

# Demonstrating The Entity Registry System: Implementing 5-Star Linked Data Without the Web

Marat Charlaganov<sup>1</sup>, Philippe Cudré-Mauroux<sup>2</sup>, Cristian Dinu<sup>1</sup>,  
Christophe Guéret<sup>1</sup>, Martin Grund<sup>2</sup>, and Teodor Macicas<sup>2\*</sup>

<sup>1</sup> DANS, Royal Dutch Academy of Sciences—The Netherlands  
{`firstname.lastname`}@dans.knaw.nl

<sup>2</sup> eXascale Infolab, University of Fribourg—Switzerland  
{`firstname.lastname`}@unifr.ch

**Abstract.** Linked Data applications often assume that connectivity to data repositories and entity resolution services are always available. This may not be a valid assumption in many cases. Indeed, there are about 4.5 billion people in the world who have no or limited Web access. Many data-driven applications may have a critical impact on the life of those people, but are inaccessible to such populations due to the architecture of today’s data registries. In this demonstration, we show how our new open-source ERS system can be used as a general-purpose entity registry suitable for deployment in poorly-connected or ad-hoc environments.

## 1 Introduction

There is an estimated number of 2 billion individuals who have access to the Internet and can thus use centralized cloud hosted solutions for sharing data. Many of these centralized solutions are well-known (Facebook, Wikipedia, WikiData, etc.) and make it possible to share semi-structured data about entities. Linked Data comes into this picture as a solution to interlink the isolated data silos by linking those entities through semantically rich connections. The expected outcome being a globally connected data space everyone can contribute to.

Unfortunately those who do not have access to seamless data connectivity and web hosting services can not benefit from Linked Data. Even when computers are interconnected through local mesh networks, the dependency on web platforms makes it impossible to de-reference the description of an entity.

For example, let us consider the case of the XO laptops deployed by the OLPC (One-Laptop-Per-Child) foundation<sup>3</sup>. OLPC brings Information and Communication Technology (ICT) to young learners in the poorest areas of the world so that they can develop new skills and work collaboratively using multimedia applications. So far, two million children world-wide have received an XO and use it to work with their peers. Data-sharing is however limited to synchronous messages using XMPP-based channels between two running instances of an application. In

---

\* Authors are listed in alphabetical order.

<sup>3</sup> <http://one.laptop.org/>

this context, the asynchronous editing of a database shared by different applications is a challenging architectural problem. External data-hosting, pre-defined schemas and data-caching can be a solution: “Sugar Network”<sup>4</sup>, a data-sharing service built for Sugar—the learning environment of the XO—implements such a platform for community support.

This kind of approach is however limited in scope and requires to have some connectivity to the central server. The goal of the Entity Registry System (ERS)<sup>5</sup> is to provide a lightweight, versatile, linked data publication tool that does not rely on third party data hosting or services. ERS replaces the Web as a platform for publishing linked data. It lets a swarm of small devices interconnected in an intermittent way create/update/delete entities within a globally shared data-space. By having the triples hosted directly on the machines creating them, the system supports different connectivity contexts.

ERS tackles one of the three challenges for accelerating the adoption of Linked Data and data-intensive applications in developing parts of the world [1, 3].

## 2 The Entity Registry System (ERS)

ERS is designed around lightweight components: Contributors, Bridges, and Global Servers, which collaboratively support data-sharing and data-intensive applications in intermittently connected settings. It is compatible with the RDF data model and makes use of the available connectivity to share data, but does not base its content publication strategy on the Web. No single component is required to hold a complete copy of the registry. The global content consists of the union of what every component decides to share. We hereafter briefly describe the components and the implementation. The interested reader is invited to consult [2] for more details on the system and on performance considerations.

### 2.1 Components

**Contributor:** Contributors read and edit the content of the registry. They may create and delete entities, look for entities, and contribute to the description of entities. Every contribution is identified by the contributor name so that the collectively-created description of an entity can be traced back to individual contributors. Contributors are free to make any statement about any entity in the system. They use a local data-store in which they persist their contributions to the description of the entities. They may also cache the contributions of others when appropriate.

**Bridge:** Bridges do not directly contribute to the content of the registry. They are used to connect isolated closed networks and improve the availability of the individual descriptions shared by the contributors. Bridges can theoretically store content coming from any contributor, but will typically store the data only for a limited amount of time.

