

Towards a Cross-Border Reference Architecture for the Once-Only Principle in Europe: an enterprise modelling approach

Jaak Tepandi¹, Eric Grandry^{2,3}, Sander Fieten⁴, Carmen Rotuna⁵, Giovanni Paolo Sellitto⁶, Dimitris Zeginis^{7,8}, Dirk Draheim¹, Gunnar Piho¹, Eftimios Tambouris^{7,8}, and Konstantinos Tarabanis^{7,8}

¹ Information Systems Group, Tallinn University of Technology, Estonia
<http://is.taltech.ee>

{[jaak.tepandi](mailto:jaak.tepandi@taltech.ee), [dirk.draheim](mailto:dirk.draheim@taltech.ee), [gunnar.piho](mailto:gunnar.piho@taltech.ee)}@taltech.ee

² Luxembourg Institute of Science and Technology

³ Ministry of Mobility and Public Works, Luxembourg

eric.grandry@tr.etat.lu

⁴ Holodeck B2B

⁵ ICI BUCHAREST

⁶ ANAC, IT

⁷ Centre for Research & Technology Hellas, Greece

⁸ University of Macedonia, Greece

{[zeginis](mailto:zeginis@uom.edu.gr), [tambouris](mailto:tambouris@uom.edu.gr), [kat](mailto:kat@uom.edu.gr)}@uom.edu.gr

Abstract. The *Once-Only Principle* states that citizens and businesses provide data only once in contact with public administrations. So far, many European countries have started to implement the Once-Only Principle at national level, but its cross-border implementation is still fragmented and limited. This paper presents the development of a Reference Architecture for the Once-Only Principle in Europe. The case study, stemming from the EU-funded Once-Only Principle project (TOOP) highlights the challenges faced by the architecture team when developing the Reference Architecture that tackles the Once-Only Principle across different countries and policy domains. The architecture is not built from scratch, but re-uses and enhances already available building blocks in order to seamlessly preserve interoperability and to comply with regulations and existing technical standards, leaving at the same time enough space for vendors and open source developers to propose their compliant solutions, whatever is their business model.

Keywords: Once-Only Principle · EU Single Digital Gateway · Reference Architecture · TOOP · EU Building Blocks · TOOPRA

1 Introduction

The Once-Only Principle (OOP) states that “public administrations should ensure that citizens and businesses supply the same information only once to a

public administration. Public administration offices take action if permitted to internally re-use this data, in due respect of data protection rules, so that no additional burden falls on citizens and businesses” [5]. So far, many European countries have started to implement the OOP at national level, but its cross-border implementation is still limited. This paper presents the development of a Reference Architecture for the OOP in Europe, as well as lessons learned in this process.

Development of reference architectures is currently often focused on some specific concern, which acts both as a reference source for Business Requirements and a canvas to compose existing Solution Building Blocks. This is increasingly becoming part of a wider evidence based policy-making cycle⁹. Pilot socio-technical systems are used to prove the viability and the feasibility of some policy objective that requires the usage of Information and Communication Technologies (ICT) platforms in order to be attained. In this context, the development of Large Scale Pilot systems along with a Reference Architecture can support decision-makers with quantitative results about the effectiveness of the devised regulations. Often the Reference Architecture itself later becomes part of a Regulation Act, as long as it does not hamper the market restraining the implementation possibilities to a single solution.

The OOP Reference Architecture (TOOPRA) development cycle within the EU-funded Once-Only Principle project (TOOP¹⁰) is representative of such a process, as the overarching policy objective is part of a wider European strategy, aimed at fostering competitiveness reducing duplicated work for the citizens, businesses, and government officers. This objective can only be attained through the use of an ICT platform. The architecture development cycle in this case does not start from scratch, but from a gap-analysis activity based on the assessment of the existing relevant building blocks (Core Components) aimed at deciding whether and how they can be composed or evolved to attain the OOP while complying with other legal and technological constraints. Moreover, architecture had not to be a bottleneck for the developers of OOP Pilot Solutions. Thus the development of TOOPRA proceeded with a continuous consultation with legal and domain experts working on concrete piloting use cases and with the developers taking care of the governance of the Building Blocks.

To complement the experience, during the development of the architecture (2017-2019), a new EU Single Digital Gateway Regulation¹¹ (SDGR) was issued by the European Commission that mandates the cross-border application of the OOP in a number of procedures and foresees a technical system based on OOP.

⁹ Cf. ”Better Regulation” and Regulatory Impact Analysis (RIA), which is a systemic approach to critically assessing the positive and negative effects of proposed and existing regulations and non-regulatory alternatives.

