



DOCKER CONTAINER LIFECYCLES - PROBLEM OR OPPORTUNITY?



About your speaker

github.com/jbaruch

@jbaruch

linkd.in/jbaruch



Baruch Sadogursky

J*, G* and Public Speaking Geek with JFrog FTW.

Israel | Computer Software

Current

Developer Advocate at JFrog Ltd



stackoverflow.com/users/402053/jbaruch

Developer who?

Job Description:

1. Hang out
with DevOps

2. Talk about it

Developer who?



What Frog?



What Frog?



What Frog?



What Frog?





JFrog Artifactory



JFrog Artifactory



JFrog Bintray



JFrog Artifactory



JFrog Bintray



JFrog Mission Control

Poll Time!



Poll Time!

- ✓ **Heard about Docker**
- ✓ **Can do the tutorial**
- ✓ **PoCing, playing etc.**
- ✓ **Production, baby!**

A close-up photograph of a grey cat's face. The cat has large, expressive orange eyes with dark pupils. Its fur is a mottled grey. The background is dark and out of focus.

Y U NO

DOCKER PRODUCTION?

I LIKE YOU

BUT I DON'T TRUST YOU

CONTINUOUS INTEGRITY.

JFrog Artifactory + Docker



Why we are good with docker

Why we are good with docker

- we have decade of experience with managing binaries

Why we are good with docker

- we have decade of experience with managing binaries
- We did image registry before there was docker registry

