

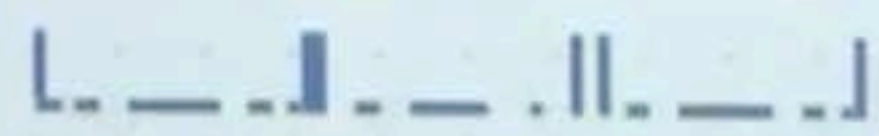
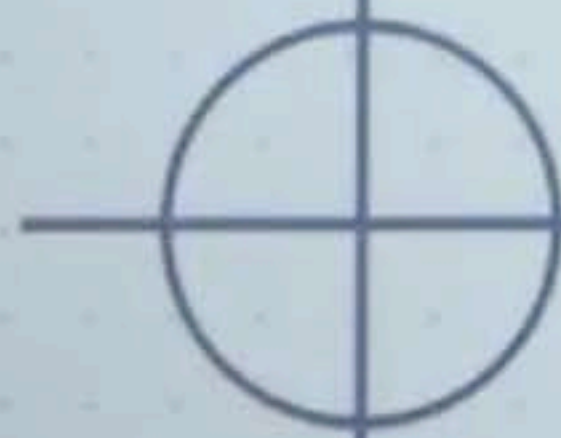


SCALING YOUR RAILS APP CODEBASE

C.B.R.A.



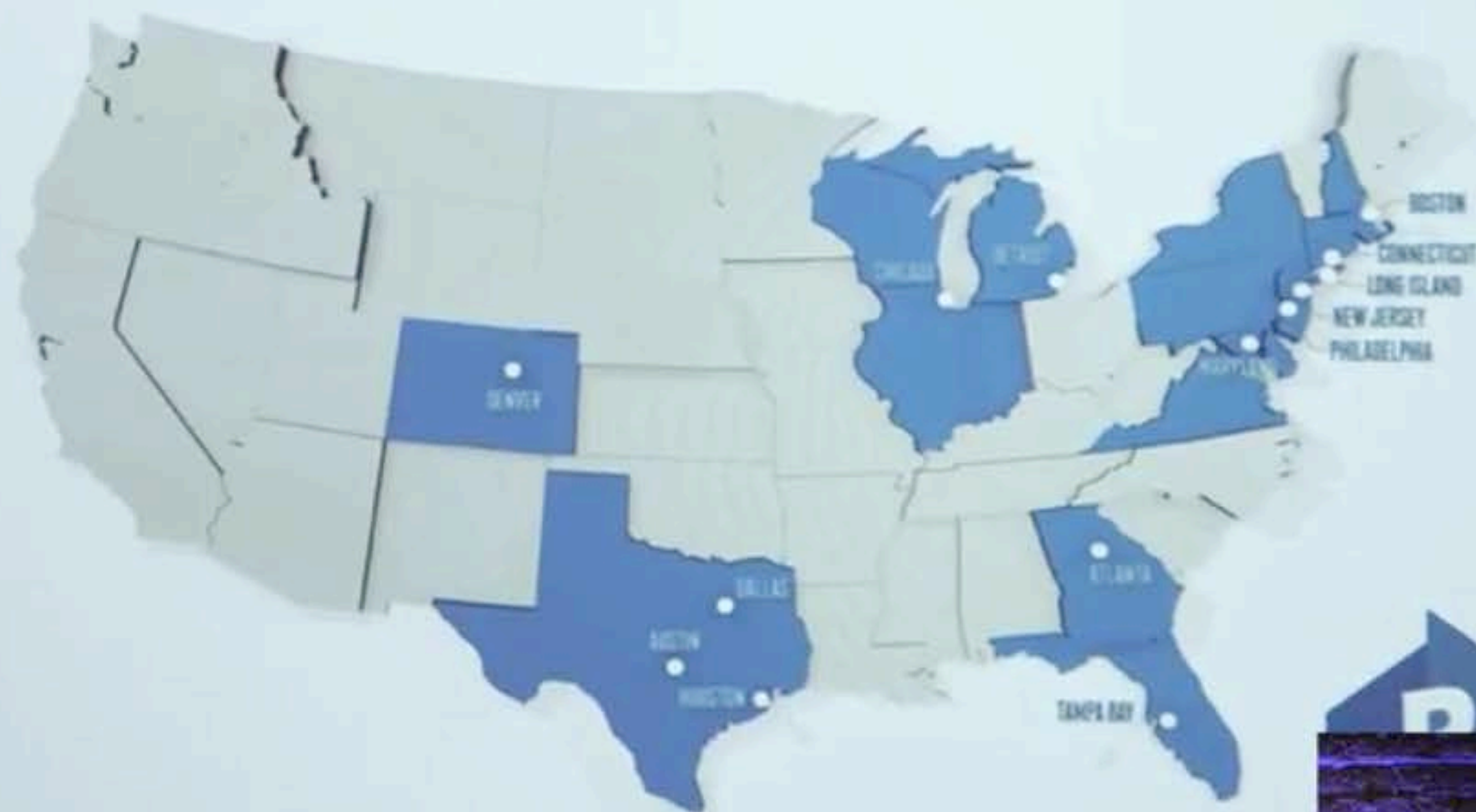
COMPONENT-BASED RAILS ARCHITECTURE



Ben Klang



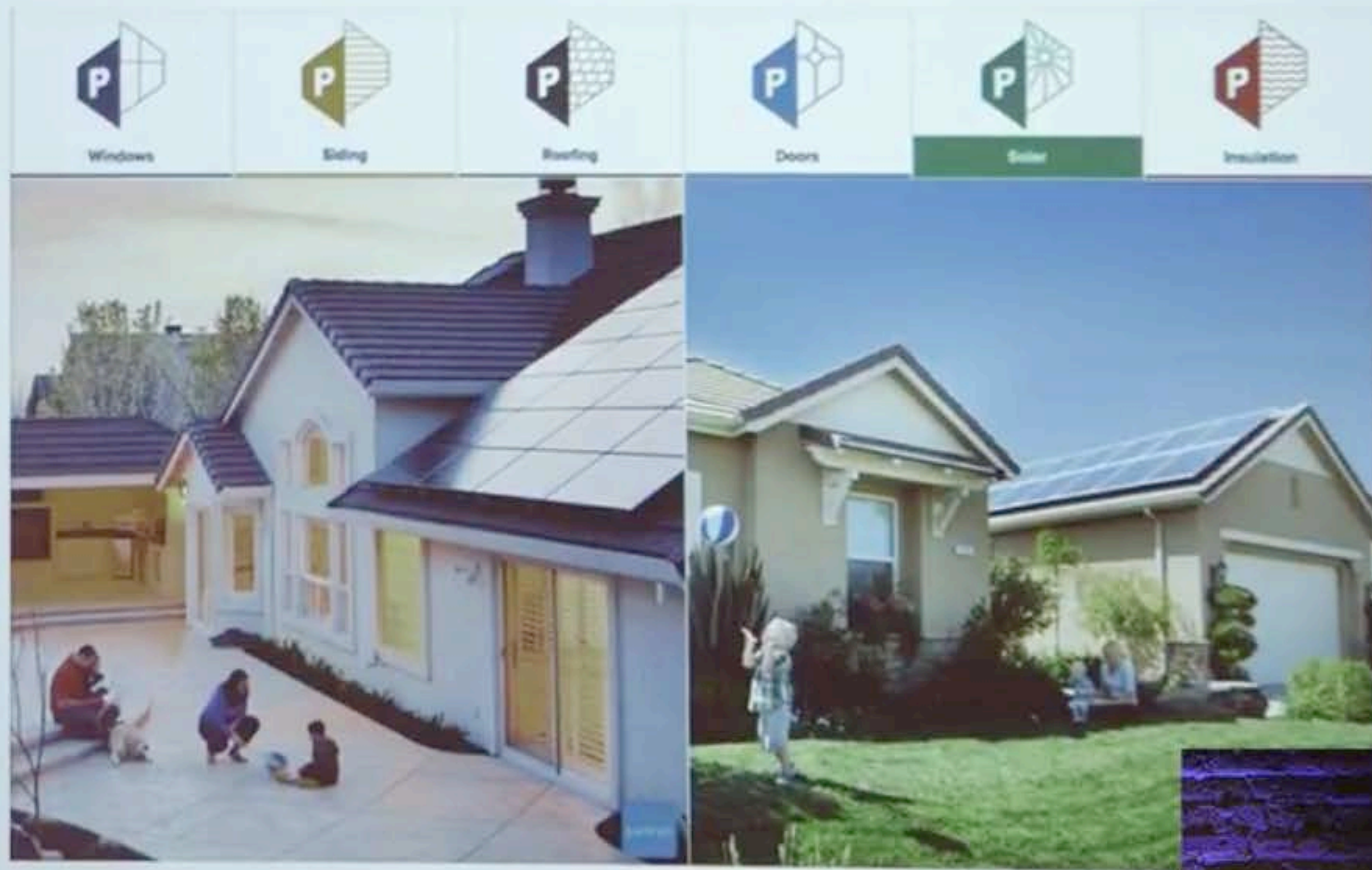
HOME
REMODELING



Ben Klang



HOME
REMODELING



Ben Klang

10 YEARS WITH RUBY

20 YEARS IN OPEN SOURCE



HOME
REMODELING

SUPPORTING:

2,800 END USERS

10-YR-OLD RAILS APP

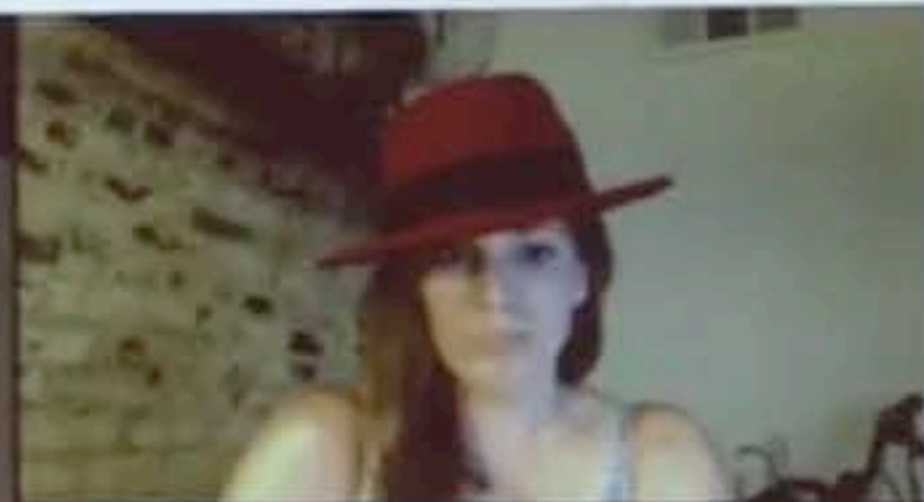
59 TECHNOLOGISTS



Rafael Garcia



The Heroes



Kraig Schwenn



Timothy Wernhold



Power Room



Ben Klang



5125



Adam Phillips



POW

- Operations 12%
- Permits
- Project Management
- Reports
- Sales
- Sales Dashboard
- Scheduling
- Inspection Routes
- Service Routes
- Scrum
- Support Calls
- Support Tickets
- System Settings
- System Support
- Talent Acquisition



