Bring Your Own Data

Public Values in Practice

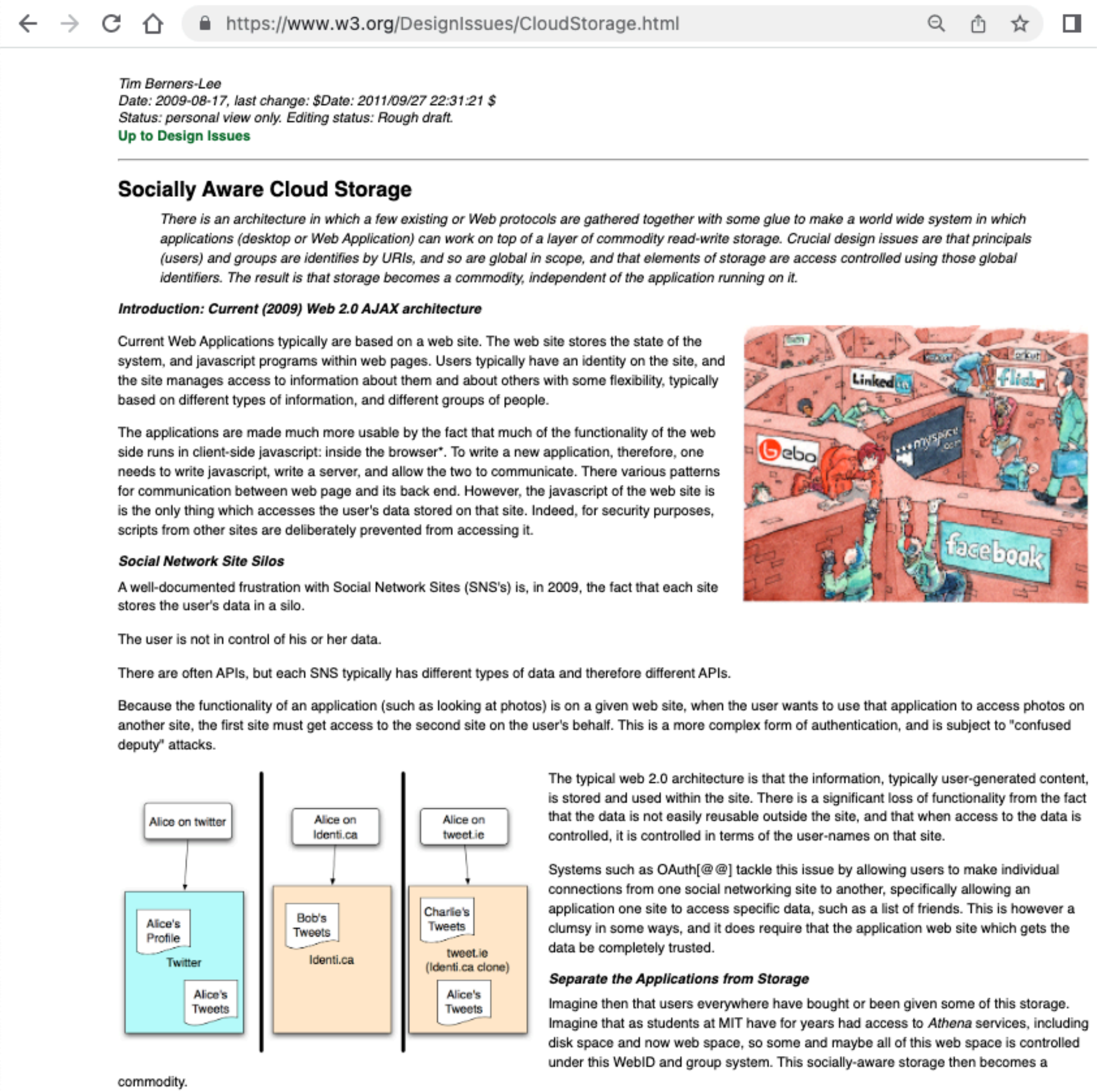
Michiel de Jong, post-growth entrepreneur at Ponder Source.

