

DELIVERABLE



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



Deliverable 2.3 Software building blocks enabling cross-border VCD



Revision: 1.0



Editors:
Maria A. 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
0.1	20101122	M. Wimmer	UKL	Initial version
0.2	20101210	M. Schett	PEPPOL.AT	Input from PEPPOL.AT
0.3	20101221	L. Leontaridis, B. Butteau	UPRC, ADETEF	Input from Greece and parts from France
0.4	20101222	D. Reiser	UKL	Input from UKL
0.5	20101223	P. Milani	InfoCamere	Input from IC (Italy)
0.6	20101228	W. Groiss, D. Ipsmiller	PEPPOL.AT	Input from PEPPOL.AT
0.7	20101231	M. Wimmer	UKL	Consolidating inputs, revising intro chapter, adding chapter 2 contributions, and restructuring some contents and reviewing document
0.8	20110104	S. Weiß, D. Ipsmiller, W. Groiß, M. Schett	PEPPOL.AT	Final Review from PEPPOL.AT with comments and questions added, short spec of Compound VCD added and short paragraph on harmonization of GUI (chapter 2.3 and 2.4).
0.9	20110106	A. Mondorf, M. Wimmer	UKL	Conclusions and revisions
1.0	20110106	M. Wimmer	UKL	Implementing final comments from Piero Milani and Ansgar Mondorf, and final revisions

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This deliverable contains original unpublished work except where clearly indicated otherwise.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



This deliverable is released under the terms of the **Creative Commons Licence** accessed through the following link: <http://creativecommons.org/licenses/by/3.0/>.

In short, it is free to

Share — to copy, distribute and transmit the work

Remix — to adapt the work

Under the following conditions

Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Contributions

Organisations

University of Koblenz-Landau (UKL), Germany

PEPPOL.AT, Austria

InfoCamere, Italy

UPRC, Greece

ADETEF, France

Persons

Bruno Boutteau, ADETEF, France (contributor)

Arianna Brutti, InfoCamere, Italy (contributor)

Christina Daskalaki, UPRC, Greece (contributor)

Jerry Dimitriou, UPRC, Greece (contributor)

Paola Fumiani, InfoCamere, Italy (contributor)

Wolfgang Groiss, PEPPOL.AT, Austria (contributor)

Doris Ipsmiller, PEPPOL.AT, Austria (contributor)

Lefteris Leontaridis, UPRC, Greece (contributor)

Flora Malamateniou, UPRC, Greece (contributor)

Giuliana Marzola, InfoCamere, Italy (contributor)

Piero Milani, InfoCamere, Italy (contributor)

Ansgar Mondorf, UKL, Germany (contributor)

Panagiotis Nicolaou, UPRC, Greece (contributor)

Andriana Prentza, UPRC, Greece (contributor)

Daniel Reiser, UKL, Germany (contributor)

Markus Schett, PEPPOL.AT, Austria (contributor)

Silke Weiß, PEPPOL.AT, Austria (contributor)

Maria A. Wimmer, UKL, Germany (editor, contributor)

Table of Contents

1	Introduction	7
1.1	Reading Instructions	7
1.2	Work Package 2 (Virtual Company Dossier) and Deliverable Objectives	7
1.3	Deliverable Summary and Structure	10
1.4	The PEPPOL Project	12
1.5	PEPPOL eProcurement Objectives	12
1.6	PEPPOL Interoperability Approach.....	13
1.7	PEPPOL Design Approach	15
2	VCD Specification Adjustments.....	19
2.1	Different stakeholder support levels of the VCD solutions	19
2.2	VCD Schema adjustments	22
2.2.2	VCD Skeleton types.....	23
2.2.3	Code lists	24
2.3	Initial process specification for a Compound VCD.....	24
2.3.1	Process and Nomenclature	24
2.3.2	Manual Process	25
3	VCD Software Artifacts.....	26
3.3	Reasoner Components	36
3.3.1	Purpose	36
3.3.2	Implementation environment.....	36
3.3.3	Deployment environment.....	36
3.3.4	Reference implementation	36
3.3.5	Outlook on functionality extensions	37
3.4	Ontology Schema Files	37
3.4.1	Purpose	37
3.4.2	Implementation Environment	37
3.4.3	Deployment Environment	37
3.4.4	Reference Implementation	38
3.4.5	Outlook on functionality extensions	38
3.5	Ontology Instance Files	38
3.5.1	Purpose	38
3.5.2	Implementation Environment	39
3.5.3	Deployment Environment	39
3.5.4	Reference Implementation	39
3.5.5	Outlook on functionality extensions	39
3.6	Packaging/Disassembly Components.....	39
3.6.1	Purpose	39
3.6.2	Implementation environment.....	40
3.6.3	Deployment environment.....	40
3.6.4	Reference implementation	40
3.6.5	Outlook on functionality extensions	40
3.7	EVS/NVS Service Interface (EVS side)	40
3.7.1	Purpose	40
3.7.2	Implementation environment.....	41
3.7.3	Deployment environment.....	41
3.7.4	Reference implementation	41

3.7.5 Outlook on functionality extensions	42
3.6 User Interaction Fragment Configurations.....	42
3.6.1 Purpose	42
3.6.2 Implementation Environment	42
3.6.3 Deployment Environment	42
3.6.4 Reference Implementation	43
3.6.5 Outlook on functionality extensions	43
3.7 Issuing Body Interfaces.....	43
3.7.1 Purpose	43
3.7.2 Implementation environment.....	43
3.7.3 Deployment environment.....	43
3.7.4 Reference implementation.....	44
3.7.5 Outlook on functionality extensions	44
3.8 Ontology Management System.....	44
3.8.1 GUI Description for Ontology Management System	44
3.8.1.1 Purpose	44
3.8.1.2 Implementation Environment	44
3.8.1.3 Deployment Environment	44
3.8.1.4 Reference Implementation	44
3.8.1.5 Outlook on functionality extensions	45
3.8.2 Ontology Export Queries.....	45
3.8.2.1 Purpose	45
3.8.2.2 Implementation Environment	45
3.8.2.3 Deployment Environment	45
3.8.2.4 Reference Implementation	45
3.8.2.5 Outlook on functionality extensions	45
3.9 m2n Intelligence Management Framework.....	45
3.9.1 Purpose	45
3.9.2 Implementation environment.....	46
3.9.3 Deployment environment.....	46
3.9.4 Reference implementation	46
3.10 EVS User access and role management incl. authentication	46
3.10.1 Purpose	46
3.10.2 Implementation environment.....	46
3.10.3 Deployment environment.....	46
3.10.4 Reference implementation	47
3.10.5 Outlook on functionality extensions	47
3.11 VCD Designer (incl. NVS-EVS Interface on the NVS-side).....	47
3.11.1 Purpose of the VCD Designer.....	47
3.11.2 Implementation environment.....	49
3.11.3 Deployment environment.....	49
3.11.4 Reference implementation	49
3.11.5 Outlook on functionality extensions	50
3.12 VCD Builder	50
3.12.1 Purpose of the VCD Builder	50
3.12.2 Implementation environment.....	51
3.12.3 Deployment environment.....	51
3.12.4 Reference implementation	52
3.12.5 Outlook on functionality extensions	52
3.13 VCD Viewer	53
3.1.1 Purpose of the VCD Viewer.....	53

3.1.2 Implementation environment.....	53
3.1.3 Deployment environment.....	54
3.1.4 Reference implementation.....	54
3.1.5 Outlook on functionality extensions.....	54
4 Software components planned in the next period to extend the VCD solutions.....	55
5 Summary and Outlook.....	59
Annex I: Overview of VCD Schema specifications in V 1.0.....	61

List of Figures

Figure 1: Overall sketch of the software building blocks of the PEPPOL VCD Full Support	9
Figure 2 Structure of PEPPOL VCD Deliverable 2.3	11
Figure 3: Structure of PEPPOL project	12
Figure 4 BII profiles relevant for PEPPOL BISs	15
Figure 5 PEPPOL Architectural Concepts.....	16
Figure 6: Main PEPPOL Schedule	16
Figure 7: Deliverable Development Approach.....	18
Figure 8: PEPPOL VCD solutions as presented in D 2.2	19
Figure 9: Stand-alone VCD Builder solution for European Countries, where the EVS contains the ontology, but there is no NVS in place	21
Figure 10: Manual VCD Editor for European Countries, where neither the EVS contains the ontology nor a NVS is in place.....	22
Figure 11: NVS/EVS interaction	23
Figure 12: Pre-VCD Skeleton types: T-, TC- and TCE-Skeleton	24
Figure 13: Components of the VCD Designer.....	47
Figure 14: Components of the VCD Builder.....	50
Figure 15: Components of the VCD Viewer	53

List of Tables

Table 1 - Main building blocks existing as per delivery of this deliverable.....	26
Table 2 - List of VCD common components	28
Table 3 - Building blocks planned for release by April 2011 or later (*).....	55
Table 4: Details of planned future components to extend the VCD solution	57

1 Introduction

1.1 Reading Instructions

This document includes the same common, generalized introduction as all other PEPPOL eProcurement deliverables so this introduction can be assumed for subsequent deliverable documents. This deliberate duplication allows each Deliverable to be read in isolation.

This common introduction includes a description of the approach adopted for the implementation of the eProcurement software building blocks according to the specifications in D 2.2.

1.2 Work Package 2 (Virtual Company Dossier) and Deliverable Objectives

In the overall PEPPOL context, the VCD addresses the pre-award phase and qualification of tenderers. The VCD specification and pilot implementation form a key building block for pan-European eProcurement through its uniquely described data to be exchanged between tenderers and Contracting Authorities across Member State countries.

The overall aim of PEPPOL Work Package 2 is

- a) to provide interoperable solutions for Economic Operators in any European country to utilise company information already registered somewhere, to assemble this company information into an electronic package and to submit this package electronically to any public sector awarding entity in Europe when these Economic Operators decide to apply for public contracts.
- b) for Contracting Authorities in any European country to (i) simplify and facilitate the submission of company information of tenderers through interoperable and easily understandable electronic criteria lists as well as to (ii) facilitate the evaluation of digitally submitted company information via appropriate tool support.

This way, both Economic Operators and Contracting Authorities can benefit from the PEPPOL Virtual Company Dossier (VCD) solutions. With the PEPPOL VCD solutions, especially SMEs are supported in better understanding criteria to evidence mappings across different national legal frameworks and in faster compilation of qualification documents for cross-border public procurement.

To achieve this goal, deliverable 2.1 (D2.1, resulting from phase 1 of the project) provided an analysis, synthesis and assessment of existing company dossier structures of individual Member State countries (AT, DE, IT, NO, etc.) and of other standard specifications. Therewith, it fulfilled the respective goal laid down in the technical annex. The deliverable also described the overall vision of the VCD, which was conceptualised throughout the first phase of work package 2. Besides that, the legal, organisational and technical requirements for the VCD solution to be implemented were elaborated.

In phase two of the WP 2 activities, the organisational, semantic and technical specifications of the overall VCD concept were elaborated. Deliverable 2.2 (Specification of architecture and components enabling cross-border VCD) based on the status quo report, the insights into existing company dossier structures, the VCD vision elaborated in phase 1 and the specific VCD requirements as defined in D2.1. The activities in phase two comprised (i) a consolidation of existing solutions with the PEPPOL needs, (ii) a formal technical specification of the VCD concept for pan-European eProcurement and (iii) organisational and legal specifications for the pan-European VCD implementation and application. The specification of the VCD was mutually agreed upon by the participating partner countries. It

therewith realized the second goal expressed in the technical annex of PEPPOL. The specifications of the VCD components form the interoperability architecture for cross border VCD, which addresses all interoperability layers in European Interoperability Framework (EIF) version 2.0.

Phase 3 of the PEPPOL VCD work focused on the implementation of the major software components to realize the VCD solutions and it continued advance the **VCD schema** specifications (cf. D 2.2). The VCD Schema specifications consist of standardised document and container schema specifications for VCDs, VCD packages and VCD containers that are used by the European VCD System and National VCD Systems at distinct VCD Container production stages. The VCD schema provides the core interoperability specification for PEPPOL's VCD solutions.

The major software components (building blocks) elaborated in phase 3 of WP 2 comprise of (see Figure 1 for an overall architecture for the PEPPOL VCD Full Support and D 2.2 for more detailed descriptions of the specifications):

- (1) the **European VCD System (EVS)** is a central component, which serves as decision support system that provides the information of which evidences are to be delivered by an Economic Operator or a Consortium in order to fulfill non exclusion and qualification criteria even in cross-border tendering processes. It is used by National VCD Systems as an embedded service as well as by Economic Operators and Contracting Authorities via a user interface. The rule set is formulated as a modular ontology representing the different implementations of the respective EU directive in each country's national public procurement acts. Besides that, complexity aspects within the different legal frameworks such as specific tenderer constellations and legal forms of Economic Operators are considered in the reasoning. As a central component, the EVS is generating a VCD Skeleton Container, i.e. it sets out the basic structure for the VCD artefact. This way, it fulfills stage 1 of the overall VCD vision described in D 2.1. The European VCD System can either be approached directly by the user (Economic Operator / Contracting Authority) or can act as an embedded service for National VCD Systems via a WSDL interface. In the context of PEPPOL's project scope, Economic Operators from the following countries can benefit from this solution: Austria, France, Germany, Greece, Italy. Further European countries may benefit from this sophisticated solution and participate easily by realising their public procurement acts in the EVS via a national ontology. An intermediate solution to generate a VCD skeleton container is described below (via the Manual VCD Skeleton Editor). This component will be offered to Economic Operators from those European countries, where the public procurement act is not yet implemented in the EVS via a national ontology. The ontology as the central rule set is managed and maintained via the **VCD Ontology Management System** decentrally by the legal teams of the respective participating countries.
- (2) the **National VCD System (NVS)** provides the compilation service of the VCD Container for the Economic Operator. It consists of the PEPPOL reference implementations VCD Designer (used for the interaction with the EVS) and VCD Builder (the compilation service) – both components will be further described in chapter 3 – or it may be built on a country's own infrastructure. It may directly interact with the EVS through an EVS-NVS interface in case the respective national ontology is available (see (1) above). The NVS may be deployed in different variants (cf. specifications of D 2.2. and scenario descriptions on pp. 17-18 of D 2.2):
 - a) *as one unique national central service in a European country.* In this solution, the NVS may directly interact with external issuing bodies (e.g. via system interfaces) to collect relevant evidences on behalf of the Economic Operator – though this simplification feature depends on the deployment and implementation in a country. A National VCD System exists at this stage in Austria (which is built upon a proprietary solution and which offers direct interfaces to currently two external issuing bodies). Greece and Italy will implement the NVS in the coming months using the VCD Designer and VCD Builder reference implementations of PEPPOL.

- b) as a service offered e.g. by different eTendering platform providers or prequalification bodies. Also in this solution, the VCD Designer and VCD Builder are provided as the reference implementation. Whether direct interaction with external issuing bodies can be integrated in the solutions depends on the countries' contexts and the position of such an actor in the respective business models. France has chosen this option, i.e. it will integrate the VCD Designer and VCD Builder into different eTendering platforms and Tendering Service Providers (such as Achat Public) or like a SAAS (Software as a service such as DILA offers) possibly called by existing eTendering platforms (such as SAE/PMI at www.marches-publics.gouv.fr).
- (3) as a **Manual VCD Builder**, which any Economic Operator across Europe can use. It is especially targeted to Economic Operators in countries, where no other solution is offered (neither a) nor b)) and it consists of the VCD Builder and a Manual VCD Skeleton Editor. The latter is used to generate the initial structure of a VCD container artefact, i.e. it serves as intermediate solution until the country has implemented the national procurement laws in the EVS (see (1) above). The Economic Operator in such countries can consult eCertis to get an understanding of which evidences in his or her country correspond to the selection and exclusion criteria set out in the call for tender or contract notice of a Contracting Authority in a different European country.
In the European countries, where national solutions or solutions integrated into eTendering platforms or provided by prequalification bodies are offered, it is recommended that these solutions are used by Economic Operators to generate VCD containers as a higher level of content quality can be provided.
- (4) the **VCD Viewer** is a component to view and navigate through the content of any VCD Container without having the possibility to edit or change content. This component is dedicated to Contracting Authorities primarily. It can also be used by Economic Operators to view content that needs not to be changed any more. In the other cases, Economic Operators use the solutions described in (2).

