EENA Technical Committee Document

What is needed for Interoperability Testing?

Title	What is needed for interoperability testing		
Version	1.0		
Revision date	19/09/2016		
Status of the document	Draft	For comments	**Approved**

Authors and contributors to this document

This document was written by members of EENA:

Authors	Country/Organisation
Wolfgang Kampichler	Frequentis, EENA Technical Committee Chair

Contributors	Country/Organisation
Ian Colville	Aculab
Cristina Lumbreras	EENA
Darren Terry	Nice

Legal Disclaimer

This document is authored by EENA staff members with contributions from individual members of EENA and represents the views of EENA. This document does not represent the views of individual members of EENA, or any other parties.

This document is published for information purposes only and it does not declare to be a statement or interpretation of EU law or the national law of EU Member States. This document is entirely without prejudice to the views of relevant national statutory authorities and their legal functions and powers, whether under EU law or the national law of their Member State. Accordingly, under no circumstances may reliance be placed upon this document by any parties in compliance or otherwise with any applicable laws. Neither may reliance be placed upon this document in relation to the suitability or functionality of any technical specifications, or any other matters discussed in it. Legal advice, technical advice and other advice as relevant, may be sought as necessary.

Table of contents

1 Executive Summary

This document provides an analysis and evaluation of what is needed for successful interoperability testing of emergency communication standards. One of the key motives for the development of emergency communication standards is to facilitate interoperability between products in a multi-vendor, multi-network and multi-service environment. Standards need to be designed and tested to ensure that products and services complying with them do, indeed, achieve interoperability. Interoperability ensures that users have a much greater choice of products and that manufacturers benefit from the economies of scale that a wider market brings. Testing and interoperability are therefore crucial factors in the success of next generation emergency technologies. Interoperability testing involves connecting devices from different vendors and operating them in a variety of real-life scenarios. Usually this is done at so called interoperability events that, in order to be successful, lay down certain factors. This document evaluates that range of factors and concludes that a good mix of vendors that implement a variety of services and features as well as a comprehensive test specification, combined with a proper infrastructure that supports lab and pretesting, are essential for successful testing. Recommendations discussed include:

- Use pre-defined data sets covering all needed elements
- Be stricter in pre-testing and conduct more formalised pre-testing, e.g. run the basic tests
- Define more complex routing policies to cover real world scenarios

This document also provides an example interoperability test scenario in order to get a better understanding of the planning process.

2 Introduction

Next generation (NG) emergency communication systems are designed to close the gap between quickly evolving technologies (fixed and mobile IP-based communications) and the more conservative or traditional approaches adopted by the emergency communications industry. NG emergency communications enable citizens to contact emergency services in different ways, using the same types of technology as those they use to communicate every day. Furthermore, it creates new opportunities and new challenges with regard to the design and the implementation of emergency communication systems. Different services from multiple providers need to be interconnected and therefore it must be ensured that chosen components are interoperable. When implementing NG, a few key questions arise: Will it all work? Are there common interfaces I can use to route emergency calls? Will the NG-based emergency communication systems work as well as or better than a traditional solution? This document will cover key facts related to interoperability testing based on discussions involving the EENA Technical Committee and experience gained from past interoperability testing projects.

3 General Aspects

Prior to any technical matter, the following factors are important for successful interoperability testing:

- Standards or umbrella documents that explain the use of various standards with a certain level of maturity such that the industry is ready to adopt them
- A solid idea of what’s being tested, ideally documented in a high level scenario
- Volunteers that contribute to the technical planning of an interoperability test event
- An industry that is willing to implement and test new features and interfaces
- A location that not only attracts people, but also provides the required technical infrastructure paired with IT expertise

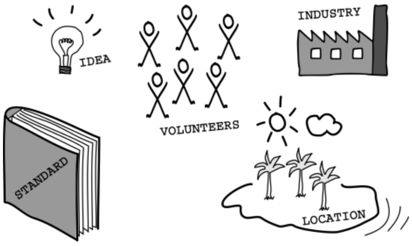

Figure 1: Non-technical factors.

