Open Virtual Mobility

O3-A1.2: Competency directory requirements (backend and frontend) and O3-A1.3 Competency directory – functional prototype

- Final -

Outcome 3 Activity A1.2 & A1.3 Competency Directory and Matching Tool	
Document submission and review information	
Declared due date of deliverable	28.02.2019 agreed in partnership to: 31.03.2019
Reviewed due date of deliverable	31.03.2019
Actual submission date	12.03.2019
Organisation name of lead contractor	Beuth University of Applied Sciences, Berlin
Revision	1.3
Author and reviewer information	
Name of the author	Johannes Konert
Affiliation of the author	Beuth University of Applied Sciences, Berlin
Name of the reviewer	Diana Andone
Affiliation of the reviewer	UNIVERSITATEA POLITEHNICA TIMISOARA

The creation of these resources has been (partially) funded by the ERASMUS+ grant program of the European Union under grant no. **2017-1-DE01-KA203-003494***. Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.*

Imprint

Imprint: This publication is O3-A1.2 & O3-A1.3 of the Open Virtual Mobility Erasmus+ strategic partnership founded by the European Commission 2017 - 2020 under **2017-1-DE01-KA203-003494,** URL: https://www.openvirtualmobility.eu/

This paper is to present the requirement analysis and resulting requirements for the prototype of a competency directory of the Open Virtual Mobility project. The designed and implemented architecture is presented, including a REST-API that provides JSON-LD semantic data of all 33 competency entries of the project. Additionally, a user interface as Single Page Web Application is described, which allows end-users to browse and find competency definitions, e.g. to cross-reference them from Open Badge definitions.

PDF download

A full-text PDF of this report is available as a free download from:
https://www.openvirtualmobility.eu/

Social media

Find us on Twitter: @openVM_erasmus
Give us your feedback on social media with the following hashtag: #openvirtualmobility

Suggested citation

Konert, J. (2018) O3-A1.2/A1.3: Competency Directory requirements and functional prototype. Open Virtual Mobility Erasmus+ (2017-2020). Berlin, Germany. Retrieved from https://www.openvirtualmobility.eu/topics/outputs

Corresponding author

Johannes Konert
Beuth University of Applied Sciences
Luxemburger Straße 10, 13353 Berlin, Germany
johannes.konert[at]beuth-hochschule[dot]de

Table of Contents

Executive summary

The competency directory of the Open Virtual Mobility project consists of a REST-API (Level 2) which provides unique URL access to the 33 competency definitions and their cross-references as defined by Output O1 of the project. The data is available in JSON-LD format referencing the schema definitions (e.g. from ESCO). Additionally, the prototype implementation includes a user interface as Single Page Application, based on React.js, providing browsing and searching of the competency entries. As database behind, Neo4J is used. All software components are released as Open Source on GitHub repositories including docker container definitions and installation guidelines. The document describes results of the conducted survey, the list of requirements, and explains all resulting components of the implementation.

What are the objectives of this paper?

- Requirement analysis (survey based)
- Requirements definition
- State-of-the-art discussion of existing related work
- API-design for other systems to use/fetch open virtual mobility skill definitions
- Semantic scheme as a basis for cross-referencing (aligned to ESCO)
- User-interface (search and browse)
- Installation of the competency directory

Who is this paper for?

- Technicians interested in a.) using semantic competency definitions or b.) setting up an own competency directory.
- Pedagogues and didacticians interested in the technical infrastructure behind semantic competency definitions.

What topics are addressed in this paper?

Requirement analysis, result discussion, prioritization, list of requirements, related work discussion, Neo4J base, REST-API design, JSON-LD schema, example entries, user interface, docker containers, installation guideline

Contributors

- Johannes Konert, Beuth University of Applied Sciences, Germany
- Andreij Ternauciuc, Centrul Multimedia, Universitatea Politehnica Timisoara, Romania
- Marica Franchi, CINECA, Italy

Acknowledgements

Essential parts of this document are supported by work done in Bachelor thesis at Beuth University of Applied Sciences. Namely work is included from (Kiauka, 2019; Stoye, 2019). We thank the authors for their scientific contribution and open source code releases.

1. Aims and Scope

The semantic competency directory will provide an overview and (graph-based) search possibility open to all interested parties to find and browse existing definitions of virtual mobility skills. Additionally, the directory provides an API with unique URI to competency entries which provide JSON-LD based data to be used for referencing the competencies, e.g. from Open Badges.

The document provides an overview of the results of the conducted requirement analysis (based on an expert survey among project partners) and discussed their prioritization. Afterwards the system design, API concept and provided user interface is presented as well as the installation process. The overall aim is to provide a clear description of the current capabilities of the prototype, where they came from and how interested parties can use the competency directory and/or install an instance.

2. Requirement analysis

The collection of requirements to the competency directory were described in prior document O3-A1.1 (see Appendix *A1 Survey items for requirements analysis*). The survey among all partner organizations was conducted as an online survey based on Google Sheets between 12.03.-12.04.2018. Six participants responded, including lead of Outputs O3 (the author himself), O5 (Credentials and Gamification), O6 (OERs and MOOC), O7 (Quality Assurance and Sustainability). The responses are missing leaders of Outputs O1 (Competency Framework), O2 (Learning Hub), and O4 (Assessment).

2.1 Survey results

The resulting answers are encoded as [-1 I disagree to have that in the project, 0 not needed, 1 optional but useful, 2 needed uncritical, 3 definitely needed] and were summed up per item[1]. The result, ranked by sum, is shown in Table 1.