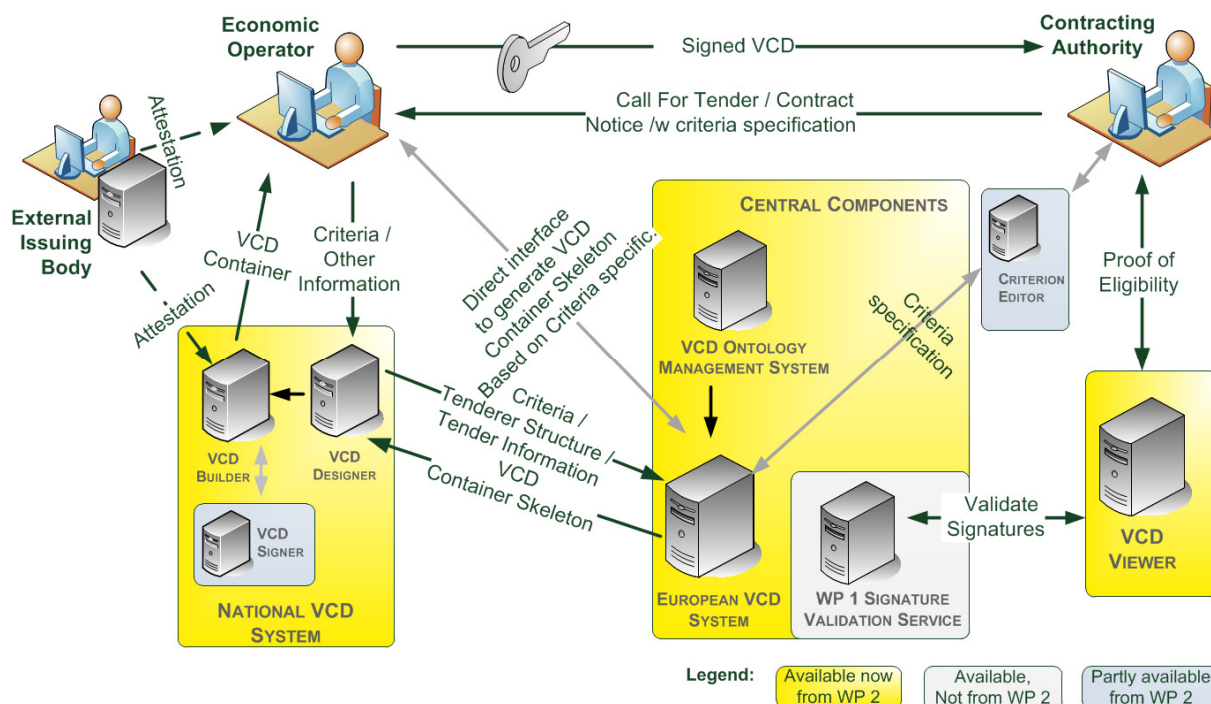


Figure 1: Overall sketch of the software building blocks of the PEPPOL VCD Full Support

This document represents Deliverable 2.3 (D 2.3: Software building blocks enabling cross-border VCD) of the PEPPOL project (Pan-European Public Procurement OnLine). The aim of this deliverable is to support the implementation of the PEPPOL VCD pilots by providing building blocks as shown in Figure 1, which are needed for

1. connecting to the European VCD System and

2. to use and deploy VCD solutions across Europe.

These building blocks are dedicated to four different target groups:

- a) Contracting Authorities (using the VCD viewer and recommending the VCD solution in public tenders),
- b) National VCD System providers,
- c) eTendering platform providers and prequalification bodies or other tendering solutions providers, who may want to integrate the VCD solutions in their existing tendering environments, and
- d) Economic Operators (using the VCD solutions in public tenders either offered by target groups mentioned in b) or c), or - if not offered in their country - by using a manual VCD solution offered by PEPPOL – see more details in NVS description above).

In particular PEPPOL has focused its software building blocks on enabling small and medium sized enterprises (SMEs), who represent the audience with the greatest potential benefit from enabling cross-border eProcurement.

Implementation of the specifications, in conjunction with the PEPPOL Infrastructure (Deliverables 1.2 and 8.2), addresses the organizational, semantic and technical interoperability layers of the European Interoperability Framework (EIF) version 2.0¹.

This deliverable, which focuses on pre-award building blocks, is aligned with other PEPPOL deliverables (which are mostly in post-award; these are D 3.3 eCatalogue building blocks, D 4.3 eOrdering building blocks and D 5.4 eInvoicing building blocks) to enable consistent implementation of eProcurement processes.

The pre-award building blocks were prepared by PEPPOL Work Package 2 as an outcome from PEPPOL's Specification Phase for VCD (D 2.2). It is anticipated that updates will be required during the Production Pilot phase of the PEPPOL project as part of PEPPOL's overall support and governance policy.

1.3 Deliverable Summary and Structure

In the first phase of the Pan European Public Procurement On Line (PEPPOL) project, Work Package 2 (WP 2) focused on the identification of the challenges for interoperability of Virtual Company Dossier in cross border public procurement in the pre-awards phase. These results, reported in Deliverable 2.1, represented the starting point for the second phase of PEPPOL WP 2. The result of the second phase was the provision of specifications enabling cross border VCD. Therby the specifications were addressed along two work streams:

1. Specifications:

Developing technical specifications of the architecture and the building blocks that are necessary to implement the high level scenarios outlined in Deliverable 2.1. The specifications (Deliverable 2.2) have been implemented by PEPPOL WP 2 partners in phase 3 of PEPPOL WP 2 to create suitable implementations for pilot operations of the VCD.

2. Tool Development:

¹ <http://ec.europa.eu/idabc/servlets/Doc?id=31597>

Developing software tools (based on the specifications) suitable for Contracting Authorities and Economic Operators (especially SMEs) to facilitate the connection and interoperability of their business processes. Some reference implementations have already been provided together with Deliverable 2.2. Most reference implementations are, however, targeted in phase 3 of PEPPOL WP 2 and are documented in this deliverable.

Therefore, this document (D 2.3) provides the results of phase 3, which confer mostly to the second work stream (Tool Development) as well as minor revisions to the first work stream (Specification).

All PEPPOL deliverables follow a similar overall structure. The main document of Deliverable D 2.3 will describe the software building blocks and provide revisions to the specifications as provided in D 2.2. Figure 2 provides an overview of the building blocks indicated in Figure 1 and to be described in the report at hand.

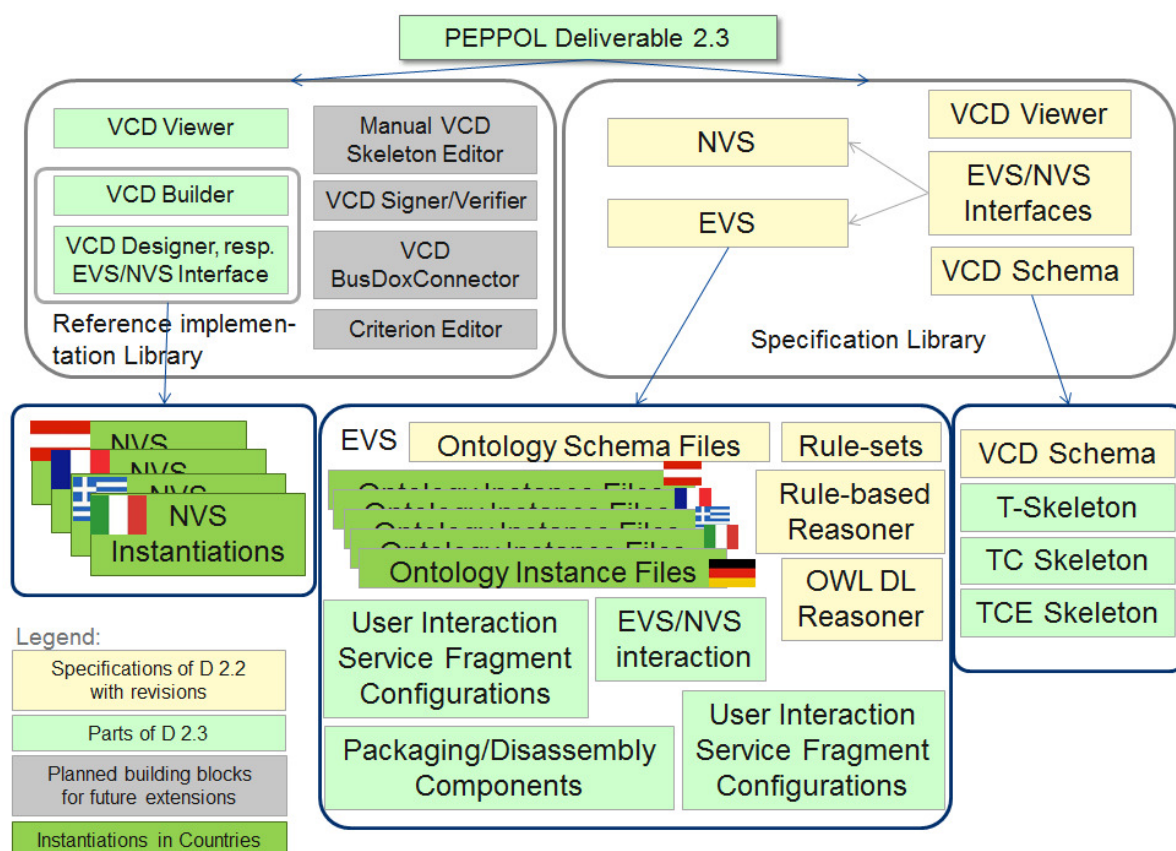


Figure 2 Structure of PEPPOL VCD Deliverable 2.3

The Deliverable 2.3 provides in chapter a summary of the revisions to specifications of components provided in the previous Deliverable D 2.2.

Chapter 3 provides an overview of the relevant software artifacts that support PEPPOL implementors in the integration of the software building blocks and end users in the usage of the provided software. It also provides brief descriptions of the WP 2 building blocks delivered together with this deliverable.

Chapter 4 gives an overview of future implementations of software components to enrich the VCD solution architecture.

Chapter 5 concludes the deliverable with a brief resumé of what has been achieved and which work lays ahead to improve the VCD solution in the next releases.

1.4 The PEPPOL Project

PEPPOL (Pan European Public Procurement On Line) is a 42 month (May 1st 2008 – October 31st 2011, with an intended prolongation till April 30th 2012, which is pending EC approval) pilot project under the European Commission's CIP (Competitiveness and Innovation Programme) initiative.

The project aims to align business processes for eProcurement across all Government Agencies within Europe. The vision is that any company and in particular small and medium-sized enterprises (SMEs) in the EU can communicate electronically with any European governmental institution for the entire procurement process.

On May 1st 2010, following a specification phase and a development phase, PEPPOL entered its test pilot phase and from November 1st 2010 will be supporting production pilots.

The PEPPOL consortium comprises of the leading public eProcurement agencies in Austria, Denmark, Finland, France, Germany, Italy, Norway and Hungary. These have recently been joined by agencies from Greece, Portugal, the UK (Scotland) and Sweden.

The scope and structure of the PEPPOL project is shown in Figure 2. In addition to the work packages shown, WP 6 provides project administration and WP 7 supports awareness, training and consensus building.

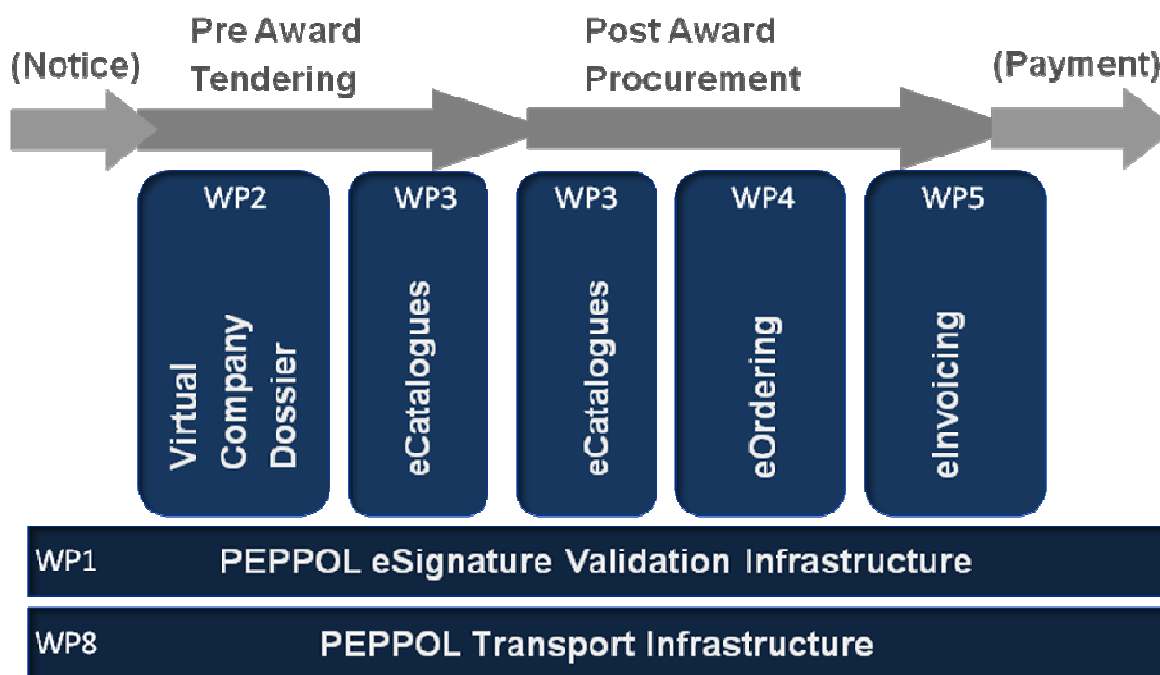


Figure 3: Structure of PEPPOL project

For an introduction to PEPPOL and the setup of the project see www.peppol.eu.

1.5 PEPPOL eProcurement Objectives

The broader vision for PEPPOL is that any company (incl. SMEs) in the EU can communicate electronically with any EU governmental institution for all procurement processes.

The objectives for eProcurement are set by PEPPOL stakeholders. These include:

- Project owners: The sponsors of PEPPOL i.e. EU commission and the beneficiary Member States.
- WP participants: Member States participating in specific PEPPOL work packages.

- Non-beneficiary Member States: stakeholders that gain benefits from the pilot i.e. EU member countries not participating in PEPPOL.

The project owner objectives can be deduced from the I2010² strategy, CIP ICT PSP³ project call and country specific reasons for joining the project. Collectively this can be viewed as supporting a single European market, competitiveness and innovation by

- Removing barriers for cross-border eProcurement
- Learning through implementation and operation of eProcurement pilot systems
- Raising awareness of eProcurement benefits through a pilot.

The PEPPOL eProcurement Community consists of parties who share the common objectives of PEPPOL and have implemented one or more components of the PEPPOL Reference Architecture. Deliverable 7.4 describes the benefits in entering the PEPPOL community.

PEPPOL has adopted a broad definition for cross-border eProcurement. In a typical case a Contracting Authority and an Economic Operator (who may be an SME) are situated in different Member States. However, there are also cases where an eProcurement platform is operated in a country different from either the Contracting Authority or the Economic Operator. In the scope of PEPPOL these are also considered as cases when the “cross-border” characteristic can be a barrier to interoperability.

Pilot participant objectives can be deduced from country specific reasons for participating in the project:

- Leveraging existing solutions to handle cross-border eProcurement
- Create traction on interoperability model, thereby securing the investment in the chosen eProcurement interoperability model
- Influence on standardization activities in such a way that they meet the requirements of the participant.

There is a strong desire by both Contracting Authorities and Economic Operators for automation and efficiency across the procurement process. This requires good interoperability - that is a common information and process model ensuring a mutual understanding of the flow of information between different parts of the process. As mentioned above these interoperability requirements have been analyzed according to the European Interoperability Framework.

Non-beneficiary Member State objectives can be deduced from country specific reasons for joining the reference group, for example:

- Leveraging and building upon the experience of the PEPPOL eProcurement project.
- Cost saving by adopting a proven eProcurement interoperability model

Two separate outcomes for PEPPOL deliverables have been identified:

- Interconnecting the eProcurement platforms of Contracting Authorities in participating countries for engaging Economic Operators in other countries.
- Making available open source software together with tools to deal with eProcurement both for Contracting Authorities and Economic Operators (especially SME's).

PEPPOL has built upon existing work in these areas and continues cooperation with current initiatives.

1.6 PEPPOL Interoperability Approach

A PEPPOL BIS (Business Interoperability Specifications) defines, in detail, the requirements and specifications on all interoperability layers to a specific part of the PEPPOL eProcurement process.

² http://ec.europa.eu/information_society/eeurope/i2010/strategy/index_en.htm

³ http://ec.europa.eu/information_society/activities/ict_psp/index_en.htm

The European Interoperability Framework's⁴ (EIF) goal is:

- To serve as the basis for European seamless interoperability in public services delivery, thereby providing better public services at EU level.
- To support the delivery of pan-European eGovernment Services (PEGS) by furthering cross-border and cross-sector interoperability.
- To supplement the various National Interoperability Frameworks in the pan-European dimension.

Version 2.0 of the EIF defines these interoperability layers as Political, Legal, Organizational, Semantic and Technical interoperability.

The CEN/ISSS BII (Business Interoperability Interfaces) Workshop Agreement⁵ defines common requirements for business interoperability interfaces for European public eProcurement. These enable implementers of BII (such as PEPPOL) to:

- Identify and document the required business interoperability interfaces related to pan-European electronic transactions in public procurement expressed as a set of technical specifications, developed by taking due account of current and emerging international standards in order to ensure global interoperability.
- Co-ordinate and provide support to pilot projects implementing the technical specifications in order to remove technical barriers preventing interoperability.
- To facilitate implementation of electronic commerce in a standardized way, thereby enabling the development of standardized software solutions as well as efficient connections between trading partners without case by case specification of the data interchange.
- Document required Business Interoperability Interfaces (BII) as formal profile descriptions.
- Reduce the cost of implementing electronic commerce to a level that is economical for small and medium size companies (SMEs) and institutions.

A CEN ISSS WS/BII profile description is a technical specification describing business processes. It includes a detailed description of the way trading partners intend to play their respective roles, establish business relations and share responsibilities to interact efficiently with the support of their respective information systems i.e.

