# DELIVERABLE D2.1

_user requirements for the publication platform from target organizations, including the map of typical environments_

<table>
<thead>
<tr>
<th>Project</th>
<th>Components Supporting the Open Data Exploitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acronym</td>
<td>COMSODE</td>
</tr>
<tr>
<td>Contract Number</td>
<td>FP7-ICT-611358</td>
</tr>
<tr>
<td>Start date of the project</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; October 2013</td>
</tr>
<tr>
<td>Duration</td>
<td>24 months, until 31&lt;sup&gt;st&lt;/sup&gt; September 2015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Date of preparation</th>
<th>26&lt;sup&gt;th&lt;/sup&gt; February 2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Martin Nečaský, Tomáš Knap, Ján Marček, Gabriel Lachmann, Ľubor Illek, Miroslav Konečný, Roberto Cornachia, Andrea Maurino,</td>
</tr>
<tr>
<td>Responsible of the deliverable</td>
<td>Peter Hanečák</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:peter.hanecak@eea.sk">peter.hanecak@eea.sk</a></td>
</tr>
<tr>
<td>Reviewed by</td>
<td>Miroslav Konečný</td>
</tr>
<tr>
<td>Status of the Document</td>
<td>Final version</td>
</tr>
<tr>
<td>Version</td>
<td>1.0</td>
</tr>
<tr>
<td>Dissemination level</td>
<td>RE (Restricted to a group specified by the consortium (including the Commission Services))</td>
</tr>
</tbody>
</table>
4.2.7 Scalable implementation ................................................................. 34
4.2.8 Operation behind firewall ............................................................... 35
4.2.9 Well defined release policy and planning for ODN .......................... 35
4.2.10 Support for ODN has to be quick and of high quality ....................... 35
4.2.11 Support for a wide range of dataset types ....................................... 36
4.2.12 “Open Data ready” concept ............................................................ 36
4.2.13 Clear explanation of Open Data, the motivation and benefits ............ 37
4.2.14 Clear explanation of Open Data publishing process ......................... 37
4.2.15 Publisher of the data remains owner of the data ............................. 37
4.2.16 Dataset wishlist ............................................................................ 38
4.2.17 Feedback about quality of data ....................................................... 38
4.2.18 First impression .............................................................................. 38
4.2.19 “What are the benefits for me?” ..................................................... 38
4.2.20 Graphical content, attachments ..................................................... 39

5 Assessment of environments within selected organizations .................. 40
  5.1 Map of typical environments ............................................................... 41
    5.1.1 Cloud ............................................................................................ 41

6 Common integration strategies .............................................................. 42
  6.1 Tight integration, at the publisher’s premises ....................................... 42
  6.2 Loose integration, at almost any location ............................................ 44
  6.3 No explicit integration, deployed usually at 3rd parties ....................... 45

7 Requirements List .................................................................................. 47

8 Summary ................................................................................................. 48

9 References ............................................................................................... 49

ANNEX 1: COMSODE Requirements List
1 Deliverable context

1.1 Purpose of deliverable

This document represents the main output of Task 2.1 (Assessment of existing tools and environments within organizations along with user requirements for the publication platform). It contains the following project achievements:

1. Assessment of existing tools (see chapter 3”);
2. Assessment of environments within organizations and map of typical environments (see chapter 5);
3. Typical requirements which the COMSODE publishing platform (Open Data Node - ODN in short) will have to satisfy (see chapter 4);
4. Common integration strategies for the publishing platform (see chapter 6).

It will later serve as main input for Task 2.2 (Architecture and design of the publication platform - ODN) and Task 6.5 (Proof-of-concept/pilot publishing project).

1.2 Related Documents

- List of related documents from project:
  - Description of Work – WP2
  - Deliverable D2.3 – Architecture and design documentation for COMSODE development tasks
2 Methodology used

2.1 Main steps

COMSODE consortium members first formulated a target group (or “market segment”) to be engaged and provided with tools for Open Data publication. It comprises:

1. **Public bodies** (and possibly other organizations) who want to publish their data as Open Data
2. **Application developers** who want to collect Open Data from other organizations, keep it always up-to-date and use it in their applications

Based on that we formulated an initial set of preliminary requirements as part of the project setup (see Part A, Part B and Description of Work).

After the project started, we contacted many organizations and individuals who fit into our target group (public bodies, SMEs, etc.) and formed User Group. In order to make the group more diverse (to obtain better feedback) we also included several other representatives who are related to Open Data (research institutes, universities, activists, etc.). From the members of User Group we then selected some as member of the more formal COMSODE User Board. We also collected contact information to other relevant people via members of consortium, associated partners and members of User Group.

Then we began to collect inputs needed for the assessment. We used all the contacts we gathered and used several methods to gather information, each particular method adapted to the expertise and availability of each particular contact person:

- We performed guided interviews with associated partners and other potential users (customers) for the planned publication platform mapping and assessing their environments and compiling a list of further requirements for the publication platform.
- We organized a first round of collaboration with User Group centered around first meeting of User Board in January 2014 in Prague. This included presentation of our plans, results achieved so far and series of dialogs to solicit feedback useful to the project.
- We took part at Open-data related seminars and events where potential users and publishers expressed their experiences and needs (e.g. “Opening the cities” seminar in Prague, Open Data Forum Bratislava, conference Lets Inspire Bratislava, etc. - see also COMSODE Dissemination Plan).
- We collected various smaller pieces of input (tools used, problems faced, ideas, requirements, etc.) from the rest of the contact persons.
- We searched for and assessed various existing tools, solutions and projects for the publication of (Linked) Open Data determining that are relevant and in what way for COMSODE project.

After that we analysed all the inputs and prepared parts of this deliverable:

- typical requirements for the publication platform,
- map of expected typical environments,
- common integration strategies,
- requirements list.
2.2 Preliminary high-level overview of ODN

In order to a) solicit feedback from potential users of Open Data Node publishing platform and b) present the findings of our assessment, we need to first describe some basic concepts and capabilities of this platform, as they are currently planned. For that we are including the current preliminary version of ODN architecture in this document.

2.2.1 Schema of the basic use-case for ODN

**Input:** Public bodies have a lot of data in heterogeneous environments, using a wide variety of formats and employing many different technologies to access and process that data. ODN is harvesting that data using a variety of interfaces, data formats and algorithms.

**Transformation:** ODN performs extraction and processing (conversion, cleansing, anonymization, etc.) of harvested data and stores the results.

**Output:** Open Data is about publishing that data in open and machine-readable formats to the general public and businesses including automated efficient distribution of updated data (dataset replication). ODN will provide the data from its storage in such formats.

The key point is that ODN helps publishers with the complexity of source data and delivers easy to use and high quality Open Data to users.

2.2.2 Actors and typical functions of ODN

Open Data Node can be deployed many times by many actors and can help with needs specific to each particular actor:

- government organizations, municipalities, etc. want to publish a majority of their information as Open Data
- other government bodies need to work with some data published by other government bodies
- non-profits and application developers want to run specific tasks using copies of official data, for example analytics and visualization applications, data integration, etc.
2.2.3 ODN hierarchies

When both data publisher and user of data use ODN, i.e. when one ODN instance is harvesting data from other ODN instance, we call that ODN hierarchy. Hierarchies can be formed spontaneously (by both side merely choosing to use same tool, without coordination and maybe even without being aware of that fact) or on-purpose (both side deliberately coordinating and choosing ODN).

Hierarchies can be simple (just two OPDN instances, one harvesting data from the other) or complicated (multiple ODN instances, many organizations, even two-way data flows, etc.).

Hierarchies can form mainly because of geography (for example aggregator harvesting datasets which relates to certain location) or topic (for example application developer harvesting datasets related to tourism).

How will two ODN instances cooperate? As ODN is Open Source solution publishing Open Data, we will not develop a secret proprietary interface to “promote" and provide “advantage” for ODN hierarchies (to gain competitive advantage). ODN will be merely able to harvest data using same formats and interfaces it is employing to publish data.

But we will make sure that flow of data between two Open Data Nodes will be tested and optimized for multiple possible scenarios (simple dataset mirroring, subset or superset republication, via file dumps, via API, etc.).
2.2.4 Typical use-cases

- **Publisher of Open Data** (public body)
  - Integration tools for extracting data from internal systems
  - Automated and repeatable data harvesting: extraction and processing (conversion, cleansing, anonymization, etc.) of data, both:
    - Initial harvesting of whole datasets (first import)
    - Periodical harvesting of incremental updates
  - Internal storage for the data and metadata
  - Data publishing in open and machine-readable formats to the general public and businesses including automated efficient distribution of updated data and metadata (dataset replication)
    - integration with data catalogues (like CKAN) for automated publication and updating of dataset metadata
  - Internal data catalogue of datasets for maintenance of dataset metadata

- **User of Open Data** (citizen, data analyst, etc.)
  - Unlike in all other use-cases, in this case the user is merely accessing data in ODN instance maintained by someone else, user is not running his own instance
  - User is downloading data dumps and calling APIs to get the data which he is interested in
  - Very often, user will also access the data indirectly, for example via 3rd party data catalogue (or other applications), which - in order to allow the user to search for datasets and show the preview or visualization of data - have to first download data and metadata from ODN

- **Aggregator of Open Data** (public body, NGO, SME, etc.)
  - Functions same as in previous use-case plus also:
  - Support for Open Data Node hierarchies for efficient dataset replication
  - Automated and repeatable data integration and linking

- **Application developer using Open Data** (SME, NGO, etc., public body too)
  - Functions same as in previous use-cases but optimized for:
  - Tools for automated generation of API and custom API development

2.2.5 Architecture of ODN: modules and data flows

Open Data Node will consist of these main parts:

- **ETL part**: responsible for the harvesting and processing of data
- **Storage part**: responsible for the storage of harvested and processed data
- **Publication part**: responsible for the publication of harvested and processed data
- **Management part**: allowing ODN user to manage ODN and data in it (harvesting, transformations, storage, publication)
ETL (Extract, transform, load[1]) is essentially the core part of Open Data Node responsible for harvesting, anonymization, enhancement, cleansing, quality assessment, provenance tracking, integration, linking and enrichment of the data.

