Project Number:	**FP7-257123**
Project Title:	**CONVERGENCE**
Deliverable Type:	**Report**
Dissemination Level	**Public**
Deliverable Number:	**D7.1**
Contractual Date of Delivery to the CEC:	**30.06.2011**
Actual Date of Delivery to the CEC:	**18.07.2011**
Title of Deliverable:	**Tools and Sample Applications Plans and Vision**
Work-package contributing to the Deliverable:	**WP 7**
Nature of the Deliverable:	**Report**
Editors:	**Stelios Pantelopoulos and Panagiotis Gkonis**
Author(s):	**Stelios Pantelopoulos, Panagiotis Gkonis, Anastasios Tsopanoglou, Dimitrios Tsirikos (SIL), Aling Hang (LMU), Francis Lemaitre (FMSH), Leonardo Chiariglione, Angelo Difino (CEDEO), Iakovos Venieris, Dimitra Kaklamani, George Lioudakis, Konstantinos Papadopoulos, Nikolaos Dellas, Angelos – Christos Anadiotis, Aziz Mousas, Charalampos Patrikakis (ICCS), Daniel Sequeira (WIPRO), Leonardo Lemmi, Andrea de Polo (ALINARI), Mihai Tanase, Lucian Corlan (UTI).**
Abstract:	**The goal of this first deliverable from WP7 is to provide a general vision and functional specifications for the tools and sample applications that will be used in the CONVERGENCE real life trials**
Keyword List:	**Tools, applications, roadmap, trials, walkthroughs, GUI, APIs, prototypes, Java middleware, network**

Executive Summary

The goal of this first deliverable from WP7 is to provide a general vision and functional specifications for the tools and sample applications that will be used in the CONVERGENCE real life trials. The deliverable begins by presenting an agreed roadmap for the implementation of the tools and applications needed by the trials. It goes on to identify common functionalities present in the four scenarios to be implemented in the trials [1], to specify the server – client engines needed to implement the scenarios and to provide the corresponding walkthroughs and BPMN diagrams. On this basis, it identifies a set of tools required by all the scenarios: User Registration, Content Registration, Publish VDI, Unpublish VDI, Revoke VDI, Subscription to VDI, Browse Event Report and Create Annotation VDI. The analysis is complemented with preliminary GUI designs for applications.

Moving from design towards implementation, the deliverable maps the tools to applications, analyzes how tools and applications will be integrated with the CONVERGENCE middleware, and describes the implementation framework for Java-based and Browser-based applications. Reviewers are asked to note that, for ease of reading, parts of the functional descriptions presented in this deliverable will also be included in deliverable D.8.1: “Plans for trials and experimentations”; however, duplicate parts will be kept to a minimum.

TABLE OF CONTENTS

TABLE OF FIGURES

1 Introduction

The goal of this deliverable is to define a general vision and functional specifications for the tools and sample applications to be used in the CONVERGENCE real life trials. These tools and applications will serve as a gateway between the needs of the user and the functionalities provided by CONVERGENCE. In order to give more flexibility to CONVERGENCE users, the applications level is split into two layers: a tools layer and a user applications layer. Tools consist of code that can be reused by many applications (e.g. for data input, visualization of search results) but is not necessarily used by all of them. A typical tool interacts with CoMid by calling an appropriate set of Technology Engines.

Reviewers are asked to note that the definition of a tool is a matter of convenience only and that are no golden rules for deciding whether a given portion of an application will be treated as a tool.

The tools and applications described in this deliverable are based on:

1. The functional requirements and use cases identified in WP2 [2]
2. The technical specifications for Versatile Digital Items (VDIs) in WP4 [2],
3. The CONVERGENCE middleware protocols described in WP5 [3]
4. Technology Engines developed in WP6.

The role of applications and tools, and their interactions with Protocol and Technology Engines is depicted in Figure *1-1*.

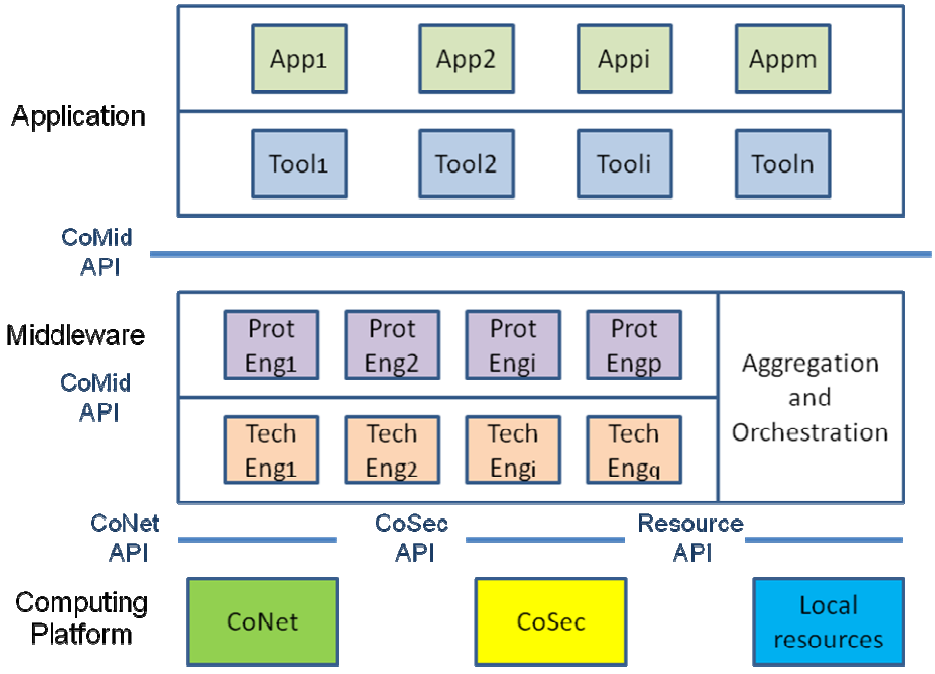

Figure 1-1: Architecture of a CONVERGENCE peer

1.1 Implementation Roadmap

In general, the tasks associated with WP7 can be summarized as follows:

- **T.7.1** will design and create technical specifications for the tools and sample applications, which will be used to validate the CONVERGENCE concept and implementation.
- **T.7.2** will implement the tools and sample applications specified in T7.1.
- **T.7.3** will perform technical tests to identify bugs and verify the performance of the tools and applications and to implement required corrective measures.

To proceed towards the implementation of the early and final prototypes in a failsafe way, WP7 will adopt the staged approach illustrated in Figure 1-2.

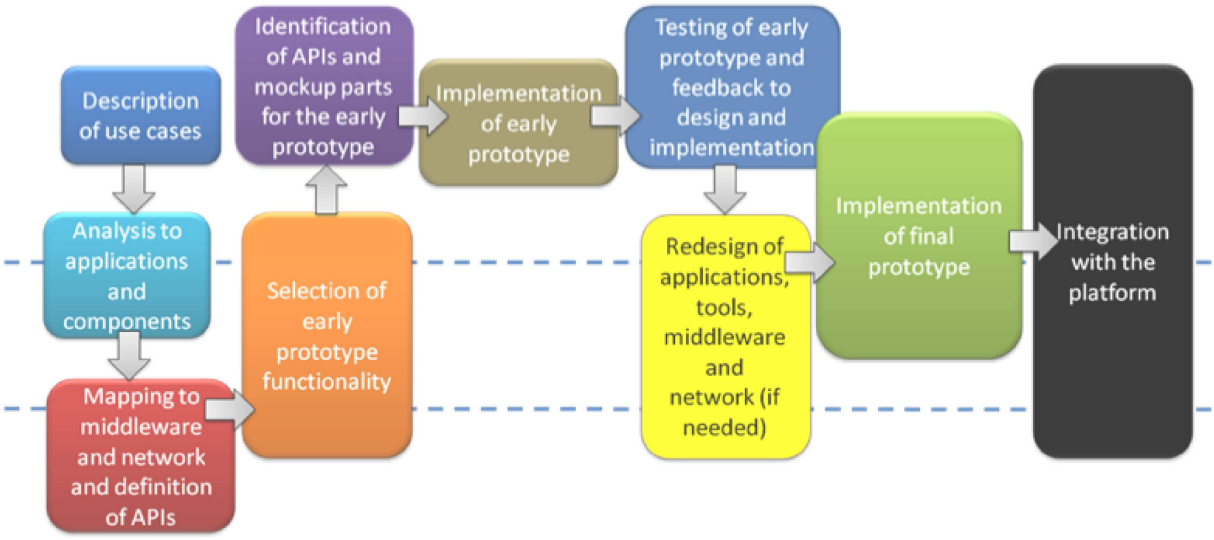

Figure 1-2: From use case description to integration

The main stages in the process can be summarized as follows:

1. **Description of use case**; analysis of the functional requirements of each use case
2. **Analysis in terms of applications and components**: identification of the applications and components necessary to implement the scenario
3. **Mapping to middleware and network, and definition of APIs**: Identification of the middleware functionality required for each use case and the way applications will use middleware APIs
4. **Selection of early prototype functionality**: design of an early prototype including identification of all required functionality
5. **Identification of APIs and partial mockups of the prototype**: creation of mockups of critical parts of the prototype including the interaction with the middleware
6. **Implementation of early prototypes**
7. **Testing of early prototype and feedback to design and implementation**. Definition and implementation of test cases; reporting of results
8. **Redesign of applications, tools, middleware and network (if required)**: specification of changes required to resolve problems identified in testing
9. **Implementation of final prototype**s
10. **Integration with the platform**: delivery of the final prototypes ready for integration in a single CONVERGENCE platform.

This deliverable covers the first two steps, namely the description of use cases and the analysis in terms of applications and components and includes a discussion of issues related to the integration of tools and applications in CoMid.

1.2 Contents of D.7.1

The rest of this deliverable is organized as follows.

Chapter 2: This chapter analyzes common functionalities required by all the scenarios and that will be implemented as CONVERGENCE tools. It goes on to map them to a list of Technology Engines.

Chapters 3, 4, 5, 6: Each of these chapters is dedicated to an analysis of functional requirements for one of the user scenario and provides:

1) A brief description of the scenario
2) A bird's eye view of scenario in terms of basic tools and applications components
3) A description of the technologies that will be used in the implementation of the scenario
4) A functional overview of relevant CoMid Interfaces

For each scenario, the description includes BPMN diagrams and preliminary ideas on GUI design.
Reviewers will observe that at least two scenarios require payment support. Discussions are in progress to decide whether the trials should use a single payment technology or whether each of them will choose different technologies, according to their specific needs.

Chapter 7: This chapter addresses the issue of integration between applications and middleware engines and provides details of the applications that will be integrated with CoMid.

Chapter 8: This chapter summarizes the key findings reported in the deliverable.

2 Basic CONVERGENCE Tools

2.1 Introduction

In this chapter, we analyze basic functionalities required by all CONVERGENCE applications. Each of these functionalities will be implemented as a CONVERGENCE tool. In each case, the user interface to the tool is provided by the application, with different applications presenting different interfaces. Elements in applications call client engines, which initiate the procedure by communicating with server engines. For each procedure we present a brief outline of the required functionality, a walkthrough, a table showing the sequence of engines called during the procedure (server - S, client - C, Peer - P) ([3]-[4]) and a BPMN diagram [5].

Reviewers are asked to note that the list of tools presented here is non-exhaustive and that future work may identify other tools. We recall here that CONVERGENCE has defined, so far, 4 types of VDI: Resource (R-VDI), Publication (P-VDI), Subscription (S-VDI) and User (U-VDI) [4] .

2.2 User registration

This tool allows the system to identify a CONVERGENCE user, the first time she connects to the system, and to authenticate her on subsequent occasions.

2.2.1 Walkthrough

1. User identifies herself to CONVERGENCE the first time she connects to the system
2. User authenticates herself to CONVERGENCE on subsequent occasions

2.2.2 Sequence of Engines

Application elements	Client Engines	Server Engines
User Identification		
	Identify User PE	Identify User PE
		Security TE
User Authentication		
	Authenticate User PE	Authenticate User PE
		Security TE

Table 2-1: Engine sequence for user registration

2.2.3 BPMN diagrams

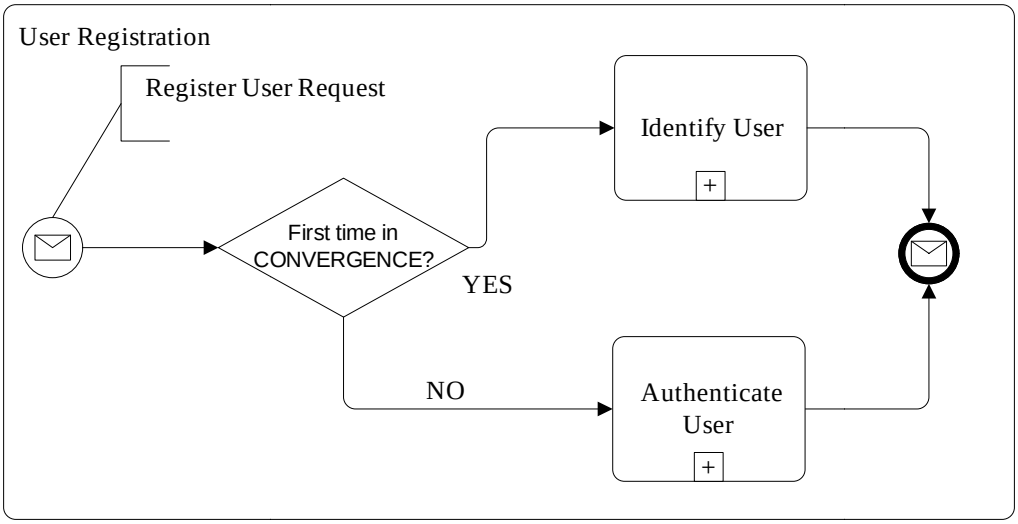

Figure 2-1: User registration workflow

2.2.4 List of Engines

Technology type	Technology	S	C
Protocol Engines			
	Identify User	X	X
	Authenticate User	X	X
Technology Engines			
	Security	X	

***Table 2-2*:** Engines for user registration

2.3 Content registration

This tool provides an identifier for a VDI or for a component of a VDI. The identifier allows authentication of the VDI on later occasions.

2.3.1 Walkthrough

1. User invokes a Content Identification service providing requested data about the content to be identified

2. Content Identification service provides requested identifier

2.3.2 Sequence of Engines

Application elements	Client Engines	Server Engines
Content Identification		
	Identify Content PE	Identify Content PE
Content Authentication		
	Authenticate Content PE	Authenticate Content PE
	Security TE	Security TE

Table 2-3: Engine sequence for content registration

2.3.3 BPMN diagrams

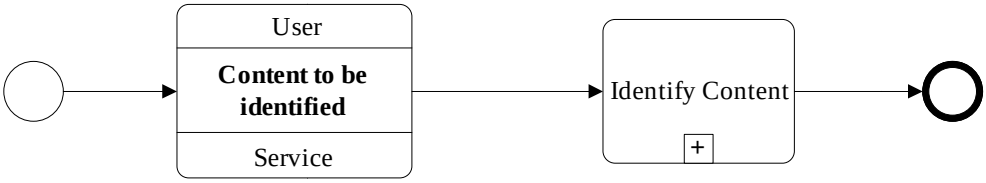

Figure 2-2: Content registration workflow

2.3.4 List of Engines

Technology type	Technology	S	C
Protocol Engines			
	Identify Content	X	X
	Authenticate Content	X	X
Technology Engines			
	Security	X	X

Table 2-4: Engines for content registration

2.4 Publish VDI

This user allows a user who has successfully created a VDI to create and inject a Publication VDI (P-VDI).

2.4.1 Walkthrough

Client side:

1. User
    a. Provides
        i. P-VDI metadata
        ii. License
        iii. ERR
    b. Gets identifier of P-VDI
    c. Gets signature of P-VDI
    d. Injects P-VDI

In the cloud:

1. Peer in the cloud
    a. Receives P-VDI
    b. Opens P-VDI
    c. Extracts
        i. Metadata
        ii. License
        iii. ERR
    d. Stores extracted data in “match table”
    e. Matches Publication VDI with any outstanding Subscription VDI
    f. If match exists checks that ER destination in ERR is in the License’s principal list
    g. If yes
        i. Makes ER
        ii. Sends ER (notification)

2.4.2 Sequence of Engines

Application elements	Client Engines	Server Engines	Peer in the Cloud
Publication VDI Creation			
	Request Content PE	Request Content PE	
		REL TE	
		Security TE	
	Describe Content PE	Describe Content PE	
		CDS TE	
		Metadata TE	
	Create License PE	Create License PE	
		REL TE	
		Security TE	
	ER TE		
	Create Content PE	Create Content PE	

		VDI TE	
	Identify Content PE	Identify Content PE	
	Security TE		
Store P-VDI in CoNet			
	Package Content PE	Package Content PE	
		MP21FF TE	
	Store Content PE	Store Content PE	
		CoNet TE	
Send P-VDI to peer in the cloud			
	Inject Content PE	Inject Content PE	
		Overlay TE	
		CoNet TE	
Processing in the cloud			
			Overlay TE
			CoNet TE
		Request Content PE	Request Content PE
		REL TE	
		Security TE	
			VDI TE
			Metadata TE
			Match TE
			CDS TE
			ER TE
		Store Event PE	Store Event PE
		ER TE	

Table 2-5: Engine sequence for publish VDI

2.4.3 BPMN diagrams

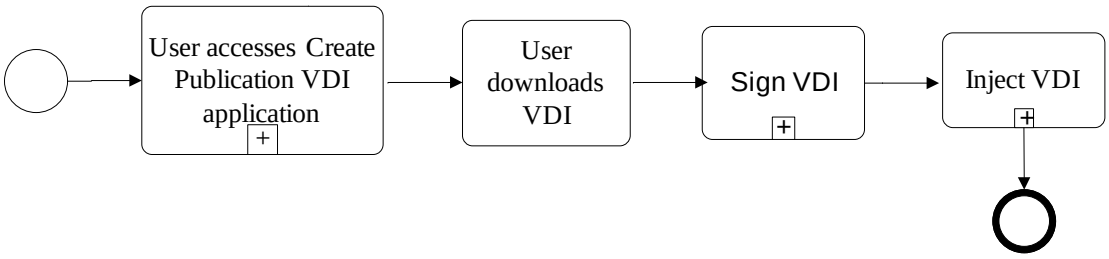

Figure 2-3: Publication of VDI workflow

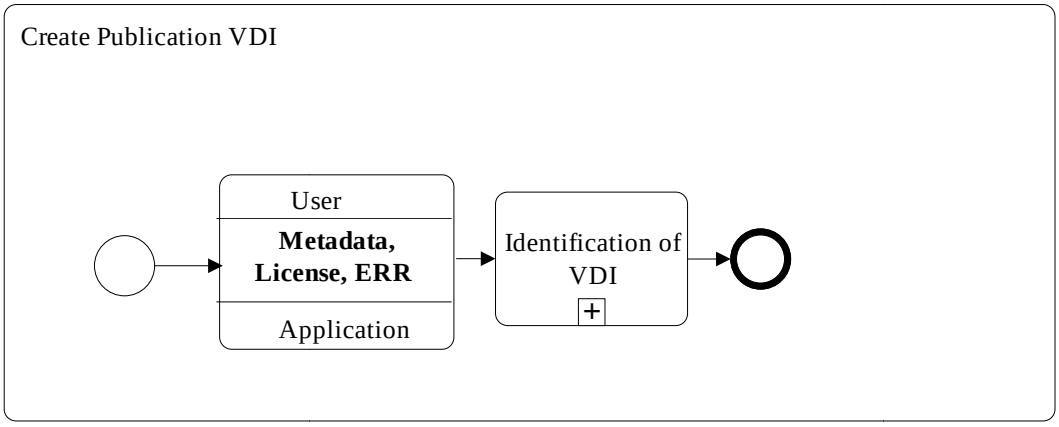

Figure 2-4: Create Publication VDI application workflow

2.4.4 List of Engines

Technology type	Technology	S	C	P
Protocol Engines	Identify Content	X	X	
	Create Content	X	X	
	Store Content	X	X	
	Inject Content	X	X	
	Describe Content	X	X	
	Create License	X	X	
	Request Content	X	X	X
	Store Event	X		X
	Package Content	X	X	
Technology Engines	VDI	X		X
	CDS	X		X
	Metadata	X		X
	REL	X		
	ER	X	X	X

	Security	X	X	
	Overlay	X		X
	CoNet	X		X
	MP21FF	X		
	Match			X

Table 2-6: Engines for publishing VDI application

2.5 Unpublish VDI

With the help of this tool, a CONVERGENCE user can revoke a P-VDI.

