

User Interface consolidation

There are currently 2 User interfaces for Egeria.

- **General UI** - polymer based that calls the UI chassis. This has asset exploration rex and tex
- **Ecosystem UI**, uses Node and React and view services and is multi tenant. This has rex, tex, dino, server author , glossary author.

This page is a working document detailing the plans to consolidate these UIs, so we can

- prevent code duplication
- reuse code where appropriate
- create a more coherent, consistent, intuitive experience for users of the UI(s)

~~Agreed in the Community meeting on the 29th of March 2022 and dev meeting from the 31st of March.~~

- ~~React UI work~~
 - ~~Some work has occurred to replace the existing expired Egeria certificates with up to date valid certificates, as part of this, the certificates are validated.~~
 - ~~The Egeria React UI 3.7 introduced new environment variables to allow the user to provide their own pfx ca and passphrase.~~
 - ~~Further work is being done to allow the pfx ca and passphrase to be specified per platform.~~
- ~~The concern was raised that this sort of routing should not be the responsibility of the the deployer of the UI. It would be more appropriate to use something like Ingenx or Istio to handle:~~
 - ~~routing~~
 - ~~authentication~~
 - ~~certificates~~
- ~~there seems to be a need for both these approaches:~~
 - ~~a manual approach allowing pocs, education and demos without requiring an overarching framework and a lot of enterprise orientated setup. This would run our node application.~~
 - ~~an enterprise approach that would:~~
 - ~~have a single authentication approach~~
 - ~~remove routing from the React UI~~
 - ~~remove the need for a node application.~~

~~Steps agreed in the TSC meeting on the 25th of January 2022.~~

- ~~Frontend~~
 - ~~package level first integration at the static file level only.~~
 - ~~consistent css (allow custom styling of the UI)~~
- ~~Minimal Considerations on reusing React UI components~~
 - ~~Accessing view service configuration (or equivalent) in General UI components (without this the react components will not be able to used asis)~~
- ~~Backend~~
 - ~~single authentication for both UIs.~~
 - ~~adopt Multi tenancy View services vs ui chassis~~

Workshop on the 24th of April 2023.

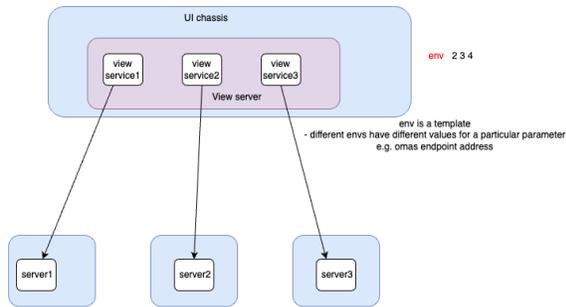
Agreed to have view services running in the UI chassis, the configuration will be held in a config file.

We have the concept of a view server having multiple view services, these are 'templated' so that they could be started with different OMAS endpoints. For example running in a dev environment and production one.



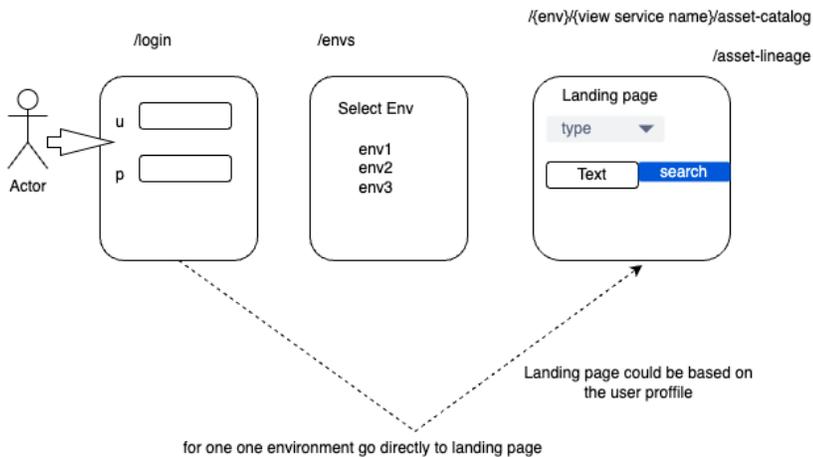
The proposed architecture is:

https://host-name/{env1}/asset
/lineage
/import



env = view server group ?
case 1) OMVS -> OMAS
case 2) OMVS -> OMRs
case 3) OMVS -> 2 OMAS

The ui experience would be



Work has been started on a Type script Tex

- on the last day we showed that the happy graph could display Tex data in a horizontal tree.
- Issues <https://github.com/odpi/happi-graph/issues/150> and <https://github.com/odpi/egeria-ui-components/issues/245> are tracking the tasks to complete this.

Other considerations

- Agree Git repositories
- How is react context is handled.
- Agree scope of Hackathon to show UIs working in a consolidated way
- Execute Hackathon