In general, interoperability testing involves testing whether a given software application or technology is compatible with others and promotes cross-use functionality. Applications may provide certain features or functional capabilities that are typically part of a specific service. Such services are orchestrated in order to fulfil a certain task (e.g. routing emergency calls to the most appropriate PSAP).

The factors in interoperability testing include syntax and data format compatibility as well as sufficient physical and logical connection methods. The main objective is to be able to route data back and forth without causing operational issues, losing data, or otherwise losing functionality. In order to facilitate this, each component needs to recognise incoming data from other services, process data depending on its role in the NG emergency communications architecture, and provide useful results. Data formats, and physical and logical connection methods, are collectively identified by the term ‘interface’ and considered as such in standard documents. Defined interfaces will be different, depending on the services that are interconnected for a specific purpose.
It is evident that if testing end-to-end, several different interfaces, features and services need to be considered as shown in Figure 2 (from **IN** to **OUT**).

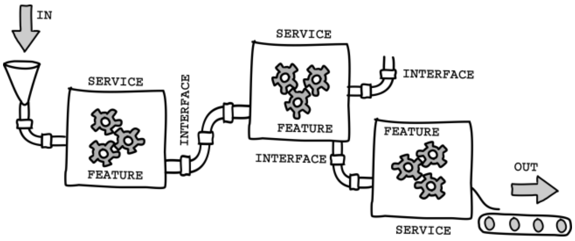

Figure 2: End-to-end testing.

Figure 2 also depicts a specific configuration – the combination of services and specific features – that is used for end-to-end interoperability testing and the interfaces used to interconnect functional elements. An important point is that interconnected services implement compatible features, e.g. an element that requests location-by-reference from a Location Information Service (LIS) assumes that a LIS implements such functionality, otherwise interoperability will not be possible as shown in Figure 3.

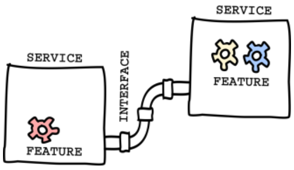

Figure 3: Feature incompatibility.

The same holds true for interfaces that interconnect services (Figure 4). Even if features are compatible, e.g. a LIS that supports location-by-value, there is no guarantee that data can be routed back and forth without causing issues. For instance, if the standard requires a secure transport protocol that is not supported by either party.

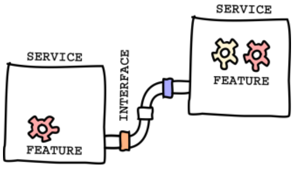

Figure 4: Interface incompatibility.

Participants that contribute functional elements to interoperability test events need to make sure that both interface and feature compatibility are considered. This is considered by Interoperable Functions Statements (IFS) that identify standardised functions of a specific feature. These functions can be mandatory, optional or conditional (depending on other functions), and depend on the role of a certain functional element or service. The IFS can also be used as a proforma by a vendor to identify the functions that its service will support when interoperating with corresponding functions from other vendors.

A participating vendor need not implement all features, but at least should provide a feature that is able to connect to a remote service, receive data via a common interface, and process it according to standard definitions, as shown in Figure 5.

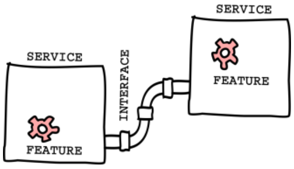

Figure 5: Interface and Feature compatibility.

