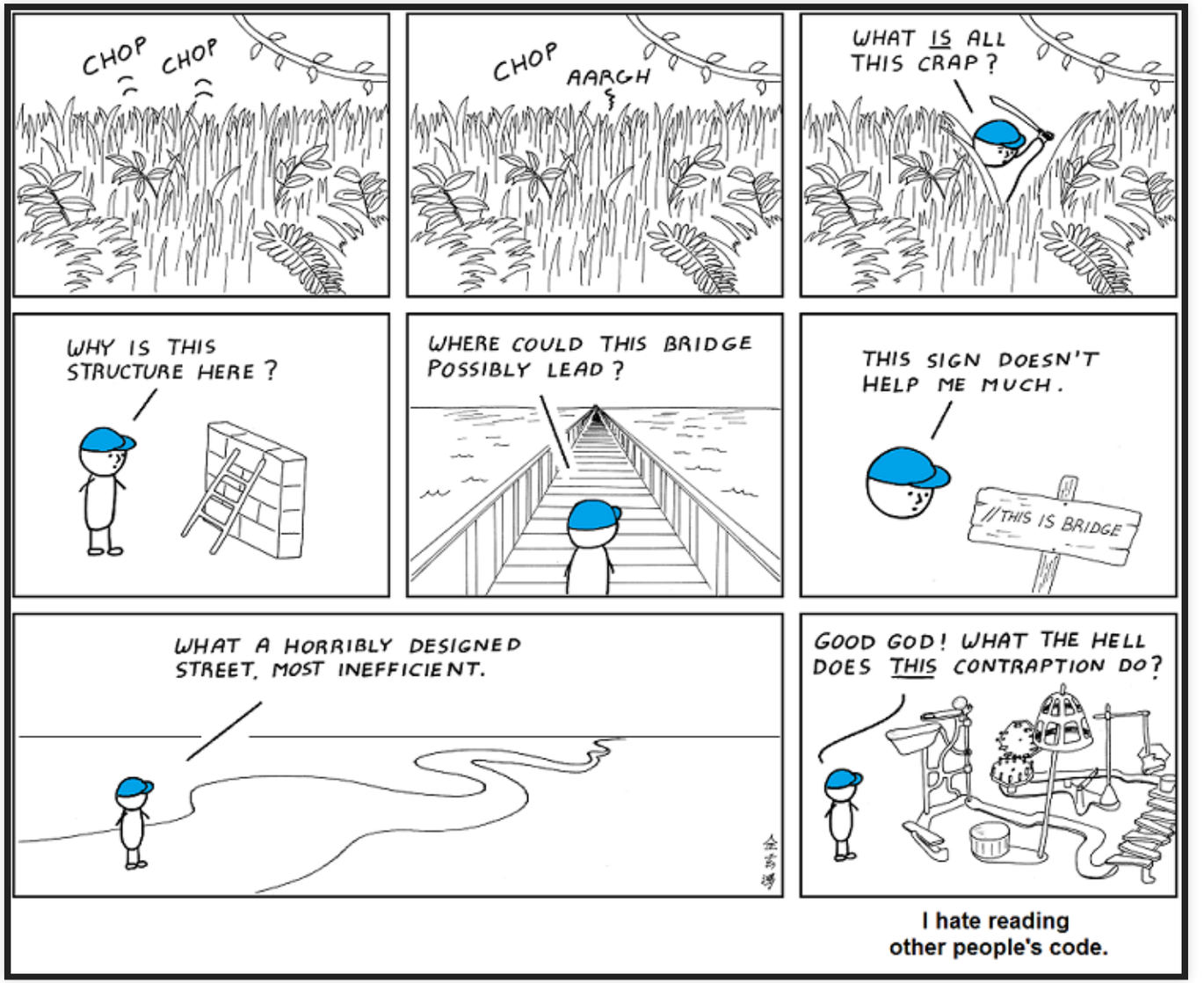
Unlock the power of feature-based JS development - JSConf 2018

