Project Number:	**FP7-257123**
Project Title:	**CONVERGENCE**
Dissemination Level:	**Public**
Deliverable Number:	**D6.3**
Contractual Date of Delivery to the CEC:	**31.07.2012**
Actual Date of Delivery to the CEC:	**22.11.2012**
Title of Deliverable:	**Report on Network and Middleware integration and testing**
Workpackage contributing to the Deliverable:	**WP6**
Nature of the Deliverable:	**Report**
Editors:	**Aziz Mousas, Dimitra Kaklamani, Charalampos Z. Patrikakis, Richard Walker**
Author(s):	**Angelos-Christos Anadiotis (ICCS), Aziz Mousas (ICCS), Sofia Kapellaki (ICCS), Nikolaos Dellas (ICCS), Georgios Lioudakis (ICCS), Vasso Giotopoulou (ICCS), Iakovos Venieris (ICCS), Andrea Detti (CNIT), Matteo Pomposini, Nicola Blefari Melazzi, Stefano Salsano (CNIT), Fernando Almeida (INESC), Angelo Difino (CEDEO), Panagiotis Gkonis (SIL)**
Abstract:	**Following the implementation of the CONVERGENCE middleware and network protocols, reported in D6.2, D6.3 describes the integration of the middleware components and the results of component and integration testing.**
Keyword List:	**Middleware, network, testing, integration, MXM, technology engines, protocol engines**

Executive Summary

This deliverable reports the integration and testing of the CONVERGENCE middleware engines and network and concludes the implementation work performed in WP6. In the deliverable, the particular tests designed and performed for the evaluation of the middleware and network are described, together with the corresponding results.

The structure of the deliverable follows the philosophy behind the planning and selection of testing methodology, which has been influenced by the type of engine (technology, orchestrator or protocol engine) and the functionalities it provides.

Therefore, the deliverable is divided into three sections: 1) Middleware Engines testing, 2) Integrated Orchestration testing, and 3) Network testing, and closes with a short conclusions section.

Two Annexes contain indicative test cases datasets and MAVEN build output in order to give a complete view of the testing of individual engines and integrated platform.

Table of Contents

1 Introduction

Following the implementation of the middleware and network prototypes reported in D6.2 [2], this deliverable describes the procedures used to verify their design and to validate their implementation, complementing performance testing in WP8. The following sections of this report describe first, tests of the middleware, verifying the design and implementation of individual engines, then integrated orchestration tests, verifying the correct execution of relevant protocols, orchestrations and workflows, and finally network tests designed to check network components. All tests are described using the same common template, shown below.

Name of test case	The name of the test case
Initial conditions	Conditions that have to be set up before the start of the test
Description of testing functionality	The functionality to be tested: tests are designed to check the behavior of the software in all possible execution environments.
Engines involved	The engines needed to perform the test.
Detailed steps	Detailed step-by-step description of the test case.
Dataset	The dataset used for the test: usually input files but also output files generated during the test (for examples see Annex A [Section 7]).
Output	Output files or results used to check if the test was successful
Expected result	The expected result of the test
Variants	Variant versions of the text (e.g. changes to the initial conditions or the input files)..

2 Middleware tests

This chapter describes the tests used to validate the implementation and the proper functioning of the middleware engines, specifically the Technology Engines; Protocol Engines are tested in the Integrated Orchestration tests discussed in the following section.

As described in D6.2, the middleware engines were all developed as MAVEN projects [3]. One of the advantages of this approach is that JUnit tests [5] are automatically incorporated in the build procedure: whenever a project is build, MAVEN automatically executes the corresponding JUnit tests. When the tests fail they generate a build failure – even when the code has compiled correctly – informing the developer of logical errors invisible to the compiler. This means that when refactoring breaks, existing functionality developers are notified instantly. Conversely, if a build is successful the developers can be confident that that the changes they have introduced have not damaged the proper functioning of the software. Annex B [Section 8] provides examples of the output from failed and successful builds.

2.1 CDS TE

The Community Dictionary Service TE (CDS TE) maintains a repository of local ontologies offering functionality to request ontology entities and expand metadata, based on inferred knowledge. The test cases below verify the proper functioning of the engine during the loading of an ontology, a request for an ontology entity and the expansion of metadata.

2.1.1 Ontology parsing

Name of test case	testOntologyParsing
Initial conditions	*Wordnet ontology file [4]*
Description of testing functionality	This test case verifies the proper parsing of a given ontology and prints descriptive information about the ontology, such as the number of statements in the ontology, the number of inferred statements and the number of ontology entities.
Engines involved	CDS TE
Detailed steps	<none> CDS loads local ontologies automatically on startup
Dataset	*Wordnet ontology file*
Output	Prints information about the ontology
Expected result	Print of successful parsing of the ontology, the number of statements it contains, the number of inferred statements and its ontology entities.

Variants	*This test has been implemented for TTL, RDF and OWL files.*

2.1.2 Request ontology entity

Name of test case	testRequestOntologyEntity
Initial conditions	*Wordnet ontology file*
Description of testing functionality	This test case verifies the results returned by the engine upon receiving a request for an ontology entity. The parameters to a request may include the URI for the ontology, an entity type, and a label prefix.
Engines involved	CDS TE
Detailed steps	1. Create multiple requests for ontology entities a. Set ontology URI b. Set entity type c. Set prefix 2. Check if returned results match the expected results
Dataset	*Wordnet ontology file*
Output	Prints of the matching ontology entities.
Expected result	The ontology entities returned for each request should match the expected results.
Variants	*N/A*

2.1.3 Expand metadata

Name of test case	testTagMetadataExpansion
Initial conditions	*Wordnet ontology file*
Description of testing functionality	This test case verifies the results of multiple tag expansion requests against the expected results. The CDS repository contains the Wordnet ontology and expected results are synonyms of the Wordnet ontology.
Engines involved	CDS TE
Detailed steps	1. Create multiple requests for tag expansion and check if returned results match the expected results
Dataset	*Wordnet ontology file*
Output	Prints of the equivalent tags.
Expected result	Returned tags for each request should match the expected

	results.
Variants	*This test has been implemented also for keyword and RDF metadata expansion.*

2.2 CoNet TE

The Information Centric Network of CONVERGENCE, named CONET, has been tested versus different aspects that regard:

i) the functions provided by the API,
ii) the underlying networking functionalities supporting the API,
iii) and performance evaluation in both laboratory and PlanetLab scenarios.

The deliverable D8.3, Section 5.1 [7], already includes the descriptions and results of these tests. Therefore, we avoid duplicating this content in this deliverable.

2.3 ConvergenceMetadata TE

The ConvergenceMetadata TE provides methods for creating and parsing the fundamental metadata structures inside R-VDIs, P-VDIs and S-VDIs, which contain descriptive information about each VDI. The test cases below verify the proper functioning of the engine, both during creating and parsing of the inner ConvergenceMetadata elements.

2.3.1 ConvergenceMetadata creation

Name of test case	testPublicationVDIMetadataCreation
Initial conditions	*Validated Publication ConvergenceMetadata file*
Description of testing functionality	This test case verifies the proper creation of the ConvergenceMetadata XML fragment intended for a P-VDI. The XML fragment is stored on disk and is checked with the metadata file in the dataset
Engines involved	ConvergenceMetadataTE
Detailed steps	1. Create ConvergenceMetadata a. Set publication fractal b. Set corresponding R-Vdi Id c. Set sequence identifier d. Set expiry date e. Add field/value metadata f. Add RDF metadata 2. Store ConvergenceMetadata to disk 3. Check if the stored ConvergenceMetadata and the dataset ConvergenceMetadata file are identical

Dataset [7.1.1]	Validated P-VDI ConvergenceMetadata file
Output	Publication ConvergenceMetadata file
Expected result	All intermediate schemahandler objects must be created without any exceptions and lead to the successful generation and storage of the ConvergenceMetadata XML fragment. The file created must be identical with the already validated P-VDI ConvergenceMetadata file in the dataset.
Variants	*This test has been implemented for parsing all kinds of ConvergenceMetadata.*

2.3.2 ConvergenceMatadata parsing

Name of test case	testPublicationVDIMetadataParsing
Initial conditions	*Validated Publication ConvergenceMetadata file*
Description of testing functionality	This test case verifies the proper parsing of a local file containing the ConvergenceMetadata XML fragment of a P-VDI. During the test, information inside the ConvergenceMetadata element are printed and checked with expected values.
Engines involved	ConvergenceMetadata TE
Detailed steps	1. Parse the local file and create ConvergenceMetadata a. Extract, print and check the publication fractal b. Extract, print and check the R-VDI Id c. Extract, print and check the sequence identifier d. Extract, print and check the expiry date e. Extract, print and check the field/value metadata f. Extract, print and check the RDF metadata
Dataset [7.1.1]	Validated P-VDI ConvergenceMetadata file
Output	Prints the inner elements of the given ConvergenceMetadata XML fragment
Expected result	File parsing as well as intermediate schemahandler parsing must be carried out without any exceptions and lead to the successful extraction and check of the inner elements of the given ConvergenceMetadata XML fragment
Variants	*This test has been implemented for parsing all kinds of*

	ConvergenceMetadata.

2.4 DigitalItem TE

The DigitalItem TE offers functionalities for creating and parsing DI XML fragments. The test cases below verify the proper functioning of the engine both during creation and parsing of digital items and its inner components.

2.4.1 DI creation

Name of test case	testDICreation
Initial conditions	*Validated DI file*
Description of testing functionality	This test case verifies the proper creation of a Digital Item XML fragment. The XML fragment is stored on disk and is checked with the DI file in the dataset.
Engines involved	DigitalItem TE, EventReport TE, IPMP TE, REL TE, Security TE, Metadata TE
Detailed steps	1. Create Item1 a. Set its identifier b. Add a reference to a resource c. Add a license about the resource 2. Create Item2 a. Set its identifier b. Add MPEG-7 metadata 3. Create TopLevelItem a. Set its identifier b. Add a related identifier c. Add a reference to a resource d. Add an event report request about the resource e. Add Item1 and Item2 as inner items f. Embed its signature 4. Create DigitalItem a. Set TopLevelItem as its top level item b. Embed its signature 5. Store the DigitalItem to disk 6. Check if the stored DigitalItem and the dataset DI file are identical
Dataset [7.2.1]	Validated DI file
Output	DI file
Expected result	All intermediate schemahandler objects must be created without any exceptions and lead to the successful

	generation and storage of the Digital Item. The file created must be identical with the already validated DI file in the dataset.
Variants	*N/A*

2.4.2 DI parsing

Name of test case	testDIParsing
Initial conditions	*Validated DI file*
Description of testing functionality	This test case verifies the proper parsing of a Digital Item XML fragment, stored in a local file. During the test, information inside the Digital Item is printed and checked against expected values.
Engines involved	DigitalItem TE, EventReport TE, IPMP TE, REL TE, Security TE, Metadata TE
Detailed steps	1. Parse the local file and create a DigitalItem a. Get its TopLevelItem b. Check if it has and print its Signature 2. Parse TopLevelItem a. Check and print its identifier b. Check and print its MPEG-7 metadata c. Check and print its EventReport d. Check and print its reference to a resource e. Check and print its Signature f. Extract its InnerItems 3. For each InnerItem a. Check and print its identifier b. Check and print its related identifier c. Check and print its metadata d. Check and print its reference to a resource e. Check and print its licenses
Dataset	Validated DI file
Output	Prints information embedded in the DigitalItem
Expected result	The engine should always parse the file and the intermediate schemahandler. The parsing should lead to the correct extraction and verification of information stored in the DigitalItem.
Variants	*N/A*

2.5 EventReport TE

2.5.1 Event report creation

Name of test case	testERGeneration
Initial conditions	ER file
Description of testing functionality	This test case uses dummy data to verify the creation and storage of an event report (ER) object
Engines involved	EREngine
Detailed steps	1. Create ER 2. Store ER
Dataset [7.3.1]	ER file
Output	ER file
Expected result	The output ER file should match the input; the test should not throw any exception
Variants	*N/A*

2.5.2 Event report parsing

Name of test case	testERParsing
Initial conditions	XML event report (ER)
Description of testing functionality	This test case loads and parses an event report (ER)
Engines involved	EREngine
Detailed steps	1. Load ER 2. Parse ER
Dataset	ER file
Output	Prints the information contained in the ER
Expected result	The engine should parse and verify the ER field without throwing any exception
Variants	*N/A*

2.5.3 Event report request creation

Name of test case	testERRGeneration

Initial conditions	ERR file
Description of testing functionality	This test case uses dummy data to verify the creation and storage of an event report request (ERR) object
Engines involved	EREngine
Detailed steps	1. Create ERR 2. Store ERR
Dataset [7.3.2]	ERR file
Output	ERR file
Expected result	The output file should match the input; the test should never throw an exception
Variants	*N/A*

2.5.4 Event report request parsing

Name of test case	testERRParsing
Initial conditions	XML event report request(ERR)
Description of testing functionality	This test case loads and parses an event report request (ERR)
Engines involved	EREngine
Detailed steps	1. load ERR 2. parse ERR
Dataset	ERR file
Output	Prints information of the ERR
Expected result	The engine should parse the ERR field. The test should never throw an exception
Variants	*N/A*

2.5.5 Event report storage and retrieval

Name of test case	testGetMultipleERsByERRId
Initial conditions	ER repository may be empty or have some ERs/ERRs registered
Description of testing functionality	This test case checks storage and retrieval of ERRs in the EREngine internal repository

Engines involved	EREngine
Detailed steps	1. Create ERR 2. Store ERR 3. Create ER 1 (ERRIdembedeed) 4. Store ER 1 5. Create ER 2 (ERRIdembedeed) 6. Store ER 2 7. Create ER 3 (ERRIdembedeed; same values as ER 1) 8. Store ER 3 9. Get list of ERIds by ERRId 10. Get list of ERIds by DIDLid (store inside ER 1)
Dataset	<none>
Output	Prints successful messages at each step. Prints the ERs that have been retrieved.
Expected result	ER objects should be registered in the ER repository. When the ER repository is queried using ERRId or DIDLId the engine should return the expected list of ERs.
Variants	*N/A*

2.6 IPMP TE

The Intellectual Property Management and Protection TE (IPMP TE) offers functionality for creating and parsing IPMP XML fragments. The test cases below verify the proper functioning of the engine during both creation and parsing.

