



HORIZON 2020
Information and Communication Technologies
Integrating experiments and facilities in FIRE+

Deliverable D2.4
Online interop test core enablers 2nd
iteration

Grant Agreement number: 687884

Project acronym: F-Interop

**Project title: FIRE+ online interoperability and performance test tools to support emerging technologies from research to standardization and market launch
The standards and innovations accelerating tool**

Type of action: Research and Innovation Action (RIA)

Project website address: www.finterop.eu/

Due date of deliverable: M23

Dissemination level: PU

This deliverable has been written in the context of the Horizon 2020 European research project F-Interop, which is supported by the European Commission and the Swiss State Secretariat for Education, Research and Innovation. The opinions expressed and arguments employed do not engage the supporting parties.



Co-funded by the
European Union



Co-funded by the
Swiss Confederation

Document properties

Responsible partner	INRIA
Author(s)/editor(s)	Federico Sismondi, César Viho
Version	1.0
Keywords	Interoperability Testing, Remote Testing, Online, Platform, Testing components, Test enablers

Abstract

This short report corresponds to the deliverable **D2.4 – Online Interoperability test core enablers 2nd iteration**. It provides complementary information to the deliverable D2.4 which was delivered in the shape of a source code package along with a video demonstration of the interoperability tool and enablers. It describes the advancement between the 1st iteration and the current status of the F-Interop online remote interoperability framework and the key enablers needed.

Table of Contents

Table of Contents	3
List of Acronyms	4
1 Introduction	5
1.1 About F-Interop	5
1.2 Deliverable Objectives	5
1.2.1 Work package Objectives	5
1.2.2 Task Objectives	5
1.2.3 Deliverables Objectives and Methodology	6
2 Achievements of the 2nd iteration	7
2.1 User’s journey for remote online interoperability testing.	7
2.1.1 Executing remotely conformance and interoperability in F-Interop	7
2.1.2 User’s journey – Test configuration phase (step 0 to 4)	8
2.1.3 User’s journey – Test execution and results phase (step 5 to 6)	11
3 Conclusion and future work	14
4 Annex	15
4.1 F-Interop session	15
4.2 An example of CoAP interoperability test description from ETSI plugtest CoAP#4, IETF89 (London)	16
4.3 An example of Test Extended Description (YAML file)	17
4.4 An example of CoAP Interoperability Test Script	18

List of Acronyms

6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4e
CoAP	Constrained Application Protocol
EC	European Commission
ETSI	European Telecommunications Standards Institute
EU	European Union
GPS	Global Positioning System
GUI	Graphical User Interphase
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and Communication Technologies
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO	International Standards Organization
IT	Information Technology
MAC	Media Access Control
OS	Operating System
R&D	Research & Development
RR	Resources Repository
SME	Small Medium Enterprise
SO	Session Orchestrator
TAT	Test Analysis Tool
TED	Test Extended Description
TL	Task Leader
TT	Testing Tool
URL	Uniform Resource Locator
WP	Work Package
W3C	World Wide Web Consortium

1 Introduction

1.1 About F-Interop

F-Interop is a Horizon 2020 European Research project, which proposes to extend the European research infrastructure (FIRE+) with online and remote interoperability and performance test tools supporting emerging technologies from research to standardization and to market launch. The outcome will be a set of tools enabling:

- Standardization communities to save time and resources, to be more inclusive with partners who cannot afford travelling, and to accelerate standardization processes;
- SMEs and companies to develop standards-based interoperable products with a shorter time-to-market and significantly lowered engineering and financial overhead.

F-Interop intends to position FIRE+ as an accelerator for new standards and innovations.

1.2 Deliverable Objectives

1.2.1 Work package Objectives

- Research and develop the online remote interoperability test key enablers
- Develop the conformance test enablers
- Implement and fine tune the requested tools with a modular architecture for extensibility

1.2.2 Task Objectives

1.2.2.1 T2.1: Online interop test core enablers M1-M33 (Task Leader: Inria)

Work. The main objective of this task is to define and implement the components of the F-Interop online remote interoperability-testing framework. This includes: the cloud-based interoperability test script repository (as well as its management), the test servers and test suites automation, as well as the libraries, adapters, API and hardware interfaces, and its reporting capability. This task includes developing new methods simplifying online remote interoperability testing. This task also considers associated security and authentication issues.

Roles: Inria will lead the task, and will integrate contributions from ETSI, UL and EANTC.

Outcome: All components and key enablers needed for online remote interoperability testing.

T2.2: Complementary conformance test enablers M1-M33 (Task Leader: Inria)

Work. Conformance is a pre-requisite for interoperability. This task will provide the enablers for online remote conformance testing which complement the key enabled developed in task T2.1. We will develop new methods and/or adaptations of existing conformance testing tools to take into account the specific case of interacting online and remotely with the implementation under test (IUT). Examples include sniffers platforms, tools for measuring end-to-end latency (for example based on GPS synchronization), a protocol dissector.

