Specification

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

Virtual Company Dossier
Use cases and other UML representations of the National VCD system

Revision: 1.01

Authors:

Ansgar Mondorf (UKL)
Maria Wimmer (UKL)

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
P	**Public**	**X**
C	**Confidential, only for members of the consortium and the Commission Services**	

Revision History

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

Contributors

Organisations

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

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

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

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

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

PEPPOL.AT, Austria

UKL (University of Koblenz-Landau), Germany

Persons

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

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

Table of Contents

1. VCD service specifications: Overall modelling

The overall modelling strategy for the systems that will get local and centralized implementations adopts the multidimensional design possibilities given by the UML language through the coordinated production of Use-Case, Activity, Package and Sequence diagrams.

1.1 VCD service specifications: Overall modelling

This basic overview and the following diagrams will be referenced by the following inter-related Use Cases, all making part of this specification document:
CompileVCD; CompileVCDPackage; ViewVCD; SignVCD; SignVCDPackage; VerifyVCDSignature; VerifyVCDPackageSignature; AssembleVCDContainer; ManageVCDContainer; CreateVisibleVCDPackage; AccessData; AccessDocuments; CreateDataAndDocuments

The packaging strategy follows a criterion that differs from the Use Cases breakdown. It gives emphasis to the implementation possibilities and to the sharing strategy that the consortium will undertake. On the other side the Use-Cases description covers the functional perspective that the specification must provide. The summary of the supported use cases is given here below:

1.1.1 Use-Case Group “VCD Builder and User Interface”
1.1.2 Use-Case group “User Identity and Access management incl. authentication”
1.1.3 Use-Case-group “Signature Functions”
1.1.4 Use-Case-group “Signature validation interface”
1.1.5 Use Case group “VCD System”
1.1.6 Use-Case Group “VCD Builder”
1.1.7 Use-Case Group “VCD Container Creator”
1.1.8 Use-Case group “VCD transportation interface incl. authentication”
1.1.9 Use-Case “Interface to VCD dictionary (incl. Code lists, schemas...)”
1.1.10 Use-Case Group “Interface to PEPPOL registries”
1.1.11 Use-Case Group “Interface to European VCD service”
1.1.12 Use-Case-group “interface to notification systems”
1.1.13 Functional specification
1.3.1 VCD Viewer
1.4.1 Function “Display VCD package” – Alternative 1: Embedded XSLT
1.4.2 Overall description for the function ”Display VCD package” – Alternative 2: Desktop-Application
1.4.3 Function “Display VCD package” – Alternative 3: Web-Application
1.4.4 Understandable and easily navigable representation of the output
1.4.5 Uploading of VCD package (Web application)
1.4.6 Input of path to VCD package (Desktop application)
1.4.7 Unpacking of VCD package
1.4.8 Transformation of VCD package content/Generation of output
1.5.1 Comparison of approaches and final decisions
1.5.2 Data handling to be implemented by a VCD service provider

1.1.1 Use-Case Group “VCD Builder and User Interface”

The compilation activity for generating a VCD Container instance is considered a very specific task with low generalization options for the implementation. This is due to the strong impact made by this activity over the existing organizations, the process models, and technologies they will adopt. The peculiarity of the compilation activity affects the way each organization will take to manage the information and document set along with the generation of values for the VCD metadata. The present specification will show a reference design model that outlines the basic functions. The actual implementation will be a matter affecting the WP2 piloting partners. The specification of the compiling functions gets full coverage in chapters 8.1.8 within the VCD Builder section.

1.1.2 Use-Case group “User Identity and Access management incl. authentication”

About every system requests authentication and so a user identity and access management system. This is the goal of this Use-Case.

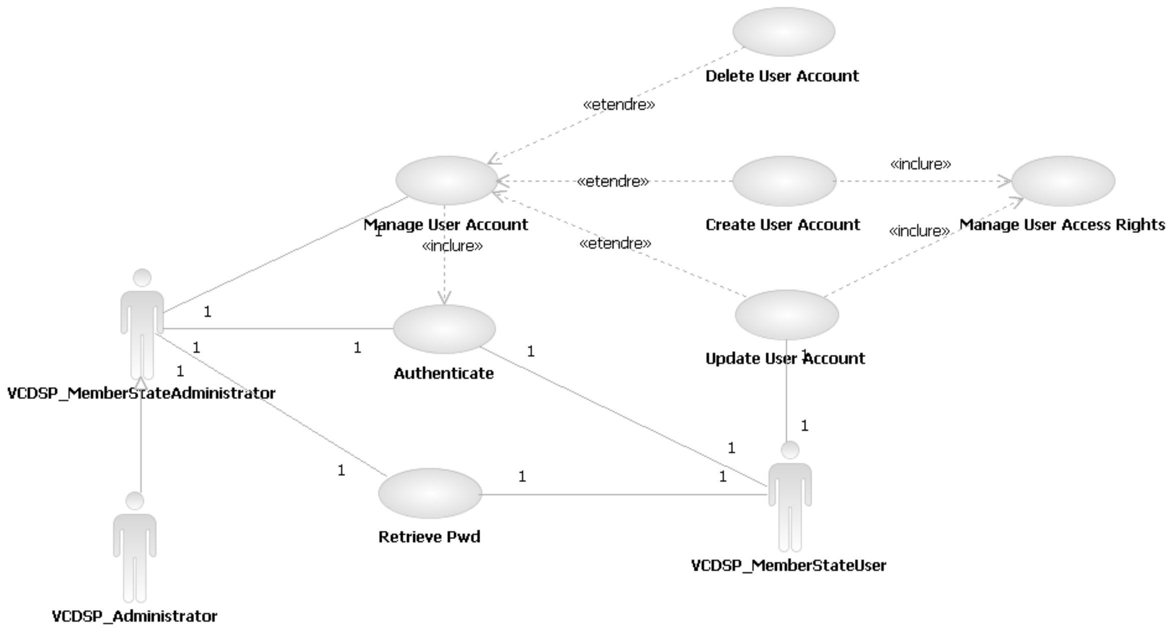

Figure 1-1 : Use-case of user identity and Access management including authentication

This Figure describes the big picture for user access and role management, including authentication, management tasks of user accounts (create, delete, update), management of user access rights, password retrieval by the (Member State) user.

Use-Case “Manage User Account”

Manage User Account will let a “VCDSP_Administrator” (EU level) /or a “VCDSP_MemberStateAdministrator” to manage a User Account, or a “VCDSP_MemberStateUser” to manage his own account.

Input-Output data:

Activity	Input	Output	User/System
Authenticate	Login, password	None	“VCDSP_Administrator” (EU level) / “VCDSP_MemberStateAdministrator” and “VCDSP_MemberStateUser”
Delete User Account	User name	Delete_UserAccount_Request or Error response	UIAManager
Create User Account	User name	Create_UserAccount_Request or Error response	UIAManager
Update User Account	User name or none	Update_UserAccount_Request or Error response	UIAManager
Manage User Access Rights	User name	Manage_UserAccess_Rights or Error response	UIAManager
Error response	Error response	None	UIAManager

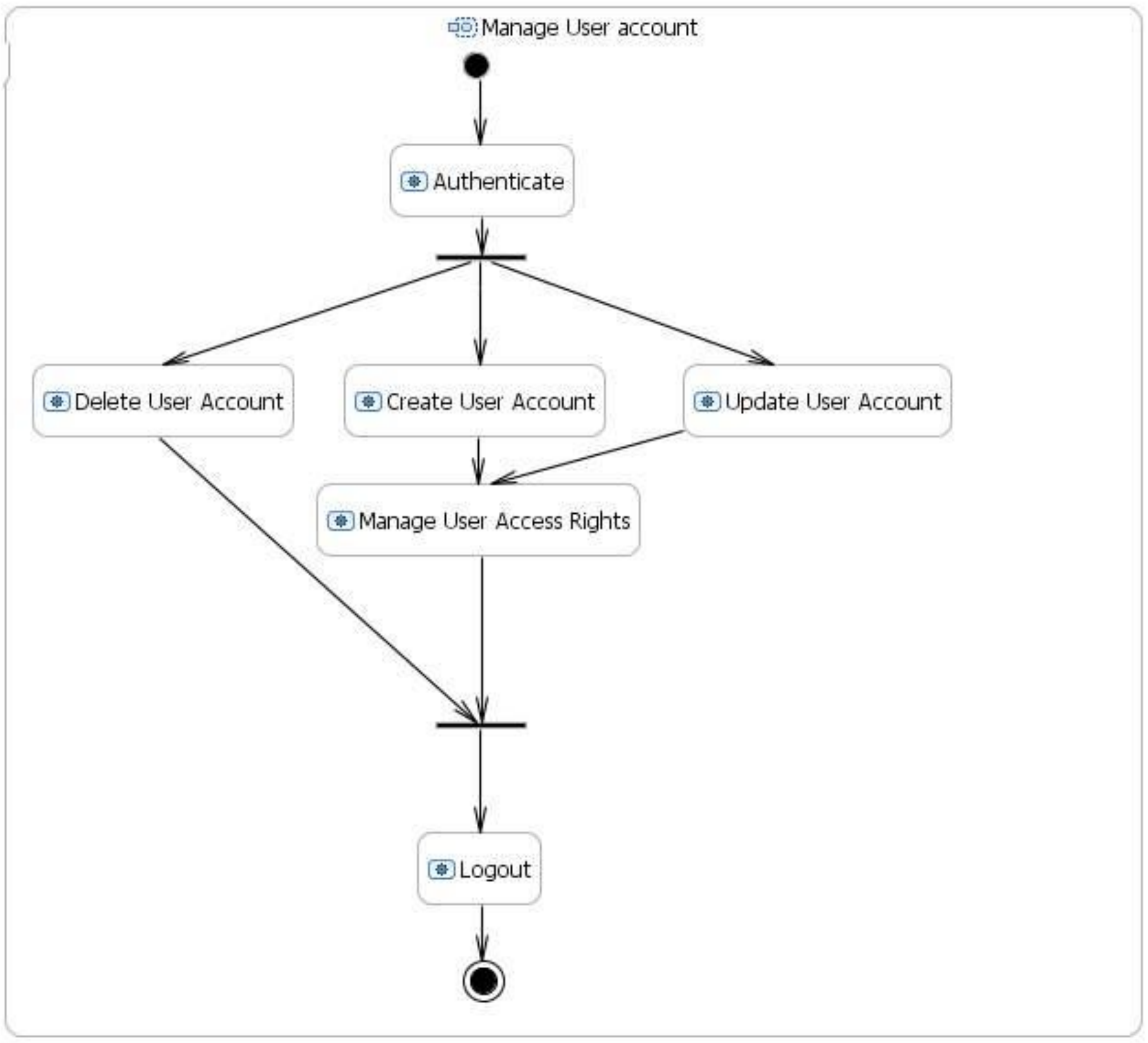

Figure 1-2 : Activity diagram of Manage User account including authentication

Aspect	Description
Objective	Allowing access to the set of sub use cases
Results (postconditions in case of success)	Directory hosting user accounts is updated
Precondition	The actor is authenticated
Postcondition in case of failure	Error message is displayed
Actor(s)	“VCDSP_Administrator“ (EU level) and “VCDSP_MemberStateAdministrator” (MS level)
Initiating event	
Description of interaction procedure with VCD Service (standard run)	
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	UCs “Delete User Account“ / “Create User Account“ / “Update User Account“ / ”Manage User Access Rights”

Table 1-1: Manage User Account

Use-Case “Delete User Account”

Delete User Account will let a “VCDSP_Administrator” (EU level) /or a “VCDSP_MemberStateAdministrator” to delete an existing User Account

Input-Output data:

Activity	Input	Output	User/System
Request User Name	User Name	None	“VCDSP_Administrator” (EU level) / “VCDSP_MemberStateAdministrator”
Verify User Name existence	User name	Delete_UserAccount_Request or Error response	UIAManager
Error response	Error response	None	UIAManager

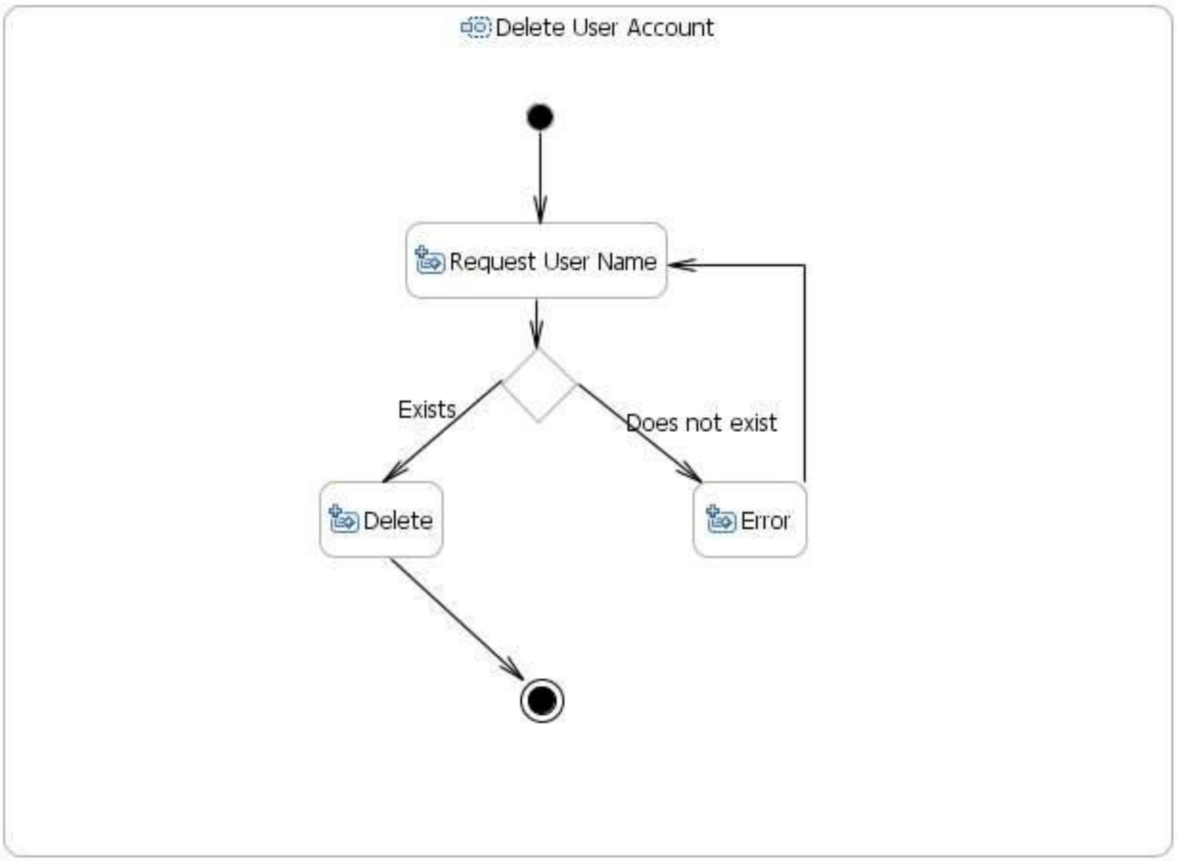

Figure 1-3 : Activity diagram of Delete User account

Aspect	Description
Objective	Deleting a “VCDSP_MemberStateAdministrator” or “VCDSP_MemberStateUser” account

Results (postconditions in case of success)	Directory entrance hosting user account is deleted The owner of the account deleted is notified
Precondition	The actor acting is authenticated The user account to be deleted exists (!)
Postcondition in case of failure	Error message is displayed
Actor(s)	“VCDSP_Administrator” (EU level) and “VCDSP_MemberStateAdministrator” (MS level)
Initiating event	
Description of interaction procedure with VCD Service (standard run)	The actor selects the account to be deleted The actor confirms its choice The account selected is deleted
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	
LEG/ORG	
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-2: Delete User Account

Use-Case “Create User Account”

Create User Account will let a “VCDSP_Administrator” (EU level) /or a “VCDSP_MemberStateAdministrator” to create a new User Account.

Input-Output data:

Activity	Input	Output	User/System
Request User Name and informations	User Name and informations	None	“VCDSP_Administrator” (EU level) / “VCDSP_MemberStateAdministrator”
Verify User Name existence	User name	Create_UserAccount_Request or Error response	UIAManager
Error response	Error response	None	UIAManager

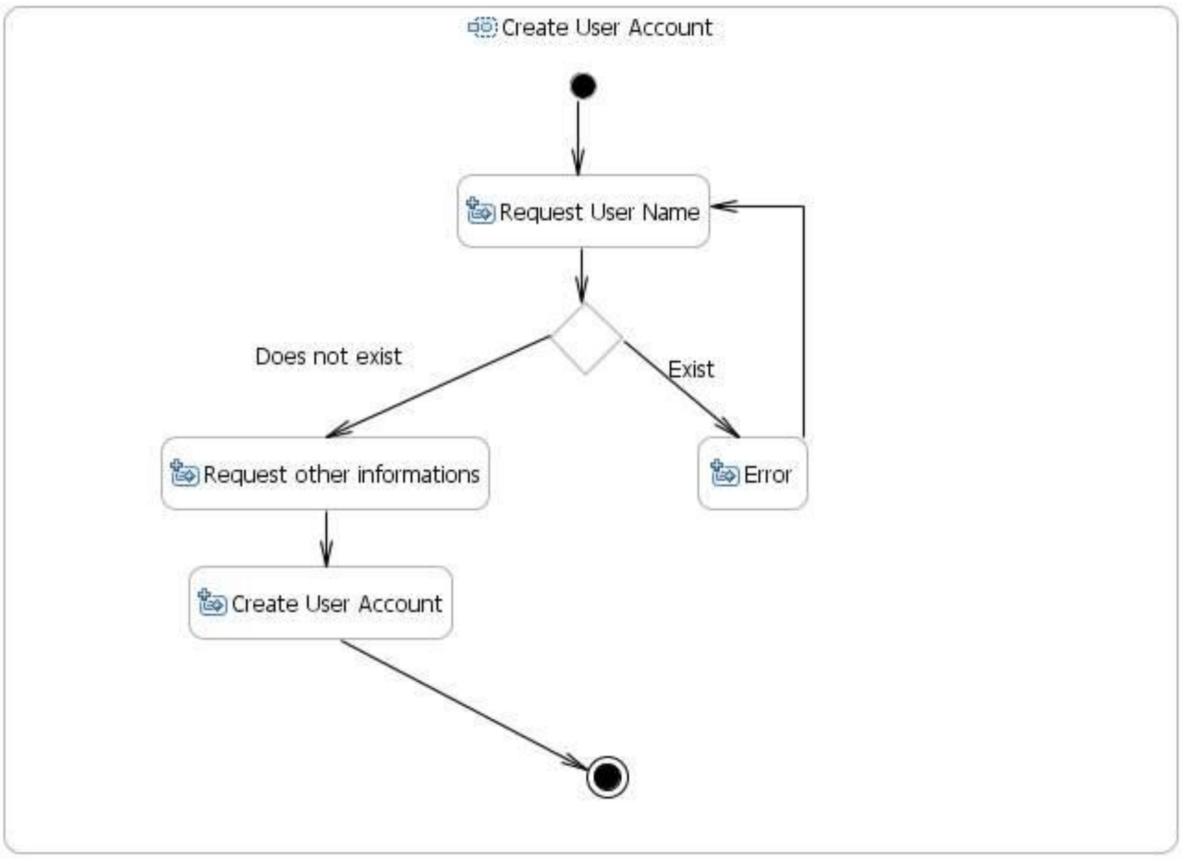

Figure 1-4 : Activity diagram of Create User Account

Aspect	Description
Objective	Creating a "VCDSP_MemberStateUser" / "VCDSP_MemberStateAdministrator" account
Results (postconditions in case of success)	Directory entrance hosting user account is created The owner of the account created is notified
Precondition	The actor acting is authenticated The user account to be created does not exist (!)
Postcondition in case of failure	Error message is displayed
Actor(s)	"VCDSP_Administrator" (EU level) and "VCDSP_MemberStateAdministrator" (MS level)
Initiating event	
Description of interaction procedure with VCD Service (standard run)	The actor fills in the form The actor confirms its choice The account selected is created
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	
LEG/ORG	
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-3: Create User Account

Use-Case "Update User Account"

Update User Account will let a "VCDSP_Administrator" (EU level) /or a "VCDSP_MemberStateAdministrator" to update an existing User Account

Input-Output data:

Activity	Input	Output	User/System
Request User Name and informations	User Name and informations	None	"VCDSP_Administrator" (EU level) / "VCDSP_MemberStateAdministrator" and "VCDSP_MemberStateUser"
Verify User Name existence	User name	Update_UserAccount_Request or Error response	UIAManager
Error response	Error response	None	UIAManager

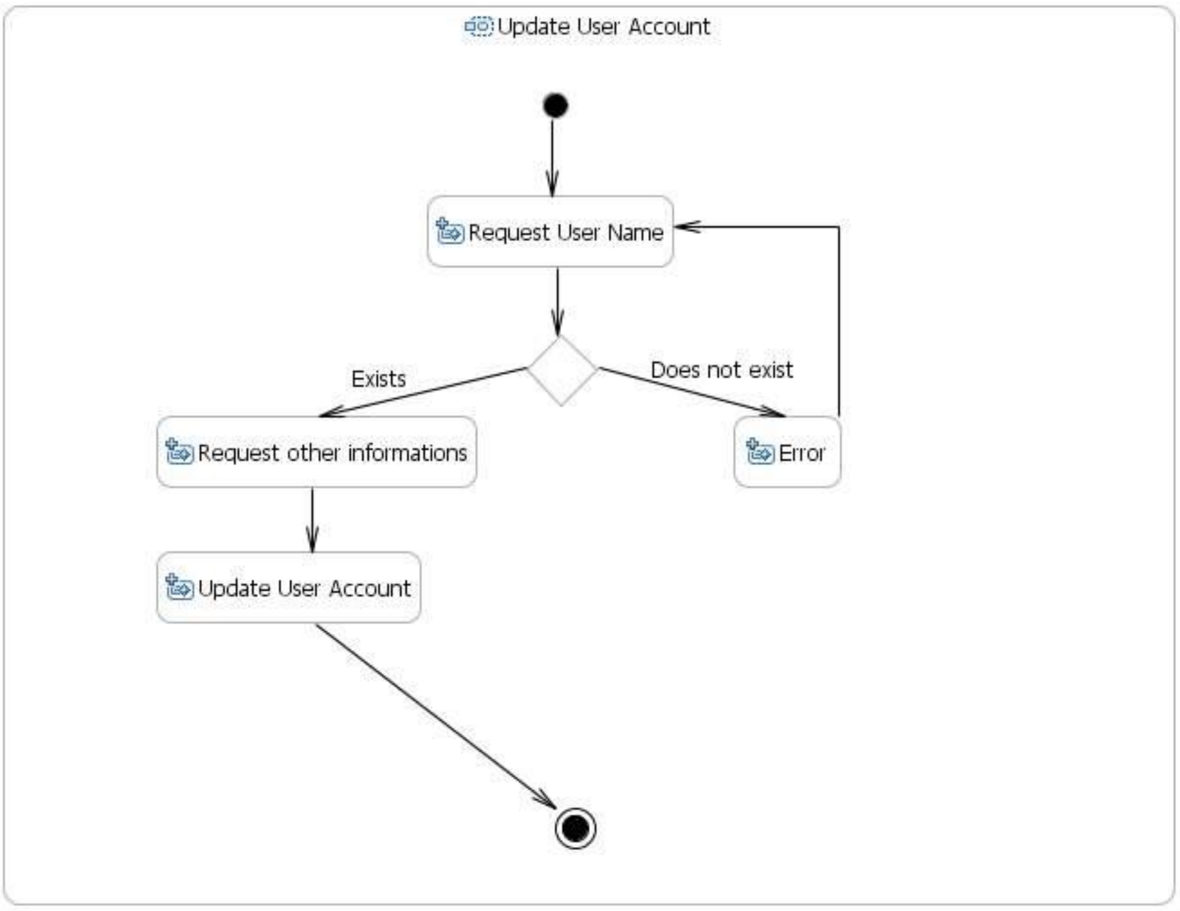

Figure 1-5 : Activity diagram of Update User Account

Aspect	Description
Objective	Updating a "VCDSP_Administrator" / "VCDSP_MemberStateAdministrator" / "VCDSP_MemberStateUser" account
Results (postconditions in case of success)	Directory entrance hosting user account is updated The owner of the account created is notified
Precondition	The actor acting is authenticated The user account to be updated exists (!)
Postcondition in case of failure	Error message is displayed
Actor(s)	"VCDSP_Administrator" (EU level) / "VCDSP_MemberStateAdministrator" and "VCDSP_MemberStateUser" (MS level)
Initiating event	
Description of interaction procedure with VCD Service (standard run)	The actor fills in the form The actor confirms its choice The account selected is updated
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	
LEG/ORG	
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-4: Update User Account

Use-Case “Manage User Access Rights”

Manage User Access Rights will let a “VCDSP_Administrator” (EU level) /or a “VCDSP_MemberStateAdministrator” to manage User Access Rights of an existing User Account

Input-Output data:

Activity	Input	Output	User/System
Request User Name	User Name	None	“VCDSP_Administrator” (EU level) / “VCDSP_MemberStateAdministrator”
Verify User Name existence	User name	Display User Access Rights form or Error response	UIAManager
Fill form	User Access Rights information	Update_User Access_Rights or Error response	UIAManager
Error response	Error response	None	UIAManager

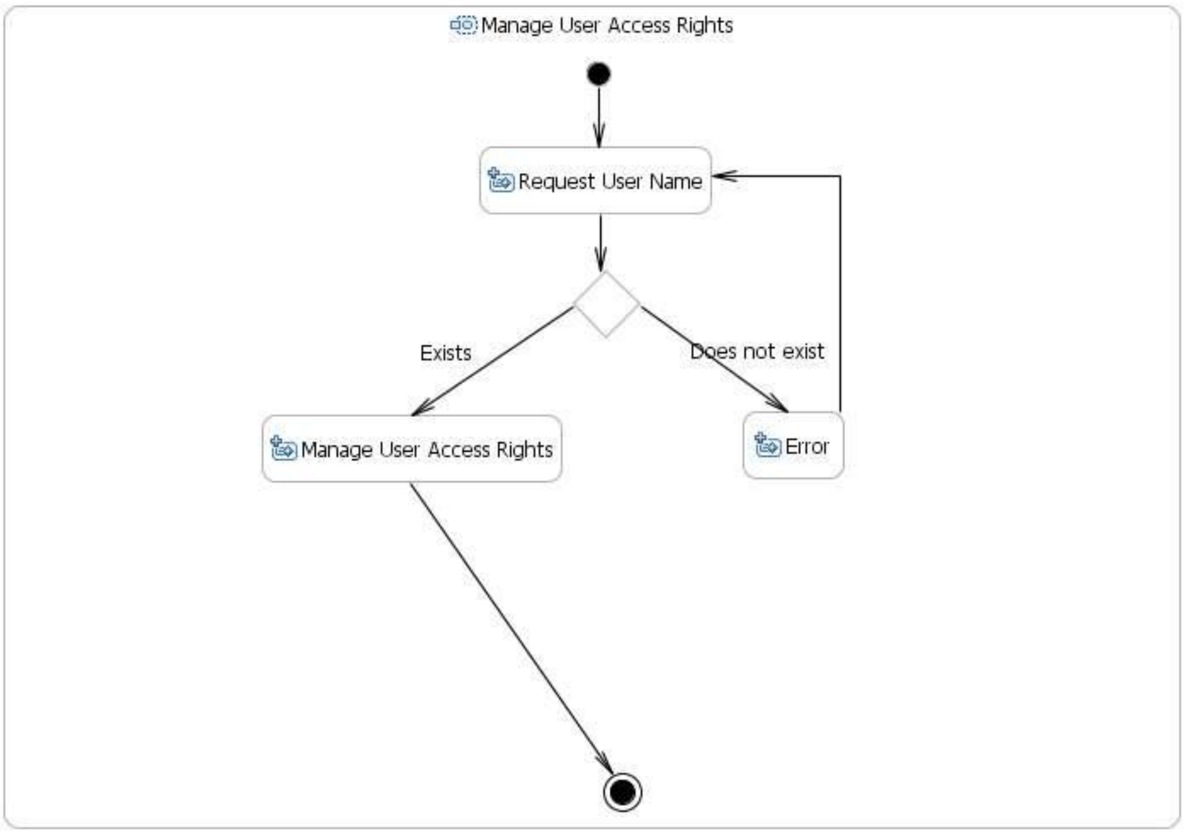

Figure 1-6: Activity diagram of manage User Access Rights

Aspect	Description
Objective	Updating a "VCDSP_MemberStateAdministrator" / "VCDSP_MemberStateUser" access rights account
Results (postconditions in case of success)	Directory entrance account hosting user access rights is updated The owner of the account is notified
Precondition	The actor acting is authenticated The user access rights account to be updated exists (!)
Postcondition in case of failure	Error message is displayed
Actor(s)	"VCDSP_Administrator" (EU level) / "VCDSP_MemberStateAdministrator" or "VCDSP_MemberStateUser" (MS level)
Initiating event	
Description of interaction procedure with VCD Service (standard run)	The actor fills in the form (State area) The actor confirms its choice The account selected is updated
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	
LEG/ORG	
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-5: Manage User Access Rights

Use-Case "Authenticate"

Authenticate User made authentication of a User ("VCDSP_Administrator" (EU level) / "VCDSP_MemberStateAdministrator" and "VCDSP_MemberStateUser").

