Technical description about a few projects

Everything that is described below has been designed and developed by me, and I will be able to explain any part of it in any level of detail.

Since these products are used by governments security, accessibility, performance etc have been of great importance all the time.

EngagemenHQ

Its an online engagement platform mostly used by governments, that I developed from scratch. First version of it was in Rails 2. We had only a couple of clients then, and we used to run one instance of the app per client. Then before we knew we had over 75 clients and we ended up running 75 instances of the app, which made deployments difficult and hardware usage inefficient.

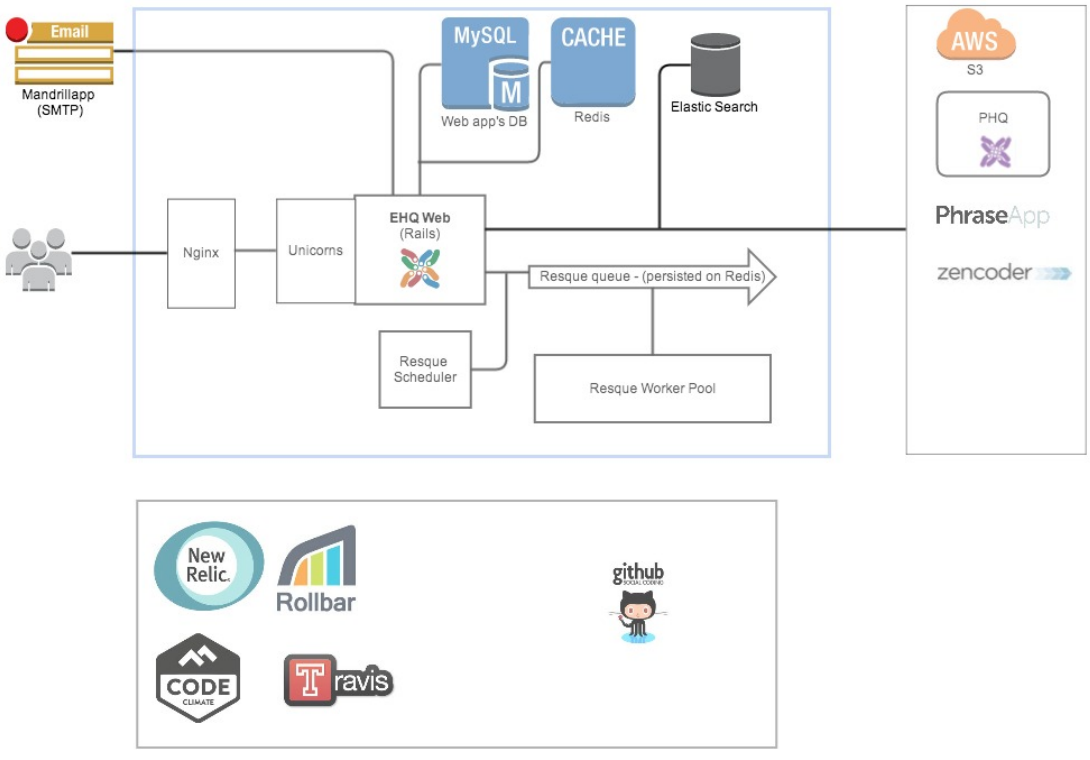
We then built the v2 of EngagementHQ, which was a multi tenanted app. ie. we had a single instance of the app running, which serves multiple web applications from the same app instance for different clients, with a shared database. We have about 250 different clients supported on this platform in Australia alone. We also have deployments in Canada, NZ and UK.

EHQ has various tools used for online engagement like forums, surveys, guest books, brainstormer, Q&A. It also has video/photo galleries, widgets etc. Admin section allows extensive customisation and theming which allows clients to match their sites to their corporate sites. There is also a Insights (qualitative and quantitative analytics) section which derives valuable insights from the data we collect, and presents it in a form that is directly consumable by the clients.

A few of our client sites:

http://sydneyyoursay.com.au/
http://yoursaywarringah.com.au/
http://haveyoursaywilloughby.com.au/
http://shapeyourcityhalifax.ca/

A diagram showing how EHQ has been architected, and various components that work together in production:



This shows a single monolithic Rails app. This has been futher split up into 4 Rails applications, one serving the user facing side (which has support for theming etc, and also handles 90% of the traffic), another for the admin section, one for the Analytics section and one for the Platform management.

EHQ also exposes an JSON API which is used by the web application as well as third party clients that are integrated to EHQ. This is also intended to be used by mobile and other clients that we build in future.

BudgetAllocator

http://hudsonave.budgetallocator.com/#ba
http://federalbudget2015.budgetallocator.com/#ba

Authoritah (Rubygem)

This is the access control module used by EngagementHQ. EHQ has complex access control requirements, with over 7 different types of users having different levels of access across the app. It becomes even more complex as the roles are dynamic. If a user U has role R1 when he visits a project P1, he might have role R2 when he is on project P2. To handle these requirements, and to keep the access control rules abstracted out of the code, I developed Authoritah. Authoritah stores access control lists in CSV files which can be modified without modifying the code. Please see the gem's Readme for more details.

https://github.com/unnitallman/acts_as_authoritah
https://rubygems.org/gems/acts_as_authoritah