ETL in ODN will support not just usual tabular data. The main potential for data integration, enhancement, quality assessment, provenance and linking is in usage of Semantic Web[2] technologies (RDF[3], OWL[4], etc.), i.e. when the ETL is converting to and working with Linked Data[5].
3    Reuse of existing tools, solutions and projects

As part of our effort to cooperate with and reuse outputs of other FP7, EU and international projects, we’ve mapped and assessed several existing tools, solutions and projects which are relevant for publishing of (Linked) Open Data to determine what components can be reused, including the state of each (quality, license, usability/reusability by COMSODE) and the purpose they can serve. Final verdict about what and how will be reused will be done as part of Architecture and Design (Task 2.2).

The main motivation for reuse is the need to avoid unnecessary duplicate development thus saving project resources (from business perspective) but also to adhere to best state of the art Open Source practices.

Please refer to section “Preliminary high-level overview of ODN” for more details about architecture of Open Data Node (ETL part, storage part, publication part, etc.).

3.1.1    Quality of the tools

We’re taking closer look at quality of existing tools because if we want to reuse some existing components, it should be mature enough so as to not risk spending of significant part of our resources on enhancements and bugfixing of a particular tool.

3.1.2    Licensing

Main differentiator of COMSODE publishing platform (Open Data Node - ODN) is it’s Open Source licensing. In order for any other tool to be reusable by COMSODE, this tool has to be Open Source too.

Plus licensing models of all the tools and modules we choose to reuse will help us determine exact licensing needed for the publication platform itself (exact description of licensing is subject of other deliverable).

3.1.3    Reusability

We’re assessing whether tool, project or service fits COMSODE project needs:

- Is it providing functions relevant to ODN?
- Is it using right mix of technologies?
- Can we reasonably expect to integrate it easily with the rest of ODN?
- Etc.

3.1.4    List of reviewed tools, solutions and projects

Overview of reviewed tools, solutions and projects is described in the table below. More detailed information about each item is provided in subsequent sections.

<table>
<thead>
<tr>
<th>Tool, solution or project name</th>
<th>Reuse value for COMSODE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigData</td>
<td>Maybe, in storage part</td>
</tr>
<tr>
<td>CAM - Content Assembly Mechanism toolkit</td>
<td>No or in very limited fashion</td>
</tr>
<tr>
<td>Cascading</td>
<td>Portions usable, mainly for ETL part.</td>
</tr>
<tr>
<td>Tool</td>
<td>Status</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CKAN</td>
<td>Yes, in management part, but further review needed</td>
</tr>
<tr>
<td>Clover ETL - community edition</td>
<td>No</td>
</tr>
<tr>
<td>CODE</td>
<td>No</td>
</tr>
<tr>
<td>CUSTODEA OAI-PMH provider</td>
<td>Yes, in publication part</td>
</tr>
<tr>
<td>DaPaaS</td>
<td>No</td>
</tr>
<tr>
<td>Data Pipes</td>
<td>Yes, but most probably not directly</td>
</tr>
<tr>
<td>Deri Pipes</td>
<td>No</td>
</tr>
<tr>
<td>DKAN</td>
<td>Yes, pending further review of similar tools</td>
</tr>
<tr>
<td>Etalab</td>
<td>Yes, pending further review of similar tools</td>
</tr>
<tr>
<td>“Old” Open Data Node</td>
<td>No</td>
</tr>
<tr>
<td>FI-WARE</td>
<td>No or only very limited</td>
</tr>
<tr>
<td>GovData - Open Data Platform</td>
<td>Yes, pending further review of similar tools</td>
</tr>
<tr>
<td>iCity project</td>
<td>No</td>
</tr>
<tr>
<td>Junar</td>
<td>No</td>
</tr>
<tr>
<td>Karma</td>
<td>No, or maybe</td>
</tr>
<tr>
<td>LDIF</td>
<td>Yes, portions of it in ETL part</td>
</tr>
<tr>
<td>mingle.io</td>
<td>Maybe, in storage and publication part</td>
</tr>
<tr>
<td>Nomenklatura</td>
<td>Yes, in ETL part</td>
</tr>
<tr>
<td>OAIHarvester2</td>
<td>Yes, in ETL part</td>
</tr>
<tr>
<td>ODCS v2</td>
<td>Yes, in ETL part</td>
</tr>
<tr>
<td>Okkam Entity Name System (ENS)</td>
<td>Yes, in ETL part</td>
</tr>
<tr>
<td>Open311</td>
<td>Very little</td>
</tr>
<tr>
<td>OpenCivic</td>
<td>Maybe, in publication part</td>
</tr>
<tr>
<td>OWLIM Lite</td>
<td>No</td>
</tr>
<tr>
<td>restSQL</td>
<td>Yes, in publication part</td>
</tr>
<tr>
<td>Sentient Data Manager</td>
<td>No</td>
</tr>
</tbody>
</table>
Sesame OpenRDF | Maybe, in storage part
---|---
Sieve | Yes, in ETL part
Silk | Yes, in ETL part
Sindice | Yes, in ETL part
Socrata | No
Talend Open Studio | No
The DataTank | Maybe, in publication part
Virtuoso Open Source | Yes, in storage part

### 3.1.5 Assessment of reviewed tools, solutions and projects

**BigData**


"BigData(R) is a scale-out storage and computing fabric supporting optional transactions, very high concurrency, and very high aggregate IO rates. The bigdata RDF/graph database can load 1B edges in under one hour on a 15 node cluster. Bigdata operates in both a single machine mode (Journal), highly available replication cluster mode (HAJournalServer), and a horizontally sharded cluster mode (BigdataFederation). ..."

State:
- quality: production
- license: GPLv2

**Reusable: Maybe, in storage part.**

This tool covers all relevant needs of Open Data Node for storage of Linked Open Data. Preliminary benchmarking suggest acceptable level of performance for both read and write operations with higher potential for clustered operation (which was not tested). And compares well with other triplestores (OWLIM, Sesame OpenRDF, Virtuoso).

The only disadvantage is that ODCS is so far being implemented using Virtuoso and has not been tried with Sesame.

**CAM - Content Assembly Mechanism toolkit**


"The CAM editor is the leading open source toolkit for building and deploying information exchanges and Open Data APIs using XML or JSON with SQL. The CAM toolkit provides an intuitive approach using a WYSIWYG visual structure editor to dramatically simplify the process of developing and managing XML business information exchanges and schema. This gives developers control, insights and analysis that are needed for consistent, interoperable and reliable exchanges."
Reusable: No or in very limited fashion.
CAM could be reused in publication part of Open Data Node to design APIs for the datasets published via ODN but it is interactive tool (thus not suitable for automated mode) and its license is not compatible with GPL (which is for now planned to be used by COMSODE implying we can't reuse at least portions of its source code). We may at bundle it on ODN installation media along with other tools.

*Cascading*

“Cascading is an application framework for Java developers to simply develop robust Data Analytics and Data Management applications on Apache Hadoop.”

Reusable: Portions usable, mainly for ETL part.

‘Cascading’ addresses the same basic needs as Open Data Node (automated and repeatable publication of data) but aims for different “customer”: programmers or organizations with very specific needs requiring custom solutions (whereas ODN aims to be a generic tool not requiring - but allowing - custom development).

Portions of this project might be used by COMSODE, for example Lingual module to allow SQL based workloads to use Apache Hadoop as backend.

We should at least mention this tool in our Methodology to give specific publishers more options.

*CKAN*

“CKAN is a powerful data management system that makes data accessible – by providing tools to streamline publishing, sharing, finding and using data. CKAN is aimed at data publishers (national and regional governments, companies and organizations) wanting to make their data open and available.“

Reusable: Yes, in management part, but further review needed.

CKAN data catalogue bundled with Open Data Node can provide multiple features relevant for data publishers (management of dataset metadata, synchronization/harvesting of that data by national and EU-level data catalogues, etc.). We want to reuse this tool but there are some
issues (like incomplete support for DCAT-AP and ad-hoc nature of the source code - leading to possible stability, extendibility and integration problems) we will need to investigate it further and more thoroughly (including its forks and clones or similar solutions like Etalab, DKAN, etc.).

**Clover ETL - community edition**


“The CloverETL Community Edition is based on our Open Source transformation engine and also includes a limited CloverETL Designer. It is a great way for people with modest data transformation and ETL requirements to start with one of the most powerful tools available in the industry. Our Community Edition is free.”

State:
- quality: production
- license: LGPL

Reusable: No.

This tool covers a lot of expected functionality of Open Data Node but it lacks support for RDF data management and, when compared to ODCS v2 it lacks also some other features (scheduling, monitoring, debugging, etc.). We will thus not reuse this tool.