Jeff Barczewski

jeff@codewinds.com
@jeffbski

https://codewinds.com/jsconf2018



Artist: Abstruce Goose CC-BY-NC 3.0 US

Jeff Barczewski

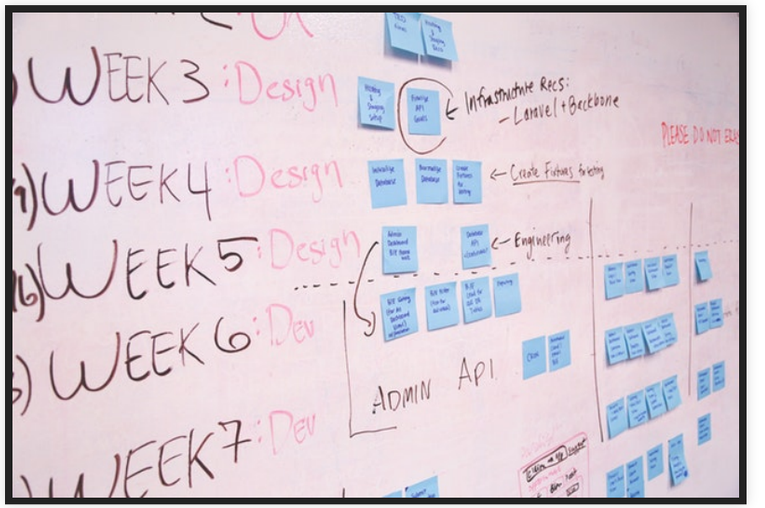
- Married, Father, Catholic
- 28 yrs as developer (be nice to the old guy :-)
- JS (since 95, my focus for last 7 years)
- Open Source: redux-logic, pkglink, …
- Founded CodeWinds, training/consulting
  - Functional JS, React, Redux, RxJS, Node.js
- I love teaching and mentoring, contact me

CodeWinds

- Live training (in-person or webinar)
- Need training/mentoring for your team on any of these?
  - React, Redux, redux-logic
  - JavaScript, Node.js
  - Functional JS
  - RxJS
  - GraphQL, Apollo, Prisma
  - FaaS, Serverless
- Code reviews, architecture help, small projects
- I'd love to work with you and your team

Why feature-based development

- software is complex
- MVP, sprints, evolve code
- git commits, PR's merge a feature at a time
- organize by feature
- A/B testing, feature switches, early adopters

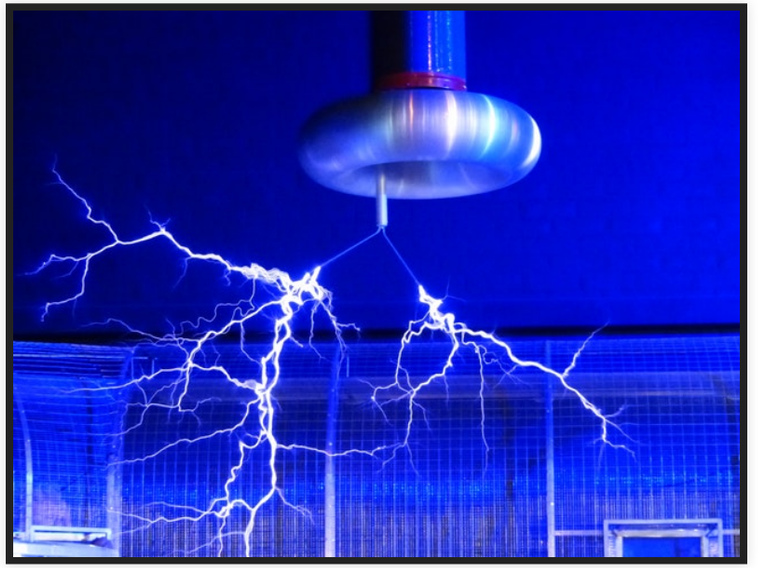


Challenges

- maintaining flexibility
- wiring, boilerplate
- cross-feature sharing
- hooks, dependencies

Previous Attempts (and limitations)