¹⁰ <http://toop.eu/>

¹¹ Regulation (EU) 2018/1724 of the European Parliament and of the Council of 2 October 2018 establishing a single digital gateway to provide access to information, to procedures and to assistance and problem-solving services and amending Regulation (EU) No 1024/2012 Text with EEA relevance, <https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1549716594539&uri=CELEX:32018R1724>

Furthermore, the European Interoperability Reference Architecture (EIRA), the Open Group Architecture Framework and the Archimate Modeling Language underwent significant changes. All of the above mentioned items had an impact on the TOOPRA and on the project, leading to a continuous shifting of the objectives backed by the availability of new tools and frameworks to support the work of the Enterprise Architects.

The rest of this paper presents how the OOP Reference Architecture was designed to comply with a new regulation and integrate standard building blocks, explains the usage of various modelling techniques and tools, analyses layers and concepts in each layer needed to customize enterprise modelling to the needs of OOP architecture, and discusses benefits, difficulties and lessons learned using the enterprise modelling approach in OOP.

2 Motivation and approach

2.1 Why designing a Reference Architecture?

The general concepts of software, systems and enterprise architecture have been widely studied, have a long history and diverse content [8, 2, 15].

In the case of the TOOP, a reference architecture is intended as a multi-stakeholder frame of reference with the specific goal of supporting OOP in a cross-domain and cross-border environment. Cross-domain means that it is intended to facilitate the creation of systems that connect multiple independent organizations having different field of business, managing heterogeneous ICT platforms and subject to different policy domains. Cross-border means that the actors come from different countries and therefore they are subject to different legal and regulatory framework - translations will be needed not only for the language, but also at the level of the semantics of the data and documents that they exchange. The reference architecture [2] must address the main OOP concern by providing a common and standardized definition of the problem domain and of the solution blueprint, without hampering interoperability between the different organizations and without compromising security, privacy and flexibility, which are qualities that come from the architectural solutions adopted by the constituent Building Blocks [10].

The specific concern of TOOPRA is the Once Only Principle: motivation elements include relevant material coming from regulations and legislation, but the approach to interoperability shall encompass all of the architecture layers from Strategy to Technical Implementation, since the TOOPRA is intended as a multi-domain and cross-border tool to develop OOP compliant systems. At the very early phase of the project, various practical aspects of OOP, such as its expected benefits, drivers, barriers, citizen perceptions, support, initiatives, legislative measures, costs and benefits, and perspectives, have been analysed in [1, 9, 16]. There are however no defined architecture of OOP systems. The current paper aims to provide this, extending and generalizing the TOOP deliverables [13, 7], as well as taking into account the European Interoperability Framework

(EIF) [4], the European Interoperability Reference Architecture [6], the Connecting Europe Facility Digital Service Infrastructures, and the ISA2 Study on functional, technical and semantic interoperability requirements for the Single Digital Gateway (SDG) implementation^{12 13}.

The attainment of the OOP in an EU-wide and cross-border mode entails a re-adjustment and the re-engineering of the administrative processes and the development of an infrastructure that purposely supports the interoperability between Administrations at various levels [LOST - Legal, Organizational, Semantic, Technical] and the exchange of information in the respect of data protection rules. Part of this infrastructure is an architecture for OOP related projects presented in the current paper.

TOOPRA improves understanding of how to achieve interoperability between domains with diverse legislation, organisations, applications, technology. It provides architecture for large-scale cross-border OOP implementation, uses EU Building Blocks to implement OOP, and contributes to implementation guidelines for SDGR. It provides support for developers of OOP projects and is based on the Connecting Europe Facility (CEF) Digital Service Infrastructures (DSIs), on the Building Blocks consolidated by the e-SENS project, and in justified cases, on new building blocks.

Organisations can benefit from TOOPRA, as it enables to select solutions without reference to a vendor - the architectural model can be included in the technical specifications of a call to reduce the risk of vendor lock in. Member States can benefit from the reference architecture, since it makes the development process of OOP-compliant applications more efficient, contributing to the implementing act of the SDGR.

2.2 Why applying enterprise modelling?

Selecting modelling techniques and tools for TOOPRA is based on the main drivers and decisions of the project:

- The Once-Only Principle reduces the workload of system users who need to provide only minimum information to receive a specific public service.
- The legal environment of the OOP, in particular the SDGR and Member State regulations, represents the legal basis that should be considered in the implementation.
- The existing frameworks, standards, and building blocks provided by CEF, e-SENS, and other initiatives, should be re-used to minimize development effort and improve interoperability. Examples of frameworks that need to be taken into account are the European Interoperability Framework [4] and the European Interoperability Reference Architecture [6]. Examples of building blocks are the CEF DSIs, including the CEF eDelivery, eID, and eSignature.