2.5.1 Walkthrough

1. User sends unpublish message to appropriate peers in the cloud
2. Peer in the cloud
    a. Receives message
    b. Removes all previously sent data related to the Publication VDI from the match table
    c. Sends confirmation message to rights holder's peer

2.5.2 Sequence of Engines

Application elements	Client Engines	Server Engines	Peer in the cloud
Send unpublish message			
	Revoke Content PE	Revoke Content PE	
		REL TE	
		Security TE	
		Overlay TE	
		CoNet TE	
Processing in the cloud			
			Overlay TE
			CoNet TE
			Match TE

Table 2-7: Engine sequence for unpublishing VDI

2.5.3 BPMN diagrams

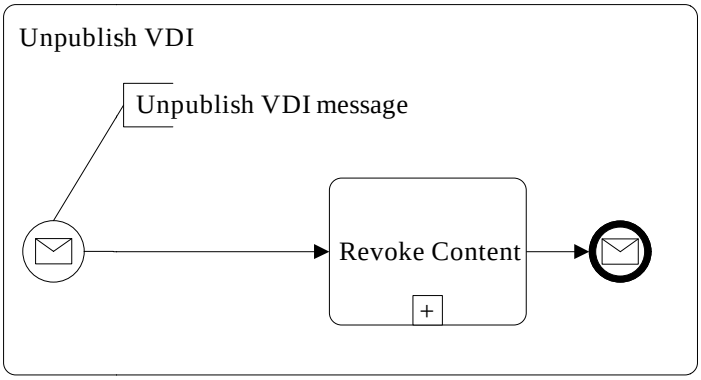

Figure 2-5: Unpublish VDI application workflow

2.5.4 List of Engines

Technology type	Technology	S	C	P
Protocol Engines	Revoke Content	X	X	
Technology Engines	Overlay	X		X
	CoNet	X		X
	Match			X
	REL	X		
	Security	X		

Table 2-8: Engines for unpublishing VDI application

2.6 Revoke VDI from CoNet

This tool allows a CONVERGENCE user to revoke a Resource VDI.

2.6.1 Walkthrough

1. Authentication
2. Send a revoke Resource VDI request
3. CoNet revokes the VDI

2.6.2 Sequence of Engines

Application elements	Client Engines	Server Engines
Revocation request		
	Authenticate User PE	Authenticate User PE
		Security TE
	Revoke Content PE	Revoke Content PE
		CoNet TE

Table 2-9: Engine sequence for revoking VDI

2.6.3 BPMN diagrams

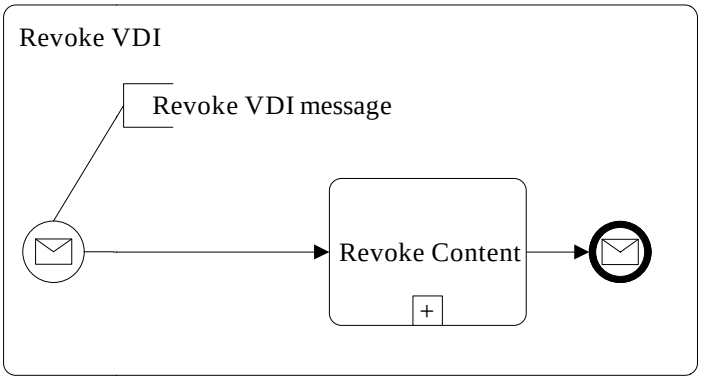

Figure 2-6: Revoke VDI application workflow

2.6.4 List of Engines

Technology type	Technology	S	C
Protocol Engines	Authenticate User	X	X
	Revoke Content	X	X
Technology Engines	Security	X	
	CoNet	X	

Table 2-10: Engines for revoking VDI

2.7 Subscription to VDI

This tool allows a user to subscribe to VDIs that match her interests.

2.7.1 Walkthrough

1. User
    a. Provides data for
        i. Metadata
        ii. License
        iii. ERR
    b. Gets Identifier for S-VDI
    c. Gets Signature of S-VDI
    d. Injects S-VDI
2. Peer in the cloud
    a. Receives Subscription VDI
    b. Opens Subscription VDI
    c. Extracts and stores in “match table”
        i. Metadata
        ii. License
        iii. ERR
    d. Matches Subscription VDI with any outstanding Publication VDI
    e. If match exists checks that ER destination in ERR is within the principal list in license
    f. If yes makes and sends ER (notification)

2.7.2 Sequence of Engines

Application elements	Client Engines	Server Engines	Peer in the cloud
Subscription VDI Creation			
	Describe Content PE	Describe Content PE	
		Metadata TE	
		CDS TE	
	Create License PE	Create License PE	
		REL TE	
		Security TE	
	ER TE		
	Create Content PE	Create Content PE	
		VDI TE	
	Identify Content PE	Identify Content PE	

	Security TE		
Store S-VDI in CoNet			
	Package Content PE	Package Content PE	
		MP21FF TE	
	Store Content PE	Store Content PE	
		CoNet TE	
Send to peer in the cloud			
	Inject Content PE	Inject Content PE	
		Overlay TE	
		CoNet TE	
Processing in the cloud			
			Overlay TE
			CoNet TE
		Request Content PE	Request Content PE
		REL TE	
		Security TE	
			VDI TE
			Metadata TE
			Match TE
			CDS TE
			ER TE
		Store Event PE	Store Event PE
		ER TE	

Table 2-11: Engine sequence for subscription to VDI application

2.7.3 BPMN diagrams

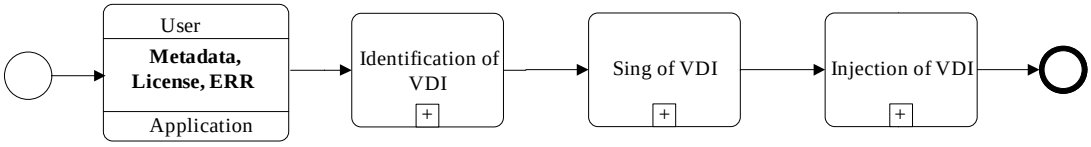

Figure 2-7: Create of subscription to VDI application workflow

2.7.4 List of Engines

Technology type	Technology	S	C	P
Protocol Engines	Identify Content	X	X	
	Create Content	X	X	
	Describe Content	X	X	
	Inject Content	X	X	
	Store Content	X	X	X
	Create License	X	X	
	Request Content	X		X
	Package Content	X	X	
Technology Engines	CDS	X		X
	Metadata	X	X	X
	REL	X	X	
	ER	X	X	X
	Security	X	X	
	Overlay		X	X
	Match			X
	CoNet	X		X

Table 2-12: Engines for subscription to VDI application

2.8 Browse Event Report

This tool allows a CONVERGENCE user to browse a previously created event report.

2.8.1 Walkthrough

1. User
    a. Authenticates, as described in User registration
    b. Request to the Service Provider the ERs based on his User identifier
    c. Browses the received ERs
2. Service Provider
    a. Receives the Event Report Request (ERR) form the User
    b. Searches for ERs related to this ERR
    c. Sends the set of ERs to the User

2.8.2 Sequence of Engines

Application elements	Client Engines	Server Engines
ER browsing		
	Request Event PE	Request Event PE
		Event Reporting TE

Table 2-13: Engines for browse event report application

2.8.3 BPMN diagrams

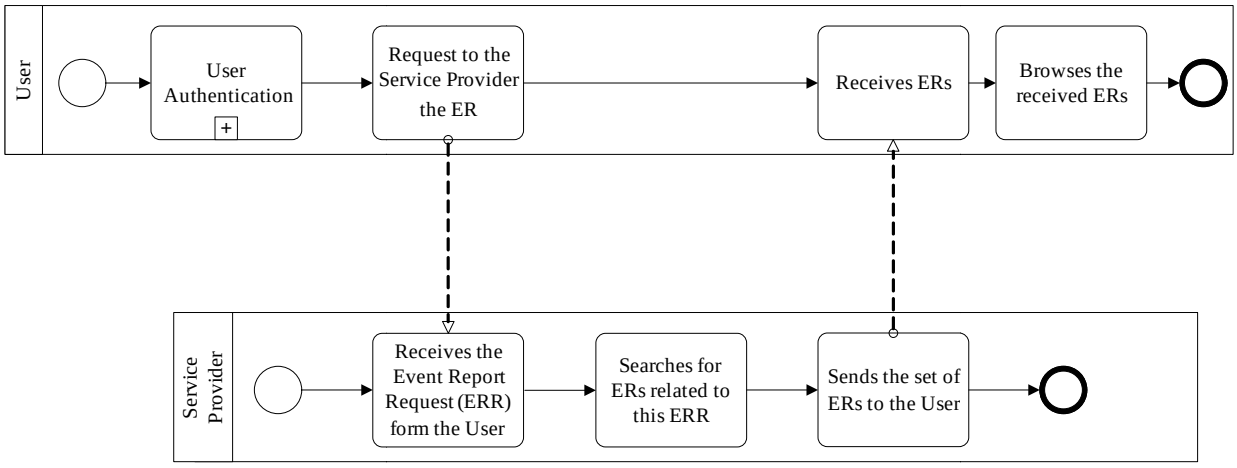

Figure 2-8: Browse event report application work flow

2.8.4 List of Engines

Technology type	Technology	S	C
Protocol Engines	Request Event	X	X
Technology Engines	Event Reporting	X	

Table 2-14: Engines for browse event report application

2.9 Create Annotation VDI

This tool helps CONVERGENCE users to create annotations based on specific resource VDIs.

2.9.1 Walkthrough

1. User:
    a. Provides data for
        i. Metadata (application-related metadata are added as well)
        ii. License (Recipients of Annotation)
        iii. ERR
    b. Embeds Text resource
    c. Gets Identifier for Annotation
    d. Signs VDI
    e. Packages VDI as a file
    f. Stores Annotation VDI
2. User uses the publish VDI tool described in Publish VDI

2.9.2 Sequence of Engines

Application elements	Client Engines	Server Engines
Annotation Creation		
	Describe Content PE	Describe Content PE
	Metadata TE	Metadata TE
		CDS TE
	Create License PE	Create License PE
		REL TE
		Security TE
Content Registration	CONVERGENCE Tool	
Signature of VDI		
	Security TE	

Packaging of VDI		
	MP21FF TE	
Storage of VDI		
	Store Content PE	
	CoNet	
Publish Annotation VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 2-15: Engine sequence for create annotation application

2.9.3 BPMN diagrams

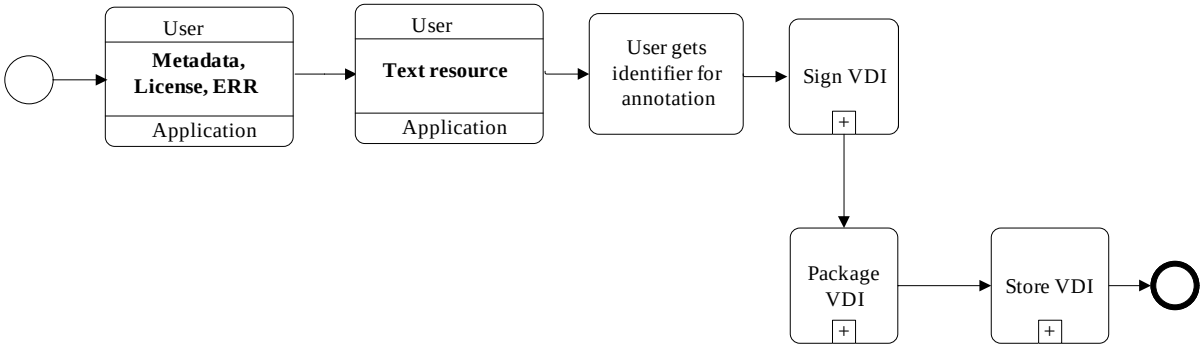

Figure 2-9: Creation of annotation VDI application work low

2.9.4 List of Engines

Technology type	Technology	S	C
Protocol Engines	Identify Content	X	X
	Create Content	X	X
	Describe Content	X	X
	Inject Content	X	X
	Store Content	X	X
	Create License	X	X
	Package Content	X	X
Technology Engines	CDS	X	
	Metadata	X	

	REL	X	
	Security	X	
	CoNet	X	

Table 2-16: Engines for create annotation VDI application

2.10 Summary of engines

Technology type	Technology	S	C	P
Protocol Engines				
	Identify User	X	X	
	Authenticate User	X	X	
	Identify Content	X	X	
	Authenticate Content	X	X	
	Create Content	X	X	
	Store Content	X	X	X
	Inject Content	X	X	
	Describe Content	X	X	
	Create License	X	X	
	Request Content	X	X	X
	Store Event	X		
	Package Content	X	X	
	Revoke Content	X	X	
	Request Event	X	X	
Technology Engines				
	Security	X	X	
	VDI	X		X
	CDS	X		X
	Metadata	X	X	X
	REL	X	X	
	ER	X	X	X
	Overlay	X	X	X
	CoNet	X		X
	MP21FF	X		
	Match			X
	Event Report	X		
	Present Content		X	

Table 2-17: Engines for basic applications

3 Photos in the Cloud and Analyses on the Earth (real world trial 1)

3.1 Brief description of use case

This use case tests whether CONVERGENCE can provide useful support for new business models in the photography business. The goal is to make it easier for photographers to contribute and describe photos, improve access for users and generally facilitate the management of relevant services. All photographs are represented by VDIs, each containing:

1. The photo itself (or a link to the photo), possibly accompanied by a low-resolution version of the photo
2. Metadata describing the photo defined by Alinari and including the date, time and place where the photo was taken, legal data on the author and owner of the photo, technical data about the photo (camera, lens, shutter time, aperture, ISO etc.), historical data about the site represented in the photo, and other metadata contributed by Alinari staff and third parties
3. Licensing information representing the conditions at which photos can be licensed by a photographer to Alinari and by Alinari to an end user. Licensing information is represented using the CONVERGENCE REL.

Alinari manages the server used in the trial and the photo archive using a dedicated server that runs custom applications on top of the CONVERGENCE framework and the CONVERGENCE network. The dedicated applications provide Alinari staff with a user interface making it easy for them to create, publish, un-publish, describe and update the photos and VDIs. Free-lance photographers can access the Alinari service to create Resource VDIs and Publication VDIs. End-users can subscribe to photos and possibly buy photos using a local application connected to the Alinari server. The CONVERGENCE framework automatically prevents access to photos that have expired and performs garbage collection to purge expired copies from network storage.

3.2 Bird's eye view of the deployment framework

The "Alinari scenario" involves:

1. Three types of users:
    1. Photographer
        i. Freelance, independent of Alinari
        ii. With a business relationship (e.g. registered) with Alinari
    2. Alinari photo archive manager
    3. Customer of Alinari photo licensing service
2. The following devices:
    1. Alinari server containing copies of current Alinari photos and metadata in CONVERGENCE format (in this deliverable called Alinari* server)
    2. Cloud of CONVERGENCE peers in the overlay
    3. Various clients (peers) used by participants

3. The following data:
    1. Photos from the Alinari server and Photo VDIs created from the Alinari photo metadata
    2. Photos and Photo VDIs created by photographers
    3. Processed Photos and validated Photo VDIs created by Alinari personnel (called Photo* and Photo VDI*, respectively) from Alinari's and photographers' Photos and Photo VDIs
4. The following applications:
    1. User registration
    2. Photographer publishes Photo VDI
    3. Rights holder unpublishes Photo VDI
    4. Alinari subscribes and downloads photo
    5. Photographer uploads Photo VDI to Alinari server
    6. Alinari publishes new Photo VDI
    7. Customer purchases interesting photos
    8. Customer visits Alinari Museum

Figure 3-1 depicts the role of CONVERGENCE, users and apps.

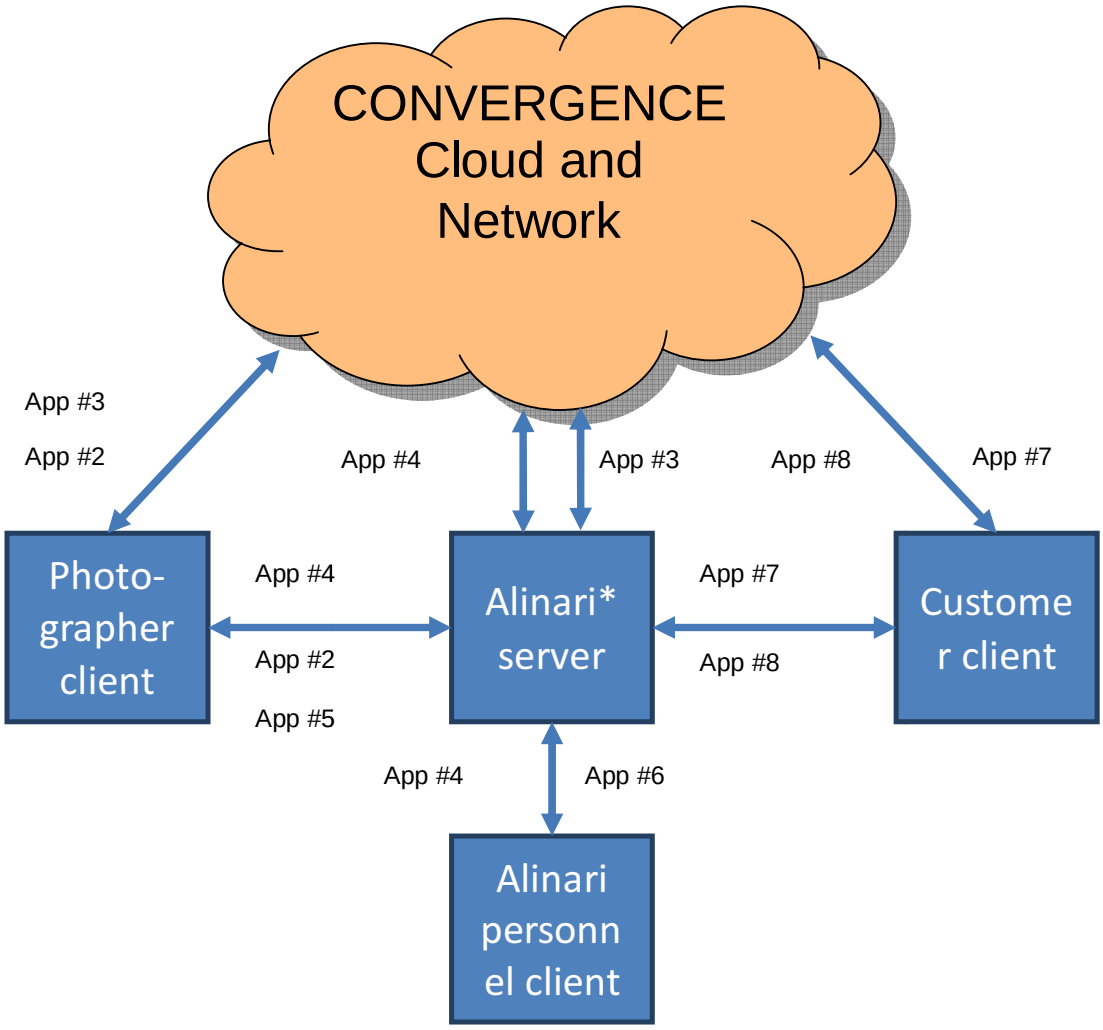

Figure 3-1: Schematic description of Alinari scenario

Our analysis is based on the following assumptions:

1. Alinari runs a photo licensing service that aims to:
    a. Create Photo VDIs of the photos/metadata stored in the Alinari* server
    b. Collect Photos (resources) and Photo VDIs from photographers who are
        i. registered with Alinari and create/store their Photo VDIs to the Alinari* server
        ii. freelance and publish their Photos and Photo VDIs on the CONVERGENCE cloud
    c. Artistically refine those photos, clear any rights, create Photo VDIs
    d. Publish Photo VDIs to the CONVERGENCE cloud.
2. Alinari owns the Alinari Museum and provides apps helping visitors to design their own personal visit
3. Customers:
    a. Search the CONVERGENCE cloud to find photos of their interest
    b. Purchase consumer and professional Licenses of photos of their interest
    c. Design their own personal visit to the Alinari Museum
4. Photographers use an Alinari application to create Photo VDIs of their photos:
    a. If registered with Alinari
    b. If not registered with Alinari (for reasons of expediency in the CONVERGENCE trials, otherwise we would need to set up another server providing the needed independent services; however in case b the GUI will be different, see section on GUIs, 3.5).

