

F-Interop Platform

Remote Conformance & Interop Testing

**OMA Device Management (DM)
And Interoperability (IOP) Working Group**
Thomas Watteyne, Remy Leone

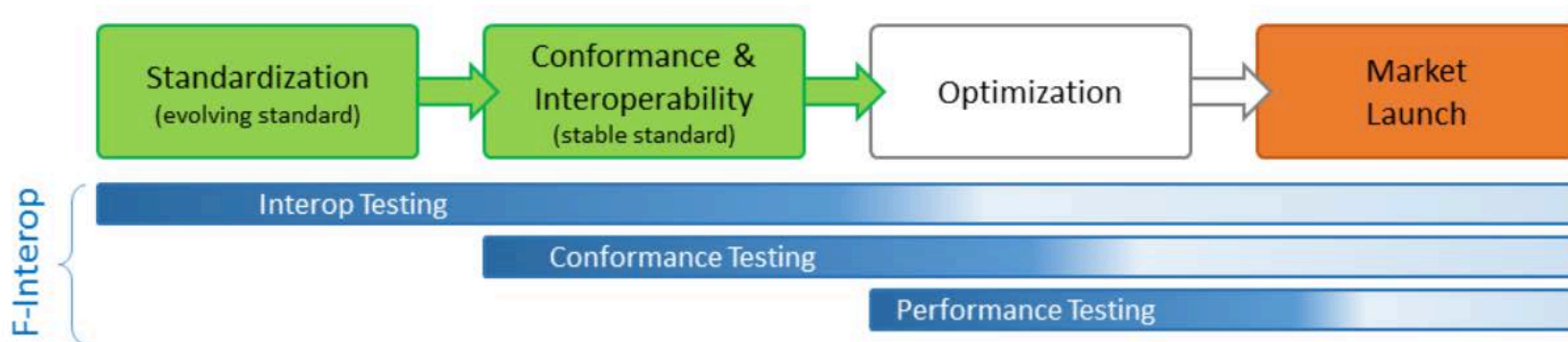
Goals



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



Why remote conformance & interop?