Roles: Inria will lead the task, and will integrate contributions from ETSI, UL and EANTC.

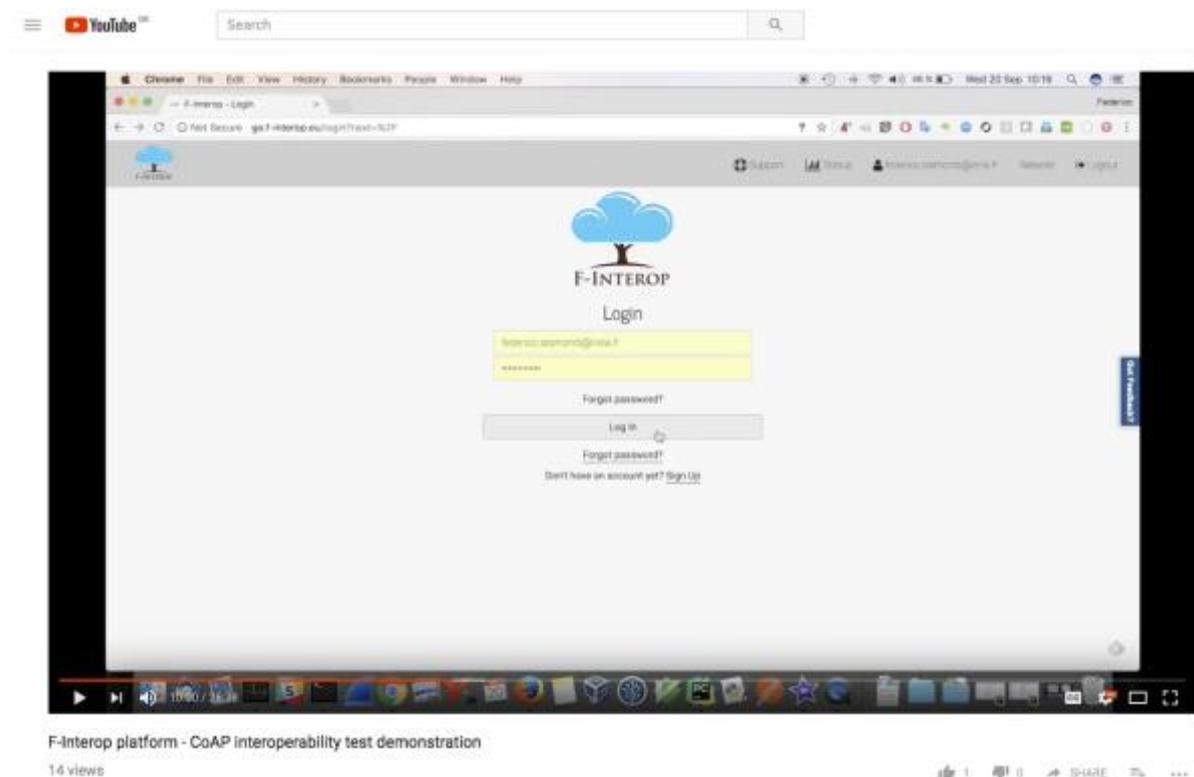
Outcome: Additional components needed for online remote conformance testing. These components will complement the key enablers developed in task T2.1.

1.2.3 Deliverables Objectives and Methodology

The deliverable corresponds to “Online interop test core enablers 2nd iteration”. The deliverable itself is in the shape of source code and a video demonstrating a CoAP interoperability test session in action using these enablers. The video presents the setup and execution of a test which correspond to steps 0 to step 6 of the F-Interop session actions (see F-Interop session annex). The current document acts as complementary information of the deliverable D2.4, and focuses on stating what has been achieved and demonstrated since the 1st iteration.

The deliverable is composed by the following resources:

- (1) video demonstrating a CoAP interoperability test session in action



the video can be found at: <https://youtu.be/BbWIGjcpVag>

- (2) the source code of the deliverable which has been tagged in F-Interop gitlab as 'D2.4':

- Testing Tool:
source code: https://gitlab.f-interop.eu/f-interop-contributors/coap_testing_tool/tags/D2.4
- Agent:
source code : <https://gitlab.f-interop.eu/f-interop-contributors/agent/tags/D2.4>
- Session orchestrator:
source code : https://gitlab.f-interop.eu/f-interop/f-interop_ietf/tags/D2.4
- CLI:
source code : <https://gitlab.f-interop.eu/f-interop-contributors/utills/tags/D2.4>

2 Achievements of the 2nd iteration

This section presents what has been achieved and demonstrated since the 1st iteration. This presentation of the work for achievements of the second iteration uses as artefact the “**user journey**” (of an interop session) and also the definition of the different steps defined in the **F-Interop session**. This way of presenting the information has proved to give the reader a better understanding of the session flow, and also helps understanding **how** and at **which moment** each component of the architecture participates into the F-Interop session.