- the business rules governing the execution of that business process.
- possible run-time scenarios and the business commitments achieved.
- the electronic messages exchanged as part of the business process.
- the sequence in which these documents are exchanged.
- the information content of the electronic messages exchanged.

As well as determining which documents are used, a profile restricts document content in terms of elements, their content and cardinality. The key aspect of the BII profile description is thus in the standardized semantics rather than the syntax. Consequently the messages within a profile can be structured based on different message standards/syntax as long as the chosen standard contains all the necessary data elements.

Figure 3 provides an overview of the BII profiles relevant for PEPPOL's Business Interoperability Specifications (BIS).

⁴ <http://ec.europa.eu/idabc/servlets/Doc?id=31597>

⁵ CEN WORKSHOP AGREEMENT CWA 16073-1 see: <http://www.cen.eu/cwa/bii/specs/>

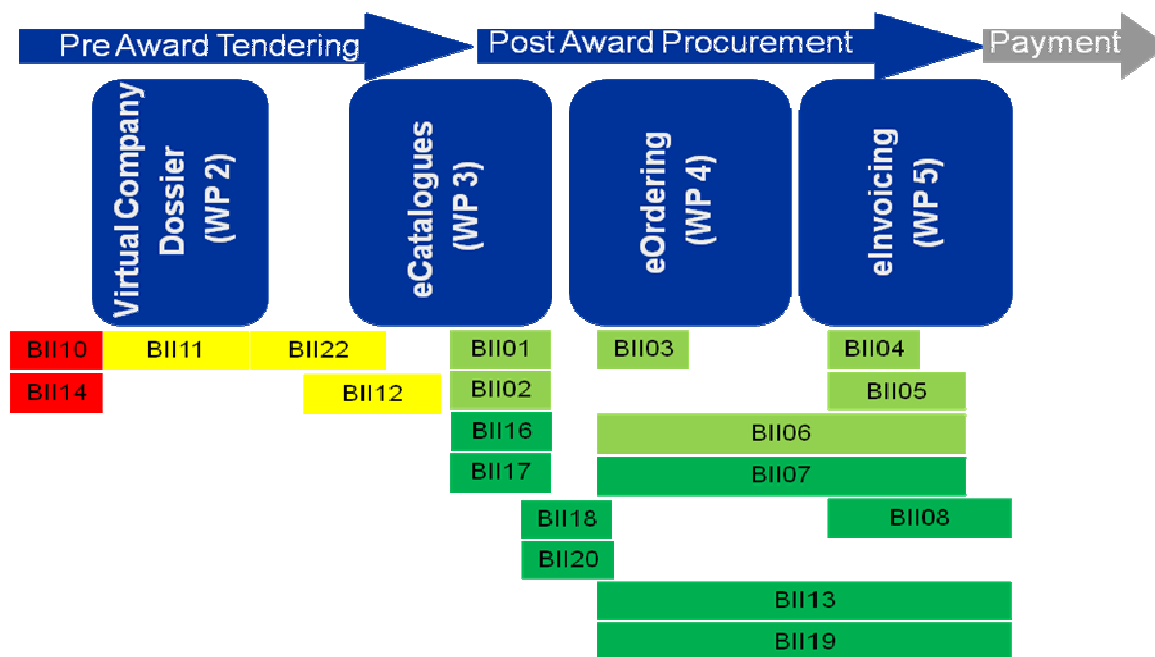


Figure 4 BII profiles relevant for PEPPOL BISs

The newly formed CEN BII2 Workshop⁶ and its timeframe of 3 years make it the perfect partner for ongoing governance and maintenance of these BII specifications. In addition, PEPPOL will work with the CEN BII2 Workshop (and other related CEN Workshops) to develop a common architecture and validation artefacts to support the profiles used.

1.7 PEPPOL Design Approach

This section introduces the design approach followed when developing the PEPPOL pre-award eProcurement specifications.

The architectural concepts adopted by PEPPOL are outlined in Figure 4. This describes the terms used and relationships between them and so provides a framework for the approach taken when developing the PEPPOL building blocks for the VCD.

An overall PEPPOL Reference Architecture has several options for implementation (we call these implementation models). Every PEPPOL beneficiary has chosen to implement their architecture based on one of these models. Each component in the architecture is implemented using building blocks, which may (or may not) be those provided in open source form by PEPPOL. Equally some of the building blocks created by beneficiaries may become PEPPOL Reference Implementations.

⁶ http://www.cen.eu/CEN/sectors/sectors/iss/activity/Pages/ws_bii.aspx

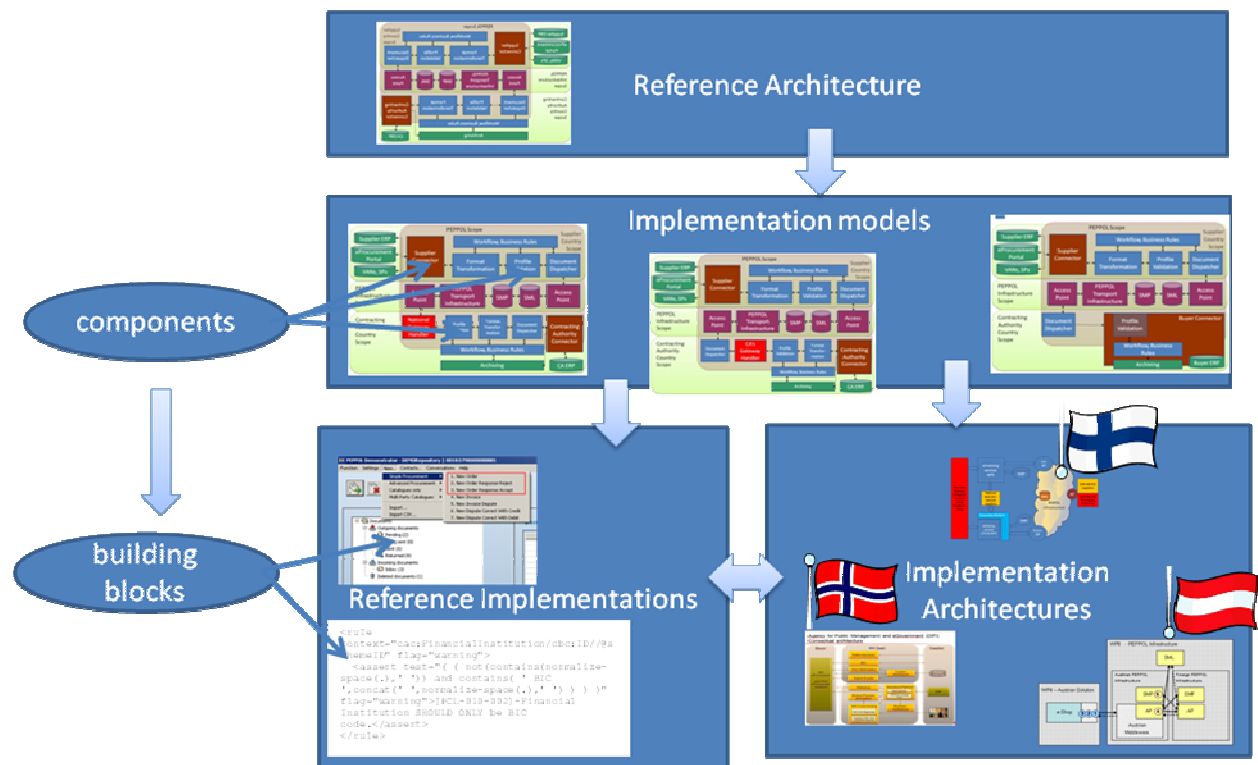


Figure 5 PEPPOL Architectural Concepts

In terms of this document, the Reference Architecture and options for Implementation Models are explained in this section (Chapter 1. Introduction). Reference Implementations and the Implementation Architectures adopted by PEPPOL beneficiaries are described in Deliverable 2.2.

Figure 6 describes how the PEPPOL project is divided into three phases. Each work package follows the processes specified in PEPPOL using a SCRUM⁷ based approach.

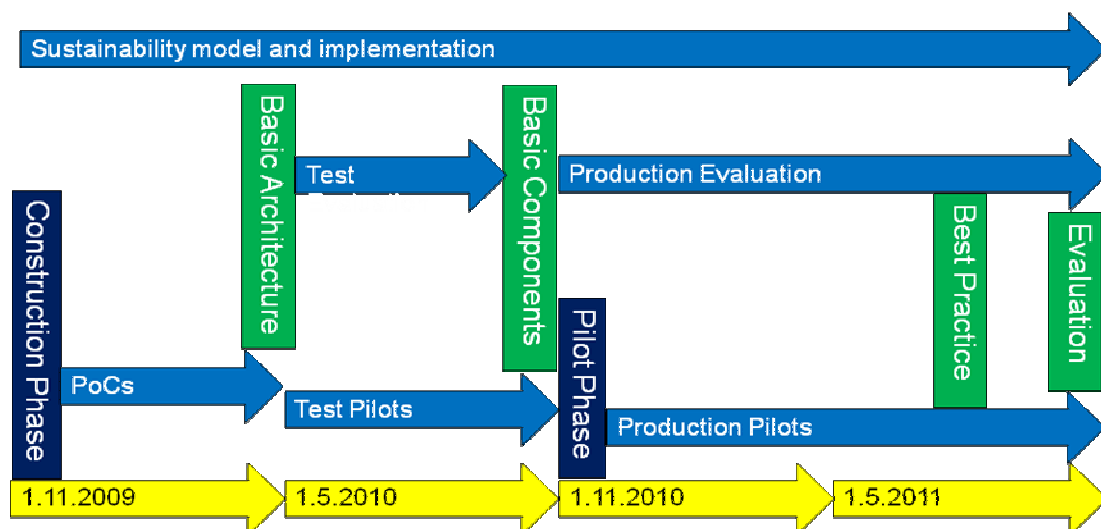


Figure 6: Main PEPPOL Schedule

⁷ Schwaber, Ken (1 February 2004). *Agile Project Management with Scrum*. Microsoft Press

A SCRUM based approach creates a fast, flexible and agile way of creating specifications and enabling software building blocks through each phase of the project:

- **Proof of Concept Pilot (PoC)** – Simulated operations (artificial participants, artificial data in a closed environment). This phase was mainly for addressing various eProcurement issues on one or more of the interoperability levels of the VCD solutions. The results are documented in D 2.2 as specifications for the VCD solutions.
- **Test Pilots** – Involve actual cases of submitted qualification documents in e-tenders and test the VCD pilot solutions to generate the same results as the tender would have requested. This phase is performed with real test data but it is not operational with real business transactions.
- **Production Pilots** – Involve actual Contracting Authorities and Economic Operators using their actual business applications and actual data (i.e. true operational business transactions). This phase is currently prepared. It is to be mentioned that the VCD pilots are different from post-award pilots as in pre-award, the Economic Operators cannot be known in advance by the Contracting Authorities.

Along the SCRUM method to develop the specifications of the VCD, WP 2 divided its work into six task force groups:

- **Pre-VCD Mapping task force:** This task force took care of the technical specifications of the mapping resulting in the ontology specifications and modelling and the specifications for the European VCD System. A set of workshops was carried out in order to build the ontology on top of national inputs.
- **VCD Pilot General task force:** This task force took care of the specification and architecture concept for the national VCD systems, including the specification of the common components. The work resulted in a set of use case descriptions for the functions that national VCD systems shall fulfill. Also implementation plans for PEPPOL Beneficiaries to enable Cross-Border Virtual Company Dossier where carried out in this task force (instantiations of the reference NVS implementation).
- **VCD Schema task force:** This task force elaborated the VCD schema specifications thereby trying to align already as much as possible with CEN BII and UBL in order to pave the way for standardising the VCD schema specifications. The major work in the task force was to define the VCD data model including a syntax binding to UBL and the definition of VCD specific code lists, identifiers and VCD container format. The task force is also concerned with XML validation in order to ensure interoperability of the components developed.
- **Legal and Organisational task force:** This task force investigated legal aspects of the VCD concepts. Foremost, this group took care of mapping the national criteria and evidences with the European criteria of the EC directive via so-called “mapping tables”. The mapping tables were transformed into the ontology representation of the European VCD System within a set of workshops that have been carried out with the partner countries involved and the pre-VCD mapping task force. This task force has also been responsible for the development of process models, evaluation guidelines, production pilot preparation, policies and the development of Governance and sustainability plans with regard to the VCD implementations.
- **Shared Components task force:** This task force is responsible to provide the development environment and to coordinate the development of the common components of WP 2 in particular the VCD builder as the core component for the NVS and the VCD viewer as well as the integration with other WP 2 and cross WP components such as the European VCD System or Signature Validation Service (WP 1).
- **Pilot Recruitment task force:** This task force has been installed in autumn 2010 and is responsible to ensure pilot cases from each WP 2 pilot partner in the VCD context. As piloting pre-awards solutions is quite different from post-awards cases (no buyer-supplier pairs, strict legal compliance with tendering procedures, neutrality and not necessarily knowing all potential tenderers in a public tender), this task force first of all addresses Contracting Authorities in order to convince them to participate with a public tender in the near future

(within the next 6 months). Participating with a public tender means that the Contracting Authority will recommend in the call for tender or contract notice the use of the VCD solution to the Economic Operators and it will set the legal grounds that the VCD artefacts delivered by the tenderers will be accepted by the Contracting Authority as the qualification documents to be submitted. This also includes accepting on the side of the Economic Operator that in case problems with the VCD generation emerge and incomplete VCD artefacts are submitted, this shall not exclude a tenderer (i.e. an Economic Operator). This task force will in future work closely with work package 7 on pilot recruitment.

New specifications and software based on new requirements emerging from the Test Pilot and Production Pilot may be produced. These new specifications and software will go through the same cycle of development: PoC Pilot, Test Pilot and Production Pilot within dedicated task forces.

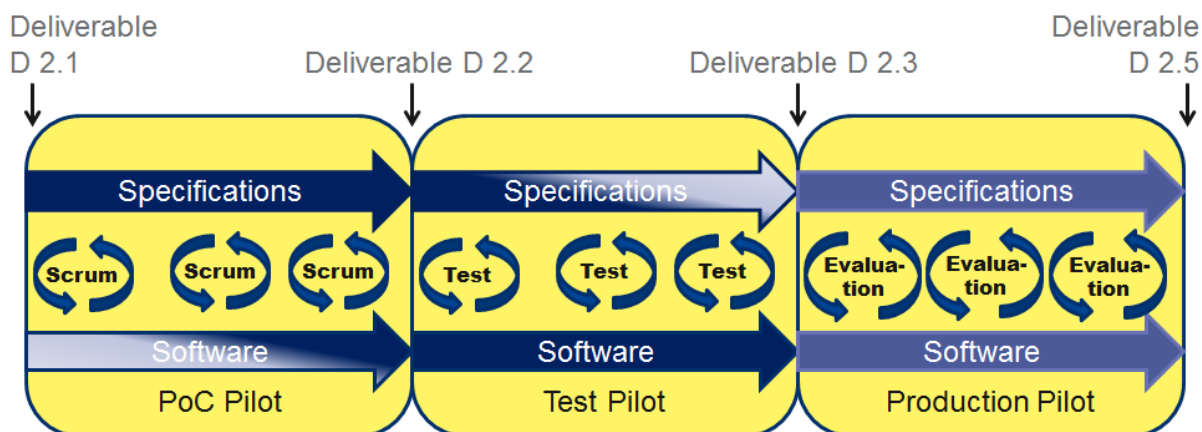


Figure 7: Deliverable Development Approach

As depicted in Figure 7 the adopted SCRUM based methodology together with the overall planning of the project, creates a hybrid, where specifications and software is created in parallel in the PoC phase. This continues into the Test Pilot, where both Software and Specifications are matured towards production. In the Production Pilot, the Specifications and Software is further matured to be scalable and pan-European. The consequence of this approach is that both Specifications and Software will evolve and new releases will be published throughout the project period.

2 VCD Specification Adjustments

2.1 Different stakeholder support levels of the VCD solutions

In Deliverable D 2.2, the overall scope of PEPPOL VCD solution has been presented as shown in Figure 8. In the specifications it was already foreseen to provide different levels of stakeholder support of the VCD solution. The different scenarios have been described in D 2.2 on pages 13 – 14.

