

Semantic reusable Web Components: A Use case in e-government interoperability

Slavko Žitnik, Karmen Kern Pipan, Miha Jesenko and Dejan Lavbič

Slavko Žitnik, Karmen Kern Pipan, Miha Jesenko and **Dejan Lavbič**. 2022. **Semantic reusable Web Components: A Use case in e-government interoperability**, *Uporabna informatika (UI)*.

Abstract

Advancements in technology and software engineering strive to efficiently build robust technologies and deliver them as fast as possible. It has been shown that both could be achieved by reusing existing implementations, libraries, components, and even frameworks. In the field of public administration, which needs to follow and implement national and EU regulations, it is essential to not only have compatible semantic data structures and processes but also to enable cross-sector and cross-border integrations. In this paper, we present the Digital Europe for All project, more specifically, a use case of a reusable prototype - A web component based on semantic data representation. We show that the component can be easily integrated into an arbitrary Web site and supports arbitrary evidence types, represented using the project's ontologies. Reusing such components enables faster creation of interoperable digital public services and opens new opportunities, better mobility, faster processes, and reduced implementation costs.

Keywords

reusable components, building blocks, digital public services, reusability, software engineering, semantics

1 Introduction

Nowadays, we use many software systems based on large frameworks used by developers to build unifying architectures and easy-to-use graphical interfaces for the end users. Users leverage such software systems to support their business processes, and during that, they work on similar activities – e.g., entering data for multiple employees, showing diverse tables or updating data. Building blocks that enable business processes, for example, to enter the address of an employee or a company, should behave similarly. For software developers to achieve that, they need to define common criteria for their software system to unify them and the users. Building blocks are so-called components, defined as follows: “An individual software component is a software package, a web service, a web resource, or a module that encapsulates a set of related functions (or data)¹.”

Software is rarely built from scratch, but to a great extent, some parts of code can be reused, copied and adapted to fit new requirements. For that reason, software process models were developed to provide guidance and create high-quality software at predictable costs. Most of the new software is thus based on the systematic use of well-defined components designed for various projects. Software engineering researchers defined the need for components already in the 1960s (McIlroy et al. 1968), while the subfield Component-based software engineering (CBSE) more widely developed and provided architectural solutions in the 1990s (Sametinger 1997). During that time, prominent commercial

¹https://en.wikipedia.org/wiki/Component-based_software_engineering (Accessed: July 29, 2022)

computing infrastructure vendors followed mentioned ideas and presented their solutions for their programming frameworks, such as ASP.NET or Java (Kozaczynski and Booch 1998).

Some researchers argue that focusing on independent components for modern software engineering is a mistake. Still, the focus should instead be on frameworks that even enable the existence of components (Wallace 2010). The recent rise of Web programming frameworks showed precisely the opposite. These frameworks define components transpiled (i.e., translation of a source code to another language) into JavaScript, can be used completely independent of specifically used Web technologies and allow for custom parameterized user interface styling (Gackenheim and Paul 2015).

Large data economy systems that we can find in the public sector deal with enormous variety and amounts of data. This data is useless if it cannot be easily and correctly understood, accessed or combined. As software components can achieve interoperability and reuse on a technical level, they cannot solve problems on a semantic level. In that sense, governments pushed the development of open data initiatives where all public data would be described in a machine-readable format. Semantic Web technologies describe the data, provide unifying schemas, and overcome problems of matching data fields only by short textual descriptions (Narducci, Palmonari, and Semeraro 2013).

The technologies mentioned above are the main building blocks of future interoperable systems. For the needs of globalization and cooperation of public administrations between EU member states, it is also crucial to enable semantic cross-sector and cross-border data interchange. Of course, privacy, security, storage, and other aspects of large software ecosystems need to be provided, but they are not the focus of this paper (GovStack 2022).

This paper presents a use case for building a Web-based and semantically enabled reusable prototype - A web component for cross-border public administration interoperability. The rest of the paper is structured as follows. We first review technologies that have already tried to standardize and interconnect different public administrations in the EU. Then we continue with the brief presentation of the Digital Europe for All project.

This article delves into specific use-case dealing with semantic and reusable prototype components as part of the field in the project group for Semantic interoperability solutions, where the authors of this article have been actively involved. We evaluate its adaptability and reusability and provide our view for further development.

2 Related technologies supporting digital public services

The previous section briefly overviewed general ideas for building reusable components in software engineering. A well-defined market segmented by Gartner already exists, and it is predicted that by 2024, 65% of application development will be low code. In addition, recently, low-code and no-code platforms are becoming popular, leveraging reusable components. In this section, we focus on the state of provisioning the same set of functionalities to public administration.

Ideas for building reusable components in the public administration domain followed the development of CBSE in the 1990s (Castano, Antonellis, and Pernici 1995). At that time, data was more tightly coupled with application logic, and therefore first approaches focused on identifying similar subschemas in Entity-Relational diagrams. Based on those, the development of common components was proposed. Also, most of the work later focused on model-driven development for digital public services. There have been a lot of different models proposed by the UK, the EU and the USA to provide interoperability on various levels (Peristeras, Loutas, and Tarabanis 2008; Peristeras, Tarabanis, and Goudos 2009).

The United Nations member states defined 17 sustainable development goals representing the 2030 Agenda to connect all countries in a global partnership, improve health and education, reduce inequality, and spur economic growth. For the nations to even enable actions, their systems supporting digital public services must be interlinked. Apart from country-specific initiatives, multi-stakeholder alliances have also been formed that try to provide open-source software, models, content, software architectures and standards – e.g., Digital Public Goods Alliance, GovStack or eGovernments Foundation.