Input-Output data:

Activity	Input	Output	User/System
Request login	None	None	UIAManager
Input login	login	None	VCDSP_Administrator" (EU level) / "VCDSP_MemberStateAdministrator" and "VCDSP_MemberStateUser"
Request password	Password	None	UIAManager
Input password	Password	None	VCDSP_Administrator" (EU level) / "VCDSP_MemberStateAdministrator" and "VCDSP_MemberStateUser"
Check Directory	Login, password	User Access_Rights or Error response	UIAManager
Error response	Error response	None	UIAManager

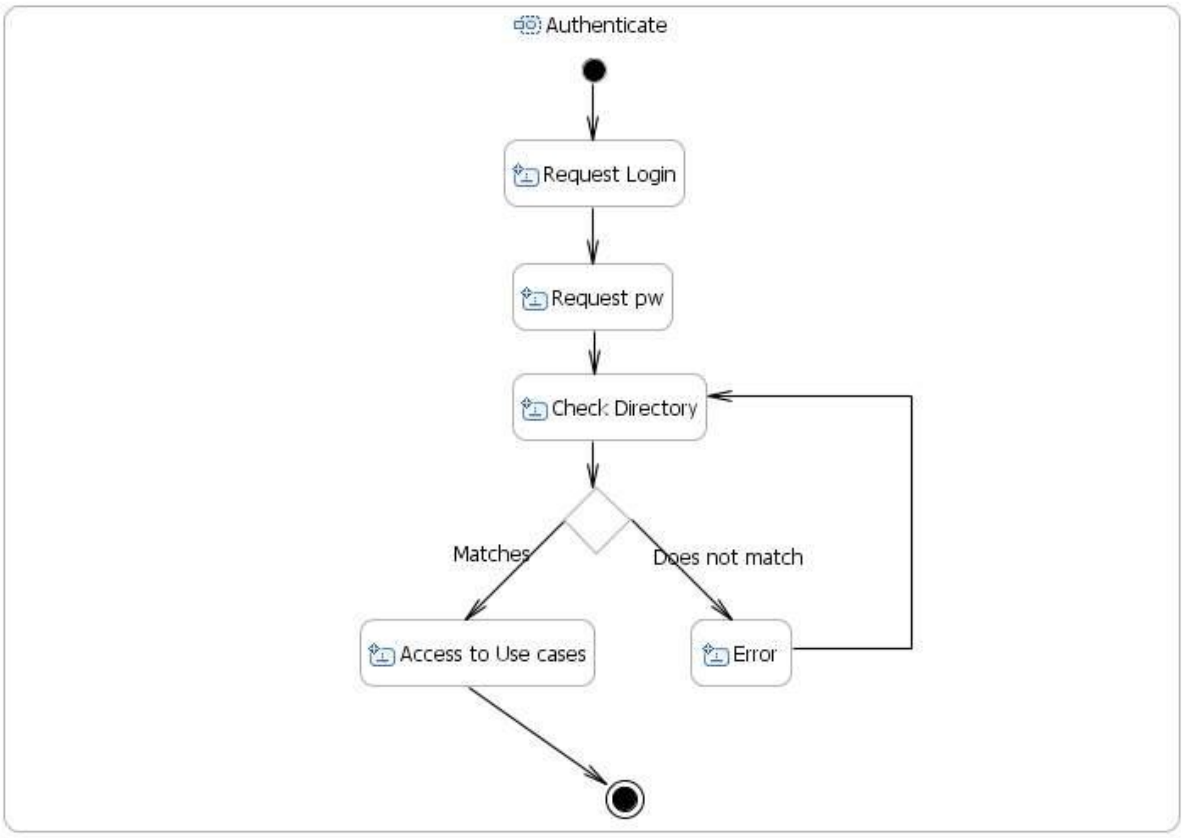

Figure 1-7: Activity diagram of authentication of a User

Aspect	Description
Objective	Be insured the user is who he claims to be
Results (post-conditions in case of success)	Allow access to use cases
Precondition	
Postcondition in case of failure	Error message is displayed
Actor(s)	"VCDSP_Administrator" (EU level) / "VCDSP_MemberStateAdministrator" or "VCDSP_MemberStateUser" (MS level)
Initiating event	
Description of interaction procedure with VCD Service (standard run)	The actor fill in the form (login and password)
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	
LEG/ORG	Need of specific Use-Case for strong authentication for "VCDSP_Administrator" (EU level) / "VCDSP_MemberStateAdministrator"?
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-6: Authenticate a user

Use-Case “Retrieve Pwd”

Retrieve Pwd lets a “VCDSP_Administrator” (EU level) /or a “VCDSP_MemberStateAdministrator” or “VCDSP_MemberStateUser” to retrieve a new password.

Input-Output data:

Activity	Input	Output	User/System
Request informations	None	Display a form	UIAManager
Fill form	Login and informations	none	VCDSP_Administrator” (EU level) / “VCDSP_MemberStateAdministrator” and “VCDSP_MemberStateUser”
Check Directory	Login, password	Send new password or Error response	UIAManager
Error response	Error response	None	UIAManager

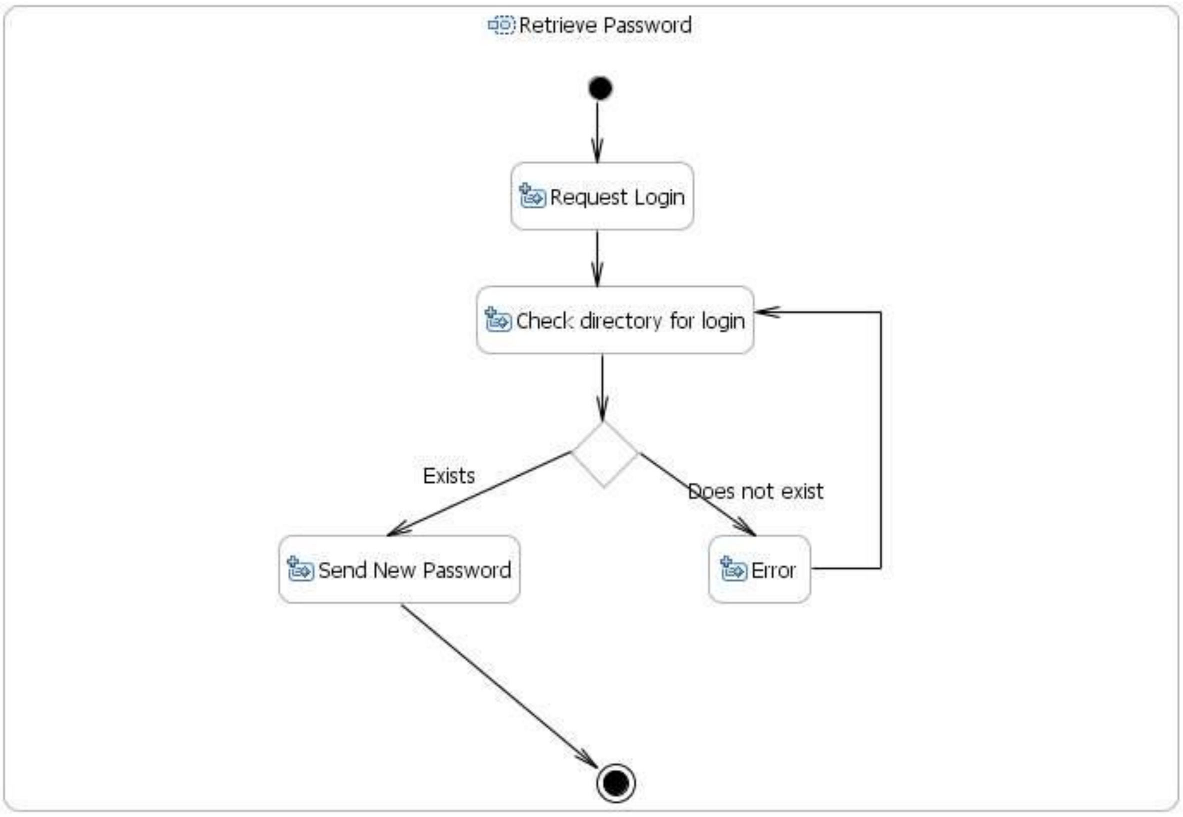

Figure 1-8: Activity diagram of Retrieve Password

Aspect	Description
Objective	Send to a user a new access key
Results (post-conditions in case of success)	New password is sent
Precondition	Directory entrance account hosts user login
Post-condition in case of failure	Error message is displayed
Actor(s)	"VCDSP_Administrator" (EU level) / "VCDSP_MemberStateAdministrator" or "VCDSP_MemberStateUser" (MS level)
Initiating event	
Description of interaction procedure with VCD Service (standard run)	The actor fill in the form (login) User submits its login A new password is sent
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	
LEG/ORG	Need of specific Use-Case for "VCDSP_Administrator" (EU level) / "VCDSP_MemberStateAdministrator"?
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-7: Retrieve Pwd

Use-Case-group "Issuing body interfaces"

The use cases and functions for this system part will be described in the specific VCD service annexes.

Use-Case-group "Manual upload interface (e.g. self declarations)"

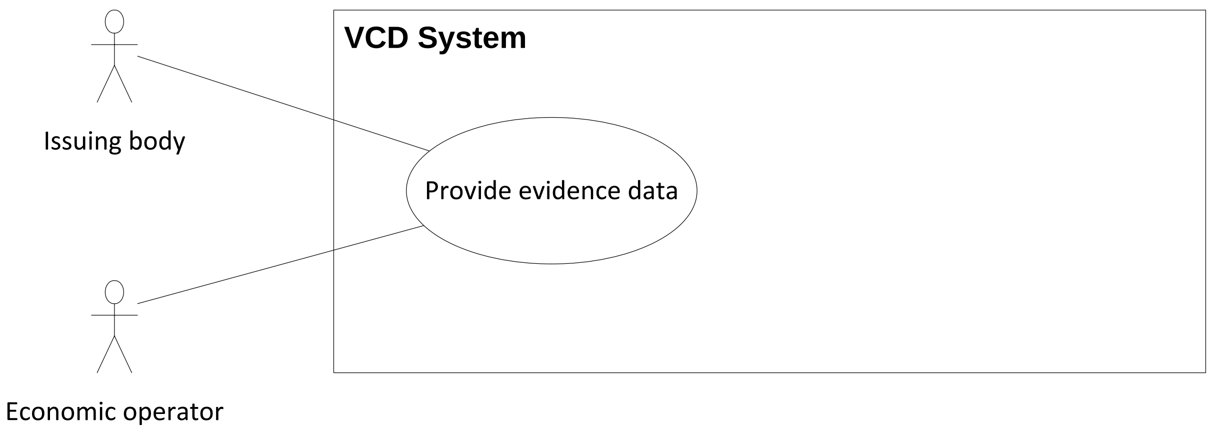

Figure 1-9: Use-Cases of the VCD system (Manual upload interface (e.g. self declarations))

Aspect	Description
Objective	The user should be enabled to provide evidence data, which are required in accordance to the rule set, and which are not provided by automated issuing services. This includes evidence documents as well as structural and context specific data.
Results (postconditions in case of success)	The respective evidence data has been provided by the user and is stored in the system (at least for the time span of the VCD compilation until the VCD package is delivered).
Precondition	The user is registered and logged in The VCD package production is in the state of evidence data acquisition (e.g. the final list of the to be provided evidences has been fixed).
Postcondition in case of failure	The system is in the same state as before, no respective evidence data is stored in the system.
Actor(s)	Economic operator Issuing bodies (personnel performing manual input)
Initiating event	The system requires manual evidence data input
Description of interaction procedure with VCD Service (standard run)	The system provides a user interface for manual data input (depending on the respective evidence and data set to be filled in, derived from the underlying ontology). This user interface can be either a single user form or a wizard dialog. The user fills in the requested data fields and selects one or more local files for upload (if required). The user confirms the authenticity, consistency and correctness of the uploaded document(s) and provided data (depending on the policy of the service provider). The user submits the data. The system validates the data in accordance to the ontology. In case of positive data validation, the system stores the data temporary. The system logs the user-input (user-id, hash for document, date/time, service-id, status, user-confirmation).
Description of interaction procedure with VCD Service (alternative runs)	5.a) in case of negative validation, the system informs the user about the validation results 6.a) the user re-enters the wrong data and submits the data The process continues with step 4.
Extension(s)	None
LEG/ORG	
Used in stage	This Use-Case is relevant for stages 2, 3 and 4

Table 1-8: Provide evidence data

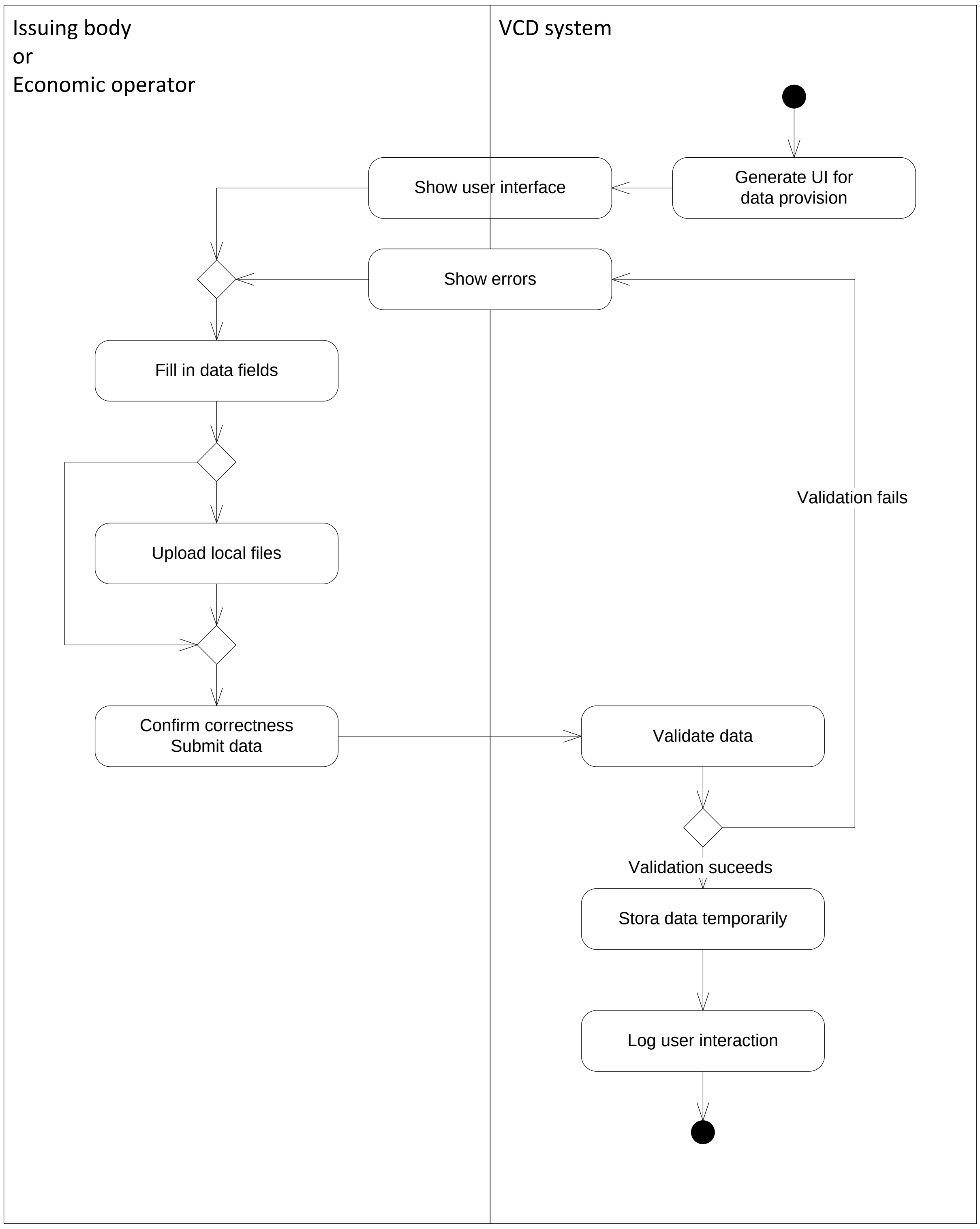

Figure 1-10: Activity diagram for use case “Provide evidence data”

Input-Output data:

Activity	Input	Output	User/System
Generate UI fpr data provision	None	UI	VCD system
Show user interface	UI	UI	VCD system
Fill in data fields	UI	UI with local filled in data	User
Upload local files	UI	Local files	User
Confirm correctness and submit data	UI with local filled in data, optionally local files, confirmation data	Submit request	User
Validate data	Submit request	Error response or submit request	VCD system
Store data temporarily	Submit request	Local ontology data	VCD system
Log user interaction	Local ontology data	User interaction log entry	VCD system

1.1.3 Use-Case-group "Signature Functions"

The signing of generated VCD objects represents that part of the building activity that proves the commitment and the level of responsibility endorsed by the stakeholders performing the VCD compilation. The signing activities can be applied to the following artefacts:

- VCD Container
- VCD Package
- Visible VCD Package
- VCD
- Documents.

The interaction of different stakeholders with the system is represented by the following Use Case representations and the connected Activity diagrams.

SignVCD Use-Case

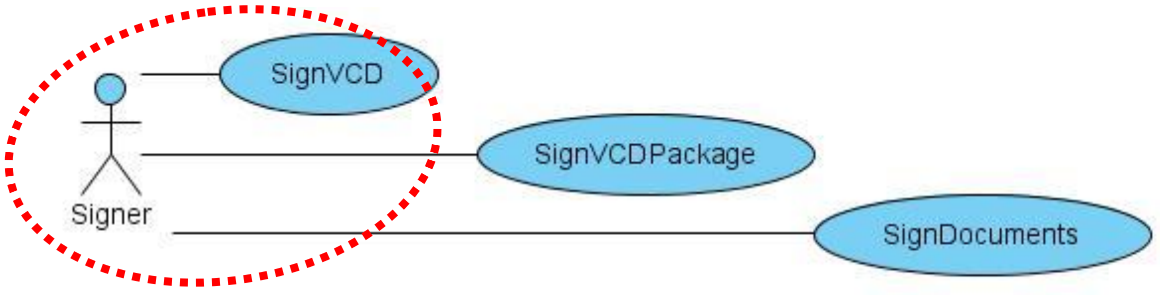

Figure 1-11: Digital Signature operations

The "SignVCD" operation shown in the above UseCase diagram is the sub-task of a more general activity, i.e. the VCDSignature activity within the VCD Builder subsystem. It's activated by the Signer within our Use Case. It affects the VCD (XML) instance created by the "CompileVCD" operation.

The signature will be compliant with the directive EC/1999/93 and by applying the UBL2.0 compatibility principle it will have a specific profiling described in a draft document under evaluation (OASIS TC [UBL XAdES Profile Version 1.0 - http://lists.oasis-open.org/archives/ubl-security/200908/doc00000.doc]). The UBL-XAdES compatibility represents the current sign of convergence between Peppol and the work coordinated by the Technical Committees: OASIS Universal Business Language (UBL) TC; OASIS UBL Security SC.
There is at the moment an open discussion about the use of joined or separated methods, i.e. XADES (ETSI TS 101 903) or PADES (ETSI TS 102 778 part.5). This is related to the chosen Human readable representation, i.e the "visible signature" concept under debate and standardization.
Concerning the pure XML object, i.e. the VCD (XML), it should be used the method defined in: http://www.w3.org/TR/xmldsig-core/#def-SignatureEnveloped .

This allows for easier parsing activities, independent from the signature. The choice is also particularly convenient when adopting the UBL format. The formats to be used are XAdES-BES or XAdES-T. More details are given by accessing the sources listed in the Reference section in Attachment B

Aspect	Description
Objective	The user invokes a function exposed by a web application protected by strong authentication. The final outcome must be a signed VCD that conforms the specified and agreed standards
Results (postconditions in case of success)	Signed VCD (XML) file Note: the signature will not affect the VCDPackage and the VCDContainer
Precondition	A VCD(XML) instance ready for consolidation, generated after a complete compilation process. A security token with a valid X509 qualified digital certificates that conforms with the directive EC/1999/93 and ETSI standard [ETSI TS 101 862 V1.3.3] - http://pda.etsi.org/exchangefolder/ts_101862v010303p.pdf A running system offering sign/verify services.
Postcondition in case of failure	No signature generated
Actor(s)	Signer: VCD Service Provider, Economic Operator
Initiating event	Start SignVCD
Description of interaction procedure with VCD Service (standard run)	Follows the VCD(XML) compilation, anticipates the VCDPackage(XML) production
Description of interaction procedure with VCD Service (alternative runs)	The signing of the VCD can be omitted in case the production of a VCDPackage takes place at the same VCD Service Provider location.
Extension(s)	The attestation documents, referenced by the VCD are also evaluated for their existing signatures, when present. Metadata are generated and inserted into the structure: “Results of Verification”. The signed XML document may require a companion document offering the visual representation of the VCD content. The visible VCD MAY be signed as well.
LEG/ORG	LEG: Leads to the “responsibility” issue based on VCD production; ORG: linked to the VCD service policy when signing is included
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-9: SignVCD

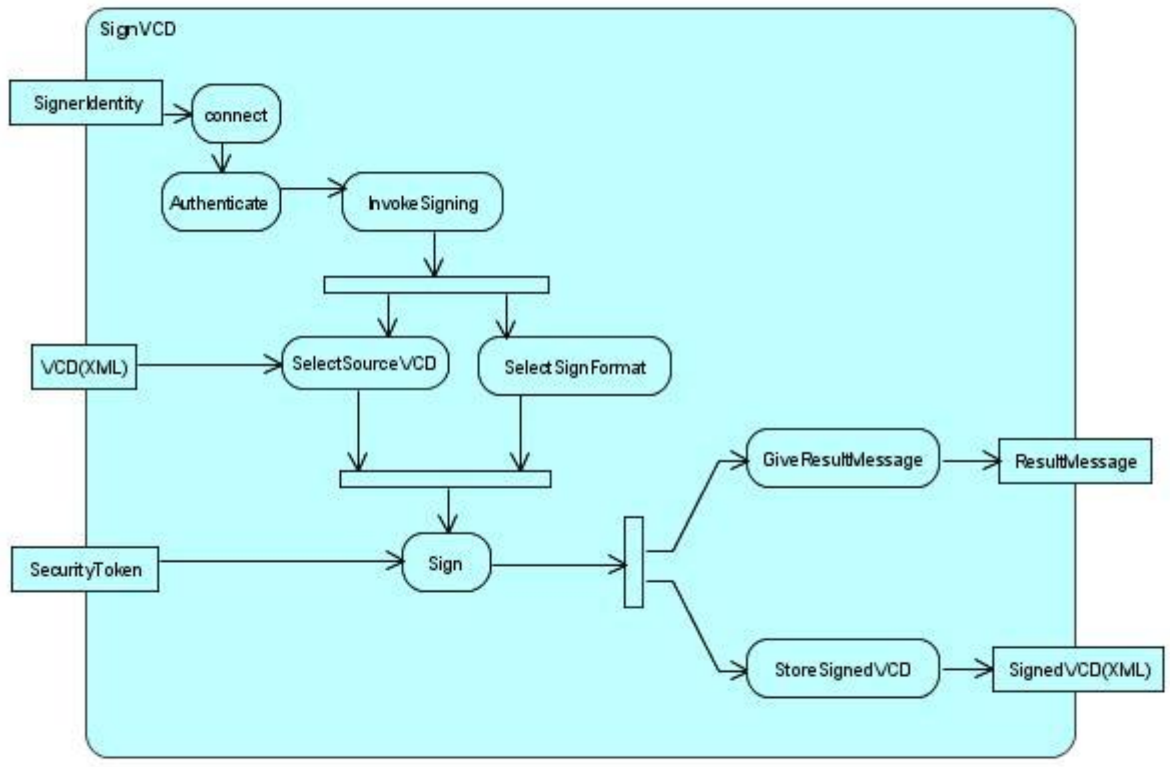

Figure 1-12: SignVCD_AD

SignVCDPackage Use-Case

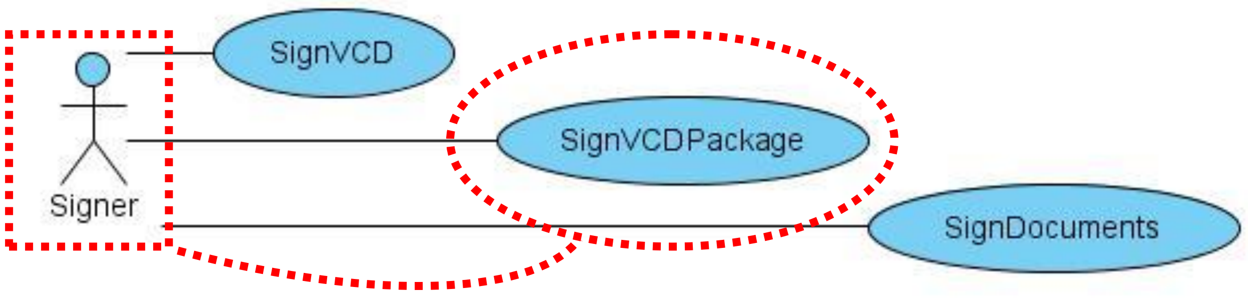

Figure 1-13: SignVCDPackage

The “SignVCDPackage” operation shown in the above UseCase diagram is the sub-task of a more general activity, i.e. the VCDSignature activity within the VCD Builder subsystem. It’s activated by the Signer within our Use Case. It affects the VCDPackage(XML) instance created by the “CompileVCDPackage” operation.

The signature will be compliant with the directive EC/1999/93 and by applying the UBL2.0 compatibility principle it will have a specific profiling described in a draft document under evaluation (OASIS TC [UBL XAdES Profile Version 1.0 - http://lists.oasis-open.org/archives/ubl-security/200908/doc00000.doc]). The UBL-XAdES compatibility represents the current sign of convergence between Peppol and the work coordinated by the Technical Committees: OASIS Universal Business Language (UBL) TC; OASIS UBL Security SC.

There is at the moment an open discussion about the use of joined or separated methods, i.e. XADES (ETSI TS 101 903) or PADES (ETSI TS 102 778 part.5). This is related to the chosen Human readable representation, i.e the "visible signature" concept under debate and standardization.
Concerning the pure XML object, i.e. the VCDPackage (XML), it should be used the method defined in: http://www.w3.org/TR/xmldsig-core/#def-SignatureEnveloped .
This allows for easier parsing activities, independent from the signature. The choice is also particularly convenient when adopting the UBL format. The formats to be used are XAdES-BES or XAdES-T. More details are given by accessing the sources listed in the Reference section in Attachment B.

Aspect	Description
Objective	The user invokes a function given by a XAdES enabled application. The final outcome must be a signed VCDPackage that conforms the specified and agreed standards
Results (postconditions in case of success)	Signed VCDPackage (XML) file
Precondition	A VCDPackage(XML) instance ready for consolidation, generated after a complete packaging process. A security token with a valid X509 qualified digital certificates that conforms with the directive EC/1999/93 and ETSI standard [ETSI TS 101 862 V1.3.3] - http://pda.etsi.org/exchangefolder/ts_101862v010303p.pdf A running system offering sign/verify services.
Postcondition in case of failure	No signature generated
Actor(s)	Signer: VCD Service Provider, Economic Operator
Initiating event	Start SignVCDPackage
Description of interaction procedure with VCD Service (standard run)	Follows the VCDPackage(XML) compilation, anticipates the VCDContainer production
Description of interaction procedure with VCD Service (alternative runs)	The signing of the VCDPackage can override the signing of the included VCD when both objects come from the same location.
Extension(s)	The signed XML document may require a companion document offering the visual representation of the VCDPackage content. The visible VCDPackage MAY be signed as well.
LEG/ORG	LEG: Leads to the "responsibility" issue based on VCD production; ORG: linked to the VCD service policy when signing is included
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-10: SignVCDPackage

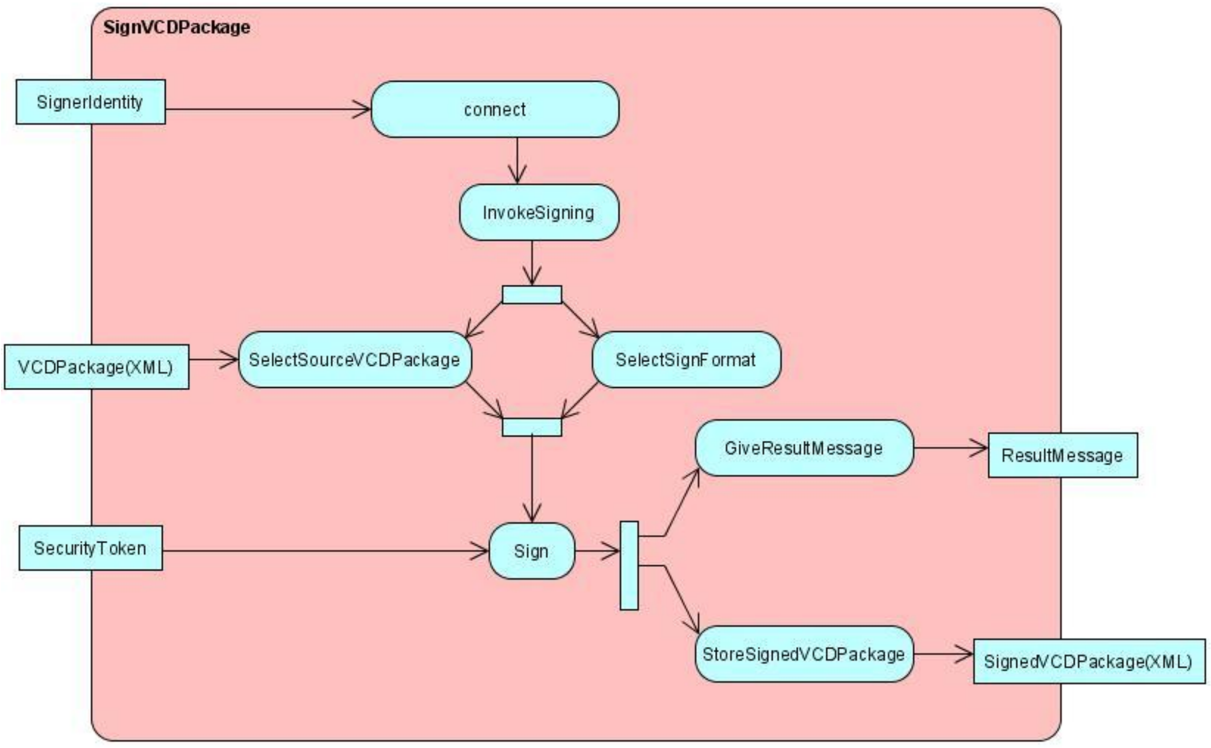

Figure 1-14: Sign VCD Package

Sign Documents - Use-Case diagram

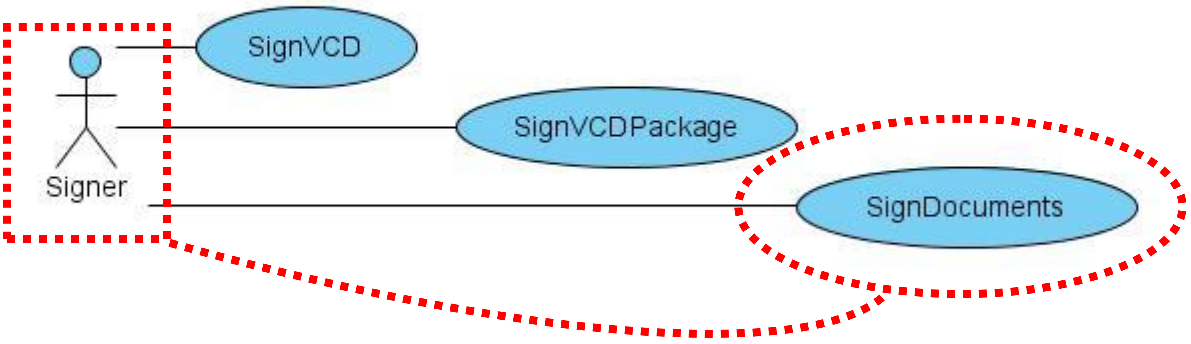

Figure 1-15: Sign Documents

This is the task offered to document creators, in particular Economic Operators, that intend to use a signing capability offered by a platform that commit to produce signatures conforming to the adopted Peppol_WP1 recommendations.