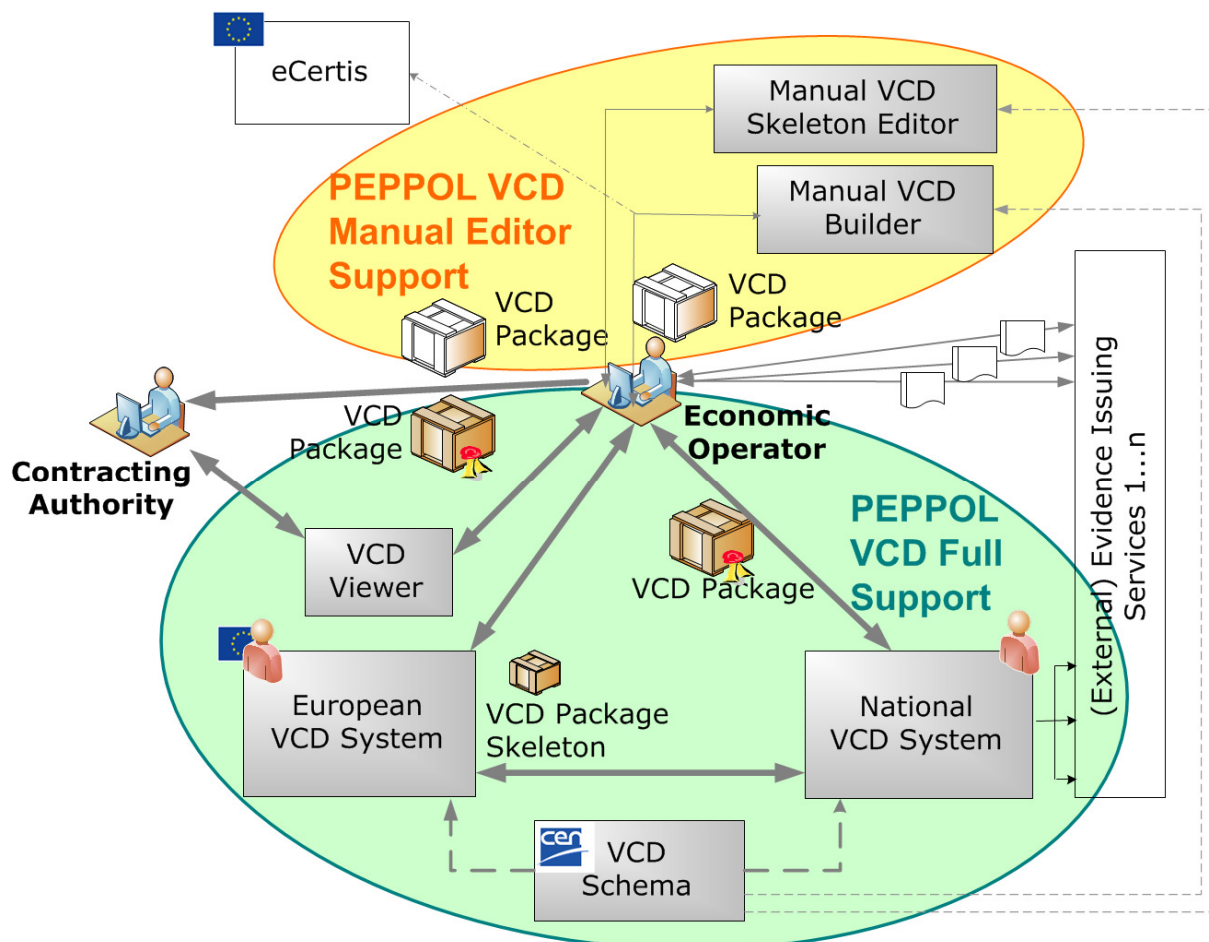


Figure 8: PEPPOL VCD solutions as presented in D 2.2

Throughout the test pilot phase, a few specifications and formal representations have been updated and slightly changed. In the following, the main modifications are summarised. More detailed updates of specifications will be provided along the planned regular updates of D 2.2, the next release being scheduled for end of April 2011:

- The ontologies for Austria, France, Germany and Italy have been updated and revised. Furthermore, the Greek ontology has been added. In consequence, the European VCD System fully supports the decision process in these countries, which simplifies the work of Economic Operators as well as Contracting Authorities in understanding, which evidence existing in the Economic Operator's country meets best the requested criterion set out by the Contracting Authority (in a different country).
- For the countries not having ontologies represented in the EVS, eCertis as well as other sources serve as information basis for the Economic Operator in order to perform this criteria to evidence mapping across borders manually via the Manual VCD Skeleton Editor.

- The VCD schema specification has been refined to support different levels of completion of a VCD artefact, for example used by the EVS-NVS interface. New schemas have been defined as subsets of the full VCD schema specified in D2.2 as follows:
 - T-Skeleton: This schema contains an interoperable standard structure to communicate a tenderer structure (single tenderer, bidding consortium, subcontractors; all with legal form of individual Economic Operators). This schema structures the data passed from the NVS to the EVS when initializing the decision support process.
 - TC-Skeleton: This schema extends the T-Skeleton with the criteria laid out in a call for tender or contract notice by the Contracting Authority. The specific skeleton maps out the exact criteria to be applied to each single Economic Operator. This schema structures the data passed from the EVS to the NVS when suggesting criteria for a specific tenderer setting and the data passed from the NVS to the EVS as input for suggesting possible evidences.
 - TCE-Skeleton: This schema represents the full structure of a VCD artefact generated by the European VCD System and passed either to the NVS or to the Economic Operator. It indicates for each Economic Operator and selected criterion, which evidences the party might provide to prove that selection or exclusion criterion based on the party's role in the tenderer structure and its legal form. This schema structures the data passed from the EVS to the NVS when suggesting possible evidences.
- The National VCD System (NVS) specification as conceptualized in D 2.2 accomodates the implementation of national solutions, which may use components of a reference implementation as well as background systems and are therefore best fitting to the specific national needs and preconditions. Throughout the PEPPOL WP 2 phase 3, NVS open source reference implementations have been programmed, which consist of the VCD Builder and the VCD Designer. Throughout the test pilot phase, piloting partners have evaluated the possibilities for instantiations in their countries. As already indicated in D 2.2, different solutions will be realized:
 - **Austria** has implemented a full National VCD System, based on the same backend infrastructure as the European VCD System, making use of the nationally provided background system in Austria. This National NVS System also implements interfaces to Austrian issuing services.
 - **France, Greece and Italy** will pick up the NVS reference implementations and will set up VCD solutions of different kinds. While Greece will implement a national VCD solution, Italy will offer the VCD service via InfoCamere. France will integrate the VCD Builder solution into eTendering platforms and offer it as a SAAS (software as a service) that can be called from existing tendering solutions.
 - New European countries may deploy the NVS reference implementations according to their specific needs with the following recommended options:
 - As unique national VCD service deploying the reference implementation of the NVS (such as Greece is planning)
 - As VCD solution integrated in eTendering platforms deploying the reference implementation of the NVS reference implementation (such as France is planning).
 - Prequalification bodies may benefit from the NVS reference implementations and upgrade their existing business solutions to provide pan-European interoperable VCD packages. Prequalification bodies are especially targeted to implement stage 4 of the VCD vision (see Deliverable D 2.1) with the VCD

solution. This way, they can implement in a simple way article 52 of the EC eProcurement directive.

- As decentralised VCD service offered e.g. by IT-service providers or as SAAS solution as e.g. France is planning.
- For Economic Operators from countries, where no VCD solution as mentioned above exists yet in their country, two scenarios are foreseen:
 - In case the ontology is in the EVS (**Germany** at the moment), the NVS reference implementations can be used as a stand-alone solution. In this case, The Economic Operator either directly consults the EVS or also deploys the VCD Designer through which he/she can interact with the EVS to generate the VCD Container Skeleton. Subsequently, the VCD Builder component is used to compile the VCD Container. As there is no direct interface with External Issuing Bodies, the Economic Operator will have to handle the VCD compilation fully manually through the Stand-alone VCD System. Figure 9 shows the slightly modified overall picture for this scenario.

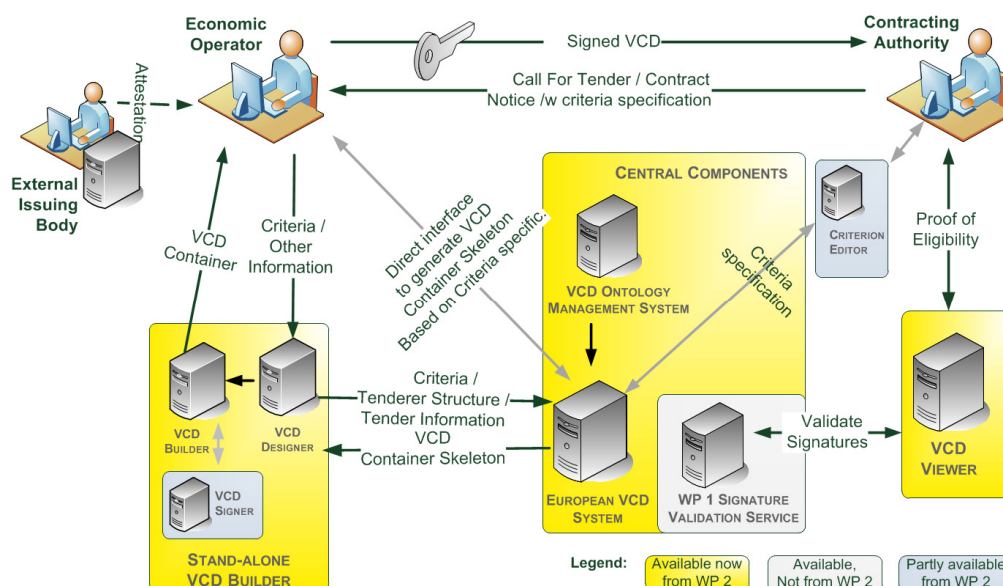


Figure 9: Stand-alone VCD Builder solution for European Countries, where the EVS contains the ontology, but there is no NVS in place

- The least support PEPPOL WP 2 can provide to Economic Operators regards countries, where neither a VCD solution as mentioned above exists nor the ontology is in the EVS (currently all European countries, except Austria, France, Germany, Greece, Italy). In this case, the Economic Operator will manually compile the VCD container through the stand-alone Manual VCD Builder. The Economic Operator will have to manually compile a VCD Container Skeleton (through a Manual VCD Skeleton Editor that will be provided in the coming months). Thereby mapping information may be retrieved from the eCertis database and from further information sources to understand which evidences prove the respective criteria of qualitative selection requested by the Contracting Authority. Subsequently, the Economic Operator can use the VCD Builder component of the reference implementation to compile the VCD container. The VCD schema specifications as presented before accommodate also manual compilation of a VCD structure. Figure 10 shows the modified overall picture for this scenario.

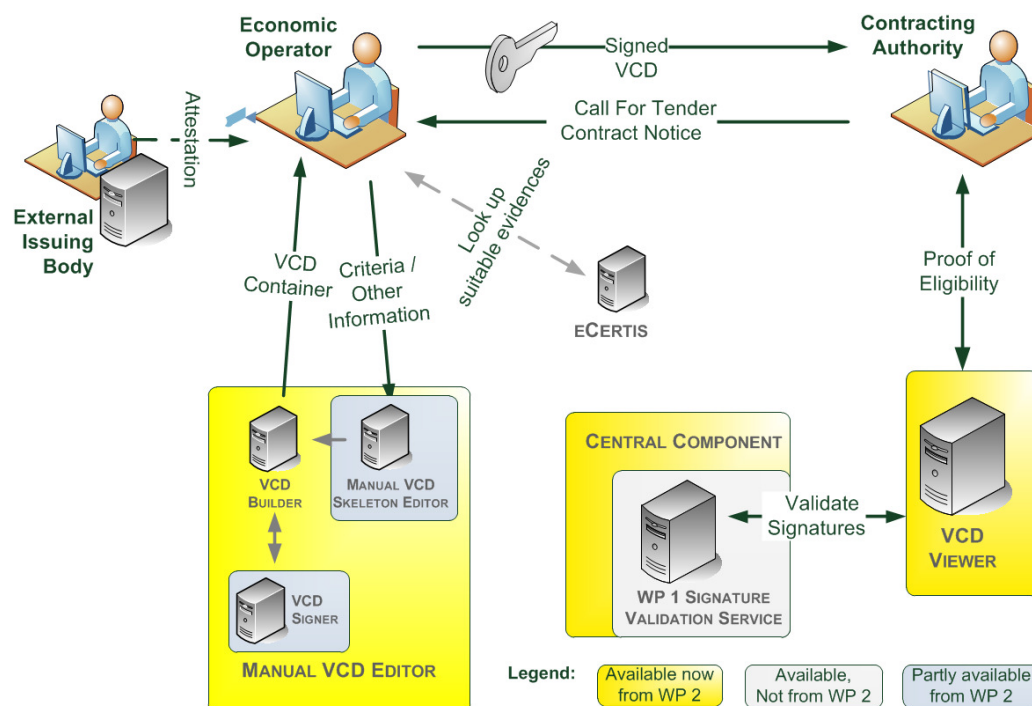


Figure 10: Manual VCD Editor for European Countries, where neither the EVS contains the ontology nor a NVS is in place

- To incorporate the decision support provided by the European VCD System directly within the NVS, an EVS/NVS service interface has been specified and implemented on both sides – the EVS as well as the NVS side. On the side of the EVS, this interface is called EVS/NVS service interface, on the side of the NVS it is encapsulated within the VCD Designer, which is part of the NVS.
- The process for building a Compound VCD Package has been specified. A Compound VCD Package is a VCD Package containing VCDs for tenderers from different countries, as defined in D 2.2 maturity level 3. As each NVS can only support tenderers from its own country, multiple NVSs have to cooperate to build a Compound VCD Package. Section 2.2 gives a more detailed description of this concept, which will be implemented in the near future.

The VCD solution for different stakeholder support levels as presented above accommodates the needs that every Economic Operator throughout Europe can be supported in creating VCD artefacts. This way, the PEPPOL WP 2 approach facilitates cross-border tendering by using even the Manual VCD Skeleton Editor in case no sophisticated VCD solution is implemented in the country.

2.2 VCD Schema adjustments

Deliverable 2.2 provided the specifications of the VCD data model and VCD schema. It defined the physical organisation of a VCD Container as well as the data structure and XML schemas for VCDs and VCD Packages. During the implementation phase, several changes and updates have been made to these XML schemas. Major updates are described in the subsequent sections. An overview of the entire VCD schema specification and the respective references is provided in Annex I: Overview of VCD Schema specifications of this deliverable.

2.2.2 VCD Skeleton types

A major amendment to the specification of D2.2 is related to the generation of a VCD Skeleton, which is a subset of a full VCD Package. It only contains information about the Economic Operators as well as the requested criteria and suitable evidences. By inputting concrete evidence documents, a VCD Skeleton becomes a full VCD, several VCDs of different Economic Operators then comprise a full VCD Package that is delivered as a ZIP file, the so called VCD Container.

According to the interaction between the National VCD System and the European VCD System taking place during the generation of a VCD Skeleton, three Pre-VCD Skeleton types exist. For each, a sub-schema has been derived from the common base VCD schema, according to the stages of this interaction (see Figure 11). In the first step, the Economic Operator provides information about the tenderer structure (i.e. different Economic Operators preparing a specific tender and their relationships such as main contractors as sub-contractors). In the next two steps, the EVS provides information about the criteria as well as suitable evidences by performing the criteria-evidence-mapping.

This leads to the following Pre-VCD Skeleton types:

1. T-Skeleton: Skeleton only containing information about the tendering Economic Operators (T = Tenderer).
2. TC-Skeleton: A T-Skeleton enriched with information about the criteria, for which Economic Operators have to provide suitable evidences. This information is derived by the EVS based on the information about the nationality of each tenderer as well as their relationship among each other (main contractor, sub-contractor).
3. TCE-Skeleton: A TC-Skeleton enriched with information about the evidences that each of the Economic Operators has to provide. This information is derived by the EVS with respect to the different nationalities of Economic Operators and requested criteria.

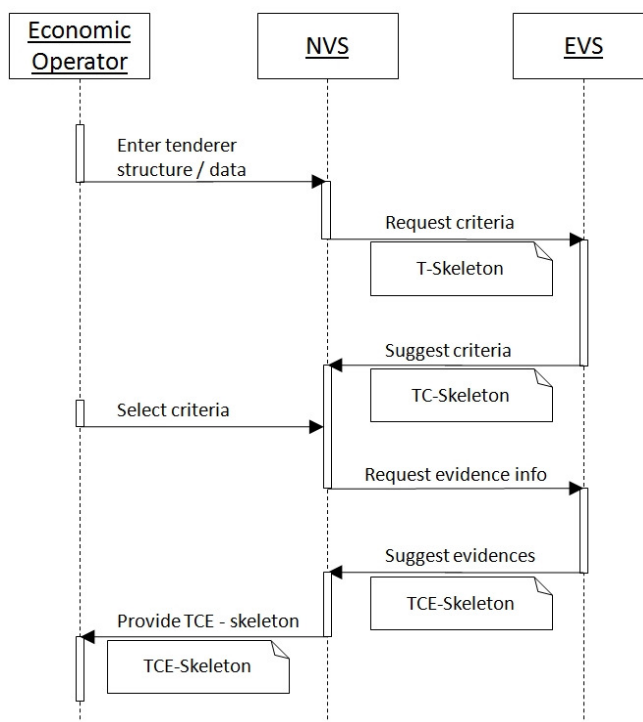


Figure 11: NVS/EVS interaction

Figure 12 illustrates the hierarchy and relationship between these skeleton types as well as the amount of information each type contains.

