D02.03 Reusability Factsheet Template

SC73 Assessment of Trans-European Solutions supporting EU policies

Under Framework Contract: DI/07172 - ABCIII

17 August 2015

This study was carried out for the ISA Programme by:

Authors: Debora DI GIACOMO, Constantinos SIMATOS,

part of KonSulT consortium

Internal Identification:

Specific Contract No: 73

Framework Contract: DI/07172 - ABCIII

Disclaimer

Table of Contents

List of Tables

Revision History

The following table shows the development of this document.

Date	Version	Description	Author(s)	Reviewed by
05/06/2015	V0.01	Development of fist version of the document and table of Contents	Constantinos Simatos	Debora Di Giacomo
16/06/2015	V0.02	Draft document presented to the client	Constantinos Simatos	Raul Abril
30/06/2015	V0.03	Implementation of comments from the Project officer and further development of the deliverables	Constantinos Simatos	Debora Di Giacomo
02/07/2015	V0.03 V0.04	Implementation of comments from the Project officer and further development of the deliverables	Debora Di Giacomo	Constantinos Simatos
07/07/2015	V0.05 V0.06	Internal review and updates	Debora Di Giacomo	Constantinos Simatos
08/07/2015	V1.00	Delivered for clients acceptance	Debora Di Giacomo Constantinos Simatos	Raul Abril Szabolcs Szekacs
17/07/2015	V1.01 V1.02	Implementing comments received from the client. Creation of Annex A and Annex B.	Debora Di Giacomo Constantinos Simatos	Debora Di Giacomo Constantinos Simatos
20/07/2015	V2.00	Version sent to Sandro D'Orazio (DIGIT) for validation of the data of the reusability factsheets for e-TrustEx and CIPA eDelivery	Debora Di Giacomo Constantinos Simatos	Sandro D'Orazio
17/08/2015	V2.01	Version including the comments and input by Sandro D'Orazio on version V2.00.	Constantinos Simatos	Debora Di Giacomo
17/08/2015	V3.00	Final version delivered for acceptance	Debora Di Giacomo Constantinos Simatos	Raul Abril
21/08/2015	V4.00	Final version implementing comments from Raul Abril. Version accepted by the client.	Debora Di Giacomo Constantinos Simatos	Raul Abril Margarida Abecasis

Introduction

This report falls under the scope of Specific Contract 73 under ABC III Framework Contract. This contract is performed in the context of Action 2.14 - Assessment of Trans-European solutions (TES) [1] supporting EU policies of the ISA Programme. This action supports the European Commission in implementing an overall strategy to rationalise and streamline the IT systems it develops, maintains and operates.

ISA Action 2.14 has also developed the EU Cartography; an inventory of Interoperable European Solutions (IES) described according to the European Interoperability Reference Architecture (EIRA)[2]. The EU Cartography is available via the Cartography Tool (CarTool)[3], a tool that can be used to describe and catalogue the Solution Building Blocks (SBBs) of enterprise architectures and visualise these SBBs according to the Architecture Building Blocks (ABBs) of the EIRA.

The purpose of this report is to propose a template for a factsheet that would facilitate the assessment of solutions reuse by providing useful and detailed information that should be considered when evaluating reusing a solution in a specific context. This factsheet should be considered as complementary to the solution's documentation available on Joinup's catalogue of interoperability solutions as well as on the EU Cartography. It is important to note that this factsheet focuses on reusability of technical solutions both as a software component and as a service. Future work envisages the extension of the scope towards semantic, organisational and legal solutions.

This report is comprised of the following parts:

- Prerequisites that should be taken into account by solution owners, when filling in the template of the reusability factsheet, and by solution architects or portfolio managers when reading the reusability factsheet (section 1);
- Identified use cases of the proposed reusability factsheet (section 2);
- The most important aspects that should be considered when evaluating reusing a solution included in the reusability factsheet (section 3);
- A guide on how to use the information contained in the reusability factsheet for evaluating the overall feasibility and impact of reusing a documented solution (section 4);
- A mock-up of the reusability factsheet (section 5); and
- Two examples of the reusability factsheet applied to operational solutions (Annex A Reusability factsheet for e-TrustEx, Annex B Reusability factsheet for CIPA e-Delivery).

[1] Trans-European Solutions are solutions developed by the European Commission or other bodies (in some cases co-funded by MSs), that facilitate cross-border exchange of information and delivery of electronic public services between Public Administrations in support to the implementation and advancement of EU policies.

[2] European Interoperability Reference Architecture (EIRA) is the reference architecture for digital public services. It defines the required capabilities to enhance interoperability as a set of Architecture Building Blocks (ABB). It is aligned with the European Interoperability Framework and is divided in four views: legal, organisational, semantic and technical. The EIRA is SOA (Service-Oriented Architecture) based, uses the Archimate modeling language and has a strong focus on interoperability in public sectors. (Source: https://joinup.ec.europa.eu/asset/eia/description)

[3] It is envisaged that the data currently available via the CarTool will be migrated to Joinup and serve as basis for the European Interoperability Cartography (EIC).

1 Prerequisites for using the reusability factsheet

The following subsections highlight the aspects that should be taken into account by system owners when filling in the reusability factsheet and by solution architects or portfolio managers when using it to evaluate the feasibility of reusing solutions.

1.1 Requirements filling in the reusability factsheet

Filling in the reusability factsheet requires **deep understanding of the solution considered reusable.** Therefore, it is recommended that the system owner or a person designated by him/her take the responsibility for filling it in.

The reusability factsheet is aligned with and refers to the EIRA, hence the system owner should have a **good understanding of the EIRA** in order to provide high quality and complete information.

The reusability factsheet will be linked to the European Interoperability Cartography (EIC)[4], a repository of interoperability solutions identified to be reusable and interoperable in the context of the implementing a public policy. Consequently, it is advocated that each **solution documented in the EIC must have a reusability factsheet**.

Finally, the reusability factsheet will be subject to data quality assurance which aims at ensuring that all factsheets provide consistent, complete and accurate information. Therefore it is strongly recommended that system owners **complete all the information requested** including the domain of applicability, general reusability aspects and specific reusability aspects.

1.2 Considerations using the reusability factsheet

The reusability factsheet aims at providing useful and detailed information that should be considered when evaluating a solution's reuse. Solution architects and portfolio managers should take into consideration the following points when consulting its contents:

- **Knowledge on the EIRA:** the reusability factsheet aims at being self-explanatory by providing definitions and examples; however, taking into account the references made to the EIRA, prior knowledge on this reference architecture would be beneficial;
- **Consider further information available via the European Interoperability Cartography (EIC)[4]:** The EIC is a repository of interoperability solutions identified to be reusable and interoperable in the context of implementing a public policy. The information available in the EIC is aligned and structured according to the EIRA building blocks. The factsheet reader may consider consulting the EIC for further details.
- **Good understanding on the solution that would benefit from reuse:** Having a good understanding of the solution's characteristics where reuse is envisioned enables a better

[4] More available at https://joinup.ec.europa.eu/catalogue/repository/european-interoperability-cartography-eic

interpretation of the information available. The reason for this is that the weight of different aspects can vary significantly based on the context for which reuse is being investigated.

- **Consider Interoperability aspects beyond reuse**: The European Interoperability Framework (EIF) principles[5] are key when considering the interoperability of a solution, with reusability itself actually representing one such principle. Note that in the reusability aspects that follow, certain interoperability principles are addressed where they could represent factors for consideration depending on the reuse scenario at hand. The aspects that refer to interoperability principles are highlighted as such in the factsheet.

[5] The EIF principles are: Subsidiarity and proportionality, User-centricity, Inclusion and accessibility, Security and privacy, Multilingualism, Administrative simplification, Transparency, Preservation of information, Openness, Reusability, Technological neutrality and adaptability, and Effectiveness and efficiency. The EIF is available online at: http://ec.europa.eu/isa/documents/isa_annex_ii_eif_en.pdf

2 Reusability factsheet use cases

This section details the identified reusability factsheet use cases:

- Document solution reusability aspects (section 2.1);
- Assess solution reuse (section 2.2);
- Evaluate solution reusability aspects for portfolio rationalisation (section 2.3); and
- Analyse a solution's Total Cost of Reuse (section 2.4).

2.1 Use Case: Document solution reusability aspects

Table 1: details the 'document solution reusability aspects' use case of the reusability factsheet.

Table 1: Use Case - Document solution reusability aspects

Use Case Name	Document solution reusability aspects	
Motivation and Context	An organisation wants to promote reuse of one or more solutions of its portfolio considered reusable. A user documents all relevant reusability aspects of each solution using the reusability factsheet template and makes it available online via the EIC[6].	
Outcome	One or more reusability factsheets are completed and made available via the EIC.	
Involved roles	System Owner	
Comments	This use case is relevant when a reusable solution is identified.	
Description	**Step**	**Action**
Basic flow	1	The user access the reusability factsheet template to identify the reusability aspects that need to be documented.
	2	The user fills in all fields of the reusability factsheet template with information about the solution being documented.
	3	The user submits the filled in reusability factsheet template for data quality review and publication.
	4	The data provided is reviewed for its quality and completeness.
Alternative flow	5a	**If data provided is complete and consistent:** The reusability factsheet is made available online via the EIC.
	5b	**If data provided is not complete or consistent:** The user is requested to update the reusability factsheet.
	6b	The user updates the reusability factsheet template and follows step 3 of this use case.

[6] Technicalities on how the reusability will be made available in the EIC should be decided once the EIC is operational.

2.2 Use Case: Assess solution reuse

Table 2: details the ‘assess solution reuse’ use case of the reusability factsheet.