➤ SDOs

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

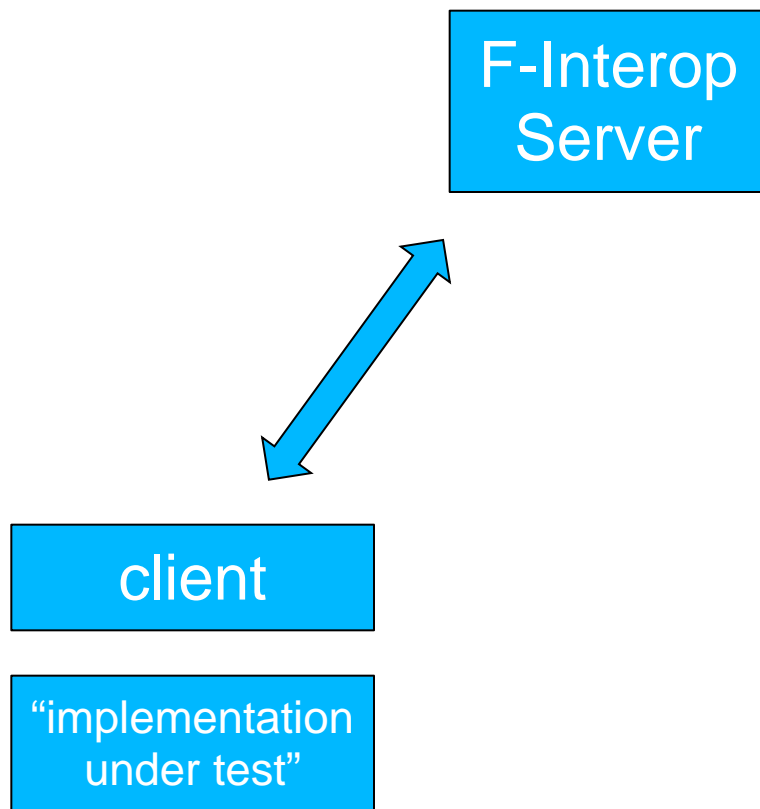
➤ SMEs and companies

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

→ more standards-based products



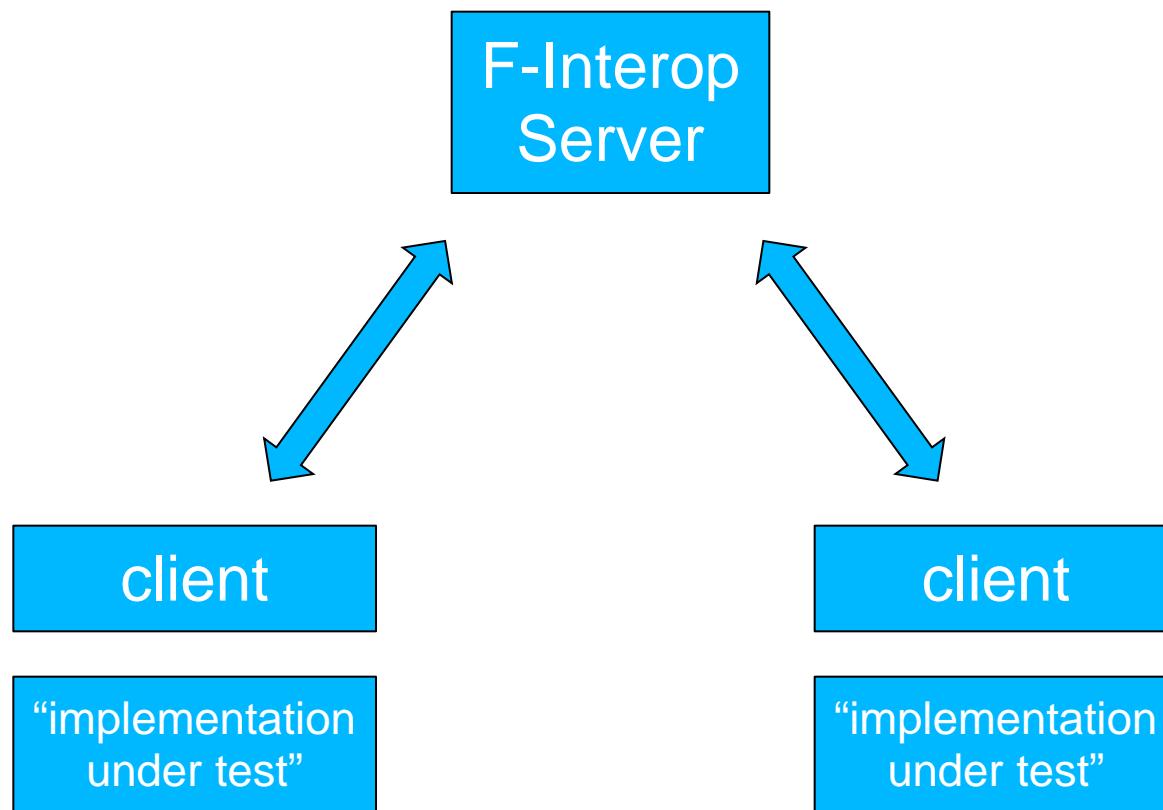
Core Idea



Conformance Testing



Core Idea



Interop Testing



Different Configurations



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



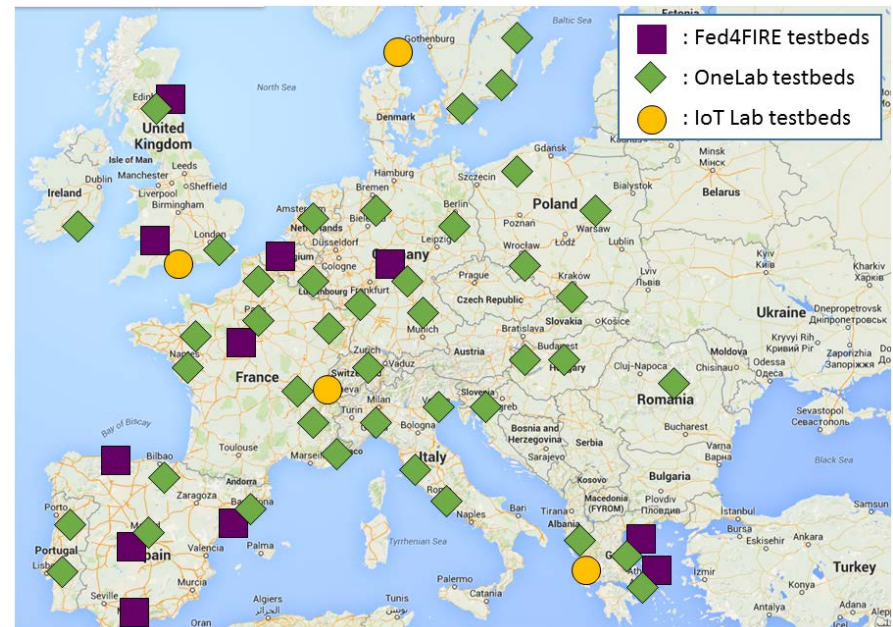
- www.f-interop.eu
- 1 November 2015 – 31 October 2018
- *develop and provide online interoperability and performance test tools to support emerging technologies from research to standardization and market launch*
- 9 partners



