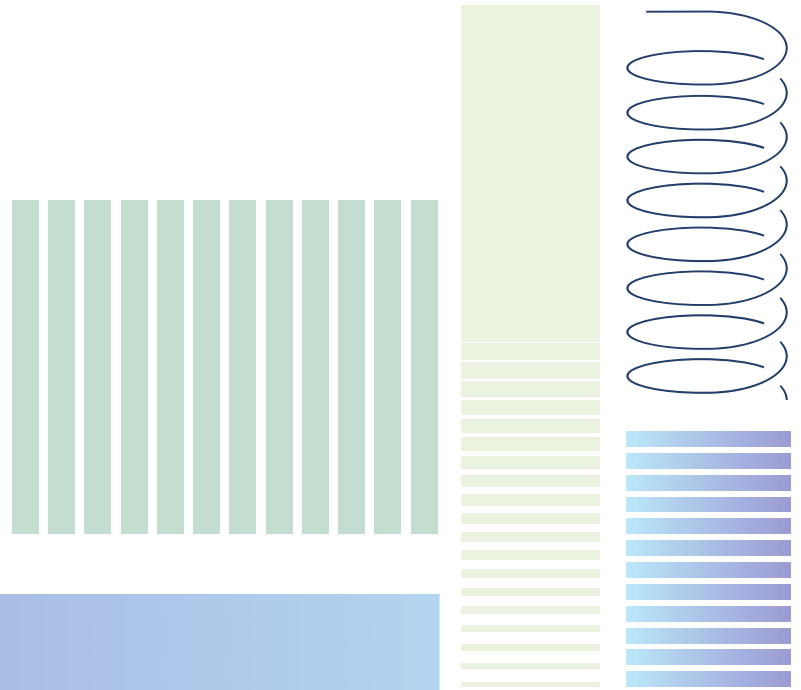


D6.3: Ontology Web Server



Deliverable number and name	D6.3 – Ontology Web Server
Due date	30/11/2018
Work package	WP6
Deliverable leader	EURECOM
Authors	Raphaël Troncy
Reviewers	Javier Sevilla, Cristina Portalés
Approved by	UVEG
Dissemination levels	Public
Version	V1.1



Document revision history

Version	Date	Contributor	Comments
1.0	25/01/2019	Raphael Troncy	Full deliverable
1.1	14/02/2019	Raphael Troncy	Add a new section for API and data access

List of acronyms

CRM	Conceptual Reference Model
RDF	Resource Description Framework
OWL	Ontology Web Language
SPARQL	SPARQL Protocol and RDF Query Language
SKOS	Simple Knowledge Organization System

SILKNOW

Table of Contents

1. Introduction	4
2. Browsing the SILKNOW Thesaurus	5
2.1 Producing the SKOS version of the thesaurus	5
2.2 Visualizing the SILKNOW thesaurus	6
3. Browsing the SILKNOW Ontology and Knowledge Graph	8
4. API Access to the SILKNOW Knowledge Graph	11
4.1 Using SPARQL as an API	11
4.2 Using the SILKNOW Linked Data API	12
5. References	14

This deliverable describes how to access the SILKNOW thesaurus and ontology and their documentation. It presents the tools used together with their specific configuration as part of the SILKNOW platform.

SILKNOW

1. Introduction

The deliverable D6.3 describes the software that have been used to provide access and documentation of the SILKNOW thesaurus described in D3.1 (Section 2) and the SILKNOW ontology described in D3.2 (Section 3). We provide the various URLs where those tools are being developed and deployed.

The overall architecture of the SILKNOW platform described in D6.1 is depicted in the figure below.