Aspect	Description
Objective	The user invokes a function given by an application enabled to create signatures. The final outcome must be a signed Document that conforms the specified and agreed standards
Results (postconditions in case of success)	Signed (XML) file Signed (non XML) files
Precondition	Any existing document (XML/non XML) instance. A security token with a valid X509 qualified digital certificates that conforms with the directive EC/1999/93 and ETSI standard [ETSI TS 101 862 V1.3.3] - http://pda.etsi.org/exchangefolder/ts_101862v010303p.pdf A running application/system offering sign and verify services
Postcondition in case of failure	No signature generated
Actor(s)	Signer:
Initiating event	Start SignDocument
Description of interaction procedure with VCD Service (standard run)	Follows a determination taken by the Economic Operator to apply a signature over documents under his authority (e.g. self declarations, VCD Container).
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	The documents referenced by the VCD are also evaluated for their signature. Metadata are generated: “Results of Verification”
LEG/ORG	LEG: Leads to the “responsibility” issue based on “attestations” production; ORG: linked to the VCD service policy when signing is included
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-11: Sign Documents

1.1.4 Use-Case-group “Signature validation interface”

The validation of signed objects follows a general strategy outlined on the diagram below. In the UseCase diagram we also present the level of interaction with servers getting implementation through other project activities, i.e. the WP1 subsystem and the XKMS responder implementation giving support to the “ValidateIdentity” operation.

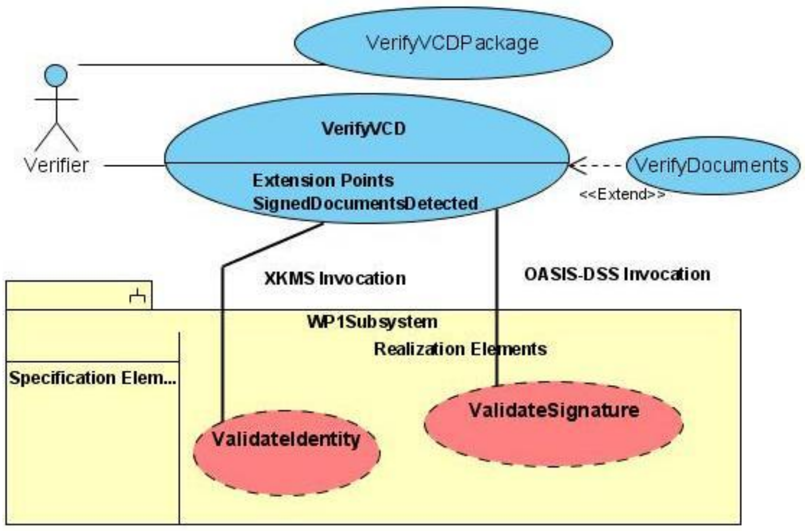

Figure 1-16: Verifier Use Case and XKMS

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

– The basic attestations in every possible electronic format (affected by the "VerifyDocuments" process);
– The VCD(XML) instance (affected by the "VerifyVCD" process);
– The VCDPackage (XML) instance (affected by the "VerifyVCDPackage" process).

A valid Peppol business transaction can be performed without any Electronic (Advanced or not) Signature. Nonetheless, Peppol cannot ignore the existing regulation at the European and national level to give legal validity to electronic documents while producing or sharing them across the network. The Peppol validation strategy will tentatively follow the ETSI validation process model.

The Validation Process validates an electronic signature, the output status of the validation process can be: a) □**invalid**; b)□ **incomplete validation**; c) □**valid**.
An **Invalid** response indicates that either the signature format is incorrect or that the digital signature value fails verification (e.g. the integrity check on the digital signature value fails or any of the certificates on which the digital signature verification depends is known to be invalid or revoked).
An **Incomplete Validation** response indicates that the format and digital signature verifications have not failed but there is insufficient information to determine if the electronic signature is valid. For example; all the required certificates are not available or the grace period is not completed. In the case of Incomplete Validation, the electronic signature may be checked again at some later time when additional validation information becomes available. Also, in the case of incomplete validation, additional information may be made available to the application or user, thus allowing the application or user to decide what to do with partially correct electronic signatures.
A **Valid** response indicates that the signature has passed verification and it complies with the signature validation Policy.

Use-Case "VerifyVCD"

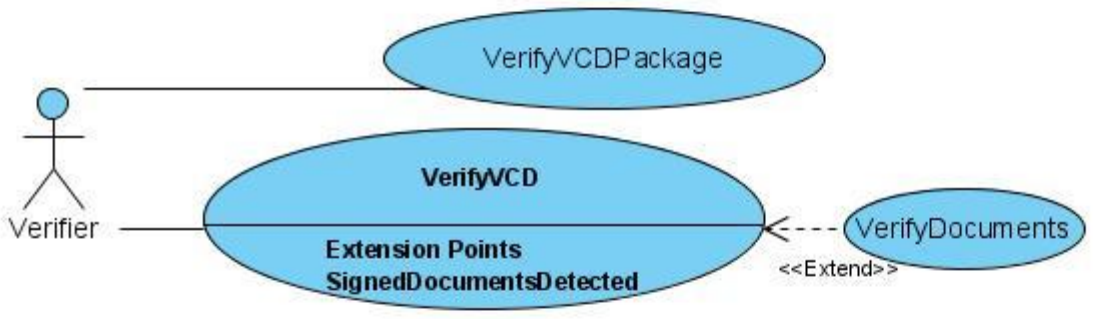

Figure 1-17: Verifier UseCase

Aspect	Description
Objective	1: Verify the signature validity on a VCD 3: Verify the identity of signers
Results (postconditions in case of success)	As stated above, the output status of the validation process can be : a) □invalid; b) incomplete validation; c) □valid.
Precondition	A running system offering sign/verify services A signed VCD instance associated to the set of documents referenced by its [VCD::evidence::DocumentReference] structure.
Postcondition in case of failure	Validation process error generated
Actor(s)	Verifier (any entity passing the authentication phase)
Initiating event	Invocation of the Verify function
Description of interaction procedure with VCD Service (standard run)	The Verify can be initiated at any time; The Verify generates the “VerifyReport” ; The VerifyReport hosts different validation indicators according to the service policy; The “VerifyReport” is external to the VCD, the VCDPackage and the VCDContainer.
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	The attestation documents, referenced by the VCD through the [VCD::evidence::DocumentReference] structure are also evaluated for their signature.
LEG/ORG	LEG: Leads to the “responsibility” issue based on VCD production, CAN be used as element of trust on the “Signer”. ORG: linked to the VCD service policy when “Validation” is covered
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-12: VerifyVCD

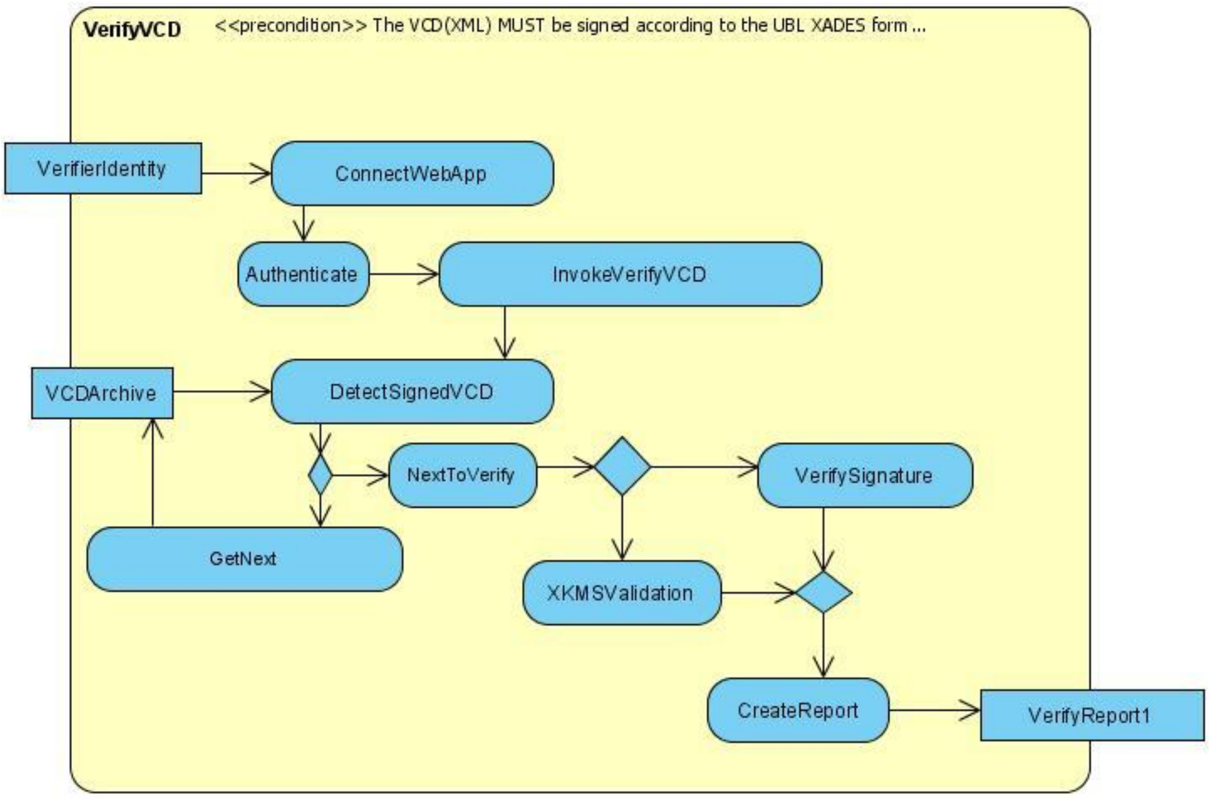

Figure 1-18 - VerifyVCD_AD

Use-Case “VerifyVCDPackage”

The VCD pilot applications will accept and support the presence of Digitally Signed objects at different levels. We pay attention here to the VCDPackage (XML) instance.

Aspect	Description
Objective	1: Verify the signature validity on a VCDPackage 3: Verify the identity of signers
Results (postconditions in case of success)	As stated above, the output status of the validation process can be*:* *a)* ***invalid****; b)* ***incomplete validation****; c)* ***valid****.*
Precondition	A running system offering sign/verify services
Postcondition in case of failure	Validition process error generated
Actor(s)	Verifier (any entity passing the authentication phase)
Initiating event	Invocation of the Verify function
Description of interaction procedure with VCD Service (standard run)	The Verify can be initiated at any time; The Verify generates the “VerifyReport” ; The VerifyReport hosts different validation indicators according to the service policy; The “VerifyReport” is external to the VCDPackage and the VCDContainer.
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	
LEG/ORG	LEG: Leads to the “responsibility” issue based on VCDPackage

	production, CAN be used as element of trust on the “Signer”. ORG: linked to the VCD service policy when “Validation” is covered
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-13: VerifyVCDPackage

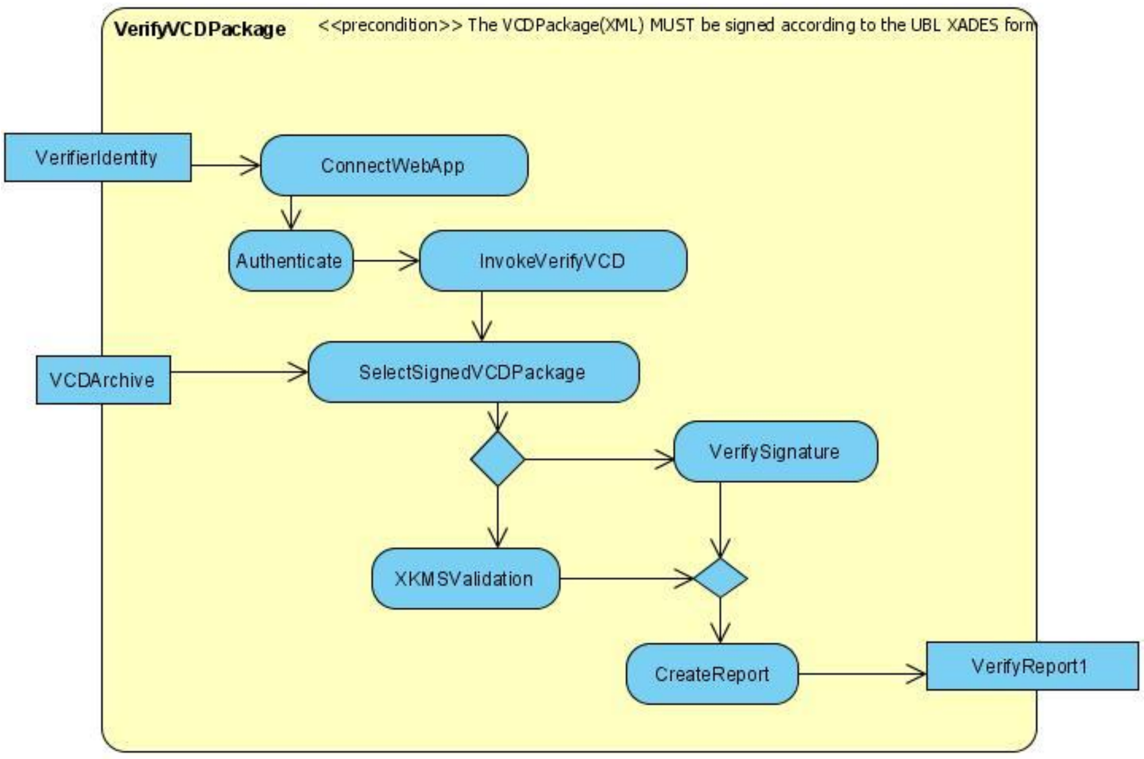

Figure 1-19: VerifyVCDPackage_AD

Use-Case VerifyDocuments descriptions

The VCD pilot applications will accept and support the presence of Digitally Signed Objects at different levels. VCD pilots pay attention to the basic attestations or similar documents that are received as contained objects within a “VCDContainer” acting as physical container. These “Digitally Signed Objects” are also referenced by the [VCD::evidence::DocumentReference] structure.
The present Use Case represents an extension of the VerifyVCD Use-Case and defines an activity to validate documents classified as evidences or non-evidences by the VCD schema definition. The Verifier gets a summary of documents identified by the VerifyVCD activity and resulting in the “VerifyReport” described in [Use-Case “VerifyVCD”]. That VerifyVCD activity returns an integrity indicator affecting the whole VCD. The Verifier has a second option i.e. to challenge the initial verify done at the VCD Compilation time and resulting into the “Result of Verification” data structure. The document validation activity relies on the XKMS server support to validate the identity of original signers.

Aspect	Description
Objective	Verify the signature validity on a Document present within a VCD Container or equivalent containers.
Results (postconditions in case of success)	the output status of the validation process can be: a) **invalid**; b) **incomplete validation**; c) **valid**.
Precondition	A running system offering sign/verify services A set of candidate documents hosted by a valid VCDContainer or

	equivalent container.
Postcondition in case of failure	A VerifyReport , including error, is generated
Actor(s)	Contracting Authority, Economic operator, VCD service provider
Initiating event	Explicit intention of operator
Description of interaction procedure with VCD Service (standard run)	The operation can be executed BEFORE the VCD compilation over candidate attestation documents. The generated “VerifyReport” CAN be used as INPUT metadata for the [VCD::evidence::DocumentReference::ResultOfVerification] structure. The operation can be executed AFTER the VerifyVCD operation over the VCD referenced documents.
Description of interaction procedure with VCD Service (alternative runs)	The operation can be executed ANYTIME.
Extension(s)	
LEG/ORG	LEG: Leads to the “responsibility” issue based on VCD production; ORG: linked to the VCD service policy when verify is included
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-14: VerifyDocuments

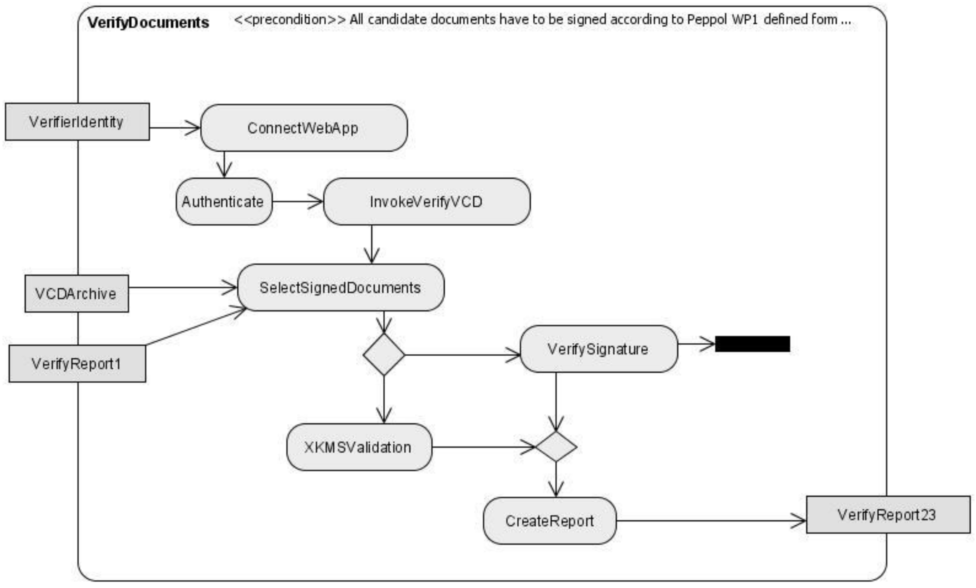

Figure 1-20: VerifyDocuments_AD

1.1.5 Use Case group “VCD System”

The Use Case group will address functionalities that are all required to BUILD an instance of a VCD along with the VCD Package and the VCD Container. The relationship among the three artefacts can be observed in the following pictorial representation.

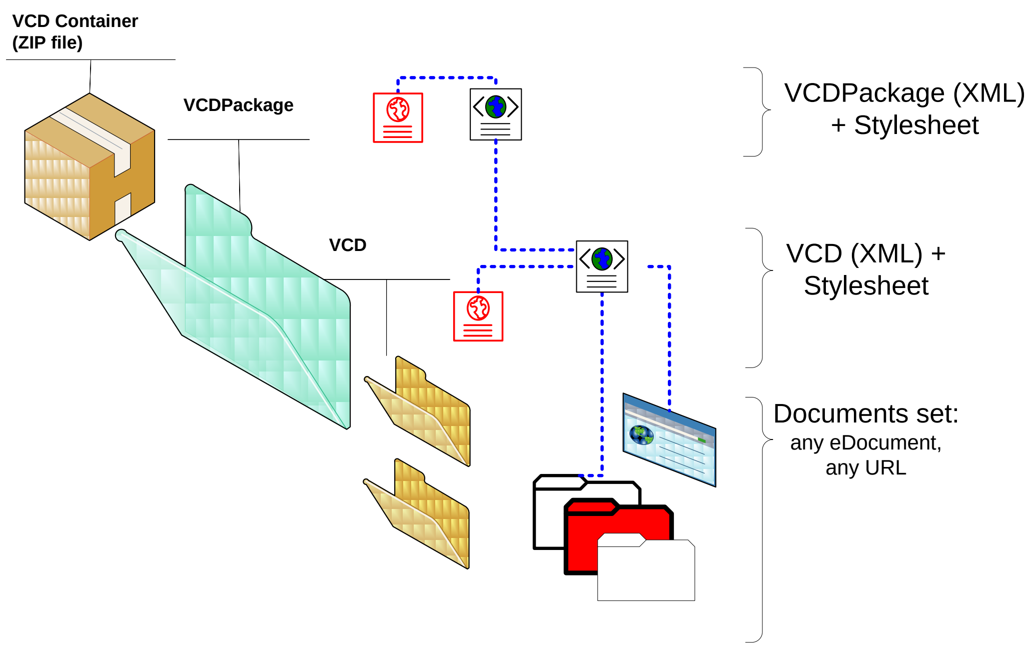

Figure 1-21: VCD system functions (system view)

The steps required to deal with the activity are several and dispersed across different systems, sub-systems and packages. These different entities are generically indicated as "components" through the whole specification. Anyway, the current specification work will provide a description of the functions we are getting from them. For an easy reading of the next chapters, a components' tree has been elaborated and briefly reproduced here below. The grouping criteria follows the physical appearance of the different entities, more details are given in the "Legend" attached to the table. An overview diagram comes along with the table.

Reference	System	Sub-systems	Packages	Functions/Use cases
1.1.6	National VCD System			
1.1.8		VCD Builder		
1.1.8.1				CompileVCD
1.1.8.2				CompileVCDPackage
				ViewVCD
				SignVCD
				SignVCDPackage
				VerifyVCDSignature
				VerifyVCDPackageSignature
1.1.9		VCD Container Creator		
1.1.9.1				AssembleVCDContainer
1.1.9.2				ManageVCDContainer
		VCD Package Viewer		CreateVisibleVCDPackage
				Manual VCD container retrieval
				Send VCD container to PEPPOL

				end point
				Receive VCD Container from Peppol end point
		Document Manager		AccessData AccessDocuments CreateDataAndDocuments
1.1.7	European VCD System			
1.1.7.**1**		VCD Skeleton Packager		CreateVCD Package Skeleton
1.1.7.x		VCD Package Viewer		CreateVisibleVCDPackage
		EuroVCD Interface		HandleRequests HandleResponses

Table 1-15: VCD system functions (system view)

Table Legend:

VCD System: The overall thing we are going to specify and implement
VCD Sub-system: The macro level of separation between subsystems taking care of macro area of functionality.
Packages: Are implementation units where classes and interfaces are well organized and properly identified. Packages can become shareable units among the implementation teams. The packaging strategy will be an implementation specific action, driven by the different choices made by WP2 piloting partners.
Functions/Use Cases: Are the perceivable/intentional operations that a requester can invoke

VCD Services List

The table below provides the linking/matching between the “VCD Services List”, scrutinized by the General Pilot Task Forces and the “functions” expected by the VCD Subsystem. Both tables have to be used for a cross check between the “VCD Pilot services” expected by the service users and the “VCD system functions” identified and under specification by the designers.

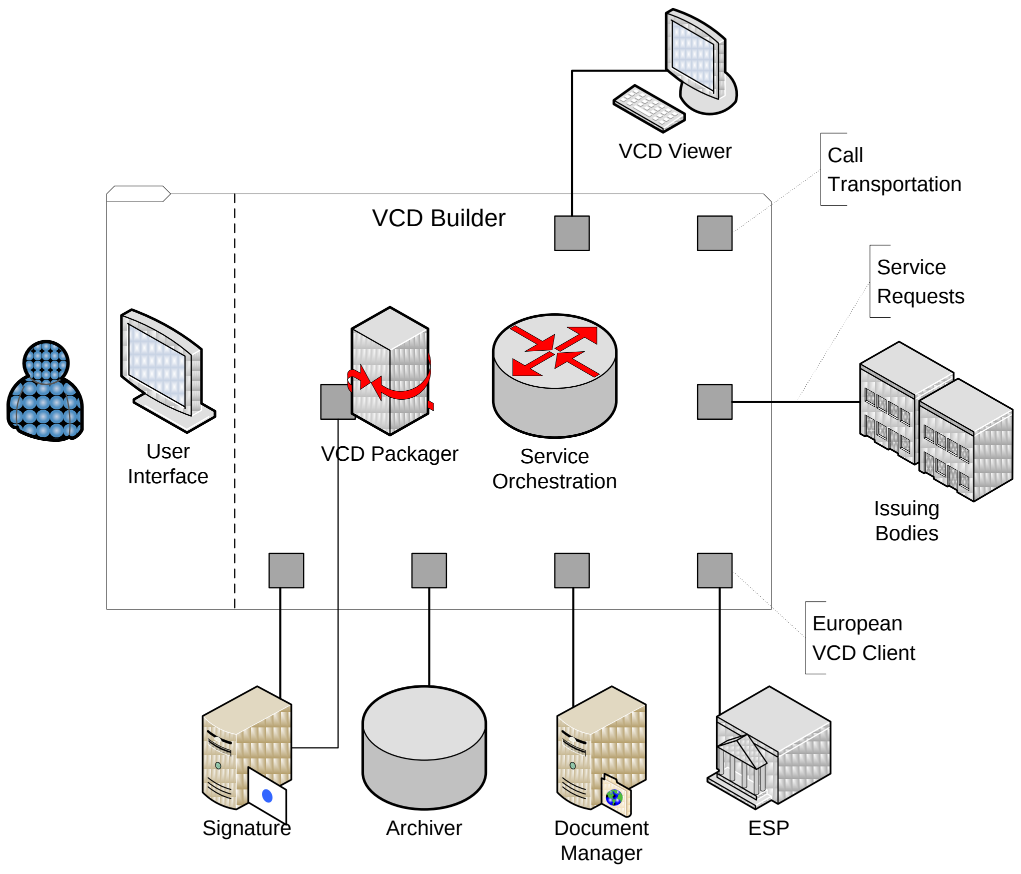

Figure 1-22: VCD Pilot services

ID	VCD Services List -	VCD Subsystem functions
	VCD Compiler incl. UI	CompileVCD
	User access and role management incl. authentication	
	Issuing body interfaces (not subject to PEPPOL funding)	AccessData AccessDocuments
	Manual data input interface	CreateDataAndDocuments
	Signature interface	SignVCD SignVCDPackage
	Signature validation interface	VerifyVCD VerifyVCDPackage
	VCD Packager	CompileVCDPackage
	VCD Container Creator	AssembleVCDContainer ManageVCDContainer
	VCD transportation interface incl. authentication (download)	GetVCDContainer
	VCD transportation interface incl. authentication (WP8)	SendVCDContainer ReceiveVCDContainer
	Interface to VCD dictionary (incl. code lists, schemas, …)	... Client functions on definition
	Interface to PEPPOL registries	... Client functions on definition
	Interface to European VCD service	SubmitRequests FetchResponses ImportVCDPackageSkeleton
	TED interface	

Table 1-16: VCD Pilot services

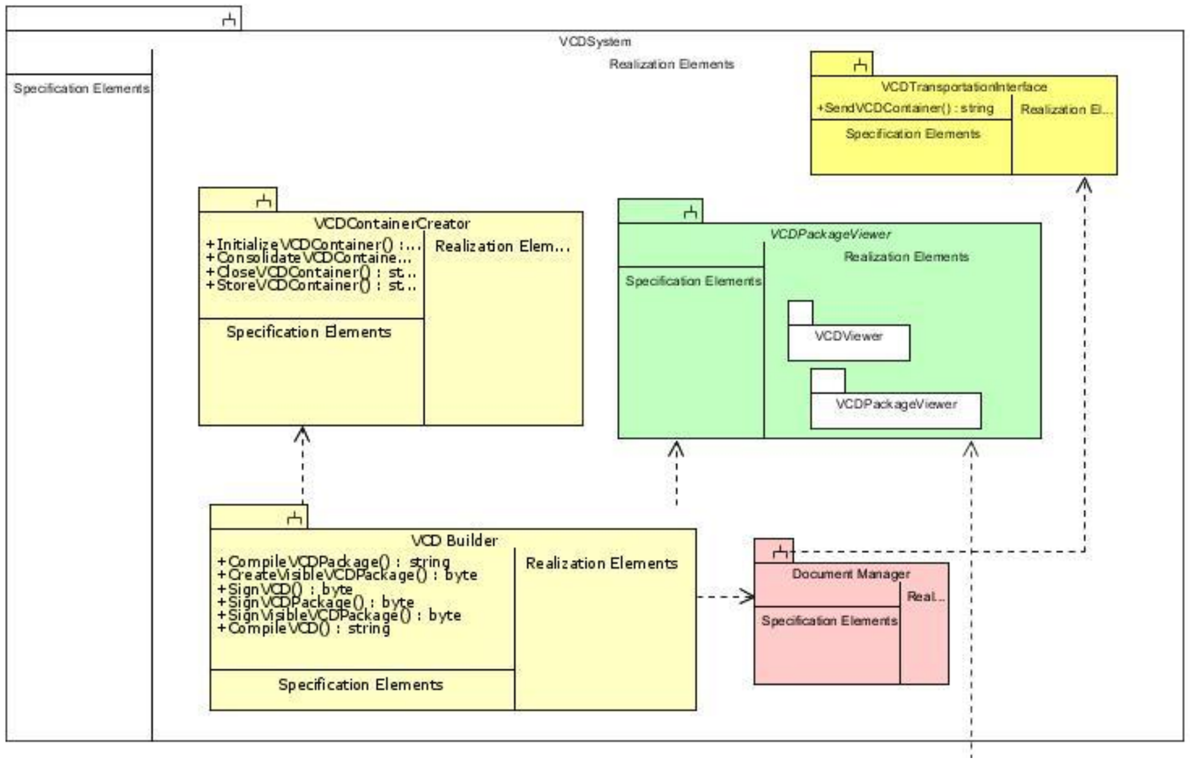

Figure 1-23: VCDSystem

1.1.6 Use-Case Group “VCD Builder”

The VCD Builder defines a sub-system where several package components are located, among them we define the “VCD Compiler” and the “VCD Packager” that are getting specification on next sections. The VCD Builder specification starts with a preliminary wider representation based on “UML package diagrams”.

The VCD Builder sub-system manages resources, activities and interactions to create VCDs and VCDPackages piling them up along with other companion elements to form the physical container that the project is planning to deliver or send for temporary archiving. The physical container will be later referenced as the VCD Container. The VCD Container represents the physical container for delivery of every object generated during the VCD Package compilation and consolidation. It also contains all the preparatory documents that the VCD has classified and referenced.