One of the key players in enabling and supporting cross-border and cross-sector interoperability between countries is the EU. It defines standards, connects multiple countries or stakeholders, and funds programmes and projects. Under the European Commission (EC) umbrella, several events are

organized (e.g., SEMIC²) to promote and share new knowledge. EC also leads some projects (e.g., eSENS³, TOOP⁴) to address common topics or provide general frameworks that multiple stakeholders in member states can reuse (e.g., the DE4A project⁵, SDG⁶). In the last years, CEF Digital⁷, ISA, ISA2⁸ and Interoperable Europe⁹ have delivered many valuable building blocks and solutions for public administrations to build interoperability at all levels, following European Interoperability Framework¹⁰.

In this regard, DE4A aims to establish a culture of co-creation, transparency, accountability and trustworthiness. Citizen and business-oriented pilots shall highlight chosen aspects of the technology ecosystem available for the Single Digital Gateway Regulation (SDGR¹¹) implementation at the European and Member State level, prove their technical viability and gauge the performance and degree to which non-functional requirements can be accommodated. We believe experiences, good practices, and information gained in the DE4A project will be of utmost importance for the further development of cross-border digital public services in member states and Slovenia.

3 Digital public services and cross-border interoperability

This section presents the details of the Digital Europe for All (DE4A) project, specifically related to the component we will more technically describe in the second part of the section. The DE4A project shows a possible path to full interoperability between different cross-border digital public services. The DE4A project started with the main building blocks supporting the SDGR's Once-Only Technical System (OOTS) and offers a way to interconnect cross-border public services. Figure 1 shows the stages of building such a platform.

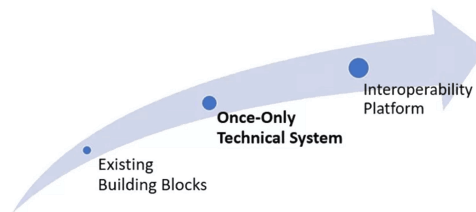


Figure 1: Stages towards a fully interoperable platform (source: <https://www.de4a.eu>, accessed July 29, 2022)

3.1 The DE4A project

The Digital Europe For All (DE4A) project builds on prior work toward EU Digital Single Market to realize the Once-Only Principle. DE4A effectively puts forward a new Member State-driven large-scale pilot aimed at compliance with Single Digital Gateway and aligned with EU eGovernment Action Plan 2016-2020¹², Tallinn Declaration and EIF Implementation Strategy. Its overarching goal is to reinforce trust in public institutions and to unleash multiple measurable positive impacts in terms of efficiency gains and reduction of current administrative burden and costs, rooted in a Toolkit for extended semantic interoperability and in the secure, privacy-preserving and trustworthy realization of fundamental once-only, relevant-only and digital by default principles, through state-of-the-art,

²<https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/semic-conference> (Accessed: July 13, 2022)

³<http://www.esens.eu/> (Accessed: October 11, 2022)

⁴<https://www.toop.eu/> (Accessed: October 11, 2022)

⁵<https://www.de4a.eu> (Accessed: July 13, 2022)

⁶https://single-market-economy.ec.europa.eu/single-digital-gateway_en (Accessed: October 12, 2022)

⁷<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/CEF+Digital+Home> (Accessed: July 14, 2022)

⁸https://ec.europa.eu/isa2/home_en/ (Accessed: July 14, 2022)

⁹<https://joinup.ec.europa.eu/collection/interoperable-europe/interoperable-europe> (Accessed: July 26, 2022)

¹⁰https://ec.europa.eu/isa2/eif_en/ (Accessed: July 14, 2022)

¹¹https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2018.295.01.0001.01.ENG (Accessed: October 11, 2022)

¹²https://ec.europa.eu/archives/isa/library/documents/eu-egovernment-action-plan-2016-2020_en.pdf (Accessed: July 29, 2022)

usable and high-quality fully online procedures accessible through the Single Digital Gateway defined in SDGR. The DE4A takes reality as its starting ground - the needs and the capacities of the Member States (MS), ensures compliance with all other relevant regulations (GDPR¹³, eIDAS¹⁴, Services Directive¹⁵, among others) as well as EU and MS strategic eGovernment and Single Market goals and aims to transform governance towards open collaboration and innovation, adaptable to depart from the variety of current environments¹⁶. It will allow for full regulatory compliance, establishing a culture of co-creation, transparency, accountability and trustworthiness. For a completely working digital single market, effectively enabling the cross-border exercise by citizens and businesses of their Single Market rights, EU MS must address several challenges in delivering better services. DE4A is a three-year pilot-led project started on January 1 2020, in support of the SDG across Europe ((EU) 2018/1724) and helping to make the single digital market a reality. DE4A includes 22 partners, including eight Member States (Austria, Luxembourg, Netherlands, Portugal, Romania, Slovenia, Spain, and Sweden). The project establishes a culture of co-creation, transparency, accountability and trustworthiness. Its goal is to facilitate migration towards European Digital Public Services co-delivered across borders, across sectors and with different participants. The project has received funding from the European Union's Horizon 2020 research and innovation program under GA No. 870 635¹⁷. ATOS leads the DE4A project, and it consists of several different work packages (project groups) covering different areas as follows: Inventory of current eGovernment landscape; Architecture vision and framework; Semantic interoperability solutions; Cross-border pilots for citizens, business and evaluation; Common component design & development; Sustainable impact and new governance models; Legal and ethical compliance and consensus building; Stakeholder dialogue, dissemination and communication; Project coordination and management and Ethics requirements.

