

**PLATFORM FOR ONLINE  
INTEROPERABILITY AND  
PERFORMANCE TEST**

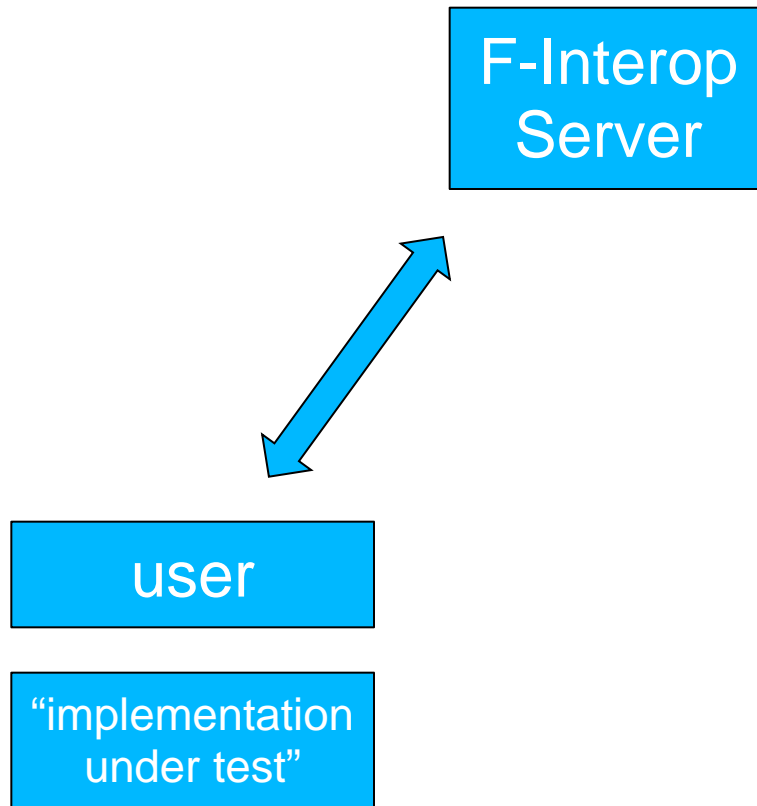


# Technical Overview of F-interop

Rémy Leone



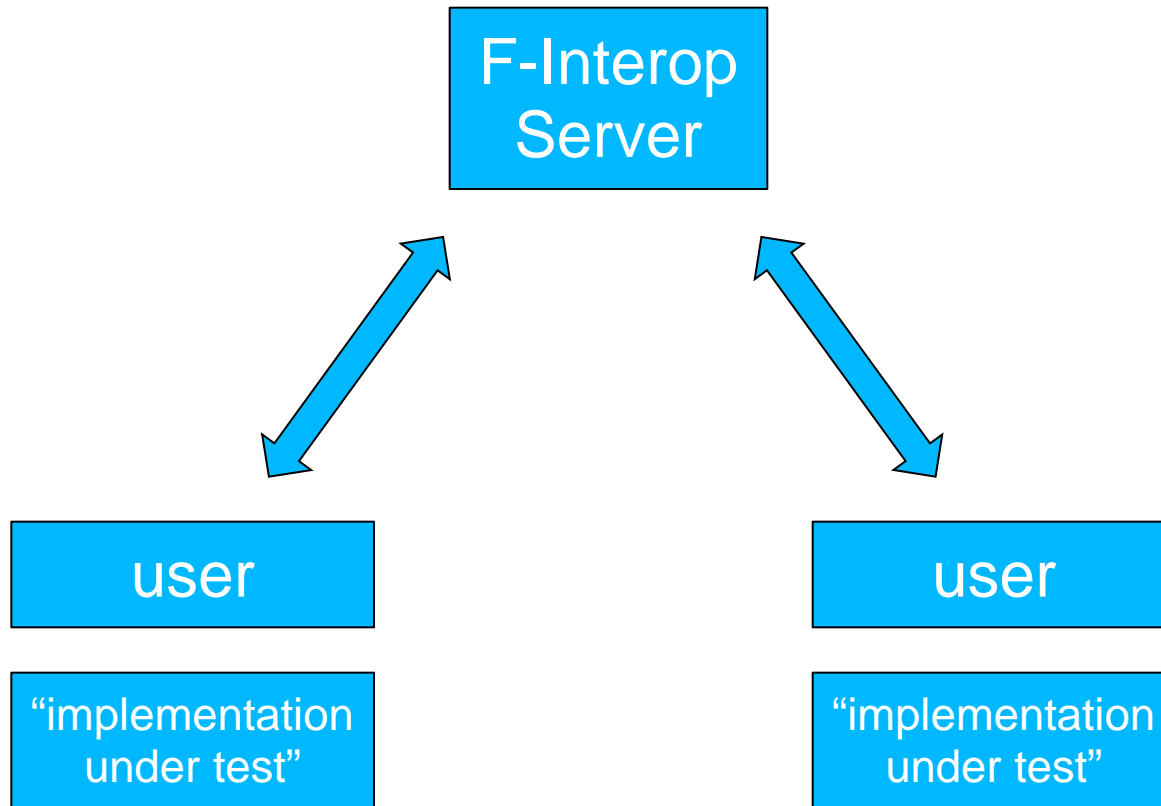
# Core Idea



Conformance Testing



# Core Idea



Interop Testing



# State of the art



- Plugtests organized face to face
  - Error-prone
  - Expensive
  - Long-time before testing again