**CODE**


“Our use case focuses on research papers as a source for mining facts and their integration into LOD repositories and light-weight ontologies. Hence, we will leverage the wealth of knowledge contained in research publications on a semantic, machine-readable level by creating the Linked Science Data cloud.”

State:
- quality: beta
- license: “open source” but source repository lacks clear licensing information

Reusable: No.

CODE project addresses very similar goals as Open Data Node but as interactive, one-off, not automatically repeatable tasks thus it aims for different user group (analysts using some lower quality data from elsewhere, etc.) as Open Data Node (publishers seeking automated publishing of their own data). Thus, as a whole, we will not reuse results of this project.

We might be able to reuse its smaller components (like DoSeR) but first it’s licensing and time plan needs to be clarified (as CODE project is in it’s first year as FP7 project but the website lacks clear information about what and when when will be delivered and current source code repository does not contain proper licensing information).

**CUSTODEA OAI-PMH provider**


“Implementation of interface for providing metadata from repository to other systems using OAI-PMH protocol.”
State:
- quality: production
- license: LGPLv3

Reusable: Yes, in publication part.
COMSODE consortium member implemented this tool to publish metadata about digitized cultural works to the rest of Europeana project through OAI-PMH protocol. What is seen as “metadata” in Culture segment is “data” in Open Data segment. Thus we will reuse this tool to provide OAI-PMH based API for datasets published via Open Data Node.

**DaPaaS**
Home page: [http://project.dapaas.eu/](http://project.dapaas.eu/)

“The core goal of the DaPaaS project is to provide a Data- and Platform-as-a-Service environment, where 3rd parties (such as governmental organisations, SMEs, developers and larger companies) can publish and host both data sets and data-intensive applications, which can then be accessed by end-user applications in a cross-platform manner.”

State:
- quality: being implemented
- license: service, build using mainly Open Source components

Reusable: No.
DaPaaS project is co-funded from same FP7 programme and same call and started at almost same time. It aims to solve very similar problems and parts of its implementation may overlap with COMSODE. The main difference is that while COMSODEAijms to supply software as solution, DaPaaS aims to provide service.

COMSODE ODN seems as good with as one of the building blocks for DaPaaS. But as DaPaaS is currently seeking blocks which are mature, ODN does not fit with that (as it is only going to be implemented). From COMSODE point of view it’s very similar: we can’t reuse DaPaaS as a whole as it is only to be implemented. But later on it might be interesting to reuse some functionality aimed at end users (bookinking and notification functionality, mobile access clients, etc.).

We agreed to exchange information between our projects and reevaluate plan for possible cooperation each time either of us reaches some milestone.

**Data Pipes**

“Data Pipes is a simple open-source node.js webapp ... providing streaming, "pipe-like" data transformations on the web – things like deleting rows or columns, find and replace, head, grep etc.”

State:
- quality: beta
- license: MIT
Reusable: Yes, but most probably not directly.

This web application provides a web service for transformation of CVS data. As Open Data Node’s ETL part will provide a way to use CSV data from Web, this service will be usable in some processing tasks by ODN users. And it can be even bundled so as to provide runtime independent of official okfnlabs.org service for those users who would need that. Based on results of WP3 (Tasks 3.1 to 3.3, i.e. datasets selection and ways to transform them) we will make final decision whether this tool will be either just supported or also bundled with ODN.

**Deri Pipes**


“Inspired by Yahoo’s Pipes, DERI Pipes is an engine and graphical environment for general Web Data transformations and Mashup.”

State:

- quality: beta, seems little bit stale (last update in March 2009)
- license: GPLv2

Reusable: No.

It is very similar to Data Pipes but provides support for wider set of formats (XML, RDF, JSON) but is also much richer in features to the point where it is comparable to ODCS (with ODCS judged better in such comparison). We will not reuse this project directly on Open Data Node (given less flexibility when compared to ODCS and also its stale status. But users of ODN still might be able to ingest data from Deri Pipes into ODN given the formats supported.

**DKAN**


“DKAN is an open source data platform with a full suite of cataloging, publishing and visualization features that allows governments, nonprofits and universities to easily publish data to the public. … Based on content management system Drupal that makes it easy to integrate with blogs and websites.”

State:

- quality: stable
- license: GPLv2

Reusable: Yes, pending further review of similar tools.

Similarly to CKAN, it covers part of features relevant to data publishers. It is an alternative for CKAN and Etalab if further analysis concludes that CKAN is not feasible to be reused by COMSODE.

**Etalab**

Home page: [https://github.com/etalab](https://github.com/etalab)

“Forked version of CKAN used by French data.gouv.fr”
State:
- quality: stable
- license: AGPLv3

Reusable: Yes, pending further review of similar tools.

Similarly to CKAN, it covers part of features relevant to data publishers. It is an alternative for CKAN and DKAN if further analysis concludes that CKAN is not feasible to be reused by COMSODE.

**“Old” Open Data Node**

Home page: [https://github.com/OpenDataSk/open-data-node](https://github.com/OpenDataSk/open-data-node)

“a proof-of-concept implementation of Open Data Node“

State:
- quality: alpha
- license: GPLv3

Reusable: No.

Developed by OpenData.sk initiative, it partially implements some basic features planned for COMSODE Open Data Node. But is has not reached level of maturity comparable to ODCS or other ETL tools so it will not be used by COMSODE.

**FI-WARE**


“FI-WARE is a core platform that eases the creation of innovative applications by lowering the costs and complexity of serving large numbers of users globally and handling data at a large scale. Its final objective is to create a sustainable innovation ecosystem engaging entrepreneurs, application sponsors and platform providers.

FI-LAB is live instance of FI-WARE available to developers for free experimentation with the technology. You can create your FI-LAB account by connecting to the FI-LAB site.

Besides a number of portals enabling you to self-service the virtual infrastructure to support your experiments, you can access a rich library of Generic Enabler implementation instances offered "as a Service" through well-defined Service End Points, or created by you using tools provided by the Cloud at FI-LAB. You may obtain information about FI-WARE Generic Enabler implementations by visiting the FI-WARE Catalogue.“

State:
- quality: beta
- license: “open source” claimed

Reusable: No or only very limited.

FI-WARE offers “FI-WARE API specifications” on “public and royalty-free” basis and “driven by the development of an open source reference implementation”. We were not able to try the platform as [lab.fi-ware.eu](http://lab.fi-ware.eu) was unavailable at the time of review and source code is not
published in very organized manner (version control is not publicly accessible and presumably source code releases at http://forge.fi-ware.eu/frs/?group_id=15 are mixed with binaries, without clear licensing information). So the conclusion so far is, that no significant portions of FI-WARE will be usable in COMSODE and that maybe at least some smaller tools will be reusable in some fashion (like REST Client Generator).

**GovData - Open Data Platform**

Home page: [https://github.com/fraunhoferfokus/GovData](https://github.com/fraunhoferfokus/GovData)

"*Das Datenportal für Deutschland / The data portal for Germany*" is functionally similar to CKAN, but it is a separate implementation based on Liferay.

State:

- quality: beta
- license: AGPLv3

**Reusable: Yes, pending further review of similar tools.**

Similarly to CKAN, it covers part of features relevant to data publishers. It is an alternative for CKAN if further analysis concludes that CKAN is not feasible to be reused by COMSODE.

**iCity project**


*The iCity project vision makes a step further on the concept of Open Data offering a novel approach of Open Infrastructures where the municipal ICT networks already deployed in urban spaces will be made available and accessible to the general public with the objective of maximizing the number of deployed services of public interest.*

State:

- quality: hard to assess
- license: unknown, assumed proprietary

**Reusable: No.**

Project is still running, with pilots in several European cities (Barcelona, Bologna, Genova, London) and as it goes far beyond publication of Open Data, its use and reuse is (as of now) very limited just to these cities. While the project does aim to provide solutions also to data publishers, we were not able to identify any components reusable by COMSODE.

**Junar**

“Junar delivers all the benefits of SaaS (Software-as-a-Service) to help organizations Open Data to spur innovation. Junar makes it easy to deal with complex end-to-end Open Data projects and turns the difficult task of opening data into a secure and controlled process. This allows you to focus on transforming data into readable, searchable, and usable insights that everyone can share.

Junar delivers the easiest-to-use, cloud-based open data platform that enables businesses, governments, NGOs, and academia to free their data to drive new opportunities, collaboration, and transparency. Some of the world's leading companies trust Junar with their most valuable assets: their data and the end users who are using it.

With Junar you can open data fast, decide what is kept for internal use and what is made public, and optimize use of the data by adding meta information which drives search engine optimization.

For innovative organizations, Junar is the fastest way to publish data.”

State:
- quality: production
- license: paid service (SaaS)

Reusable: No.

As similar solutions (like Socrata) there is very little for COMSODE to reuse. We only have to take into account that they offer their services to some of the same groups we’re targeting (data publishers, application developers) so we need to make sure we have some “unique selling point” when offering Open Data Node.

Karma

Home page: http://www.isi.edu/integration/karma/

“Karma is an information integration tool that enables users to quickly and easily integrate data from a variety of data sources including databases, spreadsheets, delimited text files, XML, JSON, KML and Web APIs. Users integrate information by modeling it according to an ontology of their choice using a graphical user interface that automates much of the process. Karma learns to recognize the mapping of data to ontology classes and then uses the ontology to propose a model that ties together these classes. Users then interact with the system to adjust the automatically generated model. During this process, users can transform the data as needed to normalize data expressed in different formats and to restructure it. Once the model is complete, users can published the integrated data as RDF or store it in a database.”