¹² <https://joinup.ec.europa.eu/community/once-only-principle/home>

¹³ <https://www.scoop4c.eu/>

- TOOPRA is developed in interaction with the three TOOP pilot projects. These pilot projects develop and implement the TOOP Common Pilot Solution Architecture which is a specific instantiation of TOOPRA.
- The architecture must take into account and be consistent with the user requirements from the EU Member States elicited during development of the TOOP pilots.

This list demonstrates the need for the following modelling techniques and tools, among others:

- Describing the architecture from the viewpoints of multiple users and their concerns, as well as showing the conflicts and trade-offs made.
- Managing continuous changes in legislation and technology.
- Providing interoperability with the existing frameworks, standards, and building blocks, to minimize development effort.
- Enabling cooperation and mutual support between development of the TOOPRA and the TOOP pilot projects.
- Facilitating consistent user requirements engineering from different EU Member States.

These techniques and tools are best covered by enterprise modelling. A comparison of enterprise architecture frameworks is given in [15]. From the list of available frameworks, TOGAF was selected because it satisfies all the needs highlighted above and is openly accessible. TOGAF 9.2 considers an "enterprise" to be any collection of organizations that have common goals. As the architecture attempts to connect and simplify different governmental services from different Member States, the enterprise in consideration is a group of weakly linked independent governmental entities that cooperate to achieve common goals defined by OOP principles.

ArchiMate¹⁴ was selected as the architecture description language (ADL), since it supports multiple views' modelling and, moreover, it is the modelling language adopted in the development of the EIRA.

3 Enterprise Modelling of the OOP Architecture

Development of the OOP architecture starts from the Once-Only Principle and the selected enterprise modelling framework (TOGAF), takes into account legal and regulatory requirements, makes use of the existing building blocks, and interacts with OOP piloting.

Based on these foundations, the business, information system and technology layers of the architecture were designed. Cross-cutting concerns include security and trust, as well as semantics. TOOPRA is aligned with EIRA, which has the primary objective to facilitate interoperability of public services while reusing existing Building Blocks (BB) at the EU level. The Building Blocks are basic

¹⁴ <http://pubs.opengroup.org/architecture/archimate3-doc/>

digital service infrastructures - key enablers to be reused in more complex digital services. They are described as a set of technical specifications and standards that the implementation of a BB has to comply with.

3.1 Business Architecture

The business layer of TOOPRA represents a coherent set of business concepts: capabilities, end-to-end value delivery, information, as well as organizational structure and the relationships among the business elements, strategies, policies, initiatives, and stakeholders (the term "business" is here understood in a wide sense, involving also public sector organisations) [14]. TOOPRA Business Architecture is concerned with the description of the business operations conducted by the business actors involved in OOP, and describes core business assets such as business roles relevant for TOOP, business data exchanged between TOOP participants, business services provided by each of the roles to meet the business goals and business processes depicting the interactions amongst roles.

The process model describes the end-to-end scenario of executing OOP in the context of a public service delivery (see Figure 1): a Competent Authority delivers a public service to a legal entity. In its role of Data Consumer, that Competent Authority executes the OOP, and retrieves required information (Evidence according to SDGR) from another Competent Authority in its role of Data Provider.

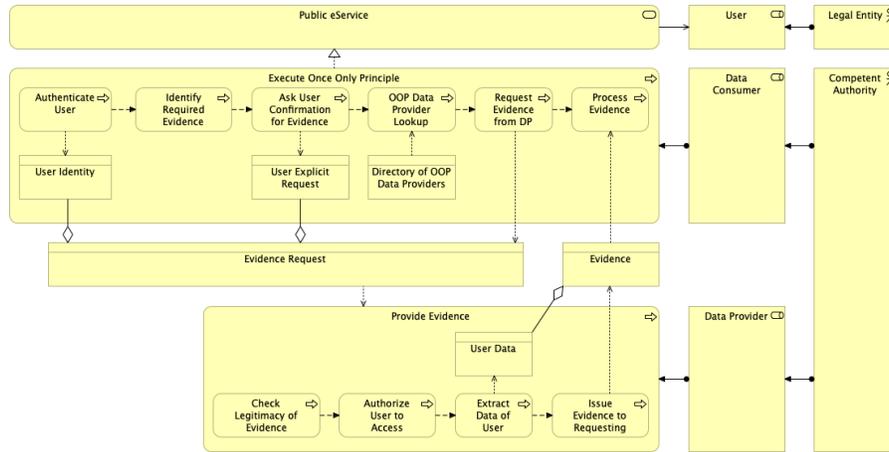


Fig. 1. TOOPRA Business Architecture Process Model

Architecturally Significant Requirements (ASR) and Architecture Principles guided the design of the Business Architecture, which realizes the business requirements expressed in the process model. TOOPRA Business Architecture