The project is validating a comprehensive and scalable approach to develop and demonstrate the potential and benefits of securely and digitally sharing relevant-only and once-only data. DE4A aims to establish a culture of co-creation, transparency, accountability and trustworthiness. Citizen and business-oriented pilots shall highlight chosen aspects of the technological ecosystem available for the SDGR implementation at European and MS levels, prove their technical viability and gauge the performance and degree to which non-functional requirements can be accommodated. The pilots comprise functional use cases requiring interaction with real users – citizens, students, businesspersons and public servants. The three pilots are as follows:

- **Studying Abroad Pilot** will enable paperless procedures for students' mobility and simplify cross-border interactions for students engaging with processes, systems and platforms used by higher education establishments. It will allow digital
 - (a) applications for higher education,
 - (b) applications for study grants,
 - (c) and diplomas/certificates/studies/ professional recognition.
- **Doing Business Abroad Pilot** will lower barriers for companies to start a business or conduct business in another EU MS. It will enable businesses to
 - (a) retrieve and keep up-to-date company data from authentic sources,
 - (b) start a business and
 - (c) digital annual reporting.
- **Moving Abroad Pilot** will enable EU citizens' mobility across the EU and support them in moving from one MS to another by creating a digitized process for exchanging information required to facilitate a change of residency and other online cross-border procedures relying on civil status certificates¹⁸. The processes will support
 - (a) registering change of address,
 - (b) civil status certificates, and
 - (c) pension claims and information.

¹³<https://eur-lex.europa.eu/legal-content/SL/TXT/?uri=CELEX%3A32016R0679> (Accessed: July 26, 2022)

¹⁴https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3AOJ.L_.2014.257.01.0073.01.ENG (Accessed: July 26, 2022)

¹⁵<https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32006L0123> (Accessed: July 26, 2022)

¹⁶<https://www.de4a.eu/about-project>, (Accessed: July 21, 2022)

¹⁷https://www.de4a.eu/_files/ugd/2844e6_1e295e0aa15c4afa8f7f1baa52001189.pdf, (Accessed: July 21, 2022)

¹⁸https://www.de4a.eu/_files/ugd/2844e6_1e295e0aa15c4afa8f7f1baa52001189.pdf, (Accessed: July 21, 2022)

The project's main deliverables will comprehensively document and validate technologies that must be implemented for the cross-border integration of public services. Figure 2 shows the central and long-term deliverable of the proposed architecture framework. Some of the final deliverables will define all aspects, such as modelling requirements, processes, semantic solutions with models, communication messages and patterns and full framework implementation. The results will be demonstrated within the production-level implementation of specific cross-border and cross-domain pilots described above.

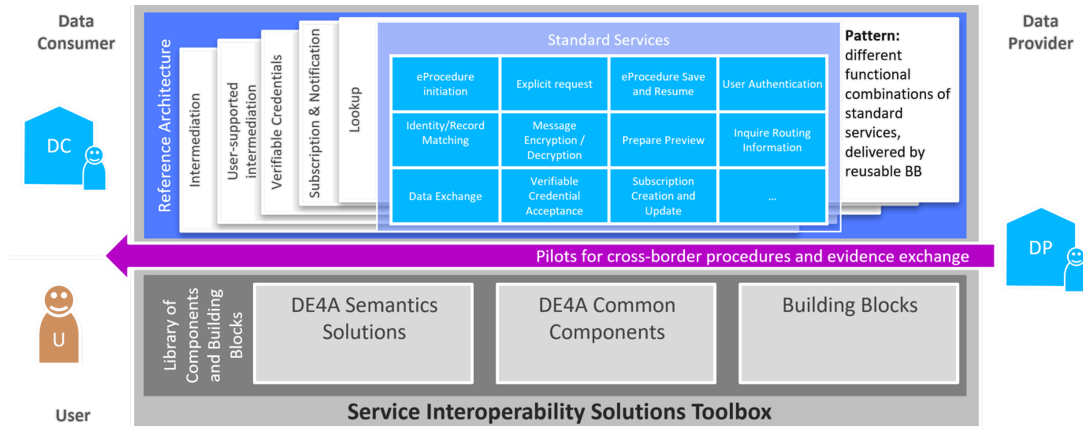


Figure 2: The DE4A project Service Interoperability Toolbox (source: <https://wiki.de4a.eu>, accessed July 29, 2022)

The key business roles identified by the DE4A are **User**, **Data Consumer (DC)**, which consists of Data Evaluator (DE) and Data Requestor (DR), **Data Provider (DP)**, which consists of Data Transferor (DT) and Data Owner (DO), and **Registrar**. User means a natural or legal person that accesses the system (uses the service). The registrar is responsible for accepting and storing formal information according to rules of law or governance. Data Consumer is an organization/administration in demand of the data, while a Data Provider is a legal entity in charge of the data deployment. It is essential to understand their specialized roles:

- **Data Consumer (DC)**
 - **Data Evaluator (DE)**: organization authorized to receive and process data from a User via OOTS.
 - **Data Requestor (DR)**: a technical role that searches and requests data. If DR is a separate entity from DE, it carries the request under the mandate of the DE.
- **Data Provider (DP)**
 - **Data Owner (DO)**: organization which governs information about the User. DO is responsible for authorization approval, data extraction (also from secondary registries) and audit control.
 - **Data Transferor (DT)**: a technical role responsible for the actual data transmission. It runs the OOTS-compliant data service. DT and DO can be different entities – e.g. when a national or sectorial intermediary is acting as an evidence broker.

These business roles can exchange structured data (i.e., evidence) or related information following Once-Only Interaction Patterns. The DE4A Architecture Framework defines five reference interaction patterns:

1. Intermediation Pattern,
2. User-supported Intermediation Pattern,
3. Verifiable Credentials Pattern,
4. Lookup Pattern, and
5. Subscription and Notification Pattern.