2.1 User’s journey for remote online interoperability testing.

2.1.1 Executing remotely conformance and interoperability in F-Interop

The two figures below show the core idea of what we intend to provide to allow executing remotely and online conformance and interoperability test in the F-Interop context.

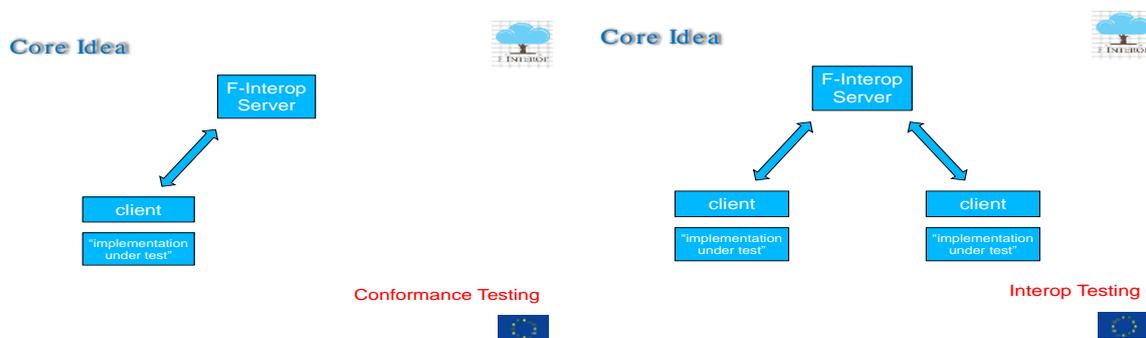


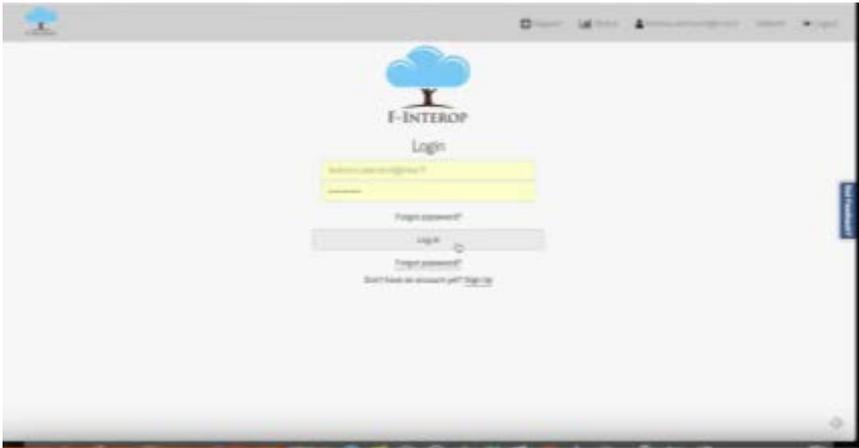
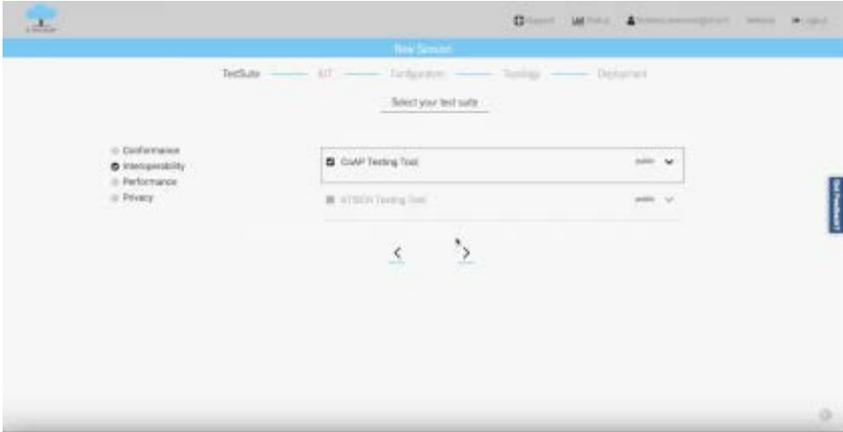
Figure 1 Executing conformance and interoperability in F-Interop