addresses two main business concerns: the Business Interactions, which show the collaboration between the actors involved in OOP, and the Capability Map, which specifies the responsibilities of each actor participating in a cross-border Evidence exchange.

The Business Interactions view identifies five main interactions: Evidence Request Exchange, Evidence exchange, Competent Authority Information Exchange, Service Offering Exchange and Identity Exchange. The identification of Business Information exchanges between entities in TOOP business network enables the stakeholders to recognize the interoperability related aspects and to address them.

The Capability Map view enables the participants to accurately identify the business capabilities required for the role they intend to play in TOOP business network. Four roles were identified: Data Consumer, Data provider, Evidence Service Broker and Identity provider. The entity in an Identity Provider role operates a business service to provide identity information of the user to the data consumer. The entity in an Evidence Service Broker role operates a business service to provide functionality to competent authorities to update their meta-data on the service offered by the authorities.

3.2 IS Architecture

The Information Systems Architecture focuses on designing Data and Application Architectures and describes how the Business Architecture is realized through Information Systems. A first step in designing the IS Architecture was a thorough assessment of the available BBs to understand what potential resources are available for OOP applications development. CEF eDelivery, eSignature, eIDAS eID and PePPOL Directory were identified as relevant technical capabilities in the context of TOOPRA. The second step consisted in assessing the technical requirements from pilots and mapping them to existing BBs. Therefore, the ASRs were analyzed, the capabilities needed to fulfill these requirements were identified and the BBs that provide the capabilities were mapped to ASRs. As a result, the IS Architecture design mixes a top-down approach, starting from ASRs and Business Architecture, together with a bottom-up approach by injecting the common pilots solution architecture.

The IS Architecture description is composed of 2 layers: the existing generic BBs and a TOOP specific layer, leveraging the generic BBs to realize the TOOP functionalities. The Data Consumer components of the IS architecture are provided on Figure 2, where a specific colour code is used to separate the TOOP specific elements (usual Archimate blue of application layer) from the generic BBs reused from the existing catalogue of solutions (coloured in purple). The architecture describes the TOOP Connector and the eID Component as the major components concerned with the realization of the business operations. The IS Architecture describes also the technical specifications and standards prescribing the implementation of the components.

The TOOP Connector is a complex architecture component that enables the cross-border exchange of Evidences. This core component fulfills some specific

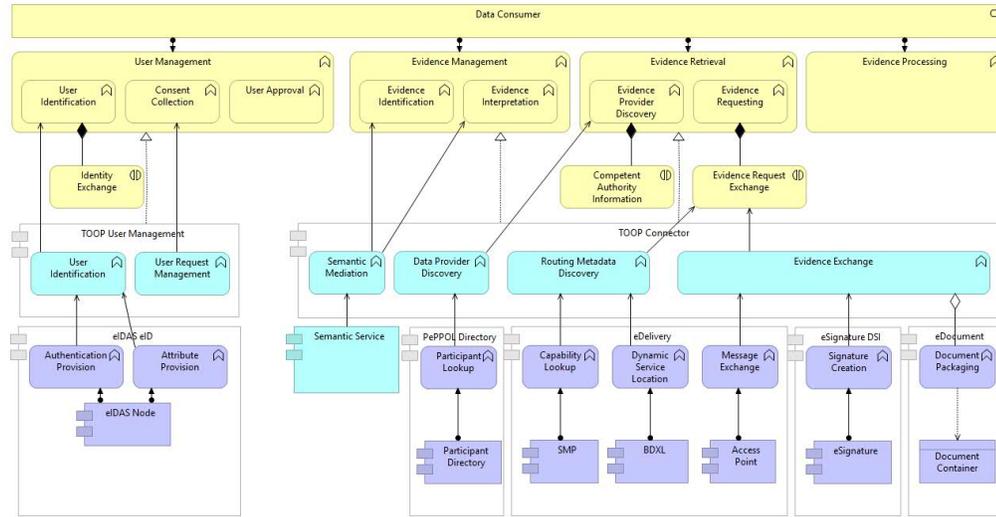


Fig. 2. TOOPRA Information System Architecture: Data Consumer

functions such as: Semantic Mediation, which transforms the data input into a TOOP message and also performs the reverse process, Data Provider Discovery handling dynamic participant lookup, Routing Metadata Discovery which enables the discovery of the endpoint address used for delivery, and Evidence Exchange handling cross-border data exchange. TOOP Connector relies on BBs that were already used in other projects and new components that were developed to meet TOOP business requirements.

