Building a Real-Time Web Observatory

Ramine Tinati, Xin Wang, Thanassis Tiropanis, Wendy Hall

***Abstract*—Real-time data streams are increasingly becoming primary ways to access data from the Web and Internet-ready devices. Real-time streams, personal devices, and sensor networks, have the potential to contain rich insights for researchers, commerce, and governments. With a vested interest in unlocking the potential benefits hidden within, there has been extensive work conducted on developing technologies to process, integrate and extract value from the data. However, exposing the value in this data is acheived via sharing the data in a secure and controllable environment. To that end, we present the Web Observatory, a Web platform with an architecture capable of harvesting, querying, and analysing multiple real-time and historic heterogeneous data, whilst providing data owners access control to their resources. We consider the current landscape and challenges of using data, analytics and visualisation, and describe a series of use cases where the Web Observatory can be used.**

Keywords*—*Web Observatory, Real-time Processing, Access Control, Data Querying

I. Introduction

We are fast becoming part of a world of digital interconnectivity, where devices such as smartphones, watches, fitness trackers, and household goods are part of a growing network, capable of sharing data and information. Increasingly, the Web has become the ubiquitous interface to access this network of devices. From sensors, to mobile applications, to fitness devices, these devices are transmitting their data to - often - centralised pools of data, which then become available via Web services. The sheer scale of this data leads to a rich set of high-volume, real-time streams of human activity, which are often is made publicly consumable (potentially at a cost) via some API.

These streams represent a global network of human and machine communication, interaction, and transaction, and with the right analytical methods, may contain valuable research and commercial insights. In domains such as health and fitness devices, the aggregation of these sources are supporting the transition towards the quantified-self, and offers rich insight into the health and well-being of individuals, with the potential of diagnosing or decreasing disease. For academia, the combination of these sources are providing social scientists and digital ethnographers a far richer understanding of society, and how we as individuals operate.

In order to handle these new forms of big, and small data, significant effort has gone into developing technologies capable of storing, querying, and analysing high-volume datasets - or streams - in a timely fashion, returning useful insights of social activity and behavior. However, herein lies a challenge, and a great opportunity. We are now in a position where the technologies used within the *big data processing pipeline* are maturing, as are the methods we use to analyse data to provide valuable insights. Yet, overshadowing these benefits are issues of data access, control, and ownership. Whilst the data being produced continue to grow, their availability beyond the walled-gardens of the data holder - whether commercial or institutional - reduce the full potential of analysis envisaged in the big data era.

Addressing the challenge described above, we introduce the Web Observatory, a globally distributed infrastructure that enables users to share data with each other, whilst retaining control over who can view, access, query, and download their data. At its core, a Web Observatory comprises of a list of architectural principles which describe a scalable solution to enable controlled access to heterogeneous forms of historical and real-time data, visualisations, and analytics.

The remainder of this paper will describe an instance of the Web Observatory, the Southampton Web Observatory (SUWO). We draw upon three real-world scenarios where the SUWO has been used, including its application for facilitating Web Science research, supporting digital government policy, and as a platform to control, share, and aggregate an individual's personal data. Finally, we conclude with considering the future growth of Web Observatories.

II. Related Work

There has been substantial work relating to the development of various components that contribute to the aspects of a Web Observatory, from early work on distributed databases [1], role-based access control mechanisms [2], data integration techniques [3], [4], distributed and federated query approaches [5], [6], to recent work looking at building complex event processing platforms [7], and open-stack real-time stream processing technologies [8]. This work has contributed greatly to challenges faced with building a Web Observatory, however, we see the challenge of building successful Web Observatories is not about the individual technologies, but how they are orchestrated and used in the processing pipeline.

Existing Web Observatories development can be broadly grouped as systems conceived and developed specifically as Web Observatories [9], and those converging on Web Observatory status from other starting positions (such as analytics platforms or data repositories) [10], [11]. Whilst the term Web Observatory is not explicitly used, we consider them a Web Observatory in terms of their functionality and capabilities. In general, existing academic and research driven Web Observatories include systems such as the NeXT social media Observatory platform [10], the CosMoS social media analysis platform [11], EventShop, a complex event-processing system [12], and the Archive Hub, a platform the accessing and analysing Web archives [13]. These systems share the desire to provide a platform where users can perform a variety of tasks relating to gathering, storing, and analysing heterogeneous forms of Web data. They work with existing technologies, and provide users with UIs capable visualising data in accessible forms, allowing for a wider group of non-technical or untrained individuals to make sense of data.