- PR's - merge conflicts, timing, deps, hard to A/B
- Boilerplate projects - rigid tech and structure, can get stale
- DI frameworks - complex, not designed for feature-based development
- Global object - collisions, no usage contracts, no wildcard selection



Feature-U

Feature Based Project Organization for React

Authored by Kevin Bridges

http://feature-u.js.org

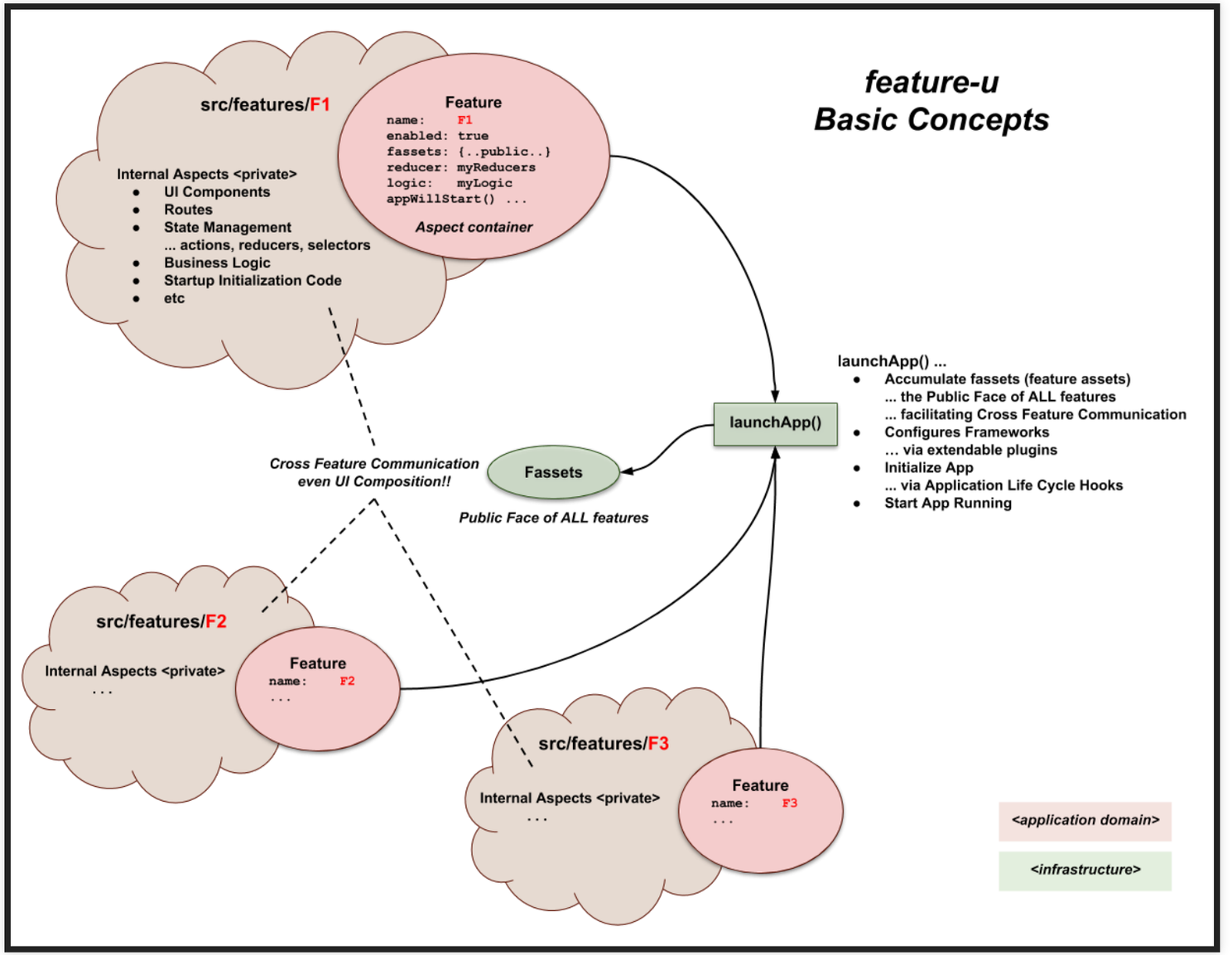
https://github.com/kevinast/feature-u

Feature-U Benefits

- designed for feature development
- simple, flexible
- cross-feature sharing
- encapsulation / org by feature
- easy to test and A/B
- dramatically reduced boilerplate/wiring
- life cycle hooks
- aspects for extension
- validation and usage contracts

