Pivotal®

Intro to Cloud Native Buildpacks

Anand Rao, Platform Architect
Scott Deeg, Platform Architect

May 2019

Agenda

- What are CN Buildpacks?
- Current Solutions
- How they work (+demo)
- Decomposition (+demo)
- Build Services

What are Cloud Native Buildpacks?

Pluggable, modular tools that translate source code into OCI images.

Buildpacks.io

- Portability via the OCI standard
- Greater modularity
- Faster builds
- Reproducible image builds
- Unprivileged containers
- Developed in partnership with Heroku
- CNCF project

The Goal

Write code

Run apps

Minimal fuss

This has been the goal of app dev teams for a long (long) time.

Containers to the Rescue?

While container technology has been around, Docker brought them to the masses. Having standardized containers is super useful, but they not the magic bullet they've been hyped to be.

- Creating non-trivial, production grade Dockerfiles is actually really hard and requires a very low level understanding what's happening
- Choosing the right image to start with has many implications (security, upgrades, etc.)
- Build images not the same as what you want for runtime
- Teams create variations which is error prone, results in inconsistent quality
- Composability was a challenge
  - Don't follow symlinks, copying files manually, no environment
- Ensuring security was a challenge
- Even if you do a good job, they drift (rot?) over time requiring constant maintenance
  - Another great source of errors
- Day 2 ops became a nightmare

“Writing a quality Dockerfile is still my users’ biggest point of friction”

- David Dollar - CEO, Convox (a k8s/Docker PaaS)

“Containers are not always a good abstraction for developers - they force you to learn about and think about very low level concerns”

- Dave Syer

CN Buildpack Features

Buildpacks provide a variety of services to create an appropriate build and run environment

- Application detection/validation
- Vend dependant components (eg: middleware stack)
- Put application in a “Stack” (OS base)
- Create a configuration
- Define how the application starts
- Output the container
- Use smart caching to enable builds and runtime images

- Application Binary Interface
- Cross repository blob mounting
- Fast is a feature!

- Local development

New Buildpack API

Detect

where an optimal selection of compatible buildpacks is chosen and a build plan is created

Analysis

where metadata about OCI layers generated during a previous build are made available to buildpacks

Build

where buildpacks use that metadata to generate only the OCI layers that need to be replaced

Export

where the remote layers are replaced by the generated layers

Components

- Platform
  - pack - Local CLI for CNB
  - knative-integration - template for using CNB with knative/tekton
- Implementation
  - lifecycle - Implementation of the Buildpack API v3
  - libbuildpack - Go language binding for the CNB API
- Core
  - spec - Buildpack API v3 specification
  - rfcs - RFCs for changes to CNB

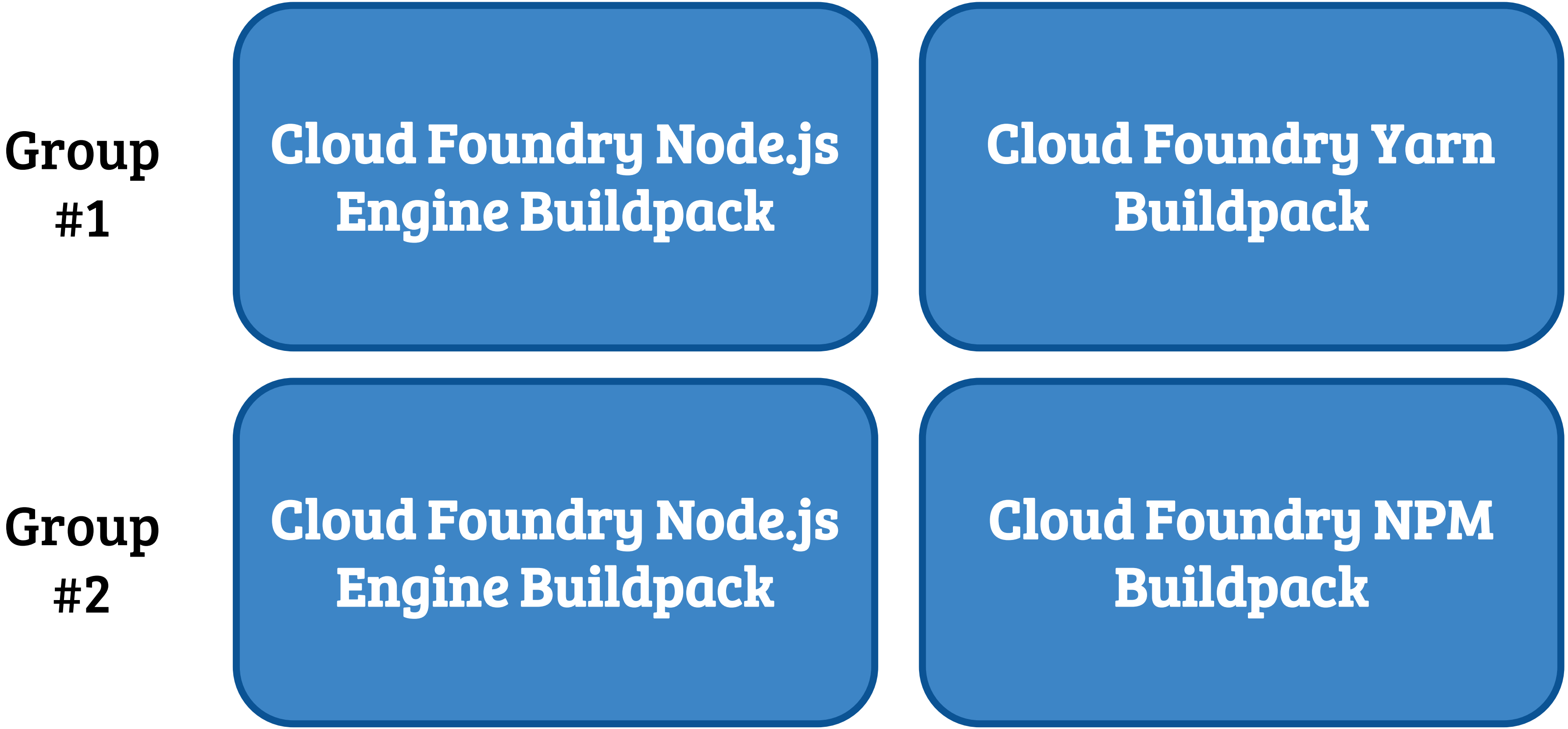
Demo

Decomposition

Cloud Foundry Buildpacks as
modular, transparent, source code processors

Cloud Foundry Node.js Buildpack (v2)