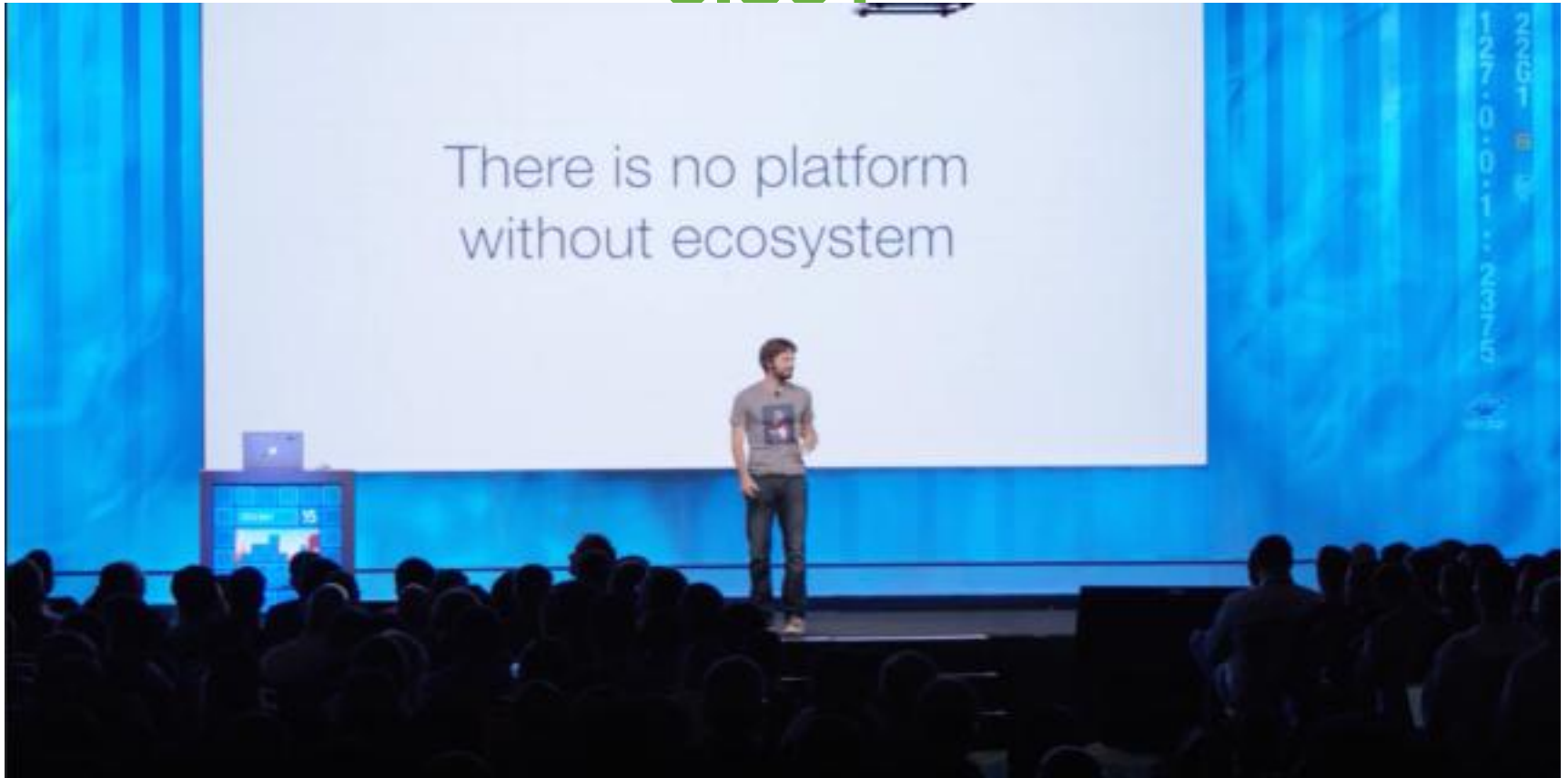
Why we are good with docker

- we have decade of experience with managing binaries
- We did image registry before there was docker registry
- we did private image hosting before docker hub