2.6.1 IPMP descriptor creation

Name of test case	testIPMPGeneration
Initial conditions	*Validated IPMP file*
Description of testing functionality	This test case verifies the proper creation of an Intellectual Property Management and Protection XML fragment. The XML fragment is stored on disk. The engine checks the fragment against the IPMP file in the dataset.
Engines involved	IPMP TE, REL TE
Detailed steps	1. Create a new REL license 2. Embed the REL license in a new IPMP license 3. Embed the IPMP license in a new Rights Descriptor 4. Add the Rights Descriptor in a new License Collection 5. Add the License Collection in a IPMP Descriptor 6. Store the IPMP Descriptor to disk 7. Check if the stored IPMP Descriptor and the dataset

	IPMP file are identical
Dataset [7.4.1]	Validated IPMP file
Output	IPMP file
Expected result	All intermediate schemahandler objects must be created without any exceptions and lead to the successful generation and storage of the IPMP description. The file created must be identical with the already validated IPMP file in the dataset.
Variants	*N/A*

2.6.2 IPMP descriptor parsing

Name of test case	testIPMPParsing
Initial conditions	*Validated IPMP file*
Description of testing functionality	This test case verifies the proper parsing of an Intellectual Property Management and Protection XML fragment. The XML fragment is read from a local file and parsed in order to extract the Signature of the Issuer of the embedded REL License.
Engines involved	IPMP TE, REL TE, Security TE
Detailed steps	1. Parse local file as an IPMP Descriptor 2. Parse the License Collection of the IPMP Descriptor 3. Parse the Rights Descriptor of the License Collection 4. Parse the IPMP License embedded in the Rights Descriptor 5. Parse the REL License embedded in the IPMP License 6. Extract the Issuer of the REL License and print the signature embedded by him
Dataset	Validated IPMP file
Output	Prints the signature embedded by the issuer.
Expected result	File parsing as well as intermediate schemahandler parsing must be carried out without any exceptions and lead to the successful extraction of issuer's signature.
Variants	*N/A*

2.7 Match TE

The Match TE is responsible for maintaining metadata and query repositories for publications and subscriptions. Moreover, it is capable to match incoming publications with existing

subscriptions and vice versa. The Match TE supports matching of field/value query subscriptions with field/value publication metadata, SPARQL query subscriptions with RDF publications as well as mixed queries with both field value conditions and SPARQL query. The test cases below verify the proper functioning of the engine under various matching scenarios.

2.7.1 Matching subscription over existing publications

Name of test case	testMatchEngineResultsValiditySubOverPub
Initial conditions	*<none>*
Description of testing functionality	This test case verifies the proper matching of incoming subscriptions against existing publications. The test starts by creating various publications and then all possible subscriptions, including subscriptions containing inequality conditions. It then checks the results returned by the engine against the expected results.
Engines involved	Match TE
Detailed steps	1. Create various publications a. Set id b. Set metadata 2. Inform engine of publications 3. Create various subscriptions a. Set id b. Set query conditions 4. Inform engine of subscriptions 5. Print and check matches detected with expected results.
Dataset	<none>
Output	Prints matching subscriptions and publications
Expected result	The matches between subscriptions and publications found by the engine should match the expected results.
Variants	*This test has been implemented with all possible metadata types and query types.*

2.7.2 Matching publication over existing subscriptions

Name of test case	testMatchEngineResultsValidityPubOverSub
Initial conditions	*<none>*
Description of	This test case verifies the proper matching of incoming

testing functionality	publications against existing subscriptions. The test starts by creating all possible subscriptions, including subscriptions with inequality conditions, and then creates various publications and checks the results returned against the expected results.
Engines involved	Match TE
Detailed steps	1. Create various subscriptions a. Set id b. Set query conditions 2. Inform engine of subscriptions 3. Create various publications a. Set id b. Set metadata 4. Inform engine of publications 5. Print and check matches detected with expected results.
Dataset	<none>
Output	Prints matching subscriptions and publications
Expected result	The matches between subscriptions and publications found by the engine should match the expected results.
Variants	*This test has been implemented for all possible metadata types and query types.*

2.8 Metadata TE

The Metadata TE offers functionalities for creating and parsing MPEG-7 XML metadata fragments. In the MPEG-M middleware, MPEG-7 is used in several different contexts and not just to describe resources. The Metadata TE thus plays a fundamental role in the middleware. The test cases below verify the proper functioning of the engine during the creation and parsing of MPEG-7 metadata.

2.8.1 MPEG-7 metadata creation

Name of test case	testMetadataCreation
Initial conditions	*Validated MPEG-7 metadata file*
Description of testing functionality	This test case verifies the proper creation of a MPEG-7 XML fragment. The XML fragment is stored on disk and is checked against the metadata file in the dataset.
Engines involved	Metadata TE
Detailed steps	1. Create MPEG-7 metadata

	a. Add a Title b. Add an Abstract Textual description 2. Store MPEG-7 file to disk. 3. Check if the stored MPEG-7 and the dataset's MPEG-7 metadata files are identical
Dataset [7.5.1]	Validated MPEG-7 file
Output	MPEG-7 metadata file
Expected result	The engine must create all intermediate schemahandler objects without ever throwing an exception. The end result should be the successful generation and storage of the MPEG-7 XML fragment. The file created must be identical to the pre-validated MPEG-7 metadata file in the dataset.
Variants	*N/A*

2.8.2 MPEG-7 metadata parsing

Name of test case	testMpeg7Parsing
Initial conditions	*Validated MPEG-7 metadata file*
Description of testing functionality	This test case verifies the proper parsing of a MPEG-7 XML metadata fragment, stored in a local file. During the test, information inside the MPEG-7 metadata is printed and checked against expected values.
Engines involved	Metadata TE
Detailed steps	1. Parse the local file and create MPEG-7 Metadata a. Check and print the title b. Check and print the abstract description
Dataset [7.5.1]	Validated MPEG-7 file
Output	Prints information embedded in the MPEG-7 metadata file
Expected result	The engine must parse files and intermediate schemahandlers without ever throwing an exception. The end result should be the successful extraction and check of information inside the MPEG-7 metadata file.
Variants	*N/A*

2.9 MP21FF TE

The MPEG-21 File Format TE (MP21FF TE) offers functionalities for creating and parsing MPEG-21 files with embedded resources. The test cases below verify the proper functioning of the engine during creation and parsing.

2.9.1 DCF creation

Name of test case	testDCFCreation
Initial conditions	DCI file name
Description of testing functionality	This test case verifies the creation of a MP21 file. The given name and the .mp21 extension are stored on disk. If the previous operation is successful the engine returns the local path for the file that has just been produced. If the operation fails, the engine returns a null value.
Engines involved	MPEG21ff TE
Detailed steps	1. Creates a new MPEG21 File Engine and File Creator 2. Set MPEG21 file name 3. Add resource to MPEG21 file 4. Store in disk
Dataset	DCI file, MPEG21 file
Output	MPEG21 file
Expected result	Output MPEG21 file shall contain the desired resource and be valid.
Variants	*N/A*

2.9.2 DCF parsing

Name of test case	testDCFParsing
Initial conditions	MPEG21 file name
Description of testing functionality	This test case verifies if the MPEG21 file was successfully created and if the MPEG21 file is valid.
Engines involved	MPEG21ff TE
Detailed steps	1. Creates a new MPEG21 File Parser 2. Extract the DCI and compare it to the original 3. Extract the Resource and compare it to the original 4. Verifies if the MPEG21 file was successfully parsed
Dataset	MPEG21 file
Output	MPEG21 file
Expected result	The output MPEG21 file is parsed successfully and is valid.
Variants	*N/A*

2.10 MPQF TE

The MPEG Query Format TE (MPQF TE) extends the functionalities of the Search TE, offering methods to create and parse custom Field/Value and SPARQL queries based on the MPQF schema. The test cases below verify the proper functioning of the engine during the creation and parsing of MPQF queries.

2.10.1 Mixed MPQF query creation

Name of test case	testMixedFieldValueAndSparqlCreation
Initial conditions	*Validated MPQF query file*
Description of testing functionality	This test case verifies the proper creation of a MPQF query XML fragment. The XML fragment is stored on disk and is checked against the query file in the dataset.
Engines involved	Search TE, MPQF TE
Detailed steps	1. Create FieldValue QueryCondition1 a. Set Field b. Set Value 2. Create SPARQL QueryCondition2 a. Set bound variable b. Set predicate c. Set object 3. Create OR QueryCondition a. Add QueryCondition1 and QueryCondition2 4. Create MPQF query a. Set as condition OR QueryCondition 5. Store MPQF query file to disk. 6. Check if the stored query and the dataset's MPQF query files are identical
Dataset [7.6.1]	Validated MPQF query file
Output	MPQF query file
Expected result	Intermediate schemahandler objects must be created without any exceptions, leading to the successful generation and storage of the MPQF query XML fragment. The file created must be identical to the prevalidated MPQF query file in the dataset.
Variants	*N/A*

2.10.2 Mixed MPQF query parsing

Name of test case	testMixedFieldValueAndSparqlParsing

Initial conditions	*Validated MPQF query file*
Description of testing functionality	This test case verifies the proper parsing of a mixed field/value and SPARQL MPQF query, stored in a local file. During the test, the conditions of the query are printed and checked against expected values.
Engines involved	SearchTE, MPQF TE
Detailed steps	1. Parse the local file and create MPQF Query a. Extract recurrently embedded QueryConditions 2. For each QueryCondition a. Check and print the conditions
Dataset [7.6.1]	Validated MPQF query file
Output	Prints the query conditions for the MPQF query file
Expected result	The engine parses files and intermediate schemahandlers without ever throwing an exception. The end result will be the successful extraction and check of conditions inside the MPQF query dataset file.
Variants	*N/A*

2.11 Overlay TE

The Overlay TE is the engine responsible for maintaining the overlay network of CONVERGENCE peers and controlling communications and propagation of messages between peers. The test cases below present the testing framework for the OverlayTE. The distributed nature of the engine required the deployment of multiple remote peers exchanging publication and subscription messages, while dynamically entering and leaving the overlay network. During the testing, therefore, multiple CONVERGENCE developers deployed CONVERGENCE peers and ran stand-alone applications to control publications and subscriptions and monitor the receipt of messages and any changes in the registry.

2.11.1 Registry update

Name of test case	testRegistryUpdateValidity
Initial conditions	*Peers connect to a pre-agreed fractal which doesn't exist on the beginning of the test.*
Description of testing functionality	This test case verifies the proper update of a pre-agreed, but not pre-existing, fractal registry. When a peer enters the fractal, it announces itself to $\log_2 N$ peers, when N is the number of peers in the fractal. Each peer should thus be

	included in $\log_2 N$ registries. For small networks with less than four peers, each peer announces itself to all other peers in the registry.
Engines involved	OverlayTE, ConetTE
Detailed steps	**Phase 1. Entry of stable peers in the fractal**. 1. PeerA creates a publication for the fractal 2. PeerA checks if its local registry file exists and if it includes PeerA. 3. … for 3 more peers 4. PeerE creates a publication for the fractal 5. PeerE should be included in floor($\log_2 5$) = 2 registries 6. … for at least 3 peers 7. PeerX should be included in floor($\log_2 N$) registries. **Phase 2. Peers lose connection** 1. PeerY loses connectivity 2. … even if $\log_2 N$ peers lose connectivity at least 1 registry includes peerY
Dataset	<none>
Output	Local registry files
Expected result	Registries are updated according to the overlay protocol, which ensures that even after $\log_2 N$ connections have dropped, each peer is still included in at least one registry.
Variants	*N/A*

2.11.2 Publication and subscription propagation

Name of test case	testPublicationAndSubscriptionPropagation
Initial conditions	*At least 5 peers are connected to a pre-agreed fractal*
Description of testing functionality	This test case verifies the proper propagation of messages in the overlay network. The test starts with at least 5 peers in the fractal. $\log_2 N$ peers then lose their connectivity. If a peer sends a publication or a subscription message it is always received by at least one peer. The Overlay Engine maintains the overlay, ensuring that all peers are in communication with at least one other peer.
Engines involved	OverlayTE, ConetTE
Detailed steps	1. At least 5 peers enter the fractal 2. … $\log_2 N$ peers lose connectivity. 3. PeerX sends a publication or a subscription.

	4. At least one peer receives it.
Dataset	<none>
Output	Print in at least one peer that it received the publication
Expected result	The Overlay Engine maintains the overlay network correctly ensuring that the most recent publication is always received by at least one peer.
Variants	*N/A*

2.12 REL TE

The Rights Expression Language TE (REL TE) offers functionalities for creating and parsing REL licenses, for checking REL licenses and for authorizing principals to execute a right over a resource, in compliance with the conditions fixed in the license. The test cases below verify the proper functioning of the engine during the creation, parsing and authorization of licenses.

2.12.1 REL license creation

Name of test case	testLicenseCreation
Initial conditions	*Validated REL license file*
Description of testing functionality	This test case verifies the proper creation of a REL license. The license is stored on disk and checked against the REL license file in the dataset.
Engines involved	REL TE, Security TE
Detailed steps	1. Create KeyHolder1 and KeyHolder2 a. Specify their public keys b. Set their alias 2. Create a Reference to a DI a. Specify the DI identifier b. Set an alias 3. Create a ValidityIntervalCondition a. Set notBefore date b. Set notAfter date 4. Create Grant1 and Grant2 a. Set as principal KeyHolder1 and KeyHolder2 b. Set rights c. Set as resource the Reference to a DI d. Set as condition the ValidityIntervalCondition 5. Create a License a. Add Grant1 and Grant2 b. Set as Issuer KeyHolder1 6. Store License to disk 7. Check if the stored license and the dataset's REL li-

	cense files are identical
Dataset [7.7.1]	Validated REL license file
Output	REL license file
Expected result	The engine must create all intermediate schemahandler objects without ever throwing an exception. This should lead to the successful generation and storage of the license. The file created must be identical with the pre-validated REL license file in the dataset.
Variants	*This test has many variants, including the specification of additional grants in the license, and the specification of different conditions, rights and resources.*

2.12.2 REL license parsing

Name of test case	testLicenseParsing
Initial conditions	*Validated REL license file*
Description of testing functionality	This test case verifies the proper parsing of a REL license, stored in a local file. During the test, all grants included in the license are parsed, printed and checked against expected values for principal, right, resource and conditions.
Engines involved	REL TE, Security TE
Detailed steps	1. Parse the local file and create a REL License a. Check and print its identifier b. Extract its Issuer c. Check and print Issuer public key d. Extract its Grants 2. For each Grant a. Check and print principal b. Check and print REL right c. Check and print resource d. Check and print condition
Dataset	Validated REL license file
Output	Prints the grants expressed in the REL license
Expected result	The engine must parse files and intermediate schemahandlers without ever throwing an exception, leading to the successful extraction and verification of license grants and their components.
Variants	*This test has many variants, including the use of licenses with different numbers of grants, different conditions, different*

	rights and different resources.

2.12.3 REL query authorization

Name of test case	testAuthorization
Initial conditions	*REL license file*
Description of testing functionality	This test case verifies the authorization functionalities of the REL engine. The test uses an input license file that governs a given resource and queries the REL engine to authorize principals to execute a right against the resource under different conditions. The authorization results are checked and printed.
Engines involved	REL TE, Security TE
Detailed steps	1. Parse the local file and create a REL License 2. Create License queries a. Set interested principal b. Set interested resource c. Set right d. Set conditions 3. Check and print authorization results
Dataset	REL license file
Output	Prints the authorization query results
Expected result	The engine must parse the license without throwing any exceptions. Subsequent queries to the REL engine must return the expected results.
Variants	*This test has many variants, including the use of licenses with different numbers of grants, different conditions, different rights and different resources.*

2.13 Security TE

The Security TE provides cryptographic functionalities to the CONVERGENCE middleware, including symmetric and asymmetric encryption/decryption, key pair generation, certificate generation, signature signing and verification. The engine provides a pure software solution and a smart card based solution. The tests below verify the proper functioning of the engine with and without the use of smart cards.