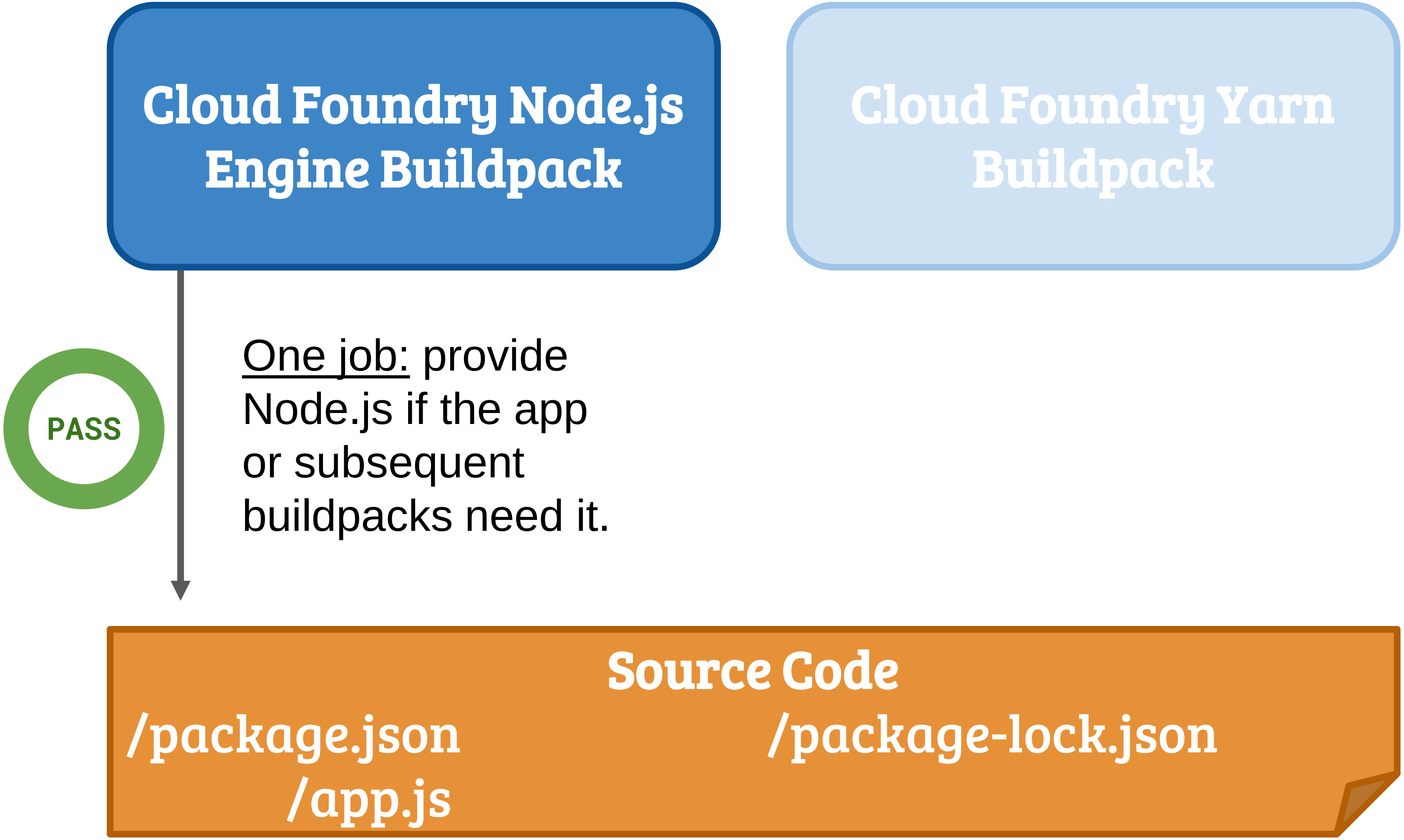
Build Configuration (for detection step)



Group #1	Cloud Foundry Node.js Engine Buildpack	Cloud Foundry Yarn Buildpack
Group #2	Cloud Foundry Node.js Engine Buildpack	Cloud Foundry NPM Buildpack

Buildpack Detection

A mechanism for automated buildpack, tool, and dependency selection



Cloud Foundry Node.js Engine Buildpack

Cloud Foundry Yarn Buildpack

PASS

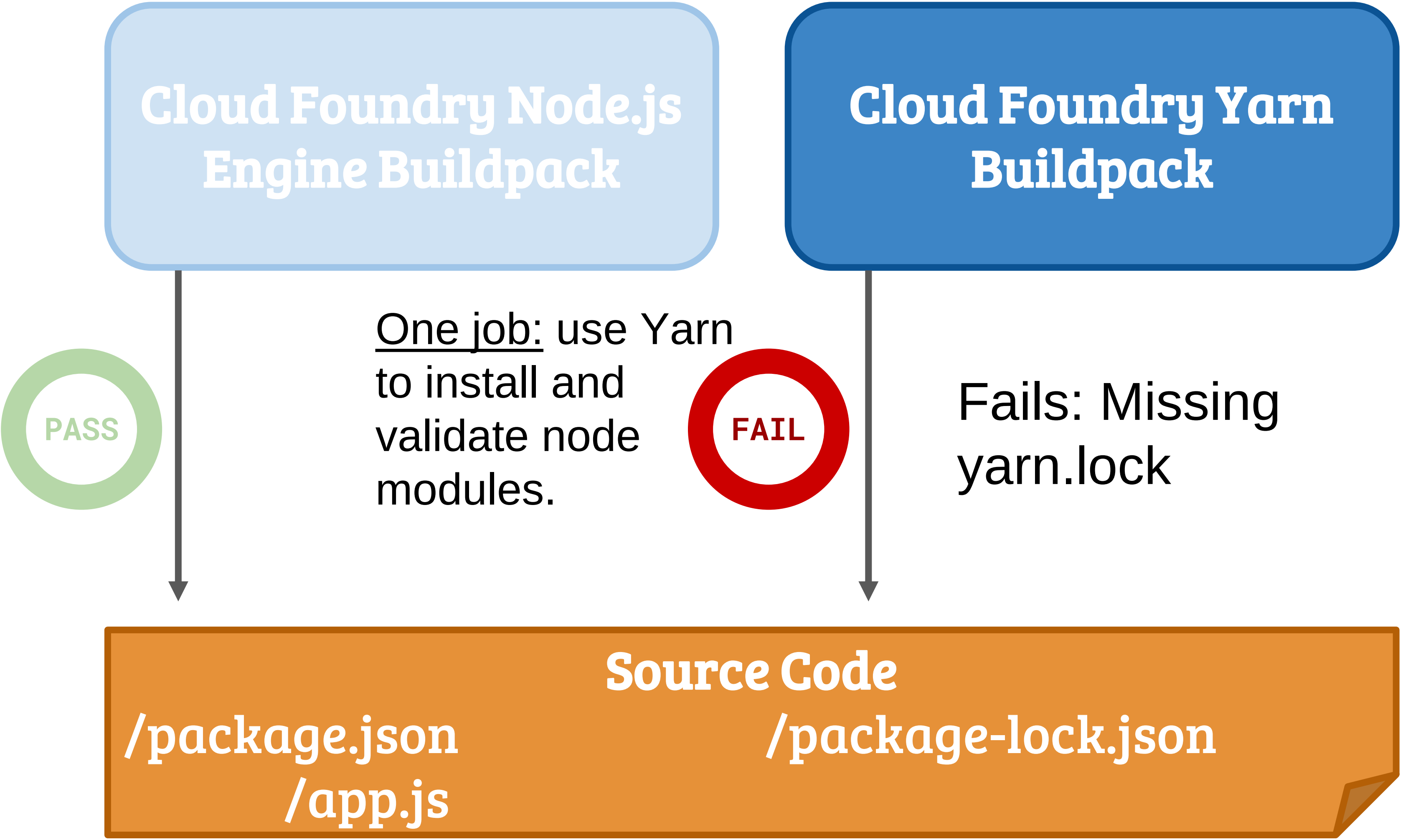
One job: provide Node.js if the app or subsequent buildpacks need it.

Source Code

/package.json

/package-lock.json

/app.js



Cloud Foundry Node.js Engine Buildpack

Cloud Foundry Yarn Buildpack

PASS

One job: use Yarn to install and validate node modules.

