

## GALORE UI Screen Design

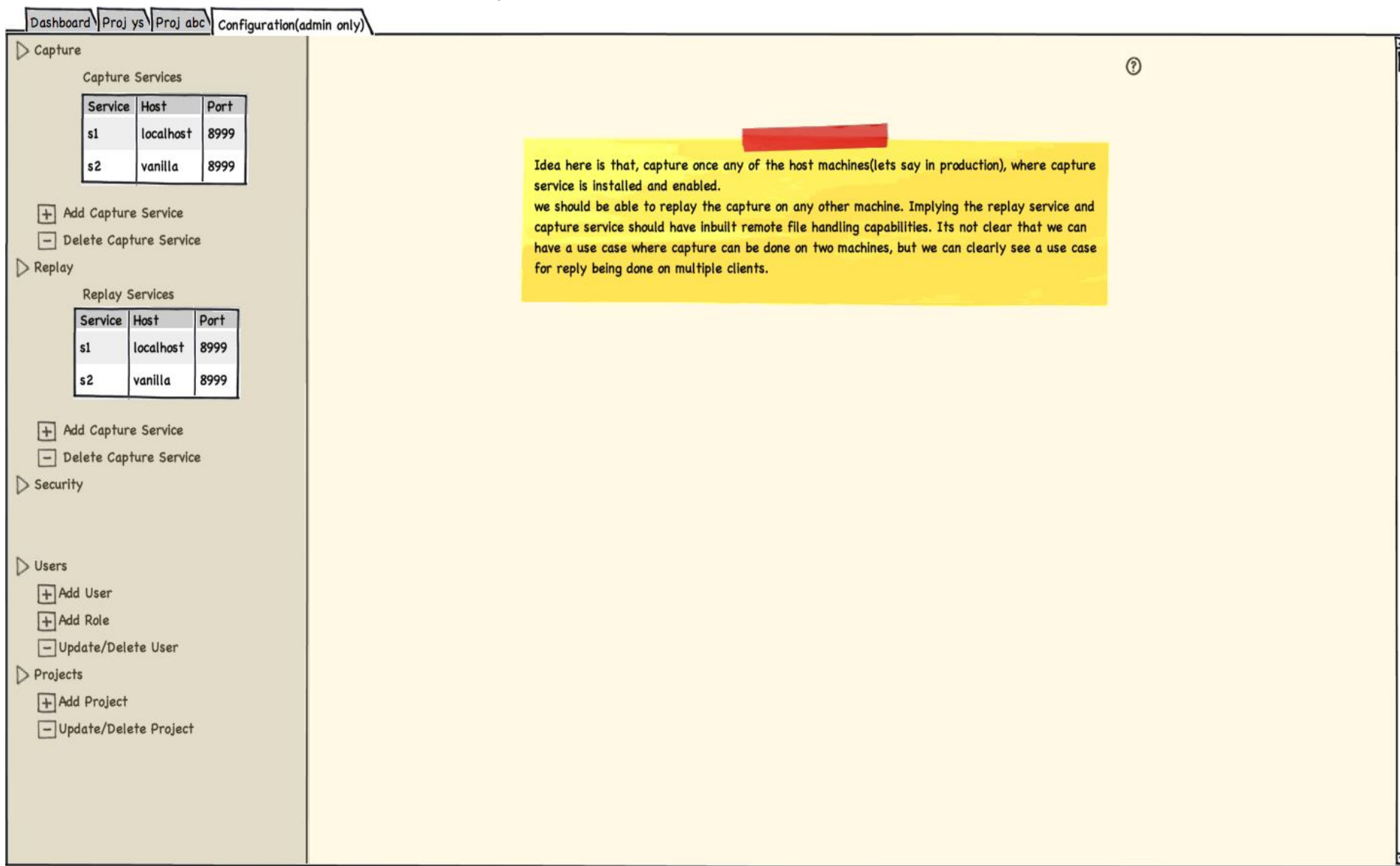
It would be nice to have some GUI from the beginning to be able to test the end to end use cases, thanks to iterative development model :-)

These are very basic not ready for non-developer consumption yet. However, this can be a good start to have the discussion going.

### ADMINISTRATION

First step after installing the product is to configure services. Galore can be installed on any number of machines and in particular any of its services can be hosted on many machines. Two main services that Galore provides are capture and replay. The administration UI screen should be able to configure any of these services as shown in the below screen. Ideally, there should be a discovery option(not shown in the screen; futuristic), when turned on should be able to figure out the existing services in the subnet and be able to auto configure.

Ability to create projects. A project is a collection a capture and multiple replay runs being managed in a single entity. Each project can be associated to one or more capture and replay services.

