



Designing an API for the Provision of Public Service Information Based on CPSV-AP

Christos Pappis¹, Dimitris Zeginis^{2,3}(✉), Efthimios Tambouris³,
and Konstantinos Tarabanis^{2,3}

¹ Hellenic Open University, Patras, Greece
std142986@ac.eap.gr

² Centre for Research and Technology - Hellas, 6th km. Charilaou-Thermi RD,
Thessaloniki, Greece
{zeginis,kat}@iti.gr

³ University of Macedonia, Egnatia 156, 54636 Thessaloniki, Greece
{tambouris,zeginis,kat}@uom.edu.gr

Abstract. The provision of public services (PS) is at the heart of public authority operations as it directly affects citizens' lives and the prosperity of society. Part of PS provision is publishing PS descriptions in online catalogues to inform citizens and promote transparency. The European Union has developed the Core Public Service Vocabulary Application Profile (CPSV-AP) as a PS data model to facilitate PS catalogue creation and promote semantic interoperability. The use of CPSV-AP is increasing within the European Union leading to a number of relevant PS descriptions that are also potentially beneficial for the implementation of the Single Digital Gateway (SDG) regulation. However, the use of heterogeneous, some times not very popular technologies (e.g. RDF) to provide access to such data hinders their further uniform exploitation. Towards this problem definition, Application Programming Interfaces (APIs) can provide a uniform, easily-consumable way of serving such data hiding any heterogeneity of the underlying technologies. The aim of this paper is to design and implement an API to serve data based on CPSV-AP facilitating the implementation of the SDG and the further exploitation of data at added value services and tools e.g. chatbots. The paper describes the requirements, use cases, design choices and implementation details of the API. We anticipate that the proposed API will help towards the further adoption of CPSV-AP by facilitating and homogenizing the consumption of relevant data.

Keywords: eGovernment · API · public service · CPSV-AP

1 Introduction and Motivation

The provision of public services (PS) is the main task of most public authorities and a major part of their operations in total. PS provision is usually the main

task for the implementation of a governmental policy targeting the fulfillment of citizens' needs. As PS provision is a major part of public sector operations, the opening of information on PSs, i.e. publishing structured PS descriptions, highly contributes to the realization of the Open Government vision, promoting transparency and trust between public administrations and citizens.

Every eGovernment system that provides PSs is based on an underlying PS model [15]. PS models and standards are considered as the main enablers for promoting interoperability and quality in the PS provision domain. Embracing the need for a standard PS model, the European Commission has developed the Core Public Service Vocabulary (CPSV) [9] that enables the description of public services.

The adoption of CPSV is expected to facilitate semantic interoperability of public service descriptions throughout Europe. This will be beneficial for the implementation of the Single Digital Gateway (SDG) regulation [14], since the European coordinator of the SDG has to collect the descriptions of public services from European public administrations in one unique portal. However, the use of heterogeneous technologies (e.g. SQL, RDF) to store the data produced based on CPSV-AP as well as the unfamiliarity with some of the technologies related to the CPSV-AP (e.g. RDF) may hinder the collection of such data by the SDG as well as their exploitation by added value services and tools.

Towards this problem definition, APIs can provide uniform access to data described based on CPSV-AP hiding any heterogeneity of the underlying technologies [20]. Additionally, the use of most CPSV-AP data requires skills and tooling (e.g. RDF, SPARQL) that are less widespread than some other web technologies (e.g. JSON, Javascript). For example, there are many Javascript visualization libraries that consume JSON data (e.g. D3.js, charts.js), while there are just a few that consume RDF and their functionality is limited.

In order to unleash the full potential of CPSV-AP described data there is a need to standardize the interaction (i.e. input, output and functionality) in a way that facilitates: i) the development of reusable services and tools and ii) the automatic collection of public service descriptions from different catalogues. This paper describes the requirements, use cases, design choices and implementation details of an API that aims to exploit and unify CPSV-AP data stored using heterogeneous technologies while also making data available in a structure and format that is familiar to a larger group of developers. The API offers a uniform way to access the data, thus enabling the development of generic services and tools that can be reused across datasets and even cross-boarder.

