

2020-12-09 DM - UIs

Date

09 Dec 2020

Attendees

- [Mandy Chessell](#)
- [David Radley](#)
- [Nigel Jones](#)
- [Graham Wallis](#)
- [Maryna Strelchuk](#)
- [Ljupcho Palashevski](#)
- [Cezar Sirbu](#)
- [Bogdan Sava](#)

Goals

- Egeria UIs: closer look at 2 UIs that we currently have (reasoning, purpose). Proposal for moving forward, roadmap. – unifying view services
- UI discussion: Using electronjs to have dedicate browser for the UI. electronjs as an alternative for LINUX, MAC, and WINDOWS
- Update on Egeria TSC work

Discussion items

Time	Item	Who	Notes

David's Summary of the meeting

We had some very rich, open discussions in the User Interfaces convergence. I wanted to share my understanding and check it out with you all :

We talked of the following options for server side convergence :

- 1) Leave the Polymer UI and React UIs as is. In this case there are 2 implementations of Tex and Rex server side (and client side). Double fixing and potentially divergent designs; as the view service configuration drives the Tex and Rex behaviour.
- 2) Amend the Polymer UI so it uses a view service. So the Polymer UI chassis still has Spring security (that does role based authorisation) and forwards on the requests to the view service. This would mean that the same view services could be used for Tex Rex (and Dino if required) by the Polymer UI. Other UI capabilities in the Polymer UI could either stay in the UI chassis or move to the view service.
- 3) Investigate replacing Spring authorisation with the standard pluggable way Egeria does security and enhance the Egeria way if required. The idea being that we should not force a particular framework such as Spring, so it can be used by organisations that do not use Spring. IBM for example uses micro profile rather than Spring.
- 4) Multi tenancy would probably need to be implemented in some sense in the Polymer UI to enable 2. Either exposing the tenant in the url for the UI, or faking in the UI Chassis server to look like one tenant.
- 5) Change the UI Chassis to the presentation server using Express and node. There were concerns on this approach as it would lose the role based spring security. When we talked about Spring security in the OMAG servers we talked of Spring being a proprietary opinionated framework (not adopting the open standards in many cases) and we need to be able to switch Spring out for other options for example <https://projects.eclipse.org/projects/technology.microprofile>
- 6) There was an initial concern that the Polymer Ui has role based security and the React based UI was persona based security. Mandy felt that we could find an architecture that encompassed both role based and persona based security models, this would allow the UI to load components based on the role.

Can you confirm which approach(es) are being favoured / investigated? If I have missed or misrepresented one, please let me know.

Client side convergence

- 1) Leave as is.
- 2) Leave the Polymer and React frameworks and refactor Tex and Rex UI components so they can be re used. I think this means that the 2 Uis would need to embrace Web Components in some sense.
- 3) Re write the Polymer Ui in React. We talked about this option and there was concern that though react is very popular now, it is a proprietary standard (Facebook's) and web components built on open standards would be preferable. At the same time we have found that Using a Node React eco-system has enabled us to bring in an existing team to develop the Egeria React UI; the Server author Ui was developed very quickly as the team already had expert FED skills in this space.
- 4) Rewrite / wrap the React components as web components that could be reused (this may turn out to be the same as 2)

Can you confirm which approach(es) you were favouring / investigating? If I have missed or misrepresented one, please let me know?

One UI

If the Polymer and React UIs convergence will be in one UI, we need to create / agree a story board to drive all development of the UI |(not thinking about the technology) including the initial experience and navigation. This is the most important thing to do. As part of this we will need to:

- agree on community profile usage
- agree on multi tenancy
- agree on security model client side and server side
- current UI audiences are difference, Polymer is currently read-only for use by a mature metadata driven organisation like ING with an existing metadata repository focussing on lineage. The React UI should allow an organisation to implement their metadata driven vision using Egeria, including authoring of glossary terms and semantic assignments.

Action items

- [Ljupcho Palashevski](#) Review David's summary and confirm what actions his team are going to do.