



# DOME 4.0

## Deliverable D2.3 – Data Sovereignty and Provenance System (also covering D2.4 – Data transaction and clearing system)

<b>Responsible Partner:</b>	SINTEF and UCL	2023.05.31
<b>Contributor(s):</b>	Bjørn Tore Løvfall (SINTEF), Jiawei Lai (CMCL), Chung Ting Lao (CMCL), Kok Foong Lee (CMCL), Alfredo Sanchez Garcia (SINTEF), Noel Vizcaino (UKRI), Adham Hashibon (UCL), Kristine Wiik (SINTEF)	2023.05.31
<b>Reviewer(s):</b>	Adham Hashibon (UCL), Martin Uhrin (EPFL)	2023.05.31
<b>Coordinator:</b>	CMCL Innovations	2023.05.31
<b>Dissemination Level:</b>	Public	
<b>Due Date:</b>	M30 (May, 2023)	
<b>Submission Date:</b>	31.May.2023	

## Project Profile

<b>Programme</b>	Horizon 2020
<b>Call</b>	H2020-NMBP-TO-IND-2020-twostage
<b>Topic</b>	DT-NMBP-40-2020 Creating an open marketplace for industrial data (RIA)
<b>Project number</b>	953163
<b>Acronym</b>	DOME 4.0
<b>Title</b>	Digital Open Marketplace Ecosystem 4.0
<b>Start Date</b>	December 1 <sup>st</sup> , 2020
<b>Duration</b>	48 months



This document is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953163. It is the property of the DOME 4.0 consortium and do not necessarily reflect the views of the European Commission.

## Document History

Version	Date	Author	Remarks
V0.1	2023-05-05	Bjørn Tore Løvfall	Initial version
V0.2	2023-05-10	Jiawei Lai, Chung Ting Lao, Kok Foong Lee	Added section on Hyperledger Fabric
V0.3	2023-05-12	Alfredo Sanchez Garcia	Added provenance model
V0.4	2023-05-18	Noel Vizcaino	PROV-O and SimPhoNy
V0.5	2023-05-22	Adham Hashibon	Update SimPhoNy, Add PROV-O-EMMO alignment, Add Provenance System
V1.0	2023-05-30	Adham Hashibon	Final draft combining D2.4

## Executive Summary

A system for logging and displaying the data provenance and sovereignty has been implemented. This system is using a Hyperledger Fabric blockchain technology to track where data originated, and what licenses apply to a given data set in addition to other relevant provenance information. Due to the increased resources needed to implement a Hyperledger based solution, a new technology was sought that enables rapid yet secure transactions without the complexity of the Hyperledger which proved to be an overkill for the purposes of data sharing in the context of DOME 4.0 eco system. In addition, due to overlap of base system for provenance and clearing house, one common deliverable is proposed that describes both the provenance and the clearing house systems.

Seeing the overlap in focus, content and timing between this report and deliverable D2.4 (Data transaction and clearing system) it was decided to combine both reports into a single document. All contents can be found in this report.

## Table of Contents

Executive Summary.....	2
Table of Contents.....	3
List of Figures .....	3
1. Introduction .....	4
1.1 Objective .....	4
2. Hyperledger Fabric Approach .....	9
3. W3C Provenance Data Model.....	12
3.1 PROV-O .....	12
4. The DOME 4.0 SimPhoNy OSP-CORE Approach.....	14
5. The DOME 4.0 Provenance Data Model .....	16
6. Conclusions / Next steps.....	17
7. Lessons learnt .....	18
8. Deviations from Annex 1.....	20
9. References .....	21
10. Acknowledgement .....	22
Annex 1 .....	23

## List of Figures

Figure 1: Example of permissioned blockchain network.....	10
Figure 2: A screenshot showing the speed at which a DOME search gets executed. Returning a result took over 30s in this case, in part because of the time spent accessing the Hyperledger service. ....	11
Figure 3: The basic classes and relations in PROV-O .....	13
Figure 4: Simple proof of concept showing how DOME 4.0 front end communicated to the SimPhoNy based new service in the bckend. The next step is to replace the messages with Json-LD serialised PROV-O based common universal data structures (CUDS).....	15
Figure 5: Main concepts, data and object properties of PROV-O relevant for DOME 4.0 .....	16
Figure 6: Example sequence of events for the DOME 4.0 provenance tracking .....	17

# 1. Introduction

This document reports on the activities of Task 2.3 of the DOME 4.0 project, named "Data Sovereignty and Provenance System" as well as the activities of Task 2.4 "Data Transaction Clearing and Reporting System (Lead: UCL; Partners: CMCL).

The main outcomes of this task are the actual data sovereignty and provenance system available through the DOME 4.0 platform (<https://dome.the-marketplace.eu>) and the foundations of the clearing house which rely on the provenance model intrinsically. The development of the data sovereignty and provenance system as well as the clearing house builds upon the overall architecture of DOME 4.0 that was portrayed in deliverable D1.3.

## 1.1 Objective

The data sovereignty and provenance system (D2.3) have significant commonalities with the clearing house (D2.4), and the data collected through the clearing house will be used to ensure that the provenance is tracked and vice versa. According to the W3C: "Provenance is information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness." [1]. The Clearing house is concerned with the automated clearing and tracking of all data transactions facilitated by DOME 4.0, hence it also requires, and in fact relies on a clear logging and provenance system. Due to the close connection between the two, we decided to create one single underlying base system and a joint report for logging the data needed for both systems. The responsibilities, however, are different, and the perspective of the clearing house coming from D2.4 part of this report are clearly highlighted.

The first requirements gathering for Task 2.3 was done as part of Task 2.1 and is part of deliverable D2.1 Technical requirements for data tools and services:

The main objective of this task is to provide a system that enables the users of the DOME 4.0 platform to track where data originated, and what licenses applies to a given data set in addition to other relevant provenance information.

