```
$ git clone https://github.com/shageman/sportsball.git
$ cd sportsball
$ ./build.sh
```

What brings you here?

Getting started with #cbra

Component-Based Rails Applications

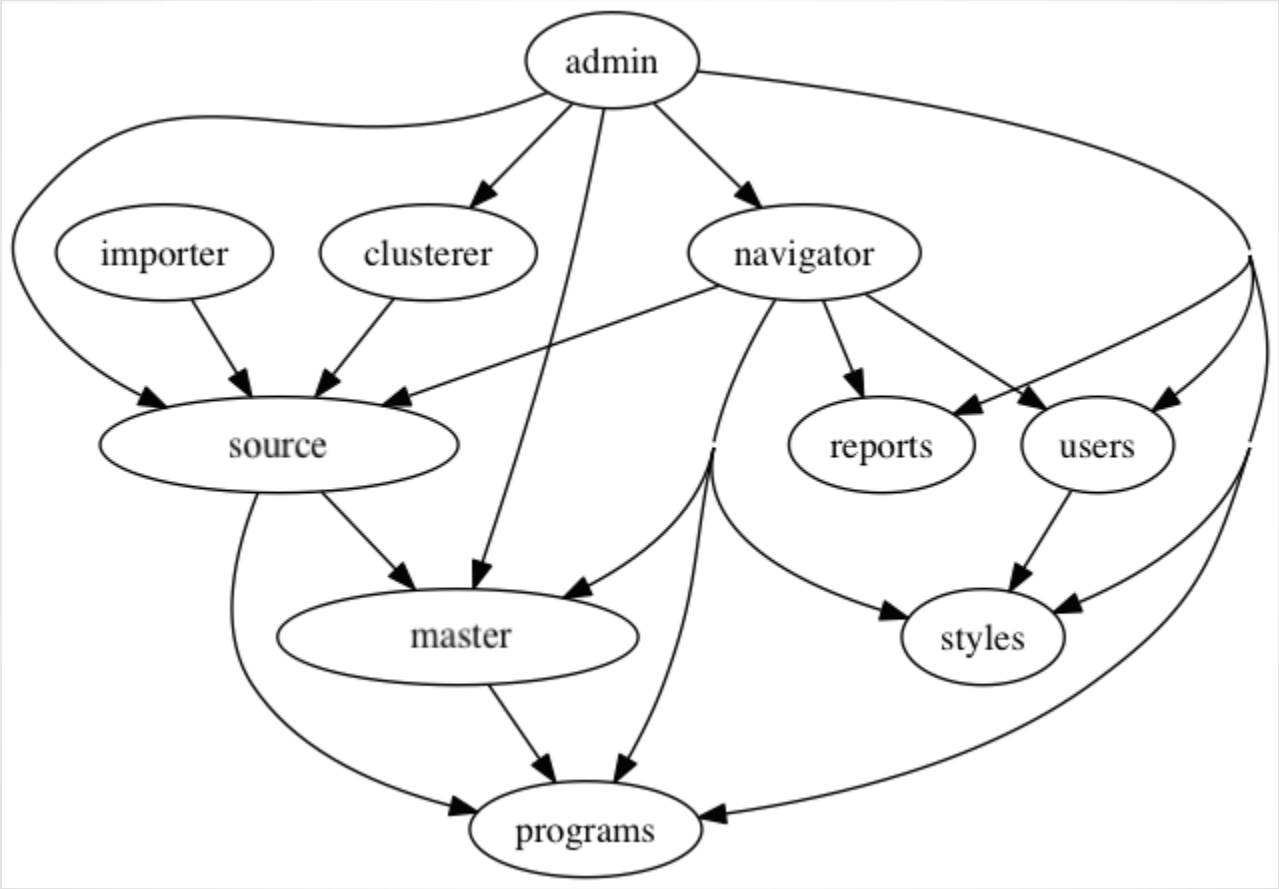
RailsConf 2015, Labs Workshop

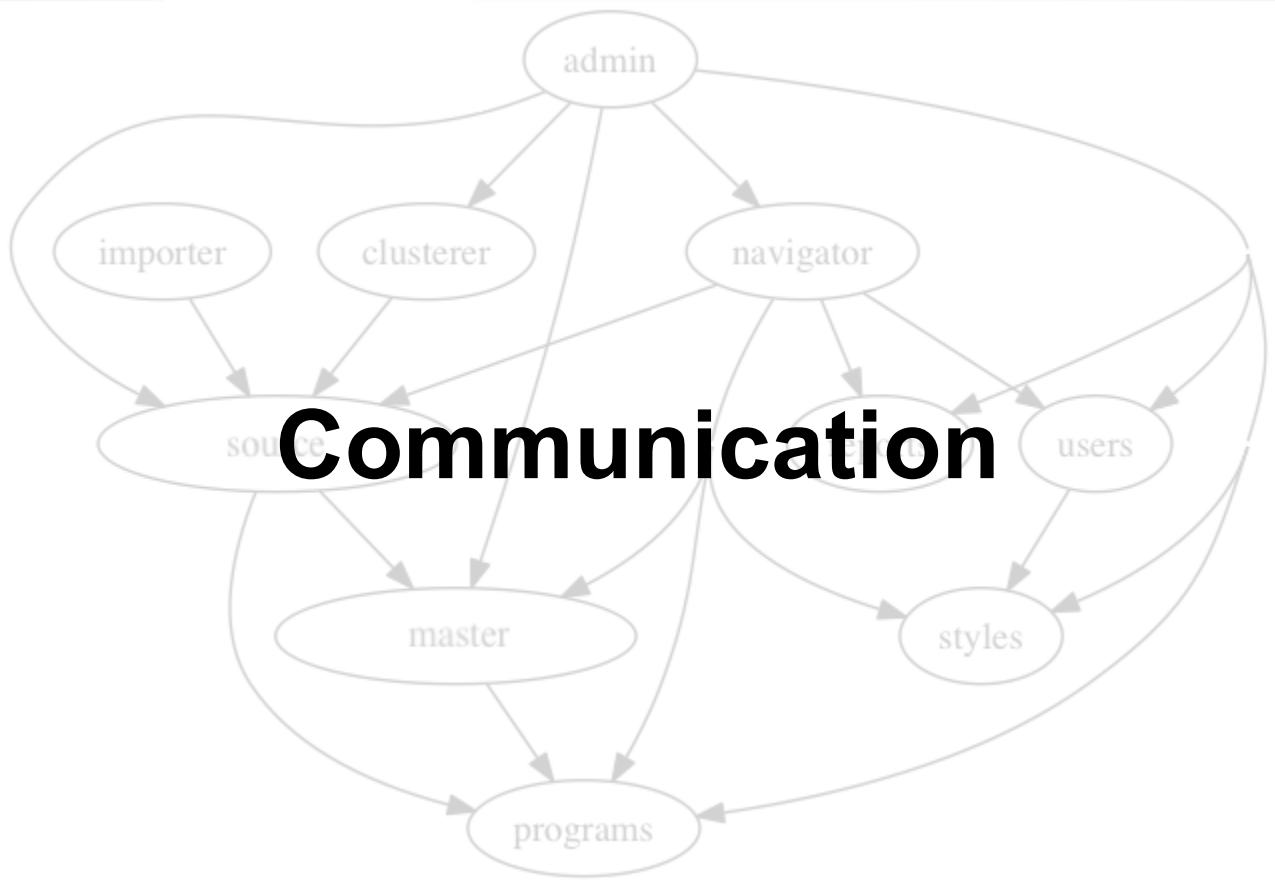
Stephan Hagemann

@shageman
shagemann@pivotal.io
Pivotal Labs

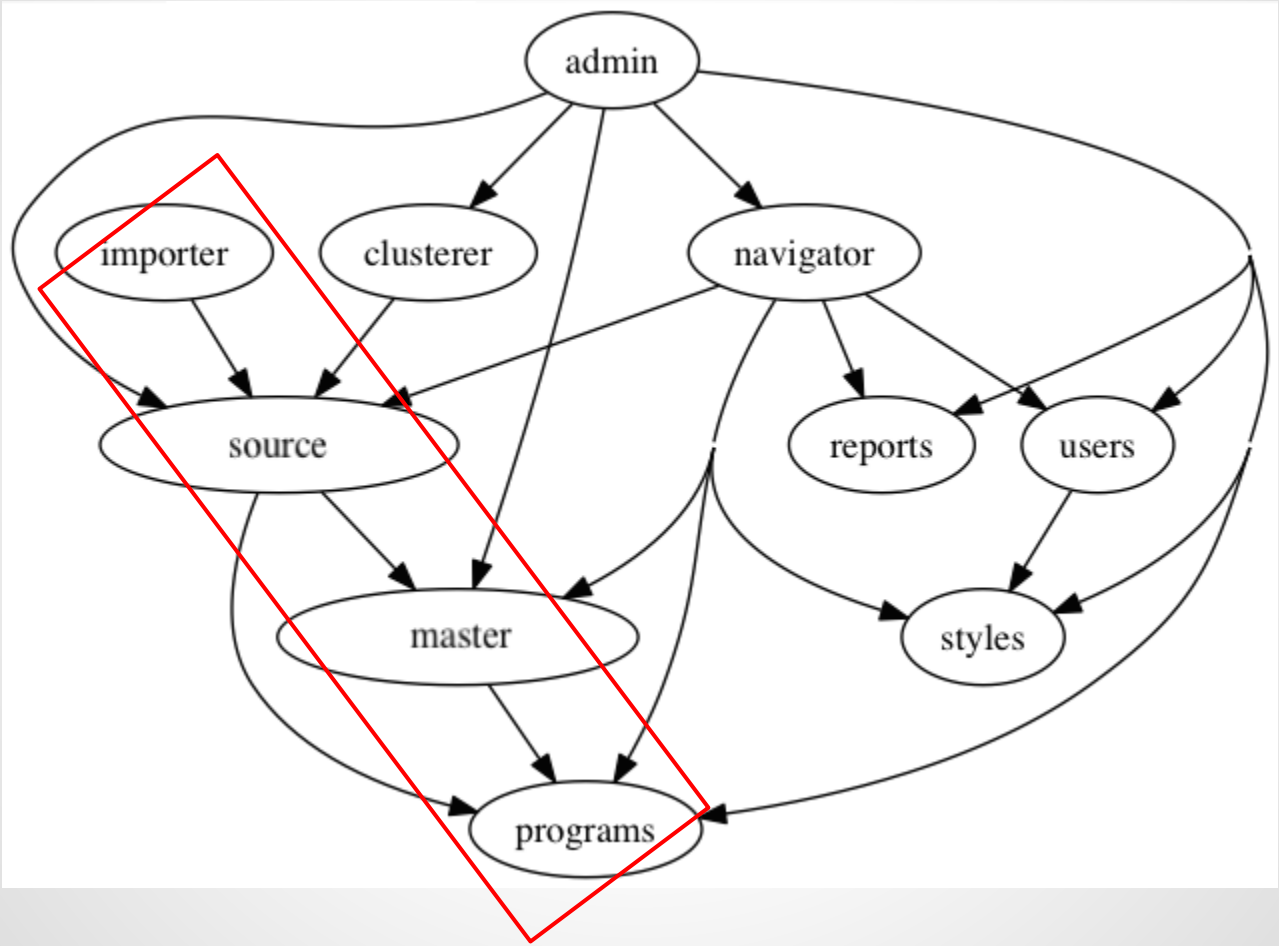
Why #cbra?

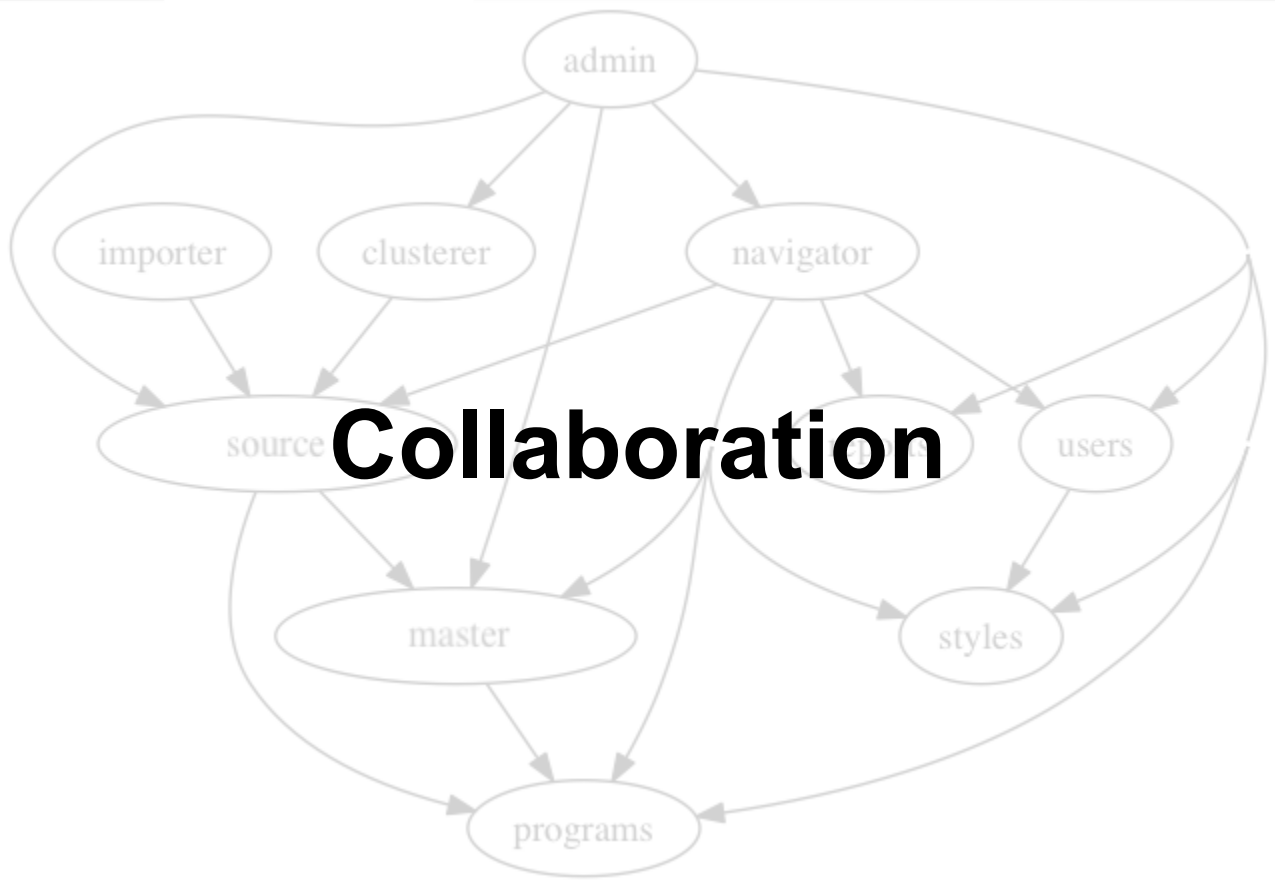
Improved Communication
Improved Collaboration
Improved Creation
Improved Maintenance
Improved Comprehension