2.13.1 Certificate and key pair generation

Name of test case	testCertificateManager

Description of testing functionality	Tests the generation of certificates and RSA key pairs and checks their validity.
Precondition	Pin number set via set property method
Engines involved	Security TE
Detailed steps	1. Two 2048 byte RSA-key pairs are generated and stored in different locations on the card or on the hard drive 2. The two public keys are read from the card and checked against the keys that have just been generated 3. A `X509Certificate` is generated using a specially-generated key pair 4. The certificate is verified
Output	Certificate, RSA key pair
Expected result	Certificate and key pair should be generated correctly

2.13.2 Hybrid encryption and decryption

Name of test case	testKeyManager
Description of testing functionality	Tests the hybrid-encryption and decryption mechanism with a random message
Precondition	Pin number set via set property method
Engines involved	Security TE
Detailed steps	1. Generate random message and random symmetric AES key 2. Encrypt (wrap) generated AES key with user's public key 3. Decrypt (unwrap) generated AES key with user's private key 4. Check if the unwrapped key matches the previously generated one 5. Message is encrypted using unwrapped key 6. Message is decrypted using unwrapped key
Output	AES key for encryption, decryption
Expected result	Unwrapped key matches generated key, decrypted message matches the generated one

2.13.3 Signature creation and verification

Name of test case	testSignatureManager

Description of testing functionality	Tests the signing of a message using a specially generated signature
Precondition	Pin number set via set property method
Engines involved	Security TE
Detailed steps	1. Check if the “auth”, “conf” and “sig” key pairs are present. If there is no signature key pair it is generated 2. A random message is generated 3. A signature is generated for the random message 4. The random message is signed using the generated signature 5. The signature is verified using the public key of the user 6. Verify signature
Output	Signature
Expected result	Signature is verified

2.14 VDI TE

The Versatile Digital Item TE (VDI TE) extends the functionalities of the Digital Item TE to support creation and parsing of Relationship XML fragments describing the relations between DigitalItems. The test cases below verify the proper functioning of the engine during the creation and parsing of digital items and the relationship information they contain.

2.14.1 VDI creation

Name of test case	testVDICreation
Initial conditions	*Validated VDI file*
Description of testing functionality	This test case verifies the proper creation of a Relationship element and its addition in a new Versatile Digital Item. The VDI is stored on disk and is checked against the VDI file in the dataset.
Engines involved	VDI TE, DigitalItem TE
Detailed steps	1. Create TopLevelItem a. Set its identifier b. Add a relationship between another digital item by i. Setting the ontology URI ii. Setting the subject DI iii. Setting the relationship type

	iv. Setting the object DI 2. Create VersatileDigitalItem a. Set TopLevelItem as its top level item 3. Store the VersatileDigitalItem to disk 4. Check if the stored VersatileDigitalItem and the dataset VDI file are identical
Dataset [7.8.1]	Validated VDI file
Output	VDI file
Expected result	All intermediate schemahandler objects must be created without any exceptions and lead to the successful generation and storage of the Versatile Digital Item. The file created must be identical with the already validated VDI file in the dataset.
Variants	*N/A*

2.14.2 VDI parsing

Name of test case	testVDIParsing
Initial conditions	*Validated VDI file*
Description of testing functionality	This test case verifies the proper parsing of a Versatile Digital Item XML fragment which contains information about its relationships with other Digital Items and is stored in a local file. During the test, information about the aforementioned relationships is printed and checked against expected values.
Engines involved	VDI TE, DigitalItem TE
Detailed steps	1. Parse the local file and create a VersatileDigitalItem a. Get its TopLevelItem 2. Parse TopLevelItem a. Check and print its identifier b. Extract its Relationships 3. For each Relationship a. Check and print its ontologyURI b. Check and print its subject DI c. Check and print its relationship type d. Check and print its object DI
Dataset	Validated VDI file
Output	Prints information embedded in the Relationships element of the VersatileDigitalItem
Expected result	The engine must parse files and intermediate

	schemahandlers without ever throwing an exception. This should lead to the successful extraction and verification of relationship information inside the VersatileDigitalItem.
Variants	*N/A*

3 Integrated orchestration testing

This chapter describes the testing of the orchestrators, which the CONVERGENCE framework uses to support middleware protocols (e.g. Revoke Content), and to handle incoming events (e.g. a VdiArrivalEvent). Each of the tests takes the form of a small application with checkpoints to verify the correct execution of the workflow.

3.1 Middleware demo

This small application tests the core functionalities provided by the CONVERGENCE middleware, namely creating and parsing a license, creating RDF metadata, publishing a resource and subscribing with a matching SPARQL query. The table below lists the steps in the workflow.

Name of application	MiddlewareDemo
Initial conditions	<none>
Description of testing functionality	This application is divided into four phases. In the preparation phase, the application initializes the middleware and creates a license for a resource. In the creation phase, it creates an R-VDI containing the license and metadata for the resource. In the publication phase it publishes the VDI together with an event report request. Finally, in the subscription phase, it creates a subscription to be matched against the publication created previously. Notification of the match demonstrates the proper functioning of the protocol, orchestrator and technology engines, and closes the application.
Engines involved	CreateLicensePE, IdentifyContentPE, CreateContentPE, CreateLicenseOrchestrator, IdentifyContentOrchestrator, CreateContentOrchestrator, MetadataOrchestrator, OverlayEngineOrchestrator, VdiOrchestrator, ConetEngineOrchestrator, MatchEngineOrchestrator, EREngineOrchestrator, REL TE, ConvergenceMetadata TE, IPMP TE, VDI TE, Overlay TE, Conet TE, EventReport TE, Match TE

Detailed steps	**Phase 1. Prepare**
	1. Initialize the middleware
	  1. Load all middleware engines
	  2. Startup Match and EventReport repositories
	  3. Schedule action for overlay registries update
	2. Startup listener for Match events
	3. Create License with CreateLicensePE
	  1. Set Principal
	  2. Set resource reference
	  3. Add grants

	4. Set conditions 4. Check created license with REL license in dataset. **Phase 2. Create** 1. Create ConvergenceMetadata 1. Set keywords 2. Set expiry date 2. Create R-VDI with CreateContentPE 1. Set ConvergenceMetadata 2. Set License 3. Check created R-VDI with the R-VDI in dataset. 4. Identify R-VDI with IdentifyContentPE 5. Check and print the returned R-VDI identifier **Phase 3. Publish** 1. Parse R-VDI and extract ConvergenceMetadata 2. Create P-VDI with CreateContentPE 1. Add ConvergenceMetadata 2. Set fractal 3. Check created P-VDI with the P-VDI in dataset 4. Identify P-VDI with IdentifyContentPE 5. Check and print the returned P-VDI identifier 6. Publish the P-VDI **Phase 4. Subscribe** 1. Create an EventReportRequest for match 2. Create ConvergenceMetadata 1. Set keywords conditions 2. Set expiry date 3. Create S-VDI with CreateContentPE 1. Add the EventReportRequest 2. Add ConvergenceMetadata 3. Set fractal 4. Check created S-VDI with the S-VDI in dataset 5. Identify S-VDI with IdentifyContentPE 6. Check and print the returned S-VDI identifier 7. Publish the S-VDI 8. Exit when the Listener returns a notification
Dataset [7.9]	REL license file, R-VDI file, P-VDI file, S-VDI file
Output	The application prints the results of the various checkpoints.
Expected result	The application should end with the arrival of a notification arriving at the peer. If there is an exception or the application detects a fault, it prints an error message and exits.
Variants	*N/A*

3.2 RDF and SPARQL app

This small application tests the matching functionalities provided by the CONVERGENCE middleware and, in particular the proper handling of mixed subscription queries and mixed publication metadata. The table below lists the steps in the workflow.

Name of application	RDFandSPARQLapp
Initial conditions	<none>
Description of testing functionality	This application is divided into five phases. In the preparation phase, the application initializes the middleware. In the creation phase, it creates an R-VDI containing a mixed description with both RDF and keyword metadata about the resource. In the publication phase it publishes the VDI. In the subscription phase, it creates six subscriptions of which four should match the publication created earlier. Finally, in the re-publish phase, it republishes the R-VDI, which again should match the subscriptions created previously. When it receives notifications for these matches, the application exits automatically, first verifying the proper matching procedure and the correct functioning of the protocol, orchestrator and technology engines.
Engines involved	IdentifyContentPE, CreateContentPE, IdentifyContentOrchestrator, CreateContentOrchestrator, MetadataOrchestrator, OverlayEngineOrchestrator, VdiOrchestrator, ConetEngineOrchestrator, MatchEngineOrchestrator, EREngineOrchestrator, ConvergenceMetadata TE, VDI TE, Overlay TE, Conet TE, EventReport TE, Match TE

Detailed steps	**Phase 1. Prepare**
	1. Initialize the middleware
	  1. Load all middleware engines
	  2. Startup Match and EventReport repositories
	  3. Schedule action for overlay registries update
	2. Startup listener for Match events
	Phase 2. Create
	1. Parse RDF file and wrap them in StructuredData
	2. Create ConvergenceMetadata
	  1. Add StructuredData
	  2. Add keywords
	  3. Set expiry date
	3. Create R-VDI with CreateContentPE
	  1. Set ConvergenceMetadata
	4. Check created R-VDI with the R-VDI in dataset.
	5. Identify R-VDI with IdentifyContentPE
	6. Check and print the returned R-VDI identifier
	Phase 3. Publish
	1. Parse R-VDI and extract ConvergenceMetadata
	2. Create P-VDI with CreateContentPE
	  1. Add ConvergenceMetadata
	  2. Set fractal
	3. Check created P-VDI with the P-VDI in dataset
	4. Identify P-VDI with IdentifyContentPE

	5. Check and print the returned P-VDI identifier 6. Publish the P-VDI **Phase 4. Subscribe** 1. Create 6 S-VDIs 1. Create and set EventReportRequest for match 2. Create and set ConvergenceMetadata 1. S-VDI-1: Matching field/value subscription 2. S-VDI-2: Matching SPARQL subscription 3. S-VDI-3: Matching field/value and SPARQL subscription 4. S-VDI-4: Non-matching field/value and SPARQL subscription 5. S-VDI-5: Matching field/value, non-matching SPARQL subscription 6. S-VDI-6: Non-matching field/value, matching SPARQL subscription 2. Check created S-VDIs with the S-VDIs in dataset 3. Identify S-VDIs with IdentifyContentPE 4. Check and print the returned S-VDI identifiers 5. Publish the S-VDIs 6. Wait for notifications for S-VDI-1, S-VDI-2, S-VDI-3, S-VDI-5, S-VDI-6 **Phase 5. Re-publish** 1. Parse R-VDI and extract ConvergenceMetadata 2. Create P-VDI with CreateContentPE 1. Add ConvergenceMetadata 2. Set fractal 3. Check created P-VDI with the P-VDI in dataset 4. Identify P-VDI with IdentifyContentPE 5. Check and print the returned P-VDI identifier 6. Publish the P-VDI 7. Wait for notifications for S-VDI-1, S-VDI-2, S-VDI-3, S-VDI-5, S-VDI-6 and exit
Dataset [7.10]	R-VDI file, P-VDI file, S-VDI-1 file, S-VDI-2 file, S-VDI-3 file, S-VDI-4 file, S-VDI-5 file, S-VDI-6 file
Output	The application prints the results of the various checkpoints.
Expected result	When the application receives the notifications that have arrived at the peer, it exits. If there is an exception or the application detects a fault, it prints an error message and exits
Variants	*N/A*

3.3 Revoke app

This small application tests the revocation functionalities offered by the CONVERGENCE middleware. The table below lists the steps in the workflow.

Name of application	RevokeApp
Initial conditions	VDI to be revoked
Description of testing functionality	This application verifies the revocation of a VDI from the CoMid and the CoNet.
Engines involved	Revoke PE, VDI TE, Overlay TE, CoNet TE
Detailed steps	**Phase 1. Unpublish from CoMid.** 1. Process Digital Item 2. Extract Fractals 3. Create unpublish VDI 4. Create a new Gorest Message 5. Set fractals 6. Check propagation of unpublish VDI over Overlay Engine **Phase 2. Revoke from CoNet** 1. Creation of NID object 2. Check the successful Store and Advertise of named-resource
Dataset	VDI File
Output	Print each phase checks.
Expected result	If both phases are completed successfully, the test is deemed to have been successful.
Variants	*N/A*

4 Network testing

4.1 Introduction

Network testing involves tests of four distinct aspects of the CONVERGENCE Information Centric Network (CONET):

i) The functions provided by the networking AP
ii) The underlying networking functionalities supporting the API
iii) Network performance in a laboratory scenario
iv) and Network performance in a PlanetLab scenario

Tests (i), (ii) and (iii) have already been described in deliverable D8.3, Section 5.1. The results of the PlanetLab test are reported in [6] and included in D8.4. To avoid duplication, therefore, we will limit the discussion to just a single aspect of the testing that has not been described elsewhere, namely the CONET toolkit used for the PlanetLab tests.

The CONET toolkit provides Linux/MacOs users with the software tools necessary to set up a single CONET Autonomous System, based on the Lookup-and-Cache architecture described in D5.3 (Section 4.2). For the sake of completeness we begin by briefly describing the architecture. We then go on to describe the toolkit.

4.2 The CONET Lookup and Cache architecture.

A CONET consists of different types of devices:

- Clients (or end-nodes)
- Servers (or serving-nodes)
- Nodes
- Name Routing System (NRS) node.

CONET devices (clients, server and nodes) are interconnected by "sub-systems". A sub-system can be a public or private IP network, an overlay IP/UDP link, a layer 2 network, a PPP link, etc.

A CONET Autonomous System is composed of CONET sub-systems, interconnected by CONET nodes and is logically served by a single Name Routing System (NRS) node. Fig. 1 depicts an Autonomous System, formed of 19 CONET (border) nodes, where each node serves a subsystem, with internal CONET clients and servers.

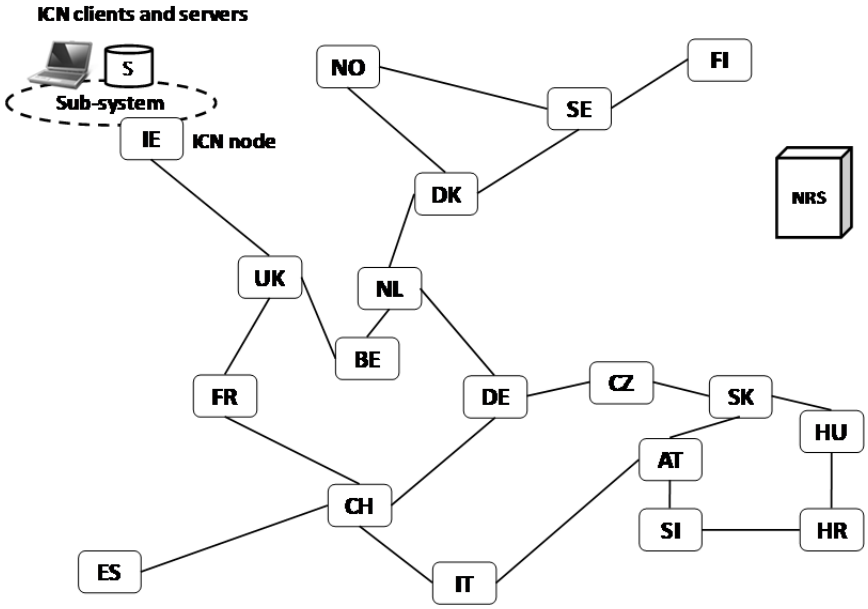

Fig. 1. CONET Autonomous System

CONET Servers are devices that publish contents (aka named-data), split them into blocks of data, called chunks or named-data CIUs, and assign a unique Network IDentifier (NID) to each chunk. The NID is a hierarchical name, in the form Principal/Label, as specified in D5.1 (section 5.4)

Clients are devices that request chunks by means of Interest messages (aka Interest CIUs).

