Guideline

Project Acronym:	**PEPPOL**
Grant Agreement number:	**224974**
Project Title:	**Pan-European Public Procurement Online**

Virtual Company Dossier Implementation Architecture Plans for PEPPOL Beneficiaries to enable Cross-Border Virtual Company Dossier

Revision: 1.01

Authors:
Ansgar Mondorf (UKL)
Maria Wimmer (UKL)

Project co-funded by the European Commission within the ICT Policy Support Programme

Dissemination Level		
P	**Public**	**X**
C	**Confidential, only for members of the consortium and the Commission Services**	

Revision History

Revision	Date	Author	Organisation	Description
1.0	20100430	Maria Wimmer	UKL	First version (pending EC approval)
1.01	20101001	Klaus Vilstrup Pedersen	DIFI	EC Approved

Statement of originality

This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Statement of copyright

Contributors

Organisations

ADETEF (Assistance au developpement des echanges en techologies economiques et financieres), France

CNIPA (Centro Nazionale per l'Informatica nella Pubblica Amministrazione[1]) , Italy, www.cnipa.gov.it

DIFI (Direktoratet for forvaltning og IKT[2]), Norway, www.difi.no

EKEVYL (The Research Center for Biomaterials), Greece, www.ekevyl.gr

IC (InfoCamere), Italy, www.infocamere.it

PEPPOL.AT, Austria

UKL (University of Koblenz-Landau), Germany

Persons

Ansgar Mondorf, UKL (editor)
Bruno Boutteau, ADETEF
Bruno Deschemps, ADETEF, France
Daniel Schmidt, UKL
Doris Ipsmiller, PEPPOL.AT
Dörthe Körner, DIFI
Elisabeth Sundholm, DIFI
Ellen Lücke, Beschaffungsamt, Germany
Jean-Philippe Nadal, ADETEF
Josef Makolm, PEPPOL.AT
Lefteris Leontaridis, EKEVYL
Maria A. Wimmer, UKL
Markus Müller, UKL
Markus Schett, PEPPOL.AT
Paola Fumiani, IC
Piero Milani, IC
Sverre Bauck, DIFI
Trygve Laake, DIFI
Wolfgang Groiß, PEPPOL.AT

[1] From 29th December 2009, CNIPA will be renamed DigitPA (Legislative Decree 1st December 2009, n. 177)
[2] English: Agency for Public Management and eGovernment

Table of Contents

1. Introduction

This Attachment describes the architecture implementation plan for every beneficiary participating in PEPPOL WP2.

These implementation architectures describe the PEPPOL beneficiary’s operations and technologies, the business processes (PEPPOL Profiles) they will enable, connectivity methods and the timelines for their implementation.

It is important to recognize these implementation plans and resultant experience will be used as a guide for newcomers and other participants in the PEPPOL project.

2. VCD service specifications and implementation plans per pilot

1.1 Austria

1.1.1 VCD service provider and governance specification

Complex and multinational projects like PEPPOL have to carefully deal with the management and maintenance of information, processes, tools and applications to ensure reliable results. Well defined governance rules are crucial for the sustainable success of the project both during the project's runtime and beyond the project.

The following subchapter provides an overview on this topic and works as a starting base for future comprehensive elaboration of the governance specification in alignment with the general statements on "governance" (see chapter 13).

Governance can be defined as "the establishment of policies and continuous monitoring of their proper implementation by the members of a governing body of an organization. It includes the mechanisms required to balance the powers of the members (with the associated accountability), and their primary duty of enhancing the prosperity and viability of the organization."[3]

According to the EIF[4], governance is concerned with the ownership, definition, development, maintenance, monitoring and communication of the various elements (policies, standards, requirements, components etc).[5] In PEPPOL governance implies mastery of the technology, systems and organisations in question, ensuring that their combined activities serve the strategic goals and objectives set out by the EC, the governing board and the beneficiaries for the run-time of the project and beyond. Following the EIF approach on governance in a first step a Governance structure/model has to be defined, encompassing involvement of the stakeholders in the governance activities. This model should mainly focus on the following aspects:

- **Specifying decision rights**:
  Which kind of decisions need to be made? Who can make them?
- **Specifying and managing the life-cycles for the artefacts and components**:
  This includes periodic reviews, top-down re-assessments, and taking into account paradigm shifts when they occur in respect to changing environment;
- **Measuring effectiveness**:
  Defining metrics (e.g. key success indicators) as well as using them to evaluate and monitor WP2 related artefacts and take appropriate actions whenever needed.
- The governance model has to be defined in a way that various requirements are taken into consideration:
- **Timeline**:
  Governance has to define rules and processes valid for the project runtime (which is the easier task and includes roles on the piloting) as well as beyond.
- **Managing level**:
  Different stakeholders and interest on different levels (political, legal, organisational, semantic and technical aspects) have to be considered.
- **Regional level**:
  There have to be statements on european level as well as rules on national level which have to be aligned. One has also to take into consideration that there might be rules valid for the PEPPOL-scope and others of a broader base (e.g. EC-wide definitions, data models).
- **Component level**:
  Rules which describe the management and maintenance of the necessary tools/artefacts (in a wide sense) and application.
- **Functional level**:
  Governance has to deal with the different stages of a VCD (from the pre-VCD-stage to stages

[3] http://www.businessdictionary.com/definition/governance.html
[4] The European Interoperability Framework (EIF) was developed within the Interchange of Data between Administrations (IDA) programme of the European Commission and presents a framework for a common understanding of interoperability.
[5] http://ec.europa.eu/idabc/servlets/Doc?id=31597

2 and 3 and – visionary – stage 4) and with the related roles and responsibilities and the decision on the competent actors.

1.1.2 Overall Requirements, Goals and Setting

The Austrian VCD-System will support the Economic Operator in the compilation of a VCD Package and the Austrian Contracting Authority in the validation of a VCD Package from abroad (including VCD Packages of all countries with a rule set taken into account in the ontology). The Austrian Economic Operator turns only to the Austrian VCD-System, which carries out all necessary interaction steps e.g. with the ESP and provides a flexible architecture which, as a long term goal, is able to call all Evidence Issuing Services which are providing an adequate data interface.

Besides supporting Economic Operators and Contracting Authorities, the Austrian VCD-System has to fulfill strong requirements concerning flexibility and efficiency in handling maintenance issues and supporting governance processes:

1. The Austrian VCD-System should be able to take changes in the overall ontology (European ontology and national ontology parts of other participating member states) very efficiently into account. Therefore, the Austrian VCD-System should operate on exactly the same ontology as the ESP without intermediary rule set specific code. If another state, for example, changes their rule set on the ESP, Austria will download and import the whole ontology and operate on top of this new rule set with no further maintenance steps to be taken into account.
2. On the other hand, the Austrian VCD-System must provide an interface for maintaining the Austrian national parts of the ontology and providing means to upload this ontology to the ESP.
3. The Austrian VCD System should also be able to quickly react to changes concerning the Austrian Evidence issuing Services. This includes changes in existing services as well as flexible means to introduce new services into the system.

All together, the Austrian VCD-System must provide a wide set of functionalities regarding the VCD assembly process on the one hand and the maintenance and governance processes on the other hand.

In order to meet those requirements, a semantic service assembly approach has been chosen. Due to the innovation factor of this approach, the technical basis is developed in a research project outside of the national and international PEPPOL project and the results are brought into the Austrian VCD-System.

Roles and Interaction of the Austrian VCD-System and the European VCD-System

Acting as the single point of contact for the (Austrian) Economic Operator, the Austrian VCD-System interacts with the user and optionally the ESP for calculating the needed Evidences and provides the user with the VCD Package as a result of this process.

After initialising the VCD compilation process, the Austrian VCD-System iteratively asks the Economic Operator to provide input regarding the tenderer structure. Alternatively, the user is able to upload an existing VCD Package. In the course of this data input, the Austrian VCD-System already enhances the tenderer structure via reasoning mechanism (OWL/DL Reasoner). In order to calculate the possible Evidences for this specific VCD instance, respectively the given tenderer structure, it calls the European VCD-System or alternatively performs these reasoning steps (Rule Based Reasoning) by itself. Due to the necessity for the Economic Operator to chose and confirm Criteria as well as Evidences suggested by the system, two reasoning steps are to be taken into account (either performed by the ESP or the Austrian VCD-System itself), each followed by a user interaction:

- In the first step, suggestions for the CA-N. (Contracting Authority National) Criteria are delivered and the user has to select the Criteria to be proven (that is, confirm or alter the suggestion).
- In the second step, suggestions for the T-N. (Tenderer National) Evidences are made, again followed by a user decision of which Evidences are to be delivered finally.

In interaction with the Economic Operator, the Austrian VCD-System discovers suitable Evidence Issuing Services, assembles them into service chains, calls the services and collects their output.

Finally, the data is packed and a VCD Package is created, signed and delivered to the Economic Operator. Figure 2-1 gives an overview over this process; the single steps are summarized in Table 2-1.

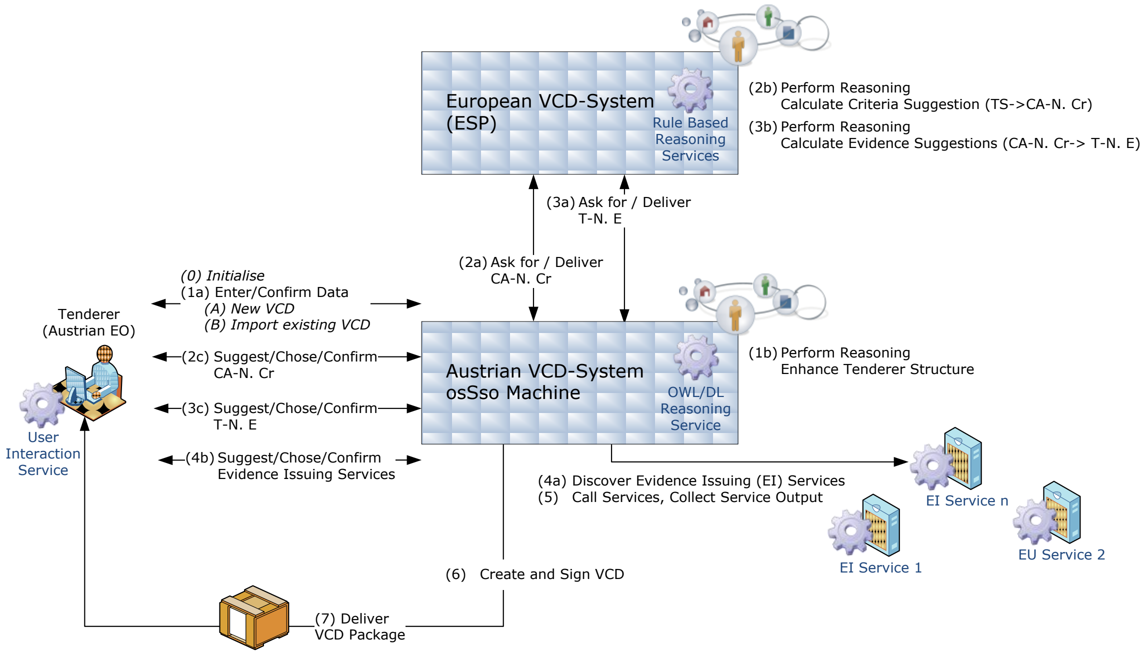


Figure 2-1: Overview of interaction processes between the Austrian Economic Operator, the Austrian VCD-System and the European VCD-System

Step	Name	Description
0	Initialise	The process is initialised with a new VCD Package Request by an (Austrian) Economic Operator.
1a	Enter/Confirm Data	In the case of a new VCD, the Economic Operator is asked to enter tender and tenderer structure data. Alternatively, the Economic Operator can upload an existing VCD.
1b	Perform Reasoning: Enhance Tenderer Structure	In iteration with step 1a, the OWL/DL-Reasoner ads triples to the entered graph according to the underlying ontology.
2a	Ask for / Deliver CA-N. Cr	Once the Tenderer Structure has been entered, the Austrian VCD-System asks the European VCD-System (precisely the ESP-Rule Based Reasoning Service) for the suggested CA-N Criteria for each specific Tenderer Structure Element (TSE= Specific structure of Economic Operators attending a Call for Tender).
2b	Perform Reasoning: Calculate Criteria Suggestion (TS->CA-N. Cr)	In order to answer this (2a) request, the ESP provides a reasoner (Rule Based Reasoning Service 1), which calculates the required CA-N. Criteria, which are again passed to the Austrian VCD-System.
2c	Suggest/Chose/Confirm CA-N. Cr	The Austrian VCD-System passes these Criteria back to the Economic Operator, who, in turn, can accept or alter the suggested criteria list.
3a	Ask for / Deliver T-N. E	Once all Criteria for each TSE have been accepted, the Austrian VCD-System asks the second Rule Based Reasoner Service at the ESP (3b) for the suggested Evidence-List per Criteria. The Austrian VCD System does not utilize the VCD Skeleton Package, since it is capable of performing the necessary steps on its own.
3b	Perform Reasoning: Calculate Evidence Suggestions (CA-N. Cr-> T-N. E)	The Rule Based Reasoner Service 2 calculates all possible Evidences according to the rule set (ontology) and passes this Evidence List to the Austrian VCD-System.
3c	Suggest/Chose/Confirm T-N. E	Again, this suggestion is passed to the Economic Operator, including contextual information like the “Mapping Tree” per Evidence. The Economic Operator can alter this list.
4a	Discover Evidence Issuing (EI) Services	In order to fill in the needed Evidences, the Austrian VCD System has to discover possible Evidence Issuing Services and assemble them to suitable service chains.
4b	Suggest/Chose/Confirm Evidence Issuing Services	Those services / service chains are passed to the Economic Operator for approval.
5	Call Services, Collect Service Output	The approved services are called and the service output is collected by the Austrian VCD System.
6	Create and Sign VCD	After all services have been called, the VCD Package is created, signed and...
7	Deliver VCD Package	... delivered to the Economic Operator, e.g. via email-notification and download.