A summary of the user stories is as follows:

- As a data consumer one can access the provenance of a piece of data, so that I can find the original source.
- As a data consumer one can determine the license for a piece of data, so that I can adhere to the terms of use.
- As a data consumer one can determine the jurisdiction of a piece of data, so that I can conform to the appropriate data handling laws.
- As a data provider one can determine who accessed my data, and when the data was accessed.

The main objectives of Task 2.4 "Data Transaction Clearing and Reporting System (Lead: UCL; Partners: CMCL) M7-M30]" of which D2.4 is a deliverable report is to **develop a data service for the automated and comprehensive clearing and tracking of all data transactions facilitated by DOME 4.0**. The service

will enable the fully automated **cross check** of data **sovereignty and provenance** for a requested data transaction to either accept or reject according to internal and external rules.

After a thorough analysis of the user stories and based on the architecture document (D1.3), and further user stories shown in Annex 1, it becomes clear that this requires the following assets to be available on the platform:

1. A reliable holistic **provenance** system: All transactions and in general actions regardless of their nature, on a data resource, is to be recorded in a secure log.
2. A clear set of **rules** for open data access: A reliable service that keeps track of the internal and external rules for data sharing. We emphasise that data sharing is not necessarily limited to free sharing (as in “free beer”) but in an open-source sense, where the data is to be shared on the basis of rules agreed between the data provider and data consumer (which is in general any agent, i.e., a human or a service etc).
3. A clear system of data **access** and **acquisition** rules that is easy to maintain and are consistent and themselves traceable (just like any other asset on DOME 4.0)
4. A reliable **process pipeline** that checks consistently each transaction and applies the provenance and rules above.
5. A clear way of **delegating** the application, or further acquisition of access rights to the data broker and other services on DOME 4.0

While point 1, the holistic provenance system is explained above as the portion of the D2.3 in this joined report, and provided already on the platform, the remaining points are the focus of ongoing efforts with initial proof of concept of the underlying service.

As a key distinguishing factor of DOME 4.0 is that it is based, from the ground up on semantic, ontology-based foundations, the set of rules for access and sharing should be ontologically maintained. Hence, we need initially an ontology for representing various users, agents, organizations, platforms, etc., preferably with a social network information (to augment the profile of a user with additional data relevant for trust and transparency) and user profiles. As the main authentication system used in DOME 4.0 is Keycloak, this ontology needs to be compatible with the OpenID standards used there as well. The most widely supported user profile and information ontology is the W3C FOAF ontology. While not entirely EMMO compliant, it can be integrated within EMMO in the same manner that DCAT has been integrated (see D3.1, and ongoing work in the ontology group of DOME 4.0). Moreover, there is a need to street line the above ontologies with the DOME 4.0 eco system ontology (see D3.2) which also includes agents as well as users; however, this does not provide the needed fine grained access rights and rules.

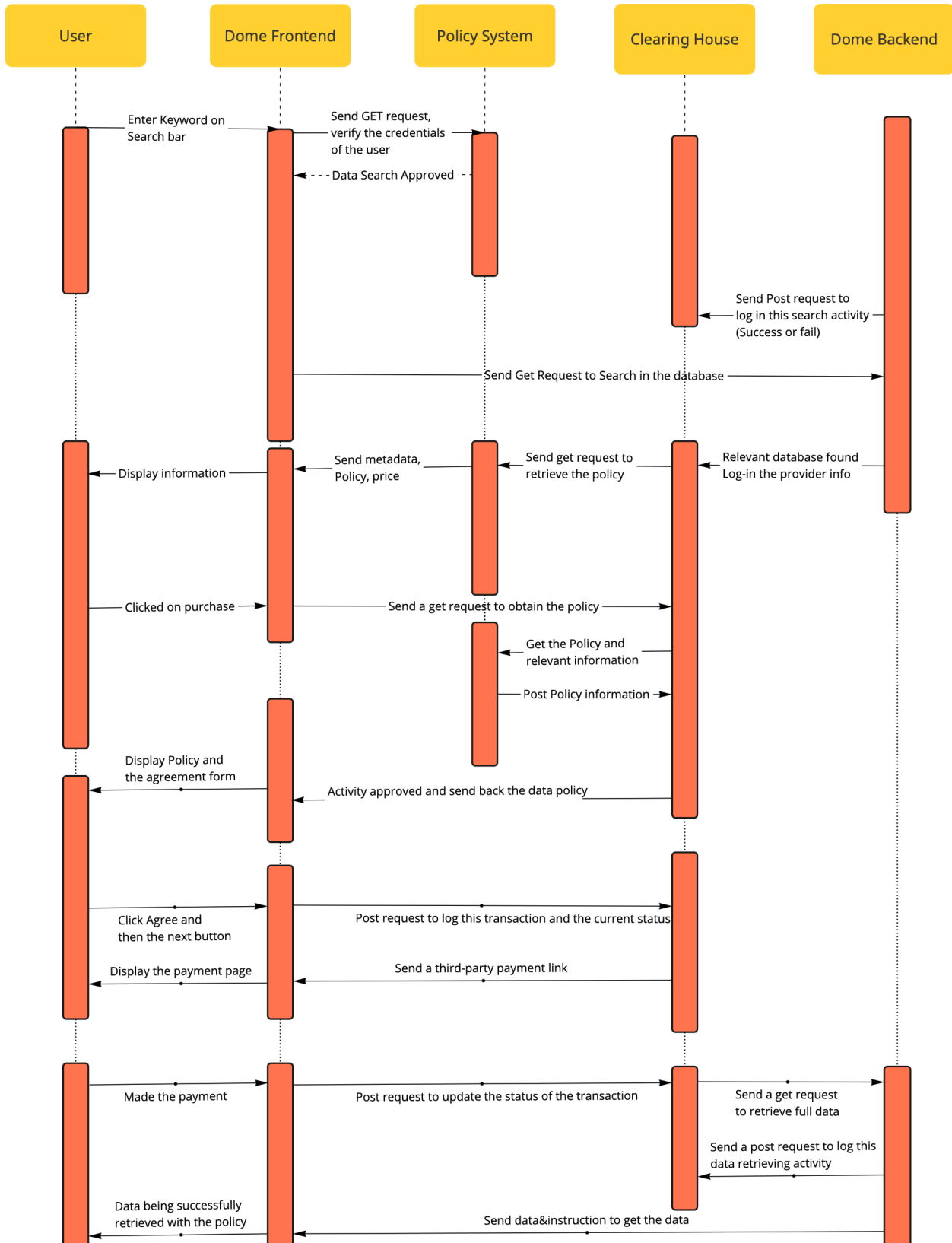
In this report, and for the implementation of the clearing service, a consolidated user profile and rules, access rights ontology is being developed that will integrate the above. In the meantime, the FOAF ontology is used as is with additional entities to cover the concepts needed for the compatibility with the Keycloak endorsed OpenID standard. These include the key concepts of Users, Usernames, Roles, Realms, Groups, identity token (A token that provides identity information about the user. Part of the OpenID Connect specification)) and access token (A token that can be provided as part of an HTTP request that grants access to the service being invoked on. This is part of the OpenID Connect and