admin
importer
clusterer
navigator
source
reports
users
master
styles
programs

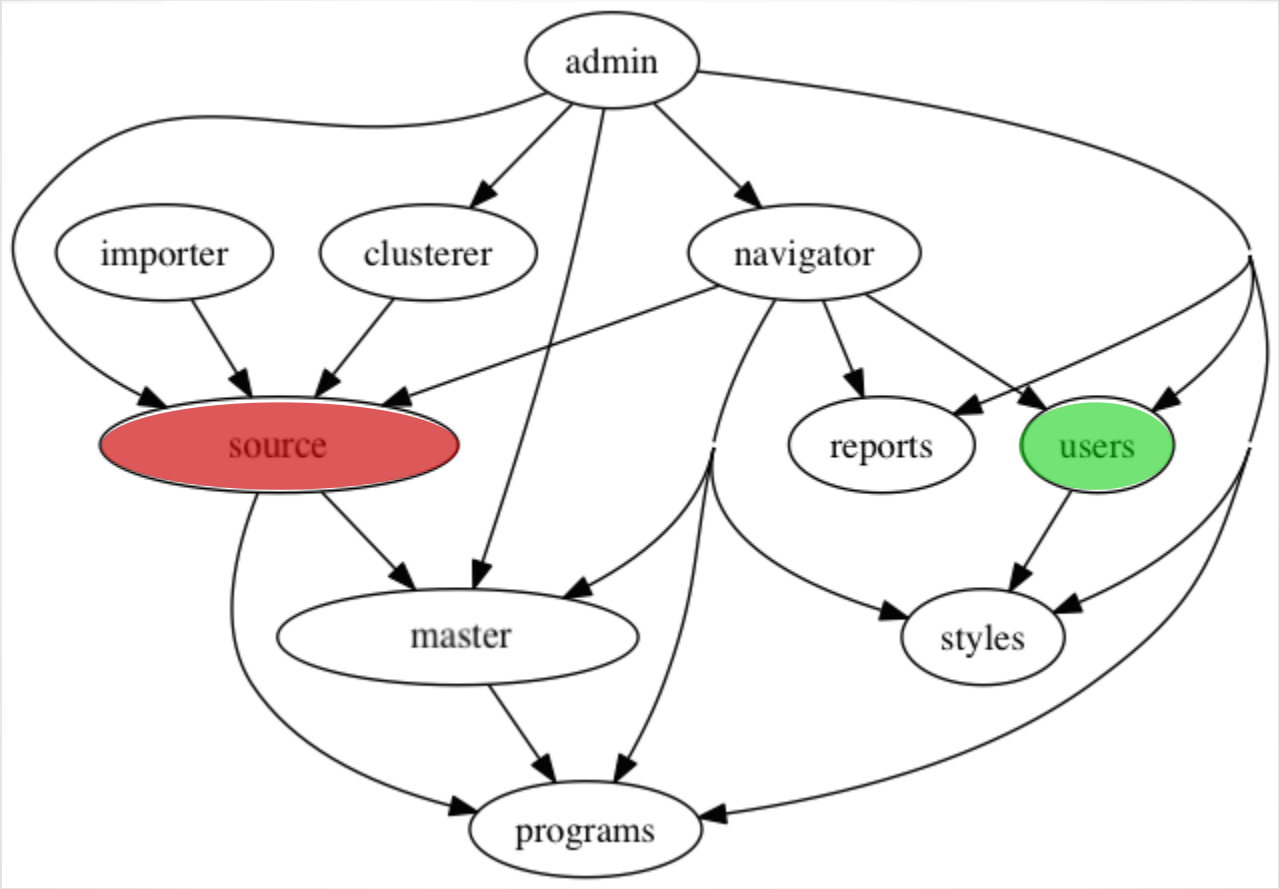


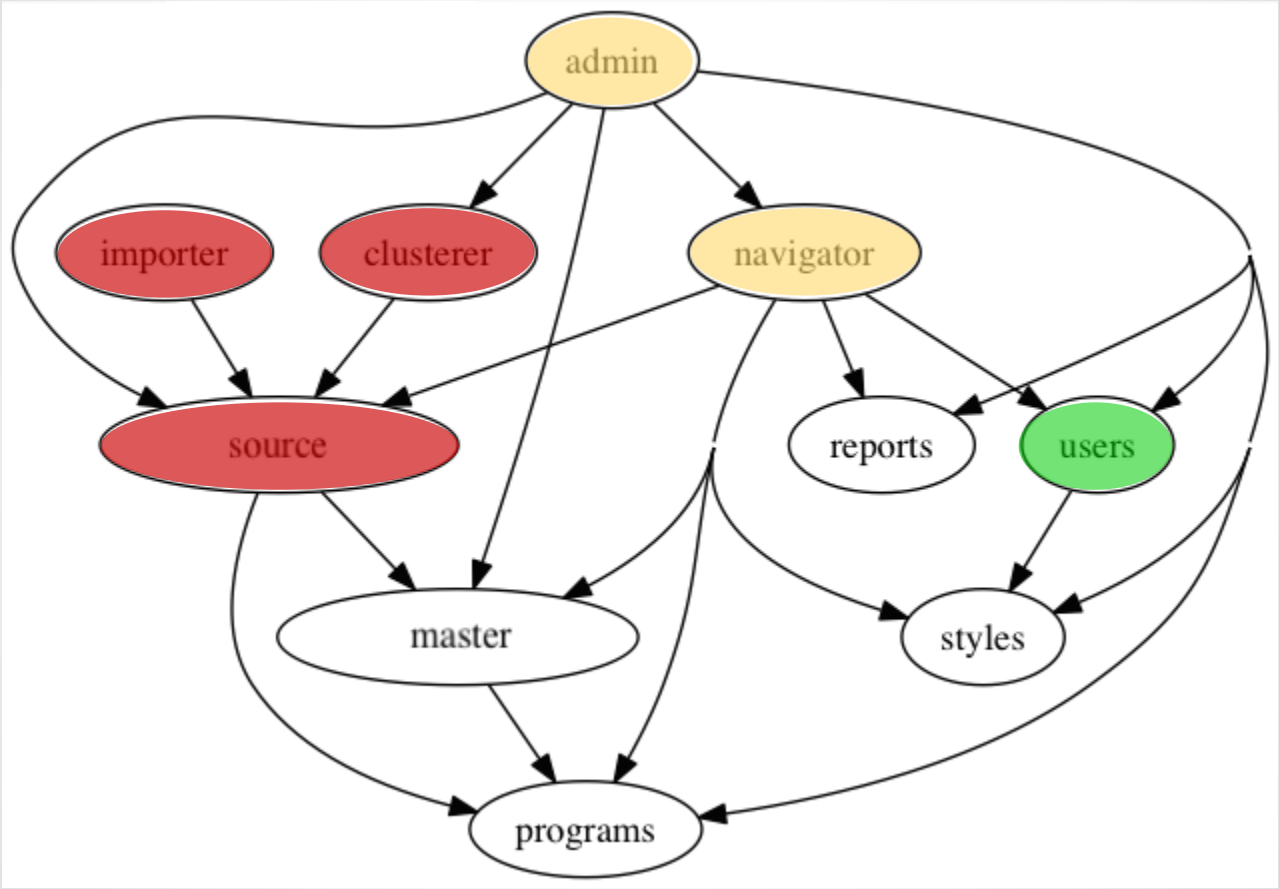
Communication

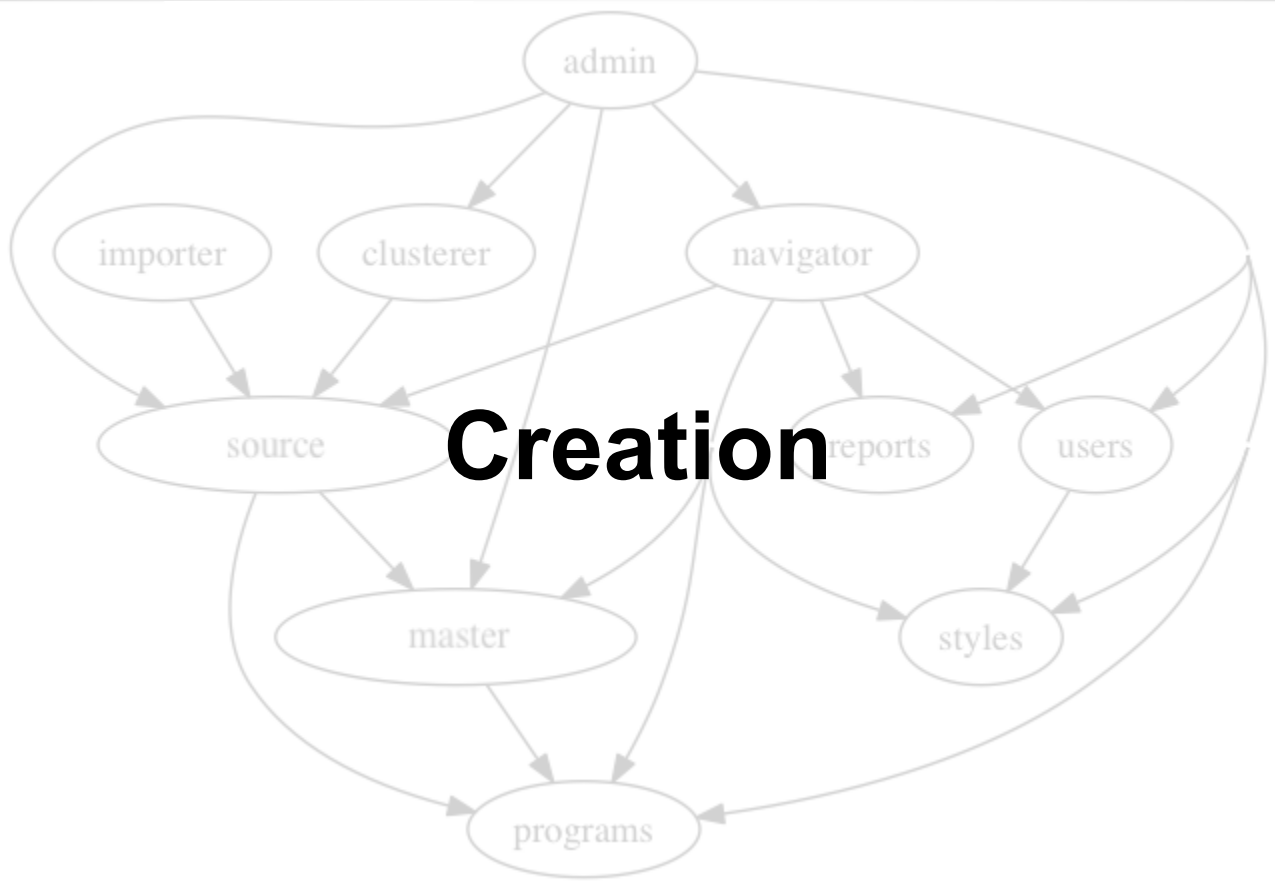

admin
importer
clusterer
navigator
source
reports
users
master
styles
programs



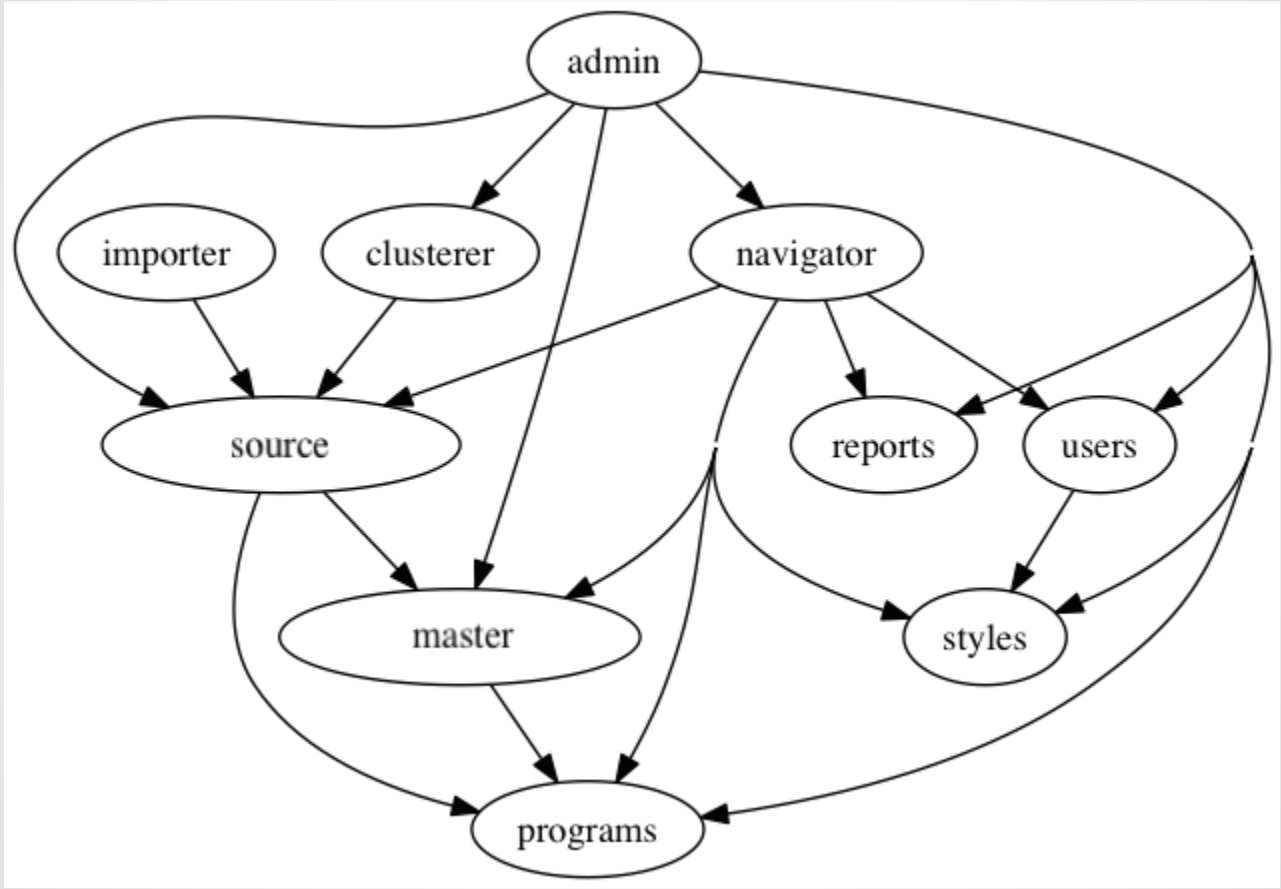
Collaboration


admin
importer
clusterer
navigator
source
reports
users
master
styles
programs


admin
importer
clusterer
navigator
source
reports
users
master
styles
programs

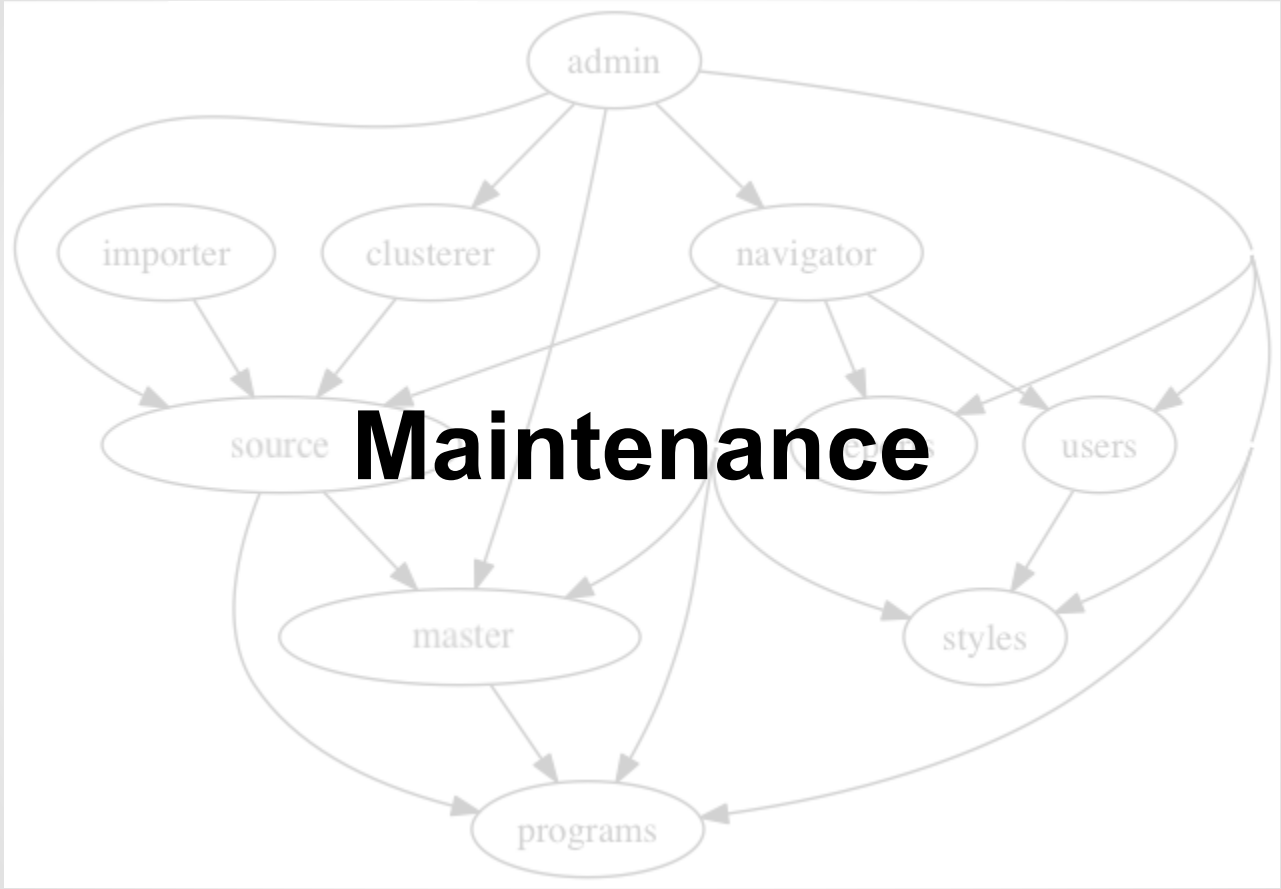

admin
importer
clusterer
navigator
source
reports
users
master
styles
programs