3.3 Technologies

The trial will use a custom application integrated with native CONVERGENCE technologies.

3.4 Functional overview and CoMid interfaces

3.4.1 User registration

Users will need to register (i.e. identify) themselves with CONVERGENCE. To do this, they will use an instance of the User Registration tool as described in User registration.

3.4.2 Photographer publishes Photo VDI

This application allows users (typically photographers) to publish Photo VDIs to the CONVERGENCE cloud. In a real scenario, the creation of the VDI could be handled by a third party service. In the trial, the VDI creation service will be provided by Alinari.

3.4.2.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Photo VDI Creation		

	Create Content PE	Create Content PE
		CDS TE
		REL TE
		ER TE
		Security TE
Download Photo VDI		
	Request Content PE	Request Content PE
Package Photo VDI		
	MPEG-21 FF TE	
Store Photo VDI		
	Store Content PE	
Publish VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 3-1: Engine sequence for publishing a photo VDI by a photographer application

3.4.2.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Authenticate User PE	X	X
	Identify Content PE	X	X
	Create Content PE	X	X
	Store Content PE		X
	Inject Content PE		X
Technology Engines	CDS TE	X	
	Metadata TE	X	X
	REL TE	X	X
	ER TE	X	X
	Security TE	X	X
	Overlay TE		X

Table 3-2: Engines for publishing a photo VDI by a photographer application

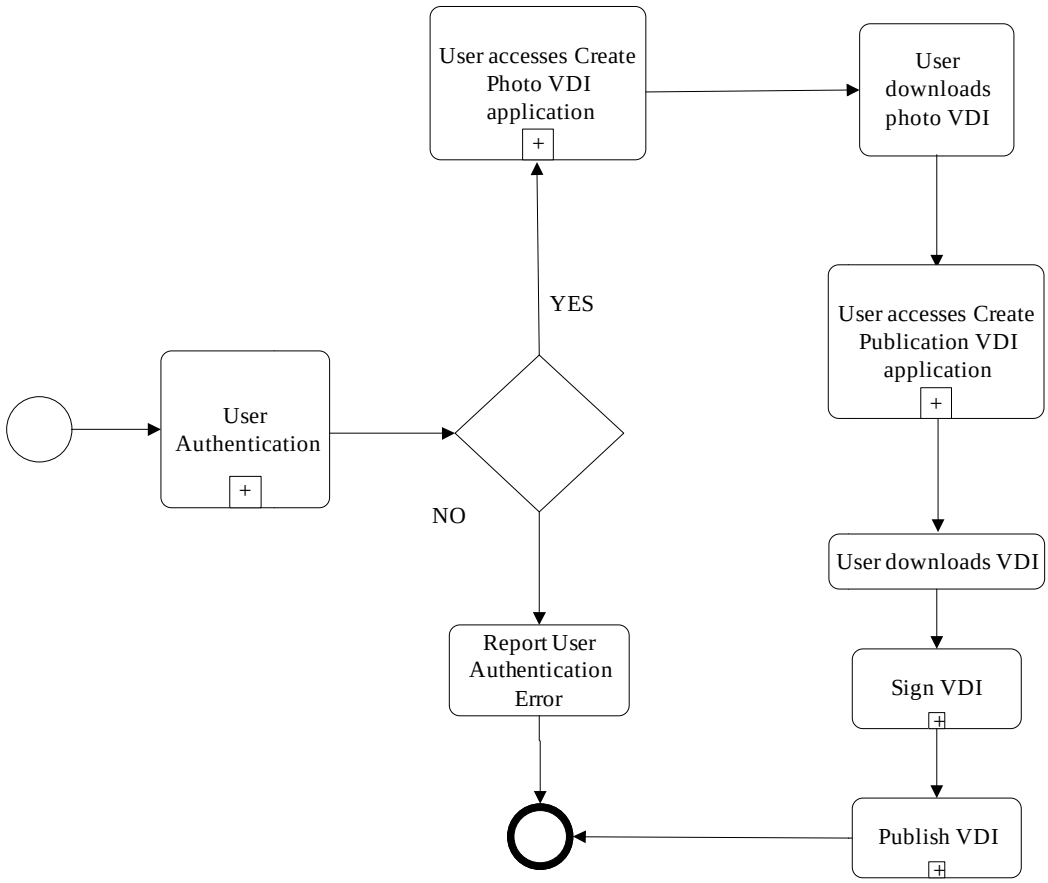

Figure 3-2: Publication of a Photo VDI by a photographer: workflow

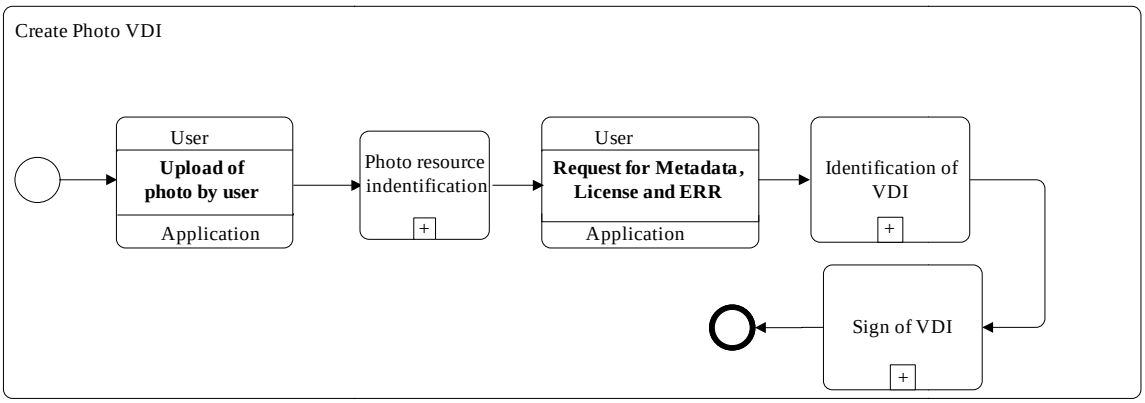

Figure 3-3: Creation of photo VDI application: workflow

3.4.3 Rights holder unpublishes Photo VDI

This application will allow a rights holder (Photographer, Alinari employee) to unpublish (revoke) a Photo VDI.

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Revoke P-VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 3-3: Engine sequence for unpublishing a photo VDI by a rights holder application

3.4.4 Alinari subscribes and downloads photo

This application – running on a corporate Alinari PC and interacting with a companion application in the server – will allow an Alinari photo archive manager to subscribe to, and download Photo VDIs satisfying a set of search criteria.

3.4.4.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Subscription to VDI	CONVERGENCE Tool	CONVERGENCE Tool
Download Photo VDI		
	Request Content PE	Request Content PE
View Photo		
	Process Content PE	
	Metadata TE	
	REL TE	
	Media Framework TE	
	ER TE	
Store Photo VDI		Store Content
		CoNet TE

Table 3-4: Engine sequence for subscription by Alinari and download photo application

3.4.4.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Authenticate User PE	X	X
	Identify Content PE	X	X
	Create Content PE	X	X
	Describe Content PE	X	X
	Inject Content PE		X
	Request Content PE	X	X
Technology Engines	CDS TE	X	
	Metadata TE	X	X
	REL TE	X	X
	Media Framework TE		X
	ER TE	X	X
	Security TE		X
	Overlay TE		X
	CoNet TE	X	

Table 3-5: Engines for subscription by Alinari and download photo application

3.4.5 Photographer uploads Photo VDI

This application allows a photographer to create and upload a Photo VDI directly to the Alinari* server. From a user point of view, the procedure is the same as the procedure already illustrated in Figure 3-3.

3.4.5.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Photo VDI Creation		
	Create Content PE	Create Content PE
		Identify Content PE
		CDS TE
		REL TE
		ER TE
	Security TE	

Table 3-6: Engine sequence for photographer uploads photo VDI application

3.4.5.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Authenticate User PE	X	X
	Identify Content PE	X	X
	Create Content PE	X	X
Technology Engines	CDS TE	X	
	Metadata TE	X	X
	REL TE	X	X
	ER TE	X	X
	Security TE	X	X

Table 3-7: Engines for upload of photo VDI by a photographer application

3.4.6 Alinari publishes new Photo VDI

This application enables Alinari personnel to

1. Clean, enhance and edit a photo*
2. Validate a photo (clear rights etc.) and create a photo VDIs
3. Publish the Photo VDI* to the CONVERGENCE cloud.

3.4.6.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Photo VDI* Creation		
	Create Content PE	Create Content PE
		Identify Content PE
		CDS TE
		REL TE
		ER TE
		Security TE
		Package Content PE
		MPEG-21 FF TE
Publication VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 3-8: Engine sequence for publishing new photo VDI by Alinari application

3.4.6.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Authenticate User PE	X	X
	Identify Content PE	X	X
	Create Content PE	X	X
	Package Content PE	X	
	Inject Content PE		X
Technology Engines	CDS TE	X	
	Metadata TE	X	X
	REL TE	X	X
	ER TE	X	X
	Security TE	X	
	Overlay TE		X

Table 3-9: Engines for publishing of new photo VDI by Alinari application

3.4.7 Customer purchases interesting photos

This application leverages the CONVERGENCE cloud to reach a broader number of potential customers.

3.4.7.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Subscription to VDI Creation	CONVERGENCE Tool	CONVERGENCE Tool
Download Photo VDI		
	Request Content PE	Request Content PE
View Photo		
	Process Content PE	
	Metadata TE	
	REL TE	
	Media Framework TE	
	ER TE	

Store Photo VDI		

Table 3-10: Engine sequence for purchasing interesting photos by customer application

3.4.7.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Authenticate User PE	X	X
	Identify Content PE	X	X
	Create Content PE	X	X
	Describe Content PE	X	X
	Inject Content PE		X
	Request Content PE	X	X
Technology Engines	CDS TE	X	
	Metadata TE	X	X
	REL TE	X	X
	Media Framework		X
	ER TE	X	X
	Security TE		X
	Overlay TE		X

Table 3-11: Engines for customer purchases interesting photos application

3.4.8 Customer visits Alinari Museum

This application improves the user experience of visitors to the Alinari Museum by exploiting semantic linking of VDIs to let visitors design the museum visit that best fits their interests. The visitor typically starts with a “visit template” that she proceeds to customize. Users can learn about exhibits by browsing museum catalogues with annotations from other users. They can also search for photos by filling metadata, and upload photos taken during their visit.

3.4.8.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Design your museum visit		
	Process Content PE	
	VDI TE	

	Request Content PE	Request Content PE
	Create Content PE	Create Content PE
		Identify Content PE
		CDS TE
		REL TE
		ER TE
		Security TE
		Package Content PE
		MPEG-21 FF TE
	Request Content PE	
Publication VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 3-12: Engine sequence for customer visits Alinari museum

3.4.8.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Authenticate User PE	X	X
	Identify Content PE	X	X
	Process Content	X	X
	Create Content PE	X	X
	Package Content PE	X	
	Inject Content PE		X
Technology Engines	CDS TE	X	
	Metadata TE	X	X
	REL TE	X	X
	ER TE	X	X
	Security TE	X	
	Overlay TE		X

Table 3-13: Engines for customer visits Alinari museum

3.4.9 Summary of Engines

Table *3-14* summarizes all engines of this scenario.

Technology type	Technology	S	C
Protocol Engines	Identify Content	X	X
	Request Content	X	X
	Process Content		X
	Create Content	X	X
	Package Content	X	
	Inject Content	X	
	Revoke Content	X	
Technology Engines	Metadata		
	REL		X
	Media Framework		X
	CDS	X	X
	ER	X	
	Security	X	X
	CoNet	X	X
	MPEG-21 FF	X	
	Overlay	X	

Table 3-14: Engines for photos in the cloud and analysis on the earth scenario

3.5 GUI design

Figure 3-4 below shows preliminary GUI designs for the Alinari application environments. Typical users will include photographers, designers, art directors, museum curators, architects, historians, students, professors, publishers, editors, and journalists and will be expected to register with Alinari before using the service.

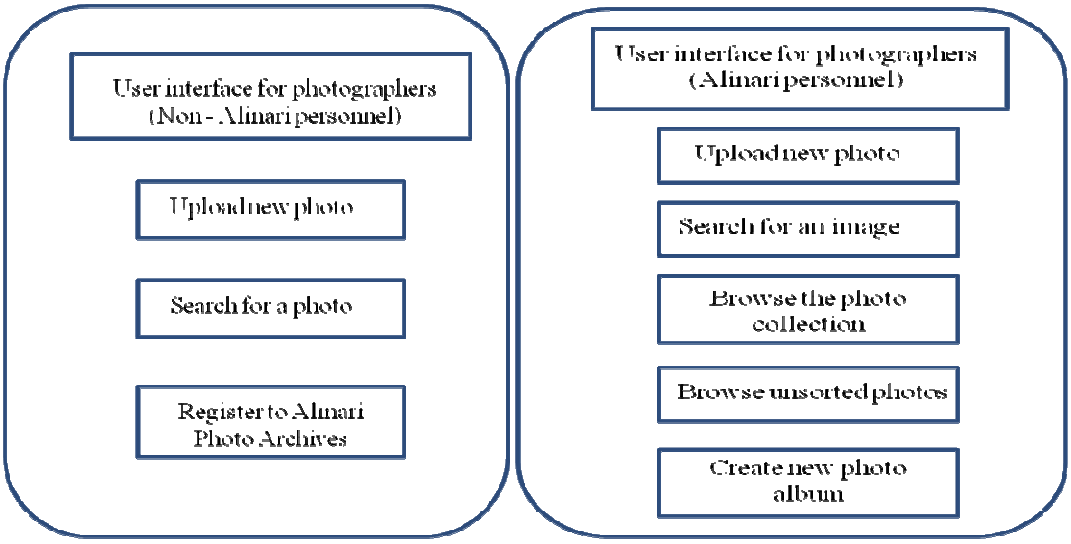

Figure 3-4: Application design for freelancers (left) and Alinari personnel (right)

Photographers who have no business relationship with Alinari will be able to access the following functionality.

- Upload new photo. The user uploads the photo, providing the data required to generate the metadata and the license. The application provides the same functionality, regardless of whether the user is uploading the photo to the cloud or directly to the Alinari server.
- Search for photos. The user can perform semantic-based searches for photos.
- Registration to the Alinari Photo Archives. The photographer fills in a form with personal data, which an Alinari Manager evaluates before creating a business relationship with the photographer.

The first two functionalities (Upload new photos and Search for photos) will be provided by a generic stand-alone application; registration to the Alinari Photo Archives will be through an Alinari web-application. Alinari photographers (right part of Figure *3-4*) will have access to the following additional functionality.

- Browse photo collection. The photos will be organized in albums.
- Browse unsorted photos: Photographers who upload a photo directly to the Alinari server may choose to not to place it in a predefined album. Alinari staff and Alinari Managers will be able to browse these images, and move them to an existing album.
- Create new album: Alinari staff and managers will be able to create new photo album.

During photo upload, the service will ask the user to provide all related metadata (e.g. image ID, date of photography, caption, keywords, etc.) as shown in the figure below.

The same metadata will provide the basis for semantic search. If the search finds photos matching a user's search criteria, it will display a list of the photos with a thumbnail of each. Users who click on a photo will have the following options (depending on their rights).

- View and study metadata (customers)
- Update image by adding new metadata (if necessary) (Alinari staff and managers)
- Categorize photo if unsorted, or place photo in a different album (Alinari staff and managers).

- Delete image (registered users, Alinari Manager).
- Purchase image (customers).

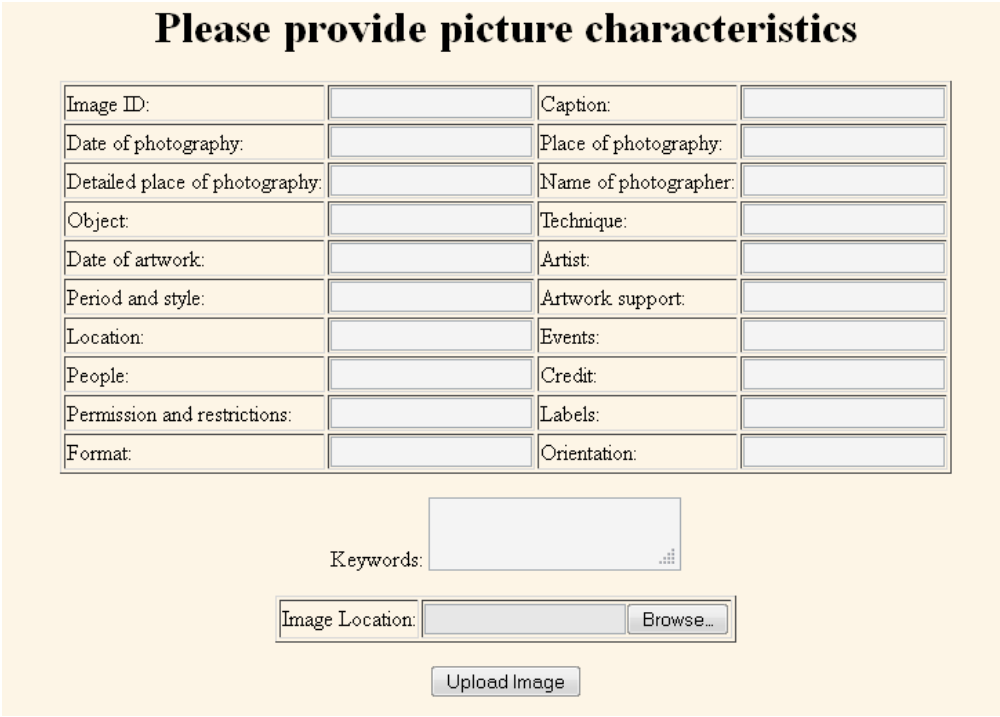

Figure 3-5: Photo upload application

4 Videos in the Cloud and Analyses on the Earth (real world trial 2)

4.1 Brief description of use case

The "Videos on the Cloud and Analyses on the Earth" scenario has been designed to improve the management of audiovisual archives and to exploit the potential of semantic techniques, when the same video resources are exploited several times in different contexts of use (analyses using different domain ontologies, posting on different video channels).

This scenario involves the following users.

1. Video Material Owners (VMO) provide videos to the archive. They encrypt videos, upload them on CoNet and advertise the existence of the videos to specific analysts. They want to be notified when their videos are manipulated.
2. Analysts describe and interpret the content of a video resource, i.e. through the analysis of audiovisual topics (or subjects), the analysis of visual and/or acoustic frames, the linguistic and cultural adaptation of source videos to a chosen (French speaking) target public, etc. They subscribe to videos, download them from CoNet and decrypt them. They upload their analyses on CoNet and notify certain VCOs of the existence of their analyses. They want to be notified when their analyses are manipulated.
3. Video Channel Owners (VCO) are responsible for video channels, and manage content posted on their channel. They subscribe to analyses and post them on their channel. They want to notify certain VCUs of these posts.
4. Video Channel Users (VCU) explore the content of Channels, subscribe to new posts on channels and browse channels.
5. FMSH/ESCoM is a technology provider for Video Channel and Video Analysis Applications.

4.2 Bird's eye view of the deployment framework

a. VMO
    1. Creates and stores a Video VDI
    2. Creates and injects a Publication VDI
    3. Revokes a Video VDI
    4. Revokes a Publication VDI
b. Analyst
    1. Creates and injects a Subscription VDI
    2. Downloads a Video
    3. Creates and stores an Analysis and an Analysis VDI
    4. Creates and injects a Publication VDI
    5. Revokes an Analysis VDI
    6. Revokes a Publication VDI
c. VCO
    1. Creates and stores a Channel and a Channel VDI

2. Creates and injects a Subscription VDI
3. Posts an Analysis and an Analysis VDI
4. Creates and injects a Publication VDI
5. Unposts an Analysis VDI
6. Revokes a Publication VDI

d. VCU
1. Creates and injects a Subscription VDI
2. Browses web channel, reads analyses and plays video

The scenario involves the following types of VDI

1. Video VDI : represents a video resource
2. Analysis VDI: contains analytical appreciations of a video
3. Channel VDI: aggregates a set of Video Analysis VDIs
4. Publication VDI
5. Subscription VDI
6. User VDI

4.3 Technologies

The scenario involves the use of the existing FMSH/ESCoM environment. This includes the following technologies.