OAuth 2.0 specification). This reporting of this ontology is out of the scope of this deliverable which will focus on the clearing house itself.

Furthermore, each data transaction facilitated by DOME 4.0 is tracked and logged by this service for reporting and invoicing. Logging of data transactions will be limited to minimally needed metadata for reporting and invoicing purposes and is sufficiently anonymised to protect the platform participant's privacy. The data transaction clearing and reporting system is developed in close collaboration with T2.3 and integrated into the platform in cooperation with T1.1. This task has releases as internal deliverable at M18 with continued improvements and releases up to the final deliverable at M30 (D2.4).

Once a basic ontology for users is implemented (as mentioned is currently based on FOAF and general added OpenID specific attributes) which focuses on the ontology terms needed for the minimal user metadata, where we initially developed a solution for a trusted clearing house dependent on the use of Hyperledger and block chain technology. Significant efforts were made into this implementation, which is currently implemented on the platform. A thorough description is given in the next chapters and while this provides well trusted and highly secure way of recording and keeping track of all transactions, the implementation proved to be an “overkill” for the purposes of DOME 4.0 within the TRL scope expected of it. Moreover, the use of blockchain and the need to reach consensus on every transaction including multiple players introduced huge demand on computational resources. Hence an alternative light weight approach is proposed and is being added to the platform as another option for providing both provenance and clearing house. This approach is based on utilising the SimPhoNy Open Simulation Platform core (OSP-CORE) that provides an ontology-based Python framework for managing large sets of semantically related data.

In either approach described below, the same pipeline for clearing house is implemented allowing seamless transition from the Hyperledger to the SimPhoNy OSP-CORE approach, where a similar API will be used in each case. The figure below shows the sequence of events ensuring that each transaction is logged, and checked for access rights and clearing before being authorised:





The key element of the implantation is to compare the access rights of a user and those of a data set, if a user attribute shows a signed access to the specific data set, permission to access the resource is granted, otherwise a session enabling the user to inspect and eventually initiate a contract to acquire the data set is performed.

Each record is stored along with a certificate generated using a public-private key encryption or as a block chain hash (depending on the approach). All data including the keys are stored in the SimPhoNy data space backend system for the SimPhoNy OSP CORE approach or in the ledger itself in the Hyperledger approach.

## 2. Hyperledger Fabric Approach

The first implementation of the logging system for the clearing house and the provenance and sovereignty of DOME 4.0 is a blockchain-based solution that utilises Hyperledger Fabric to manage data access, purchase, and distribution for data providers and consumers. This system offers secure, transparent, and reliable transactions, ensuring data providers can set access restrictions while consumers can purchase and download data in accordance with established rules, and all logs can be extracted from the system.

A decentralised and collaborative network is established to facilitate secure and transparent transactions by leveraging key features of Hyperledger Fabric, including smart contracts, consensus, world state, and transaction logs. A set of key functions is developed within the chaincode to manipulate ledger entries, enabling operations such as asset creation, user access management, and data querying.

In conjunction with the blockchain network, a clearing house service is implemented as a RESTful API, allowing various functions to be invoked through HTTP requests. This service interacts seamlessly with Hyperledger Fabric, providing a versatile interface for users. The information gathered by the clearing house will then be used by the provenance and sovereignty service to tell the consumer where the data came from and what terms and conditions were imposed on the data. At the same time a data owner can get access logs for the data they own, to have access to the information on who retrieved their data and when.

The README.md file in the project's GitHub repository offers detailed instructions for setting up the blockchain network and clearing house service, complete with necessary configurations and modifications. The guide also outlines the available functions and demonstrates how to interact with them through the RESTful API. This is currently hosted as a private repository, but to gain access please contact [info@dome40.eu](mailto:info@dome40.eu) with your GitHub username to gain access.