Table 2-1: Steps in the interaction process between the Austrian Economic Operator, the Austrian VCD-System and the European VCD-System

Architectural Overview and Component Description

Figure 2-2 gives an overview of the architecture of the Austrian VCD-System.
In general, the system is based on the m2n Intelligence Management Framework, which supports the ontology based, service oriented development of applications and provides a wide variety of

functionality for graph- and ontology management, repository management, interface management, GUI building and interaction management, as well as service orchestration and execution handling. Those components act as “background systems” in terms of the Grant Agreement – are therefore funded outside of the PEPPOL project and are marked blue in the architectural drawing below.
On top of this framework, components for specific functionality of the VCD-System are built and are incorporated as plug-ins respectively services (marked as green or white squares). Also the picture contains external services (like, for example the Business Register Extract Service), which are physically located outside the Austrian VCD-System and are called via “Service Wrappers”. The light green boxes represent the reasoners and reasoning services (Rule Based Reasoner and OWL/DL Reasoner), which may optionally be run on the European Service Provider. The components are briefly described individually beneath.

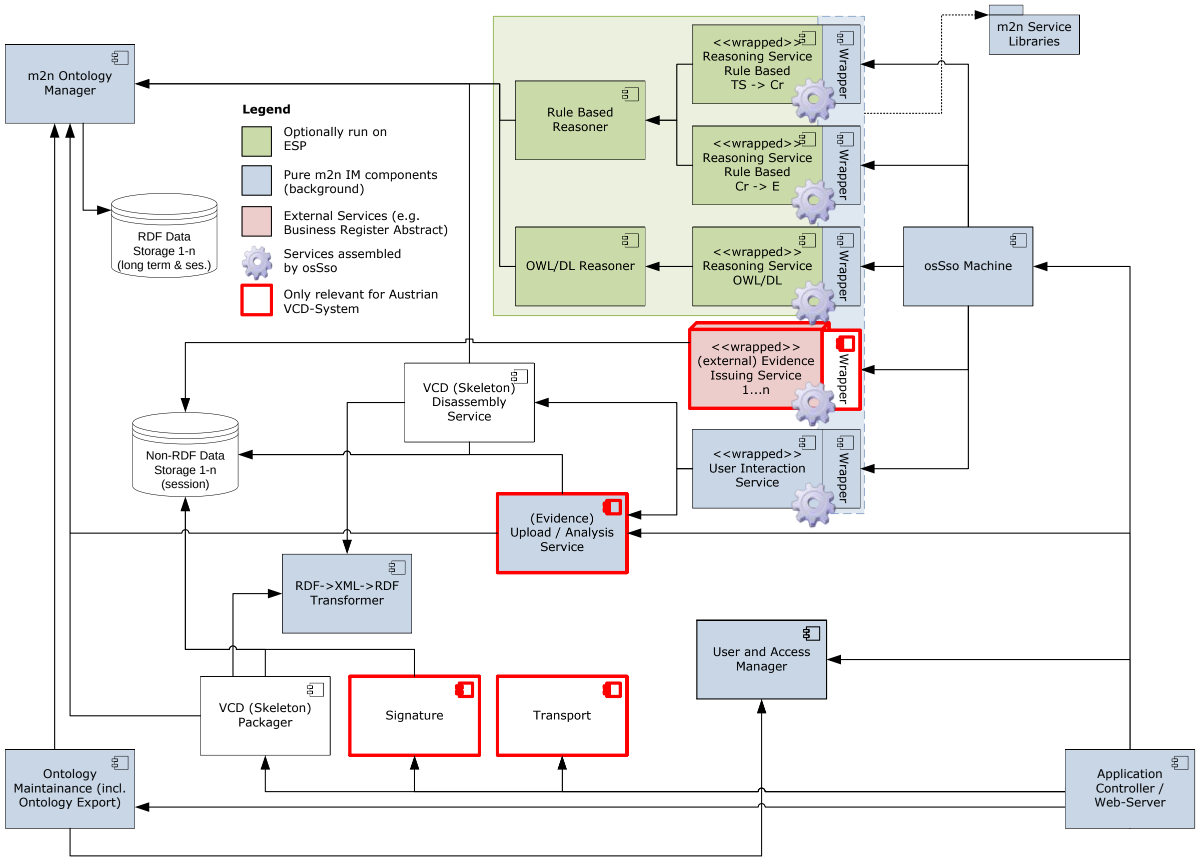

m2n Service Libraries
m2n Ontology Manager
Legend
Optionally run on ESP
Pure m2n IM components (background)
External Services (e.g. Business Register Abstract)
Services assembled by osSso
Only relevant for Austrian VCD-System
RDF Data Storage 1-n (long term & ses.)
Rule Based Reasoner
<<wrapped>> Reasoning Service Rule Based TS -> Cr
<<wrapped>> Reasoning Service Rule Based Cr -> E
OWL/DL Reasoner
<<wrapped>> Reasoning Service OWL/DL
Wrapper
osSso Machine
<<wrapped>> (external) Evidence Issuing Service 1…n
<<wrapped>> User Interaction Service
VCD (Skeleton) Disassembly Service
Non-RDF Data Storage 1-n (session)
(Evidence) Upload / Analysis Service
RDF->XML->RDF Transformer
User and Access Manager
VCD (Skeleton) Packager
Signature
Transport
Ontology Maintainance (incl. Ontology Export)
Application Controller / Web-Server

Figure 2-2: Architectural overview

Application Controller

The Application Controller is in charge of the main application flow. The m2n Execution Engine acts as a service bus, calling the methods and services according to ontology based workflow models and routing the data flows between them. Its responsibility also includes – among others - exception handling and process persistency.

m2n Ontology Manager

The m2n Ontology Manager is a set of components which consolidates and manages RDF data stores. To its client components, it presents a union of relevant sets of data store models as one “virtual” application graph. The m2n Ontology Manager is one of the core components of the whole system, since not only user data is handled here but also any kind of system configuration information.

The m2n Ontology Manager supports a variety of mechanisms for graph querying, model manipulation, several backends for graph persistency, transactional manipulation of graphs and provides event handling and listeners. Various reasoning mechanisms are also part of this component bundle.

RDF Data Storages

Graphs can be stored in various different physical formats like relational databases, native triple stores and files (e.g. n3, rdf/xml) all acting as RDF Data Storages. The data is consolidated by the m2n Ontology Manager.

Reasoners and Reasoning Services

Reasoners infer additional knowledge from an input graph and therefore enhance the graph data the m2n Ontology Manager is operating on. For the scope of the VCD, various types of reasoners will be needed, apart from m2n internal reasoning mechanisms (which acts transparently in the background) the Rule Based Reasoner and he OWL/DL Reasoner (acting as services upon explicit invocations at specific points in time). Whereas the Rule Based Reasoner performs reasoning according to explicit rule sets, the OWL/DL Reasoner performs description logic reasoning on graphs.

The Reasoning Services wrap the reasoners and may additionally expose a web-service interface (not shown in the picture). They delegate the actual reasoning to the base reasoners.
The Reasoning Service OWL/DL calls the OWL/DL Reasoner to enhance the tenderer data. The Rule Based Reasoner is called and wrapped by two Reasoning Services:

- "Reasoning Service Rule Based TS -> Cr" calls the rule based reasoner with the rule set for calculating the Criteria suggestion according to the Tenderer Structure (TS)
- "Reasoning Service Rule Based Cr -> E" calls the rule based reasoner with the rule set for calculating the Evidence suggestion from the Criteria set.

The Rule Based Reasoner and the OWL/DL Reasoner, as well as the three reasoner services may optionally be operated at the ESP. In this case, a proxy service will be put in place on the Austrian VCD Service Provider that acts as an intermediary between the Semantic Service Wrapper and the Reasoner Services.

osSso Machine

The osSso Machine is the central semantic service assembly engine of the Austrian VCD System. It uses a goal oriented approach instead of a conventional process oriented one. For assembling services, osSso does not follow a predefined workflow of service calls. Instead it automatically finds the best service combination to reach a desired state ("goal"). This state is defined by the constraints the underlying ontology poses on its instances. osSso evaluates the validity of an instance graph according to the ontology, finds semantic services that modify the graph into a state closer to full compliance and repeats this process until full graph validity is reached. osSso does not only assemble the Evidence Issuing Services but starts at the very first point of user interaction, orchestrating user input, reasoning services and evidence issuing services. A further description is given in chapter "0 Semantic Service Assembly".

m2n Service Libraries and Service Wrapper

Service Wrappers describe the services semantically in terms of preconditions and predicted results and handle the interaction with the osSso Machine. They form the "semantic shell" around the wrapped service. They translate semantic service calls into native service calls (depending on the underlying service technology, this could be plain Java calls, RMI, web-service calls, etc., or triggering semi automated services by e.g. sending e-mails). m2n Service Libraries provide common methods to be used by various Service Wrappers. Semantic Service Wrappers are briefly described in chapter "0 Service Wrapper".

Evidence Issuing Services

Evidence Issuing Services are VCD-external services that provide Evidence Documents and Evidence Data like the Business Register Extract. The existing automated services are utilized, wherever possible. Where no automated services are available, also semiautomatic or fully manual processes can be triggered.

User Interaction Service (UIS)

The User Interaction Service introduces “the user” as a service to the osSso Machine. The UIS allows user interactions to be treated as service calls, provides meaningful user interfaces to the user and supports user interaction. The User Interaction Service is called several times within the VCD creation process: In interaction with the OWL/DL Reasoner, the user inputs tenderer data via the UIS. In a second and third step, the user confirms or alters criteria suggestions and then evidence suggestions made by the Rule Based Reasoners. As a last step, the UIS allows the user to select the Evidence Issuing Services from the list provided by the reasoner.

RDF->XML->RDF Transformer

The RDF->XML->RDF Transformer acts as a generic translator between RDF and XML based on ontologies, XML schemas and mappings between those schema types using modelled mappings from the underlying ontology to the VCD (Sub-)Schema(s). This transformer is used by the VCD (Skeleton) Packager for RDF to XML transformation as well as by the VCD (Skeleton) Disassembly Service for XML to RDF transformation.

VCD (Skeleton) Packager

The VCD (Skeleton) Packager utilizes the RDF->XML->RDF Transformer to build VCD Packages (and, if needed, VCD Skeleton Packages). RDF data corresponding to the ontology and BLOBs (like Evidence Documents) are transformed into VCD Package or a VCD Skeleton Package, according to the input (Evidence Data and Documents provided or not).

Signature and Transport

The Signature and Transport Components signs the built VCD (Skeleton) Package using national signature services and makes the packages accessible to the user. In a first step, this will be handled via a download function and e-mail notification and might later be extended to utilize the WP8 infrastructure.