CONET nodes are devices that route-by-name the Interest packets, using a longest prefix matching forwarding and a name-based routing table, usually called the Forwarding Information Base (FIB). A name-based route is an entry in the FIB and has the form `<name-prefix, next-hop>`.

A FIB has a limited size and, hence, may not contain all possible name-based routes. A route miss event occurs when an Interest message arrives to a CONET node and the FIB lacks the routing entry necessary to route-by-name the message. In this case, the CONET node *Lookups* the routing entry in a NRS node, *Caches* the entry in the FIB, according to a specific *cache replacement policy* (e.g. ITO and LRU, see D5.3), and then forwards the Interest message (see Fig. 2).

The NRS node is a centralized routing engine, containing all the possible name-based routes of each CONET nodes of the Autonomous System. These routes are contained in a data base, called Routing Information Base (RIB).

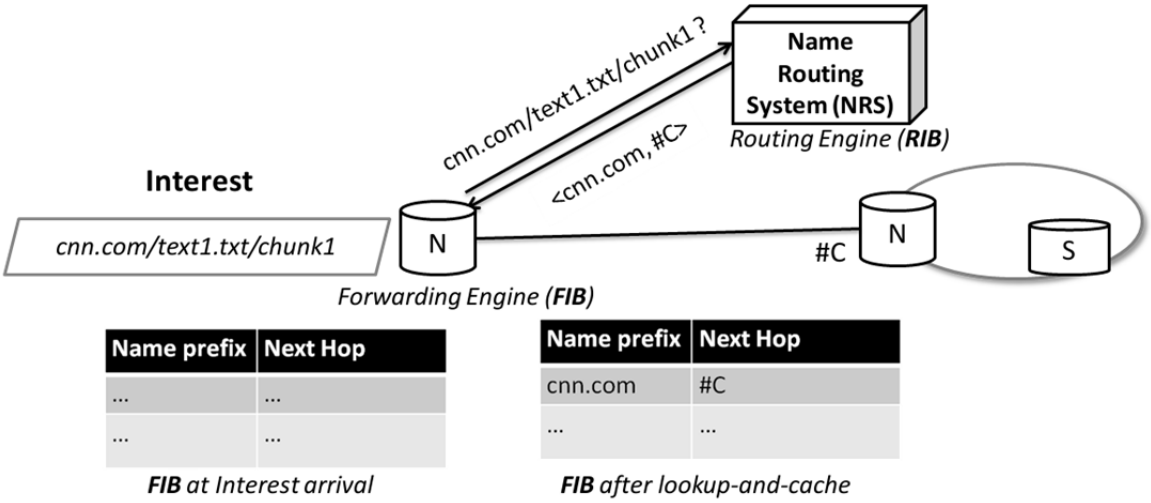

Fig. 2. CONET Lookup and Cache Architecture

4.3 The toolkit

The software toolkit (Fig. 3) makes it possible to set up an autonomous system. It contains the following modules:

- The CCNx package (www.ccnx.org), including a modified version of the CCND module
- The Lookup-and-Cache Server;
- The Name Routing System module.

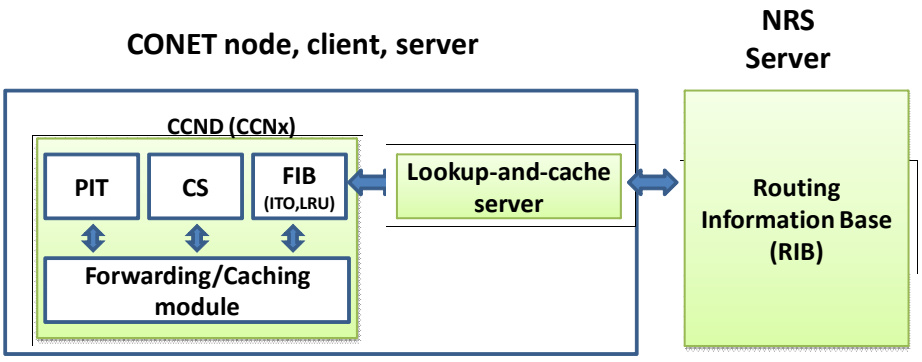

Fig. 3. Software modules in the CONET Toolkit

All CONET devices (clients, servers and nodes) run our version of CCNx, where the routing-by-name process is supported by the Lookup-and-Cache procedure, previously described.

In case of a route miss event, the CCND module notifies the miss to the local Lookup-and-cache Server, which lookups the route in the remote NRS node[1].

4.3.1 CCNx with Look-and-Cache functionality

The CCNx software, whose core functions are implemented in the daemon (CCND), provides the functionality necessary to:

- Request a named-data by name using Interest messages
- Route-by-name these messages toward the server that has published the named-data
- Deliver named-data chunks to the client

To carry out these functions, CCND manages three data structures:

- The Forwarding Information Base (FIB)
- The Pending Interest Table (PIT)
- The Content Store (CS)

In our implementation, we use the FIB as a *route cache*, implementing LRU and ITO cache replacement policies, as described in D5.3.
Once CCNx has been installed, the CCND environment variables are contained in the file ~/.ccnx/ccndrc. The toolkit implements a number of special control variables, which can be set as follows:

- `CCND_LC_ROUTING=1` (enable lookup-and-cache functionality)
- `CCND_LC_FIB_SIZE=x` (Limit the size of the FIB size to x)
- `CCND_LC_REP=1` (for LRU), `CCND_LC_REP=4` (for ITO) (Configure the route cache replacement policy)
- `CCND_LC_ISIT=x` (initial value in ms for the smoothed inter-arrival time; default=1000 – used only with ITO)
- `CCND_LC_MAX_FIB_BTIME` x (maximum pressuring time in ms; default =1000 – used only with ITO).

[1] We chose to decouple CCND and NRS by interposing the Lookup-and-cache server. This choice makes possible to implement NRS with different technologies (e.g., DNS, MySQL, etc.), without changing CCND implementation, but only changing the Lookup-and-Cache interface towards the NRS. For instance in the laboratory testbed described in D8.3 (Section 5.1.3.2.2) we use a DNS server as NRS, whereas in the Planetlab testbed, described in next D8.4, we used the Java based implementation of NRS included in the CONET package.

To run the CCND, fist one has to compile the CCNx package and then execute the command "`ccndstart`". To publish and retrieve contents, and to use CCNx API, refer to the documentation at http://www.ccnx.org/releases/latest/doc/.

4.3.2 Lookup-and-Cache Server

Lookup-and-Cache (LC) Server is a Java module, contained in the folder "conetToolkit/lc_server". The server takes as input a configuration file, named "conteRE.conf". The file contains the following parameters:

- `nre_type`
- `local_NRS`
- `nre_peers`
- `nre_parents`
- `remote_bg`
- `local_bg`
- `net_mask`

where:

- `nre_type` is the type of node on which the LC Server is running. If it is a CONET node the parameter should be set to "`host`" (i.e., `nre_type = host`)
- `local_NRS` is the IP address of the NRS that serves the node (e.g., `local_NRS = 193.1.201.27`)
- `nre_peers, nre_parents, remote_bg, local_bg` should be set to `NONE` (these parameters will be used in the next release of the software to support inter-domain routing)
- `net_mask` is the network IP address and mask (/32) of the node where the LC Server is running (e.g., `net_mask = 157.159.226.72 /32`)

To run the lookup-and-cache server, execute `java –jar lc_server.jar full_path_to_conetRE.conf nrs_port` (e.g. java –jar lc_server.jar /home/foo/conetRE.conf 9999).

4.3.3 Name Routing System (NRS)

The Name Routing System (NRS) serves all nodes contained in its Autonomous System (AS). NRS uses the topology of the AS to compute the shortest path from each node towards each name-prefix (e.g. cnn.com).

The NRS is implemented in a Java module (contained in the folder "/conetToolkit/nrs_project"). The software takes as input a configuration file named `hashtable.conf`. The file is structured as a sequence of rows in which each row has the following form:

```
name-prefix <ip-origin> <netmask-origin> <ip-nexthop> 0
```

where:

- `name-prefix` is a name-prefix, e.g. "/ccn.com/" (be careful to include the first and last slash character);
- `ip-origin` is a list of IP addresses for the nodes in the AS;
- `netmask-origin` is a list of netmask (e.g. /32) associated with the `ip-origin` list;
- `ip-nexthop` is a list, which contains the IP addresses of the next-hop for each AS node represented in the `ip-origin` list: e.g., in the case of `name-prefix,` the first IP address in `ip-origin` has as next-hop the first IP address in `ip-nexthop,` and so forth.

A typical row might be defined as follows.

```
/cnn.com/ 160.80.82.100 172.14.56.0 /32 /24 160.80.82.1 150.23.45.1
0
```

In this case, if the route towards the name-prefix "ccn.com" is requested from the node 160.80.82.100/32, then the next CONET node of the path has address 160.80.82.1. Otherwise, if the route towards the name-prefix "ccn.com" is requested from a node that belongs to the network 172.14.56.0/24, the next-hop is the CONET node with address 150.23.45.1.

When a query for a given name-prefix arrives at the NRS, the NRS node replies with a packet containing the *whole* related row. This information is parsed by the querying node (i.e., the lc_server), which singles out the proper next-hop address from `ip-nexthop`, on the base of its IP address, using traditional IP longest prefix matching applied on `ip-origin` and `netmask-origin`. This receiver-driven logic means that replies from the NRS node can be cached by intermediate nodes and reused for subsequent NRS queries, from other nodes.

In addition to hashtable.conf, the NRS software also uses a second file, called conetRE.conf, where:

- `nre_type=routing_engine`
- `local_NRS=NONE`
- `nre_peers, nre_parents, remote_bg, local_bg` must be set to `NONE` (these parameters will be used in the next release of the software, designed to support inter-domain routing)
- `net_mask` is the network IP address and mask (/32) of the node where the NRS Server is running (e.g. net_mask = 193.1.201.27 /32)

The $HOME directory has to contain the hashtable.conf file. To run the NRS software, execute `java -jar conet_routing_engine.jar` from a folder that also contains conetRE.conf.

Automatic generation of hashtable.conf via Matlab

It is possible to generate hashtable.conf using the Matlab script "computeNetworkRIB.m", contained in the folder conetToolkit/script/matlab/. The script takes as input three parameters:

- the adjacency matrix `A` for the ICN nodes served by NRS
- the vector `Index2IP`
- the cell array `Index2Prefixes`

`A` is a matrix *nxn*, where *n* is the number of AS nodes. The entry a_{ij} is equal to 1, if node *i* is connected with node *j*, otherwise the value is zero.

`IndexToIP` is a vector *1xn*, where the element a_j represents the IP address of the node *j*.

`IndexToNamePrefix` is a cell-array *1xn* where each cell contains a list of name-prefixes. Cell a_j contains the list of name-prefixes published by the node *j*.

The script computes the shortest path towards all the name-prefixes for each AS and gives as output the hashtable.conf file.

4.4 Deployment on the PlanetLab Testbed

Here we describe how to reproduce the Planetlab testbed, initially described in D8.4.

As shown in Fig. 1, the network topology is composed of nineteen Planetlab nodes and a Name Routing System node (NRS). Each node virtualizes the role of a Client, a Server and a simple Node.

The list of nodes that we have chosen for the testbed is contained in the file "/ccnToolkit/conf/geant.txt". The file is composed of three columns:

```
<node index, node name, node IP address>
```

The first nineteen nodes in the list are CONET nodes, the twentieth node is the NRS server.

Each node has a CCNx Repository (see CCNx documentation) in which it publishes named-data (i.e. contents). Each content has a unique name, composed of a name-prefix and a file name. The name-prefix is an actual domain-name, e.g. twitter.com, whereas the file name is a meaningless string, e.g. “c1.ccn”. In this case, the content name would thus be twitter.com/ c1.ccn.

In our test-bed each node handles twenty unique domain-names. For each domain-name, it publishes five items of content, each containing 500kB of data (e.g. “cnn.com/c1.ccn”, “cnn.com/c2.ccn”, “cnn.com/ c5.ccn”). In this way, the network contains 380 domain-names and 1900 items of content (231800 chunks), uniformly distributed among network nodes.

Each client generates 300 requests for contents with an inter-arrival time that follows a negative exponential distribution, with mean 4 seconds.

To select an item of content, a client first chooses a domain-name according to a Zipf distribution with alpha=1; then it randomly selects one of the 5 items of content associated with the selected domain-name.

When a node does not have a routing entry to route an Interest message, it makes a lookup request to the NRS.

4.4.1 Configuration of PlanetLab Nodes

To set up a CONET node compile and execute the ccnd of the "ccnx-0.0.5.lc" software and the Java software module "conetToolkit/lc_server". This procedure should be repeated for each node.

As explained in section 4.3.2, lc_server requires the conetRE.conf file to run properly. The configuration files for each node can be found in the "conetToolkit/conf/conetRE/" folder.

The conetRE1.conf file is the configuration file for the first node in"/conetToolkit/conf/geant.txt" list, conetRE2.conf is the file for the second node, and so forth.

4.4.1.1 Setting up the repository

Each CONET node has to set up a CCNx Repository (using `ccnr` command), which will contains the content published by the node.

Each node handles 20 unique domain-names. For each domain name, it publishes 5 items of content, each containing 500kB of data. We have generated 19 different bash script files, one for each node in the test-bed. These scripts are useful to put contents in their CCNx Repository.

To generate these script files we used a Matlab script "conetToolkit/script/Matlab/CreateRepoFile.m".

This takes as input the file:

- top-1m.txt

The file consists of a list of domain-names representing the name-prefixes used in the testbed.

The script outputs five different types of file:

- index2name.txt
- index2size.txt
- index2repo.txt
- 19 bash script file (Repo1.sh,...,Repo19.sh)
- 19 xml files (Repo1.xml,...,Repo19.xml)

where:

index2name.txt contains the list of all 1900 items of content to be published in the nineteen CONET nodes.

index2size.txt gives the size of each item of content, expressed in bytes.

index2repo.txt is a list where each row represents an item of content, described in the format:

```
<Content name, content size, node index>
```

(the fields specify the name, and size of each item of content and the index of the node where it is to be published).

The nineteen bash script files (Repo1.sh,...,Repo19.sh) and the nineteen xml files (Repo1.xml,..,Repo19.xml) make it possible to set up the repositories for each node. To do this, upload the bash script and xml file (e.g., Repo1.sh and Repo1.xml for the first node contained in "/conetToolkit/conf/geant.txt), and run the command `sh Repo<node-index>.sh`

4.4.2 Name Routing System configuration

The NRS node executes the conet_routing_engine.jar software, whose sources are contained in "/conetToolkit/nrs_project".

The first step is to generate a proper hashtable.conf file and to save it in the $HOME directory for the NRS node. As already mentioned, this requires three parameters:

- The adjacency matrix `A` for the ICN nodes served by NRS
- The vector `Index2IP`
- The cell array `Index2Prefixes`

The Matlab file "/conetToolkit/script/Matlab/geant.mat" and "/conetToolkit /script/Matlab/planetlab.mat" contain these data structures. So, running the script "/conetToolkit/script/Matlab/computeNetworkRIB.m" generates the hashtable.conf for the testbed.