Map - Marketer Est. Appts Corporate Events Schedulers Stats Watcher Mexico 2017 - Early Arrivals Signup Support Tickets Templates Green

Reload Showing PHL Created On, Marketer Created, PHL Markets, 13/09/2017 change filter

Marketer Est. Appts

Total Field Mtg Events Retail

Marketer Created Created On
PHL, PHL Markets | Wed 13/09/2017

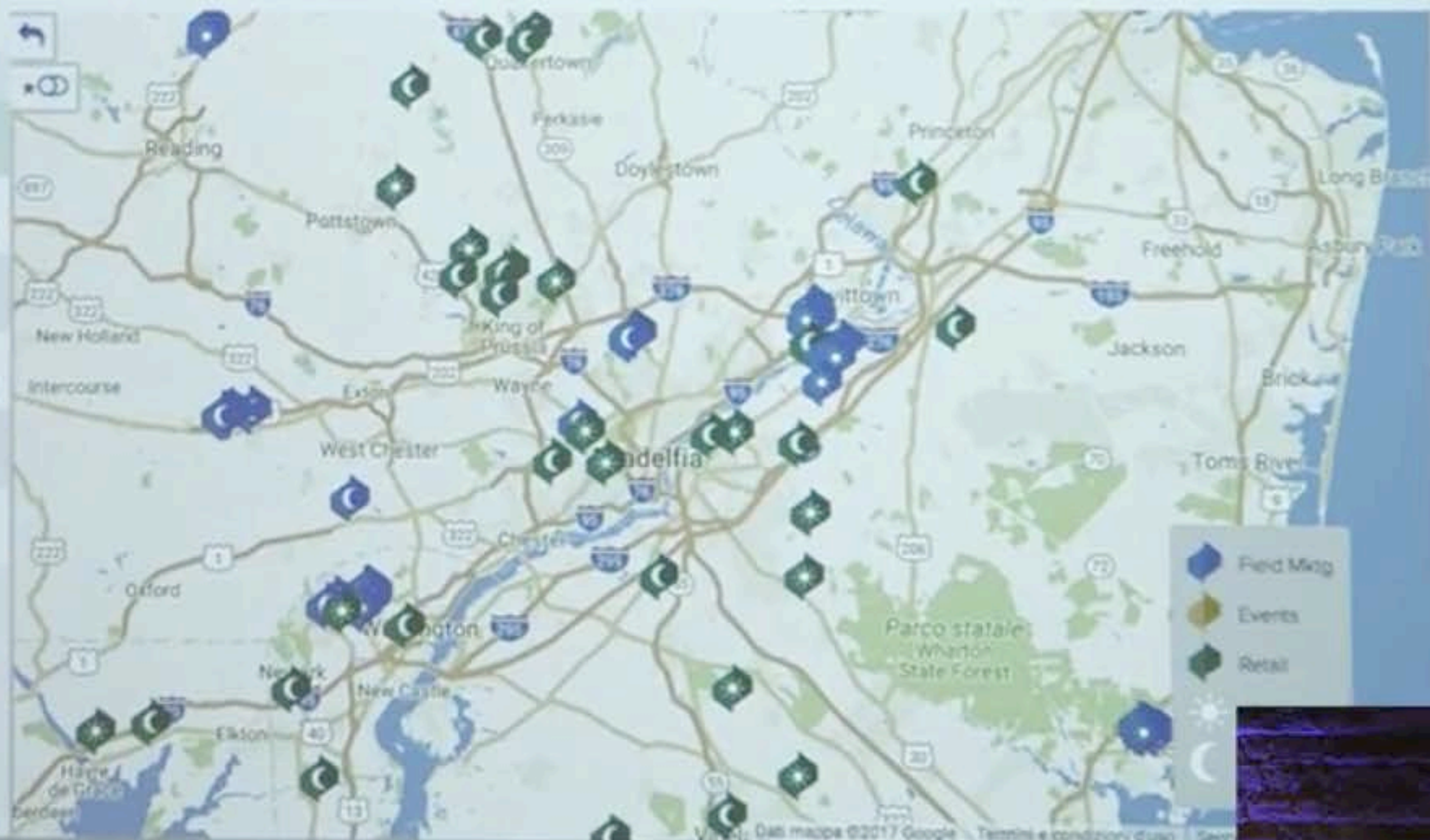
145	104	0	41
47/98	34/70	0/0	13/28

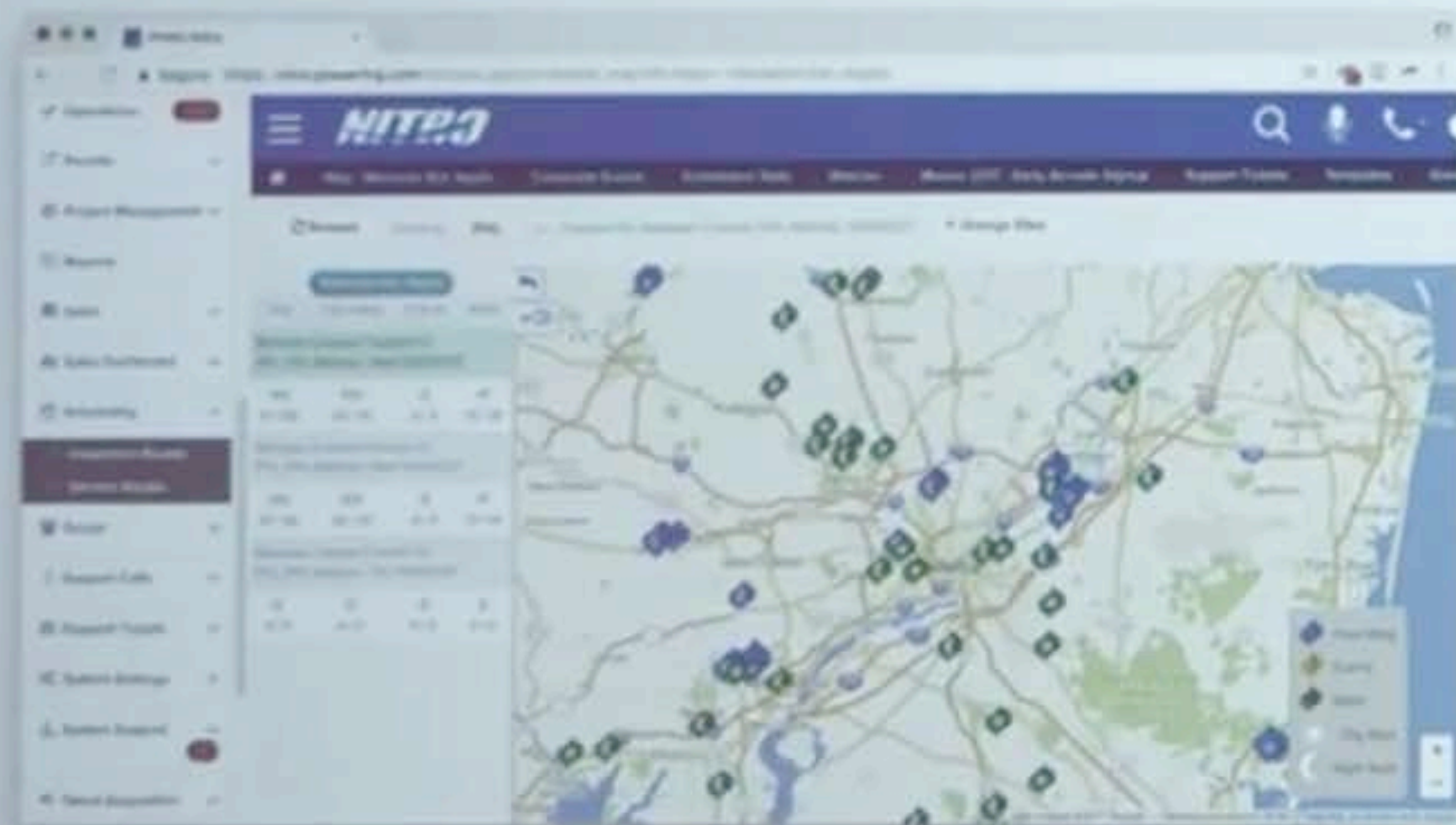
Marketer Created Created On
PHL, PHL Markets | Wed 13/09/2017

145	104	0	41
47/98	34/70	0/0	13/28

Marketer Created Created On
PHL, PHL Markets | Thu 14/09/2017

0	0	0	0
0/0	0/0	0/0	0/0





- LAUNCHED 2007
- \$500M REVENUE
- 773 ACTIVE RECORD MODELS
- 401 MAIN MENU ITEMS
- 491,910 LOC

The Ship

```
class User < NitroComponentTransition::ApplicationRecord
  include UserProfile::Avatar
  include Directory::BranchSync
  include ReplicatesToTraining
  include Redis::Objects
  include User::TrackableAttendance
  include Publishable
  include HasDAMirror
```

```
# NOTE: These are listed in explicit order. First match for the user is the default.
CONTROLLER_ACCESS = ActiveSupport::OrderedHash.new
CONTROLLER_ACCESS["reps"]           = { :action => :read, :subject => Sale }
CONTROLLER_ACCESS["leads"]          = { :action => :read, :subject => Lead }
CONTROLLER_ACCESS["inspections"]    = { :action => :read, :subject => InspectionAppointment }
CONTROLLER_ACCESS["dashboard"]      = { :action => :view_dashboard, :subject => Sale }
CONTROLLER_ACCESS["finance_projects"] = { :action => :read, :subject => FinanceProject }
```

```
CONTROLLER_ACCESS["report"]      = true
CONTROLLER_ACCESS["reports"]     = true
CONTROLLER_ACCESS["schedules"]   = { :action => :schedule, :subject => User }
CONTROLLER_ACCESS["users"]       = { :action => :read, :subject => User }
CONTROLLER_ACCESS["summaries"]   = { :action => :read, :subject => Sale }
```

```
UPDATE PROWLING_FEATURE_FLAG = "profile_user_updates".freeze
UPDATE PROWLING_STACK_KEY = "user_update_stack_entry".freeze
UPDATE PROWLING_FIELD_KEY = "user_update_field".freeze
```

