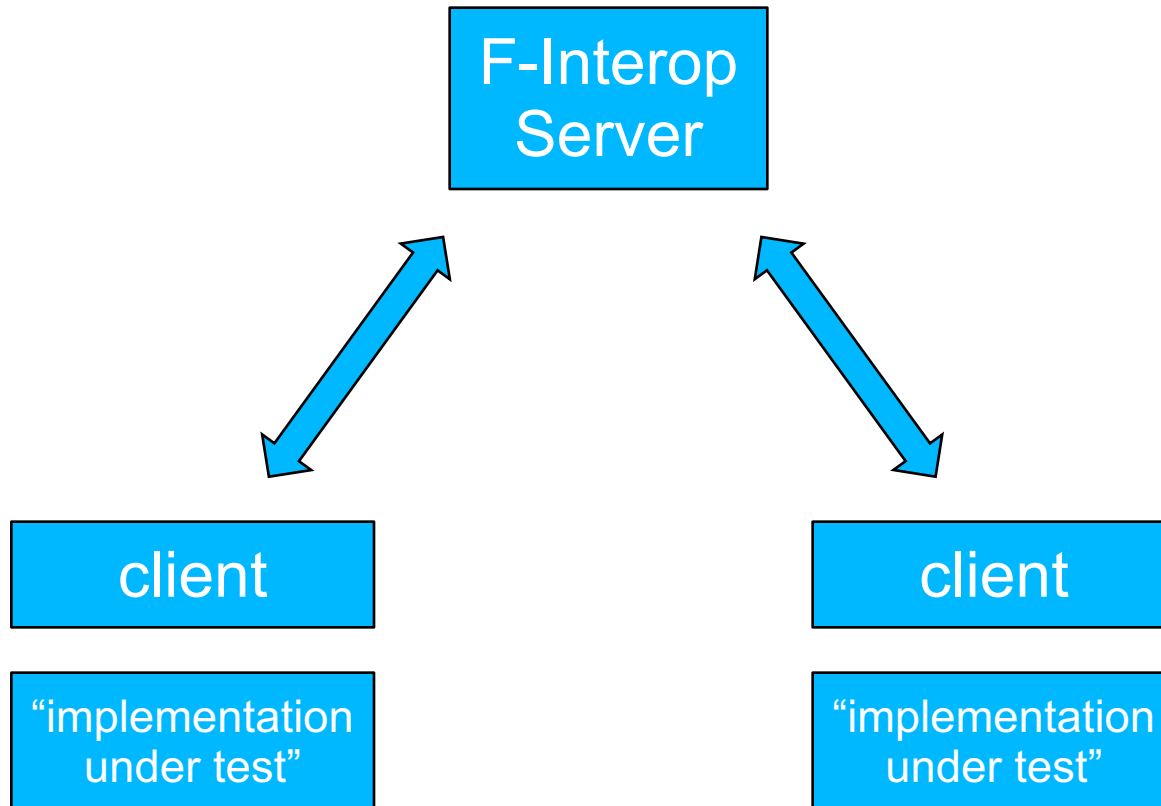
# Core Idea



Conformance Testing



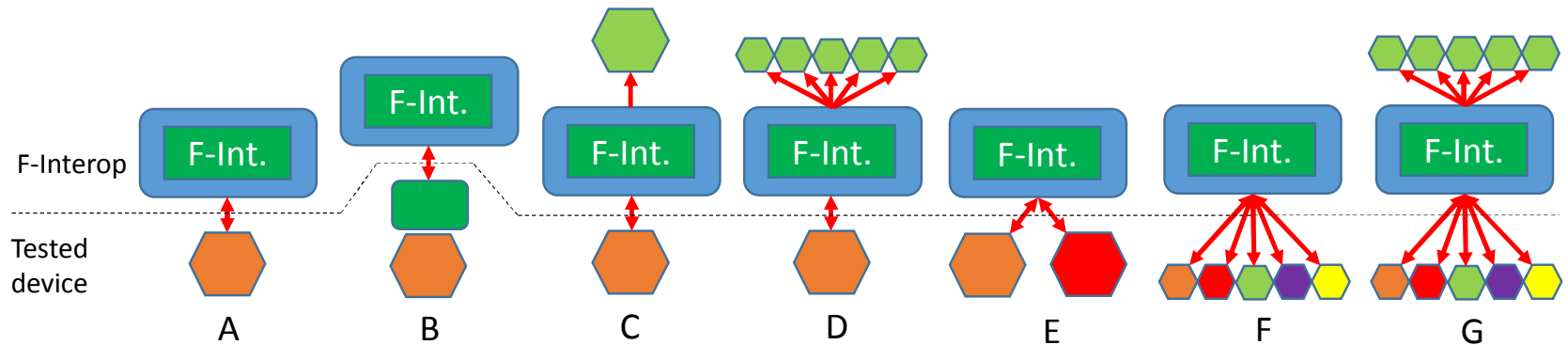
# Core Idea



Interop Testing



# Different Configurations



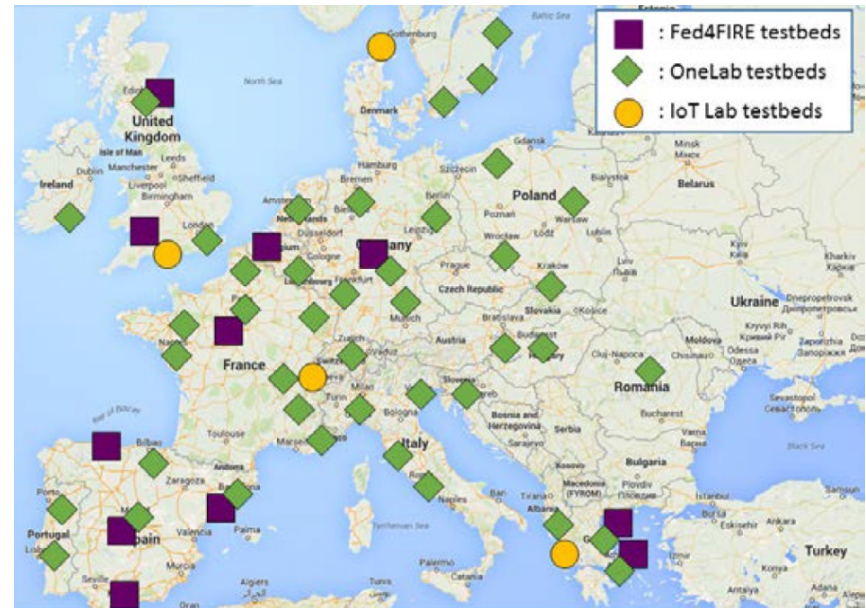
- A. Tested Device  $\leftrightarrow$  F-Interop test server
- B. Deported test with downloaded resource
- C. Remote interop with 2 participants
- D. Interop against testbed
- E. Local interop
- F. Remote interop with N participants
- G. Remote interop with N participants and testbeds

# Testbeds



32 testbeds, 4755 nodes

- **Fed4FIRE**  
([www.fed4fire.eu/testbeds](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
  - 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)