Table 2: Use Case - Assess solution reuse

Use Case Name	Assess solution reuse	
Motivation and Context	The user is looking for an existing reusable solution which can be used in his/her specific case. The user needs to further assess whether any existing reusable solution would answer his/hers specific needs.	
Outcome	Existing reusable solutions documented in a reusability factsheet are assessed.	
Involved roles	Solution Architect	
Comments	This use case is relevant when a solution is planned to be developed or updated and a user identifies specific needs that can be addressed by reusing existing solutions.	
Description	**Step**	**Action**
Basic flow	1	The user accesses reusability factsheets available online to identify existing reusable solutions in line with his/hers specific needs.
	2	The user verifies the detailed reusability aspects of the solutions documented in the reusability factsheets and evaluate whether the documented solution can be reused in the specific development/improvement case.
	3a	**If a solution fulfils the user’s requirements**, the user can contact the system owner of the reusable solution for further investigation and arrange the reuse details.
	3b	**If no reusable solution fulfils the user’s requirements**, the user would need to either develop a new solution or update the existing solution. The user can start a development project by taking into account the reusability (and other) requirements described in the factsheet.

2.3 Use Case: Evaluate solution reusability aspects for portfolio rationalisation

Table 3: details the ‘evaluate reusing a solution when rationalising a portfolio’ use case of the reusability factsheet.

Table 3: Use Case - Evaluate solution reusability aspects for portfolio rationalisation

Use Case Name	Evaluate solution reusability aspects for portfolio rationalisation
Motivation and Context	The organisation faces the need of rationalising its IT portfolio of solutions by improving efficiency and lowering its Total Cost of Ownership (TCO).
Outcome	Highly reusable solutions are identified and favoured during a rationalisation of a portfolio of IT Solutions.
Involved roles	Portfolio Manager

Comments	The portfolio manager would use the factsheet as a support tool for the rationalisation effort to assess the reusability of the solutions being considered. In this way, if one solution seems less reusable than another it would be less favoured. The use case 'Use Case: Document solution reusability aspects' is a prerequisite of this use case.

Description	Step	Action
Basic flow	1	The user identifies the need of rationalising the existing portfolio of solutions.
	2	The user uses the reusability factsheets as a supporting tool to evaluate to which extent the solutions included in the portfolio are reusable.
	3	Solutions found to be more reusable are favoured in the rationalisation exercise.

2.4 Use Case: Analyse a solution's Total Cost of Reuse

Table 4: details the 'Analyse a solution's Total Cost of Reuse' use case of the reusability factsheet.

Table 4: Use Case - Analyse a solution's Total Cost of Reuse

Use Case Name	Analyse a solution's Total Cost of Reuse	
Motivation and Context	The user wants to understand the overall financial impact of reusing a solution by performing a Total Cost of Ownership (TCO) analysis that in this context can be referred to as the analysis of the Total Cost of Reuse (TCR).	
Outcome	The financial impact (monetary figure) of reusing a solution on the specific context is identified.	
Involved roles	Solution Architect	
Comments	Having executed the use case 'Use Case: Assess solution reuse' is a prerequisite of this use case.	
Description	**Step**	**Action**
Basic flow	1	The user had evaluated reusing a solution and found no blocking constraints (use case 'Use Case: Assess solution reuse').
	2	The user consults the reusability factsheet in order to find guidelines for performing a Total Cost of Reuse analysis.
	3	The user takes into account the solution reusability aspects with potential impact on cost listed in the reusability factsheet in order to assess direct and indirect costs of reusing the identified solution.
	4	The user finalises the Total Cost of Reuse analysis and obtains a monetary figure representing the total cost of reusing a solution. The user can compare this figure with the cost of other alternatives, such as reusing another solution or starting a new development project.

3 Reusability aspects

The following sections provide the information pertinent to reusing the current solution. The first point addressed is the domain of applicability for its reuse that answers the following questions:

- Is the solution meant to be used as a component building block, a centralised service or both?
- How does the solution relate to the EIRA's Architecture Building Blocks?

Once the high level context of reuse is established the document continues with the reusability aspects to be taken into consideration. These are split in three groups that apply depending on how the solution is foreseen to be reused:

- General aspects that apply to all solutions
- Further aspects specific to solutions reused as software components
- Further aspects specific to solutions reused as services

Note that aspects specific to components and services are not mutually exclusive. It could be the case that a solution may be reused as a centralised service, but also installed as a software component to satisfy specific needs.

3.1 Domain of applicability and reuse type

The first key information provided for a solution through the reusability factsheet is its domain of applicability. In simple terms this could be considered as the type of the solution. To best capture this and link to the EIRA, this is represented by:

- The **EIRA view** the solution relates to.
- The specific **architecture building block(s)** (ABBs) addressed by the solution.

Note that the current factsheet is specific to the EIRA's technical view (application and infrastructure). Nonetheless, explicitly stating the related view serves to immediately clarify the focus of the solution in question.

A subsequent step, specific to technical solutions, is the clarification of the solution's intended form of reuse, notably whether the solution can be reused as a software component, as a service, or potentially both. The type of reuse for the solution in question defines the factsheet information that is to be considered applicable:

- **Reuse only as a software component**, the aspects detailed in section 3.2 and 3.3 apply;
- **Reuse only as a service**, the aspects detailed in section 3.2 and 3.4 apply;
- **Reuse as a service and as a software component,** the aspects detailed in section 3.2, 3.3 and 3.4 apply.

The combination of the EIRA reference and the reuse type can be viewed as the initial general information required to position and assess a solution's reuse for a specific case. To complete this introductory information, it is important to note that the factsheet acts as a complement of concise reuse-related

information to the solution's comprehensive documentation in Joinup. As such, the URI link to the **solution's documentation space in Joinup** is also provided.

3.2 General reusability aspects

These are aspects of solutions that have general applicability when considering their reuse. They apply both in the case of reusing the solution as a service and also when the solution is reused as a software component.

Solution aspect	Description
Actual reuse	Cases where the solution has already been reused, either as a service or as a software component. The extent to which a solution is already reused is a good indication of its maturity and reusability, both in technical terms but also potentially in terms of policy domains. *Example: e-TrustEx is reused as the data exchange component in e-Prior in the Procurement domain.*
Planned reuse or extension to other domains	Cases for which the solution is planned to be reused, within the same policy domain, or even extending to different policy domains, including the foreseen reuse date. Coupled with cases where the solution has already been reused, the plans for its reuse or extension to other policy domains offer an indication of the solution's maturity and ability to act as a generic, cross-domain or multi-domain solution building block. *Example: A solution initially developed in the domain of Environment, Consumers and Health that is planned to be reused in 2018 in the domain of Climate Action.*
Design aspects favouring reuse	Measures taken to ensure that this solution has been built as reusable from the ground up. These measures could relate to the choice of non-proprietary technologies, choice of a modular software architecture, or adoption of a SOA approach in exposing reusable services. *Example: Exposure of all business capabilities through a SOAP-based web service API.*
Dependencies and reuse constraints	Dependencies or constraints that this solution entails with regards to its reuse. These dependencies can either be of a technical nature (e.g. reliance on a specific third party product) or relative to specific legislation or business domain. Moreover, the solution may be reusable but only within a specific community or geographic location. *Example: A software component with a strong dependency on Oracle RDBMS 10g as a database provider.*

Solution aspect	Description
Use of standards	The standards that the solution uses and conforms to ranging from business and data standards to technical and communication standards. The standards used by a solution could either make it an ideal reuse candidate for a specific solution or exclude it due to conflicting choices already made. Note that the standards listed as interoperability specifications[7] for the solution's related architecture building blocks (as per the EIRA), are considered as endorsed for being best-in-class with regards to enabling interoperability. *Example: Use of SAML v2.0 for the communication of user authentication, rights and attributes, and AS4 for data exchange of business documents.*
Solution portfolio	Whether or not the solution is part of a portfolio of compatible solutions, either components or services, designed to work alone or together. If so, this is an indication of the solution's modularity and design for reuse, but also opens up possibilities to reuse other solutions from its portfolio. *Example: The CEF (Connecting Europe Facility) eID solution building block for cross-border authentication that is part of a European Commission catalogue of generic building blocks built for reuse (eSignature, eInvoicing, eDelivery, Automated Translation).*
Intellectual Property Rights and licensing[8]	The formal definition that refers to or describes the licensing mechanism, ownership rights, restrictions, and user responsibilities related to the distribution and reuse of the solution. In case the solution is usable both as a service and as a software component with different licensing schemes per case, all cases are documented. *Example: A software component with an MIT licence that can be reused without copyright restrictions.*
Documentation	References to the solution's available documentation. This documentation ranges from background information on the solution such as its history and development team, to technical documentation describing the services exposed, core processes, the architecture, as well as its installation and configuration (if applicable). The availability of rich documentation not only allows for a detailed investigation of the solution but also offers a strong indication of the solution's maturity and support level. *Example: Reference to a solution's wiki documenting all aspects of its design and use.*
Maturity	The status of the solution in terms of maturity, to indicate its ongoing development and whether its design and API, if applicable, are stable. The solution's development status is expressed using the following values: (1) Plan – (2) Design – (3) Development – (4) Integration and testing – (5) Deployment – (6) Maintenance/Operation (7) Disposal. *Example: A solution currently in production that is being maintained has the status of (6) Maintenance/Operation.*

[7] List of endorsed standards (interoperability specifications) were not publicly available by the time this report was concluded. Once available a reference of this list of endorsed standards should be included in the reusability factsheet.

[8] Related to EIF principle 9 "Openness" with regards to software openness defined in a formalised license.