As the reusable prototype component we describe in this paper uses only the User-supported Intermediation Pattern or Intermediation Pattern, we will present these only. Other patterns are primarily used for communication between systems or credentials-related needs. Figure 3 shows an illustration of the Intermediation Pattern. In this pattern, a user initiates a procedure through a public service request. The DC is missing the required evidence that can be automatically acquired from a known DP. After the evidence (one or more) is retrieved, they are shown to the user, who can decide to complete the procedure. In some cases, the preview part might also allow a user to upload signed evidence in his possession manually.



Figure 3: Illustration of the Intermediation Pattern

In Figure 4, we show an example of a User-supported Intermediation Pattern. In this pattern, a user initiates a procedure through a public service request. In case when a DC needs additional evidence, it notifies a user and redirects him to identify and authenticate himself with a selected DP. In the meantime, the DC has requested the missing evidence, shown to the user via a preview component. If the user approves the evidence, DP sends it to the DC. If DC is missing more evidence, this process will be repeated. When all evidences are acquired, the user can decide to submit the initiated procedure.

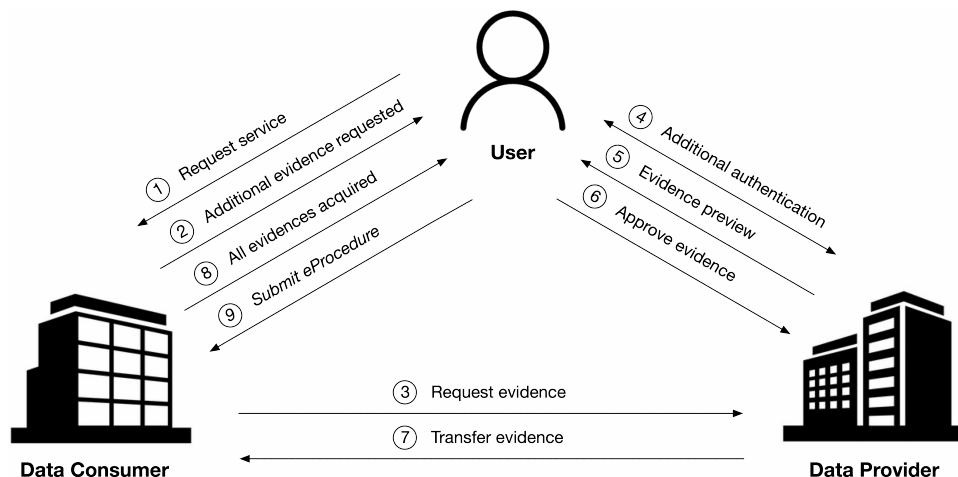


Figure 4: Illustration of the User-supported Intermediation Pattern (* in a case when additional evidence is requested)

3.2 Reusable prototype component use case: Multilingual Ontology Repository (MOR)

In the scope of the DE4A project, we have designed and tested the Multilingual Ontology Repository (MOR) prototype component for possible further use in semantic data exchange among different EU member states. The MOR component aims to register and provide information on concepts and terms involved in the evidence exchange and support the automatic generation of customizable user interfaces (i.e., for explicit request, preview, and additional parameters request) that contain complex terms in any EU official language. The component is developed as a prototype and is meant to be integrated into an arbitrary organization/administration portal that offers functionalities to users (e.g., evidence preview in step 5 of User-supported Intermediation Pattern communication).

The DE4A defined clear semantic and functional requirements for the component. Semantic requirements are highly related to MOR terms which are semantic elements (simple or complex) represented by a

Uniform Resource Identifier (URI), syntax and multilingual meaning. Examples of complex types are canonical evidence types (e.g., birth certificate¹⁹) that MOR needs to be able to process. Each term must be described using a label, description and an example in every EU official language. Suppose a description and an example are not provided for the target language. In that case, the MOR component needs to translate values automatically – these values must be shown to the user using the “non-verified” label. Complex terms are modelled as a tree hierarchy of terms, and a path within the hierarchy uniquely identifies them. If a complex term [X] is defined of type [Y], the [X] sub-terms have the same URI as the [Y] sub-terms except for the root element, so the root of [Y] term paths is replaced by the [X] URI. For example, the sub-term “Gender” of the complex term “BirthEvidence/Child” of type “Person” corresponds to the “Person/Gender” sub-term but with URI “BirthEvidence/Child/Gender” MOR also contains code lists for enumeration types. All terms must be reusable. For example, the complex term “BirthEvidence/Child” of type “Person” may overload the multilingual meaning label and description for newborn persons instead of persons in general.

Apart from semantic and syntax support, MOR must enable the automatic generation of customizable user interfaces for any complex term in any EU official language. There are three cases where this functionality can be of help:

- The explicit request functionality should inform the user of the information to be requested as evidence, so the MOR can help to create a building block that generates such a user interface for any canonical evidence type and language.
- The preview functionality should show the user the evidence to be incorporated into the procedure, so the MOR can help to create a building block that generates such a user interface for any canonical evidence type and language; audits can also use this building block to help auditors to understand any canonical evidence in any language. This is particularly interesting when the preview space is located on the evidence provider side. The language on this side can differ from the procedure requiring cross-border evidence, and the user may not understand the provider’s language.
- The additional parameters functionality requires the user to request some fields through a form, so the MOR can help create a building block that generates such a form for any additional parameters and language set. This is of particular interest when additional parameters have to be required on the evidence evaluator side because the evidence provider sets these parameters. In this case, the type of the terms is the key to generating the appropriate input field in the form (calendar, select list, text box, etc.)