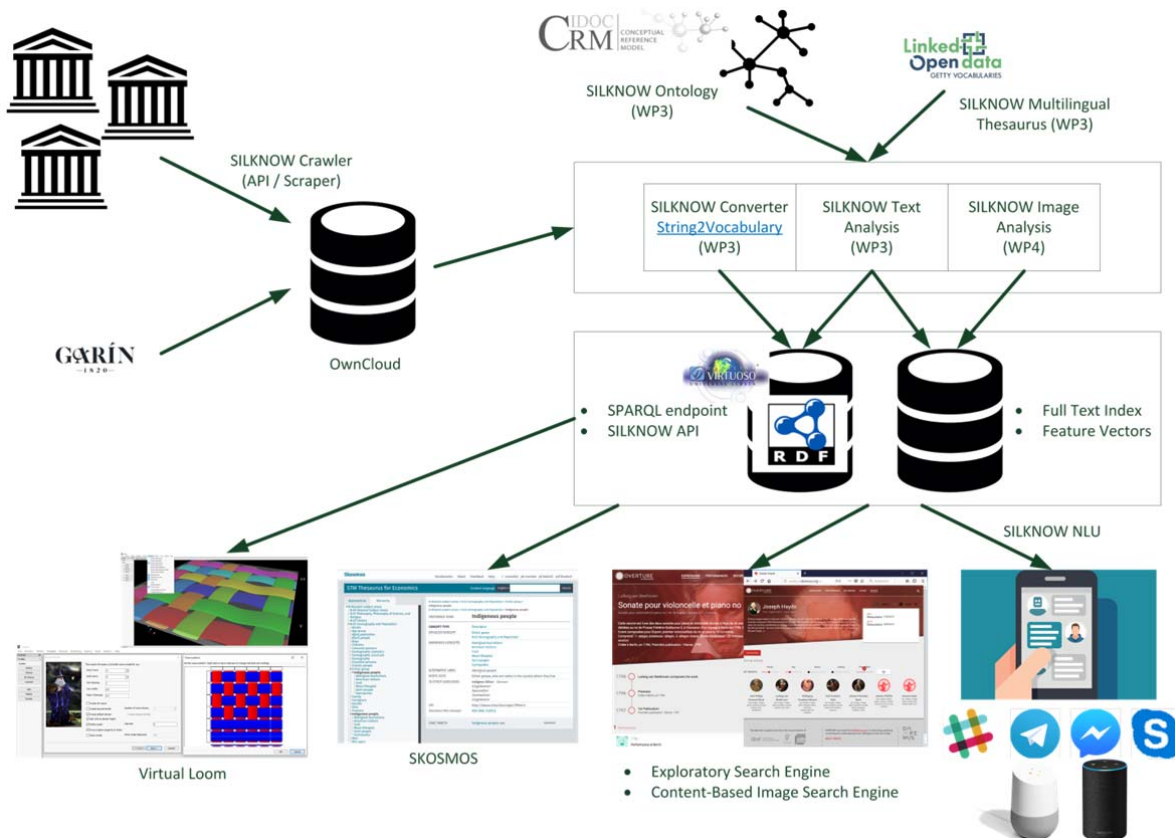


Figure 1: General architecture of the SILKNOW platform (borrowed from D6.1)

SILKNOW

2. Browsing the SILKNOW Thesaurus

2.1 Producing the SKOS version of the thesaurus

The SILKNOW thesaurus has been first primarily edited in Spanish, and it is currently being translated in other languages such as English, French and Italian.

The SILKNOW thesaurus construction has been described in the Deliverable D3.1. The result is a Google spreadsheet available at <https://docs.google.com/spreadsheets/d/1rrVbIaqkjhFw1iPAq-Y2sAvmnnimoDOH-ubMU7TSYQ/edit#gid=976709904>. This spreadsheet can be exported in various CSV files, one file per tab. Next, we developed an open source converter that takes as input such a CSV and convert it into SKOS. The software code is available at <https://github.com/silknow/thesaurus>.

For using the converter, the following command needs to be executed:

- `npm install --production` ## download the dependencies
- `npm run convert` ## start the conversion

The parameters are:

- `--src, -s, source folder`: it parses all csv files inside this folder. By default, the folder is `./raw-data`
- `--dst, -d, output file`. By default, the output file is `'thesaurus.ttl'`

For each concept, we mint a new URI following the pattern: [http://data.silknow.org/vocabulary/\[ID\]](http://data.silknow.org/vocabulary/[ID]). Finally, we create the general SKOS concept scheme <http://data.silknow.org/vocabulary/silk-thesaurus> that groups all the top concepts of the thesaurus and has for label: "Thesaurus describing silk related techniques and material"@en

We post-process the result using the Skosify tool available at <https://github.com/NatLibFi/Skosify>. This tool further clean the SKOS files by detecting double preferred labels in the same language which is forbidden by the SKOS recommendation. It also removes trailing spaces and add relationships in both ways. To run the Skosify tool, the following commands need to be executed:

- `pip install --upgrade skosify`
- `skosify thesaurus.ttl -o thesaurus.ttl`

SILKNOW

Finally, we load the final version of the thesaurus in RDF triplestore available at <http://data.silknow.org/sparql>

2.2 Visualizing the SILKNOW thesaurus

We deployed the Skosmos open source tool developed at <https://github.com/NatLibFi/Skosmos> to visualize the thesaurus.