The rest of the paper is organized as follows, Sect. 2 presents relevant background and related work, Sect. 3 describes the methodology followed for the development of the API, Sect. 4 presents the use cases and requirements that need to be covered by the API, Sect. 5 describes the design decisions for the API, Sect. 6 presents the implementation details, Sect. 7 describes some exaples scenarios using the API and finally Sect. 8 concludes the paper.

2 Background Work

This section presents some background work needed to understand the context of the paper and the developed API.

2.1 The Core Public Service Vocabulary (CPSV)

The Core Public Service Vocabulary (CPSV) [9] (Fig. 1) is a data model for describing public services and the associated life and business events (e.g. “having a baby”, “starting a new business”). The two main classes of the model are the “Public Service” and the “Public Organization” that is responsible for offering the service. Except from these two classes there are many other describing other aspects of the public service including the required input to execute the service (i.e. “Evidence”), the “Output” of the service, the “Cost” of the service, the “Channel” through which the service is offered, the “Contact Point” of the service etc.

Based on CPSV a linked data application profile CPSV-AP was also developed. An Application Profile is a “specification that re-uses terms from one or more base standards, adding more specificity by identifying mandatory, recommended and optional elements to be used for a particular application, as well as recommendations for controlled vocabularies to be used” [13]. For example, CPSV-AP reuses classes and the properties of existing vocabularies including FOAF [4], DCT [11] and DCAT [1] as well as other Core Vocabularies e.g. CPOV. It also proposes the use of controlled vocabularies to populate the properties e.g., for the names of countries [17] and languages [18]. The more recent version of CPSV-AP is version 3.0 and was published in 2022.

The classes and properties of CPSV-AP are categorized as mandatory or optional. The minimum requirements of a public service description, in order to comply with CPSV-AP, is to provide at least information on the mandatory properties of the mandatory classes. Optional classes can still have mandatory properties for which information should be provided when an optional class is used.

The CPSV-AP belongs to a set of Core Vocabularies that have been developed by the European Commission in order to conceptualise the public service provision domain. These vocabularies include, apart from the CPSV-AP, the Core Criterion and Core Evidence Vocabulary (CCCEV) [6], Core Person Vocabulary (CPV) [7], the Core Public Organisation Vocabulary (CPOV) [8] and the Core Business Vocabulary (CBV) [5].

Currently there are many known uses of CPSV-AP throughout Europe [12]:

- Belgium used the CPSV-AP to harmonise public services data from different regional sources and centralise them into a common system. Additionally, the region of Flanders adopted a slightly updated version of the CPSV-AP as their regional model for describing public services.
- The Estonian Ministry of Economic Affairs extended CPSV-AP to address local needs, as well as to cover the public service life-cycle. They also developed a service to harvest data based on their extension.

- Slovakia is working on a national data model based on the CPSV-AP to support the mapping of the Slovakian central meta information system of public administration to the CPSV-AP.
- Spain is working on a pilot with Portugal for which they are using the CPSV-AP to create a cross-border catalogue of public services.
- The Netherlands reused certain CPSV-AP tools to develop a simple way of creating public service descriptions matching the Dutch national data model.

2.2 Semantic Web and Linked Data

The term semantic web refers to an extension of the current “web of documents” in order to build a “web of linked data”. In order to achieve the vision of linked data, a set of principles have been proposed to “publish data on the web in such a way that it is machine-readable, its meaning is explicitly defined, it is linked to other external datasets, and can in turn be linked to from external datasets” [3]. Semantic Web and linked data are empowered by technologies such as RDF [10] and SPARQL [21] in order to publish data on the web.

More specifically, RDF is a W3C standard model for data interchange on the Web. It uses URIs to name things and their relationships that are expressed as “triples” $\langle X, Y, Z \rangle$ (e.g., $\langle \text{PublicService1}, \text{hasCompetentAuthority}, \text{Organization1} \rangle$). RDF has many serialization formats such as RDF/XML [23], Turtle [22] and JSON-LD [24]. The JSON-LD is the most recently developed serialization that aims to be lightweight and easy for humans to read. It is based on the successful JSON format and is ideal for APIs. SPARQL is a language to express queries across diverse RDF data sources.

