UMA User Experience

- Apple Watch Access Approval Notification
- ForgeRock OpenUMA Health Data Sharing
- SMART Project Implementation
- Simplified CV Scenario
- Setting Policies and Terms
- Real-Time Consent Gathered Out-of-Band
- Access Analytics
- UMA Trusted Claims
- Exploring visualization techniques to enhance Privacy Control UX for UMA
- Python Prototype

One of the biggest issues in user-managed access is the “management” part that users must perform, such as setting sharing preferences (policies), approving access (whether at attempted-access time in some fashion, through some notification scheme introduced by the authorization server, or well after access has been attempted by a requesting party), and introducing resource servers to chosen authorization servers.

This page collects screenshots and wireframes produced by UMA participants, including our User Experience editor, Domenico Catalano. The SMART project in the past has notably conducted user experience (UX) studies that have informed the design of the UMA protocol.

Apple Watch Access Approval Notification

This wireframe from Domenico illustrates how Apple Watch could be used as a client tied to an authorization server for access approvals.

ForgeRock OpenUMA Health Data Sharing

The YouTube video of the May 16, 2015 UMA webinar includes a video demonstration of the current state of the ForgeRock OpenUMA implementation, which demonstrates the experience of a user, Alice, performing scoped sharing resources from two different resource servers (after getting PATs for them with the same authorization server) with two different requesting parties, and one of those parties, Bob, succeeding in getting access to a resource after acquiring an AAT.

SMART Project Implementation

See the SMART FAQ for up-to-date information about the SMART project.

Simplified CV Scenario
This set of wireframes was originally developed for the UMA webinar held on 29 Jan 2010 to map protocol details to user experience realities. The webinar overview slides explain the scenario setup and how often the various interactions are performed (most are one-time-only setup; the rest are optional).

Click the image to see the wireframes. (There are "click-throughs" on the screens to give you a simulated user experience; the instructions on the right-hand side indicate where most of the links are.)

Setting Policies and Terms

This is a single screen that was developed early in the ProtectServe/UMA process to explore policies and negotiation terms that might be interesting to an authorizing user. It is discussed further in the Issues section of the Scenarios and Use Cases document.
Real-Time Consent Gathered Out-of-Band

If some sort of "ask me for consent" option is presented to a user as shown in the above sections, in many cases the consent will need to be gathered in an out-of-band fashion. Mobile devices are popular for such a purpose. This set of wireframes was developed to explore how consent might be gathered from an authorizing user given that a requester has approached a host at a time when the user is not logged in to either of them. (Some reasons for needing real-time consent are discussed in the Requester Identification section of the Scenarios and Use Cases document.)

Click the image to see the wireframes.
Access Analytics

The benefit of being able to forge data-sharing relationships with others would be blunted if the authorizing user didn’t have the opportunity to review the details of resource access after the fact. This set of wireframes was developed to explore the kinds of resource-access analytics a user might want to review, partly to manage sharing at the right level and partly as a tool for assessing privacy risk.

Click the image to see the wireframes.
UMA Trusted Claims

The UMA protocol supports the policy-driven ability of an AM to demand claims from a requesting party before authorization is granted. The claims may be self-asserted or third-party-asserted. This set of wireframes was developed to explore a person-to-person data sharing scenario in which the Authorizing User wants to restrict sharing to a specific Requesting Party identity. For such a policy to be meaningful it requires the AM to establish of trust of third-party identity claim issuers. The Wireframe includes the subject registration and the Claims Host registration as part of bootstrapping trust process.
Exploring visualization techniques to enhance Privacy Control UX for UMA

The study explores visualization techniques to enhance privacy control user experience for UMA protocol, applied to Newcastle University’s SmartAM project. The goal is to mitigate risks of lost of privacy and the exploitation of online personal data caused from user difficulty to maintain data control, correlate web resources and assign privileges for specific scope in the data sharing process. The approach (see the following presentations) introduces the Connection and Control bridge concepts and visualization tools for this purpose.

UMA/SmartAM UX Study (Part 1) (SmartAM UX study introduction, requirements, privacy control concepts)

UMA/SmartAM UX Study (Part 2) (UMA Connection concept/UX)

UMA/SmartAM UX Study (Part 3) (Control bridge concept/UX)

Python Prototype

Christian Scholz’s Python prototype (now out of date) has generated these screens:

- Getting started at a host
- AM interface
- Forcing a (human) requesting party to self-assert a label for themselves