---

<sup>4</sup> [http://wiki.sugarlabs.org/go/Sugar\\_Network](http://wiki.sugarlabs.org/go/Sugar_Network)

<sup>5</sup> <http://worldwideseanticweb.org/projects/entity-registries/>

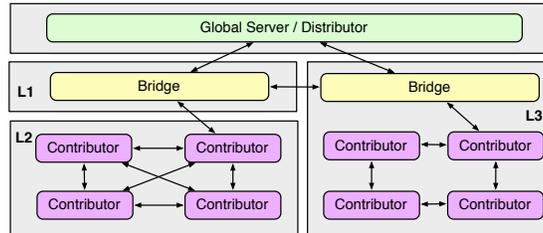
**Global Server:** ERS deployments can feature any number of bridges and contributors. In addition, some use-cases may require the presence of global servers that contain a copy of all the data going through the bridges. A global server provides a single entry point to the registry content. It exposes the contents of an ERS to other systems, for instance to the Web of Data.

## 2.2 Implementation

URNs of the form `urn:ers:<path>:<identifier>` are used to uniquely identify entities and contributors within an ERS. Individual contributions, in the form of triples, are stored in CouchDB instances run by the contributors. CouchDB's synchronisation system is used to propagate these contributions in the network by replicating them with other contributors or bridges. In addition, a search feature enables running federated queries over a set of CouchDB instances. The system source code is available at <https://github.com/ers-devs/ers> under an open licence.

## 3 Demonstration scenario

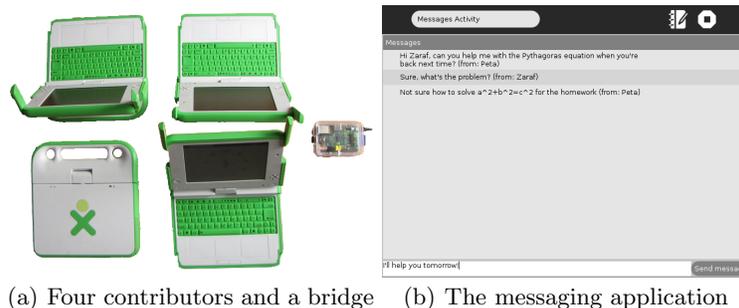
Figure 1 shows the sample deployment we created for this demonstration featuring three different physical locations, eight contributors, two bridges, and a global server. In our setup, we create one physical class-room scenario with multiple semi-connected devices consisting of multiple OLPC XO laptops, a class-room bridge server on a RaspberryPi and a dedicated global server that is connected via Internet from Fribourg, Switzerland.



**Fig. 1.** An example ERS deployment across three different locations

The contributors (XO laptops) are creating, consuming and storing structured data about entities. One bridge is used to ensure information flow and data distribution between the nodes, even if there is no reliable direct connection between two contributors. The global server is used to expose the entities within ERS as de-referencable HTTP URIs.

In our sample application, we support asynchronously discussion among school pupils. ERS is used to edit the content of a global Q&A database. In contrast to common approaches, the messages are stored and served from the laptop



**Fig. 2.** Demo setup 2(a) and messaging application 2(b)

of their publishers directly. These questions and answers are stored as entities described and interlinked using common vocabularies (SIOC, RDF, etc.). To post a message, the software creates a new entity and puts the text, the name of the creator and a visibility status (public/private) as part of the description of the said entity. When appropriate, these triples get then automatically replicated to other devices, eventually transiting through a bridge. Links between messages are established by referring to the identifiers of the entities when adding new messages, thereby creating conversation threads.

A video has been recorded to show the asynchronous dispatch of messages between XO devices. This scenario involves two XOs from the 2007 generation and a RaspberryPi model B used as a bridge. The video can be seen on Vimeo at <https://vimeo.com/70883238>.

## Acknowledgment

This work was supported by the Verisign 2012 *Internet Infrastructure Grant* program.

## References

1. The World Wide Semantic Web community. <http://worldwidesemanticweb.org/>, visited Aug 20, 2013.
2. Marat Charlaganov, Philippe Cudré-Mauroux, Cristian Dinu, Christophe Guéret, Martin Grund, and Teodor Macicas. The Entity Registry System: Implementing 5-Star Linked Data Without the Web. *arXiv preprint*, August 2013.
3. Christophe Guéret, Stefan Schlobach, Victor De Boer, Anna Bon, and Hans Akkermans. Is data sharing the privilege of a few ? Bringing Linked Data to those without the Web. In *Proceedings of ISWC2011 - "Outrageous ideas" track, Best paper award*, pages 1–4. Best paper award, 2011.