Many researchers suggest that the linked data paradigm is ideal for Public Sector Information publishing, enabling the potentiality of breaking the bureaucratic silos [16]. Towards this direction, CPSV-AP incorporates linked data as an underpinning technology. Thus, public service descriptions, that are modeled using CPSV-AP, can be published as linked data and become part of the linked data cloud.

However, linked data technologies, apart from its benefits, puts some significant challenges [25] [16] hindering the adoption of relevant vocabularies such as CPSV-AP. For example, using linked data requires skills and tooling (e.g. RDF, SPARQL) that are less widespread than some other web technologies (e.g. JSON, Javascript). Thus, there is a need to standardize the interaction (i.e. input, output and functionality) with linked data in a way that facilitates the development of reusable services and tools.

2.3 APIs for Public Service Descriptions

The Application programming interfaces (APIs) are essential enablers of the transformation towards digital governments due to their modularity, reusability and scalability [20]. The APIs are the connection nodes that standardise the interactions at digital architectures and thus foster interoperability through data transfer.

In the context of public service cataloging, EU has already identified cases [19] where either there is limited use of relevant APIs or cases where a CPSV-AP based API will be beneficial. For example a case where there is a limited use of APIs, is the Autonomous Province of Trento. They have established a common datastore¹ of concepts (e.g. Public Services) which is used by several local authorities. The data exchanged among local authorities is shared by means of a REST API and published in JSON format. In order to map the shared data structure to other data models, such as the CPSV-AP, a mapping tool has been developed so that data is transformed from JSON to JSON-LD format.

Another case that is quite mature but there is still no relevant API is the Flemish Information Agency. They have developed a model, called OSLO-Dienstencatalogoog², which extends concepts of CPSV-AP in order to uniform their public services. On top of this model they also provide a JSON-LD context that can be shared between APIs.

3 Methodology

This section presents the methodology that has been followed in order to develop an API for the provision of public service information based on CPSV-AP:

- Define the use cases and the corresponding requirements that need to be covered by the API (Sect. 4).
- Define the design criteria and decisions for the API (Sect. 5).
- Implement the API (Sect. 6).
- Apply the API at some relevant scenarios (Sect. 7).

The following sections describe in detail these steps.

4 Use Cases and Requirements

The main requirement of the API is to support Create, Read, Update, Delete (CRUD) operations on all the CPSV-AP concepts. Additionally, there is also a need for basic search functionality of public services based on some criteria that have been identified by the bibliography [15]. The main use cases that need to be supported by the API are depicted in a UML use case diagram (Fig. 2) and are the following:

- Retrieve instances of CPSV-AP concepts (e.g. public service). The API should support retrieving either all the instances or a specific instance (e.g. based on its id). The description of the instances should include all the relevant properties defined by CPSV-AP.

¹ <http://ontopa.opencontent.it/openpa/classes>.

² <https://data.vlaanderen.be/doc/applicatieprofiel/dienstencatalogoog/>.

- Search for Public Services. This includes: i) searching based on the competent authority of the service, ii) searching based on the input/output of the service, iii) searching based on the event (i.e. life event or business event) related to the public service and iv) search for dependent public services (based on the properties “requires” and “related” of CPSV-AP).
- Create a new instance of a CPSV-AP concept.
- Update an existing instance of a CPSV-AP concept.
- Delete an existing instance of a CPSV-AP concept.

