Context and terminology

In this chapter, we will describe the principal elements needed to understand the context and motivation of this thesis. First, we will go through a quick recall of the history of the web and its current challenges. Then we will talk about the motivation behind Solid and describe its main components.

History and current state of the World Wide Web

In the 1980s, the European Organization for Nuclear Research, also known as CERN, was looking for a way to make physicists easily share their work and data around the globe. Internet was at its beginning, the TCP/IP protocol was freshly invented, but at this time, sharing data across multiple computers was laborious and inefficient[1]. A layer was lacking to write, transmit and store information easily through the network. In 1989, Sir Tim Bernes Lee submitted a proposal for an information management system to tackle the former issue with new concepts such as hypertext links, web browsers, the HTML language, and the HTTP protocol[2] [3].

On 1991 August 6, the first website available on the world wide web, "http://info.cern.ch" was born.

Today's web might be far from the decentralized digital utopia imagined by Sir Tim Berners Lee in the early 90s. Currently, most services, data, and patents related to web technologies belong to only a few companies, commonly referred to as GAFAM[4]. This oligopoly is in part due to a phenomenon called the "vendor lock-in": users tend to stay with the same platform since it is too hard to switch to another one . Furthermore, The centralization of web services and data has been a the heart of the latest internet scandal such as Cambridge Analytica or the worldwide NSA information gathering revealed by Edward Snowden[5].

Solid: Tim Berners-Lee proposal to Re-Decentralise the web

To tackle those problems, Tim Berners-Lee has started the Solid project in 2016 as a mean to re-decentralize the web[6]. Solid take advantage of the Semantic Web (also known as Web 3.0 but not to be confused with web3), a new set of standards extending the current World Wide Web to make the internet's data machine-readable. It is made machine-readable through Linked Data, a new way of formatting structured data to ensure its integration, interpolation, and interpretation by machines in the Semantic Web. Solid - sometimes stylized

[1] https://eu.usatoday.com/story/tech/news/2019/03/12/world-wide-web-turns-30-berners-lee-contract-thoughts-internet/3137726002/

[2] https://www.vox.com/2019/3/12/18260709/30th-anniversary-world-wide-web-google-doodle-history

[3] https://www.w3.org/History/1989/proposal.html

[4] https://www.cairn.info/article.php?ID_ARTICLE=COMLA_188_0061#

[5] https://www.theguardian.com/technology/2018/sep/08/decentralisation-next-big-step-for-the-world-wide-web-dweb-data-internet-censorship-brewster-kahle

[6] https://en.wikipedia.org/wiki/Solid_(web_decentralization_project)

SoLiD - stands for **So**cial **Li**nked **D**ata. It aims to give back data control to internet users and improve their privacy online. Solid is not a technology in itself, but rather a set of specifications that allows decoupling of web applications' authentification, data, and app logic[7] . The end goal is to give internet users complete control over their data[8] . However, the web needs a couple of new artifacts and vocabulary to allow this decoupling; the two most important are the Pod (decoupling data storage) and the Identity Provider (decoupling authentification).

A Pod (Personal Online Datastore) is where Solid users store their data online. Like a Unix filesystem, the WAC (Web Access Control) controls who can read, write or delete files stored in a Pod. Each of these permissions is defined in `.acl` "Access Control List" files[9].

WebIDs are the standardization of a universal identifier used for authentification. More than replacing the traditional username, it is a full URI that, once dereferenced, can give more information on the end end user. It usually has the form of: `https://my-webid-host.net/my-username/profile/card#me` . The Identity Provider (IDP) permits the user to login into various applications with their WebID in the fashion of today's "Sign in with Google/Facebook". It uses The Solid-OIDC protocol - built on top of the Open ID Connect - that manages authentification in a Solid environment.

Identity Provider and Pod Provider are two decoupled services, but a Solid Server usually serves both.