Table 1: Ranked requirements from survey. Colored in green are the candidates which are kept as requirements, in yellow are optional non-critical requirements, in red are those dropped.

ID	Requirement(s)	SUM
C1 → R1	list of all competencies	18
C2 → R2	search by keyword	17
C3	levels (as advancements in ~same~ competency)	17
C4 → R3	unique URL to individual competency	17
C5 → R4	multi-language support (other interfaces than English)	16
C6 → R5	direct access by ID	15

[1] The four Likert scale values have a number and a text value and are thus classified as ordinal values, but not strictly intervals. The calculation of sums is thus not a robust reliable measure to build upon. Therefore, interested readers find the raw answers in Appendix A2 Survey result table. The sum is used only as an indicator and basis for the sorting and discussion. We refrain from calculating standard deviations or distances.

C7 → R14	browsing by exploring search results …	15
C8 → R6	description (for each competency)	15
C9 → R7	cross-referencing of competencies (similarity, includes, ..) within the VM competency framework	15
C10 → R15	competency management (adding, editing, …)	14
C11	search by framework	14
C12	synchronization/copy of the competency fw. into Moodle	14
C13 → R16	direct editing of competencies (add, edit, remove)	13
C14	mobile interface support (small displays)	13
C15 → R8	search by existing language translation	13
C16	criteria definitions for a competency	13
C17 → R9	low latency on response to requests (<200ms)	13
C18 → R10	encrypted communication via HTTPS	13
C19 → R11	REST-Level 2 conform HTTP-API	13
C20 → R12	JSON-LD format for input/output	13
C21 → R13	multi-language support for competency definitions (same competency in x languages)	12
C22 → R17	reference editing (similar/contains/prerequisite ..)	12
C23 → R18	translation support (web interface)	12
C24	cross-referencing of competencies with other frameworks	12
C25	evidence definitions for a competency …	12
C26	identity management (login, credentials)	11
C27	read/write to an xAPI	11
C28	role management (for limited editing, reading of competencies)	10
C29	multiple competency framework support	10
C30	search by country of origin (of the framework)	10
C31	Service Level Agreement (SLA) as accessibility of a high level (95%)	10
C32	assertion management (sets of competencies per user)	9
C33	RDFa format for input/output	9
C34	import of complete other competency frameworks from different sources	8
C35	SOAP WebService conform HTTP-API	8

2.2 Discussion and prioritization

Table 1 was amended by an ID for easier reference in the text. All possible candidates for requirements are named C1-C25. The candidates which are copied as a requirement for the competency directory are given additional IDs R1-R18. Must-Have requirements of higher priority are marked green (R1-R13) and the requirements which are not essential and nice-to-have (lower priority) are marked yellow (R14-18).

Before the list of requirements R1-18 was finalized the raw survey results were presented in a transnational partner meeting in Timisoara (24.09.2018). For some of the items the rated importance and sum values were quite surprising, for the author of this document as well as for the partner members present. Likewise, for some items the range of ratings spread the full spectrum

from -1 (I disagree to have that in the project) to 3 (definitely needed), as listed in Appendix A2 Survey result table.
In the following all lower or non-prioritized candidates are briefly discussed.

C3: levels (as advancements in ~same~ competency)

C3 was rejected as the competency framework created for open virtual mobility skills (Output O1) contains no levels. Additionally, the European Skills/Competences, Qualifications and Occupations Framework (ESCO) has no levels in its scheme[2]. During discussion all partners agreed that instead of levels the (existing) concept of references of type "is a prerequisite of" is used.

C11: search by framework

The competency directory will merely contain competency definitions of one framework – the open virtual mobility competency framework. It could technically support cross-referencing to external frameworks (e.g. ESCO)(C24), but all organizations members at the meeting agreed that this is not a needed requirement for any of the outputs (O1-O7).

C12: synchronization/copy of the competency framwork into Moodle

As neither the eAssessment parts (O4) nor the OERs or MOOC sections (O6) will cross-reference directly to the competency areas or individual skill-sets of the open virtual mobility framework (from O1) there is no use for the project consortium of having a copy or synchronization of the competency definitions from the competency directory into Moodle competency lists.

C14: mobile interface support (small displays)

The competency directory will not be used by end-users, but primarily by project partners while defining open badges. During the badge design a screen with high resolution is used to fill all the references and text fields for a badge. In this scenario, from the found (C2) competency definitions (C8) the unique URLs are copied (C4/C6) into the `AlignmentObject`[3] field(s) of the Open Badge standard. No mobile (smartphone) support of the user-interface is required (even though most web-design approaches implement a mobile-first approach anyway).

C16: criteria definitions for a competency

Like C25 (evidence definitions for a competency) criteria are needed to decide whether or not an individual has shown the competency, and this can receive a certificate for it. The competency directory is meant to list competencies for open virtual mobility with as few context dependencies as possible. Listing of criteria to fulfill for a certain competency (or listing of evidence needed to be shown) is beyond that scope and should better be defined in the context (e.g. OER, MOOC or assessment) where the learner wants to proof his competency and needs to fulfill certain requirements (criteria to pass, evidence to show).

The following candidates are all related to editing competency definitions or extending the existing framework. All discussion partners agreed that this would be nice-to-have, but practically there is

[2] https://ec.europa.eu/esco/portal/escopedia/Skills_pillar , last accessed 05.03.2019

[3] https://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/index.html#Alignment, last accessed 05.03.2019

no need for editing functionality as the resulting competency framework from Output O1 has been created based on the group concept mapping with many experts and this the partnership will not easily change the definition.