# Requirements for remote interop tests

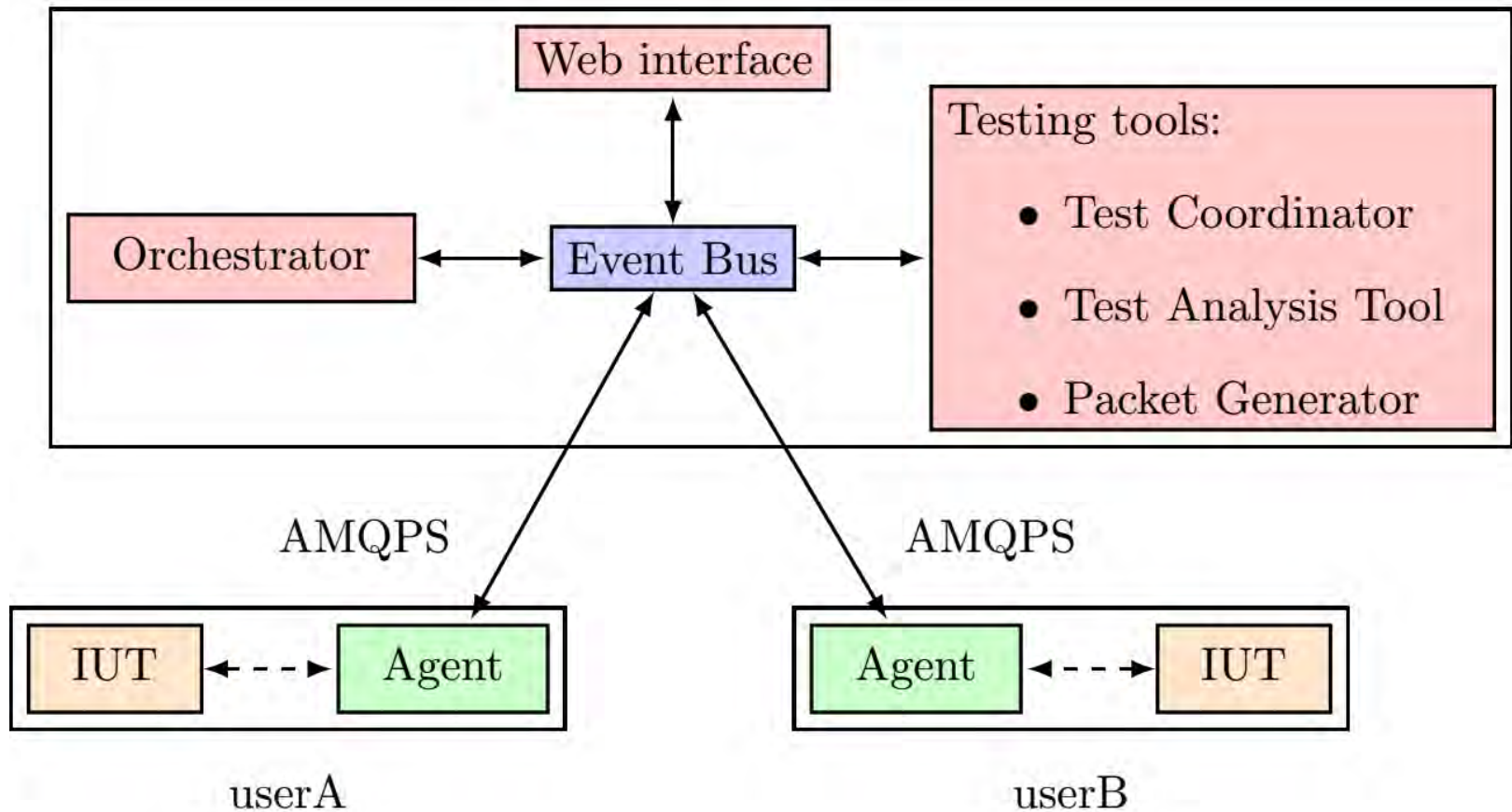


- Modular by design
  - Message-oriented architecture
  - Allows for easy scaling, features extensions, ...
- Observers as close as possible of the devices
  - Network packets, radio sniffing, precise timestamping
- Easy to use interface
  - Results debugging, archiving, ...
- Integrated with Testbeds



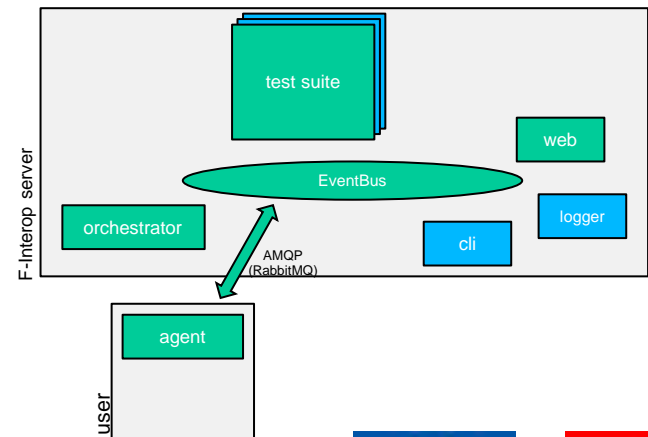
# F-interop Architecture Overview

F-Interop server



# Implementation

- AMQP (RabbitMQ)-based **event-bus core**
- Python-based **agent** code allowing a user to connect
- ttpROTO-based initial **web interface** to select/control test execution
- OpenWSN-based integration with low-power wireless **devices**
- Initial test suites
  - ttpROTO-based **CoAP testing**
  - [WIP] first **6TiSCH testing**



**PLATFORM FOR ONLINE  
INTEROPERABILITY AND  
PERFORMANCE TEST**



# Demo

*Complete run-through of a user  
executing a CoAP test*

Rémy Léone, Federico Sismondi (*Inria*)