VCD (Skeleton) Disassembly Service

The VCD (Skeleton) Disassembly Service also calls the RDF->XML->RDF Transformer, but, contrary to the VCD (Skeleton) Packager to pre-fill the ontology based on an existing VCD (Skeleton) Package. Input data is therefore a VCD (Skeleton) Package, which is uploaded by the user (via the UIS). The output of this step is RDF data according to the ontology and optionally evidence documents that are analyzed and stored (Non-RDF Data Storages and RDF Data Storages) for further use.

Non-RDF Data Storages

These storages store blob data for evidence documents as well as uploaded or built VCD (Skeleton) Packages.

(Evidence) Upload / Analysis Service

The (Evidence) Upload / Analysis Service allows the uploading of existing evidence documents by the user and analyses them to build RDF representation of their metadata according to the ontology. This component is called by the UIS.

User and Access Manager

The User and Access Manager manages known system user, handles authentication (tieing sessions to user accounts) and authorization (is user X allowed to do Y). In Austria the PvP (Portal Verbund Protokoll) will be used to ensure a seamless user experience.

Ontology Maintenance

Ontology Maintenance consists of various components responsible for keeping the ontology up to date. This includes for example a wide variety of features supporting the manual manipulation of

Criteria, Evidences, Criteria Requirements and Evidence Restrictions for the Austrian national parts of the ontology as well as uploading and merging other ontology parts (provided, for example by the ESP) into the whole ontology. Ontology Maintenance also supports the export of ontology (parts), potentially in a variety of formats (n3, rdf/xml, etc.)

Semantic Service Assembly

The osSso Machine is the service discovery, selection and assembly engine of the Austrian VCD-System. It is based on the results of a national research project outside of the PEPPOL project and is, like the other “background” components, part of the m2n Intelligence Management Framework.

Reaching Graph Validity: The osSso Machine

In order to meet the requirements of flexibility regarding changes in the European and Austrian rule set as well as service maintenance, the Austrian VCD-System is directly operating on top of the ontology, that is, the ontology is directly interpreted by the system for service assembly, without the need for intermediary service chains like in conventional SOA approaches. Regarding the requirements to be met, those SOA approaches show various limitations:

- The service descriptions are formulated in SOAP/WSDL and are therefore functional syntactic and not semantic.
- The services chains are defined by a programmer, who has implicit knowledge about why the services are related in a specific manner. This knowledge is not made explicit and is therefore not machine readable.
- Changes in regulation or services require manual adaptation of the processes, which often aggregate and lead to rather complex change processes.

Semantic SOA as a goal oriented approach for service orchestration is the next step in SOA development. An explicit model of the required end status is provided. Rigid execution rules or even code based workflow programming are no longer needed. A semantically rich service description is provided, which is human AND machine readable. Semantic services “know” by themselves which output they can provide under which preconditions. Preconditions (including input data) and output are described in terms of the ontology, also acting as the underlying rule set when describing the valid end target.

Taking this ontology as well as the services into account, osSso

- Validates a given instance against the ontology
- Discovers missing graph parts
- Discovers appropriate services to reach a valid status
- Selects the services that fit best
- Assembles the service chains
- Calls the services, collects their output and adds it to the instance graph.

This process (steps 1-5) is shown in Figure 2-3 and the steps are summarised in Table 2-2.

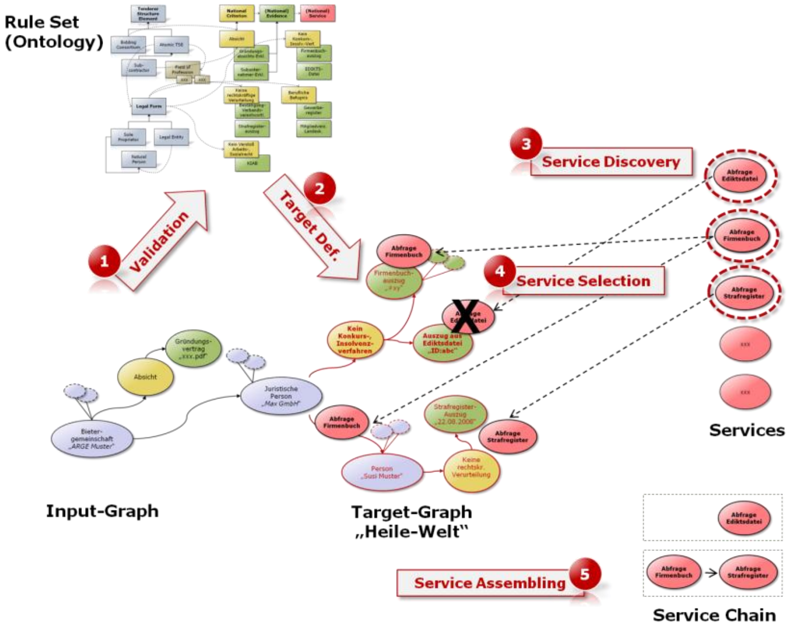


Figure 2-3: Making a graph valid

Step		Description
1	Validation	A specific instance graph is validated, with regards to the rules formulated in the ontology.
2	Target Definition	After discovering violations, one or more virtual "target graphs" are defined.
3	Service Discovery	Services (UIS, Reasoners or Evidence Issuing Services) are discovered, that are able to fill the target graph. If full validity cannot be guaranteed to be reached, services that lead the graph to a just *more* valid status can be utilized to improve validity.
4	Service Selection	The best fit services are discovered according to the defined preferences, taking the quality-of-service parameters into account. In some planning scenarios, this step is performed AFTER step 5, allowing consideration of whole service chains.
5	Service Assembly	The services are assembled to service chains. Service chains are necessary, if one service needs another service to provide the required preconditions.
6	Service Call, ...	After the best service chains are found (and, if required by the rule set, the user has agreed to their invocation), the services are called, their output is collected and added to the instance graph. These further steps are not shown in the picture above. Depending on the validity of the result graph, the process can be repeated iteratively.

Table 2-2: Steps in reaching graph validity

Service Wrapper

Semantic Service Wrappers define - in terms of the ontology - preconditions and expected output of services orchestrated by osSso. Wrappers also transform graphs into input-data of the "native" service they represent, call the service and transform the output of the native service into a graph representation which is then used for graph enhancement. This process is shown in Figure 2-4:

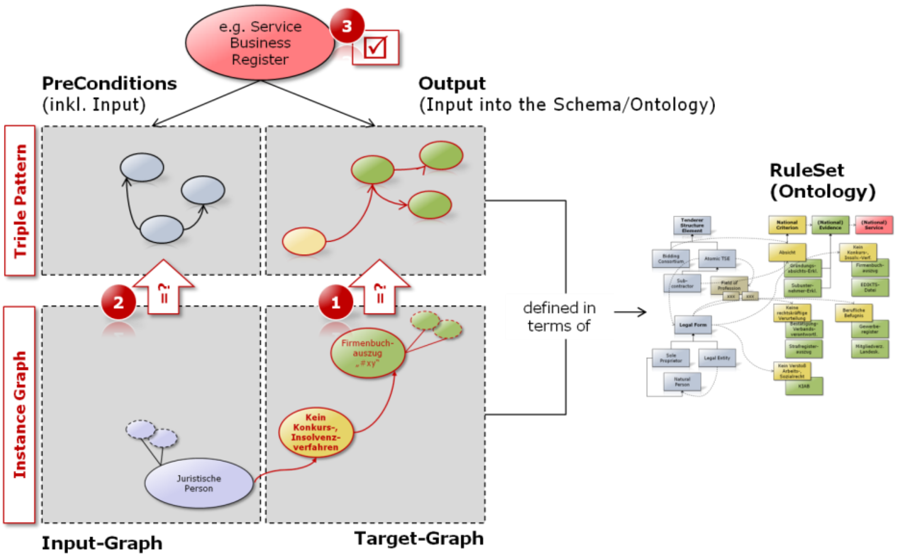


Figure 2-4: Service Wrapper

A service's preconditions and output (e.g. an Evidence Issuing Service like the Business Register) are defined as triple patterns in terms of the ontology. Preconditions also cover the input data required by the service, the output represents the graph enhancement that can be expected from a successful service call. To decide the suitability of a service, osSso considers the expected output and whether having that added to the instance graph can bring the instance graph closer to a valid state.
For this purpose, osSso checks whether the output triple patterns of the service match the identified missing triples that form the target graph. If this is the case, osSso evaluates, whether the preconditions are met by comparing the input graph to the triple pattern defining the preconditions. If no match is found, the preconditions amend the target graph, defining a new goal to be met by other services. This process is iterated, until satisfying service chains are found which can be ranked relatively to each other by calculating their “cost functions” using quality of service parameter. From this ranked list, a “best” service chain can be selected automatically or suggested to the user, who can then agree to it, or select a different service chain.

The osSso Orchestration Process in the VCD Context

The following picture (Figure 2-5:)sketches the specific results of the service assembly process orchestrated by the osSso Machine in the Austrian VCD-System. On the vertical axis, the services (User Interaction Service, Reasoner Services and Evidence Issuing Services) are outlined. The horizontal axis represents time. The Process progresses from left to right.

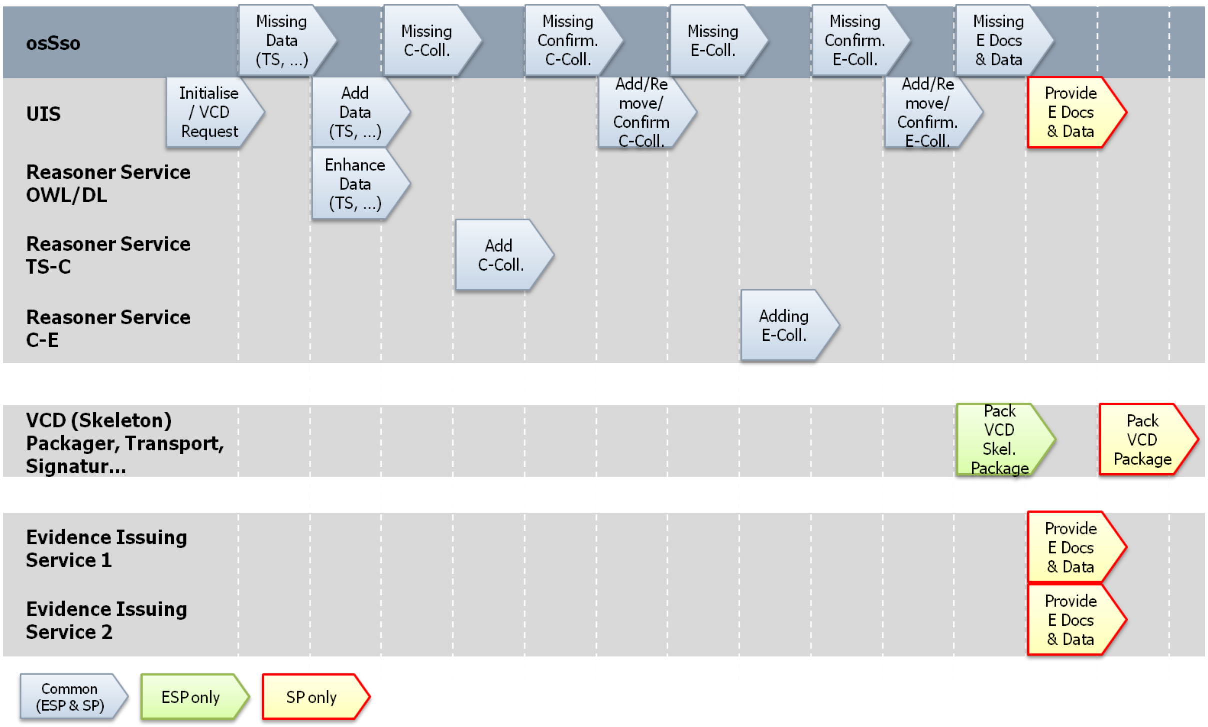


Figure 2-5: The osSso Service Assembly Process

After the process has been initialized by the user (via the UIS – User Interaction Service), osSso discovers by comparing the existing data to the schema represented in the ontology, that the tenderer data (e.g. Tenderer Structure, tenderer name, name of legal representative) is missing. There are two services, which can provide these missing graph parts: the user via the UIS and successively the Reasoner Service OWL/DL enhancing the data input of the user.