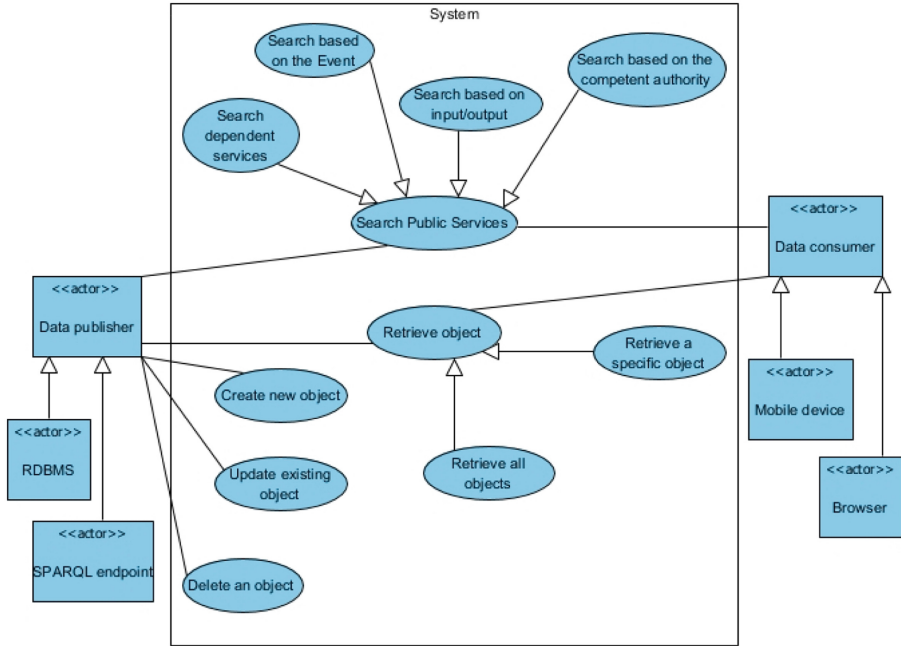


Fig. 2. UML Use Case diagram of the API

Mainly there are two actors/systems that are expected to use the API. The “data publisher” that is responsible for publishing and maintaining the public service descriptions using the API. The “data publisher” can use any storage technology e.g. RDBS or SPARQL endpoints. The “data consumer” can be any application (e.g. mobile device or browser) that retrieves or searches for public service descriptions in order to present them to the end user.

5 API Design

Based on the use cases and requirements described at Sect. 4 the main functionality of the API is to support CRUD operations on all the classes of CPSV-AP,

but also support searching of public services based on specific criteria. The architectural approach that has been selected for the design of the API is the REST, since it offers many advantages including the simplicity, scalability, speed, and ability to handle all data type through simple HTTP requests. The functionality of the API, in terms of supported REST calls, is depicted at Table 1.

Table 1. Supported REST API calls

API call	Description
GET endpoint/class	Returns all the instances of the class
GET endpoint/class/id	Returns a specific instance of the class based on the id
GET endpoint/public_services	Returns all the instances of the class “public service”. This call accepts also search parameters for: i) the competent authority, ii) the event (e.g. life event), iii) the input/output and iv) the related services (using the properties requires/related of the public service)
POST endpoint/class	Creates a new instance of the class
PUT endpoint/class/id	Updates an existing instance of the class based on the id
DELETE endpoint/class/id	Deleted an existing instance of the class based on the id

Another functionality of the API is the support of different storage technologies in a transparent way to the end user. Specifically, the API supports both RDBS and SPARQL endpoints to retrieve relevant results that are then returned to the end user in a homogenized way. Towards this direction, a design choice that was made is to use JSON-LD as the format of the results that is capable of “blending” relational data with RDF data using also the CPSV-AP model.

6 API Implementation

The API is implemented in PHP language making use of the Symfony 6.0 web application framework that supports the Model-View-Controller architecture. In such architectures the Model component undertakes the management of the data (i.e. storage, processing and retrieval), the View component manages the presentation of information to the end user and the Controller component is the interface between Model and View components. The code of the developed API follows this approach.

Additionally, the developed API uses the api-platform 2.6 framework that facilitates the building of web APIs, the Doctrine platform that supports the

abstracted use of data bases and the Composer platform for dependency management. The code of the API is available at Github³.

The code is organized in a way where a separate source file is created for each of the CPSV-AP classes and contains information about the way to handle each class. For example, Listing 1 presents the code that defines the URI of the relevant CPSV-AP class (i.e. Public Service) as well as the supported REST API calls, while Listing 2 presents the code that defines the supported search parameters for the Public Service class.

Listing 1. Supported API calls for the class PublicService

```
#[ApiResponse(
    uri: 'http://purl.org/vocab/cpsv#PublicService',
    collectionOperations: ['get', 'post'],
    itemOperations: ['get', 'put', 'delete'])]
```

Listing 2. Search parameters for the class PublicService

```
#[ApiFilter(SearchFilter::class, properties: [
    'identifier' => 'exact',
    'name' => 'exact',
    'description' => 'partial',
    'hasCompetentAuthority.id' => 'exact',
    'hasCompetentAuthority.preferredLabel' => 'exact',
    'isGroupedBy.id' => 'exact',
    'isGroupedBy.name' => 'exact',
    'produces.id' => 'exact',
    'produces.name' => 'exact',
    'requires.id' => 'exact',
    'requires.name' => 'exact',
    'related.id' => 'exact',
    'related.name' => 'exact',
    'hasInput.id' => 'exact',
    'hasInput.name' => 'exact'])]
```

The returned results of the API are in JSON-LD format that extends JSON by adding linked data annotations. In this way the returned results are also compatible with CPSV-AP.