The aim of the eID component (identification/authentication) is to authenticate the user/Data Subject over the eIDAS network and to establish trust between the Public Authorities that are exchanging Evidence. In agreement with eIDAS recommendations, a minimum data set for eIDAS natural person identification attributes must be provided.

3.3 Technology Architecture

The Technology layer of the architecture provides technology components in a way that support the deployment of the logical components described in the IS Architecture. The components can be deployed either at central European level or at Member State level. The latter can be: i) components deployed at the national level (i.e. shared between competent authorities) by an authority or a private business entity and ii) components deployed at the competent authority level. Due to the many different types of deployment, there are several topologies beneficial in different contexts.

Within TOOP project a topology that enables Member States to share common services is identified as beneficial. In this topology (Fig. 3) the Semantic

Server, the TOOP Directory Server and the Business Document Exchange Network Location (BDXL) implementations (BDXL Server and DNS Server) are implemented as central European components; the eIDAS Node, the SMP and Access Point are deployed at national level. The Data Consumer Competent Authority operates and maintains its own TOOP Connector. This topology simplifies business organization for Competent Authorities, since they do not need to maintain the SMP and Access Point. Additional effort is required at the Member State level, but if many different Competent Authorities use the same deployed components, then this variation is beneficial. A similar topology is used by the Data Provider Member State and Competent Authority.

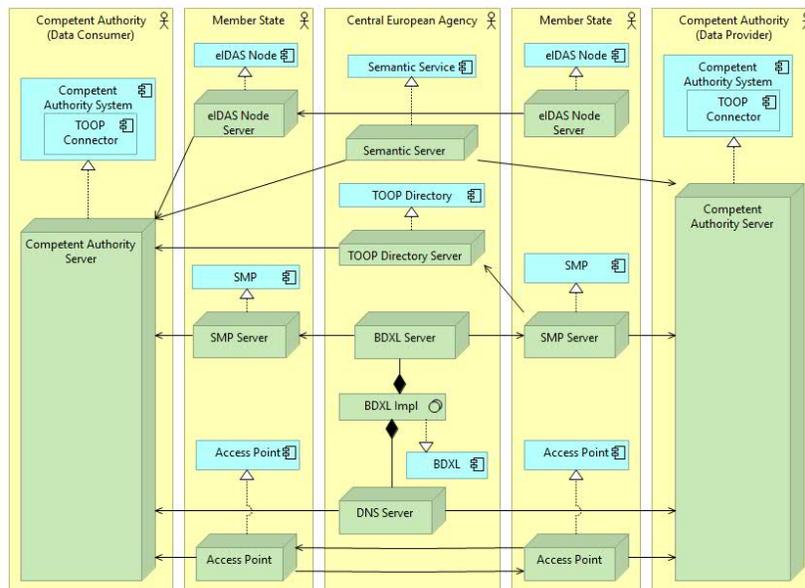


Fig. 3. Technology Architecture Variation 1

Although the above variation is beneficial within the context of the TOOP project, other topologies could also be envisaged under different context, such as: i) the Data Consumer Competent Authority deploys locally the SMP, Access Point and TOOP Connector - for example, in the initial introduction of the architecture, where the Member State infrastructure is lacking and some advanced Competent Authorities want to participate, or ii) a national OOP Layer Provider offers the components needed to on-board to the architecture (TOOP Connector, SMP and Access Point) with an additional cost.

Solutions accepted for TOOPRA were often results of thorough choices made between several options. The length of the article does not allow for a detailed presentation of the choices and decisions made, but here are two examples.

As the first example, it would have been in principle possible to present TOOPRA just as a collection of standards and interfaces specifying the central European components of the OOP infrastructure and leave implementation of the Member State components to the Member States. Although this approach has its advantages, it was not selected for the following reasons: valuable experience gained from the pilot implementations would have been lost; usage of common solutions on MS level allows to reduce costs; and certain Building Blocks for the MS level are already available.

The second example concerns a variation introduced by one of the TOOP pilots. In this pilot, there is no need for the Data Provider discovery, as the Data Consumer already knows which Data Provider has the requested data. As both variations are valid in specific situations, they both have been included in TOOPRA.