Creation

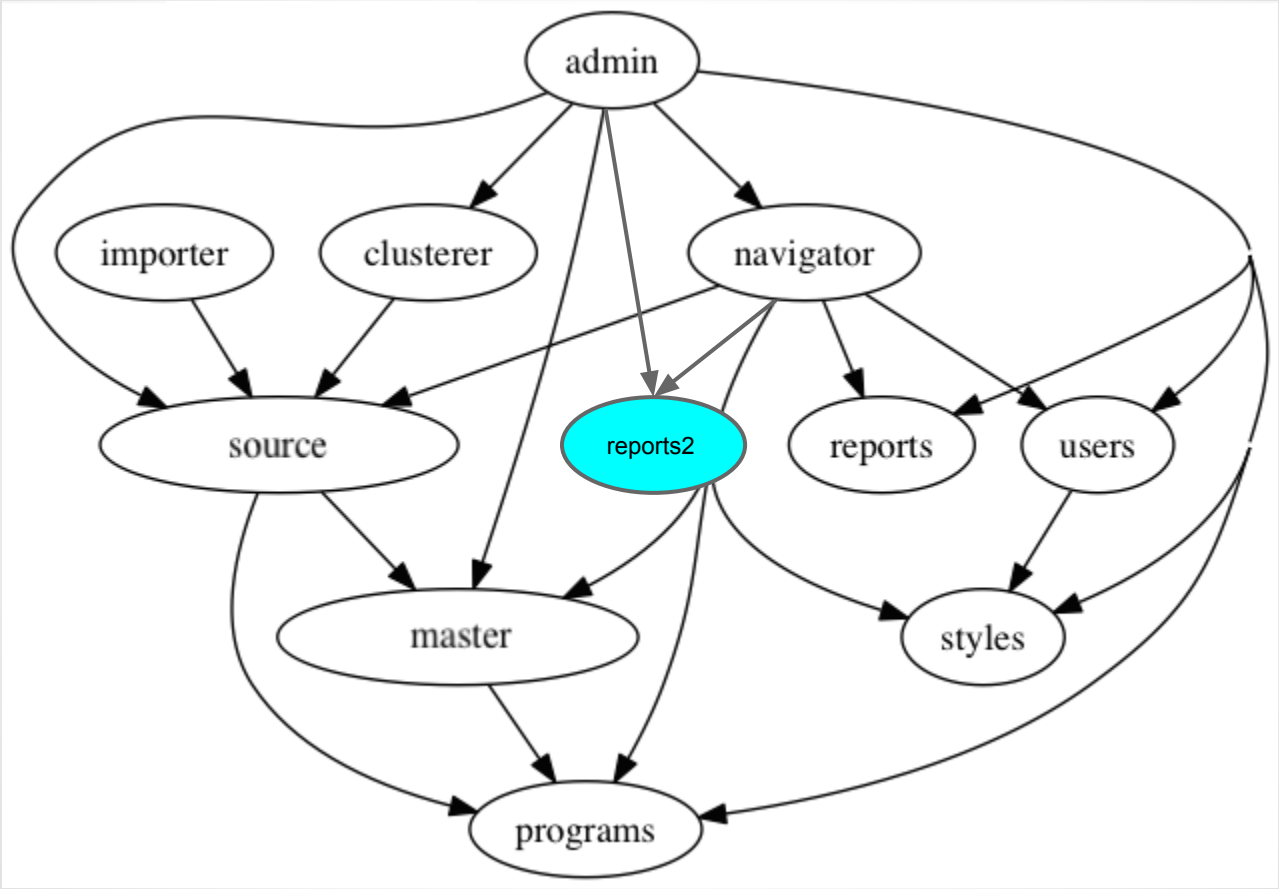


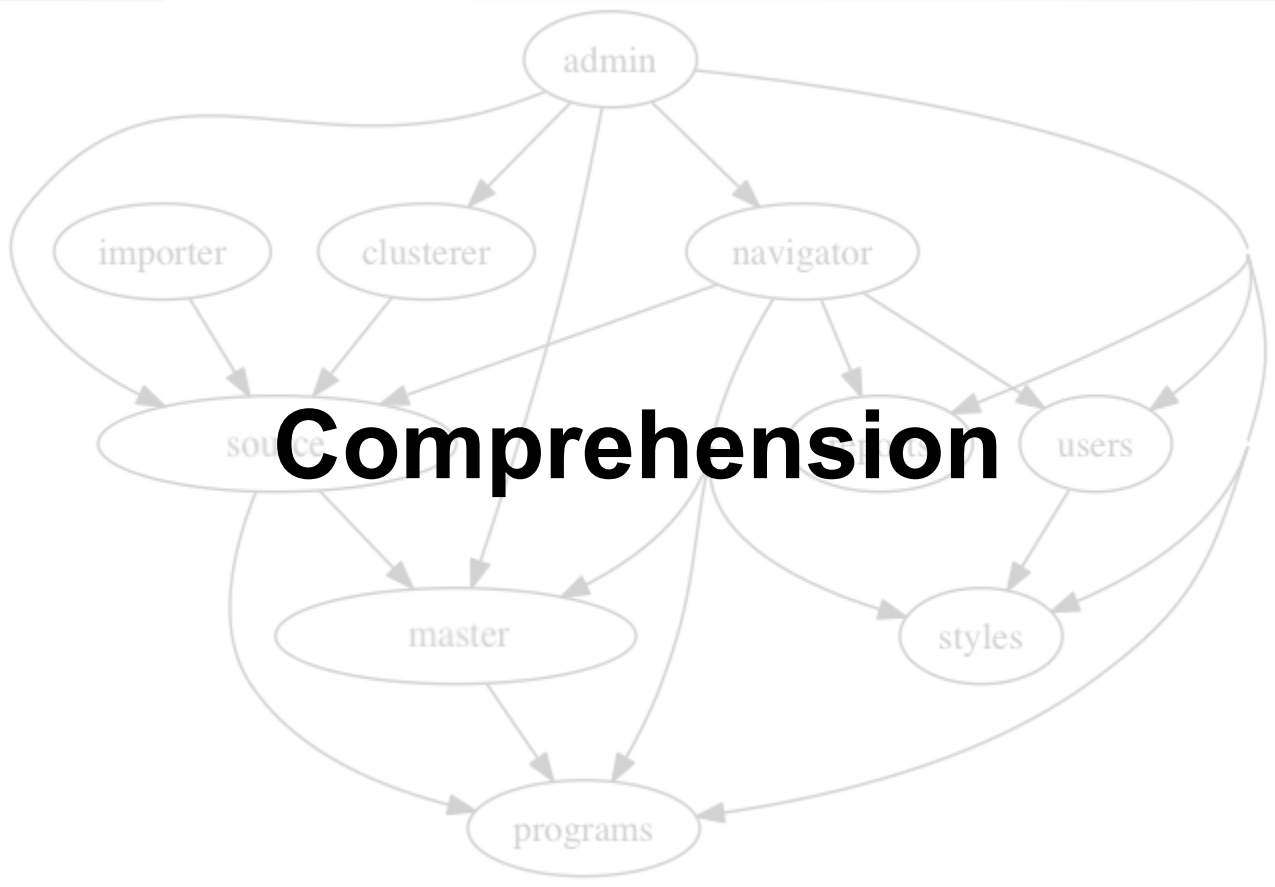
on/off

graphs

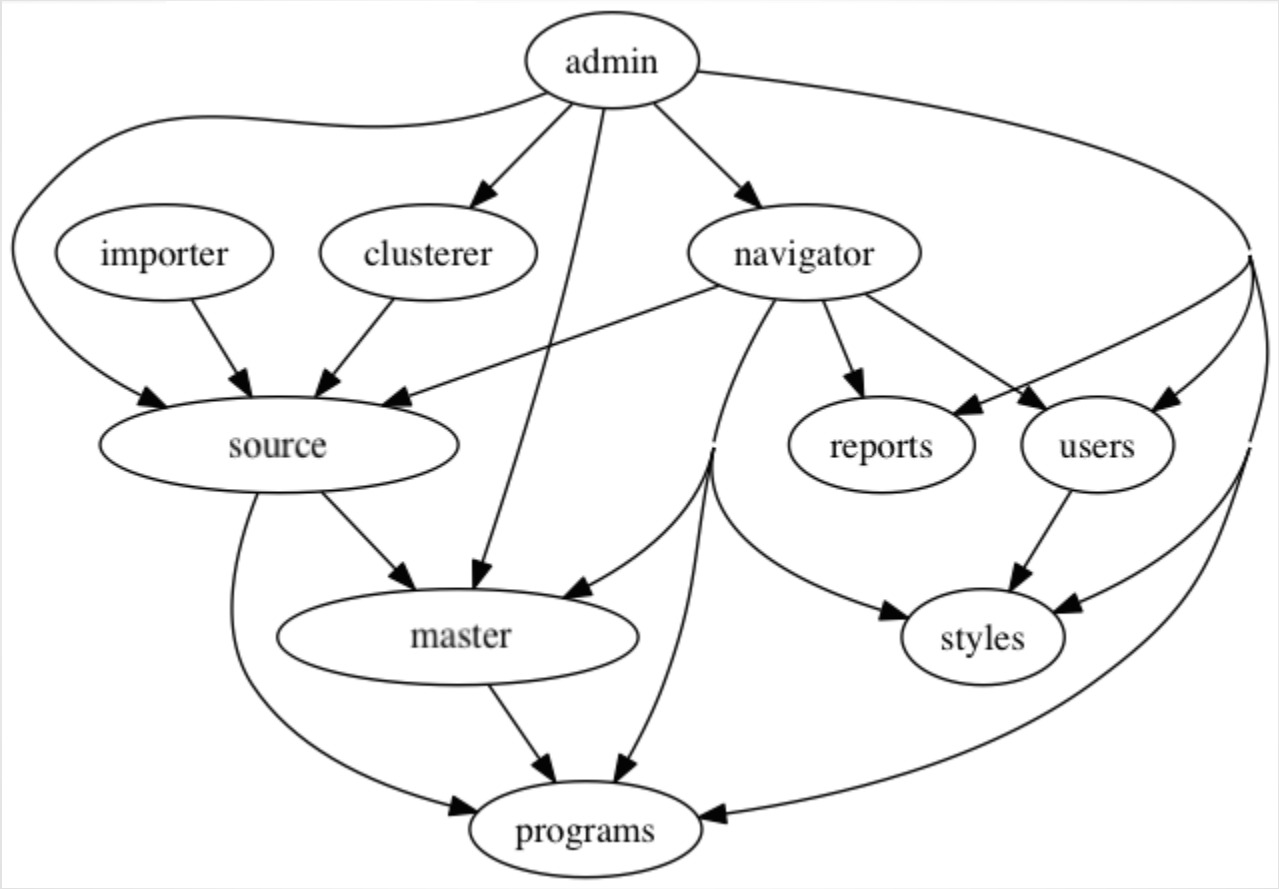


Maintenance


admin
importer
clusterer
navigator
source
reports2
reports
users
master
styles
programs



Comprehension

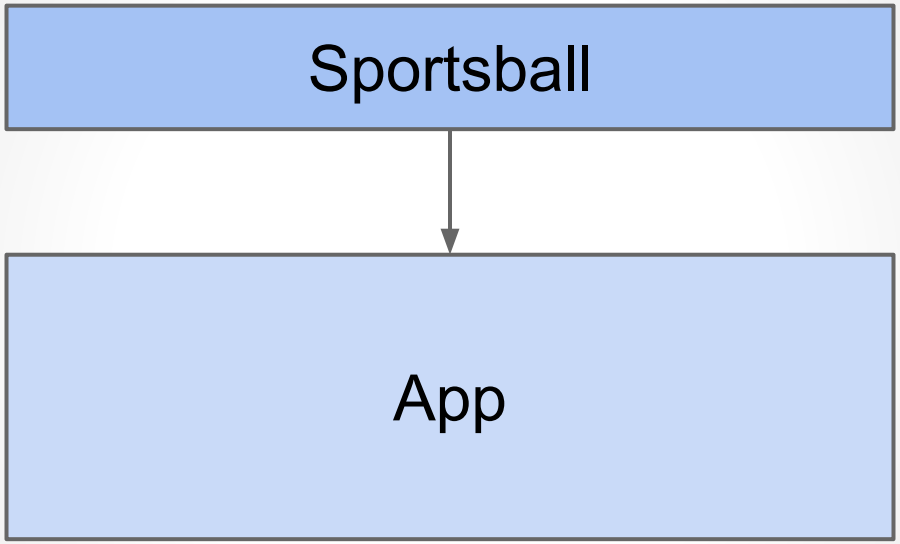

admin
importer
clusterer
navigator
source
reports
users
master
styles
programs


navigator

```
$ git clone https://github.com/shageman/sportsball.git
$ cd sportsball
$ ./build.sh
```

Sportsball - what does it do?

Status Quo



Rails
Engine
Gem

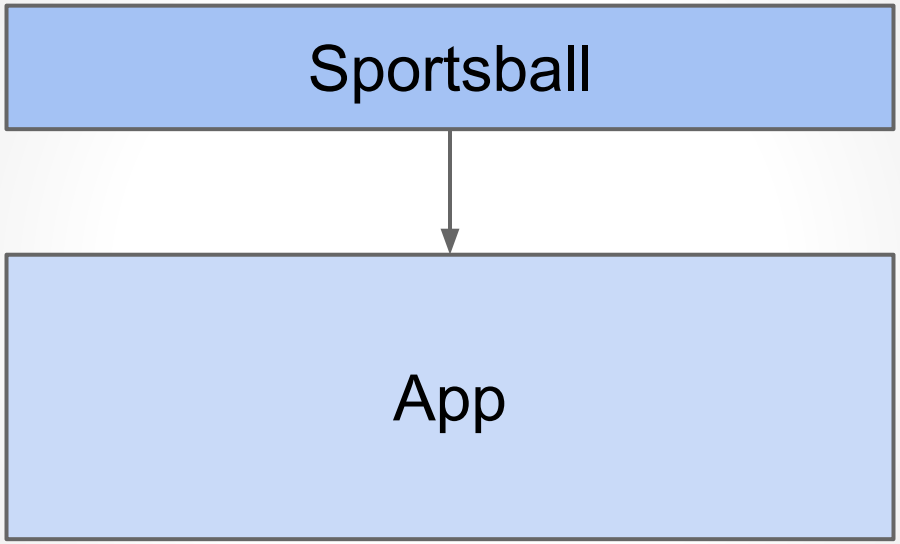
Code Review!

Code Review Recap

