Micro Frontend Squad Notes

Directory

- To Address Next Meeting
- Meeting Notes
  - 2020-08-28
  - 2020-08-27: Demos of ongoing work: identifier generation, extensions
  - 2020-08-20: Sprint review & extensions discussion
  - 2020-08-10: Design Call
  - 2020-08-07: Insights from Ajay Brar re. Style Guides vs Design Systems
  - 2020-08-03 Design Call
  - 2020-07-27 Condition Widget - Modelling Design Call
  - 2020-07-23

To Address Next Meeting

1. FYI - Passcode change coming to zoom meetings - “1”
2. Outcome of Lightning vs Carbon review & implementation approach review (Style Guide vs Design System)
3. Extensions, and implications for an implementer.
   a. Community feedback
   b. Run by on-the-ground implementers.
4. Test Results updates
   a. Krishna (Nori) Teja to be included in design calls - BA to support and lead Test Results work. krishna.teja@thoughtworks.com

Meeting Notes

Sprint Board: https://issues.openmrs.org/secure/RapidBoard.jspa?rapidView=231&selectedIssue=MF-265
Priorities List: https://wiki.openmrs.org/display/projects/MFE+-+Proof+of+concept+release

2020-08-28

Notes

2020-08-27: Demos of ongoing work: identifier generation, extensions

PM Eric: Introduced Eric Achillah as new squad PM. Will be involved in planning and coordinating sprints and user testing, w/ support from Grace & JJ. Welcome! 😊

Discussion re. possible need to have style guide. To guide the how to use Carbon Design in the OMRS setting.

Demos

Identifiers - Samuel

How to help user with auto-generating a user id

Discussion on the different Patient Identifier types - there are multiple different OMRS IDs:
**Extensions demo - Brandon**

- Extensions are like a way people can add/change features within an Add On/app. Doesn't replace need for workflow to find/share/grab/add apps/addons/modules.
- RFC: Extensions implementation plan (#27)
- See repo: esm-core https://github.com/openmrs/esm-core
- Work this week from Brandon: https://github.com/openmrs/esm-core/tree/master/packages/esm-extension-manager
- ESM implementer tools. Goal is: OMRS SPA repo will contain very little JS.
- TODO: Brandon to share updated plan on Talk

**2020-08-20: Sprint review & extensions discussion**

- Address templates & validation done
- Unique identifiers discussion
- Extensions presentation (see PPT here!)

**2020-08-10: Design Call**

1. Implementer UX for Extensions
   a. 90% of apps made will be made for local usage.
   b. Widget comes by default with app, you don't need to configure it.
   c. Extension points: will be for specific extensibility; apps will not subscribe to a "class" or "type" of extension point.
   d. Widget implications: "I turn on an HIV widget, HIV things are then available elsewhere as well"
   e. Definitions:
      i. **App** = Space in single-page application that has things available elsewhere. A vertical solution to an issue (I have a problem, app solves it). There are settings I can adjust. e.g. Diabetes app: might add or make available a new widget for my patient dashboard that is diabetes-centric. Avoid places where you'd need to write code to adjust basic settings.
      ii. **Extension** = Extend the app by adding additional relevant things (e.g. HIV-related headings in the navigation)
   f. 3 central questions:
      i. Implementer interface or JSON-only?
      ii. Need more people from on-the-ground implementations to weigh-in - b/c this is about their user experience.
      iii. TODO: Brandon (& JJ & Burke review): Document description of different options, and implications for an implementer using the platform. Then can run this by on-the-ground implementers.

2. Design
   a. **tl;dr** takeaways: Picking a Design System helps us right away, but later when breaking changes are introduced, massive scale of updates needed. (Though - this could be a good clear task for the community to fix.)
   b. So: Select a design system to follow its patterns & set standards, but not to import into our code. Need mechanism for creating shared components. Sharing components across different teams. (e.g. Zeplin). Commit to a given look & feel, use the CSS from a Style Guide for all things that are already there - so vast majority of dev time is not spent creating UI elements. For things people couldn't find, we direct them to a shared Style Guide (like our own appendix) where people can share what they've custom-built where it wasn't well represented in the Style Guide.
   c. So (1) go to 3rd party Style Guide website for guidance; if you don't find something that really meets your needs, (2) go to supplemental OMRS website to see what others have adapted specifically for OMRS.
   d. Some people would copy-paste from somewhere else and adapt it and share it there. Like the weird case of a button that has a drop-down menu built in when you click it. You might want to copy the CSS from the 3rd party, but adapt it, and then share it for other OMRS contributors to pull from.
   e. Problem: This requires people becoming good at CSS up front rather than just importing.

   Confusion: what a pattern is and where the designers document that.

   TODO: JJ F/u with Ciaran. Start w/ next week?

3. Test Results
   a. TODO: JJ/Brandon: Test Results Repo creation for ThoughtWorks team to start working in (Krishna leading)

**2020-08-07: Insights from Ajay Brar re. Style Guides vs Design Systems**
Caveat: From context of client sites, not open source environments.