4 Evaluation of the Architecture

This section provides an evaluation of the proposed architecture in terms of feasibility and applicability at EU level, but also demonstrating compliance between members states at a cross-border setting. Numerous approaches for architecture evaluation have been proposed [11]. For the TOOPRA evaluation, two approaches are adopted: i) Scenario based evaluation [3] - the architecture is implemented and evaluated within the TOOP project through five pilot scenarios in real conditions proving its feasibility at the EU level and ii) Provision of a model to deploy an OOP solution architecture based on TOOPRA. Existence of such a model indicates feasibility of using TOOPRA on a wide scale and provides an inexpensive evaluation in the spirit of the ideas of the Tiny Architectural Review Approach (TARA) technique [17].

The evaluation presented in this section provided valuable feedback for the improvement of the architecture. Final evaluation will be done with the intended end users of OOP services and the stakeholders of the reference architecture. This evaluation will take time and need significant resources.

4.1 Scenario based evaluation

Five pilot scenarios have been tested that focus on the exchange of company data in the context of cross-border services (e.g. a foreign citizen wants to expand his business in a country abroad, and his application form is automatically pre-filled with data from his national business register). The scenarios have been tested in real setting i.e. by EU public services and public authorities serving either as data consumers or data providers. The tested scenarios are:

- *eProcurement*: demonstrates the automatic retrieval of necessary evidences during an eTendering procedure.
- *Licenses & Permissions*: demonstrates the registration of cross-border services. The required evidences are provided by the competent authority of the origin country to the registration service of the destination country.

- *Company data & mandates*: demonstrates the way a service from a foreign country can derive data and mandate information about a company directly by the business registry of the home country of the company.
- *Business Register Data Provision*: demonstrates the exchange of information between the home business register of a company and a foreign public authority in order to authorize access to a service.
- *Business Register Interconnection*: demonstrates the exchange of information between business registers of different countries for the collaborative administration of a company and its foreign branches.

In order to evaluate the scenarios, TOOPRA has been implemented and used by the EU members states and public administrations. Specifically, three types of components have been implemented: i) central European components, namely the Semantic Service, the TOOP Directory Service and the BDXL service, ii) member state components, namely the Service Metadata Publisher service and the eIDAS node and iii) competent authority components, namely TOOP connector and TOOP interface that both facilitate the connection of the competent authority's back-end system with TOOP solution.

The implementation has been tested through the “TOOP Playground” - a virtual environment implemented as a Ganeti VM Cluster that emulates a virtual Europe (with multiple virtual member states) for a more realistic deployment environment. In order to facilitate testing, TOOP commander has been created that is a simple Java command line app which creates data consumer and data provider endpoints for receiving messages from the TOOP connector. It also provides means for sending requests from command line between the data consumer and the data provider.

Two types of testing activities have been performed: i) testing within the member states in order to ensure that a certain member state implementation has been installed and integrated appropriately and ii) testing across member states in order to ensure that different member state implementations communicate as expected. The later included 20 different end-to-end connectivity tests that cover all the five scenarios presented at the beginning of this section (e.g. receive a valid and complete result according to the eID values entered, or receive an appropriate error message). The connectivity tests were performed between the systems of 8 different member states (Greece, Romania, Italy, Austria, Slovakia, Poland, Sweden and Norway). Some of the member states act as data consumers (Greece, Poland, Norway) other as data providers (Italy, Romania, Slovakia) and other both (Sweden, Austria). The end-to-end tests were performed at 10 organized online “Connectathons” that test the connection between a data consumer in one member state and a data provider in another member state. The success rate of the end-to-end test reached 77% (77% of the 20 test succeeded between all the different combinations of member states).

4.2 From TOOPRA to OOP Solution Architecture

In order to successfully deploy a specific OOP project based on TOOPRA, the stakeholders need to fulfill certain organisational, semantic, and technical pre-

conditions. In particular, the reference architecture given by TOOPRA must be used to define the solution architecture for the project under development. Due to space limitations the summary below focuses on semantic and technical preconditions.

As for the semantic processing according to TOOPRA, the stakeholders must have their data described in machine-readable formats together with metadata. The data must be accessible, standard vocabularies must be available. It should be also possible to implement necessary semantic mapping services.

A technical precondition is that the infrastructure of all stakeholders must be sufficient to initiate the project. Especially, the data and public services required by TOOPRA must exist, and data needs to be available, discoverable, and accessible. The stakeholder security and trust levels, data quality, and other characteristics must be adequate.