Testbeds

32 testbeds, 4755 nodes

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



Example CoAP Test

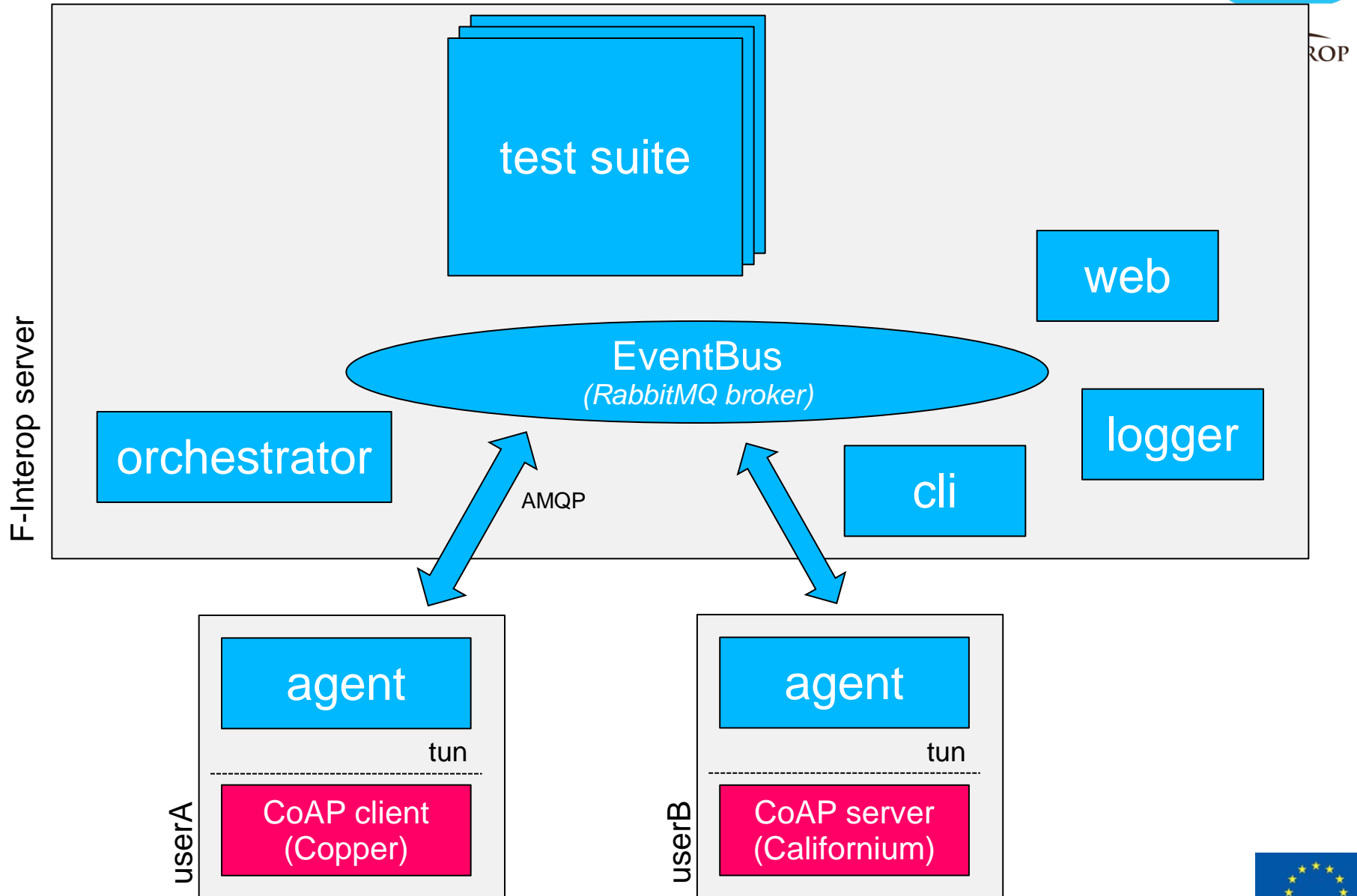


- From ETSI plugtest CoAP#4, IETF89 (London)

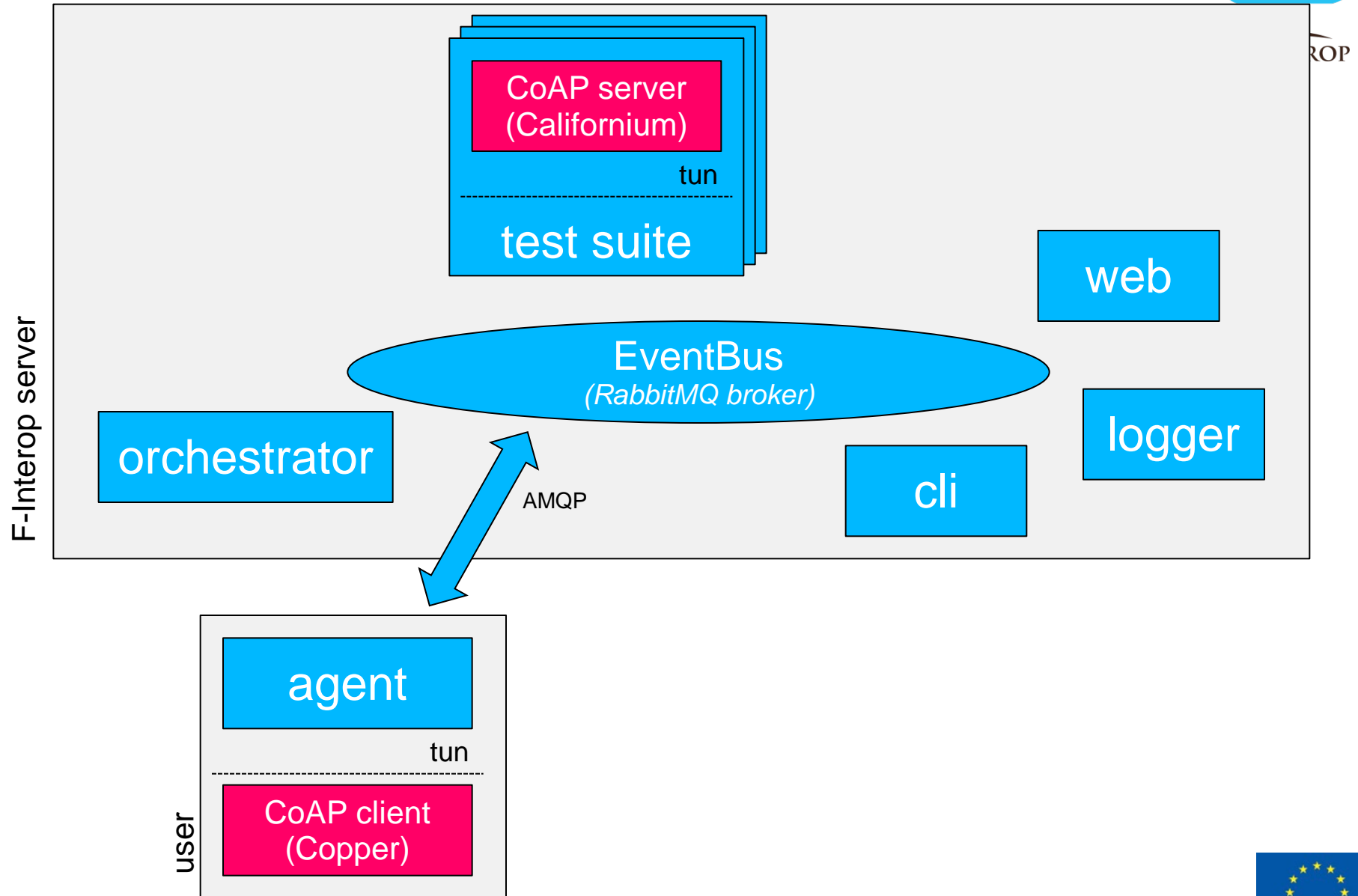
Interoperability Test Description		
Identifier:	TD_COAP_CORE_01	
Objective:	Perform GET transaction (CON mode)	
Configuration:	CoAP_CFG_BASIC	
References:	[COAP] 5.8.1, 1.2, 2.1, 2.2, 3.1	
Pre-test conditions:	Server offers the resource /test with resource content is not empty that handles GET with an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