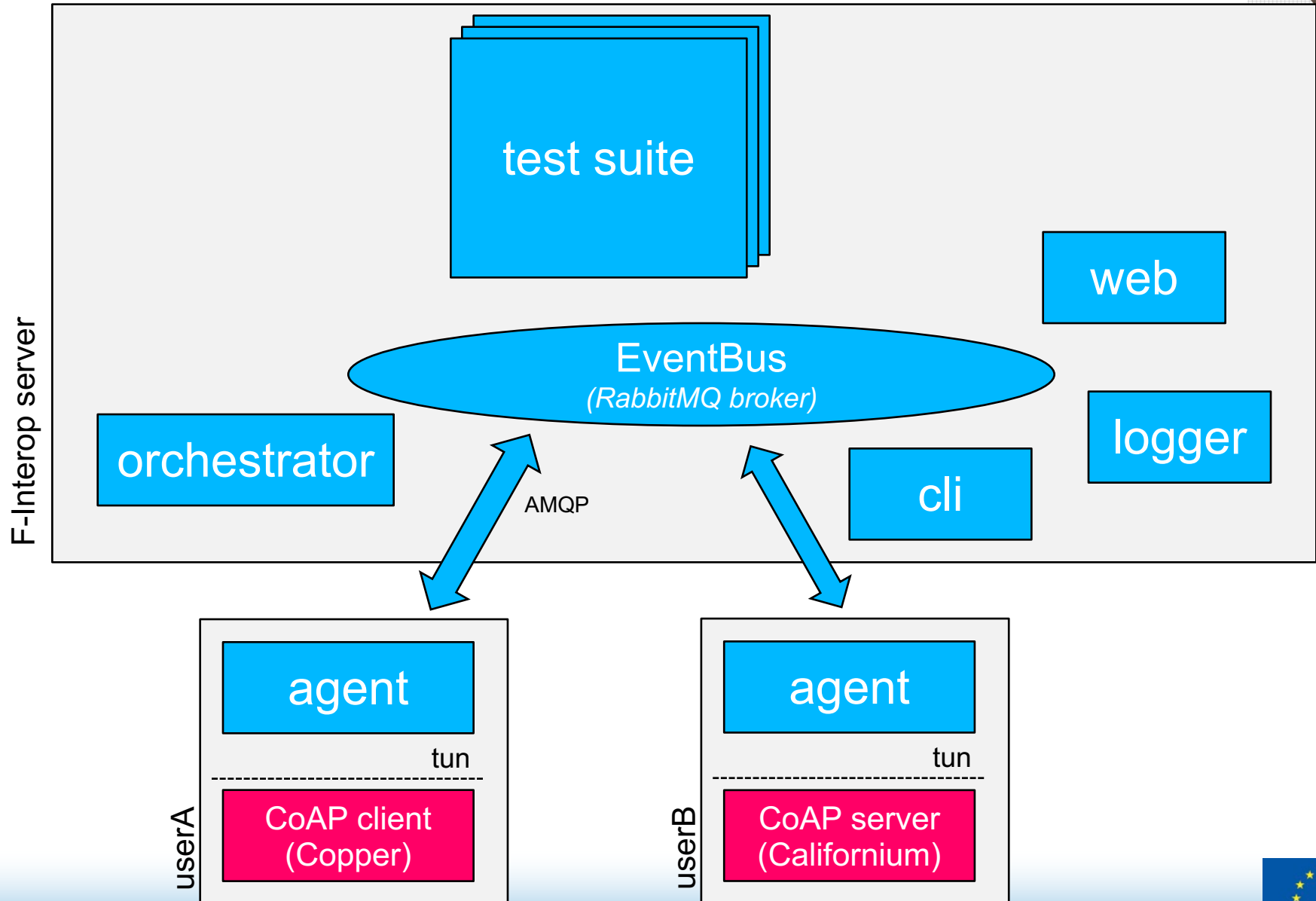
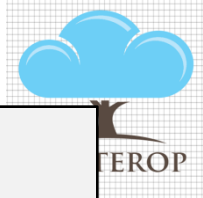
base.html x

