A minimal governance layer for the web of linked data
Alexander Grüneberg, Jonas Almeida
Div. Informatics, Dept. Pathology, University of Alabama at Birmingham, USA

Introduction
With the maturation of both the Linked Data Platform [1] and JSON-LD [2], RESTful principles and web linking present an opportunity to break away from the rigid world of RDF, and to promote the re-decentralization of the web to truly evolve it into a global data space [3]. Furthermore, there is a clear rationale in web development towards moving business logic into the client, replacing complex and costly backends by simpler, comoditizable data bases. However, NoSQL databases that expose RESTful APIs and offer native support for the web-friendly data-interchange format JSON have very limited governance modeling options.

To solve this problem, we describe a formal governance layer for RESTful Web Services that consists of a governance engine that modulates access to resources, and a special resource at a well-known location with which users can manage governance-related metadata described in a simple rule format that specifies the relationship between operators and agents. A simple hierarchical inheritance model that links the metadata of one resource to that of another facilitates the management of governance. We demonstrate the flexibility of this model in an implementation of a governed RESTful document store called Bounce.

Methods
HTTP is used for all interactions between agents and resources that are identified by URIs. Links between resources are either made using HTTP Link headers or linked data-interchange formats such as JSON-LD or HAL [4]. Agents authenticate to the server using basic access authentication. CORS [5] is used to enable cross-origin requests in browsers. Bounce is written in Node.js and stores data in MongoDB. All source code was kept under revision control and is available under MIT license at [https://github.com/agrueneberg/Bounce](https://github.com/agrueneberg/Bounce)/.

Results
The governance layer acts as an intermediary and sits on top of the resources that are exposed by a RESTful Web Service. It consists of a governance engine and a dedicated resource to manage governance-related metadata on a per-resource basis.

Implementation
To validate the approach and inform further development, the governance layer was implemented on top of the document-oriented NoSQL database MongoDB. Bounce adds governance and user management to MongoDB without interfering with its native analysis capabilities such as MapReduce. It stores JSON documents and files in collections that are contained in a single database. A RESTful APIs allows creating, reading, updating, deleting, and querying collections, documents, and files.

Conclusions
We identified a minimal mechanism to provide governance-related metadata for RESTful Web Services. An open source prototype supporting it was developed for Node.js to govern JSON documents stored in MongoDB ([https://github.com/agrueneberg/Bounce](https://github.com/agrueneberg/Bounce)). The simple inheritance mechanisms proposed relies on the basic operator-agent governance description that is also a core feature of distributed governance in the S3DB model [7]. Abstracting governance-related metadata into a dedicated resource and linking to it from governed resources also enables support for arbitrary file types, placing them under the same uniform interface constraints of other data elements. Reflecting the inherent extensibility and dynamic nature of the proposed mechanism, it places no limitations on the identification of access negotiation algorithms or restrictions on support for metadata formats.

References