FIELD ACQUIRED INFORMATION MANAGEMENT SYSTEMS PROJECT:

TAG Elaboration Report Discussion

Brian Ballsun-Stanton
16 February 2021

MACQUARIE University

Australian Research Data Commons

Purpose of the Meeting

- Report on the Elaboration Plans
  - We will assume you've read the report
  - Clarify, explain, answer questions
- Summarise current feedback
- **Gain your acceptance of the Elaboration Report**
- Note issues in the report which will require more feedback
- Prepare recommendations for Steering Committee

Design Objectives

- In-use FAIMS 2.6 features in 3 modules
- Self-service module creation and customisation
- Cross-platform
- external round-trip
- Improved performance and record capacity

Technologies: Programming Language & Datastore

- NODE.JS
- CouchDB + PouchDB
- Progressive JS Single Page Application wrapped in Native code for cross-platform support using Capacitor

Technologies: Framework

- React.js or small-scale frameworks
- React is very heavy-weight and may present complications in terms of dependencies.
- Choice of React or no overarching framework will be determined by what we need to write or rewrite.

Technologies: Packages

- JSON Forms
  - Probably inspiration than specifically useful due to limitations in JSON Schema
  - Specifying UI via JSON makes for portable, inspectable specifications for better open-science
- Leaflet
  - GIS-"lite" engine
  - Our hopes for map rendering
- Almost everything will be a plugin
  - Less monolithic application
  - Easier to replace plugins than the entire thing

Technologies: APIs

- CouchDB will be world-writable (with authentication)
  - Richard raised a desire for an abstraction layer for more consistent interactions.
  - Accomplished through the mechanisms we use to enable round-trip data export.
- Views will maintain version consistency for CouchDB interactions
- The application will also have a public API to communicate with each other and to allow data exchange initiated by users into external services
- Exporters much like 2.6

Technologies: DevOps, QA, CI/CD

- As much automated testing as possible:
  - Unit and integration tests
  - Reduce ongoing testing costs to make a more consistent maintenance burden
- CSIRO to manage QA
- Automated builds (and automated testing) through a CI/CD pipeline using Github Actions.

Non-elaborated Functionality

- Geolocation
- File access
- Camera/Barcodes
- Audio/Video
- Raw Bluetooth Serial
- Document Scanner
- External GPS
- GIS
- Offline maps
- Dynamic UI
- Module Generator
- Data importers

Team Composition

- Macquarie Uni
  - Overall planning and execution
  - Product Owner, sprint planning, subject matter experts
  - DevOps
- AAO
  - Primary Developers
  - "Sprint" like development cycle
- CSIRO
  - QA -- integration and end to end tests
  - UAT

Future Decisions

- Minimise upstream dependencies
- Focus only on used GIS features
- Explore the round-trip problem
- Collaborate with external partners