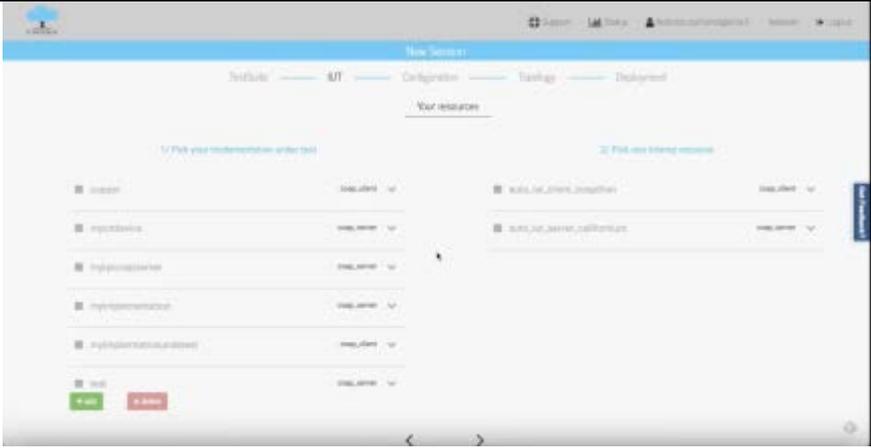
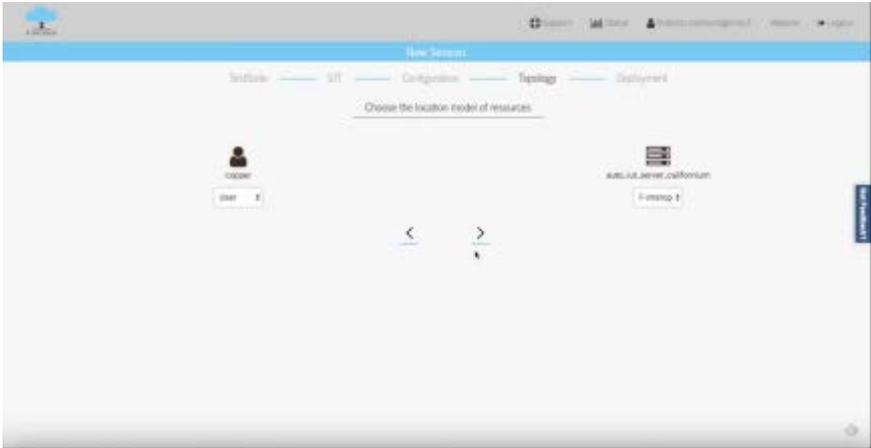
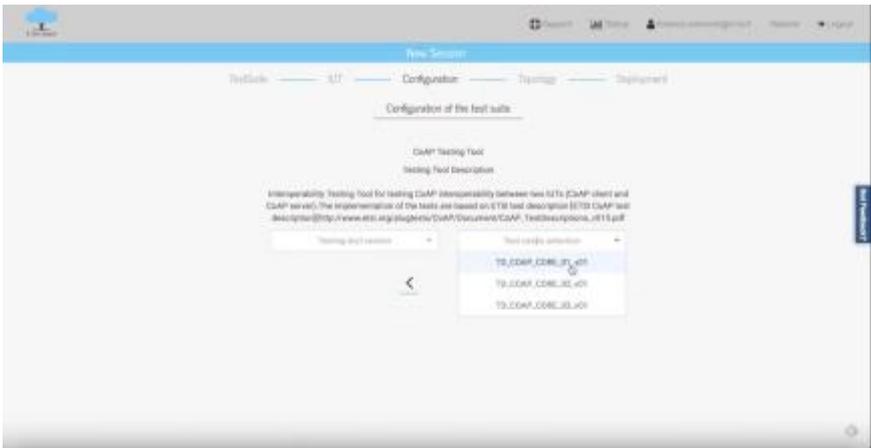
For the conformance testing on the left of Figure 1, the testing tool on a F-Interop server executes remotely the tests against a distant IUT. A client or agent will be in charge of managing the interaction between the testing tool and the IUT. In the case of interoperability on the right side of the Figure 1, the interoperability testing tool is on the F-Interop server and is executed remotely against two or more distant IUTs.

