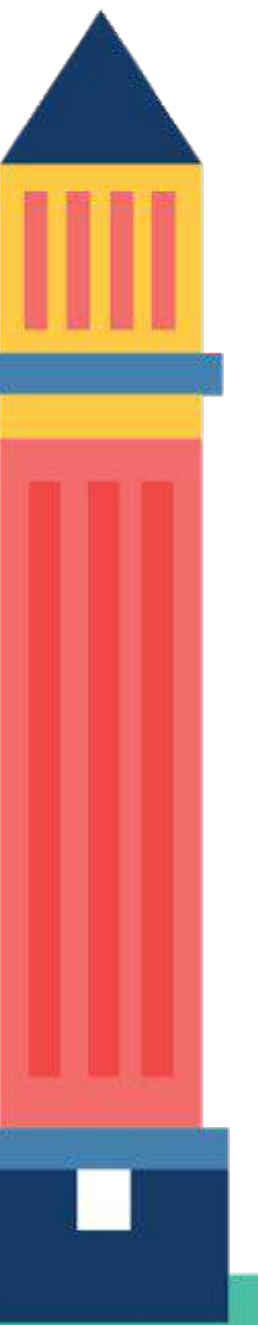
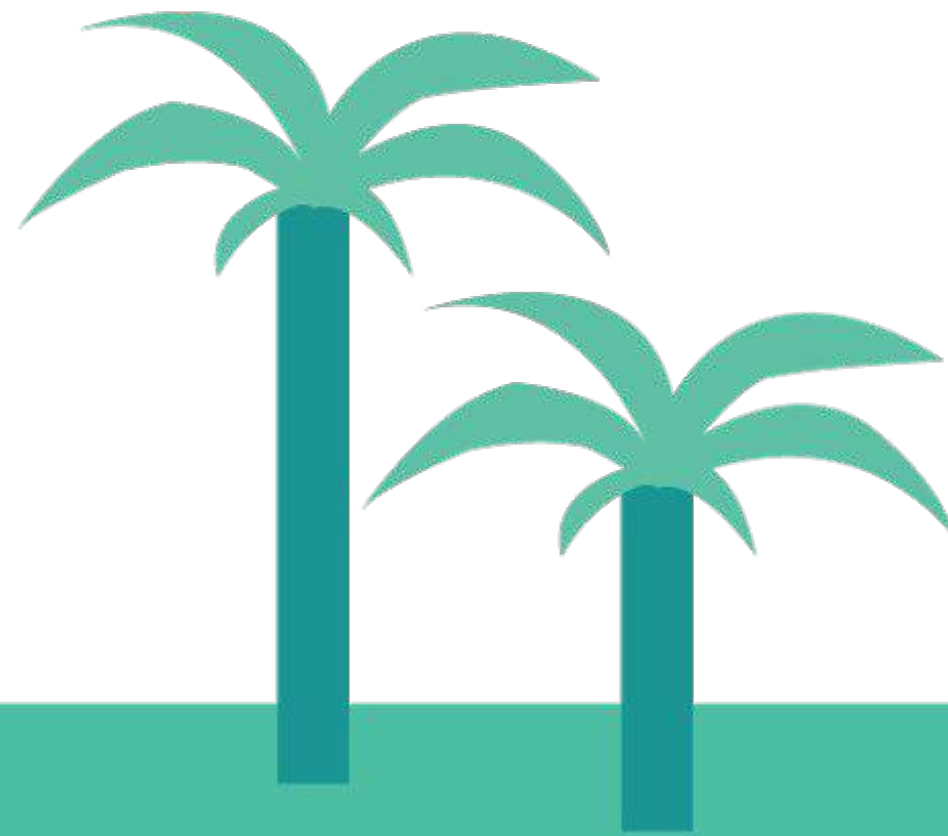
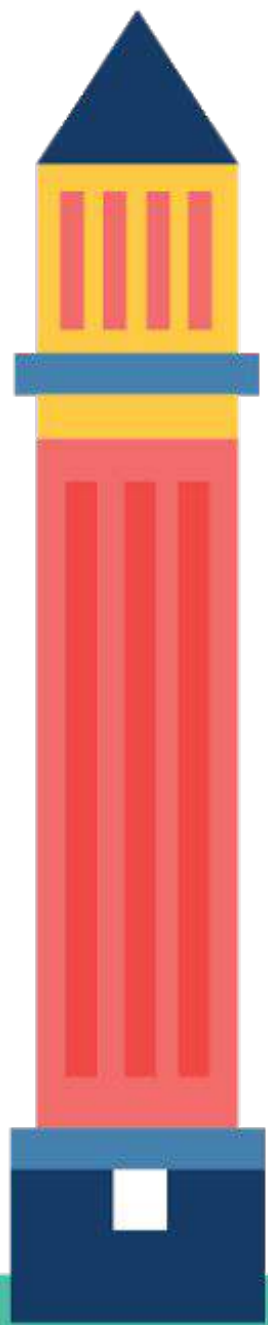




# App-in-a-Box with Docker App

Michael Irwin

Application Architect, Virginia Tech  
Docker Captain  
[@mikesir87](#)



# **Warning: Lots of content ahead!**

# About Virginia Tech

- Founded in 1872
- Main campus in Blacksburg, VA, USA
  - Six campuses throughout Virginia and one in Switzerland
- ~35,000 students and ~4,500 faculty/staff
- Manages research portfolio valued more than \$520 million





# Summit Overview

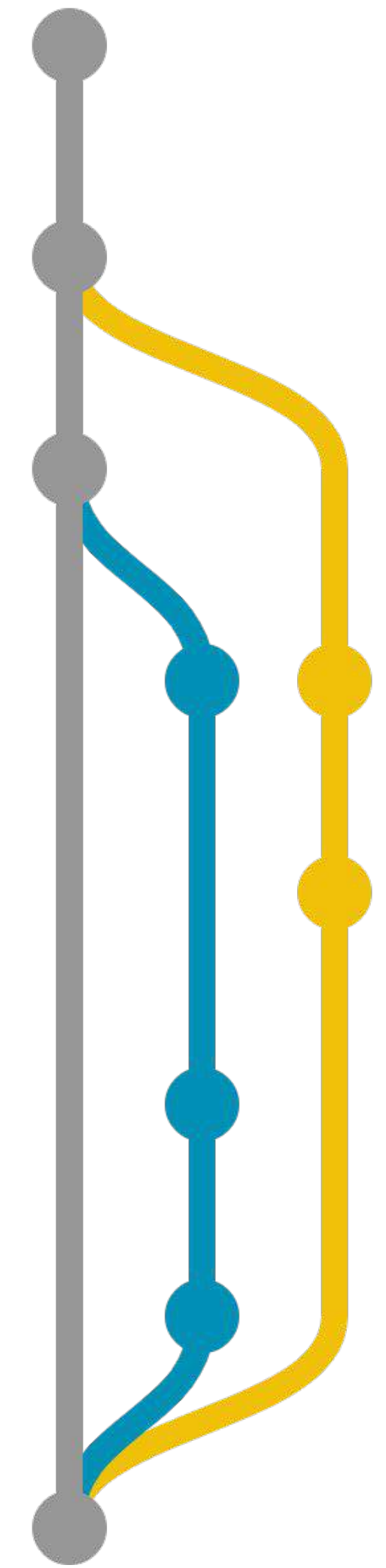
- Software designed to manage sponsored research processes
  - Document management, collaboration, compliance, and documentation
- First VT IT app to use containers and deploy to public cloud
- System is composed of several separately deployed components

# SUMMIT

The screenshot displays the SUMMIT web application interface. On the left, a sidebar contains navigation links: Notifications, WORK ITEMS, Open (with sub-links: My Open, Admin Review, People), Pending Sponsor Decision, Closed, APPROVALS (with sub-links: To be approved, Pending, Approved, Canceled), SPONSORS, ORGANIZATIONS, and ASSOCIATES. The main content area is titled 'SUMMIT' and includes a 'Get Started' button. Below the title, there's a search bar and a list of proposals. The first proposal is 'Persinger, Vanjan' with ID '020300 - Aerospace and Ocean Engineering (AOE)', dated '11/24/2018', and status 'Preparing'. The second proposal is 'Persinger, Vanjan' with ID '020300 - Aerospace and Ocean Engineering (AOE)', dated '11/26/2018', and status 'Preparing'. The third proposal is 'Persinger, Vanjan' with ID '020300 - Aerospace and Ocean Engineering (AOE)', dated '11/26/2018', and status 'Preparing'. The fourth proposal is 'Pratt, Jane' with ID '040100 - Building Construction (BC)', dated '01/01/2019', and status 'In Routing'. The fifth proposal is 'Sadhana, Dal' with ID '040100 - Building Construction (BC)', dated '01/01/2019', and status 'Preparing'. The sixth proposal is 'Sadhana, Dal' with ID '040100 - Building Construction (BC)', dated '01/01/2019', and status 'Preparing'. The right side of the interface shows a detailed view of a proposal titled 'Vanjan Persinger - YN 11/12 4:04 revise - US Department of Defense (DOD)'. The 'General Information' section includes fields for Work Label, Full Title, Type of Work, Originated From, Program Type, Research Theme, Research Initiative, Building (Where Work Performed), and Keywords (optional). The 'Personnel' section is also visible.