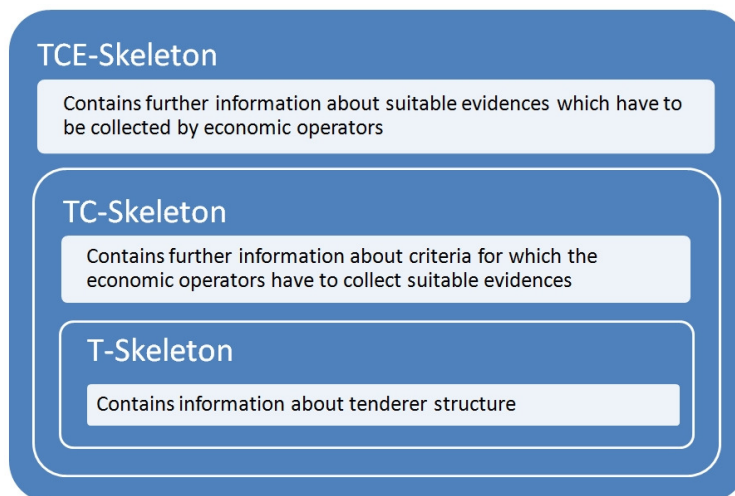


Figure 12: Pre-VCD Skeleton types: T-, TC- and TCE-Skeleton

2.2.3 Code lists

Regarding the elements of the data model that require controlled vocabularies of allowed values, D2.2 already specified the use of code lists. WP2 has started to define a set of code lists using spreadsheets to define and maintain values and different translations for every code list. Some of them rely on existing standards, such as ISO 3166-1⁸ for country codes; others require values internally defined by WP2, such as ValidationResultCode. To provide these code lists in a machine readable format that can be used by the WP2 software components, the genericcode format⁹ is used. The entire set of Code list can be found in Annex I: Overview of VCD Schema specifications of this deliverable.

2.3 Initial process specification for a Compound VCD

When Economic Operators from different countries cooperate in a single tender, e.g. as a bidding consortium or through subcontractorship, their respective NVSs have to cooperatively build a Compound VCD Package.

2.3.1 Process and Nomenclature

The ideal process would be as follows (assuming that in all countries of participating Economic Operators an NVS exists and the ontology is in the EVS):

1. *The Coordinating User* requests a VCD Package on *the Coordinating NVS*.
2. This VCD Package is built along the same process as a Non-Compound VCD, but the tenderer structure will contain tenderers from countries other than that of the Coordinating NVS. These

⁸ http://www.iso.org/iso/english_country_names_and_code_elements

⁹ http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=codelist

Economic Operatores will be called *foreign tenderers*, whereas the others will be called *local tenderers*.

3. The normal process for building the VCD Package is followed up to the TCE-Skeleton level: The Coordinating User selects the criteria for each Economic Operator (foreign as well as local).
4. The Coordinating NVS cannot provide evidence documents for the foreign tenderers. Instead, for each foreign tenderer in the tenderer structure, it contacts the NVS of that country, delegating the process of providing evidence documents to this NVS which therefore is called *Delegate NVS*.
5. The process continues as normal for the local tenderers. Of course, the full VCD Package cannot be built yet, as the Coordinating NVS is still missing the evidences for the foreign tenderers.
6. The Delegate NVSs notify the users responsible for the respective VCDs (*Delegate Users*). The Delegate NVSs allows Delegate Users to modify the selection of criteria, select the desired evidences and upload (or otherwise provide e.g. via service calls) the Evidence Documents. Once the VCD is finished, the Delegate NVS contacts the Coordinating NVS and provides the finished VCD.
7. The Coordinating NVS allows the Coordinating User to see which foreign VCDs are still missing.
8. Once all foreign VCDs are provided, the Coordinating NVS merges them together to form the final VCD Package Container including both local and foreign tenderers and their full VCDs.
9. The Coordinating NVS allows the Coordinating User to download the finished VCD Package Container.

2.3.2 Manual Process

If no suitable Delegate NVS exists for the Foreign Tenderer, in Step 4 of section 2.3.1, the Coordinating NVS should allow the Coordinating User to download the TCE-Skeleton and manually pass it on to the Delegate User who can then provide the requested Evidences through the Stand-alone VCD System as described in Figure 9, through the Manual VCD Editor as described in Figure 10, or through a NVS that doesn't support the interface necessary for automatic operation. The Delegate User would then manually send his completed VCD back to the Coordinating User who uploads it to the Coordinating NVS where it is included as per steps 7 and 8. Thus we can substitute a manual process for steps 4 and 6.

The Delegate NVS in this scenario still needs to support finishing a single VCD from a given TCE-Skeleton Package.

The Coordinating NVS in this scenario still needs to support exporting a TCE-Skeleton mid-way through the process and merging the resulting VCDs into a single VCD Package.

3 VCD Software Artifacts

This section provides an overview of the software artifacts delivered in D 2.3. The software building blocks are based on the specifications as presented in deliverables D 2.1 and D 2.2, with the revisions as pointed out in chapter 2 to be taken into consideration. Table 1 lists all PEPPOL software components for the VCD solution as depicted in the overviews charts in Figure 1 and Figure 7.

Table 1 - Main building blocks existing as per delivery of this deliverable

Building Block	Description	Decentral/ Central	Specification	Software as Open Source¹⁰
European VCD System (EVS)	The European VCD System provides the decision support for deriving the evidences in order to meet the required criteria pursuant to the underlying legal rulesets (represented as machine interpretable ontologies). The EVS hosts the ontologies, provides the reasoning and makes decision support accessible via system and user interfaces.	Central	D 2.2	Depending on detailed component – see subsequent Table 2
Ontology Management System	The Ontology Management System provides the editing and management functionality for the different ontologies the EVS is basing its decision support on. The Ontology Management System can be used simultaneously by the different ontology editing teams to keep the legal ruleset up to date.	Central	D 2.2	Depending on detailed component – see subsequent Table 2
National VCD System (NVS)	The National VCD System provides a full range of VCD functionalities to the Economic Operator from the initial selection of criteria (via the VCD Designer) to the finalization of a validated VCD Container (through the VCD Builder). Depending on	Decentral	D 2.2	Depending on Country's decisions

¹⁰ Available in full as open source under e.g MPL1.1 license and or EUPL



	national implementation architecture decisions, implementers can use all or part of the reference reference implementations for a NVS.			
EVS/NVS interaction	As a logical building block, this includes EVS/NVS Service Interface Components. Depending on national implementation architecture decisions, it may use the generic VCD Designer for the NVS-side functionality, which is interacting with the EVS/NVS service interface of the EVS.	Decentral	D2.2	EUPL (see notes below)
VCD Builder	The VCD Builder is a web or desktop application allowing users to create/build a VCD Package and is part of the NVS reference implementation	Decentral	D2.2	EUPL
VCD Designer	The VCD Designer is a component allowing users to create a TCE Skeleton that can be used by the VCD Builder for the creation of the desired VCD, taking into account the tender structure and the suggestions from the EVS. Together with the VCD Builder, it can be used in a desktop or web-based NVS environment	Decentral	D2.2	EUPL
VCD Viewer	The VCD Viewer is a web application allowing users to view the contents of VCD Packages.	Decentral / Central	D2.2	EUPL

In the subsequent Table 2, the above main building blocks are detailed in terms of common components. The components are further described along a unique template to detail the software components' main features. The subsequent sections also contain descriptions for the main building blocks of the VCD Viewer and the VCD Builder, which are not detailed further in sub-components in Table 2.

Table 2 - List of VCD common components

Component	Description	Belonging to Main Building block	Decentral/Central	Specification in D 2.2	Software as Open Source¹¹
Reasoner Components as part of the EVS					
OWL/DL Reasoner incl. rule set	Customization and amendment of Jena's OWL-DL reasoner to improve performance and reasoning results.	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
Rule Based Reasoner incl. rule sets	Custom rule based reasoner built on Jena's generic rule based reasoner. Includes rulesets for the reasoning steps performed by EVS/NVS.	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
OWL/DL Reasoner semantic service wrapper	Wrapper component to allow the OWL-DL reasoner to be used from the osSso Semantic Service Orchestration Engine.	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
Rule Based Reasoner semantic service wrapper 1	Wrapper component encapsulating the rule based reasoner and the ruleset for deriving criteria from tenderer and CA data, allowing it to be used from the osSso Semantic Service Orchestration Engine.	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL*
Rule Based Reasoner semantic service wrapper 2	Wrapper component encapsulating the rule based reasoner and the ruleset for deriving evidences from tenderer data and selected criteria, allowing it to be	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL*

¹¹ Available in full as open source under e.g MPL1.1 license and or EUPL



	used from the osSso Semantic Service Orchestration Engine.				
Ontology Schema Files as part of the EVS					
VCD ontology files (criterion and evidence schema)	OWL DL Ontology for describing Criteria and Evidences.	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
VCD ontology files (semantic service schema)	OWL DL Ontology for describing semantic services that can provide evidence documents.	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
VCD ontology files (tenderer schema)	OWL DL Ontology for describing tenderer structures.	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
VCD ontology files (tenderer criterion schema)	OWL DL Ontology for formalizing criterion suggestions and restricting evidences to certain types of tenderers.	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
VCD ontology files (collector schema)	OWL DL Ontology for describing intermediate and final reasoning results for traceability including criterion and evidence suggestions.	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
Ontology Instance Files as part of the EVS					
VCD ontology files (EU ontology)	The modelled European Criteria according to the Criterion & Evidence Schema	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
VCD ontology files (legal domain ontologies: Austria, 2x Germany, France, Norway (draft version), Italy, Greece)	For each Legal Domain: National Criteria, mappings to EU Criteria, Evidences, Criterion Requirements and Evidence Restrictions and the interrelations between them.	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
Packaging/Disassembly Components as part of the EVS and EVS/NVS interaction					
VCD (Skeleton) Packager incl.	Component to generate VCD Packages	EVS and NVS	Central/	PEPPOL D2.2 +	EUPL

validation	(including skeleton variants).		Decentral	Attachement A+ Attachement C	
VCD (Skeleton) Disassembler incl. validation	Component to disassemble, validate and handle VCD Packages (including skeleton variants).	EVS and NVS	Central/ Decentral	PEPPOL D2.2 + Attachement A+ Attachement C	EUPL
EVS/NVS interaction					
EVS/NVS service interface	Java Implementation for the SOAP Interface that allows NVSs to call the EVS in order to retrieve: <ul style="list-style-type: none"> Criteria suggestion. Evidence suggestion. 	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL*
EVS/NVS service interface VCD schema derivatives	Derivatives of the full VCD Package Specification: <ul style="list-style-type: none"> T-Skeleton: VCD Package reduced to only Tenderer structure and data. TC-Skeleton: extending T-Skeleton with criteria (depending on context either as suggested by the EVS or as selected by the User) TCE-Skeleton: TC-Skeleton extended with Evidences suggested by the EVS. 	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
EVS/NVS service interface description	WSDL and XSD files describing the SOAP interface in an implementation-independent manner.	EVS	Central	PEPPOL D2.2 + Attachement C	EUPL
User Interaction Service Fragment Configurations as part of EVS and NVS					
Configuration of User Interaction Service Fragments (EVS)	EvidenceReviewFragment: displays suggested evidences.	EVS (PEPPOL.AT)	Central	PEPPOL D2.2 + Attachement C	EUPL

Configuration of User Interaction Service Fragments (NVS)	<ul style="list-style-type: none"> ○ EvidenceSelectorFragment: similar to EvidenceReviewFragment, but allows actual selection of evidences. ○ ServiceSelectorFragment: allows selection of issuing services to get evidence documents from. ○ UploadFragment: allows upload of evidence documents. 	NVS (PEPPOL.AT)	Decentral	PEPPOL D2.2 + Attachement A	EUPL
Configuration of User Interaction Service Fragments (EVS and NVS)	<ul style="list-style-type: none"> ○ CreateNewVCDRequestFragment: asks for initial data about the new VCD request (CA nationality, ...) ○ TendererStructureFragment: allows entering of tenderer structure and data. ○ CriterionSelectorFragment: displays all possible criteria highlighting suggestions; allows selection of criteria. ○ DownloadFragment: allows downloading of finished VCD Package (or TCE-Skeleton in case of EVS). ○ UserStateFragment: displays all VCD requests of the current user with their current state. 	EVS and NVS (PEPPOL.AT)	Central/ Decentral	PEPPOL D2.2 + Attachement A+ Attachement C	EUPL
Issuing Body Interfaces (wrapper)	Library for accessing some Austrian	NVS (PEPPOL.AT)	Decentral	PEPPOL D2.2 +	EUPL*

and m2n service libraries)	Evidence Issuing Services as Services in osSso Semantic Service Orchestration Engine: <ul style="list-style-type: none"> ○ Ediktsdatei ○ Firmenbuch (Business Register) 			Attachement A	
Ontology Management System					
GUI description for Ontology Management System	Semantic model of the Ontology Management System's GUI.	EVS (Ontology Management System)	Central	PEPPOL D2.2 + Attachement C	EUPL
Ontology export queries	SPARQL queries for exporting Ontology Instance Files from a triple store.	EVS (Ontology Management System)	Central	PEPPOL D2.2 + Attachement C	EUPL
m2n Intelligence Management Framework	The Framework provides functionality for Ontology Management and maintenance, user and access amangement, repository management, GUI management, etc. It acts as the central application controller for the 3 systems EVS, NVS and Ontology Management System.	EVS and NVS, Ontology Management System (PEPPOL.AT)	Central/ Decentral		Background system (proprietary solution)
EVS User access and role management incl. authentication (User Access Manager)	Programmatic interface to enable external components to manage user accounts for the EVS.	EVS	Central	PEPPOL D2.2 + Attachement C	Interface: EUPL Actual external component: Background system
VCD Designer					
NVS Model	An internal module of the VCD Designer,	VCD Designer	Decentral	PEPPOL D2.2 +	EUPL



	<p>it represents data structure model concerning T-Skeleton Package; Used by NVS-Assembler component to assemble a T-Skeleton Package from the initial tender data which are provided by the Economic Operator; Integrates exceptions based on the XML T-Skeleton schema constraints. Dependencies: VCD Schema (T-Skeleton) Main implementation points: Builder design pattern used for implementation.</p>			Attachement A	
NVS-Assembler	<p>An internal module of the VCD Designer, it assembles initial tender data in a structured document (XML instance of T-Skeleton Package) based on the input by the Economic Operator. Dependencies: NVS-Model, automatically generated artifacts based on the provided XML Schemas Main implementation points: Java architecture for XML binding (JAXB) is used to automatically generate T-Skeleton artifacts</p>	VCD Designer	Decentral	PEPPOL D2.2 + Attachement A	EUPL
NVS-Invoker	<p>An internal module of the VCD Designer, it places VCD Package-Skeleton order request using the EVS/NVS interface. Process consists of two phases: -Criteria Suggestions (Input: the generated T-Skeleton Package. Output: the TC-Skeleton with suggested criteria from the European VCD Service). -Evidence Suggestions (Input: the modified (according to the user selection) TC-Skeleton. Output: the</p>	VCD Designer	Decentral	PEPPOL D2.2 + Attachement A	EUPL

	<p>TCE-Skeleton Package from the European VCD Service).</p> <p>Dependencies: WSDL document provided by European VCD Service</p> <p>Main implementation points: Java API for XML Web Services (JAX-WS) used to automatically generate client's web service artifacts</p>				
NVS-Utills	An internal module of the VCD Designer, it provides utilities to all NVS-EVS Interface components.	VCD Designer	Decentral	PEPPOL D2.2 + Attachement A	EUPL
NVS-Validator	<p>An internal module of the VCD Designer, it validates any document instance based either on TC or TCE Skeleton Package XML Schema</p> <p>Dependencies: Automatically generated artifacts based to provided xml schema</p>	VCD Designer	Decentral	PEPPOL D2.2 + Attachement A	EUPL
NVS-UI	<p>An internal module of the VCD Designer, it represents a GUI implementation for the two main phases of communication between the NVS and the European Service Provider. Firstly, the GUI constructs a T-Skeleton Package derived by the initial tender data gathering. Secondly, it graphically represents the suggested criteria obtained by the European VCD Service. User is able to select his/her preferred mapped criteria.</p> <ul style="list-style-type: none"> • Initial tenderer structure UI • Criteria selection UI 	VCD Designer	Decentral	PEPPOL D2.2+ Attachement A	EUPL
EVS/NVS service interface VCD schema derivatives	<ul style="list-style-type: none"> • T-Skeleton • TC-Skeleton 	VCD Designer	Decentral	PEPPOL D2.2+ Attachement A	EUPL

	<ul style="list-style-type: none"> TCE-Skeleton 				
jaxb-model generated classes	<p>Java interfaces and implementation classes corresponding to VCD skeleton schema.</p> <p>Used within the VCD Designer for XML Schema integration to application logic, mapping between VCD skeleton documents and Java code, which enables other NVS components to use generated code to read, manipulate, and re-generate XML data.</p>	VCD Designer	Decentral	PEPPOL D2.2 + Attachement A	EUPL
NVS reference implementation Integration modules					
VCD Designer / VCD Builder integration	<p>A software module responsible for the inter-communication of the VCD Designer and the VCD Builder components. It passes a validated TCE-skeleton package through a proxy service to the VCD builder.</p>	NVS reference implementation	Decentral	PEPPOL D2.2 + Attachement A	EUPL
Integrated NVS User Management	<p>Provides Single sign-on user management implementation across the VCD Builder and the VCD Designer using spring security. This module can support access control to any other NVS component may be integrated to the NVS, as a separate OSG1 bundle</p>	NVS reference implementation	Decentral	PEPPOL D2.2 + Attachement A	EUPL

* Open source component under EUPL license, using m2n Intelligence Management Framework as background system (proprietary system)



3.3 Reasoner Components

3.3.1 Purpose

These components provide ontology reasoning for the EVS and the Austrian NVS in order to derive criteria, evidences and services according to a specific tenderer setting.