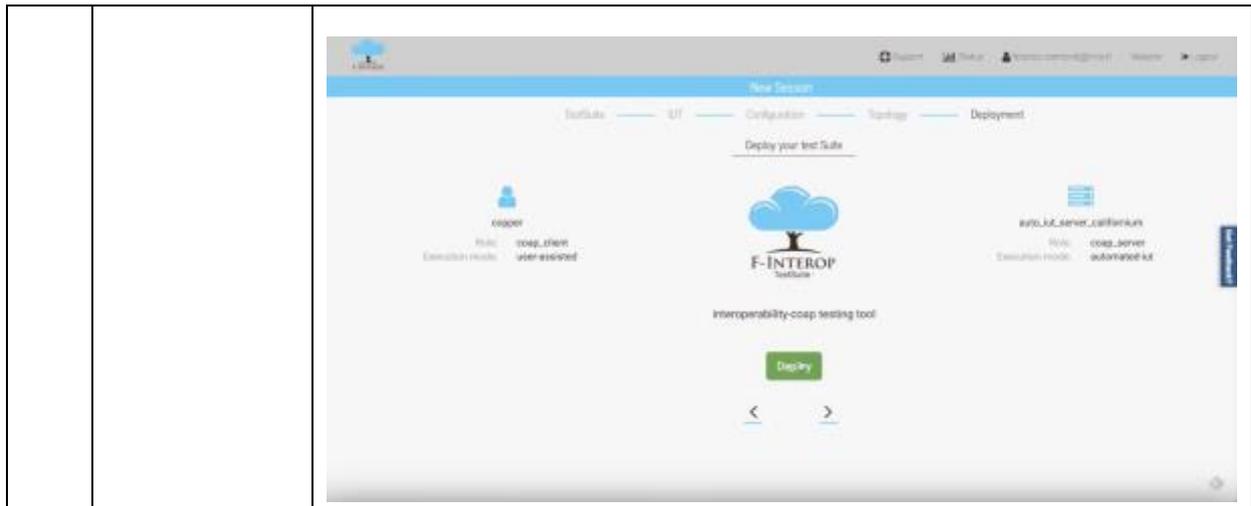
2.1.2 User's journey – Test configuration phase (step 0 to 4)

This phase of the journey is associated with steps 0 to 4 of the F-Interop session. These steps were demo-ed in the deliverable video by using the GUI. For the sake of readability, we included GUI screenshots of these steps:

2.1.2.1 What has been demonstrated

Step	Action	Description
0	FI-User authentication and authorization. IUT registration / identification	
1	Test suites discovery and selection	<p>(FI-User starts by discovering the available test suites and by selecting the one he/she wants to execute.)</p> 
2	Resource description	(resources selection)

		 <p>(location model selection)</p>  <p>(testing tool configuration)</p> 
3	Resource reservation	No need for the time being of pre-reserved resources. This step will be considered once F-Interop tools are integrated with testbeds.
4	Resource provisioning, configuration and session setup	(the instantiation of the F-Interop-Platform resources)



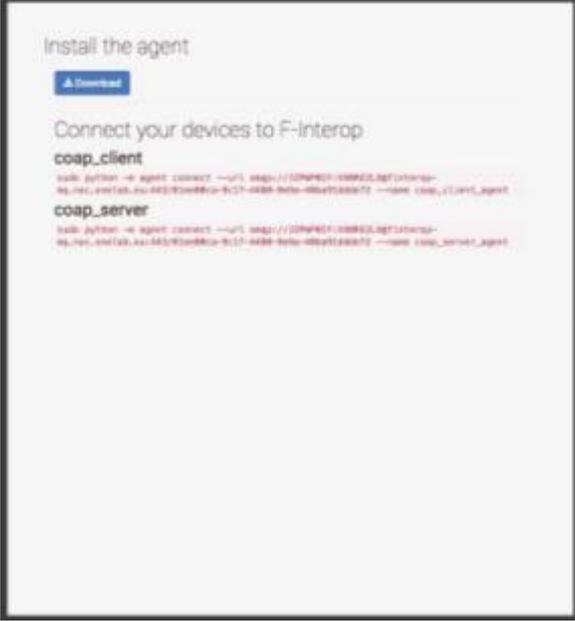
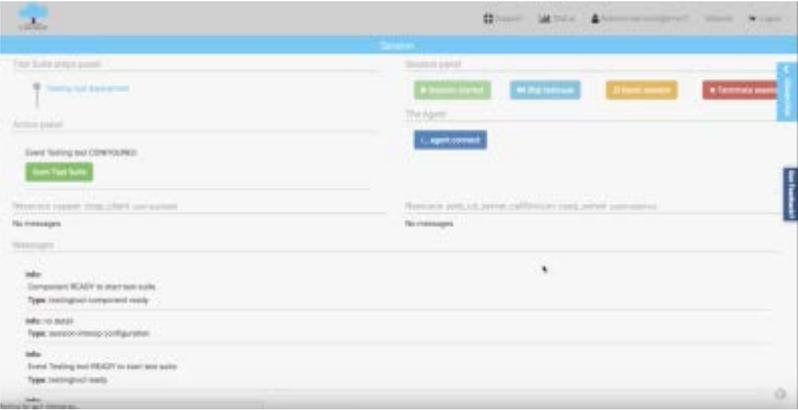
2.1.2.2 What has been achieved

During this first phase of the F-Interop session, we have achieved a full integration between several components of the F-Interop architecture. The ones that regard WP2 and D2.4 are:

- GUI <--> SO integration:
 - o GUI queries SO about available test suites and the configurations (needed inputs from the user) that are needed to run the tests.
 - o GUI triggers test suites instantiations into SO (central server test suites in docker containers deployment).
- SO <--> TT integration:
 - o Right after deployment, SO sends the configuration message to the TT. The TT confirms that it has been configured to the SO, and the GUI.

