Project: Harbinger

Design Proposal

By Kevin S. Scardina

Table of Contents.

Overview

The Harbinger Project is the next phase of the previous Novum Concepts Prototype for sending secured messages from a field to a base.

This document tries to explain “How” the messages are sent and “How” the devices are identified as well as specify what technologies used in the project.

The project is treed out as follows:

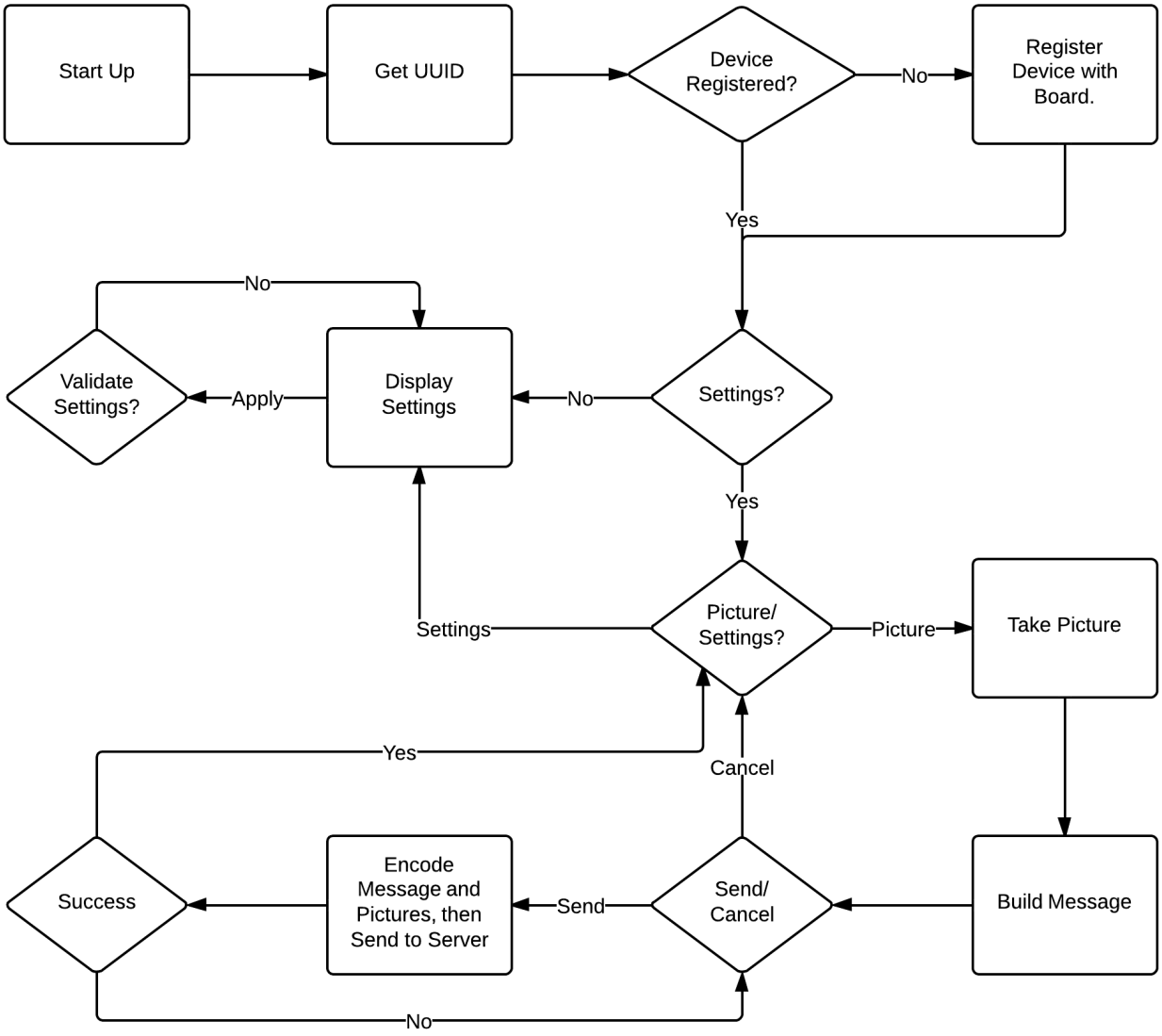
- Harbinger
  - Clients
    - Field
    - Base
  - Server
    - Board

So the project is defined as a Client-Server application with one server named Board, and two types of clients, one used by field personal named Field, and one stationary at a base named Base.