# Example CoAP Test: TD\_COAP\_CORE\_01



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

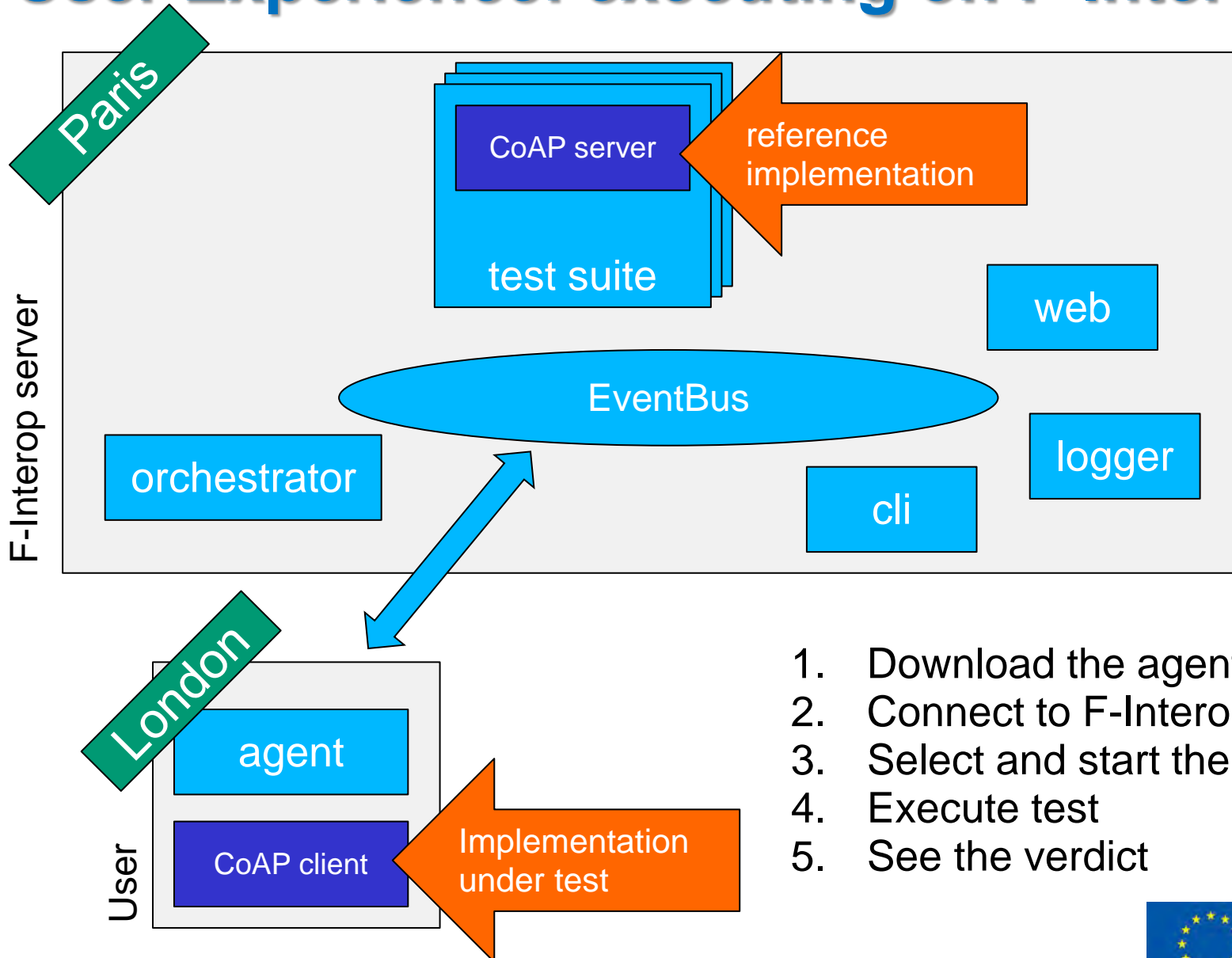
base.html x

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



# User Experience: executing on F-Interop



1. Download the agent
2. Connect to F-Interop server
3. Select and start the test case
4. Execute test
5. See the verdict