Solution aspect	Description
Support	Information on the support offered by the solution's team to parties that would reuse it. Pertinent information in this case are the existence or not of a helpdesk, possibly with multiple levels of support, that is available to external parties. Additional important information is the size and composition of the team that is actively supporting the solution that would be available to resolve support issues. *Example: A service that offers a helpdesk accessible via email or telephone during weekdays and working hours, with a 5 person third-level support team (2 analysts, 3 developers) dedicated to resolving support issues.*
Scalability	The solution's scalability level, including measures that are taken in relation to this. Information on a solution's performance and ability to gracefully scale can be of significant importance depending on the scalability requirements of the overall solution. *Example: A service deployed on cloud infrastructure that offers high on-demand horizontal scalability depending on current needs.*
Security[9]	Any constraints or implications that the solution may bring in relation to security. Such security considerations can include communication aspects, data handling or involved processes. *Example: A data exchange component that supports encryption at the communication channel level but also on exchanged payloads.*
Testability	Test resources that are available to test integration or interoperability of the component or service in the overall envisaged solution. Such resources relate to available test suites but also the availability of test instances to facilitate integration and interoperability testing during the overall solution's elaboration process. *Example: A test instance of a service that is publicly available for interoperability testing purposes.*
Localisation and language[10]	Capabilities of the solution that enable its localisation. Localisation relates not only to the language used in reports and user interfaces but also to currencies, date formats and metrics. In addition, the possibility to access the solution's documentation in multiple languages enhances its accessibility for developers and potentially also for end users. *Example: A component offering localisation of produced reports based on XML configuration files per locale.*

[9] Related to EIF principle 4 "Security and privacy" with regards to the solution's security constraints or requirements.

[10] Related to EIF principle 5 "Multilingualism" with regards to its realisation within the solution.

Solution aspect	Description
Inclusion and accessibility[11]	Capabilities of the solution that allow the non-discrimination of its users. This can relate to measures taken in terms of design, information content and delivery to support impaired people or support user choice through multichannel delivery. *Example: A solution's user interface that allows audio user prompts in support of visually impaired users.*
Cost considerations	Cost information that can serve as input to the Total Cost of Ownership (TCO) analysis for the solution. A solution's TCO is defined as the combination of direct and indirect costs that its reuse entails. Indirect costs such as the need to adapt existing processes is specific per case, but certain information, especially with regards to the direct cost calculation, is common. Such costs range from ongoing licensing fees, initial setup costs for required hardware, but also to an estimated effort to train new users in the solution's use. Note that for solutions reusable both as services and components where cost considerations may differ per case, information is provided to address all reuse types. *Example: A software component that contains a licenced library resulting in an ongoing fee for its (re)use.*

3.3 Reusability aspects for solutions as software components

These represent further aspects to consider for a solution when evaluating its reuse as a software component. This component may be reused as a standalone installation within an overall solution, or possibly as a library forming an integral part of a software system. In addition, software components can be reused in an as-is manner or potentially modified to cater for customisations needed for the specific case.

Solution aspect	Description
Technology stack	The set of platforms and technologies used to develop the component. The platform (e.g. Java Enterprise Edition or .Net) defines the overall technological approach and is complemented by the individual libraries, frameworks and tools, including their version information, used to address individual technical concerns. This information is important to determine the compatibility of the component towards the overall solution and can serve to identify potential conflicts with regards to an organisation's technological guidelines. *Example: Use of Oracle 11g RDBMS for persistence, JEE7 as the overall platform, Hibernate 3 as an Object-Relational-Mapping (ORM) framework ...*

[11] Related to EIF principle 3 "Inclusion and accessibility" on its realisation within the solution (if applicable).

Solution aspect	Description
Architectural considerations	Information on the architecture of the component and any expectations that the component would have of the overall solution. Such considerations could relate to the configurability of the component, the way it expects to be interacted with, exposed APIs etc. Such information is important to determine whether from an architectural perspective the component can be seamlessly integrated into the overall solution. *Example: A component following an event-driven design that integrates with other components using the asynchronous exchange of messages.*
Hosting needs	Component requirements in terms of infrastructure resources. These range from capacity needs on storage, processing and networking, to restrictions relating to operating system configuration and virtualisation technologies. The hosting-related needs of a component could determine whether it complies with an organisation's hosting constraints and whether it is possible to fulfil specific capacity requirements. *Example: A component provided in the form of a VMWare image for reuse.*
Extensibility[12]	Information regarding the possibility to extend of modify the component to suit a specific solution's needs. This goes past using the component as a black box by adapting it at the source code level. Such adaptations are typically subject to the component's licence, but also are facilitated by the quality of the code and the presence of rich inline code commenting. *Example: A component provided as an open source reference implementation of a standard that is meant for reuse and customisation.*

3.4 Reusability aspects for solutions as services

These aspects represent points to consider when the solution is to be reused as a service. In this case the service is operated by a third party and is consumed remotely through appropriate service calls.

Solution aspect	Description
Availability	The availability level of the service. A service's availability characteristics are important to consider with respect to the availability of the overall solution and could be prohibitive for reuse if not at a certain level. The service's availability could be formalised through an SLA or at least documented as part of its non-functional requirements. However, even if this is not recorded in a formal manner, an indication of the service's availability would provide useful insight for its reuse. *Example: A service guaranteeing through its SLA 99% availability.*

[12] Related to EIF principle 9 "Openness" with regards to the software components source code.

Solution aspect	Description
Integration	Technical considerations that relate to accessing the service. These can range from network connectivity restrictions to security-related prerequisites. *Example: A service only accessible internally within the European Commission following ECAS authentication.*
Service agreement	A reference to a formalised service agreement or a description of the considerations and prerequisites regarding the use of the service. For a service meant to be used by end-users this could restrict its use to specific user groups whereas for a service consumed by another system it could entail mandatory interoperability tests as a prerequisite step. *Example: A service that is usable only by EU citizens.*
Data privacy[13]	Considerations regarding the handling of data within the service that relate to data privacy. This relates to the service's data policies, referring where possible to specific applied legal texts, but also to specific measures taken within the solution to ensure that data privacy requirements are upheld. *Example: A service that treats all received data as fully public.*
Data location and federation	Considerations regarding any datasets that the service maintains based on the data it handles. This could relate both to the location of stored data (if any) and to how data if federated into a central dataset (if applicable). Such considerations could be important when considering reuse for solutions that have specific requirements on how and where data is recorded. *Example: A service that is guaranteed to store handled data only in DIGIT's EU Institutions (EUI) datacentre.*

[13] Related to EIF principle 4 "Security and privacy" with regards to the privacy of data.

4 How to consider solution reusability aspects

The current section provides a guide on how to use the information contained in the current document in support of the use cases listed in section 2, ‘Reusability factsheet use cases’.

4.1 Step 1: Evaluation of overall feasibility of reuse

The first step is the evaluation of the overall feasibility of reusing the solution in question. When doing this it is important to consider that the recorded information provides only half of the input to the analysis, the other half coming from the specificities and requirements of the overall context for which reuse of the documented solution is being considered.

With this in mind the recorded information needs to be considered to identify **constraints** that would rule out reuse altogether. Example cases would be ties to a specific policy domain, or incompatible dependencies that cannot be subsequently applied to the overall solution. Furthermore, and considering that reuse in this case refers to reusing a building block from the EIRA’s technical view, the precise **nature of the solution** plays an important role. Reusing a service is typically simpler than reusing a software component, considering that a software component’s integration into the overall solution needs to be tighter than that of an external service.

Finally, certain reusability aspects may have **different weights** depending on the characteristics of the overall solution. For example, a solution that will deal with fully public data has simpler or even no requirements with regards to security and data privacy. Similarly, a service’s availability can be a critical evaluation point when being considered for reuse in a solution with strategic importance that cannot afford any, even partial, downtime.

With the envisioned solution’s characteristics in mind, a quick overview of the recorded information can provide a first indication of whether or not reuse is possible. An in-depth analysis can subsequently follow to determine the full impact of a candidate component or service’s reuse.

4.2 Step 2: Total Cost of Ownership analysis

Assuming that no blocking constraints are identified from the initial reuse evaluation, the next step is the detailed analysis of what reusing the component or service entails for the solution as a whole. A means of measuring this impact is a Total Cost of Ownership (TCO) analysis that in this context can be referred to as the analysis of the Total Cost of Reuse (TCR). The goal is to provide a monetary figure for the reuse, serving on one hand to provide a holistic estimation of its cost, and on the other as a quantifiable means of comparing multiple reuse candidates.

Calculating the Total Cost of Reuse is an exercise that takes into account two sources of costs:

- **Direct costs** relating to well-known or easily identifiable costs that typically apply regardless of context. An example is an ongoing fixed licensing cost for a reused component.
- **Indirect costs** relating to implications of the reuse that are not immediately apparent but need to be estimated. Such costs are typically highly dependent on the specific context within which the

reuse is planned. An example could be the estimated cost of the overall solution's downtime as a result of the unavailability of a reused service.

What is important to point out is that estimating the Total Cost of Reuse is an exercise that needs to take place each time reuse of a solution is considered. This is not an analysis that can be done once and shared as it depends significantly on the specific context. To facilitate the analysis the following table revisits certain reusability aspects to provide an indication of how they could affect the total cost in a direct or indirect manner.

