Decentralizing the Semantic Web through Incentivized Collaboration

Ruben Verborgh

IDLab, Dep. of Electronics and Information Systems, Ghent University – imec
ruben.verborgh@ugent.be

Abstract. Personal data is being centralized at an unprecedented scale, and this comes with widely known and far-reaching consequences, considering the recent data scandals with companies such as Equifax and Facebook. Decentralizing personal data storage allows people to take back control of their data, and Semantic Web technologies can facilitate data integration at runtime. However, such data processing over decentralized data requires far more expensive algorithms, while at the same time, less processing power is available in individual stores compared to large-scale data centers. This article presents a vision in which nodes in decentralized networks are incentivized to collaborate on data processing using a distributed ledger. By leveraging the collective processing capacity of all nodes, we can provide a sustainable alternative to the current generation of centralized solutions, and thereby put people back in control without compromising on functionality.

1 Decentralizing personal data storage to regain control

The past couple of years, we have witnessed an unprecedented centralization of personal data on the Web. Large-scale social media networks collect our information, with or without our conscious approval, and store and process it centrally in powerful data warehouses. People are requested to hand over the control of their personal data in order to receive the services they want. For instance, on many social platforms, creating a photo album for sharing with family members involves uploading your photos to those platforms. Serious data scandals with companies such as Equifax and Facebook point to the inherent dangers of bringing such large amounts of data together in one place. Unsurprisingly, *taking back control of our own data* and *obtaining trusted information* are two of three major challenges formulated by Web inventor Tim Berners-Lee in 2017 [2].

Putting people back in control of their data means offering them the *choice* of storing that data wherever they want, independently of the applications they want to use. This is a core idea behind initiatives such as Solid [5]: data is *decentralized* in the sense that everyone can store their data in their own space, and applications are *decoupled* from data because resources created with one application can be read and modified by another. An example can be seen in Fig. 1, where a social feed can display pictures and events created by other applications. Moreover, the social feed is constructed by querying data from *multiple* storage locations, without prior centralization. This way, people are free to choose their storage provider and their application provider independently, and can move their data away at will. They can give applications, other people, or companies access to specific parts of their data as they see fit, and revoke or restrict that permission at any given point in time. This results in true data ownership and full control.

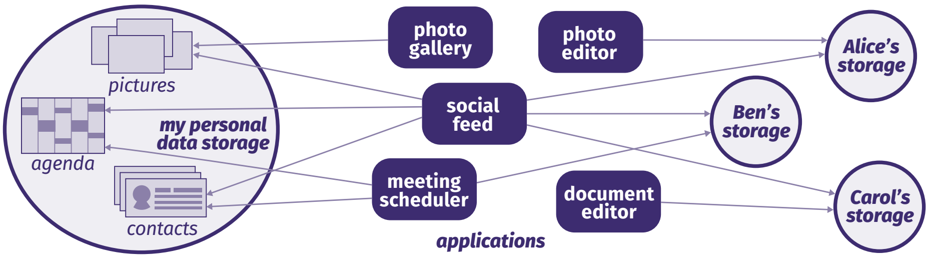

Fig. 1. Rather than demanding ownership, applications query data from decentralized locations.

Such a wide cross-application interoperability without strong prior agreements can be achieved by encoding *semantics* along with data and queries, as is possible with Semantic Web technologies like RDF and SPARQL. Data can be represented through a choice of widely used and custom ontologies. Every person is free to pick their ontologies and, because of semantics, reasoning can bridge ontological differences. In other words, the decentralized aspects of Linked Data and the uncoordinated nature of RDFS and OWL ontologies are a good fit for such scenarios [5].

2 Performance problems of decentralization

Compared to centralized systems, decentralized systems are facing a double disadvantage: individual nodes are not only solving a *harder problem*, they are doing so with *far fewer resources*. On the one hand, algorithms for decentralized data processing require significantly more processing power and network bandwidth than their centralized counterparts, because of heterogeneity and distribution. On the other hand, each individual node in the network—be it a data store or a client running an application—possesses far less computational power and bandwidth than large centralized data centers.

Furthermore, many of our data processing algorithms are not prepared for the scale of decentralization entailed by full data ownership. As a simple but realistic example, building the social media feed of a person with 500 friends requires executing a query over 500 different data sources in the worst case, where each of those friends store their data at a different location. State-of-the-art federated SPARQL query engines consider use cases of a *dozen* of *large* datasets with entirely *different* data shapes. In contrast, decentralized data storage will require federated queries over *hundreds* of *small* datasets with highly *similar* shapes. Current summarization and source selection strategies, crucial to federated performance, are not designed to function under such conditions.

Finally, exposing personal data storage through query endpoints comes with challenges of its own. Federated SPARQL query engines are usually benchmarked in private networks. On the public Web, SPARQL endpoints have long suffered from availability problems [3], and regardless of whether the causes are technological or managerial, there is a non-negligible risk that such problems would manifest themselves with at least a part of personal data stores. While less expressive query interfaces have shown promise on public networks [7], as data becomes spread across an increasing number of nodes, we can expect to run into severe bandwidth usage and associated query slowdowns.

3 Leveraging strength in numbers through collaboration

Decentralized networks have a particular asset: even though individual nodes have limited resources compared to large-scale server clusters, *collectively*, these nodes possess a far larger amount of computational power and bandwidth. Every single personal data store, as well as every client (computers, smartphones, tablets, . . .), brings their own CPUs—which, in a centralized environment, are typically underused. If we find a way for these nodes to *collaborate*, we solve the resources problem in decentralized networks. If we take optimization measures, such as performing preparatory work on the nodes closest to the data, we can counter the increased complexity of decentralized algorithms.