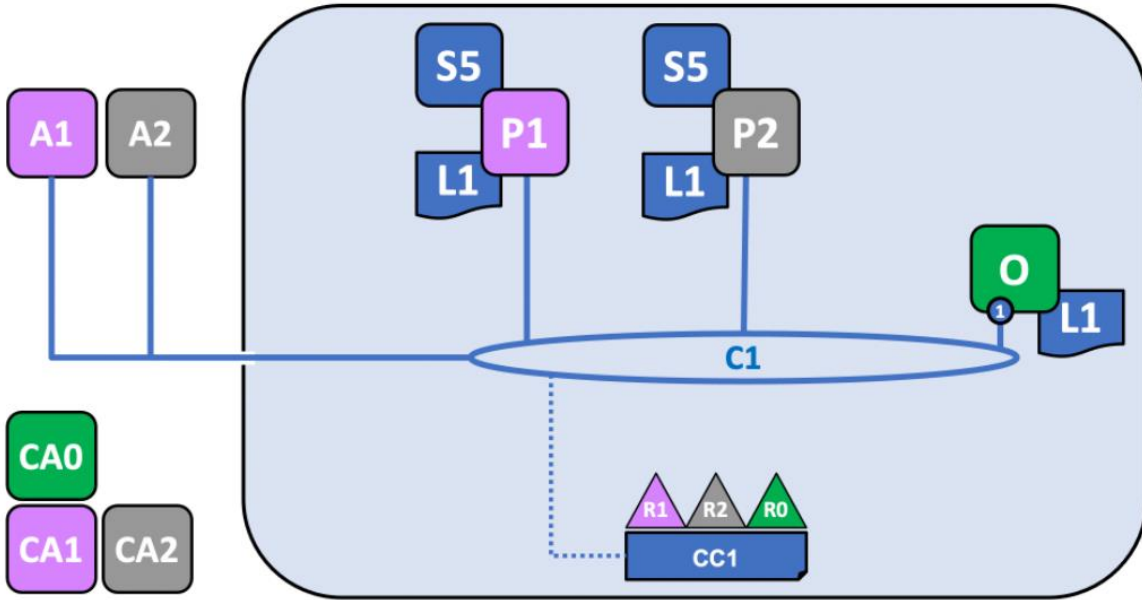


Figure 1: Example of permissioned blockchain network

The diagram showcases the permissioned blockchain network utilised in the Hyperledger project. The network consists of multiple organisations (R0, R1, R2), representing individual showcase owners, peers (P1, P2), and other components, each with its own set of peers, certificate authorities (CA0, CA1, CA2), and configurations (CC1).

In conclusion, the DOME 4.0 logging system successfully leverages Hyperledger Fabric to create a secure and transparent platform for managing data access, purchase, and distribution. This demonstrates the potential of blockchain technology for various industrial marketplace applications. The drawback of our implementation is that the calculations required for each transaction render the logging system a significant bottleneck when it came to the user experience, as the web platform became very non-responsive. Whether this is a limitation to our implementation, or if this is a general problem with blockchain technology we did not investigate further. With the limited time and resources available, we decided to investigate a more conventional logging system as an alternative. This is not to say that we will not revisit the Hyperledger Fabric in the future.

## Search Results

Keywords: carbon  
Creator: NIST

---

Keywords: C5H17AIN2O8P2

**Search**

**Filter** ⓘ

**Topic**

Topography

Sea vessels

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms	20.48 s	40.96 s
200	POST	dome.the-marketplace.eu	results	document	html	32.29 kB	32.04 kB			30192 ms
200	GET	cdn.jsdelivr.net	bootstrap.bundle.min.js	script	js	cached	0 B			0 ms
200	GET	code.jquery.com	jquery-3.2.1.slim.min.js	script	js	cached	0 B			0 ms
200	GET	cdnjs.cloudflare.com	popper.min.js	script	js	cached	0 B			0 ms
200	GET	maxcdn.bootstrapcdn.com	bootstrap.min.js	script	js	cached	0 B			0 ms
404	GET	dome.the-marketplace.eu	favicon.ico	FaviconLoader.jsm:1...	html	cached	207 B			0 ms

Figure 2: A screenshot showing the speed at which a DOME search gets executed. Returning a result took over 30s in this case, in part because of the time spent accessing the Hyperledger service.

## 3. W3C Provenance Data Model

In early stages of the development phase, the team investigated the potential applicability of the W3C Provenance Data Model (PROV-DM) as a tool for developing a provenance system for DOME 4.0.

The W3C PROV-DM is a standard that offers an extensible and formal framework for capturing and expressing provenance information in a machine-readable format. The key concepts in PROV-DM include entities, activities, agents, and the relationships between them. Entities represent things that exist in the world, such as files, documents, and datasets. Activities represent processes or events that manipulate entities, such as data transformations or document revisions. Agents represent the software, organization or person that initiate or control activities. Relationships such as `used`, `wasGeneratedBy`, and `wasDerivedFrom` connect these concepts to each other and describe their dependencies and dependencies of the entities they relate to. PROV-DM also provides a set of constraints and rules that ensure the consistency and validity of provenance information. For example, it defines rules for asserting the identity of entities, ensuring that activities are properly initiated and controlled by agents, and tracking changes in entity states over time.

Following a thorough evaluation, it was decided not to include PROV-DM directly into the DOME 4.0 platform as PROV-O is designed to be a system allowing the entire data lifecycle to be recorded. DOME 4.0 acts as a common platform that is neither expected to maintain nor to access the entire provenance of data sets that are essentially stored on remote third-party servers. The DOME 4.0 user stores (see Section 1.1 is concerned mainly with the source (as in: which platform provides access), license, and data handling information, as well as who got access to the set through the DOME 4.0 platform. As a result, only a small subset of PROV-O ontological keywords are needed.

### 3.1 PROV-O

The PROV Ontology (PROV-O) expresses the PROV Data Model using the OWL2 Web Ontology Language (OWL2) which is considered a higher semantic accuracy standard. This is the de facto scientific standard for provenance due to its widespread adoption.