During design, it is necessary to decide on a deployment topology as indicated in TOOPRA - whether it is Member State shared services, local deployment, or a national OOP layer. The topology determines to a great extent the components to be deployed on a national and Competent Authority level.

During the component selection, deployment, and operation phases a decision is needed whether to use open source software, purchase proprietary software, or develop a custom solution. The stakeholders must provide integration with data providers and infrastructure components (eg, the Access Points). In general, activities on these phases are of a more generic type and less specific to TOOPRA.

5 Lessons Learnt and Discussion

The enterprise modelling approach has given many benefits, but also entailed issues and difficulties to be encountered. The subsequent discussion highlights some of the lessons learnt.

In the first version of the architecture, the component-based software engineering methodology [12] was followed based on the assumption that the existing Building Blocks were sufficient to develop the solution. In the further versions of the architecture, the needs for designing more detailed content, demonstrating compliance, as well as presenting views from various stakeholders became evident. Therefore enterprise modelling approach was switched to. Association with the TOGAF standard has given an overall methodology and framework of views and artefacts to rely upon. Usage of the enterprise modelling approach was not intended as a TOGAF application exercise, but rather as a methodology to improve the overall quality of the result. For this reason, the selected TOGAF phases have been broadly followed in the architecture development process.

One issue encountered has been related to the notion of Enterprise Architecture (EA). Initially EA was proposed in the context of organizations. When the enterprise modelling approach is used for TOOPRA, it may be not clear where is the Enterprise - the OOP landscape includes multiple organizations

and stakeholders. However, the notion of EA has evolved and for TOGAF, an enterprise will often span multiple organizations.

As an architecture description language (ADL), Archimate 3.0 has been used starting from the very first version. There have been discussions on whether it is too stringent for TOOPRA, but these discussions have resulted in conclusion that the benefits of Archimate outweigh its drawbacks. We however faced specific issues related to the very nature of designing a reference architecture.

When defining a reference architecture, we put a specific effort in the standardization and reuse of existing components and technologies. According to TOGAF and as illustrated, we use the concept of building block to capture this unit of standardized architecture component. Archimate however does not directly support the concepts of building block and technological standards: we therefore needed to develop some kind of Archimate dialect, reusing existing modelling elements of the language to support the concepts required in the description of our reference architecture (mainly the application function and the application component). We also used a colour code to visually distinguish the generic building blocks from the specific TOOP components and functions. This solution works locally, but we could think about specializing the language and create additional modelling elements.

By definition, a reference architecture is a blueprint to a (set of) solution architecture. Those TOOPRA instances can also be described with Archimate; however, there is currently no way to relate elements of the solution architecture to elements of the reference architecture (such as realization or instance-of). This problem can obviously be solved by specific Archimate editors, however it would be valuable to define the concept of reference element as part of the language specifications. This could also solve the previous issue, as reusing a building block from a catalogue of solutions could also be seen as introducing elements of another reference architecture.

We used Archimate to describe the usual layers of an enterprise (business, IS, technology). There is however a specific concern to be addressed in the reference architecture, namely the interoperability aspect of TOOP. We introduced a specific modelling pattern in Archimate to actually capture the needs for interoperability at the business level: this allows to easily pinpoint the interaction points amongst the business partners that require interoperability solutions. We then modelled the realization of those interactions through the use of building blocks (including technical standards). The approach is compatible with the EIRA and goes further with the ability to isolate the interoperability concerns in the overall architecture.

Besides the usual views of the enterprise, we also developed specific views to describe cross-cutting concerns, and especially trust and security. The experience here is that insofar as the reference architecture requires specific building blocks to enforce trust and security, Archimate is a suitable tool to model these blocks of the trust and security framework. However, modelling the security architecture based on the ISO/IEC 27000-series of standards involves policies, procedures, guidelines, and associated resources and activities. In our opinion, these are

very detailed and representing them in Archimate would give little additional value.

Experience gained in the development of the architecture, as summarized in Chapter 3, leads to the conclusion that a continuous architecture development approach is necessary, especially when new requirements coming from pilots are frequently incorporated.

One of the lessons learnt was the usefulness of evaluation in providing feedback and stimulating improvements for the architecture development as shown in Section 4.

During our reference architecture design, we identified variations in the way the individual process steps can be executed (e.g., in the discovery of the Data Provider that can supply the required Evidence and the actual Evidence exchange), as well as variations in terms of deployment topology. An Archimate view was designed for each of them. We were however not able to express that they are linked by a variation relationship. Such a relationship might be a useful additional concept in Archimate.