SEPARATING

APPS

FROM DATA

CERN
World Wide Web Initiative
EIZO



Tim Berners-Lee

Date: 2009-08-17, last change: $Date: 2011/09/27 22:31:21 $
Status: personal view only. Editing status: Rough draft.
Up to Design Issues

Socially Aware Cloud Storage

> *There is an architecture in which a few existing or Web protocols are gathered together with some glue to make a world wide system in which applications (desktop or Web Application) can work on top of a layer of commodity read-write storage. Crucial design issues are that principals (users) and groups are identifies by URIs, and so are global in scope, and that elements of storage are access controlled using those global identifiers. The result is that storage becomes a commodity, independent of the application running on it.*

Introduction: Current (2009) Web 2.0 AJAX architecture

Current Web Applications typically are based on a web site. The web site stores the state of the system, and javascript programs within web pages. Users typically have an identity on the site, and the site manages access to information about them and about others with some flexibility, typically based on different types of information, and different groups of people.

The applications are made much more usable by the fact that much of the functionality of the web side runs in client-side javascript: inside the browser*. To write a new application, therefore, one needs to write javascript, write a server, and allow the two to communicate. There various patterns for communication between web page and its back end. However, the javascript of the web site is is the only thing which accesses the user's data stored on that site. Indeed, for security purposes, scripts from other sites are deliberately prevented from accessing it.

Social Network Site Silos

A well-documented frustration with Social Network Sites (SNS's) is, in 2009, the fact that each site stores the user's data in a silo.

The user is not in control of his or her data.

There are often APIs, but each SNS typically has different types of data and therefore different APIs.

Because the functionality of an application (such as looking at photos) is on a given web site, when the user wants to use that application to access photos on another site, the first site must get access to the second site on the user's behalf. This is a more complex form of authentication, and is subject to "confused deputy" attacks.

The typical web 2.0 architecture is that the information, typically user-generated content, is stored and used within the site. There is a significant loss of functionality from the fact that the data is not easily reusable outside the site, and that when access to the data is controlled, it is controlled in terms of the user-names on that site.

Systems such as OAuth[@@] tackle this issue by allowing users to make individual connections from one social networking site to another, specifically allowing an application one site to access specific data, such as a list of friends. This is however a clumsy in some ways, and it does require that the application web site which gets the data be completely trusted.

Separate the Applications from Storage

Imagine then that users everywhere have bought or been given some of this storage. Imagine that as students at MIT have for years had access to *Athena* services, including disk space and now web space, so some and maybe all of this web space is controlled under this WebID and group system. This socially-aware storage then becomes a commodity.

Separate the Applications from Storage

Imagine then that users everywhere have bought or been given some of this storage. Imagine that as students at MIT have for years had access to *Athena* services, including disk space and now web space, so some and maybe all of this web space is controlled under this WebID and group system. This socially-aware storage then becomes a commodity.

This then allows a market in independently written applications.