Let us illustrate all those new terms with an example: Alice has a pod hosted on `http://alice.pod.org` where she stores her data, including a holiday picture at `http://alice.pod.org/picture/holiday.png`. By associating the `READ` permission to Bob's WebID `https://bob.anotherpod.org/profile/card#me` in the ACL file `http://alice.pod.org/picture/holiday.acl`, now Bob, if authenticated with his WebID will be authorized to access the picture at `http://alice.pod.org/picture/holiday.png`.

Those specifications are developed by the World Wide Web Consortium (W3C), an organization of web standard founded by Tim Berners-Lee, which promote technologies compatibility around the web. Tim Berners-Lee has also created a startup called Inrupt that aims to build a Solid commercial solution. Inrupt is financing a team of researchers from Ghent University to develop the Community Solid Server.

Solid Community Server, an implementation of the Solid specifications

The Community Solid Server is an open-source Solid Server - i.e. a Pod and Identity Provider - that implements Solid Specification. It can also deliver

[7] https://solid.github.io/specification/protocol
[8] https://en.wikipedia.org/wiki/Solid_(web_decentralization_project)
[9] https://en.wikipedia.org/wiki/Solid_(web_decentralization_project)

WebIDs. Currently, only two implementations fulfill the Solid-specification: SCS and NSS (Node Solid Server). SCS can be considered a new replacement for the legacy NSS that power `https://solidcommunity.net`, currently the most used solid server. SCS is a newborn software under active development: version 1.0 was released in the symbolic month of August 2021, exactly 30 years after the World Wide Web first webpage, and version 3.0 was released the 23 February 2022 [10]. Inrupt financially support IDLab from Gent University (Belgium) to build the software. It's copyrighted by Inrupt and IMEC research and development hub under the MIT license. Built in a modular fashion, it has been designed for researchers and developers who want to test Solid App and/or design new features and experience with Solid[11]. Such modularity is empowered using components.js, a dependency injection framework at the core of SCS.

Component.js: a dependency injection that powers SCS modularity

Component.js is a javascript dependency injection developed by SCS authors. A Dependency Injection (DI) implements a form of inversion of control, a programming principle where part of a program receives its execution flow from a framework. Dependency injection will dynamically create (inject) the dependencies between the different components of a computer program. Therefore, the program execution flow is expressed not only through static code but also dynamically assigned during execution. In particular, components.js lets us describe the dependencies between SCS components from a JSON configuration file, usually named `config.json` or `config_<config description>.json`. Even if the SCS authors have written components.js mainly to answer SCS needs, it has been built as a general-purpose dependency injection framework and can be used for other software.

The innovation of components.js, compared to other javascript DI frameworks such as `inversify` or `typedi`, is to be built around the concept of Linked Data. In other words, the configuration files leverage the power of the semantic web: each component can be uniquely and globally identified through a URI. Furthermore, having configuration files machine-readable and built under the same vocabulary and makes it easy to generate, parse, compare, or edit them in a script.

CERN's IT infrastructure

CERN is primarily a high-energy physics laboratory. Counting 12,400 users from instititution from more than 70 countries [12], strong computing infrastructure is of paramount importance. To facilitate the host and deployment of web applications inside its computing environment, CERN has deployed a Platform-as-a-Service (PaaS) to its users.

[10] https://github.com/solid/community-server/releases
[11] https://github.com/solid/community-server/
[12] https://en.wikipedia.org/wiki/CERN

PaaS is a cloud computing service meant for developers. Its goal is to simplify workload by offering the developer to quickly initiate, run and manage one or more web applications without worrying about the computing infrastructure part such as networking, storage, OS and others. [13]

OKD4 or Openshift 4 is Red Hat's PaaS solution. The community version used at CERN is free and open source under the Apache 2 license [14]. It is powered by other popular open-source technologies such as Docker and Kubernetes.

CERN's Openshift allows its developers to quickly deploy web applications with strong DevOps tooling and provides them with a high-level integration to CERN's computing environment, such as user and access management[15].

[13] https://en.wikipedia.org/wiki/Platform_as_a_service
[14] https://okd.io/about/
[15] https://paas.docs.cern.ch/]