6 Conclusion

This paper presents experience and lessons learnt in designing the Reference Architecture for implementing the Once-Only principle to share data legally, securely, and efficiently. The architecture is designed to comply with a new EU Single Digital Gateway Regulation and to integrate standard building blocks.

The need for applying enterprise modelling techniques and tools stems from the architecture drivers and decisions as well as from the demand for multiple views, concerns from various stakeholders, compliance, reuse of building blocks and standards, etc.

Customization of an enterprise modelling framework to the needs of the current development is presented, together with illustrations and model excerpts. The architecture comprises business, information systems, technology, semantics, security, and trust components.

Evaluation of the architecture is based on tested use cases and on a model deployment an OOP solution architecture based on TOOPRA.

The benefits and difficulties of using the enterprise modelling approach in OOP are discussed, allowing the reader to apply lessons learnt in similar projects.

Plans for the future work include elaboration of the content of the architecture on a more detailed level together with further refinement of special issues such as semantics and security aspects.

Acknowledgements. This work was supported by the European Union Horizon 2020 research and innovation program (The Once-Only Principle Project, grant agreement no 737460). The authors would like to thank the three anonymous reviewers for their suggestions and comments.

References

1. Akkaya, C., Krcmar, H.: Towards the implementation of the eu-wide “once-only principle”: Perceptions of citizens in the dach-region. In: *Electronic Government*. pp. 155–166. Springer International Publishing, Cham (2018)
2. Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., Bone, M.: The concept of reference architectures. *Systems Engineering* **13**(1), 14–27 (2010)
3. Dobrica, L., Ovaska, E.: A survey on software architecture analysis methods. *Software Engineering, IEEE Transactions on* **28**, 638– 653 (08 2002). <https://doi.org/10.1109/TSE.2002.1019479>
4. EC: European Interoperability Framework Implementation Strategy. Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions, no. COM(2017) 134 final:9
5. EC: EU eGovernment Action Plan 2016-2020 - Accelerating the Digital Transformation of Government. Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions 2016 (179):111. (2016)
6. EC: European Interoperability Reference Architecture (EIRA) release v3.0.0 (2019)
7. Grandry, E., Fieten, S., Rotuna, C., Sellitto, G.P., Tepandi, J., Täks, E., Zeginis, D.: Generic Federated OOP Architecture (3rd version). Deliverable D2.3 of the TOOP project. European Commission (2018)
8. ISO/IEC/IEEE: ISO/IEC/IEEE 42010:2011. *Systems and Software Engineering – Architecture Description*. International Organization for Standardization (2011)
9. Krimmer, R., Kalvet, T., Toots, M., Cepilovs, A., Tambouris, E.: Exploring and Demonstrating the Once-Only Principle: A European Perspective. In: *Proceedings of dg.o '17*. pp. 546–551. ACM, New York, NY, USA (2017). <https://doi.org/10.1145/3085228.3085235>
10. Proper, H.A., Lankhorst, M.M.: Enterprise Architecture - Towards essential sense-making. *Enterprise Modelling and Information Systems Architectures* **9**, 5–21 (2014)
11. Shanmugapriya, P., Suresh, R.M.: Software Architecture Evaluation Methods A survey. *International Journal of Computer Applications* **49** (2012)
12. Sommerville, I.: *Software Engineering*. Addison-Wesley Publishing Company, USA, 9th edn. (2010)
13. Tepandi, J., Verhoosel, J., Zeginis, D., Wettergren, G., Dimitriou, J., Rotuna, C., Carabat, C., Albayrak, O., Yilmaz, E., Lampoltshammer, T., Täks, E., Prentza, A., Brandt, P., Kavassalis, P., Leontaridis, L., Streefkerk, J.: Generic Federated OOP Architecture (1st version). Deliverable D2.1 of the TOOP project. European Commission (2017)
14. The Open Group Architecture Forum: TOGAF Version 9.2. The Open Group (2018)
15. Urbaczewski, L., Mrdalj, S.: A comparison of enterprise architecture frameworks. *Issues in Information Systems* **7**(2), 18–23 (2006)
16. Wimmer, M.A., Tambouris, E., Krimmer, R., Gil-Garcia, J.R., Chatfield, A.T.: Once Only Principle: Benefits, Barriers and Next Steps. In: *Proceedings of dg.o '17*. pp. 602–603. ACM, New York, NY, USA (2017). <https://doi.org/10.1145/3085228.3085296>
17. Woods, E.: Industrial Architectural Assessment Using TARA. *Journal of Systems and Software* **85** (06 2011). <https://doi.org/10.1109/WICSA.2011.17>