High-level Overview

- create and register features
- instantiate and register aspects
- launch app with features and assets
- features define fassets for cross-feature sharing
- features use fassets as necessary
- feature-u can help manage/validate fassets


feature-u
Basic Concepts
src/features/F1
Feature
name: F1
enabled: true
fassets: {..public..}
reducer: myReducers
logic: myLogic
appWillStart() ...
Aspect container
Internal Aspects <private>
UI Components
Routes
State Management
... actions, reducers, selectors
Business Logic
Startup Initialization Code
etc
launchApp()
launchApp() ...
Accumulate fassets (feature assets)
... the Public Face of ALL features
... facilitating Cross Feature Communication
Configures Frameworks
... via extendable plugins
Initialize App
... via Application Life Cycle Hooks
Start App Running
Cross Feature Communication
even UI Composition!!
Fassets
Public Face of ALL features
src/features/F2
Internal Aspects <private>
...
Feature
name: F2
...
src/features/F3
Internal Aspects <private>
...
Feature
name: F3
...
<application domain>
<infrastructure>

Usage - Structure

```
src/
  index.js                  ... launches app using launchApp()

  features/
    index.js              ... accumulate/promote all Feature objects (within the app)

    featureA/             ... a feature (within the app)
      actions.js
      comp/
        ScreenA1.js
        ScreenA2.js
      index.js            ... promotes featureA object using createFeature()
      logic.js
      reducer.js
    featureB/             ... another feature
      ...
```