FAIL

Fails: Missing yarn.lock

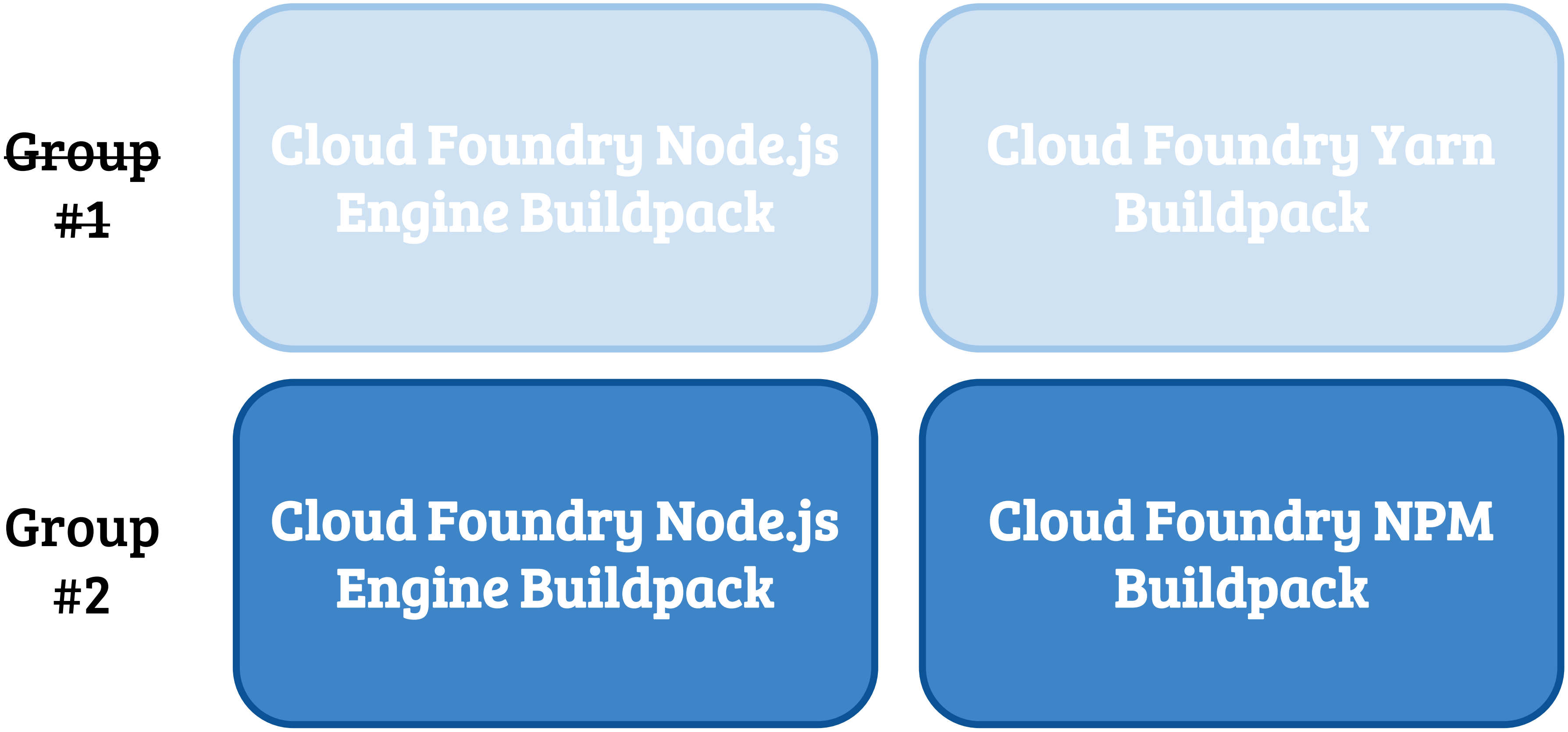
Source Code

/package.json

/package-lock.json

/app.js

Build Configuration (for detection step)