# 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*



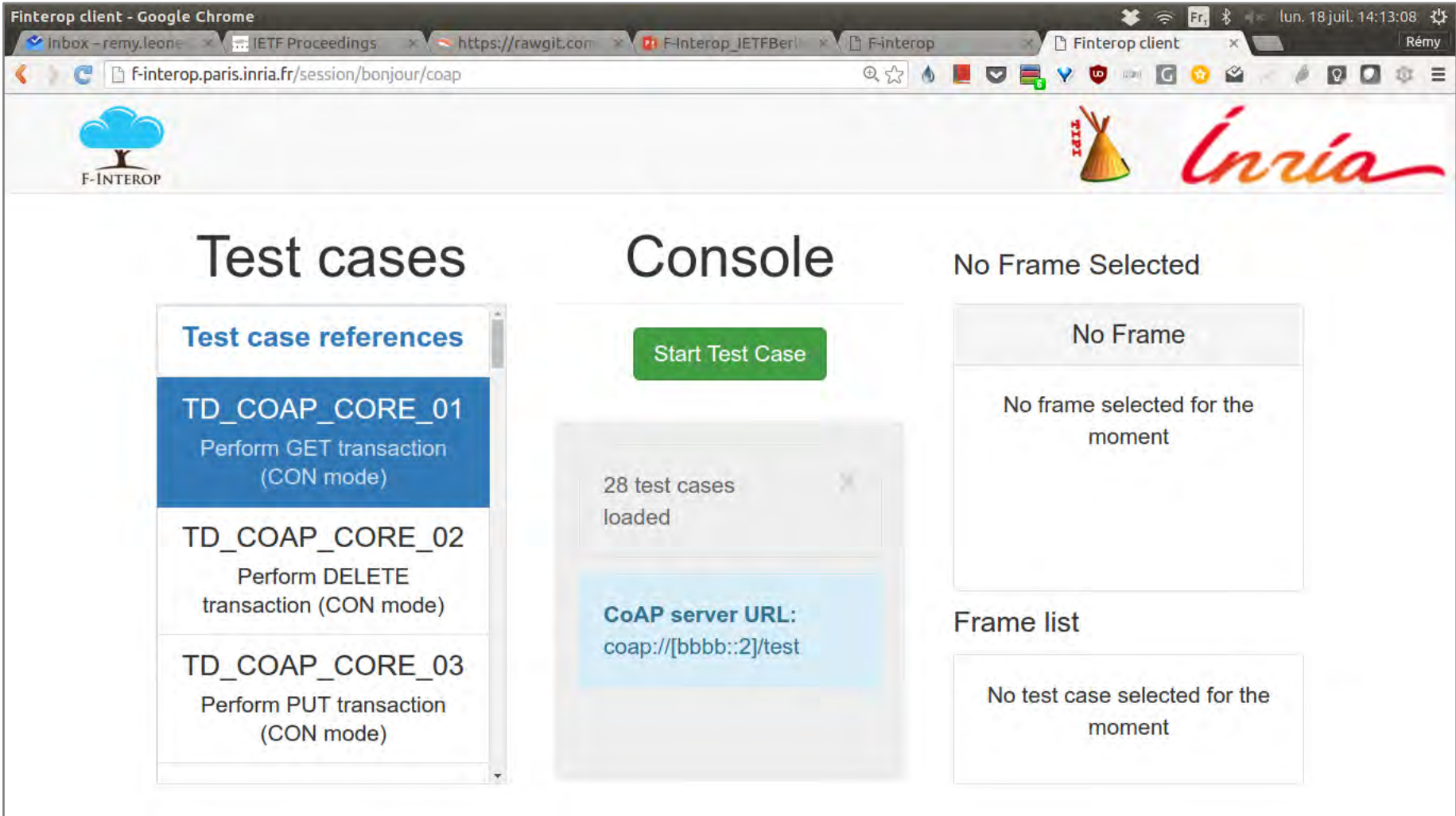
# Connect to F-Interop Server



```
# 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



The screenshot shows the Finterop client web application in a Google Chrome browser. The browser tabs include 'Inbox - remy.leone', 'IETF Proceedings', 'https://rawgit.com', 'F-Interop\_JETFBerl', 'F-interop', and 'Finterop client'. The address bar shows the URL 'f-interop.paris.inria.fr/session/bonjour/coap'. The page header features the F-INTEROP logo on the left and the Inria logo on the right. The main content area is divided into three panels:

- Test cases:** A list of test case references is shown, with 'TD\_COAP\_CORE\_01 Perform GET transaction (CON mode)' selected and highlighted in blue. Other visible items include 'TD\_COAP\_CORE\_02 Perform DELETE transaction (CON mode)' and 'TD\_COAP\_CORE\_03 Perform PUT transaction (CON mode)'.
- Console:** A green 'Start Test Case' button is visible at the top. Below it, a message box indicates '28 test cases loaded'. A light blue box displays the 'CoAP server URL: coap://[bbbb::2]/test'.
- No Frame Selected:** This panel shows 'No Frame' and 'No frame selected for the moment'. Below this, a 'Frame list' section shows '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	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
```

- [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/... F-Interop\_JETFBerl... F-interop... Finterop client

