**FHIRBlocks Technical Architecture**

**October 2, 2017**

***Working Draft***

**Forward**

I'll start with answering some nagging questions, and will answer strictly from the perspective of the POC (NOT painting the long-term view here, but providing us something thats actionable in the short term.

First, I'll introduce a new code name: Shiraz. Shiraz is the software development project undertaken by HARBINGER, Bharat's name for the Indian team. Shiraz is a "Client System 104" as described in the patent. Peeling apart Shiraz, there's a few key components:

- A user interface tier, that the user uses to interact with Shiraz, and by extension, FHIRBlocks itself.

- A crypto processing component, that creates and stores keys

- A logic layer, that implements the functionality of Shiraz

- An API implementation layer, more particularly, an implementation of the client-side of the SMOAC protocol. This component should be provided as part of the OpenSource effort, and is our iOS "Software Developer Kit" (SDK). This SDK is to be maintained in a separate/discrete GitHub repo, and be separately buildable with no dependencies on the rest of the Shiraz code base. We do this so that we can keep Shiraz itself to be proprietary, yet provide a SDK to all other iOS developers using the Apache license.

Shiraz will be an IoS application, available through Apple's developer test network-- whatever the old TestFlight product has morphed into (I think Apple itself may have acquired it). For those of you not in the know, TestFlight was a platform whereby one could create a new IOS application and deploy it to a handful of phones, for testing purposes, and without going through the Apple application approval process.

So with that as the backdrop, on to some of the more interesting areas to probe:

* **A Digital Wallet (interface to FHIRBlocksChain)** - Yes, Shiraz provides this capability. Shiraz creates and stores keys, it also provides connectivity from the device (phone) to FHIRBlocksChain, via the SMOAC protocol.
* **Registration & Identity (CSI) System** - When the user first downloads Shiraz and installs it on their phone, there is a one-time initialization process. In general:
  + User demographic information is collected from the user. The details of what comprises this information set is contained in SMOAC.
  + Shiraz requests a Globally Unique Identifier(GUID) from FB via SMOAC. A GUID is simply a psuedo-random number that's never been used before.
  + Shiraz calculates a key pair
  + Shiraz then sends a "request for access 204"
  + Shiraz then sends demographic info 210
  + Shiraz receives "set up complete" 216.
* **Applications’ Subscription System (App Store Module)** - During the POC, Shiraz is provided to authorized testers by Apple's DevKit capability (see above).
* **Applications’ Permissions Establishment System** - The UX and Logic tier (see above) of Shiraz allows the users to establish/set permissions. Shiraz then "saves" those permissions into FB by using the SMOAC protocol.
* **Applications Subscription & Permissions/Conditions Review System -** Shiraz provides this capability. Shiraz may "fetch" current permissions out of FB by using SMOAC. Shiraz may then display permissions to the user, and allow the user to modify those permissions. Modified permissions are rewritten to FB using SMOAC
* **Applications Launch System** - Shiraz will be "launched" from the IOS phone top. Just touch the icon, and off it goes.

**Architecture Overview**

FHIRBlocks is a permission management system that allows authorized actors (typically patients) to provide "access control lists" to their health information contained within a variety of electronic health record systems. One unique aspect is FHIRBlocks decentralized architecture, based on BlockChain (more particularly HyperLedger)

At the most fundamental level, FHIRBLOCKS comprises several actors:

**Client Systems** - Generally speaking, these are the applications that consume health information and provide some benefit to a use.

**Resource Servers** - These servers provide health information (contained within "electronic health records) to Client Systems. Resource Servers ONLY provide this information subject to a set of AuthZ/AuthN Principles, based on OAuth contained within the request

**Identity Management** Services - These services create Principles based on requests from Client Systems and permissions previously placed into the FHIRBlocks chain.

**Appendix A**

**SMOAC API**

**/util/ping - (GET) -** Used to determine if a FB node is up and running, returns versioning information. Available in Swagger.

**/util/getNodeEndPoints (GET)** - Returns a list of IP endpoints for all known nodes in the FB system. Available in Swagger.

**/get/getGloballyUniqueIdentifier (GET)** - Provides a globally unique identifier, and used as part of CSI construction Available in Swagger

**getRequiredDemographicMetaData (GET**) - Returns a list of all demographic data required for identity management to function.

**requestCsiRegistration (PUT**) - Used by an application to register its CSI with FB.

**getPermissions (GET)** - Get the permissions that have been granted BY a given CSI.

**getMyPermissions (GET)** - Get the permissions that have been granted TO a given CSI

**publishProof (PUT)** - Publishes a ZK proof to the chain. Includes both the prover and verifier components.

**proveProof (PUT)** - publishes the results of the prover side of a ZK package.

**validateProof (GET) -** executes the validator side of a ZK proof.

**publishPublicFact (PUT)** - Publishes a publically discoverable (and as yet unconfirmed) fact

**signFact(PUT)** - Signs a public fact, attesting to it's validity.

**getFactSignatures (GET)** - returns the set of signatures on a public fact.

**Appendix B**

**Unknowns**

1. How does a CSI and MRN become associated (thus we know which identity maps to which patient record)
2. How do new IDNs join FB?
3. Is there a base process to add a new application (NOT APPLICATION INSTANCE ON A USER DEVICE) to FB?
4. Can Oauth be configured to use the CSI as the basis for uniqueness (think so!, just have to make sure).
5. 5. Just what is the mapping function between OAUTH and FB? What's missing from the HEART spec (as a starting point
6. How does a IDN specify the data set required in order to associate a FB identity to an MRN
7. How is identity info conveyed to IDN, so the CSI-MRN mapping can be provided
8. what standard mechanism exists to encode a CSI into the EHR
9. what is the process by which a FHR resource server will implement the logic to enforce permissioned access
10. should there be an "override" permission set by the IDN to disallow permissioned access
11. how are new IDN's provisioned in FB
12. How are IDN's deprovisioned in FB