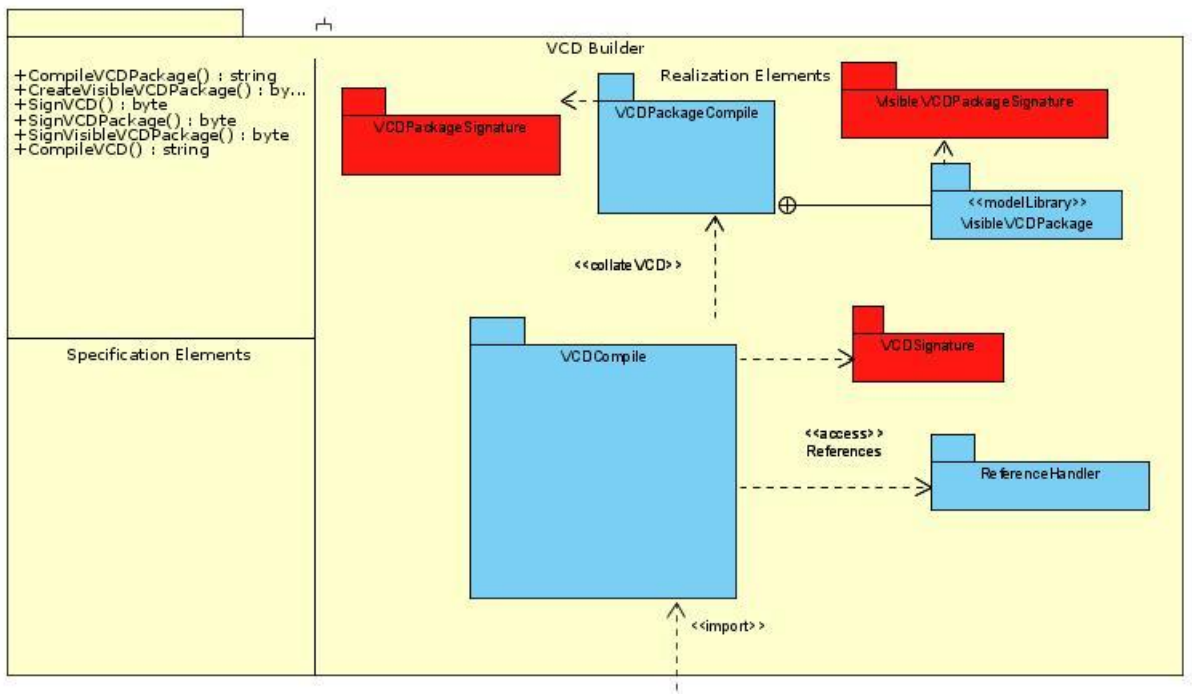

Figure 1-24: VCD Builder High-Level overview (including concurrent objects)

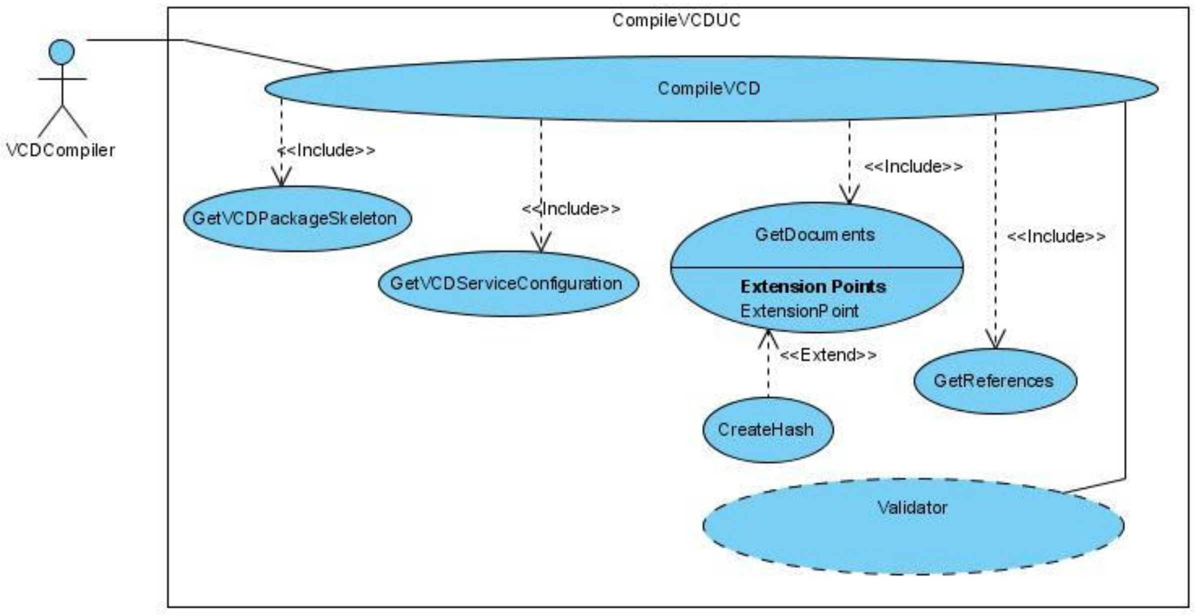

Figure 1-25: Compile VCD Use-Case

Aspect	Description
Objective	To create a complete instance of a VCD. It results from an interaction between a VCD Requester and the VCD System running a VCD service.
Results (postconditions in case of success)	A VCD (XML) instance validated
Precondition	Service Request presented by the requester A valid VCDPackageSkeleton (checked and approved by the Requester) A valid set of attestation instances A Service Configuration data file available
Postcondition in case of failure	Error message returned
Actor(s)	Economic Operator, Contracting Authority,, European VCD System
Initiating event	Request for a “VCD”.
Description of interaction procedure with VCD Service (standard run)	
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	Allows National VCD System to Euro VCD system interactions Allows VCD instance visualization Allows VCD Digital Signature generation Allows VCD Digital Signature Verification Allows interactions with Users
LEG/ORG	LEG: the overall production requires legal support ORG: workflow model must adapt to the different stages, keleton validation follows ORG directions. NSPT 2.1 . NSPT 2.6
Used in stage	Stage 2 to 4

Table 1-17: Compile VCD

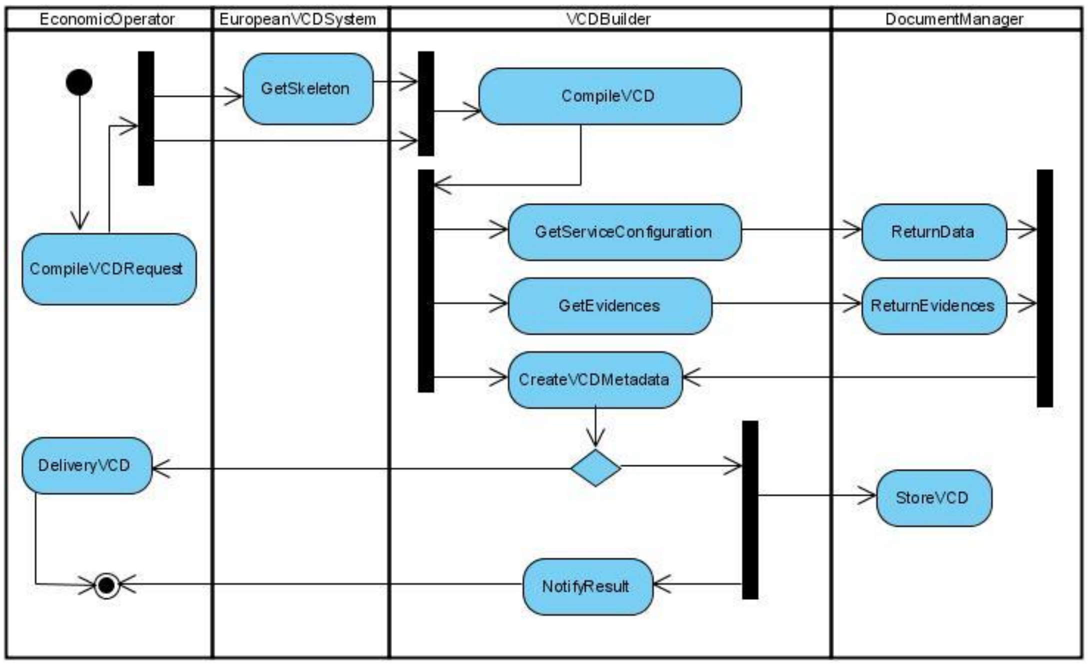

Figure 1-26: CompileVCDPackage Use-Case

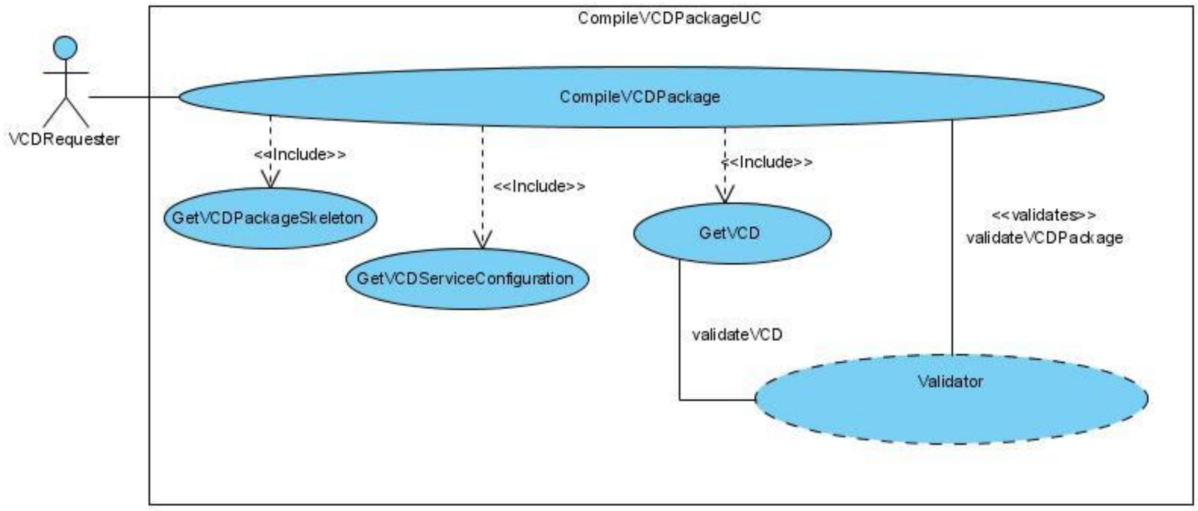

Figure 1-27: CompileVCDPackage Use-Case

The compilation of a “VCDPackage” implies the gathering and referencing of one or more VCD instances. It also requires the adaptation of the VCDPackage instance to the specific tendering activity that the Economic Operator intends to perform. The VCDPackage requires the processing of data that are giving an image of the “Bidding Consortium” and the target “Contracting Authority”. It works also for the consolidation of VCD instances already created and validated. The reference structure is offered by the “VCDPackageSkeleton”.

Aspect	Description
Objective	To create a complete instance of a VCDPackage. It results from an interaction between a VCD Requester and the VCD System running a VCD service.

Results (postconditions in case of success)	A VCDPackage (XML) instance validated
Precondition	Service Request presented by the requester A valid VCDPackageSkeleton (checked and approved by the Requester) A valid set of VCD instances A Service Configuration data file available Data from the Requester to fill empty elements
Postcondition in case of failure	Error message returned
Actor(s)	Economic Operator, Contracting Authority,
Initiating event	Request for a "VCDPackage".
Description of interaction procedure with VCD Service (standard run)	
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	Allows VCDPackage instance visualization Allows VCDPackage Digital Signature generation Allows VCDPackage Digital Signature Verification Allows interactions with Users
LEG/ORG	LEG: the overall production requires legal support ORG: workflow model must adapt to the different stages, validation follows ORG directions. NSPT 2.1 .. NSPT 2.6
Used in stage	Stage 2 to 4

Table 1-18: CompileVCDPackage

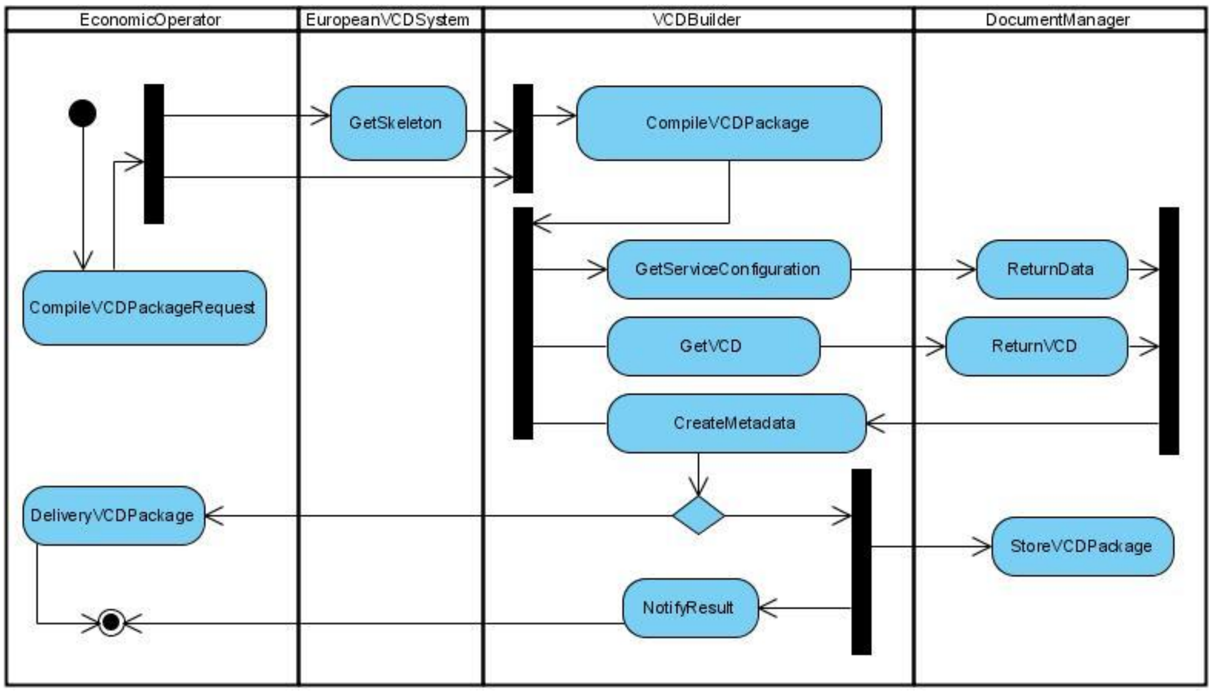

Figure 1-28: ComplieVCD Package

1.1.7 Use-Case Group “VCD Container Creator”

The "VCDContainerCreator" sub-system creates the running platform to two main packages, i.e. the "VCDContainerManager" and the "VCDContainerAssembler". Four basic operations get support within the sub-system. The list of such operations identifies the following functions: a) Initialize; b) Consolidate; c) Close; d) Store. These operations are all directed to form and transform every instance of the "VCDContainer".

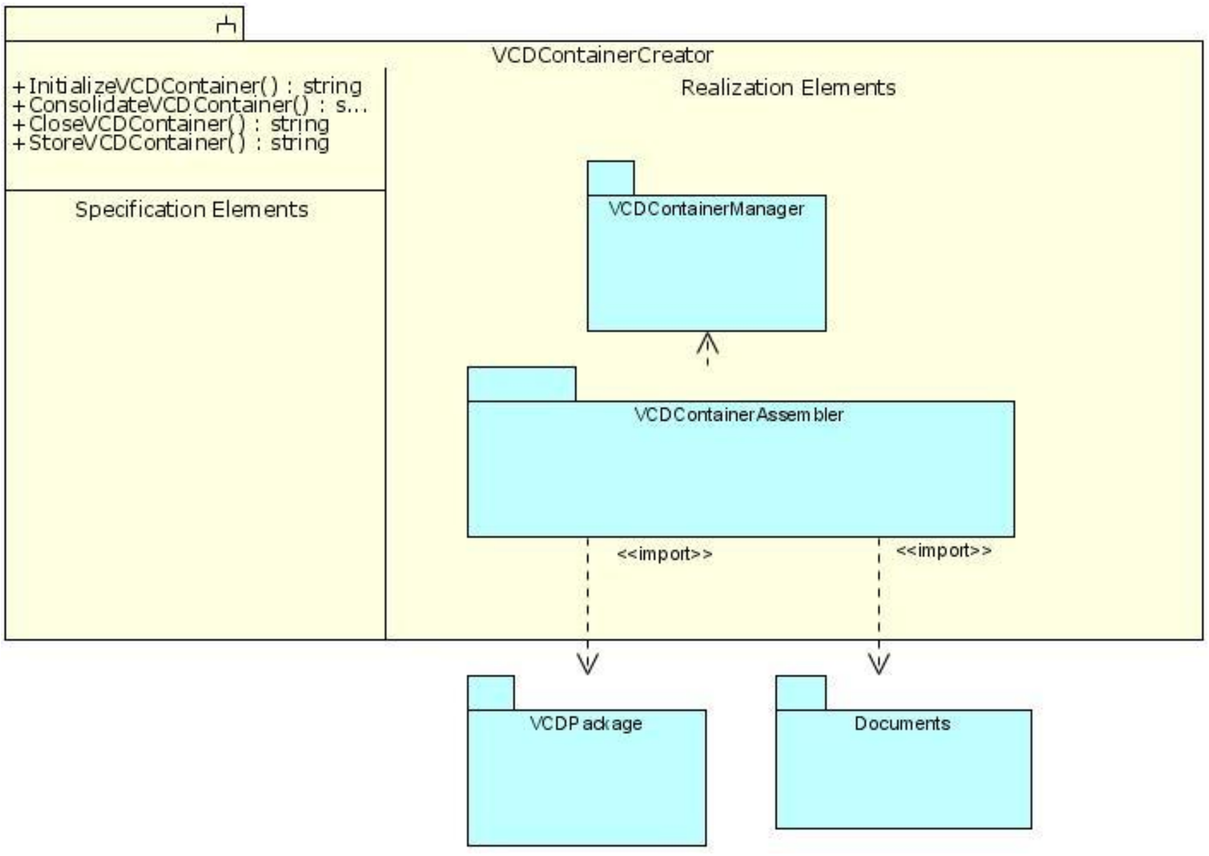

Figure 1-29: VCD Container creator

AssembleVCDContainer Use-Case

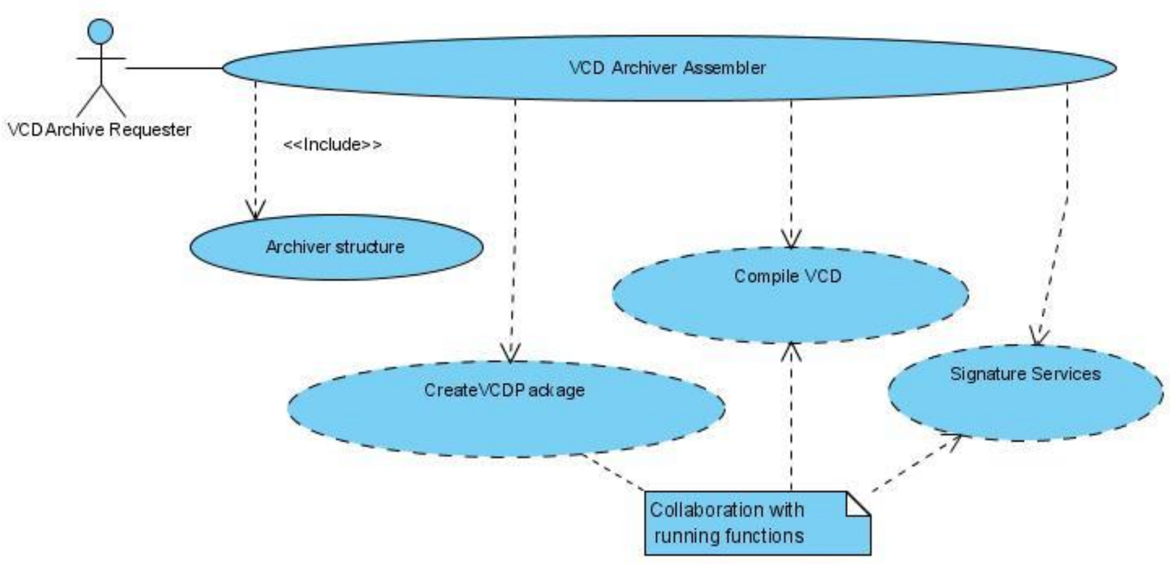

Figure 1-30: AssembleVCDContainer representation

Aspect	Description
Objective	To assemble all the components required to make a consistent delivery of a VCD package instance. The VCD package is assembled along with all the concurrent documents and files. The final object being a containerready for delivery and storage into any suitable media. The VCD Container must address and provide the correct facilities for a correct viewing of the complete package. The VCD Container must organize its contained parts and support the access to those items.
Results (postconditions in case of success)	A container based on multiformat components
Precondition	A System generatedevent driving the transition of a valid VCDPackage into the corresponding VCD Container. Configuration data file available.
Postcondition in case of failure	Error message returned
Actor(s)	VCD System
Initiating event	Request for a “VCD Container”.
Description of interaction procedure with VCD Service (standard run)	
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	Adopts a standardized structure; Conforms to container standards; Addresses Delivery and Management; Conforms to European initiatives on containers management; National VCD System to Euro VCD system interactions Allows VCD Containerinstances visualization Allows VCD Containers Digital Signature generation Allows VCD Containers Digital Signature Verification
LEG/ORG	LEG: the overall production requires legal support ORG: workflow model must adapt to the different delivery options.
Used in stage	Stage 2 to 4

Table 1-19: AssembleVCDContainer

ManageVCDContainer Use-Case

Handles the post generation stages of a VCD Container. This includes storage (when offered by the service) and delivery.

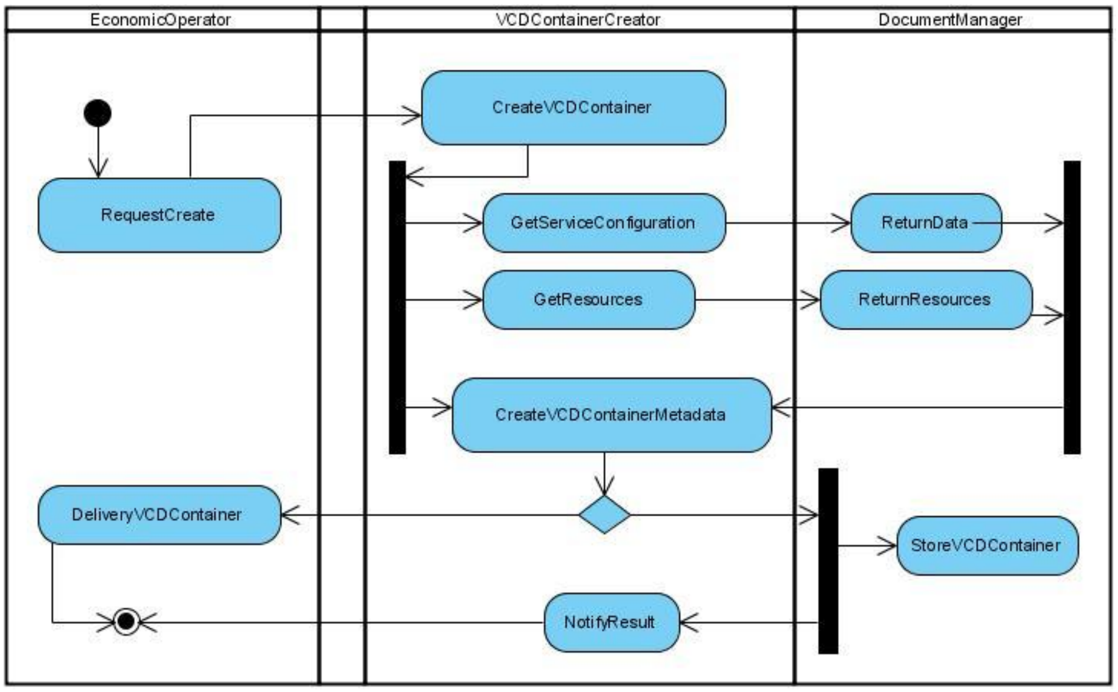

Figure 1-31: pre and post generation stages of a VCD Container

1.1.8 Use-Case group “VCD transportation interface incl. authentication”

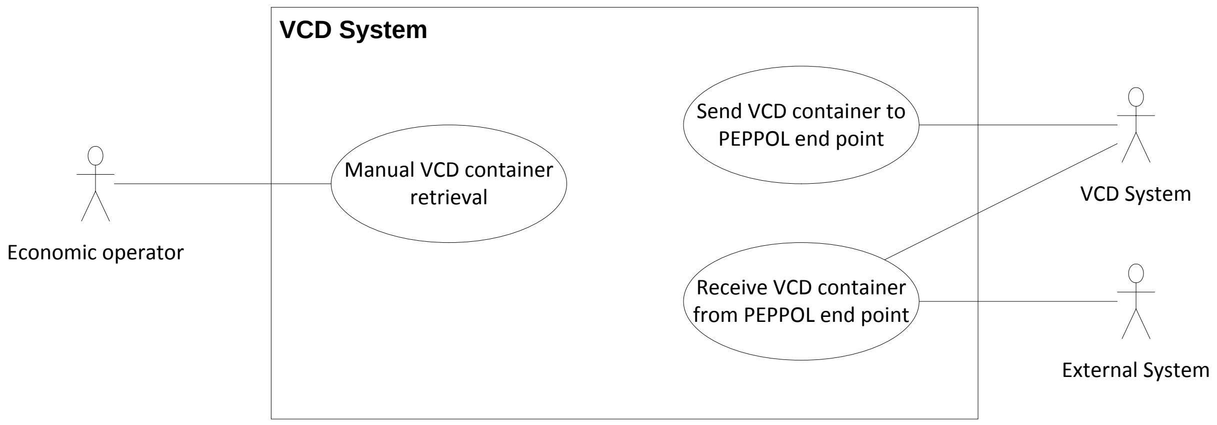
VCD System
Send VCD container to PEPPOL end point
Manual VCD container retrieval
Receive VCD container from PEPPOL end point
Economic operator
VCD System
External System

Figure 1-32: Use-Cases of the VCD system (VCD transportation interface incl. authentication)

Use-Case “Manual VCD container retrieval”

Aspect	Description
Objective	The user should be enabled to „download” a VCD container from the VCD service
Results (postconditions in case of success)	The user has retrieved the VCD container and confirmed the retrieval.
Precondition	A VCD container has been generated and is ready for retrieval.
Postcondition in case of failure	The user has not retrieved the VCD container.
Actor(s)	Economic operator
Initiating event	A VCD container has been generated and is ready for retrieval.
Description of interaction procedure with VCD Service (standard run)	The system places the generated VCD container in a defined download area. The system notifies the user about the availability of the VCD container in the download area and provides an access-link to this VCD container. The user follows this link, logs into the system and downloads the VCD container. The user confirms the successful download by filling in a checkbox in a user form and submits the confirmation.
Description of interaction procedure with VCD Service (alternative runs)	[optional] The system deletes the VCD container data from the temporary storage and from the download area.
Extension(s)	None
LEG/ORG	
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-20: Manual VCD container retrieval

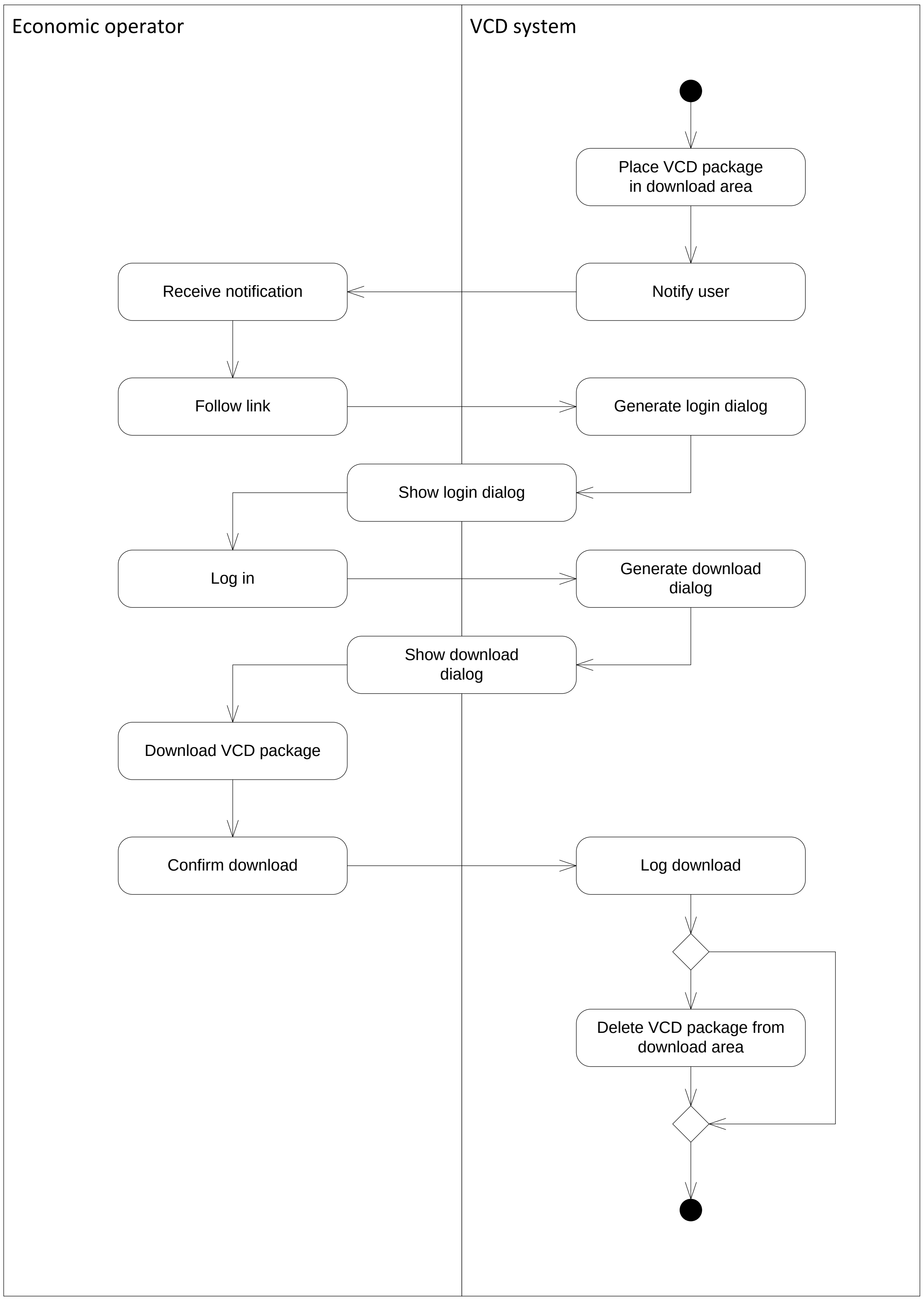

Figure 1-33: Activity diagram for use case “Manual VCD container retrieval”

Input-Output data:

Activity	Input	Output	User/System
Place VCD container in download area	VCD container	VCD container in download area	VCD system
Notify user	VCD container in download area	User notification (e.g. E-Mail)	VCD system
Receive notification	User notification (e.g. E-Mail)	None	Economic operator
Follow link	User notification (e.g. E-Mail)	Request to download area	Economic operator
Generate login dialog	Request to download area	Login screen	VCD system
Show login screen	Login screen	Login screen	VCD system
Login	Login screen, login data	Login request	Economic operator
Generate download dialog	Login request	Download UI	VCD system
Show download dialog	Download UI	Download UI	VCD system
Download VCD container	Download UI	VCD container	Economic operator
Confirm download	Download UI	User Confirmation	Economic operator
Log download	User Confirmation	User log entry	VCD system
Delete VCD container from download area	User Confirmation	Cleaned up download area	VCD system

Send VCD Container to Peppol end point

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

procedure with VCD Service (alternative runs)	
Extension(s)	The Recipient (Peppol EndPoint) returns confirmation The System is capable to detect an incoming confirmation message
LEG/ORG	
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-21: Send a VCD Container to PEPPOL Busdox

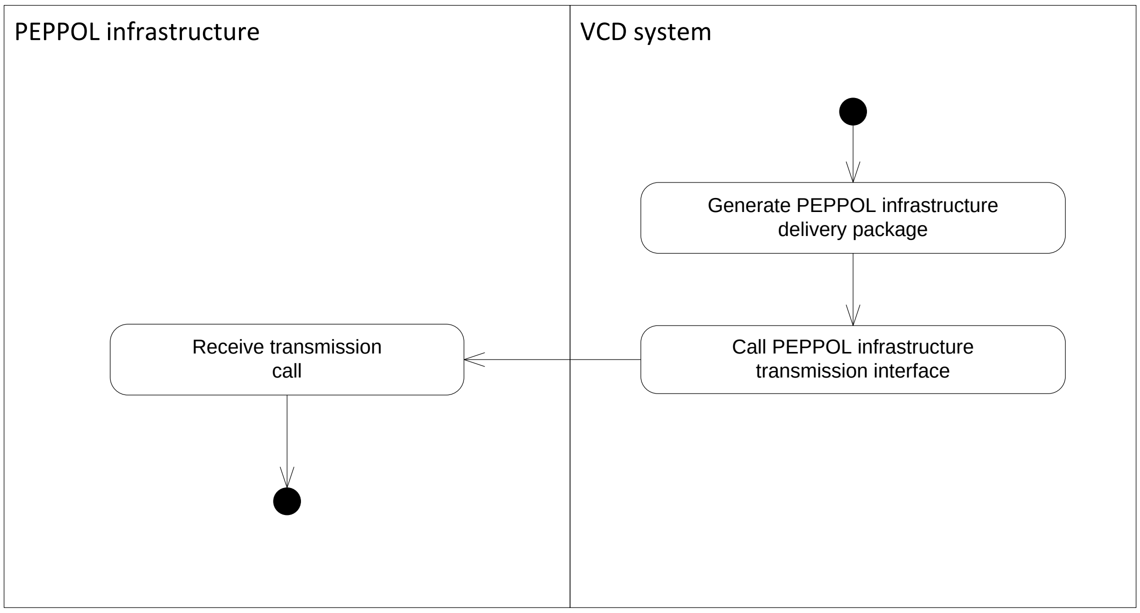
PEPPOL infrastructure
VCD system
Generate PEPPOL infrastructure delivery package
Call PEPPOL infrastructure transmission interface
Receive transmission call