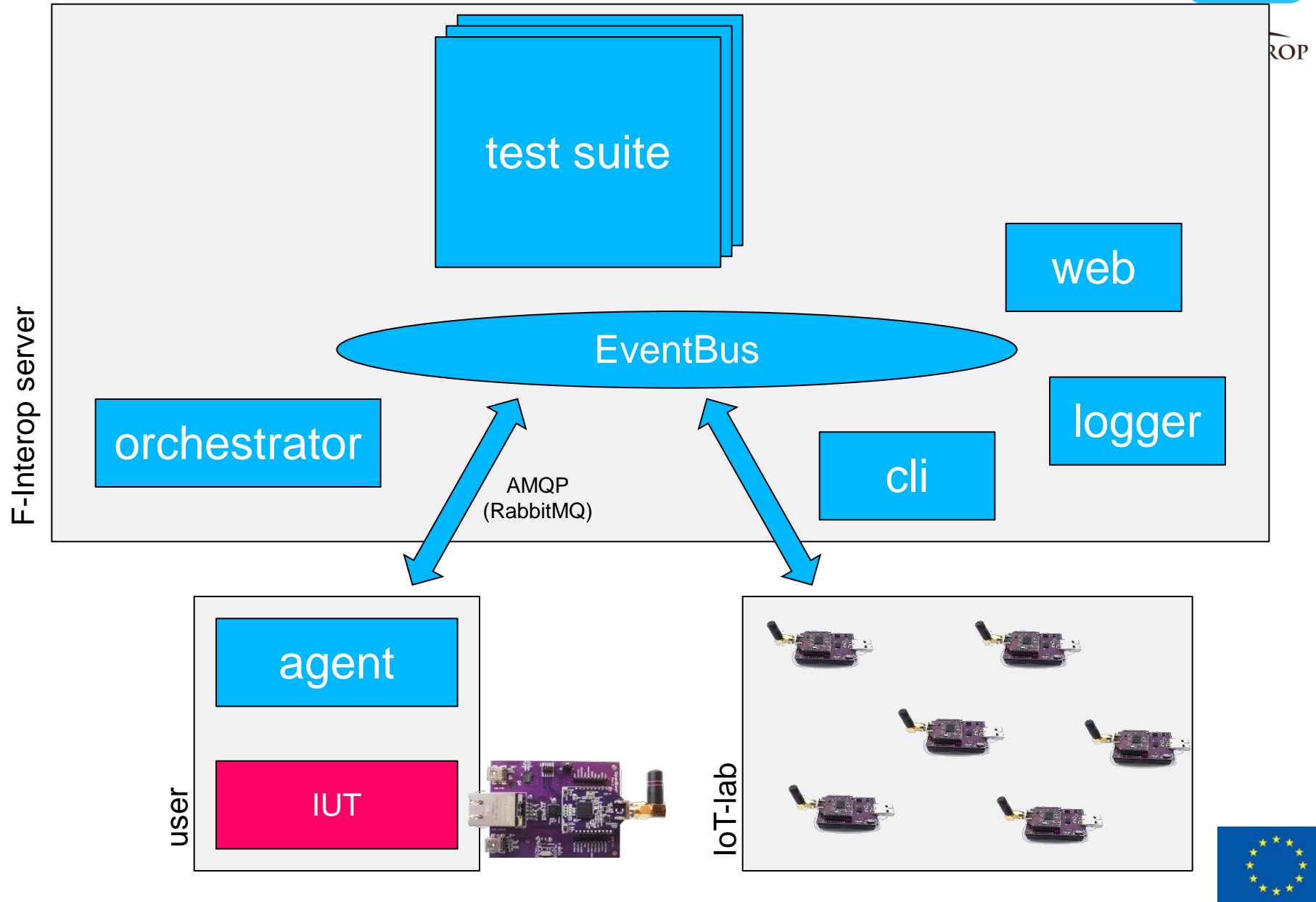
Base Architecture (CoAP interop)