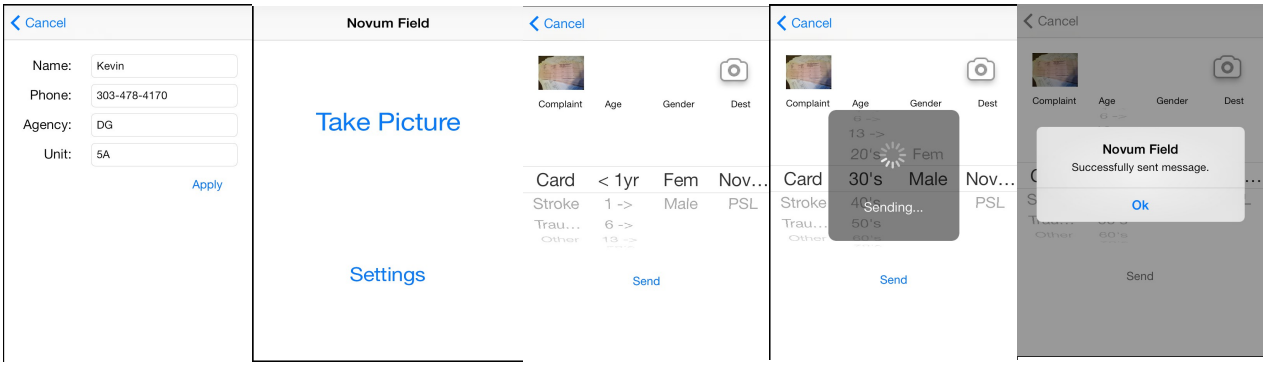
In this document I will specify the flow of the client applications and specify the technologies used with the different clients. I will then describe the server modal, and technologies used to create the server. Security will be defined and I go over how transactions are secured though SSL and how the data is encoded by clients for specific bases using an asynchronous key encoding, consistent with how secure email works (CMS).

Field

Novum Field Clients are used by field personal and as such do not contain information or do any reading of encoded information of the Board. Field only sends encoded data to the server and then deletes its data. Below is the flow of how a field client works.