Mission:

when creating
users, the created by is

would contain exactly 4 rows (id: 42, login: "bubba@192.168.1.100", first_name: "Bob", last_name: "Bingo", created_at: nil, current_updated_at: "2017-04-24 14:17:00", created_by_id: nil, updated_by_id: nil and idempot: id: 42, login: "bubba@192.168.1.100", first_name: "Bob", last_name: "Bingo", created_at: nil, current_updated_at: "2017-04-24 14:17:00", created_by_id: nil, updated_by_id: nil).

```
when branch param is sent
  returns the location of every file that should
```

```
update_phone_config
with a phoneless user
doesn't do much
when role changes
```

when successfully saving the config

instructor?

initials

The Ship

```
class User < NitroComponentTransition:ApplicationRecord
  include UserProfilerAvatar
  include Directory::BranchSync
  include ReplicatesTraining
  include Aedis::Objects
  include Users::TrackableAttendance
  include Publishable
  include HasGDPRMirror
```

```
# NOTE: These are listed in explicit order. First match for the user is the default.
CONTROLLER_ACCESS = ActiveSupport::OrderedHash.new
CONTROLLER_ACCESS["reps"] = { :action => :read, :subject => Sale }
CONTROLLER_ACCESS["leads"] = { :action => :read, :subject => Lead }
CONTROLLER_ACCESS["inspections"] = { :action => :read, :subject => InspectionAppointment }
CONTROLLER_ACCESS["dashboard"] = { :action => :view_dashboard, :subject => Sale }
CONTROLLER_ACCESS["finance_projects"] = { :action => :read, :subject => FinanceProject }
```

```
CONTROLLER_ACCESS["report"]      = true
CONTROLLER_ACCESS["reports"]     = true
CONTROLLER_ACCESS["schedules"]  = { :action => :schedule, :subject => User }
CONTROLLER_ACCESS["users"]      = { :action => :read, :subject => User }
CONTROLLER_ACCESS["summaries"]  = { :action => :read, :subject => Sale }
```

```
UPDATE PROFILING FEATURE_FLAG = "profile_user_updates".freeze
UPDATE PROFILING STACK_KEY = "user_update_stack_entry".freeze
UPDATE PROFILING FIELD_KEY = "user_update_field".freeze
```

Mission: Features

initials

The Mission: Ship Code

```
require 'digest/sha1'
require_dependency "user/work_schedule"
require_dependency "user/rep_leads"
require_dependency "user/rep_draws"
require_dependency "user/rep_commissions"
require_dependency "user/authentication"
require_dependency "user/project_management"
require_dependency "user/avatar"
require_dependency "user/territory_history"
require_dependency "user/connect_user"
require_dependency "user/nitro_reminders"
require_dependency "user/performance_teams"
require_dependency "user/contests"

class User < NitroComponentTransition::ApplicationRecord
  include UserProfile::Avatar
  include Directory::BranchSync
  include ReplicatesToTraining
  include Redis::Objects
  include User::TrackableAttendance
  include Publishable
  include HasLDAPMirror

  # =====
  # CONSTANTS
  # =====

  # =====
  # These are listed in explicit order. First match for the user is the default.
  CONTROLLER_ACCESS = ActiveSupport::OrderedHash.new
  CONTROLLER_ACCESS["reps"] = { :action => :read, :subject => :Sale }
  CONTROLLER_ACCESS["leads"] = { :action => :read, :subject => :Lead }
  CONTROLLER_ACCESS["inspections"] = { :action => :read, :subject => :InspectionAppointment }
  CONTROLLER_ACCESS["dashboard"] = { :action => :view_dashboard, :subject => :Sale }
  CONTROLLER_ACCESS["finance_projects"] = { :action => :read, :subject => :FinanceProject }

  CONTROLLER_ACCESS["estimates"] = { :action => :read, :subject => :Project }
  CONTROLLER_ACCESS["projects"] = { :action => :read, :subject => :Project }

  CONTROLLER_ACCESS["report"] = true
  CONTROLLER_ACCESS["reports"] = true
  CONTROLLER_ACCESS["schedules"] = { :action => :schedule, :subject => :User }
  CONTROLLER_ACCESS["users"] = { :action => :read, :subject => :User }
  CONTROLLER_ACCESS["summaries"] = { :action => :read, :subject => :Sale }

  CONTROLLER_ACCESS["tasks"] = { :action => :read, :subject => :Task }
  CONTROLLER_ACCESS["lead_time_methods"] = { :action => :read, :subject => :LeadTimeMethod }
  CONTROLLER_ACCESS["project_tracks"] = { :action => :read, :subject => :ProjectTrack }
  CONTROLLER_ACCESS["project_states"] = { :action => :read, :subject => :ProjectState }

  CONTROLLER_ACCESS["installation_crews"] = { :action => :read, :subject => :InstallationCrew }

  ACTIVE_STATES = %w(active)
  INACTIVE_STATES = %w(hired inactive no_show)

  UPDATE_PROFILING_FEATURE_FLAG = "profile_user_updates".freeze
  UPDATE_PROFILING_STACK_KEY = "user_update_stack_entry".freeze
  UPDATE_PROFILING_FIELD_KEY = "user_update_field".freeze

  # =====
  # ATTRIBUTES
  # =====
```

```
converted?
when the user is not have candidate
when the user is a candidate
locking for update
when updating
updates the updated by id
when creating
recruiter?
when login: "MANAGER START WITH USER ACCOUNTS?", first name
last name, email, connect, updated at: "2017-09-24 14:17:00", branch
id, id, updated by, id, role, role, login: "MANAGER WITH FIRST NAME: MATT/JOHN/STEF
first name: "Mey", last name: "Rosen", phone, id, connect, updated at: "2017-09-24 14:17:
00", updated by, id, role, updated by, id, role
set_active
sets active to false if user moves to an inactive state
sets active to true if user moves to an active state
marketer_activity_at_shift
when branch param is sent
returns the marketer activity on that branch
when branch param is not sent
returns the activity for the user regardless of branch
when there's no activity
returns nil
upcoming_training_hours
returns a hash of total hours by date in the time zone of the user
callbacks
update_phone_config
with a phoneless user
don't do much
when role changes
with a user that has a phone
when successfully saving the config
when it fails saving the config
instructor?
is true for users teaching at least one class
is false for users who not teach
started on validation
validation prevents if start date when changing start date
validation prevents if start date on new records
only not validate presence for existing records when not changing start date
initials
```


The Villain



The Lackeys

Process Drag

Feature Fragmentation

Scope Bloat




Process Drag

				App Foundation	Setups	Docs & Lint	Tests
Average stage times:				4min 3s	10min 57s	6min 53s	2h 2min
#4	Sep 14 11:34	No Changes	ⓘ	2min 2s	10min 41s	6min 46s	2h 11min failed
#3	Sep 13 17:44	No Changes	ⓘ	1min 57s	10min 26s	6min 37s	1h 53min failed
#2	Sep 13 17:34	No Changes		10min 0s failed			
#1	Sep 13 11:33	No Changes	ⓘ	2min 12s	11min 44s	7min 17s	2h 1min failed

Feature Fragmentation

Who would you like to Connect with?

 Briana



Briana West

PHL Philadelphia Business Technology - Nitro Developer



Briana Williams

NJ New Jersey Marketing - Outbound Marketer



Brian Ray

ROOMMATE PREFERENCE

Select my roommate

S

Hassan Tanksley

Johnathan Rodgers

Kraig Schwerin

Individual User Memberships

✕ Ben Klang ✕ Garrett Arrowood ✕ David Nelson

✕ Jimmie Butler JIII

Jillian Tankersley (PHL Business Technology)

NAME

starts with

Winningham

FIRST NAME

starts with

LAST NAME

starts with

Lieberman

Order Gillespie
Matt Gorman
Kerry Grant
Chris Harding
Lenny Hellman
Kevin Hise
Zach Jones
Stephen Kane
Sydney Kassaman
Andrew Knight
Lakesha Lee
Jonathan Levy
Frankie Magazu
Angela Malle
Vince Mancano
Dyke Marshall
Jared Matthews
Gert McAllister
Cody McDonald
Michael Medley
Ruben Mendes
Zach Mordche
Sam Musco
Andrew Nash
Chris Osh