Base Architecture (CoAP interop demo)



Advanced Architecture (testbed example)



CoAP 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



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

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 **Reset**

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

F-INTEROP

Idix 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'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")
```

Under the Hood: Interface Docs



The screenshot displays the F-INTEROP API documentation website. The left sidebar contains a search bar and a navigation menu with the following items: Introduction, Event Bus (MOM layer), Components description, Messaging pattern types used in..., Examples, Application Layer Messages, Orchestrator, Test Coordinator, Sniffer, Test Analyser (TAT), Protocol Dissector, Agent, Packet Generator, Result repository, Graphical interface and other ..., and Other common event. The main content area is titled 'doc.f-interop.eu/#messaging-pattern-types-used-in-f-interop' and features a diagram illustrating the interaction between 'component 1' and 'component 2' via an 'event bus'. The diagram shows two message flows: one from component 1 to the event bus and another from the event bus to component 2. Each message flow includes a 'topic' and a 'token'. Below the diagram, the 'Examples' section is marked as 'TBD' (To Be Determined). The 'Application Layer Messages' section includes a 'Message Header' table with the following fields: Field, Type, and Description. The table lists '_type' as a string, describing the message's purpose. A note below the table states: 'TBD: consensus on: every piece of information inside content must be an object that has an attribute _type for object self describing purposes?'. On the right side of the page, there is a 'Json Example' section with a code block showing a JSON object for a service start sniffer.