The components consist of several distinct subcomponents:

- The OWL DL Reasoner with its corresponding ruleset implements OWL DL reasoning. It is based on Jena's implementation of OWL DL reasoning with a amendments to improve reasoning performance and the clarity of the output model.
- The Rule-Based reasoner with its corresponding rulesets implements custom rule-based reasoning for the PEPOL-specific reasoning steps performed by the EVS/NVS. It is based on Jena's general-purpose rule-based reasoner. It contains distinct rulesets for:
 - Suggesting criteria based on the criterion requirements defined in the CA-national ontology.
 - Finding the corresponding T-national criteria for all CA-national criteria and manifesting the full path utilized for this reasoning.
 - Suggesting possible evidences based on the results of the previous steps. This step is able to tolerate certain types of irregularities in the national ontologies (e.g. missing national criteria or national criteria that have no corresponding EU criteria)
 - Suggesting possible evidence issuing services based on selected evidences.
- The different semantic service wrappers for the reasoners encapsulate reasoning steps or groups of reasoning steps and facilitate the integration of the reasoners with the osSso (One-Stop Semantic Service Orchestration) engine in the m2n Intelligence Management framework.

3.3.2 Implementation environment

Technology / Tool	Version	Description
Java Development Kit (JDK)	JDK SE 6	Java Development Kit Standard Edition
Maven	2.2	Apache Maven Software Project Management and Comprehension Tool
Jena	2.4	Jena Semantic Web Framework

3.3.3 Deployment environment

The components are to be deployed as java libraries and can be used in any kind of Java SE or EE environment.

3.3.4 Reference implementation

The reference implementation is available online as part of the EVS and the Austrian NVS.

Source code and binaries are provided as part of the software distribution for this deliverable:

Directory	Description	Software as Open Source
src/reasoners	OWL DL Reasoner, Rule-Based reasoner.	EUPL



bin/reasoners	Including the respective rulesets and a maven pom describing dependencies.	
src/reasoner-wrappers bin/reasoner-wrappers	The Service wrappers as described above, including a maven pom describing their dependencies.	EUPL

3.3.5 Outlook on functionality extensions

The core functionality of the reasoners is expected to remain relatively stable, but the rulesets will be expanded to support tenderers from and tenders in countries for which no or only limited formal modelling of criteria and evidences has been done up to now.

These extensions will be at the heart of enabling and easing use of the results of WP2 in countries not originally involved in PEPPOL WP 2.

3.4 Ontology Schema Files

3.4.1 Purpose

The ontology schema files describe the schema for the different ontologies used in the EVS and the Austrian NVS. They are split by topic:

- The Criterion and Evidence schema specifies how criteria and evidences (instances) have to be modelled.
- The Semantic Service Schema specifies how Evidence Issuing Services have to be modelled.
- The Tenderer Schema specifies how Tenderer Structure and Tenderer Data has to be modelled.
- The Tenderer Criterion Schema specifies how criteria and evidences can be tied to certain types of tenderers. Examples of such rules are:
 - “For Italian tenders, it should be suggested that all representatives of all the companies involved in the tender prove Absence of Conviction.”
 - “The Austrian Criminal Record Extract can only be provided for natural persons, not for companies.”
- The Collector Schema specifies how suggested/selected criteria, evidences and services are tied to the tenderer instances. As such it specifies the input and output data format for most of the rule-based reasoners' rules and provides traceability of reasoning results (especially regarding the derivation of criteria and evidences).

3.4.2 Implementation Environment

Technology / Tool	Version	Description
m2n Intelligence Management	6.3	m2n Intelligence Management Semantic Application framework

3.4.3 Deployment Environment

The ontology components are N3 files and can be used with any compatible tool or framework that fully supports OWL DL).

3.4.4 Reference Implementation

The reference implementation is used as part of the EVS and the Austrian NVS, available online¹².

The N3 files for these components are provided as part of the software distribution for this deliverable:

Directory	Description	Software as Open Source
plaintext/ontology-schema	All ontology schemas described above, each a separate N3 file.	EUPL

3.4.5 Outlook on functionality extensions

These schemas are expected to remain stable as they are a major integration point.

It is possible that the Criterion and Evidence schema will have to be slightly extended in the course of supporting countries that can only provide limited ontologies (e.g. by allowing mapping national evidences directly to EU criteria instead of national criteria).

3.5 Ontology Instance Files

3.5.1 Purpose

Part of WP 2 was modelling the EU and national criteria and the national evidences as well as the interrelations between them and tenderer-type specific criterion requirements and evidence restrictions for all of the participating countries – see D 2.2.

The modelled ontology instances follow the Criterion and Evidence schema and the Tenderer-Criterion schema as described above.

These ontologies provide the basic set of information upon which the Rule-Based reasoners base their decisions.

The modelled ontology instances are split up by legal domain:

- countries.n3 specifies all the legal domains and their names
- eu.n3 contains all the EU criteria
- at.n3 – Austria
- de_vol.n3 – Germany, VOL (services contracts)
- de_vob.n3 – Germany, VOB (construction domain)
- fr.n3 – France
- gr_sp.n3 – Greece (Services and Products)
- it.n3 – Italy
- no.n3 – Norway (not finalized ontology)

¹² The EVS is available at <http://eu-vcd.peppol.at/vcd/vcd/>, the Austrian NVS at <http://vcd.peppol.at/nspp/webclient/>.

3.5.2 Implementation Environment

<i>Technology / Tool</i>	<i>Version</i>	<i>Description</i>
m2n Intelligence Management	6.3	m2n Intelligence Management Semantic Application framework

3.5.3 Deployment Environment

These components are N3 files and can be used with any compatible tool or framework that supports full OWL DL.

3.5.4 Reference Implementation

The reference implementation is used as part of the EVS and the Austrian NVS, available online¹³.

The N3 files for these components are provided as part of the software distribution for this deliverable:

<i>Directory</i>	<i>Description</i>	<i>Software as Open Source</i>
plaintext/mappings	All ontologies described above, each a separate N3 file.	EUPL

3.5.5 Outlook on functionality extensions

Extensions can be expected concerning:

- Testing and further usage might uncover discrepancies between the model and juridical reality, necessitating adaptations of the national ontologies.
- Further legal domains will have to be supported (e.g. new countries).
- The legal frameworks in the countries might evolve, which will have to be reflected in the corresponding national ontologies.

Those extensions, however, can be included fairly easily due to the modelling approach chosen within PEPPOL WP 2 works in phases 2 and 3.

3.6 Packaging/Disassembly Components

3.6.1 Purpose

These components transform a VCD Package into a single, unified XML Tree (Disassembler) or transform the XML Tree into a VCD Package (Packager). The unified XML Tree includes all information regarding the VCD Package and all included VCDs.

These components also handle included documents such as evidence documents or non evidence files.

They also validate the VCD Packages against the Schema before disassembly and after packaging.

¹³ The EVS is available at <http://eu-vcd.peppol.at/vcd/vcd/>, the Austrian NVS at <http://vcd.peppol.at/nspp/webclient/>.

Apart from the full VCD Package schema, the Skeleton schemas used for communication between the EVS and NVSs in the process of building a VCD Package are supported.

3.6.2 Implementation environment

<i>Technology / Tool</i>	<i>Version</i>	<i>Description</i>
Java Development Kit (JDK)	JDK SE 6	Java Development Kit Standard Edition
Maven	2.2	Apache Maven Software Project Management and Comprehension Tool
Jena	2.4	Jena Semantic Web Framework

3.6.3 Deployment environment

The components are to be deployed as java libraries and can be used in any kind of Java SE or EE environment.

3.6.4 Reference implementation

The reference implementation is available online as part of the EVS and the Austrian NVS¹⁴.

Source code and binaries are provided as part of the software distribution for this deliverable:

<i>Directory</i>	<i>Description</i>	<i>Software as Open Source</i>
src/packager bin/packager	Packager and Disassembler, including a maven pom describing dependencies. XML Schema Files are required but not included here, as they are provided separately as part of this deliverable.	EUPL

3.6.5 Outlook on functionality extensions

The basic structure of the VCD Package (which these components rely on) is expected to remain stable. However, schematron validation rules will be defined by the VCD Schema Task Force and support for these should be added to the Packager and Disassembler components.

3.7 EVS/NVS Service Interface (EVS side)

3.7.1 Purpose

The EVS provides reasoning services for the NVSs as it derives criteria and evidence suggestions regarding a specific tenderer setting. To allow the NVSs to call these services, a SOAP interface is provided by the EVS. The calls to and replies from this SOAP interface contain unfinished VCD Packages. The exact states as modifications of the full VCD Package Specification (Skeleton Schemas) are also specified.

¹⁴ The EVS is available at <http://eu-vcd.peppol.at/vcd/vcd/>, the Austrian NVS at <http://vcd.peppol.at/nspp/webclient/>.

This Component group contains

- The Java implementation of the interface on the EVS side, including the JAX-WS annotations used to generate the WSDL files.
- Three different Skeleton Schemas (T-Skeleton, TC-Skeleton, TCE-Skeleton – see also chapter 2), each specified as
 - A XSLT “patch” against the full VCD Package Schema
 - The output of this “patch” when applied to Version 1.0 of the VCD Package Schema
- The WSDL and XSL files specifying the interface in an implementation-independent way.

3.7.2 Implementation environment

Technology / Tool	Version	Description
Java Development Kit (JDK)	JDK SE 6	Java Development Kit Standard Edition
Maven	2.2	Apache Maven Software Project Management and Comprehension Tool
JAX-WS	2.1	Java API for XML Web Services

3.7.3 Deployment environment

The EVS/NVS Service Interface implementation is to be deployed as a WAR Web Application.

The WSDL and XSL files specifying the Interface can be used to automatically or manually generate stub implementations in any suitable programming environment.

The Skeleton Schemas (T-Skeleton, TC-Skeleton, TCE-Skeleton) are XML Schemas and can be used with any tool supporting XSD.

3.7.4 Reference implementation

The reference implementation is available online as part of the EVS¹⁵.

These components are provided as part of the software distribution for this deliverable:

Directory	Description	Software as Open Source
src/peppol-esp-services bin/peppol-esp-services	Java Implementation of the Interface, including a maven pom detailing dependencies	EUPL
plaintext/peppol-esp-services-wsdl	WSDL and XSL files describing the Interface in an implementation-independent way.	EUPL
plaintext/skeleton-schema	The various Skeleton schemas.	EUPL

¹⁵ The EVS/NVS Service interface responds to SOAP calls to <http://test-vcd-esp.peppol.at/peppol-esp-services/ESPInterface> and provides its WSDL at <http://test-vcd-esp.peppol.at/peppol-esp-services/ESPInterface?wsdl>.

3.7.5 Outlook on functionality extensions

This interface is expected to remain stable. However, continuing work on the Compound VCD functionality (multiple NVSs cooperating to build a multi-national single VCD Package, see Chapter 2.3) will require the EVS to keep VCD UUIDs consistent between its input and output data, which it currently doesn't. This implies no change for clients of this interface, but the implementation on the EVS will need some amendments.

The future NVS-NVS interface for the Compound VCD can also be seen as an extension of this component, evolving it into a collection of WP 2 inter-system interfaces.

3.6 User Interaction Fragment Configurations

3.6.1 Purpose

The EVS and the Austrian NVS use osSso (One-Stop Semantic Service Orchestration) technology as a full semantically orchestrated service bus. As part of this approach, their user interface is fully described in ontology form as a configuration for the osSso UIS (User Interaction Service).

Each distinct screen (called a UIS fragment) shown to the user is described purely in RDF.

Some fragments are used by both the EVS and the Austrian NVS, others by only the EVS or the NVS:

- Common
 - Create New VCD Request Fragment
 - Tenderer Structure Fragment
 - Criterion Selector Fragment
 - User State Fragment
 - Download Fragment
- EVS
 - Evidence Review Fragment
- NVS
 - Evidence Selector Fragment
 - Service Selector Fragment
 - Upload Fragment

3.6.2 Implementation Environment

<i>Technology / Tool</i>	<i>Version</i>	<i>Description</i>
m2n Intelligence Management	6.3	m2n Intelligence Management Semantic Application framework

3.6.3 Deployment Environment

These components are N3 files and can be used with any compatible tool or framework that supports fully OWL DL.

3.6.4 Reference Implementation

The reference implementation is used as part of the EVS and the Austrian NVS, available online¹⁶.

The N3 files for these components are provided as part of the software distribution for this deliverable:

Directory	Description	Software as Open Source
plaintext/fragments	All fragments described above, each a separate N3 file.	EUPL

3.6.5 Outlook on functionality extensions

A major rework of the GUI of all WP 2 components will be necessary in order to harmonise the different user interface approaches of the building blocks, which will affect most of the fragments, introduce new fragments and might also replace some of the existing ones,

3.7 Issuing Body Interfaces

3.7.1 Purpose

The Austrian NVS orchestrates several Austrian Evidence Issuing Services to provide a user-friendly way of automatically retrieving evidence documents for Austrian Tenderers.

The evidence issuing services currently supported are:

- Firmenbuch (Business Register)
- Ediktsdatei

The component provides the service libraries to interface with these services as well as the necessary wrapping for using these services as semantic services in osSso (One-Stop Semantic Service Orchestration).

3.7.2 Implementation environment

Technology / Tool	Version	Description
Java Development Kit (JDK)	JDK SE 6	Java Development Kit Standard Edition
Maven	2.2	Apache Maven Software Project Management and Comprehension Tool
Jena	2.4	Jena Semantic Web Framework
m2n Intelligence Management	6.3	m2n Intelligence Management Semantic Application Framework

3.7.3 Deployment environment

This component is dependent on the m2n Intelligence Management Framework as background component. Due to this fact and as it is specific to the Austrian context, it cannot directly be used in other contexts. Of course, lessons and insight into how these evidence issuing services can be accessed programmatically can be shared upon request.

¹⁶ The Austrian NVS is available at <http://vcd.peppol.at/nspp/webclient/>.

3.7.4 Reference implementation

The implementation is implemented as part of the Austrian NVS.

Source code and binaries are provided as part of the software distribution for this deliverable:

<i>Directory</i>	<i>Description</i>	<i>Software as Open Source</i>
src/issuing-service-wrappers bin/issuing-service-wrappers	The issuing service wrappers.	EUPL

3.7.5 Outlook on functionality extensions

More evidence issuing services will be included in iterative steps as the project progresses.

3.8 Ontology Management System

3.8.1 GUI Description for Ontology Management System

3.8.1.1 Purpose

The Ontology Management System used for working on the national and EU ontologies is based on m2n Intelligence Management technology. The user interface of the Ontology Management System is therefore itself modelled in OWL DL.

This component describes the semantic model of the Ontology Management System GUI.

3.8.1.2 Implementation Environment

<i>Technology / Tool</i>	<i>Version</i>	<i>Description</i>
m2n Intelligence Management	6.3	m2n Intelligence Management Semantic Application framework

3.8.1.3 Deployment Environment

The Ontology Management System is based on the m2n Intelligence Management Framework, which forms the background system for the Ontology Management System as well as for the EVS and the Austrian NVS.

The semantic model of the ontology Management System GUI is represented as N3 files and can be used with any compatible tool or framework that supports full OWL DL.

3.8.1.4 Reference Implementation

The Ontology Management System is provided via citrix by the Austrian Federal Computing Centre. Its use is currently limited to a select few members of WP2.

The N3 file for the semantic model of the ontology Management System GUI is provided as part of the software distribution for this deliverable:

<i>Directory</i>	<i>Description</i>	<i>Software as Open Source</i>
plaintext/modeller-gui	The semantic model of the Ontology Management System GUI.	EUPL

3.8.1.5 Outlook on functionality extensions

It is intended to make the Ontology Management System available as a web application by the first Quarter of 2011. This will probably be combined with optimizing the UI.

3.8.2 Ontology Export Queries

3.8.2.1 Purpose

The Ontology Management System stores its user data in one or more general-purpose triple stores. To extract Ontology Instance Files, SPARQL queries are used to select the relevant instances (which are then exported with all of their properties in n3 files).

The exported Ontology Instance Files can then e.g. be deployed to the EVS or included in the software distribution for this deliverable.

3.8.2.2 Implementation Environment

<i>Technology / Tool</i>	<i>Version</i>	<i>Description</i>
m2n Intelligence Management	6.3	m2n Intelligence Management Semantic Application framework

3.8.2.3 Deployment Environment

The exported Ontology Instance Files are N3 files and can be used with any compatible tool or framework that supports full OWL DL.

3.8.2.4 Reference Implementation

The Ontology Management System is provided via citrix by the Austrian Federal Computing Centre. Its use is currently limited to a select few members of WP 2.

The N3 files for this component are provided as part of the software distribution for this deliverable:

<i>Directory</i>	<i>Description</i>	<i>Software as Open Source</i>
plaintext/modeller-export-queries	The SPARQL queries for exporting the Ontology Instance Files described above, as a N3 file.	EUPL

3.8.2.5 Outlook on functionality extensions

More queries will be added as more legal domains will be included in the project.

3.9 m2n Intelligence Management Framework

3.9.1 Purpose

The m2n Intelligence Management framework is a semantic application framework suitable for flexibly building complex information management, knowledge discovery and analysis or decision support systems. m2n Intelligence Management strictly follows a model only approach, including a powerful semantic service orchestration engine (osSso). It is used as the fundamental basis for the EVS, the Austrian NVS and the Ontology Management System.