~~Group #1~~

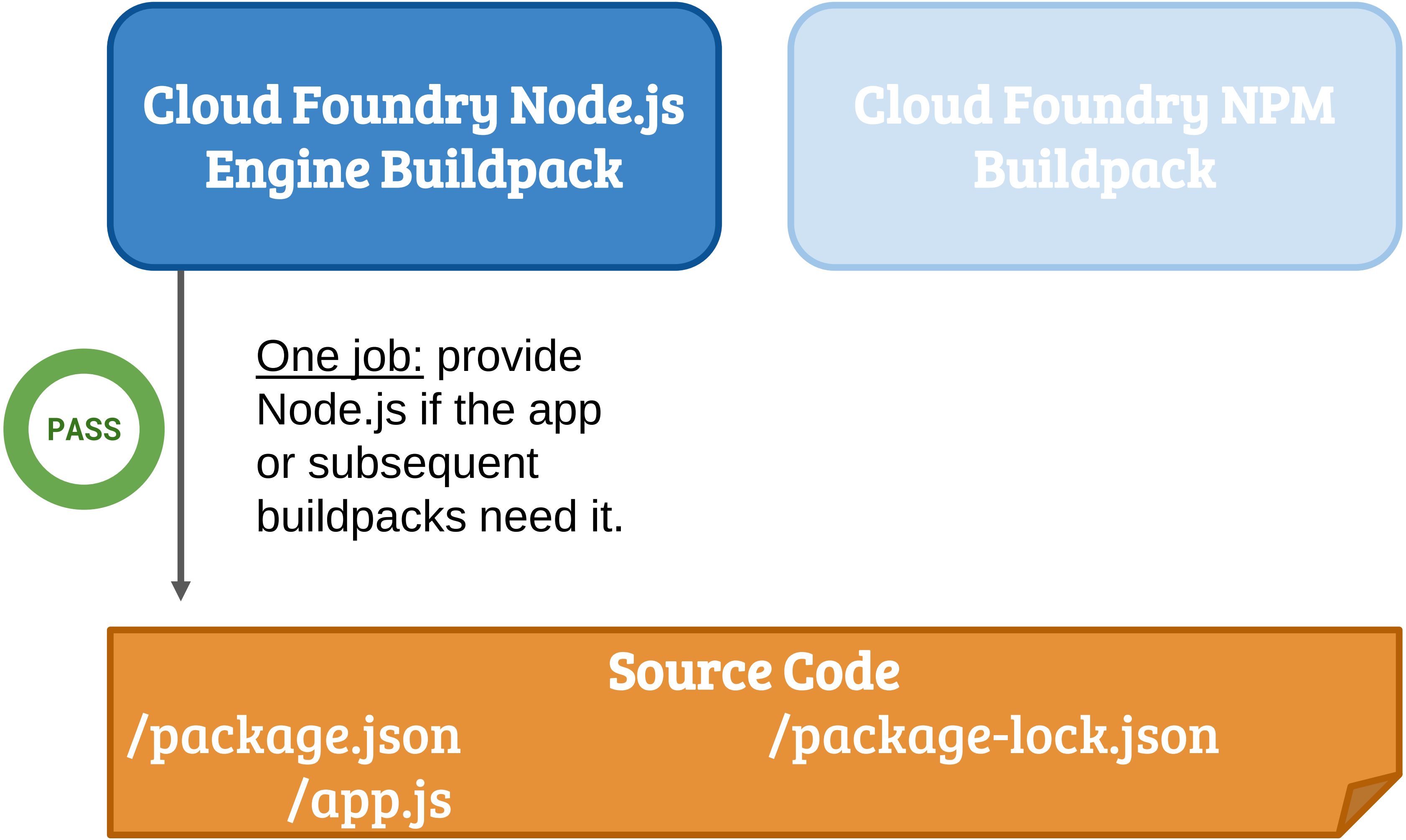
Cloud Foundry Node.js Engine Buildpack

Cloud Foundry Yarn Buildpack

Group #2

Cloud Foundry Node.js Engine Buildpack

Cloud Foundry NPM Buildpack



Cloud Foundry Node.js Engine Buildpack

Cloud Foundry NPM Buildpack

PASS

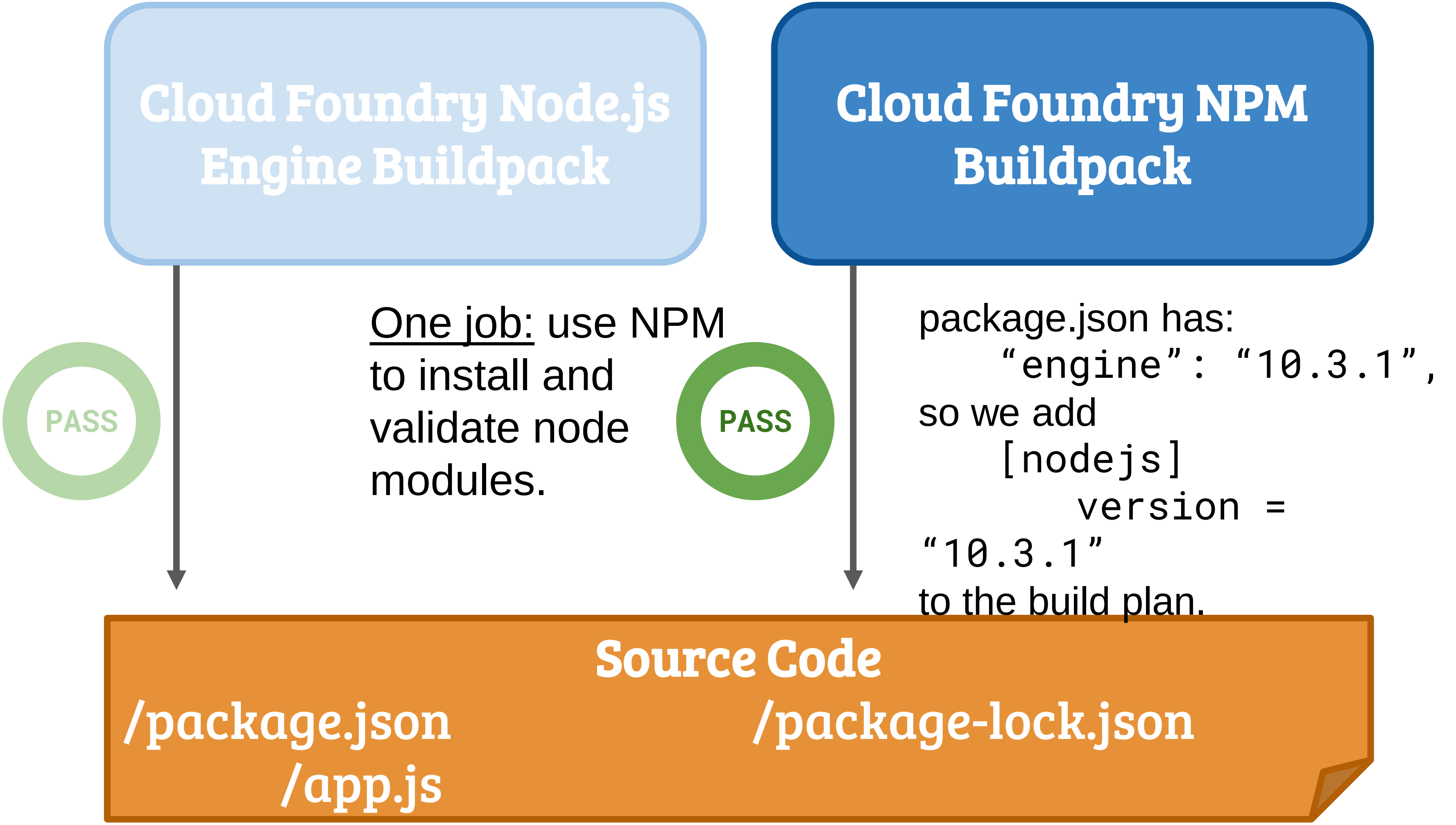
One job: provide Node.js if the app or subsequent buildpacks need it.

Source Code

/package.json

/package-lock.json

/app.js



Cloud Foundry Node.js Engine Buildpack

Cloud Foundry NPM Buildpack

PASS

PASS

One job: use NPM to install and validate node modules.

package.json has:
    “engine”: “10.3.1”,
so we add
    [nodejs]
        version = “10.3.1”
to the build plan.

Source Code

/package.json

/package-lock.json

/app.js

Build Configuration (for detection step)

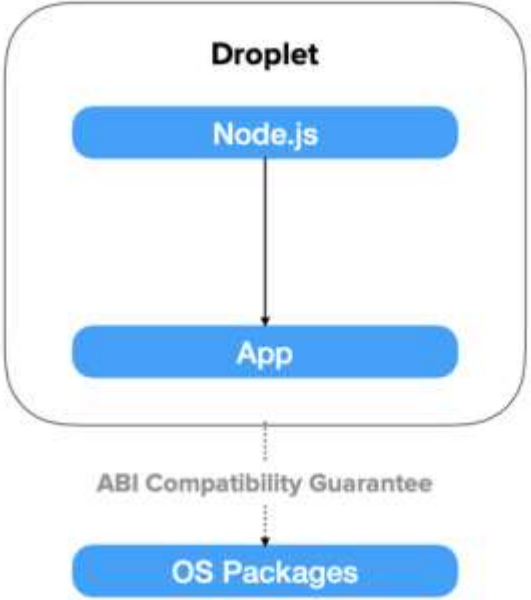
~~Group #1~~	Cloud Foundry Node.js Engine Buildpack	Cloud Foundry Yarn Buildpack
Group #2	Cloud Foundry Node.js Engine Buildpack	Cloud Foundry NPM Buildpack

Demo

Updating OS Layer

A mechanism for fast updates

Node.js App



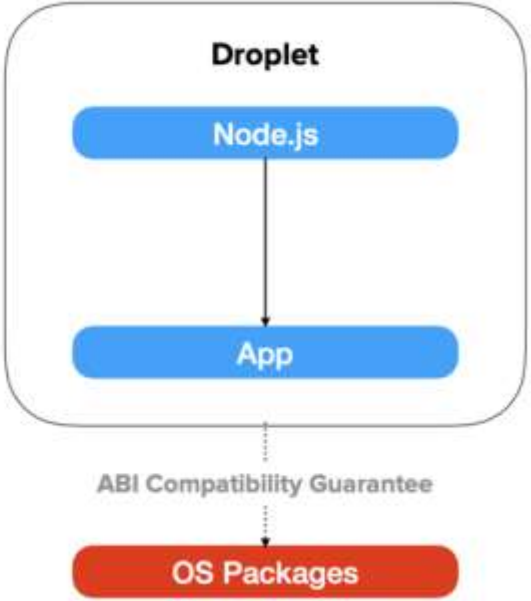
`buildpack: nodejs-buildpack`

```
rootfs: cflinuxfs3
(with OpenSSL)
```

ABI Compatibility?

- ABI = Application **Binary** Interface
- Operating System vendors (ex. Canonical for Ubuntu) guarantee a degree of compatibility when patching
- cflinuxfs3 is based on Ubuntu 18.04 LTS