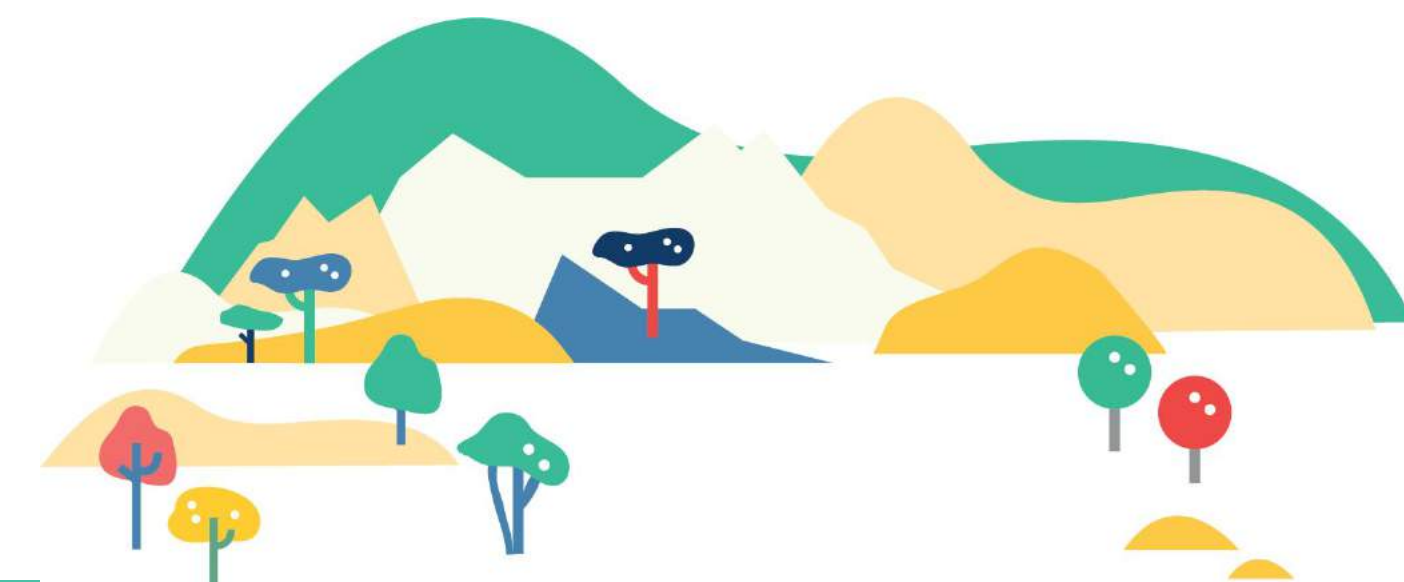
# We use GitHub Flow

- The **master** branch is always production-quality code
- New branch created for new work
- Work committed on new branch
- Open a merge request when ready
- Code review performed and code merged



# With Simple Apps...

- Working on a feature is simply checking out a branch
  - Get all source code and environment together
  - May need to restart containers



# Summit Components

## Desktop Client

AngularJS

## Mobile Client

Angular 4

## User Guide/Docs

Static HTML generated  
from mkdocs

## Admin Client

React

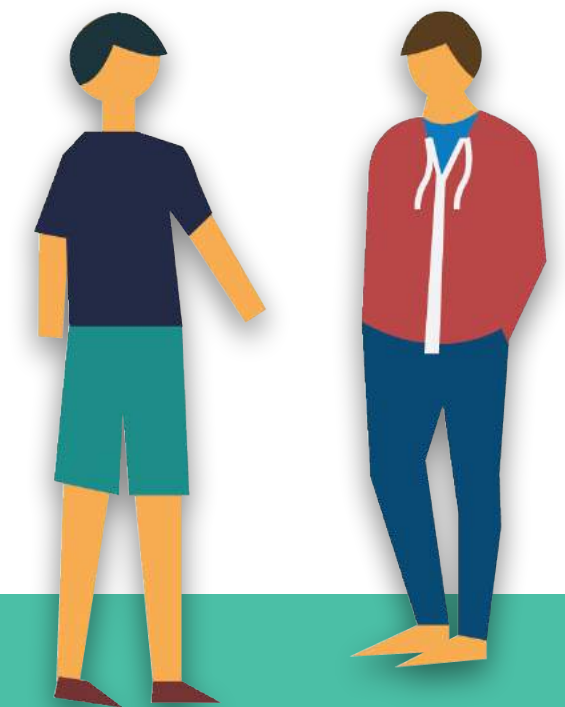
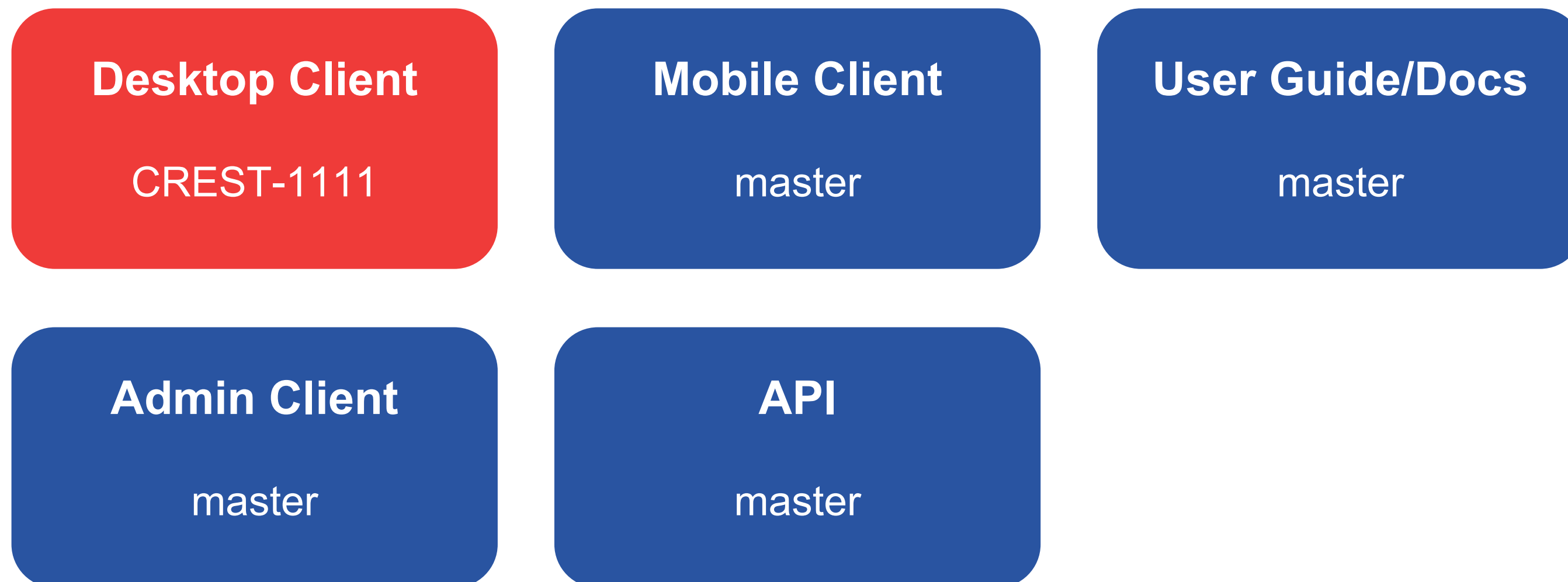
## API