1. Ontology Edition Application developed in C# (.Net Framework 4).
   i. *Ontologies are produced using a FMSH/ESCoM proprietary metadata format.*
   ii. *FMSH/ESCoM will develop a technology engine (in Prolog) for converting ontologies into an OWL metadata format, suitable for use by the CONVERGENCE CDS.*
   iii. *2 OWL domain ontologies will be stored in the CDS Server*
   iv. *The application is 100% non- CONVERGENCE.*
2. Video Analysis Application developed in C# (.Net Framework 4).
   i. *Analyses are produced using a FMSH/ESCoM proprietary metadata format.*
   ii. *FMSH/ESCoM will develop a technology engine (in Prolog) for converting ontologies into an OWL metadata format, suitable for use by the CONVERGENCE CDS.*
   iii. *When creating an Analysis VDI, OWL metadata previously produced by FMSH/ESCoM application is injected into the CONVERGENCE "Analyses Manager" application.*
   iv. *This application is 100% non- CONVERGENCE.*
3. FMSH/ESCoM database, using Microsoft SQL Server 2008.
4. FMSH/ESCoM web server, using IIS7.
5. FMSH/ESCoM Windows Media Server (streaming wmv files over mms protocol)
6. FMSH/ESCoM Flash Red5 Server (streaming flv files over rtmp protocol)
7. Video Channel Management Application developed in ASP.Net/C# (.Net Framework 4).

i. *Analyses are posted in a channel by injecting metadata (in FMSH/ESCoM metadata) in the FMSH/ESCoM database*
ii. *This application is 100% non-CONVERGENCE.*

8. Video Channel Application developed in ASP.Net/C# (.Net Framework 4).
   i. *Dynamic web pages display the content of the FMSH/ESCoM database and stream videos*
   ii. *This application is 100% non-CONVERGENCE.*

This scenario involves the development of the CONVERGENCE Tools described later in this document:

1. Tools will be developed in Java. Tools will
   i. *Communicate with CoMid via web services.*
   ii. *Communicate with CoNet via CoMid or directly using CoNet libraries.*

4.4 Functional overview and CoMid interfaces

4.4.1 User registers

This functionality will be provided by the basic "User registration and authentication" tool (see User registration).

4.4.2 User authenticates

This functionality will be provided by the basic "User registration and authentication" tool (see User registration).

4.4.3 VMO creates and uploads a Video VDI

A VMO wants to make a Video available to analysts and FMSH/ESCoM, but does not want people to access the clear text.

1. Authenticate
2. Encrypt Video
3. Create Video VDI with encrypted video resource, static metadata, licenses & ERRs
4. Identify, Sign, Package & Store Video VDI

4.4.3.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Video Encryption		
	Security TE	
Video VDI Creation		
	Metadata TE	

	Create License PE	Create License PE
		REL TE
		Security TE
	ER TE	
	Create Content PE	Create Content PE
		VDI TE
VDI Registration	CONVERGENCE Tool	CONVERGENCE Tool
VDI Signature		
	Security TE	
VDI Packaging		
	Package Content PE	
	MPEG-21 FF TE	
VDI Storage		
	Store Content PE	Store Content PE
		Security TE
		CoNet TE

Table 4-1: Engine sequence for VMO creates and uploads a video VDI application

Figure 4-1 illustrates the application workflow.

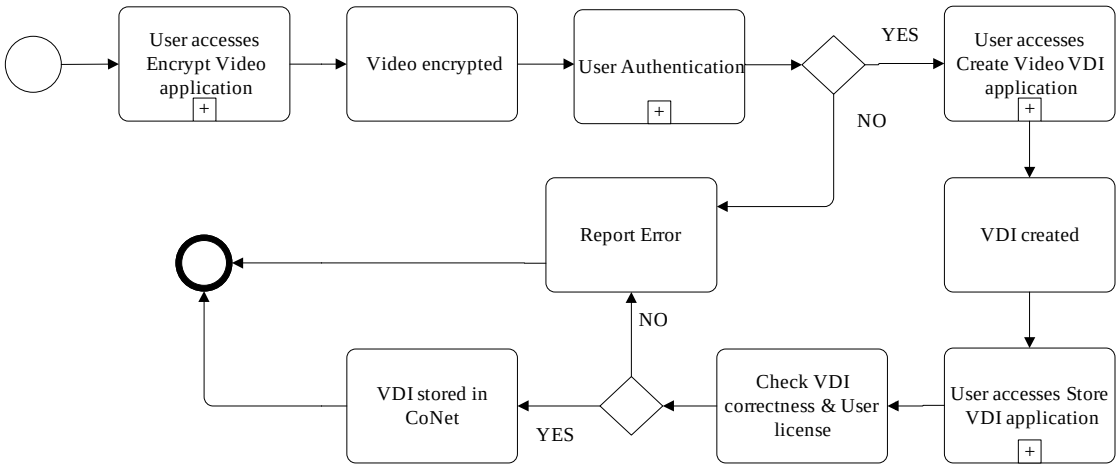

Figure 4-1: Create and upload video VDI application workflow

4.4.3.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Create License	X	X
	Create Content	X	X
	Package Content		X
	Store Content	X	X
Technology Engines	Security	X	X
	Metadata		X
	REL	X	
	ER		X
	VDI	X	
	MPEG-21 FF		X
	CoNet	X	

Table 4-2: Engines for VMO creates and uploads a video VDI application

4.4.4 VMO creates and injects a Publication VDI

A VMO wants to notify analysts that she has published a video.

1. Authenticate
2. Create & Inject Publication VDI with metadata extracted from Video VDI, licenses and ERRs

4.4.4.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Publish VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-3: Engine sequence for creation and injection of a publication VDI by a VMO application

N.B. in this case, and in other cases below, the sequence of engines refers to CONVERGENCE tools, described in Chapter 2; in such cases, we do not repeat the list of engines to reduce the number of tables.

4.4.5 Analyst creates and injects a Subscription VDI

An analyst wants to be notified of new published videos.

1. Authenticate

2. Create & Inject Subscription VDI with conditions on Video VDIs metadata and licenses, licenses and ERRs

4.4.5.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Subscription	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-4: Engine sequence for creation and injection of subscription VDI by an analyst application

4.4.6 VMO revokes a Video VDI

A VMO wants to revoke a Video VDI.

1. Authenticate
2. Browse Video VDIs created by VMO
3. Revoke Video VDI

4.4.6.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Revoke VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-5: Engine sequence for revocation of a video VDI by a VMO application

4.4.7 VMO revokes a Publication VDI

A VMO wants to revoke the publication of a Video VDI.

1. Authenticate
2. Browse Publication VDIs created by VMO
3. Revoke Publication VDI

4.4.7.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Revoke VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-6: Engine sequence for revocation of a publication VDI by a VMO

4.4.8 Analyst and FMSH/ESCoM download Video

After receiving a notification of a new published video, the Analyst downloads it for analysis. FMSH/ESCoM downloads a video to get it ready for streaming. This involves the following steps.

1. Authenticate
2. Browse Event Reports
3. Request Video VDI whose link is in Event Report
4. Download Video VDI
5. Decrypt Video Resource
6. Store Video Resource locally
7. Store Video in FMSH/ESCoM media server (using a ftp client)

4.4.8.1 Sequence of Engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Browse Event Reports	CONVERGENCE Tool	CONVERGENCE Tool
Video VDI downloading		
	Request Content PE	Request Content PE
		REL TE
		Security TE
		CoNet TE
	Process Content TE	
	Store Content PE	
Video decryption		
		Security TE
	Security TE	
	Process Content TE	
	Store Content TE	

Table 4-7: Engine sequence for download video by analyst and FMSH/ESCOM application

Figure 4-2 shows the workflow of the Download video application

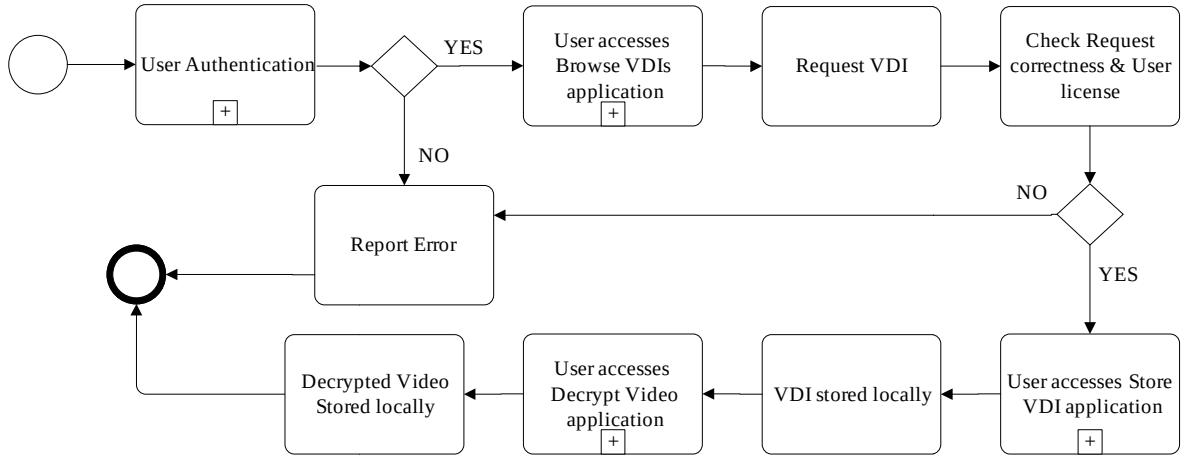

Figure 4-2: Download video application workflow

4.4.8.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Request Content	X	X
	Store Content		X
	Process Content		X
Technology Engines	Security	X	X
	CoNet	X	
	Process Content		X
	REL	X	

Table 4-8: Engines for download video by analyst and FMSH/ESCOM application

4.4.9 Analyst creates an Analysis and an Analysis VDI

Analyst wants to analyze a video, i.e. provide analyses of the discourse, the topics, the visual & acoustic plans, etc. of the whole video, as well as for parts of the video. This analysis is driven by specific contexts of use and by a specific knowledge domain (i.e. domain ontology).

1. Create analysis metadata interacting with CDS (using a FMSH/EScoM non-CONVERGENCE application)
2. Authenticate (as described in application 4.2)
3. Create Analysis VDI with metadata created with FMSH/ESCoM app., licenses, ERRs & referencing a Video VDI
4. Identify, Sign, Package & Store Analysis VDI

4.4.9.1 Sequence of engines

Application elements	Client Engines	Server Engines
Analysis Metadata Creation	Metadata TE	
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Analysis VDI Creation		
	Describe Content PE	Describe Content PE
		CDS TE
		Metadata TE
	Create License PE	Create License PE
		REL TE
		Security TE
	ER TE	
	Create Content PE	Create Content PE
		VDI TE
VDI Registration	CONVERGENCE Tool	CONVERGENCE Tool
VDI Signature		
	Security TE	
VDI Packaging		
	Package Content PE	
	MPEG-21 FF TE	
VDI Storage		
	Store Content PE	Store Content PE
		Security TE
		CoNet TE

Table 4-9: Engine sequence for creation of analysis and analysis VDI by an analyst application

Figure 4-3 illustrates the application workflow.

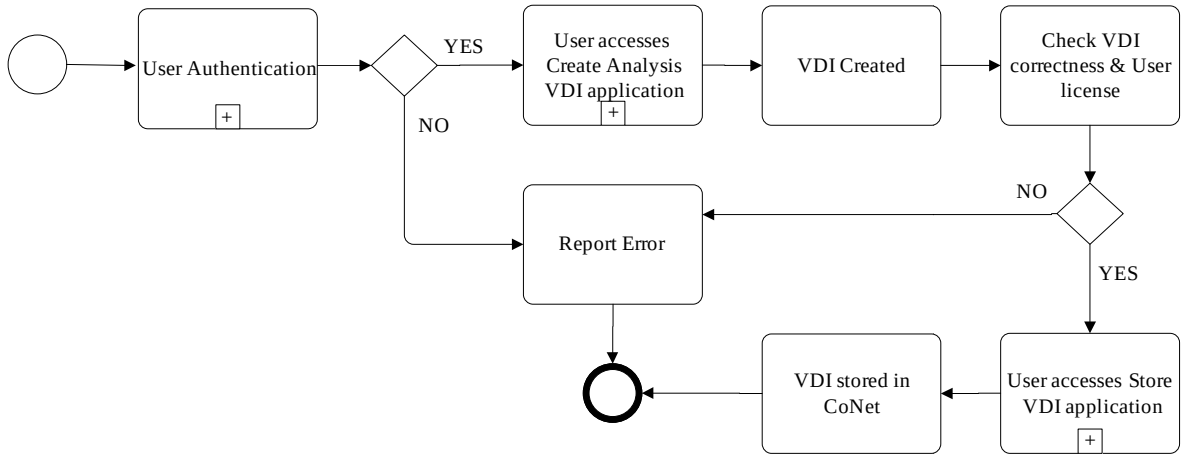

Figure 4-3: Create analysis application workflow

4.4.9.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Create License	X	X
	Create Content	X	X
	Package Content		X
	Store Content	X	X
Technology Engines	Security	X	X
	Metadata	X	X
	REL	X	
	ER		X
	VDI	X	
	MPEG-21 FF		X
	CoNet	X	
	CDS	X	

Table 4-10: Engines for creation of an analysis and analysis VDI application

4.4.10 Analyst creates and injects a Publication VDI

An analyst wants to notify VCOs that her analysis has been published. This involves the following steps.

1. Authenticate
2. Create & Inject Publication VDI with metadata extracted from Analysis VDI, licenses and ERRs

4.4.10.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Publish VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-11: Engine sequence for creation and injection of a publication VDI by an analyst

4.4.11 VCO creates and injects a Subscription VDI

A VCO wants to be notified of published analyses, dealing with specific topics.

1. Authenticate
2. Create & Inject Subscription VDI with conditions on Analysis VDI metadata interacting with CDS and licenses, licenses and ERRs

4.4.11.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Subscription	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-12: Engine sequence for creation and injection of subscription VDI by VCO application

4.4.12 Analyst revokes an Analysis VDI

An analyst wants to revoke an Analysis VDI.

1. Authenticate
2. Browse Analysis VDIs created by Analyst
3. Revoke Analysis VDI

4.4.12.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Revoke VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-13: Engine sequence for revocation of analysis VDI by analyst application

4.4.13 Analyst revokes a Publication VDI

An analyst wants to revoke the publication of an Analysis VDI. This involves the following steps.

1. Authenticate
2. Browse Publication VDIs created by Analyst
3. Revoke Publication VDI

4.4.13.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Revoke VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-14: Engine sequence for revocation of publication VDI by analyst application

4.4.14 VCO creates and stores a Channel and a Channel VDI

A VCO wants to create a web channel providing videos and analyses on a particular topic. This involves the following steps.

1. Authenticate
2. Create Channel VDI with static metadata, licenses and ERRs
3. Identify, Sign & Store Channel VDI
4. Create a channel on FMSH/ESCoM server by deploying & configuring the web application and updating the database (non-CONVERGENCE application)

4.4.14.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Channel VDI Creation		
	Metadata TE	
	Create License PE	Create License PE
		REL TE
		Security TE
	ER TE	
	Create Content PE	Create Content PE

		VDI TE
VDI Registration	CONVERGENCE Tool	CONVERGENCE Tool
VDI Signature		
	Security TE	
VDI Packaging		
	Package Content PE	
	MPEG-21 FF TE	
VDI Storage		
	Store Content PE	Store Content PE
		Security TE
		CoNet TE

Table 4-15: Engine sequence for creation and storage of Channel and Channel VDI by VCO application

4.4.14.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Create License	X	X
	Create Content	X	X
	Package Content		X
	Store Content	X	X
Technology Engines	Security	X	X
	Metadata		X
	REL	X	
	ER		X
	VDI	X	
	MPEG-21 FF		X
	CoNet	X	

Table 4-16: Engines for creation of Channel and Channel VDI by VCO application

4.4.15 VCO posts an Analysis and an Analysis VDI

After receiving a notification of an interesting analysis, a VCO wants to post it on her channel. This involves the following steps.

1. Authenticate
2. Browse Event Reports
3. Download Video Channel VDI created by the VCO
4. Update Channel VDI with new reference to the Analysis VDI to post
5. Send an Update Channel VDI request
6. Store the VDI in a CoNet serving node
7. Revoke old VDI
8. Update FMSH/ESCoM database (non-CONVERGENCE application)

4.4.15.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Browse Event Reports	CONVERGENCE Tool	CONVERGENCE Tool
Channel VDI downloading		
	Request Content PE	Request Content PE
		Security TE
		CoNet TE
	Process Content TE	
	Store Content PE	
Channel VDI Processing		
	Metadata TE	
	Create License PE	Create License PE
		REL TE
		Security TE
	ER TE	
	Process Content PE	Process Content PE
		VDI TE
VDI Registration	CONVERGENCE Tool	CONVERGENCE Tool

VDI Signature		
	Security TE	
VDI Packaging		
	Package Content PE	
	MPEG-21 FF TE	
VDI Storage		
	Store Content PE	Store Content PE
		Security TE
		CoNet TE
Revoke VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-17: Engine sequence for posting of Analysis and Analysis VDI by VCO application

Figure 4-4 illustrates the application workflow.

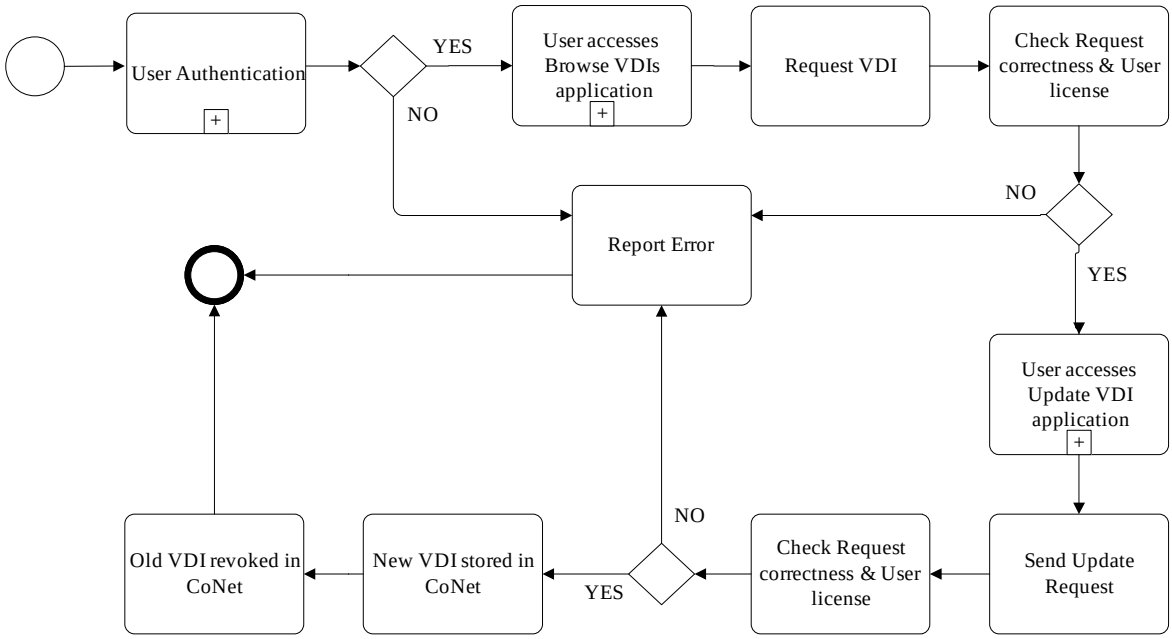

Figure 4-4: Post Analysis VDI application workflow

4.4.15.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Create License	X	X
	Package Content		X
	Store Content	X	X
	Request Content	X	X
	Process Content	X	X
Technology Engines	Security	X	X
	Metadata		X
	REL	X	
	ER		X
	VDI	X	
	MPEG-21 FF		X
	CoNet	X	

Table 4-18: Engines for posting of Analysis and Analysis VDI by VCO application

4.4.16 VCU creates and injects a Subscription VDI

A VCU wants to be notified of analyses on a specific topic

1. Authenticate
2. Create & Inject Subscription VDI with conditions on Analysis VDIs metadata interacting with CDS and Channel VDIs identifiers, licenses and ERRs.