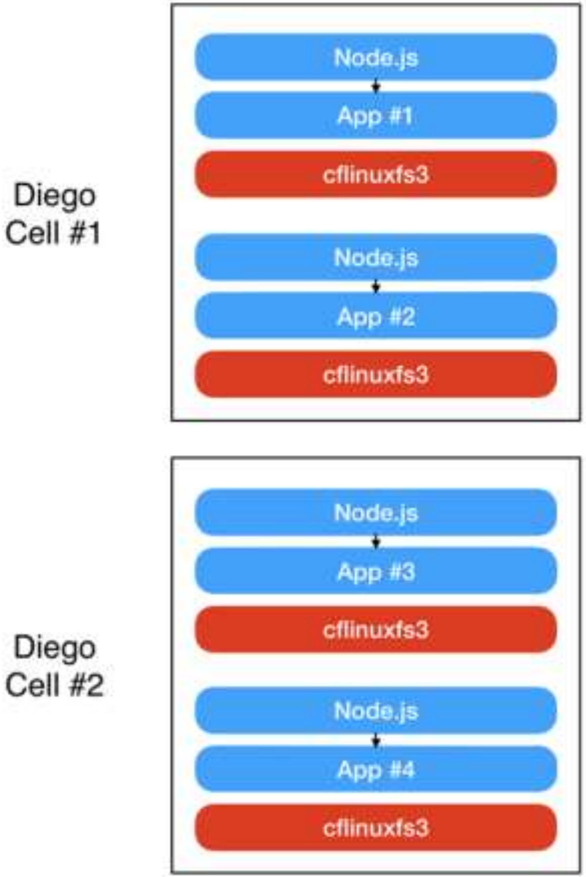
Node.js App



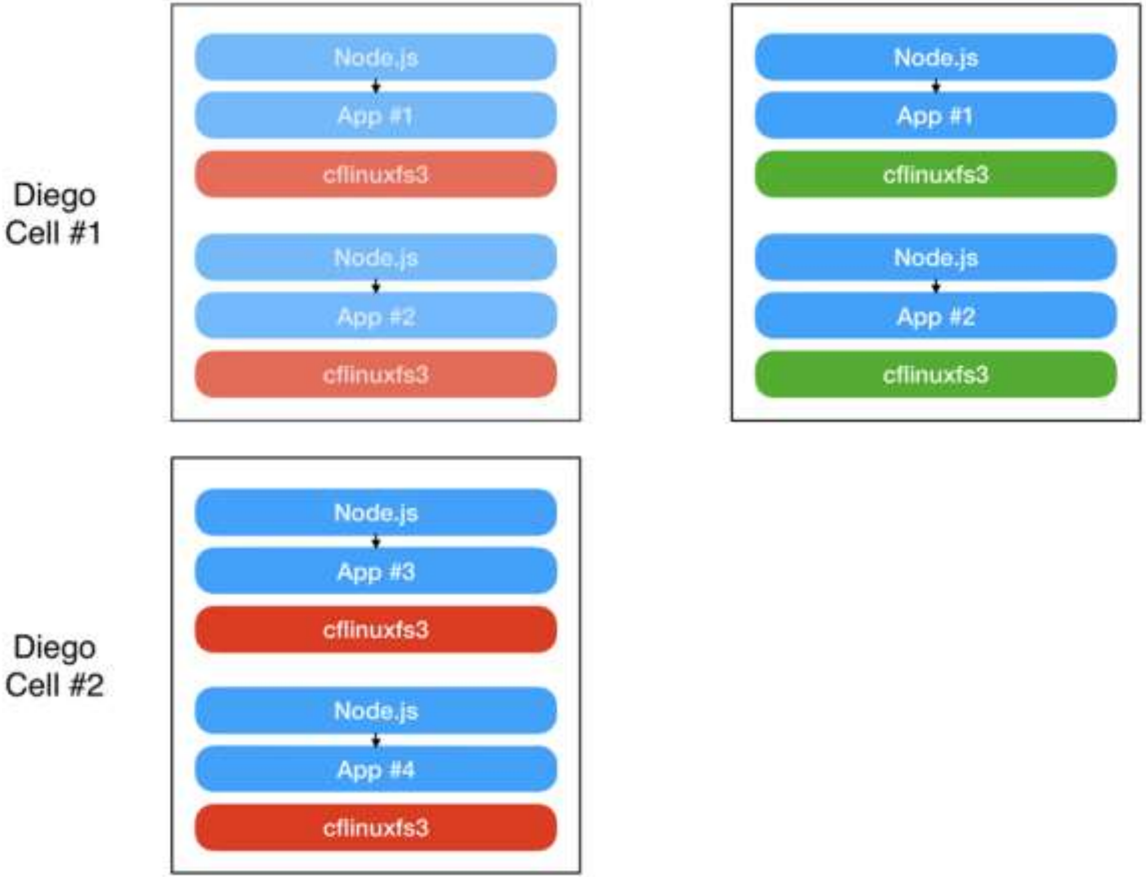
```
buildpack: nodejs-buildpack
```

```
rootfs: cflinuxfs3
(with OpenSSL)
```