State:
- quality: stable
- license: APLv2

Reusable: No, or maybe.

‘Karma’ is covering similar use case as Open Data Node but as interactive tool (as opposed to automated functioning of ODN) so it is similar to tools like CAM or CODE with one interesting feature: “Scalable processing: Users work with a subset of their data to define the models that integrate their data sources. This enables Karma to offer a responsive user interface when users are defining the model that integrates their data. Karma can then use these models in
batch mode to integrate large data sources.” We will mention this tool in COMSODE Methodologies, as it seems very capable for the phase of planning and designing publication processes. Depending on the further outcomes of WP3 (Tasks 3.1 to 3.3, i.e. datasets selection and ways to transform them) we may consider also some level of integration between Karma and ODN to aid automated publication of big datasets.

**LDIF**

Home page: [http://ldif.wbsg.de/](http://ldif.wbsg.de/)

“LDIF translates heterogeneous Linked Data from the Web into a clean, local target representation while keeping track of data provenance”

State:
- quality: beta
- license: APLv2

**Reusable: Yes, portions of it in ETL part.**

LDIF consists of Scheduler, data import jobs and Integration Runtime Environment. It thus implements a major part of ODN ETL part. But compared to ODCS, as “ETL runtime” it is less flexible (lacking support for data formats other than RDF, less capable administration GUI, etc.). But as LDIF concepts are quite similar to concepts used in ODCS (DPUs, etc.) we plan to reuse portions of LDIF related directly to processing of Linked Data (like provenance tracking).

**mingle.io**

Home page: [https://mingle.io/](https://mingle.io/)

This project is mainly a team of Open Data practitioners offering various services related to Open Data. But they also provide some open Source tools, one of them being bandicoot tool:

“Bandicoot is an open source programming system with a new set-based programming language, persistency capabilities, and run-time environment. The language is similar to general purpose programming languages where you write functions/methods and access data through variables. Though, in Bandicoot, you always manipulate data in sets using a small set-based algebra (the relational algebra).”

State:
- quality: production
- license: APLv2

**Reusable: Maybe, in storage and publication part.**

Bandicoot provides REST/JSON based API for data stored in it. Thus it fits into publication part of ODN. But we want to limit the number of store solutions used in ODN to very few (triple store, relational database and maybe SOLR index) so in this regards this tool is not a good fit and the plan is to not use it, unless we encounter problems with other solutions we plan to reuse for RESTfull APIs (restSQL).

**Nomenklatura**

Home page: [http://nomenklatura.okfnlabs.org/about](http://nomenklatura.okfnlabs.org/about)
“nomenklatura is a reference data recon server. It allows users to manage a list of canonical entities (e.g. person or organisation names) and aliases that connect to one of the canonical entities. This helps to clean up messy data in which a single entity may be referred to by many names.”

State:
- quality: ...
- license: MIT

Reusable: Yes, in ETL part.

We can use this service (or its implementation) in ETL part of Open Data Node to help users with conversion of non-linked data to Linked Data, allowing them to assign entity identifiers more easily avoiding unnecessary duplicates, managing them if they do occur. It is similar to Okkam ENS (but provides also source code, which is an advantage for us), so the final decision to use it will be based on results of WP3 (Tasks 3.1 to 3.3, i.e. datasets selection and ways to transform them).

**OAIHarvester2**


“The OAIHarvester2 Open Source Software (OSS) project is a Java application that provides an OAI-PMH harvester framework”

State:
- quality: production
- license: APLv2

Reusable: Yes, in ETL part.

COMSODE consortium member implemented this tool to harvest metadata about digitized cultural works for some members of Europeana project through OAI-PMH protocol. What is seen as “metadata” in Culture segment is “data” in Open Data segment. Thus we will reuse this tool allow ODN to load data from other sources using OAI-PMH based API.

**ODCS v2**


“ODCleanStore is an ETL (Extract-Transform-Load) tool for processing RDF data. Such tool supports creation of data processing pipelines consisting of data processing units (DPUs) - extractors, transformers, and loaders. The tool is easily extensible, new DPUs may be added as plugins. Pipelines may be scheduled to be executed anytime. The tool has graphical user interface for the administration, debugging, and monitoring of the ETL process.”

State:
- quality: beta
- license: GPLv3/LGPLv3

Reusable: Yes, in ETL part.
ODCS even in current beta provides a lot of functionality planned for ETL part of Open Data Node: versatile processing pipelines with support for debugging, graphical management GUI, support for processing of RDF data etc. When compared with other solutions (like Deri Pipes, “old” Open Data Node, LDIF) it provides additional benefits. Final very important aspect is, that is being developed by CUNI, a consortium member, thus giving as big leverage when it comes to steering of its development and also allows us quickly to reuse their existing know-how. Thus we chose to use ODCS in ETL part of ODN.

Similar choice was done by Semantic Web Company (associated partner of the COMSODE), which is replacing its LODMS with ODCS planning also to include ODCS into LOD2 stack in cooperation with CUNI.

**Okkam Entity Name System (ENS)**


“The Entity Name System (ENS), applying the OKKAM razor principle, makes the systematic reuse of globally unique identifiers possible. The ENS stores the identifiers for entities and provides a collection of core services to support their creation, maintenance and pervasive reuse.”

State:

- quality: production
- license: free to use service

Reusable: Yes, in ETL part.

We can use this service in ETL part of Open Data Node to help users with conversion of non-linked data to Linked Data, allowing them to assign entity identifiers more easily avoiding unnecessary duplicates. It is similar to Nomenklatura, so the final decision to use it will be based on results of WP3 (Tasks 3.1 to 3.3, i.e. datasets selection and ways to transform them).

**Open311**


“a collaborative model and open standard for civic issue tracking: Open311 is a form of technology that provides open channels of communication for issues that concern public space and public services. Primarily, Open311 refers to a standardized protocol for location-based collaborative issue-tracking. By offering free web API access to an existing 311 service, Open311 is an evolution of the phone-based 311 systems that many cities in North America offer.”

State:

- quality: depending on particular tool or library
- license: depending on particular tool or library

Reusable: Very little.

We’ve investigated Open Source components of the project to determine how much of them are usable in the process of publishing Open Data. The answer is “not much” as the core of Open311 functionality lies in integration of various applications and services directly with
information services of various municipalities and government organizations. That is something beyond the concepts of “just” publishing Open Data. But at least Open311 strengthens our commitment to provide RESTful API for the data (confirming the demand for such) and will serve as guide in particular data analysis task in WP3 (to look for data formats interoperable with Open311 specifications).

**OpenCivic**


“OpenCivic is an open source resource cataloguing, hackathon and app store management platform designed to help organizations better collaborate in developing, sharing and maintaining information and apps that solve civic problems.”

State:
- quality: production
- license: GPLv2, but not fully sure

**Reusable: Maybe, in publication part.**

As its main purpose is to document applications built using Open Data, it may complement data catalog/management functions in ODN. But such functionality is not directly in scope envisioned for Open Data Node so (based on MoSCoW methodology) the reuse is categorized as “Could” (i.e. “desirable but not necessary”).

**OWLIM Lite**


“OWLIM is a family of semantic repositories, or RDF database management systems, with the following characteristics:

- native RDF engines, implemented in Java
- delivering full performance through both Sesame and Jena
- robust support for the semantics of RDFS, OWL 2 RL and OWL 2 QL
- best scalability, loading and query evaluation performance

**OWLIM is used in a large number of research projects and software tools.”**

State:
- quality: production
- license: free for non-commercial use, not open source

**Reusable: No.**

This tool covers all relevant needs of Open Data Node for storage of Linked Open Data. Preliminary benchmarking suggests acceptable level of performance for both read and write operations. And compares well with other triplestores (BigData, Sesame OpenRDF, Virtuoso).

But is is not an Open Source solution so we will not use it in Open Data Node.
Note: Maybe in the future, based on needs of potential customers, we may attempt to adapt ODN to be able to work with this triplestore.

**restSQL**


“restSQL is an ultra-lightweight data access layer for HTTP clients. restSQL is a persistence framework or engine in the middle tier of a classic three tier architecture: client, application server and database. It may also be embedded in any middle-tier as a Java library.”

State:
- quality: beta
- license: MIT

**Reusable: Yes, in publication part.**

Usable for automated provisioning of RESTful APIs for datasets published using ODN. The main advantage compared to other solutions (like Bandicoot, The DataTank or modules from Socrata) is its lightweight implementation. The main disadvantage of not just this tool but also other similar ones is, that automatically provisioned API for given dataset will never be the best possible one, compared to API designed by developer directly for particular dataset. But such generic APIs are expected to be “good enough” for many datasets so that publishers can concentrate their resources efficiently to design high quality API only to most prominent/useful datasets.

**Sentient Data Manager**


“IO Informatics’ Data Manager provides automated input and organization of data from any source. Data Manager extracts flat files and proprietary file formats into open, compliant standards. This efficiently creates well structured and contextually integrated data objects individually or in batch, from virtually any existing data, including data acquired from instruments or third party applications.”

