Dataverse Single Page Application: Authentication Analysis and Design Summary

The development of a Single Page Application (SPA) front-end for Dataverse will require changes to how users login to Dataverse and the underlying design of the authentication mechanism. The key issue driving the need for change is that the third-party authentication mechanisms, as currently implemented in Dataverse, would not be secure when used with an SPA whose code is running in the user's browser. These mechanisms involve a secret shared between the Dataverse instance and the authentication provider that assures all parties understand that the login is for that given Dataverse. With an SPA and the current authentication design, that secret would have to be in every user's browser where it could potentially be exposed.

Fortunately, there is a design available that addresses this issue while retaining backward-compatibility with all of the authentication providers usable with Dataverse today. In the Dataverse - SPA Authentication V1 document, the Dataverse team has identified required and desirable functionality, documented possible design options, reviewed these options in the context of supporting the new SPA and other future Dataverse clients, identified an overall winning design, and developed an incremental plan for development. This document presents the outcomes of that effort with an emphasis on the path forward.

OpenID Connect (OIDC) with Proof Key for Code Exchange (PKCE)

OIDC is a popular protocol - supported by Google, ORCID, and many others - in which an application redirects the user to a third-party authentication provider for login and receives, via a multi-step authentication workflow, proof that the user has successfully logged in, along with basic information about the user such as their name and email. OIDC is already supported in Dataverse, along with the option for built-in users (where the user's encrypted password is stored in Dataverse's database), OAuth2 and Shibboleth/SAML, a protocol widely used in academia.

The PKCE (pronounced "pixie") variant of OIDC adds a challenge/response mechanism suitable for SPAs that guarantees that an OIDC exchange initiated by an application can only be completed by the software initiating the exchange. A further variant of PKCE involves the application back-end (e.g. Dataverse) in the final exchange allowing it to add a client secret and avoid storing the final access token in the browser, which increases security.

The Dataverse core team has selected OIDC+PKCE as the long-term choice for Dataverse. The only other option usable with an SPA would involve maintaining the current login/logout pages from the current UI. While this may make sense as a short-term option during SPA development, maintaining these pages for the long-term was not seen as viable.

OIDC Brokering and Keycloak

When OIDC+PKCE is implemented, Dataverse and the SPA would work with a single OIDC provider. For sites currently supporting more than one OIDC provider, it should be possible to retain that capability by maintaining the ability in Dataverse to register more than one provider, assuming those providers support PKCE. However, this approach would not provide backward compatibility for Dataverse instances currently using a non-PKCE OIDC provider, a Shibboleth/SAML or OAuth2 provider, and/or built-in users.

To support these cases, the design team has looked into the possibility of using an external OIDC-based broker - an application that can interact with Dataverse via OIDC+PKCE and then broker connections to other providers using OIDC, OAuth2, Shibboleth/SAML, LDAP, and/or even make custom connections to a database - as a way to provide this backward compatibility. Keycloak is an example of an open-source OIDC broker that also offers additional useful capabilities such as harmonizing the user attributes from brokered providers or even offering multi-account logins. With such a broker, Dataverse would only need to support a single OIDC+PKCE connection to the broker, simplifying the back-end code.

While running Keycloak or another OIDC broker as the way to support Shibboleth/SAML providers could mean that Dataverse installations will need to manage this additional component, it is also possible that institutions will stand up such services for campus-wide use. For installations that choose to run their own Keycloak (or other broker instance), the Dataverse team expects that a standardized base configuration and/or containerized version can be provided.

An Incremental Development Plan

As SPA development proceeds, the team expects that the migration to this new OIDC+PKCE with Optional Broker design can be done in stages:

- As an initial step, primarily for development and UI testing, Dataverse will support login via API and/or the existing login mechanism with the SPA, then calling the Dataverse API with an included session cookie for authentication. This mechanism can currently be enabled via a feature flag (new in Dataverse v5.14), although it is not secure enough for production use. If needed, this mechanism could be hardened for production use if better alternatives are not ready in time.
- A future Dataverse release will support OIDC+PKCE and Bearer Token based API Authentication, allowing initial testing against OIDC+PKCE providers.

- The SPA will then implement OIDC+PKCE with a choice (tbd) made as to whether to use the more secure back-end variant. When this is implemented, it should be possible to run the SPA and the current Dataverse UI securely at the same time.
- Guidance and standardized configuration/containers for running Keycloak as an OIDC broker with Dataverse will be developed. With this component, installations should be able to migrate from use of multiple OIDC and/or Shibboleth providers to doing so with the Keycloak broker and thereby be able to run the SPA and current UI with multiple providers.
- The Dataverse team will explore using a custom plugin to Keycloak or a migration script to allow it to be used as a way to login with built-in providers. When this is done, it will be possible to standardize the SPA, and potentially the current UI as well, to only allow login via OIDC+PKCE and to rely on Keycloak to provide support for any combination of built-in, Shibboleth, and OIDC providers. Whether/when direct support for Shibboleth in the existing UI and/or direct support for login to built-in accounts in the existing UI and SPA can be removed is a decision that can be made later after evaluation of the relative costs of running Keycloak versus maintaining the existing code.

Open Issues

There are variations within this overall design and development path where the Dataverse team will need to make decisions, including some that might affect others, as the developers and community gain experience with OIDC during 2023.

For example, there are choices to be made in how to authenticate individual API calls after login. OIDC specifies an access token that can be used for this purpose but does not define whether this token should be considered opaque or can include information such as the user's account name. The latter case, which is supported by Keycloak, and could be assured if Dataverse implements the backend variant of OIDC+PKCE, is more scalable. Making a decision involves an assessment of the tradeoffs between efficiency, coding complexity, ease of operations, etc. that will in turn be affected by how widely backend OIDC+PKCE is supported, whether non-opaque tokens are supported widely, etc.

Further information on these and other issues, e.g. some related to logout mechanisms, and to whether built-in users should be migrated to an OIDC broker, are discussed further in the longer analysis document linked in the introduction. The Dataverse team expects decisions in these areas as to what to support initially and longer-term, and whether further options should be supported over time, will be made as development proceeds and as early adopters gain more experience in using OIDC and Keycloak and interacting with institutional OIDC services.