



Supporting Sustainable Publishing and Consuming of Live Linked Time Series Streams

Julian Andres Rojas Melendez^(✉), Gayane Sedrakyan, Pieter Colpaert,
Miel Vander Sande, and Ruben Verborgh

IDLab, Department of Electronics and Information Systems,
Ghent University - IMEC, Ghent, Belgium
`julianandres.rojasmelendez@ugent.be`

Abstract. The road to publishing public streaming data on the Web is paved with trade-offs that determine its viability. The cost of unrestricted query answering on top of data streams, may not be affordable for all data publishers. Therefore, public streams need to be funded in a sustainable fashion to remain online. In this paper we present an overview of possible query answering features for live time series in the form of multidimensional interfaces. For example, from a live parking availability data stream, pre-calculated time constrained statistical indicators or geographically classified data can be provided to clients on demand. Furthermore, we demonstrate the initial developments of a Linked Time Series server that supports such features through an extensible modular architecture. Benchmarking the costs associated to each of these features allows to weigh the trade-offs inherent to publishing live time series and establishes the foundations to create a decentralized and sustainable ecosystem for live data streams on the Web.

Keywords: Semantic web · Open linked data
Linked Data Fragments · Time series · Data streams

1 Introduction

The development of Internet of Things technologies has fostered the creation of live data streams in multiple domains. Specifically in the public domain, examples of such data streams can be found as sensor observations about air quality, noise level, street occupancy, vacant parking spaces, temperature, river water level, wind speed, state of public lighting systems, traffic lights, among others. Furthermore, in Europe thanks to the European Public Sector Information directive¹, public authorities are required to publish such data in an open fashion on

¹ <https://ec.europa.eu/digital-single-market/en/european-legislation-reuse-public-sector-information>.

the Web. This raises new challenges for data publishers as they cannot anticipate the amount of users or type of queries required on the Web, and might not be able to afford expensive infrastructures, required to maintain availability and scalability.

Studying the trade-offs introduced by Linked Data Fragments [5] to publish data streams on the Web, helps to understand possible ways to reduce server costs by transferring query answering related tasks to clients. This requires clients to implement the logic to answer a given query, increasing their complexity and the time required to process the data on the client side. Anticipating this, data publishers may provide multidimensional interfaces [3] containing pre-processed data, relevant for answering common queries and offer them as a service that could benefit clients by reducing query response times and implementation complexity, without limiting query flexibility. The type of interfaces to offer depend directly on the type of data and the related use-cases. Such an approach may help to create revenue sources for data publishers while contributing to the sustainability of the Open Data streams on the Web.

In this demo paper we present an overview of possible multidimensional interfaces, containing query answering features, that may be implemented and offered as a service on different Open Stream data use cases. Furthermore, we introduce the initial developments of a Live Time Series server that support the creation of such interfaces through an extensible modular architecture.

2 Related Work

RDF Stream Processing (RSP) [1] defines a framework for continuous query answering over data streams. RSP engines can take into account one or more RDF streams to answer queries which results will be computed at several time instants to consider new available data on the streams. Triple Pattern Fragments Query Streamer (TPF-QS) [4] was introduced as an alternative to server-side RSP engines. with the goal of making RDF stream server-side publishing possible at a low cost, with a client-side RSP engine. In this approach, several time-annotation techniques were investigated, of which annotation using named graphs caused the least overhead. The results however showed that this approach has scalability issues when querying historic data.

The work presented in [2] raises the fundamental question of the sustainability of the Web of data and introduces a marketplace for federated query answering, giving clients the option to decide from which sources do they want to retrieve the data needed to answer a certain query and who will process the data to obtain that answer. The cost of the answer(s) of a given query can be derived from the cost of hosting the related data. However determining what is the cost of computing such answer(s) is still an open issue. In this direction, benchmarking mechanisms could be used to determine it in terms of computational costs. For instance, the solution proposed in the HOBBIT project which provides a generic platform for benchmarking question/answering processes centered around the challenges of data heterogeneity and scalability could be used to determine the costs associated to answering a certain query.

3 Multidimensional Interfaces

Multidimensional Interfaces [3] were introduced for generically fragmenting data with a specific order and publishing these fragments in an interface-level index. These interfaces can make multidimensional ordinal data automatically discoverable and consumable by clients using hypermedia controls. The goal of these interfaces is to raise the server expressivity while maintaining low server costs. A vocabulary² to formally describe multidimensional interfaces was introduced. It defines the concepts of *Range Fragments* and *Range Gates*. A *Range Fragment* is a Linked Data Fragment that specifies an ordinal interval for a predefined fragmentation strategy. A *Range Gate* is a Linked Data interface which exposes a set of *Range Fragments*. Using these concepts it is possible to define different fragmentation strategies that can be exposed as multidimensional interfaces.

In a general sense, for live time series originated from sensor observations it is possible to define ranges as follows:

Time Ranges. Time constrained intervals can be used to create *Range Fragments* or summaries that compute statistical variables. For example to expose average values of measurements at hour, day, week, month and year level.

Geospatial Ranges. Sensors locations can be used to create Range Fragments that comprises predefined geographical areas. For example, street occupation can be given on a neighborhood, city or country level.

Depending on the type of data and the specific use case other type of fragmentations and even combinations of them can be further defined.

