DRAFT: THE IMS-GLC

COMMUNITY APP STORE ARCHITECTURE (CASA) INITIATIVE

THE CONCEPT: ADDING AN APP STORE ON TOP OF THE BROWSER

The World Wide Web is a vast, mildly curated repository of information. While search engines fairly accurately filter the Internet based on content, they are less effective at filtering based on functionality. For example, they lack options to identify mobile-capable sites, sites that provide certain interoperability mechanisms, or sites related to certain industries or with certain content rating levels. There is a space where such a model already exists: the "app stores" that pervade the native mobile app landscape. In addition to the app itself, these hubs have deep awareness of application metadata, such as mobile and/or tablet support. Another deficit of search engines is their inability to allow organization-based configuration, defining a worldview with trust relationships, filters and transformations to curate the results they present to end users. Native app stores use a star (hub-and-spoke) topology with a central hub for publishing, which lacks this fine-grain customizability, but an alternative peer-to-peer topology, as is used for autonomous systems across the Internet, restores this freedom.

Based on these considerations, the Community App Store Architecture (CASA) provides a model for publishing and sharing web applications with metadata about features and functions; further, it includes mechanisms for filtering and transforming this data during the publishing and sharing processes. This enables the construction of affinity-focused environments that may connect with peers to which it may share apps and publishers from which it may receive apps. Use cases for CASA include an LTI Tools store, a K-12 website or an organization's mobile dashboard, such as http://m.ucla.edu.

NETWORK TOPOLOGY

Traditional app stores employ a star topology whereby publishers add apps to a central hub and the hub makes these apps available to end-users. This approach requires a homogeneous worldview, a centralized trust model and an agreeable aggregator, none of which have been shown to exist on the World Wide Web. The alternative is a distributed model, the very model of the Internet architecture itself. In the same way that autonomous systems in the Border Gateway Protocol share routes they're willing to serve and select routes they're willing to use, CASA presents a model for accepting and sharing applications in a consistent way based on their metadata; further, as the peer-to-peer model is a superset of hub-and-spoke and complete configurations, it supports sub-graph composition best described as a small-world network. When surveying organizations about how they manually ascertain and curate environments such as the ones CASA seeks to enable, it is this small-world topology that most adequately models their publishing and sharing relationships. From an efficiency standpoint, where a star topology requires a central aggregator which can support connections, and a complete topology requires ——— total connections, this approach has reduced performance footprint based on the number of relationships maintained by every peer in the network, rather than the total size of the network.

LEARNING TOOL INTEROPERABILITY

In the last several years, IMS-GLC has made progress with tool interoperability, by developing the LTI specification, which was intended to allow instructors to easily and seamlessly leverage third party tools in their LMS's. To create this integrative layer of code, The IMS community created the Learning Tool Interoperability (LTI 1.0) standard. This allowed the LMS system to pass authentication and authorization from the LMS to the Tool.

HOW DO I FIND, COMPARE AND EVALUATE LTI TOOLS?

LTI enables a vision of many niche products being pluggable into the LMS; however, without a storefront, the acts of publishing and locating these tools are key problems. The IMS-GLC website provides a list of LTI-compatible tools, but, just as a search engine provides generalized results, this list is also rather broad. There is no easy way for an LMS manger to research LTI math tools to see which ones are most popular, to view review other users' comments, and to generally peruse software listings the way they do with native app stores. IMS-GLC and its constituents should thus present a solution that enables LMS managers to parse through LTI tools, searching via a tool's attributes and viewing detailed metadata and product views before making a selection. The Community App Store Architecture (CASA), with its ability to share and propagate apps with their metadata, satisfied this need such that a network of browser-based app stores may peer together to cultivate organization-based or affinity-based app stores of that may contain LTI and other types of tools.

THE COMPONENTS OF A COMMUNITY APP STORE ARCHITECTURE (CASA)

The Community App Store Architecture (CASA) enables the sharing of web apps and associated metadata ("mi casa es su casa") through the URI Sharing Environment (USE) Protocol, as described by https://github.com/uShare/protocol.