We also note that there are enterprise-level systems and platforms that offer similar characteristics and functionalities with a Web Observatory which tackle the challenge of developing high-volume, message queuing and passing systems [14]. However, systems such as those developed by TIBCO are more relevant to dealing with specific forms of data in sectors such as finance or heavy industries, and unlike a Web Observatory, they are closed-source, proprietary solutions, without the underlying principles of sharing and openness.

III. Web Observatory Definition

A Web Observatory is a distributed infrastructure that provides users the ability to share resources with each other, whilst retaining control over who can view, access, query, and download such resources [15]. One of the primary functions of the Web Observatory is to provide data cataloguing capabilities with security as an inherent feature of its design. *Data discovery* is also an essential component of a Web Observatory, which is why embedded within the cataloguing mechanism is the use of a common metadata schema which provides a rich resource containing the type and contents of the data being shared.

In addition to the core capabilities of data discovery and sharing, a Web Observatory is able to query data of heterogenous types, stored in various formats and repositories. Web Observatory users can integrate their analysis and visualisation of a dataset for other users to view, reuse, and modify. The metadata schema used to describe the data also extends across these functionalities, providing vocabulary to describe the relationship between a visualisation and the analysis tool, dataset, and users that it is built upon.

A. Architectural Principles

A Web Observatory will vary in design, architecture, UI, and configurability, based on the context of its use. However, there are four fundamental principles which need to be considered when building a Web Observatory [15], [16], which in application allows a Web Observatory to become a wrapper for many different scenarios and use cases. Below is a summarised account of these is listed below:

- *Not all datasets or applications can be public.* Access to some datasets needs to be restricted for licensing, privacy or other reasons. The Web Observatory allows its users to list or host datasets that are public or private. Access to private datasets is managed by the user who hosts them on the Web Observatory. Since access to datasets can be restricted, access to applications that make use of those datasets needs to be restricted as well.
- *Web Observatories list two main types of resources: datasets and analytic application (including visualisations).* The link between a listed analytical application and the datasets that it uses must always be made explicit, even if the used datasets are listed as private, with restricted access.
- *Not all listed resources need to be locally hosted.* Listed datasets or analytic applications can be hosted in remote servers managed by third parties.
- *Metadata describing the listed resources and projects are published.* This way, descriptions of resources can be harvested and listed in other Web Observatories or Web-based resources.

B. A Network of Web Observatories

The true potential of the Web Observatory can be found when the resources (datasets and analytic applications) are discoverable within a *network of Web Observatories*. Analogous to the Web, connecting multiple Web Observatories increases the richness and variety of searchable datasets, analytics and visualisations. Our approach to creating a network of Web Observatories borrows discovery ideas from existing Web services such as WSDL and UDDL, where each observatory emits a heartbeat to a named Web Observatory node using a common protocol, which then acknowledges the existence of the Observatory and crawls the available datasets based on their access control settings. Using this architectural configuration allows the network to grow using an arbitrary number of listening nodes which themselves emit a heartbeat to other listeners. Essentially this forms a peer-to-peer, decentralised configuration. Facilitating the discoverable of Web Observatory data and visualisations is the use of metadata represented in an interoperable schema[1] as described above.

Independent to the implementation of the platform, the core of a Web Observatory will contain embedded micro-data which describes the available datasets, analytics, and visualisations. The *Web Observatory* micro-data provides the underlying mechanism for crawlers to extract the available (and open) resources and enables Web Observatory users to be able to search for datasets and visualisations.

IV. The Southampton Web Observatory

The Southampton Web Observatory (SUWO) builds on the core Web Observatory principles of providing controllable access to heterogeneous data sources and visualisations. As Figure 1 illustrates, SUWO's distinguishes itself from other Web Observatories by taking a decoupled approach to the data storage, analytics and visualisations, and the Web portal. Each component communicates with each other using the SUWO Web Observatory API, and the portal provides an end-point for users to search and query datasets, and view visualisations associated with the datasets listed. The SUWO platform also enforces that listed visualisations must be associated with the datasets that they have been build with, even if the dataset is not publicly accessible. The SUWO Portal also handles the embedded metadata used to describe the listed datasets and visualisations in order to make them discoverable to a wider network of Observatories.

