Building Decentralized Applications for the Social Web

Andrei Sambra
Decentralised Information Group, CSAIL, MIT, Cambridge, USA
andrei@w3.org

Amy Guy
School of Informatics, University of Edinburgh, Edinburgh, UK
amy@rhiaro.co.uk

Sarven Capadisli
Enterprise Information Systems Department, University of Bonn, Bonn, Germany
info@csarven.ca

Nicola Greco
Decentralised Information Group, CSAIL, MIT, Cambridge, USA
ngreco@mit.edu

Keywords
decentralization, personal data stores, linked data, read-write web, social web

1. INTRODUCTION
Recent advancements in technologies and protocols mean that it is easier than ever to integrate social features into diverse web applications, and increased awareness of privacy concerns means that it is pertinent to consider empowerment of application users when doing so. Many developers are already familiar with the notion of personal data stores; this tutorial will demonstrate how to access or provide such stores for users, and build simple web applications which read and write to them. This advantages developers in two ways: by removing the burden of storing and maintaining a canonical copy of user data; and by enabling access to and ease of integration with data created through other applications, creating richer, seamless experiences. From the application users' perspective, they need no longer commit and become bound to particular services, but can mix, match and move between those that best meet their needs.

We will introduce *Solid*, a set of protocols based on existing W3C recommendations, for reading, writing and access control of the contents of a personal data store, which can be layered up in order to integrate various social features into new or existing web applications. Our intent is to encourage and facilitate decentralized, cross-application interactions. Our intended audience include:

WWW 2016 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2891060

- Individuals who wish to have full control of their own data through a personal data store, or who wish to provide this service for others.
- Developers of existing web applications who now want to add social features to enhance user experience or increase engagement.
- Those with an idea for a new application which would be enriched by social features, and are working out how to get started.
- Anyone who thinks web users should take control of their personal data and wants to see how this can be done with existing web technologies, today.

Attendees will leave with an understanding of Solid and how different parts of the protocols can work together, and having written some code to implement the parts that interest them most. They will also have hands on experience with existing libraries and tooling to facilitate working with the Solid protocols. Those who stay for the full day will have an opportunity to build a small but complete web application with decentralized social features, and to collaborate with others to see the advantages of sharing data between multiple applications.

2. SCHEDULE
We propose a full day tutorial; the morning serves as an introduction to important concepts and some guided coding examples, and the afternoon is a less structured hack session through which the morning's work can be explored and expanded. The following four key areas will be covered over the course of the day.

Data Ownership: Principles, Pros & Possibilities

The design of the Solid protocols stems from the original vision of the Web, and we will recap some key principles as a foundation for the rest of the session. This will introduce the motivations behind the Solid initiative and the benefits of distributed data ownership to both application developers and users, as well as a discussion of social, political and technical challenges yet to be overcome.

Servers

Solid servers - the home of user data, decoupled from applications which read and write it - will be introduced through creating accounts at the existing public personal data store service at databox.me. Those who wish to may install their own instances of any of the open source server implementations. Further, the protocol itself will be introduced, so that attendees may understand how to implement their own Solid-compliant server, potentially as part of an existing server stack they already use.

Clients

Solid clients are the interface to read and write data to users' personal data stores through the Web. Participants will be guided through creation of simple JavaScript applications that operate on data in their Solid-compliant stores. There will be a brief walkthrough of available libraries and tooling to help with the creation of Solid-compliant clients, and participants are encouraged to explore and experiment in the direction of their own particular interest once the key parts of the protocols are understood.

Integration & Interoperability

For many, the most exciting possibility will be implementing the Solid protocols into existing applications or technology stacks. We emphasise the importance of this, and strongly encourage participants to bring along existing projects that they wish to add social features to, or for which they wish to replace a centralised backend with a user-controlled data store. We will give guidance on how to prompt a user to point to their POD (Personal Online Datastore), and how to manage data there.

We anticipate that participants will work together to experiment with interoperability of independently created Solid-compliant applications. For example, data created with one app may be read and processed by another, and edited and deleted by yet another, according to access granted by the user. The protocols additionally allow data federation through the passing of notifications between applications and between users, via the personal data stores. Participants will be challenged to replicate functionality of their favorite centralised web applications by conceptually breaking such systems down and implementing smaller pieces which can work together.

Solid is an ongoing project, evolving with the diverse needs of users of the social Web. Workshop participants are encouraged to contribute on an ongoing basis by building Solid clients and servers, as well as updating the protocols as new requirements arise or become apparent through their own research and development work. Solid can be followed on Github: github.com/solid.

PREREQUISITES

- Some web programming/scripting experience; any language, but JavaScript is beneficial.
- Understanding of linked data is a plus but not required (key principles will be taught in the introduction).
- Participants should bring laptops equipped with a code editor and a Web browser.

ORGANISERS

- Andrei Sambra (MIT/W3C)
- Amy Guy (University of Edinburgh & MIT)
- Sarven Capadisli (University of Bonn & MIT)
- Nicola Greco (MIT)

The organisers are among the original developers of the Solid protocols, engaged in ongoing related standardisation work with the W3C, and actively developing services which use Solid from a number of different perspectives. As such, we are ideally placed to introduce the fundamental concepts and guide and support development activity during this full day tutorial.