3.3 Technical implementation

In this section, we overview the implementation of the prototype - MOR component, show options for its possible usage, and comment on its reusability and extensibility. The implementation is publicly available in a GitHub repository²⁰.

MOR implementation has two aspects:

- (a) the storage of the MOR terms according to the specification, and
- (b) the standard Graphical User Interface (GUI) components to provide, in a multilingual way, a customizable user interface for the explicit request, evidence preview and requesting additional parameters for the record matching.

Within the DE4A project, GUI for requesting additional parameters has not been considered, and some simplifications have been adopted to implement the MOR storage to reduce the technical complexity of its implementation. Since only the DE4A Moving Abroad pilot will use the component, MOR storage contains only canonical evidence types for Birth (**BirthEvidence**) and Marriage (**MarriageEvidence**).

¹⁹https://wiki.de4a.eu/index.php/Birth_Canonical_Evidence (Accessed July 26 2022)

²⁰<https://github.com/de4a-wp3/MOR> (Accessed: July 15, 2022)

3.3.1 Simplified MOR storage

MOR needs to store all MOR terms required for the three functionalities. As in the production environment, evidence types can be modified, or new types can be added; MOR should access these types via centralized services.

In our case, MOR storage is implemented as JSON files, one per available language, offered by a REST API. After the modelling, we generated files that represent all supported canonical evidence types (i.e., `BirthEvidence` and `MarriageEvidence`), complex terms used as types of other complex terms (e.g., `Person`, `Location`), and code lists (e.g., the EU Member States, administrative-territorial units, languages codes, currencies, gender). Available languages are the ones related to the Member States that participate in the Moving Abroad pilot: French (`fr`), Portuguese (`pt`), Slovenian (`sl`), Spanish (`es`), and, as the common language, English (`en`).

Each MOR term is comprised of five elements:

1. **Term URI:** URI is the path corresponding to the XML element of the term in the XML schema of the complex. For instance, the root MOR term of the `BirthEvidence:1.0` canonical evidence type would be “`BirthEvidence`”, i.e., the root element of the corresponding XML schema. The rest of the sub-elements of such XML schema are specified as separate MOR terms, such as “`BirthEvidence/CertifiesBirth/Child`”.
2. **Origin:** Source of the type of term, either DE4A or other XML namespaces, such as the ones used by ISA2. As the MOR component does not need to be aware of its origins, this field is not included in the MOR schema.
3. **Type:** Depending on the nature of the term, according to the defined schema:
 - XML base type with prefix “`xsd:`” (e.g., `xsd:string`) for a simple term.
 - “`xsd:enumeration`” for a complex term defined as a code list.
 - “`xsd:token`” for a simple term defined as an entry of a code list.
 - URI of another MOR term specified in a different schema (e.g., MOR term `LocationAddress` as a type of the MOR term `BirthEvidence/CertifiesBirth/Child/PlaceOfBirth`).
 - No type for root MOR terms. A complex term whose sub-elements are included in the same schema, so their URI starts with the URI of the complex term (e.g., `BirthEvidence`).
4. **Cardinality:** Represented as two Boolean values. The first value represents if the term is mandatory, and the second represents if the term can have multiple values. Root MOR terms have no cardinality.
5. **Comments:** Optional text to describe the term specification.

3.3.2 Multilingual Ontology Repository Application Programming Interface (MOR API)

The MOR API is defined to be the same for future implementations of the MOR storage with an existing database system. The MOR API retrieves all MOR terms for a given language via one REST endpoint: “`/mor/{lang}`”.

The JSON object returned by the MOR API contains MOR term values (presented in the previous section) and language-specific type values. These values are needed as metadata to present the retrieved schema to the user. The “`{lang}`” JSON object consists of a label, a description, an example, and a verified flag.

Examples of MOR term objects when Spanish is the selected language:

```
"BirthEvidence": {
  "type": "",
  "cardinality": "",
  "comment": "",
  "es": {
    "label": "Prueba de nacimiento",
    "description": "Información sobre un nacimiento que prueba su registro.",
    "example": ""
```



```

    "verified": "true"
  }
},
"BirthEvidence/CertifiesBirth/Child/PlaceOfBirth": {
  "type": "Person/PlaceOfBirth",
  "cardinality": "10",
  "comment": "Country and place of birth",
  "es": {
    "label": "País y lugar de nacimiento",
    "description": "País y lugar donde nació un niño",
    "example": "Spain, Cercedilla",
    "verified": "true"
  }
}
}

```

Note that “BirthEvidence” is the root term in Birth Canonical Evidence Type, so values of “type” and “cardinality” are empty.

The JSON specification of the MOR API response is aimed at easing the finding of MOR terms when using a JavaScript JSON variable. Because of this reason, the MOR specification of the term with URI “BirthEvidence/CertifiesBirth/Child/PlaceOfBirth” can be easily obtained by the expression `javascriptJsonVar[URI]`.

3.3.3 Reusable and customizable MOR GUI

Explicit request and evidence preview are required functionalities by every DE and DO when using a User-supported Intermediation Pattern. Multilanguage support is vital as a user from any member state can use services in any other member state.

As evidence types were thoroughly defined using semantics, schemas make it possible to define automatically customizable GUI. The look and feel of the component can be completely changed by providing a customized Cascading Style Sheet file (CSS). Each HTML DOM object within MOR is identified by its id; therefore, an integrator can completely redesign the component’s user interface. The component is implemented as an AngularJS component and can be included into any Web page using HTML placeholder `<app-mor-er ...></app-mor-er>`.