The SUWO has been designed as a decoupled solution in order to offer reconfigurablity and scalability. The separation

[1] Web Observatory is an extension of the Schema.org vocabulary: http://schema.org/Dataset/WebObservatoryDataset

of the data storage, analytics and visualisations, enables the platform to be configured and operated in a distributed fashion, where co-ordination is handled by either the Web Observatory API, or external middleware (such as message passing protocols).

A. Data Types and Formats

Data can be contained in various types of stores from SQL, NoSQL, and triple/RDF stores, to non-structured formats such as CSV. Resources can be queried using the Web Observatory API, which uses a JSON structured query language, and the mappings to the various types of datastores is handled by the Web Observatory API, and processed server-side. Unlike a distributed database architecture which can be considered as a single logical database, we wish to provide a middle layer which provide a NoSQL like query syntax that uses the Web Observatory API to query multiple datastores solutions. Moreover, this approach enables us to query both historic datasets and real-time streams using the same NoSQL-like syntax, which in practise allows Web Observatory users to build applications using a mix of historical and real-time data.

C. SUWO Real-Time Data

Many Web services provide programmatic access to platforms via APIs. Depending on the Web service, using the API can provide the complete collection of activity (the *firehose*) in real-time. Many of the social media platforms such as Twitter, Facebook, and Weibo, offer full-to-limited access to their social data, which typically contains information regarding their members' communications, interactions and activity. As shown in Figure 2, the first stage of the real-time processing pipeline is the connection to external APIs, followed by the *Pre-Processing Stage* of enrichment and unification of various real-time streams from Web services. This process is achieved by individual Web harvesters connected, which are jointly controlled by a federated processing component. Each harvester works independently to establish an external data stream, which is transformed into the SUWO real-time stream JSON form. As part of the unification processes, we perform lightweight enrichment of the data order to ensure consistency across streams. This enrichment process, performed in real-time, ensures that each record has at a minimum, a timestamp (ISO8068), source identifier (i.e. 'wikipedia_revisions') and unique record identifier.

After the initial pre-processing stage, the enriched and restructured data streams are then processed in the *Streaming Stage*, which uses the Advanced Message Queueing Protocol (AMQP) [17] to restream the incoming data sources. In this configuration, AMQP acts as the message-passing middleware for our publish-and-subscribe approach, capable of providing a scalable solution for maintaining high-volume message processing and passing [18].

Many of the Web sources being harvested produce high-volume feeds which which is resource intensive when using hold messages holding queues - in-memory or on disk - until clients connect and pop them off the queue. Additionally, as we are serving messages to multiple clients, queues become problematic as we want all clients to have the same access to the data. An alternative approach, which is far more suited to handling multiple, high volume data streams, is to uses Exchanges, which represents a publish-and-subscribe service. This enables multiple clients to connect to a 'temporary' queue and retrieve the incoming messages. The advantage of this method is that it requires significantly fewer resources than a queueing approach, and clients are able to receive the *latest* data as soon as they connect. However, the disadvantage of this approach is that clients are offered no 'cache' when connecting to the exchange, thus if there is no incoming messages, the stream may appear to be inactive. For each pre-processed stream we instantiate a single exchange, and for combined streams, we construct a consume-and-emit pipeline, by where we connect to multiple single exchanges, perform additional