Ideally, interfaces between adjacent services should be bilaterally tested prior to an event. This reduces debugging and tracing time during an interoperability test event, and in general helps in testing newly developed interfaces. In order to support bilateral testing it is required to provide at least:

- An overall system configuration that explains which services are required
- A list of adjacent elements to test
- A basic set of data in order to either configure services or support testing
- Simple test scenarios and supporting elements if required

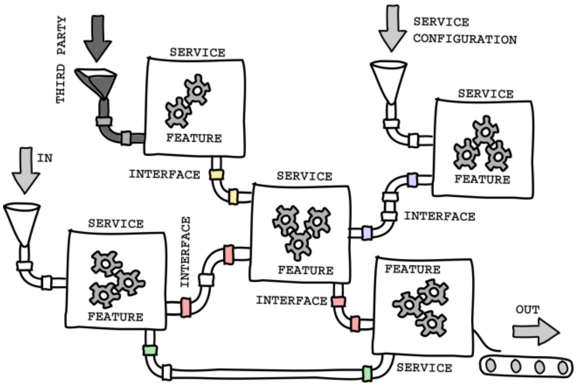

Figure 6: Overall System Configuration.

Figure 6 shows an overall system configuration for end-to-end and bilateral testing. Please note that adjacent neighbours may differ depending on which interfaces are tested. As mentioned earlier, testing requires a set of data in order to configure individual services (e.g. network and transport layer parameters) and to provide certain functionality (e.g. location based call routing). The former is considered when planning the IT infrastructure and the latter depends on the scenarios to be tested. In terms of emergency communications, it is necessary at least to determine Public Safety Answering Point (PSAP) boundaries and mapp them to PSAP URIs (i.e. communication end-points). End-to-end testing of emergency communications also includes access to third party features and services (see Figure 6) that may be, for instance, different types of originating networks. The term third party in this context means access to an infrastructure that is based on standards different to those that are the subject of testing. Therefore, it is important to include such elements in a test specification, at least as "black box" elements that operate a single interface to the infrastructure under test.

A vital element that contributes to successful interoperability testing is a comprehensive test specification. The test specification introduces the main objectives of a specific event and provides mandatory technical information. Basically, it should be a document that evolves over time where each test event maintains its own edition. This not only helps to avoid duplication of work, but also ensures that future events are based on past experience and effort. A test specification lists normative and informative standards that specify or describe required services and interfaces (summarised in specific IFSs). Furthermore, it introduces mandatory system configurations and lists test scenarios combined with a test description that includes optional and mandatory IFS and individual test steps (as depicted in Figure 7). Test steps list the stimulus for a specific test, what needs to be checked in the message sequence, and finally, what should be verified in order to report a successful test.

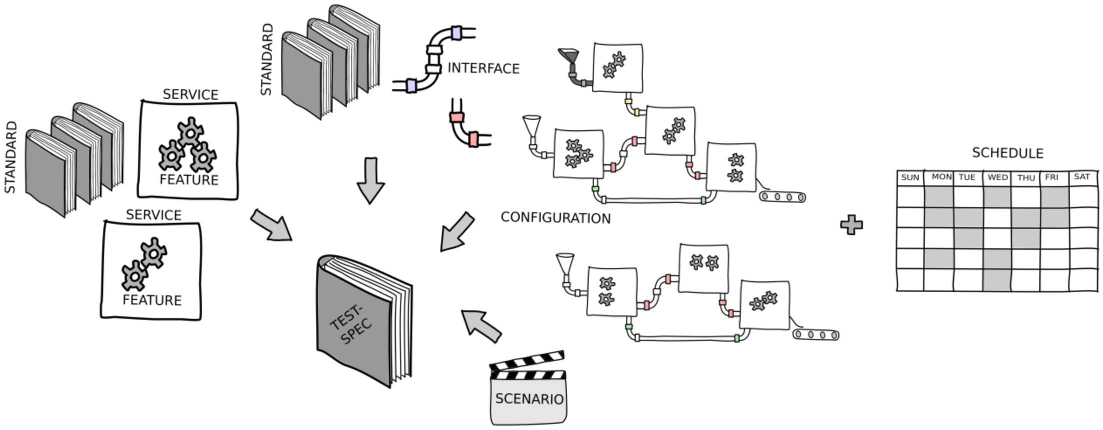

Figure 7: Test Specification and Schedule.

