Non-admin Users (NAU) Implementation

Problem Statement

Each HDA has admin users (AUs) and non-admin users (NAUs). AUs have full access to all the data and settings in the system. NAUs have a subset of the access rights to the system, as defined by the HDA AUs.

Users in Amahi.org are different from users in the HDAs. Users in the HDAs will generally have different credentials as Amahi.org users.

The problem is to allow access as easily as possible to AUs and NAUs to the data in their HDA from mobile devices using the Amahi Anywhere server running in HDAs.

Current Setup

Each HDA-User has a pin in the HDA. On logging into the client app with amahi.org credentials, admin users see the list of HDA's available to them. On selecting any HDA, they are asked to enter the pin for accessing that HDA. This supplies the client app with an auth token, which has to sent in every further request to HDA until the user logs out (at which point both the admin and NAU are logged out).

The problem is, to authenticate a NAU against a HDA, the HDA first needs to be identified.

There are several options which can thought of to handle this:

- OPTION A (Unique ID)
    - In this each HDA will have a unique (say 6 digit alphanumeric) identifier.
    - The unique identifier of the HDA will be displayed on platform dashboard.
    - The client apps will have a separate login screen for NAU login.
    - This screen will ask the user for the HDA's unique identifier and their pin.

  Pros:
    - ★ No need to separately implement friending feature as it will be auto implemented with it.
    - ★ Works on both local as well as remote.
    - ★ From usability standpoint, it will be a easier for NAUs as well as AUs to login. AUs will simply need to share the HDAs Unique Identifier with other NAUs.

  Cons
    - ★ Hard to create unique identifiers for already existing HDA's.
- OPTION B (Probing)
    - To identify a HDA to connect to, the client app will ping each available ip in the subnet and try to detect any HDAs in the local network.
    - For detecting a HDA, the client app will make a simple GET request to every ip which replied to the ping. The HDA (if present) is supposed to respond with a few details like its name etc.
    - NAU login screen will then display a list of all the detected HDAs.
    - The user can then login by selecting a HDA from the list and providing his/her pin.

Pros:

- ★ Not much effort in the backend side, so will be easier to launch.
- ★ From usability standpoint, it will be the easiest for the NAUs, as the user doesn't have to remember any unique identifier or ask another admin to login.

Cons

- ★ Only available in Local mode.
- ★ Cannot be extended over remote mode in future.
- ★ Probing is not an efficient solution as a complex network may be spread across different subnets.

- OPTION C (Remembering)
  - The client app will remember the details of the previously connected HDAs on that network. On NAU login screen it will display those remembered HDAs to connect to.

Pros:

- ★ No effort in the backend side, so will be easier to launch.

Cons

- ★ Requires an admin login at least once.
- ★ Cannot connect to an hda which has not been discovered yet.