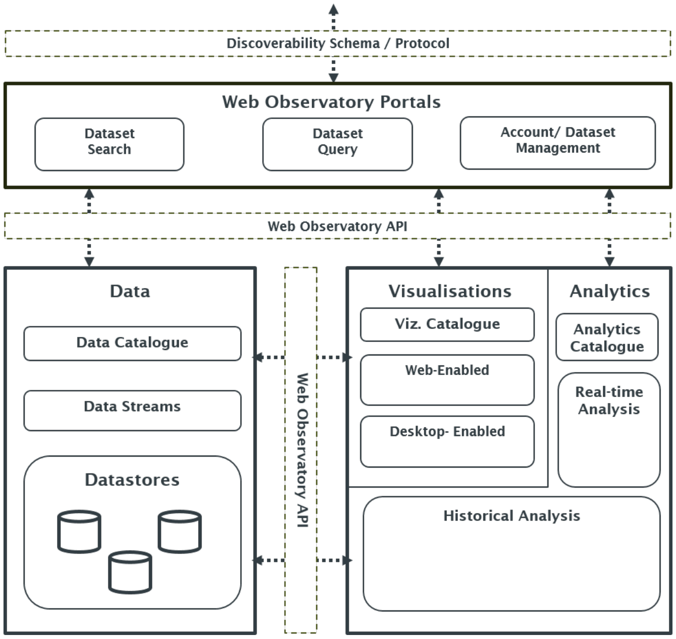

Fig. 1. SUWO decoupled architecture. The three major compoents of SUWO: Data, Visualisations, and the Web Portal. The Web Observatory federates access between them. Note: the datastores do not represent a distributed database, but a set of independant databases and filestores accessible via the Web Observatory API and Web Portal.

B. Supplier-to-User Metamodel Transformation

In many cases, the ingested real-time and static data do not share common schemas, thus requires significant effort to make sure that the data provided to the users is represented with a homogeneous schema, with consistent fields and identifiers. This is particularly important for data integration, where matching between sources requires consistent closely aligned data. In order to achieve this we use a mix of manual hand-coded mapping-annotations and automated discovery processes in order to automatically detect the SUWO resource metamodel.

processing, and then publish this on a new exchange.

The final stage to the real-time stream processing workflow involves two components, internal consumption for archiving purposes, and a AMQP HTTP middleware in order to make the exchanges available via the Web Observatory portal. Although discussed in more detail in Section IV-D, the middleware used to connect to the AMQP exchanges is a lightweight service that does not process data, but provides the necessary handshaking and security layer between the client, the API and the exchange.

1) Challenges: There are various challenges associated with processing real-time streams of heterogeneous, unstructured data, with different data schemas. One of the core challenges faced is processing high volume streams in a timely fashion, which may include restructuring, enriching the data streams to maintain a consistent stream. Many of the enrichment subprocesses required calls to external services (e.g. geographic location enrichment), which potentially slows down processing time. A simple approach to this is to employ an internal cache of recently looked-up resources is used to ensure that processing of streams is performed in a timely manner, however, this has resource limitations, thus smart caching strategies are required.

Possibly the most challenging topics for a Web Observatory involves stream integration. There has been extensive related work in complex event processing [7], sensor data integration [19], and personal information management [20]. In the SUWO, we take a naive approach and perform stream integration by matching common fields or 'pattern' between streams. In many cases, this matching is based on a seed list of topics, keywords or, by matching records with matching timestamp, or within a given time window. We also developing dynamic methods of integrating streams which rely on both user input, a simple machine learning techniques in order to dynamically change the pattern used to match and integrate streams. Take for instance a collection of social data streams, we monitor the overall message rate of a given set of streams (i.e. posts per minute (PPM)), which we learn as a baseline value as the 'normal' value of activity. If a stream's PPM changes for a given period of time, these fluctuations and bursts notify the stream-integration middleware to begin integrating streams based on a pre-defined set of fields (i.e. common identifier or keyword), and instantiate a new stream with the result.

D. Public Access - Web Observatory API

In order to provide end-user access to historic and real-time streams, the SUWO offers an API capable of acting as a secure middle layer between the dataset locations, and the end-user connections, whether this be via direct access on the Web Observatory portal, or via programmatic use via the development of 3rd party applications and visualisations. Resource access relies on two pieces of information: the *access URL* and its *media type*[2]. The access URL gives the location where the resource is available; the media type indicates the protocol and procedure for accessing the resource. The media type should be standard if possible, or a definition of the media type should be provided.

For real-time streams that use a messaging protocol such as AMQP, we use a custom media type "application/amqp"[3]. Applications built on multiple streams can select resources of the type "application/amqp" and filter relevant ones based on the resource descriptions and keywords. It is also possible to combine static datasets and live streams in the same way. Once a stream is connected to, we establish a Web Socket to enable a client, which may be a 3rd party application or a open socket via the SUWO portal, to receive messages from the stream as they arrive at the message exchange.