file:///C:/Users/Thomas/Desktop/base.html

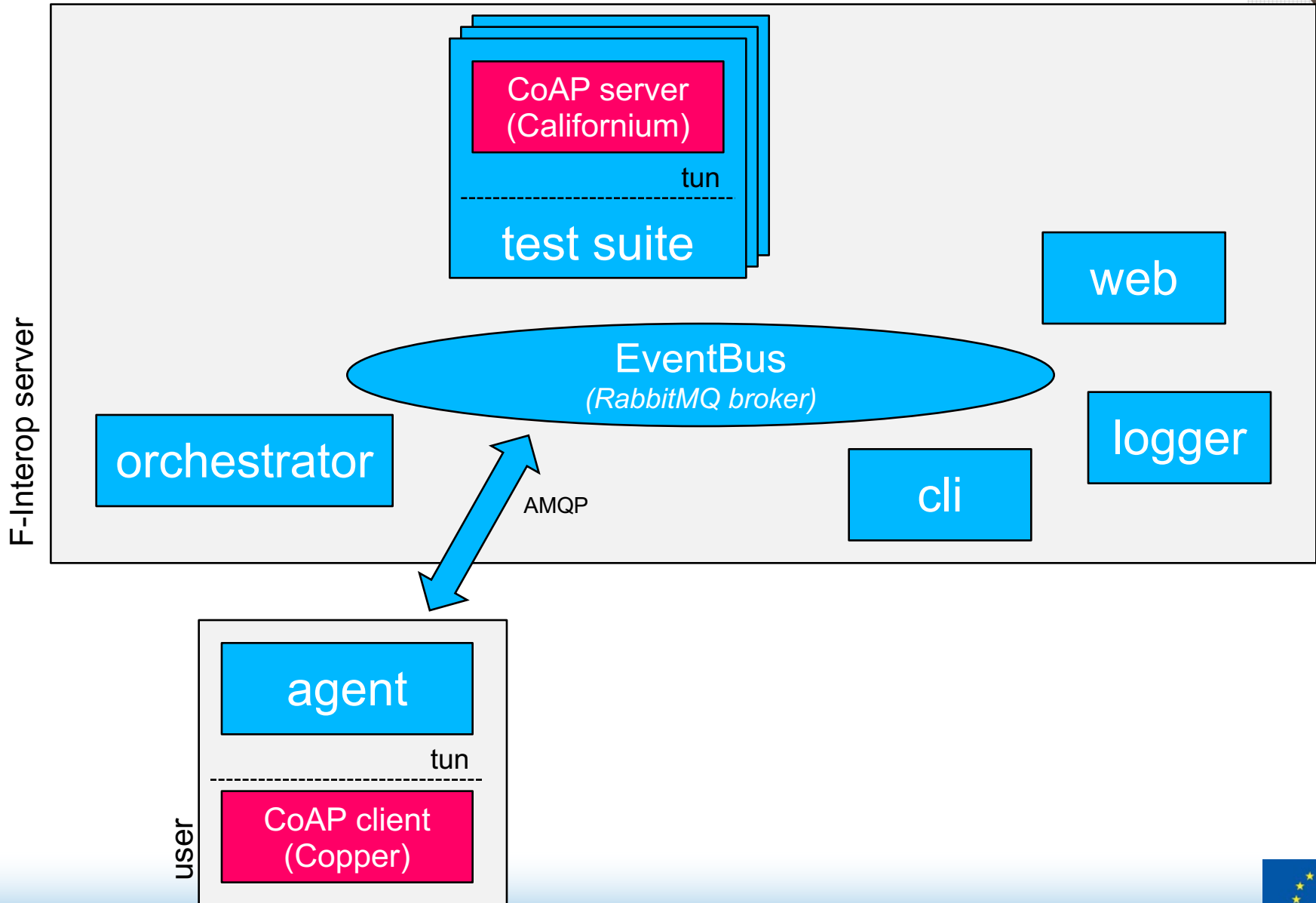
| Interoperability Test Description |  |          |  |
|-----------------------------------|--|----------|--|
| <b>Identifier:</b>                | TD_COAP_CORE_01  |          |  |
| <b>Objective:</b>                 | Perform GET transaction (CON mode)   |          |  |
| <b>Configuration:</b>             | CoAP_CFG_BASIC   |          |  |
| <b>References:</b>                | [COAP] 5.8.1, 1.2, 2.1, 2.2, 3.1   |          |  |
| <b>Pre-test conditions:</b>       | Server offers the resource /test with resource content is not empty that handles GET with an arbitrary payload |          |  |
| <b>Test Sequence:</b>             | 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 interop)



# Base Architecture (CoAP interop demo)



# Download the Agent



**F-interop**  
A platform for interoperability testing

**Home**

- 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

A screenshot of a web browser showing the Finterop client interface. The browser tabs include "Inbox - remy.leone...", "IETF Proceedings", "https://rawgit.com/", "F-Interop\_IETFBerlin", "F-interop", "Finterop client", and "Rémy". The address bar shows "f-interop.paris.inria.fr/session/bonjour/coap". The page header includes the F-INTEROP logo, a logo for "I311" (a yellow cone), and the "Inria" logo. The main content area is divided into three sections: "Test cases", "Console", and "No Frame Selected".  
**Test cases**  
This section is titled "Test case references" and contains a list of three test cases:

- 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**  
This section features a green "Start Test Case" button. Below it, a notification box states "28 test cases loaded". A light blue box displays the "CoAP server URL: coap://[bbbb::2]/test".  
**No Frame Selected**  
This section has a header "No Frame" and a message "No frame selected for the moment". Below it, a "Frame list" section also displays "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

[bbbb::2]:5683

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

| Value               | Option         |
|---------------------|----------------|
| T... Acknowledgment | Content-F... 0 |
| C... 2.05 Content   | Max-Age ...    |
| ... 63915           |                |
| T... empty          |                |