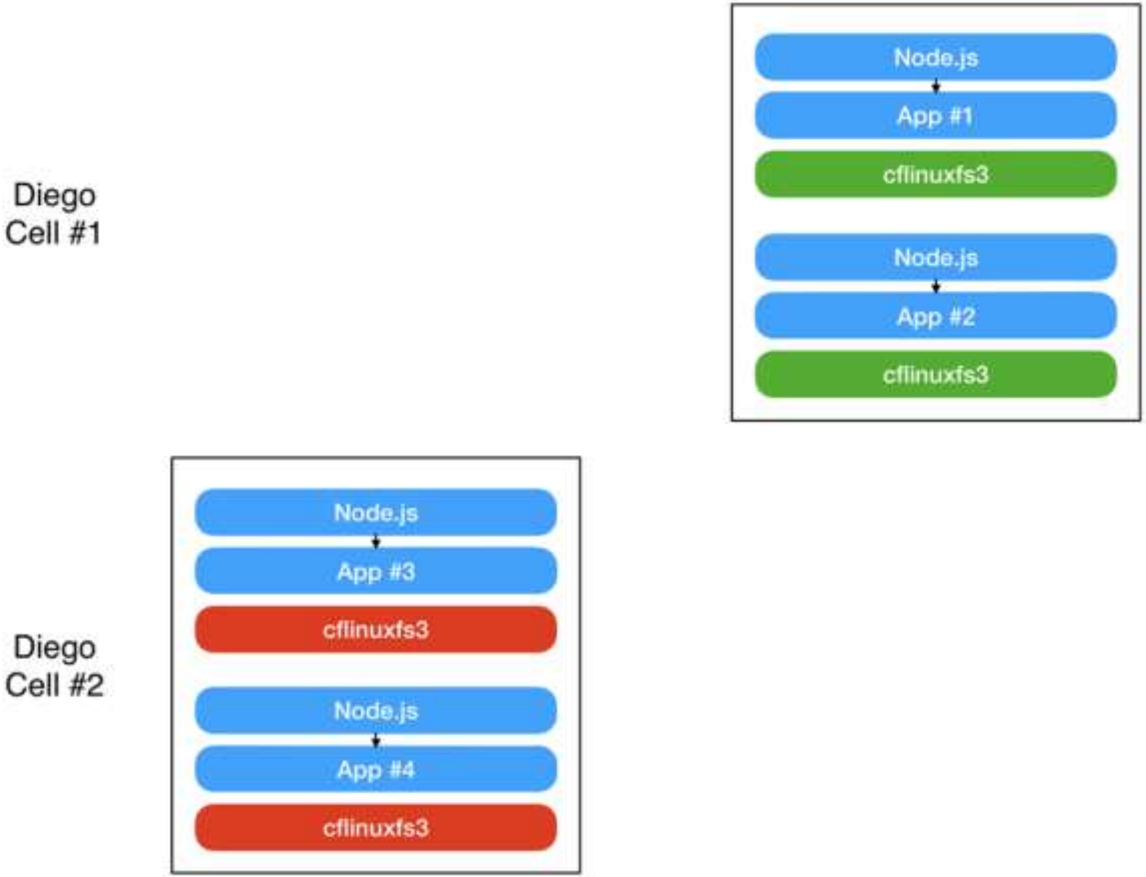
Cloud Foundry



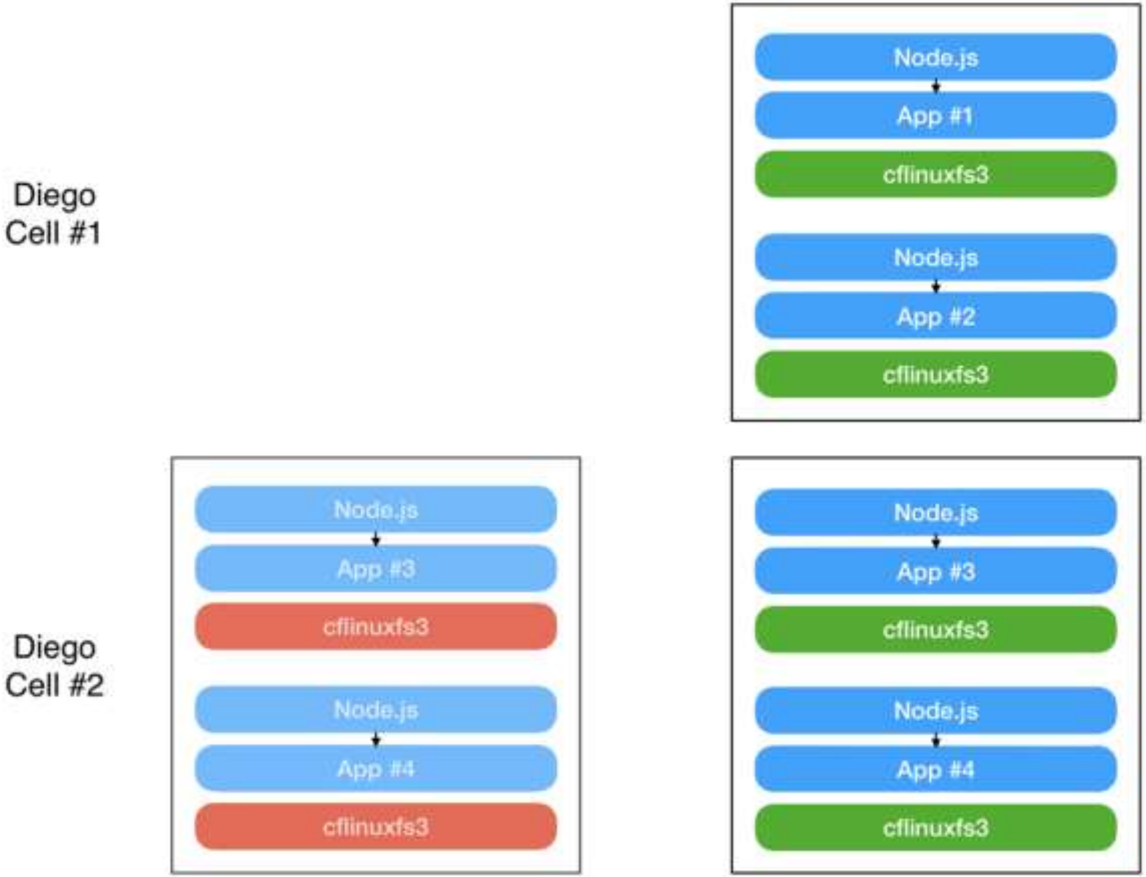
Cloud Foundry



Cloud Foundry



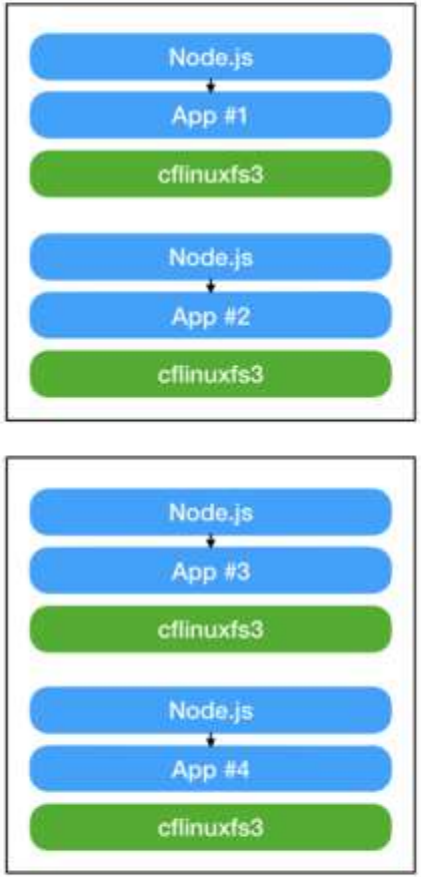
Cloud Foundry



Cloud Foundry

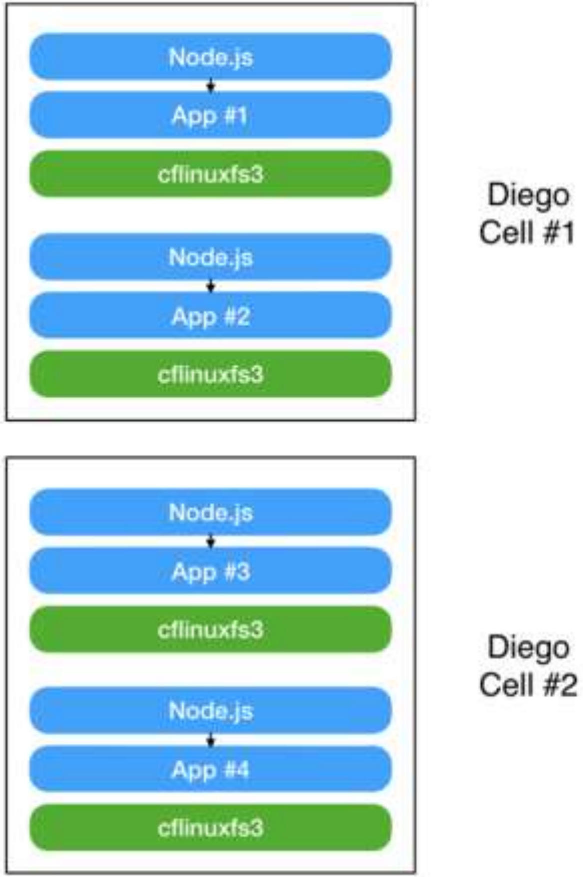
Diego
Cell #1

Diego
Cell #2



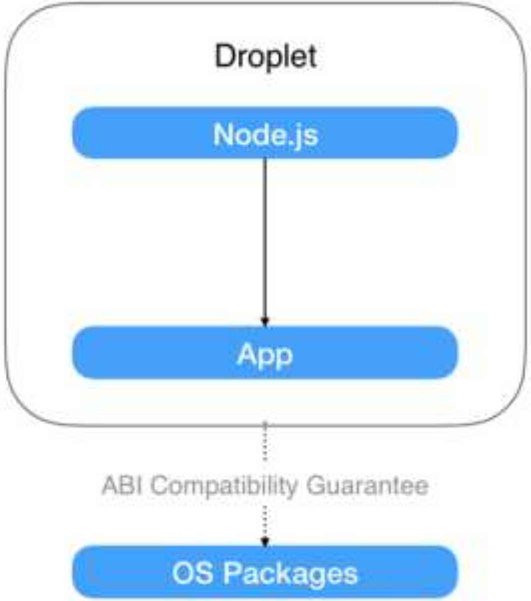
Cloud Foundry

<u>ETA to mitigation:</u>
Time roll cells,
usually a few hours



Buildpacks.io

Node.js App



`buildpack: nodejs-buildpack`

`rootfs: cflinuxfs3`

Node.js App

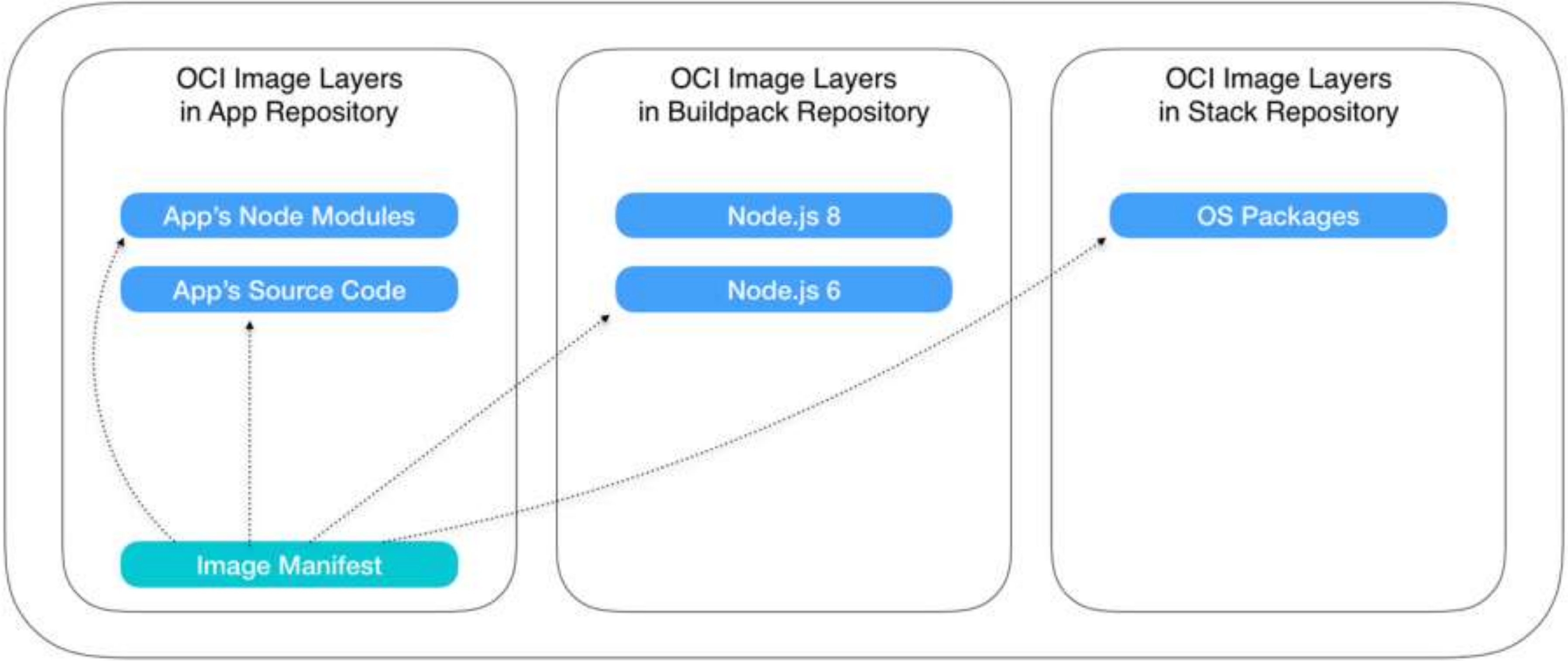


Image Registry (with OCI image and cross-repo blob mount support)