1) Accessing Embedded Metadata: The SUWO maintains metadata for all resources, whcih are internally represented using the Data Catalog Vocabulary (DCAT) [21]. The SUWO API is built by mapping DCAT documents to a REST API in a way that preserves the semantics of DCAT. The SUWO API follows the *Hypermedia as the Engine of Application State* (HATEOAS) constraint of REST[4], that enables applications to explore the whole API from a single entry without referring to external documentations. We use the DCAT-to-REST approach as they provide advantages such as offering *rich mapping semantics*, *interoperability and automation* for resource discovery, *reversible mappings* between the API and DCAT, and *simple document federation* when combining multiple resources.

2) Protecting resources using OAuth 2.0: Not all resources are open to the public. The SUWO API adopts OAuth 2.0 to provide comprehensive yet flexible protection for both public and private resources. As shown in Figure 3, the SUWO acts as a reverse proxy of registered resources. That is, all requests for access are controlled by the SUWO before accessing the resources. Using OAuth users can authorise applications (either their own or third-party) to act on behalf of the users, and the authorised applications gain the same permission as the users. To access resources, authorised applications firstly authenticate themselves against OAuth 2.0, and the SUWO verifies whether the applications - the users who are viewing the application - are allowed to access the resources. As the resources are never locally held, the owner has full access control at the SUWO layer and at the datastore.

V. Web Observatory Use Cases

In this section we consider a selection of uses cases where a Web Observatory is suitable for supporting the access and sharing of data and analytics. We base these scenarios on the SUWO architecture which is broad-based to enable it to be re-purposed and be used as a wrapper for different contexts.

A. Web Observatory for Web Science

An initial use case of the Southampton Web Observatory was inspired by the pressing need from Web Science researchers to access various sources of Web data, including

[2] A comprehensive list of media types is available at http://www.iana.org/assignments/media-types/media-types.xhtml

[3] There is a media type for streams, "application/octet-stream". However in our case the media type has to be specific enough to enable applications to automatically access the resource.

[4] Strictly speaking an API is not RESTful without HATEOAS

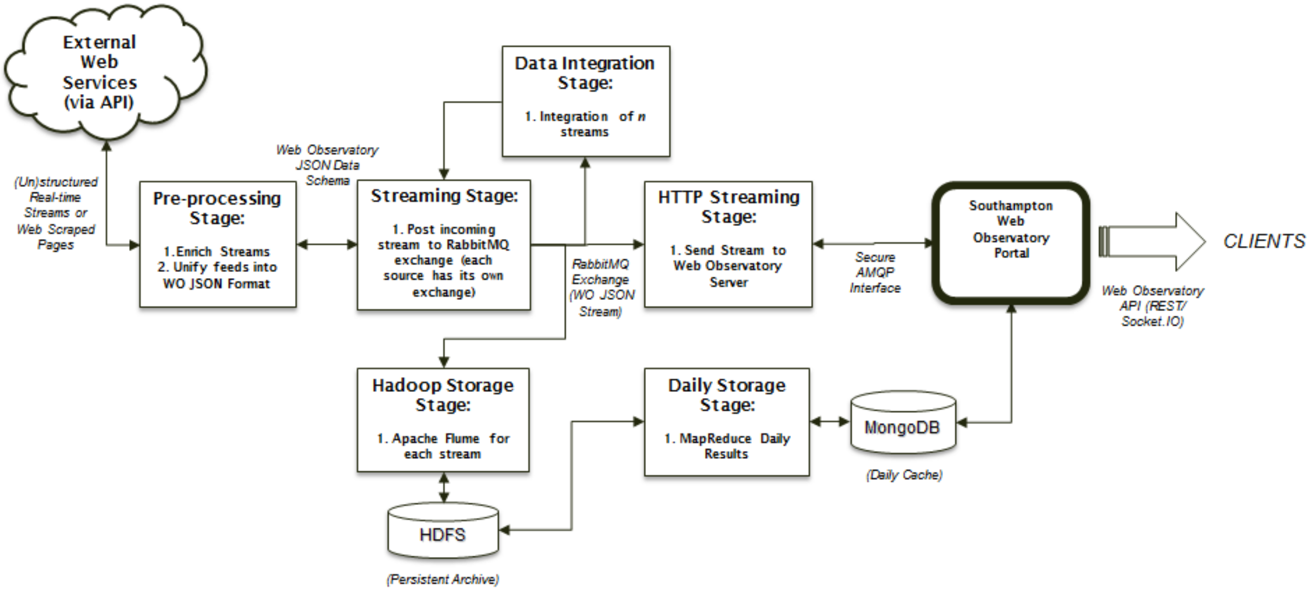
External Web Services (via API)
(Un)structured Real-time Streams or Web Scraped Pages
Pre-processing Stage:
1. Enrich Streams
2. Unify feeds into WO JSON Format
Web Observatory JSON Data Schema
Streaming Stage:
1. Post incoming stream to RabbitMQ exchange (each source has its own exchange)
Data Integration Stage:
1. Integration of n streams
RabbitMQ Exchange (WO JSON Stream)
HTTP Streaming Stage:
1. Send Stream to Web Observatory Server
Secure AMQP Interface
Southampton Web Observatory Portal
CLIENTS
Web Observatory API (REST/ Socket.IO)
Hadoop Storage Stage:
1. Apache Flume for each stream
HDFS
(Persistent Archive)
Daily Storage Stage:
1. MapReduce Daily Results
MongoDB
(Daily Cache)

