A generic data-api for implementing GSIM:

Linked Data Store

**Keywords: GS**IM, Linked Data, OpenSource, Java 11, REST, RAML, JSON, JSON-LD, Docker,

# Introduction

In modernization program of Statistics Norway the information platform is a key component. The platform is intended to logical store all metadata and data produced in the production of statistics; however, we have failed to discover a reusable implementation that fits the envisioned microservice architecture. In this paper we introduced the Linked Data Store (1)

The Linked Data Store is designed and developed to support storage of structured metadata and data based on the GSIM standard (2). in which forms the basis for the Logical Data Model — the concrete domain specific model to be used in the national statistic office

In Chapter 2 we discuss the drivers, motivation and main requirements for developing LDS. In Chapter 3 we will present the Linked Data Store. In chapter 4 we present the conclusion and further work.

# Drivers and motivation for LDS

## Technology and vendor independent

Different physical data storage technologies have their strengths and weaknesses. To ensure louse coupling to technology and vendors, the data storage need to support multiple data storage technologies through abstractions like services.

This provides flexibility, while also providing better cost control (in a cloud context) about where data is located. One example is data storage such as AWS Glacier, which is a very cheap data storage service, but the disadvantage is that reading from data storage takes a long time. For example, a similar service could be used for long-term storage of data where access to data occurs very rarely.

## Metadata driven

To ensure negligence and traceability in our data while ensuring reusability, data and metadata should be integrated. Data processing should be done through a standardized streamlined production chain that uses metadata for access to data.

## Standards

By designing and sharing new business tools according to common standards, statistical organizations benefit from each other’s’ expertise. (3).

The physical data model in the information platform will be based on GSIM a conceptual model that provides a set of standardized, consistently described information objects, which are the inputs and outputs in the design and production of statistics ([4)](https://kuscholarworks.ku.edu/handle/1808/11045) forms the basis for the physical data model of the information platform.

A microservice architecture is chosen and Rest is the defacto architectural pattern for API

* The data API should use REST and designed as a microservice

UNECE (United Nations Economic Commission for Europe) also prescribes the use of Linked Data for modelling statistical metadata as one of one of its principal aims.” (5)

According to Page et. al (6) REST and Linked Data, principally centred on the notion of resources and the relationships between them. There is, however, a key difference in the motivation for resource identification:

* RDF and ontologies, resources are identified to encapsulate the underlying data model. While Linked Data extends this idea so that sections of the model can be retrieved by dereferencing resources, linking in the returned representation is used to bind sections of the model rather than transition state.
* RESTful systems, resources and their relationships are identified and exposed to enable a client to retrieve data and transition to other resources; in effect, they define an API to enable application operation and state transition. Linking is the mechanism to navigate the API; link relations encode semantics to enable this.

# Results

## About LDS

The Linked Data Store is developed and designed to support storage of structured metadata and data based on the GSIM standard, in which forms the basis for the Logical Data Model — the concrete domain specific model to be used in Statistics Norway.

Linked Data Store (LDS) manages structured data by using JSON Schema and a directive to describe linking of JSON objects with another. It is a JSON store bound to a well-defined resource-oriented REST API. The LDS is smart and allows you create data in a free order. It supports creation of data that links to other data, even before its linked data has been created.

LDS supports a Generic Logical Data Model. RAML is used as a modeling tool. LDS converts RAML files to JSON Schema which forms the physical data model. Figure 1. Illustrates the process.

,

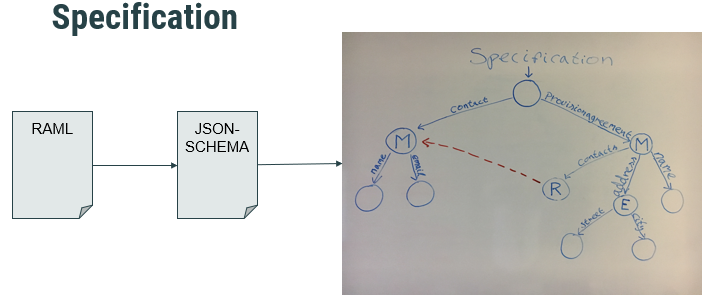


Figure 1. Specification

LDS consist of three data entities: *Managed (GSIM Objekt) (figure 2), Embedded (Properties) (Figure 3), References (Linking) (Figure 4).*

|  |  |
| --- | --- |
|  |  |
| Figure 2. Managed resource | Figure 3. Embedded Resource |

