Linked Data Principles for Services and Streams

Andreas Harth
Karlsruhe Institute of Technology

June 10, 2011

Introduction A recent trend in data management involves real-time access to dynamic data sources, driven by the increasing deployment of sensors. In addition, more and more web sites expose data and computation via services, often in form of web application programming interfaces (APIs). The current Semantic Web technology stack does not explicitly provide support for dynamic data (Figure 1[1]). To enable interlinking and integration of these dynamic sources, we advocate a lightweight layered approach for publishing and consuming dynamic data on the web.

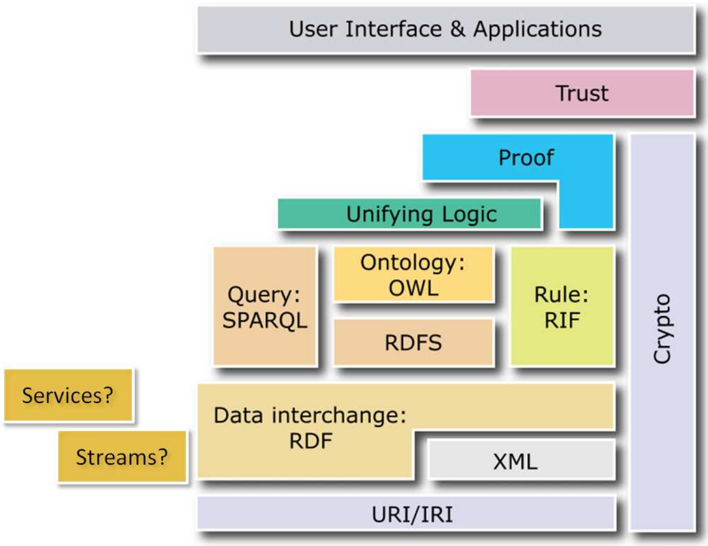

Figure 1: Services and stream sources are missing in the Semantic Web stack.

Given the success of Linked Data, which comprises a uniform access mechanism (Hypertext Transfer Protocol, HTTP) and data representation (Resource Description Format, RDF) to mostly static data, we argue that these principles should be adapted to also work on dynamic data sources, to facilitate access, interlinking and integration.

Central to the idea of Linked Data – and indeed the web – is the notion of layering: a set of basic principles provide a solid foundation for elaborate functionality built on top. The layering enables a flexible architecture and avoids large monolithic systems. The architecture of the web relaxed some of the characteristics of traditional Hypertext architectures, for example, the need for back-links or a central catalogue ([2]). Focussing on core functionality allowed the web to grow at a rapid pace. At the same time, the web created the link structure used by the likes of AltaVista, Yahoo, Ask, Google and Bing to build centralised catalogues.

[1] adapted from `http://www.w3.org/2007/03/layerCake.png`

Requirements An architecture for managing dynamic data sources on the web should consist of simple technologies to provide a low barrier to entry, and should provide means for decentralised publishing and interlinkage. Given the flexibility of existing web standards, a slight adaptation of currently deployed standards and best practices enables the publication and consumption of streams and services on the web.

Data Streams Stream data appears in a wide variety of application areas: news streams (e.g., from newspapers or blogs), social activity streams (e.g., from microblogging sites or social networking sites), sensor streams (e.g., in the Smart Grid). There is currently a Babylonian confusion of tongues when it comes to accessing stream data. News streams are often published using RSS in various flavours. However, given the request/response model of HTTP, clients have to frequently poll the RSS feed for updates. A similar architecture is employed by sites such as Facebook which offer social activity streams. Other sites, such as Twitter, have started to provide streaming APIs, which serve data in Java Simple Object Notation (JSON) format over HTTP. In the area of sensor networks, monolithic systems such as GSN [1] provide access to a wide variety of sensors, however, do not fit well into the layered web architecture as there is little support for publication of sensor streams for external use.

Existing Semantic Web technologies already offer support for streaming data. HTTP supports streaming data. RDF as universal data format could be used to encode streaming data. Recently proposed compression methods for RDF can reduce the amount of data transmitted over the wire [3]. We are working on a note documenting best practices for publishing streaming data in accordance to Linked Data principles [4]. Elaborate descriptions of sensors [6] could be used to describe streams identified via URIs.

Services Large amounts of data will not be published as a fully materialised knowledge base. Reasons include i) constant change of data (e.g., stock quotes), ii) data generated on arbitrary input (e.g., route between two geographical points), iii) data provider requires limited access (e.g., price of flight tickets). Services offer means to control access to those data sources. The most widely deployed dynamic services on the web are APIs offered by popular web sites such as Twitter and Facebook. Such APIs do often not only provide access to data but also allow for operations with side effects, such as posting a new message. Both data and operations should be made accessible in a uniform manner to facilitate integration of data and functionality.

In contrast to [7] which consists of a monolithic service environment managing publishing, registration and invocation, we argue for a layered architecture for services specifying access protocols and languages which are inter-operable with current web infrastructure, independent of implementation. We are in the process of specifying a minimal foundational layer [8], [5] upon which more complex functionality can be built upon. A layered service model with direct exposition of basic functionality further enables the adaptation of work done by the database community, e.g. on schema mapping, to web services.

Conclusion We have argued for the need of providing uniform means for accessing data streams and services on the web, and have identified Linked Data principles as a sensible starting point for doing so. The best practices and foundational standards for publishing dynamic sources should be minimal and allow for the layering of more complex functionality on top. Such an ecosystem could involve distributed complex event processing engines in the case of data streams, and service registries or service composition in the case of services. We provide a non-partisan platform consisting of a wiki [2] and a mailing list to serve as consolidation point for the community.

[2] `http://linkedservices.org/`

References

[1] K. Aberer, M. Hauswirth, and A. Salehi. A middleware for fast and flexible sensor network deployment. In *VLDB*, pages 1199–1202, 2006.

[2] D. W. Connolly. An Evaluation of the World Wide Web with respect to Engelbart's Requirements, August 2002. W3C Note, `http://www.w3.org/Architecture/NOTE-ioh-arch`.

[3] J. D. Fernandez, M. A. Martinez-Prieto, C. Gutierrez, and A. Polleres. Binary RDF Representation for Publication and Exchange (HDT), March 2011. W3C Member Submission, `http://www.w3.org/Submission/2011/SUBM-HDT-20110330`.

[4] A. Harth and R. Stühmer. Publishing Event Streams as Linked Data , June 2011. `http://km.aifb.kit.edu/sites/lodstream/`.

[5] R. Krummenacher, B. Norton, and A. Marte. Towards linked open services and processes. In *Proceedings of the Third future internet conference on Future internet*, FIS'10, pages 68–77, 2010.

[6] L. Lefort, C. Henson, and K. Taylor. Semantic Sensor Network Incubator Report, 2011. `http://www.w3.org/2005/Incubator/ssn/wiki/Incubator_Report`.

[7] C. Pedrinaci and J. Domingue. Toward the next wave of services: Linked services for the web of data. *Journal of Universal Computer Science*, 16(13):1694–1719, 2010.

[8] S. Speiser and A. Harth. Integrating Linked Data and Services with Linked Data Services. In *ESWC*, pages 170–184, 2011.