Fig. 2. SUWO Real-Time processing architecture This illustrates a top-level configuration of the processing pipleine from supperlier-to-user for real-time streams; the intermediary proceeses and middleware are not shown. Static data access and query middleware are also not present, but follow a similar architecture of data integration and access.

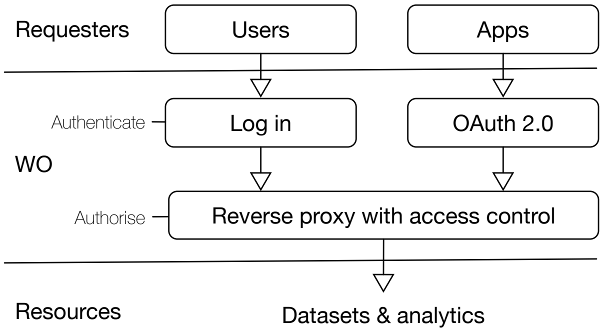

Fig. 3. Requesters are users or applications initiating requests to access resources; SUWO authenticates requesters and authorises their requests, and resources, that are datasets and analytics shared on the web observatory.

historic stores and real-time streams of Web data. Many datasets were collected and used within research projects, however, accessing and sharing them between teams, projects and institutions was costly or impractical. In light of this, SUWO was developed as a platform for users to initially list and share their data, followed by the capability of listing their visualisations and analytics with each other.

In this application context, the Web Observatory facilitates Web Science researchers interesting in studying the Web, human activity and related topics which require datasets collected from the Web. In respects to data access, the SUWO enables researchers, from different departments and institutes, to share and query a global set of resources using a simple query language via the Web Observatory API, reducing the complexity of obtaining data. In additional to being able to share and access various forms of data, SUWO enables users to share visualisations and the embedded analytics, supporting reproducablity and validation. Visualisations also provide non-technical users the ability to inspect data without data processing knowledge, enabling them to directly work with the data without domain-specific expertise.

We envision the future developments in this space will lead to improvements in methods and approaches for conducting real-time analysis about human and machine interaction. The capabilities of real-time stream processing offered by the SUWO will become an essential component to examine the digital traces of human activity at scale, in real-time. A platform such as the SUWO will simplify the process of obtaining access to real-time data, providing end-users a single point of access, where data is structured using a common schema. Furthermore, by offering a secure layer of access control via mechanisms such as the Web Observatory API, data providers can share potentially sensitive and commercially valuable data streams.

B. *Web Observatory for Digital Government Policy*

Increasingly governments are in the position where they require access to various data services which they internally, and externally produce, and often these manifest into an ecosystem of social systems [22]. In the last decade, the move towards a *digital government* has led to many of the services previously part of an offline process to become digitised and accessible via the Web. Take for instance, the UK and US government now offer thousands of datasets produced by city-sensor networks, live transport and traffic data, to demographic and crime reports. This technological-turn has led to a government that are in the position where a variety of datasets are in abundance, offering internal organisational value, reducing operational costs, increasing transparency, and improving citizens' living conditions [23]. Many of these datasets are produced in real-time, such as traffic data, or from the variety of smart-city initiatives using sensor networks.