3.9.2 Implementation environment

Due to its generic nature and high maturity level, the m2n Intelligence Management framework has numerous relationships to other software components, both at build time and at run time. The most central of them are:

Technology / Tool	Version	Description
Java Development Kit (JDK)	JDK SE 6	Java Development Kit Standard Edition
Maven	2.2	Apache Maven Software Project Management and Comprehension Tool
Jena	2.4	Jena Semantic Web Framework

3.9.3 Deployment environment

This component suite is operating as a background system. For more information see <http://www.m2n.at> or contact office@m2n.at.

3.9.4 Reference implementation

This component suite is used as the base of the EVS, the Austrian NVS and the Ontology Management System. Since it is acting as a background system, it is not provided as part of the software distribution for this deliverable.

3.10 EVS User access and role management incl. authentication

3.10.1 Purpose

This component supports a programmatic way for other components to provide user registration services for the EVS.

3.10.2 Implementation environment

Technology / Tool	Version	Description
Java Development Kit (JDK)	JDK SE 6	Java Development Kit Standard Edition
Maven	2.2	Apache Maven Software Project Management and Comprehension Tool
Jena	2.4	Jena Semantic Web Framework
m2n Intelligence Management	6.3	m2n Intelligence Management Semantic Application Framework

3.10.3 Deployment environment

This component describes the interface that enables other components to provide user data to the EVS. It cannot be deployed on its own but can be used as a guideline when implementing such a component.

3.10.4 Reference implementation

The reference implementation is available online as part of the EVS¹⁷.

Source code and binaries are provided as part of the software distribution for this deliverable:

Directory	Description	Software as Open Source
src/evs-user-management-interface bin/evs-user-management-interface	The interface and current implementation on the EVS side.	EUPL

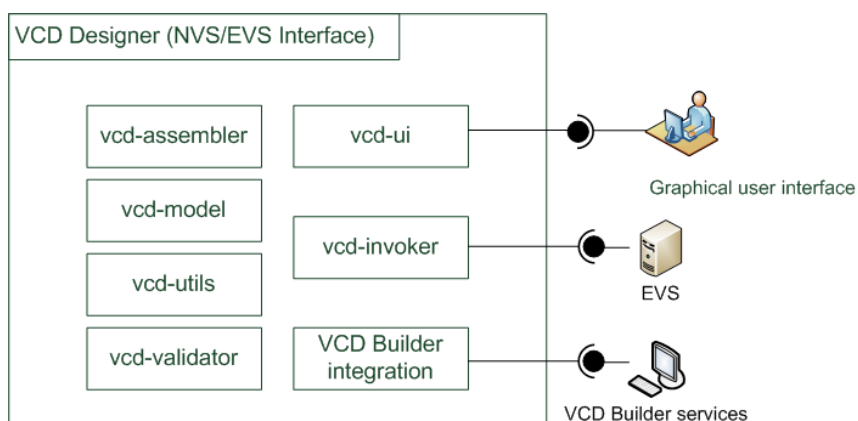
3.10.5 Outlook on functionality extensions

No extension planned.

3.11 VCD Designer (incl. NVS-EVS Interface on the NVS-side)

Figure 13 provides an overview about the components of the VCD Designer.

Figure 13:
Components of the
VCD Designer



3.11.1 Purpose of the VCD Designer

The purpose of the VCD Designer component in its v1.0 version is to construct a VCD TCE Skeleton Package taking into account the tender structure as interpreted by the Economic Operator, and the suggestions of the EVS. The VCD Designer supports all steps the end user needs to follow from the initial input of tender data structure until the final compilation of the VCD Skeleton Package that is passed on to the VCD Builder component.

VCD Designer is available in two configurations:

- A standalone client configuration that be executed from the user's desktop
- A Web Client

General description

The overall process is being represented by a user interface that introduces two main steps to the user.

- The first step is entering the initial tender data and sending of the T-Skeleton Package to the European VCD System, in order to get the suggested criteria.

¹⁷ The EVS is available at <http://eu-vcd.peppol.at/vcd/vcd/>.

- b. The second step is the criteria selection and the sending of the TC-Skeleton Package to the European VCD System, in order to get the suggested evidences.

At the end of the process, the user is able to either download the TCE-Skeleton Package received or redirect the workflow to the VCD Builder. This depends on whether the user wants to continue the VCD creation workflow seamlessly in synchronous mode or prefers to execute each step separately in an asynchronous mode.

The VCD Designer depends on the following specific models, services and modules:

- **NVS-Model:** representing the different data structures derived from the corresponding VCD Schemas that represent the T-, TC- and TCE-Skeleton, are used throughout the application.
- **Services:** Two main external services, provided by the EVS and VCD Builder components, are used by the VCD Designer component in order to integrate all process steps within its scope, as outlined above. The EVS provides a web service which is called via SOAP messaging and the VCD Builder (see section 3.12) provides a proxy service which is called via OSGi as the two components are integrated.
- **Modules:** In order to complete a full process of each phase of the VCD Designer operations, four sub-processes are required: User Interaction, assembly, invocation and validation. Each process is developed in its own separate component:
 - **NVS-UI:** The NVS-UI component is responsible for the generation of the User Interface and thus the interaction of the User (CA or EO) with the NVS System. It provides the GUI implementation for the two main phases of communication between the NVS and the European VCD System. Firstly, the GUI constructs a T-Skeleton Package derived by the initial tender data gathering. Secondly, it graphically represents the suggested criteria obtained by the European VCD System.
 - **NVS Assembler:** The NVS Assembler is responsible for the construction of a T-Skeleton Package based on the appropriate designed model for this purpose.
 - **NVS-Invoker:** The NVS Invoker module is responsible for the communication and interoperability between the EVS and the NVS. It sends the compiled VCD T- and TC-Skeleton Packages to the EVS SOAP Interface and send the responses back to NVS-UI
 - **NVS-Validator:** All receivable packages from the EVS need to be validated against the VCD Schema. The NVS-Validator module is responsible for the validation
 - There is also a utility module, the **NVS-Utils**, which provides helper functionality for all the above modules.
- **NVS Integration:** The VCD Designer is integrated with the VCD Builder within the NVS components through two modules:
 - **Designer-Builder Integration module:** A software module responsible for the inter-communication of the VCD Designer and the VCD Builder components. It passes a validated TCE-Skeleton Package through a proxy service to the VCD Builder.
 - **Integrated NVS User Management:** Provides Single sign-on user management implementation across the VCD Builder and the VCD Designer using spring security. This module can support access control to any other NVS component may be integrated to the NVS reference implementation, as a separate OSG1 bundle

3.11.2 Implementation environment

Technology / Tool	Version	Description
NetBeans Integrated Development Environment (IDE)	6.9.1	NetBeans Java EE IDE
Java Development Kit (JDK)	JDK SE 6	Java Development Kit Standard Edition
Apache Maven	2.2.1	Build and Dependency Management
Equinox OSGi container	3.6.0	Implementation of the OSGi R4 core framework
Vaadin web framework	6.4.5	RIA framework built on GWT-based widgets
JAXB	RI 2.1	Java XML binding
Axis2 JAX-WS	RI 2.1.3	Java for XML Web Services. Apache Axis2 is a core engine for Web services in Java.

3.11.3 Deployment environment

The VCD Designer component is distributed as a set of OSGi software bundles integrated with the VCD Builder component. These two components share the same deployment environment, in particular the VCD Bootstrap component, which is a customized OSGi container based on Equinox.

The aforementioned set of software bundles consists of five JAR files and one WAR file. Concerning the WAR file, it is deployed on a Jetty Application Server that also runs as a separate OSGi bundle inside VCD Bootstrap.

Bundles are: NVS–Model, NVS–Assembler, NVS–Invoker, NVS–Validator, NVS–Utils, NVS–UI.

3.11.4 Reference implementation

The reference application of the VCD Designer is available online deployed on the embedded Jetty Application Server bundle of VCD Bootstrap component. Version 1.0 of the VCD Designer supports version 1.0 of the VCD Schema.

Item	Resource	Version
WAR	NVS-ui-1.0.0.RELEASE.war	1.0.0
JAR	nvs-assembler-1.0.0.RELEASE.jar nvs-invoker-1.0.0.RELEASE.jar nvs-model-1.0.0.RELEASE.jar nvs-utils-1.0.0.RELEASE.jar nvs-validator-1.0.0.RELEASE.jar	1.0.0
Embedded Application Server	jetty-6.1.25.jar	6.1.25
VCD Schema	VCD Schema	1.0

In order to install the VCD Designer, the VCD Bootstrap software component has to be installed first (following the installation instructions¹⁸ of the VCD Bootstrap). Once the customized container is

¹⁸ <http://coeus.ds.unipi.gr/wiki/display/VCDWIKI/Bootstrap+Packaging+And+Configuration>

installed and running, the VCD Designer application can be accessed via a web browser¹⁹ by logging in and choosing “Create a new VCD Package”.

3.11.5 Outlook on functionality extensions

Functional extensions to version 1.0 of the VCD Designer will be implemented within future software releases. These future versions will be released in incremental cycles and will cover the following functionality:

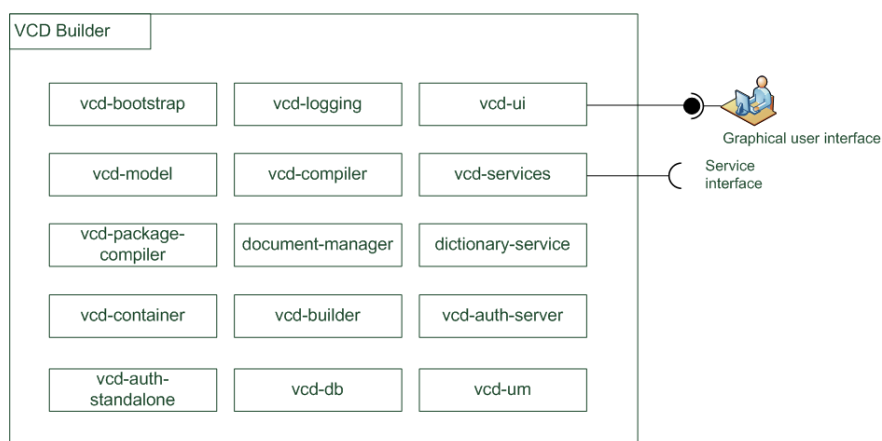
- Provision of an interface for the Contracting Authorities to be able to input the initial tender structure themselves, making them available for each prospective Economic Operator that wants to use the NVS to design and then build a VCD for this particular tender
- Provision of an EVS-NVS interface that should perform two functions:
 - Import a tender structure provided by a CA in another NVS
 - Support Compound VCD scenarios among Economic Operators from different countries and/or using different NVS providers (see Chapter 2.3)
- Multi-lingual user interface: Ability to choose between different languages to visualize user interface elements in other languages (e.g. French, Greek).
- Multiple VCD Package support.
- Database and file management support in order to be able to store, view and re-use TCE Skeletons.
- UI improvements and adoption of the unified GUI, as all VCD components.
- Digital Signatures support (future CA interface only, if applicable) .

3.12 VCD Builder

Figure 14 provides an overview about the components of the VCD Builder.

Figure 14: Components of the VCD Builder

3.12.1 Purpose of the VCD Builder



The purpose of the VCD Builder component in its v1.0 version is to compile a VCD Container on the basis of a TCE-Skeleton the VCD Designer (see section 3.11) has delivered. For this purpose, the VCD Builder requests the evidence documents (either from the Economic Operator or directly via

¹⁹ <http://Server-Name/vcd-ui>

external issuing service interfaces) and then builds/creates the VCD. The VCD Builder supports all steps the end user needs to follow for the final compilation of the VCD Package.

VCD Builder is available in two configurations:

- A standalone client configuration that be executed from the user's desktop
- A Web Client

General description

The first step is to call the VCD Designer who processes embedded tasks as described in section 3.11.

Step 2 of the VCD Builder delivers the evidences themselves.

In step 3, the VCD is build. The VCD Builder can manage to have uncompleted VCD and so lets the Economic Operator proceed in subsequent phases.

A variant gives the possibility to the Economic Operator to upload the VCD Skeleton without calling the VCD Designer.

The VCD Builder component depends on the following concrete models, services and modules:

- **Models:** Various models, representing the different data structures derived from the corresponding VCD Schemas that represent the VCD Skeleton, and the VCD Package are used throughout the application.
- **Services:** One external service, provided by the VCD Designer component, is used by the VCD Builder component in order to request information from the EVS.

3.12.2 Implementation environment

<i>Technology / Tool</i>	<i>Version</i>	<i>Description</i>
NetBeans Integrated Development Environment (IDE)	6.9.1	NetBeans Java EE IDE
Java Development Kit (JDK)	JDK SE 6	Java Development Kit Standard Edition
Apache Maven	2.2.1	Build and Dependency Management
Equinox OSGi container	3.6.0	Implementation of the OSGi R4 core framework
Vaadin web framework	6.4.5	RIA framework built on GWT-based widgets
JAXB	RI 2.1	Java XML binding
Axis2 JAX-WS	RI 2.1.3	Java for XML Web Services. Apache Axis2 is a core engine for Web services in Java.

3.12.3 Deployment environment

The VCD Builder package is distributed as a set of OSGi software bundled with the VCD Designer component. These two components share the same deployment environment, in particular the VCD Bootstrap component, which is a customized OSGi container based on Equinox.

The aforementioned set of software bundles consists of JAR and WAR files. Concerning the WAR file, it is deployed on a Jetty Application Server that also runs as a separate OSGi bundle inside VCD Bootstrap.

3.12.4 Reference implementation

The reference application of the VCD Builder is available online deployed on the embedded Jetty Application Server bundle of VCD Bootstrap component. Version 1.0 of the VCD Builder supports version 1.0 of the VCD Schema.

Item	Resource	Version
WAR	nvs-ui-1.0.0.RELEASE.war	1.0.0
JAR	nvs-assembler-1.0.0.RELEASE.jar nvs-invoker-1.0.0.RELEASE.jar nvs-model-1.0.0.RELEASE.jar nvs-utils-1.0.0.RELEASE.jar nvs-validator-1.0.0.RELEASE.jar	1.0.0
Embedded Application Server	jetty-6.1.25.jar	6.1.25
VCD Schema	VCD Schema	1.0

In order to install the VCD Builder, you just install the VCD Bootstrap software component (following the installation instructions²⁰ of the VCD Bootstrap). Once the customized container is installed and running, the VCD Builder application can be accessed via a web browser²¹ by logging in and choosing the different options.

3.12.5 Outlook on functionality extensions

Functional extensions to version 1.0 of the VCD Builder will be implemented within future software releases. These future versions will be released in incremental cycles and will cover the following functionality:

- Provision of an interface for the Contracting Authorities to be able to input the initial tender structure themselves, making them available for each prospective Economic Operator that wants to use the NVS to design and then build a VCD for this particular tender
- Support Compound VCD scenarios among Economic Operators from different countries and/or using different NVS providers as explained in chapter 2.3.
- Multi-lingual user interface: Ability to choose between different languages to visualize user interface elements in other languages (e.g. French, Greek).
- Multiple VCD Package support.
- Database and file management support in order to be able to store, view and re-use VCD
- UI improvements and adoption of the unified GUI, as all VCD components.
- Digital Signatures support (future CA interface only, if applicable) .

²⁰ <http://coeus.ds.unipi.gr/wiki/display/VCDWIKI/Bootstrap+Packaging+And+Configuration>

²¹ <http://Server-Name/vcd-ui>

3.13 VCD Viewer

Figure 15 provides an overview about the components of the VCD viewer.

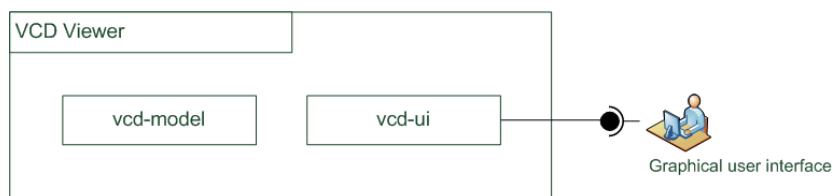


Figure 15: Components of the VCD Viewer

3.1.1 Purpose of the VCD Viewer

The VCD Viewer is a web application used to visualize the content of VCD Packages which can be uploaded via the GUI. The application extracts all documents and information from the container and the meta-data files and presents it via a web browser. The Viewer is developed to support Contracting Authorities in visualizing the content of received tenders, as well as evaluating the Economic Operators with respect to the required criteria and submitted evidence documents. Economic Operators may also use this tool to visualize the content of created VCDs before submitting them to a Contracting Authority.

The following features have been implemented in the current version:

- Upload of VCD Container zip files
- Visualization of VCD Package information (extracted from the VCD Package meta-data file)
- Open/download of attached documents
- Visualization of the tenderer structure, provided as a clickable tree to view the content of single VCDs
- Visualization of VCD information (extracted from the VCD meta-data file)
- Visualization of requested criteria (including different legislation levels: European regulation, Economic Operator national regulation, Contracting Authority national regulation)
- For each criterion, information about the provided evidences are shown
- Possibility to filter evidences according to the legislation level
- Possibility to open/download evidence documents as well as additional attached files or documentation

3.1.2 Implementation environment

<i>Technology / Tool</i>	<i>Version</i>	<i>Description</i>
Eclipse Integrated Development Environment (IDE)	Galileo (Version: 1.2.1)	Eclipse Java EE IDE for Web Developers
Java Development Kit (JDK)	JDK SE 6	Java Development Kit Standard Edition
Google Web Toolkit (GWT)	2.0.4	Google software development kit
Maven Eclipse Plugin (m2eclipse)	2.0	Maven Integration for Eclipse
JAXB	2.0	Java XML binding

3.1.3 Deployment environment

The VCD Viewer is distributed as a web application, in particular as a web application archive (WAR). This WAR can be deployed in any suitable Java Servlet Container, such as the Apache Tomcat application server.