In addition to the Test Specification, a proper test schedule is needed. Such a schedule provides timeslots for testing permutations, with the aim to have as many combinations as possible of different vendor products (Figure 7). Input parameters are test scenarios, IFS and system configurations. As this is a very complex task, it is desirable to have tools that automatically generate a schedule and allow for manual adaption. Finally, there are several key success indicators of interoperability testing and a few critical success factors as listed below:

- A good mix of vendors (at least 2 or 3 per functional element or service)
- A mix of products that implement a variety of services and features or functions
- Scenario planning and scheduling that evenly considers all possible configurations
- A comprehensive test specification (configuration, test description, IFS, …)
- A proper infrastructure that supports lab and pretesting
- Tools to capture test results

4 Example

In order to get a better understanding of the planning process, let us consider a test scenario with an emergency call that originates in an IMS network (Voice over LTE, VoLTE), and, just for testing purposes, where location is provided via AML (SMS) to a location hub (LIS). The system configuration (CFG-1) is shown in Figure 8.

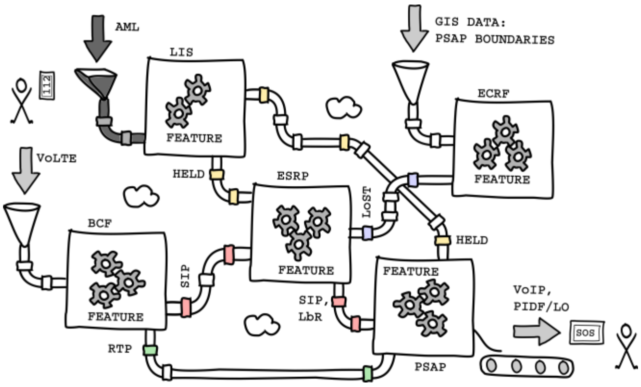

Figure 8: Example Configuration – CFG-1.

Service elements[1] providing specific features (from left to right in Figure 8) are:

- BCF, Border Control Function
- LIS, Location Information Service
- ESRP, Emergency Service Routing Proxy
- PSAP, Public Safety Answering Point
- ECRF, Emergency Call Routing Function

Each service implements features or functions that are summarised as IFS. The following IFS table serves as an example and is simplified.

Service	IFS (mandatory)
BCF	Does the BCF support SIP and RTP?
LIS	Does the LIS support AML, HELD, LbV, LbR?
ESRP	Does the ESRP support SIP, LoST, HELD, LbR, LbV?
PSAP	Does the PSAP support SIP, RTP, HELD, LbR, LbV?
ECRF	Does the ECRF support LoST (point, circle)?

The following vendor table assumes that there are at least two participants/vendors per service.

Service	Vendor	IFS
BCF	A	Supports SIP and RTP?
	B	Supports SIP and RTP?
LIS	C	Supports AML, HELD, LbV, LbR
	D	Supports HELD, LbV, LbR?
ESRP	E	Supports SIP, LoST, HELD, LbR
	F	Supports SIP, LoST, HELD, LbR, LbV
PSAP	G	Supports SIP, RTP, HELD, LbV
	H	Supports SIP, RTP, HELD, LbR, LbV
ECRF	I	Supports LoST (point, circle)
	J	Supports LoST (point, circle)

[1] Refer to http://www.eena.org/uploads/gallery/files/pdf/2013-03-15-eena_ng_longtermdefinitionupdated.pdf

Summarising, we start with a scenario that requires a certain system configuration that describes service elements and their combination. In addition, we define a list of IFSs that are mandatory for a specific test case and a vendor list. The next step is to identify vendors that support mandatory IFS in order to be scheduled for testing. Please note that it is not mandatory to implement each feature that is listed as IFS. Typically, there are test scenarios that differ in mandatory IFS.