C10 → R15 (competency management (adding, editing, …))
C13 → R16 (direct editing of competencies (add, edit, remove))
C22 → R17 (reference editing (similar/contains/prerequisite)
C23 → R18 (translation support (web)

All other candidates C26-C35) received a score below 12 and were agreed during the transnational partner meeting to be not considered for implementation.

2.3 Requirements

The derived list of requirements (without any priority indicated by the order) is:

R1 list of all competencies
R2 search by keyword
R3 unique URL to individual competency
R4 multi-language support (other interfaces than English)
R5 direct access by ID
R6 description (for each competency)
R7 cross-referencing of competencies (similarity, includes, ..) within the VM competency framework
R8 search by existing language translation
R9 low latency on response to requests (<200ms)
R10 encrypted communication via HTTPS
R11 REST-Level 2 conform HTTP-API
R12 JSON-LD format for input/output
R13 multi-language support for competency definitions (same competency in x languages)

The list of optional, nice to have functionality is:

R14 browsing by exploring search results …
R15 competency management (adding, editing, …)
R16 direct editing of competencies (add, edit, remove)
C16 criteria definitions for a competency)
R17 reference editing (similar/contains/prerequisite ..)
R18 translation support (web interface)

3. Related work

The former document for milestone O3-A1.1 about the scientific background contained a brief section about existing software solutions. This section included the Competency and Skills System (CASS) and the competency directory of the Open Badge Network Europe (OBN) project. Both are repeated here for a better overview. The list is amended by the neo4j graph database, as this technology finally proved to be the best basis for (re)implementation.

3.1 Competency and Skills System (CASS)

In 2016 the Competency and Skills System (CASS) started to work on an Open Source implementation[4] of a directory to host competency frameworks, identities (of learners) and assertions (competency profiles) while supporting several CFs in parallel (Robson, 2017). During 2017 the project improved its applicability by providing several example implementations for client systems (e.g. JAVA, C#, JavaScript)[5] and application plugins (e.g. Moodle, xAPI)[6] to connect applications to the directory. While CASS defines another different vocabulary to express and store CFs (CASS Project, 2018b) compared to ESCO, the project aims at being compatible with existing formats and vocabularies by using the concept of converters and adapters (CASS Project, 2018a). For example, an import of ESCO defined competencies and skills was already done. CASS supports JSON-LD as format to express competency frameworks, relations, identities and assertions in import and export.

CASS was developed to provide competence-based solutions for lifelong training, instruction and training of employees. Other software solutions should use CASS to collect and share information about learners' achievements and competencies. Thus, beside storing competency frameworks, CASS provides user accounts to connect indicators/assumptions and probabilities about existing competencies with such user accounts. Based on this, applications can use the CASS-API to request whether or not a person has probably a specific competency.

[4] https://github.com/cassproject/

[5] http://devs.cassproject.org/index.html?doc=1oPEFo7M5RHv9kB8DoVwCFBEcsctiYMpU-QDVcOyXIz8

[6] http://devs.cassproject.org/index.html?doc=1DZlrMrPd8Me2BsYHB0vUovtocWUMaj_VvKQD2_Ibb70

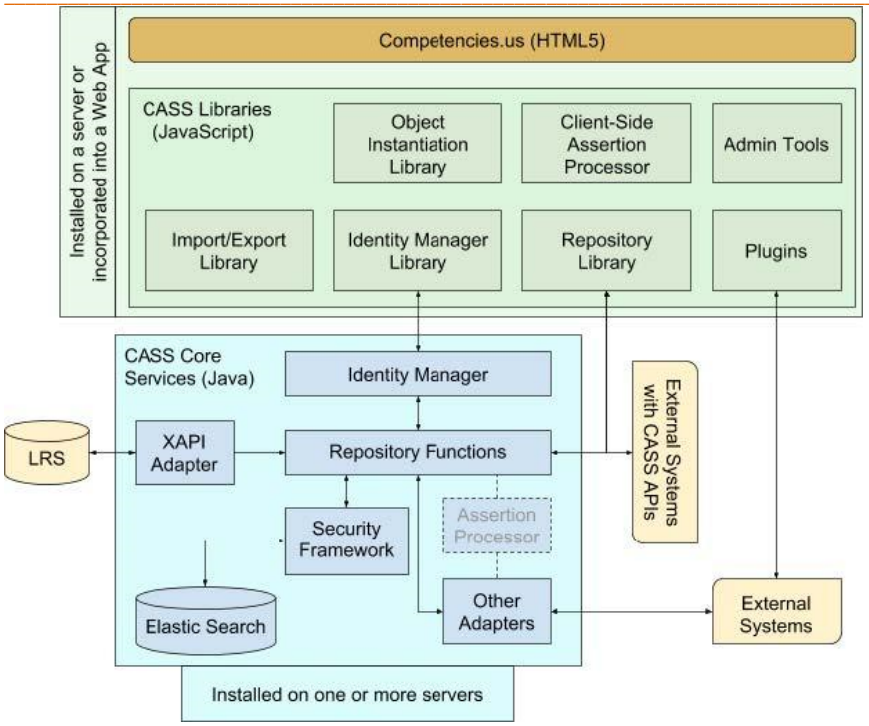


Figure 1: CASS architecture, here example competencies.us instance (Robson, 2017)

A CASS instance consists of the CASS Core Services, which mainly form the repository. On top of it operate the libraries, which can be integrated into web applications. The functionalities of inserting, modifying and deleting frameworks, skills and references are provided by the CASS Editor (Ray & Wood, 2019).