4.4.16.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Subscription	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-19: Engine sequence for creation and injection of subscription VDI by VCU application

4.4.17 VCO creates and injects a Publication VDI

A VCO wants to notify VCUs that an analysis has been posted. This involves the following steps.

1. Authenticate
2. Create & Inject Publication VDI with metadata extracted from Analysis VDI & Channel VDI, licenses and ERRs

4.4.17.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Publish VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-20: Engine sequence for creation and injection of publication VDI application

4.4.18 VCO unposts an Analysis VDI

VCO wants to remove a posted analysis from his channel. The walkthrough is very similar to the walkthrough in section VCO posts an Analysis and an Analysis VDI ("VCO Posts an Analysis VDI"). The difference is that in this case the Analysis VDI is removed from the Channel VDI instead of being added. The list of engines is exactly the same (see Table *4-18*).

4.4.19 VCO revokes a Publication VDI

VCO wants to revoke publication VDI advertising the posting of an analysis in a channel.

1. Authenticate
2. Browse Publication VDIs created by VCO
3. Revoke Publication VDI

4.4.19.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Revoke VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 4-21: Engine sequence for revoking publication VDI by VCO application

4.4.20 VCU browses an analysis on a channel

After receiving a notification of an interesting new analysis, a VCU wants to browse it by reading metadata and watching corresponding video. This involves the following steps.

1. Authenticate
2. Browse Event Reports
3. Browse Publication VDI whose link is inside Event Report
4. Present Metadata in the device
5. Redirect VCU channel, where he can browse the analysis and watch the video (non-CONVERGENCE application)

4.4.20.1 Sequence of engines

Application elements	Client Engines	Server Engines
Browse Event Reports	CONVERGENCE Tool	CONVERGENCE Tool
Analysis VDI browsing		
	Request Content PE	Request Content PE
		REL
		Security TE
		CoNet
	Process Content PE	

Table 4-22: Engine sequence for browsing an analysis on a channel by VCU application

Figure 4-5 illustrates the application workflow.

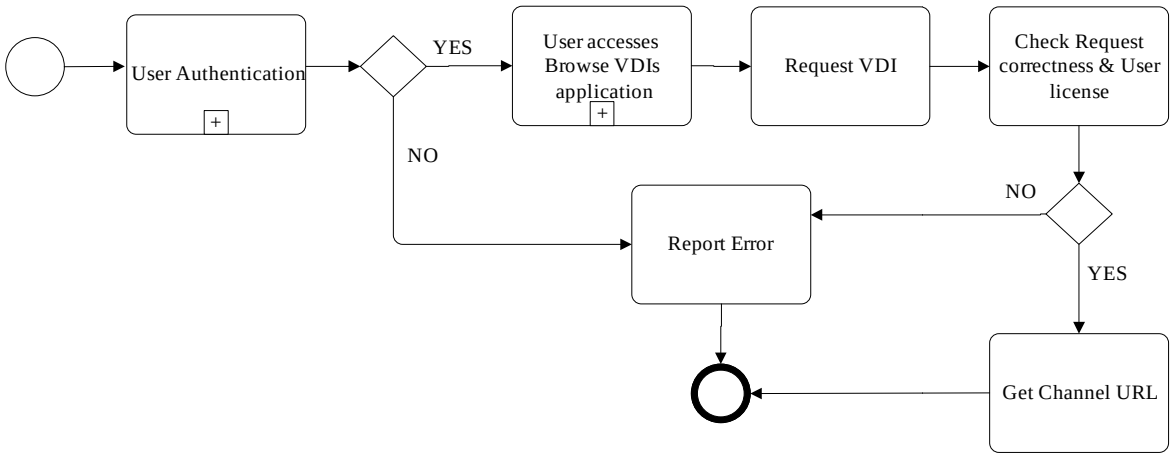

Figure 4-5: Browse Analysis on a Channel application workflow

4.4.20.2 List of engines

Technology type	Technology	S	C
Protocol Engines	Request Content	X	X
	Process Content		X
Technology Engines	Security	X	
	CoNet	X	
	REL	X	

Table 4-23: Engines for browsing an analysis on a channel by VCU application

4.4.21 Summary of Engines

Table *4-24* summarizes the engines of this scenario

Technology type	Technology	S	C	P
Protocol Engines				
	Identify User	X	X	
	Authenticate User	X	X	
	Identify Content	X	X	
	Authenticate Content	X	X	
	Create Content	X	X	
	Store Content	X	X	X
	Inject Content	X	X	
	Describe Content	X	X	
	Create License	X	X	
	Request Content	X	X	X
	Store Event	X		
	Package Content	X	X	
	Revoke Content	X	X	
	Request Event	X	X	
	Process Content	X	X	
Technology Engines				
	Security	X	X	
	VDI	X		X
	CDS	X		X
	Metadata	X	X	X
	REL	X	X	

	ER	X	X	X
	Overlay	X	X	X
	CoNet	X		X
	MP21FF	X	X	
	Match			X
	Event Report	X		
	Present Content		X	
	Process Content		X	

Table 4-24: Summary of engines

4.5 GUI design

Figure 4-6 provides a preliminary design of the GUI for the scenario.

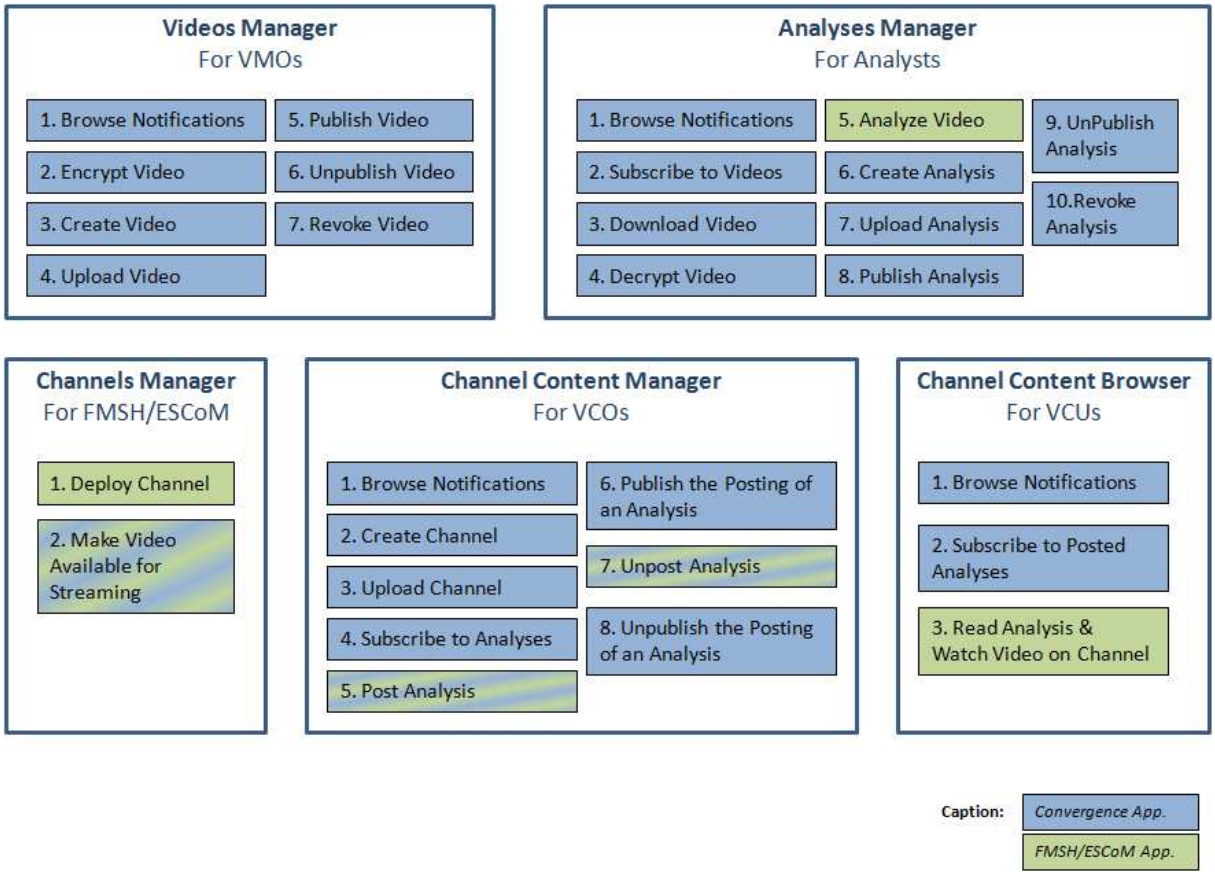

Figure 4-6: Videos in the Earth and Analyses on the Cloud - Application Design

The scenario provides 5 applications, all including a CONVERGENCE User & Device Authentication Tool.

1. **Videos Manager**. This application provides the following functionalities for Video Material Owners.

- Browse Notifications (default interface): browse Event Reports and VDIs referenced in Event Reports.
- Encrypt Video: encrypt a Video Resource.

- Create Video: create a Video VDI including a previously encrypted Video Resource, static metadata (see Figure *4-7*), licenses and ERRs.

- Upload Video: identify, sign, package and store a previously created Video VDI on CoNet.

- Publish Video: create and inject into the cloud a Publication VDI with metadata, licenses and ERRs extracted from a previously uploaded Video VDI.

- Unpublish Video: unpublish a Publication VDI previously injected into the cloud.

- Revoke Video: revoke a previously uploaded Video VDI from CoNet.

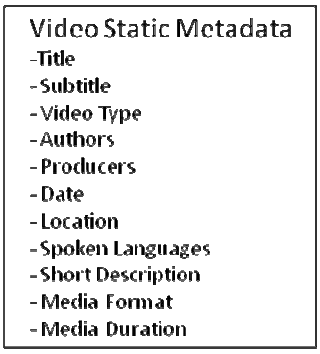

Figure 4-7: Static metadata of a Video VDI

2. **Analyses Manager**. This application provides the following functionalities for Analysts.

- Browse Notifications (default interface): browse Event Reports and VDIs referenced in the Event Reports.

- Subscribe to Videos: create and inject a Subscription VDI defining conditions on metadata, licenses and ERRs.

- Download Video: download a Video VDI and the included Video Resource from CoNet to local device.

- Decrypt Video decrypt a previously downloaded Video Resource.

- Analyze Video: create metadata related to a previously downloaded and decrypted Video (see Figure 4-8 and Figure 4-9).

- Create Analysis: create an Analysis VDI including previously created metadata, licenses and ERRs.

- Upload Analysis: identify, sign, package and store a previously created Analysis VDI on CoNet.

- Publish Analysis: create and inject a Publication VDI with metadata, licenses and ERRs extracted from a previously uploaded Analysis VDI.

- Unpublish Analysis: unpublish a previously injected Publication VDI.

- Revoke Analysis: revoke a previously uploaded Analysis VDI from CoNet.

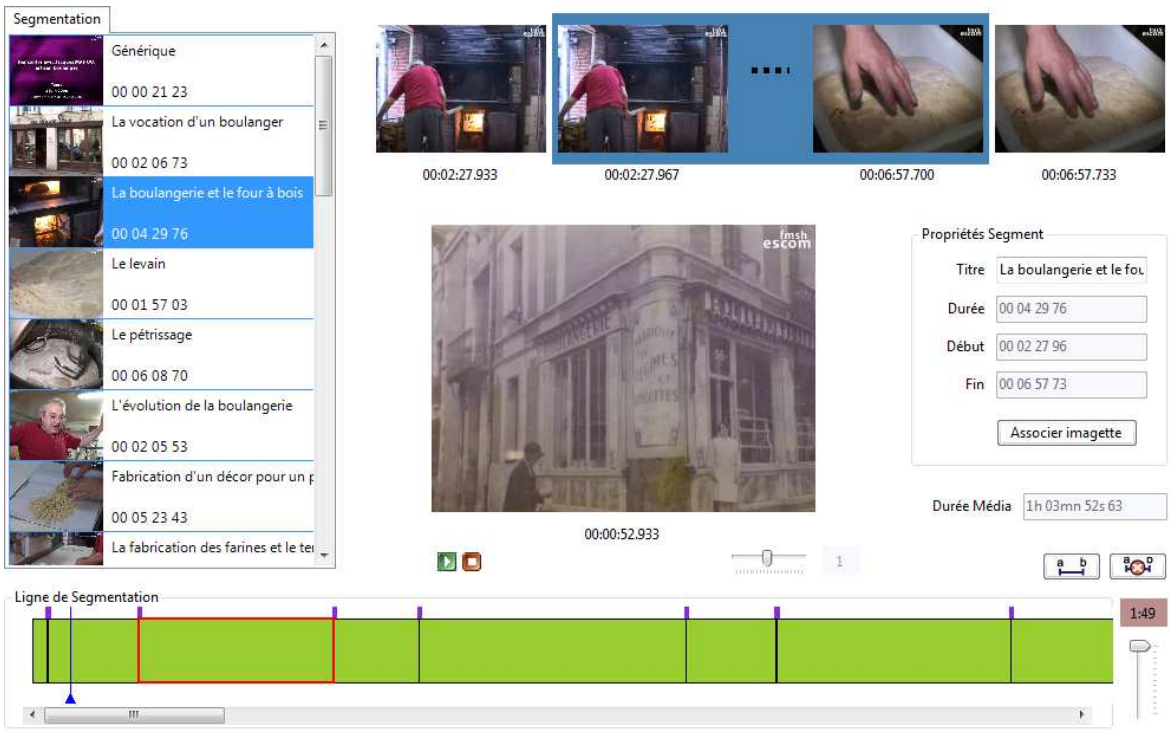

Figure 4-8: Video Segmentation in FMSH/ESoM Analysis Application

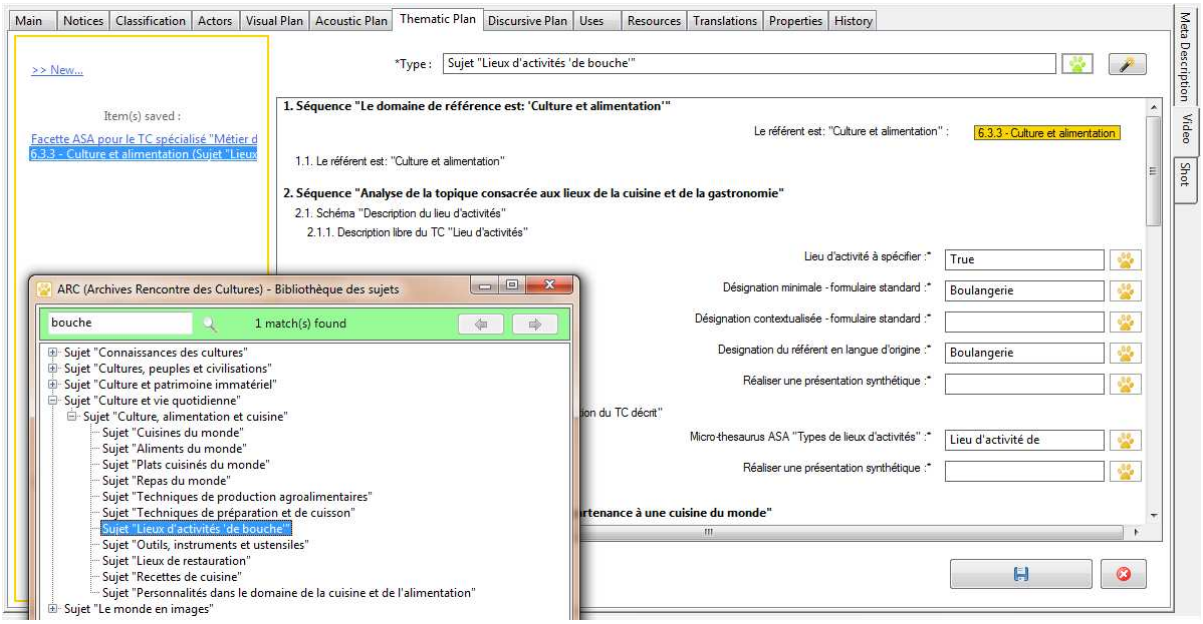

Figure 4-9: Forms in FMSH/ESCoM Analysis Application

3. **Channels Manager**. This application provides the following functionalities for FMSH/ESCoM:

- Deploy Channel: deploy a web channel using FMSH/ESCoM technology.
- Make Video Available for Streaming: download and decrypt a Video Resource (as in the Analyses Manager and upload it to the FMSH/ESCoM streaming server via ftp.

4. Channels Content Manager. This application provides the following functionalities for Video Channel Owners:

- Browse Notifications (default interface): browse Event Reports and VDIs referenced in Event Reports.
- Create Channel: create a Channel VDI including static metadata (see Figure *4-10*), licenses and ERRs.
- Upload Channel: identify, sign and store a previously created Channel VDI on CoNet.
- Subscribe to Analyses: create and inject a Subscription VDI with conditions for metadata license of Analysis VDIs, licenses and ERRs.
- Post Analysis: update a Channel VDI by adding a reference to an Analysis VDI, and by updating the FMSH/ESCoM database to make the Analysis available on the web channel.
- Publish the Posting of an Analysis: create a Publication VDI with metadata extracted from a previously uploaded Analysis VDI and Channel VDI, licenses and ERRs, and inject it into the cloud
- Unpost Analysis: update a Channel VDI by removing a reference to an Analysis VDI: update the FMSH/ESCoM database, removing the analysis from the web channel.
- Unpublish the Posting of an Analysis: unpublish a Publication VDI previously injected into the cloud.

Figure 4-10: Static metadata of a Channel VDI

5. Channel Content Browser. This application provides the following functionalities for Video Channel Users:

- Browse Notifications (default analysis): browse Event Reports and VDIs referenced in Event Reports.
- Subscribe to Posted Analyses: create and inject a Subscription VDI with conditions for metadata in Analysis VDIs, identifier of Channel VDIs, licenses and ERRs.
- Read Analysis & Watch Video on Channel: browse an Analysis VDI posted on a Channel VDI; redirect the VCU to the corresponding section in the web channel (see Figure *4-11*).

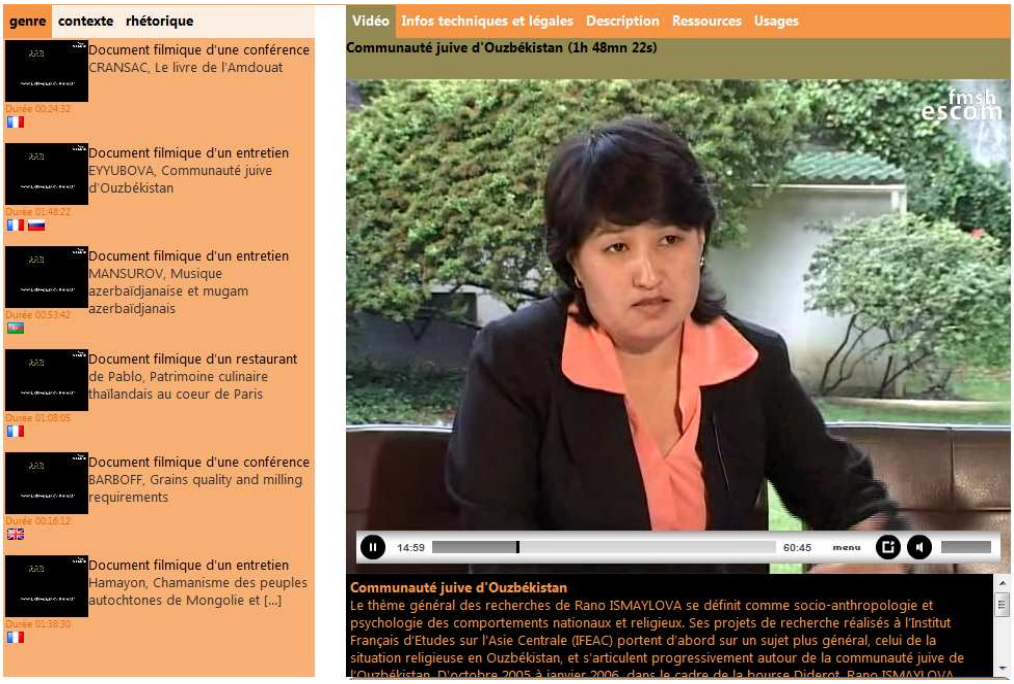

Figure 4-11: Browsing an Analysis in FMSH/ESCoM Channel Application

5 Augmented Lecture Podcast (ALP) (real world trial 3)

5.1 Brief description of use case

This scenario is based on a web-based lecture podcast application that enables students to revise lectures by watching video podcasts with synchronized slides. CONVERGENCE provides a common basis for collaboration and information exchange. Since different instances of the same file are related to each other by their VDI representations, it is possible to update one file and keep all related files in sync and ensuring that students always have access to the latest materials. When students have downloaded files onto their local device, they are notified of changes as soon as they connect to the CONVERGENCE Network.