The API supports the connection both to relational databases and RDF SPARQL endpoints. With the appropriate configuration the API can retrieve (using the GET method) data both from relational databases and SPARQL endpoints and return them in a unified way. Specifically, the GET method executes the appropriate SELECT statements on the relational database and the SPARQL endpoint, collects the results from both sources, merges them and returns the result. Currently, this functionality is supported only for the Public Service and the Public Organization classes. Additionally, the current implementation supports only the GET method for the SPARQL endpoints, while all

³ <https://github.com/chpappis/cpsv-php-api>.

the methods (GET, POST, PUT, DELETE) are supported for the relational databases.

Figure 3 presents a part of the API implementation description for the “Public Service” and the “Rule” classes.

PublicService	
GET	/api/public_services Retrieves the collection of PublicService resources.
POST	/api/public_services Creates a PublicService resource.
GET	/api/public_services/{id} Retrieves a PublicService resource.
PUT	/api/public_services/{id} Replaces the PublicService resource.
DELETE	/api/public_services/{id} Removes the PublicService resource.
Rule	
GET	/api/rules Retrieves the collection of Rule resources.
POST	/api/rules Creates a Rule resource.
GET	/api/rules/{id} Retrieves a Rule resource.
PUT	/api/rules/{id} Replaces the Rule resource.
DELETE	/api/rules/{id} Removes the Rule resource.

Fig. 3. Part of the description of the API

7 Using the API: Example Scenarios

An instance of the API has been deployed in order to execute some example scenarios. The instance of the API is available at the URL: <http://pappis.gr/api>. The API is connected at two data sources:

- A relation database with CPSV-AP data related to the public service “Getting a Greek Passport” [2].
- The SPARQL endpoint <http://data.dai.uom.gr:8890/sparql> that contains CPSV-AP based linked data descriptions of Public Services. The SPARQL endpoint uses the Virtuoso open-source RDF database.

The following scenarios are executed using the Postman application for calling the API and getting the results.

Scenario 1: search public services based on the name of the public organization (i.e. using the property `hasCompetentAuthority` of CPSV-AP) that offers them. For example, the following API call retrieves all the public services that are offered by the Public Organization with name “Passport Issuer”.

```
http://pappis.gr/api/public_services?
hasCompetentAuthority.preferredLabel=Passport Issuer
```

An excerpt of the returned result is presented at Fig. 4. The excerpt contains two public services with id 1 and 3 that are offered by the Public Organization with name “Passport Issuer” (some of the API results are in Greek since the relational database and the SPARQL endpoint exploited contain data in Greek).

```
{
  "@context": "/api/contexts/PublicService",
  "@id": "/api/public_services",
  "@type": "hydra:Collection",
  "hydra:member": [
    {
      "@id": "/api/public_services/1",
      "@type": "http://purl.org/vocab/cpsv#PublicService",
      "id": 1,
      "identifier": "Ps0001",
      "name": "Έκδοση Διαβατηρίου",
      "description": "Διαβατήριο είναι το έγγραφο που εκδίδεται ...",
      "processingTime": "Ρ0Υ0Μ3ΔΤ0Η0Μ0Σ",
      "keyword": "Έκδοση, Διαβατήριο, Ελληνική Αστυνομία, ...",
      ....
    },
    {
      "@id": "/api/public_services/3",
      "@type": "http://purl.org/vocab/cpsv#PublicService",
      "id": 3,
      "identifier": "Ps0003",
      "name": "Έκδοση Πιστοποιητικού",
      "description": "Υπηρεσία Έκδοσης Πιστοποιητικών",
      "processingTime": "Ρ0Υ0Μ1ΔΤ0Η0Μ0Σ",
      "keyword": "Έκδοση, Πιστοποιητικό",
      ...
    }
  ],
  ...
}
```

