RIVER ID TECHNICAL OVERVIEW

River ID will be a distributed reputation and trust server with a RESTful API for sending and receiving data. While the technical details will be left to the lead developer (Soe) to decide, here is a rough idea of what we're looking for:

TERMINOLOGY

RiverID - server and database for all types of information related to user trust
TrustDB - essentially a 'locker' for all the content a user trusts, as an individual
RiverID Score - users accumulated score
Source - the point of origin for content (email address, URL, Twitter Username, Phone Number etc.)
Instance - a deployed copy of Swift running on a users server

REFERENCES

- Open ID Connect http://openidconnect.com/ -Jon Gosier 5/24/10 2:53 PM
- Facebook Connect
- Twitter @anywhere http://dev.twitter.com/anywhere/begin -Jon Gosier 5/24/10 2:53 PM
- Google Friend Connect

OVERVIEW

- Allow Swift users to take their score with them across SwiftRiver instances
    - Global score (across all SwiftUsers)
    - Local score (across on instance)
    - Local score is algorithmically weighted against the global
    - Store transaction interactions between content, instance and User
- Instance ID
    - each instance of Swift needs to accumulate a score based on user interaction within that community
    - the score should be inferred, not stated. (e.x. no user should be able to do any one thing that might 'game' the algorithm)
- Build upon Oauth and Open ID Standards
- TrustDB this will allow third-party applications to hook our RiverID servers -Jon Gosier 5/24/10 2:42 PM
    - Trust DB stores each users 'trusted' sources. Swift instances can then hook that data via the API
    - "Trust" Button. Allows a user to trust specific sources across the web. Enabled via javascript. similar to the Facebook 'Like' button -Jon Gosier 5/24/10 2:44 PM
    - Also allows users to 'claim' a source as theirs. (e.x. This is my phone number, this is my email address, my blog etc.) XFN? -Jon Gosier 5/24/10 2:44 PM
- Persistent
    - All scores should always be in flux
        - If user interactions define all scores, they have to be
- Login Information stored differently than TrustDB info
    - Login platform should simply be some combination of existing Open ID/Oauth platforms
    - TrustDB should be paired with user IDs with tokens
    - The two DBs should remain separate, layer of abstraction for distributing excessive requests
- Build to scale Soe, Decide if SQL is better than NoSQL here -Jon Gosier 5/24/10 2:45 PM

SWIFT RIVER INTEGRATION

Swift isn't at quite ready for RiverID integration yet. For instance, there's no user roles to even pull or push data to. However, because we imagine RiverID being RESTful anyway, please just develop it with a sophisticated API that we can hook later.

Don't worry about interacting with SwiftRiver or Swift Web Service APIs. All this will happen in the core app as we integrate all the APIs together.