PERSONAL

ONLINE

DATA STORE

PERSONAL

ONLINE

DATA STORE

UNHOSTED

us69.cz
CENOVÉ NÁKUPNÍ HITY
REALITY UNHOŠŤ
REALITY UNHOŠŤ
RYOR
RYOR
UNHOSTED
UNHOSTED
UNHOSTED
PŘÍJEM | VÝDEJ
vodafone

REMOTE STORAGE

Webfinger
user@provider

OAuth

CORS
cross-origin
AJAX calls

GET
PUT
DELETE
key-value
store

REMOTE STORAGE

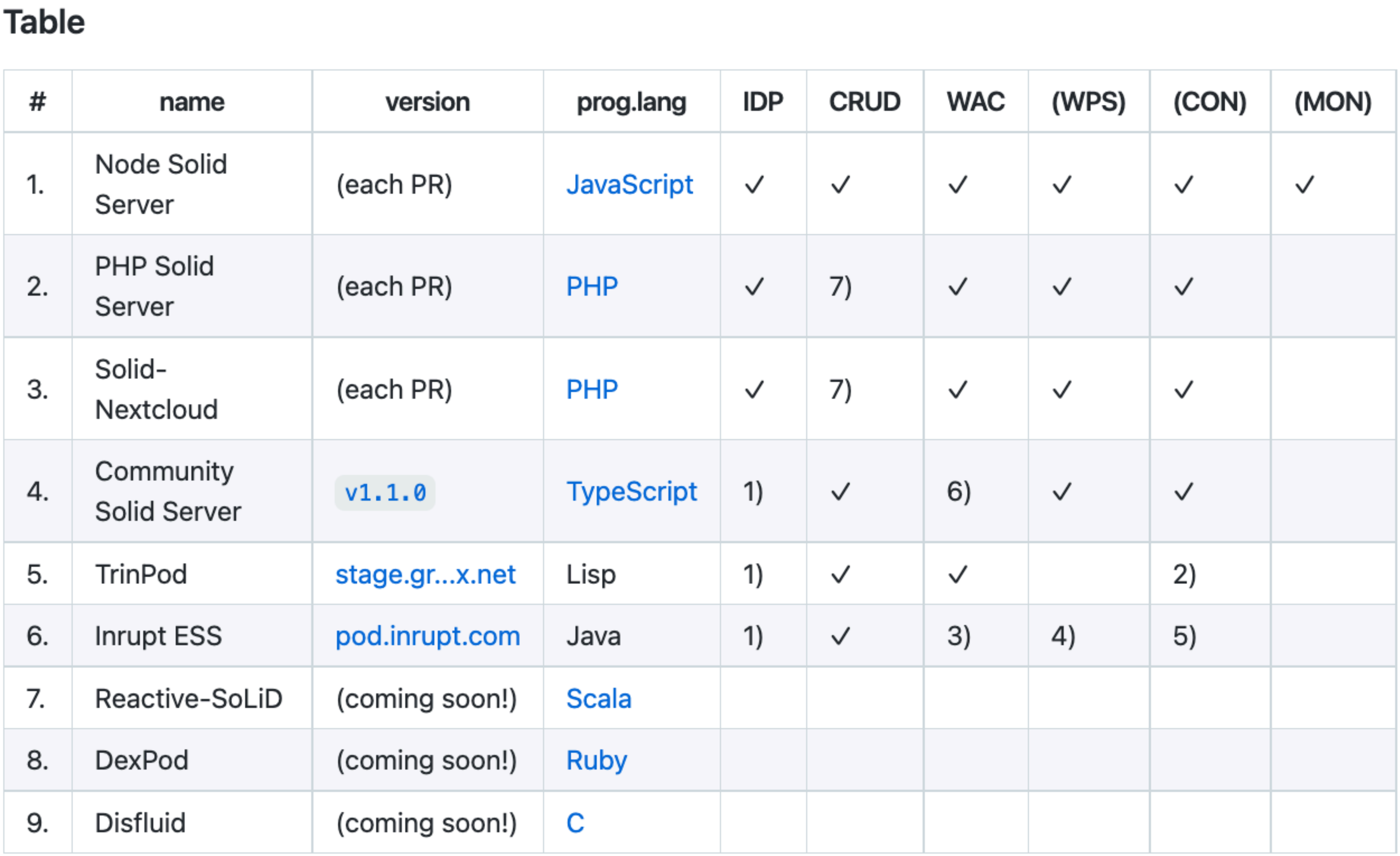
remoteStorage.js

SOLID

OPEN PROTOCOL

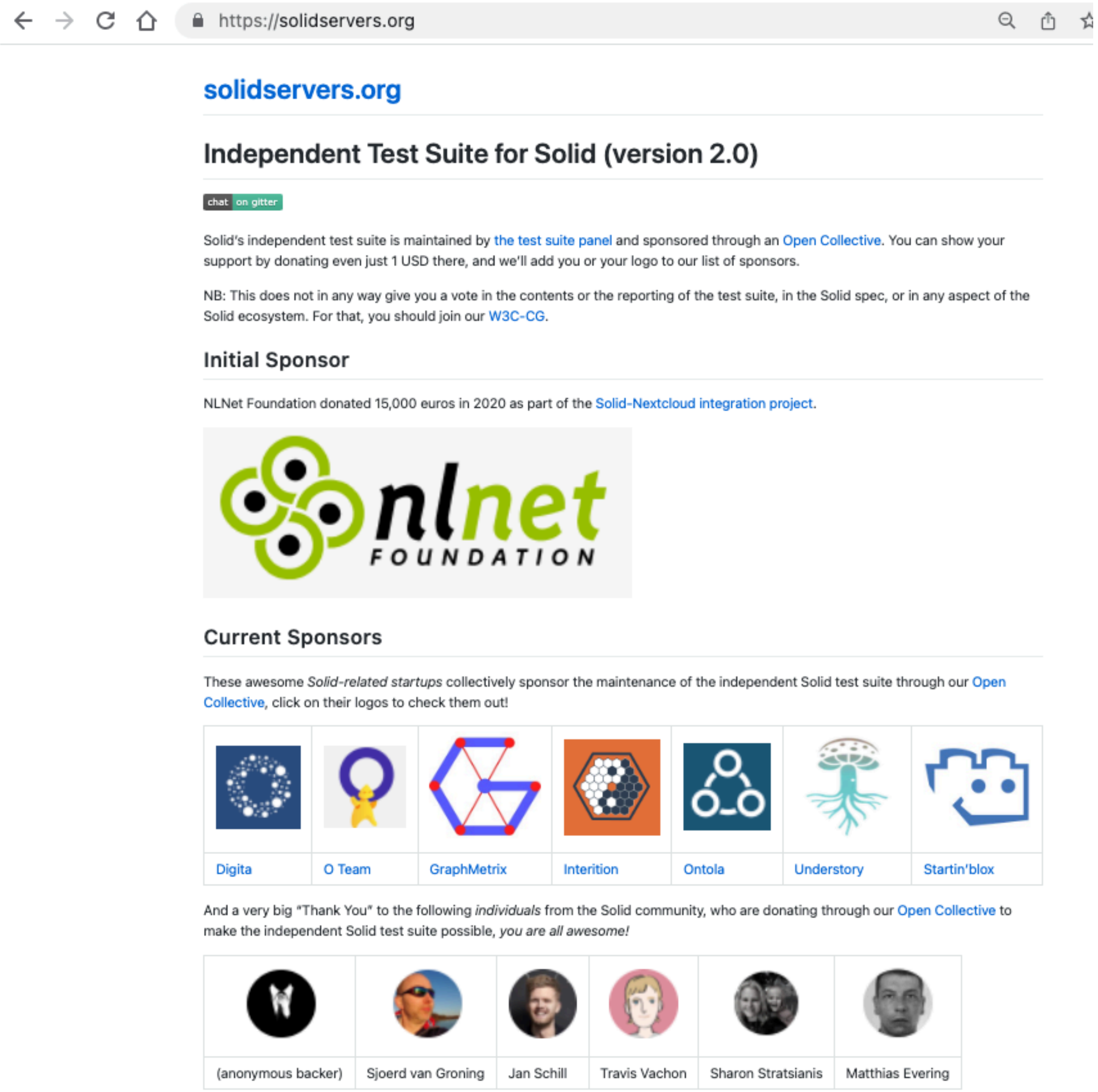
6 IMPLEMENTATIONS

Independent Test Suite



Table

#	name	version	prog.lang	IDP	CRUD	WAC	(WPS)	(CON)	(MON)
1.	Node Solid Server	(each PR)	JavaScript	✓	✓	✓	✓	✓	✓
2.	PHP Solid Server	(each PR)	PHP	✓	7)	✓	✓	✓	
3.	Solid-Nextcloud	(each PR)	PHP	✓	7)	✓	✓	✓	
4.	Community Solid Server	`v1.1.0`	TypeScript	1)	✓	6)	✓	✓	
5.	TrinPod	stage.gr...x.net	Lisp	1)	✓	✓		2)	
6.	Inrupt ESS	pod.inrupt.com	Java	1)	✓	3)	4)	5)	
7.	Reactive-SoLiD	(coming soon!)	Scala						
8.	DexPod	(coming soon!)	Ruby						
9.	Disfluid	(coming soon!)	C						



solidservers.org

Independent Test Suite for Solid (version 2.0)

Solid's independent test suite is maintained by the test suite panel and sponsored through an Open Collective. You can show your support by donating even just 1 USD there, and we'll add you or your logo to our list of sponsors.