4 Live Time Series Server

The Live Time Series Server is an ongoing implementation that aims on providing a cost efficient interface for Open Stream data publishing. Through an extensible modular architecture we allow data publishers to define multidimensional interfaces to provide query answering functionalities on top of their data. The code is available in a Github repository³, along with the instructions of how to test it.

As shown in Fig. 1, the server is composed by three main modules:

Data Event Manager. This module receives RDF stream updates and fires an event to notify the availability of new data.

Communications Manager. Handle the communication between the Multidimensional Interfaces and the clients. It can expose the data as Range Fragments, created by each interface through HTTP endpoints or by Websocket channels for publish/subscribe communication.

² <http://semweb.datasciencelab.be/ns/multidimensional-interface/RangeGate>.

³ <https://github.com/linkeditimeseries/timeseries-server>.

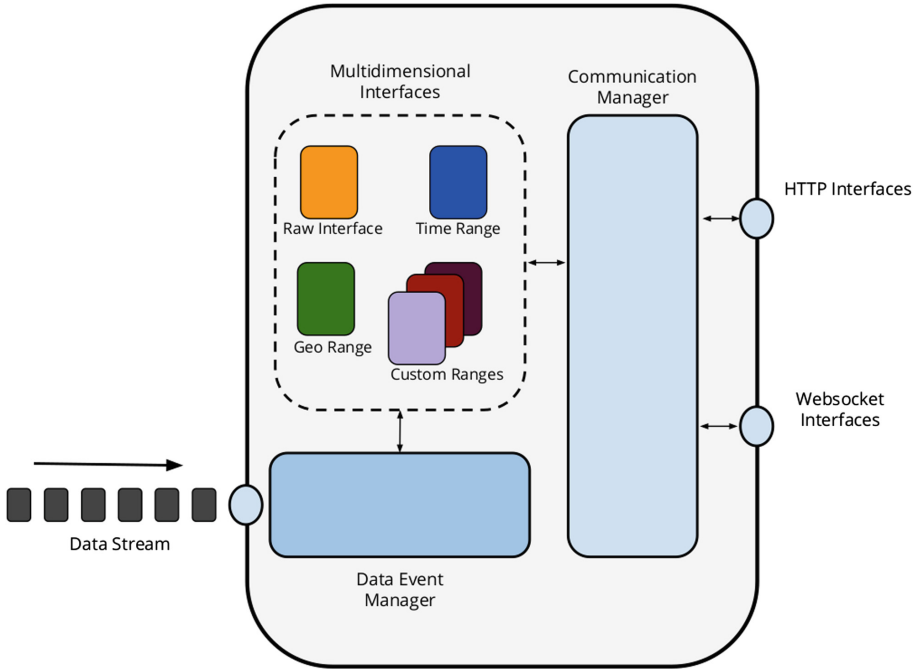


Fig. 1. Modular architecture of the Live Time Series server.

Multidimensional Interfaces. The interfaces expose the data stream according to its predefined logic. Each interface subscribes to a data event with the Data Event Manager and performs a new calculation with each update with the exception of the Raw Interface, which exposes the data as it is received. The data can be exposed as *Range Fragments* through HTTP or pushed to subscribed clients through Websocket channels.

5 Conclusion and Future Work

We introduced Live Time Series server which provides a Linked Data fragments based interface for publishing live time series on the Web. By integrating the concept of multidimensional interfaces data publishers can define modules that perform predefined calculations over the data that suit a given use case. This increases the expressivity of the server while keeping its costs low. It also reduces clients implementation complexity and data processing time to obtain a query answer. Allowing features to be turned off and on on the time series server, helps data owners to define what features they want to support and make the trade-off with their budget. Data reusers may still implement some of these features as a third party, yet then a revenue model should be thought of [2].

In future work we plan to extend this approach by defining a mechanism that allows to calculate computational cost of multidimensional interfaces through

benchmarking processes, in order to help determining their economical cost. Integrating mapping capabilities in order to work with non RDF data streams constitutes yet another future work line.

Acknowledgements. This work has been supported by HOBBIT H2020 project (GA no. 688227) and by the Smart Flanders Programme (<https://smart.flanders.be>).

References

1. Dell’Aglio, D., Della Valle, E., van Harmelen, F.: Stream reasoning: a survey and outlook. **1–2**, 59–83 (2017). <https://doi.org/10.3233/DS-170006>
2. Grubenmann, T., Dell’Aglio, D., Bernstein, A., Moor, D., Seuken, S.: Decentralizing the semantic web: who will pay to realize it? In: ISWC2017 Workshop on Decentralizing the Semantic Web, October 2017
3. Taelman, R., Colpaert, P., Verborgh, R., Mannens, E.: Multidimensional interfaces for selecting data within ordinal ranges. In: Proceedings of the 7th International Workshop on Consuming Linked Data, October 2016
4. Taelman, R., Verborgh, R., Colpaert, P., Mannens, E.: Continuous client-side query evaluation over dynamic linked data. In: Sack, H., Rizzo, G., Steinmetz, N., Mladenić, D., Auer, S., Lange, C. (eds.) ESWC 2016. LNCS, vol. 9989, pp. 273–289. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47602-5_44
5. Verborgh, R.: Triple pattern fragments: a low-cost knowledge graph interface for the web. *J. Web Semant.* **37–38**, 184–206 (2016). <https://doi.org/10.1016/j.websem.2016.03.003>