Scope Bloat

CONTROLLER

MODEL

MODEL

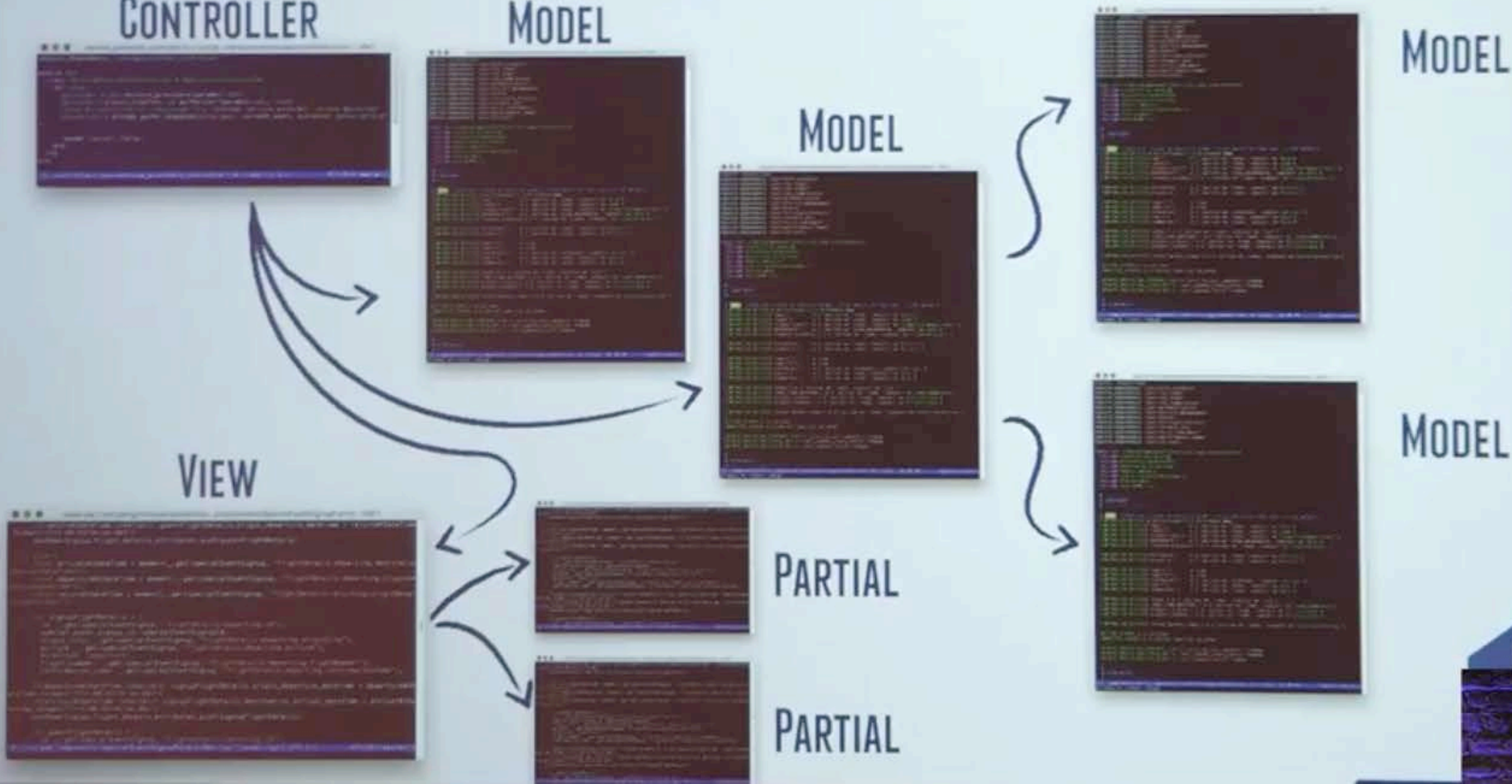
MODEL

MODEL

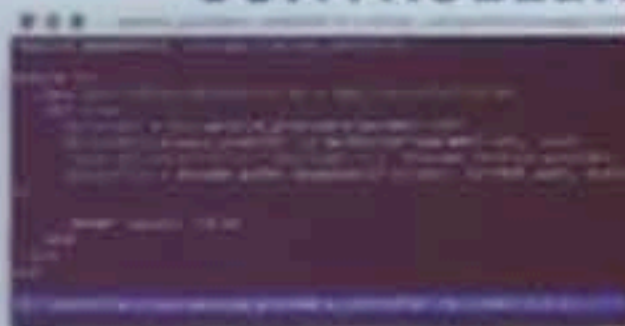
VIEW

PARTIAL

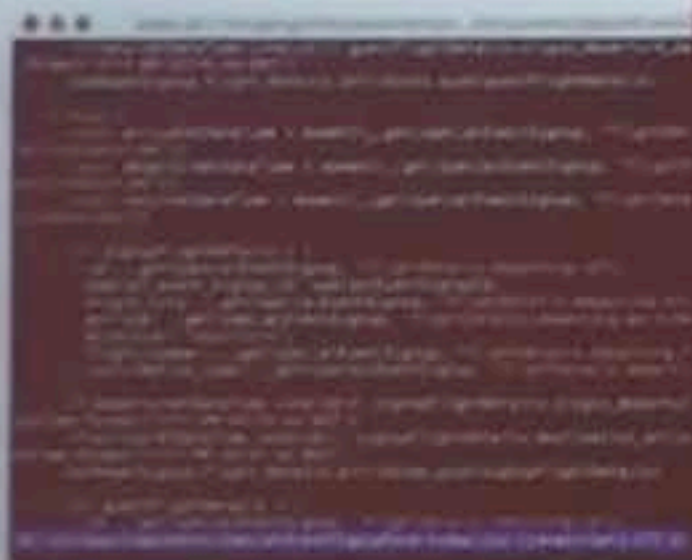
PARTIAL



CONTROLLER



VIEW

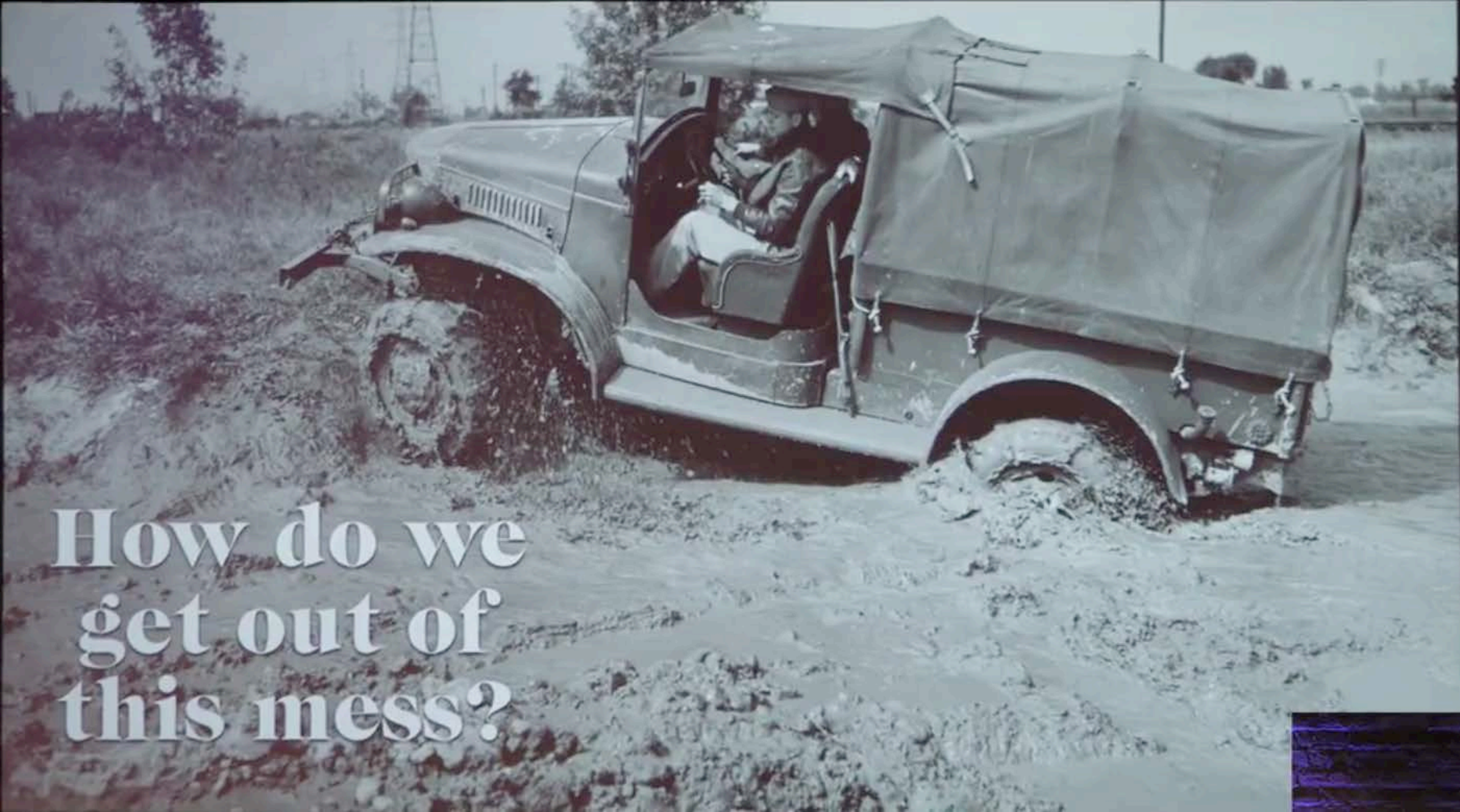


MODEL



MODEL

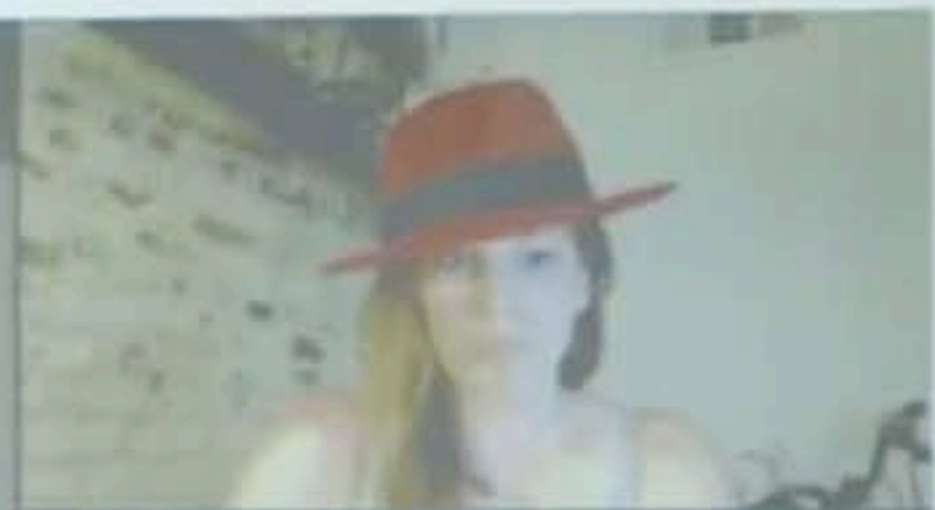




How do we
get out of
this mess?

Rafael Garcia

what if we...



Erin Johnson



Timothy Winhold



Power Room



Ben Klug



SL



Adam Phillips



What if we... ~~build a Test Pyramid?~~

- Massive effort
- Does not address code issues, only tests

What if we...

~~mock the database?~~

- Hides behavior of database layer:
Foreign Keys, Triggers, Views, Implicit Sorts
- DB behavior is critical to app behavior

What if we...

DRY our app?

- ApplicationHelper is a dumping ground
- Difficult to model dependencies
- Impossible to enforce boundaries

What if we... ~~switch to microservices?~~

- New failure mode: network partitions
- Performance Analysis tools immature
- Monitoring/High Availability much harder
- Must coordinate deploys of many apps