To complete the NRS configuration upload the conetRE.conf file ("/conetToolkit/conf/conetRE/conetREnrs.conf"). This file should be placed in the same directory as conet_routing_engine.jar.

4.4.3 Start PlanetLab Test

In our Planetlab testbed we executed a series of measurements using different values for the system parameters, (FIB size, CCND cache size, etc.).

The same workloads were used for all tests. Each CONET node requested 300 items of content (selected according to a Zipf distribution with alpha=1), with an inter-arrival time that followed a negative exponential distribution, with average 4 s. The workload was generated using the C program, contained in "/conetToolkit/workload/workload_pl.c", which can be compiled using the makefile "/conetToolkit/workload/Makefile", which outputs the executable file workload_pl.

To execute the software, workload_pl requires the file index2name.txt in the same directory. When run, the program generates the content requests and writes the results in a file named time.txt. Each row in the file is written in the format

```
< number of the request, index of requested content, download time, start time, end time>
```

The NRS module also produces the file lc_nrs_time.txt. This logs the exact times when the NRS receives lookup requests.

5 Conclusions

This deliverable described the integration and testing framework for the CONVERGENCE middleware and network. We note the key role played in this work by the MAVEN project management system, which integrates testing in the build procedure and provides developers with instant notification of build failures caused by compilation or logical errors. During our work, this possibility made it possible to spot errors early on, reducing the need for debugging during integration. It also encouraged developers to design tests at an early stage in the development process and to continuously add new ones as they added new functionality to the system.

The framework described in this report has made it possible to validate the proper functionality of the middleware engines and network individually and in integrated scenarios. Many of the tools developed in this work have been contributed to the MPEG-M standard or used to extend existing reference software. The work of WP6 is now complete.

6 References

[1] ISO/IEC JTC 1/ SC 29/ WG 11 - Information Technology – Multimedia service platform technologies 23006.

[2] CONVERGENCE project. Deliverable D6.2. System architecture. http://www.ict-convergence.eu/deliverables

[3] Apache MAVEN – Software project management tool - http://MAVEN.apache.org/

[4] Wordnet – Lexical database for English - http://wordnet.princeton.edu/

[5] JUnit testing framework - http://www.junit.org/

[6] N. Blefari-Melazzi, A. Detti, M. Pomposini, "Measurement-based experimental research: methodology, experiments and tools" in Springer Lecture Notes in Computer Science (LNCS), vol. 7586, 2012.

[7] CONVERGENCE project. Deliverable D8.3. Report on trials and experimentations after cycle 2. http://www.ict-convergence.eu/deliverables

7 ANNEX A – Test cases datasets

7.1 ConvergenceMetadata TE

7.1.1 ConvergenceMetadata creation and parsing

P-VDI ConvergenceMetadata

```
<ns21:ConvergenceMetadata>
<ns21:SequenceIdentifier>urn:sequence_identifier:temp:bec2ca4e-c0ba-459b-ada3-
ffbdbb31ebec</ns21:SequenceIdentifier>
    <ns21:VDIkind>PVDI</ns21:VDIkind>
    <ns21:ExpiryDate>2012-09-28T04:56:20.023+03:00</ns21:ExpiryDate>
    <ns21:PublicationMetadata>
<ns21:Dimension>
    <ns21:FractalAlgebra>
        <ns21:Fractal>analysis</ns21:Fractal>
    </ns21:FractalAlgebra>
</ns21:Dimension>
<ns21:RVDIid>urn:eu:convergence:server1\77e79d88-7087-4366-96a7-
9ec4a03a202d</ns21:RVDIid>
<ns21:Keyword>ORIGINAL</ns21:Keyword>
<ns21:FieldValue>
    <ns21:Field>TEAM</ns21:Field>
    <ns21:Value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns25="http://www.w3.org/2001/XMLSchema"
xsi:type="ns25:string">AEK</ns21:Value>
</ns21:FieldValue>
<ns21:StructuredData ref="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:videodescription="http://escom.msh-paris.fr/videodescription.owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
        <rdf:Description rdf:about="http://escom.msh-paris.fr/9257b284-7bf1-4272-
9542-732e566b00b9">
            <rdfs:label xml:lang="en">quechua translation</rdfs:label>
            <rdf:type rdf:resource="http://escom.msh-
paris.fr/videodescription.owl#RichDescription"/>
  </rdf:Description>
        <rdf:Description rdf:about="http://escom.msh-paris.fr/92bec06d-f033-4ea6-
836b-bd3c218826d4_Individual">
            <rdfs:label xml:lang="en">English literature is the literature written
in the English language, including literature composed in English by writers not
necessarily from England; for example, Robert Burns was Scottish, James Joyce was
Irish, Joseph Conrad was Polish, Dylan Thomas was Welsh, Edgar Allan Poe was Amer-
ican, J. R. R. Tolkien was born in the Orange Free State, V.S. Naipaul was born in
Trinidad, and Vladimir Nabokov was Russian, but all are considered important writ-
ers in the history of English literature. In other words, English literature is as
diverse as the varieties and dialects of English spoken around the world. In aca-
demia, the term often labels departments and programmes practising English studies
in secondary and tertiary educational systems. Despite the variety of authors of
English literature, the works of William Shakespeare remain paramount throughout
the English-speaking world.</rdfs:label>
```

```
  </rdf:Description>
        <rdf:Description rdf:about="http://escom.msh-paris.fr/9c33aaa5-f16c-4f27-
be6f-c7c60bfe157e_DescriptionPattern">
            <videodescription:Sign rdf:resource="http://escom.msh-
paris.fr/5c694344-c2e1-4efc-ba95-3746908229f1"/>
            <videodescription:Sequence rdf:resource="http://escom.msh-
paris.fr/8890602d-f1bb-402a-ae26-e63389524651"/>
            <videodescription:Scheme rdf:resource="http://escom.msh-
paris.fr/8dbcf418-07f4-474b-8c2a-071c302a93a8"/>
            <videodescription:Pattern rdf:resource="http://escom.msh-
paris.fr/3f5f0686-b013-47c7-ac28-c713a51f34f3"/>
            <videodescription:Object rdf:resource="http://escom.msh-
paris.fr/5d2276d5-f625-4a7d-9703-d7e67b7bcb31"/>
            <videodescription:DataType rdf:resource="http://escom.msh-
paris.fr/0665bf9a-cca1-44cb-8b6f-45af1a01703a"/>
            <rdf:type rdf:resource="http://escom.msh-
paris.fr/videodescription.owl#DescriptionPattern"/>
  </rdf:Description>
      </rdf:RDF>
</ns21:StructuredData>
                            </ns21:PublicationMetadata>
                        </ns21:ConvergenceMetadata>
```

7.2 DigitalItem TE

7.2.1 DI creation

DI file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mpegm-didl:DIDL xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS" xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:rel-
m1x="urn:mpeg:mpeg21:2005:01-REL-M1X-NS" xmlns:rel-sx="urn:mpeg:mpeg21:2003:01-
REL-SX-NS" xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:rel-
m2x="urn:mpeg:mpeg21:2006:01-REL-M2X-NS" xmlns:rel-mx="urn:mpeg:mpeg21:2003:01-
REL-MX-NS" xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS"
xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:fru="urn:mpeg:mpegB:schema:FragmentRequestUnits:2007"
xmlns:ns22="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ns23="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:mpegmipmpinfo="urn:mpeg:mpegM:schema:07-IPMPINFO-NS:2012">
    <mpegm-didl:DIDLInfo>
        <mpegm-didl:DISignature/>
    </mpegm-didl:DIDLInfo>
    <mpegm-didl:Item id="ANGELO">
        <mpegm-didl:Descriptor>
```

```
            <mpegm-didl:Statement>
                <dii:Identifier>temp:id:123456:ANGELO</dii:Identifier>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <mpegm-didl:ApplicationSpecificData/>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <mpegm-didl:ContentElementSignature/>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <er:ERR>
                    <er:ERSpecification>
                        <er:ERDeliverySpecification>
                            <er:DITransportService>
<rel-r:serviceReference>
    <rel-m1x:serviceLocation>
        <rel-m1x:url>localhost://</rel-m1x:url>
    </rel-m1x:serviceLocation>
</rel-r:serviceReference>
                            </er:DITransportService>
                        </er:ERDeliverySpecification>
                        <er:EmbeddedERR>
                            <er:ERRReference>ANGE</er:ERRReference>
                        </er:EmbeddedERR>
                    </er:ERSpecification>
                    <er:EventConditionDescriptor>
                        <er:DIOperationCondition>
                            <er:DIOperationEvent>
<er:Operation>match</er:Operation>
                            </er:DIOperationEvent>
                        </er:DIOperationCondition>
                    </er:EventConditionDescriptor>
                </er:ERR>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <dii:RelatedIdentifier relationshipType="relation">value
relat</dii:RelatedIdentifier>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Component>
            <mpegm-didl:Resource mimeType="mime--resource" ref="href--resource"
encoding="encoding--resource" contentEncoding="content--encoding--resource"/>
        </mpegm-didl:Component>
        <mpegm-didl:Item id="B0">
            <mpegm-didl:Descriptor>
                <mpegm-didl:Statement>
                    <dii:Identifier>urn:angelo</dii:Identifier>
                </mpegm-didl:Statement>
            </mpegm-didl:Descriptor>
```

```
            <mpegm-didl:Descriptor>
                <mpegm-didl:Statement>
                    <ipmpinfo:IPMPGeneralInfoDescriptor>
                        <ipmpinfo:LicenseCollection>
                            <ipmpinfo:RightsDescriptor>
<ipmpinfo:License>
    <rel-r:License licenseId="1234"/>
</ipmpinfo:License>
                            </ipmpinfo:RightsDescriptor>
                        </ipmpinfo:LicenseCollection>
                    </ipmpinfo:IPMPGeneralInfoDescriptor>
                </mpegm-didl:Statement>
            </mpegm-didl:Descriptor>
            <mpegm-didl:Component>
                <mpegm-didl:Resource mimeType="mime--resource" ref="href--
resource" encoding="encoding--resource" contentEncoding="content--encoding--
resource"/>
            </mpegm-didl:Component>
        </mpegm-didl:Item>
        <mpegm-didl:Item id="B1">
            <mpegm-didl:Descriptor>
                <mpegm-didl:Statement>
                    <dii:Identifier>unr:babbo</dii:Identifier>
                </mpegm-didl:Statement>
            </mpegm-didl:Descriptor>
            <mpegm-didl:Descriptor>
                <mpegm-didl:Statement>
                    <didl_msx:Metadata>
                        <didl_msx:StructuredData ref="urn:mpeg:mpeg7:schema:2004">
                            <mpeg7_v3:Mpeg7 timeUnit="323232"/>
                        </didl_msx:StructuredData>
                    </didl_msx:Metadata>
                </mpegm-didl:Statement>
            </mpegm-didl:Descriptor>
        </mpegm-didl:Item>
    </mpegm-didl:Item>
</mpegm-didl:DIDL>
```

7.3 EventReport TE

7.3.1 Event report creation

ER file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<er:ER                          xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012"
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"                    xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"   xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS"           xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS"           xmlns:rel-
```

```
m1x="urn:mpeg:mpeg21:2005:01-REL-M1X-NS"     xmlns:rel-sx="urn:mpeg:mpeg21:2003:01-
REL-SX-NS"                                xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"                              xmlns:rel-
m2x="urn:mpeg:mpeg21:2006:01-REL-M2X-NS"     xmlns:rel-mx="urn:mpeg:mpeg21:2003:01-
REL-MX-NS"                        xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS"
xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:fru="urn:mpeg:mpegB:schema:FragmentRequestUnits:2007"
xmlns:ns22="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ns23="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:mpegmipmpinfo="urn:mpeg:mpegM:schema:07-IPMPINFO-NS:2012">
    <er:ERDescriptor>
        <er:Description>urn:er_identifier:temp:db02bed6-f09a-48f5-86e8-
4d0b1da96c64</er:Description>
        <er:ERSource>
            <er:ERRReference>errId</er:ERRReference>
        </er:ERSource>
    </er:ERDescriptor>
    <er:ERData>
        <er:DII>DIDLId</er:DII>
    </er:ERData>
</er:ER>
```

7.3.2 Event report request creation

ERR file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<er:ERR                        xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012"       xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"   xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS"            xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS"            xmlns:rel-
sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"  xmlns:rel-m1x="urn:mpeg:mpeg21:2005:01-REL-
M1X-NS"           xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004"           xmlns:rel-
m2x="urn:mpeg:mpeg21:2006:01-REL-M2X-NS"     xmlns:rel-mx="urn:mpeg:mpeg21:2003:01-
REL-MX-NS"                        xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:fru="urn:mpeg:mpegB:schema:FragmentRequestUnits:2007"
```

```
xmlns:ns22="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ns23="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:mpegmipmpinfo="urn:mpeg:mpegM:schema:07-IPMPINFO-NS:2012">
    <er:ERSpecification>
        <er:ERDeliverySpecification>
            <er:DITransportService>
                <rel-r:serviceReference>
                    <rel-m1x:serviceLocation>
                        <rel-m1x:url>localhost://</rel-m1x:url>
                    </rel-m1x:serviceLocation>
                </rel-r:serviceReference>
            </er:DITransportService>
        </er:ERDeliverySpecification>
        <er:EmbeddedERR>
            <er:ERRReference>ANGE</er:ERRReference>
        </er:EmbeddedERR>
    </er:ERSpecification>
    <er:EventConditionDescriptor>
        <er:DIOperationCondition>
            <er:DIOperationEvent>
                <er:Operation>match</er:Operation>
            </er:DIOperationEvent>
        </er:DIOperationCondition>
    </er:EventConditionDescriptor>
</er:ERR>
```

7.4 IPMP TE

7.4.1 IPMP descriptor creation

IPMP file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ipmpinfo:IPMPGeneralInfoDescriptor
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS" xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:rel-
m1x="urn:mpeg:mpeg21:2005:01-REL-M1X-NS" xmlns:rel-sx="urn:mpeg:mpeg21:2003:01-
REL-SX-NS" xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004" xmlns:rel-
mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:rel-
```

```
m2x="urn:mpeg:mpeg21:2006:01-REL-M2X-NS" xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-
REL-M3X-NS" xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:fru="urn:mpeg:mpegB:schema:FragmentRequestUnits:2007"
xmlns:ns22="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ns23="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:mpegmipmpinfo="urn:mpeg:mpegM:schema:07-IPMPINFO-NS:2012">
    <ipmpinfo:LicenseCollection>
        <ipmpinfo:RightsDescriptor>
            <ipmpinfo:License>
                <rel-r:License licenseId="1234"/>
            </ipmpinfo:License>
        </ipmpinfo:RightsDescriptor>
    </ipmpinfo:LicenseCollection>
</ipmpinfo:IPMPGeneralInfoDescriptor>
```