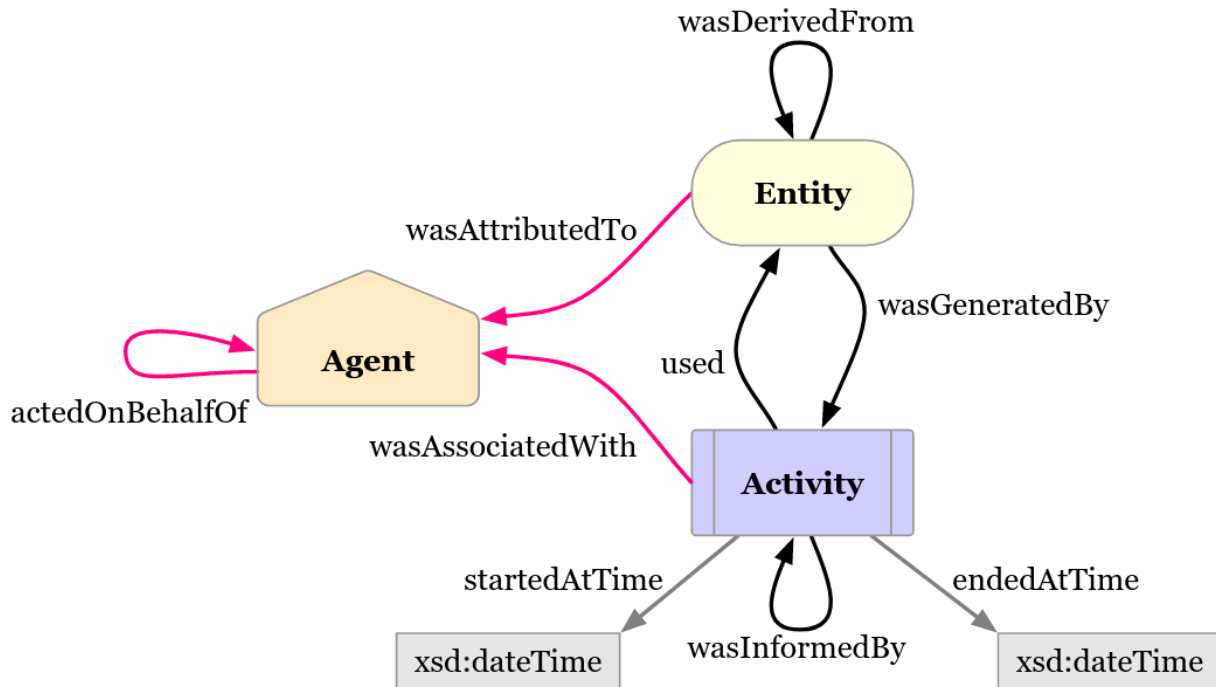


Figure 3: The basic classes and relations in PROV-O<sup>1</sup>

Any `dcate:Dataset` is also a `prov:Entity`. Having multiple '@types' is considered a good practice.

A simple solution would be to keep immediate dataset changes (I.e. closer nodes in the graph). Thus, 'wasDerivedFrom' points to the permanent identifiers (PIDS, usually URIs), a JSON list of the dataset parents.

'wasGeneratedBy' holds the PID list of our custom defined `prov:Activities`, with timestamps along with other metadata.

Persons, Organizations or Software are the Agents responsible for the Activities. A summary of this can be found in Figure 3.

The result is a recorded network (actually a Digraph, I.e., a directed graph, where the arcs joining two nodes have a direction that is semantically significant) within the metadata. We can harvest this highly structured log for many purposes. For instance, we may want to traverse the tree to recreate the history of changes of a dataset.

E.g, if we want the list of datasets generated by an entity we could use:

- A Python loop over a dataset list.
- A standardized query to the RDF dataset holdings in the knowledge graph
- The database query doing the heavy lifting and then the final touches with a Python API.

<sup>1</sup>source: <https://www.w3.org/TR/2013/REC-prov-o-20130430/>

Metadata holding can be progressively adapted to support it. Albeit not a priority, it is good to keep it in mind as it may simplify future solutions. The key here is about expecting to have such metadata while designing or updating software components.

Within the Python ecosystem there are libraries to work with PROV-O, e.g., to import PROV-O into NetworkX, opening new analytics, data conversion and visualisation options. Additionally, PROV-O can be directly imported as an ontology name space into the DOME 4.0 back and front ends which will allow to directly use it to decorate the provenance of a data set. In the future we will investigate ways of adding analytics to SimPhoNy via NetworkX or similar when the number of data sets and their provenance becomes substantial to assist in the actual analysis of the provenance.

## 4. The DOME 4.0 SimPhoNy OSP-CORE Approach

This is a novel alternative way to both the provenance and clearing house implementations that promises to be both efficient and easier to maintain. SimPhoNy is an open simulation platform (OSP) framework that is based, from the ground up, on an ontological framework [7,2,4]. We are focused here on one of the main components of SimPhoNy, namely its ability to provide semantic, built in provenance model out of the box essentially, as each data structure generated using the system is by definition connected to a unique metadata based on ontology such as PROV-O which caters for the complete log of the data structure. SimPhoNy initially targeted interoperability and integrate-ability between computational materials simulation tools. However, in its core is the OSP-CORE component that focuses on providing semantic data management based and ontology infused knowledge graphs. Essentially SimPhoNy OSP-CORE relies on knowledge graph semantic representation of data entities and relations based drawn directly from an ontology to cater for interoperability. A key feature of SimPhoNy is that it requires an ontology in the first instance to define the data itself and of course its relations. Details of the SimPhoNy DOME 4.0 extended data structures will be in the scope of D3.6. We note that SimPhoNy was developed initially by the group of Prof. Hashibon while at Fraunhofer and is currently developed jointly between UCL and Fraunhofer and is an open-source project with a permissible, industry friendly license.