Solution aspect	Potential impact on reuse cost
General reusability aspects	
Use of standards	Can be beneficial with regards to future costs if interoperability is an eventually envisaged goal, and especially so if the standards are endorsed as being interoperability enablers.
Solution portfolio	If multiple building blocks from a solution portfolio are planned to be reused, currently or as future extensions, then the fact that they are designed to work with each other can have a significant reduction in the cost of integrating them as a cohesive whole.
Documentation	Lack of rich documentation for a component or solution can entail significant effort (hence cost) in its integration in the overall solution and may result in suboptimal results.
Maturity	A mature solution that is currently in production and being actively maintained, serves to avoid potential costs that could be incurred by unforeseen changes if the solution was still in development.
Support	A well supported solution may offer significant cost reduction with regards to initial integration effort at the time of development, but also during operation to quickly resolve issues and unblock processes.
Scalability	A solution that has high scalability requirements may be faced with significant costs in lost revenue or inefficient alternative (e.g. manual) processes that need to be followed if overall scalability is negatively impacted by the limited scalability of a reused service or component.
Security	A reused component or service that compromises the overall security or a solution, assuming that this would be acceptable to begin with, may require significant costs in applying additional security measures or reacting to breaches.
Testability	If integration requirements for a reused component or service are complex, having high testability can significantly reduce integration and interoperability testing effort.
Localisation and language	An overall solution that has explicit localisation requirements may be faced with additional localisation costs, in developing custom automated localisation components or simply translating resources, if localisation is not well supported from a reused service or component.

Solution aspect	Potential impact on reuse cost
Inclusion and accessibility	A solution with poor accessibility features may incur additional costs in foreseeing alternative service delivery methods, possibly involving costly manual processes, for e.g. impaired users.
Cost considerations	This cost information represents direct costs that can be easily factored into the overall Total Cost of Reuse.
Component reusability aspects	
Architectural considerations	An architecturally incompatible component may require additional development for the solution as a whole to ensure correct integration with a reused component.
Hosting needs	Specific hosting requirements could incur additional costs for the overall solution owner depending on the infrastructure already available or foreseen.
Extensibility	Solutions that are expected to evolve significantly over time (e.g. due to policy domain changes) may have high costs evolving reused components that are not built with extensibility in mind.
Service reusability aspects	
Availability	An overall solution of strategic importance may have a high cost due to reduced availability of a reused service that would require alternative delivery methods (if even possible). In contrast, this cost could be negligible if increased availability for the overall solution is not important.
Data privacy	Similar to security concerns, reduced data privacy characteristics of a reused service may incur costs in ensuring privacy as part of the overall solution, assuming that reuse would still be possible to begin with.

5 Reusability Factsheet Mock-up

This section proposes a mock-up of the reusability factsheet (Table 5:) to give an indication of how it could be presented. The main focus is placed on the discussed reusability aspects with additional information made available in annex form:

- The prerequisites for using the reusability factsheet presented in section 1;
- The use cases presented in section 2, and;
- A guide on how to use the information contained in the reusability factsheet for assessing the solution's reuse and its impact presented in section 4.

It is very important to note that additional work is needed to refine exactly how the final factsheet will be linked to Joinup. Regarding data collection, a significant set of information will be already covered by a solution's documentation process for the EIC, through its documented EIRA solution building blocks and attributes. Information that is not covered will need to be collected as a supplement, and most likely in an online manner through Joinup. In doing so, information already recorded in Joinup, such as existing solutions of specific types, can be leveraged to facilitate the data input by e.g. providing prefilled selection lists. Similarly, the means through which the factsheet is delivered to interested parties, is also subject to further analysis. A possible approach would be to expose the factsheet online through Joinup, considering that Joinup is the underlying information source and that it represents the solutions' core documentation space.

Considering the envisaged link to Joinup, the mock factsheet that follows should be considered not as its proposed final form but rather as a proof-of-concept.

Table 5: Reusability factsheet mock-up

isa Interoperability Solutions for European Public Administrations	**Reusability Factsheet – [Solution Name]**	
I. GENERAL INFORMATION		
Domain of applicability		**Details on the domain of applicability**
The solution can be reused only as a software component	☐	*Please provide detais on the domain of reuse of the solution*
The solution can be reused only as a service	☐	**EIRA Building Blocks**[14]
The solution can be reused as a service and as a software component	☐	*Please indicate the EIRA building blocks from the technical view that are being addressed by the current information on reuse.*

[14] Available at: https://joinup.ec.europa.eu/asset/eia/asset_release/all

II. GENERAL REUSABILITY ASPECTS

The following reusability aspects have general applicability when considering solutions reuse. They apply both in the case of reusing the solution as a service and also when the solution is reused as a software component.

Actual reuse

Cases where the solution has already been reused, either as a service or as a software component. The extent to which a solution is already reused is a good indication of its maturity and reusability, both in technical terms but also potentially in terms of policy domains.

Example: e-TrustEx is reused as the data exchange component in e-Prior in the Procurement domain.

Please provide detais on the solution's actual reuse

Planned reuse or extension to other domains

Cases for which the solution is planned to be reused, within the same policy domain, or even extending to different policy domains. Coupled with cases where the solution has already been reused, the plans for its reuse or extension to other policy domains offer an indication of the solution's maturity and ability to act as a generic, cross-domain or multi-domain solution building block.

Example: A solution initially developed in the domain of Environment, Consumers and Health that is planned to be reused in 2018 in the domain of Climate Action.

Please provide detais on the solution's planned reuse or extension to other domains

Design aspects favouring reuse

Measures taken to ensure that this solution has been built as reusable from the ground up. These measures could relate to the choice of non-proprietary technologies, choice of a modular software architecture, or adoption of a SOA approach in exposing reusable services.

Example: Exposure of all business capabilities through a SOAP-based web service API.

Please provide detais on the solution's design aspects favouring reuse.

Dependencies and reuse constraints

Dependencies or constraints that this solution entails with regards to its reuse. These dependencies can either be of a technical nature (e.g. reliance on a specific third party product) or relative to specific legislation or business domain. Moreover, the solution may be reusable but only within a specific community or geographic location.

*Example: A software component with a strong dependency on Oracle RDBMS 10g as a database provider***.**

Please provide detais on the solution's dependencies and reuse constraints

Use of standards

The standards that the solution uses and conforms to ranging from business and data standards to technical and communication standards. The standards used by a solution could either make it an ideal reuse candidate for a specific solution or exclude it due to conflicting choices already made. Note that the standards listed as interoperability specifications[7] for the solution's related architecture building blocks (as per the EIRA), are considered as endorsed for being best-in-class with regards to enabling interoperability.

Example: Use of SAML v2.0 for the communication of user authentication, rights and attributes, and AS4 for data exchange of business documents.

Please provide detais on the use of standards in the solution

Solution portfolio

Whether or not the solution is part of a portfolio of compatible solution building blocks, either components or services, designed to work alone or together. If so, this is an indication of the solution's modularity and design for reuse, but also opens up possibilities to reuse other solution building blocks from its portfolio.

Example: The CEF (Connecting Europe Facility) eID solution building block for cross-border authentication that is part of a European Commission catalogue of generic building blocks built for reuse (eSignature, eInvoicing, eDelivery, Automated Translation).

Please provide detais on the portfolio this solution is a part of

Intellectual Property Rights and licensing[15]

The formal definition that refers to or describes the licensing mechanism, ownership rights, restrictions, and user responsibilities related to the distribution and reuse of the solution. In case the solution is usable both as a service and as a software component with different licensing schemes per case, all cases are documented.

Example: A software component with an MIT licence that can be reused without copyright restrictions.

Please provide detais on the solution's Intellectual Property Rights and licensing

Documentation

References to the solution's available documentation. This documentation ranges from background information on the solution such as its history and development team, to technical documentation describing the services exposed, core processes, the architecture, as well as its installation and configuration (if applicable). The availability of rich documentation not only allows for a detailed investigation of the solution but also offers a strong indication of the solution's maturity and support level.

Example: Reference to a solution's wiki documenting all aspects of its design and use.

Please provide detais on the solution's documentation

[15] Related to EIF principle 9 "Openness" with regards to software openness defined in a formalised license.

Maturity

The status of the solution in terms of maturity, to indicate its ongoing development and whether its design and API, if applicable, are stable. The solution's development status is expressed using the following values: (1) Plan – (2) Design – (3) Development – (4) Integration and testing – (5) Deployment – (6) Maintenance/Operation (7) Disposal.

Example: A solution currently in production that is being maintained has the status of (6) Maintenance/Operation.

Please provide detais on the solution's maturity

Support

Information on the support offered by the solution's team to parties that would reuse it. Pertinent information in this case are the existence or not of a helpdesk, possibly with multiple levels of support, that is available to external parties. Additional important information is the size and composition of the team that is actively supporting the solution that would be available to resolve support issues.

Example: A service that offers a helpdesk accessible via email or telephone during weekdays and working hours, with a 5 person third-level support team (2 analysts, 3 developers) dedicated to resolving support issues.

Please provide detais on the solution's support

Scalability

The solution's scalability level, including measures that are taken in relation to this. Information on a solution's performance and ability to gracefully scale can be of significant importance depending on the scalability requirements of the overall solution.

Example: A service deployed on cloud infrastructure that offers high on-demand horizontal scalability depending on current needs.

Please provide detais on the solutions' scalability

Security[16]

Any constraints or implications that the solution may bring in relation to security. Such security considerations can include communication aspects, data handling or involved processes.

Example: A data exchange component that supports encryption at the communication channel level but also on exchanged payloads.

Please provide detais on the solutions' security

[16] Related to EIF principle 4 "Security and privacy" with regards to the solution's security constraints or requirements.

Testability

Test resources that are available to test integration or interoperability of the component or service in the overall envisaged solution. Such resources relate to available test suites but also the availability of test instances to facilitate integration and interoperability testing during the overall solution's elaboration process.

Example: A test instance of a service that is publicly available for interoperability testing purposes.

Please provide detais on the solution's testability

Localisation and language[17]

Capabilities of the solution that enable its localisation. Localisation relates not only to the language used in reports and user interfaces but also to currencies, date formats and metrics. In addition, the possibility to access the solution's documentation in multiple languages enhances its accessibility for developers and potentially also for end users.