7.5 Metadata TE

7.5.1 MPEG-7 metadata creation and parsing

MPEG-7 file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mpeg7_v3:Mpeg7 xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS" xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:rel-
m1x="urn:mpeg:mpeg21:2005:01-REL-M1X-NS" xmlns:rel-sx="urn:mpeg:mpeg21:2003:01-
REL-SX-NS" xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:rel-
m2x="urn:mpeg:mpeg21:2006:01-REL-M2X-NS" xmlns:rel-mx="urn:mpeg:mpeg21:2003:01-
REL-MX-NS" xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS"
xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:fru="urn:mpeg:mpegB:schema:FragmentRequestUnits:2007"
xmlns:ns22="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ns23="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:mpegmipmpinfo="urn:mpeg:mpegM:schema:07-IPMPINFO-NS:2012">
    <mpeg7_v3:Description xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="mpeg7_v3:CreationDescriptionType">
        <mpeg7_v3:CreationInformation>
            <mpeg7_v3:Creation>
                <mpeg7_v3:Title>The BIG Lebowsky</mpeg7_v3:Title>
                <mpeg7_v3:Abstract>
                    <mpeg7_v3:FreeTextAnnotation>desciption of
work</mpeg7_v3:FreeTextAnnotation>
                </mpeg7_v3:Abstract>
            </mpeg7_v3:Creation>
```

```
        </mpeg7_v3:CreationInformation>
    </mpeg7_v3:Description>
</mpeg7_v3:Mpeg7>
```

7.6 MPQF TE

7.6.1 Mixed MPQF query creating and parsing

MPQF query file

```
<ns21:Query>
    <ns22:QueryCondition>
        <ns22:Condition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ns22:OR">
            <ns22:Condition xsi:type="ns22:Equal">
                <ns22:StringField>KEYWORD</ns22:StringField>
                <ns22:StringValue>MY_KEYWORD</ns22:StringValue>
            </ns22:Condition>
            <ns22:Condition xsi:type="ns22:QueryBySPARQL">
                <ns22:SPARQL>SELECT  ?x
WHERE
  { ?x  &lt;http://escom.msh-paris.fr/videodescription.owl#hasReferences&gt;
&lt;http://escom.msh-paris.fr/531c1582-5950-4a2a-aab2-f428c2172212&gt; . }
</ns22:SPARQL>
            </ns22:Condition>
        </ns22:Condition>
    </ns22:QueryCondition>
</ns21:Query>
```

7.7 REL TE

7.7.1 REL license creation

REL license file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rel-r:license xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS" xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:rel-
m1x="urn:mpeg:mpeg21:2005:01-REL-M1X-NS" xmlns:rel-sx="urn:mpeg:mpeg21:2003:01-
REL-SX-NS" xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:rel-
m2x="urn:mpeg:mpeg21:2006:01-REL-M2X-NS" xmlns:rel-mx="urn:mpeg:mpeg21:2003:01-
REL-MX-NS" xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS"
xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:fru="urn:mpeg:mpegB:schema:FragmentRequestUnits:2007"
```

```
xmlns:ns22="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ns23="urn:oasis:names:tc:SAML:2.0:protocol"
xmlns:mpegmipmpinfo="urn:mpeg:mpegM:schema:07-IPMPINFO-NS:2012">
    <rel-r:inventory>
        <rel-r:keyHolder licensePartId="endUserJohn">
            <rel-r:info>
                <dsig:KeyValue>
                    <dsig:RSAKeyValue>

<dsig:Modulus>YTFoa2QwVmFOMDV1Y1RaNVlqTXlkRkZsUWtsUVIyZHVlV1lyVW1KR1ZXUnVkSEJWWXps
UWNqbFNVMGx4U0d3MlRXOW1iR1V5YlRobWQwd3dWa3BhZDBGeVRVdDRURWxWV1hNNWN3b3pNV1pFYkhOU2
JraDNQVDA9</dsig:Modulus>
                        <dsig:Exponent>UVZGQlFnPT0=</dsig:Exponent>
                    </dsig:RSAKeyValue>
                </dsig:KeyValue>
            </rel-r:info>
        </rel-r:keyHolder>
        <rel-r:keyHolder licensePartId="endUserJack">
            <rel-r:info>
                <dsig:KeyValue>
                    <dsig:RSAKeyValue>

<dsig:Modulus>YTFoa2QwVmFOMDV1Y1RaNVlqTXlkRkZsUWtsUVIyZHVlV1lyVW1KR1ZXUnVkSEJWWXps
UWNqbFNVMGx4U0d3MlRXOW1iR1V5YlRobWQwd3dWa3BhZDBGeVRVdDRURWxWV1hNNWN3b3pNV1pFYkhOU2
JraDNQVDA9</dsig:Modulus>
                        <dsig:Exponent>UVZGQlFnPT0=</dsig:Exponent>
                    </dsig:RSAKeyValue>
                </dsig:KeyValue>
            </rel-r:info>
        </rel-r:keyHolder>
        <rel-mx:diReference licensePartId="diReference">
            <rel-mx:identifier>http://www.onlinemusic.com/mySong.mp3</rel-
mx:identifier>
        </rel-mx:diReference>
        <rel-r:validityInterval>
            <rel-r:notBefore>2006-01-01T00:00:00</rel-r:notBefore>
            <rel-r:notAfter>2006-12-31T23:59:59</rel-r:notAfter>
        </rel-r:validityInterval>
    </rel-r:inventory>
    <rel-r:grant>
        <rel-r:forAll varName="anyEndUser"/>
        <rel-r:forAll varName="anyProducer"/>
        <rel-r:forAll varName="anyAggregator"/>
        <rel-r:principal varRef="anyProducer"/>
        <rel-r:obtain/>
        <rel-r:validityInterval licensePartIdRef="oneYear">
            <rel-r:notBefore>2006-01-01T00:00:00</rel-r:notBefore>
            <rel-r:notAfter>2006-12-31T23:59:59</rel-r:notAfter>
        </rel-r:validityInterval>
    </rel-r:grant>
    <rel-r:grantGroup>
        <rel-r:grant>
            <rel-r:principal varRef="anyProducer"/>
            <rel-mx:play/>
            <rel-mx:diReference licensePartIdRef="diReference"/>
            <rel-r:validityInterval licensePartIdRef="oneYear">
                <rel-r:notBefore>2006-01-01T00:00:00</rel-r:notBefore>
```

```
                <rel-r:notAfter>2006-12-31T23:59:59</rel-r:notAfter>
            </rel-r:validityInterval>
        </rel-r:grant>
        <rel-r:grant>
            <rel-r:principal varRef="anyProducer"/>
            <rel-m1x:governedCopy/>
            <rel-mx:diReference licensePartIdRef="diReference"/>
            <rel-r:validityInterval licensePartIdRef="oneYear">
                <rel-r:notBefore>2006-01-01T00:00:00</rel-r:notBefore>
                <rel-r:notAfter>2006-12-31T23:59:59</rel-r:notAfter>
            </rel-r:validityInterval>
        </rel-r:grant>
        <rel-r:grant>
            <rel-r:principal varRef="anyProducer"/>
            <rel-m3x:governedAggregate/>
            <rel-mx:diReference licensePartIdRef="diReference"/>
            <rel-r:validityInterval licensePartIdRef="oneYear">
                <rel-r:notBefore>2006-01-01T00:00:00</rel-r:notBefore>
                <rel-r:notAfter>2006-12-31T23:59:59</rel-r:notAfter>
            </rel-r:validityInterval>
        </rel-r:grant>
        <rel-r:grant>
            <rel-r:principal varRef="anyEndUser"/>
            <rel-mx:play/>
            <rel-mx:diReference licensePartIdRef="diReference"/>
        </rel-r:grant>
    </rel-r:grantGroup>
</rel-r:license>
```

7.8 VDI TE

7.8.1 VDI creation

VDI file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mpegm-didl:DIDL xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS" xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:rel-
m1x="urn:mpeg:mpeg21:2005:01-REL-M1X-NS" xmlns:rel-sx="urn:mpeg:mpeg21:2003:01-
REL-SX-NS" xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:rel-
mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:rel-m2x="urn:mpeg:mpeg21:2006:01-REL-
M2X-NS" xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004"
xmlns:ns21="urn:conv:metadata:schema:2011" xmlns:ns22="urn:mpeg:mpqf:schema:2008"
xmlns:ns23="urn:conv:relation:schema:2011">
    <mpegm-didl:Item id="ANGELO">
```

```
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <mpegm-didl:ApplicationSpecificData>
                    <ns23:relationship>
                        <ns23:OntologyURI>ontologyURI</ns23:OntologyURI>
                        <ns23:SubjectVDI>subjectID</ns23:SubjectVDI>
                        <ns23:Predicate>RelationshipType</ns23:Predicate>
                        <ns23:ObjectVDI>objectID</ns23:ObjectVDI>
                    </ns23:relationship>
                </mpegm-didl:ApplicationSpecificData>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
    </mpegm-didl:Item>
</mpegm-didl:DIDL>
```

7.9 Middleware demo

REL license file

```
<rel-r:license xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#" xmlns:er="urn:mpeg:mpeg21:2005:01-
ERL-NS" xmlns:fru="urn:mpeg:mpegB:schema:FragmentRequestUnits:2007"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004" xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012"
xmlns:mpegmipmpinfo="urn:mpeg:mpegM:schema:07-IPMPINFO-NS:2012"
xmlns:ns22="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:ns23="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:rel-
m1x="urn:mpeg:mpeg21:2005:01-REL-M1X-NS" xmlns:rel-m2x="urn:mpeg:mpeg21:2006:01-
REL-M2X-NS" xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS" xmlns:rel-
mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-
NS" xmlns:rel-sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"><rel-r:grant><rel-
m1x:identityHolder><rel-m1x:idValue>AZ</rel-m1x:idValue></rel-
m1x:identityHolder><rel-mx:play/><rel-r:digitalResource><rel-r:nonSecureIndirect
URI="http://digitalResource"/></rel-r:digitalResource><rel-
r:validityInterval><rel-r:notBefore>2012-10-01T12:45:04.287+03:00</rel-
r:notBefore></rel-r:validityInterval></rel-r:grant><rel-r:issuer><rel-
m1x:identityHolder><rel-m1x:idValue>AZ</rel-m1x:idValue></rel-
m1x:identityHolder></rel-r:issuer><rel-r:issuer/></rel-r:license>
```

R-VDI file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mpegm-didl:DIDL xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012" xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS" xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:rel-
sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS" xmlns:rel-m1x="urn:mpeg:mpeg21:2005:01-REL-
M1X-NS" xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:rel-
mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:rel-m2x="urn:mpeg:mpeg21:2006:01-REL-
M2X-NS" xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004"
xmlns:ns21="urn:conv:metadata:schema:2011" xmlns:ns22="urn:mpeg:mpqf:schema:2008"
xmlns:ns23="urn:conv:relation:schema:2011">
    <mpegm-didl:Item id="A0">
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <dii:Identifier>urn:eu:convergence:server1\243a6d19-3209-4ec0-
9cef-2b428a61432c</dii:Identifier>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <didl_msx:Metadata>
                    <didl_msx:StructuredData ref="urn:conv:metadata:schema:2011">
                        <ns21:ConvergenceMetadata>

<ns21:SequenceIdentifier>urn:sequence_identifier:temp:504ac550-059f-4b4d-9424-
e52edaad0a70</ns21:SequenceIdentifier>
                            <ns21:VDIkind>RVDI</ns21:VDIkind>
                            <ns21:StartDate>2012-10-
01T12:36:41.924+03:00</ns21:StartDate>
                            <ns21:ResourceMetadata>
<ns21:Keyword>keyword</ns21:Keyword>
<ns21:Tag>tag</ns21:Tag>
<ns21:FieldValue>
    <ns21:Field>field</ns21:Field>
    <ns21:Value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns25="http://www.w3.org/2001/XMLSchema"
xsi:type="ns25:string">value</ns21:Value>
</ns21:FieldValue>
                            </ns21:ResourceMetadata>
                        </ns21:ConvergenceMetadata>
                    </didl_msx:StructuredData>
                </didl_msx:Metadata>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Item id="B0">
            <mpegm-didl:Descriptor>
                <mpegm-didl:Statement>
                    <dii:Identifier>identifier:contentelement:temp:733d0348-82d9-
```

```
4a96-baaa-cd2f3df48215</dii:Identifier>
                </mpegm-didl:Statement>
            </mpegm-didl:Descriptor>
            <mpegm-didl:Descriptor>
                <mpegm-didl:Statement>
                    <ipmpinfo:IPMPGeneralInfoDescriptor>
                        <ipmpinfo:LicenseCollection>
                            <ipmpinfo:RightsDescriptor>
<ipmpinfo:License>
    <rel-r:License>
        <rel-r:grant>
            <rel-m1x:identityHolder>
                <rel-m1x:idValue>AZ</rel-m1x:idValue>
            </rel-m1x:identityHolder>
            <rel-mx:play/>
            <rel-r:digitalResource>
                <rel-r:nonSecureIndirect URI="http://digitalResource"/>
            </rel-r:digitalResource>
            <rel-r:validityInterval>
                <rel-r:notBefore>2012-10-01T12:36:02.380+03:00</rel-r:notBefore>
            </rel-r:validityInterval>
        </rel-r:grant>
        <rel-r:issuer>
            <rel-m1x:identityHolder>
                <rel-m1x:idValue>AZ</rel-m1x:idValue>
            </rel-m1x:identityHolder>
        </rel-r:issuer>
        <rel-r:issuer/>
    </rel-r:License>
</ipmpinfo:License>
                            </ipmpinfo:RightsDescriptor>
                        </ipmpinfo:LicenseCollection>
                    </ipmpinfo:IPMPGeneralInfoDescriptor>
                </mpegm-didl:Statement>
            </mpegm-didl:Descriptor>
            <mpegm-didl:Component>
                <mpegm-didl:Resource mimeType="MIME/TYPE" ref="REF">This is a com-
ment</mpegm-didl:Resource>
            </mpegm-didl:Component>
        </mpegm-didl:Item>
    </mpegm-didl:Item>
</mpegm-didl:DIDL>
```

P-VDI file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mpegm-didl:DIDL xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012" xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS" xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:rel-
sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS" xmlns:rel-m1x="urn:mpeg:mpeg21:2005:01-REL-
M1X-NS" xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
```

```
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:rel-
mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:rel-m2x="urn:mpeg:mpeg21:2006:01-REL-
M2X-NS" xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004"
xmlns:ns21="urn:conv:metadata:schema:2011" xmlns:ns22="urn:mpeg:mpqf:schema:2008"
xmlns:ns23="urn:conv:relation:schema:2011">
    <mpegm-didl:Item id="A0">
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <dii:Identifier>urn:eu:convergence:server1\fa0cbe4e-0e1a-47dc-
ba47-4f57acd9f5d0</dii:Identifier>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <didl_msx:Metadata>
                    <didl_msx:StructuredData ref="urn:conv:metadata:schema:2011">
                        <ns21:ConvergenceMetadata>