JavaEE in Wildfly



# Scenario 1...

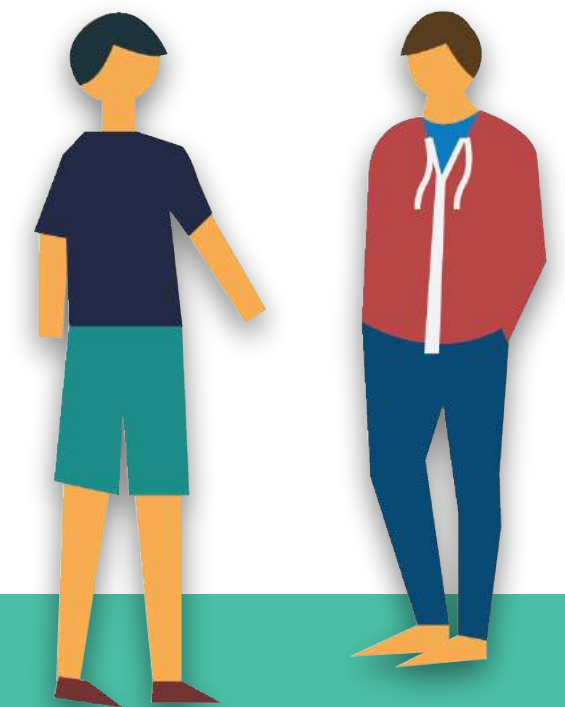
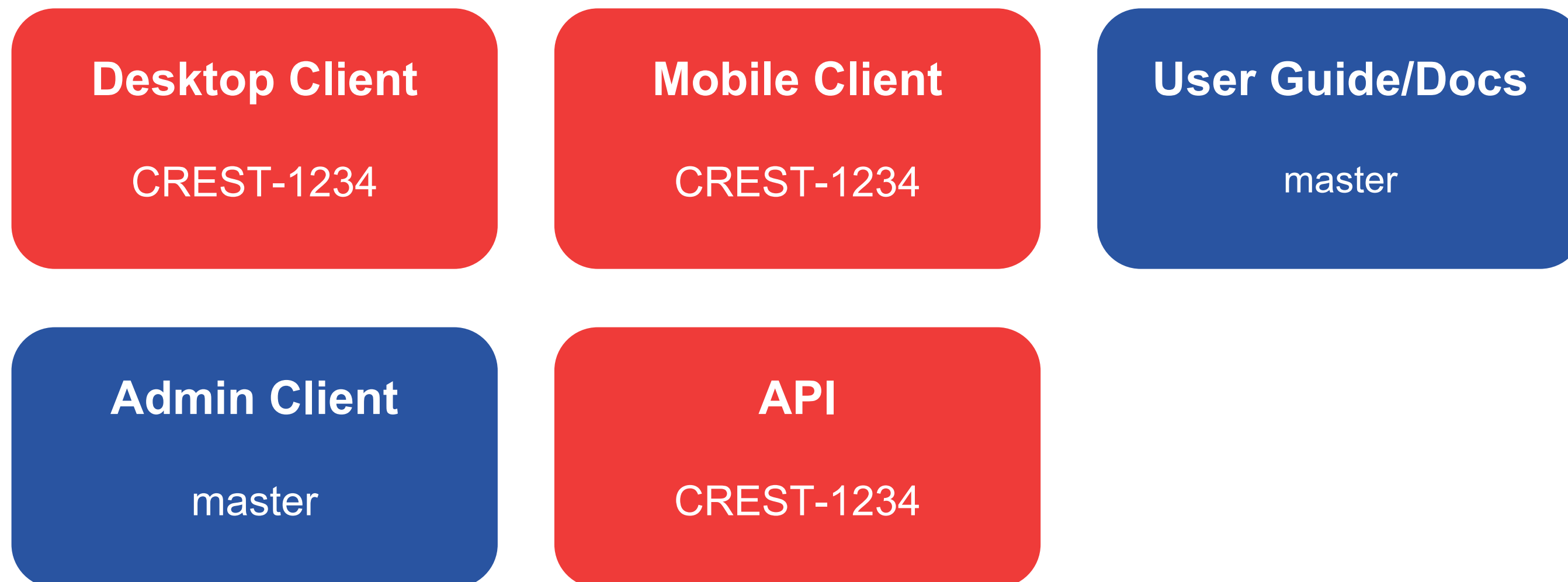
- All new work is done on only a single component





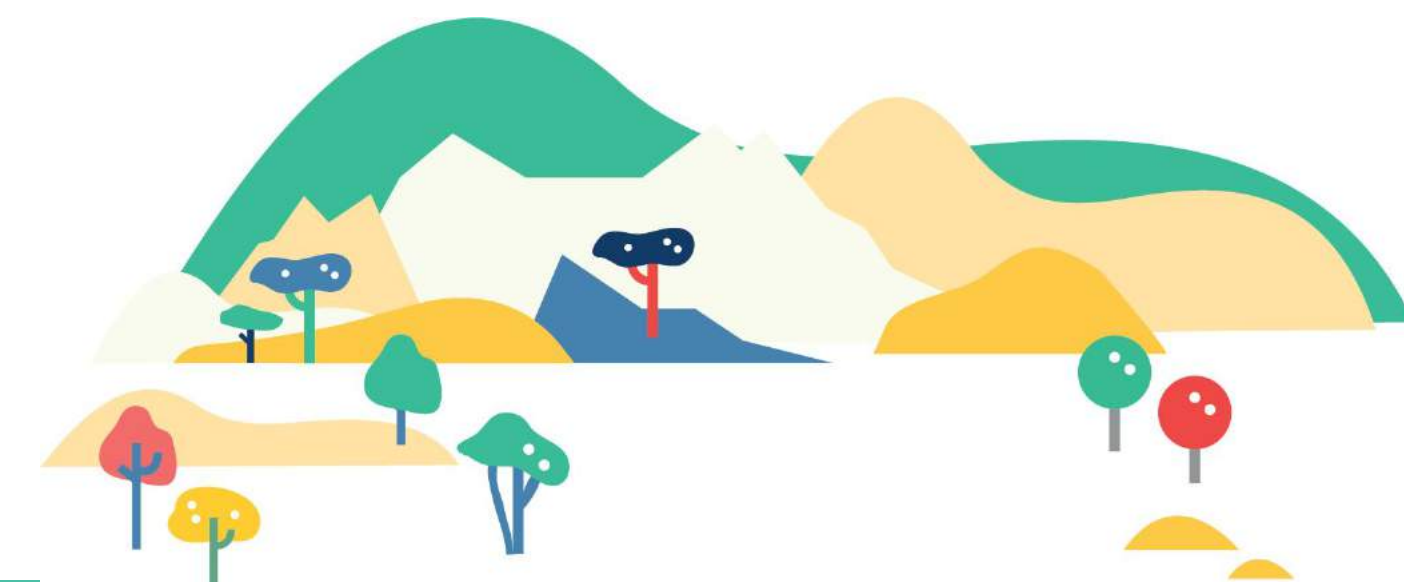
# Scenario 2...

