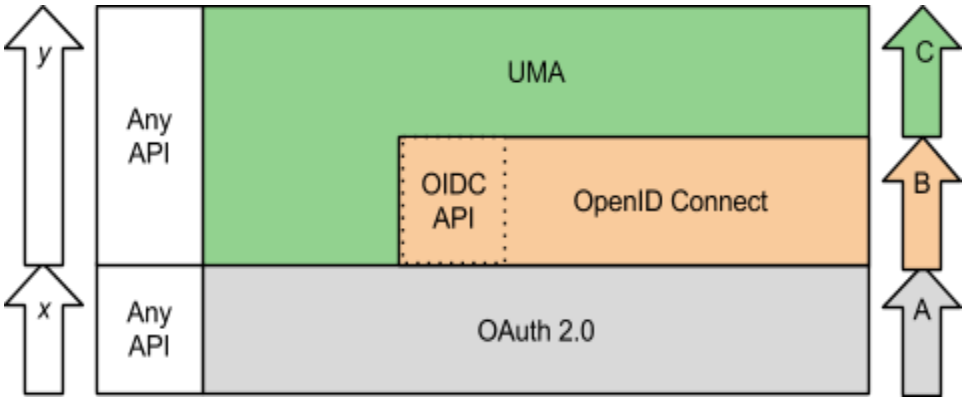
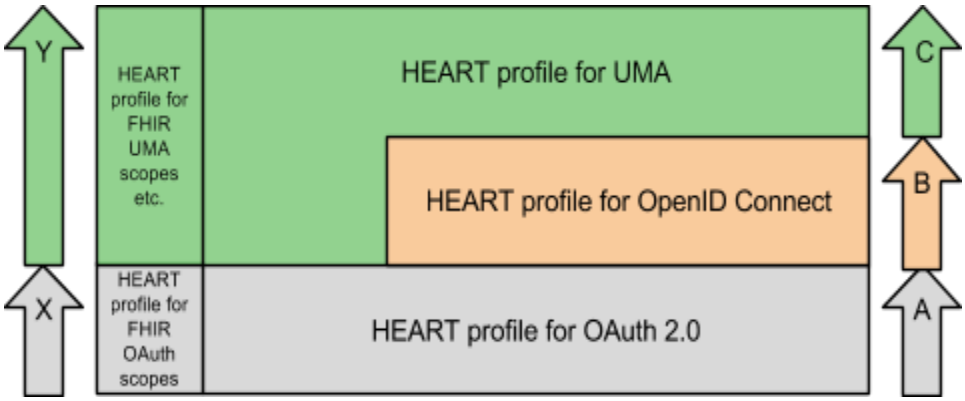
Venn technology decision tree

The HEART WG is mainly about profiling three technologies: OAuth, OpenID Connect, and UMA. OAuth is a “base” technology. OpenID Connect is built on top of OAuth. UMA is also built on top of OAuth, and optionally leverages OpenID Connect as well. In fact, each lower level is usable independently of any higher levels. Here is a (relatively crude) way that this modularity can be understood. Each succeeding letter represents a new aggregate level of functionality. (The lowercase letters on the left represent APIs that, in the general case, are either proprietary or must be standardized by communities of interest.)



If the HEART WG produces profiles according to the plan in our charter, RESTful health data sharing API deployers will have the following profiles available.



How would a deployer choose technologies, and therefore profiles? Following are the specialties of each. (There are edge cases not described here.)

- **OAuth is for app-to-service authorization.** It enables a person to let a client application call an API on his or her behalf securely and with authorized consent, without having to reveal his or her credentials (such as a username and password) to

the application. An app-specific “access token” whose operation may be scoped to some subset of the API’s capabilities is substituted for the credentials instead.

- **OpenID Connect is for portable identity.** It defines a standardized identity API and protects it using OAuth, leveraging the OAuth redirect “consent loop” that users undergo as a single sign-on mechanism between the client application (the relying party) and the API (the identity provider).
- **UMA is for privacy-enabled party-to-party sharing.** It uses OAuth, and optionally OpenID Connect, for three main purposes: 1) enabling a person to authorize other parties to access APIs under his or her control; 2) making it possible for that person to set conditions for access so that those other parties may have to provide “claims” and do step-up authentication to get access; and 3) making it possible to centralize that person’s management of all these conditions for access.
- **FHIR is a web-friendly representation of clinical concepts.** It defines a framework, including a RESTful API and a data format, for health data exchange.

To determine the technologies, and therefore the profiles, you may need for your deployment, ask these questions. NOTE: This is a high-level analysis only. The profile work itself may bring other factors into play.

- In all cases, if you’re reading this document, you’ll probably be interested in “A”.
    - Even non-healthcare-related projects may be interested in “A”, “B”, and “C” for their security properties.
- Is your project using the FHIR API?
    - If yes, you’ll probably be interested in “X” (and “Y”, depending on whether your answers to other questions make you interested in “C”).
- Does your project need single sign-on or federated identity between applications?
    - If yes, you’ll probably be interested in “B”.
- Are you interested in strong client authentication?
    - If yes, you’ll probably be interested in “B” on top of “A”.
- Will your user population always be present at the time of resource access?
    - If yes, you probably won’t need “C”.
    - If your user population must sometimes be absent at the time someone else needs to attempt to access their resources, you probably will need “C”.
- Is it valuable to centralize management of entitlements or scopes granted to apps?
    - If yes, you’ll probably be interested in “C”, even if autonomous party-to-party sharing isn’t involved.