Sharing M3 content with Partners

Overview

Approach

The proposal is to utilise the Oauth 2.0 standard to share specific pieces of M3 content with members of a partner community in a way that is secure and enables M3 to accurately track engagement.

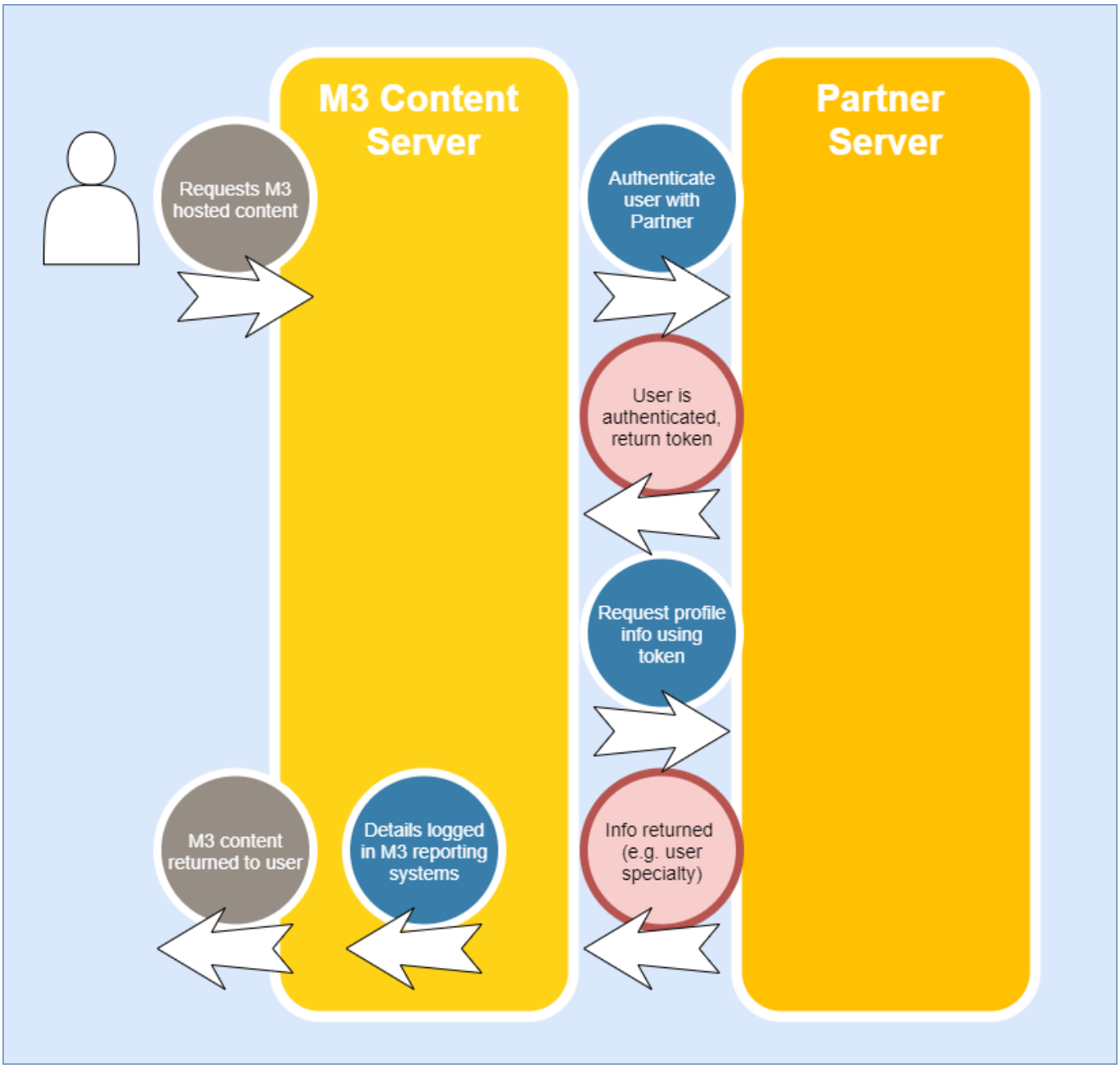
Principles of this approach:

- Partner users do not need to register with M3
- Partner users will only have access to specific agreed pieces of content, they are not "logging in" to M3 websites
- Agreed set of profile information is shared for reporting purposes only, but never enough to personally identify a user

Simplified Overview of Flow

Detailed documentation of the Oauth flow and work needed to implement it can be found later in this document but the below gives a simplied view of the process.

If already logged into the partner site the flow below will be invisible to the user and happen "behind the scenes". If not already logged in they will be presented with the Partner site login screen after the first step.



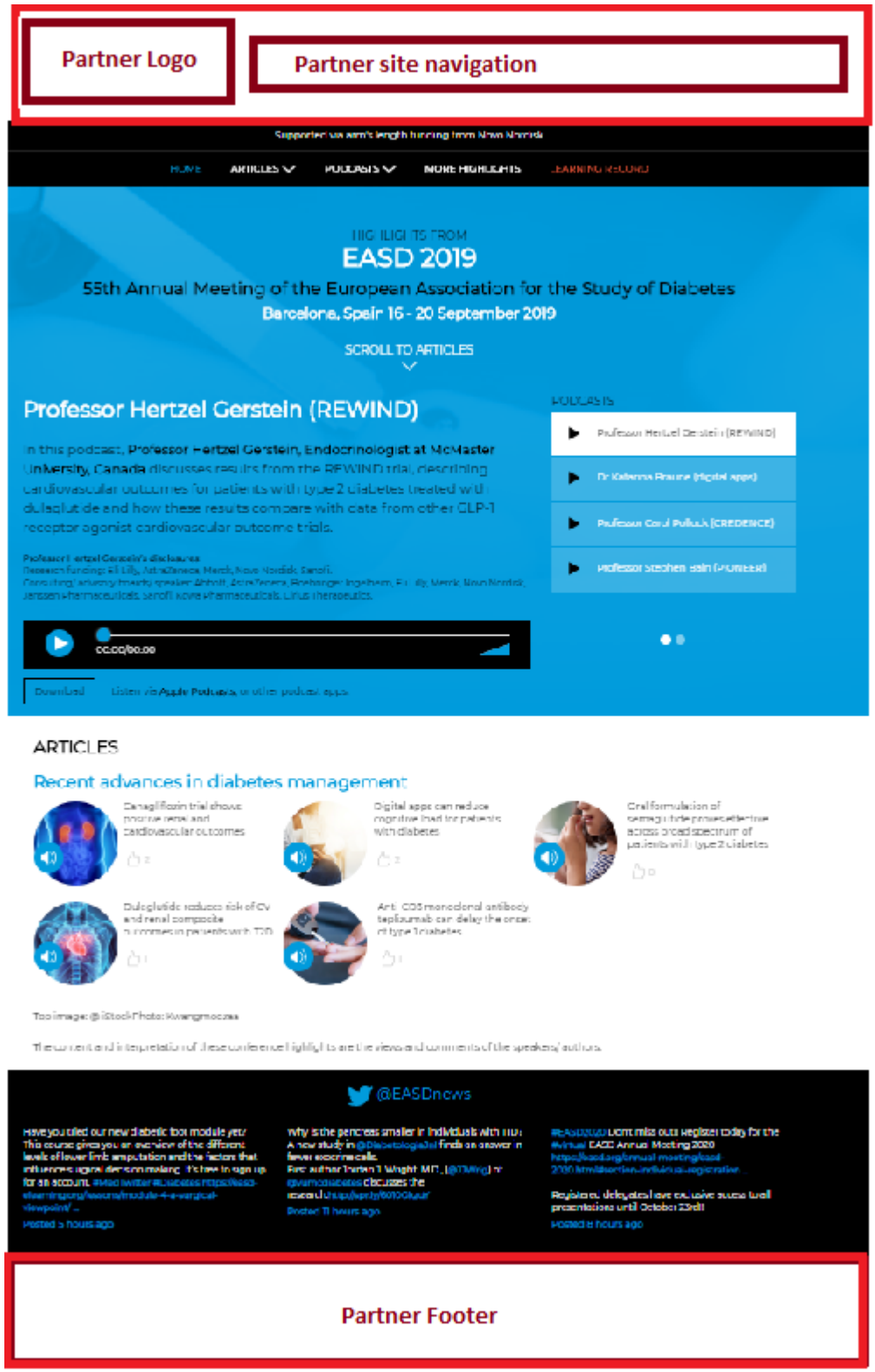
Header/Footer (optional)