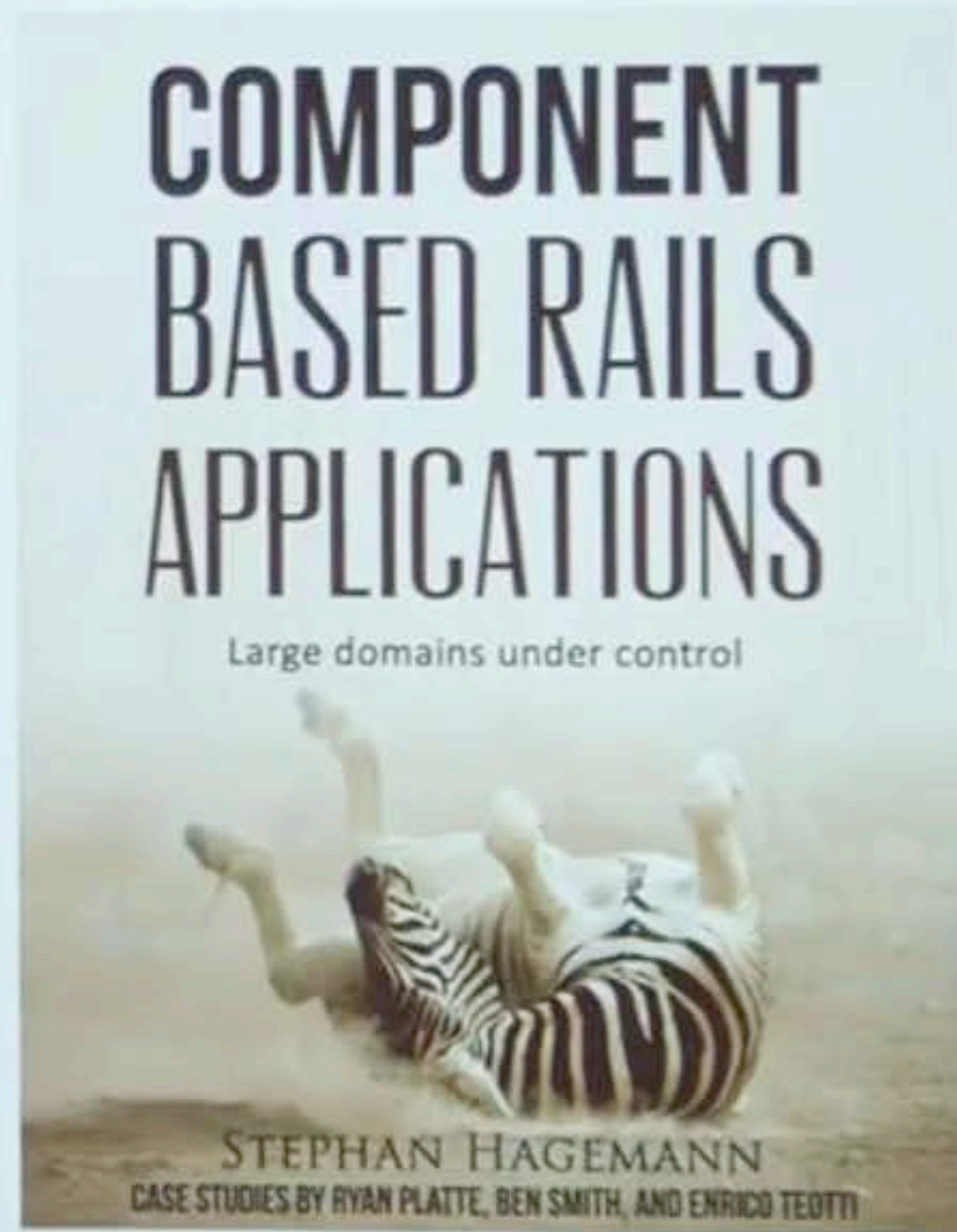
What if we...

break our app into gems?

- ✓ Clear responsibility boundaries
- ✓ Understandable dependency graph
- ✓ Reduces merge conflicts
- ✓ Allows selective testing

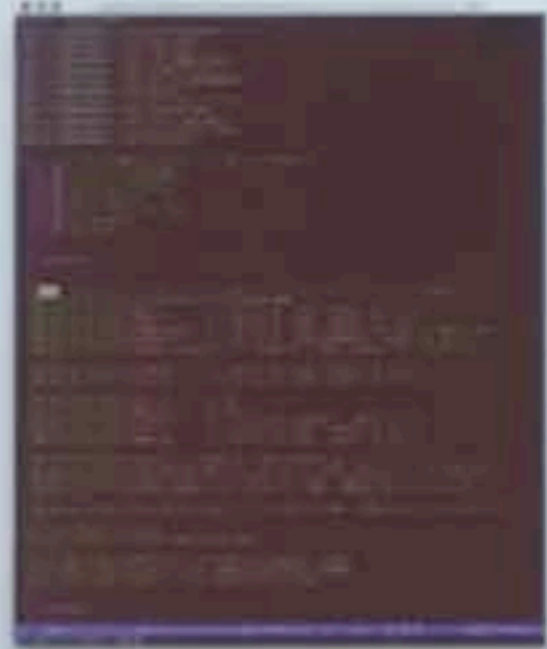
C.B.R.A.

Component Based Rails Architecture



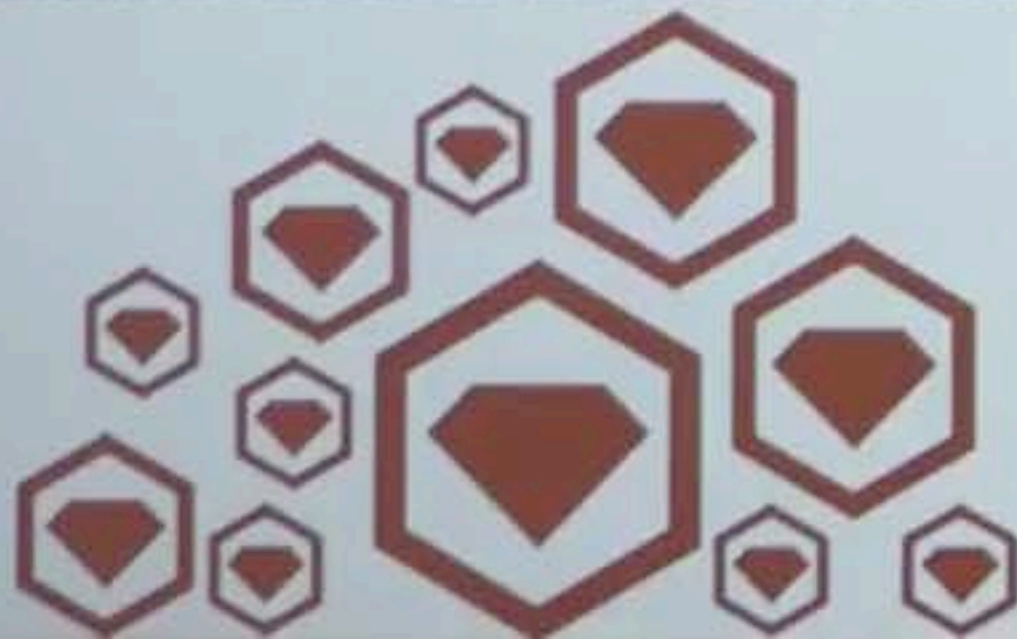
What is C.B.R.A.?