For evaluation reasons, an implementation of the open virtual mobility competency directory based on CASS was done between April 2018 and December 2018. The provided CASS Editor is complex and documentation is not easy to comprehend. Additionally, the current version is not stable. Moreover, the API delivered results very slowly (in our tests on average 3 seconds per competency entry). The achieved implementation was not satisfying due to bugs and issues with starting/running the CASS Core.

3.2 Competency Directory of OBN (copied from O3 A1.1)

For the Open Badge Network Europe Erasmus+ project (2015-2017) an implementation of a competency directory using semantic technology was realized (Konert, Buchem, Lewis, Hamilton, & Riches, 2017). The competency directory aimed for providing unique URLs and a simple web-interface for searching competency definitions in order to use the found unique URLs in Open Badge definitions. During the course of the project first an implementation for InLOC vocabulary (using RDFa and JSON-LD format) was provided, then after ESCO release, a re-implementation was

done for importing the ESCO competency definitions. In the end, prototypical implementations are available for InLOC and ESCO vocabulary[7].

Due to the fact, that this competency directory not yet supported JSON-LD format and the underlying database structure (mongoDB) seemed unsatisfying for the graph data of the competency framework, an extension of this implementation was abandoned.

3.3 Neo4J Graph database

Neo4J[8] is a graph database system which provides additional implementation examples for web applications and has its own Neo4J Browser, which is a web-application to query and visualize the database entries (see Figure 2 as an example). Neo4J offers a GPL v3 licensed community edition as open source[9].

Graph nodes can be labeled with properties (key-value pairs) as well as categorized into own category systems. Directed edges between nodes (called references) can as well be labeled with properties and weighted with a number value.

Even though it is not a relational system, a query language exists, called Cypher. Cypher allows filter queries to the database, e.g. for certain sub-graphs (parts) to match certain criteria. Cypher allows the creation of a REST-API to build own web-applications on top. (Jim Webber & Ian Robinson, 2018)

Based on the experiences with the former described implementations (CASS, mongoDB-based OBN implementation) the output leaders of O3 recommended a prototypical realization of the desired competency directory based on Neo4J. Future further extensions are expected to profit from the graph database layer used as foundation.

[7] https://github.com/openbadgenetwork/
[8] https://neo4j.com/product/
[9] https://neo4j.com/licensing/

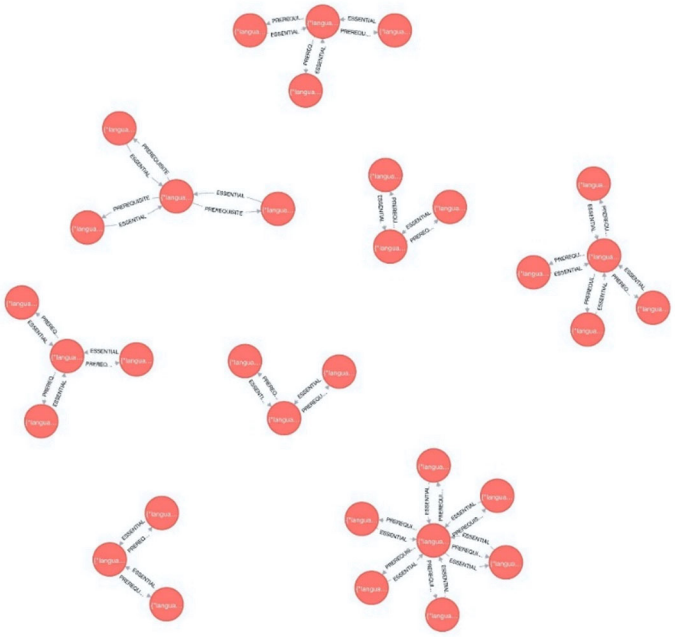


Figure 2: example of a graph visualization using Neo4J Browser (standard settings) here with the openVM skill and competency framework (existing of eight competency areas as visible). [Texts are not readable by intention as it is exemplary]

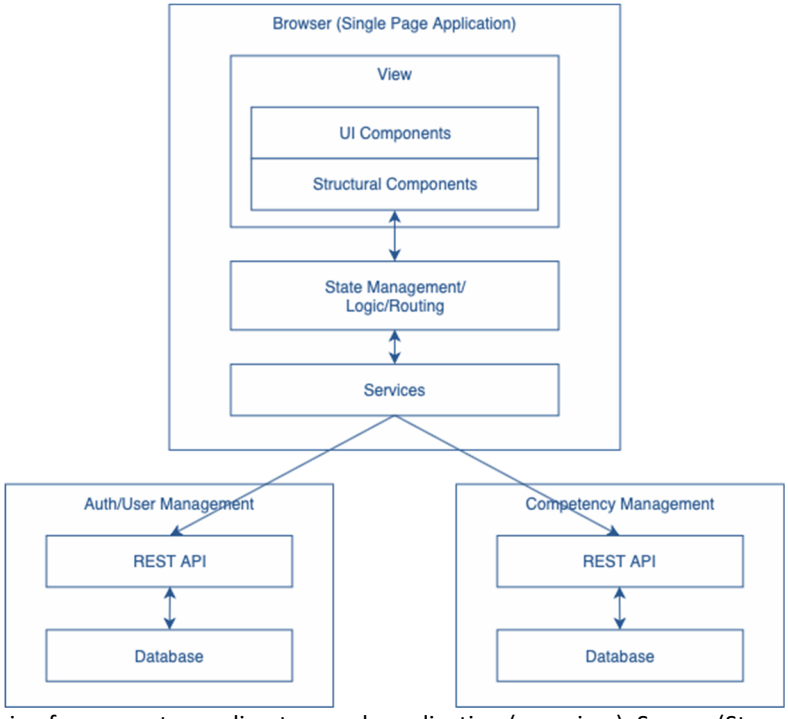


Figure 3: Application design for competency directory web application (overview). Source (Stoye, 2019, p. 25)