After all necessary tenderer data in order to reach compliance is available, osSso discovers that Criteria Collectors are missing. The only service able to provide this data is the Rule Based Reasoner Service TS-C. In the next step, osSso is missing the criteria confirmation information of the user. Again this can only be provided by the UIS; the user can alter the suggestion and finally confirm the Criteria.
In the same way, information about which Evidence is to be finally included is provided iteratively. First osSso discovers missing Evidence Collectors, the Reasoner Service C-E is able to provide this data and finally the UIS is asked for confirmation.

Finally, osSso is looking for services able to provide Evidence Documents and Evidence Data. There can be two types of suitable services. First of all the user could upload existing Evidence Documents (e.g. a valid Business Register Extract pdf) via the UIS, if this form of tenderer issued evidence is compliant according to the ontology. Alternatively, Evidence Issuing Services (e.g. Business Register) offer this information. osSso calls the best fitting services (after user confirmation), collects the data and hands the data over to the VCD Packager for creating the archive according to the internationally agreed upon VCD Schema. Finally the VCD Package is signed and delivered to the user.
For more information on the reasoning process, please refer to section.

1.1.3 Linking up VCD service providers with local evidence issuing services

As already stated above, reaching the maximum level of automation and IT support is one of the main requirements that have to be met when designing and setting up the Austrian VCD-System. Therefore,

a system will be established, built upon an architecture prepared to utilize as many existing IT services as possible in the course of creating a VCD package. Due to project limitations, the number of services connected within the scope of the project time will be also limited. Those services will be included into the system by Semantic Service Wrappers as briefly explained in chapter “0 Service Wrapper”. Wherever no fully automated IT services exists, or legal / organisational constraints prevent their inclusion, a semi automated process will be invoked by the Austrian VCD-System with an interface for a representative of an issuing body to be able to enter the data manually into the VCD-System e.g. by uploading Evidence Documents. Of course, the Economic Operator himself can act as an issuing service, if the ontology rule set defines him as a valid service. In the Austrian VCD-System the user will be supplied with a suitable Evidence Upload (and Analysis) Service.
However, it is one of the main objectives of the Austrian VCD-System to provide an architecture that allows a step by step integration of as many automated services as possible, providing the highest possible level of support to Austrian Economic Operators.

1.2 France

1.2.1 Introduction to French pilots

This document will describe the French pilots which will be conducted by French Sub-consortium with French partner including contracting authorities and economic operators for WP2 in relation with WP1 and some French tendering platforms. Those pilots will be bi-directional : from French contracting authorities to cross-border economic operators (plus national) and from cross-border contracting authorities to French economic operators.
The target is to have two families of pilots, one with stand-alone components, one with online components possibly integrated to tendering platforms.

1.2.2 Components used for the French pilots

The components we plan to use are:

- VCD Viewer
- VCD Builder
- Access to WP1 service :
  - Or thru Interface (Signature (ie : digital certificate) validation Interface)
  - Or via a client provided by WP1
- Documents Signer (if requested by Contracting authorities)
- Interface to VCD signer (Italian Signature Interface)
- Access to European VCD Service :
  - Or thru web user interface
  - Or thru interface

Sharing Overview:
The piloting activity for France combines different resourcing options:

- Project common resourcing [common components] : European VCD Service, Access to WP1 service
- Components reuse from other teams [Reuse Shared] : VCD viewer
- Internal development with sharing [Make&Share] : , VCD builder
- Use of background systems [Beneficiary specific] : tendering platform (including eProcure)
- Off-the shelf solutions : documents signer

1.2.3 VCD Builder and components

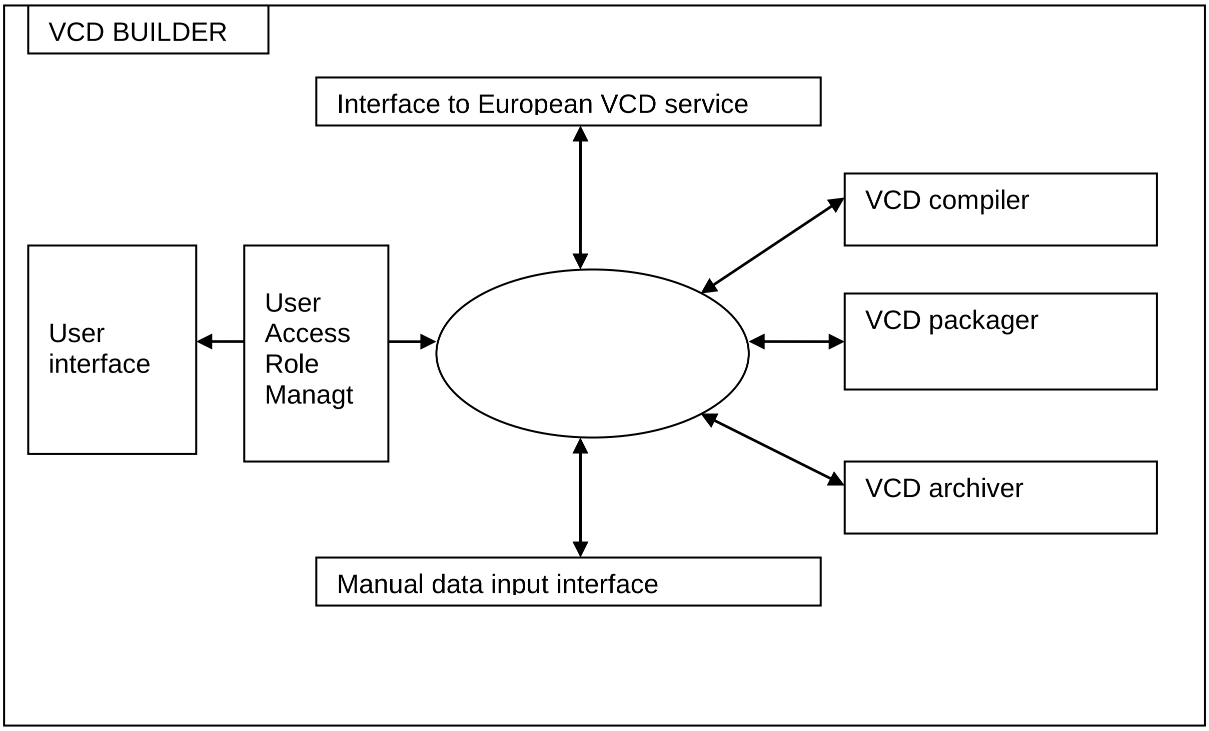


Figure 2-6: VCD Builder – Component building blocks of the National VCD system

1.2.4 List of Component building blocks

Component building blocks	Phase	Type	Licence
Manual data input interface	Test pilot	Reusable component	GPL 3
User access and role management incl. authentication	Production pilot	Reusable component	GPL 3
VCD packager (creates VCD packages)	Test pilot	Reusable component	GPL 3
VCD compiler (creates VCDs)	Test pilot	Reusable component	GPL 3
VCD Archiver (creates VCD container)	Test pilot	Reusable component	GPL 3
Interface to European VCD service	Production pilot	Reusable component	GPL 3
VCD builder (general VCD system)	Test pilot	Reusable component	GPL 3

Table 2-3: List of component building blocks

1.2.5 Test pilots scenario

- Who are the users
  - SAE
  - EBourgogne
  - CG10
  - E-Santé Alsace (ENRS)
  - UNIHA (French hospitals)

- What case will be used?
  - ENRS : medical portal : build and run
  - TBD for others

- What will be the expected results and success measures?
  - End-user Interaction thru UI with ESP using tender notice information
  - Request VCD skeleton from ESP
  - Package VCD with stand-alone VCD packager if decision (Proposal to implement VCD in Demo tool)
  - View VCD with stand-alone VCD viewer if decision (Proposal to implement VCD in Demo tool)

- What technology components will be available (target) for test pilots?
  - File signer
  - Signature verification
  - VCD Viewer (development) – stand alone application

  - VCD Builder(development) – stand alone application
  - Web user interface to ESP interface

1.2.6 Production pilots scenario

- Who are the users
  - SAE
  - EBourgogne
  - CG10
  - E-Santé Alsace (ENRS)
  - UNIHA (French hospitals)

- What case will be used?
  - ENRS : tele-medecine software
  - TBD for others

- What will be the expected results and success measures?
  - End-user Interaction thru UI with ESP or NSP using tender notice information
  - Request VCD skeleton from ESP thru UI or NSP
  - Package VCD with stand-alone VCD packager or NSP
  - View VCD with stand-alone VCD viewer or NSP
  - Verify signature on tendering platform thru WP1or thru NSP
  - No use of BUSDOX

- What technology components will be available (target) for production pilot?
  - File signer
  - Signature verification
  - VCD Viewer (version 1) (online service and stand-alone)
  - VCD Builder(version 1) (online service and stand-alone)
  - VCD Builder to ESP interface

EU Economic Operators and French Contracting Authorities

- Which cross-border dimension is contained?
  - Tender object is cross-border
  - Ontology mapping tool validation
  - VCD exchange
  - Cross-border certificate validation for VCD and bid
  - Multilanguage

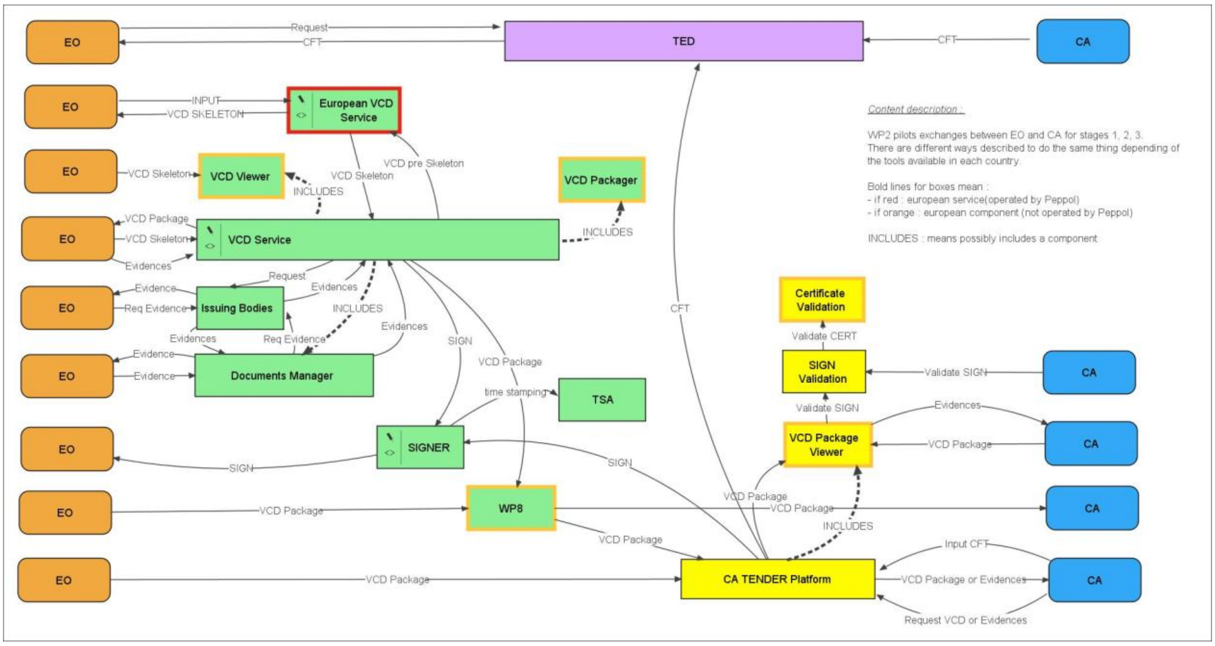
- Which cross-border dimension is NOT contained?
  - Signature done on tendering platform
  - No use of BUSDOX

- Cross-border Economic operator has to register and to use French tendering platform

EU Contracting Authorities and French Economic Operators

- Which cross-border dimension is contained?
  - Ontology mapping tool validation
  - VCD exchange
  - Cross-border bid

- Which cross-border dimension is NOT contained?
  - Use of WP8 ?
  - Shared signer to be provided



Content description :

WP2 pilots exchanges between EO and CA for stages 1, 2, 3.
There are different ways described to do the same thing depending of the tools available in each country.

Bold lines for boxes mean :
- if red : european service(operated by Peppol)
- if orange : european component (not operated by Peppol)

INCLUDES : means possibly includes a component

Figure 2-7: Representation of general French pilots processes

1.3 Greece

1.3.1 VCD service provider and governance specification

Greece is joining PEPPOL WP2 thru the enlargement process and it still defining its implementation.