YOUR APP



YOUR CODE

OTHER PEOPLES' CODE



Start Small

🔒 powerhome / **greensky-ruby** Private

<> Code

📄 Issues 0

🔄 Pull requests 0

📁 Projects 0

Ruby client for the GreenSky Loan Application API

📁 powerhome / **consent**

<> Code

📄 Issues 0

🔄 Pull requests 0

DSL for constructing CanCanCan Abilities

🔒 powerhome / **wells_fargo_retail** Private

<> Code

📄 Issues 1

🔄 Pull requests 0

Wells Fargo Credit App SOAP Client

📁 powerhome / **paperclip-eitheror**

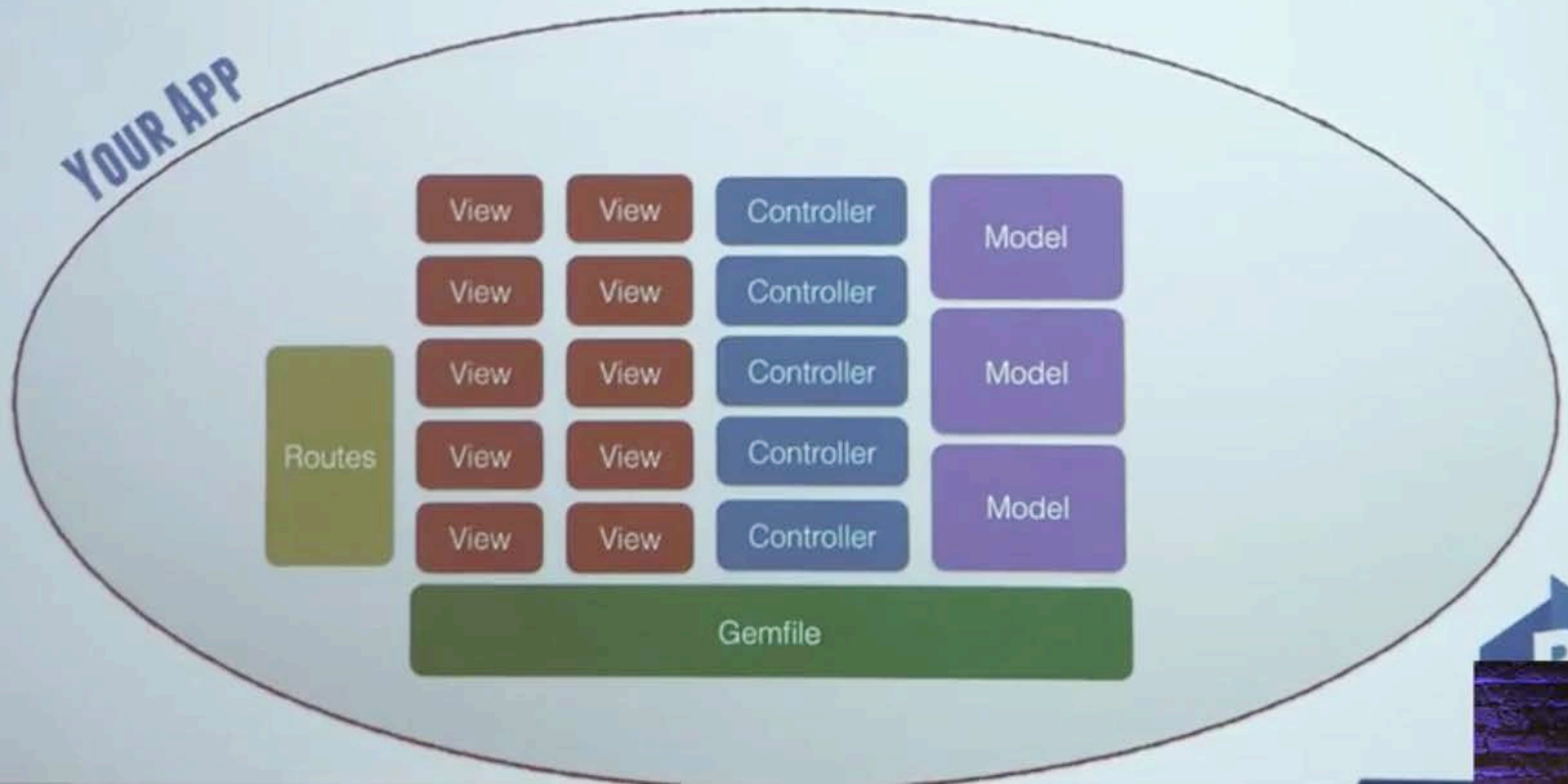
<> Code

📄 Issues 0

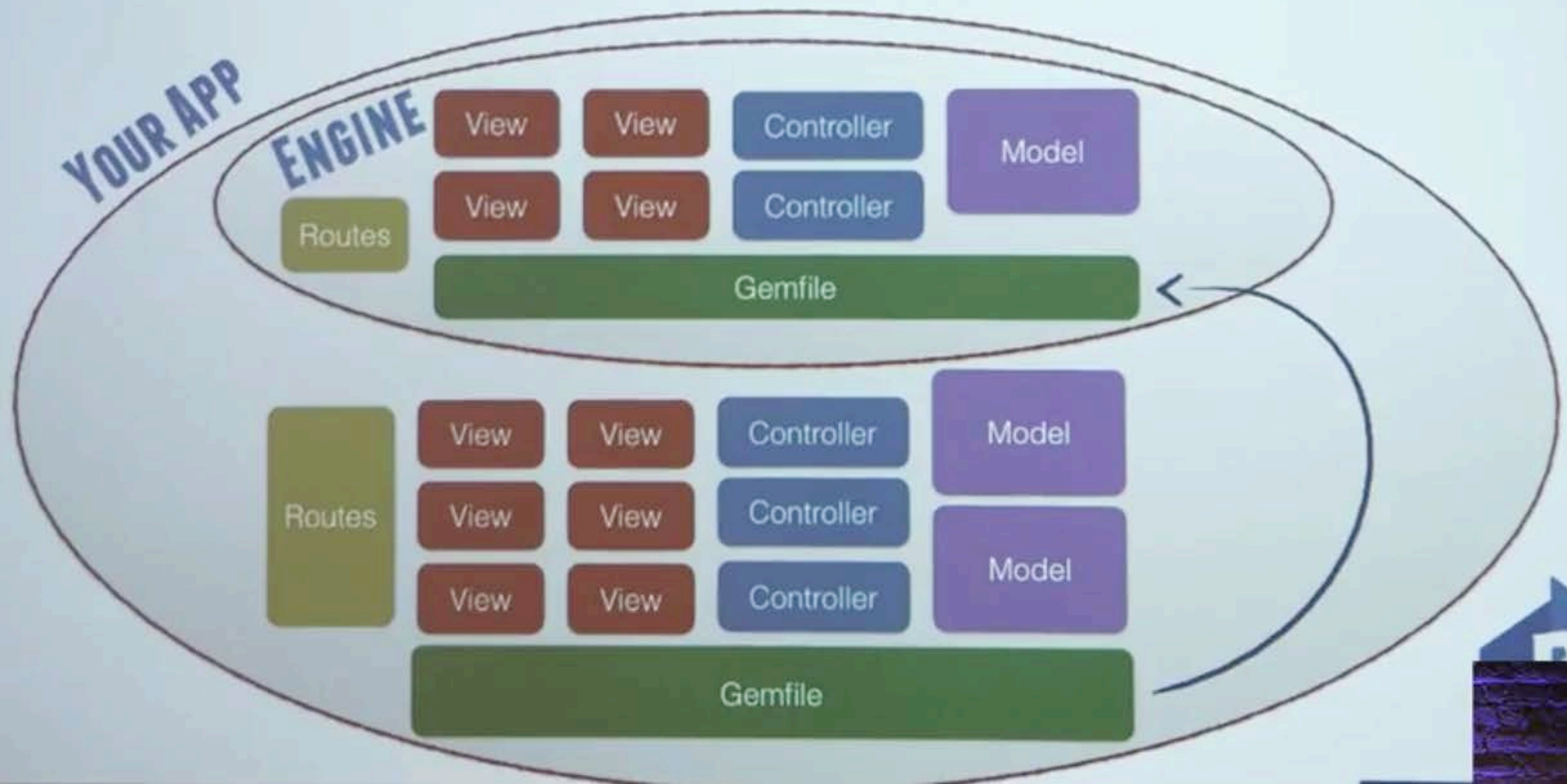
🔄 Pull requests

A Paperclip Storage with a fallback option.