State:
- quality: production
- license: proprietary

**Reusable: No.**

This tool covers a lot of expected functionality of Open Data Node but more in “interactive way” (ODN is geared more for “automated mode of operation”) plus it is a proprietary tool so it can’t be reused in Open Source publication platform.

**Sesame OpenRDF**
Home page: http://www.openrdf.org/

“OpenRDF Sesame is a de-facto standard framework for processing RDF data. This includes parsers, storage solutions (RDF databases a.k.a. triplestores), reasoning and querying, using the SPARQL query language. It offers a flexible and easy to use Java API that can be connected to all leading RDF storage solutions.”

State:
- quality: production
- license: BSD-style

Reusable: Maybe, in storage part.

This tool covers all relevant needs of Open Data Node for storage of Linked Open Data. Preliminary benchmarking suggests acceptable level of performance for both read and write operations. And compares well with other triplestores (BigData, OWLIM, Virtuoso).

The only disadvantage is that ODCS is so far being implemented using Virtuoso and has not been tried with Sesame.

**Sieve**

Home page: http://sieve.wbsg.de/

“Sieve (Linked Data Quality Assessment and Fusion) allows Web data to be filtered according to different data quality assessment policies and provides for fusing Web data according to different conflict resolution methods.”

State:
- quality: beta
- license: APLv2

Reusable: Yes, in ETL part.

The tools was tried and deemed usable for quality assessment and conflict resolution task in ETL parts of Open Data Node.

**Silk**

Home page: http://wifo5-03.informatik.uni-mannheim.de/bizer/silk/

“The Silk framework is a tool for discovering relationships between data items within different Linked Data sources. Data publishers can use Silk to set RDF links from their data sources to other data sources on the Web.”

State:
- quality: production
- license: APLv2

Reusable: Yes, in ETL part.

Tools which are part of this project will be used in ETL part of ODN for linking of Linked datasets.

**Sindice**

“The Web of Data is the web composed of pages which have semantic markup in RDF, RDFa, Microformats or Microdata (e.g. schema.org). Semantic markup can be about any topic, ranging from popular ones like people, events, reviews, products, services etc., to specific scientific annotations, media, bioscience, software and more.

Sindice is a platform to build applications on top of this data. Sindice collects Web Data in many ways, following existing web standards, and offers Search and Querying across this data, updated live every few minutes. Specialized APIs, and tools are also available”

State:
- quality: production
- license: free service

Reusable: Yes, in ETL part.

Sindice provides free services available both to humans (via web GUI) and application developers (via API). APIs can be reused by COMSODE in some ETL task, mainly enrichment and linking. Final decision to reuse the services will be based on results of WP3 (Tasks 3.1 to 3.3, i.e. datasets selection and ways to transform them).

**Socrata**


- “Open Data Portal
  Data optimized for access and use

Socrata’s open data portal moves data to the cloud for everyone to review, compare, visualize, and share.”

- “GovStat
  Government decisions made with data

GovStat puts government performance management in the hands of all your employees, and shares results with the citizens you serve.”

- “API Foundry
  Real-time data for high-impact apps

APIs create instant, accurate data sharing – and more apps. With API Foundry you create custom APIs and developer documentation in minutes.”

State:
- quality: production
- license: service is commercial, but source code is available as open source (but exact license in unclear)

Reusable: No.

As similar solutions (like Junar) there is little for COMSODE to reuse: source code is available but we were not able to determine sufficiently quickly how complete and usable it is as there is no comprehensive documentation for it. We might be able to reuse only some smaller parts.
Thus the main task for COMSODE is to take into account that Socrata offers their services to some of the same groups we’re targeting (data publishers, application developers) so we need to make sure we have some “unique selling point” when offering Open Data Node.

**Talend Open Studio**


“Talend Open Studio is a powerful and versatile set of open source products for developing, testing, deploying and administrating data management and application integration projects.”

State:

- quality: production
- license: seems LGPL

Reusable: No.

Talend Open Studio, even in its limited Open Source version, provides a lot of features and capabilities we plan for Open Data Node. But like a lot of other commercial offerings which “also provide open source” (like Socrata), it is difficult to quickly assess their open source offering and gain operational and development know-how about it. Thus we do not plan to use this in COMSODE.

**The DataTank**

Home page: [http://docs.thedatatank.com/](http://docs.thedatatank.com/)

“The DataTank is an open source tool that publishes data. These data can be stored in text based files such as CSV, XML and JSON or in binary structures such as SHP files and relational databases. The DataTank will read the data out of these structures and publish them on the web using a URI as an identifier. It can then provide these data in any format a user wants, no matter what the original datastructure was. In practical terms, this means that you can provide a JSON feed on a certain URI based on data somewhere on the web say, a CSV output from a google spreadsheet.”

State:

- quality: production
- license: AGPLv3

Reusable: Maybe, in publication part.

Once ODN has some Open Data in storage (i.e. harvested and transformed) this tool can help provide RESTfull API for it if using multiple formats (JSON, XML, …) in a similar fashion as restSQL. But it is seen as slightly worse fit for COMSODE as it is implemented in PHP (we prefer Java). But, as it provides also some dataset visualization functionalities, we might reconsider this in the future based on feedback from proof-of-concept publishing project (Task 6.5).

**Virtuoso Open Source**

“At core, Virtuoso is a high-performance object-relational SQL database. ... OpenLink Virtuoso supports SPARQL embedded into SQL for querying RDF data stored in Virtuoso’s database. SPARQL benefits from low-level support in the engine itself, such as SPARQL-aware type-casting rules and a dedicated IRI data type.”

State:

- quality: production
- license: GPLv2

Reusable: Yes, in storage part.

This tool covers all relevant needs of Open Data Node for storage of Linked Open Data. Preliminary benchmarking suggests acceptable level of performance for both read and write operations. And compares well with other triplestores (BigData, OWLIM, Sesame OpenRDF).

Additionally, Virtuoso provides common SQL capabilities thus it covers also relevant needs of Open Data Node for storage of (structured, but not Linked) Open Data which is a great plus, because we do not need to introduce another storage solution into the final solution.

Finally, ODCS is so far being implemented using Virtuoso.

So we chose this solution for storage part of Open Data Node. We might re-evaluate in the future based on from proof-of-concept publishing project (Task 6.5) or interest of potential ODN users/customers (by introducing support for different triplestores - Sesame, etc. - or RDMS - MySQL, etc.).
4 Typical requirements for the publication platform

This section contains results of our assessment and analysis of information we obtained in regards to requirements for publication platform: features, use-cases, requirements, limitations, etc. (see section “Used methodology” for more details about how we obtained the information).

As we specifically target also public bodies and other data publishers, requirements are split in two main groups:

- Requirements formulated by public bodies - those are in subsection “Required aspects of the publication platform for general usage within public bodies and other data publishers”
- Requirements formulated by users of data and other entities (analysts, application developers, etc.) - those are more general in nature so they precede requirements from public bodies and are listed before them, right in the next subsection.

4.1 Aspects of the publication platform required by data users

4.1.1 Open license for data

Members of COMSODE User Group voiced strong desire for proper licensing of published datasets.

COMSODE Methodology has to clearly explain that an integral part of Open Data publishing is the selection of appropriate open license for the data, together with the need to clearly communicate this license to all users of the data (this also implies that the license must be part of the metadata stored in ODN - see sections “Provide list of datasets as Linked Data” and “Dataset metadata available via API” below.)

Subsequently, Methodology has to also give guidance to the process of license selection and give recommendation for use of particular licenses (like ODbL or CC) based on EU-wide best practises and standards (which are being established right now - see “Consultation on guidelines on recommended standard licences, datasets and charging for the re-use of public sector information”[6]).

We should also follow those recommendations ourselves when publishing datasets as part of Pilot project (Task 6.5).

4.1.2 Publication process has to be automated

Automated publication process is seen as necessary by both publishers of the Open Data and by users of Open Data:

- **publishers**: It helps to avoid repeated manual data processing task thus lowers the costs associated with publication of Open Data.
- **users**: If some data gets published manually based on proactivity of some personnel in government, the data in question is in jeopardy when it comes to timely updates. Such personnel might get replaced, might be overworked later on or in some cases might get even reprimanded for doing something outside their “usual” or explicitly defined duties.
Data published manual fashion will, with high probability, not be regularly updated and will slowly became outdated, losing its usefulness and contributing to confusion about the data itself but also about Open Data in general.

Thus, in relation/cooperation with “Binding guidance” mentioned below, we have to make sure that the Open Data publication process is automated.

For COMSODE, this means:

- Open Data Node (ODN) publication platform itself has to support automated and repeatable publication of data.
- Methodology has to stress the need for continual updates for already released data and has to put emphasis on doing it in automated way.
- Documentation of the ODN and parts of Methodology covering specific usage of ODN has to give clear guidance and examples about how to use ODN for automated and repeatable publishing of data.

4.1.3 **Datasets published as dumps**

Application developers defined requirement, that datasets (their contents) must be accessible as so called “dumps” in files. Following formats have to be available:

- CSV
- RDF

4.1.4 **Datasets published via API**

Application developers defined requirement, that datasets (their contents) must be accessible also via API:

- REST (with priority “Must”)
- SOAP (with priority “Could”)

4.1.5 **Provide list of datasets as Linked Data**

Future users of Open Data published via Open Data Node asked for:

- List of datasets in the particular ODN instance to be available too,
  - in machine readable format in general and
  - as Linked Data in particular, based on well know vocabulary.