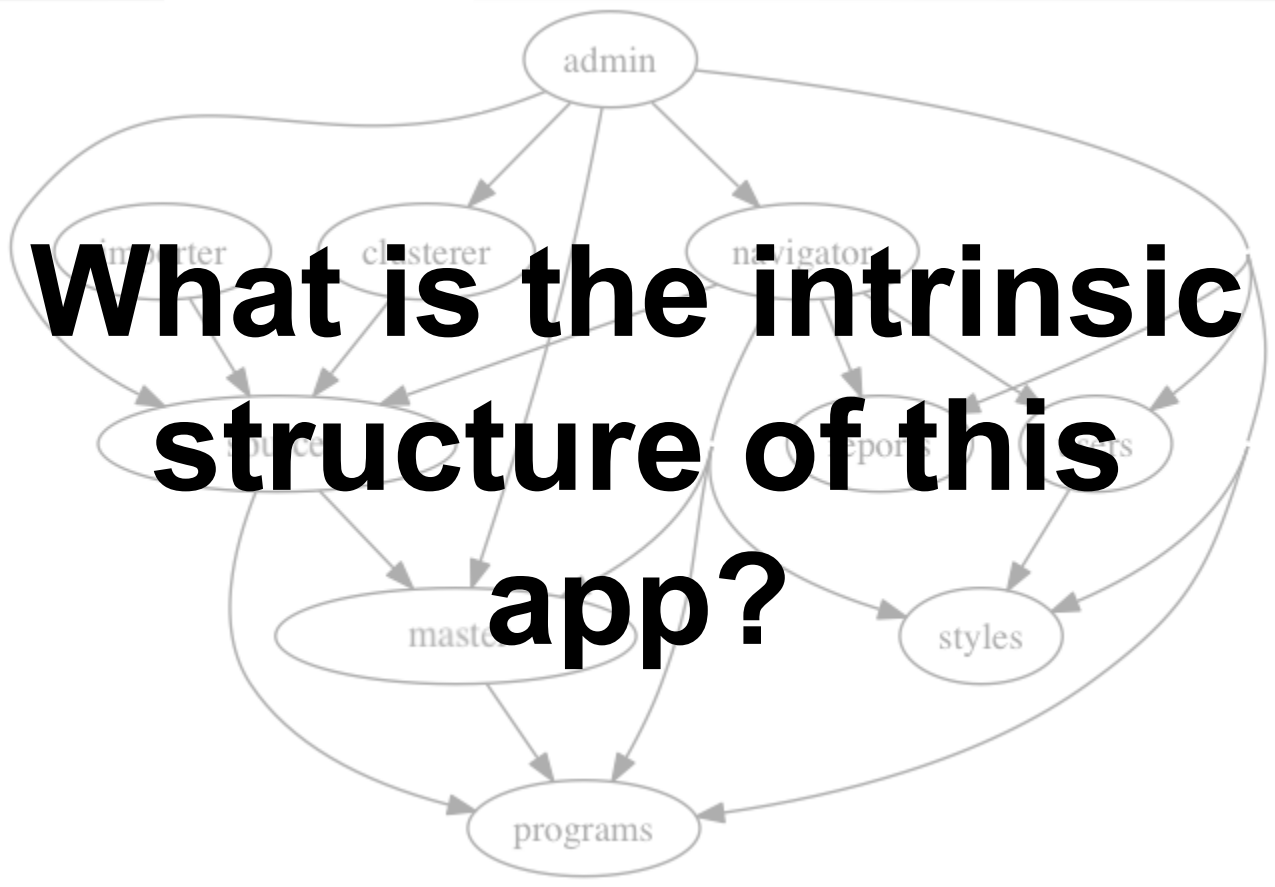
- No app folder!
- App gem loaded through path *(Gemfile)*
- App is an engine *(engine.rb)*
- App is mounted *(config/routes.rb)*
- App routes defined in engine *(app/config/routes.rb)*
- App defines and loads migrations *(app/db/migrate/, engine.rb)*
- Non-published gems are funky *(both Gemfiles)*
- Tests are at the appropriate levels and aggregated via scripts *(build.sh, test.sh)*
- Deployment is unchanged

Finding Components

What should be extracted first?



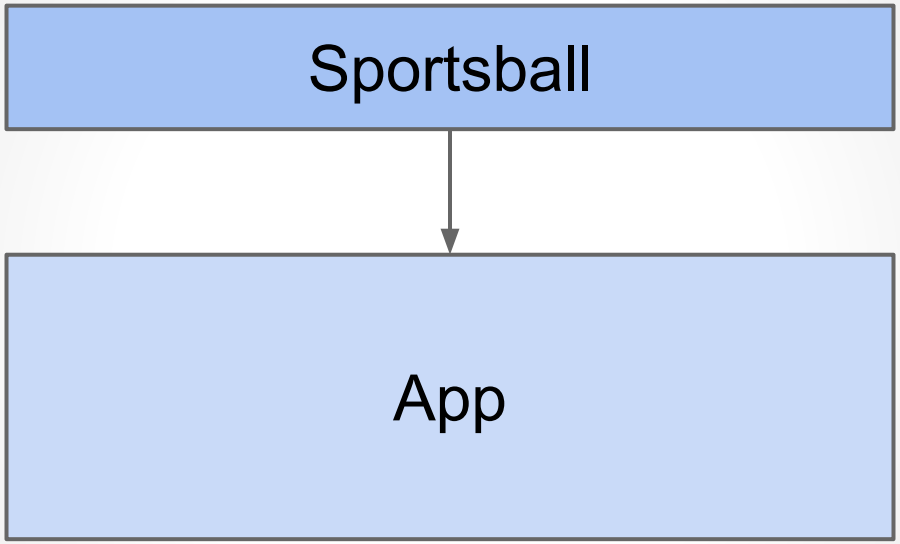
Rails
Engine
Gem



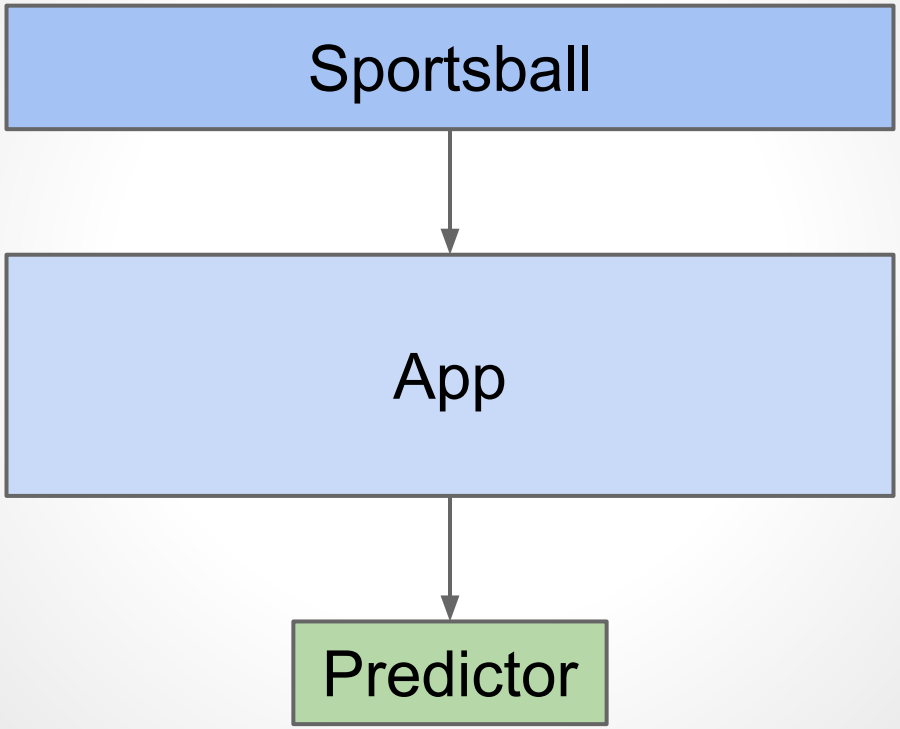
What is the intrinsic structure of this app?