Rails Engines

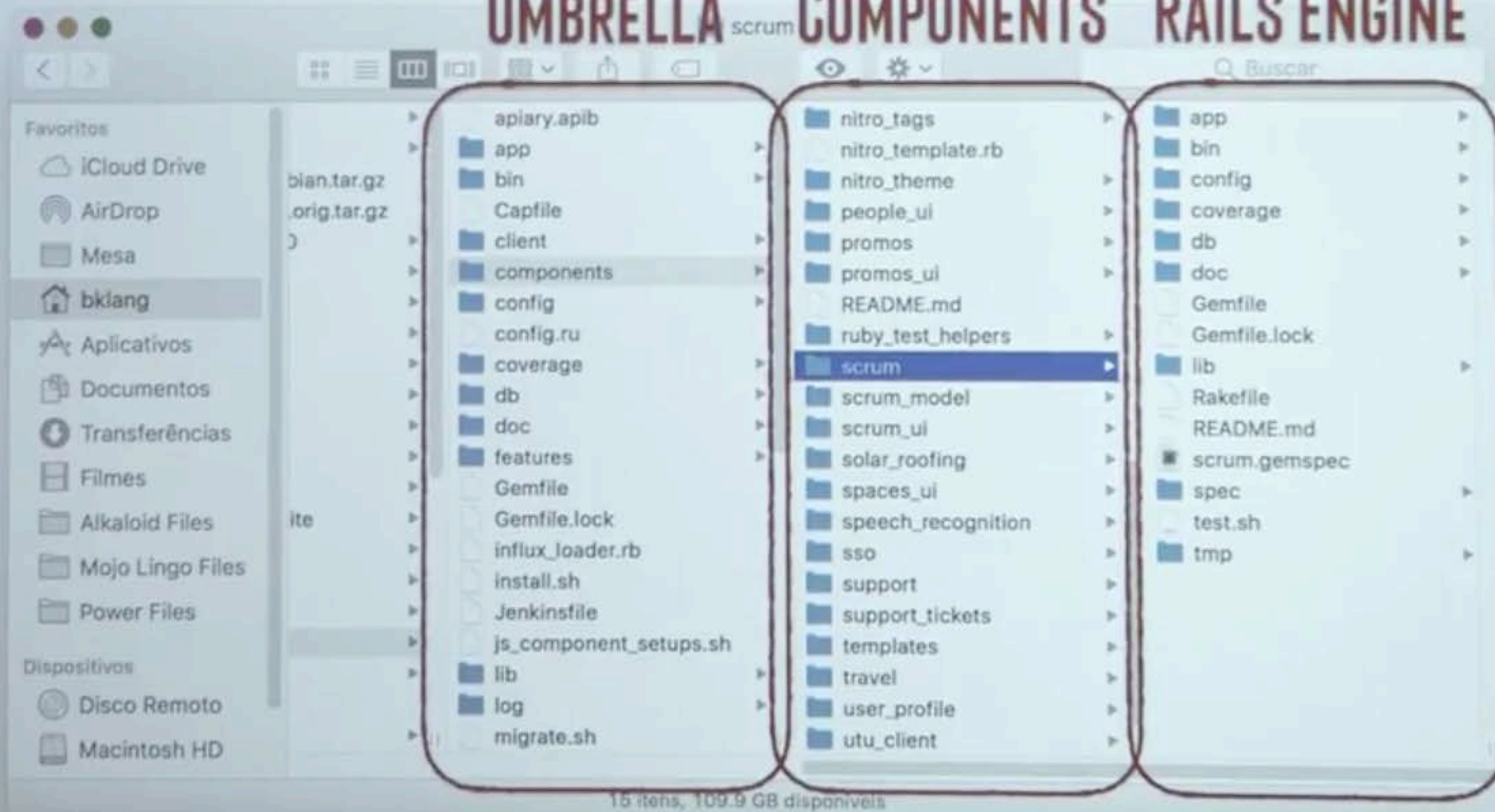


Rails Engines



App Components

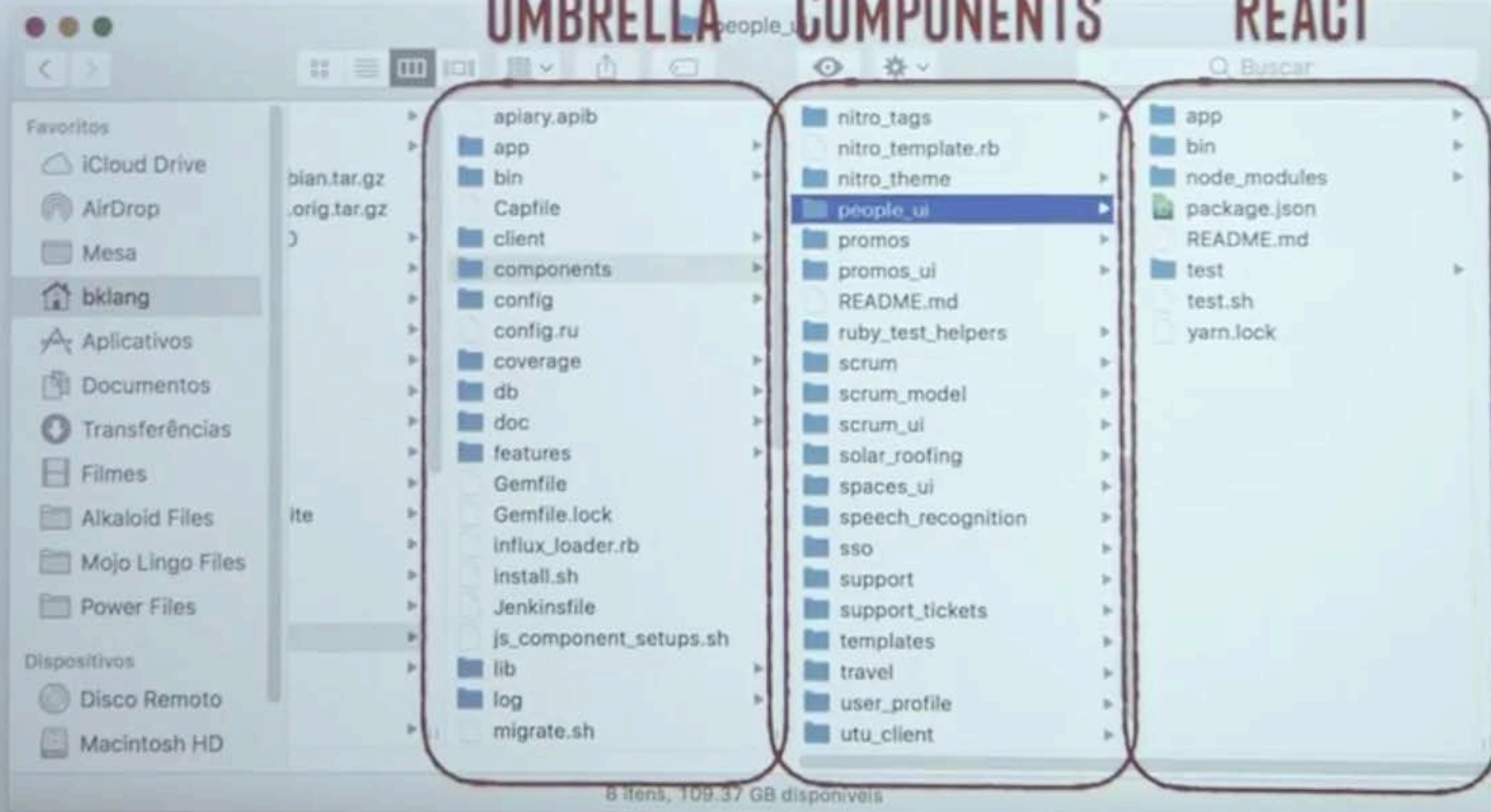
UMBRELLA COMPONENTS RAILS ENGINE



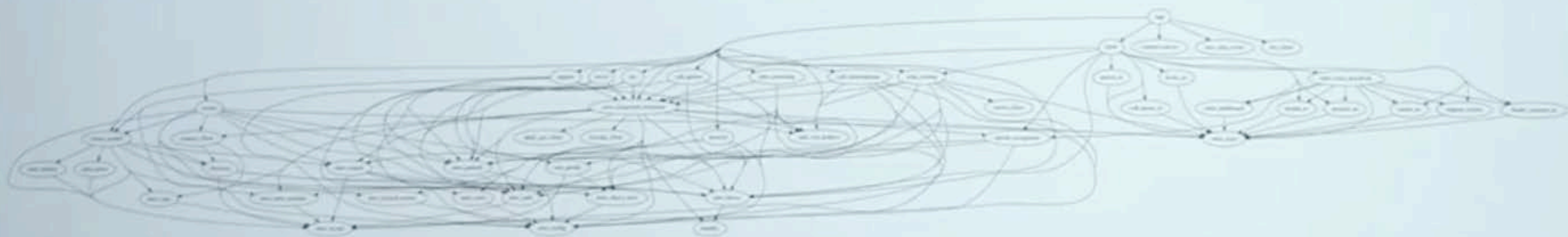
App Components

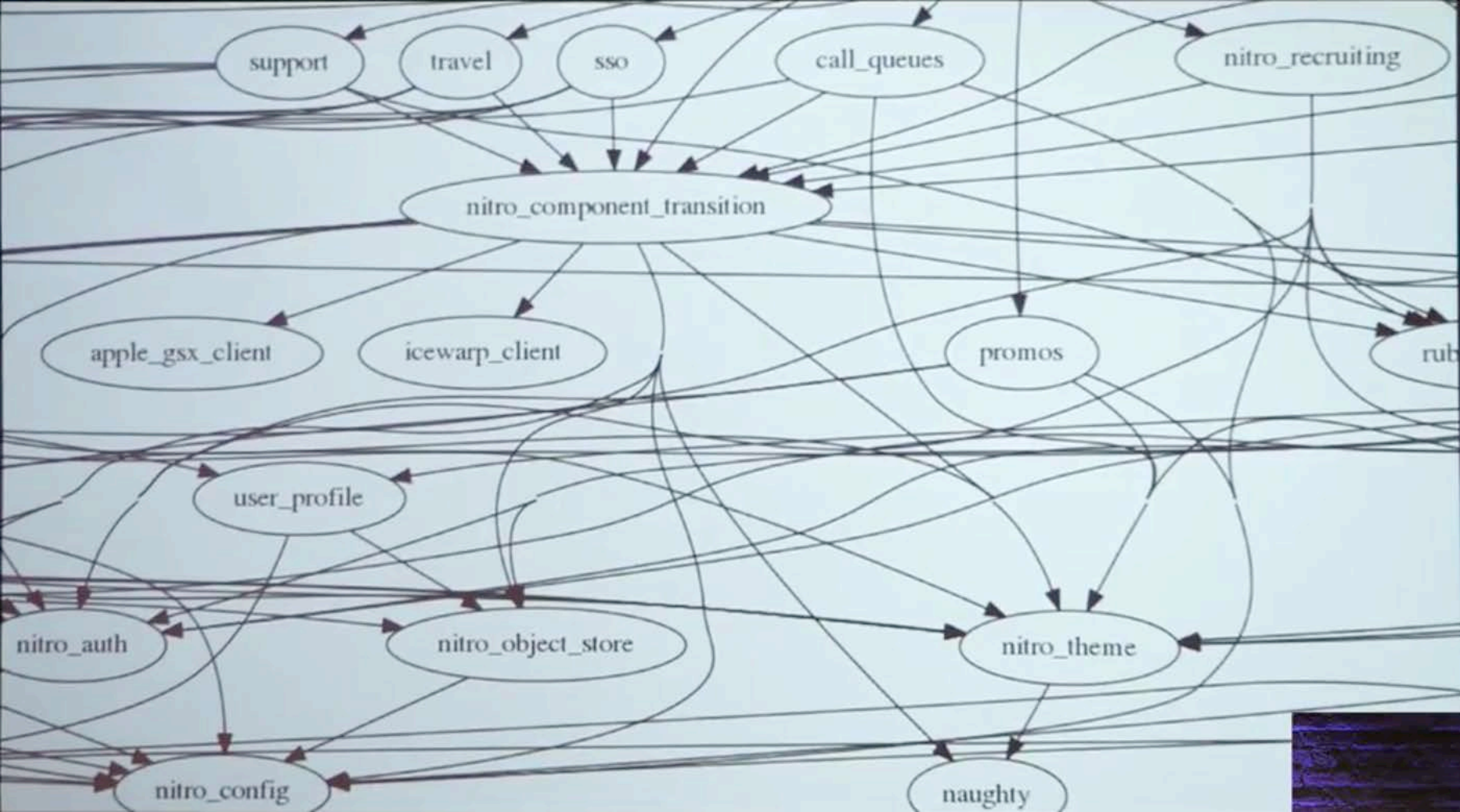
UMBRELLA COMPONENTS

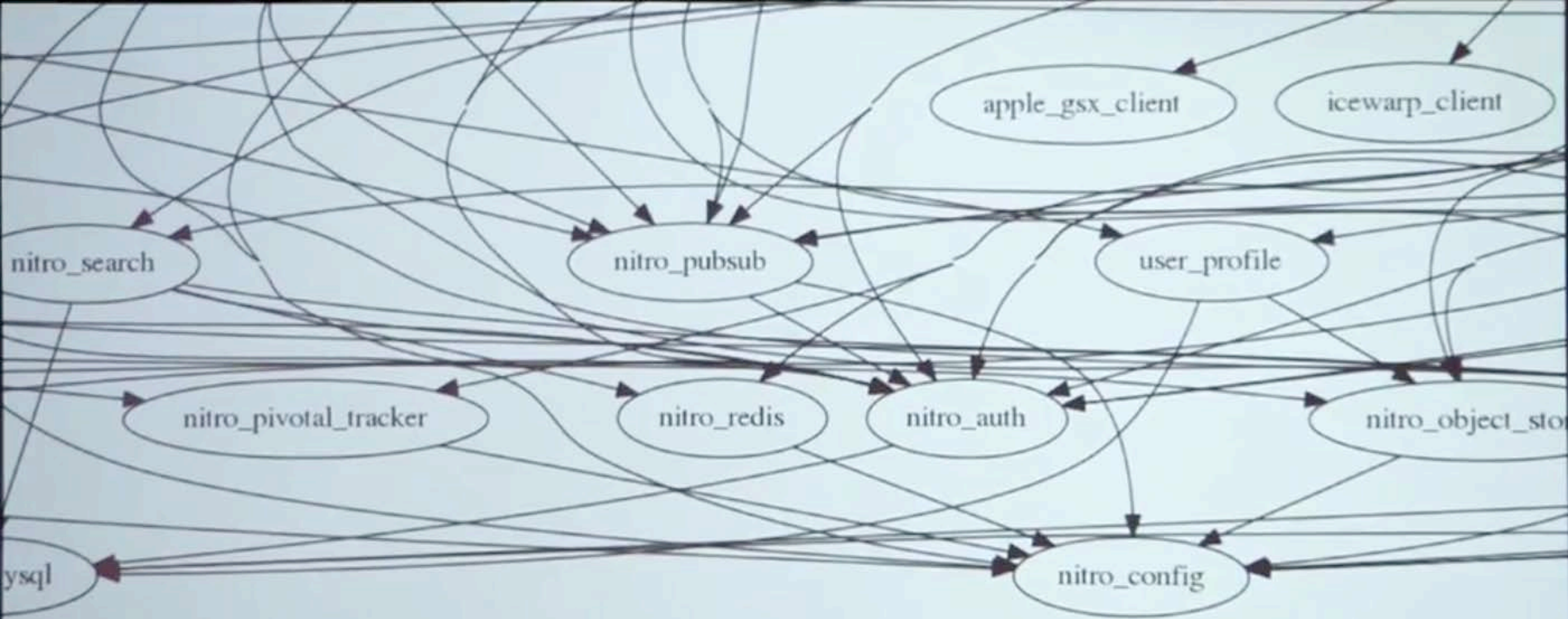
REACT

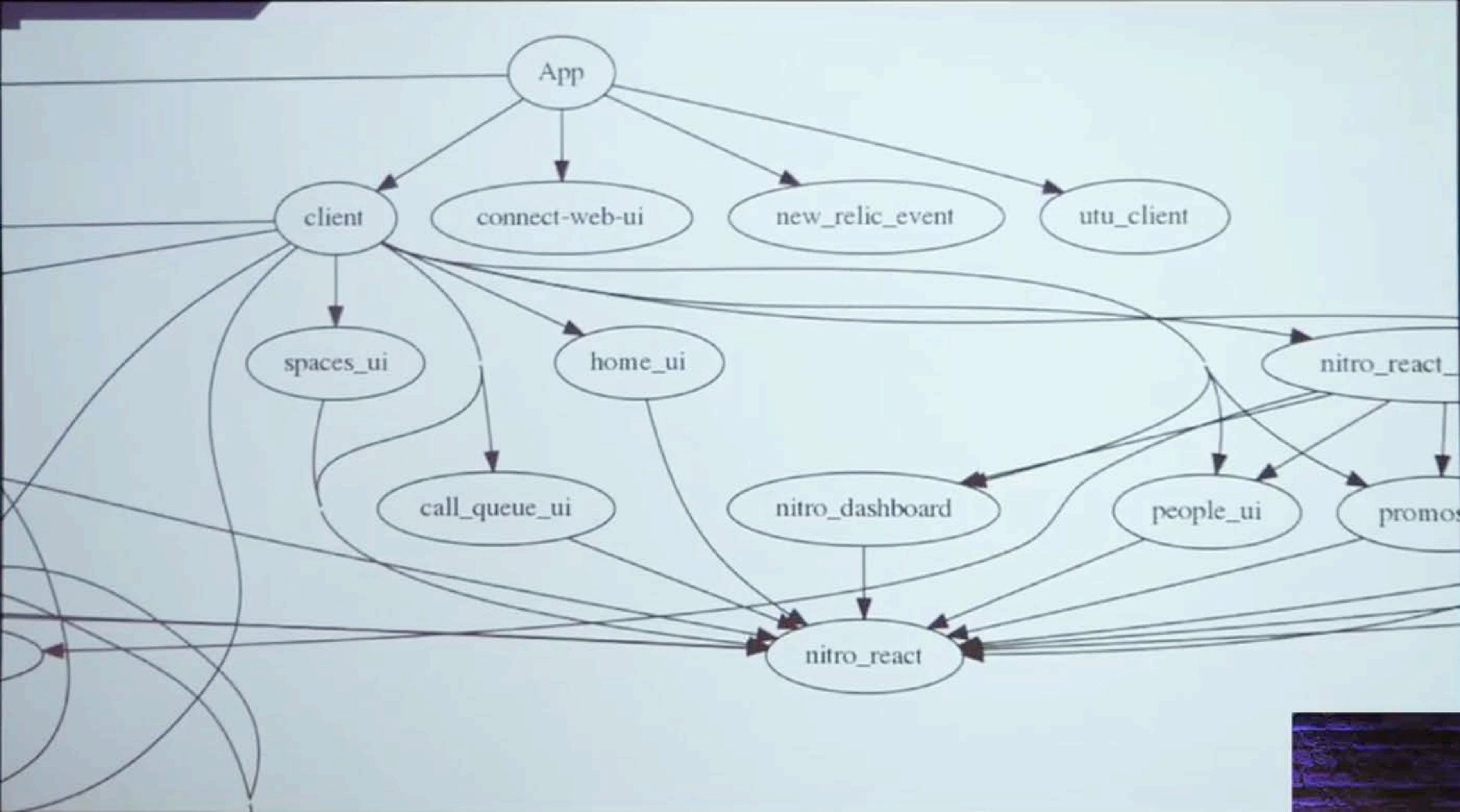


Today's C.B.R.A. Nitro









Today's C.B.R.A. Nitro

Average stage times:				App Foundation	Setups	Docs & Lint	Tests
				5min 12s	7min 37s	5min 16s	1min 6s
#2	Sep 14 07:12	No Changes	⓪	2min 4s	9min 11s	6min 32s	1min 17s
#1	Sep 12 10:29	No Changes	⓪	8min 20s	6min 4s	4min 1s	56s

RESULTS IN MINUTES, NOT HOURS

C.B.R.A. Pros and Cons

PROS

- Good code organization
- Parallelize tests