1.3.2 VCD service provider technical setup and architecture

Greece is joining PEPPOL WP2 thru the enlargement process and it still defining its implementation.

1.3.3 Linking up VCD service providers with local issuing services

Greece is joining PEPPOL WP2 thru the enlargement process and it still defining its implementation.

1.4 Italian VCD system infrastructure

1.4.1 General overview

The Italian piloting solution will establish a system that covers the service demand at national level. Services are offered to the national community of Economic Operators and Contracting Authorities.

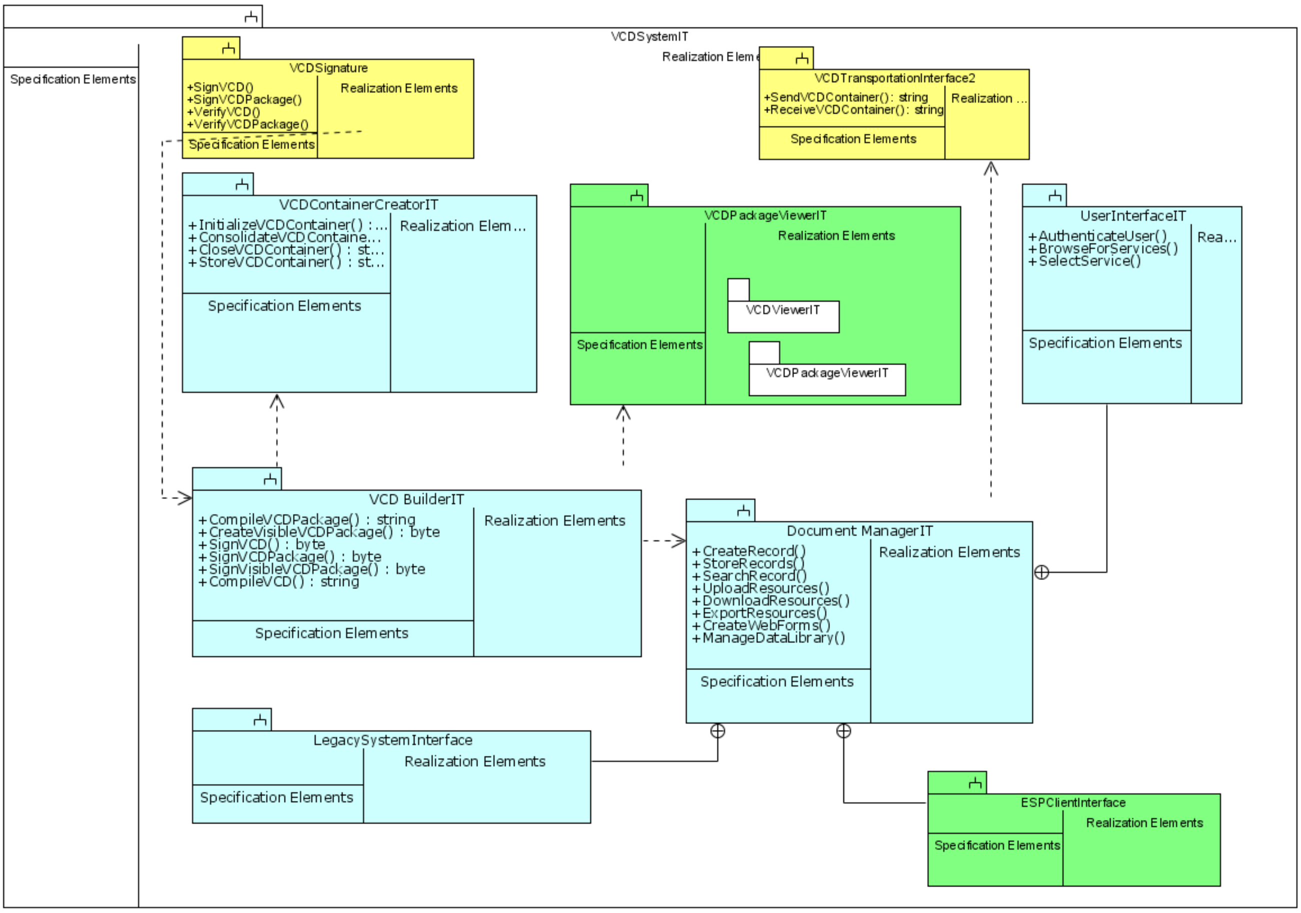


Figure 2-8: VCDSystemIT

Name	Documentation
VCDSystemIT	Represents the Italian implementation of the infrastructure required for operating the VCD service. The Italian VCD System defines a domain where different sub-systems are present with specific purposes. Every sub-system will have independence from different perspectives: 1) Design: will be partially shared with Peppol existing teams and partially operated internally at the national beneficiary level; 2) Implementation: includes multiple options that primarily include: a) integration of off-the-shelf components, b) reuse of Peppol shared resources, c) self managed implementations; 3) operations: it will activate operations under the control of three basic User roles: Administrator: the generic resources manager National Service Provider: the operator managing the added value functions Service Requester: The service requester allowed and supported by the system. It includes the Economic Operator and the Contracting Authority. There is anyway a differentiation between the roles and a corresponding distinction of functions they are supposed to perform.
VCDSignature	The “VCDSignature” sub-system implements the Signing and Verifying services that the ServiceProvider MAY activate over his created artifacts: VCD, VCDPackage.
VCDTransportationInterface2	The interface allows to interact with the BusDox transportation system
VCDContainerCreatorIT	The " VCDContainerCreator " sub-system creates the running platform to two main packages, i.e. the "VCDContainerManager" and the "VCDContainerAssembler". Four basic operations get support within the sub-system. The list of such operations identifies the following functions: a) Initialize; b) Consolidate; c) Close; d) Store. These operations are all directed to form and transform every instance of the “VCDContainer".
VCDPackageViewerIT	The sub-system implements viewing capabilities. Two integration strategies are considered: a) integration of Shared resources; b) specific design and implementation.
VCDViewerIT	The Package included in VCDPackageViewerIT sub-system.
VCDPackageViewerIT	The Package included in VCDPackageViewerIT sub-system.
UserInterfaceIT	The User Interface for the Italian NSP relies on the functions and services operated by the document management system "Document Manager IT".

VCD BuilderIT	The VCD Builder manages resources, activities, interactions to create different Peppol artifacts as the instances of VCDs, VCD Packages and VCD Containers. The VCD Container represents the physical storage and delivery unit of every object generated during the compilation of the different VCD objects. It also contains all the preparatory documents that the VCD has classified and referenced.
Document ManagerIT	The “Document ManagerIT” sub-system represents the pivot component of the Italian VCD System. The core resources in the VCD processing activity are “ Records”. These records are primarily managed by generating or importing different types of electronic documents. The critical mission of the system is to manage this large set of resources in all possible and convenient manners.
LegacySystemInterface	The LegacySystemInterface provides the linking mechanism to interconnect the Italian VCD System "VCDSystemIT" with the existing system infrastructure operated by the local NSP (National Service Provider). The Interface provides invoking services for the retrieval of resources that have already been filed by users into the legacy system.
ESPClientInterface	The ESPClientInterface allows the direct link between the "DocumentManagerIT" system and the "European Service Provider" system. The interface will be activated in low priority. The first step implies the direct interaction of users with the ESP.

Table 2-4: Overview of the Italian VCD System and its components

1.4.2 SignVerifyCore

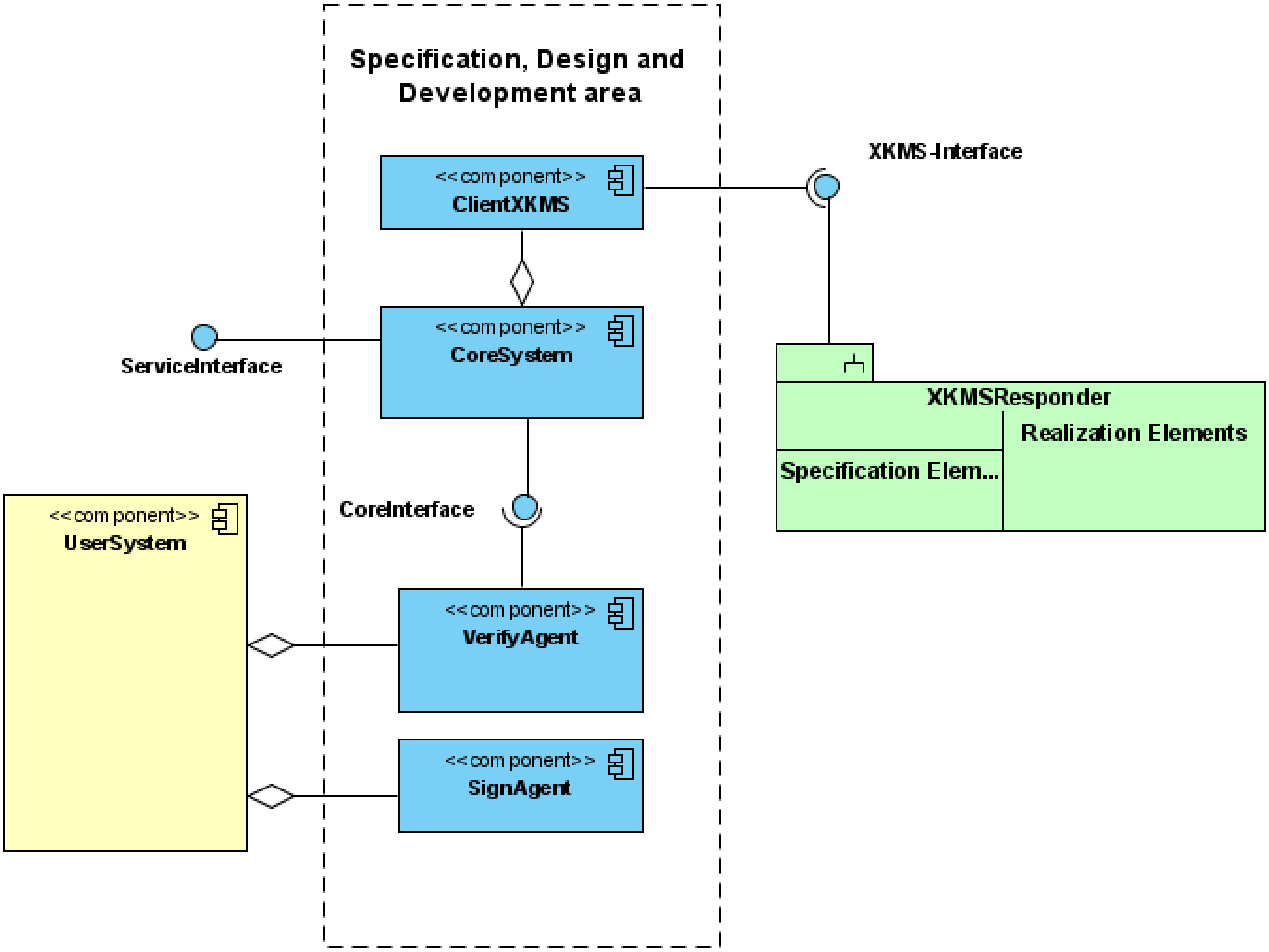


Figure 2-9: SignVerifyCore

1.4.3 Validation Service for cross-border interoperability

The current specification addresses the implementation in Italy of the Peppol WP1 strategy. It also specifies a list of functions to perform the signing and verifying of instances of different Peppol's WP2 generated artefacts as the VCD, the VCDPackage and VCDContainer. These generated objects come from the compilation processes defined in WP2 Pilots. The current specification document covers an integrated scenario where two different work-packages build an integrated solution.
Starting assumption is that every signer is entitled and enabled to generate advanced signatures according to the rules defined within his/her own domain. The Use Case depicted below defines the two actors affected by the present specification. Further details are given by the UML reference base:

- A (by the project web site): [http://bscw.uni-koblenz.de/bscw/bscw.cgi/d1464952/20091106_VCDSigner%26VerifierUseCase.PDF]
- B (by the modelling reference base):