Figure 5.1 depicts a general overview of the main activities in the scenario.

a) Lecturers:
    a. Create, store and publish Lecture Podcast VDI sequences, which are updated from time to time, when video or slides are modified
    b. Retrieve statistical information about the use of the Lecture Podcast VDIs (statistical information is collected by the service provider).
b) Students:
    a. Search, subscribe, download and watch Lecture Podcasts, and are informed if a new lecture podcast or a new version of a podcast is made available by the service.
    b. Create and publish annotations to specific portions of the podcast using an augmented lecture podcast application, which streams the podcast episode.
    c. Receive notifications of published Annotations and publish Annotations to Annotations.
    d. Delete annotations.

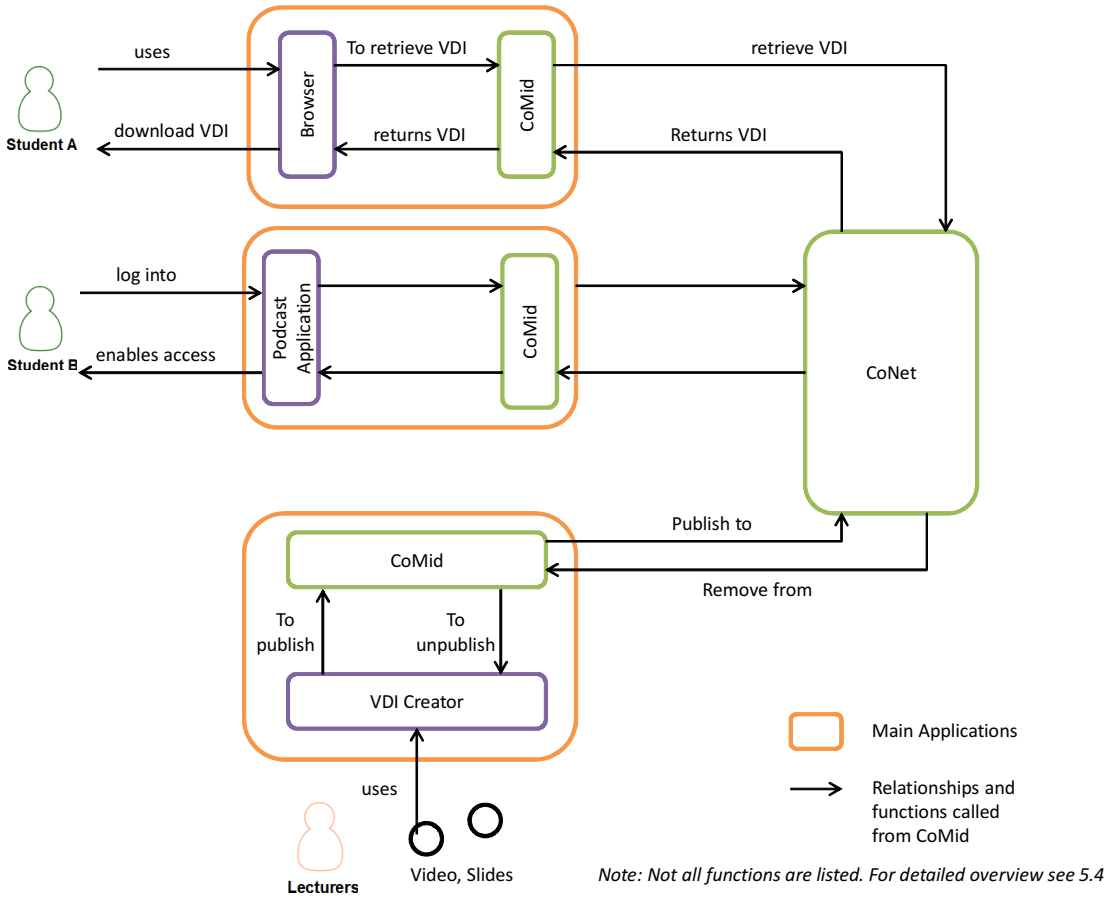

Figure 5-1: Augmented Lecture Podcast System Architecture

5.2 Bird's eye view of the deployment framework

The application space has been provisionally partitioned into seven Tools and Applications.

1. User registration
2. Lecturers create and update video fragments and corresponding slides
3. Lecturers create and publish sequences of Lecture Podcast VDIs
4. Students search for, download and subscribe to Lecture Podcasts which interest them*
5. Students receive notification of lecture updates
6. Students watch Lecture Podcasts, and write Annotations for a selected audience
7. Students read Annotations and make Annotations to Annotations by other students**
8. Qualified users access event statistics.

*The "Search and Subscribe" application should look like something one can find in today's Internet, where a user finds some resources with a search engine and decides to download and store it.

**The "Watch and make Annotations" in turn, has nothing to do with the file downloaded by the student. The "Watch and make Annotation" applications does not display the file downloaded by the student, though it streams the same content.

5.3 Technologies

5.3.1 Current Prototype (based on existing solution in [1])

The current prototype is being developed using the Adobe Flex Framework[2], MXML and ActionScript. The application runs in a standard environment using Adobe Flash Player[3] (at least version 9). The application can also be run as a web application on the user desktop using Adobe Air[4] as a runtime environment.

Figure 5-2 shows the architecture of the system.

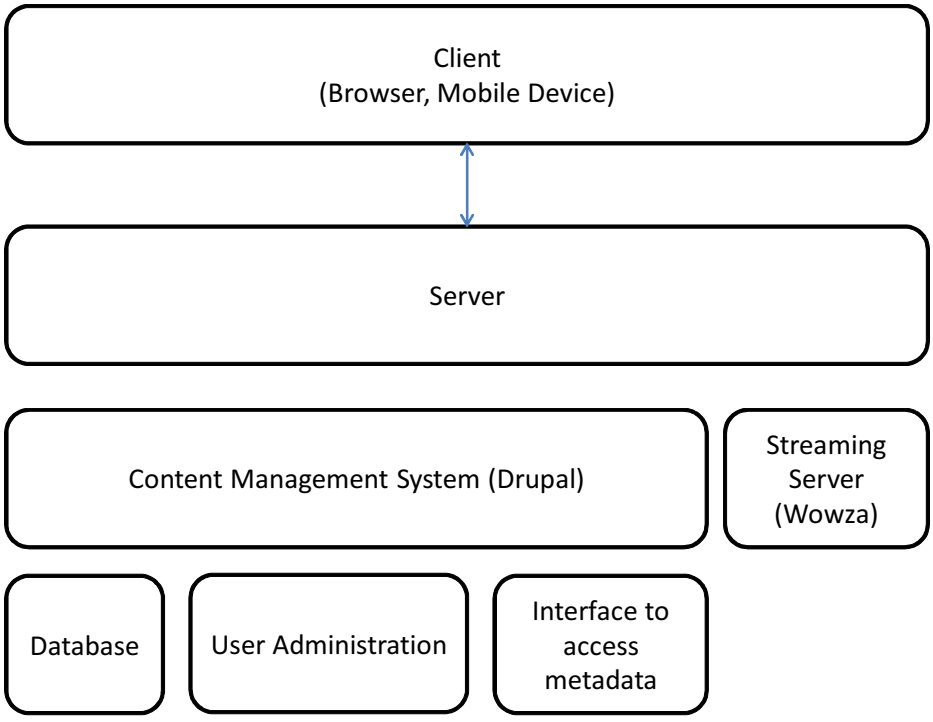

Figure 5-2: Current architecture used for lecture podcast applications

As the prototype is developed with Adobe Flex, it will be necessary to find a way of communicating with the middleware, which exposes various Web-Services for each application. In Flex we can call Web-Services using a Web Service Object that communicates via SOAP. This makes it possible to call JAVA-methods.

[1] http://videoonline.edu.lmu.de/new_features

[2] http://www.adobe.com/products/flex/

[3] http://www.adobe.com/de/products/flashplayer/

[4] http://get.adobe.com/de/air/?promoid=DAFZD

5.3.2 Other technologies

The development team is currently considering the use of the following additional technologies.

- HTML/CSS. The podcast application will be implemented in HTML. CSS will be used to specify the layout of podcast components inside the HTML-document.
- AJAX. AJAX makes it possible to retrieve information and display it on a webpage without reloading the page. In the podcast application the use of Ajax makes it possible to display Annotations without reloading the embedded video they refer to.
- Scripting Languages. The application will use JavaScript on the client side and JSP on the server side.
- QuickTime Player[5]. The lecture podcast will be played/streamed using the QuickTime player. The application will use JavaScript to control player operations (e.g. stop, play, jump…).
- PHP/MySQL. Data from the application (e.g. event report statistics) will be stored in a MySQL database, using PHP.
- Acrobat Browser Plugins. Slides accompanying the podcast will be displayed using Acrobat Browser plugins. The development team is studying solutions for Browsers that do not have a suitable plugin.

5.4 Functional overview and CoMid interfaces

5.4.1 User Registration

Users need to register with the service before they can access its functionalities. The relevant procedures are described in User registration.

5.4.2 Creation and publishing of Lecture Podcast VDI sequence

Lecturers can create, publish and update VDIs for slides, videos and podcast episodes. This involves the following steps:

1. User Authentication
2. VDI Creation for podcast component
3. VDI Creation for augmented lecture podcast
4. VDI Publication

[5] http://www.apple.com/de/quicktime/download/

5.4.2.1 Sequence of engines

Application elements	Client Side	Server Side
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Creation of ALP VDI / Segments		
	Metadata TE	
Content Registration	CONVERGENCE Tool	CONVERGENCE Tool
Synchronization of video and slides		
	Metadata TE	
Description of resources		
	Describe Content PE	Describe Content PE
		Metadata TE
		CDS
Creation of License		
	Create License PE	Create License PE
		REL TE
		Security TE
Insertion of Event Report Request		
	Event Report TE	
VDI identification	CONVERGENCE Tool	CONVERGENCE Tool
VDI signature		
	Security TE	
VDI packaging		
	MP21FF TE	
VDI storage		

	Store Content PE	Store Content PE
		CoNet
Publish VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 5-1: Engine sequence for the creation and publishing of Lecture Podcast Application

Figure *5-3Figure 4-3* illustrates the workflow.

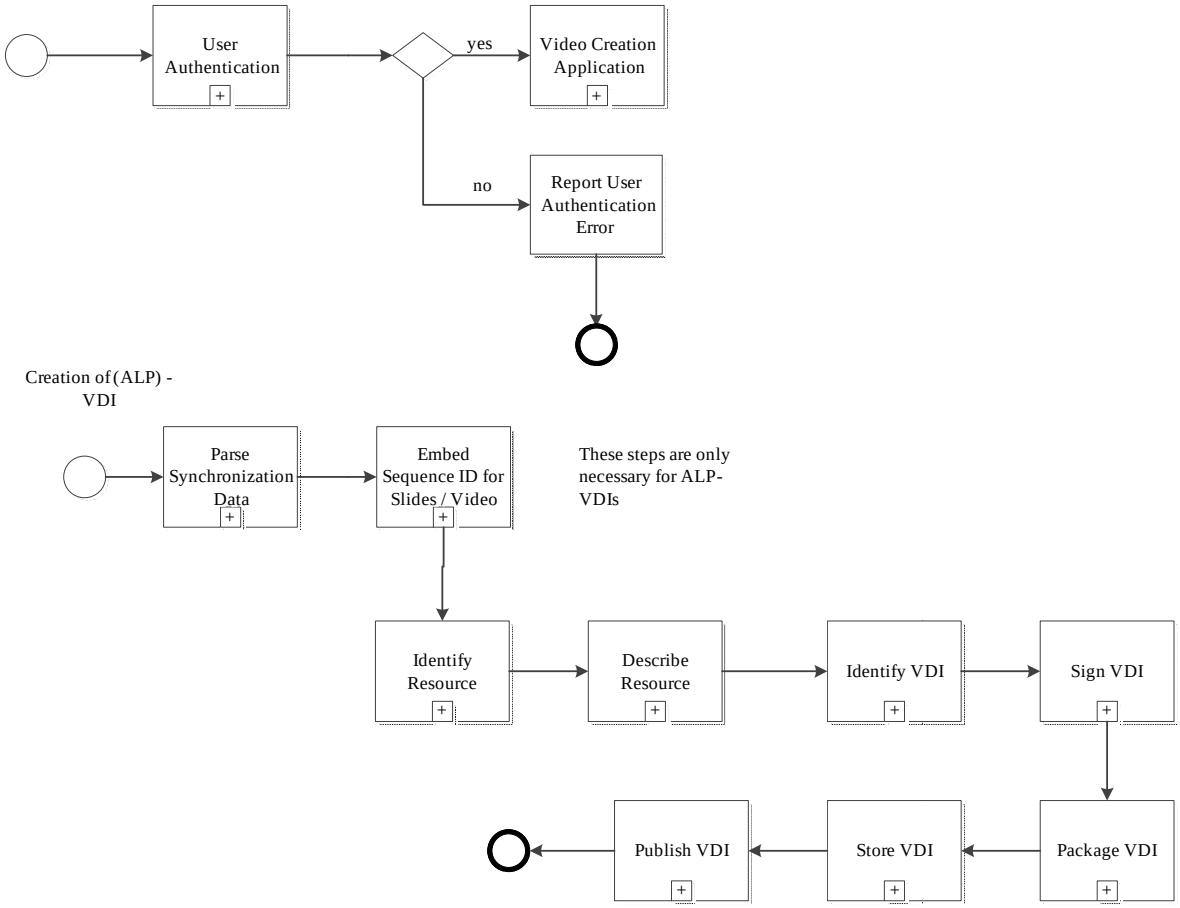

Figure 5-3: Creating and publishing of lecture podcast work flow

5.4.3 Search, subscribe to and download Lecture Podcast episode

Students can search, subscribe and download lecture podcast episodes and lecture podcast components. This involves the following steps:

1. Searching for lecture podcast
2. Receiving a series of candidate VDIs
3. Authenticating VDI of interest
4. Subscribing to VDI of interest

Subscription procedures are performed as in Subscription to VDI.

5.4.3.1 Sequence of engines

Application elements	Client Side	Server Side
Searching for Lecture Podcast		
	Search Content PE	Search Content PE
		CDS
		Metadata TE
		Match TE
Authenticating VDI of interest	CONVERGENCE tool	CONVERGENCE tool

Table 5-2: Engine sequence for search, subscribe to and download Lecture Podcast episode

5.4.4 Watch and annotate Lecture Podcast Episode

Students can use the augmented lecture podcast application to watch podcast episodes with synchronized slides. While watching, they have the possibility to make annotations to the podcast. This involves the following steps:

1. Watching the Lecture Podcast
2. Annotating the Lecture Podcast
3. Publishing Annotations

5.4.4.1 Sequence of engines

Application elements	Client Side	Server Side
Watching Lecture Podcast		
	Request Content PE	Request Content PE
	Metadata TE	
	Process License PE	
	REL TE	
VDI / Resource identification	CONVERGENCE Tool	CONVERGENCE Tool
Create Annotation	CONVERGENCE Tool	CONVERGENCE Tool
Inclusion of Event Report		

	Event Report TE	
Requesting to receive notifications (Subscribe to Annotation)	CONVERGENCE Tool	CONVERGENCE Tool

Table 5-3: Engine sequence for watch and annotate Lecture Podcast episode

Figure 5-4 illustrates the workflow.

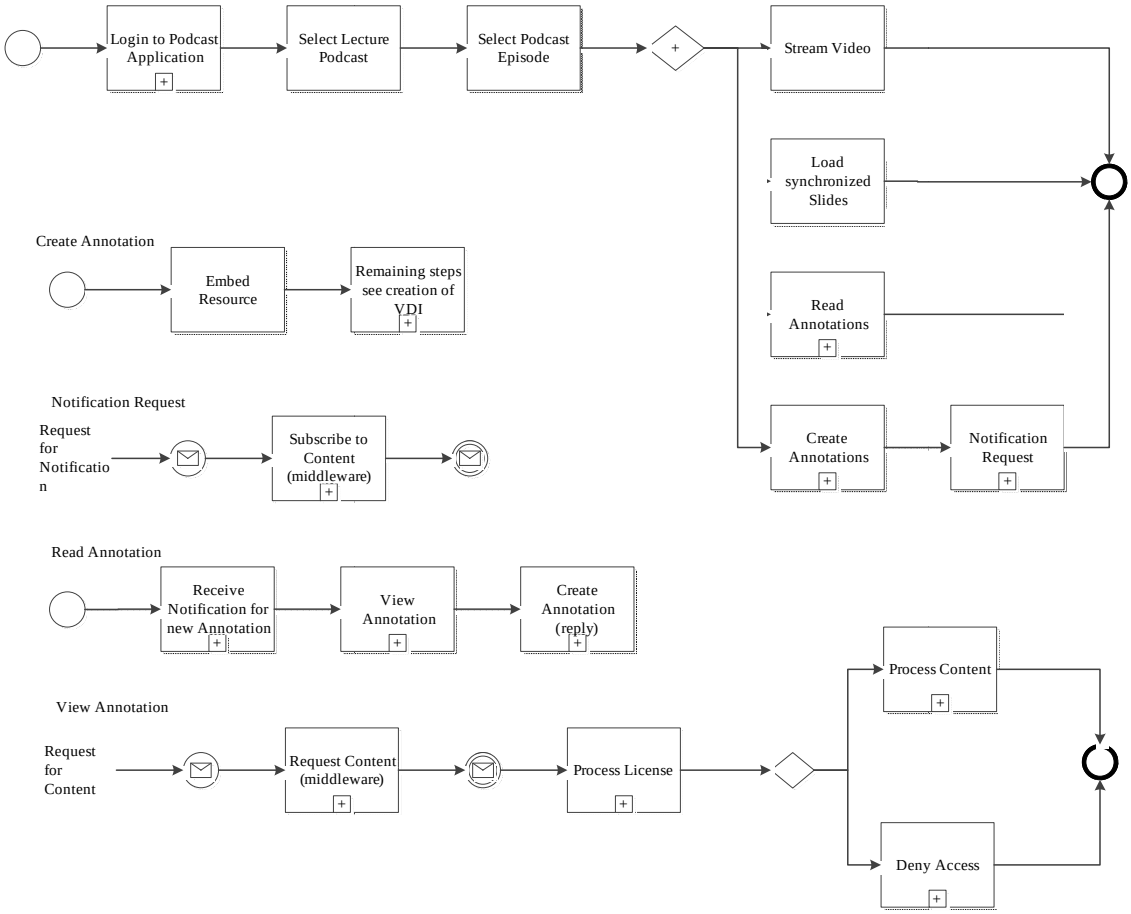

Figure 5-4: Watch Podcast/Make and create annotations work flow

5.4.5 Read Annotations to Lecture Podcast

While watching the lecture podcast, students can read annotations made by themselves or by other students. They also can reply to annotations. This involves the following steps.

1. Watching Lecture Podcast
2. Receiving Notifications about new Annotations
3. Reading Annotations
4. Reply to Annotations

5.4.5.1 Sequence of engines

Application elements	Client Side	Server Side
Watching Lecture Podcast		
	Request Content PE	Request Content PE
		CoNet
	Metadata TE	
	Process License PE	Process License PE
		REL TE
	Media Framework TE	
Receiving Notification of Annotation		
		Match TE
	Store Event PE	Store Event PE
	Event Report TE	Event Report TE
	Request Content PE	Request Content PE
		CoNet
	Metadata TE	
	Process License PE	Process License PE
		REL TE
		Security TE

Table 5-4: Engine sequence for Read Annotations to Lecture Podact

5.4.6 Revoke Lecture Podcast or Annotation

Students can delete their Annotations, which are then revoked from the CONVERGENCE Network. This functionality is described in Revoke VDI from CoNet.

5.4.7 Event Report Statistics

Lecturers can collect statistical information on the use of specific lecture podcasts. This functionality is described in Browse Event Report.