In this context, the Web Observatory provides a wrapper to expose and share government data, and more importantly, allows for third parties to access and produce analytical and visual representations of the data, while still retaining control over who can access those data. Security is an extremely important feature, given the nature of specific types of datasets. Furthermore, by being part of a network of observatories, it offers an opportunity for governments to link to non-government datasets, enriching existing data and providing new insights not originally envisioned.

Supporting data-driven Government policy making, the SUWO has been used as a platform for joint research between the Adelaide Government and the Universiry of South Australia and the University of Southampton. Inspired by the desire to examine the growing concerns of supporting an aging city, the SUWO has been used to provide access to multiple goverment and academic datasets in order to answer questions concerning the provision of public services based on the demographic landscape of the city's neighbourhoods. The next step involves the expansion with additional government departments within Australia in order to establish a network of Observatories, which can then be used to compare similar issues across cities.

C. Web Observatory for Personal Data

The expanding collection of personal wellbeing and fitness devices, sensors and applications provides a rich context for considering a Web Observatory within the context of *personal data* and personal information management. The data agnostic architecture of the Web Observatory enables it to wrap as a *Personal Web Observatory* (PWO), capable of collecting and processing data produced by the numerous personal data devices that individuals use. This can include a mix of Web data produced from social media and networking services, to more location-aware services produced by mobile applications, to health and well-being data produced by devices such as heart rate monitors, pedometers, to more specialised devices that track signals such as blood glucose, body composition, and hormone levels.

In this context, the Web Observatory becomes a platform which can draw together the various types of personal data that individuals which to store, analyse, and potentially share with a network of additional personal Web Observatories. A PWO still conforms to same principles of sharing and control, whilst empowering the data owner with control over data access and use. Moreover, the SUWO's approach to integrating historic and real-time streams of heterogeneous data offers PWO users the ability to combine their various streams of data to gain a holistic view of their data, which has the potential to be used for improving their wellbeing, advising lifestyle changes, diagnosing possible diseases, or predicting future complications. Analytics and visualisations can be designed to take advantage of the spectrum of heterogeneous data sources, offering macro-level lens of an individual's collection of personal data sources. One can envision a *personal data dashboard*, providing PWO owners to gain an overview of their various activities on the Web, with the potential of sharing this with others quantified-self enthusiast.

We also consider how a network of PWOs in combination with other types of Web Observatories can be used to support societal challenges. In such a scenario, combining data from PWOs with a Government Observatory, restricted government datasets containing sensitive health records could be queried against personal data, in order to provide value for both the individual, and also to improve public services and deliver data-driven policy-decisions. This offers an opportunity for data providers and data owners to negotiate a relationship between who can access their data, and how the data are being used and for what purpose.

VI. Future Prospects for Web Observatories

In this article we considered the changing data landscape, and introduce the Web Observatory and its architectural principles designed to empower users and data suppliers with controllable access to data produced by human and machines, on and off the Web. This vision is driven by the desire to facilitate the *sharing* and *discoverablity* of data, analytics and visualisations, while offering data providers the capability to retain control over who can use their resources. Based on these principles, we built the Southampton Web Observatory (SUWO), a modular, scalable, and discoverable Web Observatory, suitable for managing historic and real-time streams of heterogeneous data.

Driven by open standards and technologies, SUWO includes a secure access control layer via the *Web Observatory API*, which becomes a core component to enable the sharing of data, visualisations and analytics. The generality of SUWO's architecture and use of open source technologies and standards enables it to become a wrapper for many different purposes, from facilitating Web Science research, to helping users store and process their own personal data. Moreover, using the established embedded metadata and vocabularly to describe datasets and resources, connecting a network of Web Observatories - which may be across multiple domains - offers the potential to link up a global network of resources.

The future of the Web Observatory not only relies on technicalogical development, but consideration of data ethics, governance, re-use, attribution, and licensing. As with the growth of a multi-stakeholder technology, for example, the World Wide Web, we believe that these will be part of a co-constructed process with data owners, Web Observatories developers, users, and the changing technological landscape.

References

[1] D. Bell, *Distributed Database Systems*, 1st ed., J. Grimson, Ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1992.

[2] J. S. Park, R. Sandhu, and G.-J. Ahn, "Role-based access control on the web," *ACM Transactions on Information and System Security (TISSEC)*, vol. 4, no. 1, pp. 37–71, 2001.