If it is desired for partner users to "feel" like they are still on partner systems, partners can provide a header and footer to frame the M3 content.

This also provides the users a means to get back to the partner website.

If the header is personalised (for example has their name on it) then the partner should provide endpoints which will return the correct html to be injected into the page.

Alternatively if the header/footer are static and the same for all users, the partner can supply M3 the html blob to be inserted on the page, with no endpoints neccesary.



Reporting

As a minimum M3 will report on total engagement (impressions and clicks) and unique user engagement, and the only requirement for this to be possible is for an unique ID to be sent for each user as part of the Oauth flow.

For more detailed reporting M3 requires extra information to be passed, these are known as "scopes". The below defines typical scopes that can be used:

Scope	Description

Scope	Description
`User ID`	A unique ID per user, may be a string or number
`Specialty`	The user's medical specialty
`Seniority`	The user's medical seniority
`Country`	
`Doctor`	True or False. Useful for distinguishing staff/test accounts for example
`Groups`	This allows an arbitary list of tags to be assigned to a user. Can be useful if custom reporting is required that can't be satisfied by the above, for example by age groups

It is important that whatever info is transfered isn't enough to personally identify anyone. For example if postcode was passed in groups this may be fine on it's own. but in combination with user specialty it may become enough to find someone's identity.

