Server Functionality

Version 1.1 July 3, 2013

The cloud based server stores the content, sends content to the displays, allows users to control the content streams being delivered and provides the ability to upload and edit the content store.

Terms:

Client: roku, Samsung [ps3,xbox] device which connects to internet and display device

SmartDevice: an android, IOS device with which the user interacts with the server

User: a person with an account

Administrator: a member of the Zwamy team who can perform functions on the server

Server – cloud based server infrastructure which serves up images for display

Viewlist – list of images which will be displayed

Release Versions:

MVP – minimal valuable product – designed to be given to at most 100 “friends”

MVP + 1 – first system which real users can use – support up to 10,000 users

There is a many to many relationship between clients, users and SmartDevice. A user can be associated with multiple SmartDevices and clients e.g. a person has multiple TVs and both a iPad and iPhone and wants to be able to control the TV they are looking at with whichever SmartDevice is near. A client can be associated with multiple users e.g. both my wife and I are users and either of us can control the TV in the room which we are in.

Basic MVP functionality

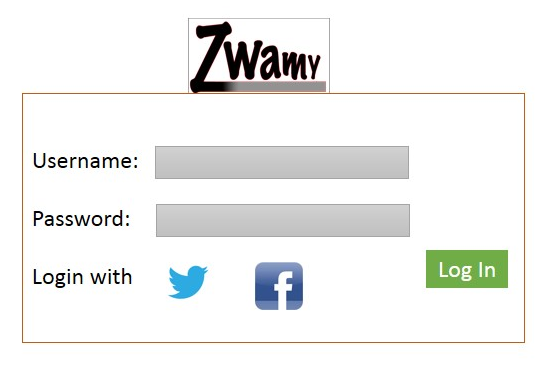
1. Content Store – the server can store at least 1000’s of images. Each image has a set of meta data fields associated with it e.g. artist name, work name, date, etc. These meta data fields need to be extensible (so we can add fields in the future) and support multiple types e.g. text, list, yes/no. Images must not be loaded into the content store; we only store a thumbnail and a URL for the full size image.
2. Content Store MVP data fields (see Zwamy data structures document)
3. Content Store Administration: images are entered into the system manually. (no user interface)
4. User administration – Users are entered into the system manually. (no user interface)
5. Client administration – only Roku clients are supported and possibly Samsung smart TVs. Manually entered into the system. (no user interface)
6. User client association – for each user the system has a list of clients which they are authorized to use. A client can be associated for multiple users.
7. Viewlists – an object which stores pointers to a set of images. In addition each viewlist has a unique per user name, owner and a set of extensible meta-data fields (like images) e.g. viewlist description, date created etc.
8. Viewlist MVP data fields: (see Zwamy data structures document)
9. Viewlists – the server stores a set of viewlists. There are both public and private viewlists. In MVP we only support public viewlists. These public viewlists are visible to all users. They are manually created by the administrator in MVP. The user can select which viewlist they are viewing as described in 11a.
10. Logging in – when a user first starts Zwamy on their SmartDevice (or through the web interface to the Server) they are presented with a login screen which allows them to authenticate themselves to Zwamy. In MVP we will support Zwamy only user registration, authentication through Facebook and Google as shown in Figure 1

Figure 1

1. Client activity –Each client has an active viewlist. When a viewlist is chosen on the SmartDevice, the first image is sent to the associated client. On the SmartDevice the user sees what is shown in Figure 2 below. This allows the user to control what is sent to the client. It shows the image being displayed on the client, selected metadata [Title, Artist, Date Created) associated with the image, and allows swiping left or right which allows movement through the viewlist. This web interface is designed to display reasonably on a web browser on the Nexus 7 tablet with 1280x800 resolution. When the end of the viewlist is reached, it starts again at the beginning. 

Figure 2

1. Options There are some options which can invoked by touching the gray box in the upper right corner as shown in figure 3



Figure 3

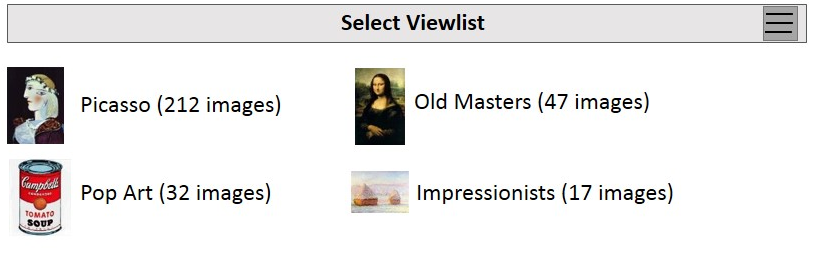
* 1. Select Viewlist allows the user to select a different viewlist for display on their player (Roku, Samsung SmartTV) as shown in Figure 4. This shows all available viewlists.

Figure 4

* 1. Select Player allows the user to choose which player they want to control with their SmartDevice. If the user has only a single player associated with their user account then this option is greyed out, otherwise a list of players is shown (figure 5). In MVP+1 the circle next to each player indicates its status. If the circle is green then the Zwamy software is running on the player and is in contact with the Zwamy cloud server. It the circle is grey then the player is not in contact with the Zwamy cloud server. If the user tries to select a grey player, then an error message is displayed, “this player is not presently connected to Zwamy. Please check to make sure the Zwamy application is running”



Figure 5

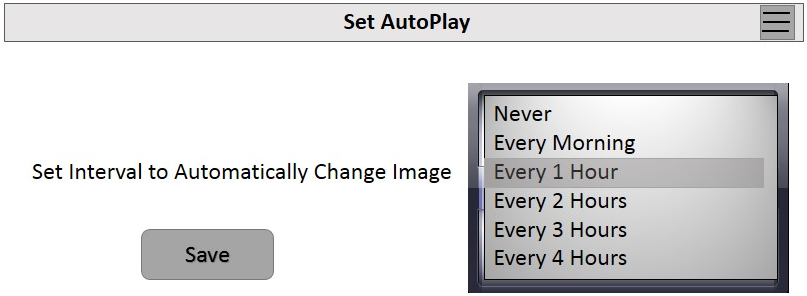
* 1. Set AutoPlay allows the user to have Zwamy automatically move forward in a viewlist, on a timed basis, without having to use a smartdevice. The user uses the scroll wheel to select the interval for image changing as shown in figure 6. 

Figure 6

* 1. Log Out lets the user log out the current user. When the user selects Log Out, Zwamy displays the log in screen (Figure 1)

MVP + 1 functionality

1. Native app support SmartDevice support multiple types of Smartdevices e.g. IOS and Android.
2. Users can also use the web display interface from MVP to control their client.

MVP + 2 functionality

1. User administration – edit/delete user account
2. Viewlist editing – there is a web interface which allows a user to create/edit viewlists by creating a viewlist and then selecting images from the content store to be members of the viewlist.
3. Viewlist sharing – there is a way to share viewlists between users
4. Client support – support for potential other clients like ps3 and Xbox 360
5. User client association – a client can be associated with multiple users (different family members can control the client with their own SmartDevice)