4. API-Design (backend)

The competency directory implementation based on Neo4J consist of two main parts: First, the backend with Neo4J database and REST-API to store and access the competency definitions and

references as a graph structure. Second, the frontend as a single page application based on React.js visualizing the user interface and offering e.g. search functionality (see architecture in Figure 3). The architecture is envisioned to be used for editing as well, as proposed in project plan, but currently editing of the final competency framework is not of major importance to the project consortium.

The REST-API for competency management has been designed as REST-API Level 2, containing unique URLs for resources and supporting the HTTP methods GET, POST, PUT and DELETE for the CRUD operations. Due to the consortium decision that editing is of lower priority, the API for authentication, user management and policies was postponed. Consequently, designed endpoints for manipulation of resources (POST, PUT, DELETE) fail due to missing authentication. Still, these are designed and implemented for easier extension with an editing functionality in near future.

REST-API Endpoints

- `/entries[/:id]` `GET`
- `/user[/:id]` `GET, POST, PATCH, PUT, DELETE` (implemented but unused)
- `/policies[:id]` `GET, POST, PUT, DELETE` (not yet implemented)
- `/auth` `GET, POST` (not yet implemented)

The endpoints provide access to the competency entries and their relations. Based on the ESCO-framework structure with the vision in mind to support future competency framework structured as well, entries can be of several skill types. For openVM only skill types *skill or competency* and *knowledge* are needed. The list of types is easily extendable.

Data model

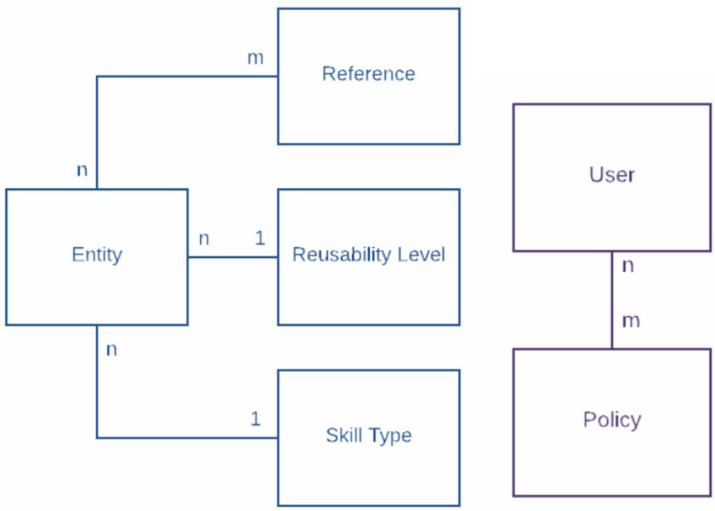


Figure 4: data model for entries which are of a certain skill type and have a reusability level (ESCO-related). entries can have multiple references (named) among each other. The user and right management is designed as a policy based structure where no direct relation is modeled between entries and users/policies.

The data model behind the REST-API models entries, which have a skill type. Skill types are skill or competency or knowledge. If in the future more competencies are defined and/or other competency frameworks are added, the list of types is easily extendable (e.g. by *occupations*). Each entry has a reusability level. Currently the reusability for all 33 entries is set to *transversal*.

References between entities are named, e.g. *needs as prerequisite* or *is essential subskill/part of*. The skill types available, the references and reusability level are all schemas based on definitions made by ESCO (European Centre for the Development of Vocational Training (Cedefop), 2019).

5. Semantic Scheme (ESCO-based)

The data is provided in a semantic format named JSON-LD (see format discussions in O3.A1.1 document). The linked data format provides references to schema definitions of all used attributes of the entries data. For schema elements provided by the openVM competency repository local reference data is given as URI, for the others external linked data URIs (mostly ESCO-references).

The semantic data is provided in a compact format or extended. See examples for both in the following listings.

Listing 1: compact JSON-LD answer accessing skill entry ID 8


```
{
  "@context": "http://cd.openvirtualmobility.eu/context/",
  "id": "http://cd.openvirtualmobility.eu/entries/8",
  "skillReuseLevel": "1 Transversal",
  "skillType": "Skill or Competence",
  "isEssentialPartOf": [],
  "isOptionalPartOf": [],
  "isSameAs": [],
  "isSimilarTo": [],
  "needsAsPrerequisite": [
    "http://cd.openvirtualmobility.eu/entries/10",
    "http://cd.openvirtualmobility.eu/entries/9",
    "http://cd.openvirtualmobility.eu/entries/11",
    "http://cd.openvirtualmobility.eu/entries/12"
  ],
  "prefLabel": {
    "language": "en",
    "value": "Interactive and collaborative learning in an
authentic international environment"
  },
  "altLabel": [],
  "description": {
    "language": "en",
    "value": "Interactive and collaborative learning in an authentic
international environment implies that the student develops teamwork skills ,
collaborates with peers across disciplines and contexts , acquiring new
international learning  experiences and interacting with authentic international
tools, systems, and resources in a foreign language"
  }
}
```