Extract Domain Gem

1st Refactor



Rails
Engine
Gem



Rails
Engine
Gem

DIY - Extract Predictor Gem

```
bundle gem predictor #answer questions
rm -rf predictor/.git*
```

Move classes, move tests

Rename module (in gem)

Fix tests (in gem)

Load gem (in engine)

Fix tests (in engine)

Check build (in main app)

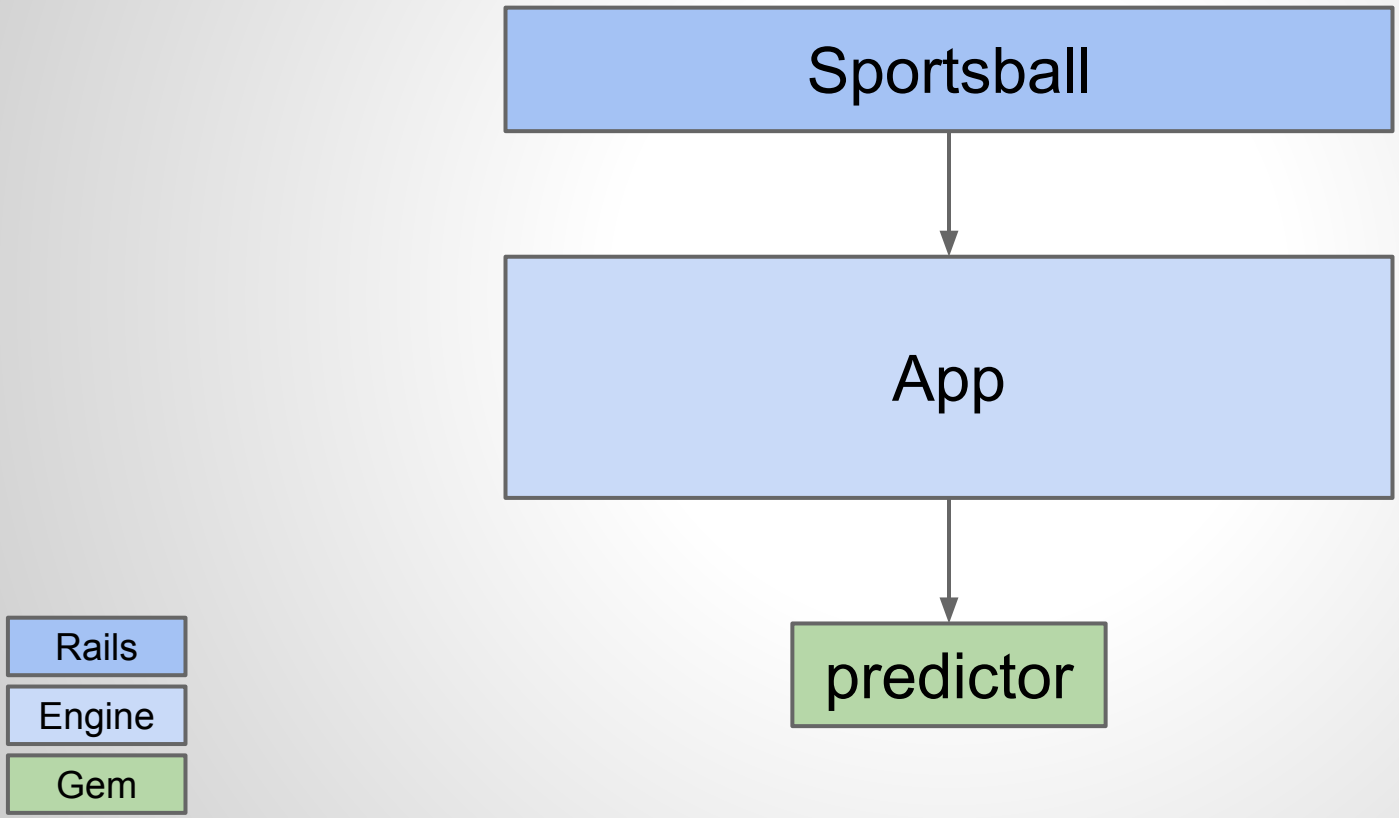
Code Review!

Code Review Recap

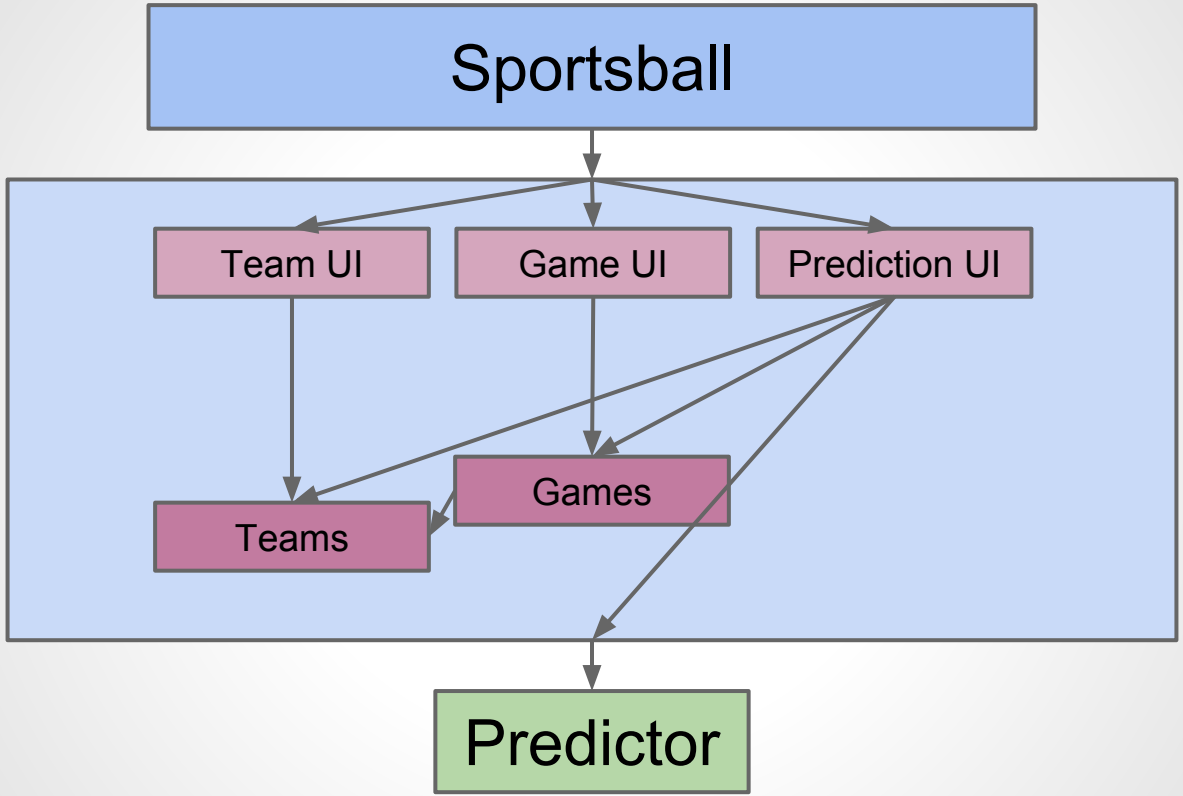
- `path … do` *(Gemfile)*
- Require gem in *(lib/app.rb)*
- Non-test code changes only in Module names *(prediction_controller.rb)*
- OpenStruct App::Team and App::Game *(predictor_spec.rb)*
- Test runner is simplified *(predictor/test.sh)*

Lessons Learned

- Have good tests around the refactor
- Keep running those tests!
- Run tests inside out (up the dependency graph)
- Reduce component dependencies as much as possible
  - Simplifies component
  - Fewest possible reasons to change
  - Keeps dependency graph flat


Sportsball
App
predictor
Rails
Engine
Gem

What else is in this app?



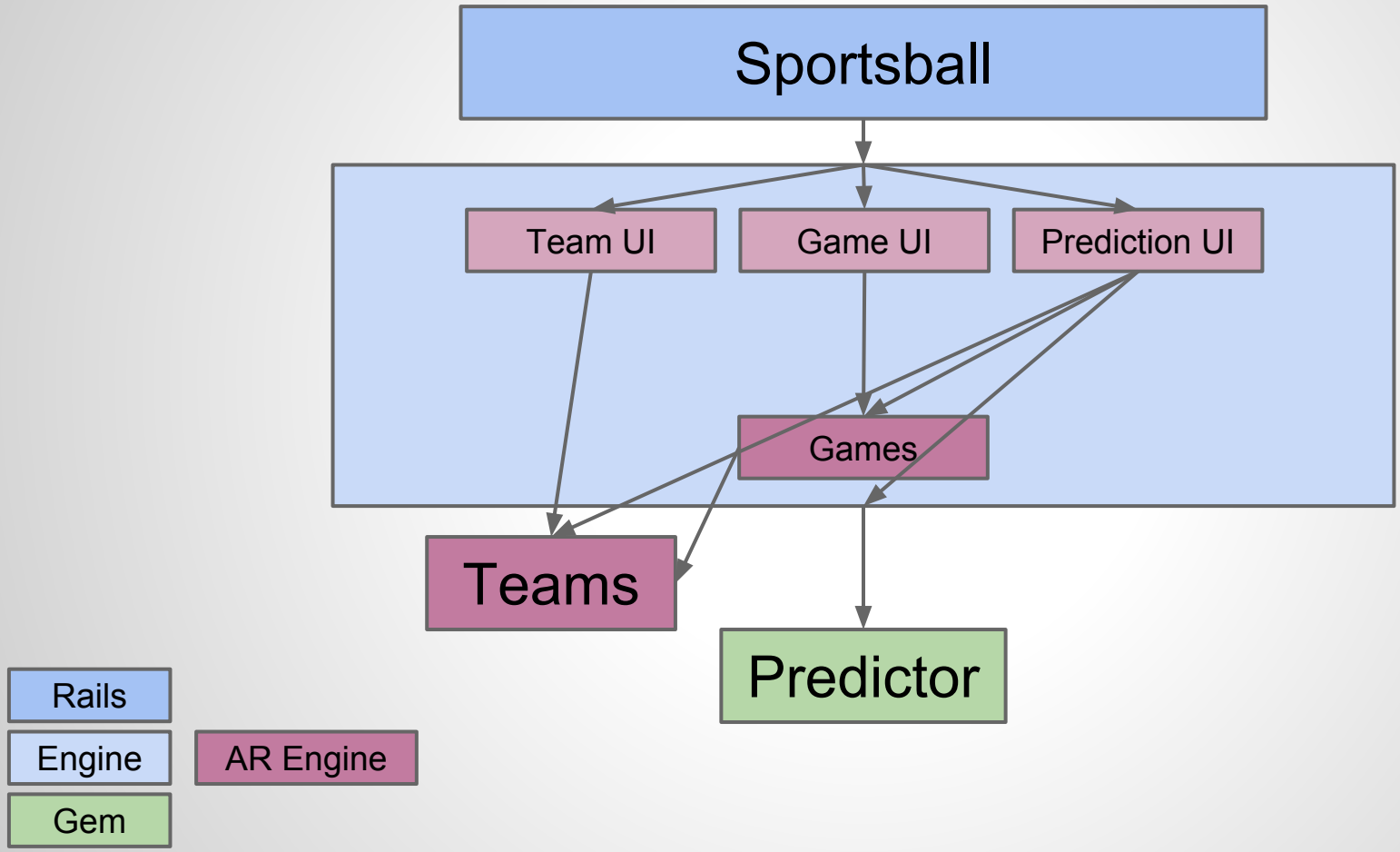
Rails

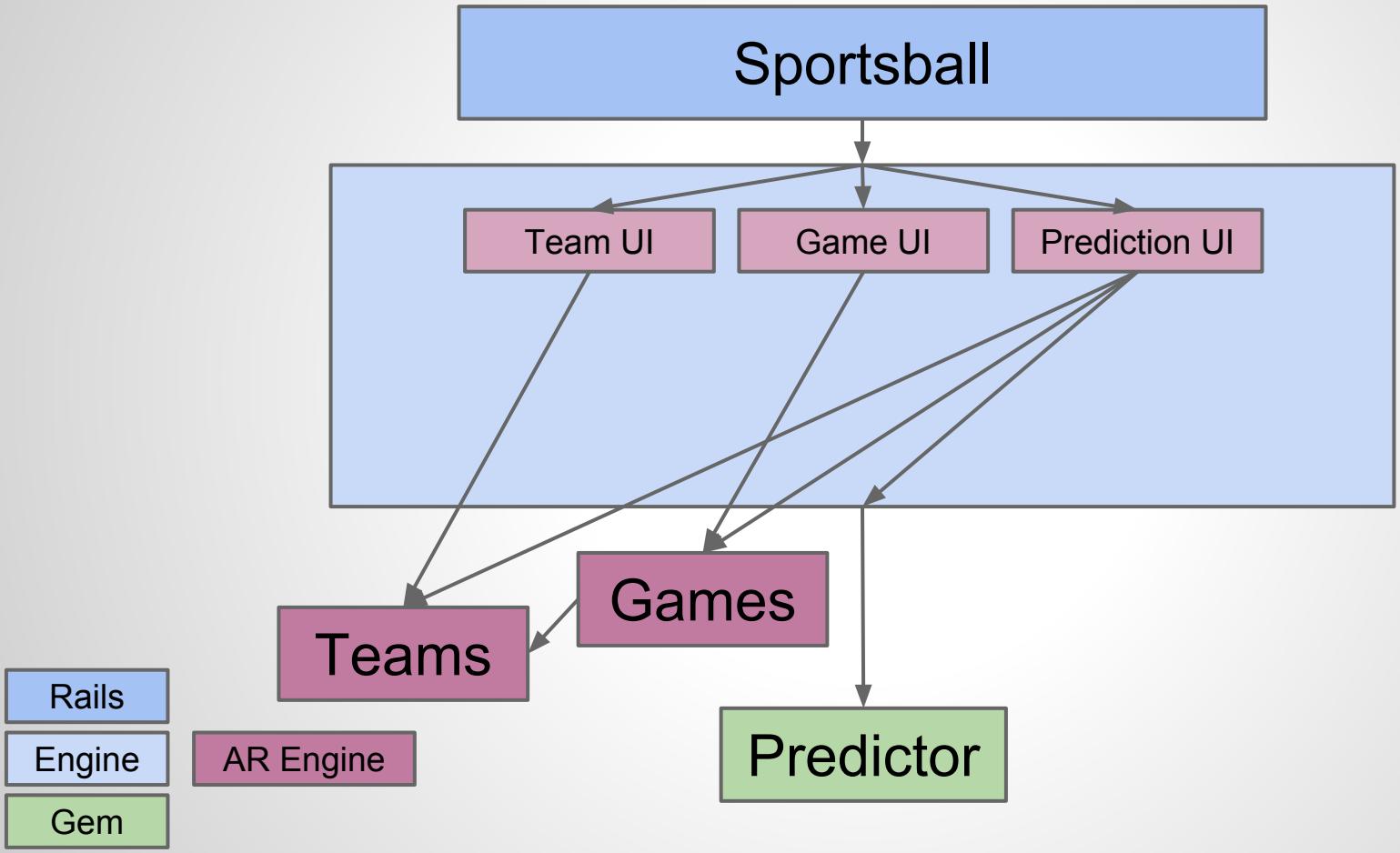
Engine

Gem

Push Persistence Down

2nd & 3rd Refactor


Sportsball
Team UI
Game UI
Prediction UI
Games
Teams
Predictor
Rails
Engine
AR Engine
Gem


