Unicon-assisted Proof of Concept re UMA for Privacy-Preserving Accessibility Support

The Scalable Privacy Project has contracted with Unicon for work in support of the following objectives:

The User-Managed Access (UMA) protocol is a profile of OAuth 2. The Scalable Privacy Project, one of the first-round NSTIC pilots, is seeking consulting help in developing a proof-of-concept implementation of privacy-preserving UMA-based accessibility support. The context for this work is the Global Public Inclusive Infrastructure (GPII.net) initiative.

1) Accessibility needs and prefs doc for several users (will be provided) conforming to ISO/IEC 24751 and AccessForAll standards to be published at UMA-protected RESTful-style URLs on an UMA-compliant Resource Server (RS).
 - Unicon to build and configure the RS leveraging SMART's Python for UMA (PUMA) library per instructions on SMART blog
 - http://smartjisc.wordpress.com/2012/04/13/releasing-puma/
 - In GPII architecture terms (http://wiki.gpii.net/index.php/A_Detailed_Tour_of_the_Cloud4all_Architecture), this RS "host" is meant to be a cloud instance of the Preferences Server.

2) There will be a GLUU-provided UMA Authorization Server (AS) configured to protect the per-user needs & pref docs on the RS; Initially supporting only one OAuth/UMA scope: "read". Time permitting, we would add an "admin" scope for user self-service create, update and delete operations on the needs & pref docs.
 - AS is currently running at Brown U; Unicon will be provided all needed access.

3) Simple UMA client/relying party (RP) app that allows access to user's own needs & pref doc (person-to-self sharing model).
 - Unicon to write RP client web app leveraging the PUMA library per instructions on SMART blog. Client to interact with RS and AS as specified above.
 - In GPII architecture (http://wiki.gpii.net/index.php/A_Detailed_Tour_of_the_Cloud4all_Architecture), this RP client is meant to be a cloud instance of the Flow Manager
 - Coding will need to include implementation of an abstract PUMA interface for a simple set of storage/persistence operations
 - RP Client to be delivered with an integrated set of test modules and logging capabilities.

4) Since the ultimate hope is that this work will be adopted by the GPII development team as a security layer for the Flow Manager and the Preference Server, and since GPII's chosen development platform is Node.js, the Unicon python code should be fully annotated to facilitate the GPII developers' task of migrating from Python to Node.js.