This SimPhoNy framework provides a novel way to organise and programmatically work with ontology in general and web resource description framework (RDF) namespaces (which are triplets) directly under a coherent Python interface (and in the near future JavaScript for web friendly applications). It comes with a built-in support for a number of available ontologies, including,

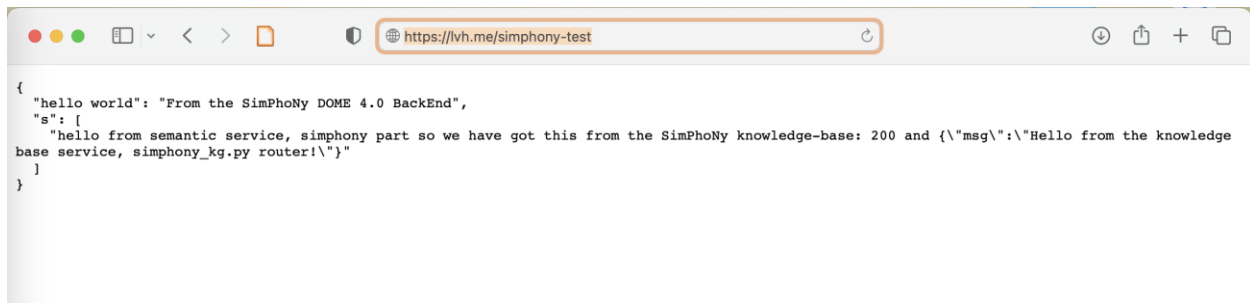
- [Elementary Multiperspective Material Ontology \(EMMO\)](#)
- [Dublin Core Metadata Initiative \(DCMI\)](#)
- [Data Catalogue Vocabulary \(DCAT\)](#)
- [Friend of a Friend \(FOAF\)](#)
- [The PROV Ontology \(PROV-O\)](#)
- [Simple Knowledge Organization System \(SKOS\)](#)
- [The City ontology](#)

Other ontologies are simple to add. Notably the PROV-O ontology is among the out of the box supported ones and hence eases the deployment in DOME 4.0 in particular for the provenance model. In particular

it can support the DOME 4.0 data set ontology and when combined with PROV-O and DCAT ontology provides a remarkably efficient knowledge graph based framework for the DOME 4.0 platform requirements. These DOME 4.0 data set ontology (DSO), DCAT, and PROV-O standards are most foundational for Knowledge representation aims. The SimPhoNy OSP-CORE utilises RDFlib [5] and thus supports its RDF serialisations as well as all features of RDFlib, including the SPARQL end points. Moreover, SimPhoNy OSP-CORE provides a built in Python native tools to manipulate and convert ontology relations and entities into data structure python classes. The key feature of SimPhoNy is that the data structures are one to one compliant with the notion of ontological individuals Moreover, RDFlib is augmented along with PyLD enabling state of the art efficient libraries to work with generic semantic RDF data in Python based on de facto standards.

The framework abstracts the databases and provides uniform wrappers for data access. There are other features, like ontology management that enable the uploading of existing ontologies. More is given in the deliverable D3.6 (CUDS classes).

While the solution with the Hyperledger is already fully implemented, the solution using SimPhoNy is a recent activity started due to the bottle necks posed by the Hyperledger. An Existing proof of concept prototype implementation for storing and retrieving SimPhoNy common universal structures already exists and will be available shortly on the DOME Git repository after code clean-up (see figure 4 for a rudimentary screen shot). The PROV-O powered common universal data structures (CUDS) of SimPhoNy once bult with PROV-O and the DS ontology of DOME will have – built in – the needed metadata to cater for provenance and enable efficient clearing house.



```
{
  "hello world": "From the SimPhoNy DOME 4.0 BackEnd",
  "s": {
    "hello from semantic service, simphony part so we have got this from the SimPhoNy knowledge-base: 200 and {\\"msg\\":\\"Hello from the knowledge base service, simphony_kg.py router!\\"}"
  }
}
```

Figure 4: Simple proof of concept showing how DOME 4.0 front end communicated to the SimPhoNy based new service in the bckend. The next step is to replace the messages with Json-LD serialised PROV-O based common universal data structures (CUDS).



## 5. The DOME 4.0 Provenance Data Model

As the scope of DOME 4.0 is not to entirely record the provenance state of each data set or asset along its entire lifecycle but to be able to record the relevant access through and on the DOE 4.0 platform (see D1.3), only a small subset of keywords are needed, these can be mapped to general DOME 4.0 EMMO based ontology and PROV-O. Similar to work done on aligning DCAT ontology to DOME as described in D3.1, the main concepts of Provo (Figure 3). Figure 5 below shows the main concepts, Data and Object Properties (connecting individuals to data and to other individuals, respectively) will be supported in the DOME 4.0 Provenance service.

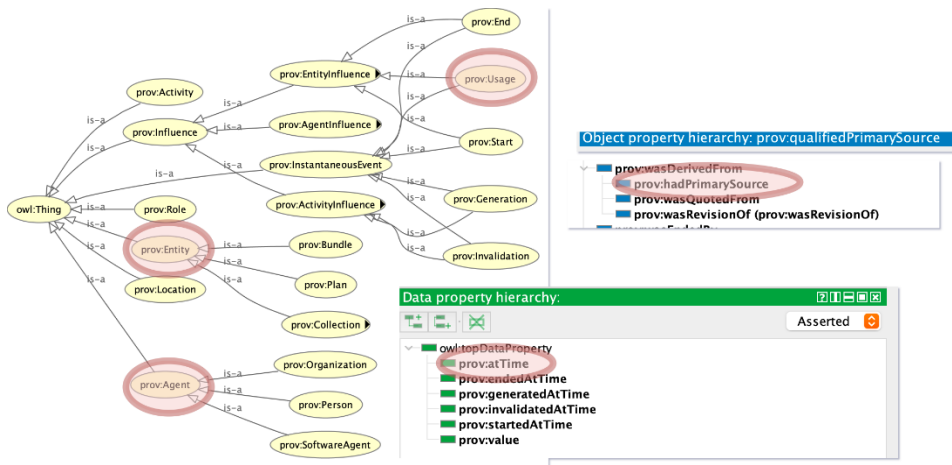


Figure 5: Main concepts, data and object properties of PROV-O relevant for DOME 4.0