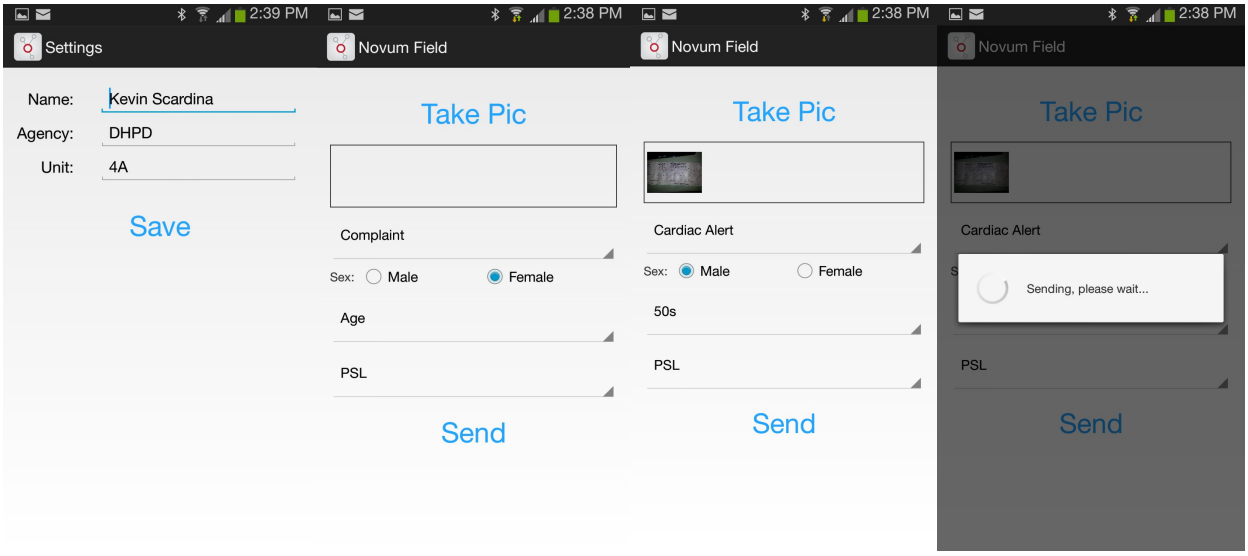


IPhone



The iPhone client is a native client written in Objective-C in Xcode. Currently no extra add-on libraries and only native frameworks are being used. Harbinger Field iPhone only requires access to the network, the camera, and the security chain. All data requests to the Board are done with NSHTTPRequests over HTTPS.

Android

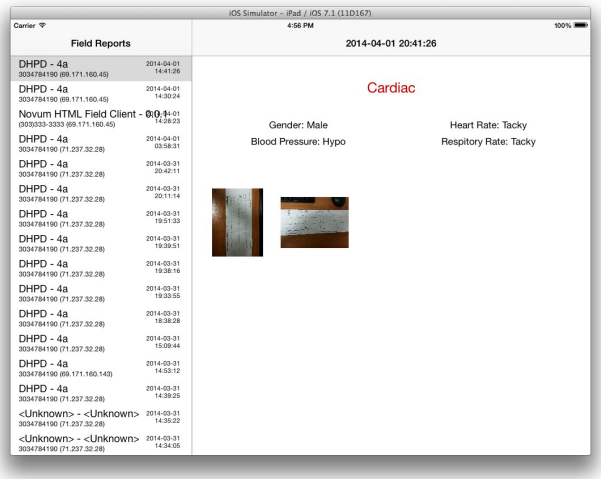


The Android client is a native Android JDK Application in Android Studio. Currently the Apache mime and HTTP pachages. All data requests to the Board are done with org.apache.http objects over HTTPS.

Base

Describe base. Add Base Flow.

iPad



The iPad base client is writen in native Objective-C with only native frameworks. Harbinger Base iPad only requires access to the network and the security chain. All data requests to the Board are done with NSHTTPRequests over HTTPS.

Possible Modules

We had discussed possible modules so I have left this here so that we I could remember to design with this in mind.

Currently I see modules as relays available from a base. All bases will have the ability to relay from one base to another base, so if the ER/main base gets a message that they want to relay to a module they can.