Fig. 4. Excerpt of the result of Scenario 1

Scenario 2: search of public services based on the life event they belong. For example, the following API call retrieves all the public services that belong to the life event “Traveling abroad” with id = 1. The searching is also possible using the life event name (i.e. isGroupedBy.name) but in this case the greek name of the event is required as input.

```
http://pappis.gr/api/public_services?isGroupedBy.id=1
```

Scenario 3: search of public services based on their input/output. For example, the following API call retrieves the public service that produces as output the “Passport with 3 years duration” with id = 1. The searching is also possible using the output name.

```
http://pappis.gr/api/public_services?produces.id=2
```

Scenario 4: search of dependent public services (i.e. using the properties “requires” and “related” of CPSV-AP). For example, the following API call retrieves the public service that requires the execution of the public service “Certificate issuance” with id = 4. The searching is also possible using the public service name.

```
http://pappis.gr/api/public_services?requires.id=4
```

An excerpt of the returned result is presented at Fig. 5. The excerpt contains one public service with id = 1 that requires for its execution the public service with id = 4.

```
{
  "@context": "/api/contexts/PublicService",
  "@id": "/api/public_services",
  "@type": "hydra:Collection",
  "hydra:member": [
    {
      "@id": "/api/public_services/1",
      "@type": "http://purl.org/vocab/cpsv#PublicService",
      "id": 1,
      "identifier": "Ps0001",
      "name": "Έκδοση Διαβατηρίου",
      "description": "Διαβατήριο είναι το έγγραφο που εκδίδεται από ...",
      "processingTime": "P0Y0M3DT0H0M0S",
      "keyword": "Έκδοση, Διαβατήριο, Ελληνική Αστυνομία, Ταξίδι, ...",
      "requires": [
        {
          "@id": "/api/public_services/4",
          "@type": "http://purl.org/vocab/cpsv#PublicService",
          "id": 4,
          "identifier": "Ps0004",
          "name": "Έκδοση Βεβαίωσης",
          "description": "Υπηρεσία Έκδοσης Βεβαίωσης",
          "processingTime": "P0Y0M2DT0H0M0S",
          "keyword": "Έκδοση, Βεβαίωση",
          ...
        }
      ]
    },
    ...
  ]
}
```

Fig. 5. Excerpt of the result of Scenario 4

8 Conclusion

The Core Public Service Vocabulary Application Profile (CPSV-AP) has been proposed as a European PS data model to facilitate PS catalogue creation and promote semantic interoperability in the PS information provision. Its broader use will also be beneficial for the implementation of the Single Digital Gateway regulation. However, the use of heterogeneous, some times not very popular technologies (e.g. RDF) to provide access to CPSV-AP data hinders their further uniform exploitation.

Towards this direction, this paper proposes the development of an API to provide a uniform, easily-consumable way of serving CPSV-AP data hiding any

heterogeneity of the underlying technologies and also facilitating the automatic collection of PS descriptions from multiple catalogues. The API currently provides CRUD operations on all the concepts of CPSV-AP and also basic search functionality of public services. The API is designed in a flexible way enabling the easy addition of more functionalities e.g. searching based on other more advanced criteria. Additionally, the API supports both relational databases and partially SPARQL endpoints enabling also their integration (only for public services and public organizations). In a future version of the API the authors aim at extending the functionality for the SPARQL endpoints by supporting also the PUT, POST and DELETE methods as well as by supporting all the CPSV-AP concepts at the GET.

We anticipate that the proposed API will help towards the further adoption of CPSV-AP by facilitating the development of relevant added value services and applications as well as the automatic collection of PS descriptions from various catalogues throughout Europe. This will provide multiple benefits to citizens including the provision of integrated information about PS throughout Europe, the increase of the quantity/quality of (mobile) applications that use CPSV-AP compliant data and the personalization of PS based on the citizen's profiles. All the above can lead to the reduction of bureaucracy and time spend to identify and execute public services that in turn can foster transparency and trust between the public administration and the citizens.

Acknowledgment. Part of this work was funded by the European Commission, within the H2020 Programme, in the context of the project inGov under Grant Agreement Number 962563 (<https://ingov-project.eu/>).