Figure 1-34: Activity diagram for use case “Deliver VCD container to PEPPOL end point”

Input-Output data:

Activity	Input	Output	User/System
Generate PEPPOL infrastructure delivery container	VCD container	PEPPOL infrastructure delivery container	VCD system
Call PEPPOL infrastructure transmission interface	PEPPOL infrastructure delivery container	PEPPOL infrastructure delivery container	VCD system
Receive transmission call	PEPPOL infrastructure delivery container	PEPPOL infrastructure delivery container	PEPPOL infrastructure

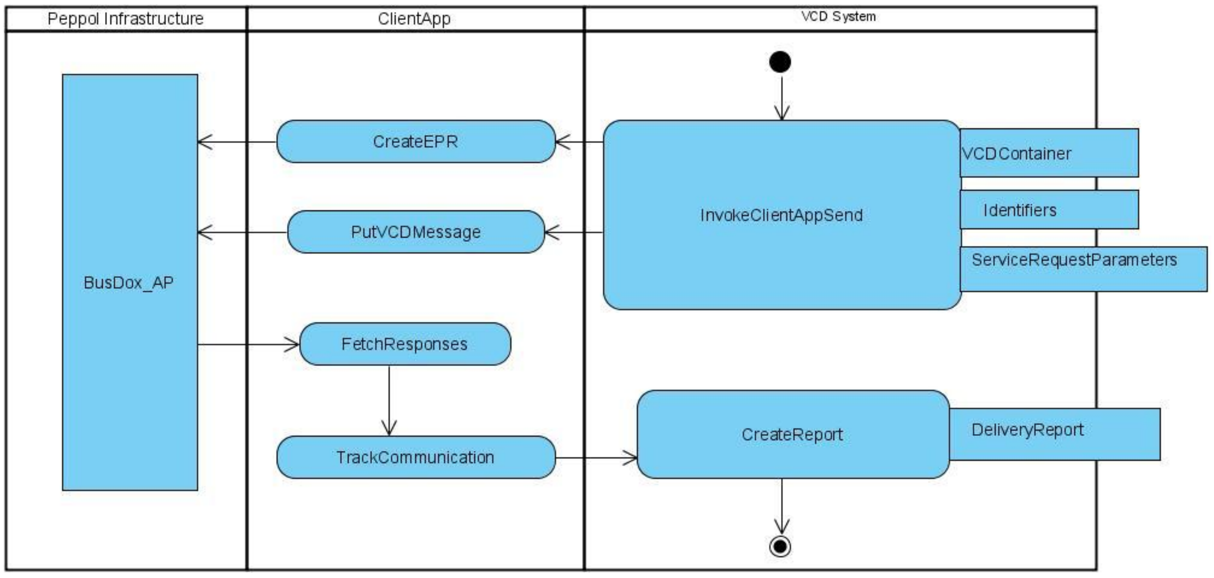

Figure 1-35: Sending a VCDContainer (NEW)

Receive VCD Container from Peppol end point

Aspect	Description
Objective	A VCD service is able to receive a VCD container from a PEPPOL end point user/system.
Results (postconditions in case of success)	The VCD container has been successfully fetched from the PEPPOL transport infrastructure. A confirmation (Acknowledge) message is returned
Precondition	A VCD container has been sent by a peer system to a valid recipient.
Postcondition in case of failure	The VCD container has not been successfully fetched by the Client Application over to the PEPPOL transport infrastructure (partially received) A negative confirmation (Negative-Acknowledge) message is returned
Actor(s)	System
Initiating event	A VCD container has been sent by a peer system and is ready for retrieval.
Description of interaction procedure with VCD Service (standard run)	The VCD system calls the PEPPOL infrastructure Client Application in order to FETCH a valid BusDox message. (Browsing for available messages or similar operations have to be designed according to the BusDox Specifications). The VCD Container is taken from the transport message. The VCD Container is validated (optional operation)
Description of interaction procedure with VCD Service (alternative runs)	
Extension(s)	VCD Container is validated
LEG/ORG	
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-22: Receive VCD Container from PEPPOL Busdox

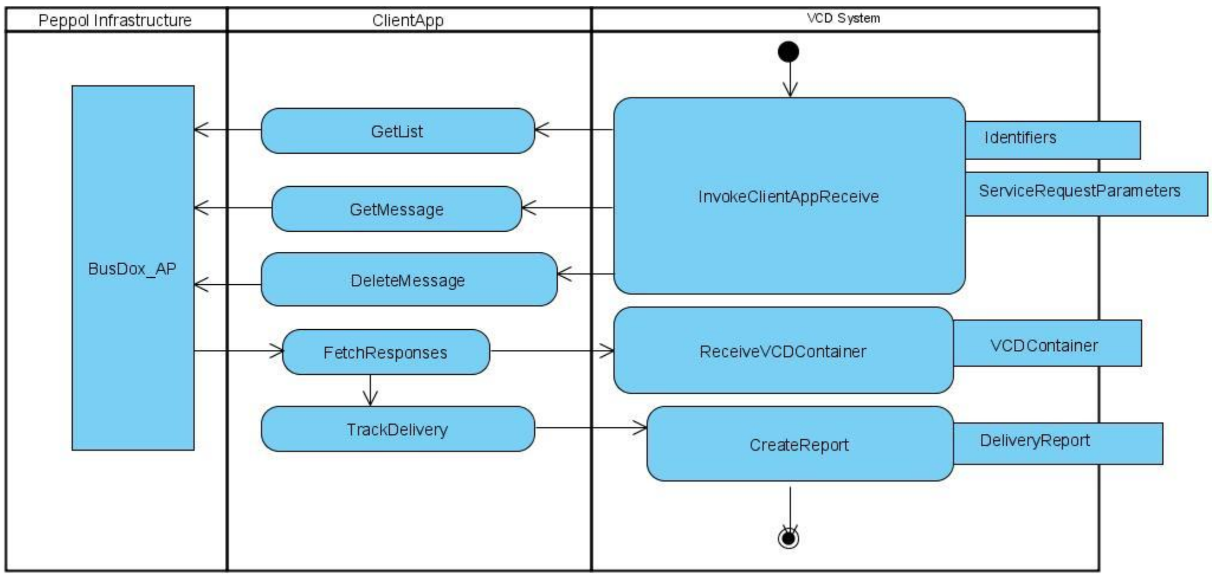

Figure 1-36: Receiving a VCDContainer

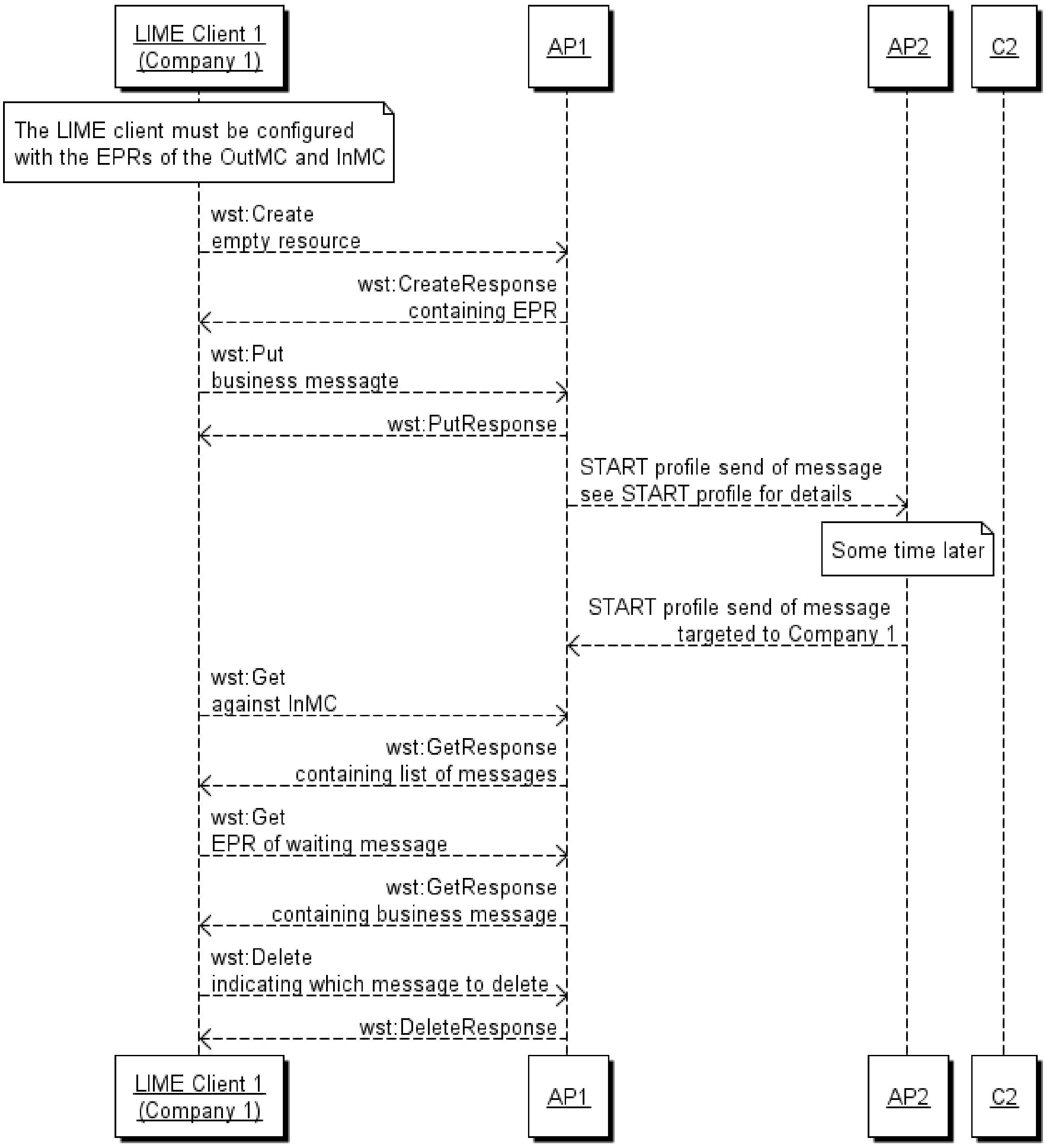
LIME Client 1 (Company 1)
AP1
AP2
C2
The LIME client must be configured with the EPRs of the OutMC and InMC
wst:Create empty resource
wst:CreateResponse containing EPR
wst:Put business messagte
wst:PutResponse
START profile send of message see START profile for details
Some time later
START profile send of message targeted to Company 1
wst:Get against InMC
wst:GetResponse containing list of messages
wst:Get EPR of waiting message
wst:GetResponse containing business message
wst:Delete indicating which message to delete
wst:DeleteResponse
LIME Client 1 (Company 1)
AP1
AP2
C2

Figure 1-37: Busdox LC capability

The description of the flow is given by the "LIMEProfile.doc" located at: https://svn.forge.osor.eu/svn/peppol/Documents/Specifications/

1.1.9 Use-Case “Interface to VCD dictionary (incl. Code lists, schemas...)”

The national VCD Service Providers will frequently need to verify that they are using up-to-date material, like code lists, identifiers, references, schemas, explanatory texts; they therefore need to have access to central ESP sources according to established procedures. The up-dating can be periodic and automated, but it can also be initiated when serving a user with special needs, like code list with texts for content of odd evidential documents.

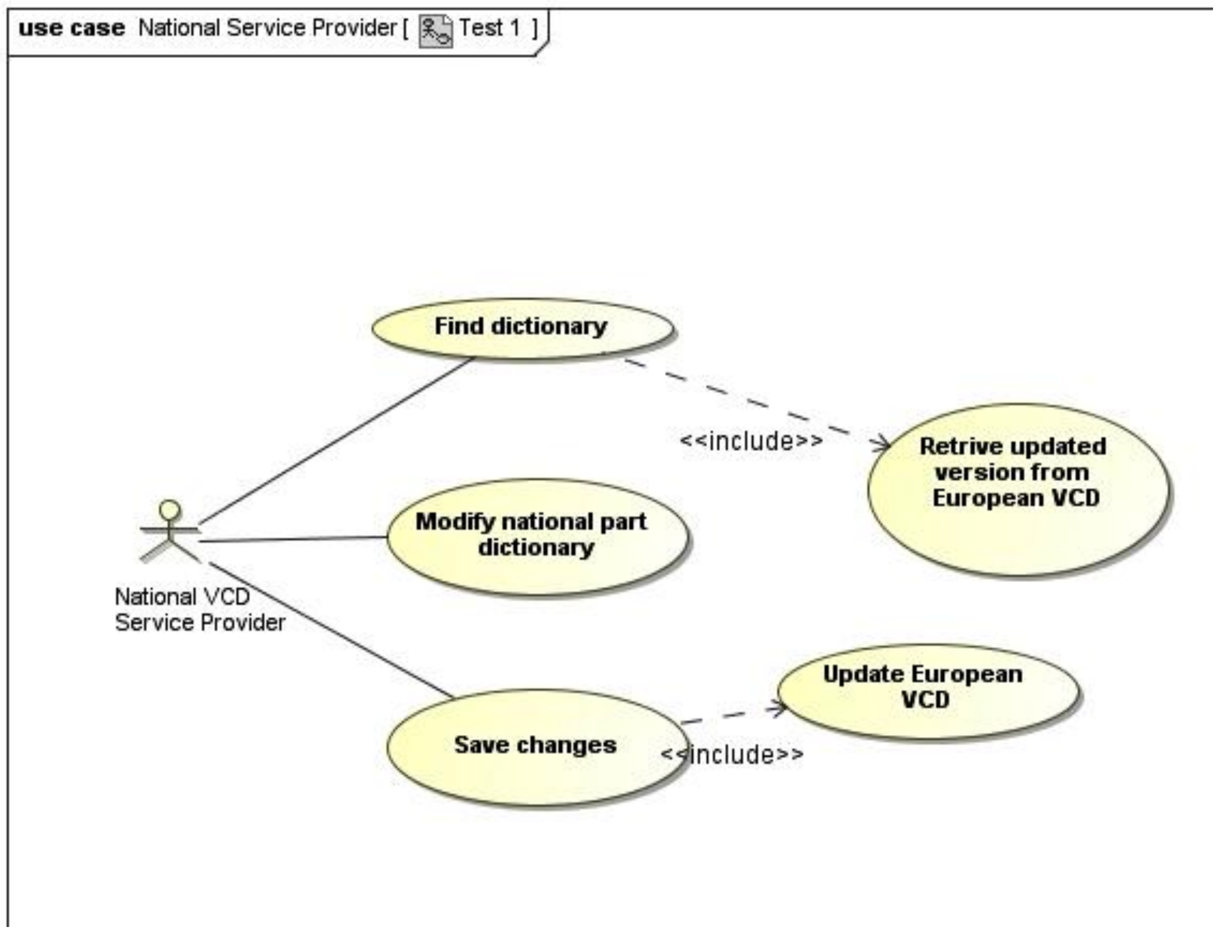
use case National Service Provider [Test 1]
Find dictionary
<<include>>
Retrive updated version from European VCD
Modify national part dictionary
National VCD Service Provider
Update European VCD
Save changes
<<include>>

Figure 1-38: Use-Case National Service provider

Use-Case descriptions

VCD Dictionaries containing code lists for evidences, lists of available versions, lists of issuing services, applications and their versions, etc will be subject to maintenance and changes. Interoperability requires that exchanged values are interpreted the same way, and all VCD Service Providers therefore need to have access to shared dictionaries through a common interface.

Use-Case “Interfacing between European VCD and national VCD Service Providers”

Aspect	Description
Objective	To have common dictionaries for all national VCD Service Providers
Results (postconditions in case of success)	Cross-border interoperability
Precondition	The different systems of VCD Service Providers have applications that share data structures.
Postcondition in case of failure	No cross-border interoperability
Actor(s)	European VCD Service and any VCD Service providers.
Initiating event	Initiation of cross-border VCD services
Description of interaction procedure with VCD Service (standard run)	Retrieve updated dictionary from European VCD Services. Modify national part of dictionary. Save changes to European VCD.
Description of interaction procedure with VCD Service (alternative runs)	No alternative
Extension(s)	
Used in Stage	Needed in stage 1 and onwards.
LEG/ORG reference	

Table 1-23: Interfacing between European VCD and national VCD Service Providers

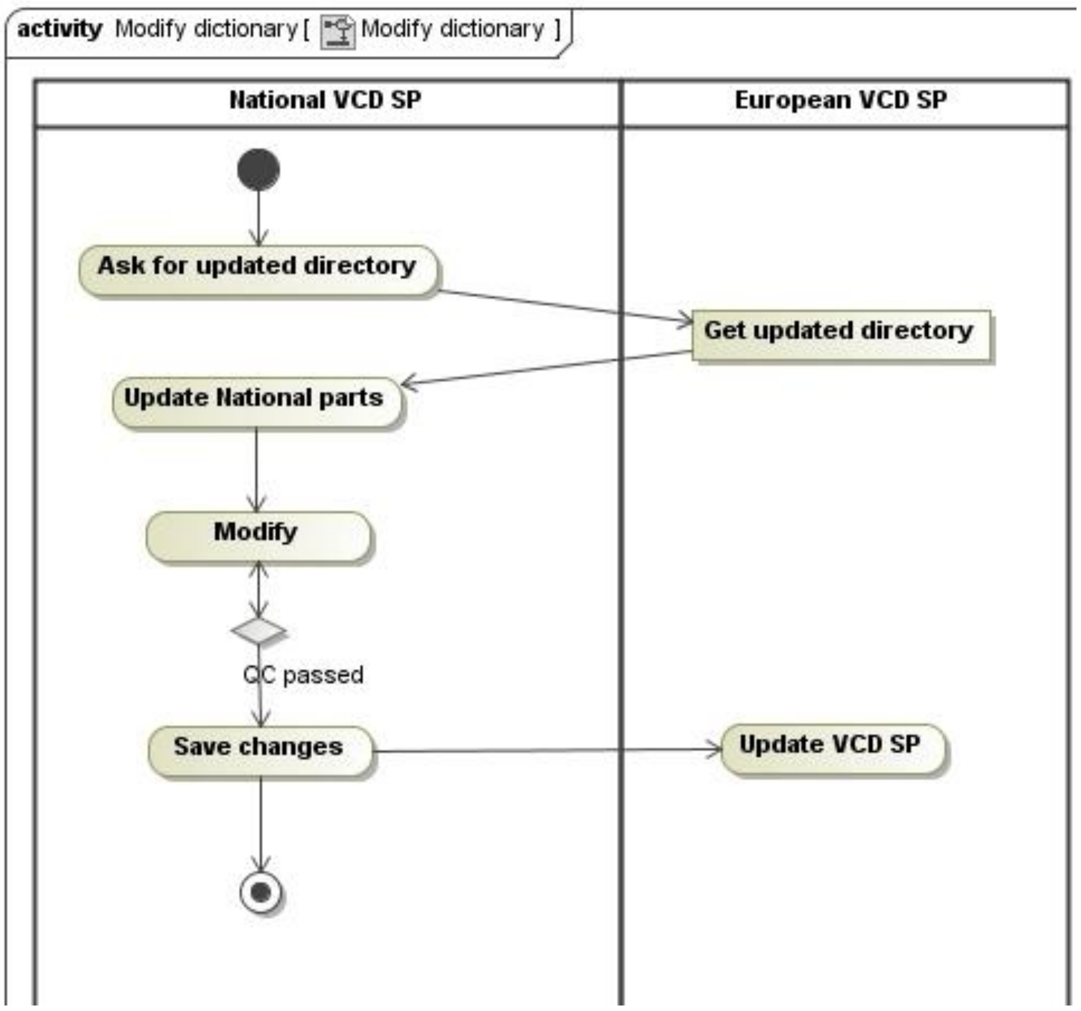

Figure 1-39: Modify dictionary

Activity	Input	Output	User/System
Modify directory	Existing directory	Updated directory	VDC SP

Storage of common code lists

The storage of common code lists can be done on a web-server with public reading access in form of xml-files. These files must be valid with regard to a defined schema. The VCD services and the ESP can then easily access these code lists either on the fly or periodically.

1.1.10 Use-Case Group “Interface to PEPPOL registries”

Use-Case diagram

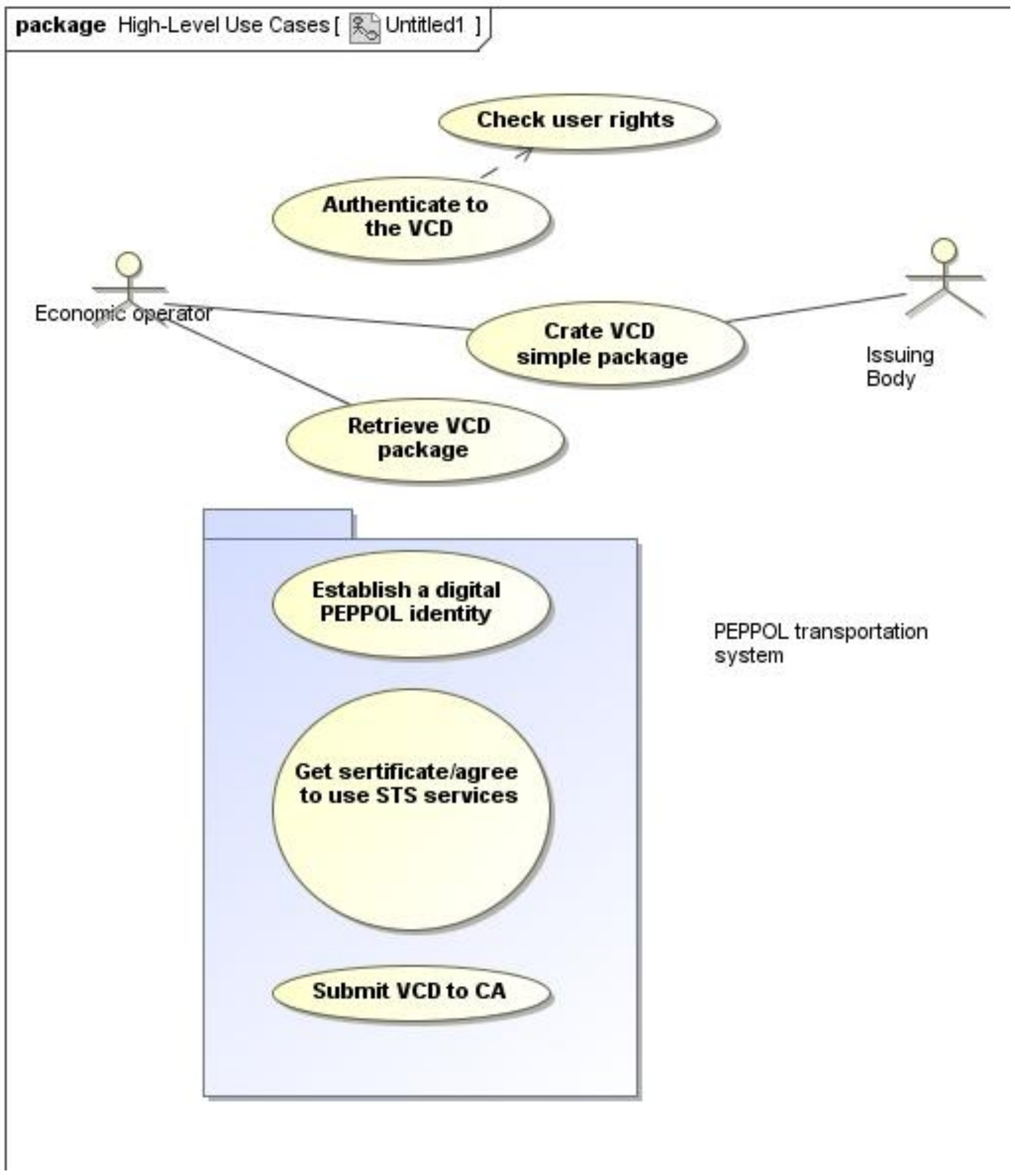
package High-Level Use Cases [Untitled1]
Check user rights
Authenticate to the VCD
Economic operator
Crate VCD simple package
Issuing Body
Retrieve VCD package
Establish a digital PEPPOL identity
Get sertificate/agree to use STS services
Submit VCD to CA
PEPPOL transportation system

Figure 1-40: High-Level Use-Case

Use-Case descriptions

When using the PEPPOL infrastructure to forward a VCD package data needed for correct transport will be needed; the VCD SP therefore need to have access to up-to-date PEPPOL SML registries to retrieve valid identifiers and code lists.

Aspect	Description
Objective	To enable sending of VCD Packages through the PEPPOL infrastructure.
Results (postconditions in case of success)	VCD packages can be sent through the PEPPOL infrastructure.
Precondition	Existence of VCD packager
Postcondition in case of failure	VCD packages will be transmitted in other infrastructures.
Actor(s)	VCD Service providers, European VCD Services and PEPPOL infrastructure manager .
Initiating event	Existence of a VCD SP application.
Description of interaction procedure with VCD Service (standard run)	Send query with sender's and receiver's identifier and identifier scheme Service Metadata LocatorReceive Recipient and Sender Identifier from Service Metadata Publisher Create BUSDOX message with identifiers Submit
Description of interaction procedure with VCD Service (alternative runs)	1 Manual query or look-up in PEPPOL registries from VCD SP
Extension(s)	
Used in Stage	All stages
LEG/ORG reference	

Figure 1-41: Interfacing PEPPOL/SML

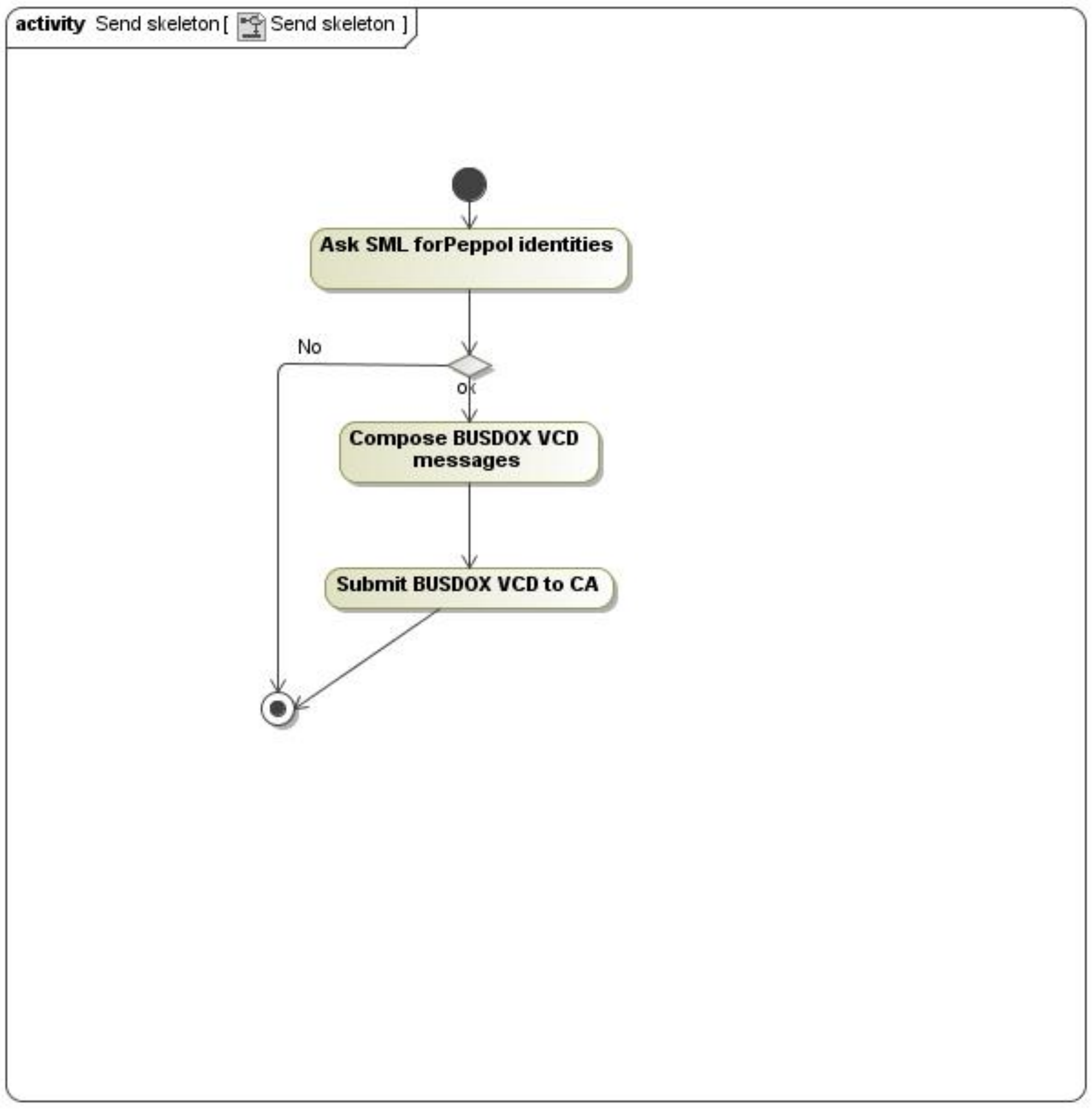
activity Send skeleton[Send skeleton]
Ask SML forPeppol identities
No
ok
Compose BUSDOX VCD messages
Submit BUSDOX VCD to CA

Figure 1-42: Send Skeleton Activity Diagram

Activity	Input	Output	User/System
Obtain BUSDOX identifier	Request for identifier from SML	Endpoint identifier	VCD SP

1.1.11 Use-Case Group “Interface to European VCD service”

The system interaction between the VCD service and European VCD service does not include any user interaction. The use cases described here therefore describe the system interaction having the systems as actors.