Example: A component offering localisation of produced reports based on XML configuration files per locale.

Please provide detais on the solution's localisation and language

Inclusion and accessibility[18]

Capabilities of the solution that allow the non-discrimination of its users. This can relate to measures taken in terms of design, information content and delivery to support impaired people or support user choice through multichannel delivery.

Example: A solution's user interface that allows audio user prompts in support of visually impaired users.

Please provide detais on the solution's inclusion and accessibility

Cost considerations

Cost information that can serve as input to the Total Cost of Ownership (TCO) analysis for the solution. A solution's TCO is defined as the combination of direct and indirect costs that its reuse entails. Indirect costs such as the need to adapt existing processes is specific per case, but certain information, especially with regards to the direct cost calculation, is common. Such costs range from ongoing licensing fees, initial setup costs for required hardware, but also to an estimated effort to train new users in the solution's use. Note that for solutions reusable both as services and components where cost considerations may differ per case, information is provided to address all reuse types.

Example: A software component that contains a licenced library resulting in an ongoing fee for its (re)use.

Please provide detais on the cost considerations of reusing the solution

II. REUSABILITY ASPECTS FOR SOLUTIONS AS SOFTWARE COMPONENTS

[17] Related to EIF principle 5 "Multilingualism" with regards to its realisation within the solution.

[18] Related to EIF principle 3 "Inclusion and accessibility" on its realisation within the solution (if applicable).

These represent further aspects to consider for a solution when evaluating its reuse as a software component. This component may be reused as a standalone installation within an overall solution, or possibly as a library forming an integral part of a software system. In addition, software components can be reused in an as-is manner or potentially modified to cater for customisations needed for the specific case.

Technology stack

The set of platforms and technologies used to develop the component. The platform (e.g. Java Enterprise Edition or .Net) defines the overall technological approach and is complemented by the individual libraries, frameworks and tools, including their version information, used to address individual technical concerns. This information is important to determine the compatibility of the component towards the overall solution and can serve to identify potential conflicts with regards to an organisation's technological guidelines.

Example: Use of Oracle 11g RDBMS for persistence, JEE7 as the overall platform, Hibernate 3 as an Object-Relational-Mapping (ORM) framework.

Please provide detais on the solution's technology stack

Architectural considerations

Information on the architecture of the component and any expectations that the component would have of the overall solution. Such considerations could relate to the configurability of the component, the way it expects to be interacted with, exposed APIs etc. Such information is important to determine whether from an architectural perspective the component can be seamlessly integrated into the overall solution.

Example: A component following an event-driven design that integrates with other components using the asynchronous exchange of messages.

Please provide architectural considerations on the solution

Hosting needs

Component requirements in terms of infrastructure resources. These range from capacity needs on storage, processing and networking, to restrictions relating to operating system configuration and virtualisation technologies. The hosting-related needs of a component could determine whether it complies with an organisation's hosting constraints and whether it is possible to fulfil specific capacity requirements.

Example: A component provided in the form of a VMWare image for reuse.

Please provide detais on the solution's hosting needs

Extensibility[19]

Information regarding the possibility to extend of modify the component to suit a specific solution's needs. This goes past using the component as a black box by adapting it at the source code level. Such adaptations are typically subject to the component's licence, but also are facilitated by the quality of the code and the presence of rich inline code commenting.

Example: A component provided as an open source reference implementation of a standard that is meant for reuse and customisation.

Please provide detais on the solution's extensibility

III. REUSABILITY ASPECTS FOR SOLUTIONS AS SERVICES

These aspects represent points to consider when the solution is to be reused as a service. In this case the service is operated by a third party and is consumed remotely through appropriate service calls.

Availability

The availability level of the service. A service's availability characteristics are important to consider with respect to the availability of the overall solution and could be prohibitive for reuse if not at a certain level. The service's availability could be formalised through an SLA or at least documented as part of its non-functional requirements. However, even if this is not recorded in a formal manner, an indication of the service's availability would provide useful insight for its reuse.

Example: A service guaranteeing through its SLA 99% availability.

Please provide detais on the solution's availability

Integration

Technical considerations that relate to accessing the service. These can range from network connectivity restrictions to security-related prerequisites.

Example: A service only accessible internally within the European Commission following ECAS authentication.

Please provide detais on the solution's integration

[19] Related to EIF principle 9 "Openness" with regards to the software components source code.

Service agreement

A reference to a formalised service agreement or a description of the considerations and prerequisites regarding the use of the service. For a service meant to be used by end-users this could restrict its use to specific user groups whereas for a service consumed by another system it could entail mandatory interoperability tests as a prerequisite step.

Example: A service that is usable only by EU citizens.

Please provide detais on the solution's service agreements

Data privacy[20]

Considerations regarding the handling of data within the service that relate to data privacy. This relates to the service's data policies, referring where possible to specific applied legal texts, but also to specific measures taken within the solution to ensure that data privacy requirements are upheld.

Example: A service that treats all received data as fully public.

Please provide detais on the solution's data privacy

Data location and federation

Considerations regarding any datasets that the service maintains based on the data it handles. This could relate both to the location of stored data (if any) and to how data if federated into a central dataset (if applicable). Such considerations could be important when considering reuse for solutions that have specific requirements on how and where data is recorded.

Example: A service that is guaranteed to store handled data only in DIGIT's EU Institutions (EUI) datacentre.

Please provide detais on the solution's data location and federation

[20] Related to EIF principle 4 "Security and privacy" with regards to the privacy of data.

Annex A Reusability factsheet for e-TrustEx

The current annex provides a completed reusability factsheet for DIGIT's e-TrustEx solution.

Table 6: Reusability factsheet for e-TrustEx

Reusability Factsheet – [Solution Name]

I. GENERAL INFORMATION

Domain of applicability		Details on the domain of applicability
The solution can be reused only as a software component	☐	*EIRA Technical View - Infrastructure*
The solution can be reused only as a service	☐	**EIRA Building Blocks**[21]
The solution can be reused as a service and as a software component	☒	*Data Exchange Service* *Data Exchange Component*

II. GENERAL REUSABILITY ASPECTS

The following reusability aspects have general applicability when considering solutions reuse. They apply both in the case of reusing the solution as a service and also when the solution is reused as a software component.

Actual reuse

Cases where the solution has already been reused, either as a service or as a software component. The extent to which a solution is already reused is a good indication of its maturity and reusability, both in technical terms but also potentially in terms of policy domains.

Example: e-TrustEx is reused as the data exchange component in e-Prior in the Procurement domain.

e-TrustEx is currently reused as the data exchange platform for the following solutions:
- e-Justice Portal (DG JUST) in the Justice and Citizens' Rights domain.
- e-CODEX (DG CNECT) in the Justice and Citizens' Rights domain.
- IMI (DG MARKT), in the Single Market domain.
- CIPA e-Delivery (DIGIT), which is a cross-domain solution.
- e-Prior (DIGIT) in the Procurement domain.
- e-Greffe (SG).
- Council IS (European Council).
- OP IS (OP).
- ASAP (SG).
- EDMA (DG COMP).

[21] Available at: https://joinup.ec.europa.eu/asset/eia/asset_release/all

Planned reuse or extension to other domains

Cases for which the solution is planned to be reused, within the same policy domain, or even extending to different policy domains. Coupled with cases where the solution has already been reused, the plans for its reuse or extension to other policy domains offer an indication of the solution's maturity and ability to act as a generic, cross-domain or multi-domain solution building block.

Example: A solution initially developed in the domain of Environment, Consumers and Health that is planned to be reused in 2018 in the domain of Climate Action.

e-TrustEx is planned for future reuse in the following solutions/domains:
- In the European Parliament.
- In GENIS (DG COMP) for submission of State Aid Notifications to DG COMP back-office and the exchange of attachments with the Member States.
- e-TrustEx has been proposed by DG TAXUD as a solution to exchange Taxation data between Member States and OECD countries.
- e-TrustEx components will be reused in the future CEF e-Delivery connector.

Design aspects favouring reuse

Measures taken to ensure that this solution has been built as reusable from the ground up. These measures could relate to the choice of non-proprietary technologies, choice of a modular software architecture, or adoption of a SOA approach in exposing reusable services.

Example: Exposure of all business capabilities through a SOAP-based web service API.

e-TrustEx foresees generic data exchange web services for the transport of any business payload that can be reused in any domain.
When used as a component, e-TrustEx's foundation on open source technologies, and its own EUPL licence, facilitates extension to include added-value domain-specific services per case.

Dependencies and reuse constraints

Dependencies or constraints that this solution entails with regards to its reuse. These dependencies can either be of a technical nature (e.g. reliance on a specific third party product) or relative to specific legislation or business domain. Moreover, the solution may be reusable but only within a specific community or geographic location.

Example: A software component with a strong dependency on Oracle RDBMS 10g as a database provider.

None, since e-TrustEx was custom developed to be a reusable data exchange service and component that can be reused in any domain.

Use of standards

The standards that the solution uses and conforms to ranging from business and data standards to technical and communication standards. The standards used by a solution could either make it an ideal reuse candidate for a specific solution or exclude it due to conflicting choices already made. Note that the standards listed as interoperability specifications[7] for the solution's related architecture building blocks (as per the EIRA), are considered as endorsed for being best-in-class with regards to enabling interoperability.