2.1.3 User's journey – Test execution and results phase (step 5 to 6)

2.1.3.1 What has been demonstrated

Step	Action	Description
5	Test execution	<p>The online F-Interop test campaign is launched and the selected (executable) test suites are executed against the IUTs.</p> <p>(first, user downloads and launches the agent)</p>  <p>(ready to execute the interop session with GUI)</p>  <p>(or with CLI)</p>

		<pre> _api_version 8.1.42 _type testcoordination.testcase.ready _description Next test case to be executed is TD_CDRP_CORE_01_v01 notes objective Perform GET transaction(CDR mode) pre_conditions ["Server offers the resource /test with resource content is not empty that handles GET with an arbitrary payload"] state testcase_id TD_CDRP_CORE_01_v01 testcase_ref http://doc.f-interop.eu/tests/TD_CDRP_CORE_01_v01 [Test Assistant] Suggested following action to executes <action to_start> or <action suggested> Event received: testcoordination.testcase.ready > > action suggested [User input] suggested Executing : to_start Do you want START test case? [y/N]: y Sending message.. [Session message] [testcoordination.testcase.start] _api_version 8.1.42 _type testcoordination.testcase.start _description Event test case START </pre>
6	Results analysis and report	<p>(a verdict is issued after each test case termination)</p> <pre> [Session message] [testcoordination.testcase.verdict] _api_version 8.1.42 _type testcoordination.testcase.verdict _description No interoperability error was detected, objective Perform GET transaction(CDR mode) partial_verdicts [{"TD_CDRP_CORE_01_v01_step_02", None, "CHECK step: postponed", ""}, {"TD_CDRP_CORE_01_v01_step_03", None, "CHECK step: postponed", ""}, {"TD_CDRP_CORE_01_v01_step_04", "pass", "VERIFY step: User informed that the information was displayed correctly on his/her DUT"}] state Finished testcase_id TD_CDRP_CORE_01_v01 testcase_ref http://doc.f-interop.eu/tests/TD_CDRP_CORE_01_v01 verdict pass Partial verdicts: ----- Step ID Partial verdict Description ----- TD_CDRP_CORE_01_v01_step_02 CHECK step: postponed TD_CDRP_CORE_01_v01_step_03 CHECK step: postponed TD_CDRP_CORE_01_v01_step_04 pass VERIFY step: User informed that the information was displayed correctly on his/her DUT post_worries_analysis_check_1 pass <Frame 1: [0000:1 -> 0000:2] CoAP [DIX 11756] GET /test> Match: CoAPType=0, code=1 post_worries_analysis_check_2 pass <Frame 1: [0000:1 -> 0000:2] CoAP [DIX 11756] GET /test> Match: CoAPType=0, code=1 post_worries_analysis_check_3 pass <Frame 2: [0000:2 -> 0000:1] CoAP [DIX 11756] 2.05 Content > Match: CoAPCode=05, rid=0x0000, tok=b'', pl=0x00'''' post_worries_analysis_check_4 pass <Frame 2: [0000:2 -> 0000:1] CoAP [DIX 11756] 2.05 Content > Match: CoAPCode=05, rid=0x0000, tok=b'', pl=0x00'''' (a report is issued after test suite completion) [Session message] [testcoordination.testresults.report] Testcase ID Final verdict Description ----- TD_CDRP_CORE_01_v01 pass No interoperability error was detected. Step ID Partial verdict Description ----- TD_CDRP_CORE_01_v01_step_02 CHECK postponed TD_CDRP_CORE_01_v01_step_03 CHECK postponed TD_CDRP_CORE_01_v01_step_04 pass VERIFY step: User informed that the information was displayed correctly on his/her DUT CHECK_1_post_worries_analysis pass <Frame 3: [0000:1 -> 0000:2] CoAP [DIX 41211] GET /test> Match: CoAPType=0, code=1 CHECK_2_post_worries_analysis pass <Frame 4: [0000:1 -> 0000:1] CoAP [DIX 41211] 2.05 Content > Match: CoAPCode=05, rid=0x0000, tok=b'', pl=0x00'''' CHECK_3_post_worries_analysis pass <Frame 4: [0000:1 -> 0000:1] CoAP [DIX 41211] 2.05 Content > Match: CoAPCode=05, rid=0x0000, tok=b'', pl=0x00'''' Testcase ID Final verdict Description ----- TD_CDRP_CORE_02_v01 pass No interoperability error was detected. Step ID Partial verdict Description ----- TD_CDRP_CORE_02_v01_step_02 CHECK postponed TD_CDRP_CORE_02_v01_step_03 CHECK postponed TD_CDRP_CORE_02_v01_step_04 pass VERIFY step: User informed that the information was displayed correctly on his/her DUT CHECK_1_post_worries_analysis pass <Frame 3: [0000:1 -> 0000:2] CoAP [DIX 41211] DELETE /test> Match: CoAPType=0, code=1 CHECK_2_post_worries_analysis pass <Frame 4: [0000:1 -> 0000:1] CoAP [DIX 41211] 2.02 Deleted > Match: CoAPCode=05, rid=0x0000, tok=b'' </pre>
7	Session storage	(still not implemented)