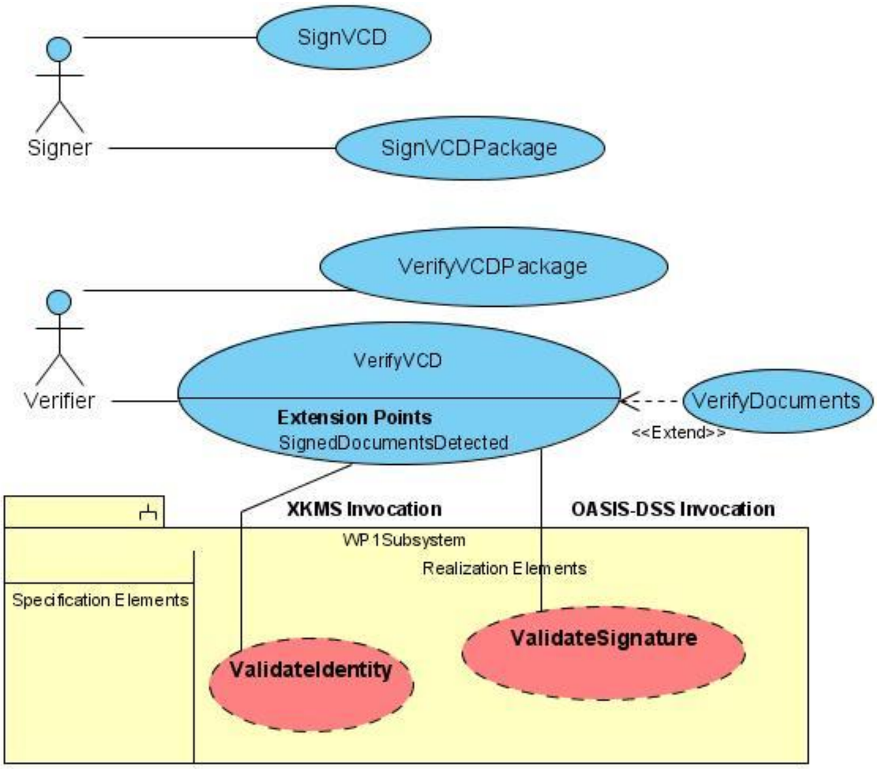


Figure 2-10: Signer & Verifier UseCase

Every business document shared within the PEPPOL community, either a VCD or the contained “Document” requires a satisfactory level of interoperability and this affects also the “digital signatures” applied by signers.
For the validation of generated signatures, Peppol_WP1 has set up a double interface:

- an XKMS interface (for checking the validity of signer digital identity)
- an OASIS DSS (for checking the validity of the signature)

The exposure of these interfaces and the set-up of the WP1 Sub-System is in line with the TRUST MODEL of PEPPOL. Every verify action has to rely on a XKMS trusted environment.

XKMS

You get full description from Peppol D1.1 part 5_XKMS Interface Specification from [http://www.peppol.eu/deliverables/wp-1/d1-1-part-5-xkms-interface-specification] .
XKMS supports validation of X.509 certificates when adopted for signing or in any further application defined in Peppol.
The WP1 architecture layout provides for a central XKMS Responder. The architecture provides also for possible satellite XKMS Responders to run under the management of the different project members.
The use of the most qualified validation server to perform the X.509 validation has to be supported by a Peppol implementation of TSL (Trust-service Status Lists) [ETSI standard TS 102 231 v2.1.1 - Trust-service Status List]
The validation of the Certification Authority starts with an inquiry to the central XKMS server, that will forward, if necessary, to the suitable local XKMS responder
A two phase implementation is planned:
XKMS_Phase#1 : The XKMS Responder will be set–up by “bos – Bremen Online Services” and will support validation against a predefined set of CAs.
XKMS_Phase#2: forwarding to ancillary XKMS Responders, if present, in case of broader coverage.

XKMS Interface (waiting for specification)

XKMS_IF01: OutBound validation requests sent to the central XKMS Responder
XKMS_IF02: InBound validation requests originated by the central XKMS Responder

Prerequisites

1 - Compliance to the XKMS profiled protocol as described in [http://www.peppol.eu/deliverables/wp-1/d1-1-part-5-xkms-interface-specification]
2 - WSDL published by "bos – Bremen Online Services"

XKMS Implementation components

XKMS_IMPL001: XKMS Client application: invokes remote validation to the Central XKMS Responder. The operation is performed through the [**XKMS_IF01:** OutBound validation request] ;
XKMS_IMPL002: local activation of an XKMS Responder. The local XKMS Responder

- adopts the standard: http://www.w3.org/TR/2005/REC-xkms2-20050628/;
- Implements the Peppol profile: WP1_D1.1 Part. 5.
- Builds a mutual trusted relationship between each local XKMS responder and the central one.

The basic functions

The basic functions to be implemented are:

XKMS_IMPL001A: XKMS Request preparation, Submission into the Central XKMS Responder, preparation for receiving and interpreting the reply from the server.
XKMS_IMPL001B: Activation and handling of the synchronous operation mode
XKMS_IMPL001C: Activation of SOAP protocol 1.2
XKMS_IMPL001D: Trust between Client and Server will be supported by signed messages (Request/Response)
XKMS_IMPL001E: The X.509 Certificate is core part of the XKMS Request
XKMS_IMPL001F: Making of a JAVA Library (web interface / java-application) for reuse by organizations interested into the validation system

TSL

Interaction with the PEPPOL/WP1 project team is requested in order to define the TSL service implementation.

1.4.4 Signing and Verifying methods

The VCD pilot applications will accept and support the presence of Digitally Signed objects at different levels. We pay attention to three primary objects, i.e.

- The basic attestations in every possible electronic format,
- The VCD(XML) instance,
- The VCDPackage (XML) instance.

Please refer to the chapter [
Glossary] for a definition of these concepts.

A valid Peppol business transaction can be performed without any Electronic (Advanced or not) Signature. Nonetheless, Peppol cannot ignore the existing regulation at the European and national level to give legal validity to electronic documents while producing or sharing them across the network. In case some Electronic Signatures are present they shall comply with the signing methods and formats listed below.
Digital time stamping, gives a trusted evidence of the fact that the signature existed at a certain time. If used, Time stamping must conform the RFC 3161 format and shall be included in the signed file (non detached).

1.4.5 Signature formats

The following signature formats have been investigated:

- PKCS#7-CMS: Largely adopted standard; based on standard RFC 2315 – RFC 3852 . The cryptographic envelope must be of type signedData (OID: 1.2.840.113549.1.7.2). It must contain the X.509 Signer's Certificate. It allows for multiple signatures (countersignature, parallel and nested signature) and time stamping
- CAdES: is described in the specification ETSI TS 101 733 ver 1.7.4. Basically, it is an extended form of CMS.The formats CAdES-BES and CAdES-T (when the Time Stamp is present) will be supported. It allows for multiple signatures (countersignature, parallel and nested signature) and time stamping.
- PDF embedded: It is based on the ISO32000-1 standard specification of the PDF format as profiled by ETSI TS 102 778 part 2 PADES-ISO32000-1. It allows multiple signatures only as the nested signature as a.
- XML Signature XAdES: It is based on the specification ETSI TS 101 903. The formats XAdES-BES and XAdES-T (when the Time Stamp is present) will be supported. It allows for multiple signatures (countersignature, parallel and nested signature) and time stamping

Further features that have been investigated are:

- Countersignature: A counter-signature is a digital signature of another digital signature. As such, it only provides authenticity of the signature, and not of the message's content. A digital signature can be countersigned by multiple countersigners. An example of a Peppol application that might use countersignatures is the signing by public officers of Declarations signed originally by Economic Operators. It is not allowed in PDF.
- Parallel signature: allowing multiple signers to apply their signature to the same content. The signature does not sign the previous signatures. It is not allowed in PDF.
- Nested signature: allowing multiple signers to apply their signature to the same content, but in this case the following signatures apply also to the previous signatures.
- time stamping: a signed evidence by a trusted Time Authority in order to prove that a file existed before a certain time.

1.4.6 Use Case: Getting a signed VCD – Virtual Company Dossier

Glossary

VCD-Container: physical container of every instance of a VCD-Package. It hosts the VCD- Package and all the electronic documents, of any type, that have been collected before compiling a VCD. This can be a file of .zip type containing any kind of files defined the VCD concept.

VCD-Package: structured electronic document in XML format containing the global set of VCDs required for attending a Public Tender. The set of VCDs is added of information necessary to connect the package to the target tender context. The current structure (draft 20091009) requires the following information: ID, Call for Tender ID, VCD Schema Version, Target Country, Contracting Authority, VCD-Package Content, Content Signature <PM comment> The XML Signature, at the VCD package level, will be addressed as priority #2. Priority#1 concerns the XML signature within the VCD document. </PM comment>.

VCD: structured electronic document in XML format containing the global set of information generated while performing a VCD Service request. The VCD, being an XML document, contains data and metadata. Data and metadata are those inserted by the editing application or service officer. Metadata in particular address the set of physical binary objects inserted into the VCD Container as a set of evidences or similar documents.

Document: Every physical electronic document (signed or not), i.e. the object inserted into the VCD-Container or placed into an external location. Warning: the documents located outside the Container, i.e. outside the control of the VCD application, will be outside the SignedVCD proposal and consequently ignored.

1.4.7 Implications for a Signature application within the VCD service modelling

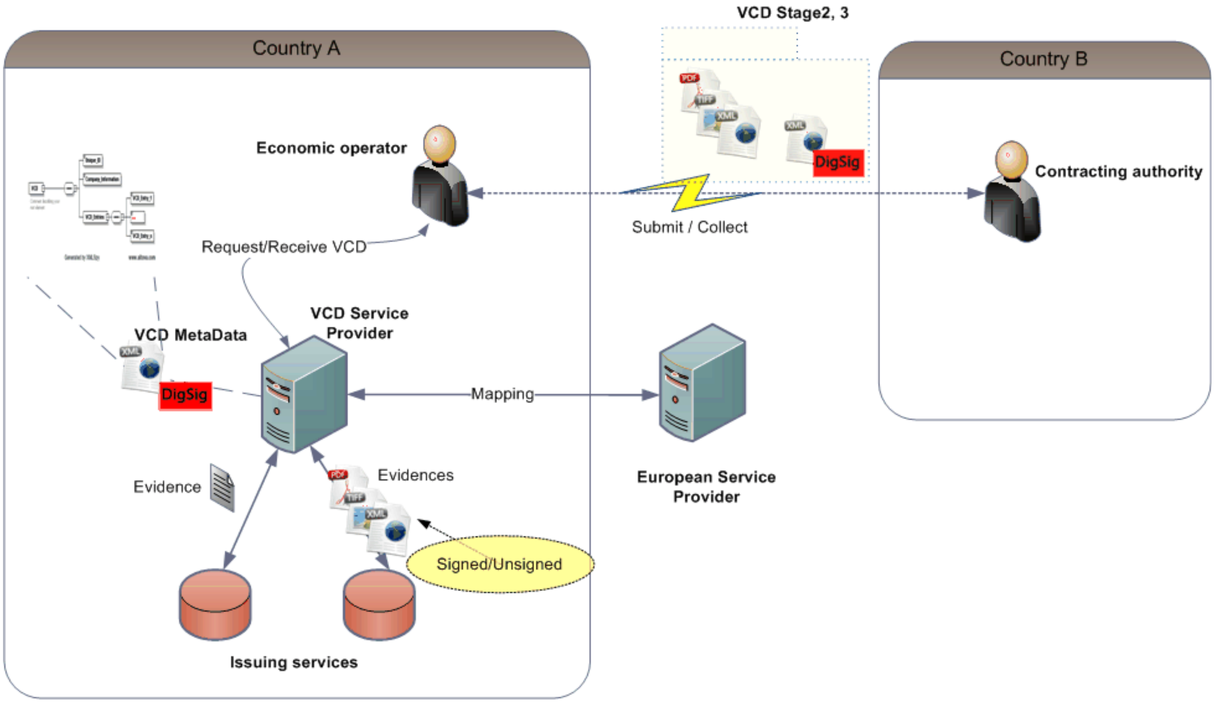


Figure 2-11: Signature application within VCD service

The VCD Piloting activity in Peppol adopts different staging solutions. The present description applies to stages#2 and stage#3 (details are offered within Deliverable D2.1 – 'Functional and non-functional requirements specification for the VCD'
Stage#2: http://www.peppol.eu/deliverables/wp-2/wp2-d1-chapter-2-2-vcd-simple-package-stage-2/view
Stage#3: http://www.peppol.eu/deliverables/wp-2/wp2-d1-chapter-2-4-vcd-advanced-package-stage-3/view

According to the business processes to be activated to support the indicated stages, the signature strategy can get local modifications.