- Fewer merge conflicts
- Reduce human RAM requirements
- Step toward microservices

CONS

- Organizational Overhead
- Testing complexity
- Tools need adapting
 - Slower migrations
 - Gemfile synchronization

Should I Componentize?

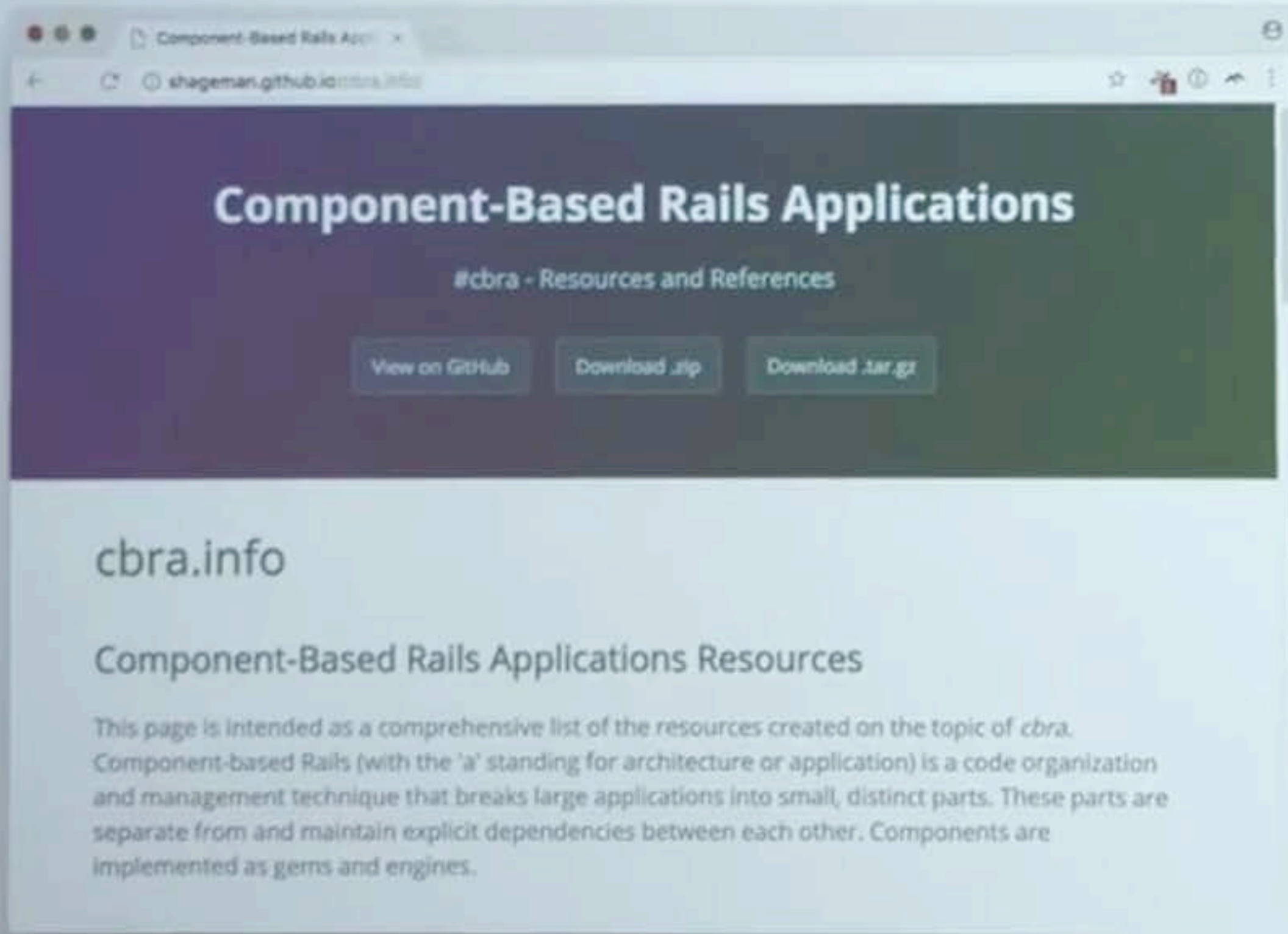
- ✓ Large Monolithic App?
- ✓ Large number of contributors?
- ✓ Frequent merge conflicts?
- ✓ Unbearably long CI run-times?
- ✓ Difficult to hold a feature in your head?

Should I Not Componentize?

- ✗ Greenfield app?
- ✗ Small team?
- ✗ Manageable, single-purpose codebase?
- ✗ Process drag not primary concern?

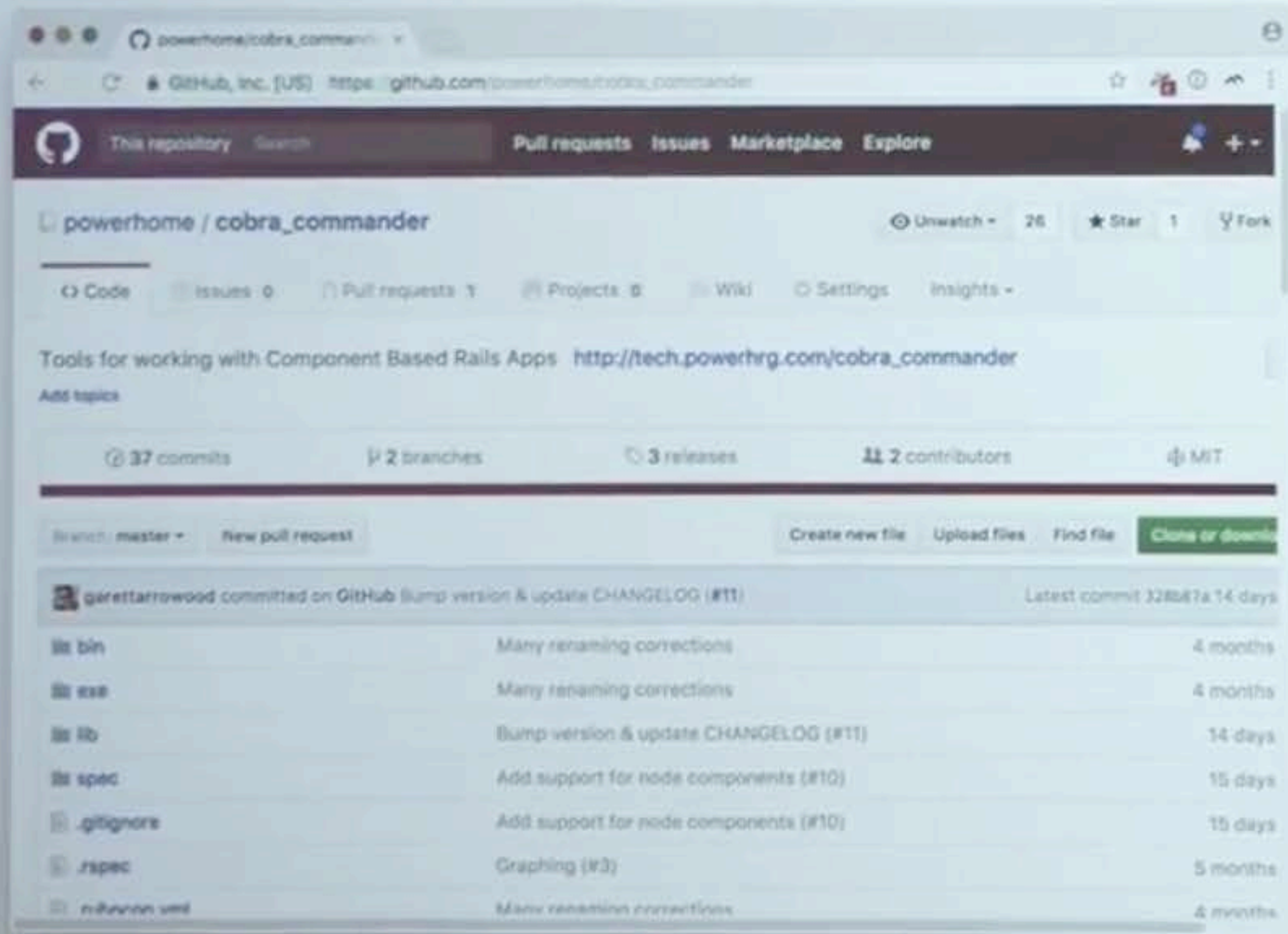


Additional Reading



[HTTPS://CBRA.INFO](https://cbra.info)

Cobra Commander



Github:
powerhome/
cobra_commander

Ben Klang
@bklang