- New work requires updates in multiple components



# With Multiple Services...

- Switching features is much more complicated
  - Source code is still in individual components
  - Environment is scattered across repositories



# A frequent problem

- We frequently noticed that we would change to a new branch, but not every component was updated

**Desktop Client**

CREST-1234

**Mobile Client**

CREST-1234

**User Guide/Docs**

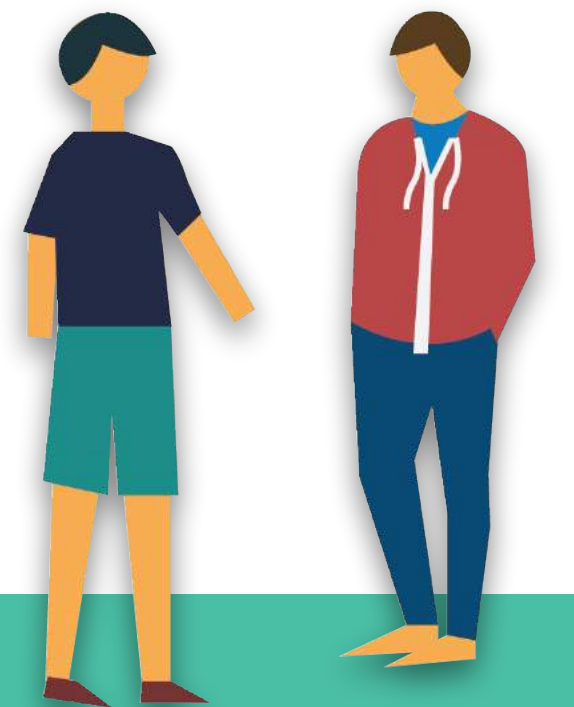
master

**Admin Client**

master

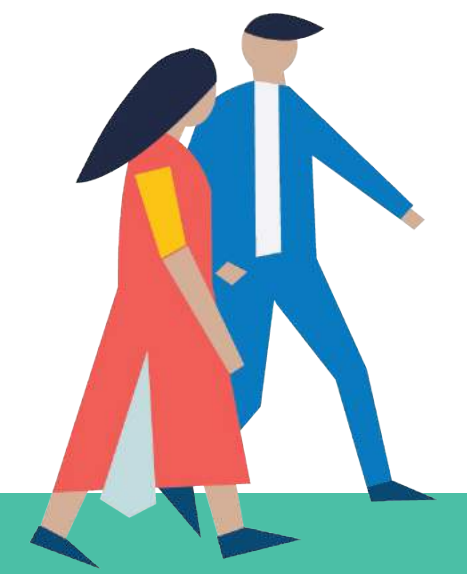
**API**

CREST-1111



# Our Objective

- We want an “App-in-a-Box!” The ability to...
  - spin up the entire application (with all components)
  - use the latest versions of each component, whether feature branch of latest **master** branch





# Our Sample App

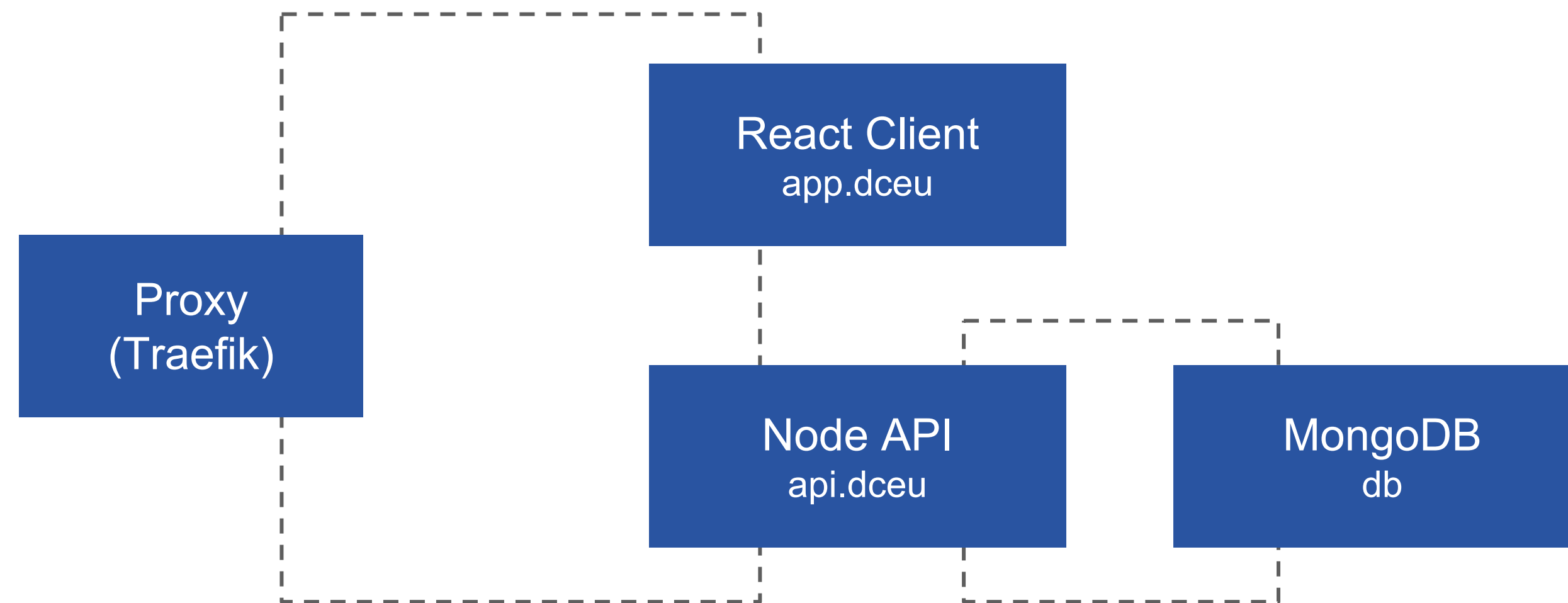
## Hello World

New item

☐ Be awesome!

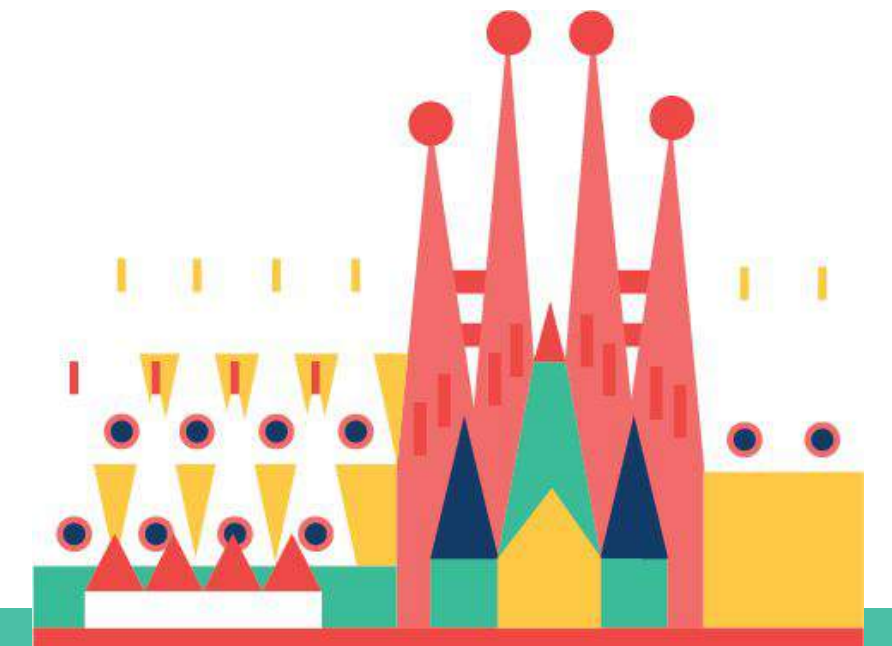
☐ Don't mess up the talk!

# The App's Components



# Our Feature

- Theme the app for DockerCon!
  - Change the welcome message (provided by API)
  - Apply a stylesheet to theme the client



**Let's check in with Sally!**



# Anyone impressed?



# The secret sauce!

```
version: "3.7"
services:
  api:
    image: summit/api:abcdef
    ...
  desktop-client:
    image: summit/desktop:defabc
    ...
  docs:
    image: summit/docs:abc123
    ...
  mobile-client:
    image: summit/mobile:123456
    ...
```