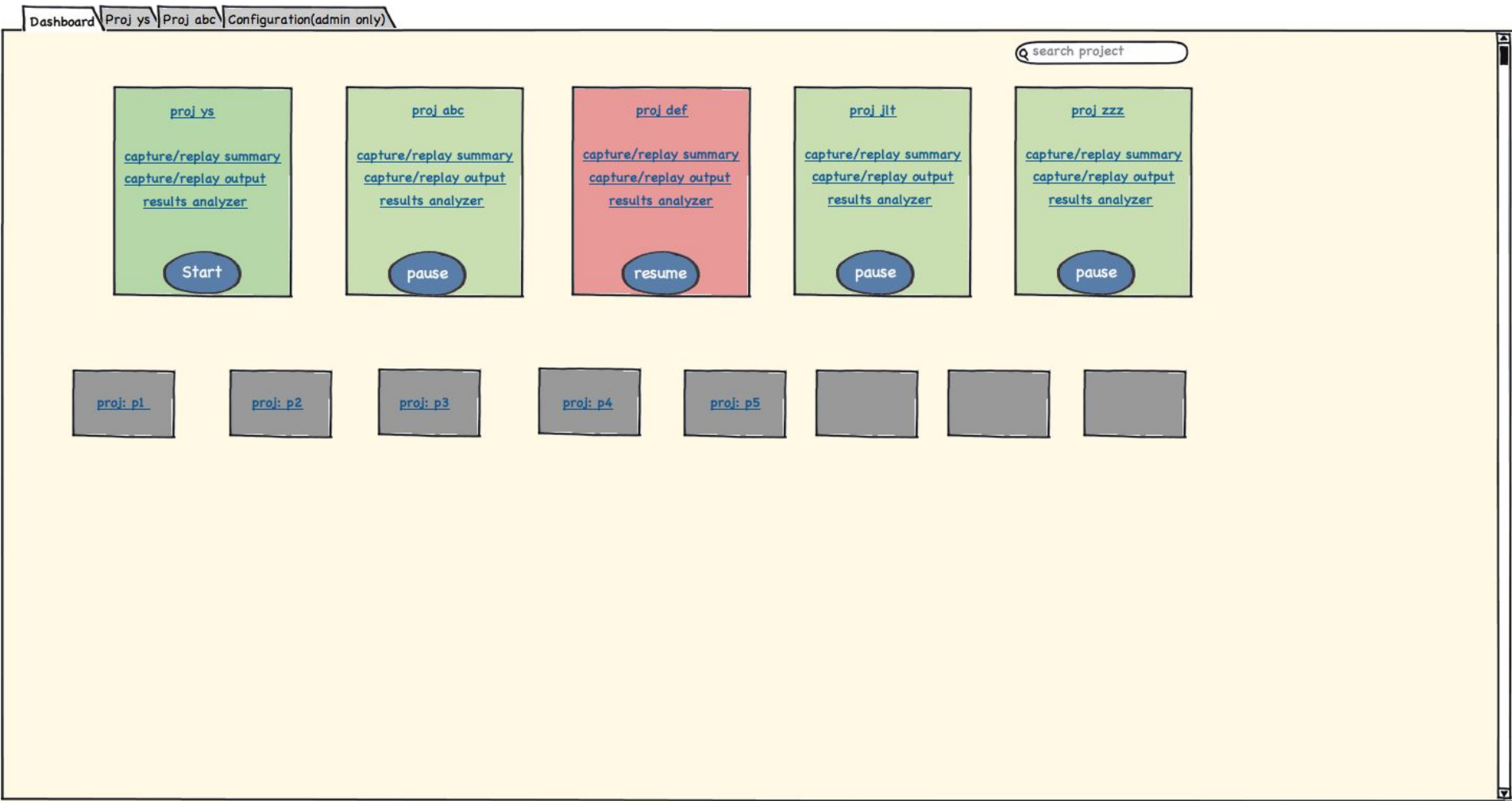


### DASHBOARD

This is the first screen when user/administrator sign in. The screen displays elliptical (shown as rectangles in the screen, but elliptical would be more nice) objects that represents projects with recent activity. Green indicates currently either capture/replay is going on and red indicates errors. All the brown ones indicates inactive projects. Each project object has links to directly go the project tab also some summary related to the activity so far done. If current activity is capture, then it can display things like, how many requests/repsponses logged so far/how much size captured so far/ how much space left on the device/alerts if any. Everything in the project object should be a link, so that when clicked on that link should open a new tab for the project and show the corresponding activity details. This would give us more details about how the project organization should be on the server

- every project should be accessible through a link(meaning like a service on the web server)
- every resource should be accessible through a link (this could also be the entire database on the back end)