This would help with discovery of datasets and also make it easier to find similar datasets (candidates for further linking).

List has to contain all relevant metadata about each dataset: ID, name, license, etc. See DCAT[7] for more exhaustive list of dataset metadata attributes although ODN will hold (and make available) also some more in order to implement its functions.

Essentially, such list is a dataset too. It can be generated by ODN automatically and published along with other datasets.

4.1.6 **Dataset metadata available via API**

Application developers defined also requirement, that dataset metadata (information about datasets) have to be available via API. That’s essentially API for list of datasets (see section “Provide list of datasets as Linked Data”).
Following specific functionalities were also requested via this API:

- obtain the metadata: source URL, license, type, update frequency, etc. (priority “Must”)
- obtain list of datasets (priority “Must”)
- search in the list of datasets (priority “Must”)

4.1.7 Additional dataset functions via API

Application developers formulated need for access to functions related to contents of datasets. Specifically they request for functions related to datasets that are being periodically updated. Such datasets have to be versioned (i.e. each update to the dataset must first copy existing version into archive and sufficient historical record must be kept and made available).

The following specific functionalities were also requested via this API:

- obtain a list of available versions of dataset (priority “Must”)
- obtain an exact version of a dataset as dump in a file (priority “Must”)
- obtain the latest version of a dataset as dump in a file (priority “Must”)
- obtain a so called “diff” between specified and previous version of a dataset (priority “Would”)
- search through content of a dataset, with matching results returned in JSON format (priority “Could”)

4.1.8 Notifications for new datasets and updated datasets

Application developers defined the requirement that ODN should be able to provide the following notifications related to the status of datasets in ODN instance:

- new dataset added,
- dataset updates (changes in either content or metadata).

Requested protocol: RSS, Atom or other suitable.

4.1.9 Non-invasive enhancements to datasets

ODN should be able to provide some basic transformations which increase the quality of datasets but do not change the meaning of data, for example character encoding fixes and conversions, removal of quoting, file format conversions, etc.

The main purpose is to make datasets more usable without compromising its credibility.

4.1.10 Linking of data entries to official registries

ODNs, as part of ETL and enrichment, have to provide functionality to clean-up (or even link, if the format allows is) data from processed dataset to official registries (i.e. other datasets).

Example: ODN has to be able to clean, normalize (and link, if say RDF format is used) ZIP numbers in the list of addresses from one dataset using official list of ZIP numbers in other dataset.

4.2 Required aspects of the publication platform for general usage within public bodies and other data publishers

This section contains results of our assessment and analysis of information we obtained particularly from public bodies in regards to requirements for publication platform: features, use-
cases, requirements, limitations, etc. (see section “Used methodology” for more details about how we obtained the information).

4.2.1 Binding guidance for government officials and clerks

As investigated in Slovak Republic, government officials and clerks are permitted to do only what is explicitly written in law as their responsibility. The same or similar situation applies also in other countries of the EU.

Thus:

- **COMSODE in general:** We need to include in COMSODE Methodology as much guidance information, references to standards, best practises and (most importantly) references to relevant EU-wide legislation (PSI directive, etc.) as possible. In regards to PSI and related directives, we need to also stress mandatory requirements of this EU-wide legislation it places on various public bodies.

- **MoI SR in Slovakia, other partners in their respective countries:** Add as much relevant country specific information as possible (reference to OGP plan, national standards for government Information Systems, etc.). And then convince national and municipal governments to adopt the Methodology (or at least parts of it) as official recommendations.

4.2.2 Anonymization requirements

As the publishers bear responsibility for what and how they publish and as there are various national and EU-level regulations regarding personal data, anonymization is seen as important part of both COMSODE publishing platform and Methodology. Most of publishers that already started to publish Open data identified anonymization as a very important part of the process. Anonymization ensures that personal data is protected (not published) but that a usable portion of data is still released to general public.

Examples of desired features:

- ETL part must be able to skip some rows and columns from source raw data.
- ETL part must be able to introduce some noise into transformed data.
- ETL part must be able to randomize transformed data.

ODN implementation has to provide sufficient features for proper anonymization of data.

ODN documentation and Methodologies have to give clear guidance about how to use those features.

4.2.3 ODN implementation has to comply with laws and regulations

As was stated in section “Binding guidance for government officials and clerks”, there are regulations and laws to be followed when publishing Open Data (in particular) and construct Information Systems in public bodies (in general). ODN has to be able to comply with such laws and regulations - EU-wide but also local ones.

As exhausting audit of all such laws and regulations in whole EU would not fit into COMSODE budget and time frame, we will mainly:

- Analyse EU-wide directives and regulations (like Directive 95/46/EC etc.),
● Analyse laws and regulations applicable to project members and associated partners (for example Italian Article 68 (Comparative analysis of solutions) - Leg. 82/2002 or Slovak law number 275/2006 Z.z. and related decrees),

● Reflect findings in ODN implementation,

● Document the findings in ODN documentation and Methodology.

ODN cannot comply with all relevant laws - but must have potential to be tuned (configured) accordingly.

**Example:** Technical documentation for “Spáva základních registrů” (“National Registers Authority”, project of Czech government)[8] published parts of its technical infrastructure that demonstrates what kinds of documentation is (might be) required by regulations in practice. Documents published at [http://www.szrcr.cz/vyvojari](http://www.szrcr.cz/vyvojari) cover various particular topics as required in Czech Republic:

- etc.

### 4.2.4 ODN integration with internal systems has to be easy and secure

Integrating something directly with internal systems is a difficult and lengthy process at majority of public bodies.

But as efficient and automated publication of Open Data does often require such close integration (direct access to database, in most cases) we have to make this process as easy as possible but also secure. Thus:

- ODN has to have little footprint (low resource requirements: CPU, storage, …).
- ODN architecture has to be simple and clearly documented.
- ODN has to be able to easily integrate with common types of databases and document stores.
- ODN has to be able to guess and recommend good settings for data publishing in order to make the process of setting up a dataset publication quick and easy (while still correct and secure).
- We have to take into account also a high probability, that a person configuring ODN might not be same as the person with deep knowledge of systems and databases that are being integrated. ODN have to provide aids to simplify this task to such persons (but we have to be clear, that ODN is not supposed to be used by unskilled personnel - such people should not be tasked with installing ODN and also should not have administrative access to databases and application servers in the first place).
● ODN not having adverse effect on the systems being integrated with is of paramount concern. It needs to be ensured and stressed that:
  ○ Publication of Open Data itself is not undermining security of the internal systems from which data comes from.
  ○ High demand for Open Data does not increase the load on internal systems.
● We need to supply easy to fill templates for official paperwork required by respective EU and national laws and regulations, for example “Bezpečnostný projekt” (“Security project”) required in Slovak government organizations (which is required for all information systems deployed there) as part of COMSODE Methodology. Guidance about what and how to fill and where to submit it is needed too. Note: The extent of what will be covered by our templates will be limited by assistance we get from MoI SR and other public bodies that will assist us as associated partners.
● We need to promote the “Open Data ready” concept to make such ODN integration in the future either much more easy or avoidable altogether.

Note: Later on (in section “Common integration strategies”) we classify this kind of integration as “tight”.

4.2.5 TCO for ODN has to be low

Limited budgets for IT in public bodies constitute a reality we cannot ignore. Thus we have to make sure that total cost of ownership (TCO)[9] for ODN is as low as possible: while the publication platform as a set of software components and documentation is free, public bodies will need to invest into personnel (for deployment and maintenance of the system), and hardware (thus, the hardware requirements of the platform should be kept low) which means we have to implement and document ODN so that it is easy to install, maintain and uses hardware resources efficiently, etc..

This is also closely related with the requirements mentioned in section “ODN integration with internal systems have to be easy”.

4.2.6 High availability

Potential publishers expressed need for “24x7” availability of the publication platform. The motivation is quite clear: nobody will use the data if it is not available for prolonged periods of time without prior warnings, etc. So this is also a requirement shared with potential users of data.

Achieving high availability is not “cheap”. It is also not purely a problem of software itself (as the availability depends also on hardware, internet connectivity, power supply, etc.). To some degree, this requirement conflicts with requirements stated in section “TCO for ODN has to be low”.

But COMSODE can still partially address this requirement:
  ● Implementation of ODN has to be simple and robust.
  ● Implementation of ODN has to allow to be deployed in high-available cluster.
  ● ODN documentation and Methodology have to give sufficient guidance about high-availability deployment options.

4.2.7 Scalable implementation

Potential publishers expressed the need for scalable implementation, which can deliver data also when requests come in frequently and the volume of data is high.
This is quite similar to what is described above in section “High availability”: scalability is determined not just by software, but also by hardware it is running on and other factors.

COMSODE will make sure that:

- ODN design will be scalable.
- ODN implementation will be able to use resources at disposal to particular instance to deliver data even under high load.
- ODN implementation will take advantage of HTTP caching in proxies, etc.
- ODN documentation and Methodology will give guidance about how to choose hardware and how to tweak configuration for higher loads.

### 4.2.8 Operation behind firewall

As a specific subcase of what is described in sections “Publication process has to be automated”, “ODN integration with internal systems have to be easy” and others, we need to specifically cover a case of ODN operating “inside” protected government networks, behind their firewalls.