Pre-push to API

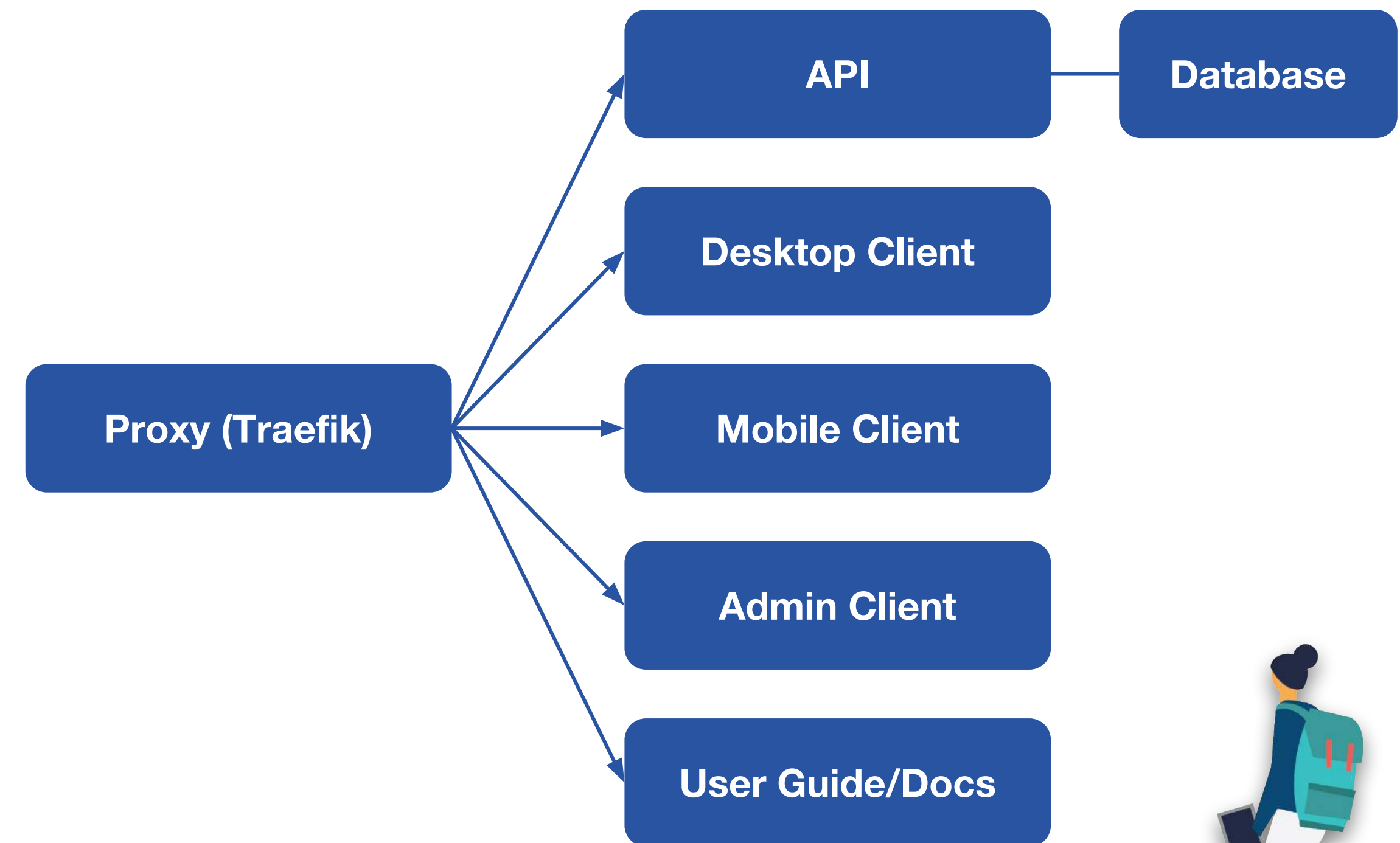


```
version: "3.7"
services:
  api:
    image: summit/api:234567
    ...
  desktop-client:
    image: summit/desktop:defabc
    ...
  docs:
    image: summit/docs:abc123
    ...
  mobile-client:
    image: summit/mobile:123456
    ...
```

Post-push to API

# Mirror your environment!

- If you're deploying to multiple hostnames, do the same locally!
- Stop worrying about port conflicts in local development
- We love using Traefik!





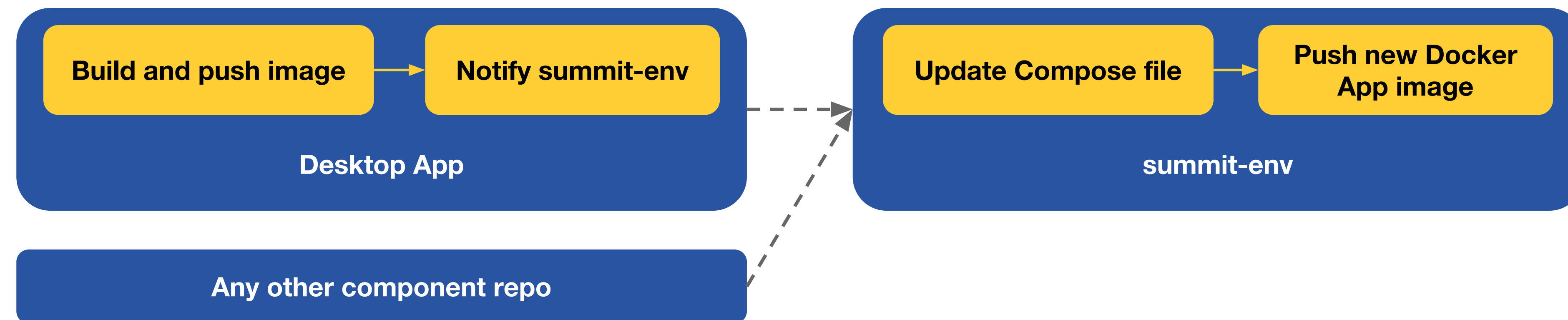
**Great, but how?**





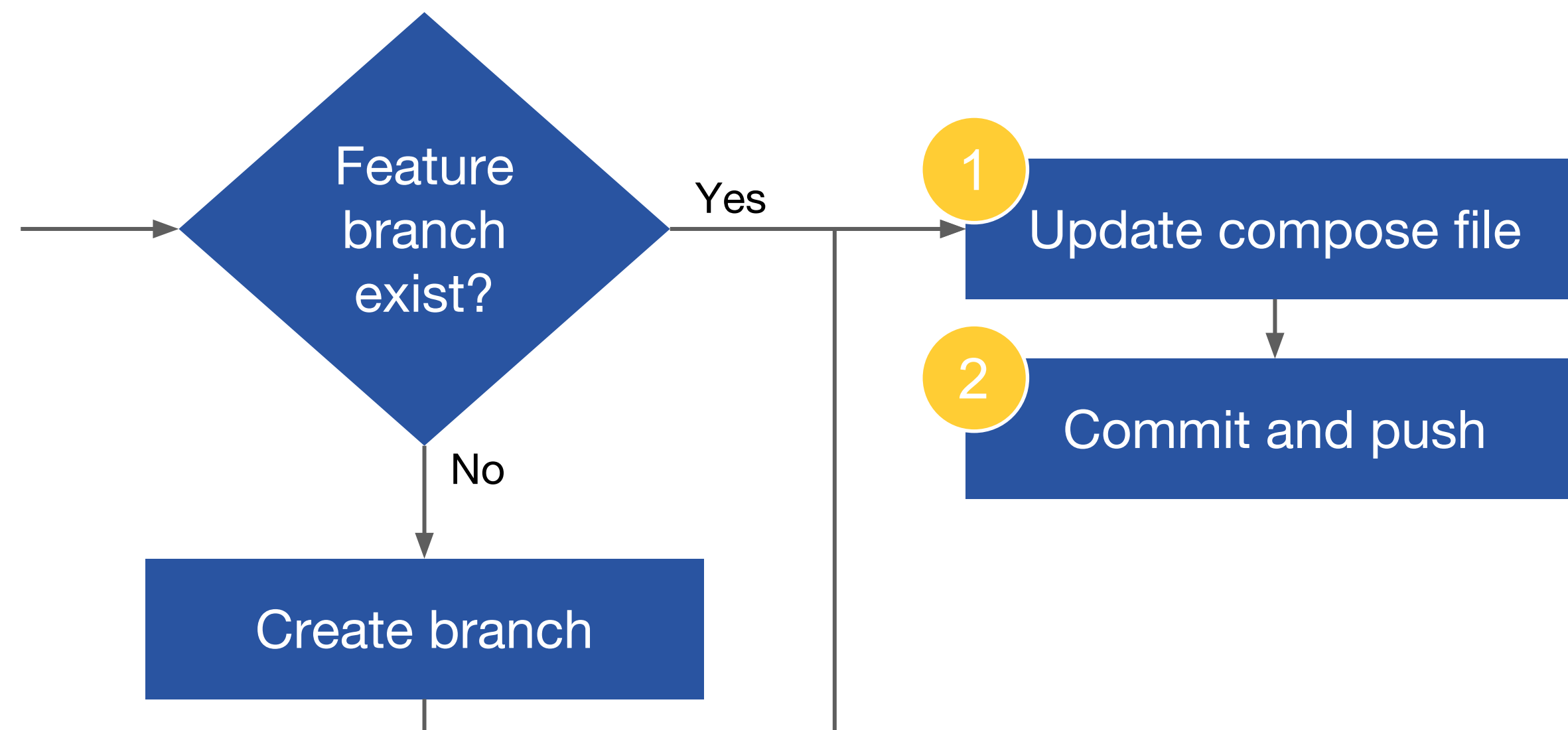
# Automatic updates

- Push “current environment” state into its own repo
- Use the CI pipelines to update env when any component is updated