Use-Case diagram

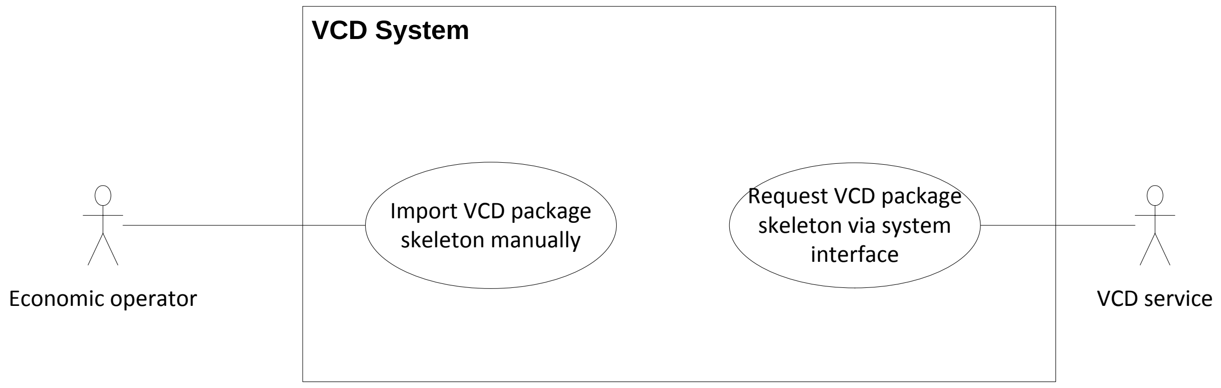
VCD System
Import VCD package skeleton manually
Request VCD package skeleton via system interface
Economic operator
VCD service

Figure 1-43: Use-Case of the VCD system (VCD transportation interface incl. authentication)

Use-Case descriptions

Aspect	Description
Objective	The economic operator requests a VCD package from the VCD service provider. In order to come up with the list of evidences needed in this case, the VCD system asks the European VCD service to pre-generate a VCD package skeleton including the list of economic operator national evidences.
Results (postconditions in case of success)	The VCD system has retrieved a VCD package skeleton from the European VCD service including the list of economic operator national evidences.
Precondition	The economic operator has provided all information needed as initiating data (formalized within the ontology, e.g. tender structure, country of the contracting authority…).
Postcondition in case of failure	The VCD system has not retrieved a VCD package skeleton from the European VCD service.
Actor(s)	VCD system
Initiating event	The VCD system needs to place an order at the European VCD service for a VCD package skeleton.
Description of interaction procedure with VCD Service (standard run)	The VCD system collects the initiating data and constructs the VCD package skeleton order request (this request should use (a derivation of) the VCD package schema much like the VCD package skeleton). The request also includes an indicator, whether all possible economic operator nation evidences for a specific criterion should be included in the VCD package skeleton or only selected ones. The latter case requires a user interaction at the European service provider. The VCD system places the VCD package skeleton order request using a system interface at the European VCD service.

	The VCD system waits for the order processing by the European VCD service. The VCD service receives the resulting VCD package skeleton. The VCD system validates the VCD package skeleton. In case of positive validation, the system stores the VCD package skeleton data temporarily.
Description of interaction procedure with VCD Service (alternative runs)	6.a) In case of negative validation, the system returns an error code to the European VCD service.
Extension(s)	None
LEG/ORG	
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-24: Interface to European VCD service

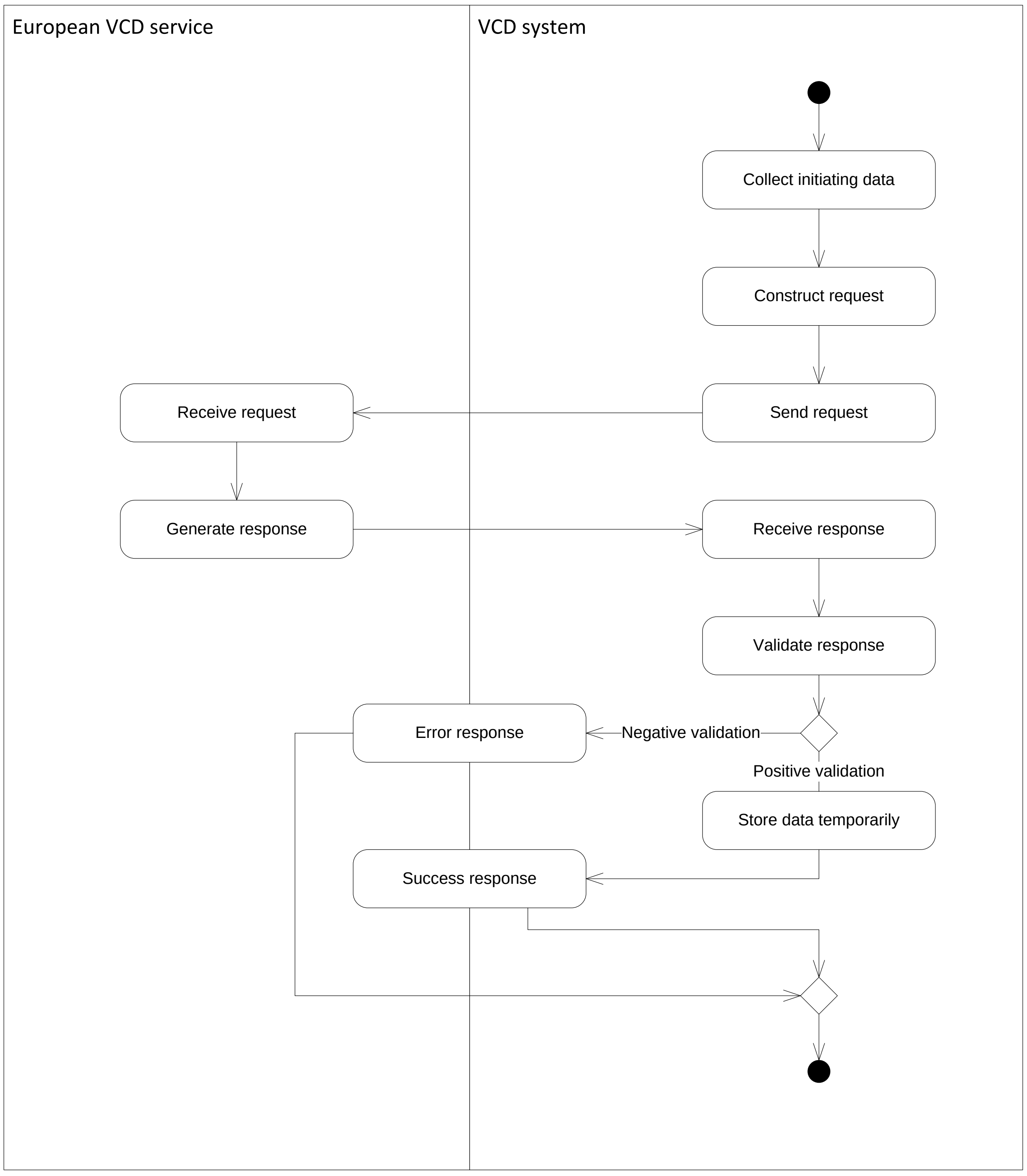

Figure 1-44: Activity diagram for use case “Request VCD package skeleton via system interface”

Input-Output data:

Activity	Input	Output	User/System
Collect initiating data	None	Tenderer structure Nationalities of Tenderer and Contracting authority	VCD system
Construct request	Tenderer structure	SOAP-Request-	VCD system

	Nationalities of Tenderer and Contracting authority	Message	
Send Request	SOAP-Request-Message	SOAP-Request-Message	VCD system
Receive Request	SOAP-Request-Message	SOAP-Request-Message	European VCD service
Generate Response	SOAP-Request-Message	SOAP-Response Message	European VCD service
Receive Response	SOAP-Response Message	SOAP-Response Message	VCD system
Validate response	SOAP-Response Message	SOAP-Error Message or SOAP-Success-Message	VCD system
Error Response	SOAP-Error Message	None	European VCD service
Success Response	SOAP-Success-Message	None	European VCD service
Store data temporarily	SOAP-Response Message	Local ontology data	VCD system

Aspect	Description
Objective	The economic operator should be enabled to upload a VCD package skeleton to a VCD service.
Results (postconditions in case of success)	The VCD service has imported the VCD package skeleton provided by the user.
Precondition	The user possesses a valid VCD package skeleton. The user is logged into the system
Postcondition in case of failure	The VCD service has not imported the VCD package skeleton provided by the user.
Actor(s)	Economic operator
Initiating event	The economic operator wants to request the compilation of a VCD package based on an existing VCD package skeleton.
Description of interaction procedure with VCD Service (standard run)	The system provides user form which enables the user to upload a VCD package skeleton The user selects the local file containing the VCD package skeleton and submits the request The system validates the VCD package skeleton In case of positive validation, the system stores the VCD package skeleton data temporarily.
Description of interaction procedure with VCD Service (alternative runs)	4.a) In case of negative validation, the system informs the user about the error.
Extension(s)	None
LEG/ORG	
Used in stage	This use-case is relevant for the stages 2, 3 and 4

Table 1-25: Import VCD package skeleton manually

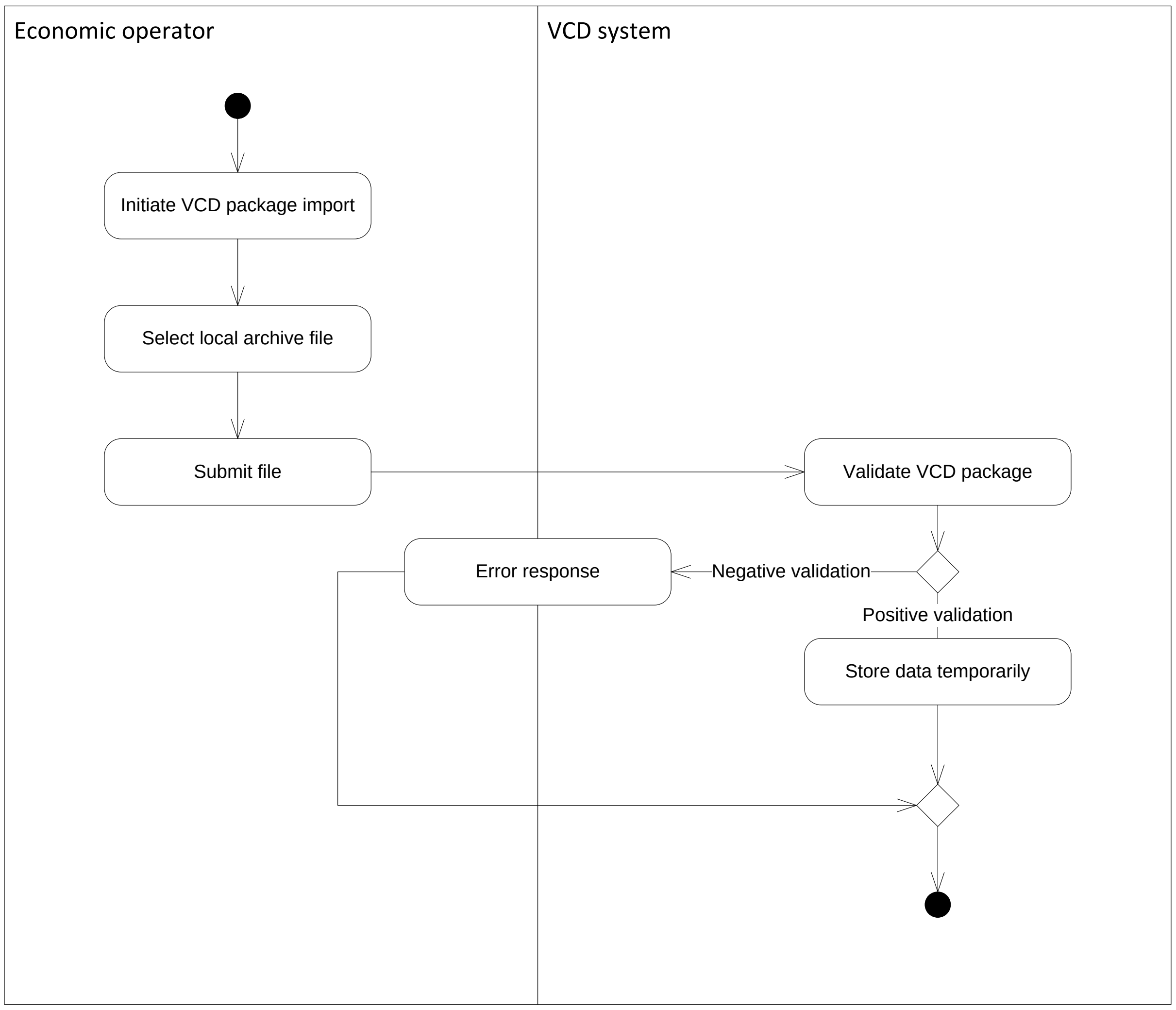

Figure 1-45: Activity diagram for use case “Import VCD package skeleton manually”

Input-Output data:

Activity	Input	Output	User/System
Initiate VCD package import	None	None	Economic operator
Select local container file	None	Local container file	Economic operator
Submit file	Local container file	Local container file	Economic operator
Validate VCD package	Local container file	Error response or local container file	VCD system
Error response	Error response	None	VCD system
Store data temporarily	container file	Local ontology data	VCD system

1.1.12 Use-Case-group “interface to notification systems”

TED and national publication systems have interfaces that make them accept structured announcements from contracting authorities and deliver structured files to the systems used by economic operators. Contracting authorities find services on http://simap.europa.eu/ojs_esenders/list_of_ojs_esenders/index_en.htm, these are also rendering services to Economic Operators by alerting on CPV codes and giving guidance on how to reply.

More than 90 % of all notices sent to TED are in structured XML format according to DTD from more than sixty services in EEA. The CIRCA project of the Publication Office is working on the upgrading to XML Schema, which will be implemented and piloted by the end of 2010; public information about the project’s results and progress is available from the TED eSender documentation folder on http://circa.europa.eu/Public/irc/opoce/eproc/library?l=/public/1_general&vm=detailed&sb=Title.
The documentation contains specification of XML Schemas for interfacing, further it specifies several identifiers that are of importance for VCD systems, and also methods and procedures for testing to become accepted partner of TED are detailed.
The eSender specifications cover presentation of fixed texts of the CFT form in all official EU languages (not Norwegian).
In order to interface TED the relevant VCD systems need match parts of the eSender XML Schema specifications. Further work therefore needs to be constrained by the referred documentation.

Overview

The general description has been used to derive Figure 1-46. The drawing shows how contracting authorities and economic operators are interoperating through TED; the parties are encircled in Figure 1-46.

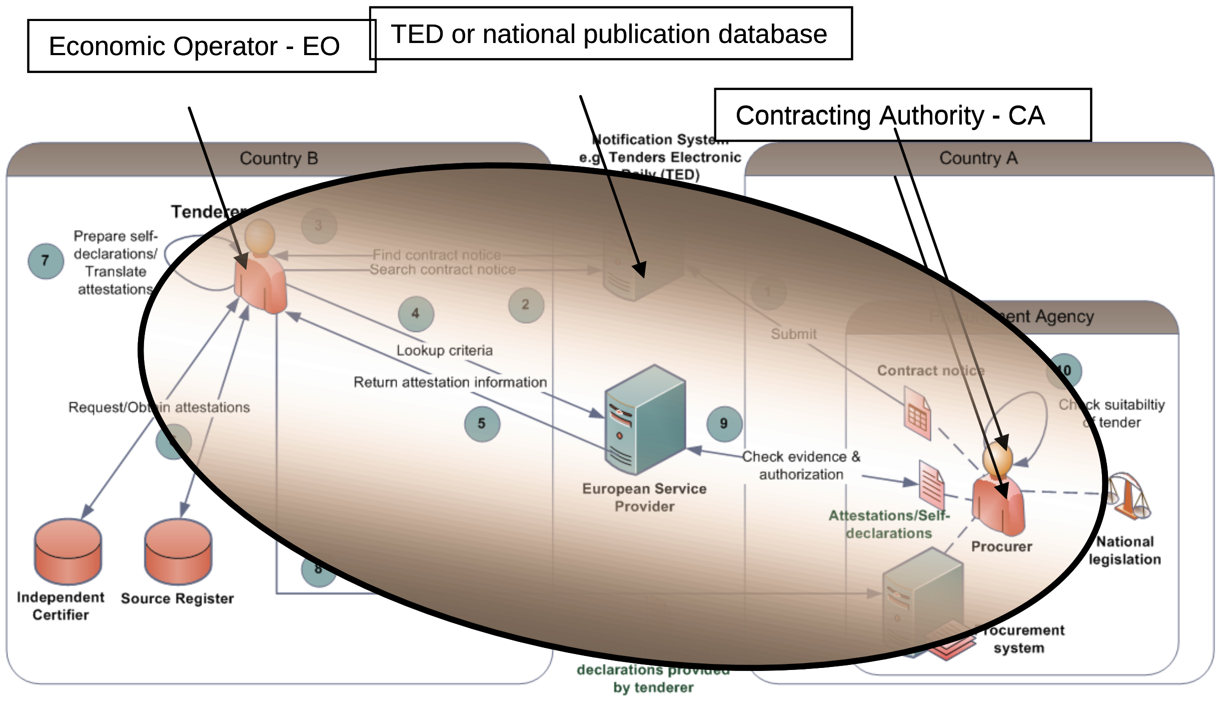
Economic Operator - EO
TED or national publication database
Contracting Authority - CA
Country B
Country A
Notification System e.g. Tenders Electronic Daily (TED)
Tenderer
Prepare self-declarations/ Translate attestations
Find contract notice
Search contract notice
Lookup criteria
Return attestation information
Request/Obtain attestations
Submit
Contract notice
Check suitabiltiy of tender
Check evidence & authorization
European Service Provider
Attestations/Self-declarations
Procurer
National legislation
Procurement system
declarations provided by tenderer
Independent Certifier
Source Register
1
2
3
4
5
7
8
9
10

Figure 1-46: Both CA and EO need a structured TED interface

National publication databases will interface TED by using the eSender specifications for:
– Interfacing

- Sharing of data and identifiers
- Testing and Conformance Assessment
- Presentation in more official languages

Both main parties need to interface TED by use of its specifications; the VCD Service Provider will need to interoperate with Contracting Authority, Economic Operator and TED, hence the TED interface specification will become essential for interoperability within its environment and must be implemented by the VCD SPs to create its interoperability.
The figure clearly shows how systems of the contracting authority and the economic operator share information in TED, and in order to become an intermediary service the VCD SP systems also need to become a sharing partner by interfacing TED.

Activity Diagram

The activity diagram shows how the Economic Operator need a TED interface to retrieve data from the Call for Tender (CFT) in TED; some of the data retrieved will be used to specify the VCD skeleton, others to label it and create time line for the processing.
Having built the skeleton package the VCD SP will be able to collect the evidences form the Issuing Bodies (IB), and the construct the complete VCD package to be submitted to the Contracting Authority (CA).

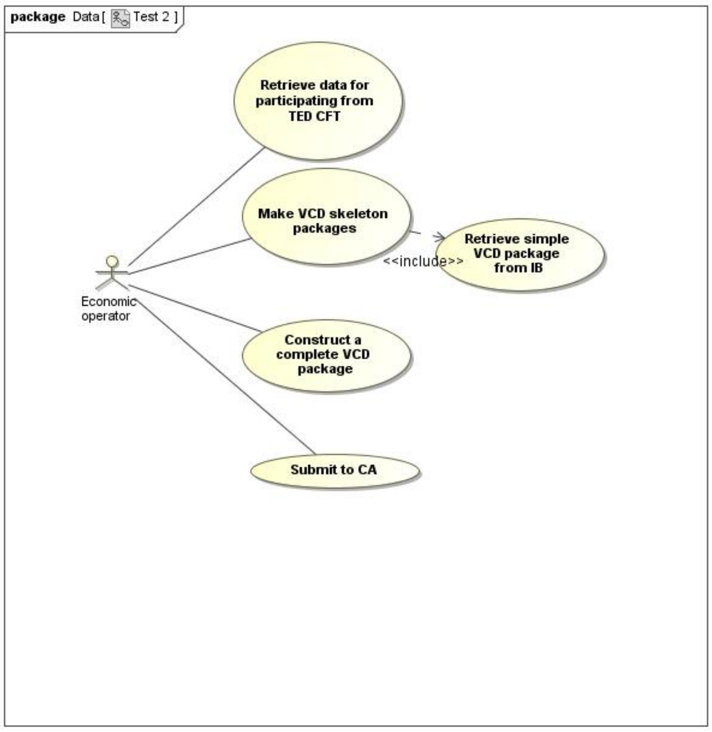
package Data [Test 2]
Retrieve data for participating from TED CFT
Make VCD skeleton packages
<<include>>
Retrieve simple VCD package from IB
Economic operator
Construct a complete VCD package
Submit to CA

Figure 1-47: Activity diagram

Functions to be interfaced

Overall description of the notification structure to be interfaced

Economic operators are using services that alert them when section II.1.5 of calls for tender contain given CPV codes. If the EO wants to prepare a response, the services extract information from header and sections in notices in TED:

- From header:

- Date of publication
- TED registration number
- Type of contract
- Type of procedure
- Country and Region of contracting authority
- Type of publication

– From section I.1:
  - Name of Contracting of contracting authority – likely to deviate from the one in the business register Address of contracting authority to be used for the tendering process – likely to deviate from the official one and the one used for other procurement processes.
  - Names, phone numbers and e-mail addresses of contact persons.

– From section II.1.1:
  - Contracting authority's reference number of tender – will also be used to label contract and as reference number later.
  - Contracting authority's naming of the case and the contract.

– From section II.1.6:
  - CPV codes

– From section III.2.1:
  - Names of evidential documents, like Company registration certificate, Value added tax certificate, Self declarations.

– From section IV.1:
  - Type of procedure.

– From section IV.2.1:
  - Time-limit for receipt of tenders or request to participate

– From section IV.3.1:
  - Administrative information: File reference number attributed by the contracting authority

– From section IV.3.6:
  - Language in which tenders or requests to participate may be drawn up. Minimum time frame during which the tenderer must maintain the tender: Duration on months.

– From section VI.2:
  - URL of additional information

The systems support the writing and inclusion of the tender documents by using the collected data, and they build up the entire file by attaching electronic copies of evidential documentation. The finished file will finally be made available for the contracting authority.

Full data models will belong to the eSender specifications, so far the following is available:

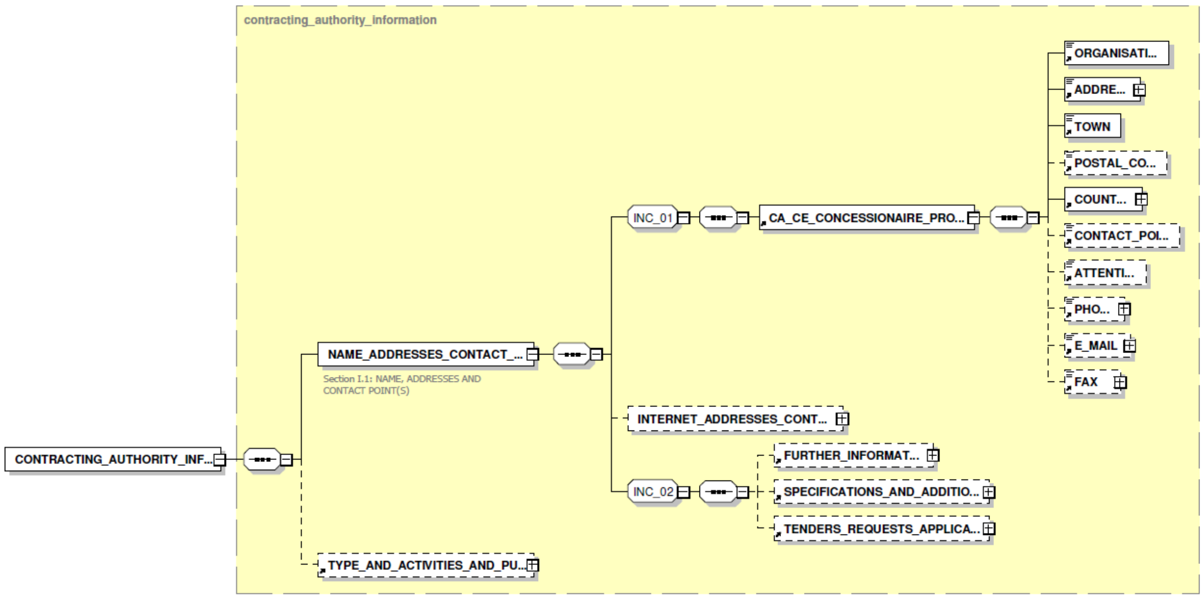
contracting_authority_information
CONTRACTING_AUTHORITY_INF...
NAME_ADDRESSES_CONTACT_...
Section I.1: NAME, ADDRESSES AND CONTACT POINT(S)
TYPE_AND_ACTIVITIES_AND_PU...
INC_01
CA_CE_CONCESSIONAIRE_PRO...
INTERNET_ADDRESSES_CONT...
INC_02
FURTHER_INFORMAT...
SPECIFICATIONS_AND_ADDITIO...
TENDERS_REQUESTS_APPLICA...
ORGANISATI...
ADDRE...
TOWN
POSTAL_CO...
COUNT...
CONTACT_POI...
ATTENTI...
PHO...
E_MAIL
FAX

Figure 1-48: Contracting Authority Information data model from eSender

There are many difficulties before to have a possibility to use the eSender XML message to Peppol purposes:
For the moment and it should be the same when the new schema will be in production (scheduled for end of 2010), the criterions (and requested evidences) are in a free text field and so are not structured to be computable.

The eSender XML message is an input message to TED not an output message (ie : not for the economic operator, there is an XML message (different from eSender XML message) for the “licence holders”, ie : companies who pay TED for this information in XML files format to build a subscription based service).

The new schema will be ready late for our purpose of using it for the pilots.

The only information we should have when the new schema will be in production with the proposed schema, is the tender information header and the CA information.

So that is why, we have proposed to use the fact than the CA are dealing with different tools to prepare their tender notices (CFT) to generate (through an evolution of their tools) a VCD pre-skeleton which will be joined to the list of tender documents. It could include all the information described in the VCD skeleton except the economic operator information.

In a far future, the same information could be retrieved automatically from an evolution of TED notices but the conditions are not met in 2010.

1.1.13 Functional specification

Core functions

Adaptive user form provision
Evidence data input user forms should be generated dynamically using a generic description of the requested data (GUI data schema). The GUI data schema could be derived out of the ontology. The user interface should only present those data fields to the user which are needed in the respective context and should also provide the user with context sensitive help. The user interface must be able

to represent all required data types which are needed for user input, including the possibility to upload local files.
The user interface must be WAI conform, at least WAI level A.

(Temporary) data storage
The system must provide means to store and persist the following data which is needed for VCD package compilation:

- Initial user input (economic operator data, tenderer structure, complexity drivers…)
- Extract of VCD package skeleton data
- Evidence data incl. files (from automated services as well as manual user input data)

Logging
The system must provide a logging mechanism which logs all important system interactions. Especially each user interaction which represents an approval of the user to a system generated result must be logged permanently.
The following data for each log entry should exist:

- Date/time of event
- Description of event
- Explicit user approval
- Source of event (system part or user id)

Input data validation

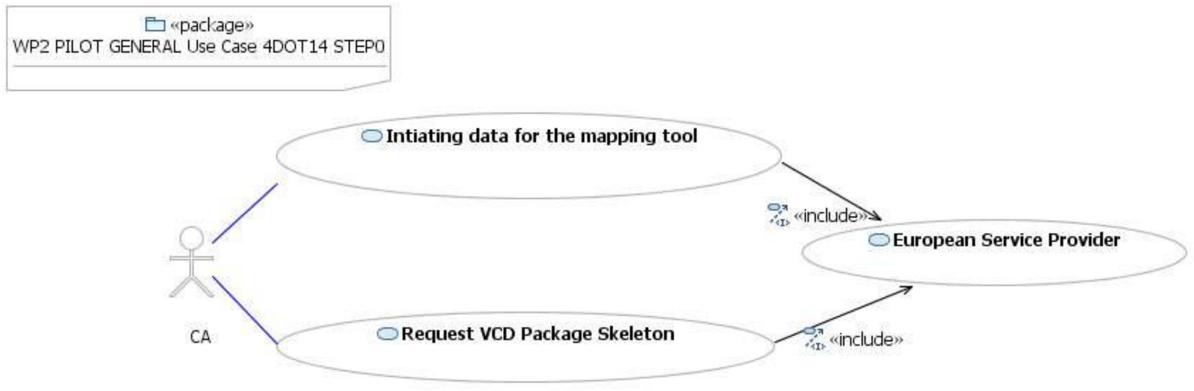
«package»
WP2 PILOT GENERAL Use Case 4DOT14 STEP0
Intiating data for the mapping tool
«include»
European Service Provider
CA
Request VCD Package Skeleton
«include»

Figure 1-49: Use-Cases of the CA preparing a VCD Skeleton

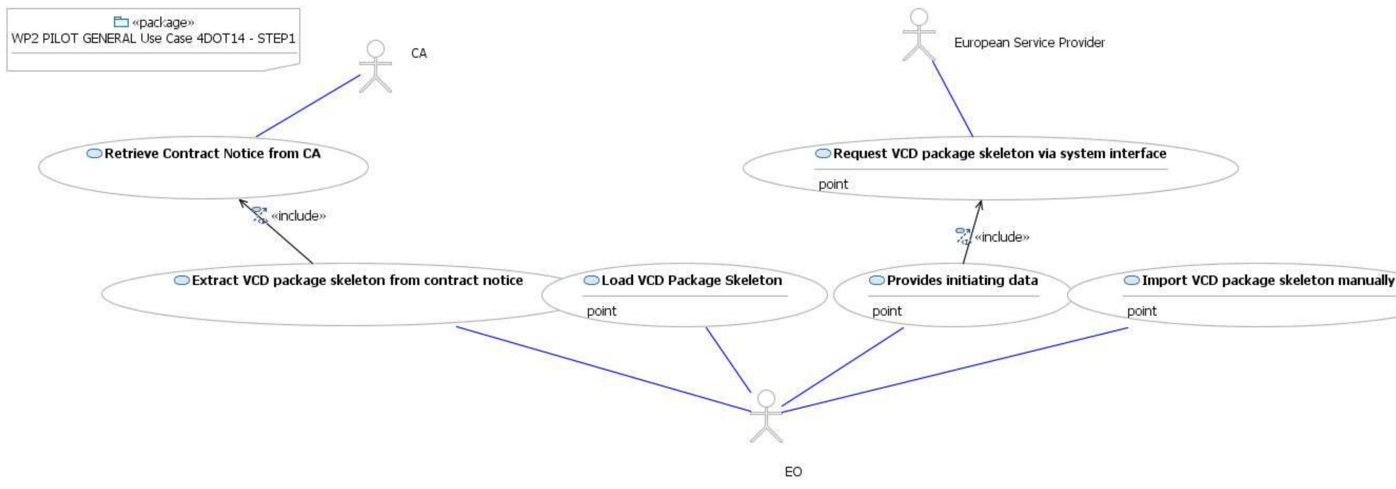
«package»
WP2 PILOT GENERAL Use Case 4DOT14 - STEP1
CA
European Service Provider
Retrieve Contract Notice from CA
Request VCD package skeleton via system interface
point
«include»
«include»
Extract VCD package skeleton from contract notice
Load VCD Package Skeleton
point
Provides initiating data
point
Import VCD package skeleton manually
point
EO

Figure 1-50: Use-Cases of the EO preparing/completing a VCD Skeleton

Validation of evidence data

The system must provide a validation mechanism which is capable of validating manual user input data with respect to the following aspects:

- Syntax (e.g. correct data type)
- Semantic correctness (e.g. check against the underlying ontology or a data schema)
- The system must inform the user about failed validity checks and should provide hints in the user interface for input correction (e.g. highlighting of erroneous input fields…)

Validation of VCD package skeletons

The system must provide a mechanism to validate a VCD package skeleton, which is either uploaded or retrieved via a system interface, with respect to following aspects:

- Syntax (correct XML files)
- Semantic (e.g. check against the ontology or a data schema)
- The system must inform either the user or the sending system interface about failed validity checks.

