

WHERE TO NOW?
EXPANDING
FULCRO

UNFINISHED WORK/FEATURES

NETWORK COMBINING

WHAT DIDN'T MAKE THE CUT

- Network read/write combining
 - Limited combining was implemented.
 - Opportunity to actually improve this story.
- Fulcro Spec client renderer
 - Using Bruce Hauman's runner now, but it isn't pretty/nice.
 - We've mostly started writing everything in CLJC so we can manually test against the JVM.
- Internationalization is partially ported, but untested
- Support viewer never matured/maintained

INTERNATIONALIZATION

NEEDS LOVE

- Fully baked in F2, unfinished for F3
- Lots of “wow” potential here, but I don’t use it (currently)
- Features:
 - Strings as keys, auto-extraction, dynamic load of locales into client.
- Cool Add-ons that would be “easy”:
 - Able to edit strings IN UI via Fulcro Inspect, submit to server, and write directly to PO file.

CLJS TEST RUNNER

GOOD ENOUGH, BUT...

- Fulcro Spec had one, but it was a mess
- Using Bruce Hauman's renderer
 - Pro: someone else's problem
 - Con: Isn't very good
- Desire: Ability to have the experience we have on CLJ (Demo?)

SUPPORT VIEWER

I PERSONALLY DON'T CARE

- Was a compelling idea
 - I have not had a use for it yet, and I don't really care too much about it.
 - Easy to build, but needs polish
 - Would require plugging in a "state recorder" on the tx system
 - Standardized middleware would be nice

RAD

SCHEMA

RAD SCHEMA

NON-REFERENCE PROPERTIES

- Can live on many things (not tied to just one entity/table)
 - Tony likes broccoli
 - Golden Retriever likes broccoli
- Might be arrays/tuples (PostgreSQL array, Datomic tuple)
- Might have a "fact" that crosses storage (image URL + image binary store)

RAD SCHEMA

REFERENCE PROPERTIES

- Many ways to model relationships
 - RAD needs to be able to generate reads and writes.
 - It is possible to model relations that go through a sequence of steps for reads.
 - Pathom does complex operations to find this data, but how do we manage writing that data? Can we leverage Pathom engine to “remember” identity of the items it traversed?

```
(defsc-form Thing [_ _]  
  {:form/attributes [:account/name :github.user/name]  
   ...})
```

RAD SCHEMA

HODUR

- Could we leverage Hodur for db management?
 - Possible to use for db artifacts, but doesn't use an attribute model.
 - No open maps.
 - Would not be authoritative.
 - Does have some nice-to-have aspects:
 - Somebody else's problems
 - Plugins for things like visualization
- Generate Hodur FROM our open maps???

UI/IO/ GENERATION

NESTED FORMS

ISSUES/THOUGHTS

- Form-state can already do proper delta/state management
 - Similar to report case, but complicated by there being multiple potential reference fields
 - Possible to solve the compound-field issue with nested form.

```
::form/subforms { :contact/phone-numbers PhoneNumberForm  
                  [ :contact/avatar :contact/avatar-url] AvatarForm }
```

RENDERING FORMS

DISPATCH

- Rendering Dispatch
 - Needs to flex on “style” (e.g. Bootstrap vs. Semantic UI) and “API” (e.g. Web, React Native, etc.)
 - MUST NOT crap on one when using the other.
 - Must support dead code elimination for parts you don’t use (multi-methods are not good for that, unless you don’t require them).
 - So far I’ve got “render-layout” and “render-field”. Both of these *potentially* need additional dispatch (style and data type)

RENDERING FORMS

LAYOUT/FORMAT IDEAS

- Dispatch? Form Attributes?

```
::form/field-style { :address/postal-code :zipcode
                    ;; What if long/lat? Two props => one "input"?
                    :address/geocoordinate :google-map
                    ;; Other "Related" inputs
                    :avatar/image :file-picker
                    ;; links to result of upload
                    :avatar/url     ??? }
```

- Layout styles
- Can do fancy things like "auto-generated" grid from arrays.

```
::form/layout [[:account/first-name :account/last-name]
                [:account/address]
                [:account/city :account/state :account/zip]]
```


FORMS I/O

COMPLEX WRITES

- Related Attributes
 - Image URL vs. Image data itself
 - Multiple facts, and probably even crosses storage medium.
 - Could write image first, return URL, then write URL.
 - Generalization???

STUFF TO BUILD

STUFF TO BUILD

TYPE SUPPORT

- CLJC + Transit:
 - Date/time/timezone (see Tick, but more focused) (PARTIAL)
 - Local time, local date.
 - Arbitrary precision numerics (we have math lib for bigdec)
 - ???

STUFF TO BUILD

FORM WIDGETS

- At least one complete “widget set” for standard data types.
 - Semantic UI is my preference
- Need to invent validation support constructs:
 - Attribute-centric (CLJC lambda?)
 - Form-centric overrides
 - Full form validation (passwords match, A comes before B, etc.)

STUFF TO BUILD

SCHEMA DECLARATION STANDARD

- What do we need to know in order to do Schema and Resolver generation for a storage system (SQL, Datomic, Firebase, etc.)?
 - Standard set of keys
 - Database adapter can define their own extended set...we use open maps.

STUFF TO BUILD

DATABASE ADAPTERS

- At least one complete database adapter (preferably two so we can work out generalizations).
 - Datomic is what I need, but I also use PostgreSQL
 - MySQL?
 - Firebase?
 - Others?
- Database adapter is responsible for schema diffing, and entity-level queries by ID, and converting data diff (by id) into writes.

STUFF TO BUILD

REPORTING

- To-one, and to-many
 - Possibly use form UI things in read-only mode for to-one case?
 -

PATHOM

TODO

- Finish query planner (In Progress)
 - UI for execution plan in Inspect
- Multi-depth batch support
- Nested inputs for resolvers
- Autocomplete for mutations/parameters (data is there, but no UI)

AUTH

PLUGGABLE HANDLERS

AUTHENTICATION/AUTHORIZATION

- Figure out how to do oauth kinds of things (redirects)
- Mechanism to check auth
- Root level authorization
 - Routes
 - Server vs client-capable permissions policies?

HTML5 ROUTING

- Needed for bookmarking and redirect handling
-