3.1.4 Reference implementation

The reference application of the VCD Viewer is available online²² deployed on an Apache Tomcat application server. Version 1.0 of the VCD Viewer supports version 1.0 of the VCD Schema. It consists of the following software components:

<i>Item</i>	<i>Resource</i>	<i>Version</i>
WAR	VCDViewer-1.0.war	1.0
Application Server	Apache Tomcat ²³	7.0
VCD Schema	VCD Schema	1.0

To install the VCD Viewer on the Apache Tomcat application server, the server first needs to be installed (following the installation instructions on the Apache Tomcat website). Once the server is installed and running, the VCDViewer-1.0.war file has to be copied to the directory “\apache-tomcat-7.0.0\webapps\”. After restarting the server, the VCD Viewer application can be accessed via a web browser.

3.1.5 Outlook on functionality extensions

Functional extensions to version 1.0 of the VCD Viewer will be implemented within future software releases. These future versions will be released in incremental cycles and will cover the following functionality:

- Multi-lingual user interface: Ability to choose between different languages to visualize user interface elements in other languages (e.g. French, German).
- User authentication: Access control to the web interface.
- Integration with other VCD software components: Deliver software components as an integrated software bundle.
- Validation of digital signatures and visualization of validation results.

²² See <http://fgwimmer2008.uni-koblenz.de:8181/VCDViewer/>

²³ See <http://tomcat.apache.org/download-70.cgi>

4 Software components planned in the next period to extend the VCD solutions

The components described in chapter 3 provide basic reference implementations to run the VCD solution with pilots. Yet further functionality is scheduled to be added. Therefore, this chapter provides an outlook on future software components, which will be included in the VCD solution in the next phase. Table 3 provides an overview of the planned components to be developed until April 2011. Table 4 provides more detailed information on these components.

Table 3 - Building blocks planned for release by April 2011 or later (*)

Building Block	Description	Decentral/ Central	Specification	Software as Open Source²⁴
VCD Verifier	The VCD Verifier supports multiple digital verify functions for any electronic document being part of a VCD instance or individually submitted	Decentral	D2.2 Specification & components – section 7.2	MPL/EUPL
VCD Signer	The signature component will provide technical means to digitally sign VCD Packages and the included XML meta-data files, as well as validate existing signatures. It will provide interfaces so that other components can directly access its functionality. The VCD Signer performs multiple digital signature functions for electronic structured documents according to the XAdES-BES for UBL specifications	Decentral	D2.2 Specification & components – section 7.2	MPL/EUPL
VCD Identity Validator	The VCD Identity Validator implements the digital identity verification based on XKMS infrastructure. It is a client component to invoke the XKMS server functions	Decentral	Extension to D 2.2	MPL/EUPL
VCD BusDox Connector	The Connector implements a client application to Send/Receive VCD instances over the PEPPOL BusDox	Decentral	D 2.2 indicates the WP 8 infrastructure use, but no	MPL/EUPL

²⁴ Available in full as open source under e.g MPL1.1 license and or EUPL

	infrastructure		specification was in D 2.2.	
Manual VCD Skeleton Editor	<p>It shall support those Economic Operators in creating a VCD Package whose national procurement legislations are not part of the EVS ontologies, i.e. the legal rule set to derive evidence information from requested criteria is not available. The mapping of the criteria to prove (as stated in the call for tender) and the suitable national evidences to be provided by the Economic Operator has to be done by the Economic Operator, eCertis (respectively other informations sources) acting as information base.</p> <p>The Manual VCD Skeleton Editor is linked with the VCD Builder. After the VCD structure is generated, the Economic Operator can use the existing VCD Builder to compile the VCD artefact.</p>	Decentral	Extension of D 2.2 specifications	MPL/EUPL
Criterion Editor (*)	<p>Intended to support Contracting Authorities in providing the selection and exclusion criteria for call for tender in a structured and standardised way that can be automatically computed via the VCD solution. It will support Contracting Authorities in creating CfTs in more efficient way and will at the same time simplify the compilation of a VCD through automatic support via the VCD solution. For this a new C-Skeleton will be introduced (to be added to the skeletons introduced in 2.2.2)</p>	Central / decentral	Extension to D 2.2. specification	MPL/EUPL
NVS (Italy)	The national implementation of VCD system based on OS platforms compatible with the OASIS-CMIS domain model	Decentral		MPL/EUPL

Table 4: Details of planned future components to extend the VCD solution

Component	Description	Belonging to Main Building block	Decentral/Central	Specification in D 2.2	Software as Open Source²⁵
API for digital verify operations	implements: – digital verify on XAdES-BES, XAdES-T, signed VCD and VCDPackage XML files; – digital verify AdES document format: XAdES-BES, XAdES-T, CAdES-BES, CADES-T, PADES part2 and PKCS7 – digital identity verification based on XKMS infrastructure; – digital reporting over structured (XML) formats; – digital reporting over human readable formats; – digital identity verification for bearer of X509 certificates (XKMS functions)	VCD Verifier	Decentral	Extension of D 2.2, no specification available yet	MPL/EUPL
API for digital sign operations	Implements: – preparation for digital signing on XAdES-BES, XAdES-T for UBL. The “sign” action relies on the PKCS#11 interface offered by different cryptographic tokens. The digital signature conforms to UBL-XAdES-Profile.	VCD Signer	Decentral	Extension of D 2.2, no specification available yet	MPL/EUPL
API for identity validation	implementats: – digital identity verification based on XKMS infrastructure; – client component to invoke the XKMS server functions.	VCD Identity Validator	Decentral	Extension of D 2.2, no specification available yet	MPL/EUPL

²⁵ Available in full as open source under e.g MPL1.1 license and or EUPL



API for hashing	Implements: hashing functions to affect any candidate electronic document regardless of its format	VCD Verifier VCD Signer	Decentral	Extension of D 2.2, no specification available yet	MPL/EUPL
API for reporting	Implements: digital reporting of the verification results according to pre-defined templates and structured documents for metadata	VCD Verifier	Decentral	Extension of D 2.2, no specification available yet	MPL/EUPL
VCDTransportationInterface	Implements: a client application for sending/receiving instances of the VCD container over the BusDox infrastructure	VCD BusDox Connector	Decentral	Extension of D 2.2, no specification available yet	MPL/EUPL
VCD_Builder (Italy)	Creates VCD and VCDPackage document instances interacting with companion components : <ul style="list-style-type: none"> • VCD Container Creator (Italy) • DocumentManager • VCD Verifier • User Interface (Italy) 	NVS (Italy)	Decentral	Deliverable D 2.2 + Attachment B	MPL/EUPL
VCD Container Creator (Italy)	Creates VCD Container instances by merging individual document instances	NVS (Italy)	Decentral	Deliverable D 2.2 + Attachment B	MPL/EUPL
Document Manager	Manages all the relevant records and electronic documents qualified as candidate parts of a VCD Container	NVS (Italy)	Decentral	Deliverable D 2.2 + Attachment B	MPL/EUPL
UserInterface	Manages the services invocation and the system operations for the primary system users as: <ul style="list-style-type: none"> • Economic Operators • Contracting Authorities • National Service Officers 	NVS (Italy)	Decentral	Deliverable D 2.2 + Attachment B	MPL/EUPL

5 Summary and Outlook

This document provides an overview about the VCD components implemented in PEPPOL WP2 according to the specifications designed in phase 2 of the project (cf. D 2.2). Phase 3 of the PEPPOL VCD work focused on the software implementation of the major components to realize the VCD solution. The major software components (building blocks) implemented in phase 3 of WP 2, which are briefly described in the context of the VCD solution as well as in regards to their main functionality and composition of software modules, are:

1. **The European VCD System (EVS):** This central VCD building block provides a decision support system for building a VCD skeleton conform to the terms of a call for tender. The EVS hosts the ontologies, provides the reasoning and makes decision support accessible via system and user interfaces. The EVS generates a VCD Skeleton Container, i.e. it sets out the basic structure for the VCD artefact that is further used by the NVS or the Manual VCD Builder. The EVS currently serves Economic Operators and Contracting Authorities in Austria, France, Germany, Greece and Italy.
2. **The National VCD System (NVS):** This VCD building block enables Economic Operators to compile a VCD Container. PEPPOL WP 2 has implemented reference implementations for the NVS consisting of the VCD Designer and the VCD Builder. The VCD designer provides an interface to the EVS and enables a direct interaction in order to create the VCD skeleton in case the respective national ontologies are available. The VCD Builder enables the Economic Operator to fill the VCD skeleton with evidences and other information about the Economic Operator in order to create a full VCD container. The NVS may be deployed in different variants and scenarios. It can be provided as a central service within a European country but it can also be used as stand alone application. Also eTendering platform provides or prequalification bodies can use the NVS by integrating it into their existing solutions.
3. **Manual VCD Builder:** This building block serves Economic Operators in European countries, where no other solution is offered (neither 1. nor 2.). This building block consists of the VCD Builder and a Manual VCD Skeleton Editor. The latter is used to generate the initial structure of a VCD container artefact, i.e. it serves as intermediate solution until the country has implemented the national procurement laws in the EVS (see 1.).
4. **VCD Viewer:** This building block is especially developed for Contracting Authorities to view and navigate through the content of any VCD Container without having the possibility to edit or change content. It can also be used by Economic Operators to view content that needs not to be changed any more.

The software for the individual components forms the major part of this deliverable. It is provided in Version 1.0. The report at hand serves as supplement to provide a textual outline of the work performed and delivered.

As indicated in chapter 2, the specifications of D 2.2 will be adjusted further. Likewise as indicated in chapters 3 and 4, the software implementations will be amended and extended with functionality and further components as planned (e.g. the VCD Signer, VCD Verifier, VCD BusDox Connector, Manual VCD Skeleton Editor) will be provided in next releases over the coming months and until the end of the PEPPOL project. Versions 2.0 of VCD building blocks will mainly aim to provide further functionality towards an integrated VCD solution considering all relevant aspects as brought forward in the VCD vision (cf. Deliverable D 2.1).

As the different VCD building blocks are providing a broad set of functionality to the user and are, often in combination, applicable in diverse organisational and technical settings, most of these building blocks (e.g. VCD Builder, VCD Designer, EVS, VCD Viewer) are implementing their own individual



graphical user interface. In consequence, information blocks repeating themselves (e.g due to the using of the same schema derivatives) are currently presented through different GUIs throughout the diverse implementations. In order to support the user with a standardized GUI, WP 2 will harmonize these different interfaces in the course of the next implementation phases according to reference design templates, still leaving space for the different (national) characteristics.

Other activities planned in the next period regard for example the ability to create compound VCDs (cf. initial process described in section 2.3), multi-lingual features or XML validation.

Beside technical aspects introduced above, the preparation and performance of real public tenders using the VCD solution will continue in the production pilot phase of PEPPOL. This will also provide important feedback to further improve the VCD solutions and to adequately evaluate the results implemented in Version 1.0. This approach will lead to a good reflection of results and enable great exchange of best practice at the end of this project.

Annex I: Overview of VCD Schema specifications in V 1.0

According to the summary of updates given in chapter 2.1 and 2.2, the table below outlines the specification of the XML schemas for the generation of document skeletons and full document instances. The schema version that gives support to the current instances generation is Version 1.0.

VCDSchema: summary of folders and companion files

<http://coeus.ds.unipi.gr/hg/VCDSchema/file/cfec36638255/>

Skeleton Sets:

- T-Skeleton: This schema contains an interoperable standard structure to communicate a tenderer structure (single tenderer, bidding consortium, subcontractors; all with legal form of individual Economic Operators). This schema structures the data passed from the NVS to the EVS when initializing the decision support process.
- TC-Skeleton: This schema extends the T-Skeleton with the criteria laid out in a call for tender or contract notice by the Contracting Authority. The specific skeleton maps out the exact criteria to be applied to each single Economic Operator. This schema structures the data passed from the EVS to the NVS when suggesting criteria for a specific tenderer setting and the data passed from the NVS to the EVS as input for suggesting possible evidences.
- TCE-Skeleton: This schema represents the full structure of a VCD artefact generated by the European VCD System and passed either to the NVS or to the Economic Operator. It indicates for each Economic Operator and selected criterion, which evidences the party might provide to prove that selection or exclusion criterion based on the party's role in the tenderer structure and its legal form. This schema structures the data passed from the EVS to the NVS when suggesting possible evidences.

T-Skeleton/	maindoc/	WP2-ContextSpecificData-1.xsd WP2-VirtualCompanyDossier-1.xsd WP2-VirtualCompanyDossierPackage-1.xsd
	common/	CCTS_CCT_SchemaModule-2.0.xsd CodeList_CurrencyCode_ISO_7_04.xsd CodeList_LanguageCode_ISO_7_04.xsd CodeList_MIMEMediaTypeCode_IANA_7_04.xsd CodeList_UnitCode_UNECE_7_04.xsd UBL-CommonExtensionComponents-2.0.xsd UBL-CoreComponentParameters-2.0.xsd UBL-ExtensionContentDatatype-2.0.xsd UBL-QualifiedDatatypes-2.0.xsd UnqualifiedDataTypeSchemaModule-2.0.xsd WP2-CommonAggregateComponents-2.xsd WP2-CommonBasicComponents-2.xsd
TC-Skeleton/	maindoc/	WP2-ContextSpecificData-1.xsd WP2-VirtualCompanyDossier-1.xsd WP2-VirtualCompanyDossierPackage-1.xsd
	common/	CCTS_CCT_SchemaModule-2.0.xsd CodeList_CurrencyCode_ISO_7_04.xsd CodeList_LanguageCode_ISO_7_04.xsd CodeList_MIMEMediaTypeCode_IANA_7_04.xsd CodeList_UnitCode_UNECE_7_04.xsd UBL-CommonExtensionComponents-2.0.xsd

		UBL-CoreComponentParameters-2.0.xsd UBL-ExtensionContentDatatype-2.0.xsd UBL-QualifiedDatatypes-2.0.xsd UnqualifiedDataTypeSchemaModule-2.0.xsd WP2-CommonAggregateComponents-2.xsd WP2-CommonBasicComponents-2.xsd
TCE-Skeleton/	maindoc/	WP2-ContextSpecificData-1.xsd WP2-VirtualCompanyDossier-1.xsd WP2-VirtualCompanyDossierPackage-1.xsd
	common/	CCTS_CCT_SchemaModule-2.0.xsd CodeList_CurrencyCode_ISO_7_04.xsd CodeList_LanguageCode_ISO_7_04.xsd CodeList_MIMEMediaTypeCode_IANA_7_04.xsd CodeList_UnitCode_UNECE_7_04.xsd UBL-CommonExtensionComponents-2.0.xsd UBL-CoreComponentParameters-2.0.xsd UBL-ExtensionContentDatatype-2.0.xsd UBL-QualifiedDatatypes-2.0.xsd UnqualifiedDataTypeSchemaModule-2.0.xsd WP2-CommonAggregateComponents-2.xsd WP2-CommonBasicComponents-2.xsd
VCD documents set: The subsequent XML schema files are required for the generation of VCD and VCD package document instances.		
common/		CCTS_CCT_SchemaModule-2.0.xsd CodeList_CurrencyCode_ISO_7_04.xsd CodeList_LanguageCode_ISO_7_04.xsd CodeList_MIMEMediaTypeCode_IANA_7_04.xsd CodeList_UnitCode_UNECE_7_04.xsd UBL-CommonExtensionComponents-2.0.xsd UBL-CoreComponentParameters-2.0.xsd UBL-ExtensionContentDatatype-2.0.xsd UBL-QualifiedDatatypes-2.0.xsd UnqualifiedDataTypeSchemaModule-2.0.xsd WP2-CommonAggregateComponents-2.xsd WP2-CommonBasicComponents-2.xsd
maindoc/		WP2-ContextSpecificData-1.xsd WP2-VirtualCompanyDossier-1.xsd WP2-VirtualCompanyDossierPackage-1.xsd
README.txt		
T-Skeleton.xsl		
TC-Skeleton.xsl		
TCE-Skeleton.xsl		
skeleton-patcher.sh		

The current generation of VCD document instances comes from the merge of specific content, in line with the specific instance to be generated and some content values taken from a pregenerated vocabulary organized into a set of lists. The table below reports the specifications available and necessary for VCD instance generation.

VCD Code Lists: summary of folder and companion files
<http://coeus.ds.unipi.gr/hg/CodeLists/file/d5f7b2046264/gc/>

<p>VCD Code Lists: The lists are organized into a twin set of</p> <ul style="list-style-type: none"> • VCD specific Code Lists, and • UBL shared Code lists. 	
<p>VCDCodeLists/</p>	<p>CharacterSetCode.gc CountrySubentityCode.gc DocumentTypeCode.gc LocaleCode.gc MaturityLevelCode.gc OntologyDomainCode.gc README.doc ValidateResultCode.gc</p>
<p>ublSharedCodeLists/</p>	<p>BinaryObjectMimeCode-2.0.gc CountryIdentificationCode-2.0.gc README.doc</p>
<p>VCD_CodeLists_R8.xls</p>	