VCD download area incl. user notification and download interface

The system must have write access to a physical data space which is capable of storing VCD packages, which are ready for download.
The system must provide a notification service which informs a registered user about the completion of his requested VCD package. Further details about the VCD package are included in the notification, as well as a link to the download area.
The system must provide a user interface which enables the user to download the VCD package from the download area following the link in the notification. The user interface also must enable the user to approve the successful download of the VCD package.
The system may provide a functionality to optionally delete the VCD package after it has been successfully downloaded by the user.

PEPPOL transport infrastructure delivery interface

The system must implement a system interface to the PEPPOL transport infrastructure which is able to hand over a VCD package and all required addressee information in order to send the VCD package using the PEPPOL transport infrastructure.
Firstly, the system of the VCD SP needs to obtain recipient information from the SMP; this is done by asking the SML for the address of his SMP. To do so he needs to send a query with the recipients ID scheme identifier and his identifier using the SML's Discovery Interface, and the SML will return

contact information for the very SMP. Full documentation on http://www.peppol.eu/News/news-archive/peppol-reference-software-version-1.0-released.
A request to the SMP will return his endpoint identifier and also information about profile versions his systems can process. The information is useful for both the VCD SP processing and for the sending; the information is collected in the BUSDOX message that ensures correct transport, delivery and receipt.

1.2 ESP interface

The system should provide a system interface which passes those data to the ESP, which is necessary to retrieve a VCD package skeleton (including an evidence list for each required criterion per economic operator in the tenderer structure). The interface also includes an indicator, which informs the ESP whether all possible economic operator nation evidences for a specific criterion should be included in the VCD package skeleton or only selected ones. The schema of the interface payload should be a subset of the VCD package skeleton schema.
The system interface must be able to wait for the retrieval of the requested VCD package skeleton. The system must provide an escalation mechanism which notifies defined addresses if the pending time of the request exceeds a predefined time span.
The system interface must be able to retrieve the requested VCD package skeleton.
The system interface must be able to use the validation mechanisms described in 0.

1.3 VCD package skeleton upload interface

The system should provide a user interface which is capable of uploading VCD package skeletons by the user.
The system interface must be able to use the validation mechanisms described in 0.

1.3.1 VCD Viewer

The VCD Viewer is an application for the visualization of the VCD package and the VCD package skeleton as well as their contents. Figure 1-51 shows the three main situations in which a VCD Viewer is needed by economic operators, VCD service providers as well as by contracting authorities to view a VCD package. Figure 1-52 shows the two main situations in which a VCD Viewer is needed by economic operators as well as by VCD Service Providers to view a VCD package skeleton.

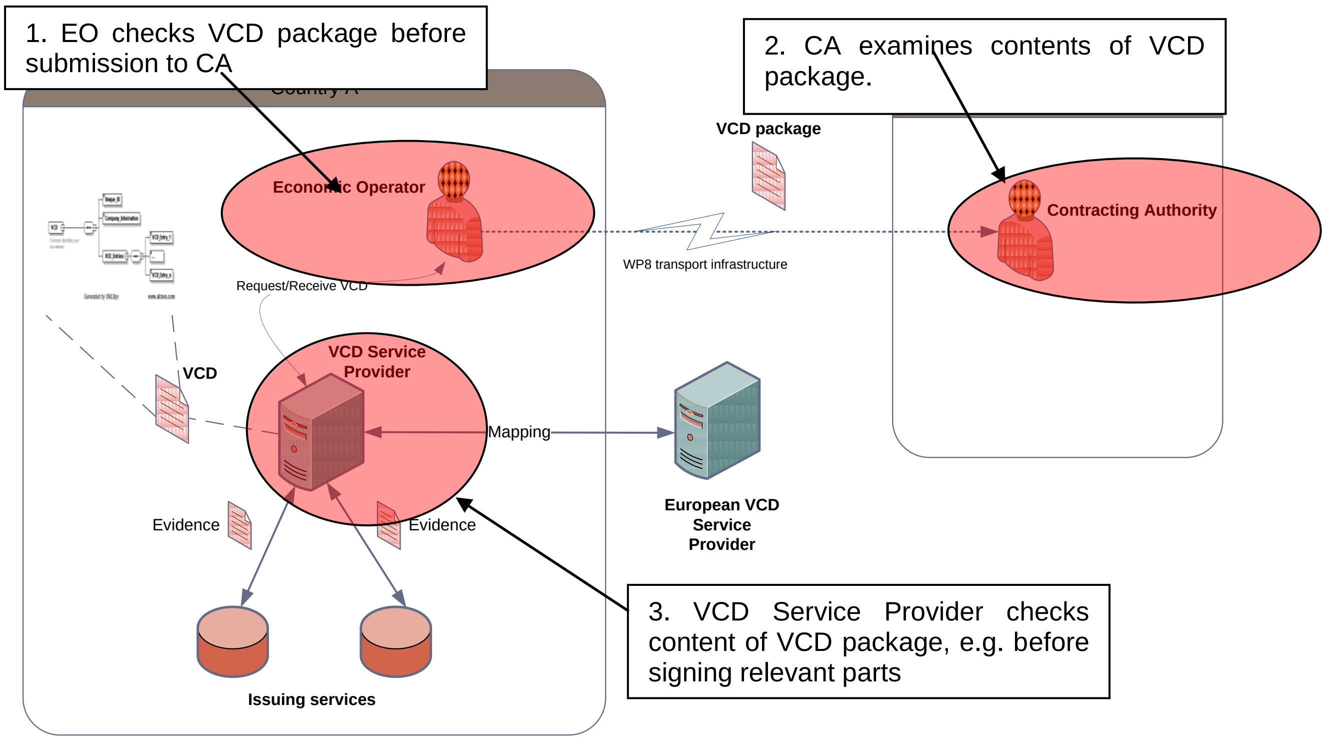
1. EO checks VCD package before submission to CA
Country A
Economic Operator
Request/Receive VCD
VCD
VCD Service Provider
Mapping
Evidence
Evidence
Issuing services
VCD package
WP8 transport infrastructure
2. CA examines contents of VCD package.
Contracting Authority
European VCD Service Provider
3. VCD Service Provider checks content of VCD package, e.g. before signing relevant parts

Figure 1-51: VCD Viewer for VCD package

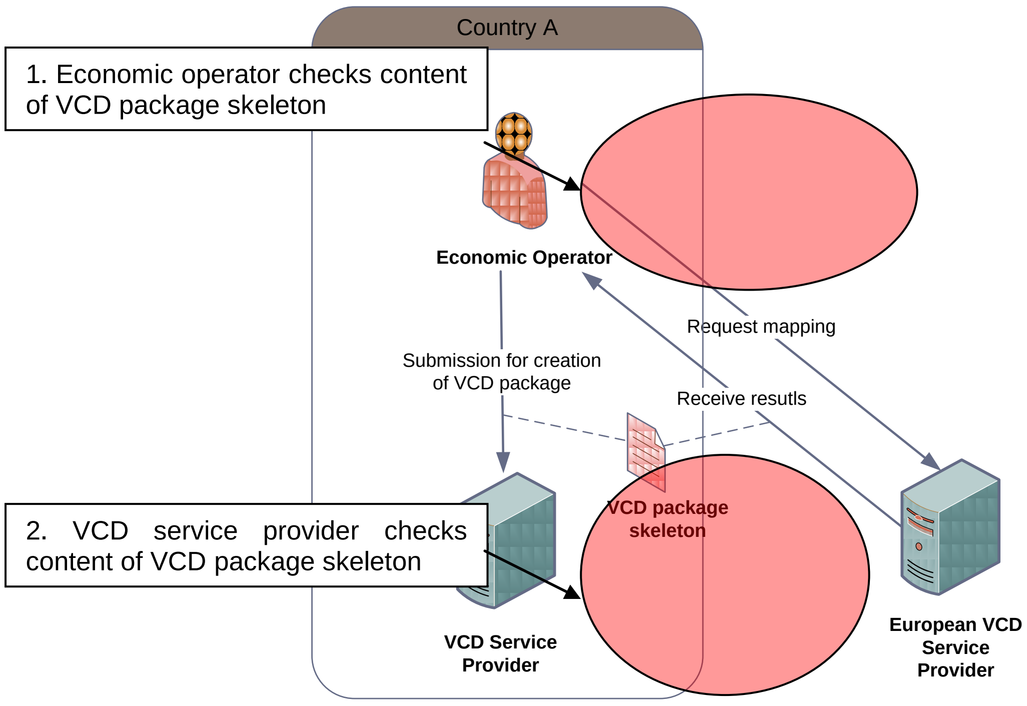
Country A
1. Economic operator checks content of VCD package skeleton
Economic Operator
Request mapping
Submission for creation of VCD package
Receive resutls
VCD package skeleton
2. VCD service provider checks content of VCD package skeleton
VCD Service Provider
European VCD Service Provider

Figure 1-52: VCD Viewer for a VCD package skeleton

Different possibilities exist for the technical realization of the VCD Viewer. Three of these are introduced in the introduced in the following sections, consisting in the embedding of XSL stylesheets in the VCD package, the package, the development of a desktop application and the development of a web application (see Figure 1-53).

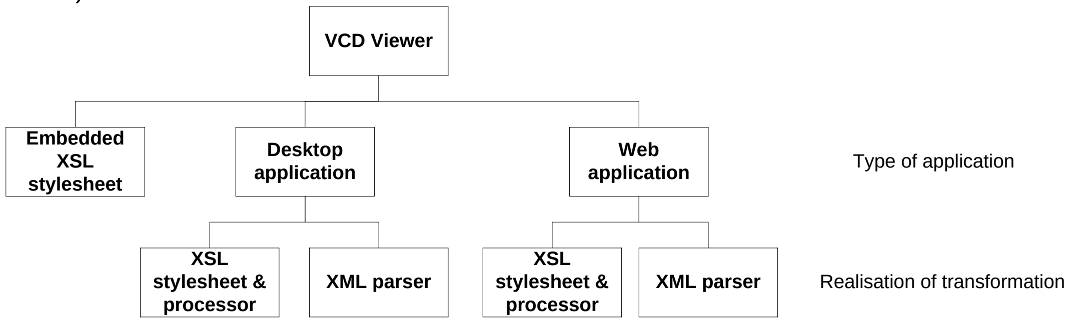

Figure 1-53: Possible implementations of the VCD Viewer and the XML transformation capabilities

The last two alternatives can perform the transformation of the XML files included in a VCD package either with the help of XSL stylesheets (e.g. stored as part of the application) and appropriate XSL processors or with XML parsers. The component diagram in Figure 1-55 depicts the main components of a desktop and a web application respectively.

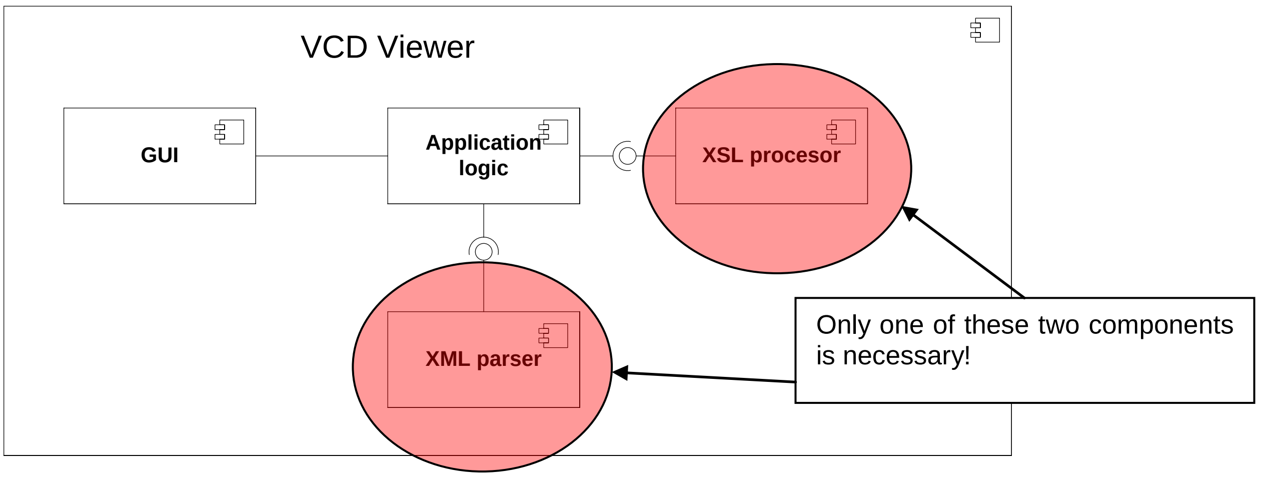

Figure 1-54: Components of the VCD Viewer (implementation alternatives 2 and 3)

A short comparison with regard to advantages and disadvantages of the three development alternatives is presented in the next chapter.

1.4 Physical Implementation Considerations

The descriptions presented below do partly refer to the physical structure of the VCD package which is assumed to be as described in the conceptual class diagram (see Figure 1-55). <PM>there MUST be a separated description affecting the VCD Package Skeleton as the two missions are different along with the practical implications.</PM>

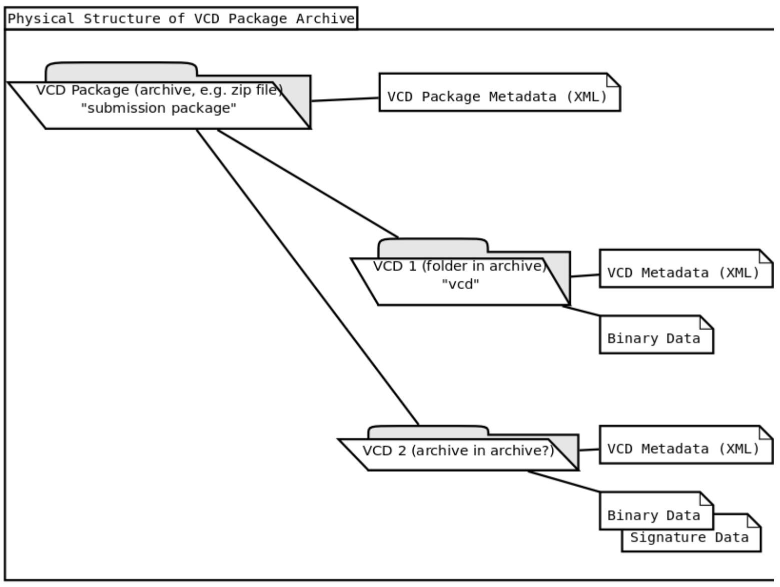
Physical Structure of VCD Package Archive
VCD Package (archive, e.g. zip file) "submission package"
VCD Package Metadata (XML)
VCD 1 (folder in archive) "vcd"
VCD Metadata (XML)
Binary Data
VCD 2 (archive in archive?)
VCD Metadata (XML)
Binary Data
Signature Data

Figure 1-55: Physical structure of the VCD package

1.4.1 Function “Display VCD package” – Alternative 1: Embedded XSLT

The first possible realization consists in the embedding of XSLT stylesheets in the VCD package. The user who wants to visualize such a package can use these stlyesheets and the respective XML files as input of a XSLT processor to generate a desired output, e.g. in the form of HTML or PDF files. The generated output representing the VCD package can then be displayed with appropriate applications, e.g. web browser or PDF viewer. In this implementation variant, the VCD Viewer is does not denote a software application in the classical sense but it just refers to the XSLT stlyesheets included in the VCD package.

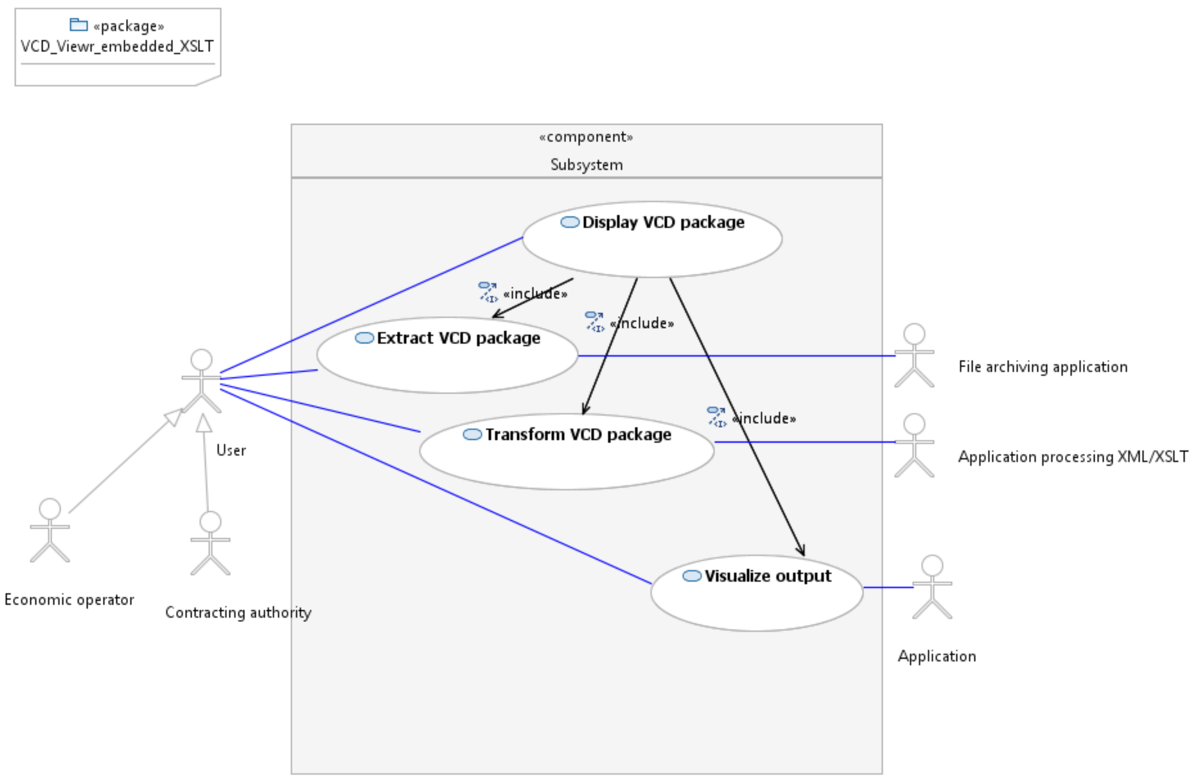

Figure 1-56: Functionalities of the VCD Viewer – Alternative 1

Aspect	Description
Objective	Access the VCD Viewer in order to get a human-readable representation of the VCD package. <PM>Get a paired set of files that make the viewing possible</PM>
Results (postconditions in case of success)	Human-readable and understandable representation of the VCD package and its contents (documents like evidences etc.). <PM>The VCD package content has different “layers” the Peppol managed part and the EO supplied documents</PM>
Precondition	The user has access to the VCD package to be displayed. The user has installed software on his computer which is capable of processing XML and XSLT stylesheets, e.g. a usual web browser.
Postcondition in case of failure	The user gets informed about the reasons for failure (e.g. missing information).
Actor(s)	Economic operator Contracting authority Auxiliary Entity
Initiating event	The user requests the displaying of the VCD package by opening the root XML element of the VCD package.
Description of interaction procedure with European VCD Service	The user extracts the VCD package to a local storage medium, e.g. local hard disk. The user transforms the VCD package with an application capable of processing XML files and assigned XSLT stylesheets (e.g. web browser) into the desired output format, e.g. HTML or PDF. The entry point for the transformation could be the XML file residing on the top-level of the VCD package. The user displays the generated output with an appropriate application (web browser, PDF Viewer). This application can –but need not necessarily - be the one that has created the output, e.g. web browser.
Extension(s)	

Table 1-26: Use case description

Figure 1-57 depicts the scenario described above.

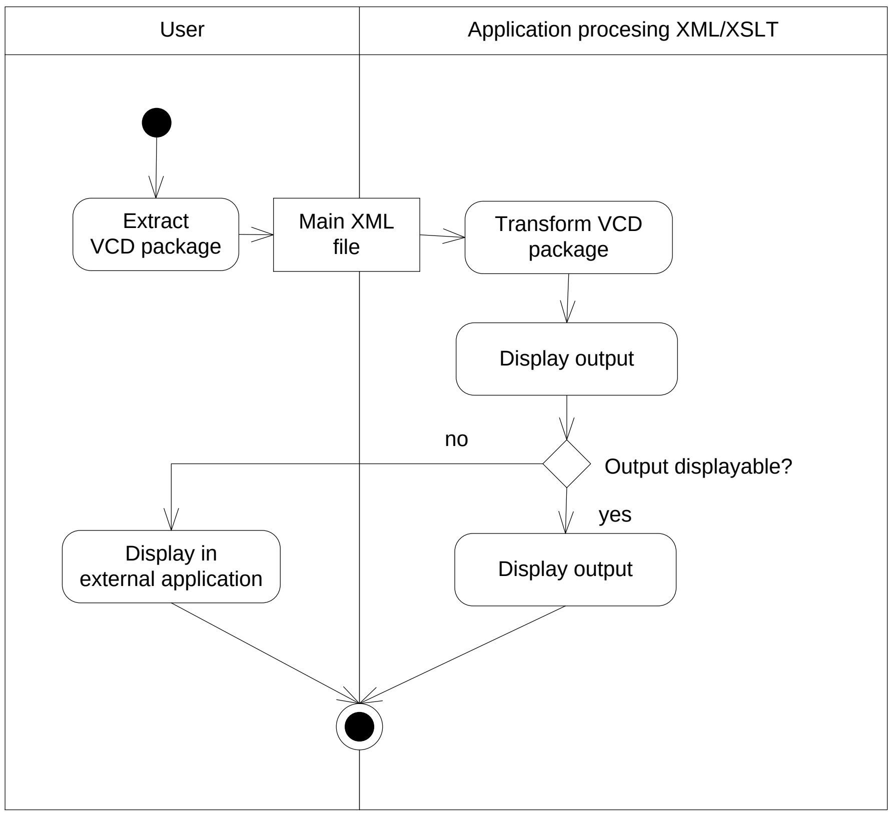

Figure 1-57: Activity diagram for the use case "Display VCD package"

1.4.2 Overall description for the function ”Display VCD package” – Alternative 2: Desktop-Application

The second implementation alternative consists in the development of a desktop application which is executed on the user’s computer. This application performs the necessary transformations and displays the output or makes use of other (installed) software for the visualization.

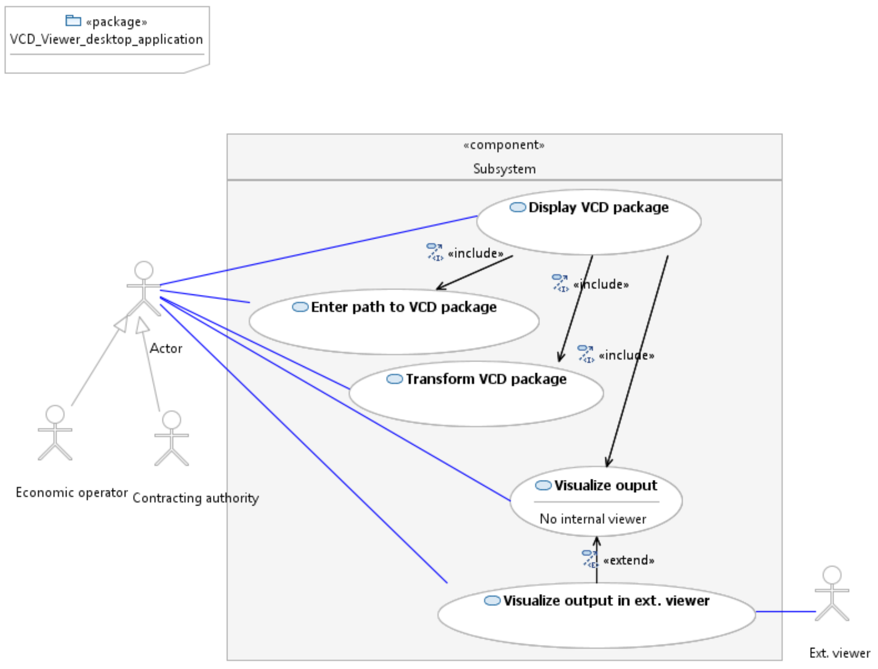

Figure 1-58: Functionalities of the VCD Viewer – Alternative 2

Aspect	Description
Objective	Access the VCD Viewer in order to get a human-readable representation of the VCD package.
Results (postconditions in case of success)	Human-readable and understandable representation of the VCD package and its contents (documents like evidences etc.).
Precondition	The user has access to the VCD package to be displayed. The user has installed the required VCD Viewer software on his computer. <PM>details should be added concerning the governance regime for the desktop software, regardless of getting it from Peppol or other organizations</PM>
Postcondition in case of failure	The user gets informed about the reasons for failure (e.g. missing information).
Actor(s)	Economic operator Contracting authority Auxiliary Entity
Initiating event	The user requests the displaying of the VCD package through the appropriate means provided by the VCD Viewer application.
Description of interaction procedure with European VCD Service	The user opens the VCD Viewer desktop application. The user requests the visualization of a VCD package, e.g. by clicking a button or by selecting a menu entry. The user enters the path to the stored VCD package. The system transforms the VCD package and generates the appropriate output. Two possibilities exist for the presentation of the generated output: The desktop application has a built-in viewer displaying the contents of the VCD package. The desktop application makes use of external programs for the visualization (PDF viewer, web browser).
Extension(s)	

Table 1-27: VCD viewer

The activity diagram in
Figure 1-59 depicts the scenario described above.

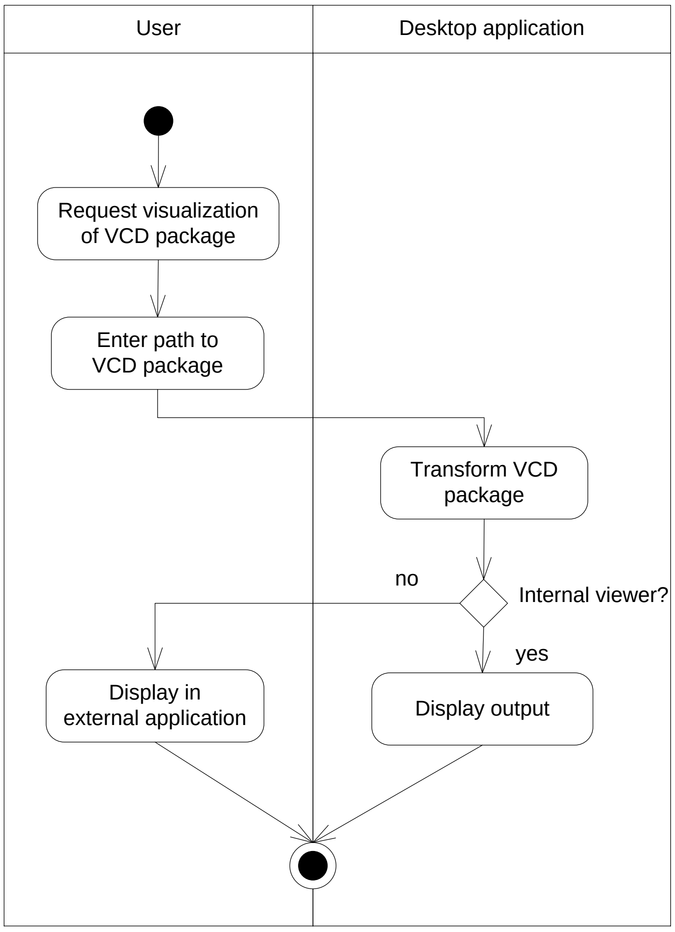
User
Desktop application
Request visualization of VCD package
Enter path to VCD package
Transform VCD package
no
Internal viewer?
yes
Display in external application
Display output

Figure 1-59: Activity diagram for the use case "Display VCD package"

Alternative 2a: Desktop application using an XML parser for the transformation

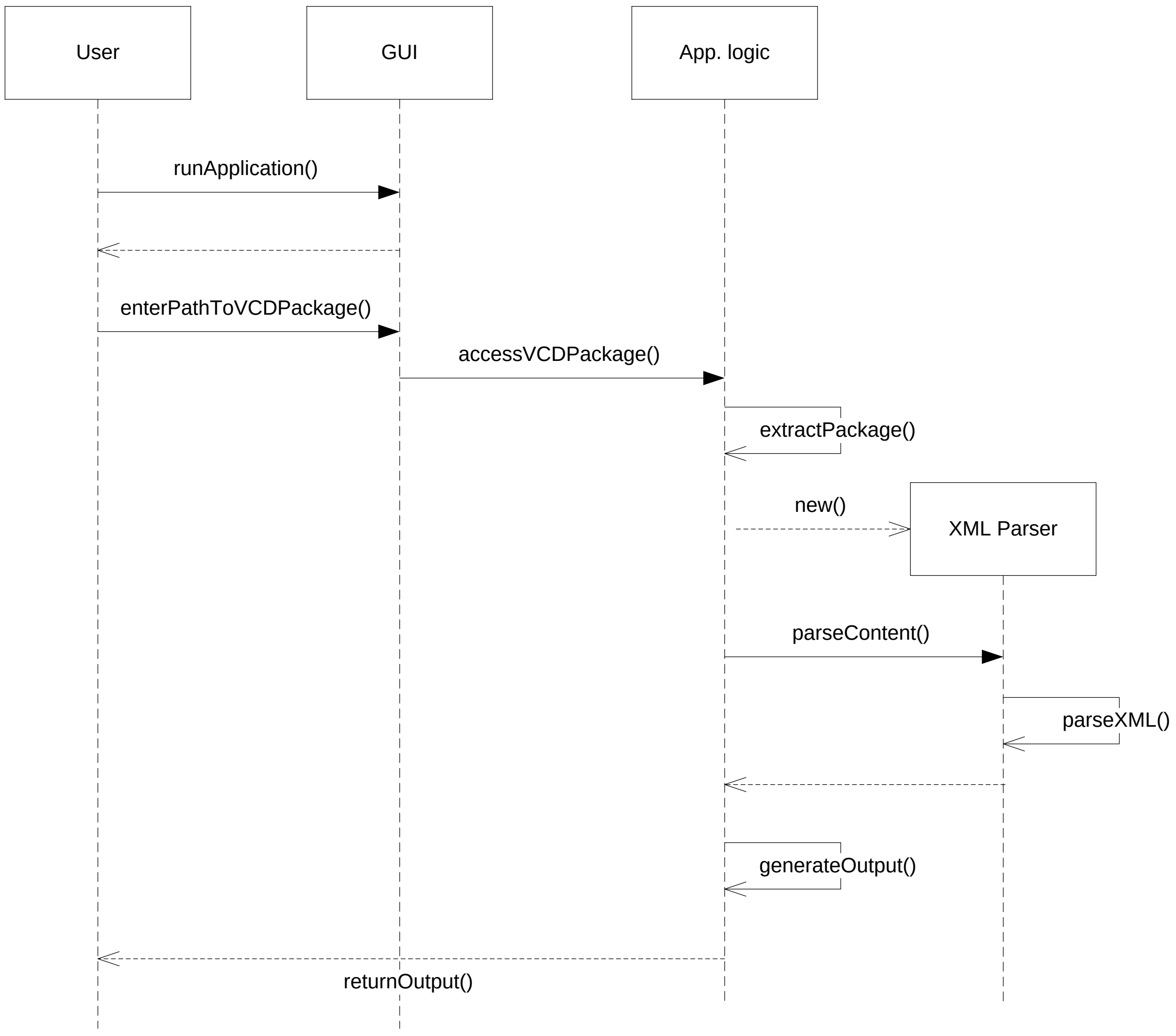

Figure 1-60: Desktop application transforming with XML parser

Alternative 2b: Desktop application using an XSL stylesheet and XSL processor for the transformation