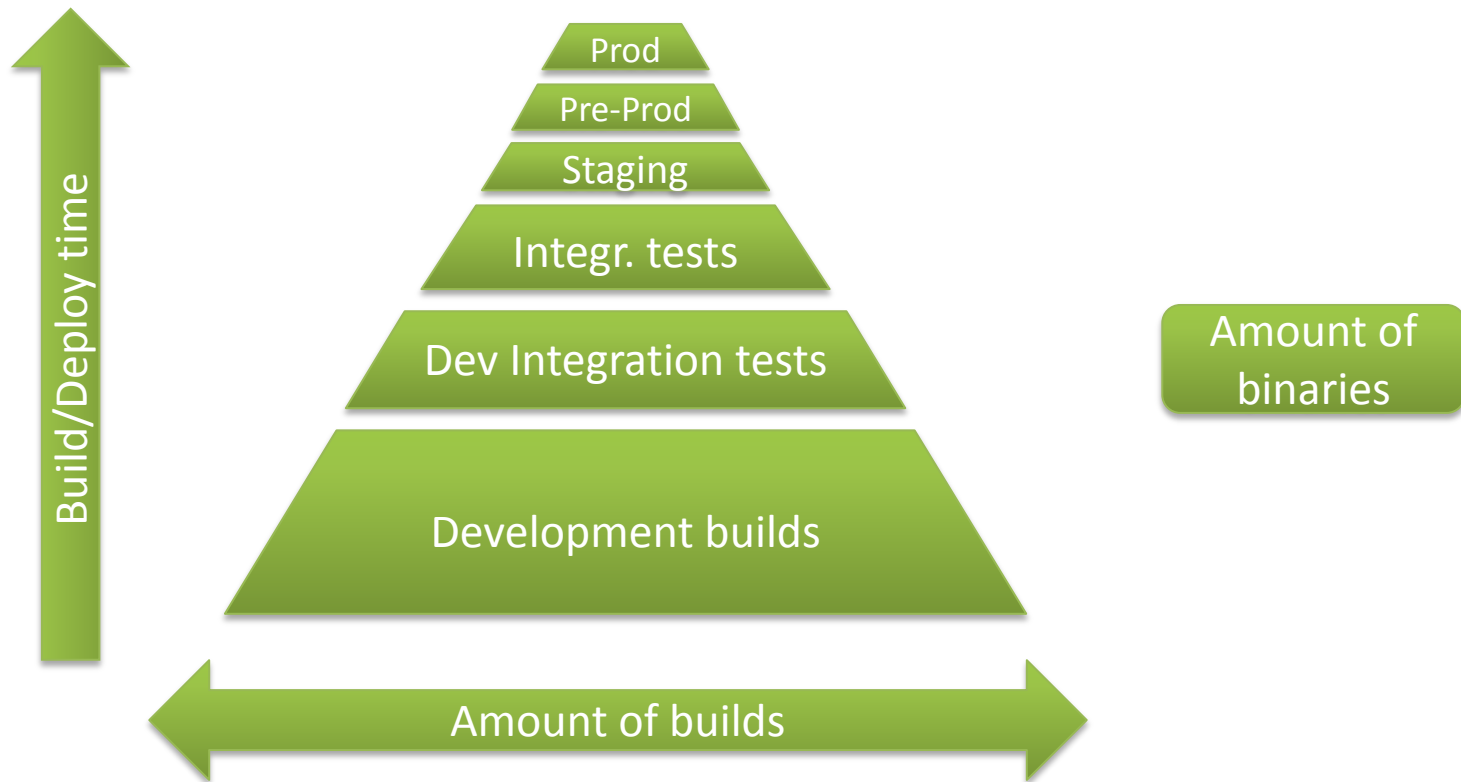
Why we are good with docker

- we have decade of experience with managing binaries
- We did image registry before there was docker registry
- we did private image hosting before docker hub
- we do promotion pipelines for docker

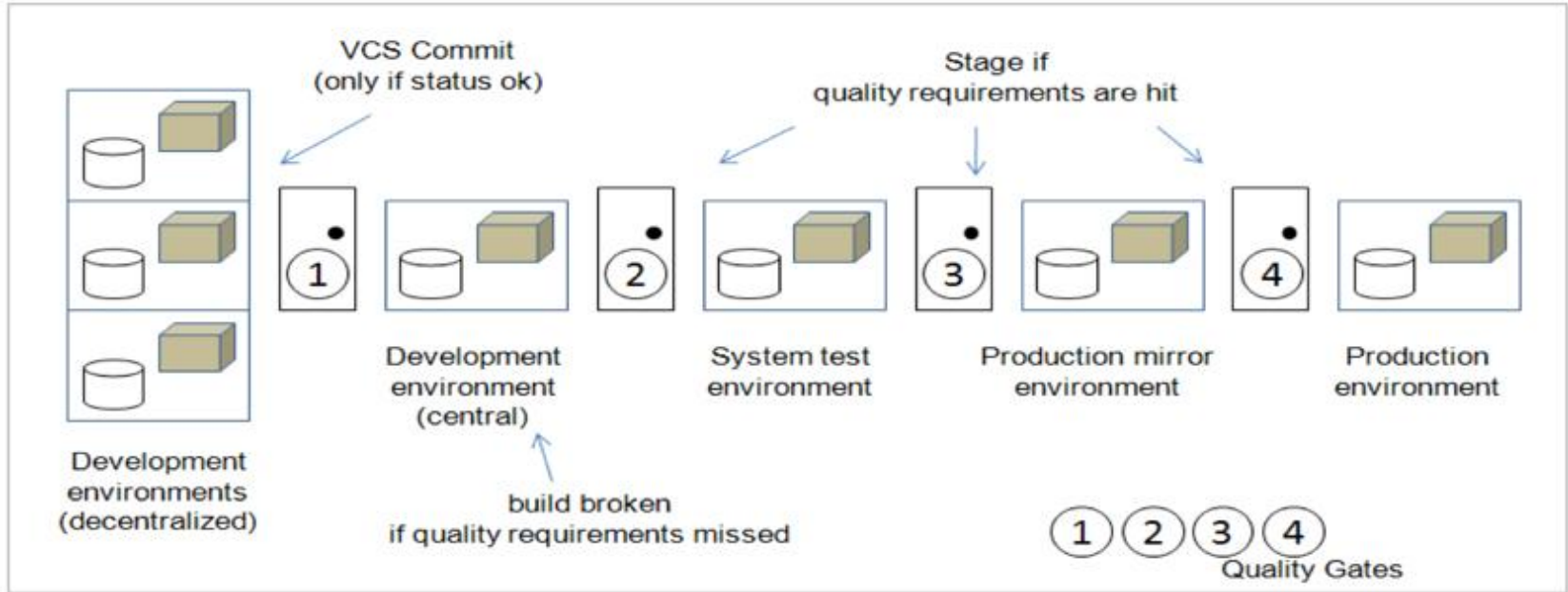
Who's using Docker and nothing else?