# The Build Pipeline

Build triggered by upstream component  
on update branch



Build triggered by update to feature branch

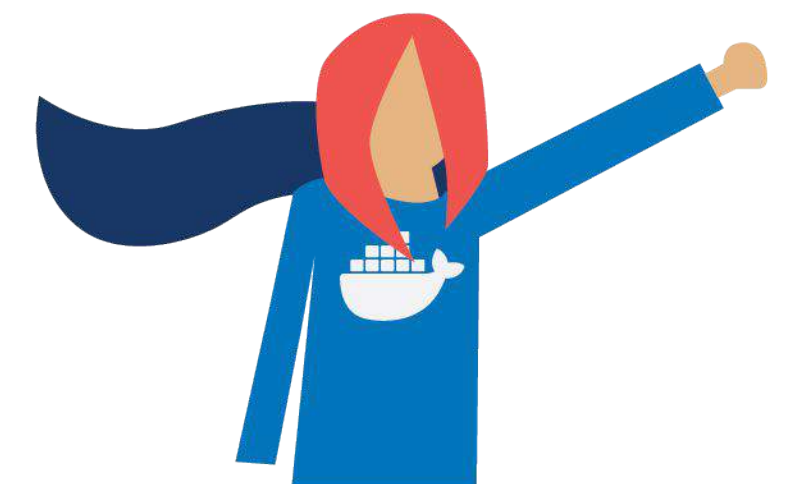






# Docker App

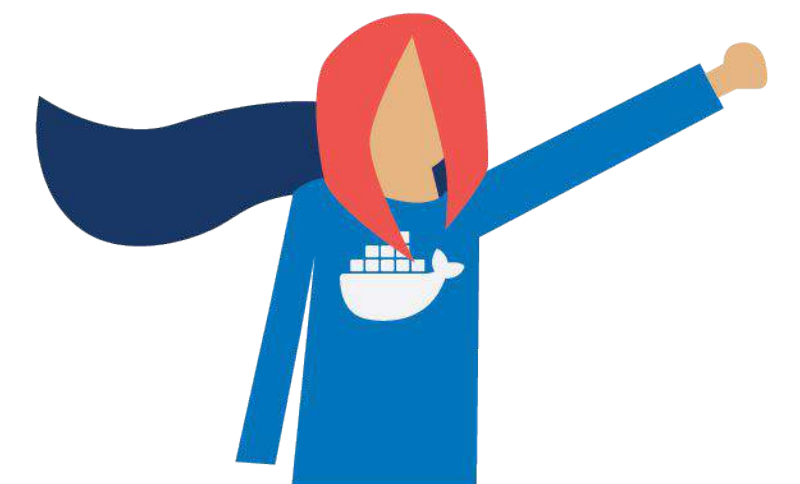
- An experimental utility designed to make compose files shareable and reusable
- Creates a Docker image containing:
  - Docker Compose file
  - Settings and metadata files
  - Other supporting files





# Docker App, cont'd.

- App shared through a registry, just as any other image
- Settings allow for configuration within the compose file
- When deploying, custom setting values can be applied
- Can generate Helm charts



# Docker App commands (a few anyways)

`docker-app init dceu`

Bootstraps a project

`docker-app push dceu`

Create an image using the local files and push it

`docker-app deploy mikesir87/dceu:master`

Pull the docker app and deploy it

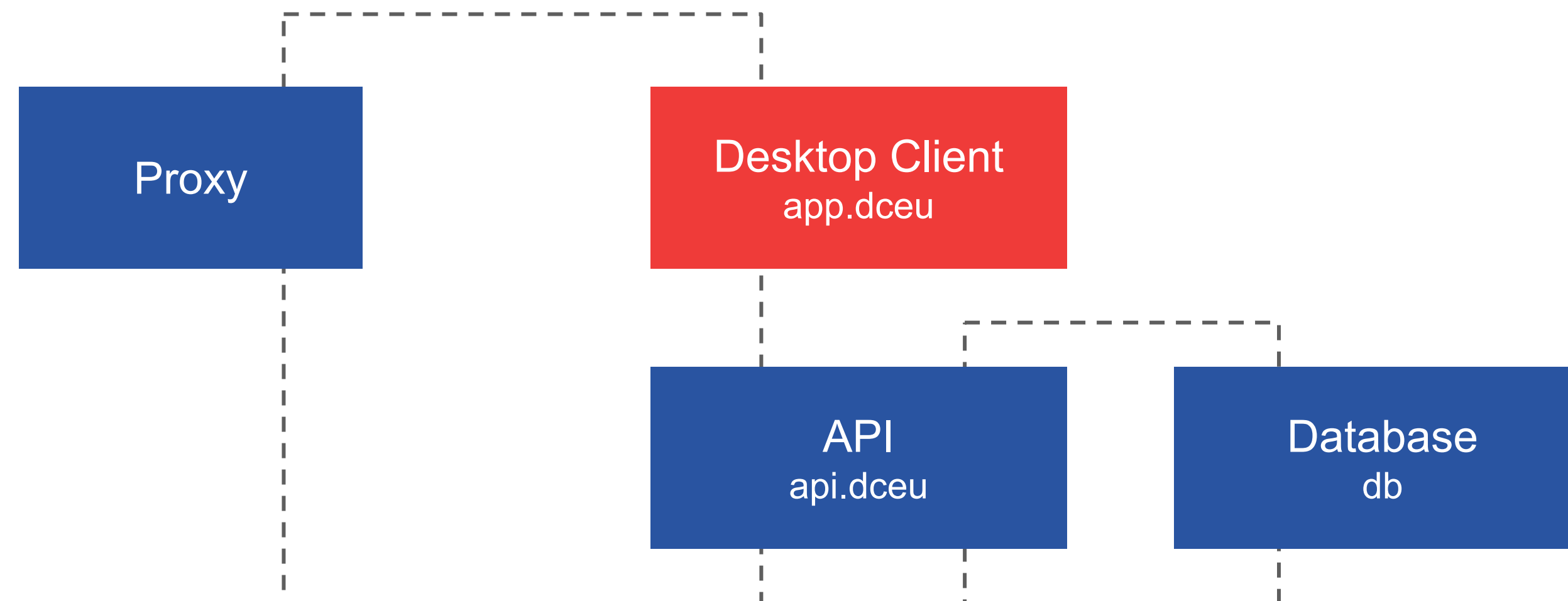
`docker-app render mikesir87/dceu.dockerapp:master`

Pull the app and output the Compose file to stdout

# Let's dive into an image!

# How's this change local dev?

# Local Dev Environment



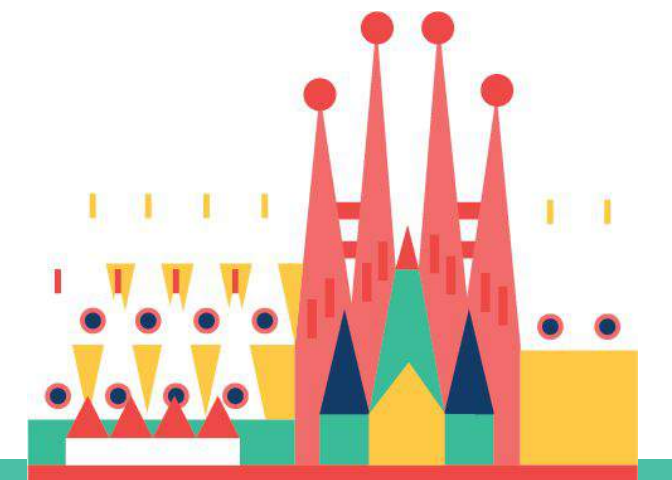


# Disabling a service

- Docker Compose file format 3.4+ allows extension fields
- Docker App observes specification of **x-enabled** on a service
- If the value is false, that service is NOT rendered/deployed