Usage - Feature Object

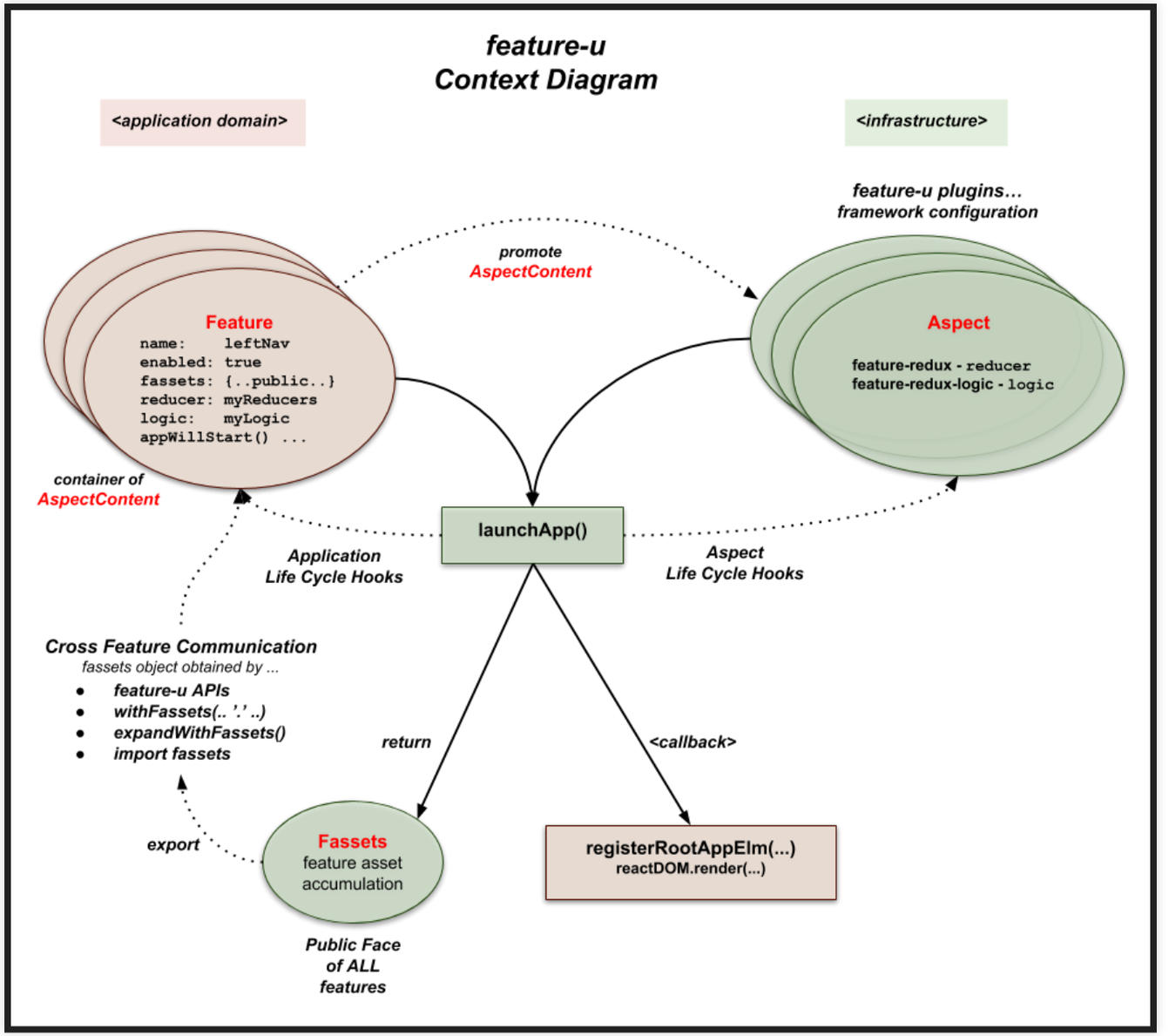
```
import {createFeature}  from 'feature-u';
import reducer          from './state';
import logic            from './logic';

export default createFeature({
  name:     'featureA', // a unique feature name
  enabled:  true, // optional, default is true

  fassets: { // use, define, and/or defineUse
    define: {
      'foo.xyz.comp':  () => ... implementation omitted,
      'foo.actions': actions
    },
  },

  reducer, // sliced reducer (feature-redux aspect content)
  logic, // logic array (feature-redux-logic aspect content)

  appWillStart({ fassets, curRootAppElm }) { ... },
  appDidStart({ fassets, appState, dispatch }){ ... }
});
```

Usage - Feature Accumulation

```
import featureA  from './featureA';
import featureB  from './featureB';
import featureC  from './featureC';
...

// promote ALL our features through a single import (accumulated in an array)
export default [
  featureA,
  featureB,
  featureC
];
```