Sportsball
Team UI
Game UI
Prediction UI
Games
Teams
Predictor
Rails
Engine
Gem
AR Engine

DIY - Push Teams Persistence Down

```
rails plugin new --help
rails plugin new teams -BGVSJT --mountable \
    --dummy-path=spec/dummy
```

Move classes, move tests, move migration

Create table renaming migration (in teams engine)

Rename module (in teams engine)

Fix tests (in teams engine)

Load gem (in app engine)

Create testhelper (in teams engine) and use (in app engine)

Fix tests (in app engine)

Check build (in main app)

Code Review!

Code Review Recap

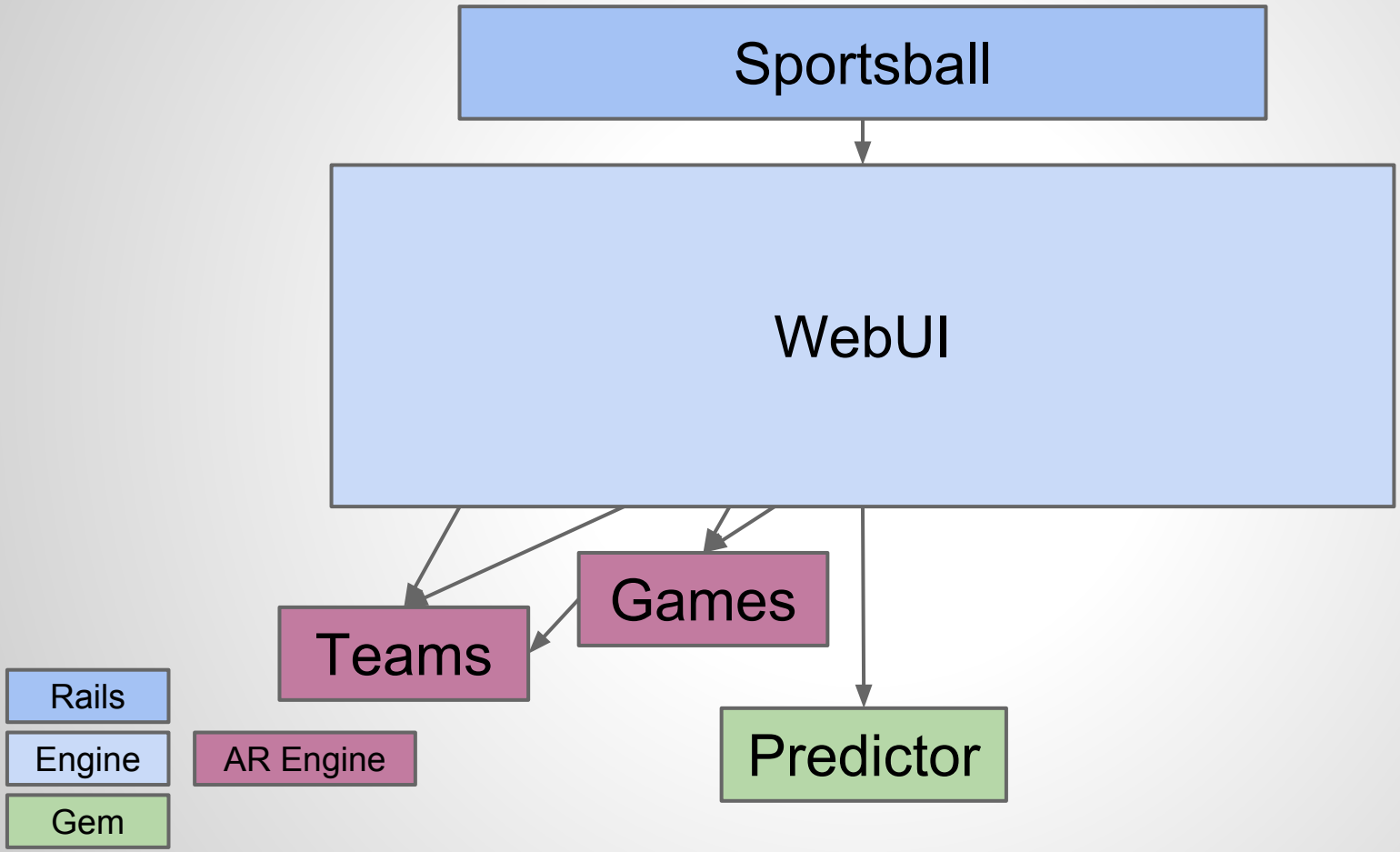
- Dummy app *(teams/spec/dummy)*
- String renames of AR associations! *(app/game.rb)*
- Teams test helper
  - Create test helper *(teams/spec/support/object_creation_methods.rb)*
  - Expose test helper *(teams/lib/teams/test_helpers.rb)*
  - Use teams test helper *(app/spec/spec_helper.rb)*

Lessons Learned

- Two ways of engine-out-of-engine extraction
  - Create fresh + move + rename
  - Duplicate + delete + rename
- Don’t mess with existing migrations!
  - Move affected existing migrations
  - Create new migration for table rename
- Test helpers nice for ActiveRecord models
- “App” is a terrible name to refactor away from

Better Naming of Component

4th Refactor


Sportsball
WebUI
Games
Teams
Predictor
Rails
Engine
Gem
AR Engine

Side note: If you feel like naming your engine...

Common

Base

Misc

General

Lib

<My Company name>

Here are some alternatives

Everything

Cruft

Random

Don’t Know

Don’t Care

Duh

Here are some alternatives (Proper module names)