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

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

## 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





F-INTEROP

**Thank you**



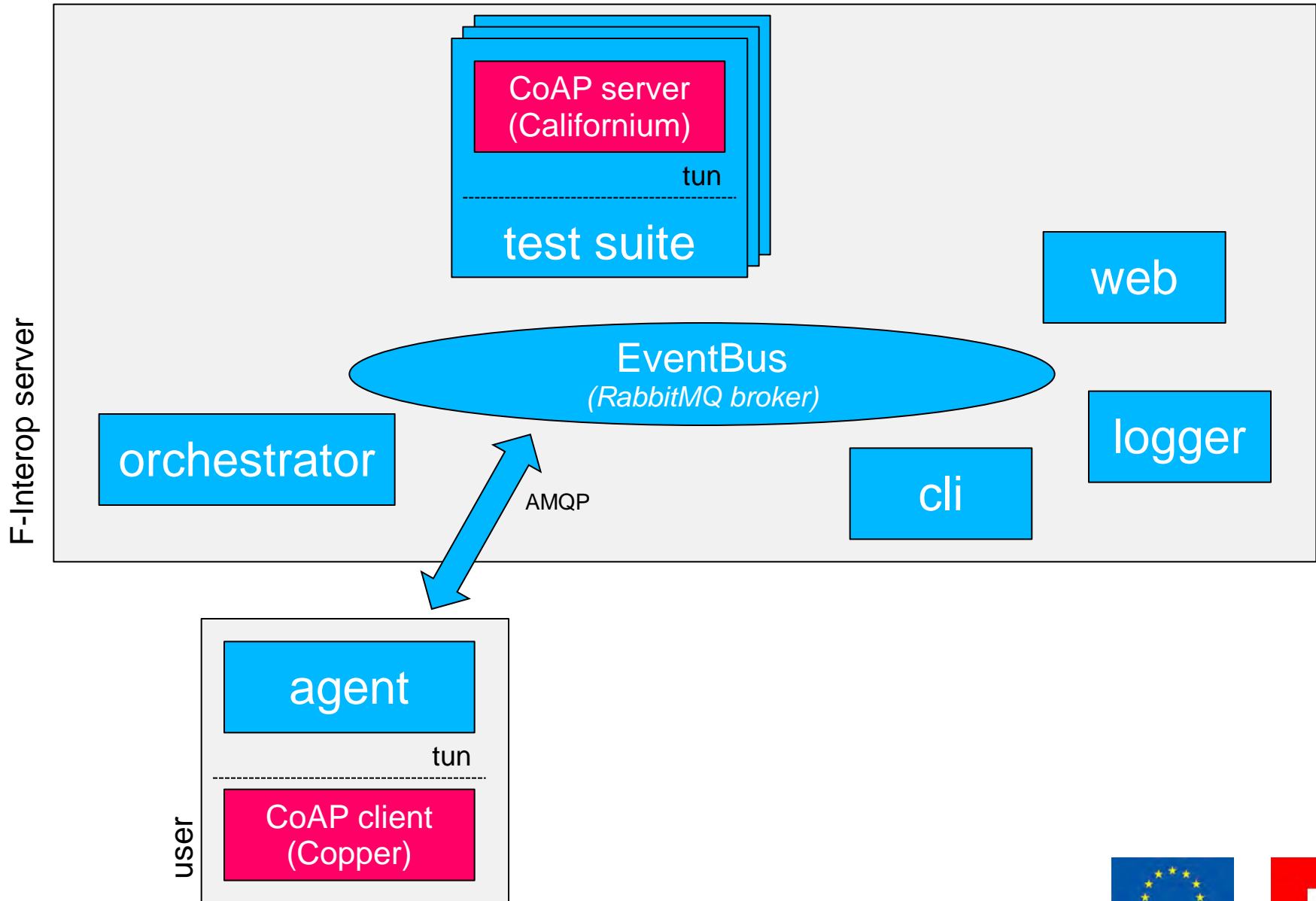
# Different Configurations



- 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