<ns21:SequenceIdentifier>urn:sequence_identifier:temp:b110acd4-cc93-47d7-8cfc-
4db83c3339bf</ns21:SequenceIdentifier>
                            <ns21:VDIkind>PVDI</ns21:VDIkind>
                            <ns21:StartDate>2012-10-
01T12:37:04.480+03:00</ns21:StartDate>
                            <ns21:ExpiryDate>2012-11-
01T12:37:04.480+02:00</ns21:ExpiryDate>
                            <ns21:PublicationMetadata>
<ns21:Dimension>
    <ns21:FractalAlgebra>
        <ns21:Fractal>aek</ns21:Fractal>
    </ns21:FractalAlgebra>
</ns21:Dimension>
<ns21:RVDIid>urn:eu:convergence:server1\243a6d19-3209-4ec0-9cef-
2b428a61432c</ns21:RVDIid>
<ns21:Keyword>keyword</ns21:Keyword>
<ns21:Tag>tag</ns21:Tag>
<ns21:FieldValue>
    <ns21:Field>field</ns21:Field>
    <ns21:Value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns25="http://www.w3.org/2001/XMLSchema"
xsi:type="ns25:string">value</ns21:Value>
</ns21:FieldValue>
                            </ns21:PublicationMetadata>
                        </ns21:ConvergenceMetadata>
                    </didl_msx:StructuredData>
                </didl_msx:Metadata>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
    </mpegm-didl:Item>
</mpegm-didl:DIDL>
```

S-VDI file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mpegm-didl:DIDL xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
```

```
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012" xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS" xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:rel-
sx="urn:mpeg:mpeg21:2003:01-REL-SX-NS" xmlns:rel-m1x="urn:mpeg:mpeg21:2005:01-REL-
M1X-NS" xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:rel-
mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:rel-m2x="urn:mpeg:mpeg21:2006:01-REL-
M2X-NS" xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004"
xmlns:ns21="urn:conv:metadata:schema:2011" xmlns:ns22="urn:mpeg:mpqf:schema:2008"
xmlns:ns23="urn:conv:relation:schema:2011">
    <mpegm-didl:Item id="A0">
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <dii:Identifier>urn:eu:convergence:server1\e223bf1f-4c4d-4e1e-
ac7b-2e1c3598c95a</dii:Identifier>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <didl_msx:Metadata>
                    <didl_msx:StructuredData ref="urn:conv:metadata:schema:2011">
                        <ns21:ConvergenceMetadata>

<ns21:SequenceIdentifier>urn:sequence_identifier:temp:a8a87bd2-9d16-492f-9047-
084476aac47d</ns21:SequenceIdentifier>
                            <ns21:VDIkind>SVDI</ns21:VDIkind>
                            <ns21:StartDate>2012-10-
01T12:37:12.746+03:00</ns21:StartDate>
                            <ns21:ExpiryDate>2012-11-
01T12:37:12.746+02:00</ns21:ExpiryDate>
                            <ns21:SubscriptionMetadata>
<ns21:Dimension>
    <ns21:FractalAlgebra>
        <ns21:Fractal>aek</ns21:Fractal>
    </ns21:FractalAlgebra>
</ns21:Dimension>
<ns21:Query>
    <ns22:QueryCondition>
        <ns22:Condition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ns22:AND">
            <ns22:Condition xsi:type="ns22:Equal">
                <ns22:StringField>field</ns22:StringField>
                <ns22:StringValue>value</ns22:StringValue>
            </ns22:Condition>
            <ns22:Condition xsi:type="ns22:Equal">
                <ns22:StringField>KEYWORD</ns22:StringField>
                <ns22:StringValue>keyword</ns22:StringValue>
            </ns22:Condition>
            <ns22:Condition xsi:type="ns22:Equal">
```

```
                <ns22:StringField>TAG</ns22:StringField>
                <ns22:StringValue>tag</ns22:StringValue>
            </ns22:Condition>
        </ns22:Condition>
    </ns22:QueryCondition>
</ns21:Query>
                            </ns21:SubscriptionMetadata>
                        </ns21:ConvergenceMetadata>
                    </didl_msx:StructuredData>
                </didl_msx:Metadata>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Item id="B0">
            <mpegm-didl:Descriptor>
                <mpegm-didl:Statement>
                    <dii:Identifier>identifier:contentelement:temp:32f9f006-2e89-
47f9-830a-aeea756def1d</dii:Identifier>
                </mpegm-didl:Statement>
            </mpegm-didl:Descriptor>
            <mpegm-didl:Descriptor>
                <mpegm-didl:Statement>
                    <er:ERR>
                        <er:ERSpecification>
                            <er:ERDeliverySpecification>
<er:DITransportService>
    <rel-r:serviceReference>
        <rel-m1x:serviceLocation>
            <rel-m1x:url>peer6.peers.ict-convergence.eu/er</rel-m1x:url>
        </rel-m1x:serviceLocation>
    </rel-r:serviceReference>
</er:DITransportService>
                            </er:ERDeliverySpecification>
                            <er:EmbeddedERR>
<er:ERRReference>urn:err_identifier:temp:af419f7a-a8ef-485d-8530-
669bfe267bd2</er:ERRReference>
                            </er:EmbeddedERR>
                        </er:ERSpecification>
                        <er:EventConditionDescriptor>
                            <er:DIOperationCondition>
<er:DIOperationEvent>
    <er:Operation>match</er:Operation>
</er:DIOperationEvent>
                            </er:DIOperationCondition>
                        </er:EventConditionDescriptor>
                    </er:ERR>
                </mpegm-didl:Statement>
            </mpegm-didl:Descriptor>
        </mpegm-didl:Item>
    </mpegm-didl:Item>
</mpegm-didl:DIDL>
```

7.10 RDF and SPARQL app

R-VDI file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mpegm-didl:DIDL xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
```

```
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012" xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS" xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:rel-
m1x="urn:mpeg:mpeg21:2005:01-REL-M1X-NS" xmlns:rel-sx="urn:mpeg:mpeg21:2003:01-
REL-SX-NS" xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:rel-
mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:rel-m2x="urn:mpeg:mpeg21:2006:01-REL-
M2X-NS" xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004"
xmlns:ns21="urn:conv:metadata:schema:2011" xmlns:ns22="urn:mpeg:mpqf:schema:2008"
xmlns:ns23="urn:conv:relation:schema:2011">
    <mpegm-didl:Item id="A0">
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <dii:Identifier>urn:eu:convergence:server1\ad9d8160-4a6d-4b67-
850f-d4f03869ea28</dii:Identifier>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <didl_msx:Metadata>
                    <didl_msx:StructuredData ref="urn:conv:metadata:schema:2011">
                        <ns21:ConvergenceMetadata>

<ns21:SequenceIdentifier>urn:sequence_identifier:temp:85bd850d-60af-4b33-b296-
9eace8ab1b0a</ns21:SequenceIdentifier>
                            <ns21:VDIkind>RVDI</ns21:VDIkind>
                            <ns21:ExpiryDate>2012-10-
01T10:43:19.104+03:00</ns21:ExpiryDate>
                            <ns21:ResourceMetadata>
<ns21:Keyword>ORIGINAL</ns21:Keyword>
<ns21:FieldValue>
    <ns21:Field>TEAM</ns21:Field>
    <ns21:Value xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns25="http://www.w3.org/2001/XMLSchema"
xsi:type="ns25:string">AEK</ns21:Value>
</ns21:FieldValue>
<ns21:StructuredData ref="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:videodescription="http://escom.msh-paris.fr/videodescription.owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
  <rdf:Description rdf:about="http://escom.msh-paris.fr/9257b284-7bf1-4272-9542-
732e566b00b9">
    <rdfs:label xml:lang="en">quechua translation</rdfs:label>
    <rdf:type rdf:resource="http://escom.msh-
paris.fr/videodescription.owl#RichDescription"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://escom.msh-paris.fr/92bec06d-f033-4ea6-836b-
```

```
bd3c218826d4_Individual">
    <rdfs:label xml:lang="en">English literature is the literature written in the
English language, including literature composed in English by writers not neces-
sarily from England; for example, Robert Burns was Scottish, James Joyce was
Irish, Joseph Conrad was Polish, Dylan Thomas was Welsh, Edgar Allan Poe was Amer-
ican, J. R. R. Tolkien was born in the Orange Free State, V.S. Naipaul was born in
Trinidad, and Vladimir Nabokov was Russian, but all are considered important writ-
ers in the history of English literature. In other words, English literature is as
diverse as the varieties and dialects of English spoken around the world. In aca-
demia, the term often labels departments and programmes practising English studies
in secondary and tertiary educational systems. Despite the variety of authors of
English literature, the works of William Shakespeare remain paramount throughout
the English-speaking world.</rdfs:label>
  </rdf:Description>
  <rdf:Description rdf:about="http://escom.msh-paris.fr/df76b56b-77ca-4053-b0dd-
1878f8f01dd1_DescriptionPattern">
    <videodescription:Sign rdf:resource="http://escom.msh-paris.fr/d3651e01-5076-
4969-bbb4-948d8dac8cb4"/>
    <videodescription:Sequence rdf:resource="http://escom.msh-paris.fr/8890602d-
f1bb-402a-ae26-e63389524651"/>
    <videodescription:Scheme rdf:resource="http://escom.msh-paris.fr/57327ae3-
a1e8-461d-a5f1-d465d9b19b6a"/>
    <videodescription:Pattern rdf:resource="http://escom.msh-paris.fr/bb974842-
adf6-41f6-a44c-c30740b36653"/>
    <videodescription:Object rdf:resource="http://escom.msh-paris.fr/f9803178-
d235-4325-988c-0351ea2bdde4"/>
    <videodescription:DataType rdf:resource="http://escom.msh-paris.fr/5b7f6e0c-
49ab-4dd0-b9b9-2d086f1ec2cd"/>
    <rdf:type rdf:resource="http://escom.msh-
paris.fr/videodescription.owl#DescriptionPattern"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://escom.msh-paris.fr/b9312a22-4c0f-4282-910d-
f18c00313e35">
    <videodescription:hasDescription rdf:resource="http://escom.msh-
paris.fr/fd7dcdea-75ae-4d79-a582-cd411dcc91ab"/>
    <videodescription:Kind rdf:resource="http://escom.msh-paris.fr/7217ab7b-5a2a-
4eb6-aa64-a77b38a37b3b"/>
    <rdf:type rdf:resource="http://escom.msh-
paris.fr/videodescription.owl#Rhetoric"/>
  </rdf:Description>
</rdf:RDF>
</ns21:StructuredData>
                            </ns21:ResourceMetadata>
                        </ns21:ConvergenceMetadata>
                    </didl_msx:StructuredData>
                </didl_msx:Metadata>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
    </mpegm-didl:Item>
</mpegm-didl:DIDL>
```

S-VDI-3 file

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<mpegm-didl:DIDL xmlns:didmodel="urn:mpeg:mpeg21:2002:02-DIDMODEL-NS"
xmlns:didl_mp21="urn:mpeg:mpeg21:2002:02-DIDL-NS"
xmlns:ipmpinfo="urn:mpeg:mpeg21:2004:01-IPMPINFO-NS"
```

```
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:mpegmipmpdidl="urn:mpeg:mpegM:schema:08-IPMPDIDL-NS:2012" xmlns:mpegm-
didl="urn:mpeg:mpegM:schema:06-didl-NS:2012"
xmlns:didl_msx="urn:mpeg:maf:schema:mediastreaming:DIDLextensions"
xmlns:dii="urn:mpeg:mpeg21:2002:01-DII-NS" xmlns:er="urn:mpeg:mpeg21:2005:01-ERL-
NS" xmlns:rel-r="urn:mpeg:mpeg21:2003:01-REL-R-NS" xmlns:rel-
m1x="urn:mpeg:mpeg21:2005:01-REL-M1X-NS" xmlns:rel-sx="urn:mpeg:mpeg21:2003:01-
REL-SX-NS" xmlns:ipmpmsg="urn:mpeg:mpegB:schema:IPMP-XML-MESSAGES:2007"
xmlns:xenc="http://www.w3.org/2001/04/xmlenc#" xmlns:rel-
mx="urn:mpeg:mpeg21:2003:01-REL-MX-NS" xmlns:rel-m2x="urn:mpeg:mpeg21:2006:01-REL-
M2X-NS" xmlns:rel-m3x="urn:mpeg:mpeg21:2006:01-REL-M3X-NS"
xmlns:ipmpinfo_msx="urn:mpeg:maf:Schema:mediastreaming:IPMPINFOextensions:2007"
xmlns:mpeg4ipmp="urn:mpeg:mpeg4:IPMPSchema:2002"
xmlns:mpeg7_v3="urn:mpeg:mpeg7:schema:2004"
xmlns:ns21="urn:conv:metadata:schema:2011" xmlns:ns22="urn:mpeg:mpqf:schema:2008"
xmlns:ns23="urn:conv:relation:schema:2011">
    <mpegm-didl:Item id="A0">
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <dii:Identifier>urn:eu:convergence:server1\53c42d17-3a5b-481b-
b71b-d3bb1608e954</dii:Identifier>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Descriptor>
            <mpegm-didl:Statement>
                <didl_msx:Metadata>
                    <didl_msx:StructuredData ref="urn:conv:metadata:schema:2011">
                        <ns21:ConvergenceMetadata>

<ns21:SequenceIdentifier>urn:sequence_identifier:temp:9f8910f5-50dd-44c8-afec-
09a53bef08e2</ns21:SequenceIdentifier>
                            <ns21:VDIkind>SVDI</ns21:VDIkind>
                            <ns21:ExpiryDate>2012-10-
01T10:47:56.965+03:00</ns21:ExpiryDate>
                            <ns21:SubscriptionMetadata>
<ns21:Dimension>
    <ns21:FractalAlgebra>
        <ns21:Fractal>analysis</ns21:Fractal>
    </ns21:FractalAlgebra>
</ns21:Dimension>
<ns21:Query>
    <ns22:QueryCondition>
        <ns22:Condition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="ns22:OR">
            <ns22:Condition xsi:type="ns22:Equal">
                <ns22:StringField>FIELD</ns22:StringField>
                <ns22:StringValue>VALUE</ns22:StringValue>
            </ns22:Condition>
            <ns22:Condition xsi:type="ns22:QueryBySPARQL">
                <ns22:SPARQL>SELECT  ?x
WHERE
  { ?x  &lt;http://escom.msh-paris.fr/videodescription.owl#hasReferences&gt;
&lt;http://escom.msh-paris.fr/531c1582-5950-4a2a-aab2-f428c2172212&gt; . }
</ns22:SPARQL>
            </ns22:Condition>
        </ns22:Condition>
    </ns22:QueryCondition>
```

```
</ns21:Query>
                            </ns21:SubscriptionMetadata>
                        </ns21:ConvergenceMetadata>
                    </didl_msx:StructuredData>
                </didl_msx:Metadata>
            </mpegm-didl:Statement>
        </mpegm-didl:Descriptor>
        <mpegm-didl:Item id="B0">
            <mpegm-didl:Descriptor>
                <mpegm-didl:Statement>
                    <dii:Identifier>identifier:contentelement:temp:79684509-9ada-
4b20-a013-d63532b33d40</dii:Identifier>
                </mpegm-didl:Statement>
            </mpegm-didl:Descriptor>
            <mpegm-didl:Descriptor>
                <mpegm-didl:Statement>
                    <er:ERR>
                        <er:ERSpecification>
                            <er:ERDeliverySpecification>
<er:DITransportService>
    <rel-r:serviceReference>
        <rel-m1x:serviceLocation>
            <rel-m1x:url>peer6.peers.ict-convergence.eu/er</rel-m1x:url>
        </rel-m1x:serviceLocation>
    </rel-r:serviceReference>
</er:DITransportService>
                            </er:ERDeliverySpecification>
                            <er:EmbeddedERR>
<er:ERRReference>urn:err_identifier:temp:03a18886-e39d-4bce-9eec-
c1d5b9391148</er:ERRReference>
                            </er:EmbeddedERR>
                        </er:ERSpecification>
                        <er:EventConditionDescriptor>
                            <er:DIOperationCondition>
<er:DIOperationEvent>
    <er:Operation>match</er:Operation>
</er:DIOperationEvent>
                            </er:DIOperationCondition>
                        </er:EventConditionDescriptor>
                    </er:ERR>
                </mpegm-didl:Statement>
            </mpegm-didl:Descriptor>
        </mpegm-didl:Item>
    </mpegm-didl:Item>
</mpegm-didl:DIDL>
```