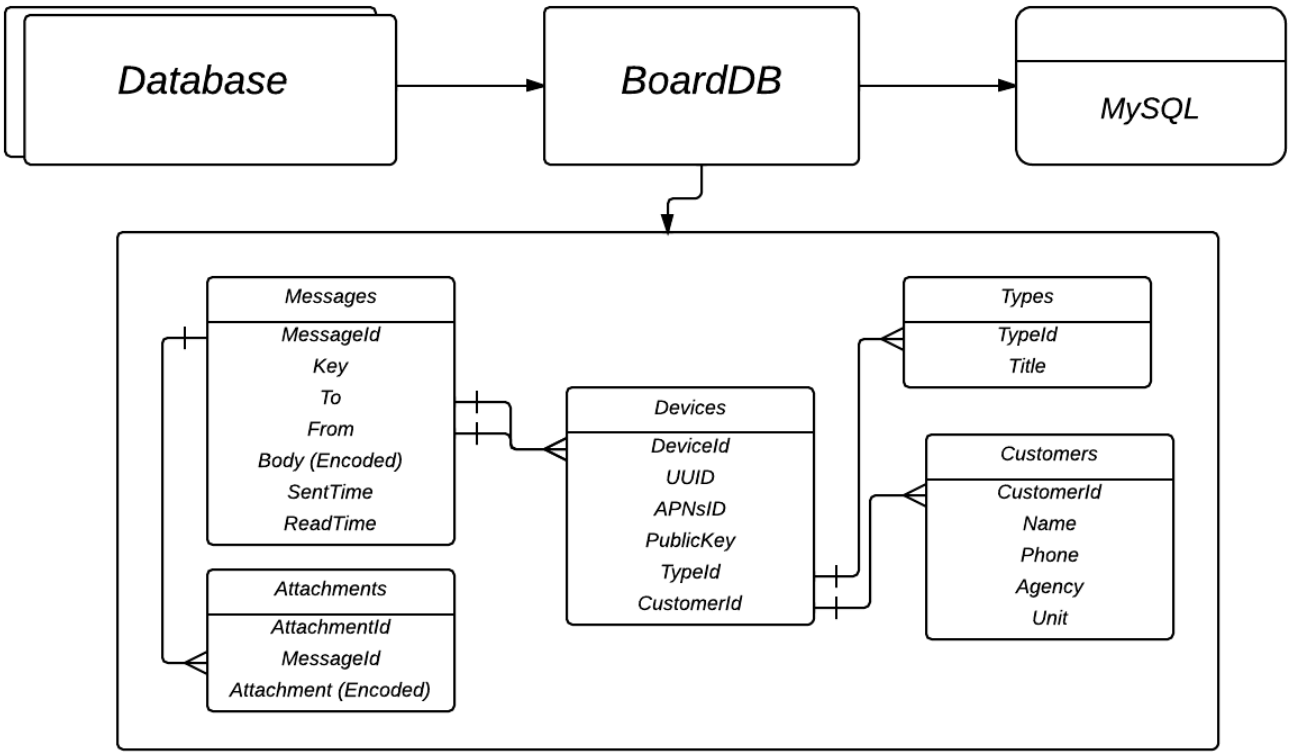
Board

Board is the “backend”, modal, database portion of the Harbinger Project. It consistests of a MySql Database, Java EE 7 (JAX-RS 2.0, JSON-P 1.0, and EJBs), along with the current JDBC using the current Oracle MySQL Java Connector. Board will be a Java RESTful Web Service, running behind the latest stable version of Glassfish Application Server.

RESTful Server

- /device/{UUID}
  - GET: {DeviceId, UUID, APNsID, PublicKey, TypeId, CustomerId}
  - queries Devices and returns the Device with UUID
- /device/{DeviceId}/{UUID}/{APNsID}/{PublicKey}/{TypeId}/{CustomerId}/
  - POST
  - updates Devices sets values where DeviceId and UUID are indexes.
- /device/{UUID}/{APNsID}/{PublicKey}/{TypeId}/{CustomerId}/
  - PUT
  - Creates Device.
- /device/{DeviceId}/{UUID}
  - DELETE
- …