A schedule with four parallel slots may look like the one shown in the table below. Please note that, in this example, PSAP vendor G just supports LbV, therefore, only combinations with ESRP vendor F, which supports both LbV and LbR, are possible.

Time	Slot 1	Slot 2	Slot 3	Slot 4
09:00 – 10:00	A – BCF C – LIS E – ESRP H – PSAP I – ECRF *CFG-1*	A – BCF C – LIS E – ESRP H – PSAP J – ECRF *CFG-1*	B – BCF C – LIS E – ESRP H – PSAP I – ECRF *CFG-1*	B – BCF C – LIS E – ESRP H – PSAP J – ECRF *CFG-1*
10:00 – 11:00	A – BCF C – LIS F – ESRP H – PSAP I – ECRF *CFG-1*	A – BCF C – LIS F – ESRP H – PSAP J – ECRF *CFG-1*	B – BCF C – LIS F – ESRP H – PSAP I – ECRF *CFG-1*	B – BCF C – LIS F – ESRP H – PSAP J – ECRF *CFG-1*
11:00 – 12:00	A – BCF C – LIS F – ESRP G – PSAP I – ECRF *CFG-1*	A – BCF C – LIS F – ESRP G – PSAP J – ECRF *CFG-1*	B – BCF C – LIS F – ESRP G – PSAP I – ECRF *CFG-1*	B – BCF C – LIS F – ESRP G – PSAP J – ECRF *CFG-1*
12:00 – 13:00	LUNCH			

Finally, a list of pre-conditions (mostly with regard to system or network configuration), and steps to follow for execution of the test case, as listed in the table below.

Step	Type	Description
1	stimulus	User dials emergency number
2	check	Dialog creating INVITE received at BCF domain
3	check	AML data string received at LIS
4	check	Dialog creating INVITE and received at ESRP
5	check	HELD request received at LIS
6	check	LoST request received at ECRF
7	check	Dialog creating INVITE + LbR received at PSAP
8	check	SIP dialog established
9	verify	PIDF/LO dereferenced at LIS by PSAP
10	verify	Call connected and location displayed

Considering 12 scheduled test runs with different combinations, an overall result may look like the one shown in the table below.

	Interoperability		Not Executed		Totals	
	OK	NO	NA	OT	Run	Results
CFG-1	9 (75%)	3 (25%)	0 (0%)	0 (0%)	12 (100%)	12 (100%)
CFG-2	…	…	…	…	…	…

NO…not ok, NA…not applicable, OT…out of time

5 EENA recommendations

Planning Process	Actions
Test Data	○ Use pre-defined data sets covering all needed elements ○ Provide data sets before the pre-testing phase as a csv file
Wiki Page	○ Update Wiki Page during the event in order to capture any changes
Testing	○ Produce a visual representation of the call path through elements, as it is difficult to have an overview of how calls are being routed ○ Use a more automated way to test the various routes ○ Define in the test plan who provides necessary information and how it should be queried ○ Sequential execution of tests is preferred if there is just a single core element
Pre-Testing	○ Be more strict in pre-testing ○ Every originating element should be pre-tested with BCF ○ Conduct more formalised pre-testing, e.g. run the basic tests
Test Scenarios	○ Include scenarios with stage-1 and stage-2 PSAPs ○ Include relay services and bridges ○ Consider testing of WebRTC, mobile users inside UC environment, and call-back calls (RFC7090) ○ Define more complex routing policies to cover real world scenarios

6 Abbreviations

AML	Advanced Mobile Location
BCF	Border Control Function
ECRF	Emergency Call Routing Function
ESRP	Emergency Service Routing Proxy
HELD	HTTP enabled Location Delivery
IFS	Interoperable Functions Statement
IMS	IP Multimedia Subsystem
IP	Internet Protocol
LIS	Location Information Service
LoST	Location to Service Translation
NG	Next Generation
PSAP	Public Safety Answering Point
RTP	Real-time Transport Protocol
SIP	Session Inititation Protocol