In Figure 5, we show the MOR component diagram. The blue denotes MOR implementation, while the red represents the external system (i.e., a specific Web page) where the MOR component could be integrated. MOR component requires a definition of a few parameters to be placed within the target Web page. Results of the component are returned into a JavaScript variable within the embedded Web page (as parameterized during placement). Below, we describe the configuration and running of explicit request and evidence preview functionalities.

3.3.3.1 Explicit request A page of the DE portal opens the MOR GUI page for the explicit request. The parameters that DE needs to provide when embedding the component into a Web page are the following:

- (a) a default component language,
- (b) the DE country,
- (c) a list of Canonical Evidence Types identified by the MOR root terms, and
- (d) the name of a JavaScript Map variable to fill by the MOR component (the hosted Web page can then use this data).

An example of embedding a MOR component for the explicit request functionality:

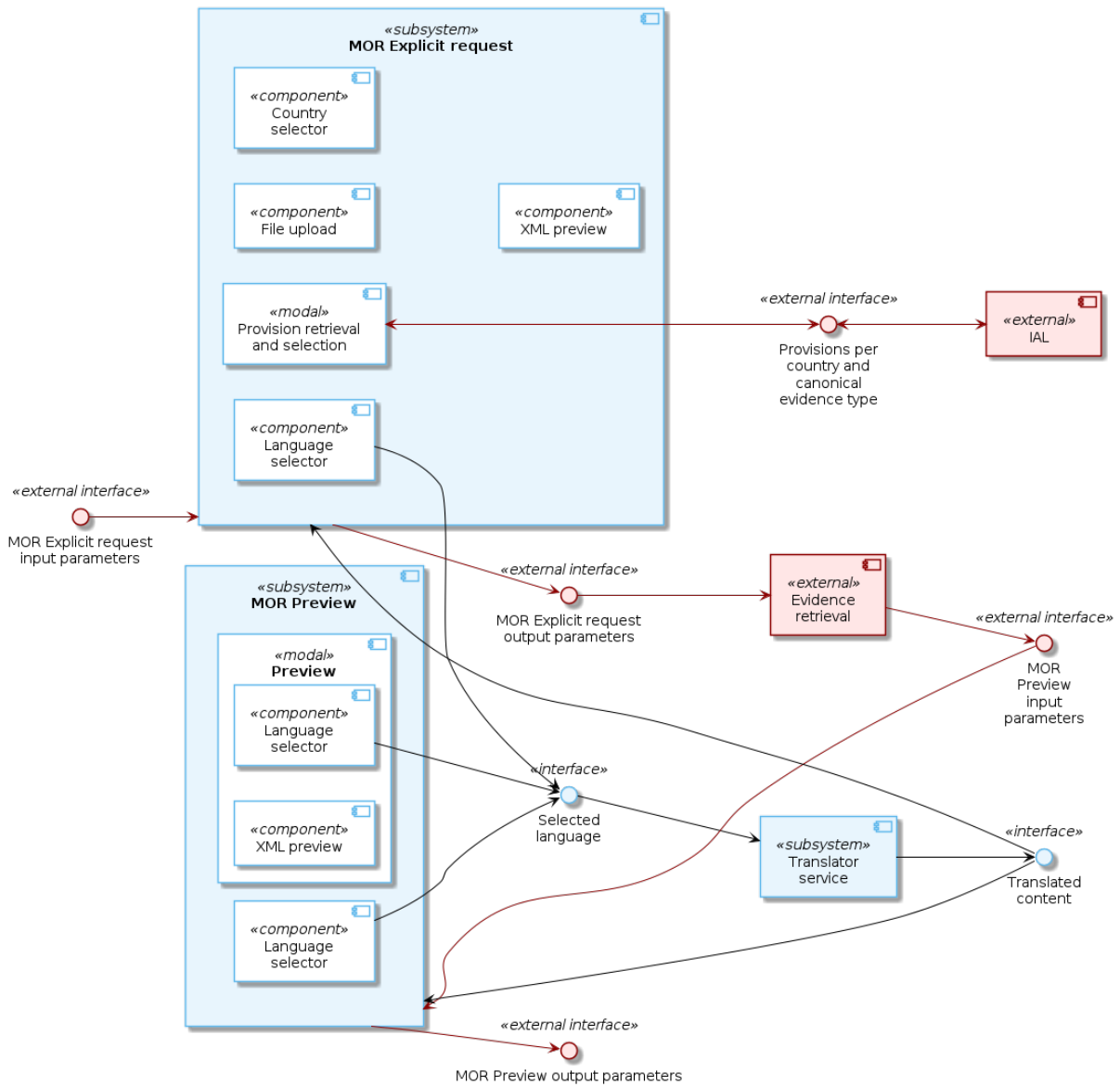


Figure 5: Multilingual Ontology Repository component diagram

```

<app-mor-er
  defaultLang = "en"
  requesterCountryCode = "BE"
  canonicalEvidenceTypes = "BirthEvidence,MarriageCertificate"
  outputJSMaId = "outputJSMaIdMorEr"
>>/app-mor-er>

```

When the page with the embedded MOR component is open, the MOR API is called to retrieve the MOR JSON in the given language. Figure 6 shows a default MOR GUI for the explicit request functionality (with the default language set in Slovene). It allows for changing the default language and provisioning of the needed evidence. For each evidence it provides for:

- Review of the schema and data fields that constitute evidence. The review is available in multiple languages and provides a list of all labels and descriptions of the fields (see Figure 7).
- Providing the evidence to the DE. The user can upload each evidence if he owns a relevant and verifiable digital document accepted by the DE (e.g., XML, PDF). The DE can automatically retrieve evidence if the user selects the country that takes the role of a DP. The evidence retrieval process follows the predefined exchange patterns (described in the previous section).