Example: Use of SAML v2.0 for the communication of user authentication, rights and attributes, and AS4 for data exchange of business documents.

e-TrustEx uses the following standards to realise its data exchange capabilities:
- SOAP (e-TrustEx's generic services are realised as SOAP web services).
- XML (used in web service communication and to structure the business payloads).
- JMS (used to realise queueing-based asynchronous processing but also as a communication means with external systems that have appropriate access).
- WS-Security (SOAP extensions to implement message content integrity and confidentiality).
- WS-Policy (used to describe web service policies).
- WS-SecurityPolicy (security related policies for web services).
- MTOM (used for efficient transmission of binary data to and from SOAP web services).

Solution portfolio

Whether or not the solution is part of a portfolio of compatible solution building blocks, either components or services, designed to work alone or together. If so, this is an indication of the solution's modularity and design for reuse, but also opens up possibilities to reuse other solution building blocks from its portfolio.

Example: The CEF (Connecting Europe Facility) eID solution building block for cross-border authentication that is part of a European Commission catalogue of generic building blocks built for reuse (eSignature, eInvoicing, eDelivery, Automated Translation).

e-TrustEx is not part of a solution portfolio.

Intellectual Property Rights and licensing[22]

The formal definition that refers to or describes the licensing mechanism, ownership rights, restrictions, and user responsibilities related to the distribution and reuse of the solution. In case the solution is usable both as a service and as a software component with different licensing schemes per case, all cases are documented.

Example: A software component with an MIT licence that can be reused without copyright restrictions.

e-TrustEx is offered as a component under the EUPL 1.1 licence. In that respect it offers:
- freedom to use or run it for any purpose and any number of users;
- freedom to obtain the Source Code (in order to study how the software works);
- freedom to share and redistribute copies of the software;
- freedom to modify, adapt and improve the software according to specific needs and to share these modifications.

Documentation

References to the solution's available documentation. This documentation ranges from background information on the solution such as its history and development team, to technical documentation describing the services exposed, core processes, the architecture, as well as its installation and configuration (if applicable). The availability of rich documentation not only allows for a detailed investigation of the solution but also offers a strong indication of the solution's maturity and support level.

Example: Reference to a solution's wiki documenting all aspects of its design and use.

e-TrustEx's documentation home is in Joinup at: https://joinup.ec.europa.eu/software/openetrustex

Maturity

The status of the solution in terms of maturity, to indicate its ongoing development and whether its design and API, if applicable, are stable. The solution's development status is expressed using the following values: (1) Plan – (2) Design – (3) Development – (4) Integration and testing – (5) Deployment – (6) Maintenance/Operation (7) Disposal.

Example: A solution currently in production that is being maintained has the status of (6) Maintenance/Operation.

Status: Maintenance/Operation

e-TrustEx is in operation since 2012.

[22] Related to EIF principle 9 "Openness" with regards to software openness defined in a formalised license.

Support

Information on the support offered by the solution's team to parties that would reuse it. Pertinent information in this case are the existence or not of a helpdesk, possibly with multiple levels of support, that is available to external parties. Additional important information is the size and composition of the team that is actively supporting the solution that would be available to resolve support issues.

Example: A service that offers a helpdesk accessible via email or telephone during weekdays and working hours, with a 5 person third-level support team (2 analysts, 3 developers) dedicated to resolving support issues.

A service that offers a helpdesk accessible via email or telephone during weekdays and working hours

Scalability

The solution's scalability level, including measures that are taken in relation to this. Information on a solution's performance and ability to gracefully scale can be of significant importance depending on the scalability requirements of the overall solution.

Example: A service deployed on cloud infrastructure that offers high on-demand horizontal scalability depending on current needs.

e-TrustEx has medium scalability. Its architecture allows horizontal scaling through use of clustering. In addition, the use of JMS-based queueing allows e-TrustEx to support high message loads without exceeding its capacity.

Security[23]

Any constraints or implications that the solution may bring in relation to security. Such security considerations can include communication aspects, data handling or involved processes.

Example: A data exchange component that supports encryption at the communication channel level but also on exchanged payloads.

Overall the security level of e-TrustEx varies depending on the domain in which is is used. It ranges from Sensitive (e.g. for procurement domain) to Critical (e.g. for competition domain).
In terms of specific measures:
- Authentication: Basic authentication over HTTPS or 2-way SSL with client authentication via its certificate.
- Authorisation: Interchange Agreement Services to model the contract between parties in the context of a specific document exchange profile (a set of transactions that define document operations).
- Channel encryption: HTTPS for channel encryption.
- Message encryption: Encryption using certificate key-pairs.
- Non-repudiation: Use of digital signatures in messages.
- Auditing: Audit trail for all requests.
- Integrity: The platform supports point to point and end to end digital signatures to ensure the integrity of the message transmission.

[23] Related to EIF principle 4 "Security and privacy" with regards to the solution's security constraints or requirements.

Testability

Test resources that are available to test integration or interoperability of the component or service in the overall envisaged solution. Such resources relate to available test suites but also the availability of test instances to facilitate integration and interoperability testing during the overall solution's elaboration process.

Example: A test instance of a service that is publicly available for interoperability testing purposes.

e-TrustEx offers a test/demo framework comprising of SOAP UI test suites.

Localisation and language[24]

Capabilities of the solution that enable its localisation. Localisation relates not only to the language used in reports and user interfaces but also to currencies, date formats and metrics. In addition, the possibility to access the solution's documentation in multiple languages enhances its accessibility for developers and potentially also for end users.

Example: A component offering localisation of produced reports based on XML configuration files per locale.

Not applicable due to the nature of e-TrustEx as a generic data exchange solution.

Inclusion and accessibility[25]

Capabilities of the solution that allow the non-discrimination of its users. This can relate to measures taken in terms of design, information content and delivery to support impaired people or support user choice through multichannel delivery.

Example: A solution's user interface that allows audio user prompts in support of visually impaired users.

No specific measures to support accessibility exist although the nature of e-TrustEx as a payload-agnostic data exchange solution make this not relevant.

Cost considerations

Cost information that can serve as input to the Total Cost of Ownership (TCO) analysis for the solution. A solution's TCO is defined as the combination of direct and indirect costs that its reuse entails. Indirect costs such as the need to adapt existing processes is specific per case, but certain information, especially with regards to the direct cost calculation, is common. Such costs range from ongoing licensing fees, initial setup costs for required hardware, but also to an estimated effort to train new users in the solution's use. Note that for solutions reusable both as services and components where cost considerations may differ per case, information is provided to address all reuse types.

Example: A software component that contains a licenced library resulting in an ongoing fee for its (re)use.

- No licensing costs for the software itself (e-TrustEx and used libraries).
- Costs could be incurred by the selected middleware used to host e-TrustEx although a deployment on open source middleware without licence costs is possible and tested.

II. REUSABILITY ASPECTS FOR SOLUTIONS AS SOFTWARE COMPONENTS

[24] Related to EIF principle 5 "Multilingualism" with regards to its realisation within the solution.

[25] Related to EIF principle 3 "Inclusion and accessibility" on its realisation within the solution (if applicable).

These represent further aspects to consider for a solution when evaluating its reuse as a software component. This component may be reused as a standalone installation within an overall solution, or possibly as a library forming an integral part of a software system. In addition, software components can be reused in an as-is manner or potentially modified to cater for customisations needed for the specific case.

Technology stack

The set of platforms and technologies used to develop the component. The platform (e.g. Java Enterprise Edition or .Net) defines the overall technological approach and is complemented by the individual libraries, frameworks and tools, including their version information, used to address individual technical concerns. This information is important to determine the compatibility of the component towards the overall solution and can serve to identify potential conflicts with regards to an organisation's technological guidelines.

Example: Use of Oracle 11g RDBMS for persistence, JEE7 as the overall platform, Hibernate 3 as an Object-Relational-Mapping (ORM) framework.

e-TrustEx is a Java Enterprise Edition solution built using the Spring framework.
Its lightweight ESB architecture is realised using Spring Integration with web services implemented using the Spring WS framework.
Its web frontend uses Spring MVC, JQuery and Spring Security.
Persistence, from the application's perspective, is realised using a relational database accessed using JPA 2.0.

In terms of middleware, e-TrustEx is flexible in its deployment as it is loosely coupled to specific middleware products. Two deployment scenarios have been envisaged and tested:
- Based on Oracle technology at the European Commission Data Centre: use of (a) Oracle Database 10g for persistence, (b) WebLogic Application Server 11g for document management and (c) Oracle Service Bus 11g for user access, service triggering, monitoring and logging.
- Based on open source: MySQL for persistence, and JBoss application server for hosting e-TrustEx's messaging backend and web applications.

Architectural considerations

Information on the architecture of the component and any expectations that the component would have of the overall solution. Such considerations could relate to the configurability of the component, the way it expects to be interacted with, exposed APIs etc. Such information is important to determine whether from an architectural perspective the component can be seamlessly integrated into the overall solution.

Example: A component following an event-driven design that integrates with other components using the asynchronous exchange of messages.

- Supports synchronous and asynchronous messaging.
- Messaging is through SOAP web services and JMS that favour e-TrustEx's integration in an overall solution in a loosely coupled manner.
- Routing, validation rules and all other considerations relating to the data exchange are configured in the database without the need to redeploy or restart upon modification.

Hosting needs

Component requirements in terms of infrastructure resources. These range from capacity needs on storage, processing and networking, to restrictions relating to operating system configuration and virtualisation technologies. The hosting-related needs of a component could determine whether it complies with an organisation's hosting constraints and whether it is possible to fulfil specific capacity requirements.

Example: A component provided in the form of a VMWare image for reuse.

- The platfrom web services must be exposed on the web and secured via SSL (one way). Setting up the access and the SSL channel is the responsibility of the implementer.
- The binary documents contained in messages are stored on the file system and thus sufficient disk space must be provided to the platform.