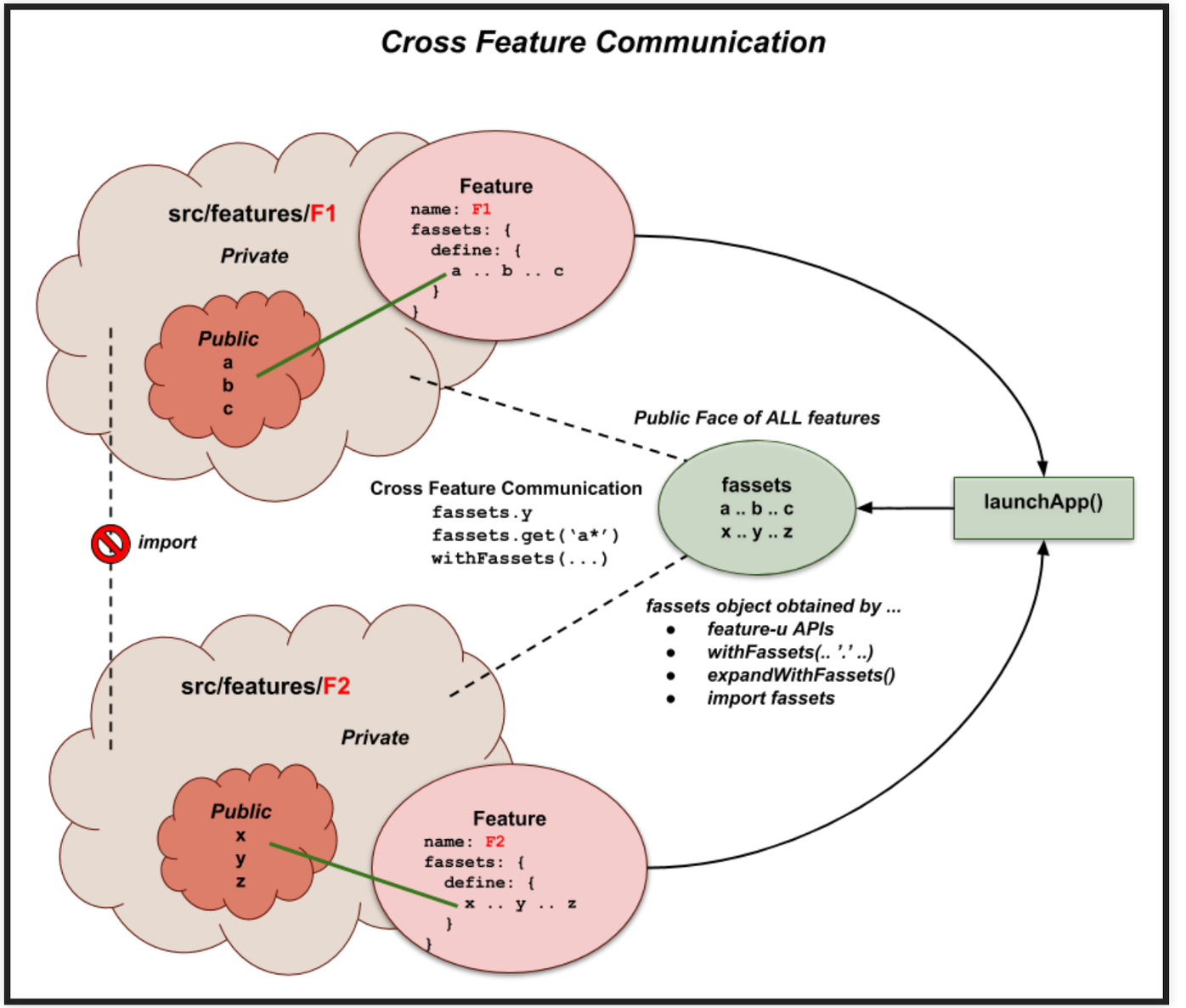
Usage - launchApp

```
import ReactDOM           from 'react-dom';
import {launchApp}        from 'feature-u';
import {createReducerAspect}   from 'feature-redux';
import {createLogicAspect}     from 'feature-redux-logic';
import features           from './features';

// launch our app, exposing the Fassets object (facilitating cross-feature sharing)
export default launchApp({                  // *4*

  aspects: [                                // *1*
    createReducerAspect(), // redux           ... extending: Feature.reducer
    createLogicAspect()   // redux-logic      ... extending: Feature.logic
  ],

  features,                                 // *2*

  registerRootAppElm(rootAppElm) {   // *3*
    ReactDOM.render(rootAppElm, getElementById('root'));
  }
});
```


feature-u
Context Diagram
<application domain>
<infrastructure>
feature-u plugins…
framework configuration
promote
AspectContent
Feature
name: leftNav
enabled: true
fassets: {..public..}
reducer: myReducers
logic: myLogic
appWillStart() ...
Aspect
feature-redux - reducer
feature-redux-logic - logic
container of
AspectContent
launchApp()
Application
Life Cycle Hooks
Aspect
Life Cycle Hooks
Cross Feature Communication
fassets object obtained by ...
feature-u APIs
withFassets(.. '.' ..)
expandWithFassets()
import fassets
return
<callback>
export
Fassets
feature asset
accumulation
registerRootAppElm(...)
reactDOM.render(...)
Public Face
of ALL
features