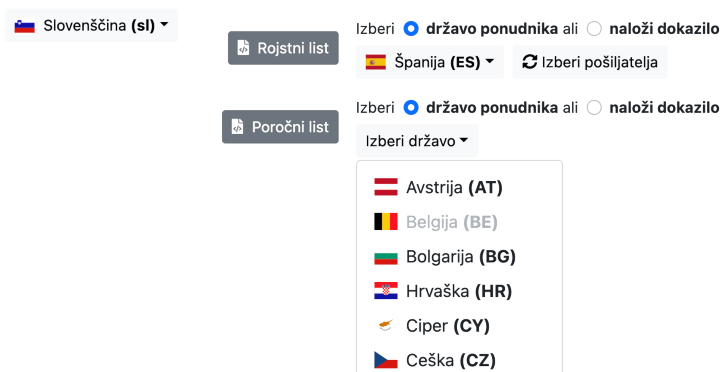


Figure 6: Default MOR component GUI for the explicit request functionality

The selection made by the user per canonical evidence type is returned by filling in the JavaScript Map variable specified. The keys in the map variable are the needed evidence type names. Each entry consists of one from two possible values depending on the user choice, that is,

- the selected provider country or
- the ASCII representation of the uploaded file, which can be either an ASCII file such as an XML file or a binary file such as a PDF.

3.3.3.2 Evidence preview After issuing the requested canonical evidence, the MOR component GUI for the evidence preview is opened by a Web page at the DE or the DO side, depending on the type of the exchange pattern (described in the previous section).

When the page with the embedded MOR component is open, the MOR API is called to retrieve the MOR JSON in the given language (if not already cached by the browser). Figure 8 shows a default MOR GUI for the evidence preview functionality. The parameters for embedding the preview part of the component include

- the default component language,
- a JSON array that contains JSON objects of specific evidence (key is the evidence type name and value is its representation), and
- a JavaScript Map variable to fill by the MOR component with the result of the evidence preview.

Evidência de nascimento ← Documento oficial ou dados provando o nascimento de uma criança.

Identificador ← Uma referência inequívoca às evidências de nascimento.

Conteúdo ← Conteúdo - String Alfa-numérica

Tipo de identificador ← O tipo de identificador emitido (se a autoridade emitir mais de um tipo de identificador).

Data de emissão ← A data em que foi emitida

Autoridade emissora ← O nome da agência emissora

ID da autoridade emissora ← O ID da agência emissora

Data de emissão ← A data mais recente em que a instância da evidência de nascimento foi emitida.

Autoridade emissora ← Uma entidade pública com autoridade oficial encarregada de emitir a evidência de nascimento.

Identificador ← Muitas organizações são referidas por um acrônimo ou algum outro identificador. Por exemplo, entre as instituições da UE, o BCE é o identificador do Banco Central Europeu, o OLAF do Organismo Europeu AntiFraud e assim por diante. Estes são formalmente reconhecidos pela Comissão Europeia, que fornece uma lista dessas siglas. Listas análogas devem ser usadas em outros contextos.

Conteúdo ← Conteúdo - String Alfa-numérica

Tipo de identificador ← O tipo de identificador emitido (se a autoridade emitir mais de um tipo de identificador).

Data de emissão ← A data em que foi emitida

Autoridade emissora ← O nome da agência emissora

Figure 7: Selected evidence type preview within a default MOR GUI

For instance:

```
<app-mor-p
  defaultLang = "en"
  postActionValue = "[{...}, {...}]"
  outputJSMaId = "outputJSMaIdMorEr"
>>/app-mor-p>
```

The component allows for reviewing the retrieved evidence types from the DO. If the evidence is not provided in machine-readable XML format following semantic schemas, the evidence is available as a binary file (e.g., PDF, image). The preview of evidence is like the explicit request part (Figure 7) with the addition of actual values for the user's evidence. Users can exclude the evidence at this step, and within the process, the system will ask them again to provide the needed canonical evidence (i.e., explicit request part).



Figure 8: Default MOR component GUI for the evidence preview functionality

The selection made by the user per canonical evidence type is returned by filling the defined JavaScript Map variable. The key to each entry is an evidence type name, while the value is of Boolean type, depending on the user's choice to include the evidence within the service.

3.4 Reusability evaluation

The MOR component building blocks (i.e., explicit request and evidence preview) have the advantage of being reusable and generic for any complex term and language, so parties in the evidence exchange do not need to develop their equivalent components, and any modification in MOR is automatically available through the MOR building blocks. For the best integration from the user experience perspective, CSSs are used. More specifically, we define the following aspects of adapting the component and its reusability:

- Configuration and integration: The component is packaged as a self-contained HTML DOM object with JavaScript functionality. It can be included on any modern Web page. The parameters for the component are set at its placement into an HTML DOM by providing values for its attributes. The results of the component are outputted to a selected JavaScript variable.

- Support of additional evidence types: The component can support an arbitrary evidence type and an arbitrary number. This is enabled using semantic schemas and MOR terms JSON provided by the DE4A services. If the services update JSON responses, the MOR component will automatically consider them and adapt the preview.
- Multilanguage support: Multiple languages support is enabled by MOR terms JSON files that are offered by the DE4A services. This JSON also includes translations for MOR component-related translations (e.g., names of buttons) which start with the root MOR term “GUI”. For example, a label of a proceed button in a selected language is available in the “GUI/proceed” field.
- Visual integration: One of the essential requirements was to support fully customizable visual integration of the component. There can be many DEs or DPs from different countries, and they already have their Web sites supporting digital public services. Therefore, the need is that the MOR component can be adapted for each one of those portals. In that sense, all the HTML DOM components in MOR have an associated id, based on which visual characteristics can be defined via the MOR CSS during the integration.