CASA is comprised of three modules which represent an implementation of the USE Protocol. The first module is the CASA Engine, which queries other CASA Engine publishers that share modules with peers based on its organization-specific configuration. The second module is the CASA Engine Manager which provides an administrative user interface for the management of the engine's organization-specific configuration, including peering relationships, filters and transformations; and allows the entity to set and manage their preferences and filters for content. The third module is the CASA Storefront, which provides an "app store" user interface for easy user access to apps received by or published into the CASA Engine. The initial CASA storefront will be developed for Moodle, but given differences in consumer environments, it is expected that eventually numerous versions of the CASA Storefront will exist.

1) The CASA Engine

The CASA Engine has two principal jobs: The first is that of App discovery and the second is that of App publishing. App discovery content filters will be configured on a per engine basis, while app publishing profiles will be created on a per App basis.

1.1 CASA Engine Job 1: Discovery

This process entails querying other CASA Engines to receive information regarding apps of which these peers have knowledge. This process is supplemented by: (1) a filter layer to drop apps that do not meet organizational requirements, and (2) a transformation layer to curate apps that the Engine has received. These enable each CASA Engine to apply a lens based on preferences of the organization. For example, one might implement an accessibility filter whereby the Engine only accepts apps with ADA and Section 508 compliance.

1.2 CASA Engine Job 2: App Publishing:

This process entails allowing other CASA Engines to query the CASA Engine, providing a configurable mechanism for publishing a catalog of web applications and metadata to peers in the network. The CASA Engine publishes apps as individual profiles, and thus each app is made available based on creator's sharing preferences. Each app may select between "local", "shared", and "shared with propagation".

2) The CASA Engine Manager Module

The CASA Engine Manager module will be a web-based interface and generally set up by the LMS technical manager. It will allow LMS administrative users to configure the CASA Engine, including defining filters and transformations, configuring peers and managing app publishing profiles.

3) The CASA Storefront Module

The CASA Storefront Module will be a web-based interface that allows the LMS users to select apps from searchable and orderable properties such as categories, tags, ratings and capabilities, as defined by individual app profiles. CASA Storefront modules will likely be implemented for Learning Management Systems, as well as other consumers such as a web portal mobile dashboard, that wish to leverage the app profiles stored within the CASA Engine.

EXAMPLES OF CROWD SOURCED ACADEMIC RESOURCES

One example of a crowd sourced academic resources is a professor that keeps a robust and ever evolving organic chemistry glossary. This glossary is something that students help with, add to, and even get extra credit in the professor's course if they contribute to it enough. This mobilized tool is useful to students doing a chemistry lab, and saves them from having to open up a laptop next to their experiments. The professor is willing to share this glossary freely, however, today, without a network of CASA Engines; there is no easy way to find it.

Another example is a school of business that wants to make its homegrown case studies into MBA-focused student simulation games. It makes sense that they would be most interested to share and publish them for simulation exercises with other business schools. This would likely be reciprocal if several other schools were in turn willing to share their case-study simulations with the business school community.

Finally, The CASA Engine will provide universities and other entities to publish in new models within a trusted network web-based apps (with functionality and content intertwined) independently. Examples of apps that might employ a variety of publishing models could include a journal of Egyptology, where subscribers might get a notification when a new article is published. A writer might decide to publish an ePUB3 novel in chapter iterations. If the readers didn't like a chapter, the writer could re-do it, allowing the public to crowd-source the editing process.

PROVIDING CROWD SOURCED RANKINGS

The CASA Engine will also satisfy the need for the LMS administrators to get their peer's perspective on how using certain tools worked within their LMS environment. There is currently no place for faculty or instructional technologists to see user's opinions and ratings, read narratives of what they did with it, or how their students liked using it etc… . The CASA Engine will provide a set of 'most useful' ratings for both the CASA Manager and the CASA Storefront modules.