Extensibility[26]

Information regarding the possibility to extend of modify the component to suit a specific solution's needs. This goes past using the component as a black box by adapting it at the source code level. Such adaptations are typically subject to the component's licence, but also are facilitated by the quality of the code and the presence of rich inline code commenting.

Example: A component provided as an open source reference implementation of a standard that is meant for reuse and customisation.

The source code of e-TrustEx is fully extensible as it is based on popular open source libraries (e.g. Spring framework) and is itself offered as free and open source through its EUPL 1.1 licence.

III. REUSABILITY ASPECTS FOR SOLUTIONS AS SERVICES

These aspects represent points to consider when the solution is to be reused as a service. In this case the service is operated by a third party and is consumed remotely through appropriate service calls.

Availability

The availability level of the service. A service's availability characteristics are important to consider with respect to the availability of the overall solution and could be prohibitive for reuse if not at a certain level. The service's availability could be formalised through an SLA or at least documented as part of its non-functional requirements. However, even if this is not recorded in a formal manner, an indication of the service's availability would provide useful insight for its reuse.

Example: A service guaranteeing through its SLA 99% availability.

The availability of e-TrustEx is 99% (including possible maintenance time).

The availability level as well as multiple additional hosting and service characteristics are defined in the provided standard DIGIT IS Hosting SLA (version 1.5 updated 26/04/2013).

Integration

Technical considerations that relate to accessing the service. These can range from network connectivity restrictions to security-related prerequisites.

Example: A service only accessible internally within the European Commission following ECAS authentication.

No restrictions exist per se in the definition of e-TrustEx's services, and moreover aspects such as use of digital signatures are optional to best suit varying needs.

Service agreement

A reference to a formalised service agreement or a description of the considerations and prerequisites regarding the use of the service. For a service meant to be used by end-users this could restrict its use to specific user groups whereas for a service consumed by another system it could entail mandatory interoperability tests as a prerequisite step.

Example: A service that is usable only by EU citizens.

Not available.

[26] Related to EIF principle 9 "Openness" with regards to the software components source code.

Data privacy[27]

Considerations regarding the handling of data within the .service that relate to data privacy. This relates to the service's data policies, referring where possible to specific applied legal texts, but also to specific measures taken within the solution to ensure that data privacy requirements are upheld.

Example: A service that treats all received data as fully public.

A data protection assessment has been carried out and a data protection statement has been submitted to the commission DPO.

Data location and federation

Considerations regarding any datasets that the service maintains based on the data it handles. This could relate both to the location of stored data (if any) and to how data if federated into a central dataset (if applicable). Such considerations could be important when considering reuse for solutions that have specific requirements on how and where data is recorded.

Example: A service that is guaranteed to store handled data only in DIGIT's EU Institutions (EUI) datacentre.

e-TrustEx's Oracle Database uses one schema for the storage of XML messages, configuration, auditing and security-related information, and a NAS (Network-Attached Storage) file system for the storage of large binary files. For additional data privacy protection all files are stored encrypted by the platform .All information is recorded and remains located at the European Commission Data Centre of DIGIT.

[27] Related to EIF principle 4 "Security and privacy" with regards to the privacy of data.

Annex B Reusability factsheet for CIPA e-Delivery

The current annex provides a completed reusability factsheet for DIGIT's CIPA e-Delivery solution.

Table 7: Reusability factsheet for CIPA e-Delivery

Reusability Factsheet – [Solution Name]

I. GENERAL INFORMATION

Domain of applicability		Details on the domain of applicability
The solution can be reused only as a software component	☐	*EIRA Technical View - Infrastructure*
The solution can be reused only as a service	☐	**EIRA Building Blocks**[28]
The solution can be reused as a service and as a software component	☒	*Data Exchange Service* *Data Exchange Component*

II. GENERAL REUSABILITY ASPECTS

The following reusability aspects have general applicability when considering solutions reuse. They apply both in the case of reusing the solution as a service and also when the solution is reused as a software component.

Actual reuse

Cases where the solution has already been reused, either as a service or as a software component. The extent to which a solution is already reused is a good indication of its maturity and reusability, both in technical terms but also potentially in terms of policy domains.

Example: e-TrustEx is reused as the data exchange component in e-Prior in the Procurement domain.

CIPA e-Delivery is currently reused as the data exchange platform for the following solutions:
- e-TrustEx (DIGIT) has interfaces with the e-Delivery solution allowing to forward documents to the e-Delivery network.
- e-Prior (DIGIT), as an extension of e-TrustEx in the Procurement domain, interfaces with the e-Delivery network
- Reused as a service in the Procurement Domain for pre-award and post-award transactions (e-Order, e-Invoicing, e-Catalogue).
- Reused as a service in the Justice domain for European Payment Order transactions.

Planned reuse or extension to other domains

Cases for which the solution is planned to be reused, within the same policy domain, or even extending to different policy domains. Coupled with cases where the solution has already been reused, the plans for its reuse or extension to other policy domains offer an indication of the solution's maturity and ability to act as a generic, cross-domain or multi-domain solution building block.

Example: A solution initially developed in the domain of Environment, Consumers and Health that is planned to be reused in 2018 in the domain of Climate Action.

CIPA e-Delivery is a generic solution that could be reused in any policy area requiring the cross-border exchange of electronic documents in an asynchronous manner.

[28] Available at: https://joinup.ec.europa.eu/asset/eia/asset_release/all

Design aspects favouring reuse

Measures taken to ensure that this solution has been built as reusable from the ground up. These measures could relate to the choice of non-proprietary technologies, choice of a modular software architecture, or adoption of a SOA approach in exposing reusable services.

Example: Exposure of all business capabilities through a SOAP-based web service API.

CIPA e-Delivery offers generic, asynchronous transfer of business documents based on use of standard B2B protocols (AS2 and AS4) and is designed to be reused in its whole in another domain or context.

Its consists of three main components which, in terms of reuse:
- The (a) Access Point can be reused to provide access to a transport infrastructure.
- The (b) Service Metadata Publisher and (c) Service Metadata Locator can be reused to provide discovery functionalities.

Reuse of CIPA as a software component, with possible source-level extensions and new deployments, is favoured through the use of non-proprietary technologies and its own EUPL 1.1 licence.

Dependencies and reuse constraints

Dependencies or constraints that this solution entails with regards to its reuse. These dependencies can either be of a technical nature (e.g. reliance on a specific third party product) or relative to specific legislation or business domain. Moreover, the solution may be reusable but only within a specific community or geographic location.

Example: A software component with a strong dependency on Oracle RDBMS 10g as a database provider.

None, considering that CIPA was designed to be a reusable data exchange service and component.

Use of standards

The standards that the solution uses and conforms to ranging from business and data standards to technical and communication standards. The standards used by a solution could either make it an ideal reuse candidate for a specific solution or exclude it due to conflicting choices already made. Note that the standards listed as interoperability specifications[7] for the solution's related architecture building blocks (as per the EIRA), are considered as endorsed for being best-in-class with regards to enabling interoperability.

Example: Use of SAML v2.0 for the communication of user authentication, rights and attributes, and AS4 for data exchange of business documents.

CIPA e-Delivery uses the following standards to realise its data exchange capabilities:
- SOAP (generic services are realised as SOAP web services).
- XML (used in web service communication and to structure the business payloads).
- AS2 (for business payload exchange).
- AS4 (for business payload exchange).
- REST/HTTP (to access the SMP metadata).

Solution portfolio

Whether or not the solution is part of a portfolio of compatible solution building blocks, either components or services, designed to work alone or together. If so, this is an indication of the solution's modularity and design for reuse, but also opens up possibilities to reuse other solution building blocks from its portfolio.

Example: The CEF (Connecting Europe Facility) eID solution building block for cross-border authentication that is part of a European Commission catalogue of generic building blocks built for reuse (eSignature, eInvoicing, eDelivery, Automated Translation).

CIPA e-Delivery realises the CEF (Connecting Europe Infrastructure) e-Delivery solution building block. As such, it is part of CEF's catalogue of generic building blocks (eID, eSignature, eDelivery, eInvoicing and Automated Translation).

Intellectual Property Rights and licensing[29]

The formal definition that refers to or describes the licensing mechanism, ownership rights, restrictions, and user responsibilities related to the distribution and reuse of the solution. In case the solution is usable both as a service and as a software component with different licensing schemes per case, all cases are documented.

Example: A software component with an MIT licence that can be reused without copyright restrictions.

CIPA e-Delivery is released under the EUPL 1.1 licence. In that respect it offers:
- freedom to use or run it for any purpose and any number of users;
- freedom to obtain the Source Code (in order to study how the software works);
- freedom to share and redistribute copies of the software;
- freedom to modify, adapt and improve the software according to specific needs and to share these modifications.

Documentation

References to the solution's available documentation. This documentation ranges from background information on the solution such as its history and development team, to technical documentation describing the services exposed, core processes, the architecture, as well as its installation and configuration (if applicable). The availability of rich documentation not only allows for a detailed investigation of the solution but also offers a strong indication of the solution's maturity and support level.

Example: Reference to a solution's wiki documenting all aspects of its design and use.

CIPA e-Delivery's documentation home is in Joinup at: https://joinup.ec.europa.eu/software/cipaedelivery

Maturity

The status of the solution in terms of maturity, to indicate its ongoing development and whether its design and API, if applicable, are stable. The solution's development status is expressed using the following values: (1) Plan – (2) Design – (3) Development – (4) Integration and testing – (5) Deployment – (6) Maintenance/Operation (7) Disposal.

Example: A solution currently in production that is being maintained has the status of (6) Maintenance/Operation.

Status: Maintenance/Operation

CIPA e-Delivery is in operation since 2011.