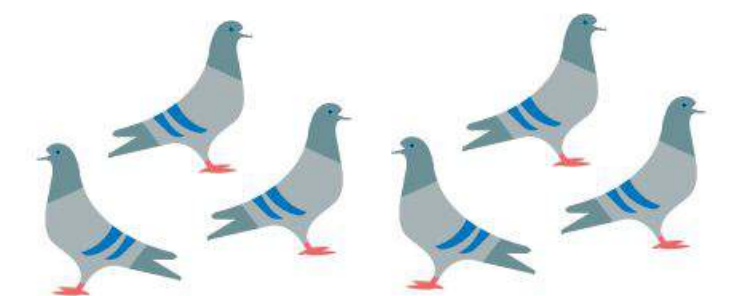
```
version: "3.7"

services:
  desktop-client:
    image: my-desktop-client-image
    x-enabled: ${enable-desktop-client}
    labels:
      traefik.frontend.rule: Host:app.localhost
    ...
```



# Hooking in a Dev Container

- Each component's **`docker-compose.yml`** will:
  - Start a dev-ready container
  - Use the same networks from the Docker App
  - Add labels to configure Traefik to send traffic



**Show me the demo again!**

# Great! What else can we do?

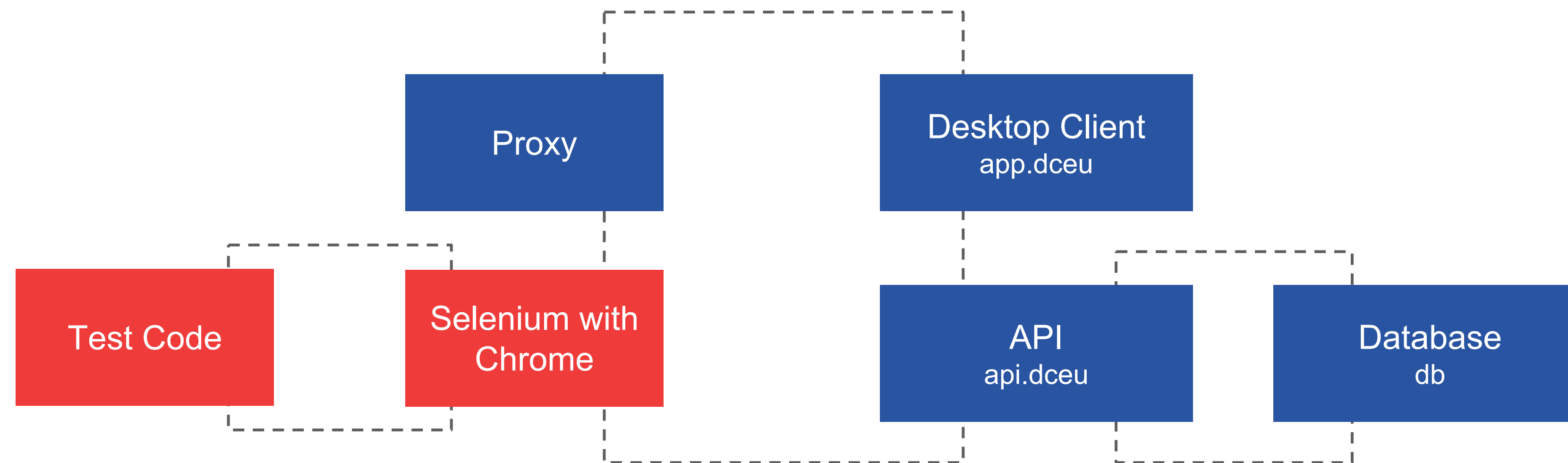
# Layering additional services

- With **docker-app render**, we can use other compose features
- **Feature One:** Docker Compose allows use of multiple compose files
  - Using multiple **-f** flags, we can add or modify services in the stack
  - NOTE: You can pipe the docker-app render directly to compose. But, you can't use a piped file AND a local file together.



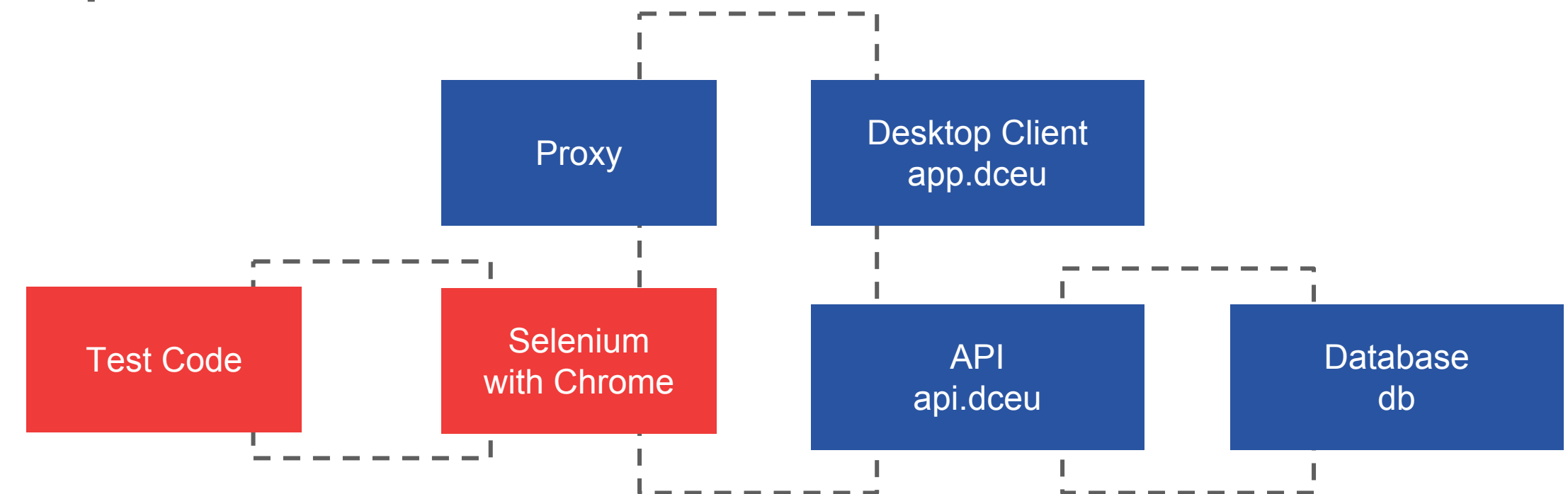


# Layering in test containers



# Running Tests with Docker

- **Feature Two:** Usage of `--exit-code-from` will:
  - Spin up all containers in the stack
  - Wait for the specified service to exit
  - Relay exit code from service as result of compose run



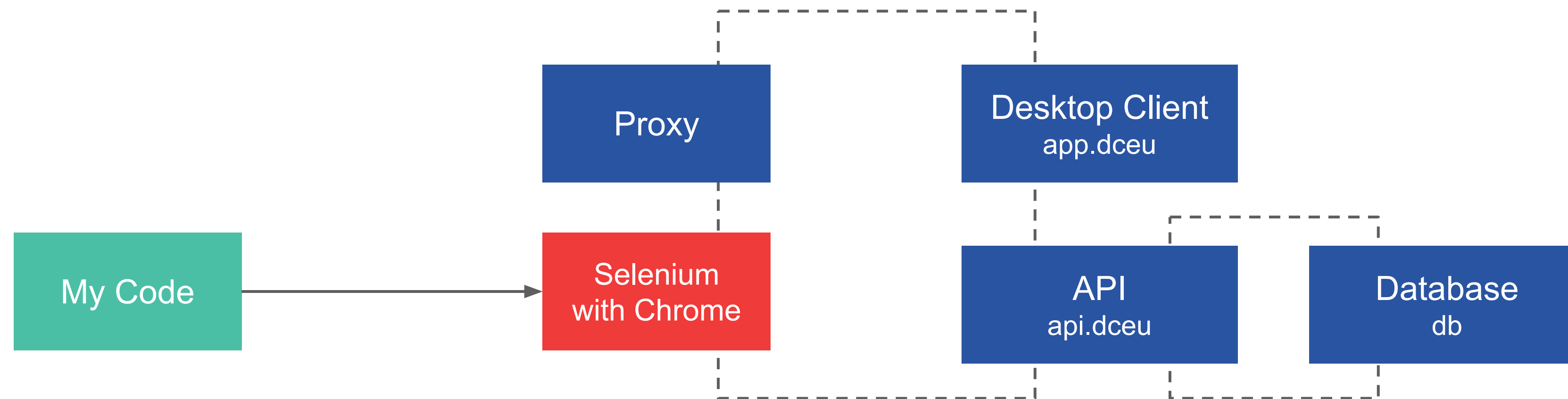
# Demo Time!