The SILKNOW thesaurus is available at <http://skosmos.silknow.org/>. The user interface is localized in both English and Spanish and adapts according to the preferred language of the user's web browser.

Skosmos is configured to load the data from the SILKNOW RDF endpoint and generates its view doing SPARQL queries directed to <http://data.silknow.org/sparql>.

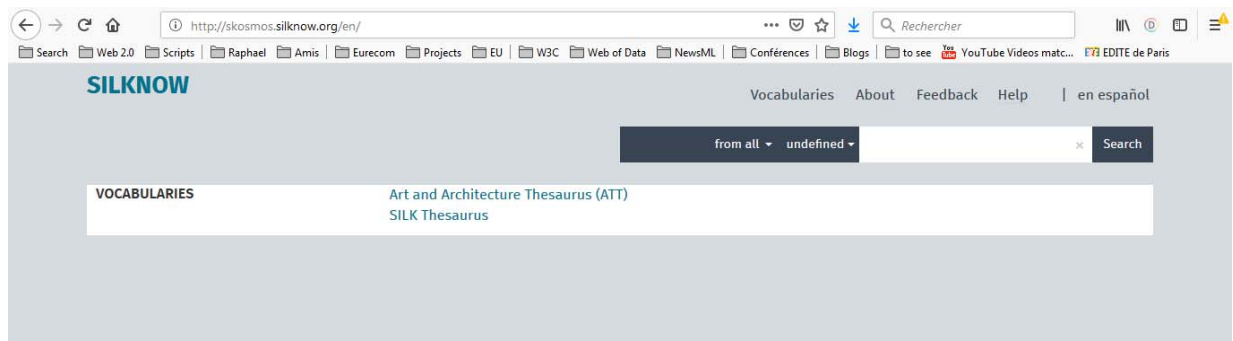


Figure 2: Homepage of Skosmos configured to browse both the AAT and the SILKNOW thesauri

The Figure 2 depicts the welcome page and enables to select to browse either the original Getty AAT thesaurus, or the SILKNOW thesaurus which is specializing AAT.