In order to get a data, ODN has to have access to internal databases or other internal systems (or be accessible from them). In order to publish a data, ODN has to be reachable from Internet. Thus, ODN has to support deployments where some of its parts might be on different servers, in different security zones, DMZs, etc.

This support has to be aligned with current practises regarding websites of public bodies, which are - from the point of data flows - a very similar problem.

All that needs to be properly documented.

### 4.2.9 Well defined release policy and planning for ODN

Each software is subject to small or even bigger updates throughout its lifetime. Potential customers require upgrades to be “backward compatible” so as to ensure that a service provided by an older version is still available to the users even after an upgrade to the newer version.

COMSODE will take this requirement into account and once there are ODN instances actually deployed, we will make sure upgrades do not disrupt functionalities already in use.

Data publishers publishing data via ODN publication platform also need to know exact, up-to-date state of implementation and need to easily find out what was changed in the latest version compared to previous version.

Documenting such information is standard practise both in software development in general and in the Open Source development model in particular. COMSODE will follow the best practises in this regard.

### 4.2.10 Support for ODN has to be quick and of high quality

Potential users of ODN will need quick and high quality support for it.

As ODN is being developed as Open Source there are more options about how COMSODE can meet this requirements:

- for users which download, install and maintain the Open Source software by themselves: COMSODE will provide support in ways usual for Open Source projects, aiming for high quality and quick response (as this is needed to build up community around ODN) but
through limited channels due to costs factors (we will have to pay for it, not the user): public Wiki, bug tracker, etc. and support directly from developers and other project participants (but on level considered “fair” given that the user is not paying for the service)

- for users who purchase the support from a COMSODE member: support will build on top of what is described above but a Service Level Agreement will be provided based on negotiations between the user and a COMSODE member (giving customer for example higher priority, guarantees limiting for response time etc.).

**Note:** As ODN will be Open Source, also other organizations and even individuals will be able to provide premium paid support for it. That is good for users, widening their options, avoiding vendor lock-in, etc. (as mentioned in the DoW).

### 4.2.11 Support for a wide range of dataset types

Public bodies expressed the need for ODN to support both static and real-time data sets, as well as both small and large datasets.

COMSODE defined criteria for dataset selection (see “Deliverable 2.2: Criteria for selection of datasets”) to also give preference to a wider range of types of datasets. We will not be able to cover all the extreme cases (e.g. truly big amount of real-time data with high velocity) but we will implement and validate most common cases in the Pilot project (Task 6.5).

### 4.2.12 “Open Data ready” concept

In relation to “backup solution” mentioned in section “ODN integration with internal systems has to be easy” we presented to our partners the concept of “Open Data ready” (quoting the DoW):

“As part of a wider strategy, the term “Open Data Ready” was coined in relation to future procurement of Information Systems by public organizations in Slovakia. It generally means that all newly purchased systems need to have interfaces that can be used to get the data from those systems and publish it as Open Data easily. The COMSODE project (implementation of the publication platform, methodologies and the pilot project) will help to clarify and test in practice what exact definition for “Open Data Ready” is needed, borrowing heavily from standards and techniques which will be part of methodologies delivered by the project.”

Many interviewees agreed that this approach will help with Open Data publishing, either with Open Data Node or with some other solution.

This confirmed our vision and plans with this concept and also provided some clarifications:

- “Open Data ready” concept has to be properly documented in Methodologies: what it means, to what information systems and situations it applies, technical parameters (API, dataformats, expected functions, etc.) and examples of what formulations should be added to procurement materials.

- Open Data Node has to be “Open Data ready” itself, i.e. it has to be able to both provide data (to others) and harvest data (from others) that way.

- The concept should be validated in practice to gain validity.

- In conjunction with “Binding guidance” mentioned above, the concept should be incorporated also to EU or national procurement and IS standards.
4.2.13 **Clear explanation of Open Data, the motivation and benefits**

We’ve heard several common questions from many interviewees:

- Why to publish Open Data?
- What are the benefits of Open Data?
- And who benefits?

COMSODE Methodology in general has to properly address and answer those questions.

For example, it has to explain how the benefits of publishing Open Data grow with the amount and variety of (linked) datasets that are published by as many organizations (and countries) as possible. That is, each particular dataset counts, but not by just itself. This also implies explaining basic concepts and benefits of Linked Data, in layman language.

4.2.14 **Clear explanation of Open Data publishing process**

We have heard several common questions from many interviewees:

- Who is the owner of published data?
- What legislation applies to the publishing process?
- What legislation and licensing applies to published data?

COMSODE Methodology in general has to properly address and answer those questions.

As Open Data Node is deployed as open source software - not as a service - such questions need to be answered also for various ODN deployment options, for example:

- Who is owner and who is responsible when an organization is publishing its own data with ODN instance, which they installed and are maintaining themselves?
- What changes if they publish their data using ODN instance maintained by some other organizations?
- Who is responsible for the software (its proper functioning and security, etc.) if an organization downloads it and uses it for free?

4.2.15 **Publisher of the data remains owner of the data**

Publishers who are potential users of ODN want to remain the owners of the data they publish.

As COMSODE is supplying only software, documentation and Methodology and is not planning to assert any ownership for data in any ODN instance (we even do not think it is legally possible to do such a thing), this requirement can be met without massive effort.

What needs to be done:

- State clearly and properly in Methodology that “ownership of the data will not change simply because ODN is being used to publish it”. This is essentially a sub-part of the requirements described in section “Clear explanation of Open Data, the motivation and
benefits” and “Clear explanation of Open Data publishing process”. Make sure data publishers know this.

- Make sure that dataset ownership information is tracked using dataset metadata in a similar fashion as information about license. I.e ODN instance have to store this information and make it available to users.
- Methodology has to explain why and how to manage ownership information both in general terms and with ODN in particular.

4.2.16  Dataset wishlist

Open Data initiatives usually assume that the social added value of Open Data - such as higher transparency, lower corruption, more jobs, higher GDP, etc - can only be observed as a mid-long consequence of the actual publication of such Open Data. Such assumptions are nowadays (after few years of Open Data practise) also validated by some studies - TODO: add some references.

On the other hand, many public bodies are still reluctant to release data without concrete reasons: Why to release exactly this data in this form? What good it will do? To whom? Etc.

So, in addition to measures mentioned in section “Binding guidance” we can also strengthen the argument for publishing of Open Data by documenting a feedback loop from citizens and other potential users of data to data publishers (public bodies in particular) - as mentioned in DoW: “solicit public demand and capture user feedback”. In particular, we will document also a “dataset wish lists” as a way to collect requests for data from future users. In this wishlist, information about reasons and potential benefits of a dataset can be documented.

We should also consider implementing such mechanism in ODN, even if it is not directly related to technical aspects of publishing Open Data.

4.2.17  Feedback about quality of data

Several publishers expressed the need to solicit feedback from users about the quality of data being published by ODN instance.

Such feedback loop is not purely in the domain of the publishing platform but - similarly to what is described in section “Dataset wish list” - we will document this in Methodology (the need for feedback mechanism and ways to implement it) and also consider implementing a feedback mechanism into ODN itself.

4.2.18  First impression

COMSODE Methodology and ODN documentation is expected to be extensive. But people in public bodies (and elsewhere) will make a “first impression” based on the first picture or scheme they will find.

Thus we need to make sure that:

- Every ODN user, coming from any environment (and public bodies in particular) has a document dedicated for his/her needs.
- First scheme in such document has to be sufficiently clear and informative for this particular kind of user.

4.2.19  “What are the benefits for me?”

Open Data is seen mainly as benefits to citizens and companies. Publishers, especially those from public bodies are usually left with just the costs and “do it for public good” arguments.
To increase the chance of ODN adoption (and thus increasing the chance of new Open Data to be published) we need to provide some enticements also to public bodies themselves. ODN has to incorporate some functions that provide some benefits to them. Examples:

- Based on “Open Data ready” and “Open Data as integration platform” concepts, we should make an argument about more efficient and less costly integration of their own information systems in the future allowing them to do more with same budget.
- Internal data catalogue can aid data management (data governance) within organizations. Many organizations currently do not have sufficient documentation about what data they have, in which systems, in what format and quality, without clear documentation about that is responsible, who is the supplier etc. So such functionality should help.

### 4.2.20 Graphical content, attachments

Several potential publishers expressed the need to publish also some graphical data (for example scans of contracts) and other files (attachments) as part of some datasets (list of contracts).

A special kind of graphical data are “land use plans” which usually consist of very large JPG or TIFF files and special tools are needed to actually display them properly, zoom in and out, move the view within them.

As of now, we are planning to process mainly tabular data, so such attachments are seen as “out of scope”. But given the expressed interest from potential users of the publication platform, we will look into this in the future if we will still have some resources to spend (i.e. “could” priority for this item in Requirements list) and it will be pursued as follow-up Exploitation strategy also outside of COMSODE project.
5 Assessment of environments within selected organizations

This section contains results from the assessment and analysis of information we obtained from public bodies in regards to their IT environments, where the publication platform is expected to be deployed (see section “Used methodology” for more details about how we obtained the information).