Pladenj Preview

Preview

The following information has been retrieved from the business register. Please review the provided information.

The information is formatted according to the European standards, therefore some information might look different from the country specific presentation.

HIGHER EDUCATION DIPLOMA

| | | | |
|-----------------|---|-----------------------|---|
| TITLE | Masters in dance | DEGREE | Masters |
| COUNTRY | http://publications.europa.eu/resource/authority/country/SVN | INSTITUTION NAME | Academy for Dance |
| STUDY PROGRAMME | Dance, choreography | MAIN FIELD OF STUDY | http://data.europa.eu/snb/iscd-f/0215 |
| MODE OF STUDY | http://data.europa.eu/europass/learningScheduleType/fullTime | DURATION OF EDUCATION | P2Y |
| SCOPE | 120 | DATE OF ISSUE | 2017-04-21 |

PLACE OF ISSUE

| | |
|------|-----------|
| NAME | Ljubljana |
|------|-----------|

HOLDER OF ACHIEVEMENT

| | | | |
|--------------|---------------|---------------|------------|
| NATIONAL ID | 0707993505389 | GIVEN NAMES | Ana |
| FAMILY NAMES | Maček | DATE OF BIRTH | 1993-07-07 |

Figure 9: Evidence preview example by the Slovenian implementation of a pilot supporting digital public services related to “Studying abroad”– Pladenj (Tray) Preview

Furthermore, we would like to mention the Slovenian national “Once-Only” platform “Tray”, which operates as a decentralized data exchange system supporting the implementation of the Government to Citizens (G2C), Government to Businesses (G2B) and Government to Government (G2G) e-procedures. The system can also obtain data from many sources through a single-entry point, exchanging a minimum data set. The central system, “The tray”, takes care of optimal data load from all the data sources with the help of the control and forecasting system based on continuous machine learning. Based on this technology, a prototype for Slovenian evidence preview has been developed in the DE4A project as one of the possible solutions. Figure 9 shows an example of evidence preview by the Slovenian implementation of a pilot supporting digital public services related to “Studying abroad” (using Tray for data exchange). MOR component can be adapted to allow the same look and feel (in comparison to the default MOR evidence type preview setting, shown in Figure 7).

MOR component might therefore be used by countries that do not yet implement such functionalities. Within the DE4A project, these countries might be Spain, Portugal or Romania, which take part in the pilots.

4 Conclusions

In this paper, we presented the main characteristics of the DE4A project aiming to support the delivery of cross-border digital public services. We showed the evolution of the development of reusable prototype components in software engineering and how it evolved for the needs of public administration systems based on OOTS principles. We briefly reviewed progress in cross-border and cross-sector public administration interoperability. We focused on specifics of Digital Europe for all projects, especially those related to the use case of a reusable prototype component. In more detail, we presented a use case for building a semantic and Web-based reusable prototype component. The pilot component Multilingual Ontology Repository is a deliverable of the Digital Europe for all project. We evaluated and showed how the component offers adaptability to any Web site and is configurable to support future semantic models.

We believe that using such prototypes and building blocks could support the EU Member States' public administrations' interoperability and would open new opportunities to support mobility, fasten processes and be very cost-effective. Also, innovative technologies will be able to serve with transformative impact, for example, blockchain (for effective notarization, fostering accountability and transparency in the distributed transactional environment) and machine learning (over usage data to automate monitoring and improve effectiveness/quality of eProcedures).

Acknowledgements

The research and development reported in this paper have received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no. 87035 (DE4A: Digital Europe for All). We would like to thank the members of the DE4A WP3 group, especially the WP leader, Thashmee Karunaratne, and Ana Rosa Guzmán. We would also like to give credit to the authors of the DE4A deliverable D2.1, from where we used Figure 2 (ATOS is the leading partner of the DE4A project).

Literature

- Castano, S, V. D. Antonellis, and B Pernici. 1995. "Building Reusable Components in the Public Administration Domain." In, 81–87.
- Gackenhaimer, C, and A Paul. 2015. *Introduction to React*. Vol. 52. Apress.
- GovStack. 2022. "Accelerating the Digital Transformation of Government Services." <https://www.govstack.global/>.
- Kozaczynski, W, and G Booch. 1998. "Component-Based Software Engineering." *IEEE Software* 15 (5): 34–36. <https://doi.org/10.1109/MS.1998.714621>.
- McIlroy, M. D., J Buxton, P Naur, and B Randell. 1968. "Mass-Produced Software Components." In, 88–98. Garmisch Pattenkirchen.
- Narducci, F, M Palmonari, and G Semeraro. 2013. "Cross-Language Semantic Matching for Discovering Links to e-Gov Services in the Lod Cloud." In, 992:21–32.
- Peristeras, V, N Loutas, and K Tarabanis. 2008. "Organizational Engineering in Public Administration: The State of the Art on eGovernment Domain Modeling." In, 580–87.
- Peristeras, V, K Tarabanis, and S. K. Goudos. 2009. "Model-Driven eGovernment Interoperability: A Review of the State of the Art." *Computer Standards & Interfaces* 31 (4): 612–28.
- Sametinger, Johannes. 1997. *Software Engineering with Reusable Components*. Springer.
- Wallace, P. 2010. "A Hole for Every Component, and Every Component in Its Hole. Existential Programming." 2010. <https://existentialprogramming.blogspot.com/2010/05/hole-for-every-component-and-every.html>.