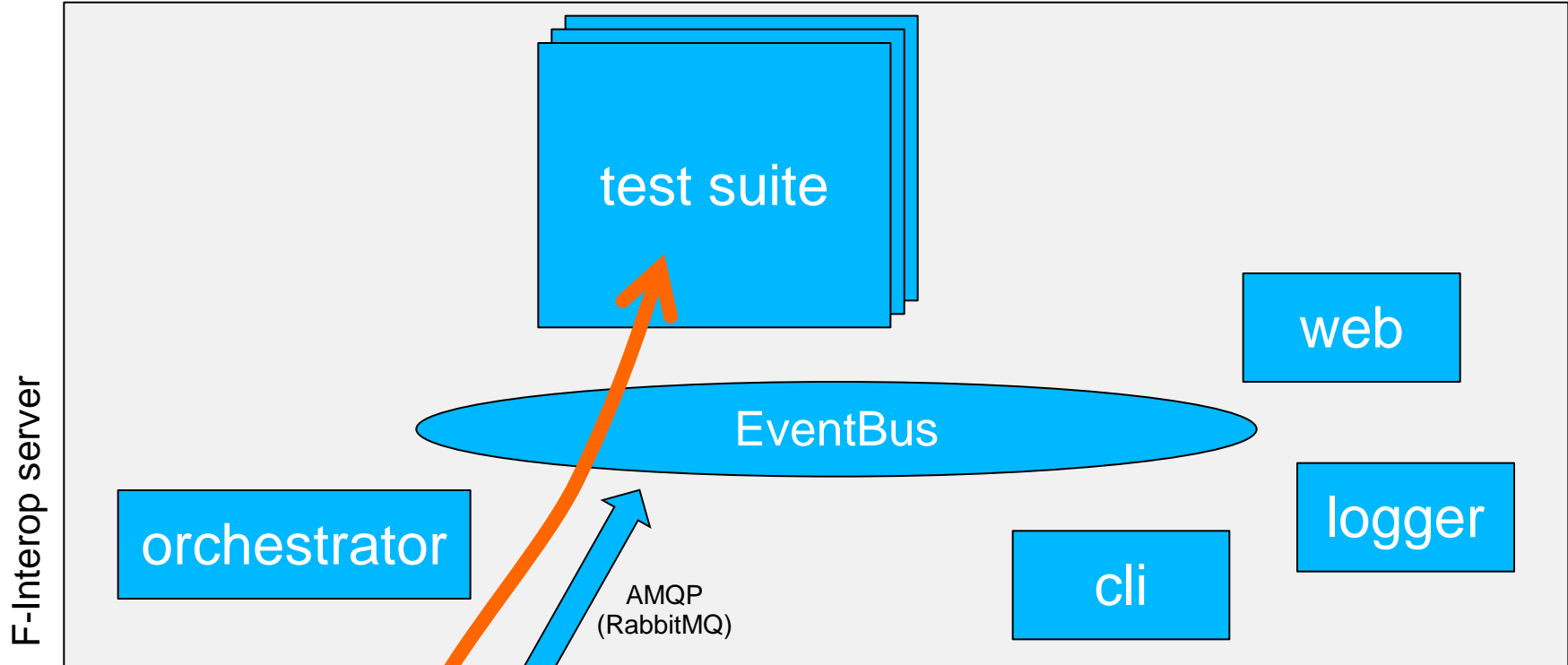
# Base Architecture (CoAP example)



# Advanced Architecture (6TiSCH example)

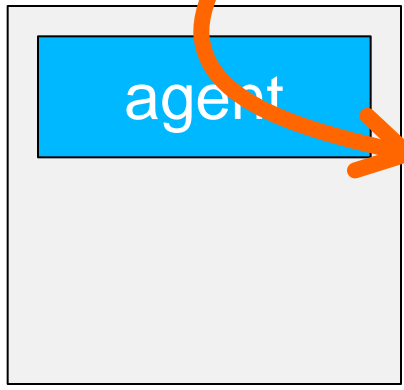


TEROP



F-Interop server

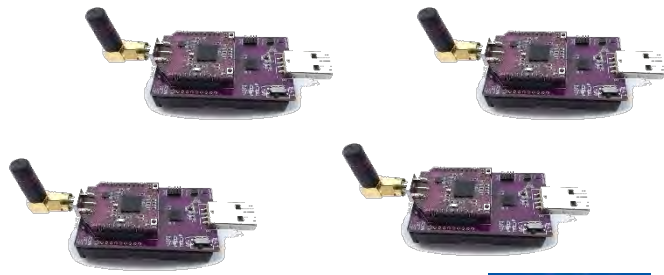
Configuration (e.g. control clock drift)



user



"Golden Device"  
DAGroot



# Advanced Architecture (testbed example)



TEROP

