

The goals of the FST Hnet system are as follows:

- Simple architectural structure
 - . Make future extensions easy
- Straightforward implementation
 - . Minimal amount of integration
 - . Keep external costs low
 - . Utilize COTS as appropriate
- Get something working soon
 - . Base platform
 - . Add features opportunistically

System Components

To keep this simple, the following components are required (recommendations in []s):

- . Network colocation
 - [InReach <http://www.inreach.com/> downtown Oakland, ~\$350/mo]
 - This is 1/4 rack ("11 U" = about 19").
 - Can hold several computers
 - Could consolidate existing IT infrastructure (web site, email?)
 - Smallest reasonable footprint available in the Bay Area
 - Good support, central location
- . Server computer
 - [Sun Sparc Netra/X1, 1U rack mount, "small" configuration, \$995 <http://store.sun.com/catalog/doc/BrowsePage.jhtml?cid=60268>]
 - Some more memory (\$100?)
 - Some more disk (\$400?)
 - This will fit into a rack, take minimal space, and will provide everything needed for this first system, demos, etc. We will configure the hardware, install software, security-prep and arrange for lights-out system administration (regular backups, etc.)
- . 3rd Party Software
 - Development Environment
 - [Java JDK (j2ee), \$0]
 - All required tools are freely available on the Internet, will be installed and configured for best performance and capability
 - Web Server
 - [Zeus v4, <http://www.zeus.com/>, ~\$2500?]
 - This is the most expensive component, but we've found no other web server that's worth it like Zeus is. We will install and configure for best performance and capability. A certificate will be required (Verisign, \$295/yr) for secure operation, but it's optional for now.
 - Database
 - [Sybase v11.9.3, \$1000]
 - Rock-solid, simple, high-performance and cheap alternative to Oracle boondoggle. We will install and configure for best performance and capability, plus lights-out system administration, backups, etc.

Application Components

These are chosen based on past experience with similar projects.

- . HNet Application Server ("HAS")
 - This is the bulk of the work, described in more detail below.
- . HAS <---> Database linkage
 - We'll provide linkage between the HAS and a "standard" database using JDBC; in theory any database should work, but in practice we've found that Sybase works best. We don't use any Sybase-specific features, to make a database change at a later time feasible. The database will be the storage mechanism for all of the data that the system collects and manages. We tend not to rely on database "programming" per se, but rather simply use it as a place to store data persistently.
- . Web Server <---> HAS linkage
 - There's some mechanism involved with getting a web hit directed to the HAS (as opposed to simply delivering static content) that has to be built, but Zeus makes this a straightforward engineering effort. Maintenance of sessions, hooks for authentication and authorization, and mapping from URLs to application functionality are provided.
- . Static HTML
 - The "output phase" of the HAS is templatized to allow a clean separation between form and function: the LFA (Look, Feel, Attitude) can be modified independently of the running system, in particular by non-Thinkbank team members. This has many plusses including separation of the work effort, but also we've found that:
 - . Changes to the LFA can be made without restarting the system
 - . Multiple LFA instances (such as "current" and "new") can co-exist
 - . Development of new modules can leverage existing templates
- . Miscellaneous
 - Some of the functionality we'll provide will reside outside the HAS, so that they can be developed independently without a dependency on the HAS itself. Examples include administrative "housekeeping" duties and the "robotic" features we talked about the other day (phony participants in the transaction system).

HNet App Server Components

- . Authenticated Participant Manager
 - All participants (identified by a name, authenticated by a password) have roles (what they can read, what they can write, what they can do) and personal attributes (email contact, address information).
- . Data Object Manager
 - All data objects in the system have attributes, such as the owner of the data, when it was last modified, and any of the specific items such as an amount for a purchase or a location for a telemetry upload.
- . Transaction Manager
 - Some mechanism is required to make sure that transactions happen in the right order and can be verified, etc. All transactions generate historical logs for auditing purposes.
- . Report Manager
 - Flexible report generation, including Excel export/download

Specific Functional Units

- . Session
 - Login
 - Logout
 - Session Timeout
- . Participant Management
 - Create
 - Modify
 - View
 - Change Password
 - Recover Password
- . My Hydrogen
 - Orders
 - . View (old/current) and track
 - . Create (email confirm)
 - . Modify
 - Reminders
 - Usage / Statistics
 - . ?trends?
 - . ?charts?
 - . ?forecasts?
 - Account Management
 - . Change of address (shipping/billing)
 - . Password
- . Customer Service
 - Catalog management
 - . Prices
 - . Inventory
 - . Descriptions, etc.
 - Customer order review/revise
 - Reports
 - . Sales data (by period, by item)
 - . Open Orders
 - . Usage (logins, changes, buyers, lurkers)
 - . New customers
 - . ?analytics?
- . Inventory Management
 - On hand
 - Orders
 - Ordered -> On hand
 - Reports