Listing 2: extended JSON-LD answer accessing skill entry ID28


```
[
  {
    "http://www.w3.org/2004/02/skos/core#prefLabel": [
```

```
        {
          "@language": "en",
          "@value": "Gaining cultural knowledge"
        }
      ],
      "http://data.europa.eu/esco/model#skillReuseLevel": [
        {
          "@value": "1 Transversal"
        }
      ],
      "http://www.w3.org/2004/02/skos/core#altLabel": [],
      "http://purl.org/dc/terms/description": [
        {
          "@language": "en",
          "@value": "Gaining knowledge about the culture they visit.Getting to know
other cultural - based perspectives of education"
        }
      ],
      "@id": "http://cd.openvirtualmobility.eu/entries/2",
      "http://cd.openvirtualmobility.eu/context/isEssentialPartOf": [
        {
          "@value": "http://localhost :6060/entries/1"
        }
      ],
      "http://cd.openvirtualmobility.eu/context/isOptionalPartOf": [],
      "http://cd.openvirtualmobility.eu/context/isSameAs": [],
      "http://cd.openvirtualmobility.eu/context/isSimilarTo": [],
      "http://cd.openvirtualmobility.eu/context/needsAsPrerequisite": [],
      "http://data.europa.eu/esco/model#skillType": [
        {
          "@value": "Skill or Competence"
        }
      ]
    }
]
```


The semantic JSON-LD data fetched from the REST-API is used by the Bestr system to cross-reference and display competency information from the Open Badges provided by the Open Virtual Mobility project. Currently only English is listed, even though the data format is capable of multi-language entries. (The definitions of competencies of Output O1 are only available in English at the moment.)

6. User-interface (frontend)

The user interface of the prototype for the competency directory is minimalistic and included all functionality listed by the requirements. As shown in Figure 5 and Figure 6 the directory provided a card-based layout. This is responsive and usable by mobile devices, too. The search functionality on top directly reduces the list of matching entries. Currently search fetches all entries from the databases (as these are only 33) and searches for matchings of the text in title and description. The details overlay shows the full description of a competency entry, the cross-references based on the reference types and the URL for this entry can be seen (and copied) from browser address bar.