Each activity in the backend of DOME 4.0 that pertains to any access to data, including passing information about data sets using the connector, searching the knowledge service etc. will automatically create the relevant provenance data sets and record it directly into the provenance and clearing house (see D2.4). Figure 6 demonstrates an example sequence of events. Once a user requests a certain access to a data set, a service in the backend is triggered to augment the data returned to the user as well as the internal log and clearing house register with the proper access metadata based on the Prov-O integrated ontology concepts. These are the `prov:Entity` which is mapped to EMMO and DCAT data set, `prov:Agent` which is mapped to the DOME 4.0 eco system ontology Platform Concept etc. Whenever more confidential implementation is needed, use of salt hashing can be a less sophisticated alternative to blockchain technologies as it will hash the relevant information and offer more security. However, this is not a real alternative to the more computationally demanding Hyperledger architecture using full block chain encryption. The use of salt hashing, however, enables DOME 4.0 to reach a TRL of 3-4 in a short time, whereby the task of upscaling the hashing can be deferred to later stages depending on market demands.

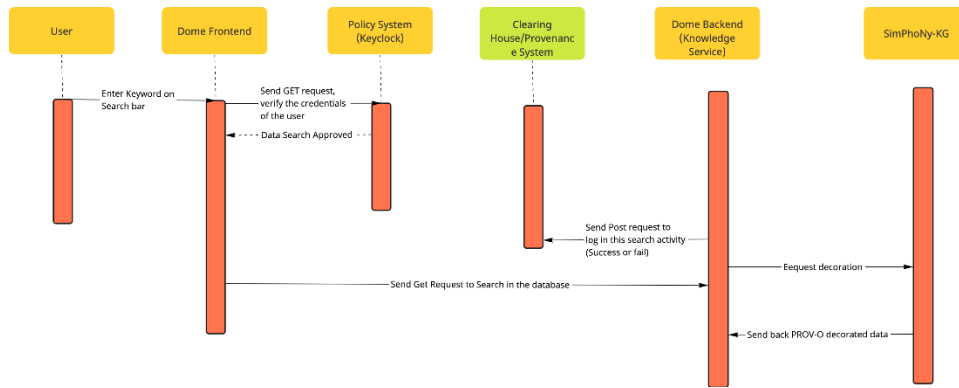


Figure 6: Example sequence of events for the DOME 4.0 provenance tracking

## 6. Conclusions / Next steps

A system for logging and displaying the data provenance and sovereignty has been implemented. The first approach, using a Hyperledger Fabric blockchain technology proved to be prohibitively slow, impairing the user experience on the DOME 4.0 platform. The inherent security feature of the Hyperledger Fabric technology is appealing, but at the relatively low TRL level of the DOME 4.0 platform, we made the compromise of implementing a different technology that has similar functionality but is faster. Should we decide to change this decision in the future, this would be possible, as the blockchain was working on the platform, and the new implementation will be using Similar API’s. A new technology based on SimPhoNy is under heavy development and already showed promising results for 1) built in provenance and 2) straight forward clearing house implementation.

## 7. Lessons learnt

This task had two challenges that is worth mentioning in this section. First, we developed a logging system based on a Hyperledger blockchain, but even though it did the job, the user experience was not satisfactory as the site become prohibitively slow. Whether this was due to our knowledge of the technology, or a problem with the technology, we did not investigate, as we found it rather complicated and went for a simpler solution.

The second lesson learnt is that we realized that a lot of aspects of provenance can be tracked, and to be sure to capture everything, we designed a very generic logging system. The provenance and sovereignty will then be a specific view on the collected data. What data is collected, and what view will be beneficial in the final version of DOME 4.0 must be left to mature with the platform.

The third lesson learnt is that often the simple, in house solutions are the most easy to use and efficient as they are designed from the onset to the problem at hand, at the same time such an implantation based on in house software should adhere to standard software interfaces (e.g. use of PROV-O and DCAT where possible) to allow interoperability and alleviate the dangerous of vendor lockdown.

Another lesson learned is that we realized that a lot of aspects of clearing house and provenance can be tracked, and to be sure to capture everything, we designed a very generic logging system. A solution that is based on state-of-the-art novel EU ontology frameworks provides a much more efficient yet secure solution which in the context of DOME 4.0, being a central authentication point, is much preferable from a user perspective, and hence is essential for the success of the platform.



## 8. Deviations from Annex 1

There are no deviations from Annex 1.

## 9. References

- [1] <https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/>
- [2] OSP core. SimPhoNy. (2023). <https://github.com/simphony/osp-core>
- [3] Hashibon, A., et al. "Common universal data structures (CUDS) and vocabulary in the SimPhoNy integrated framework." (2015).
- [4] Adler, Joan, et al. "Visualization in the integrated SimPhoNy multiscale simulation framework." *Computer Physics Communications* 231 (2018): 45-61.
- [5] Boettiger C (2018). *rdflib: A high level wrapper around the redland package for common rdf applications*. doi:10.5281/zenodo.1098478, <https://doi.org/10.5281/zenodo.1098478>.
- [6] The Friend of a Friend Ontology: An ontology is used to describe people and social relationship on the Web. <http://www.foaf-project.org/>
- [7] [https://www.keycloak.org/docs/latest/server\\_admin/](https://www.keycloak.org/docs/latest/server_admin/)
- [8] Hashibon, A., et al. "Common universal data structures (CUDS) and vocabulary in the SimPhoNy integrated framework." (2015).

## 10. Acknowledgement

The author(s) would like to thank the partners in the project for their valuable comments on previous drafts and for performing the review.