You still have to style

- Not opposed to DS. Has seen they introduce dependencies. Patches can be security related as well as usability etc etc.
- Bootstrap - found class naming difficult; wanted something very very lightweight. Switched to Foundation (can choose to just use CSS) and only used a few components
- With CSS/Global CSS, very easy to change if tomorrow you wanted something different
- What do you get in return?
- Clear building block guidelines
- Could achieve w/ visual style guide. But advantage of DS is instead of having 5 different implementations of a button, you get 1-button-code used across. Normally great but it's no longer "your" button
- Bulk from Flexibility = most difficult part
- e.g. button in Material: made to be extremely generic. ++ properties to allow changes; we may only need to use 2.
- Loading in code that 1/2 the time you don't know what it does
- Restricts you from adapting button further (e.g. with dropdown underneath it) - then you have to create custom UI components. (e.g. even Google doesn't use Material across its own applications)
- Bloat
- some allow you to chose what gets included vs what doesn't (don't have to include all the UI elements) but some don't have that capability
- est. 25-30% ^
- Handling Information Architecture (this should be laid out like this)
- Can say "follow Material guidelines the way they structure things" - e.g. "a text field should have labels on the left rather than on top"
- Security
- Anytime you have a JS library, have potential for security issues. e.g. pre-built React components. Not difficult to inject malicious code through JS.
- Having a CSS-only Style Guide would meant this isn't a concern - but that means you're not chosing a Design System (b/c whole idea is reuseable component library)
- Maintenance
- Idea of defining in design tool: Create button standards in Sketch, Figma etc, let that platform generate the CSS, so you're not dealing with patches/things that can break
- e.g. Sketch sticker sheet
- How do we make process of sharing code simple?
- Shared code repo
- Need to i.d. who owns it, who's responsible to maintaining it

2020-8-03 Design Call

Introducing Lousa Sheenaz - Palladium Kenya (HIV prevention, care & treatment)
Style Guide evaluation update

- TODO: Grace F/U w/ Ciaran re. absence of card view; expectations re. design system selection work; further detail re. his thoughts on Bootstrap
- TODO: F/U w Bahmni - what do they use?
- TODO: F/U w/ jr dev volunteers - + book wrap up

Comes down to: Is there a better choice for OMRS than Bootstrap?
2020-07-27 Condition Widget - Modelling Design Call

- Could make sure that FHIR2 module captures business flows w/ existing API
- Can't build on existing condition modelling in RefApp due to tech issues and workflow/use case issues
- If RefApp 2.11 was shipped w/ OMRS Spa, could new condition list be included out-of-the-box? Yes.
- Easiest would be if we could ship a version of the RefApp where one of the modules has some of the MFE artifacts; module can deploy taste of MFEs without having to set up separate docket container etc
- Develop on SDK itself?
- Issue around FHIR "active" status - active for patient vs active in clinician's current review? Active lists were meant to help handle adding and removing conditions to a list.

How can MFE Squad support, in general and in Sprint 3?

- Items should go on/off list; remove with single click; have undo
- Take current condition component in MFEs and make work against FHIR API
- General Goal: Get conditions working in MFEs with aim to take component and intro into RefApp

Next Steps

- TODO: Mark: Log bug tickets with clear descriptions around need to fix the API so it uses the UI correctly; flag Ian and include in FHIR2 Module release project
- TODO: Sprint 3: Ian and Brandon support: Get current MFE condition component hooked up to FHIR API
- TODO: Grace: Clean up Wiki page & talk posts for condition lists - compare to what exists, document tickets & share w/ Mark & Burke for review
- https://wiki.openmrs.org/display/projects/Production-ready+Condition+list

2020-07-23

1. Announcements

1. **FHIR2 Release: Mid August.** Should cover 70-80% of data in OMRS systems. Resources, ability to create/update/delete resources, ability to do complex searching for those resources. "Almost as much FHIR support as EPIC".

1. ThoughtWorks Intros: 6 devs from Germany joined MFE project over last week. Availability can change w/in a day; so recorded onboarding calls as a future resource. Cards picked up by pairs.

1. **End of sprint 2: Demo**/share our work together

1. PRs re. extensions - see here if you want to see ongoing discussion:

   1. https://github.com/openmrs/openmrs-esm-api/pull/43

1. Ampath had session w/ Florian re. new architecture - looking at design for flexible MFEs to come up w/ widgets that are flexible t/o MFEs


1. Frontend Development Vision statement drafted; vetting with Bahmni to see how they see this fits in for future convergence using this tech

1. Draft:

1. **Style Guides/Design Systems Update & Discussion:** @Ciarán Duffy will join us for the 2nd half of the call and we'll address next steps here and open the opportunity for questions.

1. Ciaran having period to speak to people in community and interview would give better recommendation/rationale - later in August due to admin blockers