There is a plethora of technologies being deployed in public organizations both at client and server side:

- **Client side:**
  - Various desktop operating systems, with Windows having clear majority, but with many version being used: from now ancient and soon to be unsupported Windows NT to latest Windows 8, with Windows XP, Windows 2000 and Windows 7 having major shares
  - The vast majority seems to be able to maintain most recent versions of browsers on their desktops: Firefox, MS IE, Opera

- **Server side:**
  - Various operating systems ranging from recent versions of Windows Server or Linux through Mac OS X to old versions of the same or even “ancient” operating systems like VMS or NetWare, with UNIX based and Windows based server having majority
  - Virtualization technologies are quite common

Developing our Open Data Note as PC client applications would be very difficult as the deployment and maintenance (for various operating systems) would be expensive (also, the server side would still be needed to deliver the data to users). Organizations would prefer server side software that provides a web GUI.

From the server side perspective, there is no clearly preferred platform (neither Windows, Linux or Unix). The majority of organizations are able to accommodate any major server architecture, be it Windows with .NET or Linux with LAMP stack or J2EE under any major OS.

On the other hand there is also a question of skilled staff (mainly administrators): not many organizations have staff that is able to properly support both UNIX- and Windows- based servers. In cases when internal staff does not have the know-how required for a particular solution (for example ODN based on Linux), the price for the solution will have to be higher as external support has to be purchased.
### 5.1 Map of typical environments

This is a scheme of typical deployment of Enterprise Portal, i.e. server side application with web GUI (which is conceptually very similar to ODN):

Public bodies are able to accommodate solutions with such architecture with all major combinations of application containers (J2EE, .NET, etc.), database back-ends (Oracle, MS SQL, PostgreSQL, etc.), Operating Systems (Linux, UNIX, Windows, etc.) and other components, as needed by application. Contemporary virtualization solutions are making this even easier.

Our current strategy employing server-side implementation based on J2EE with web based GUI is in line with this. But in order to accommodate cases when ODN user is not able to support certain operating system "in house" (as mentioned in assessment part of this section) we need to put more emphasis to ODN being multi-platform, i.e. mainly to run properly on both UNIX based systems and under Windows so that we can ensure lower TCO for wider set of organizations.

#### 5.1.1 Cloud

COMSODE by itself is not targeting to provide ODN as a Cloud service. It may be part of Future exploitation strategy but as mentioned in DoW, but in the scope of COMSODE project we’re
going to release just the software. And this will be our differentiation strategy ("unique selling point") in relation to Cloud based publication services.

On the other hand, in addition to using virtualization technologies in their own environments (creating so called Virtual Machines and using them to share common hardware for diverse software platforms, i.e. to collocate services), public bodies and other organizations are embracing also Virtual Machines from Cloud providers like Amazon (S3) or similar offerings from many other providers.

We will test and document ODN installation also on such offerings to make sure ODN can be used “in the Cloud" (see COMSODE Description of Work, page 48, part “Positioning (unique selling points) of Open Data Node”).

6 Common integration strategies

We will describe common integration strategies using basic deployment options - see section “Hierarchies, actors and typical functions of ODN”.

For more details about ODN, its basic use-cases and preliminary architecture, please refer back to section “Preliminary high-level overview of ODN”.

6.1 Tight integration, at the publisher’s premises

Scheme depicts the space where this strategy can be used: when the publisher itself is running ODN using any interfaces judged feasible to harvest raw source data
Main points:

- Open Data Node is tightly integrated with the publisher's internal application(s) - this means direct access to back-end databases or integrated into application(s) workflows via API (i.e. **new integration interface in designed and implemented** to enable publication of Open Data using ODN)

- Open Data Node is deployed alongside the publisher's internal application(s)
  - precise structure of deployment needs to take into account network, security and other zones

Typical scenario:

- the publisher wants to achieve high quality and efficiency and is willing to invest more

Prerequisites:

- the publisher is willing to update existing workflows and applications

Examples: In all cases, ODN is being maintained by same organization whose internal systems are being integrated with. As usual with any IT solution, ODN can be either purchased (as end-to-end or partial solution) from some vendor or organization can implement it by itself using their own staff and resources.

- ODN harvesting data from selected databases, tables and views in internal data warehouse
- ODN harvesting data from XLS files from internal shared network drive
6.2 Loose integration, at almost any location

Scheme depicts the space where this strategy can be used: when the publisher itself is running ODN or when some other entity is harvesting data from the publisher using Open Data interfaces

Main points:

- Open Data Node is integrated with the publisher's application(s) in a loose way harvesting data from some **periodical data dumps or via API which already exist** (i.e. were reused from some other solution, Open Data or not)

- Open Data Node can be deployed in several locations:
  - at publisher's premises - access to the data secured in a similar fashion as in case of tight integration
  - at collocated housing, for example datacenter shared by multiple government organizations - access to the source data secured for example using combination of IPsec, HTTPS and access controls (authentication and authorization)
  - in the Cloud - access to the source data secured with just HTTPS and access control (not suitable for sensitive data or sensitive internal systems)

- systems which are a source of the data may use ODN or some other means to publish data
Typical scenario:

- the publisher wants to achieve high quality and efficiency but has limited resources, so changes to the existing infrastructure and applications have to be limited
- aggregator affiliated with one or more publishers and willing and able to invest into tighter integration with them (ministry aggregating data from municipalities or SME planning to make business using aggregated data)

Prerequisites:

- the publisher is able to do minor modifications to existing workflows and applications

Examples: The main point are that systems used as source of the data were not altered, or were altered only in minor way (extended configuration, new user account added, etc.) but there is some sort of agreement or contract between the publisher and user. Systems providing the data may or may not provide Open Data.

- ODN maintained by ministry is harvesting and linking together data from several of its agencies using existing data integration interfaces developed previously as part of internal reporting system
- ODN maintained by application developer is harvesting, converting and integrating together data from several public bodies for use in his application employing Open Data APIs provided by public bodies. API key obtained from public bodies is used to gain increased data transmission quotas.

6.3 No explicit integration, deployed usually at 3rd parties
Scheme depicts the space, where this strategy can be used: when some other entity is harvesting data from the publisher or aggregator using Open Data interfaces

Main points:

- Open Data Node is not explicitly integrated with publisher's applications, other existing means are used to get access to the data (either Open Data or other format or API, at worst case scraping of data from website)
- deployed usually at 3rd party:
  - their own hardware
  - collocated housing
  - Cloud
- systems which are a source of the data may use ODN or some other means to publish data

Typical scenario:

- the publisher wants to publish Open Data but has severely limited resources or options, so changes to existing infrastructure and applications have to be preferably none
- a 3rd party aggregator or application developer wants to use data from one or more publishers but for some reason is not able or willing to implement a tighter integration with them

Prerequisites:

- some usable form of access to source data is possible without changing existing workflows and applications

Examples:

- Publisher is loading the data manually into ODN via administrative web GUI. Subsequent data processing and publishing is automatic. No changes in other systems of the publisher are necessary as he is simply taking files which already exists and loading them into ODN without any automated integration.
- ODN maintained by application developer is harvesting, converting and integrating together data from several public bodies for use in his application employing Open Data APIs provided by public bodies. API is used anonymously, without API key, common data transfer quota for general public applies.
- NGO is using ODN to scraping (and then clean and link) data from websites of several public bodies for use in their data analysis tasks.
7 Requirements List

The Requirements List attached to this Deliverable as ANNEX1 contains all requirements that the publication platform has to or is supposed to satisfy. The MoSCoW Method[12] was used to document the requirements. This list is preliminary, as it will be further refined based on subsequent Analysis and Design of the Publication platform and will also incorporate feedback from proof-of-concept projects (Task 6.5). Requirement List will then serve as input for development of software (WP4), development of Methodology (WP5) and for testing (WP6).
8 Summary

Based on project plan, preliminary use-cases and architecture of Open Data Node publication platform (enhanced by the feedback obtained from members of User Board, other potential users of the platform and public servants), the deliverable summarizes the following:

- Assessment and selection of existing tools, results of research projects and services suitable for reuse by COMSODE: 
  We selected CKAN data catalogue (pending further study), CUSTODEA OAI-PMH provider, LDIF (portions of it), Nomenklatura, OAIHarvester2, ODCS v2, Okkam ENS, restSQL, Sieve, Silk, Sindice and Virtuoso Open Source for reuse - see section “Existing tools, solutions and projects”,

- Additional requirements from potential users, both data publishers and data users 
  Critical mass of user requirements has been collected and will be continuously updated during the lifetime of the project - see section “Typical requirements for the publication platform”,

- A narrowed map of expected environments: 
  We will implement server-side solution based on J2EE with web based GUI and with emphasis on multi-platform support (mainly UNIX based and Windows) - see section “Map of typical environments”,

- Definition of three basic common integration strategies: 
  (1) Tight integration, (2) Loose integration and (3) No explicit integration - see section “Common integration strategies”.

All findings have been summarized in the COMSODE Requirements List (ANNEX 1).
9 References

[1] Extract, transform, load (ETL)
http://en.wikipedia.org/wiki/Extract,_transform,_load

[2] Semantic Web


[4] OWL Web Ontology Language
http://www.w3.org/TR/owl-features/

[5] Linked Data
http://en.wikipedia.org/wiki/Linked_data

[6] Consultation on guidelines on recommended standard licences, datasets and charging for the re-use of public sector information

[7] DCAT vocabulary
http://www.w3.org/TR/vocab-dcat/

[8] National Registers Authority (CZ)

[9] Total cost of ownership (TCO)

[10] Content Management System (CMS)
http://en.wikipedia.org/wiki/Content_management_system


[12] MoSCoW method
http://en.wikipedia.org/wiki/MoSCoW_Method