Node.js App

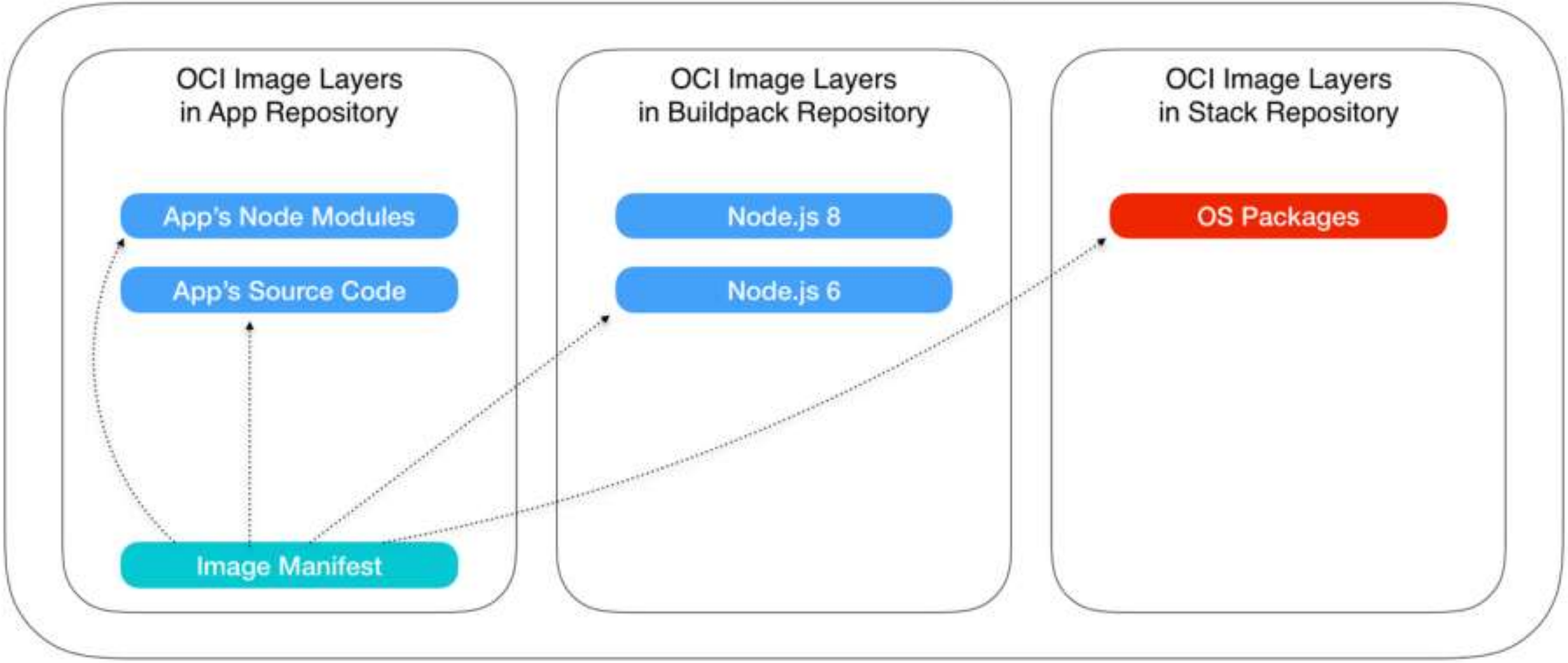

OCI Image Layers in App Repository
App's Node Modules
App's Source Code
Image Manifest
OCI Image Layers in Buildpack Repository
Node.js 8
Node.js 6
OCI Image Layers in Stack Repository
OS Packages

Image Registry (with OCI image and cross-repo blob mount support)

Node.js App

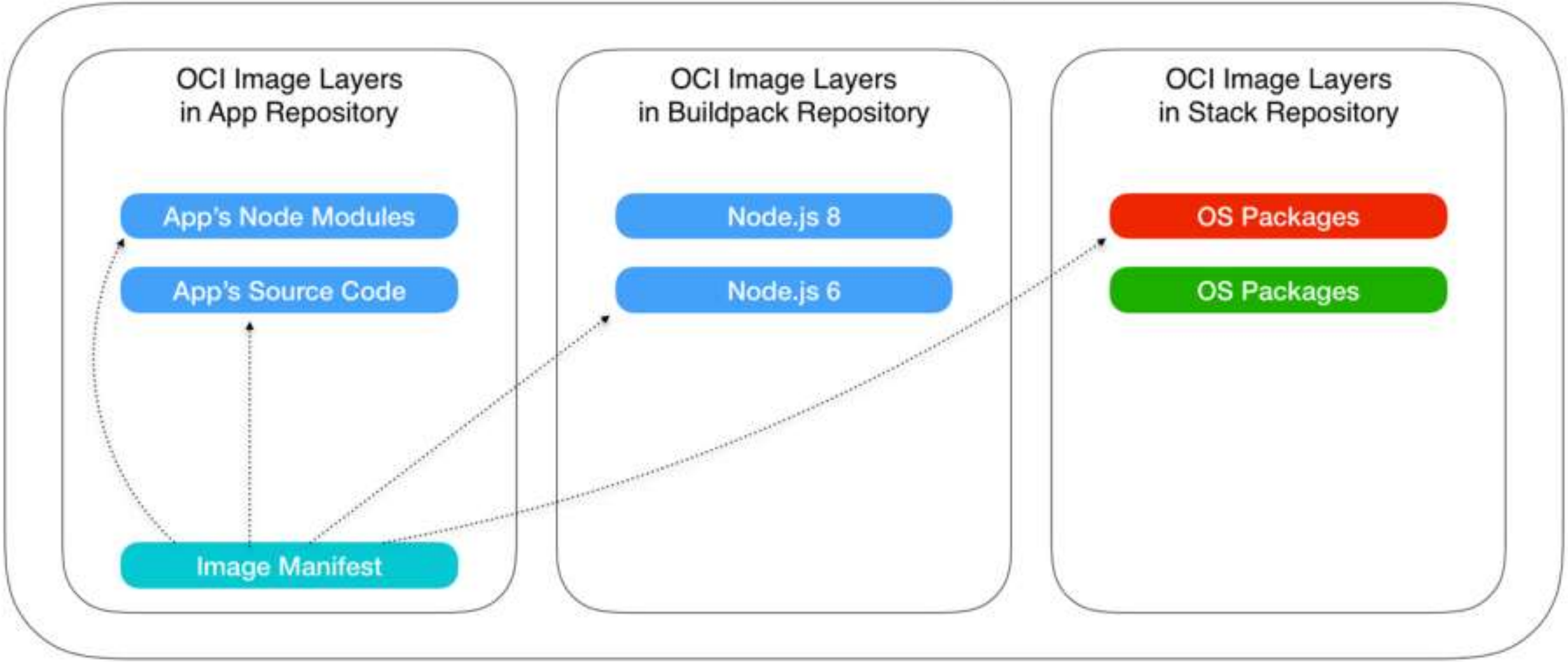

OCI Image Layers in App Repository
App's Node Modules
App's Source Code
Image Manifest
OCI Image Layers in Buildpack Repository
Node.js 8
Node.js 6
OCI Image Layers in Stack Repository
OS Packages
OS Packages

Image Registry (with OCI image and cross-repo blob mount support)