Everything

Cruft

Random

DontKnow

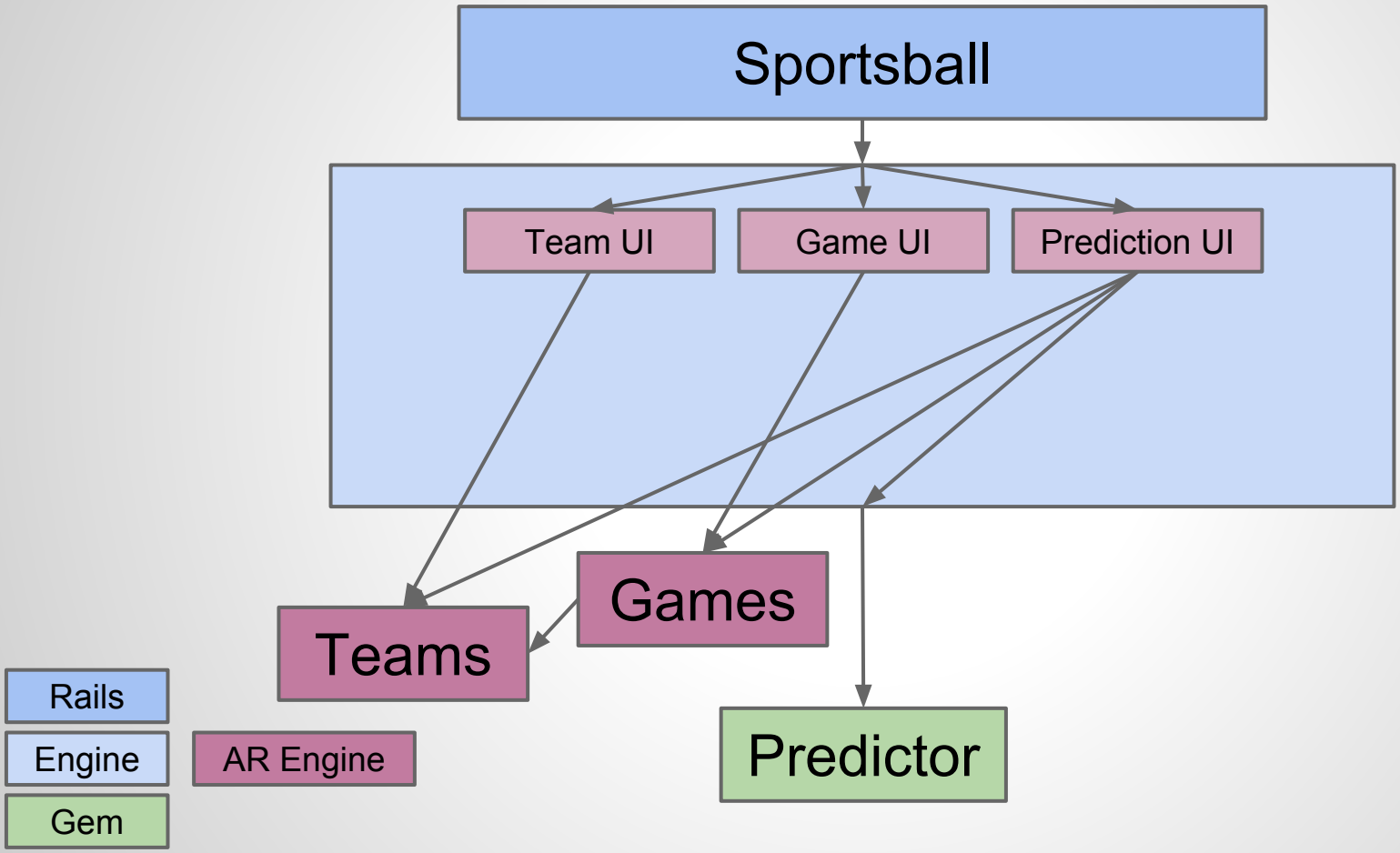
DontCare

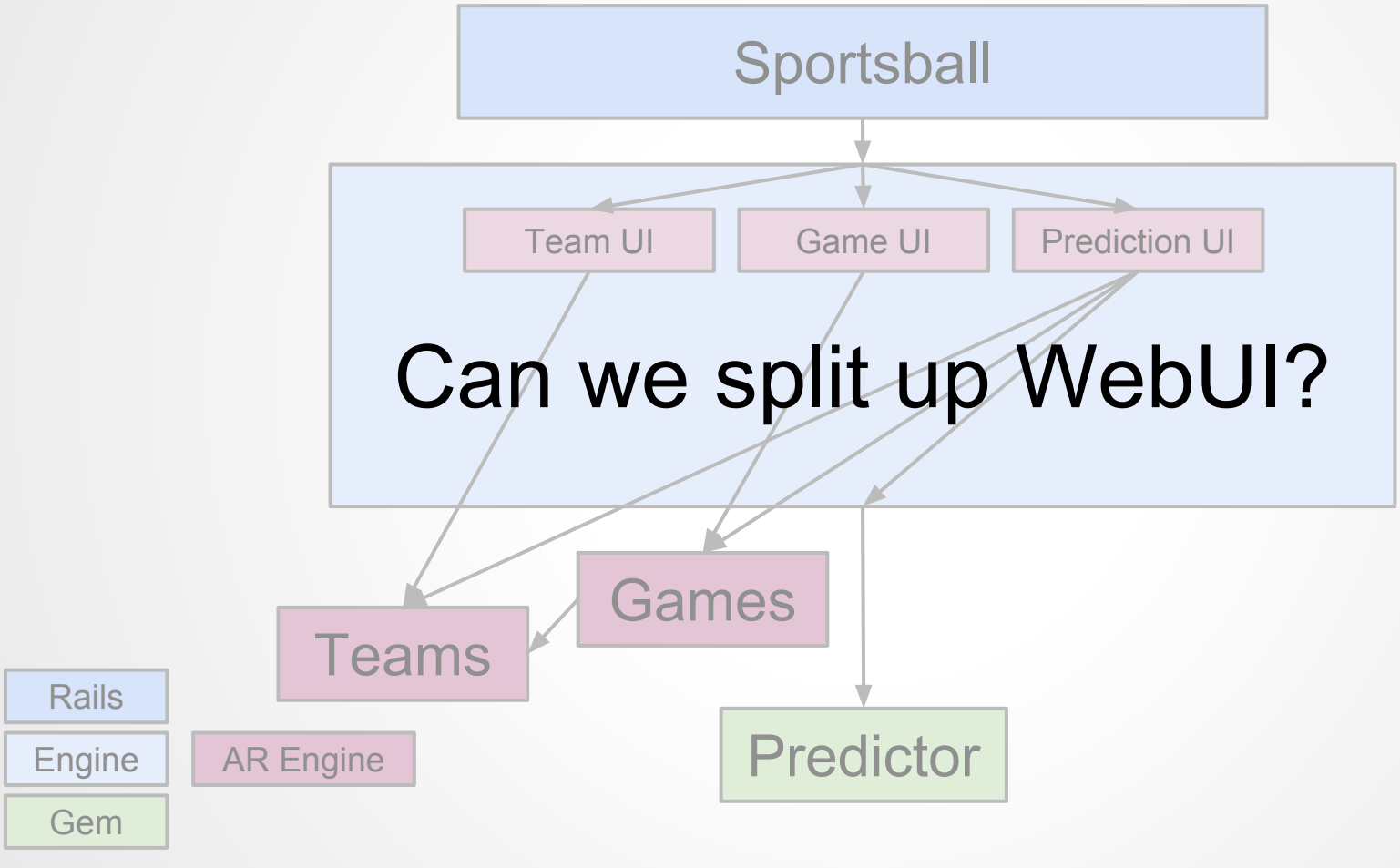
Duh

Give the **most specific** name you can think of

refactor a lot

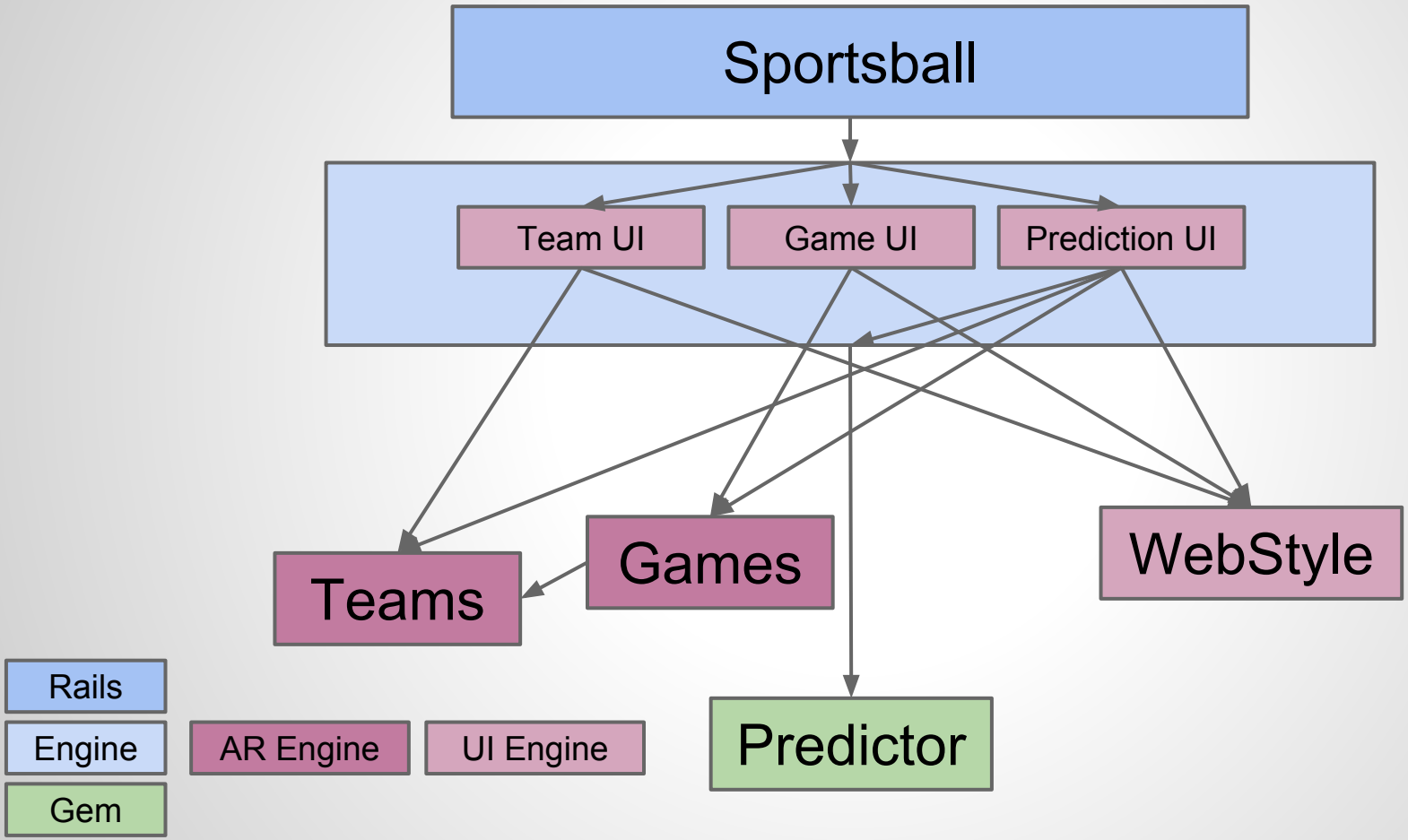
continue the **conversation**


Sportsball
Team UI
Game UI
Prediction UI
Games
Teams
Predictor
Rails
Engine
Gem
AR Engine


Sportsball
Team UI
Game UI
Prediction UI
Can we split up WebUI?
Games
Teams
Predictor
Rails
Engine
Gem
AR Engine

Extract Layout and Style

5th Refactor


Sportsball
Team UI
Game UI
Prediction UI
Teams
Games
WebStyle
Predictor
Rails
Engine
Gem
AR Engine
UI Engine

DIY - Extract WebStyle

Copy WebUI into WebStyle

Remove everything but assets/layout - don’t forget vendor! *(in WebStyle)*

Remove assets *(in WebUI)*

Fix asset loading *(in WebUI)*

Check build *(in main app)*

Code Review!

Code Review Recap

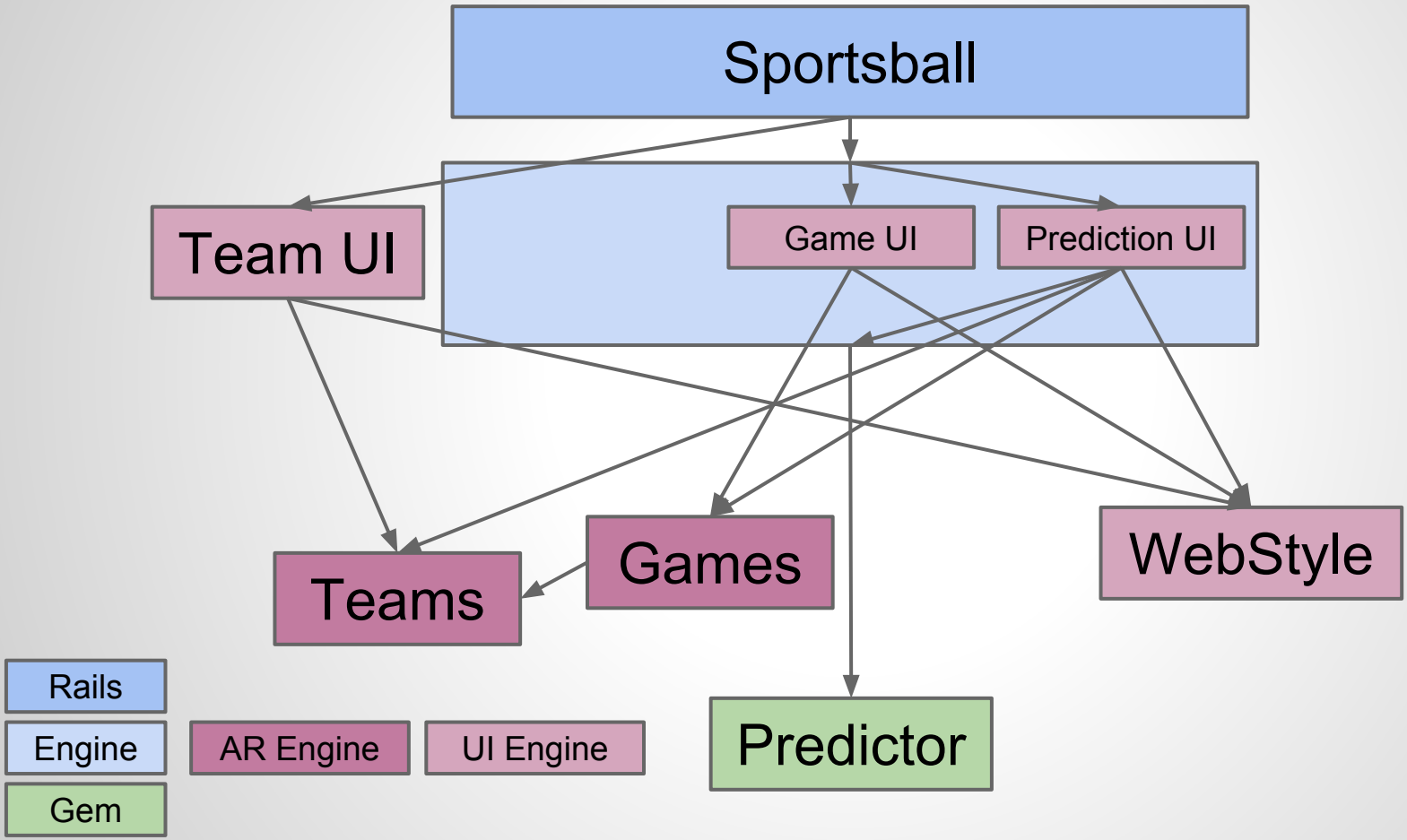
- No tests in WebStyle
- Application layout *(web_style/application.html.slim)*
- Layout selection *(web_ui/application_controller.rb)*

