Mahara integration blueprint

Proposed by

Catalyst.Net Limited

Version 1.0

September 2014

catalyst

open source technologists

Level 6, Catalyst House, 150 Willis Street, Wellington 6011
PO Box 11053, Manners Street, Wellington 6142, New Zealand
+64 4 499 2267 // info@catalyst.net.nz // www.catalyst.net.nz

Table of contents

1 Introduction

Integration for Mahara is largely influenced by two things:

- A close relationship with the Moodle community and ecosystem with tight software integration;
- The legacy integration framework MNet does not have a widely supported set of transport protocols, and service architecture.

MNet as a technology in Moodle and Mahara dates back to at least 2007. Since then web service architecture has matured considerably in software where in general token-based authentication mechanisms (e.g. OAuth) and either REST or WS-* methodologies dominate, with payload serialisation formats centred around URL encoding, JSON, and SOAP related standards.

To this end, several years ago, Moodle implemented a flexible and generic web services framework that has the capacity to support these current trends and evolve as industry directions dictate.

Historically, in order to support MNet, both Moodle and Mahara have had this legacy protocol tightly coupled to their respective code bases, which is no longer desirable. From Moodle's point of view, it makes sense to only support one web services framework, and for Mahara MNet is effectively a dead end in terms of making the platform attractive to integration with other applications and services.

For the last 3 years, there has been various attempts made to start the process of replacing this legacy support, predominantly driven by Moodle. Each of these attempts have been held back by recurring problems:

- Replacing MNet is not adding new functionality therefore is hard to make a priority;
- It requires an architecture that meets the needs of Moodle and Mahara;
- It needs to meet both projects' release cycles;
- It needs to provide a migration pathway for both user bases;
- For Mahara, at least, it is a large undertaking for a small project to build a comparably sophisticated web services framework to that available in Moodle.

This document outlines a plan that enables both projects to move forward, providing their respective user bases with independent migration paths, while allowing Mahara to offer its community a comparable set of integration tools as found in Moodle.

2 The plan

2.1 Primary consideration

Because of the cost involved in redeveloping existing functionality, it is not enough to simply replace functionality in Mahara – rather, any development undertaken must provide maximum benefit and potential to the future of the project.

2.2 Objectives

- Provide a migration path away from MNet.
- Provide a web services framework that will open the way for new and interesting integration opportunities with Mahara.
- Provide a flexible authentication and authorisation framework that:
    - Is standards based;
    - Will outsource the complexity of managing the common protocols and authentication sources from Mahara;
    - Will make it possible to network *n* Mahara and / or Moodle instances together.

2.3 Broad component requirements

- Standardise on a compatible web services architecture between Moodle and Mahara.
- Robust and generic API framework that works well with primary integration partners, but is sufficiently standards compliant to work with the broadest range possible enabling Mahara to also meet the future requirements of its community.
- Mahara event API model that enables the extension of business processes to trigger custom work flows through modular plugins (Moodle has this).
- Generic client connection manager to be used by pluggable components (client side management of access tokens and credentials).
- Provide an alternative flexible authentication framework.
- Develop modular integration components to backfill MNet functionality in both Moodle and Mahara (plugins).

2.4 Decouple from MNet

Extracting MNet from Moodle and Mahara simultaneously is difficult. Therefore, a more realistic approach is to provide a generic alternative, and then deprecate MNet features when both projects have moved forward sufficiently towards original feature parity.

2.5 Separate authentication from integration

MNet is primarily used for both authentication and provisioning of user profile data between Moodle and Mahara, however this is not necessarily required for all integration scenarios. For instance, it is common to manage account provisioning directly in either system, but add the convenience of SSO integration with an institution directory via LDAP, CAS, SAML, Shibboleth etc. In adjunct to this it may be desirable to retain password control in individual systems, but synchronise account data between them.

To meet these varied requirements that include SSO, the designed approach is to use the SimpleSAMLphp project to provide a layer of abstraction away from the common authentication sources. This reduces the requirement for maintenance of additional authentication plugins, while enabling Mahara to tap into the list of

authentication sources that SimpleSAMLphp supports, and the SSO integration it provides.

Account provisioning can be covered by three options:

- Automatic provisioning of accounts based on the SAML payload;
- Manual account provisioning via existing Mahara account maintenance tools;
- 3rd party plugin development that can take advantage of the Mahara event model, and the client side web services tools.

It is a combination of SimpleSAMLphp and registered event API plugins that will form the basis for replacing the existing MNet functionality and enhance the SSO related integration options into the future.

For non-SSO related integration scenarios, it will be perfectly feasible to use the web services and event model API related plugins on their own.

2.6 Web services framework

The chosen model for the web services framework is the Moodle web services framework. This has a number of unique benefits:

- The majority of integration partners with Mahara (outside authentication concerns) are Moodle systems;
- The original Mahara code base has a lot of similarities with Moodle making translation of the Moodle web services framework viable;
- There are aspects of the Moodle web services framework such as token handling and API / plugin development that can be shared making it easier and faster for Moodle / Mahara developers to work with (most Mahara developers are likely to be Moodle developers too);
- It will be easier to keep feature parity / compatibility between Moodle and Mahara if the web services architecture share the same underlying model.

Note: it is an assumed requirement of the Mahara / Moodle web services frameworks that SSL site encryption is required as the API access credentials can be passed in the URI.

3 What has happened so far?

- Survey gathering requirements from community – https://docs.google.com/forms/d/1vyECf8s8lyEX6YJuf9SGe7sUN5flaROcfBF7mPxY9fc/viewform with announcements in both Moodle and Mahara forums;
- Mahara event API – https://wiki.mahara.org/index.php/Developer_Area/Events_API;
- Previous work (currently available as plugin for Mahara https://wiki.mahara.org/index.php/Plugins/Artefact/WebServices) done on replicating and extending the Moodle web services framework in Mahara (partly funded by the New Zealand Ministry of Education) https://github.com/piersharding/mahara-contrib-auth-webservice;
- MNet API encapsulated in the Mahara web services framework;
- Previous work done on integrating SimpleSAMLphp into Moodle and Mahara (partly funded by the

New Zealand Ministry of Education – Moodle support https://github.com/piersharding/moodle-auth_saml, and Mahara support in core);

- New authentication plugins developed for SimpleSAMLphp to backend authentication with either Moodle or Mahara – https://github.com/piersharding/simplesamlphp-modules.

4 Main survey findings

The main areas of interest in functionality were:

- Federated SSO with Moodle/s, or Mahara/s;
- Transfer of profile data, which can be equated with automated account provisioning;
- Automatic setup of groups in Mahara based on Moodle courses;
- Portfolio export / import;
- Portfolio (artefact/collections) versioning;
- General purpose artefact APIs;
- Evidence API (including badges).

5 Roadmap

- Integrate web services framework with Mahara core;
- Develop generic client connection manager;
- Revamp Mahara auth/saml to support generic SimpleSAMLphp integration;
- Develop alternate plugin support for MNet functionality in Moodle and Mahara – local plugins that hook into event APIs for identity and profile management, and call respective APIs in partner system.

6 Next steps

- Discuss plans with Moodle HQ – planned for October 2014.
- Code review, test, and import web services plugin into Mahara core - release 2Q 2015.
- Build client connection manager – TBD.
- Revamp Mahara auth/saml – TBD.