Node.js App

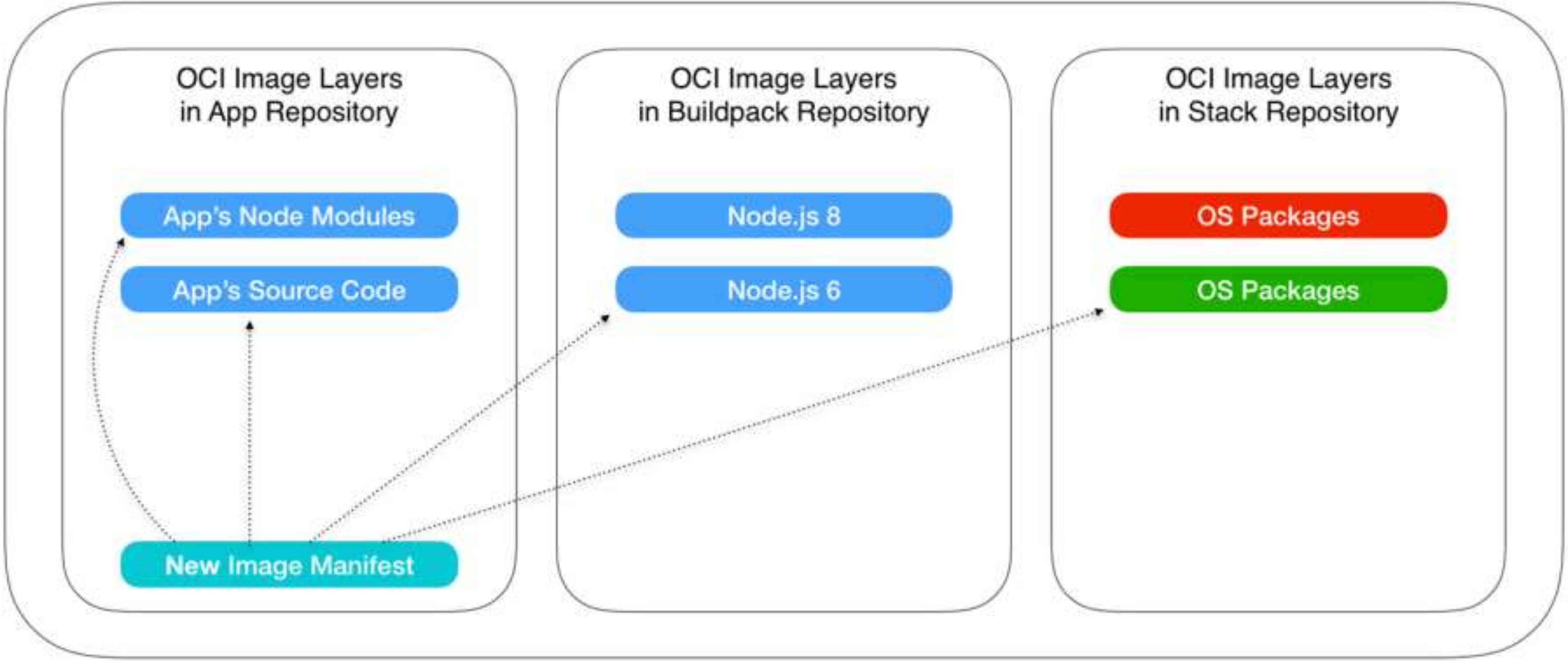
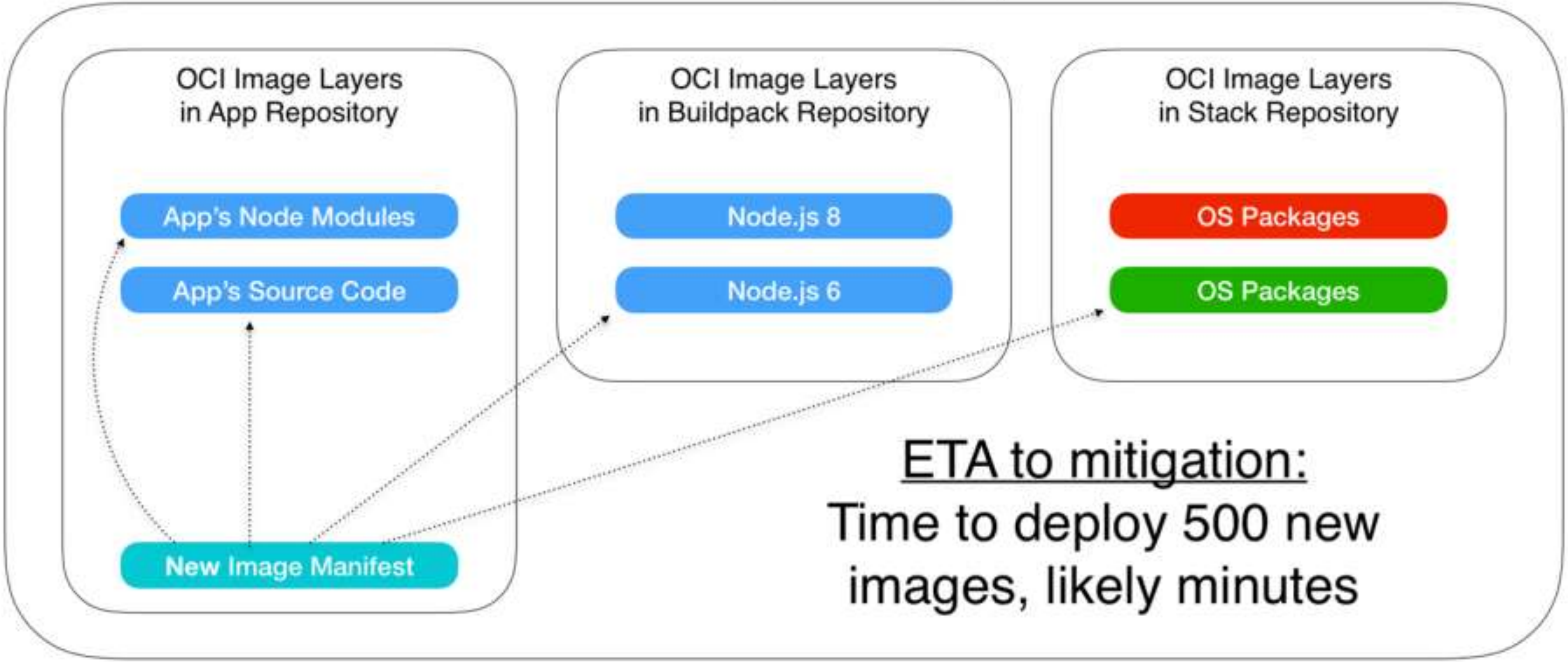


Image Registry (with OCI image and cross-repo blob mount support)

Node.js App


OCI Image Layers in App Repository
App's Node Modules
App's Source Code
New Image Manifest
OCI Image Layers in Buildpack Repository
Node.js 8
Node.js 6
OCI Image Layers in Stack Repository
OS Packages
OS Packages

ETA to mitigation:

Time to deploy 500 new images, likely minutes

Image Registry (with OCI image and cross-repo blob mount support)

Cloud Native Buildpacks

Pivotal Build Service

Pivotal Build Service: CNB + Enterprise Features

Image Promotion—No Rebuild Required

- A more intelligent approach to image updating.
- Developers can to promote images through environments, and eventually, across PCF foundations.

Automated Image Updates

- Declarative configuration model
- New images are delivered to your registry whenever this configuration falls out of sync.
- Consistent and up to date container images.

Operator Control

- Restricting buildpack usage in the apps they supervise.
- Create build configurations for different groups of developers within the org.
- These configs would govern the buildpacks that any given dev is allowed to use.

Try out pack and Buildpacks

- Second Public Beta – v0.2.1
  - pack CLI
  - Cloud Foundry and Heroku Buildpacks
  - Ubuntu-based Operating System Image

- Join us on Slack:
  - **slack.buildpacks.io**
- Join our Mailing List:
  - **lists.cncf.io/g/cncf-buildpacks**

Buildpacks.io

Pivotal®

Transforming How The World Builds Software