2.1.3.2 What has been achieved

During this first phase of the F-Interop session, we have achieved integration between several components of the F-Interop architecture. The ones that regard WP2 and D2.4 are:

- GUI <--> TT integration:
 - o The execution phase regarding step 5 of the F-Interop session has been achieved and demonstrated. Also, a new more generic way for the testing tools to use the GUI is being investigated to favour extensibility and flexibility.
- CLI <--> TT integration:
 - o as an optional way of running the execution phase the User can opt to use the CLI, which integrates all functionalities provided by the TT.
 - o CLI handles features like chat between F-Interop Users participating in an interoperability session for enhancing the coordination of the interoperability tests, commands for verifying the correct deployment and behaviour of the TT, and more.
- Agent <--> TT integration:
 - o Testing tool drives the agent from beginning to end of the session.
 - o Agent in collaboration with the TT provide IPv6 tunnelling over AMQP between both remote IUTs. This provide remote IPv6 reachability between IUTs.
 - o Agents and the IPv6 tunnelling mechanism has as extra advantage the fact that even when users don't have IPv6 connectivity in their facilities (either because their ISP doesn't provide IPv6 or because the facilities where the IUT runs hasn't adopted IPv6), the agent can still enable IPv6 in the localhost (user's PC), these IPv6 packets are then encapsulated into the AMQP messages and forwarded to the other IUT participating in the interoperability session.

Also, the achievements regarding the TT enhancements during 2nd iteration were:

- Complete AMQP integration.
- Complete interop testing cycle **from testing tool configuration to issuing of the final report** (with testcase verdicts and detail information on the exchanges analysis).
- 24 test cases implemented for CoAP CORE (RFC 7252).
- Full documentation of TT AMQP API calls on doc.f-interop.eu/#interoperability-testing-tool

3 Conclusion and future work

This document presented what has been achieved and demonstrated since the 1st iteration. This presentation of the work for achievements of the second iteration used as artefact the “**user journey**” (of an interop session) and also the definition of the different steps defined in the **F-Interop session**. This way of presenting the information has proved to give the reader a better understanding of the session flow, and also helps understanding **how** and at **which moment** each component of the architecture participates into the F-Interop session.

We presented some indicators on the status of work regarding the 2nd iteration of Online Interoperability test core enablers corresponding to deliverable D2.4. The description of the enablers used for interoperability has been omitted and can be consulted at the D2.1 given that the requirements on the components have not changed since the 1st iteration.

The 2nd iteration delivered a set of components which enable a complete interoperability session execution with the remote and online approach. This was demonstrated for the CoAP interoperability test use case, but can be extended to other protocols or technologies simply by providing:

1. extra libraries to the analysis component
2. extra test descriptions files for describing coordination of the test (see An example of Test Extended Description (YAML file))
3. extending the agent with plugins for interfacing with new IoT devices or new virtual implementation resources

The future work for the last iteration should include:

- Move the CoAP interoperability testing tool and enablers into production and fine tuning the tools for end-users
- Validate its modularity by extending it to other protocols
 - Extend the TT for including Web of Things (WoT) interoperability tests
 - Extend the TT with other F-Interop Contributors' interoperability tests (e.g. the ones provided by third parties of the Open Call)
- Design and implement a simpler and more generic interface (API) for the TT and the GUI interaction