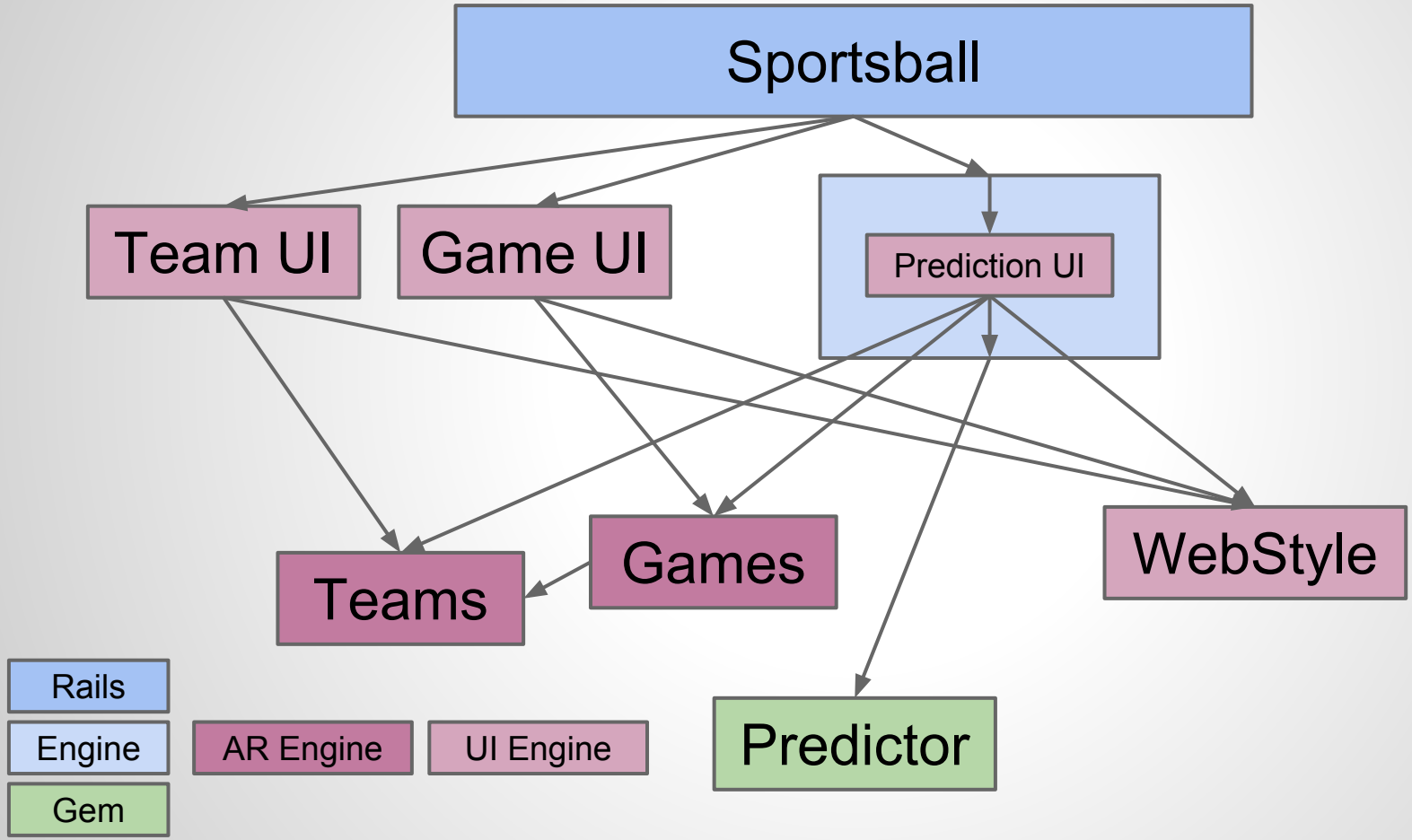
Lessons Learned

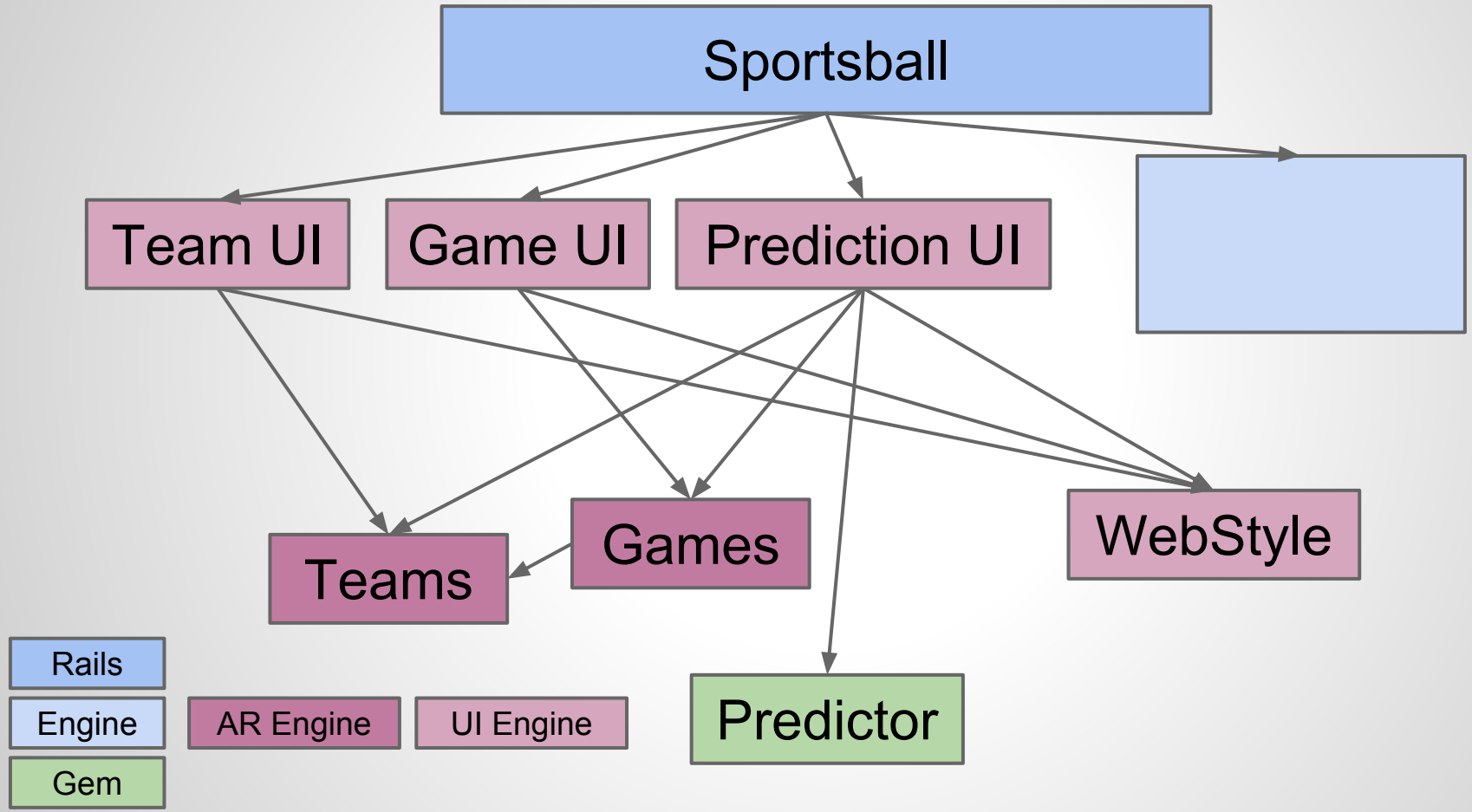
- You won’t know what components you will need to extract until you try to extract something

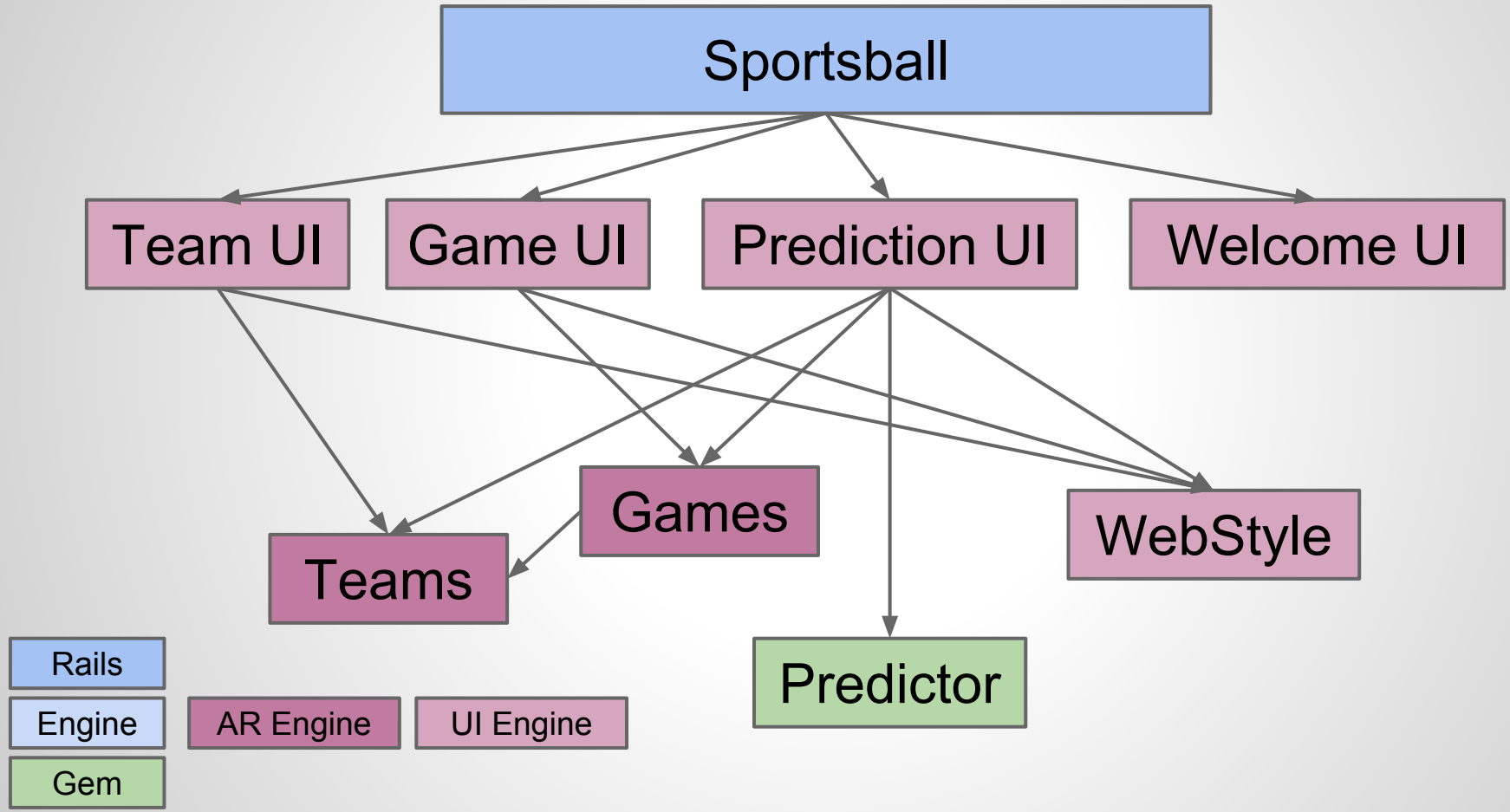
Separate independent UIs

6th, 7th, 8th Refactor


Sportsball
Team UI
Game UI
Prediction UI
Teams
Games
WebStyle
Predictor
Rails
Engine
Gem
AR Engine
UI Engine


Sportsball
Team UI
Game UI
Prediction UI
Teams
Games
WebStyle
Predictor
Rails
Engine
Gem
AR Engine
UI Engine


Sportsball
Team UI
Game UI
Prediction UI
Teams
Games
WebStyle
Predictor
Rails
Engine
Gem
AR Engine
UI Engine


Sportsball
Team UI
Game UI
Prediction UI
Welcome UI
Games
Teams
WebStyle
Predictor
Rails
Engine
Gem
AR Engine
UI Engine

DIY - Split off a UI Engine

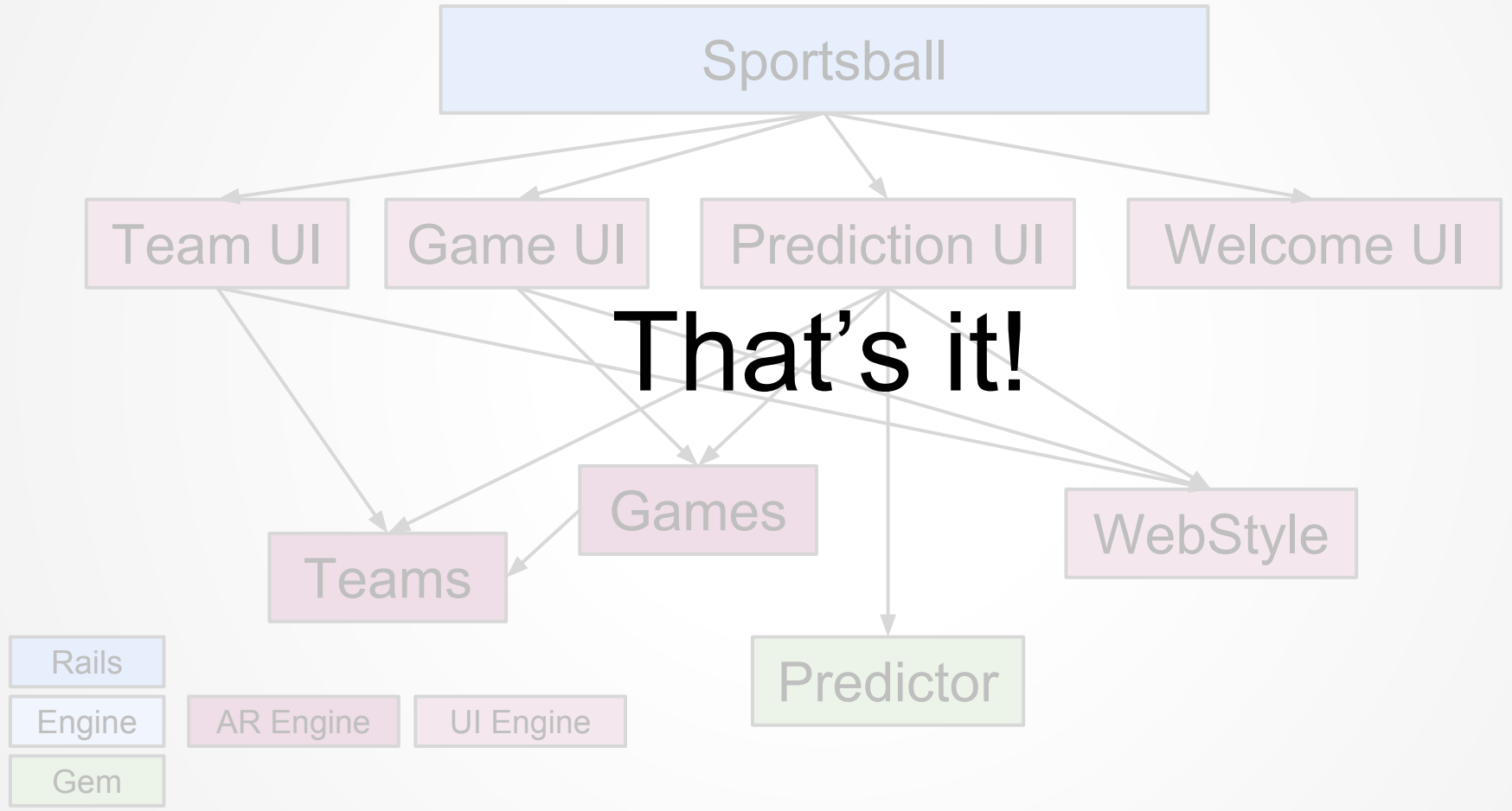
Pick any one of the potential UI engines

Code Review!

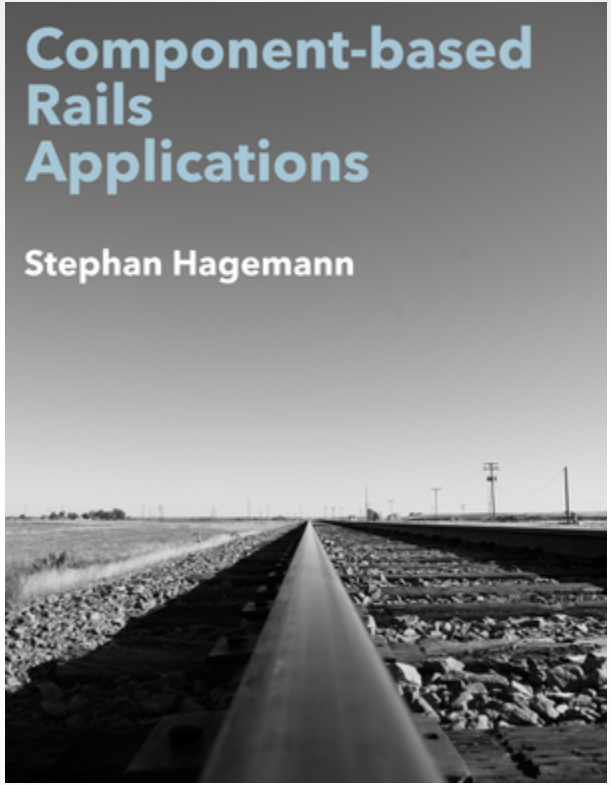
Code Review Recap

- Rerouting necessary
  - App routes (config/routes.rb)
  - Engine routes (*_ui/config/routes.rb)
  - Tests

Lessons Learned


Sportsball
Team UI
Game UI
Prediction UI
Welcome UI
That's it!
Games
Teams
WebStyle
Predictor
Rails
Engine
AR Engine
UI Engine
Gem

Now - Your App!



leanpub.com/cbra/c/railsconf

cbra.info