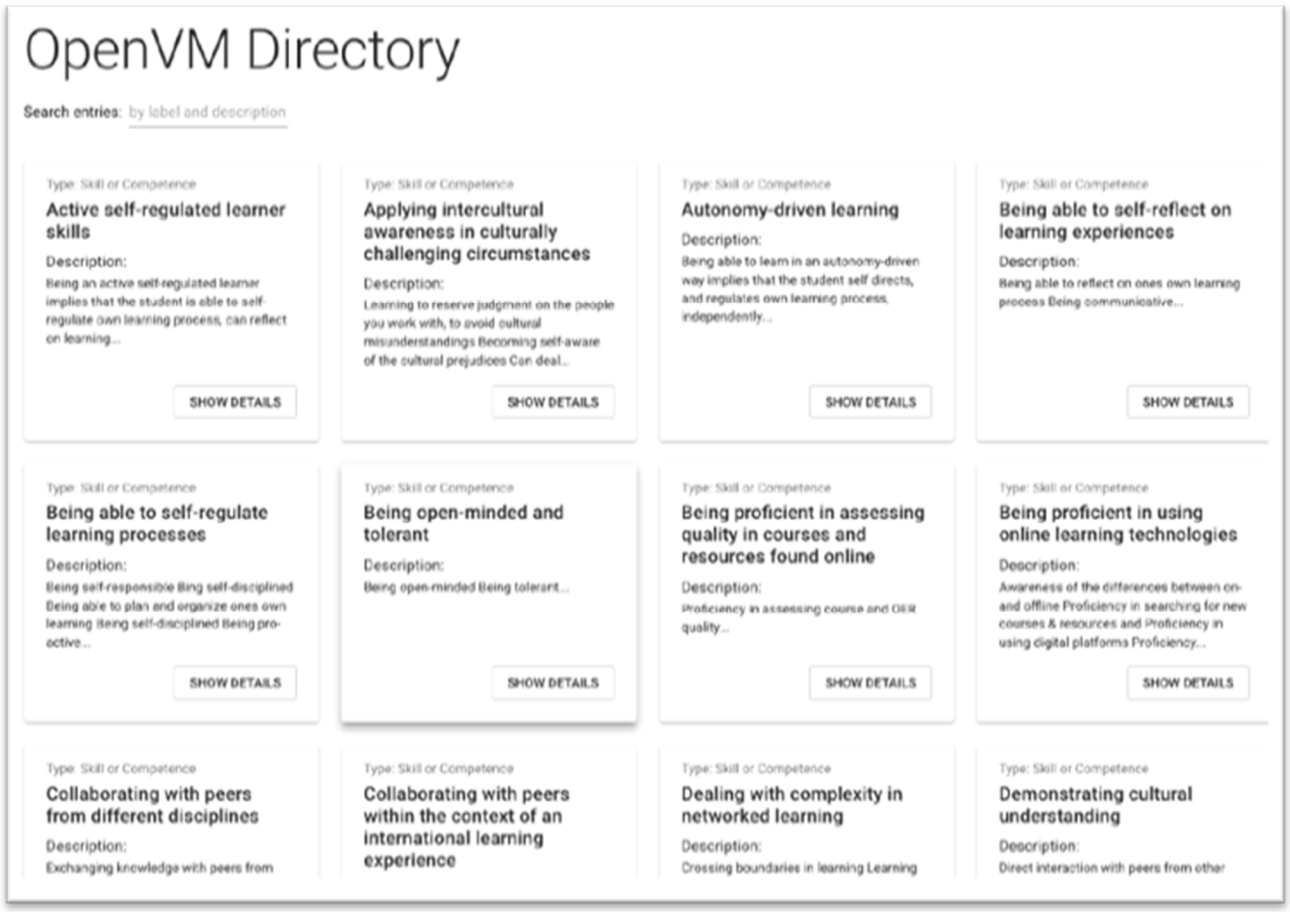


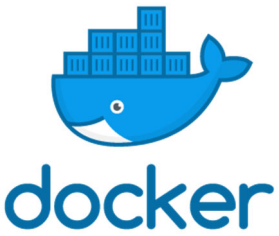
Figure 5: User interface using material design and a card-based layout. Search bar instantly reduced the displayed list of results.

Figure 6: In case one entry is opened the overlay displays all information and cross-links to other competency definitions. The URL in browser changes to the URL for this entry, too.

The user interface is currently only in English as the entries are in English. It is planned in cooperation with Output O1 leaders to ask participating project members for translations. In case these are possible (based on workload) the competency entries and user interface can be extended by multi-language support.

7. Installation and technical infrastructure

The technical realization uses virtualization based on Docker containers[10]. The three containers contain, a.) the database, b.) the backend with REST-API (using the database), and c.) a web server including the frontend (using the backend). A docker-compose script is provided to startup all three containers and connect them among each other by appropriate port-mappings.



Steps for installation

1. Clone or download the code repository, e.g. from GitHub[11]
2. If not available, install Docker[12] and docker-compose[13]
3. Run `docker-compose up` inside of the application root folder. The console shows the startup process of all three containers. The UI can then be found at http://localhost:3000 and the REST-API at http://localhost:6060.

8. Conclusion and outlook

The document provided a technical insight into the current functionality of the competency directory. The implementation has been realized using the Open Source edition of Neo4J graph database. Based on this a REST-API is provided for the entries which are delivered in JSON-LD format with semantic information. The provided cross-references to schema definitions, e.g. from ESCO allow fetching machines to interpret the data accordingly. Currently, all competency framework entries of the Open Virtual Mobility skills are provided in English only as no other languages are yet available (cf. Output 1). The data format is prepared for provision of multi-language results.

The API is planned to be used by Bestr platform to link corresponding competency definitions from text descriptions in Open Virtual Mobility Open Badges. Likewise, the API is used by the provided web-frontend application which allows listing, browsing, searching of the entries.

All requirements desired from the competency directory have been collected using an expert survey among all partner organizations to represent the insight of all output leaders in the data collection. The results were unbalanced and needed interpretation and discussion as done in section *2.2*

[10] https://www.docker.com/resources/what-container
[11] https://github.com/openVirtualMobility/competency-directory
[12] https://www.docker.com/
[13] https://docs.docker.com/compose/

Discussion and prioritization on page . Finally, 13 requirements were defined and six additional optional ones. From the list, encrypted communication using HTTPS has not been discussed in this document due to the postponed editing functionalities (R10, R15, R16, R17).

The prototype solution has been encapsulated in three Docker containers to allow easy setup and running of further instances of the competency repository. It is envisioned that soon, competency directories run on several servers worldwide and cross-reference their competency entries among each other. One major step in this direction is the provision of a running instance by the Open Virtual Mobility project as planned to be available via http://cd.openvirtualmobility.eu including all 33 competency definitions with their cross-references as defined and described by Output 1 documents.

Next steps after installation and maintenance are possible translations of the competency entries and multi-language support of the user interface.

9. References

References to tools and software are as footnotes directly in text.

CASS Project. (2018a). CASS Documentation - Import/Export. Retrieved March 13, 2018, from http://docs.cassproject.org/index.html?doc=1TPGedyaqxKIE3VgStHlCDFi379Ju2pj99GFjO4mIwGE#h.k3joxdhzdvbw

CASS Project. (2018b). CASS Schema. Retrieved March 2, 2018, from http://schema.cassproject.org/

European Centre for the Development of Vocational Training (Cedefop). (2019). Skills pilar - ESCO - ESCOpedia - European Commission. Retrieved March 12, 2019, from https://ec.europa.eu/esco/portal/escopedia/Skills_pillar

Jim Webber, & Ian Robinson. (2018). *A Programmatic Introduction to Neo4j* (1st ed.). Addison-Wesley Professional .

Kiauka, A. (2019). *Entwicklung einer Web-basierten Benutzerschnittstelle für semantische Graphen-basierte Daten am Beispiel von Kompetenzkatalogen*. Beuth University of Applied Sciences.

Konert, J., Buchem, I., Lewis, L., Hamilton, G., & Riches, T. (2017). Competency Alignment of Open Badges. In C. M. Stracke, M. Shanks, & O. Tveiten (Eds.), *Official Proceedings of the International WLS and LINQ Conference 2017* (pp. 29–36). Kristiansand, Norway: Logos Verlag GmbH Berlin.

Ray, R., & Wood, K. (2019). Competency and Skills Service -- Competency Framework Editor. Retrieved February 6, 2019, from https://github.com/cassproject/cass-editor/blob/master/README.md

Robson, R. (2017). *Competency and Skills System (CASS) Components*.

Stoye, J. (2019). *Development of an Application for a Competency and Skill Framework*. Beuth University of Applied Sciences.