```
{
  "routing_key": "control.sniffing.service.start",
  "src_id": "amqp://coordinator@finterop.org",
  "msg_id": "9b70cd6b-4a2e-45f7-bb28-12cfd9788f16",
  "token": "9b70cd6b-4a2e-45f7-bb28-12cfd9788f16",
  "contents": [...]
}
```

doc.f-interop.eu



Next Milestones



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



Thoughts about using F-Interop for testing OMA L2M2M

Using F-Interop for OMA LWM2M?



Goal:

- Automating SCR?
- Is there an interoperability TD?

Thoughts:

- Application-level protocol greatly simplifies testing
 - No tight timing requirements
 - Remote testing ideal
 - CoAP supporting tools (e.g. Wireshark, [TLV support](#))
- 4 clear interfaces (bootstrap, registration, management, data)
- Reference LWM2M client and/or server?
- Using F-Interop as “cloud version” of existing test suite possible

OMA-TS-LightweightM2M-V1_0-20160407-C

Page 86 (127)

Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCR.RULES].

B.1 SCR for LWM2M Client

B.1.1 Bootstrap Interface

Item	Function	Reference	Requirement
LWM2M-BOOT-001-C-M	Support of at least one Bootstrap Mode	Section 5.1	
LWM2M-BOOT-002-C-O	Support of Factory Bootstrap Mode	Section 5.2.2.1	
LWM2M-BOOT-003-C-O	Support of Bootstrap from Smartcard	Section 5.2.2.2, Appendix F	LWM2M-BOOT-012C-O
LWM2M-BOOT-004-C-O	Support of Client Initiated Bootstrap	Section 5.2.2.3	
LWM2M-BOOT-005-C-O	Support of Server Initiated Bootstrap	Section 5.2.2.4	
LWM2M-BOOT-006-C-M	Support of LWM2M Server Bootstrap Information	Section 5.2.1	
LWM2M-BOOT-007-C-O	Support of LWM2M Bootstrap Server Bootstrap Information	Section 5.2.1	
LWM2M-BOOT-008-C-M	Support of accepting Bootstrap Information transferred	Section 5.2.1	
LWM2M-BOOT-009-C-M	Support of Bootstrap Sequence	Section 5.2.3	
LWM2M-BOOT-010-C-M	Support of Bootstrap Security	Section 5.2.4	
LWM2M-BOOT-011-C-O	Support of Bootstrap from Smartcard with Secure Channel	Section 5.2.2.2, Appendix F	LWM2M-BOOT-012C-O AND LWM2M-SEC-007-C-O
LWM2M-BOOT-012-C-O	Retrieve & Process bootstrap data from Smartcard	Section 5.2.2.2	
