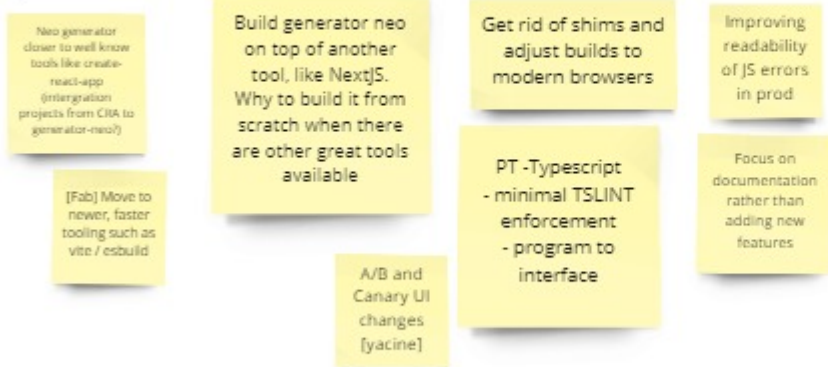




## DX



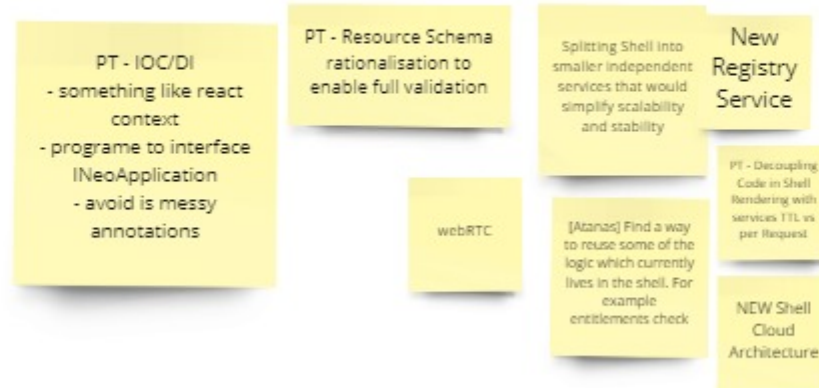
### Roadmap:

- Modernize the templates for UI projects
  - Collaboration with WMPC
  - Use modern tools to reduce build time

### Benefits:

- Better DX for contributors (less time spent waiting for code to build)
- Reduce attrition by using latest technologies
- Bring more self-service to contributors

## Webserver architecture

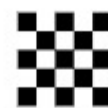


### Roadmap:

- Review shell architecture once available in Cloud
  - Split the shell into smaller microservices with less responsibilities
- Data analysis of pros / cons of bundling assets vs using http2

### Benefits:

- Increased stability - Shell is less a single point of failure
- Faster release cycle for small microservices
- Decrease inconsistencies between environments (and prod issues related to UI artifacts deployments)
- Performance improvements in loading Neo UI



## Micro frontend



### Roadmap:

- Add es modules support

### Benefits:

- Allow multiple panels (for example EVs, workspace) on the same page to load different versions for the same library
- Improve developer experience (less time spent waiting for code to build)



## Infrastructure

[Fab] Have the ability to spin up an exact copy of the shell for a given environment using cloud

Have an easy way to do rollout

Static assets served via CDN

[Fab] Use of a CDN to improve performance of serving js/css assets

Reduce load time to shared environment [yadine]

Have very prod like environments - and stop sharing dev1/dev2 - but have an environment per developer or pod (if needed)

HTTP2

Use Cloud CDN for assets (images and js artifacts)

How about HTTP3? :)  
BTW, there is a new HTTP code 103 for static assets



## Server Side Rendering

[Fab] Allowing partner teams to have their pages rendered on server side

Microservice for SSR

Find a way to be able to use NextJS - for SSR

Allow using tools such as NextJS to build SSR React applications. More SSR in Neo. Potentially it would require separate services deployments.

Prefetch data on the server [yadine]

Somehow integrate NextJS into our UI

Jamstack



## CX

Lazy loading and code splitting (es modules)

Create some way to request and cache api responses to avoid 50 requests for the same thing in a page



## Libraries

React 18 new capabilities (including server side streaming)

[Fab] React 18 upgrade + future upgrades

[Atanas] Make it super easy for apps to move to new versions of React

[Atanas] Upgrade the EV to the latest version of the webskd components and react

Unbundle /remove neo-core-js [Yadine]

### Roadmap:

- Ability to spawn short lived Neo that reflects a given environment, to be used during CI/CD
- Using HTTP2/3 once we are in Azure
- Gather data on benefits of using CDN with the goal of hosting our assets on a CDN

### Benefits:

- short lived env:
  - Better integration against prod like env, resulting in shift left testing approach
  - Faster feedback on developer changes
- HTTP2/3 + CDN:
  - Decrease load on shell and proxy
  - Decrease load time of Neo UI
  - Improve client experience



## Components

Framework agnostic UI components

[Alanes] Do not create any more instances of components :)

[Fab] Should we move to bundling everything (components?)  
Impact on EVs?



## Low-code / no-code

PT - Global Neo FDC3 - lib  
- standard compliant - open to extension  
- global taxonomy of extendable types  
- uses web rtc for comms under the hood?  
- bridge to desktop

[Fab] Provide a low-code environment

Make ACQA webapps a core Neo platform offering



## Desktop

[Fab] PWA - explore again (2 years later)

PT - OpenFin Platform API for Neo Desktop/externalise OneDesk

## Other

Recognition and connecting apps with SWC (AppDir) codes

Clean up entitlements - we will need to do this for A/B testing? :)

[Fab] mobile version - is there anything we can do to provide a better experience?