8 ANNEX B – MAVEN build output

8.1 Build success

IPMP TE build success

```
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building mxm-ipmpTE 0.3.1
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- MAVEN-resources-plugin:2.4.3:resources (default-resources) @ mxm-ipmpTE
---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- MAVEN-compiler-plugin:2.0:compile (default-compile) @ mxm-ipmpTE ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- MAVEN-resources-plugin:2.4.3:testResources (default-testResources) @
mxm-ipmpTE ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 2 resources
[INFO]
[INFO] --- MAVEN-compiler-plugin:2.0:testCompile (default-testCompile) @ mxm-
ipmpTE ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- MAVEN-surefire-plugin:2.6:test (default-test) @ mxm-ipmpTE ---
[INFO] Surefire report directory: C:\CONVERGENCE\Work\RefactoringEW\mxm\mxm-
engines\mxm-ipmpTE\target\surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running org.iso.mpeg.ipmpTE.test.IPMPEngineTest
 INFO [main] (IPMPEngineTest.java:63)-

---------- testIPMPParsing ----------
 INFO [main] (MXM.java:158)-  ------- Multimedia Extensible Middleware -------
 INFO [main] (MXM.java:159)-  - Version 0.0.1
 INFO [main] (MXM.java:160)-  - MXM Configuration file:
C:\CONVERGENCE\Work\RefactoringEW\mxm\mxm-engines\mxm-ipmpTE\target\test-
classes\MXMConfiguration.xml
 INFO [main] (MXM.java:167)- Initializing parameters...
 INFO [main] (MXM.java:171)- Initializing engines...
 INFO [main] (MXMEngineLoader.java:88)- Loaded engine: RELTE
 INFO [main] (MXMEngineLoader.java:113)- Loaded engine: IPMPTE
 INFO [main] (MXM.java:175)- DONE Initializing...
 WARN [main] (MXM.java:264)- No MXM Parameters were set.
 INFO [main] (MXMAbstractEngine.java:59)-
-------------- MXM Engine --------------
 INFO [main] (MXMAbstractEngine.java:60)- -- Engine name: RELTE --
 INFO [main] (MXMAbstractEngine.java:61)- -- Engine ID: 0
```

```
 WARN [main] (MXMAbstractEngine.java:68)- No MXM Parameters were set.
 INFO [main] (MXMAbstractEngine.java:59)-
-------------- MXM Engine --------------
 INFO [main] (MXMAbstractEngine.java:60)- -- Engine name: IPMPTE --
 INFO [main] (MXMAbstractEngine.java:61)- -- Engine ID: 2
 WARN [main] (MXMAbstractEngine.java:68)- No MXM Parameters were set.
DEBUG [main] (IPMPEngineContext.java:41)- Generating Engine JAXB Context...
DEBUG [main] (IPMPEngineContext.java:43)- Done!
DEBUG [main] (RELEngineContext.java:111)- Generating RELEngine JAXB Context...
DEBUG [main] (RELEngineContext.java:113)- Done!
id of the embedded license is:1234
 INFO [main] (IPMPEngineTest.java:97)-

---------- testIPMPGeneration ----------
 INFO [main] (MXM.java:158)-  ------- Multimedia Extensible Middleware -------
 INFO [main] (MXM.java:159)-  - Version 0.0.1
 INFO [main] (MXM.java:160)-  - MXM Configuration file:
C:\CONVERGENCE\Work\RefactoringEW\mxm\mxm-engines\mxm-ipmpTE\target\test-
classes\MXMConfiguration.xml
 INFO [main] (MXM.java:167)- Initializing parameters...
 INFO [main] (MXM.java:171)- Initializing engines...
 INFO [main] (MXMEngineLoader.java:88)- Loaded engine: RELTE
 INFO [main] (MXMEngineLoader.java:113)- Loaded engine: IPMPTE
 INFO [main] (MXM.java:175)- DONE Initializing...
 WARN [main] (MXM.java:264)- No MXM Parameters were set.
 INFO [main] (MXMAbstractEngine.java:59)-
-------------- MXM Engine --------------
 INFO [main] (MXMAbstractEngine.java:60)- -- Engine name: RELTE --
 INFO [main] (MXMAbstractEngine.java:61)- -- Engine ID: 0
 WARN [main] (MXMAbstractEngine.java:68)- No MXM Parameters were set.
 INFO [main] (MXMAbstractEngine.java:59)-
-------------- MXM Engine --------------
 INFO [main] (MXMAbstractEngine.java:60)- -- Engine name: IPMPTE --
 INFO [main] (MXMAbstractEngine.java:61)- -- Engine ID: 2
 WARN [main] (MXMAbstractEngine.java:68)- No MXM Parameters were set.
 INFO [main] (IPMPEngineTest.java:139)- The IPMP file was successfully created in
file: C:\Users\AZHAMM~1\AppData\Local\Temp\\myIPMP.xml
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 8.084 sec

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- MAVEN-jar-plugin:2.3.1:jar (default-jar) @ mxm-ipmpTE ---
[INFO] Building jar: C:\CONVERGENCE\Work\RefactoringEW\mxm\mxm-engines\mxm-
ipmpTE\target\mxm-ipmpTE-0.3.1.jar
[INFO]
[INFO] --- MAVEN-install-plugin:2.3.1:install (default-install) @ mxm-ipmpTE ---
[INFO] Installing C:\CONVERGENCE\Work\RefactoringEW\mxm\mxm-engines\mxm-
ipmpTE\target\mxm-ipmpTE-0.3.1.jar to C:\Users\Az Ham-
mer\.m2\repository\org\iso\mpeg\engines\mxm-ipmpTE\0.3.1\mxm-ipmpTE-0.3.1.jar
[INFO] Installing C:\CONVERGENCE\Work\RefactoringEW\mxm\mxm-engines\mxm-
ipmpTE\pom.xml to C:\Users\Az Hammer\.m2\repository\org\iso\mpeg\engines\mxm-
ipmpTE\0.3.1\mxm-ipmpTE-0.3.1.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
```

```
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 55.486s
[INFO] Finished at: Tue Oct 02 18:14:12 EEST 2012
[INFO] Final Memory: 4M/7M
[INFO] ------------------------------------------------------------------------
```

8.2 Build failure

IPMP TE build failure

```
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building mxm-ipmpTE 0.3.1
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- MAVEN-resources-plugin:2.4.3:resources (default-resources) @ mxm-ipmpTE
---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- MAVEN-compiler-plugin:2.0:compile (default-compile) @ mxm-ipmpTE ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- MAVEN-resources-plugin:2.4.3:testResources (default-testResources) @
mxm-ipmpTE ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 2 resources
[INFO]
[INFO] --- MAVEN-compiler-plugin:2.0:testCompile (default-testCompile) @ mxm-
ipmpTE ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- MAVEN-surefire-plugin:2.6:test (default-test) @ mxm-ipmpTE ---
[INFO] Surefire report directory: C:\CONVERGENCE\Work\RefactoringEW\mxm\mxm-
engines\mxm-ipmpTE\target\surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running org.iso.mpeg.ipmpTE.test.IPMPEngineTest
 INFO [main] (IPMPEngineTest.java:63)-

---------- testIPMPParsing ----------
 INFO [main] (MXM.java:158)-  ------- Multimedia Extensible Middleware -------
 INFO [main] (MXM.java:159)-  - Version 0.0.1
 INFO [main] (MXM.java:160)-  - MXM Configuration file:
C:\CONVERGENCE\Work\RefactoringEW\mxm\mxm-engines\mxm-ipmpTE\target\test-
classes\MXMConfiguration.xml
 INFO [main] (MXM.java:167)- Initializing parameters...
 INFO [main] (MXM.java:171)- Initializing engines...
 INFO [main] (MXMEngineLoader.java:88)- Loaded engine: RELTE
 INFO [main] (MXMEngineLoader.java:113)- Loaded engine: IPMPTE
 INFO [main] (MXM.java:175)- DONE Initializing...
 WARN [main] (MXM.java:264)- No MXM Parameters were set.
```

```
 INFO [main] (MXMAbstractEngine.java:59)-
-------------- MXM Engine --------------
 INFO [main] (MXMAbstractEngine.java:60)- -- Engine name: RELTE --
 INFO [main] (MXMAbstractEngine.java:61)- -- Engine ID: 0
 WARN [main] (MXMAbstractEngine.java:68)- No MXM Parameters were set.
 INFO [main] (MXMAbstractEngine.java:59)-
-------------- MXM Engine --------------
 INFO [main] (MXMAbstractEngine.java:60)- -- Engine name: IPMPTE --
 INFO [main] (MXMAbstractEngine.java:61)- -- Engine ID: 2
 WARN [main] (MXMAbstractEngine.java:68)- No MXM Parameters were set.
DEBUG [main] (IPMPEngineContext.java:41)- Generating Engine JAXB Context...
DEBUG [main] (IPMPEngineContext.java:43)- Done!
[Fatal Error] testIPMP.xml:17:17: The element type "dsig:SigsdasdsadadasdnedInfo"
must be terminated by the matching end-tag "</dsig:SigsdasdsadadasdnedInfo>".
ERROR [main] (SuperAbstractLevel1.java:379)- A SAXException occurred. The detailed
exception is: The element type "dsig:SigsdasdsadadasdnedInfo" must be terminated
by the matching end-tag "</dsig:SigsdasdsadadasdnedInfo>".
org.xml.sax.SAXParseException: The element type "dsig:SigsdasdsadadasdnedInfo"
must be terminated by the matching end-tag "</dsig:SigsdasdsadadasdnedInfo>".
       at org.apache.xerces.parsers.DOMParser.parse(Unknown Source)
       at org.apache.xerces.jaxp.DocumentBuilderImpl.parse(Unknown Source)
       at javax.xml.parsers.DocumentBuilder.parse(DocumentBuilder.java:208)
       at
org.iso.mpeg.mxm.baseprotocol.schema.SuperAbstractLevel1.parse(SuperAbstractLevel1
.java:356)
       at
org.iso.mpeg.mxm.baseprotocol.schema.SuperAbstractLevel1.parse(SuperAbstractLevel1
.java:319)
       at
org.iso.mpeg.mxm.ipmpTE.schemahandler.IPMPGeneralInfoDescriptor.parse(IPMPGeneralI
nfoDescriptor.java:46)
       at
org.iso.mpeg.ipmpTE.test.IPMPEngineTest.testIPMPParsing(IPMPEngineTest.java:79)
       at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
       at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
       at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
25)
       at java.lang.reflect.Method.invoke(Method.java:597)
       at junit.framework.TestCase.runTest(TestCase.java:168)
       at junit.framework.TestCase.runBare(TestCase.java:134)
       at junit.framework.TestResult$1.protect(TestResult.java:110)
       at junit.framework.TestResult.runProtected(TestResult.java:128)
       at junit.framework.TestResult.run(TestResult.java:113)
       at junit.framework.TestCase.run(TestCase.java:124)
       at junit.framework.TestSuite.runTest(TestSuite.java:243)
       at junit.framework.TestSuite.run(TestSuite.java:238)
       at
org.junit.internal.runners.JUnit38ClassRunner.run(JUnit38ClassRunner.java:83)
       at
org.apache.MAVEN.surefire.junit4.JUnit4TestSet.execute(JUnit4TestSet.java:59)
       at
org.apache.MAVEN.surefire.suite.AbstractDirectoryTestSuite.executeTestSet(Abstract
DirectoryTestSuite.java:120)
       at
org.apache.MAVEN.surefire.suite.AbstractDirectoryTestSuite.execute(AbstractDirecto
```

```
ryTestSuite.java:103)
       at org.apache.MAVEN.surefire.Surefire.run(Surefire.java:169)
       at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
       at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
       at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
25)
       at java.lang.reflect.Method.invoke(Method.java:597)
       at
org.apache.MAVEN.surefire.booter.SurefireBooter.runSuitesInProcess(SurefireBooter.
java:350)
       at
org.apache.MAVEN.surefire.booter.SurefireBooter.main(SurefireBooter.java:1021)
 INFO [main] (IPMPEngineTest.java:97)-

---------- testIPMPGeneration ----------
 INFO [main] (MXM.java:158)-  ------- Multimedia Extensible Middleware -------
 INFO [main] (MXM.java:159)-  - Version 0.0.1
 INFO [main] (MXM.java:160)-  - MXM Configuration file:
C:\CONVERGENCE\Work\RefactoringEW\mxm\mxm-engines\mxm-ipmpTE\target\test-
classes\MXMConfiguration.xml
 INFO [main] (MXM.java:167)- Initializing parameters...
 INFO [main] (MXM.java:171)- Initializing engines...
 INFO [main] (MXMEngineLoader.java:88)- Loaded engine: RELTE
 INFO [main] (MXMEngineLoader.java:113)- Loaded engine: IPMPTE
 INFO [main] (MXM.java:175)- DONE Initializing...
 WARN [main] (MXM.java:264)- No MXM Parameters were set.
 INFO [main] (MXMAbstractEngine.java:59)-
-------------- MXM Engine --------------
 INFO [main] (MXMAbstractEngine.java:60)- -- Engine name: RELTE --
 INFO [main] (MXMAbstractEngine.java:61)- -- Engine ID: 0
 WARN [main] (MXMAbstractEngine.java:68)- No MXM Parameters were set.
 INFO [main] (MXMAbstractEngine.java:59)-
-------------- MXM Engine --------------
 INFO [main] (MXMAbstractEngine.java:60)- -- Engine name: IPMPTE --
 INFO [main] (MXMAbstractEngine.java:61)- -- Engine ID: 2
 WARN [main] (MXMAbstractEngine.java:68)- No MXM Parameters were set.
DEBUG [main] (RELEngineContext.java:111)- Generating RELEngine JAXB Context...
DEBUG [main] (RELEngineContext.java:113)- Done!
 INFO [main] (IPMPEngineTest.java:139)- The IPMP file was successfully created in
file: C:\Users\AZHAMM~1\AppData\Local\Temp\\myIPMP.xml
Tests run: 2, Failures: 0, Errors: 1, Skipped: 0, Time elapsed: 7.858 sec <<<
FAILURE!

Results :

Tests in error:
  testIPMPParsing(org.iso.mpeg.ipmpTE.test.IPMPEngineTest)

Tests run: 2, Failures: 0, Errors: 1, Skipped: 0

[INFO] ------------------------------------------------------------------------
[INFO] BUILD FAILURE
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 15.882s
[INFO] Finished at: Tue Oct 02 18:16:27 EEST 2012
```

```
[INFO] Final Memory: 3M/7M
[INFO] ------------------------------------------------------------------------
[ERROR] Failed to execute goal org.apache.MAVEN.plugins:MAVEN-surefire-
plugin:2.6:test (default-test) on project mxm-ipmpTE: There are test failures.
[ERROR]
[ERROR] Please refer to C:\CONVERGENCE\Work\RefactoringEW\mxm\mxm-engines\mxm-
ipmpTE\target\surefire-reports for the individual test results.
[ERROR] -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run MAVEN with the -e
switch.
[ERROR] Re-run MAVEN using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read
the following articles:
[ERROR] [Help 1]
http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
```