# Tests failed! Let's fix them!



# Fixing the Tests

- Spin up the Selenium service
- Write tests
- Point tests to go against the Selenium service



# Generalizing things

- We noticed we frequently wanted to disable services for local development
- Keeping track of the settings to disable services is a pain
- Wouldn't it be nice to have a tool that asked us what services we want disabled and kept track of the settings?



# DevDock

- With DevDock, we add additional fields to each service in the Docker App:
  - **x-devdock-description** - a human readable description of the service
  - **x-devdock-setting** - the name of the settings that must be set to false to disable the service

```
version: "3.7"
services:
  api:
    image: my-api-image
    x-enabled: ${enable-api}
    x-devdock-description: API/Backend
    x-devdock-setting: enable-api
    ...
  desktop-client:
    image: my-desktop-client-image
    x-enabled: ${enable-client}
    x-devdock-description: Desktop Client
    x-devdock-setting: enable-client
    ...
```



# DevDock, cont'd.

Usage: **devdock [project] [dockerapp-image]**

- Two ways to run the cli:
  - No additional args - presents a cli-based UI to allow selection of services to disable
  - Additional args - passes through to an underlying Docker Compose command that's running on the rendered output

# Demo Time!

# One more thing!

- Create your own team env CLI by aliasing **devdock**!

```
alias dceu="devdock dceu mikesir87/dceu.dockerapp:{BRANCH} $@"  
dceu up -d  
dceu logs -f  
dceu down
```

# How to install DevDock?

- *Currently* distributed as a global NPM module.
  - So... just npm install!

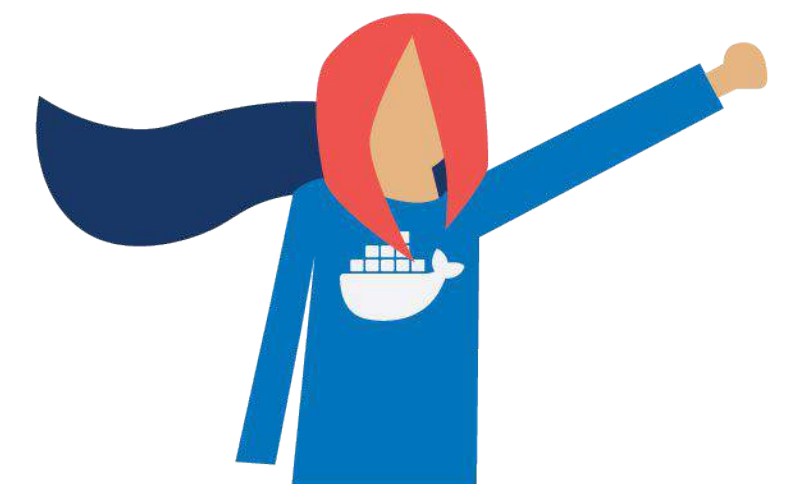
```
npm install -g devdock
```





# Recap

- Docker Compose is incredible and has great features!
- Docker App is awesome and makes local dev MUCH better!
- You can use extension fields to build tools to help your devs



# Testimonials



“I am getting better gas mileage in my Jeep now.”  
- **Jeff Mitchell**



“Summit-in-a-box healed my relationship with my wife and kids”  
- **Justin Boblitt**

# [Actual] Testimonials



“SIAB has made it even easier for me to spin up or disable different components in my stack as testing requirements change.”

- **Jeff Mitchell**



“Thumbs are raised. Summit-in-a-Box gives me more confidence that I'm running the latest versions of each component, preventing bugs and helping isolate new bugs I've introduced. Summit-in-a-Box hides the fact that we're even sharing docker-compose files.”

- **Justin Boblitt**

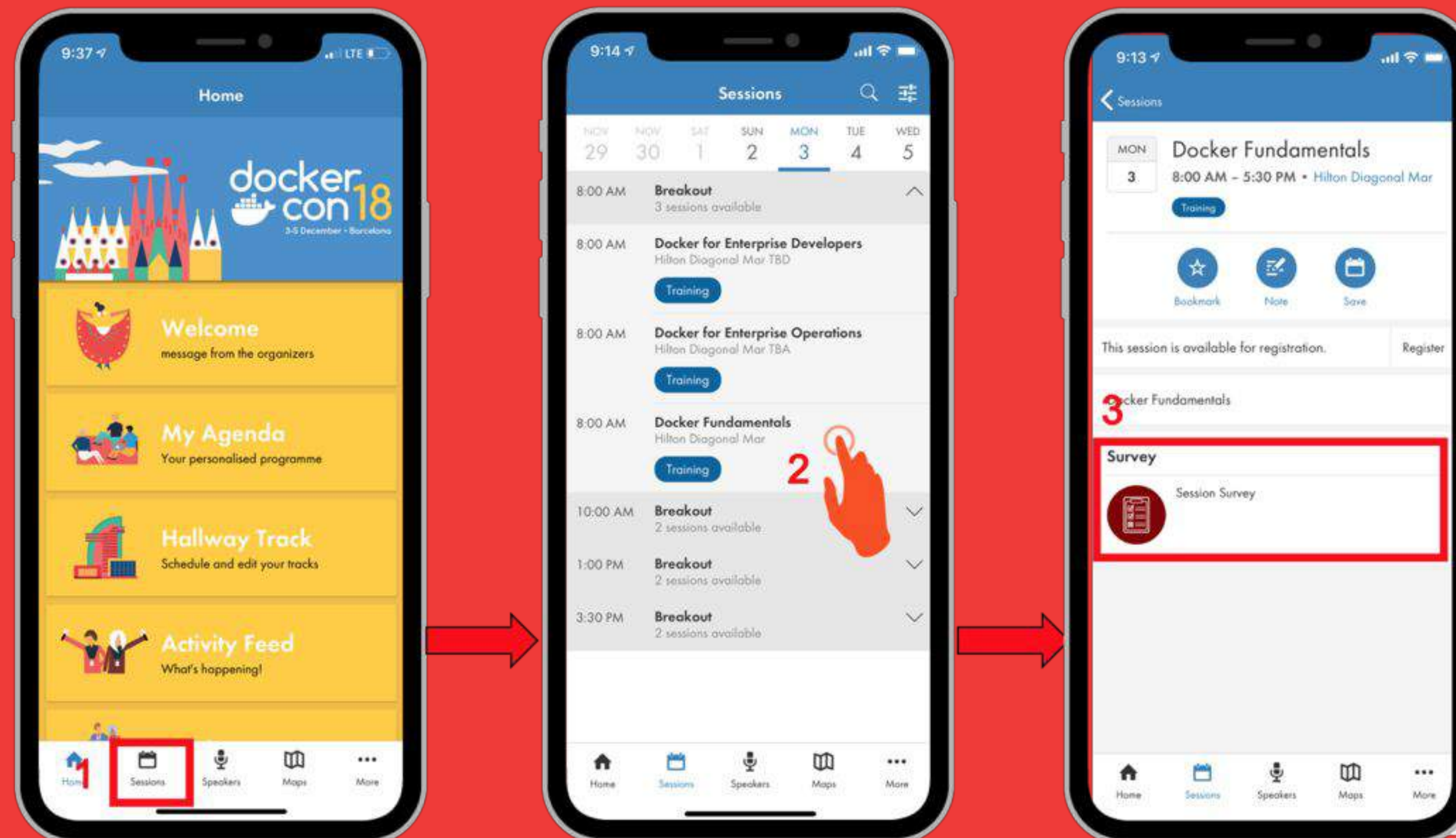
# Thanks!



# Take A Breakout Survey

Access your session and/or workshop surveys for the conference at any time by tapping the Sessions link on the navigation menu or block on the home screen.

Find the session/workshop you attended and tap on it to view the session details. On this page, you will find a link to the survey.



# Questions?