NB: This does not in any way give you a vote in the contents or the reporting of the test suite, in the Solid spec, or in any aspect of the Solid ecosystem. For that, you should join our W3C-CG.

Initial Sponsor

NLNet Foundation donated 15,000 euros in 2020 as part of the Solid-Nextcloud integration project.

Current Sponsors

These awesome *Solid-related startups* collectively sponsor the maintenance of the independent Solid test suite through our Open Collective, click on their logos to check them out!

Digita	O Team	GraphMetrix	Interition	Ontola	Understory	Startin'blox

And a very big "Thank You" to the following *individuals* from the Solid community, who are donating through our Open Collective to make the independent Solid test suite possible, *you are all awesome!*

(anonymous backer)	Sjoerd van Groning	Jan Schill	Travis Vachon	Sharon Stratsianis	Matthias Evering

SOLIDSERVERS.ORG

PRACTICAL

DIFFICULTIES

CHICKEN

EGG

USER

EXPERIENCE

DATA

FORMATS

Content, Language, Origin, Grounding, Speech Act

POD

MIGRATIONS

HYBRID

APPROACH

SOVEREIGN

BUT

CONNECTED

WHY
THIS IS
AWESOME

APPLICATION

IDENTITY

STORAGE

USER

OWNS

DATA

BRING YOUR

OWN

DATA

SHARING AT STORAGE LEVEL

APPS

COME

PRE-FILLED

LIKE

BOOKMARKS

IMPORT

DEMO

TIME

Thank you!

https://pondersource.com