According to the present WP2 VCD process modelling status, represented in [http://bscw.uni-koblenz.de/bscw/bscw.cgi/d1420667/VCD%20Compilation%20Process.jpg] and for a generic initial overview, we consider some abstract processes and the effect they have for the signing/verifying strategy:

– **VCD_Proc001**: Compilation , the task mainly performed by the VCD Service Provider (including the VCD Signing as Signing priority#1);
– **VCD Proc002**: Consolidation. the task performed bv the VCD Service Provider and the

ts compiled VCDs
e [VCD Container]
;
ntained documents
a compiled VCD.

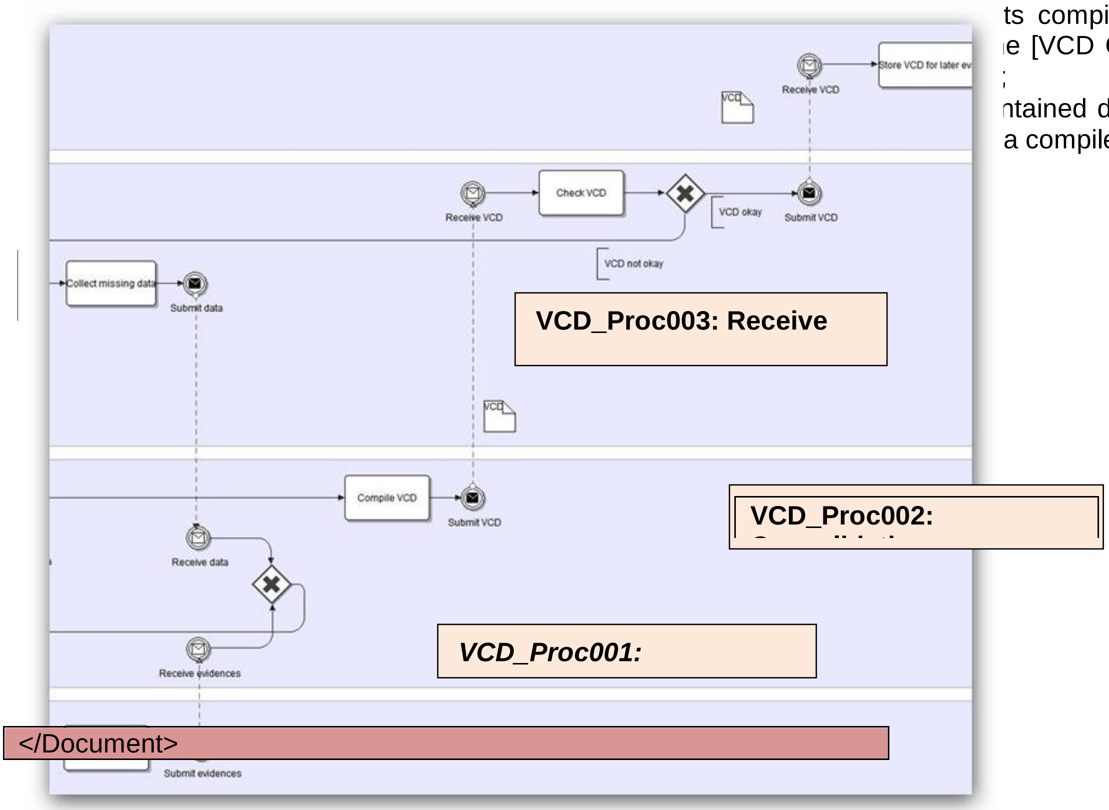


Figure 2-12: VCD compilation: Business Process snapshot (under revision)

Objects representation

For every object listed in the [0
Glossary] chapter, we will get a representation, both logical and physical by the modeling activity and the reference base that's currently located at [http://bscw.uni-koblenz.de/bscw/bscw.cgi/1428249]. In particular, the object "Document" is represented within the VCD(XML) by the element "Document" where it gets a representation close to the structure shown below. Every single instance of a physical evidence/attestation will get this generated set of metadata elements.

The yellowed segment outlines the part that comes from the verification of pre-existing signatures on evidences that are created or received during the 'VCD_Proc001: Compilation' process. This individual evidence signing/verifying activity is considered as part of existing document management solutions and is currently not affecting the current specification, i.e. the SignedVCD Use Case definition.

Results of evidences verification

VCD.document.VerificationResultIndicator
This element is used to summarize the result of the verification of existing signatures on evidences.
When evidences have already a signature, the verification is done by the existing verification procedures within the organization and the domain that is operating the compilation service. The verification results are inserted into the segment "Verification Result Indicator".
Candidate elements to enter such segment are: Identity of Signer, validity of signature, presence of Time Stamp, how the verify was done (within/outside Peppol services).

Hash

VCD.Document.Hash
Every individual evidence managed by the compilation activity will be referenced within the VCD(XML) instance by an adequate set of elements (URI, Type, etc). The linking element for preventing replacement or loss of such physical evidences will be the Hash element.
The Hash algorithm recommended by WP1 is SHA256 (size 32 characters)

Human readable representation

The visual representation of a VCD instance is a prerequisite to a signing activity bearing legal effect; we propose three initial options:

- VCD(XML) transformation into an **XHTML** representation by using a specific style-sheet;
- VCD(XML) transformation into a **PDF document** that's embedding the original VCD(XML)
- VCD(XML) transformation into a **PDF document (no embedding).** The VCD(XML) and the PDF document exists as separated documents and get separated signatures.

VCD format

The format of documents, VCD(XML) and VCD Package(XML), according to the ongoing specification activity in WP2 will be compatible with the OASIS UBL2.0 (Universal Business Language) standard.

Signing a VCD

This is the sub-task (part of VCD_Proc001) activated by the Signer within our Use Case. It affects the VCD(XML) instance created by the "VCD_Proc001: Compilation" procedure.
The signature will be compliant with the directive EC/1999/93 and by applying the UBL2.0 compatibility principle it will have a specific profiling described in a draft document under evaluation (OASIS TC [UBL XAdES Profile Version 1.0 - http://lists.oasis-open.org/archives/ubl-security/200908/doc00000.doc]).

There is at the moment an open discussion about the use of joined or separated methods, i.e. XADES (ETSI TS 101 903) or PADES (ETSI TS 102 778 part.5). This is related to the chosen Human readable representation.
Concerning the pure XML object, i.e. the VCD(XML), it should be used the method defined in: Enveloped http://www.w3.org/TR/xmldsig-core/#def-SignatureEnveloped.
This allows for an easier parsing activities, independent from the signature. That choice will be particularly convenient in case of adopting the UBL format. The formats to be used are XAdES-BES o XAdES-T.

Electronic Signature format for VCD(XML)

The signature must conform to the XAdES specification [*ETSI TS 101 903*].
[*Digest Algorithm*] The URI on property[Algorithm] of element [*DigestMethod*] is :
http://www.w3.org/2001/04/xmlenc#sha256
[*Signature Algorithm*] The URI on property[Algorithm] of element [*SignatureMethod*] is :
http://www.w3.org/2001/04/xmldsig-more#rsa-sha256 as stated in RFC 4051.

Canonicalization methods

The signing application for XML has to provide for the element [*SignedInfo*] a URI value:
http://www.w3.org/2006/12/xml-c14n11 or
http://www.w3.org/2006/12/xml-c14n11# or http://www.w3.org/2006/12/xml-c14n11#WithComments

Transformation methods

The minimum set of transformations to be supported is
Base64 through the URI: http://www.w3.org/2000/09/xmldsig#base64,
Xpath through the URI: http://www.w3.org/2002/06/xmldsig-filter2
[Canonicalization]
http://www.w3.org/2006/12/xml-c14n11# or
http://www.w3.org/2006/12/xml-c14n11#WithComments
[Enveloped] through the URI: http://www.w3.org/2000/09/xmldsig#enveloped-signature that URI ha to appear on property[Algorithm] of element [*Transform*] (specification RFC 3275).
When an XPath transformation is applied it is necessary to indicate that option for the element <ds:Transforms> within the element <ds:SignedInfo>. Transformation syntax shall comply with recommendation XPath Filter 2.0 as indicated by the URI: http://www.w3.org/2002/06/xmldsig-filter2 or http://www.w3.org/TR/xpath20

WEB application

Verifying a Virtual Company Dossier

The Receiver of a VCD Container, that's the container of any physical instance of VCD Package (XML), VCD (XML) and Documents, regardless of her/his role within the VCD business process, will check the validity and integrity of the included VCD (XML) via a Web application.
This comes through the following checks:

- Verification and Validation of signature present on every single VCD included in the VCD Container (one by one)
- Verification of integrity of the VCD Container by comparing newly generated Hash strings with those stored within the "*VCD.document*" metadata.
- Optionally, the verification of original signatures present on individual "evidences" in case the Receiver challenges the "*VCD.document.VerificationResultIndicator*" data .
-

All listed "Verification" actions, i.e. the VerifyVCD and VerifyDocument, if necessary, will make use of the XKMS Responder.

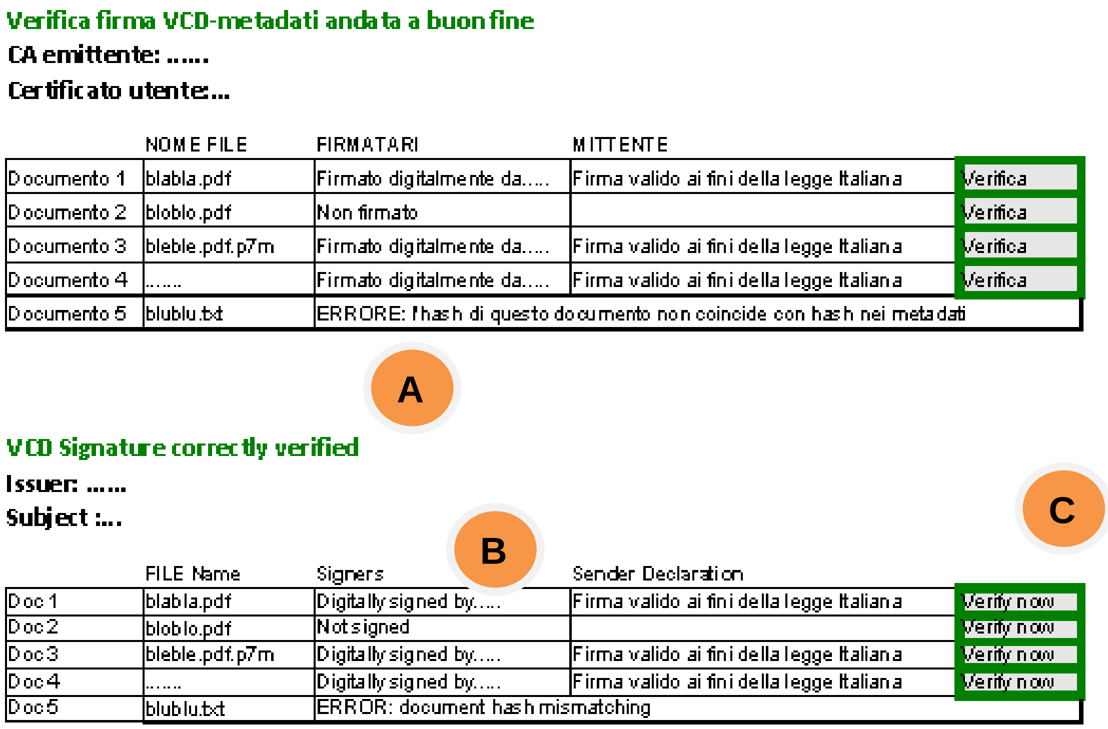
Verification results

The web interface will comply to the (W3C Web Accessibility) standards. The base application for showing the results of a VCD Verify will be set up by InfoCamere. The language set will be: Italian, English, German. Further localizations has be agreed.
The access to the Web application requires strong authentication to be performed by presenting an X.509 Certificate. The Certificate can be requested to the provider recommended by the Web Application though the point and click method. The collaboration established in Peppol WP1 can also identify other solutions to establish a specific trusted solution for accepting X.509 Certificates created by other Certification Authorities.

The resulting report will show:

- The result of the VCD verify;
- In case of positive verification. The list of referenced documents and the report of the Hash Check activity;
- Possibility to request individual verify on every single document.