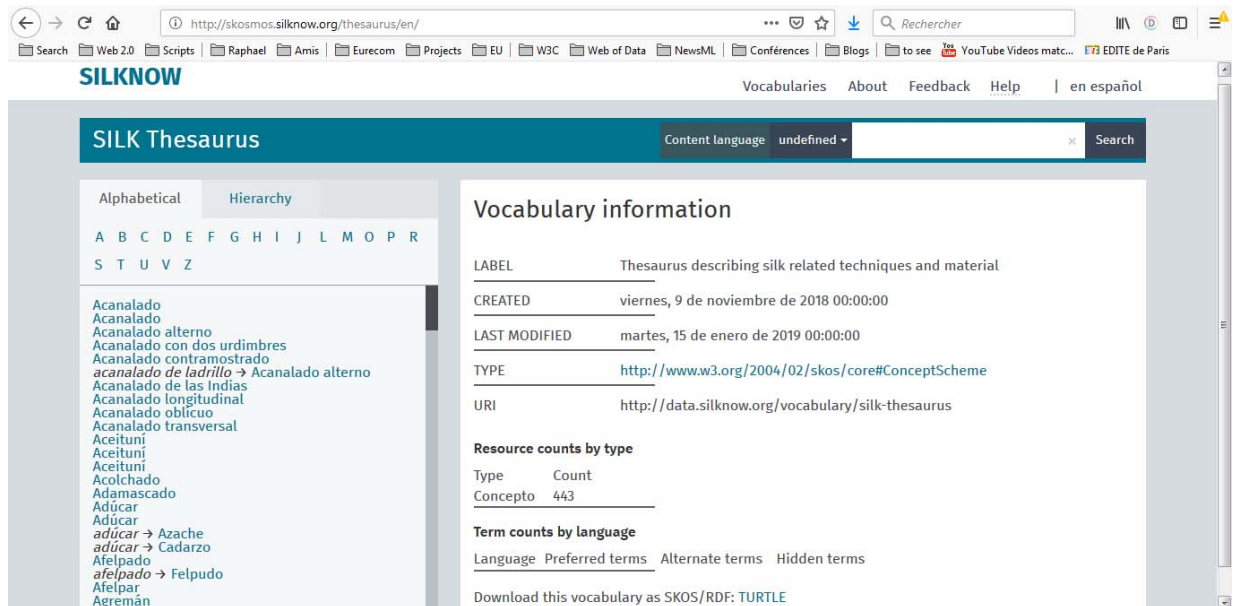


Figure 3: General page showing statistics of the SILKNOW thesaurus

SILKNOW

The Figure 3 depicts the general metadata of the SILKNOW thesaurus. The metadata includes the creation date and the last modification date of the thesaurus and the number of concepts. The version 1.0 of the SILKNOW thesaurus contains 443 concepts. On the left side, the user can click on any concept to have more detailed information. Two views are offered: the alphabetical order of the concepts or their organization in a hierarchy following the broader / narrower relationships.

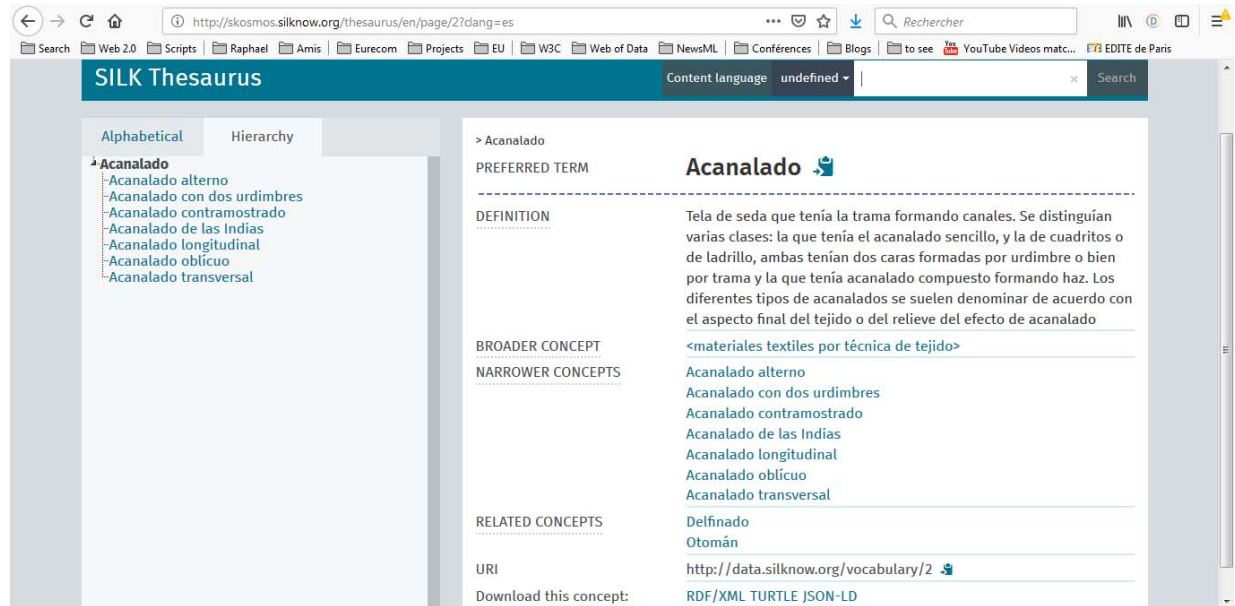


Figure 4: Detailed view of the Acanalado concept

The Figure 4 depicts the detailed view of the Acanalado concept, which definition, in Spanish, reads: *“Tela de seda que tenía la trama formando canales. Se distinguían varias clases: la que tenía el acanalado sencillo, y la de cuadritos o de ladrillo, ambas tenían dos caras formadas por urdimbre o bien por trama y la que tenía acanalado compuesto formando haz. Los diferentes tipos de acanalados se suelen denominar de acuerdo con el aspecto final del tejido o del relieve del efecto de acanalado.”*

Furthermore, this concept is defined as narrower than the AAT concept <http://vocab.getty.edu/aat/300231684> (<textile materials by weaving technique>). It is itself the broader concept of 7 other concepts and it is a related concept to both “Delfinado” and “Otomán”.