Project partners:

#	Type	Partner	Partner full name
1	SME	CMCL	Computational Modelling Cambridge Limited
2	Research	FHG	Fraunhofer Gesellschaft zur Förderung der Angewandten Forschung E.V.
3	Research	INTRA	Intrasoft International SA
4	University	UNIBO	Alma Mater Studiorum – Università di Bologna
5	University	EPFL	Ecole Polytechnique Federale de Lausanne
6	Research	UKRI	United Kingdom Research and Innovation
7	Large Industry	SISW	Siemens Industry Software NV
8	Large Industry	BOSCH	Robert Bosch GmbH
9	SME	UNR	Uniresearch B.V.
10	Research	SINTEF	SINTEF AS
11	SME	CNT	Cambridge Nanomaterials Technology LTD
12	University	UCL	University College London



*This document is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 953163. It is the property of the DOME 4.0 consortium and do not necessarily reflect the views of the European Commission.*

## Annex 1

### 10.1.1.1 Use case 1: Data provider making data available on DOME 4.0

<b>Description</b>	A data provider wants to make his/her data available on the platform and set the rules on who can access, and the price.
<b>Actors/Personas</b>	Data provider
<b>Preconditions</b>	1. Data is in the format accepted by the DOME 4.0 platform
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>Record this dataset in the clearing house database.</li> <li>Record purchase options for this dataset (maybe different price tiers for different usages)</li> <li>Records restrictions on previewing this data</li> </ol>
<b>Workflow</b>	<ol style="list-style-type: none"> <li>Data provider uploads data to the platform and sets the restrictions on the data, how much can be previewed by users who have not purchased the data</li> <li>Data provider sets the price for his/her data</li> <li>Maybe the data provider can give certain groups of users “free” access to the data without purchase through the clearing house. This then requires the data provider give further specification on the portion of the data that can be viewed “freely”.</li> </ol>
<b>Alternative workflow</b>	
<b>Exceptions</b>	If the rules cannot be accepted by the clearing house, return an error message which includes any modifications that are needed? For instance, the data provider should give different types of credentials for the data, or more information about the data (the quality of the metadata).
<b>Non-functional requirements</b>	

### 10.1.1.2 Use case 2: Data consumer purchases right to access data

<b>Description</b>	A data consumer previewed the data and wants to purchase rights to access the data.
<b>Actors/Personas</b>	Data consumer, data provider
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>-Data provider has already made the data available on the platform</li> <li>- The data consumer fulfilled the requirements that are identified by the data provider. (Terms and condition box should be ticked)</li> </ul>
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>Data consumer and software agent gain rights to access a particular dataset. <ul style="list-style-type: none"> <li>• Adds an additional metadata to the users’ profile in the database, saying this user has access to this dataset for a certain period of time agreed by the data provider.</li> </ul> </li> <li>Data provider receives money securely</li> </ol>
<b>Workflow</b>	1. Data consumer found a data he/she wants to buy



	<ol style="list-style-type: none"> <li>2. Clicks purchase (I'm assuming that the only condition to access data is by money, not restricted by any identities etc.)</li> <li>3. Use third party tool to charge consumer's credit/debit card and transfer funds to data provider</li> <li>4. Data consumer can specify which software agent can access this dataset (probably for a showcase), probably restrict the IP address of the agent</li> </ol>
<b>Alternative workflow</b>	
<b>Exceptions</b>	
<b>Non-functional requirements</b>	<i>Non-functional and special requirements for the system.</i>

### 10.1.1.3 Use case 3: Data consumer downloads data

<b>Description</b>	Data consumer wants to download a dataset that he/she has access either through purchase or agreement with the data provider
<b>Actors/Personas</b>	Data consumer
<b>Preconditions</b>	Data consumer has rights to access data
<b>Postconditions</b>	Data downloaded to consumer's computer. Clearing house database contains information on this data retrieval.
<b>Workflow</b>	<ol style="list-style-type: none"> <li>1. Data consumer selects data he/she wants to download from the front end</li> <li>2. Clearing house checks user database whether this person has access rights to the data</li> <li>3. Approves download</li> <li>4. Logs this download in the database</li> </ol>
<b>Alternative workflow</b>	-
<b>Exceptions</b>	Give rejection message if consumer does not have access rights
<b>Non-functional requirements</b>	<i>Non-functional and special requirements for the system.</i>

### 10.1.1.4 Use case 4: System admin to check transactions

<b>Description</b>	System admin wants to check the list of transactions/downloads were approved/rejected
<b>Actors/Personas</b>	System admin
<b>Preconditions</b>	-
<b>Postconditions</b>	System admin gets the data he/she wants
<b>Workflow</b>	<ol style="list-style-type: none"> <li>1. System admin makes a query through the interface, probably REST api</li> </ol>
<b>Alternative workflow</b>	-
<b>Exceptions</b>	

<b>Non-functional requirements</b>	<i>Non-functional and special requirements for the system.</i>
------------------------------------	--

#### 10.1.1.5 Use case 5: Software agent requests a dataset

<b>Description</b>	A software agent from a showcase wants to download a dataset
<b>Actors/Personas</b>	Software agent
<b>Preconditions</b>	Software agent needs to have access rights, either obtained when a data consumer makes the purchase, or set explicitly by the data provider
<b>Postconditions</b>	Software agent obtains dataset, clearing house logs this data retrieval
<b>Workflow</b>	<ol style="list-style-type: none"> <li>1. Software agent makes a GET request on this dataset through the platform's REST api</li> <li>2. Clearing house checks whether this software agent has access rights to the dataset</li> <li>3. Approves download</li> <li>4. Assign a transaction ID and record in the database</li> </ol>
<b>Alternative workflow</b>	
<b>Exceptions</b>	
<b>Non-functional requirements</b>	<i>Non-functional and special requirements for the system.</i>