Let us apply this insight to the data gathering phase of applications, which in a decentralized network amounts to *federated query evaluation*. A straightforward query to collect the recent activity of one's contacts would involve the application sending subqueries to each of those contacts' data stores. However, social media networks typically contain overlapping clusters of people, so any person on a contact list is likely to have a subset of that list as contacts too. Therefore, we can set up agreements along the lines of *"I will help you execute your query if you help me execute mine"*. Then instead of sending subqueries to, for instance, 500 contact nodes, we can delegate larger subqueries to 10 or 20 hubs in parallel. Instead of executing data gathering entirely at the server or the client [7], we thus dynamically redistribute query execution across the network.

4 Providing incentivization and trust through distributed ledgers

In order to reach sustainable collaborations, nodes need to be *incentivized* to act as a contributor to the network. Otherwise, a node cannot be sure that, if it helps other nodes while idle, the others will return the favor when needed. However, when incentives are created, nodes also gain a reason for dishonest behavior, so we will need a *trust* mechanism to verify whether the work was completed correctly. For lack of a centralized entity in the network, such incentives and trust need to be established through *decentralized consensus*. This is possible through *distributed ledgers* [6], which can keep track of the work performed and hence the right to receive help from others.

One category of distributed ledgers are *blockchains* [6], which require a *proof* in order to add something to a ledger. Whereas the popular Bitcoin ledger is known for an essentially meaningless computation as *proof-of-work*, newer types of ledgers such as Filecoin [1] introduce more meaningful purposes for this proof. With Filecoin, people can pay others to securely store and retrieve their data, and a *proof-of-replication* confirms that the data is there at all times. We would similarly need to develop a *proof-of-query-results* that captures both the work performed as well was the correctness of the results.

Figure 2 shows the architectural components of an individual node in the network. When a query arrives, the node determines what incentive it is willing to accept, and what incentives it is wiling to pay others for subquery delegation. After possibly delegating some parts, and performing the remaining work itself, it maintains *provenance* of the data and generates a correctness proof of the results. Transactions are registered on the blockchain, such that all participants receive their reward. Some nodes might start performing preparatory work, such as precomputing partial results of common queries in the network, or locally caching other stores' data to speed up querying.

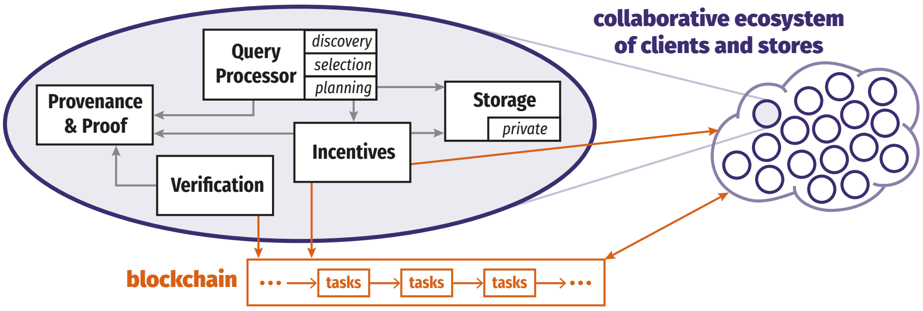

Fig. 2. Each node in the network has a query processor that can evaluate queries itself or (partially) delegate to others. Incentive modeling captures the required reward, and provenance and proof provide correctness guarantees. Performed tasks and their incentives are recorded on a blockchain.

5 Projected impact

This idea goes beyond *data marketplaces* [4] by in essence proposing a *service marketplace* between nodes in a decentralized semantic data network. While the example applies this to query execution over personal data, other kinds of services can be auctioned as well, such as *reasoning* to convert data to different ontologies. All such applications rely on the principle that client CPUs are idle most of the time, so by allowing others to use it when we do not, we can rely on them at the moment we need it ourselves.

This proposal can have a strong impact on the scale at which we apply Semantic Web technologies, especially in absence of clear business models. It opens up new directions in decentralized algorithms, and creates a connection between the Semantic Web and agent theory, as well as economic models for incentives. We also must pay attention to challenges such as privacy, perhaps through encryption. Most importantly, this vision sketches a Web-oriented future path to a Semantic Web for large *and* small players.

References

1. Filecoin: A decentralized storage network. Whitepaper, Protocol Labs (Aug 2017), *https://filecoin.io/filecoin.pdf*
2. Berners-Lee, T.: Three challenges for the Web, according to its inventor. World Wide Web Foundation (Mar 2017), *https://webfoundation.org/2017/03/web-turns-28-letter/*
3. Buil-Aranda, C., Hogan, A., Umbrich, J., Vandenbussche, P.Y.: SPARQL Web-querying infrastructure: Ready for action? In: Proc. of the 12th Int. Semantic Web Conference (2013)
4. Grubenmann, T., Dell'Aglio, D., Bernstein, A., Moor, D., Seuken, S.: Decentralizing the Semantic Web: who will pay to realize it? In: Proceedings of the Workshop on Decentralizing the Semantic Web (2017)
5. Mansour, E., Sambra, A.V., Hawke, S., Zereba, M., Capadisli, S., Ghanem, A., Aboulnaga, A., Berners-Lee, T.: A demonstration of the Solid platform for social Web applications. In: Companion Proceedings of the 25th International Conference on World Wide Web (2016)
6. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system, *https://bitcoin.org/bitcoin.pdf*
7. Verborgh, R., Vander Sande, M., Hartig, O., Van Herwegen, J., De Vocht, L., De Meester, B., Haesendonck, G., Colpaert, P.: Triple Pattern Fragments: a low-cost knowledge graph interface for the Web. Journal of Web Semantics 37–38 (Mar 2016)