4 Annex

4.1 F-Interop session

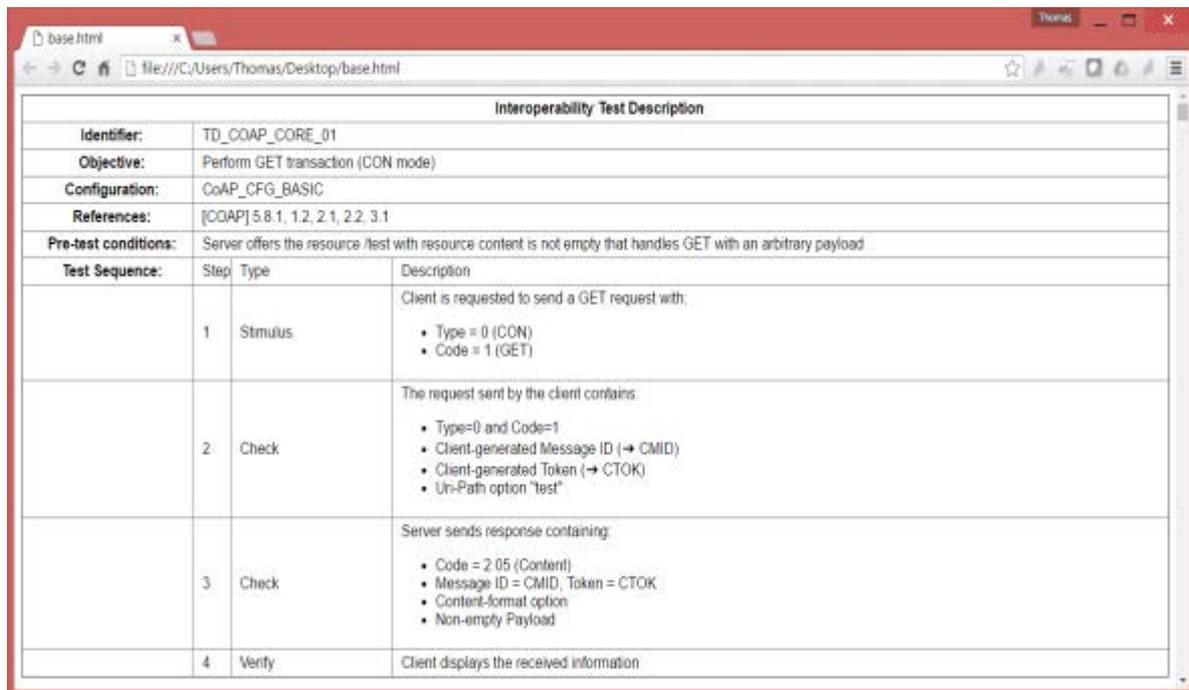
The set of actions for executing any type of tests (interoperability, conformance, performance, etc.) in the F-Interop platform are defined as the “**F-Interop session**” and are summarized in the following table.

Step	Action	Description
0	FI-User authentication and authorization. IUT registration / identification	FI-User authenticates in a secure way (prior FI-User registration needed) in FI-Platform. FI-User needs to be authorized to use FI-Platform resources. FI-User identifies which IUT he/she will test (prior IUT registration needed).
1	Test suites discovery and selection	FI-User starts by discovering the available test suites and by selecting the one he/she wants to execute.
2	Resource description	FI-User specifies/selects resources in the F-Interop-Platform that are needed for his/her F-Interop session including the location models ¹ , testing tools, libraries, etc. During this phase FI-Platform may request information from FI-User or provide information to FI-User for a coherent selection of the required resources.
3	Resource reservation	The resources selected in the previous step are actually reserved.
4	Resource provisioning, configuration and session setup	The instantiation of the F-Interop-Platform resources that fit best with the FI-User needs is done.
5	Test execution	The online F-Interop test campaign is launched and the selected (executable) test suites are executed against the IUTs.
6	Results analysis and report	Test execution information is analysed. The test results and verdicts are provided together with explanations in case of FAIL or INCONCLUSIVE verdicts or something wrong happened. A report can be provided under request in case for example the FI-User wants to apply for a certification/labelling program.
7	Session storage	Storage of the F-Interop session information (Session-id, User-id, FI-User's IUT-id, IUTs' version, test description, test version, testing tool, test log and results, etc.). This has to remain accessible beyond the F-Interop session for the involved parties.

Table 1 - F-Interop Session

¹ Location models are the different configurations for the location of components of the test. These will be defined in D1.3.1. [6]

4.2 An example of CoAP interoperability test description from ETSI plugtest CoAP#4, IETF89 (London)



The screenshot shows a web browser window with the address bar displaying 'file:///C:/Users/Thomas/Desktop/base.html'. The main content is a table titled 'Interoperability Test Description'. The table contains the following information:

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
	1	Stimulus
	2	Check
	3	Check
	4	Verify

The description for each step is as follows:

- Step 1 (Stimulus):** Client is requested to send a GET request with:
 - Type = 0 (CON)
 - Code = 1 (GET)
- Step 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"
- Step 3 (Check):** Server sends response containing:
 - Code = 2.05 (Content)
 - Message ID = CMID, Token = CTOK
 - Content-format option
 - Non-empty Payload
- Step 4 (Verify):** Client displays the received information

4.3 An example of Test Extended Description (YAML file)

```
testcase_id: TD_COAP_CORE_01
uri : http://f-interop.paris.inria.fr/tests/TD_COAP_CORE_01
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

4.4 An example of CoAP Interoperability Test Script

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")
```