SILKNOW

3. Browsing the SILKNOW Ontology and Knowledge Graph

The SILKNOW ontology is developed within the Task 3.3. This ontology is an implementation of the SILKNOW model as described in the Deliverable D2.3 – Data Model Definition. It is based on the CIDOC-CRM ontology, and more precisely, the Erlangen implementation in OWL of this ontology available at <http://erlangen-crm.org/current/>.

We developed an open source converter in JAVA (Java version > 1.8), available at <https://github.com/silknow/converter> that takes as input any of the museum data records that have been crawled and converts it in RDF. In order to use the converter, the following command needs to be executed:

- `./gradlew run --args="[--log=<logLevel>] [-o=<outputFolder>] TYPE FOLDER "`

where the required arguments are:

- TYPE. Museum source data: imatex, garin, joconde
- FOLDER. Source folder to process or a single file which is also accepted.

and where the optional parameters are:

- `--log`. The log level, according to the SLF4J nomenclature¹, by default: WARN.
- `-o, --output`. Output folder, by default: an out folder siblings to the input directory.

The following are examples of commands:

- `./gradlew run --args="imatex
../crawler/data/imatex/records/3345_en.json --log DEBUG -o ./output"`
- `./gradlew run --args="imatex
../crawler/data/imatex/records/3008_en.json --log DEBUG -o ./output"`
- `./gradlew run --args="joconde
../crawler/data/joconde/records/95.71.2.json -o ./output"`
- `./gradlew run --args="joconde
../crawler/data/joconde/records/95.71.364.json -o ./output"`
- `./gradlew run --args="garin ../Desktop/garin/T000053.xls -o
./output"`

We have processed three datasets, namely IMATEX, JOCONDE and GARIN, following the mapping rules described in the Deliverable D3.2. This generates three RDF dumps that we have loaded in the SILKNOW RDF endpoint available at <http://data.silknow.org/sparql>.

¹ <https://www.slf4j.org/api/org/apache/commons/logging/Log.html>

SILKNOW

The SPARQL endpoint is powered by the Openlink Virtuoso software, <http://virtuoso.openlinksw.com/>. We also configured a faceted browser enabling to browse the SILKNOW data, available at <http://data.silknow.org/ct/>. We strictly follow the linked data principles, and each URI identifying an object is dereferencable.

The screenshot shows a web browser displaying the Virtuoso interface for an object. The URL is <http://data.silknow.org/describe/?url=http%3A%2F%2Fdata.silknow.org%2Fobject%2F5a087a1b-2003-33ed-ac71-bb4e82d5a0b5>. The page title is "About: <http://data.silknow.org/object/5a087a1b-2003-33ed-ac71-bb4e82d5a0b5>". The type is "http://erlangen-crm.org/current/E22_Man-Made_Object". The main content is a table of attributes and values:

Attributes	Values
rdf:type	http://erlangen-crm.org/current/E22_Man-Made_Object
rdfs:comment	- conserva 1 voraviu
http://erlangen-crm.org/current/P3_has_note	- conserva 1 voraviu
http://erlangen-crm.org/current/is_identified_by	4278
http://erlangen-crm.org/current/as_representation	http://data.silknow.org/image/74b21664-12e0-3989-92ac-43cdf5dd2177
http://erlangen-crm.org/current/P43_has_dimension	http://data.silknow.org/object/5a087a1b-2003-33ed-ac71-bb4e82d5a0b5/dimension/h http://data.silknow.org/object/5a087a1b-2003-33ed-ac71-bb4e82d5a0b5/dimension/w
http://erlangen-crm.org/current/P62_depicts	floral patterns
dc:identifier	3273_en
is http://erlangen-crm.org/current/nsferred_title_of	http://data.silknow.org/event/aadac060-c1d6-3cf5-9c90-60d86621b0bc
is http://erlangen-crm.org/current/P129_is_about	3006_ca
is http://erlangen-crm.org/current/ferred_custody_of	http://data.silknow.org/event/a4cc3044-1ea7-388c-a473-40e77b30b294
is http://erlangen-crm.org/current/P41_classified_of	http://data.silknow.org/object/5a087a1b-2003-33ed-ac71-bb4e82d5a0b5/type_assignment/2 http://data.silknow.org/object/5a087a1b-2003-33ed-ac71-bb4e82d5a0b5/type_assignment/1
is http://www.ics.forth.gr/RMsc/Q8_observed_of	http://data.silknow.org/object/5a087a1b-2003-33ed-ac71-bb4e82d5a0b5/observation/1
is http://erlangen-crm.org/current/P108_has_produced	http://data.silknow.org/production/b533d348-0993-33b5-8c53-7be8f4ead3af
is http://erlangen-crm.org/current/P90_observed_of	http://data.silknow.org/object/5a087a1b-2003-33ed-ac71-bb4e82d5a0b5/dimension/height

Figure 5: Virtuoso view of an object from the IMATEX museum

The Figure 5 depicts the Virtuoso view for the object identified by <http://data.silknow.org/object/5a087a1b-2003-33ed-ac71-bb4e82d5a0b5>. This described the object 3273_en of the IMATEX museum following the SILKNOW ontology. In the details, this object depicts “floral patterns”.

The screenshot shows a web browser displaying the Virtuoso interface for a production object. The URL is <http://data.silknow.org/describe/?url=http%3A%2F%2Fdata.silknow.org%2Fproduction%2Fb533d348-0993-33b5-8c53-7be8f4ead3af>. The page title is "About: <http://data.silknow.org/production/b533d348-0993-33b5-8c53-7be8f4ead3af>". The type is "http://erlangen-crm.org/current/E12_Production". The main content is a table of attributes and values:

Attributes	Values
rdf:type	http://erlangen-crm.org/current/E12_Production
http://erlangen-crm.org/current/P126_employed	silk vegetal fibres
http://erlangen-crm.org/current/lace_on_or_within	Spain Italy
http://erlangen-crm.org/current/P4_has_time-span	17th century 16th century
http://erlangen-crm.org/current/general_technique	trama llançada patterned fabric brocatelle
http://erlangen-crm.org/current/P108_has_produced	http://data.silknow.org/object/5a087a1b-2003-33ed-ac71-bb4e82d5a0b5
is http://erlangen-crm.org/current/P129_is_about	3006_ca

Faceted Search & Find service v1.13.91 as of Apr 05 2017

Alternative Linked Data Documents: [ODE](#) Content Formats: [CMM](#) [CSV](#) [RDF](#) [N-Triples](#) [JSON-LD](#) [JSON](#) [XML](#) [ODATA](#) [ATOM](#) [JSON](#) Microdata [JSON](#) [HTML](#) [About](#)

LINKINGOPENDATA [Virtuoso](#) [SPARQL](#) [RDF](#)

OpenLink Virtuoso version 07.20.3217 as of Apr 5 2017, on Linux (x86_64-unknown-linux-gnu), Single-Server Edition (62 GB total memory)
Data on this page belongs to its respective rights holders.

SILKNOW

Figure 6: Virtuoso view of the production information of an object from the IMATEX museum

The Figure 6 depicts the Virtuoso view of the production information for the object identified by <http://data.silknow.org/object/5a087a1b-2003-33ed-ac71-bb4e82d5a0b5>. Hence, we know that this object has been made using silk and vegetal fibres, in Spain and in Italy, during the 16th and 17th century using the following weaving techniques: trama llançada, patterned fabric, brocatelle.

As future work, we will replace part of this metadata using terms from the SILKNOW thesaurus.

SILKNOW

4. API Access to the SILKNOW Knowledge Graph

4.1 Using SPARQL as an API

The SPARQL Query Language is a declarative query language (like SQL) for performing data manipulation and data definition operations on data represented as a collection of RDF statements².