The Database

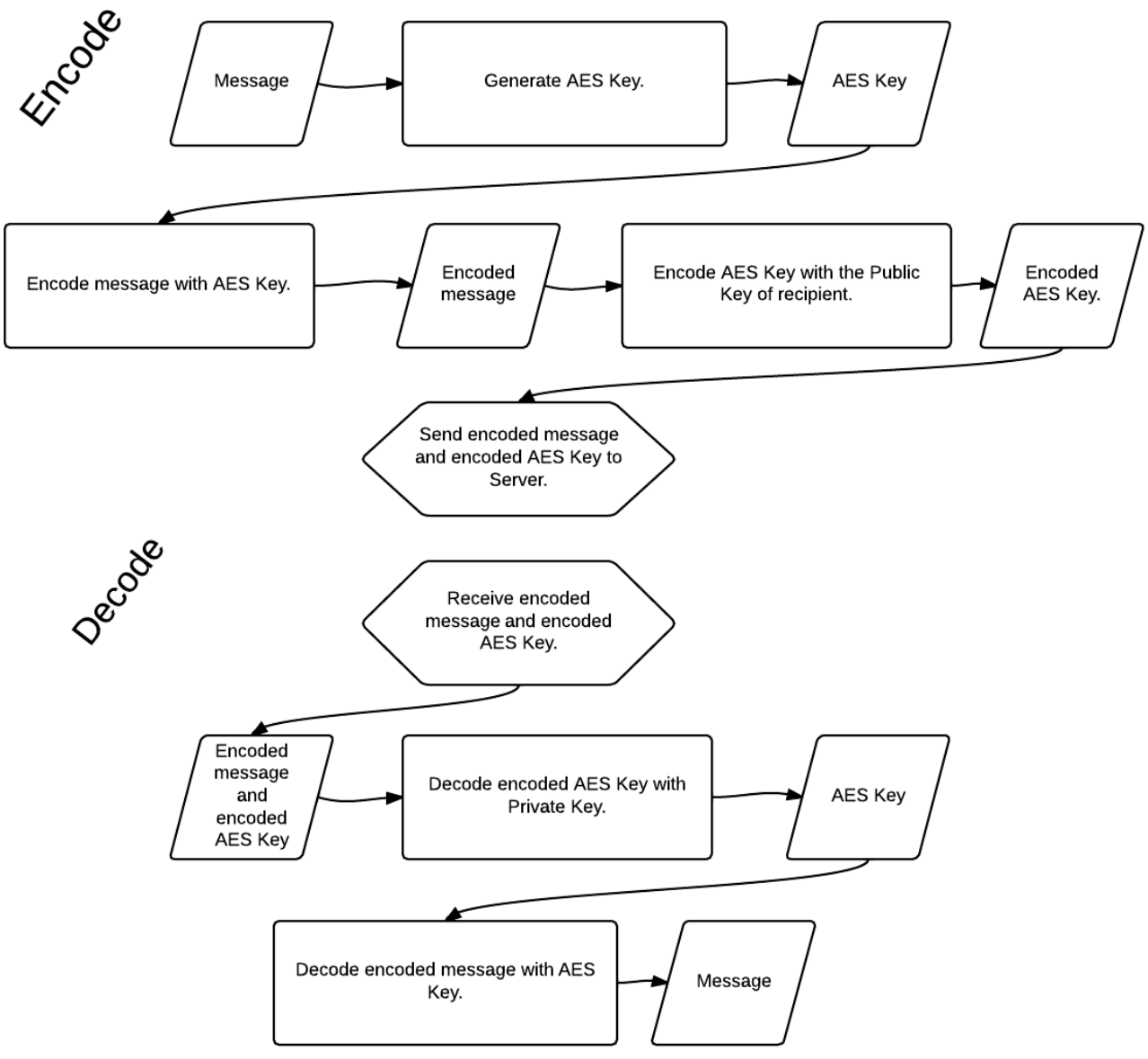


Security

All HTTP requests are sent over SSL and are handled by Glassfish and the client libraries. As long as these requests are sent on the secure socket layer, their data is secure, and transports are all secure. The data storage is also secured by client side encoding and all PHI is encoded with an AES key that is encoded with the PublicKey of the intended recipient, locking the message so only the intended recipient receives the PHI.

Client Encoding of possible HIPAA confidential information

Client Encoding is a process that makes it so at no point from point-A to point-B can the data be retrieved by anyone but the intended client. Here is how it works:



HIPAA Compliant Host

We need a HIPAA compliant host that can host GlassFish and Java EE 7. It seems like everything we look at so far will do it. We will also need SSL Certificates. I believe we also want the host to provide the mysql database and provide HIPAA on that as well.