References

1. Albertoni, R., Browning, D., Cox, S., Beltran, A.G., Perego, A., Winstanley, P.: Data catalog vocabulary (dcat) - version 2. W3C Recommendation (2020). <https://www.w3.org/TR/vocab-dcat-2/>
2. Antoniadis, P., Tambouris, E.: Passbot: a chatbot for providing information on getting a greek passport. In: ICEGOV 2021, pp. 292–297. Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3494193.3494233>
3. Bizer, C., Heath, T., Berners-Lee, T.: Linked data-the story so far. semantic services, interoperability and web applications: emerging concepts, pp. 205–227 (2009)
4. Brickley, D., Miller, L.: Foaf vocabulary specification (2004). <https://xmlns.com/foaf/0.1/>
5. Core Vocabularies Working Group: Core business vocabulary. EU Semic Recommendation (2022). <https://semiceu.github.io/Core-Business-Vocabulary/releases/2.00>
6. Core Vocabularies Working Group: Core criterion and core evidence vocabulary. EU Semic Recommendation (2022). <https://semiceu.github.io/CCCEV/releases/2.00/>
7. Core Vocabularies Working Group: Core person vocabulary. EU Semic Recommendation (2022). <https://semiceu.github.io/Core-Person-Vocabulary/releases/2.00>

8. Core Vocabularies Working Group: Core public organisation vocabulary. EU Semic Recommendation (2022). <https://semiceu.github.io/CPOV/releases/2.00>
9. Core Vocabularies Working Group: Core public service vocabulary application profile (cpsv-ap). EU Semic Recommendation (2022). <https://semiceu.github.io/CPSV-AP/releases/3.0.0>
10. Cyganiak, R., Wood, D., Lanthaler, M.: Rdf 1.1 concepts and abstract syntax. W3C Recommendation (2014). <https://www.w3.org/TR/rdf11-concepts/>
11. DCMI Usage Board: Dcmi metadata terms (2020). <https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>
12. DG-DIGIT/ISA-Programme: About core public service vocabulary application profile. <https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/core-public-service-vocabulary-application-profile/about>. Accessed 18 Oct 2022
13. DG-DIGIT/ISA-Programme: D04.01 core public service vocabulary application profile 2.0. European Commission, Technical Report (2016)
14. European Union: Establishing a single digital gateway to provide access to information, to procedures and to assistance and problem-solving services. Regulation (EU) 2018/1724 of the European Parliament and of the Council (2018)
15. Gerontas, A., Peristeras, V., Tambouris, E., Kaliva, E., Magnisalis, I., Tarabanis, K.: Public service models: a systematic literature review and synthesis. *IEEE Trans. Emerg. Top. Comput.* **9**(2), 637–648 (2021). <https://doi.org/10.1109/TETC.2019.2939485>
16. Mouzakitis, S., et al.: Challenges and opportunities in renovating public sector information by enabling linked data and analytics. *Inf. Syst. Front.* **19**(2), 321–336 (2016). <https://doi.org/10.1007/s10796-016-9687-1>
17. Publications Office of the European Union: The country authority table (2022). <https://publications.europa.eu/resource/dataset/country>
18. Publications Office of the European Union: The language authority table (2022). <https://publications.europa.eu/resource/dataset/language>
19. PwC EU Services: APIs for cpsv-ap based catalogue of services (2019)
20. Vaccari, L., et al.: Application programming interfaces in governments: Why, what and how. European Commission - JRC science for policy report (2020)
21. W3C: Sparql 1.1 overview. W3C Recommendation (2012). <https://www.w3.org/TR/sparql11-overview/>
22. W3C: RDF 1.1 turtle: Terse RDF triple language. W3C Recommendation (2014). <https://www.w3.org/TR/turtle/>
23. W3C: RDF 1.1 xml syntax. W3C Recommendation (2014). <https://www.w3.org/TR/rdf-syntax-grammar/>
24. W3C: Json-ld 1.1 a json-based serialization for linked data. W3C Recommendation (2020). <https://www.w3.org/TR/json-ld11/>
25. Zeginis, D., Kalampokis, E., Roberts, B., Moynihan, R., Tambouris, E., Tarabanis, K.: Facilitating the exploitation of linked open statistical data: JSON-QB API requirements and design criteria. *SemStats* (2017)