5.4.8 Summary of Engines

Technology type	Technology	S	C	P
Protocol Engines				
	Authenticate Content	X	X	
	Authenticate User	X	X	
	Create Content	X	X	

	Create License	X	X	
	Describe Content	X	X	
	Identify Content	X	X	
	Identify User	X	X	
	Inject Content	X	X	
	Package Content	X	X	
	Process Content		X	
	Process License	X	X	
	Request Content	X	X	X
	Request Event	X	X	
	Revoke Content	X	X	
	Search Content	X	X	
	Store Content	X	X	
	Store Event	X	X	X
Technology Engines				
	CDS	X		X
	CoNet	X		X
	Content Identification	X		
	Event Report	X	X	X
	Match	X	X	X
	Media Framework	X	X	
	Metadata	X		X
	MP21FF	X	X	
	Overlay	X		X
	REL	X	X	
	Security	X	X	
	VDI	X		X

Table 5-5: Summary of engines

5.5 GUI design

The figures below show preliminary GUI designs for the application. The figures refer to the interface that appears when students have already logged in and selected a podcast episode. The GUI consists of three main components: video, slides and annotations.

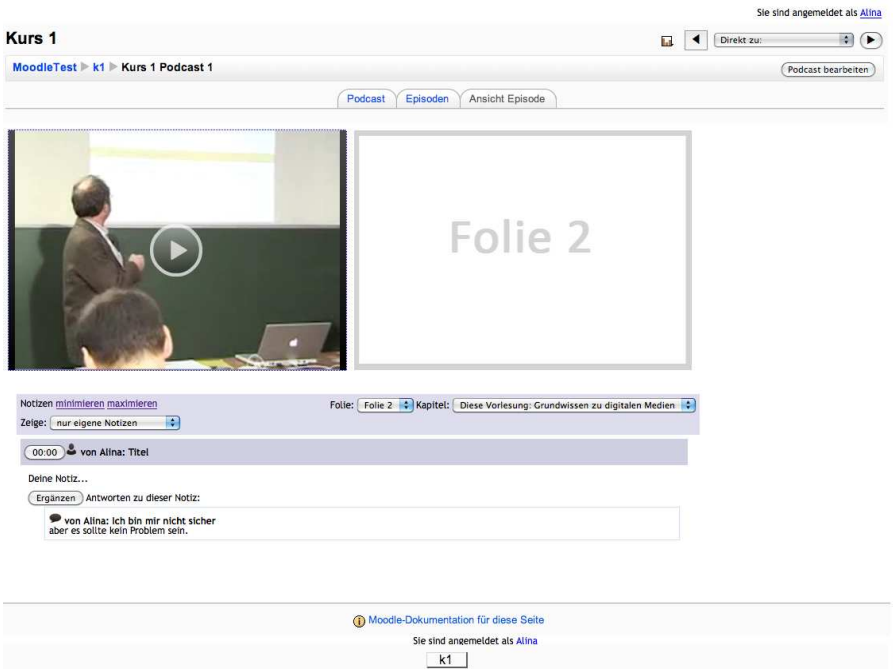

Figure 5-5: Screenshot of preliminary GUI design showing annotations to a podcast

- Watch videos. The video component (on the upper left) shows the podcast episode. Students can hover on it to see the video controls.
- Select slides: While playing the video, the slide component (upper right) displays the corresponding slides, which are synchronized with the video. When a student selects slide number from the dropdown menu or a specific chapter, the system displays the corresponding slides and the video jumps to the corresponding timestamp.

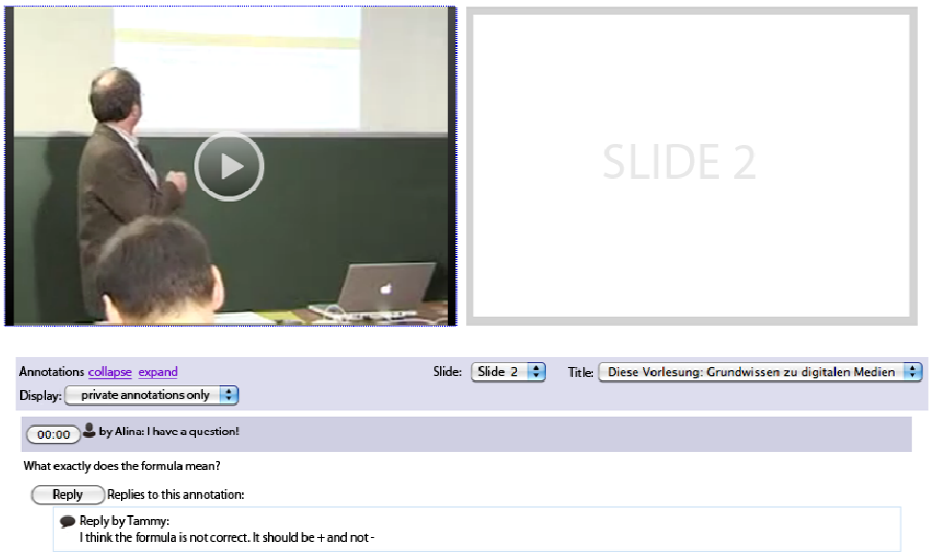

Figure 5-6: Screenshot of preliminary GUI design showing the creation of an annotation

- Make annotations. A student can make an annotation to a slide, by clicking on it. A popup window appears. The student enters a title for their annotations, fills in the text of the annotation and selects whether it is personal or public. Once the annotation has been saved, the popup disappears and the annotation appears in the annotation component (bottom). This component displays the title and the type of annotations. Selecting the title expands the annotation.

- Reply to annotations: Once an annotation is expanded, a reply-button allows students to add further comments. Replies appear inside the main annotation.

- Jump to slide: Each annotation provides a button, which enables students to jump to the slide the annotation refers to. The video will jump to the time indicated in the timestamp for the annotation.

- Annotations features: A dropdown menu allows the student to:
    i. View only personal annotations
    ii. View only public annotations
    iii. View both types of annotations

6 Smart Retailing (real world trial 4)

6.1 Brief description of use case

This scenario describes a smart retailing supply chain for electronic products. In the scenario, CONVERGENCE provides users with a wide range of services and operations. The users and beneficiaries are manufacturers, retailers, consumers and other actors in the supply chain. Manufacturers make their products available on CONVERGENCE; retailers advertise the products they sell; customers search for and compare products. Additional services improve the customer shopping experience allowing them to subscribe to information on specific products, receive notifications about the products and sales events and immediately look up information on a product by scanning its barcode.

This scenario involves the following activities:

1. Manufacturers
    a. Create, store, publish and certify product type VDIs containing reliable information such as the name of the product and its technical characteristics
    b. Profit from new flexible mechanisms for selling products to authorized retailers in a smart retailing supply chain
2. Retailers
    a. Subscribe to product type VDIs published by manufacturers
    b. Create and publish retailers' product type VDIs, augmenting manufacturer product type VDIs with information about retailer promotions, special offers, sales, etc.
    c. Enjoy improved control over the products they sell and better knowledge of their customers
3. Consumers
    a. Search and subscribe to products VDIs
    b. Receive notifications about updates on products VDIs they have subscribed to
    c. Consult certified, reliable information about products
    d. Use services to manage technical information about their products, warranties, information on repair services etc.

6.2 Bird's eye view of the deployment framework

The scenario involves the following functionalities:

1. Manufacturer
    i. Create/Store Manufacturer Product Type VDI (MPT-VDI)
    ii. Publish MPT-VDI
    iii. Un-publish MPT-VDI from the CONVERGENCE Cloud
2. Retailer
    iv. Process MPT-VDI to Create/Store Retailer Product Type (RPT-VDI)
    v. Publish RPT-VDI

vi. Process MPT-VDI to Create/Store Retailer Product Instance (RPI-VDI)
vii. Publish RPI-VDI
viii. Create/Store RPI-D-VDI
ix. Publish RPI-D-VDI
x. Un-publish RPT-VDI from the CONVERGENCE Cloud
xi. Browse VDI

3. Consumer
   i. Subscribe to VDI
   ii. Browse VDI
4. User (Manufacturer/Retailer/Consumer)
   xii. Browse Event Reports and generate statistics

The development team is currently discussing the possibility of using annotations to enhance social relationships between manufacturers, retailers, and customers, etc. This would require an additional "Create Annotation" VDI.

6.3 Technologies

Oracle Retail Environment

- Linux x64 environments (servers)
- Programming languages:
  - Java + PL/SQL
- Oracle Retail Products:
  - Retail Merchandising System (RMS)
    - *The database will be customized so as to associate VDIs with items (products) and store the VDIs*
  - Oracle Retail Point of Sale (POS): Server + Client
  - Oracle Retail Back Office and Central Office
- Oracle Forms Interfaces in Java:
  - *The interface will be customized to include CONVERGENCE functionalities*

6.4 Functional overview and CoMid interfaces

6.4.1 Creation and storage of VDI

A CONVERGENCE application will allow manufacturers to create a product VDI and store it in the CoNet. This involves the following steps:

a. User Authentication
b. VDI Creation for product
c. VDI Publication

6.4.1.1 Sequence of engines

Application elements	Client Engines	Server Engines
Creation of VDI		
	Create Content PE	Create Content PE

Content Registration		
	CONVERGENCE Tool	CONVERGENCE Tool
Description of resources		
	Describe Content PE	Describe Content PE
		Metadata TE
		CDS TE
Creation of License		
	Create License PE	Create License PE
		REL TE
		Security TE
Insertion of Event Report Request		
		Event Report TE
VDI identification		
	CONVERGENCE Tool	CONVERGENCE Tool
VDI signature		
		Security TE
VDI packaging		
	MPEG-21 FF TE	
VDI storage		
	Store Content PE	

Table 6-1: Engine sequence for creation and storage of VDI

Figure 6-1 illustrates the workflow.

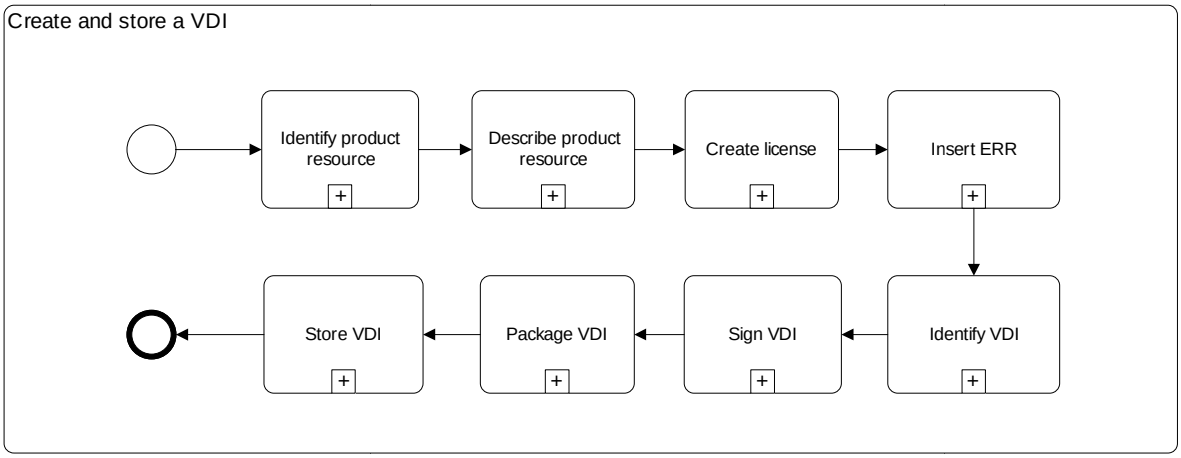

Figure 6-1: Create and store VDI workflow

6.4.2 Creation and injection of Publication VDI

Manufacturers and retailers want to inform other users about new products (technical features, support, etc.). To do this, they use a CONVERGENCE application to create and inject a Publication VDI into the cloud. An appropriate peer in the cloud performs matches between the Publication and outstanding subscriptions. This is done with the help of the Publish VDI tool described in Publish VDI. The procedure involves the following steps.

1. Authenticate
2. Create & Inject Publication VDI with product's metadata, licenses and ERRs

6.4.2.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Publish VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 6-2: Engine sequence for creation and injection of publication VDI workflow

6.4.3 Unpublish VDI

The user uses a CONVERGENCE application to send an unpublish message request to the appropriate peers in the cloud; for this purpose he uses the Unpublish VDI tool described in Unpublish VDI. The tool revokes the Publication VDI. The procedure involves the following steps:

1. Authenticate
2. Browse Publication VDIs created by the User
3. Revokes Publication VDI

6.4.3.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Unpublish VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 6-3: Engine sequence for unpublishing VDI application

6.4.4 Creation and injection of Subscription VDI

A user wants to be notified of new product types and uses a CONVERGENCE application to create and inject a Subscription VDI. An appropriate peer in the cloud performs a match between the Publication and outstanding Subscriptions. This is done with the help of the Subscription to tool described in Subscription to VDI.

The procedure involves the following steps:

1. Authenticate
2. Create & Inject Subscription VDI with conditions on Product VDIs metadata

6.4.4.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Subscription	CONVERGENCE Tool	CONVERGENCE Tool

Table 6-4: Engine sequence for creation and injection of subscription VDI application

6.4.5 Browse VDI

The user uses a CONVERGENCE application to browse VDIs and request data for a specific VDI and its components. This involves the following steps:

1. Authenticate
2. Search for products meeting domain specific criteria (product characteristics, retailer details, etc.)
3. Verify access rights to the VDI and authorize access
4. Visualize product details

6.4.5.1 Sequence of engines

Application elements	Client Engines	Server Engines
User Authentication	CONVERGENCE Tool	CONVERGENCE Tool
Searching for Product		
	Search Content PE	Search Content PE
		CDS
		Metadata TE
		Match TE
Authenticating VDI of interest	CONVERGENCE tool	CONVERGENCE tool
Browse VDI	CONVERGENCE Tool	CONVERGENCE Tool

Table 6-5: Engine sequence for browse VDI application

Figure *6-2* illustrates the application workflow.

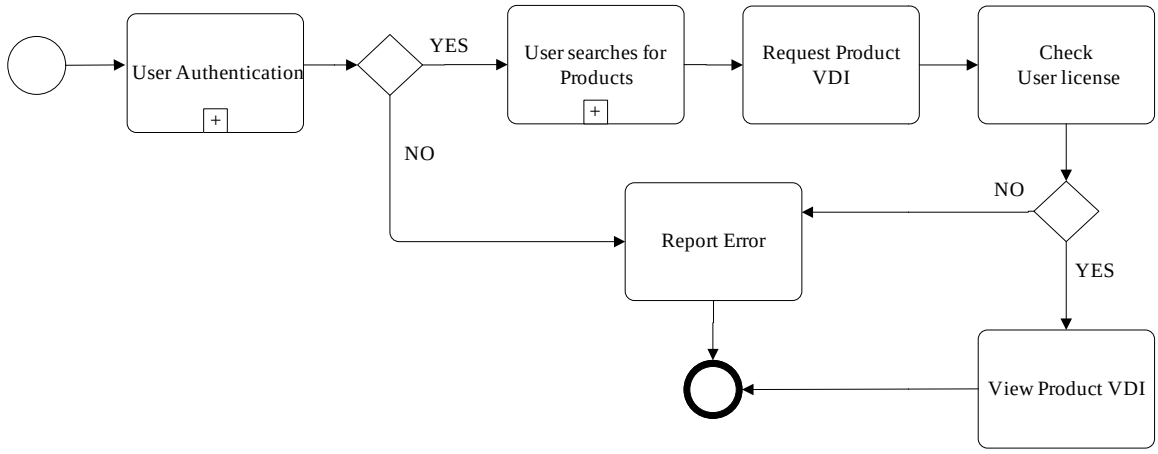

Figure 6-2: Browse VDI application workflow

6.4.6 Browse Event Reports and generate statistics

The user uses the CONVERGENCE application to browse the ERs she had received and to request data for the generation of statistics. This involves the following steps:

1. User receives ER (notifications)
2. Browse ER (Browse Event Report)
3. Generate statistical information– the statistics are generated on the server using the ERs and the aggregated data are delivered for presentation to the client.

6.4.6.1 Sequence of engines

Application elements	Client Engines	Server Engines
ER browsing		
	Request Event PE	Request Event PE
		Event Reporting TE
Stats Board	X	X

Table 6-6: Engine sequence for browse event reports and generate statistics

Figure 6-3 illustrates the workflow.

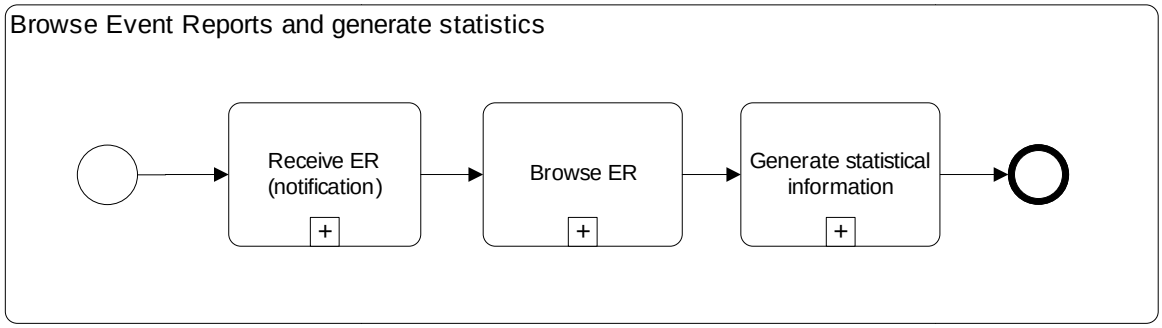

Figure 6-3: Browse event reports and generate statistics workflow

6.4.7 Summary of Engines

Technology type	Technology	S	C
Protocol Engines			
	Identify User	X	X
	Authenticate User	X	X
	Identify Content	X	X
	Authenticate Content	X	X
	Create Content	X	X
	Store Content		X
	Inject Content		X
	Describe Content	X	X
	Request Content	X	X
	Create License	X	X
	Process License	X	X
	Store Event	X	X
	Request Event	X	X
Technology Engines			
	CDS	X	
	Metadata	X	X
	REL	X	X
	ER	X	X
	Security	X	X
	Overlay		X
	Match	X	

	Event Report		X
	MPEG-21 FF	X	

Table 6-7: Summary of engines

6.5 GUI design

Different users (manufacturers, retailers and consumers) will require different GUI designs. Manufacturers will require an application GUI that facilitates the creation and publication of VDIs for the products they release to the market. Consumers will need an appropriate application GUI to help them to browse and subscribe to VDIs. However, the most critical design issues are those posed by the retailer POS (Point-of-Sale) application, an application that will be used for every sale. The figures below illustrate preliminary GUI designs for such an application. Screenshots are based on Oracle Retail Point-of-Sale software and were extracted from the user guide document available on the Oracle website (http://download.oracle.com/docs/cd/E12521_01/point_of_service/docset.html).

When selling items to customers, the sales agent begins with the "Sell Item Screen" where all the transactions begin and end. The screen always shows all items involved in the current transactions.

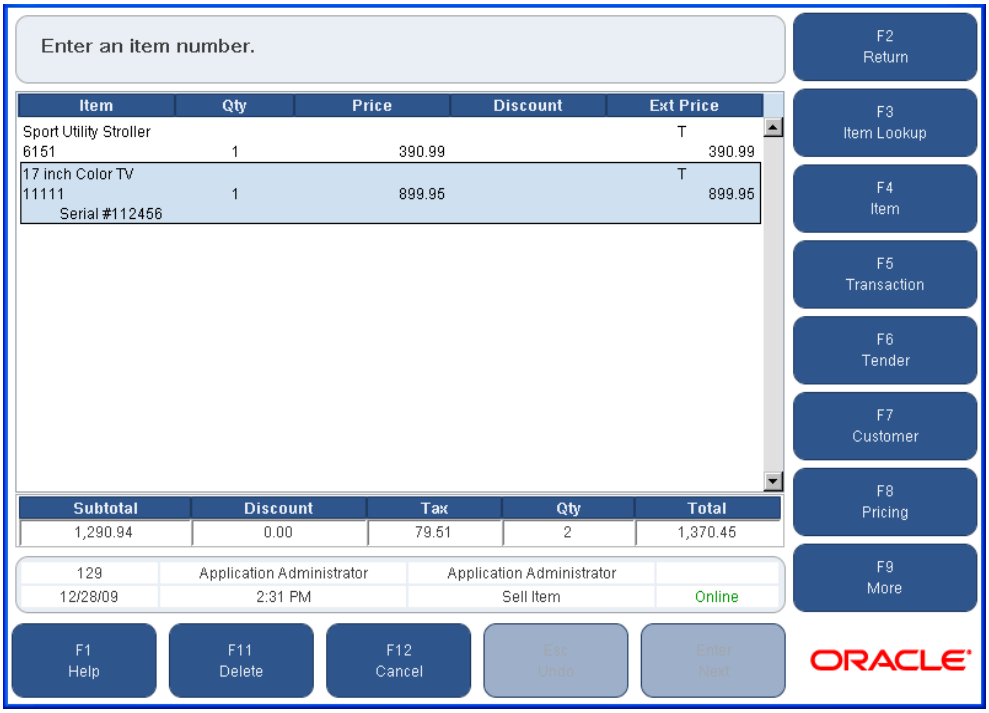

Figure 6-4: Screenshot of shell item screen

At any time during the transaction, the customer, with the help of the sales agent, can identify herself at the POS, so that she can be associated with the product VDIs for the products she is purchasing. To do this, the sales agent selects the "F7 Customer" button in the "Sell Item Screen" (shown above) and uses the "Customer Options Screen" to find, add, edit or delete customer information. The screen provides an option to read the customer CoCard (CONVERGENCE Smart Card) which contains her user VDI. The card identifies the customer to the system.