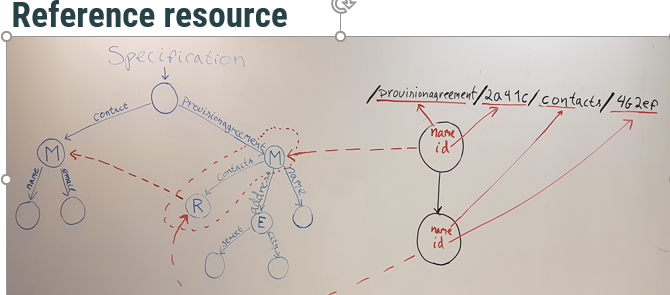


Figure 4 Reference resource

## Technolgy architechture

REST API: *Idempotent - retry safe; One HTTP status regime; Easy to use and unambiguously*

Distributed Saga: *Coordinated transaction (SEC + log), Data Recovery per se, Backup / Restore per se (7)*

Java stack: Java 11, *Undertow web server, Multiple Java libraries, No use of Spring Framework*

## Performance

The LDS is designed for high performance, we have tested latency and throughput.

Latency is the time required to perform some action or to produce some result. Latency is measured in units of time -- hours, minutes, seconds, nanoseconds or clock periods.

Throughput is the number of such actions executed or results produced per unit of time. We have measured in messages pr second.

Table 1 shows the test result for the LDS with the chosen adapters. The stress test is performed with 50 clients.

|  |  |  |
| --- | --- | --- |
| Database | Throughput (messages/second) | Latency (milliseconds) |
| Local memory | 16.000 | 1,5 |
| PostgreSQL | 3.200 | 4 |
| Neo4J | 3.500 | 7 |

Both PostgreSQL and Neo4J has more than acceptable performance.

# Conclusions and further work

## Conclusion

LDS is a high-performance generic data rest API service with a single interface. It is designed for graphy data/metadata. The LDS is easy to adapt to different backend storage solution, two adapters is developed . The LDS is open sourced and published on Github free for use under the apache licence by other statistical offices who have need of a Linked Data Store. Together with complete RAML model of GSIM, we think the reuse potential is high.

## Further work

Further work is required and planned.

* LDS should have support for clustering to achieve horizontal scaling.
* As a supplement to REST,
* GrapQL should be examined for server-side runtime for executing queries. Security must be built into the service.

All improvements will be regularly published to github.

# References

1. Statistic Norway (2018) Linked-Data-Store\*, https://github.com/statisticsnorway
2. UNECE, Clickable GSIM, (2018), https://statswiki.unece.org/display/GSIMclick/GSIM+on+a+page page-page.
3. UNECE, Building better tools, together: The CSPA project , (2018), <https://statswiki.unece.org/download/attachments/112132835/CSPA%204-%20article%20by%20Peter%20150722%20%281%29.docx?version=1&modificationDate=1450906225900&api=v2>
4. Lalor, Vale, Gregory. (2013). Generic Statistical Information Model (GSIM). Paper presented at the North American Data Documentation Initiative Conference (NADDI 2013), University of Kansas, Lawrence, Kansas, April 3, 2013. <http://hdl.handle.net/1808/11045>
5. Dreyer, Duffes, Gillman (2016), An OWL Ontology for the Generic Statistical Information Model (GSIM): Design and Implementation, Proceedings of the 4th …., ceur-ws.org
6. Page, Roure, Martinex (2011) REST and Linked Data: a match made for domain driven development?, WS-REST '11 Proceedings of the Second International Workshop on RESTful Design, p. 22-25
7. Merzky, Weidner, Jha (2015) , 5. SAGA: A STANDARDIZED ACCESS LAYER TO HETEROGENEOUS DISTRIBUTED COMPUTING INFRASTRUCTURE, Software X, Vol 1-2, p. 3-8

### 

### **[An OWL Ontology for the Generic Statistical Information Model (GSIM): Design and Implementation](http://ceur-ws.org/Vol-1654/article-03.pdf)**

Dreyer, Duffes, Gillman… - Proceedings of the 4th …, 2016 - ceur-ws.org