Attachments

A1 Survey items for requirements analysis

The following questions were used for data collection, filled by one person per partner organization:

1. Please rate the following functionality of the **competency directory backend (API)** importance for your own work in the open VM project. [-1 I disagree to have that in the project, 0 not needed, 1 optional but useful, 2 needed uncritical, 3 definitely needed]
   - identity management (login, credentials)
   - role management (for limited editing, reading of competencies)
   - competency management (adding, editing, deleting of VM competencies)
   - multiple competency framework support
   - multi-language support for competency definitions (same competency in x languages)
   - assertion management (sets of competencies per user)
   - import of complete other competency frameworks from different sources
   - other: [name here what you would need]
2. Please rate the following functionality of the **competency directory frontend (web-interface)** importance for your own work in the open VM project. [-1 I disagree to have that in the project, 0 not needed, 1 optional but useful, 2 needed uncritical, 3 definitely needed]
   - list of all competencies
   - direct editing of competencies (add, edit, remove)
   - reference editing (this competency is similar/contains/prerequisite of another)
   - mobile interface support (small displays)
   - multi-language support (other interfaces than English)
   - search by keyword
   - search by framework
   - search by country of origin (of the framework)
   - search by existing language translation
   - direct access by ID
   - browsing by exploring search results (this implies to show links to connected other competencies in the frontend)
   - translation support (web frontend for language editing of competencies)
   - other: [name here what you would need]
3. What elements (attributes) does **~one~ competency need to support in the output format** for your work in the open VM project [-1 I disagree to have that in the project, 0 not needed, 1 optional but useful, 2 needed uncritical, 3 definitely needed]
   - description
   - levels (as advancements in ~same~ competency)
   - cross-referencing of competencies (similarity, includes, ..) within the VM competency framework
   - cross-referencing of competencies (similarity, includes, ..) with other frameworks
   - evidence definitions for a competency (how it has been proven by an individual)
   - criteria definitions for a competency (what measurable aspects exactly have to be fullfilled to have that competency)
   - other: [name here what you need]
4. Please rate the following **interoperability aspects** importance for your own work in the open VM project. [-1 I disagree to have that in the project, 0 not needed, 1 optional but useful, 2 needed uncritical, 3 definitely

needed]
- synchronization/copy of the competency frameworks into Moodle (from the CD to Moodle)
- Service Level Agreement (SLA) as accessibility of a high level (95%)
- low latency on response to requests (<200ms)
- encrypted communication via HTTPS
- unique URL to individual competency
- REST-Level 2 conform HTTP-API
- SOAP WebService conform HTTP-API
- JSON-LD format for input/output
- RDFa format for input/output
- read/write to an xAPI
- other: [name here what you need]

5. Describe in short words what of your work will use the directory (technical components and/or humans)?
6. Describe in short words what data and how often it will access (or send)?
7. Something else to mention?

A2 Survey result table

Table 2: Results of the survey per participant (incl. sum of all responses). In red bold are items that were dropped as requirements for implementation due to reasons explained in the text (e.g. out of scope).

Requirement(s)	SUM	P1	P2	P3	P4	P5	P6
list of all competencies	18	3	3	3	3	3	3
search by keyword	17	3	3	3	3	3	2
levels (as advancements in ~same~ competency)	17	3	3	3	3	3	2
unique URL to individual competency	17	3	3	3	3	3	2
multi-language support (other interfaces than English)	16	2	3	3	2	3	3
direct access by ID	15	3	3	3	2	2	2
browsing by exploring search results …	15	3	3	1	3	3	2
description	15	3	3	1	3	3	2
cross-referencing of competencies (similarity, includes, ..) within the VM competency framework	15	2	3	2	3	3	2
competency management (adding, editing, …)	14	3	1	3	3	3	1
search by framework	14	2	3	1	3	3	2
synchronization/copy of the competency fw. into Moodle	14	3	1	3	2	3	2
direct editing of competencies (add, edit, remove)	13	3	3	1	3	2	1
mobile interface support (small displays)	13	3	0	2	2	3	3

search by existing language translation	13	1	3	1	3	3	2
criteria definitions for a competency …	13	3	3	0	3	2	2
low latency on response to requests (<200ms)	13	2	3	1	3	3	1
encrypted communication via HTTPS	13	3	3	0	3	3	1
REST-Level 2 conform HTTP-API	13	2	3	2	3	2	1
JSON-LD format for input/output	13	2	2	2	3	3	1
multi-language support for competency definitions (same competency in x languages)	12	2	3	2	2	0	3
reference editing (similar/contains/prerequisite ..	12	2	3	1	3	2	1
translation support (web …	12	1	3	0	3	3	2
cross-referencing of competencies with other frameworks	12	2	3	1	2	2	2
evidence definitions for a competency …	12	3	3	-1	3	2	2
identity management (login, credentials)	11	3	0	0	3	3	2
read/write to an xAPI	11	3	1	0	3	3	1
role management (for limited editing, reading of competencies)	10	2	0	1	3	3	1
multiple competency framework support	10	3	1	1	3	1	1
search by country of origin (of the framework)	10	1	1	0	3	3	2
Service Level Agreement (SLA) as accessibility of a high level (95%)	10	2	2	0	2	2	2
assertion management (sets of competencies per user)	9	3	1	1	3	0	1
RDFa format for input/output	9	3	0	1	2	2	1
import of complete other competency frameworks from different sources	8	2	1	1	2	0	2
SOAP WebService conform HTTP-API	8	3	0	0	2	2	1