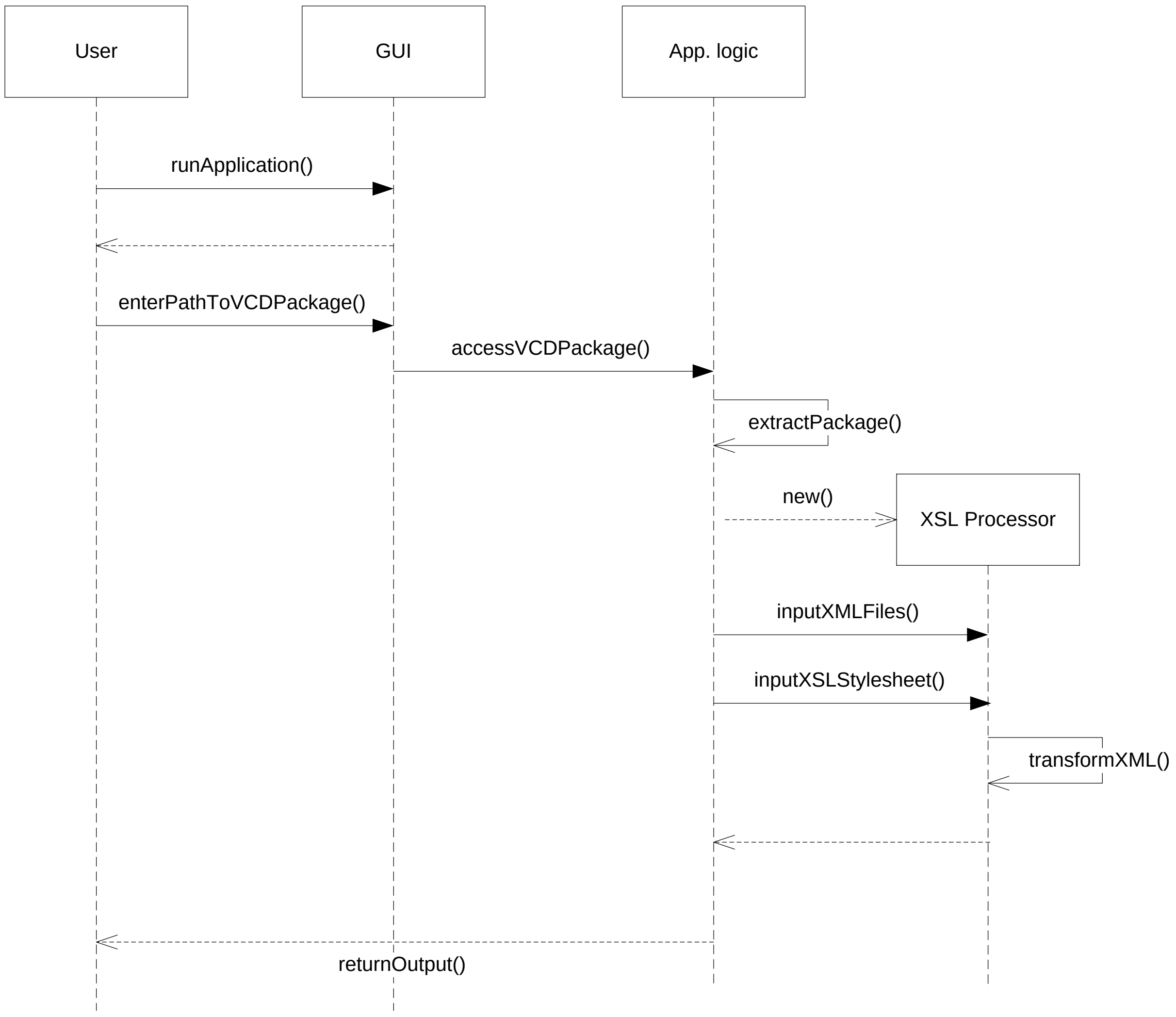

Figure 1-61: Desktop application working with XSL stylesheet/processor

1.4.3 Function "Display VCD package" – Alternative 3: Web-Application

The third possible implementation of the VCD Viewer consists of a web- based application. The user accesses this service through a web page which provides a possibility for uploading the VCD package.

The VCD package is transformed by the web-application and the generated output is returned to the user whose web browser either displays the transformed VCD package or, in case it cannot process the output format, asks the user to open the appropriate application (e.g. output is PDF but browser does not possess the appropriate plug-in).

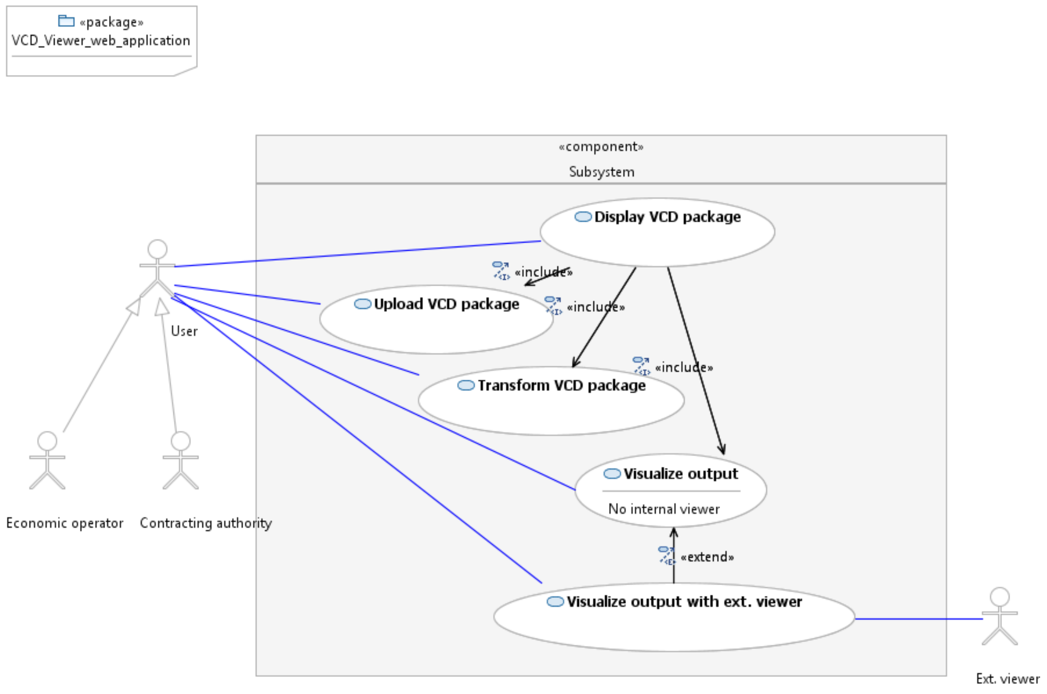

Figure 1-62: Functionalities of the European VCD mapping system (mapping component)

Aspect	Description
Objective	Access the VCD Viewer in order to get a human-readable representation of the VCD package.
Results (postconditions in case of success)	Human-readable and understandable representation of the VCD package and its contents (documents like evidences etc.).
Precondition	The user has access to the VCD package to be displayed. The user has installed the required VCD Viewer software on his computer.
Postcondition in case of failure	The user gets informed about the reasons for failure (e.g. missing information).
Actor(s)	Economic operator Contracting authority Auxiliary Entity
Initiating event	The user requests the displaying of the VCD package, e.g. by clicking a button on the web site.
Description of interaction procedure with VCD Viewer	The user navigates to the VCD Viewer web application, e.g. by using a web browser. The user requests the visualization of a VCD package, e.g. by clicking a button. The user uploads the VCD package using the means provided by the VCD Viewer web application. The system processes/transforms the VCD package and generates the appropriate output, e.g. in the form of HTML or PDF files. Two possibilities exist for the presentation of the generated output: The web application generated output in a format displayable in the user's web browser (HTML, PDF with appropriate browser plug-in etc.). The web application generated output in a different format which needs to be presented using an external application. In this case, the user would have to download the transformed VCD package for later display.
Extension(s)	

*

Table 1-28: Use case description

The activity diagram in Figure 1-63 depicts the scenario described above.

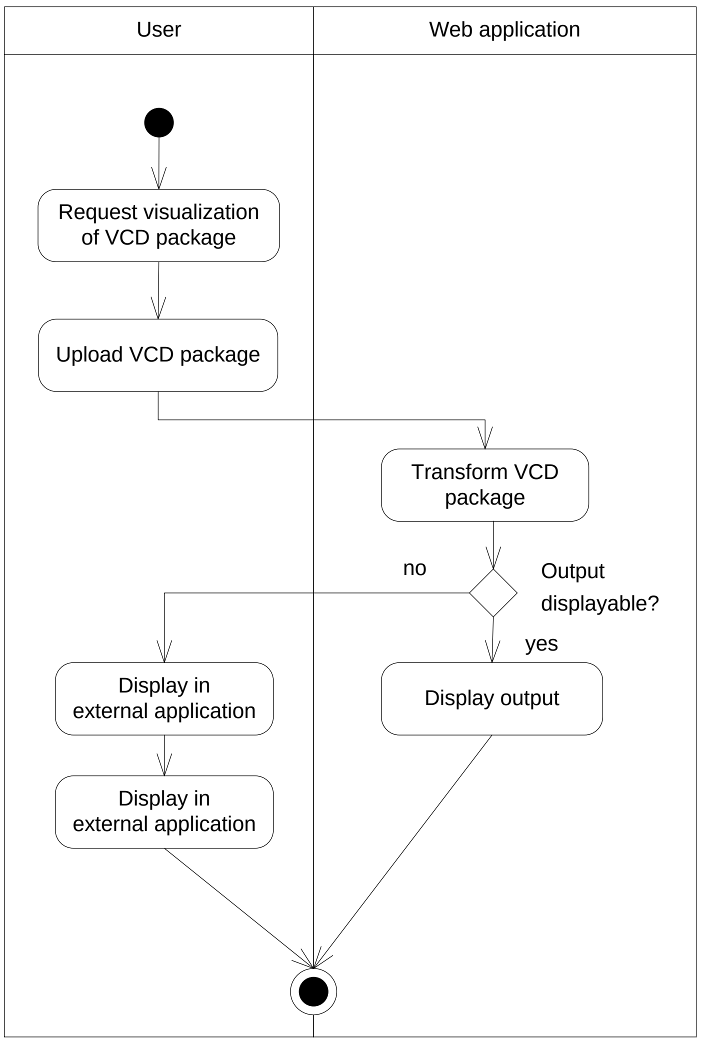
User
Web application
Request visualization of VCD package
Upload VCD package
Transform VCD package
no
Output displayable?
yes
Display in external application
Display output
Display in external application

Figure 1-63: Activity diagram for the use case "Display VCD package"

Alternative 3a: Web application using an XML parser for the transformation

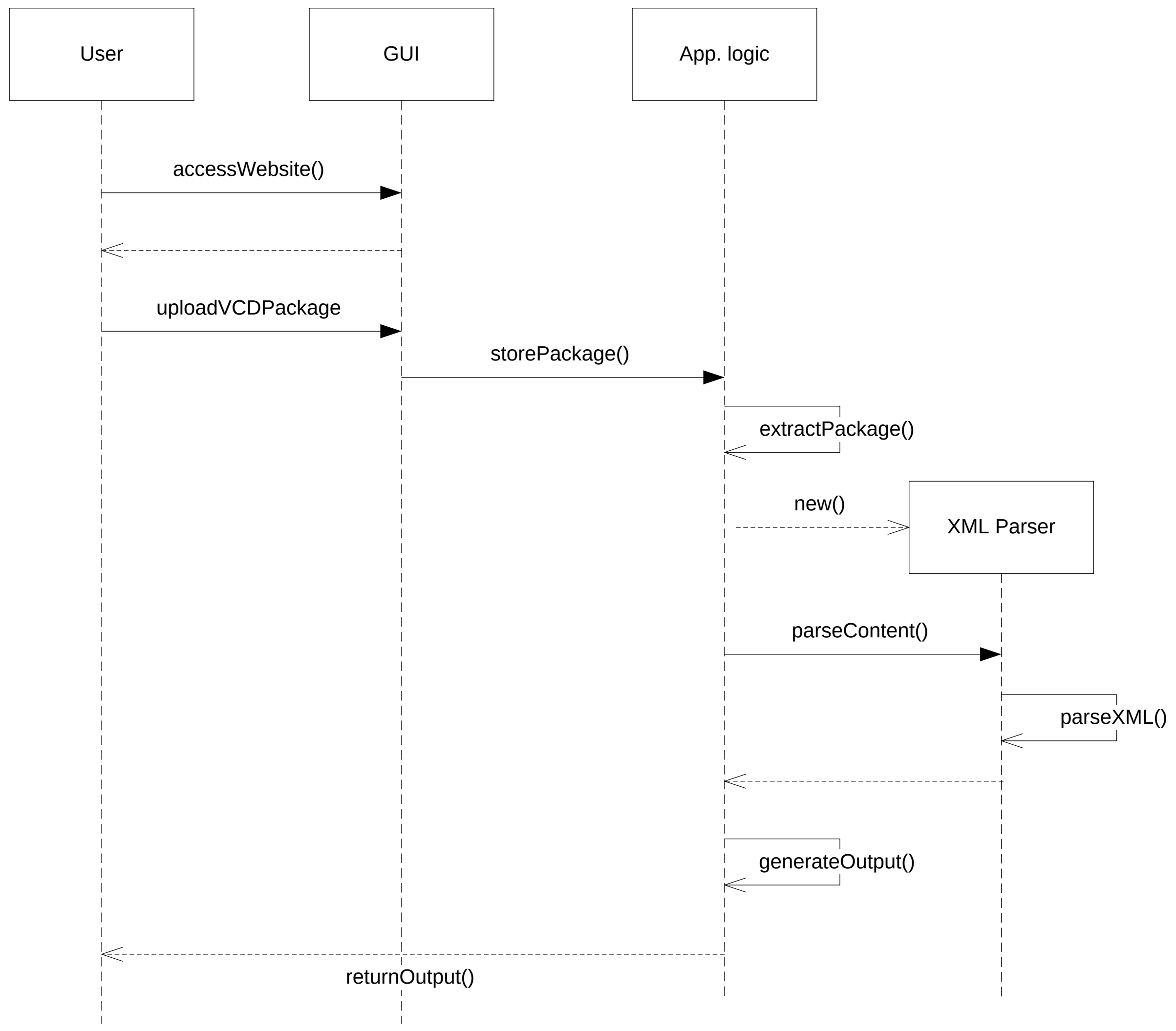

Figure 1-64: Web application transforming with XML parser

Alternative 3b: Web application using an XSL stylesheet and XSL processor for the transformation

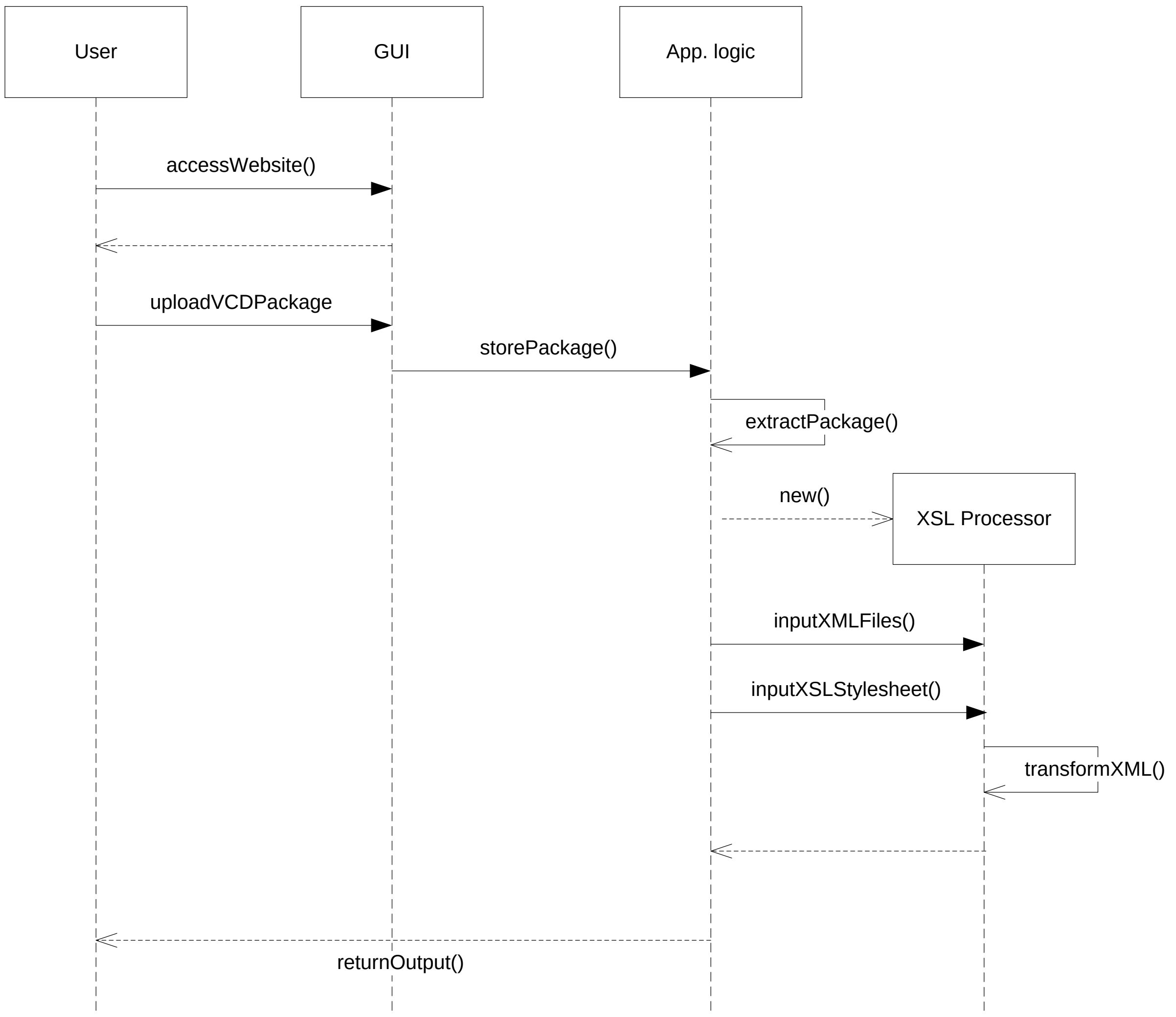

Figure 1-65: Web application working with XSL stylesheet/processor

Functions supporting user interaction

1.4.4 Understandable and easily navigable representation of the output

The representation of the transformed VCD package must be easily understandable and navigable.

1.4.5 Uploading of VCD package (Web application)

The VCD Viewer must provide a function allowing a user to upload a VCD package for further processing.

1.4.6 Input of path to VCD package (Desktop application)

The VCD Viewer must provide a function allowing a user to indicate the path to a (locally) stored VCD package for further processing.

1.4.7 Unpacking of VCD package

The VCD Viewer must have a function to unpack the contents of a VCD package in case the latter is packed, using e.g. ZIP compression. This functionality is considered necessary in order for the system (the Viewer) to be able to access all relevant files (XML, binary content).

1.4.8 Transformation of VCD package content/Generation of output

The VCD Viewer must provide functionality for the transformation of a VCD package respectively its contents into a human-readable and understandable (graphical) representation.

The content of a VCD package does mainly consist of XML and binary files as depicted in Figure 1-55. At least two approaches can be identified for the transformation of these XML files into a human-readable representation, using either XSL stylesheets or XML parsers (DOM/SAX).

In the first case, an XSL stylesheet needs to be created and referenced by the XML files to be transformed. An XML processor, which is for example included in today’s web browsers, can then transform the XML file based on the rules described in the respective XSL stylesheet. Different output formats can be created with the help of appropriate stylesheets as depicted in Figure 1-66.
An alternative approach consists in the parsing of the XML files with the help of appropriate XML parsers and the generation of appropriate output code in a desired format such as HTML.
The first implementation alternative described in section 1.4.1 is solely based on the user of XSL stylesheets and appropriate XSL processors.

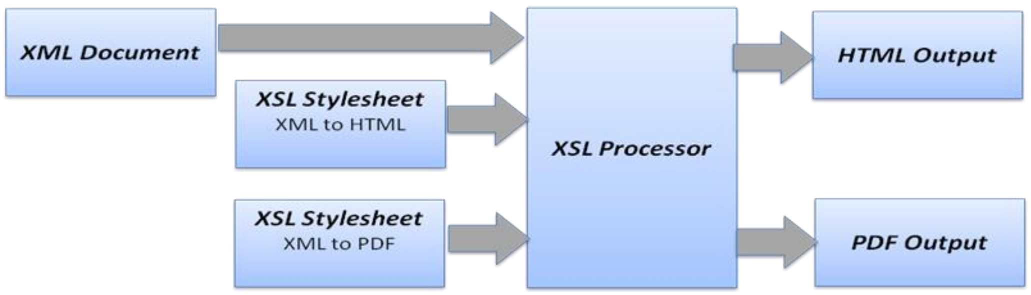

Figure 1-66: Transformation of XML using XSL

Alternatives 2 and 3, described in sections 1.4.2 and 1.4.3, could be implemented with the help of any of the two approaches.

1.5 Non-functional specifications

Regardless of the alternative actually chosen for the implementation of the VCD Viewer, the main development task consists in the creation of an appealing representation of a VCD package’s contents, taking into consideration aspects like usability and accessibility. General non-functional specifications relevant for the development of user interfaces can be found in the general specifications document.

1.5.1 Comparison of approaches and final decisions

Table 1-29 depicts some of the advantages and disadvantages of the three implementation alternatives introduced before.

	Advantages	Disadvantages
XSLT file included in VCD package	Displaying facility already included in and delivered with the VCD package Can be processed with standard web browsers, no extra software needed	Technical realization has to be examined: In case the VCD package is delivered as container (ZIP, RAR or similar), the latter would have to be extracted first. User can choose the output format but an extra style sheet has to be delivered/provided for every output format (e.g. PDF, HTML etc.)
Desktop-application	No internet connection necessary No upload of VCD package necessary Different output formats can be offered to the user	Installation and maintenance of application necessary
Web application	No installation of software necessary Maintenance of the application in a central place Different output formats can be offered to the user	Upload of VCD package to Web-application necessary Web-application has to be always online

Table 1-29: Comparison of alternative implementation solutions for the VCD viewer

Based on a careful assessment of these advantages and disadvantages it was decided to implement a desktop application. Main arguments for this decision were that with this solution, users do not need an active connection to the Internet for the visualization of a VCD package (skeleton). In addition, the same solution can be offered as a web application at the peppol.eu site for those users who do not want to download and install a separate application. The stylesheet, which is necessary as basis for the transformation, is likely to be included in the VCD package (skeleton). Alternative 1 (see section 1.4.1) assumes that the transformation of such a VCD package (skeleton) can be performed by any application capable of processing XML and XSLT as for example common web browsers. However, it is supposed that the rather complex structure of the VCD package (skeleton) necessitates the use of additional application logic as presented in alternatives 2 and 3.

Some questions remain with regard to the implementation of the desktop application. The first one concerns the user interface.

The choice for a desktop application, as presented in the previous paragraph, offers two alternatives with regard to the graphical user interface used for the visualization of the VCD (skeleton) package:

- Implementation of a standard desktop user interface. This user interface would be a hard-coded one, consisting of traditional navigation and displaying elements. These would be filled with the contents of a VCD (skeleton) package by the desktop application. For that purpose, the use of an XML-parser and appropriate application logic for the processing of the VCD package and the displaying of its contents is considered necessary. XSLT stylesheets cannot be used in this scenario.

- Generation of a HTML user interface displayed in a locally installed browser.: This user interface would consist of HTML elements, created and filled with the contents of a VCD package through the processing of an XSLT stylesheet. Consequently, a web browser, installed locally on the user's desktop, is needed to display the interface and its contents. It is to be noted that not all of the content included in a VCD package can be displayed in a web browser. While most image formats such as jpg, png or tiff can be included in HTML code and thus displayed by a web browser, other file formats such as PDF necessitate the installation of particular plug-ins . A user who has not installed these plug-ins will therefore not be able to view these files. It is proposed to create links to content which cannot be included in HTML code. By clicking on such a link, the appropriate third-party application will be started on the user's computer, provided such an application is installed and appropriately configured. The configuration of these third-partly application is up to the user and therefore not in the scope of PEPPOL WP2.

Both alternatives do have advantages and disadvantages. The first solution is considered to be easier to implement but reusability is limited. The second alternative necessitates a heavier implementation effort but provides the possibility of reusing the created XSLT stylesheet for the development of an additional web application. As the web browser solution (second solution) provides an environment, with which users are familiar, and because of the possibility of reusing the XSLT stylesheet also in the online environment, this alternative was decided.
Another issue to be dealt with concerns the choice for a programming language. A variety of languages can be used for the creation of desktop applications, e.g. Java, C++, C#, Visual Basic etc. As the VCD Viewer should be able to run on as many currently existing operating systems as possible, platform independency is a major requirement. Furthermore, the language of choice needs to provide all of the needed libraries for processing the VCD (skeleton) package, e.g. libraries for parsing and transforming XML. A good support with regard to IDEs and other development tools should be available for the chosen language. Table 1-30 compares Java and C# along a number of implementation aspects to be considered (rating scale is +: good support; 0: medium support; -: no support). The assessment was done based on experiences and expertise of project members.

Criterion/Language	**Java**	**C# (or any other .NET based language)**
Tool support	+	+
Support for (Zip-) containers and XML processing	+	+
Platform support (Windows, Linux, Macintosh)	+	0[3]

Table 1-30: Brief comparison of programming languages for the implementation of the VCD Viewer

It is proposed to use Java as it fulfils all of the aforementioned criteria, especially with regard to platform independency.

Figure 1-67 summarizes the activities during the visualisation of a VCD package (skeleton).

[3] Although the .NET framework does officially only exist for the Windows platform, the open-source implementation "Mono", developed as a community project, permits to run .NET application also on Linux and MacOS.

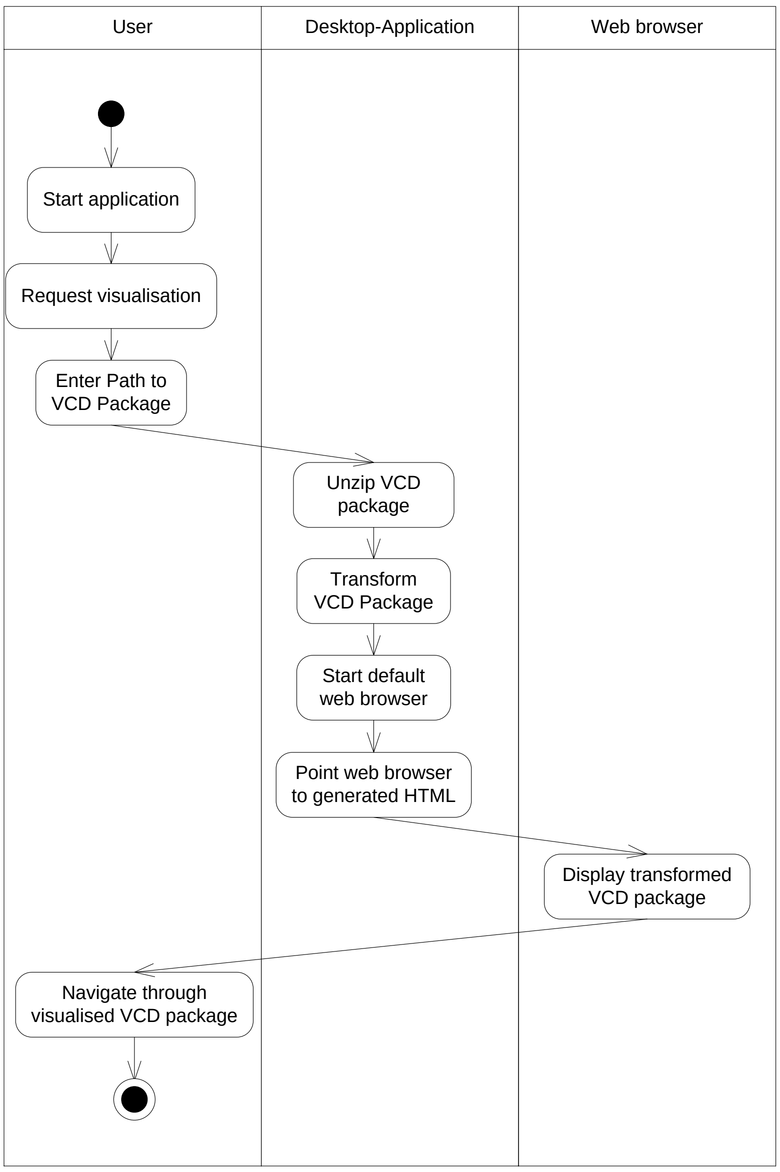
User
Desktop-Application
Web browser
Start application
Request visualisation
Enter Path to VCD Package
Unzip VCD package
Transform VCD Package
Start default web browser
Point web browser to generated HTML
Display transformed VCD package
Navigate through visualised VCD package

Figure 1-67: Visualisation of a VCD package using a desktop application

1.5.2 Data handling to be implemented by a VCD service provider

Every VCD Service Provider should devise and implement its own policy for data handling according to provisions for data protection and user privacy guarantee.
This will be left to the VCD service providers that will be connected to PEPPOL, however some principles should be common to all:

- Data minimization: Only store information that is absolutely necessary for the provision of the contracted services
- Proportionality of use: The data handled and stored as well as the policy implemented should be proportional to the objective of the service – i.e.security measures should be designed according to the degree of sensitivity of particular data.
- Informed consent: Users should be made aware of the data handling and storage policies and should have the chance to opt into any provisions after understanding the procedures implemented by the VCD service provider and their consequences for their privacy.
- Alignment with PEPPOL trust models

Such principles should enable subscribers of VCD service providers to negotiate and build trust relationships necessary for moving all transactions online.

Index of Figures

Index of Tables

Abbreviations

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