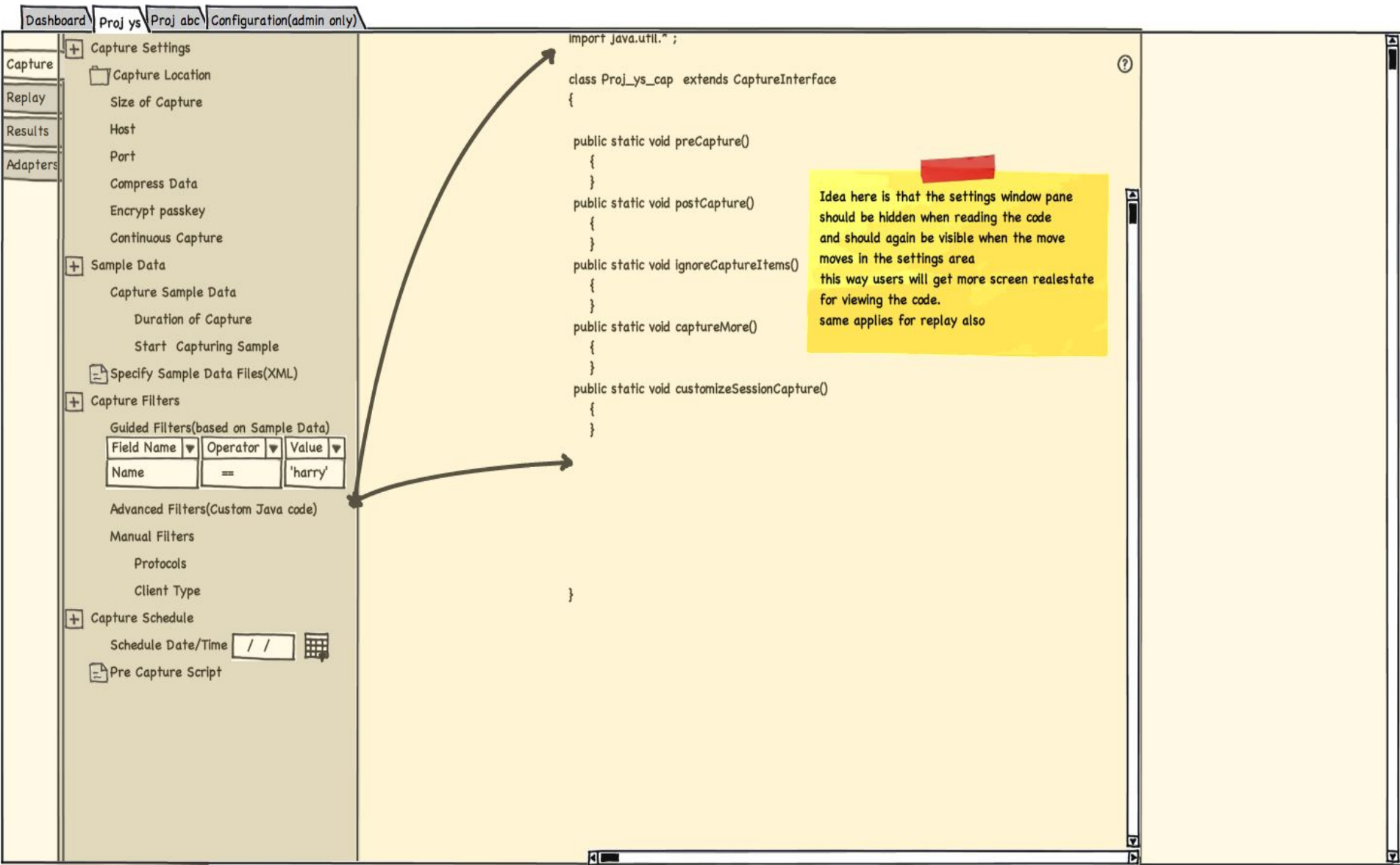
The dash board can be different for different users/roles. A regular user's dash board can simply show the list of projects he/she created in case there are no active projects. The idea here is that the user should be able to select the project from the dash board eventually. If the administrator logs in for the first time after installation, only a link to the configuration tab should be displayed on the dash board(of course in addition to welcome to clock replay message :-)



CAPTURE

One of the main things that should be considered while implementing the capture screen is to make the coding for customization/filtering of capture easy to the end user. This in fact depends on the design of the adapter framework. Also, while accessing the code, the left window pane related to settings should be hidden so that the entire screen can be used to display the existing code. The code should be nicely formatted and with markers as to where the user can write the customization. we should be able to do some syntax/semantic errors in the UI it self. Also, if we can make some sections read only and do not let users to modify, that makes it easy for validation. If the data needs to be secure, the admin can add a encryption key of his/her own and the system should encrypt the data before storing it. Eventually while replay, it can be decrypted with the same key(or we can use public private key infrastructure as well)

The capture screen should also give a nice interface for the users to execute any pre-capture scripts. Taking a back up of the database or backing up of any files necessary can be done through these scripts.



## REPLAY

Captured data can be replayed many times. so, replay details needs to be stored by the run. users can give a meaningful name to the run; the system will also create a timestamp and store it in addition to the run name. After the capture, the user should be able to view/add/modify/delete the data, which is made possible with the cover flow(dont know if we can get this control on java script, but its nice to have something like this). The user can see the cover flow either session based, i.e, all the network request/response pairs are grouped by the session and be displayed accordingly.