A SPARQL query has a solution modifier (or head) and a query body. The solution modifier provides the basis for categorizing different types of SPARQL query solutions. The query body comprises a collection of RDF statement patterns that represent the entity relationships to which a query is scoped. The solution modifier includes read-oriented data access (SELECT, ASK, DESCRIBE, CONSTRUCT) and write-oriented data access (CREATE, INSERT, UPDATE, DELETE, CLEAR, DROP).

A SPARQL Query Service is an HTTP Service (also known as a Web Service) that offers an API for performing declarative data definition and data manipulation operations on data represented as RDF sentence collections, via GET, POST, and PATCH operations that support query solution (result set) delivery using a variety of negotiable document types. SPARQL Queries are executable directly from any computer using `cURL`³, a command line tool and library for transferring data with URLs.

A first API providing access to the SILKNOW knowledge graph is therefore:

```
QUERY=$(<example-query.sparql) && curl -X POST -H "Accept:application/sparql-  
results+json" --data-urlencode "query=$QUERY" http://data.silknow.org/sparql
```

This instructs to submit a particular query to the endpoint and to get the results in the JSON format, which is the most favorite format for web developers. The full specification of the HTTP API is as follows⁴:

- HTTP methods: GET, PUT, POST
- Endpoint: <http://data.silknow.org/sparql>
- Required parameters:
 - *query*: the url-encoded of the SPARQL query
 - *named_graphs*: a vector of named graphs (or NULL to prevent overriding named graphs specified in the query in the FROM clause)
- Optional parameters
 - *dflt_graph*: the default graph URI (string or NULL)
 - *timeout*: the timeout value for query execution expressed in milliseconds (values less than 1000 are ignored)
 - *debug*: if set to on, the SPARQL Compiler will check if all variables are declared, and if there is variable that is not declared, an error will be raised

We provide below several examples of SPARL queries that can be used on the SILKNOW endpoint.

Example 1: *list all weaving techniques mentioned in the data*

² <https://medium.com/virtuoso-blog/what-is-a-sparql-endpoint-and-why-is-it-important-b3c9e6a20a8b>

³ <https://curl.haxx.se/>

⁴ See also <http://vos.openlinksw.com/owiki/wiki/VOS/VOSSparqlProtocol> for the specific implementation of Virtuoso of the http SPARQL API

SILKNOW

```
select DISTINCT *
where {
  [] ecrm:P108_has_produced [];
  ecrm:P32_used_general_technique ?tech .
}
```

Example 2: list all materials mentioned in the data

```
select DISTINCT *
where {
  [] ecrm:P108_has_produced [];
  ecrm:P126_employed ?material .
}
```

Example 3: list all objects and filter by material (e.g. 'ottoman') or technique (e.g. 'doble tela) or location (e.g. 'Italy') or period of time (e.g. '15th century')

```
select *
where {
  ?s a ecrm:E22_Man-Made_Object;
  rdfs:label | dc:identifier ?label .
  ?production ecrm:P108_has_produced ?s;
  ecrm:P126_employed ?material ;
  ecrm:P4_has_time-span / rdfs:label ?time .
  OPTIONAL {?production ecrm:P8_took_place_on_or_within / rdfs:label
?location }
  OPTIONAL {?production ecrm:P32_used_general_technique ?tech }
  ## FILTER BY MATERIAL ottoman
  # VALUES ?material {'ottoman'}
  ## FILTER BY TECHNIQUE doble tela
  VALUES ?tech {'doble tela'}
  ## FILTER BY TIME 15th century
  # VALUES ?time {'15th century'}
  ## FILTER BY LOCATION Italy
  # VALUES ?location {'Italy'}
}
```

4.2 Using the SILKNOW Linked Data API

We have started to implement a second, more classic, REST-based API on top of the SILKNOW knowledge base. Our implementation relies on ELDA⁵, an implementation of the Linked Data API proposed by the UK government⁶. In a nutshell, this implementation enables to map pre-defined SPARQL queries with REST calls via a configuration file. Hence, the query given in the example 1 above enabling to retrieve all techniques mentioned in the data could simply be served via a HTTP call such as:

```
GET http://data.silknow.org/api/material
```