**Payload (38)**

Incoming  Rendered  Outgoing

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



# 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

F-INTEROP Inria

## 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'bxda')

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'bxda'  
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?



```
--- !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
  iut : 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
  iut: 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

|                      |  |
|----------------------|--|
| <b>Identifier</b>    | <b>TC_WOT_BASE_01</b>  |
| Objective            | Read Boolean Property  |
| References           | <a href="#">3.2.3.1 Property</a> , <a href="#">3.2.4.1 Simple Data</a>   |
| Pre-test conditions  | Exposing Thing provides boolean Property   |
| <b>Test sequence</b> |  |
| 1. Stimulus          | Consuming Thing sends <code>Retrieve</code> to Property  |
| 2. Check             | Consuming Thing sends <ul style="list-style-type: none"><li>- protocol operation bound to <code>Retrieve</code></li><li>- no payload</li><li>- to Property URI</li></ul> |
| 3. Check             | Exposing Thing sends <ul style="list-style-type: none"><li>- positive response code</li><li>- payload formatted according to TD</li></ul>                                |
| 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

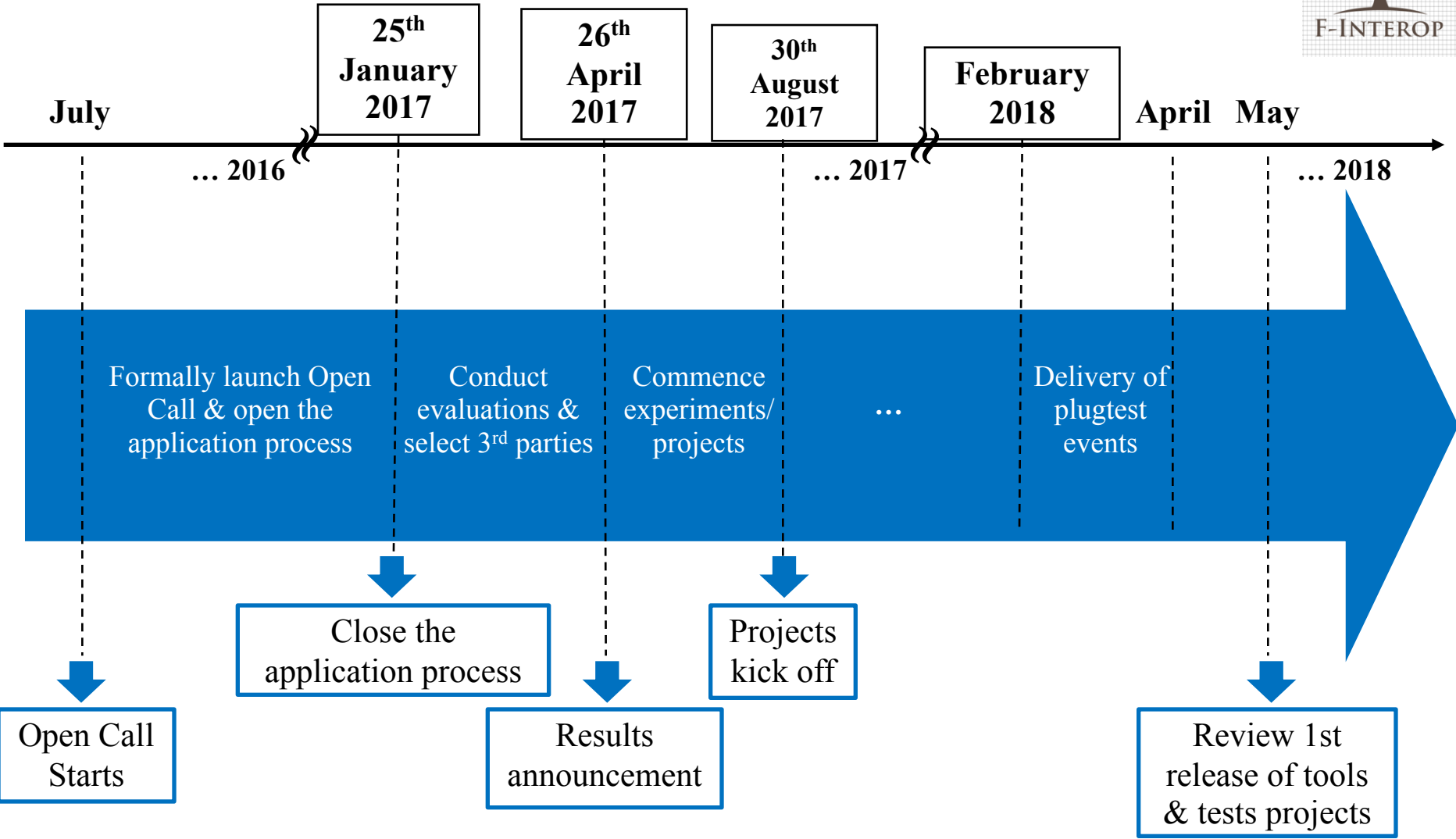


610k for 19 projects

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