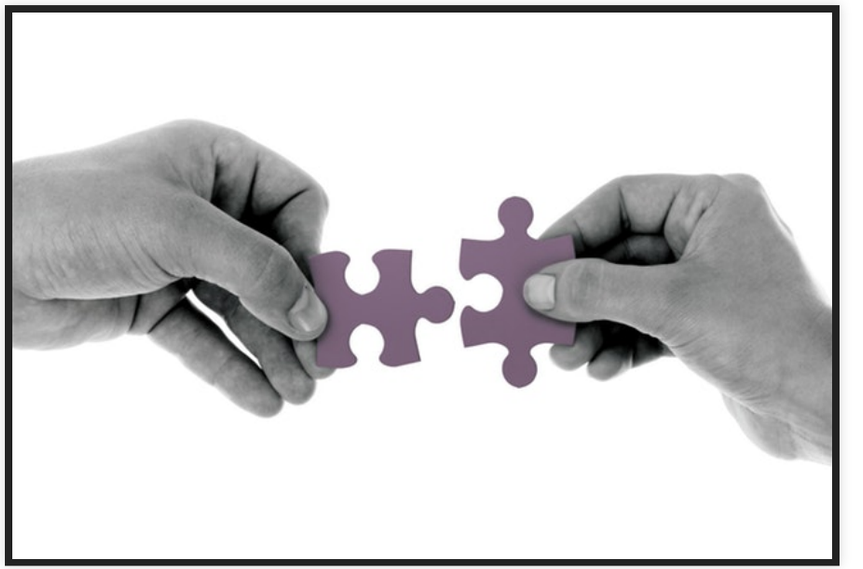
Cross feature sharing - usage contracts

```
const feature = createFeature({
  name: 'foo',

  fassets: {
    use: [ // I am using these fassets from other features
      'bar.link.comp', // default is required to exist
      ['cat.route.comp', { required: false }], // not required
      ['dog.action.d', { type: validationFn }], // type validation
      '*.main.link.comp' // wildcard match, array of comps
    ],

    define: { // defining some fassets, not required to be used
      'foo.actions': actions
    },

    defineUse: { // defining more fassets, expected to match a use
      'foo.link.comp': linkComp,
      'foo.route.comp': routeComp
    }
  }
});
```


Cross Feature Communication
src/features/F1
Private
Public
a
b
c
Feature
name: F1
fassets: {
define: {
a .. b .. c
}
}
Public Face of ALL features
fassets
a .. b .. c
x .. y .. z
launchApp()
Cross Feature Communication
fassets.y
fassets.get('a*')
withFassets(...)
import
fassets object obtained by ...
feature-u APIs
withFassets(.. '.' ..)
expandWithFassets()
import fassets
src/features/F2
Private
Public
x
y
z
Feature
name: F2
fassets: {
define: {
x .. y .. z
}
}

Usage - Accessing fassets in features

1. `withFassets` HOC
2. `fassets` provided in life cycle hooks - `appWillStart`, `appDidStart`
3. `expandWithFassets` expand feature aspect content
4. `launchApp` returns fassets object



See Cross feature Communication in feature-u docs

Code Walkthrough

- simple web app
- Expo (react-native) application

Best Practices

1. Avoid cross feature imports
   - use fassets defined by features
   - helps your features to be plug and play
   - easy to test without complex mocking
   - simple to A/B test, swap features
2. Take advantage of the usage contracts
3. Use wildcard selectors to reduce hardcoded refs
4. featureName is required to be unique
   - can use to namespace actions, state, logic, fasset names
   - if feature spans many files, export from a file

Summary

- feature-u simplifies feature-based development
  - cross-feature sharing
  - usage contracts / validation
  - life cycle hooks
  - reduced wiring and boilerplate
  - flexible and extensible
- This technique could be applied to other frameworks and languages

Thanks

- https://codewinds.com/jsconf2018 - slides
- https://feature-u.js.org - feature-u docs
- https://github.com/kevinast/feature-u - repo
- https://codewinds.com/
  (training/consulting)
- jeff@codewinds.com @jeffbski