⁵ <https://www.epimorphics.com/technology/elda/>

⁶ <https://github.com/UKGovLD/linked-data-api>

SILKNOW

This simple request simply retrieves all possible values of material found in the knowledge base. A more interesting query is:

GET

```
http://data.silknow.org/api/object/location/{location}/period/{time}/technique/{technique}/language/{lang}
```

This more complex query will retrieve all objects (description of museum records) found in the knowledge base that have been created in a particular *location* (e.g. Italy), during a particular *time period* (e.g. 15th Century) using a particular *weaving technique* (e.g. <<http://data.silknow.org/vocabulary/421>>, the identifier of the ‘doble tela’ weaving technique in the SILKNOW thesaurus), and this description must be returned in a specific *language* (e.g. Spanish).

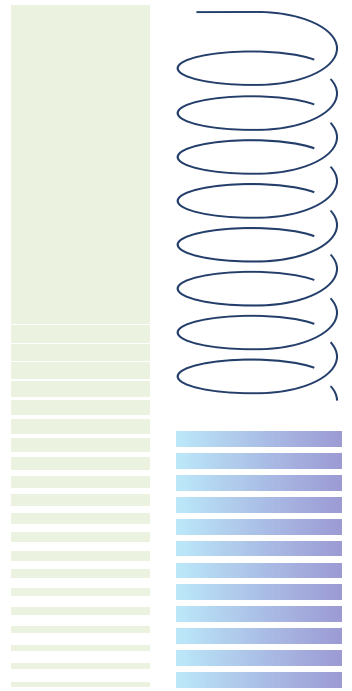
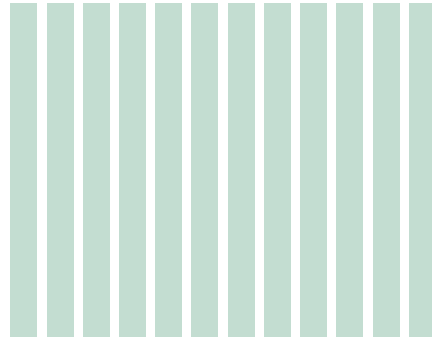
The partial specification of this REST API is as follows:

- HTTP methods: GET, PUT, POST
- Endpoint: <http://data.silknow.org/api/>
- Calls available:
 - /object/all
 - Retrieve ALL objects described in the knowledge base
 - /object/location/{location}
 - Retrieve ALL objects that have been produced in *location*
 - /object/period/{period}
 - Retrieve ALL objects that have been produced during the time *period*
 - /object/material/{material}
 - Retrieve ALL objects of a particular *material*
 - /object/technique/{technique}
 - Retrieve ALL objects made following a particular *technique*
 - /object/search/{keyword}
 - Retrieve ALL objects for which the searched *keyword* is present in the title, or description, or notes in the object record
 - /thesaurus/all
 - Retrieve ALL concepts from the SILKNOW thesaurus
 - /thesaurus/top
 - Retrieve ALL TOP concepts from the SILKNOW thesaurus
 - /thesaurus/search/{keyword}
 - Retrieve ALL concepts from the SILKNOW thesaurus for which the searched *keyword* is present in the concept name or description

SILKNOW

5. References

- [1] SILKNOW Deliverable D2.1 – Data Model Definition, May 2018.
- [2] SILKNOW Deliverable D2.2 – Initial Collected Data, July 2018
- [3] SILKNOW Deliverable D2.3 - Definition of the Graphical User Interface (GUI), August 2018.
- [4] SILKNOW Deliverable D2.4 – Pilot Scenarios Definition, September 2018.
- [5] SILKNOW Deliverable D3.1 – Historical Silk Multilingual Thesaurus, December 2018.
- [6] SILKNOW Deliverable D3.2 – Design of SILKNOW Ontology, November 2018.
- [7] SILKNOW Deliverable D6.1 – Design Overall Software Structure and Data Sharing Framework



VNIVERSITAT
DE VALÈNCIA



LINIVERSITÀ
DEGLI STUDI
DI PALERMO



Leibniz
Universität
Hannover



MONKEYFAB



Institut
"Jožef Stefan"
Ljubljana, Slovenija



Instituto Cervantes



EURECOM



GARÍN
-1820-