LWM2M-BOOT-013-C-O	Check for Bootstrap Data change in Smartcard	Section 5.2.2.2	

B.1.2 Client Registration

Item	Function	Reference	Requirement
LWM2M-CR-001-C-M	Support of “Register” operation	Section 5.3.1	
LWM2M-CR-002-C-M	Support of Endpoint Client Name parameter	Section 5.3.1	
LWM2M-CR-003-C-M	Support of Lifetime parameter	Section 5.3.1	
LWM2M-CR-004-C-O	Support of LWM2M Version parameter	Section 5.3.1	

© 2016 Open Mobile Alliance Ltd. All Rights Reserved.
Used with the permission of the Open Mobile Alliance Ltd. under the terms as stated in this document.

[OMA-Template-Spec-20160101-0]



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

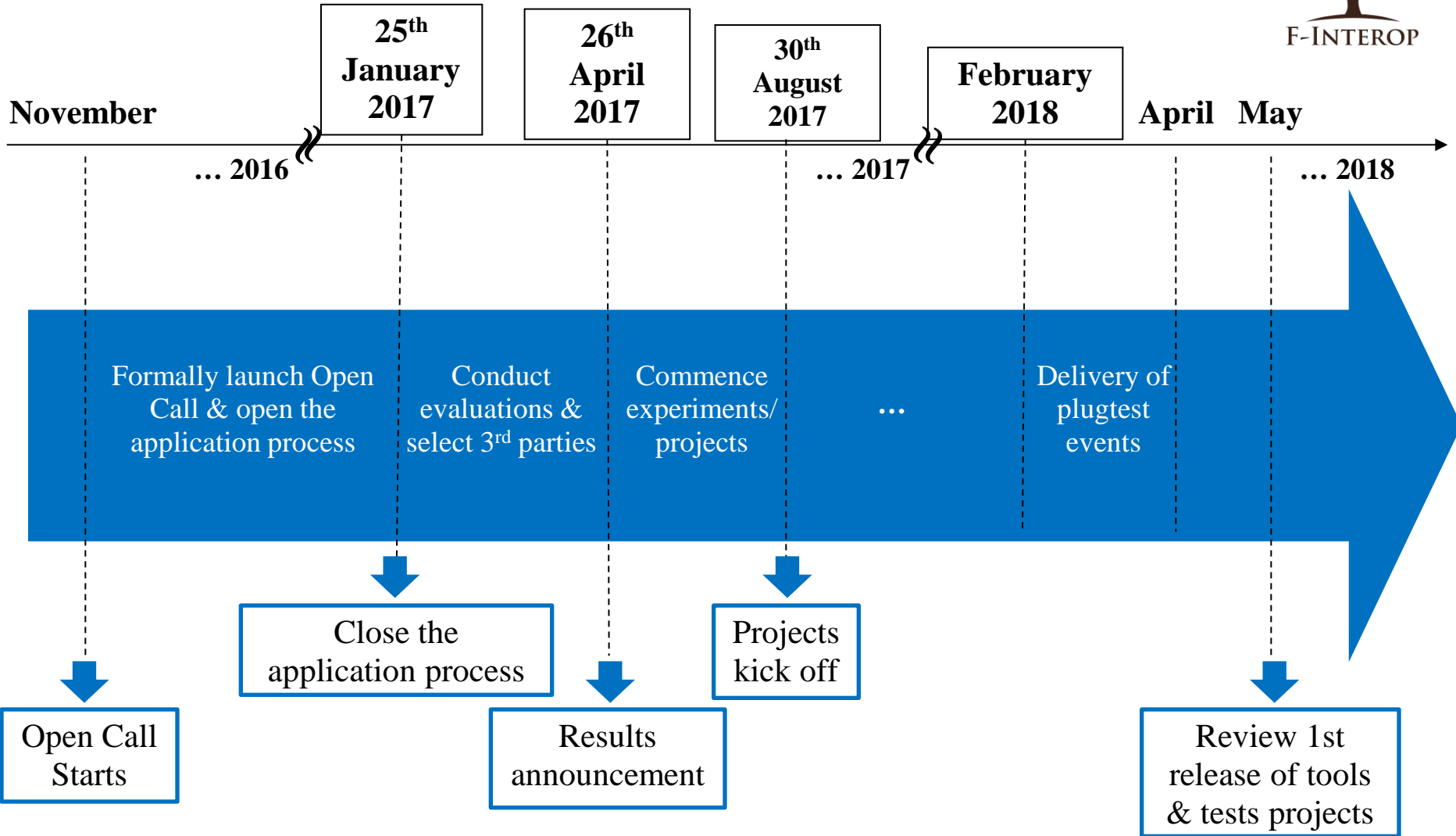


610k for 19 projects

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>





F-INTEROP



Thomas Watteyne



Thomas Watteyne (<http://www.thomaswatteyne.com/>, [@thomaswatteyne](#)) is an insatiable enthusiast of low-power wireless mesh technologies. He is a researcher at Inria in Paris, in the new EVA research team, where he designs, models and builds networking solutions based on a variety of Internet-of-Things (IoT) standards. He is Senior Networking Design Engineer at Linear Technology, in the Dust Networks product group, the undisputed leader in supplying low power wireless mesh networks for demanding industrial process automation applications. Since 2013, he co-chairs the IETF 6TiSCH working group, which standardizes how to use IEEE802.15.4e TSCH in IPv6-enabled mesh networks, and recently joined the IETF Internet-of-Things Directorate. Prior to that, Thomas was a postdoctoral research lead in Prof. Kristofer Pister's team at the University of California, Berkeley. He founded and co-leads Berkeley's OpenWSN project, an open-source initiative to promote the use of fully standards-based protocol stacks for the IoT. Between 2005 and 2008, he was a research engineer at France Telecom, Orange Labs. He holds a PhD in Computer Science (2008), an MSc in Networking (2005) and an MEng in Telecommunications (2005) from INSA Lyon, France. He is Senior member of IEEE. He is fluent in 4 languages.