Technical Implementation

Introduction

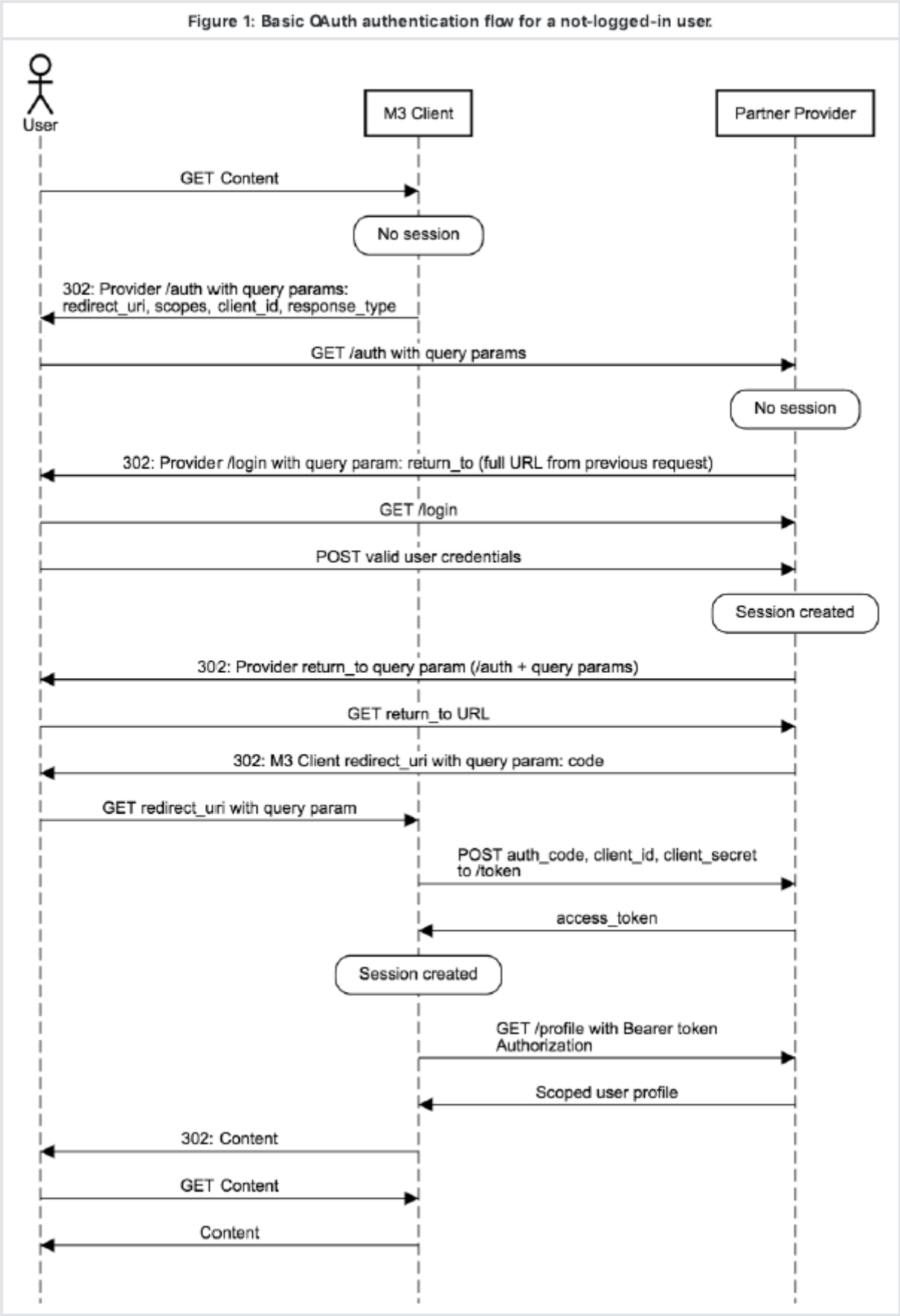
Authentication is handled using the OAuth 2.0 specification. OAuth is an open standard designed to enable users to authenticate themselves on site A using their existing account on site B. This is ideal for the scenario of sharing M3's content with partner communities.

The OAuth authentication flow uses a number of user-agent redirects, passing data using query parameters and a server-to-server request to establish a login session. Visualising the flow can be difficult, so below is a sequence diagram showing the full flow of a non-logged-in user landing on M3 content for the first time.

OAuth Authentication Flow

We will first consider the case where a user arrives without a session on the client. The steps required to establish a session on the M3 Client can be summarised by the following points:

1. Create a cookie-based session on the partner OAuth provider.
2. Pass an auth_code to the M3 OAuth Client from the Partner Provider via user-agent redirect.
3. Make a server-to-server request from the M3 OAuth Client to the provider to exchange the auth_code for an access_token .
4. Make a server-to-server request from the M3 OAuth Client to the provider to get the user profile using the access_token .
5. Create a cookie-based session on the M3 Client.



API Specification

To enable all of the above the Partner needs to implement the following API endpoints:

Method	Url	Description
GET	/oauth/auth	If the M3 client finds that the user has no valid session, then it will redirect the user to the PP Authorise endpoint.
POST	/oauth/token	After the PP has redirected the user to the M3OC callback endpoint, a server-to-server request will be made to exchange the provided auth_code for a valid access_token . The POST body will be sent with form encoding.
GET	/api/header	(OPTIONAL) Serves the site header as a snippet of HTML that can be inserted into any page
GET	/api/footer	(OPTIONAL) Serves the site header as a snippet of HTML that can be inserted into any page