[3] M. Lenzerini, "Data integration: A theoretical perspective," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2002, pp. 233–246.

[4] A. Doan, A. Halevy, and Z. Ives, *Principles of data integration*. Elsevier, 2012.

[5] F. Goasdou and M.-C. Rousset, "Querying distributed data through distributed ontologies: A simple but scalable approach," *IEEE Intelligent Systems*, vol. 18, no. 5, pp. 60–65, 2003.

[6] A. Freitas, E. Curry, J. Oliveira, and S. O'Riain, "Querying heterogeneous datasets on the linked data web: Challenges, approaches, and trends," *Internet Computing, IEEE*, vol. 16, no. 1, pp. 24–33, Jan 2012.

[7]

[8] R. Ranjan, "Streaming big data processing in datacenter clouds," *Cloud Computing, IEEE*, vol. 1, no. 1, pp. 78–83, May 2014.

[9] W. Hall, T. Tiropanis, R. Tinati, X. Wang, M. Luczak-Rosch, and E. Simperl, "The web science observatory-the challenges of analytics over distributed linked data infrastructures," *ERCIM News*, no. 96, pp. 29–30, 2014.

[10] T.-S. Chua, H. Luan, M. Sun, and S. Yang, "Next: Nus-tsinghua center for extreme search of user-generated content," *MultiMedia, IEEE*, vol. 19, no. 3, pp. 81–87, July 2012.

[11] P. Burnap, O. Rana, M. Williams, W. Housley, A. Edwards, J. Morgan, L. Sloan, and J. Conejero, "Cosmos: Towards an integrated and scalable service for analysing social media on demand," *International Journal of Parallel, Emergent and Distributed Systems*, no. ahead-of-print, pp. 1–21, 2014.

[12] S. Pongpaichet, V. K. Singh, M. Gao, and R. Jain, "Eventshop: Recognizing situations in web data streams," in *Proceedings of the 22Nd International Conference on World Wide Web Companion*, ser. WWW '13 Companion. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, pp. 1359–1368. [Online]. Available: http://dl.acm.org/citation.cfm?id=2487788.2488175

[13] M. S. Weber, "Observing the web by understanding the past: Archival internet research," in *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, ser. WWW Companion '14. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2014, pp. 1031–1036. [Online]. Available: http://dx.doi.org/10.1145/2567948.2579213

[14] P. C. Brown, *TIBCO Architecture Fundamentals*. Addison-Wesley, 2011.

[15] T. Tiropanis, W. Hall, J. Hendler, and C. de Larrinaga, "The web observatory: A middle layer for broad data," September 2014. [Online]. Available: http://eprints.soton.ac.uk/369910/

[16] T. Tiropanis, X. Wang, R. Tinati, and W. Hall, "Building a connected web observatory: architecture and challenges," in *2nd International Workshop on Building Web Observatories (B-WOW14), ACM Web Science Conference 2014*, June 2014. [Online]. Available: http://eprints.soton.ac.uk/366270/

[17] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Computing*, vol. 10, no. 6, pp. 87–89, Nov. 2006. [Online]. Available: http://dx.doi.org/10.1109/MIC.2006.116

[18] P. Houston, "Building distributed applications with message queuing middleware," 1998.

[19] D. J. Abadi, W. Lindner, S. Madden, and J. Schuler, "An integration framework for sensor networks and data stream management systems," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004, pp. 1361–1364.

[20] D. R. Karger and W. Jones, "Data unification in personal information management," *Commun. ACM*, vol. 49, no. 1, pp. 77–82, Jan. 2006. [Online]. Available: http://doi.acm.org/10.1145/1107458.1107496

[21] F. Maali and John Erickson, "Data Catalog Vocabulary (DCAT)," 2014. [Online]. Available: http://www.w3.org/TR/vocab-dcat/

[22] T. Tiropanis, A. Rowland-Campbell, and W. Hall, "Government as a social machine in an ecosystem," in *Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion*, ser. WWW Companion '14. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2014, pp. 903–904. [Online]. Available: http://dx.doi.org/10.1145/2567948.2578837

[23] G.-H. Kim, S. Trimi, and J.-H. Chung, "Big-data applications in the government sector," *Commun. ACM*, vol. 57, no. 3, pp. 78–85, Mar. 2014. [Online]. Available: http://doi.acm.org/10.1145/2500873