Table 2: Solid applications. An asterisk (*) indicates a third-party application, not developed by us.

Name	Function	Usable At
contacts	Manage a list of contacts	*http://mzereba.github.io/contacts*
contacts	Manage a list of contacts	*http://linkeddata.github.io/contacts*
calendar	Event manager	*http://mzereba.github.io/calendar*
dokieli	Decentralized authoring, annotation, and social notifications	*https://dokie.li*
pad	Shared collaborative editing	*https://github.com/timbl/pad*
profile-editor	View and update a user's profile	*http://linkeddata.github.io/profile-editor*
warp	Solid file browser	*http://linkeddata.github.io/warp*
cimba	Microblogging (cf. Twitter)	*http://cimba.co*
zagel	Instant messaging/group chat	*https://solid.github.io/solid-zagel*
***webid.im**	Instant messaging/chat	*http://webid.im*
***shamblokus**	Strategy game (cf. Blokus)	*http://deiu.github.io/Shamblokus*

store data on two different pod servers: databox.me and meccano.io. This section provides a specific demonstration scenario using these servers.

The demonstration scenario involve two users, Alice and Bob, using different pod servers. Alice will use the **gold** server at databox.me, and Bob will use the **meccano** server at meccano.io. We will show that although these are two totally different servers, both users can use the same applications to access and maintain their data. This can be shown using any of the applications in Table 2. An application will be able to create, modify, delete, and retrieve resources in the user's pod. Demonstration participants can view these resources using the **warp** file browser, and can also see the client-server interaction involved.

Besides the basic Solid functionality, the demonstration will turn to interoperability and access control. Interoperability will be demonstrated through the **dokieli** application enabling social interactions among users, and through applications using link-following queries. For example, we will demonstrate how Alice can use link-following queries in the **contacts** application to search in the public contacts of Bob. In addition to demonstrating interoperability, these examples will also demonstrate access control. They will also demonstrate other featured of Solid such as delegation[16] in order to allow a pod to speak on behalf of its owner. As before, demonstration participants can view the resources being created, observe client-server interactions, and also server-to-server interactions.

Another form of interoperability is having multiple applications use the same data. We will show that a user can use two different **contacts** applications to manage the same set of contacts. We will also demonstrate the portability provided by Solid by showing how Alice can easily migrate her pod from databox.me to meccano.io. After this migration, Alice needs to change her WebID profile to point to the new storage, and her applications will be redirected to the new pod.

5. CONCLUSION

Re-decentralizing the social Web is an important topic and an active area of research. The Solid platform is a concrete instance of a decentralized platform for social Web applications, providing decentralized authentication, decentralized data management, developer support in the form of libraries and web components, and a suite of running servers and example applications. This demonstration will show how the Solid platform can enable social applications while allowing each user to retain control of their pod. Demonstration participants will experience Solid from a user and application developer perspective. They will gain insights into the interoperability and portability features provided by Solid, the rich social features that it can enable, and the client and server machinery behind these features. A concrete appreciation of such a platform is very valuable in the ongoing discussion on re-decentralization.

[16]https://github.com/solid/solid-spec#webid-delegated-requests

6. REFERENCES

[1] L. M. Aiello and G. Ruffo. LotusNet: Tunable privacy for distributed online social network services. *Computer Communications*, 35(1), 2012.

[2] T. Berners-Lee, Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and analyzing linked data on the semantic Web. In *Proc. Int. Semantic Web User Interaction*, 2006.

[3] B. Dodson, I. Vo, T. J. Purtell, A. Cannon, and M. S. Lam. Musubi: Disintermediated interctive social feeds for mobile devices. In *Proc. World Wide Web Conf. (WWW)*, pages 211–220, 2012.

[4] D. Glazkov and H. Ito. Introduction to Web components. *W3C Working Group Note*, 14, 2014.

[5] G. Klyne and J. J. Carroll. Resource description framework (RDF): Concepts and abstract syntax. 2006.

[6] A. Malhotra, J. Arwe, and S. Speicher. Linked Data Platform Specification. W3C Recommendation, 2015. http://www.w3.org/TR/ldp/.

[7] A. Sambra, A. Guy, S. Capadisli, and N. Greco. Building decentralized applications for the social Web. Tutorial at the World Wide Web Conf. (WWW), 2016.

[8] A. V. Sambra, H. Story, and T. Berners-Lee. WebID Specification. 2014. http://www.w3.org/2005/Incubator/webid/spec/identity/.

[9] S. Schulz and T. Strufe. d2 deleting Diaspora: Practical attacks for profile discovery and deletion. In *Proc. IEEE Int. Conf. on Communications (ICC)*, 2013.

[10] M. Van Kleek, D. A. Smith, N. R. Shadbolt, and mc schraefel. A decentralized architecture for consolidating personal information ecosystems: The WebBox. In *Proc. Workshop on Personal Information Management (PIM)*, 2012.