Support

Information on the support offered by the solution's team to parties that would reuse it. Pertinent information in this case are the existence or not of a helpdesk, possibly with multiple levels of support, that is available to external parties. Additional important information is the size and composition of the team that is actively supporting the solution that would be available to resolve support issues.

Example: A service that offers a helpdesk accessible via email or telephone during weekdays and working hours, with a 5 person third-level support team (2 analysts, 3 developers) dedicated to resolving support issues.

- Helpdesk accessible via email or telephone during weekdays and working hours.
- Extended support available outside commission working days for critical issues.

[29] Related to EIF principle 9 "Openness" with regards to software openness defined in a formalised license.

Scalability

The solution's scalability level, including measures that are taken in relation to this. Information on a solution's performance and ability to gracefully scale can be of significant importance depending on the scalability requirements of the overall solution.

Example: A service deployed on cloud infrastructure that offers high on-demand horizontal scalability depending on current needs.

CIPA offers a high level of scalability. The architecture follows a four corner model allowing additional components to be added transparently and does not depend on one specific point of failure, making use of basic internet services, such as DNS lookup.

Security[30]

Any constraints or implications that the solution may bring in relation to security. Such security considerations can include communication aspects, data handling or involved processes.

Example: A data exchange component that supports encryption at the communication channel level but also on exchanged payloads.

Service providers require certificates to access the network. Certificates are also used for message-level encryption.

Authorisation of document exchange between partners is achieved by means of configuration (Pmodes for AS4 and specific AS2 configuration) defining the expected bindings between senders and receivers.

Testability

Test resources that are available to test integration or interoperability of the component or service in the overall envisaged solution. Such resources relate to available test suites but also the availability of test instances to facilitate integration and interoperability testing during the overall solution's elaboration process.

Example: A test instance of a service that is publicly available for interoperability testing purposes.

CIPA e-Delivery offers through Joinup a testing suite consisting of a preconfigured Apache Tomcat instance and a preconfigured VirtualBox appliance that can act out-of-the-box as a receiving party for the Tomcat distribution. SOAP UI test suites are also provided to perform basic connectivity tests for AS2 and AS4 message exchange.

Localisation and language[31]

Capabilities of the solution that enable its localisation. Localisation relates not only to the language used in reports and user interfaces but also to currencies, date formats and metrics. In addition, the possibility to access the solution's documentation in multiple languages enhances its accessibility for developers and potentially also for end users.

Example: A component offering localisation of produced reports based on XML configuration files per locale.

Does not apply as CIPA e-Delivery only offers a machine to machine interface for document exchange. Payload that could contain localisable elements is not the concern of CIPA.

Inclusion and accessibility[32]

Capabilities of the solution that allow the non-discrimination of its users. This can relate to measures taken in terms of design, information content and delivery to support impaired people or support user choice through multichannel delivery.

Example: A solution's user interface that allows audio user prompts in support of visually impaired users.

No specific measures to support accessibility exist although the nature of CIPA e-Delivery as a payload-agnostic data exchange solution make this not relevant.

[30] Related to EIF principle 4 "Security and privacy" with regards to the solution's security constraints or requirements.

[31] Related to EIF principle 5 "Multilingualism" with regards to its realisation within the solution.

Cost considerations

Cost information that can serve as input to the Total Cost of Ownership (TCO) analysis for the solution. A solution's TCO is defined as the combination of direct and indirect costs that its reuse entails. Indirect costs such as the need to adapt existing processes is specific per case, but certain information, especially with regards to the direct cost calculation, is common. Such costs range from ongoing licensing fees, initial setup costs for required hardware, but also to an estimated effort to train new users in the solution's use. Note that for solutions reusable both as services and components where cost considerations may differ per case, information is provided to address all reuse types.

Example: A software component that contains a licenced library resulting in an ongoing fee for its (re)use.

- No licensing costs for the software itself (CIPA e-Delivery and used libraries).

II. REUSABILITY ASPECTS FOR SOLUTIONS AS SOFTWARE COMPONENTS

These represent further aspects to consider for a solution when evaluating its reuse as a software component. This component may be reused as a standalone installation within an overall solution, or possibly as a library forming an integral part of a software system. In addition, software components can be reused in an as-is manner or potentially modified to cater for customisations needed for the specific case.

Technology stack

The set of platforms and technologies used to develop the component. The platform (e.g. Java Enterprise Edition or .Net) defines the overall technological approach and is complemented by the individual libraries, frameworks and tools, including their version information, used to address individual technical concerns. This information is important to determine the compatibility of the component towards the overall solution and can serve to identify potential conflicts with regards to an organisation's technological guidelines.

Example: Use of Oracle 11g RDBMS for persistence, JEE7 as the overall platform, Hibernate 3 as an Object-Relational-Mapping (ORM) framework.

CIPA e-Delivery is a Java Enterprise Edition (JEE) solution built using the Spring framework. Being a JEE application, CIPA can be deployed on any operating system that supports Java and any JEE server (the server provided as part of the test package is Apache Tomcat). In terms of persistence, CIPA has been tested with MySQL and Oracle databases.

Architectural considerations

Information on the architecture of the component and any expectations that the component would have of the overall solution. Such considerations could relate to the configurability of the component, the way it expects to be interacted with, exposed APIs etc. Such information is important to determine whether from an architectural perspective the component can be seamlessly integrated into the overall solution.

Example: A component following an event-driven design that integrates with other components using the asynchronous exchange of messages.

Synchronous document exchange is currently not supported (only asynchronous through SOAP web services).

New nodes can be added transparently to the e-Delivery network following CIPA's four corner model, where end entities (corners one and four) exchange messages via gateway intermediaries (corners two and three). The infrastructure only standardises communication between these intermediaries whereas communication between gateways and end entities may use different solutions. This architecture model also makes it possible to have no single point of failure.

[32] Related to EIF principle 3 "Inclusion and accessibility" on its realisation within the solution (if applicable).

Hosting needs

Component requirements in terms of infrastructure resources. These range from capacity needs on storage, processing and networking, to restrictions relating to operating system configuration and virtualisation technologies. The hosting-related needs of a component could determine whether it complies with an organisation's hosting constraints and whether it is possible to fulfil specific capacity requirements.

Example: A component provided in the form of a VMWare image for reuse.

- The hosting of the SMP and access points requires a JEE application server.
- The SML relies on DNS technology and thus it's hosting requires the connection to a Dynamic DNS server. Implemeters of the SML must own a Domain on the internet in which they can create entries.

Extensibility[33]

Information regarding the possibility to extend of modify the component to suit a specific solution's needs. This goes past using the component as a black box by adapting it at the source code level. Such adaptations are typically subject to the component's licence, but also are facilitated by the quality of the code and the presence of rich inline code commenting.

Example: A component provided as an open source reference implementation of a standard that is meant for reuse and customisation.

The source code of CIPA e-Delivery is fully extensible as it is based on popular open source libraries (e.g. Spring framework) and is itself offered as free and open source through its EUPL 1.1 licence.

III. REUSABILITY ASPECTS FOR SOLUTIONS AS SERVICES

These aspects represent points to consider when the solution is to be reused as a service. In this case the service is operated by a third party and is consumed remotely through appropriate service calls.

Availability

The availability level of the service. A service's availability characteristics are important to consider with respect to the availability of the overall solution and could be prohibitive for reuse if not at a certain level. The service's availability could be formalised through an SLA or at least documented as part of its non-functional requirements. However, even if this is not recorded in a formal manner, an indication of the service's availability would provide useful insight for its reuse.

Example: A service guaranteeing through its SLA 99% availability.

In the context of OpenPEPPOL, the availability of the centrally operated (by DIGIT B4) SML component is defined as 99% (non-working hours and maintenance time included). The availablity level as well as additional service delivery details are defined in an MoU between DIGIT B4 and OpenPEPPOL AISBL.

Integration

Technical considerations that relate to accessing the service. These can range from network connectivity restrictions to security-related prerequisites.

Example: A service only accessible internally within the European Commission following ECAS authentication.

Integration with the CIPA Access Point is through use of SOAP web services. In order to access the e-Delivery network through new CIPA Access Point instances, participants need to first request PKI certificates or mutually exchange certificates.

[33] Related to EIF principle 9 "Openness" with regards to the software components source code.

Service agreement

A reference to a formalised service agreement or a description of the considerations and prerequisites regarding the use of the service. For a service meant to be used by end-users this could restrict its use to specific user groups whereas for a service consumed by another system it could entail mandatory interoperability tests as a prerequisite step.

Example: A service that is usable only by EU citizens.

Access to the e-Delivery network is possible for participants once they have signed relevant service agreements or based on bilateral agreements.

In the context of e-Procurement this is organised through the international non-profit association OpenPEPPOL.

Data privacy[34]

Considerations regarding the handling of data within the .service that relate to data privacy. This relates to the service's data policies, referring where possible to specific applied legal texts, but also to specific measures taken within the solution to ensure that data privacy requirements are upheld.

Example: A service that treats all received data as fully public.

Data protection obligations should be taken into account prior to use of CIPA. Consideration should be given to the fact that the components are licenced as open source which could have legal implications.

A data protection study has been carried out and appropriate

Data location and federation

Considerations regarding any datasets that the service maintains based on the data it handles. This could relate both to the location of stored data (if any) and to how data if federated into a central dataset (if applicable). Such considerations could be important when considering reuse for solutions that have specific requirements on how and where data is recorded.

Example: A service that is guaranteed to store handled data only in DIGIT's EU Institutions (EUI) datacentre.

Not applicable as the only centrally operated component is the SML that records DNS lookup information of participants.

[34] Related to EIF principle 4 "Security and privacy" with regards to the privacy of data.