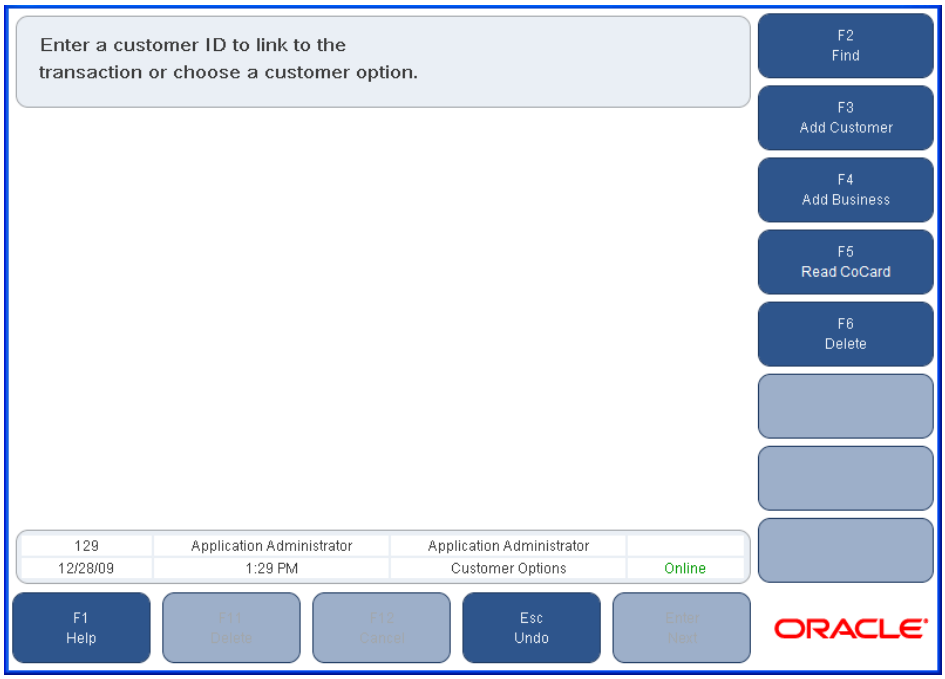

Figure 6-5: Screenshot of customer options screen

After the consumer has identified herself, a "Customer Information Screen" shows available customer information. From this moment on, the system knows the identity of the customer involved in the sale. Product VDIs created at the POS will include this information.

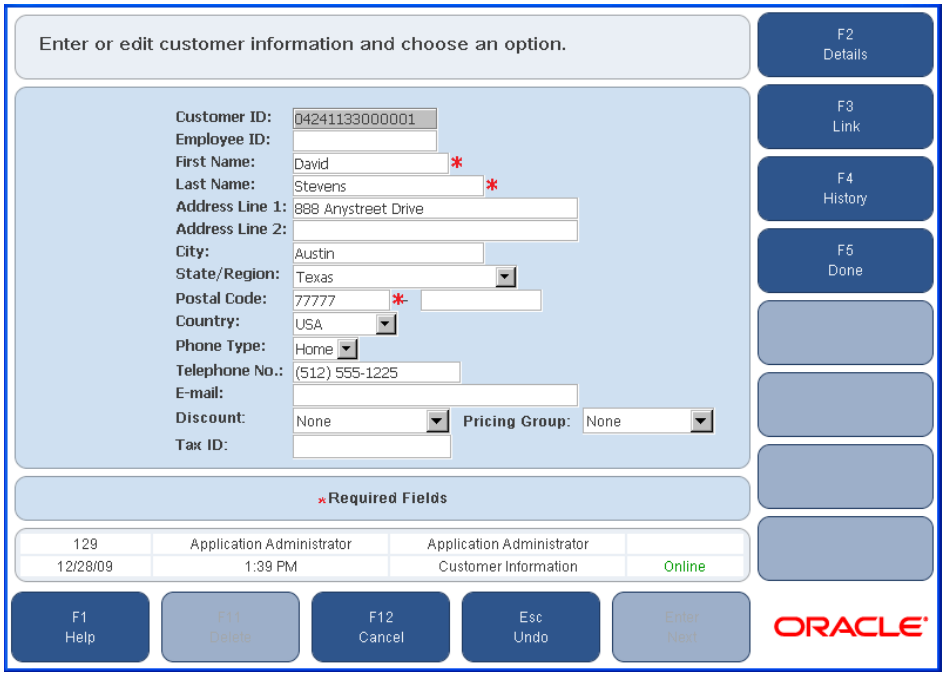

Figure 6-6: Screenshot of customer information system

The sales agent introduces the serial numbers of products sold at the POS. To do this, she selects the desired item in the "Sell Item Screen" and presses the "F4 Item" button. The system displays the "Item Options Screen" below.

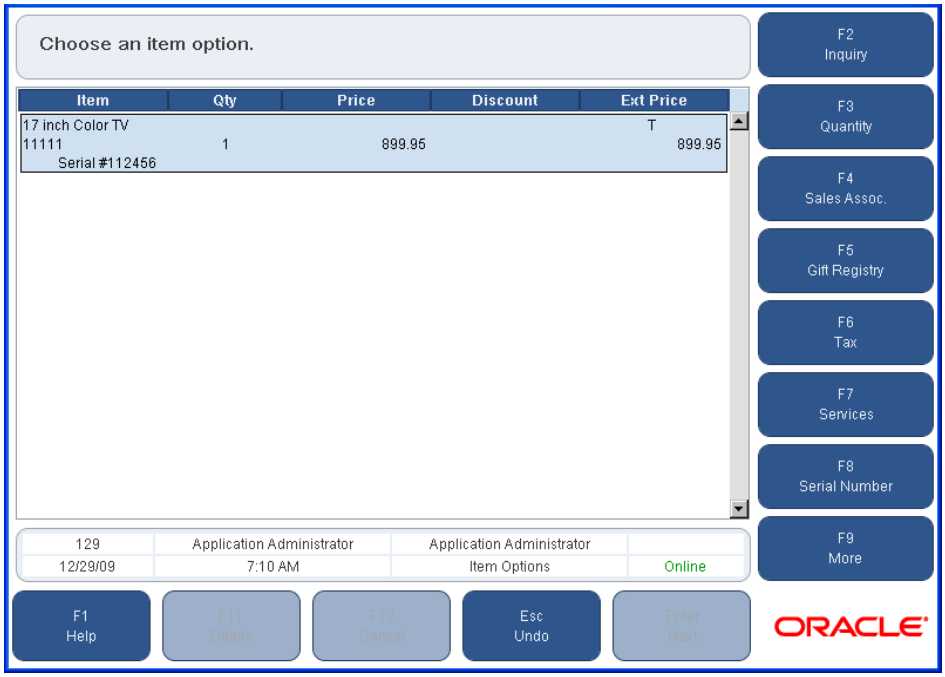

Figure6-7: Screenshot of items options screen

The sales agent presses the "F8 Serial Number" button to access the "Enter Serial Number Screen", inserts the serial number and clicks on the "Enter Next" button. The system associates the serial number with the product. The serial number will be included in the product VDI created on completion of the sale.

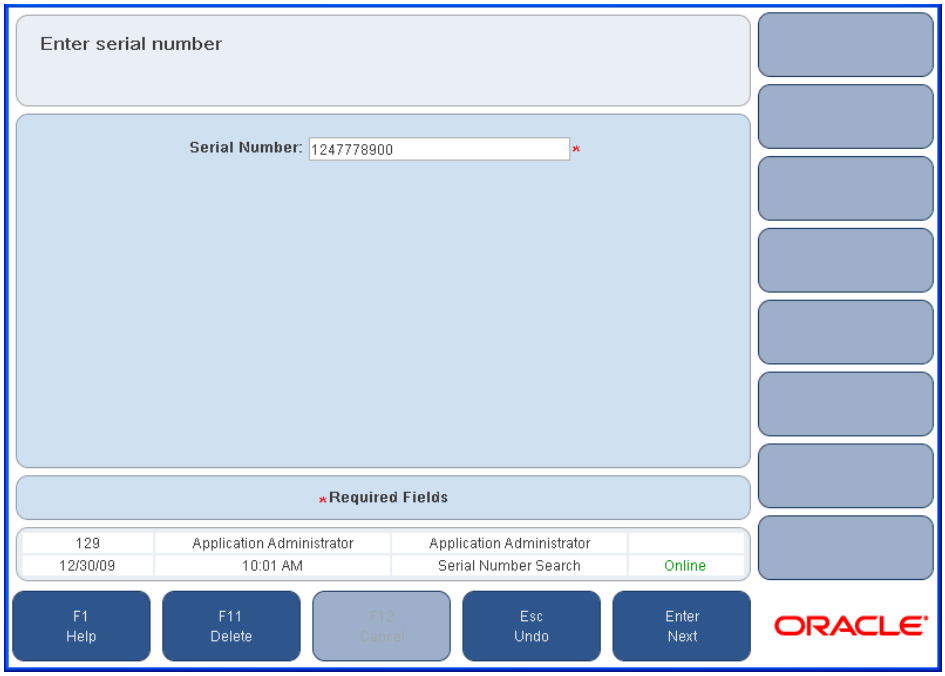

Figure 6-8: Screenshot of enter serial number screen

On completion of the sale, the system creates all necessary product VDIs including relevant consumer and serial number information and generates an electronic receipt including the IDs

of each VDI. From this moment on, the customer will be able to access the information through the CONVERGENCE cloud.

7 Integration of middleware engines with applications

CoMid will expose a Java-based API. Applications will be able to access the middleware in two ways: either by making a web service call to an elementary service or an aggregation of elementary services (which then automatically call the corresponding protocol engines to handle the request) or by making a direct call to technology engines or an orchestration of technology engines.

Web service calls are designed to provide interoperability between different platforms and programming languages and work in many different environments. By contrast, a call to a technology engine/orchestrator is less simple. If the developer is working with a Java application, she can use the API provided by CoMid. However, browser based applications can be developed by using various frameworks (JavaScript, Flash, ASP/JSP/JSF, etc.). This means that there is no a general valid way for such an application to call a Java method. To circumvent this issue, the development team has developed an HTTP Proxy that receives HTTP requests to technology engines and interprets them to local Java calls. Interpretation is performed by a Java application supplied as a plugin for the browser application. The proxy triggers the application on each request to an engine.

Figure *7-1* depicts the abstract architecture for this scheme. From the figure, we see that Java-based and Web Browser based applications can make web service calls to the protocol engines without help from the middleware. As far as concerns Technology Engines, standalone Java applications make direct calls to the middleware; HTTP requests from Browser applications go through the proxy, which calls a local Java application to call the middleware.

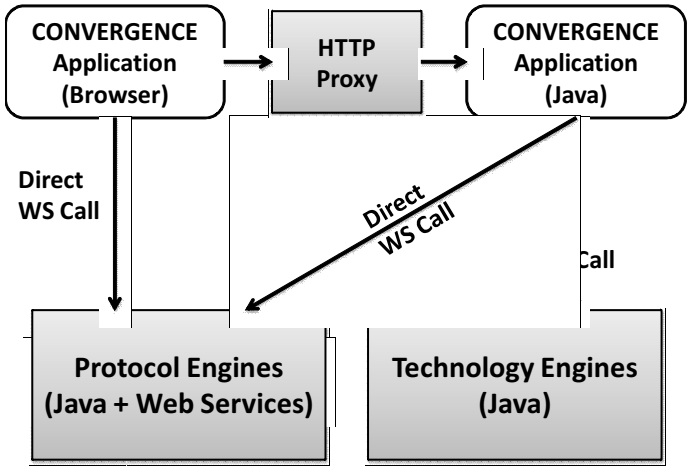

Figure 7-1: Applications access to CoMid

Figure *7-2* shows a more elaborate schema of the architecture for a Browser-based application. When the browser requests the application from the remote server (application server) it also fetches a plugin, which is installed in the local proxy. The plugin contains the part of the middleware that needs to be executed in the client (i.e. the technology engines and any orchestrations) and the mappings between the HTTP requests and the methods that make

the calls to the technology engines. These mappings are stored in a configuration file maintained by the HTTP proxy.

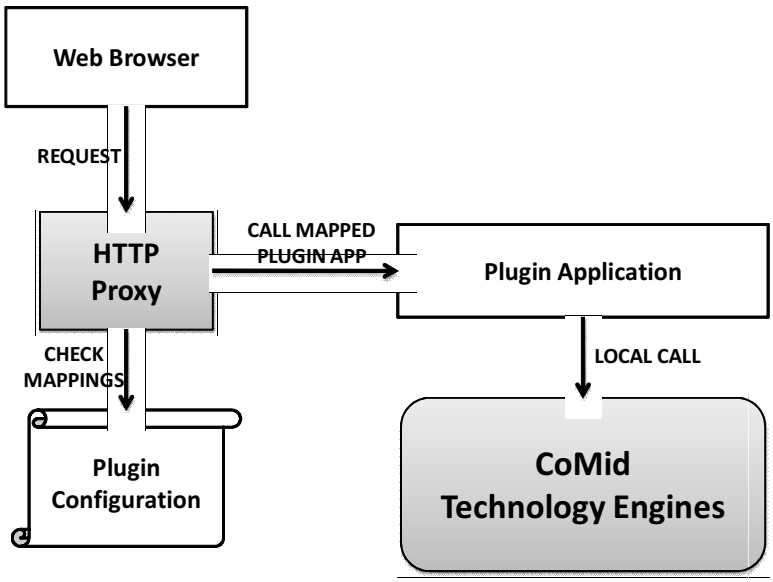

Figure 7-2: Applications access to CoMid

Consider the following example: the Browser makes a call to an application server and fetches the browser application along with the configuration file. The configuration file is stored locally in the proxy. This allows the proxy to check for mappings when it receives an application request. When the user submits an HTTP form, the HTTP POST request goes through the proxy, which checks whether the plugin configuration file for the application contains the URI for the action. If it does, the request parameters are passed to the method defined in the configuration file; which makes the local call to the middleware engines. If not, the proxy forwards the request to the application server.

Since each application may require a different instance of the middleware, the proxy holds a separate CoMid part for each application, which is fetched on the first local call; when the proxy tries to call a method defined by the plugin for the first time, the plugin returns the URI that the proxy should use to fetch the necessary part of the middleware.

Putting everything together, CONVERGENCE will support the following forms of communication between applications and middleware:

1. Application → Protocol Engine:
    A. Web Service call

    (The application can be developed in any framework that supports web services)
2. Standalone Java Application → Technology Engine:
    A. Local java call

    (The application has to be developed in Java; the client carries the part of the middleware that provides the technology engines used by the application)

3. Web Application → Technology Engine:
   (In this case, we consider a three-component communication schema that includes three components: a user terminal (equipped with a Web Browser), a Java-based application server and the middleware server)

   A. The terminal is *not* a CONVERGENCE peer (i.e. it does not include a part of the middleware) and the role of the CONVERGENCE client is performed by the application server. In this case, it is the applications server that parses user requests from the Web Browser and makes the local calls to the technology engines (just like any other Java application).

   B. The terminal is a CONVERGENCE peer (i.e. it does or it *can* include a part of the middleware) and plays the role of the CONVERGENCE client (this does not exclude the possibility that an application server could also play the same role). In this case, the HTTP proxy is responsible for receiving requests from the Web Browser and, using the web application plugin (a java application), to make the local calls to the technology engines.

For the first phase of the trials, CONVERGENCE will support communication methods 1, 2 and 3A, as depicted in Figure *7-3*, which is extracted from a preliminary version of deliverable D.8.1.

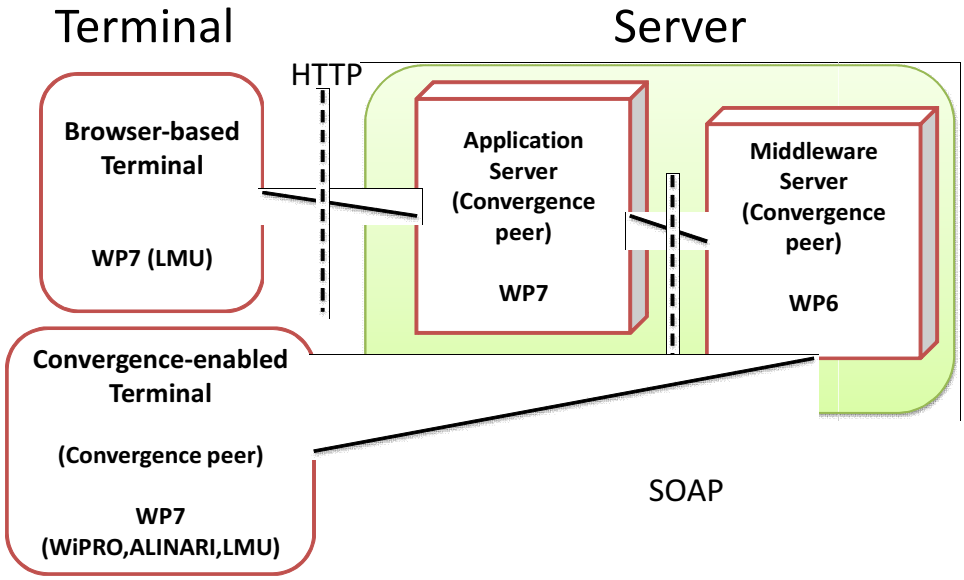

Figure 7-3: Phase 1 Trials Terminal and Server Relationship

8 Summary – Conclusions

8.1 Overview

In this deliverable, we describe the implementation framework for the CONVERGENCE project, starting from the four user scenarios that will be implemented in the trials. We begin by describing a set of tools required by all the scenarios and go on to describe additional tools meeting specific needs. In each case, we provide a walkthrough of the scenario, identify a list of engines supporting its needs and provide BPMN diagrams. In what follows we summarize our main results.

8.2 CONVERGENCE Tools

We have identified the following tools, required by all scenarios:

- User Registration
- Content Registration
- Publish VDI
- Unpublish VDI
- Revoke VDI
- Subscription to VDI
- Browse Event Report
- Create Annotation VDI

We observe that this is not necessarily an exhaustive list and that additional tools may be identified during the implementation phase.

8.3 Analysis of use cases: engines, applications and tools

We have analyzed use cases from an end-user and an applications viewpoint and used the analysis to define an implementation framework that is compliant with existing standards and can possibly extend them. As part of this work, we have identified the applications needed to support the use cases and the engines needed to support each application over the CONVERGENCE middleware and network. This approach allows maximum re-use of CONVERGENCE tools and minimizes the design effort required.

8.4 Moving towards integration

To speed up integration, we have selected the technologies that will be used in the implementation phase and created a unified approach to handling Java stand-alone and web-based applications. In this approach, local calls to technology engines by Browser-based applications will be handled by an HTTP Proxy, which will be responsible for receiving HTTP requests to engines and interpreting them to local java calls. Interpretation will be performed by a java application, which takes the form of a plug-in for the browser application, which the proxy triggers every time it receives a request to an engine.

9 References

[1] CONVERGENCE, Project Deliverable D.2.1

[2] CONVERGENCE Project Deliverable D.4.1

[3] CONVERGENCE Project Deliverable D.5.1

[4] CONVERGENCE Project Deliverable D.3.2

[5] Business Model Process and Notation (BPMN), Version 2.0, OMG Document Number: dtc/2010-06-05, Standard document URL: http://www.omg.org/spec/BPMN/2.0.