An example of the web report is shown on the following screen mock-up.



Verifica firma VCD-metadati andata a buon fine

CA emittente: ……

Certificato utente:…

	NOME FILE	FIRMATARI	MITTENTE	
Documento 1	blabla.pdf	Firmato digitalmente da….	Firma valido ai fini della legge Italiana	Verifica
Documento 2	bloblo.pdf	Non firmato		Verifica
Documento 3	bleble.pdf.p7m	Firmato digitalmente da….	Firma valido ai fini della legge Italiana	Verifica
Documento 4	……	Firmato digitalmente da….	Firma valido ai fini della legge Italiana	Verifica
Documento 5	blublu.txt	ERRORE: l'hash di questo documento non coincide con hash nei metadati		

VCD Signature correctly verified

Issuer: ……

Subject :…

	FILE Name	Signers	Sender Declaration	
Doc1	blabla.pdf	Digitally signed by….	Firma valido ai fini della legge Italiana	Verify now
Doc2	bloblo.pdf	Not signed		Verify now
Doc3	bleble.pdf.p7m	Digitally signed by….	Firma valido ai fini della legge Italiana	Verify now
Doc4	……	Digitally signed by….	Firma valido ai fini della legge Italiana	Verify now
Doc5	blublu.txt	ERROR: document hash mismatching		

Figure 2-13: Results of a VerifyVCD operation

1.4.8 VCDTransportation

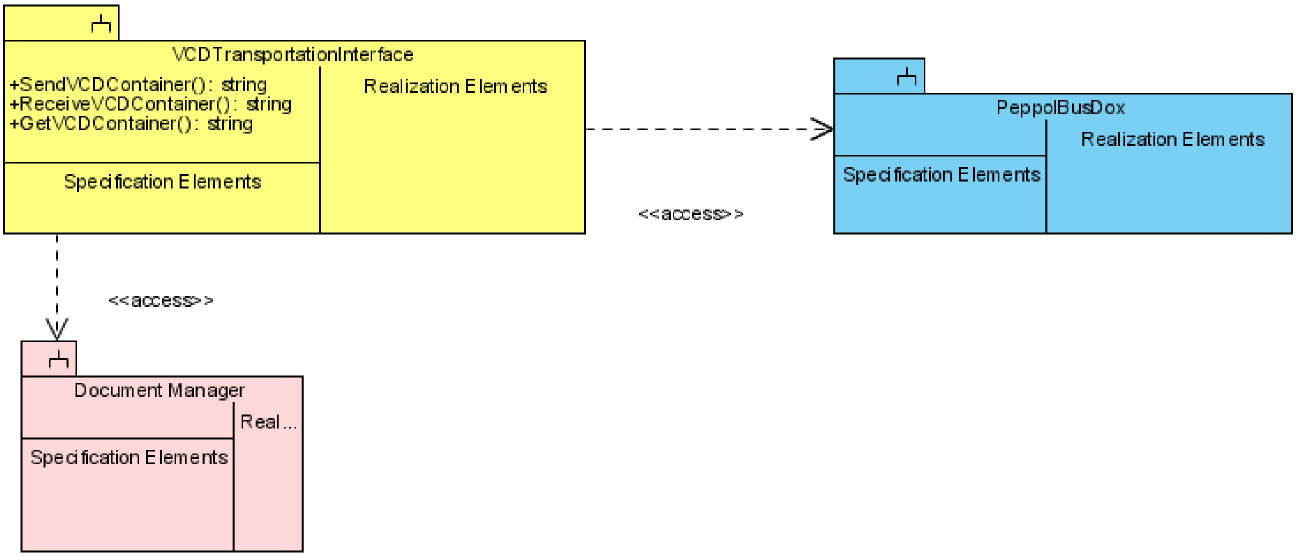


Figure 2-14: VCD Transportation

1.4.9 Strategy

InfoCamere is performing all the necessary evaluations for:

- The **setting up of a Client application** to handle the sending and receiving of VCD Containers across the BusDox infrastructure. (The Client Application will be implemented as **shared component** within Peppol)
- **Establishing agreements** with PEPPOL beneficiaries that can activate a BusDox AP and that are running an SMP.
- InfoCamere will establish the required agreements for connecting to the BusDox infrastructure during the POC pilot phase
- The first contacted BusDox operator is IntercentER (Peppol Beneficiary in Italy).
- Design will go in parallel with similar activity run by other VCD affected beneficiaries
- Development of the Client interface will follow the VCD development stages that are planned for the Pilot Testing phase

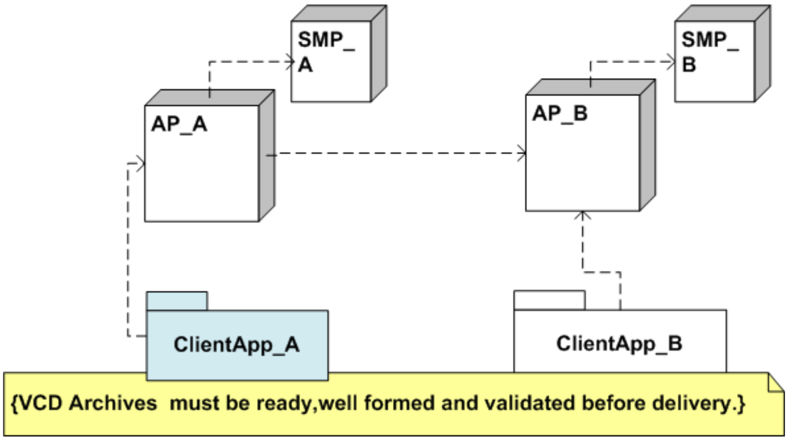

SMP_
A
AP_A
SMP_
B
AP_B
ClientApp_A
ClientApp_B
{VCD Archives must be ready,well formed and validated before delivery.}

Figure 2-15: BusDox interface for National VCD Systems

1.4.10 Send VCD Container to PEPPOL end point

Aspect	Description
Objective	A VCD service is able to send a VCD container to a PEPPOL end point user/system.
Results (postconditions in case of success)	The VCD container has been successfully handed over to the PEPPOL transport infrastructure.
Precondition	A VCD container has been created and is ready for delivery. The economic operator has provided the end point address information of the addressee.
Postcondition in case of failure	The VCD container has not been successfully handed over to the PEPPOL transport infrastructure.
Actor(s)	System
Initiating event	A VCD container has been created and is ready for delivery.
Description of interaction procedure with VCD Service (standard run)	The VCD system calls the PEPPOL infrastructure Client Application in order to CREATE a valid BusDox message. Concurrent parameters are passed (e.g. addressee). The VCD Container is linked to that message according to the procedure for the transmission of Binary objects. The Client Application validates the BusDox message and message is sent.
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	The Recipient (Peppol EndPoint) returns confirmation The System is capable to detect an incoming confirmation message

Table 2-5: Send a VCD container thru PEPPOL Busdox

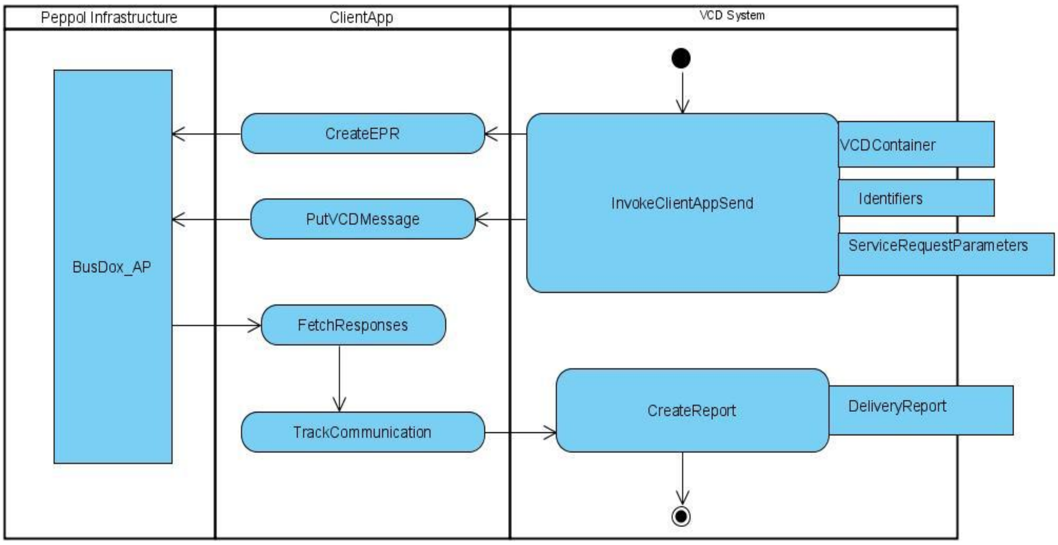


Figure 2-16: Activity diagram for the interaction among National VCD System and BusDox access point of WP 8 (sending VCD).

1.5 Receive VCD Container from PEPPOL end point

Aspect	Description
Objective	A VCD service is able to receive a VCD container from a PEPPOL end point user/system.
Results (postconditions in case of success)	The VCD container has been successfully fetched from the PEPPOL transport infrastructure. A confirmation (Acknowledge) message is returned
Precondition	A VCD container has been sent by a peer system to a valid recipient.
Postcondition in case of failure	The VCD container has not been successfully fetched by the Client Application over to the PEPPOL transport infrastructure (partially received) A negative confirmation (Negative-Acknowledge) message is returned.
Actor(s)	System
Initiating event	A VCD container has been sent by a peer system and is ready for retrieval.
Description of interaction procedure with VCD Service (standard run)	The VCD system calls the PEPPOL infrastructure Client Application in order to FETCH a valid BusDox message. (Browsing for available messages or similar operations have to be designed according to the BusDox Specifications). The VCD Container is taken from the transport message. The VCD Container is validated (optional operation)
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	VCD Container is validated

Table 2-6: Receive a VCD Container thru PEPPOL Busdox

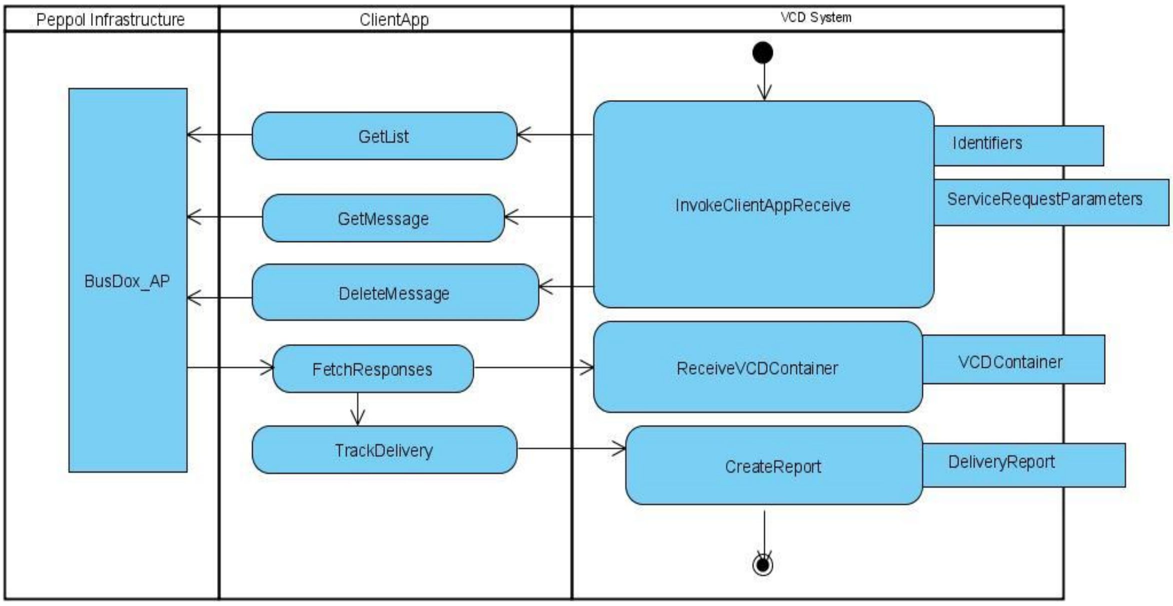


Figure 2-17: Activity diagram for the interaction among BusDox access point of WP 8 and a National VCD System (receiving VCD).

Index of Figures

Index of Tables

Abbreviations

PEPPOL Glossary of Terms and Abbreviations can be found in deliverable 7.3b.