The Promotion Pyramid



Pipeline: Quality Gates and Visibility



[Source: Agile ALM, Michael Hüttermann, Manning Publications Co.](#)

CI PIPELINES?

DOING THEM FOR YEARS.





1

Declare new dependencies



Build Tools / Dependency Managers



Dev. Team

1

Declare new dependencies



Build Tools / Dependency Managers

2

Resolve dependencies





Dev. Team

1

Declare new dependencies



2

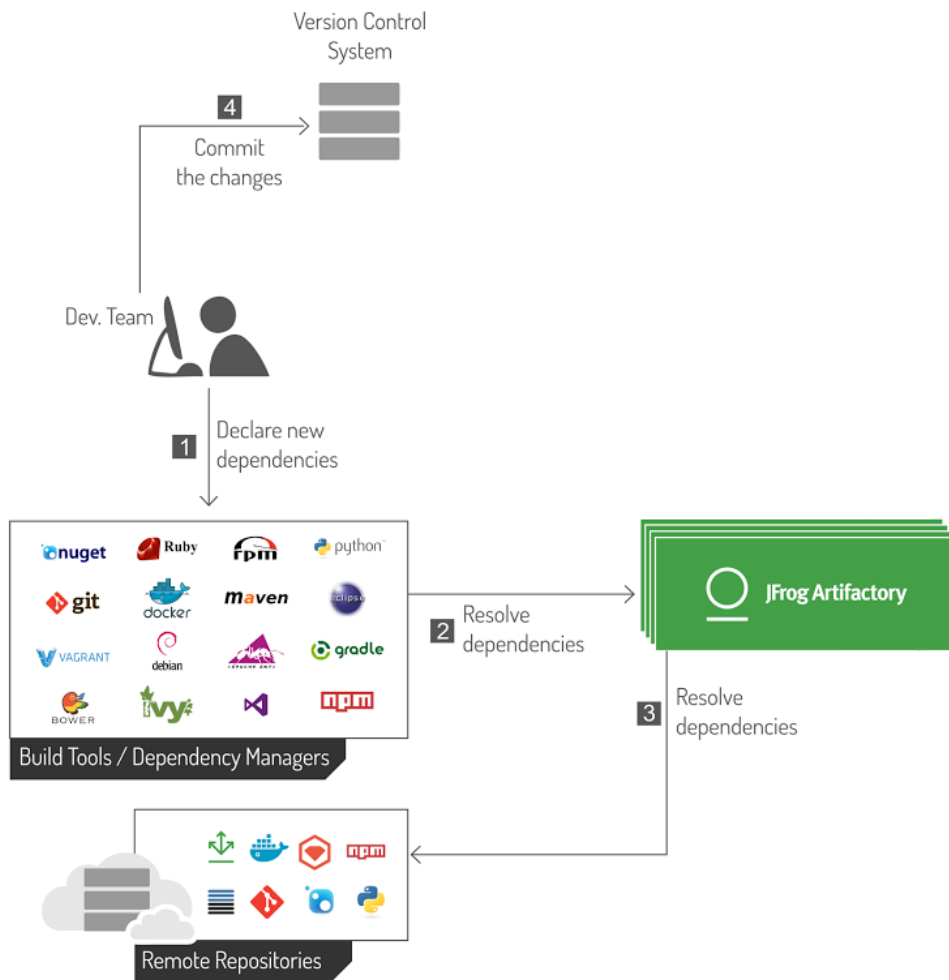
Resolve dependencies

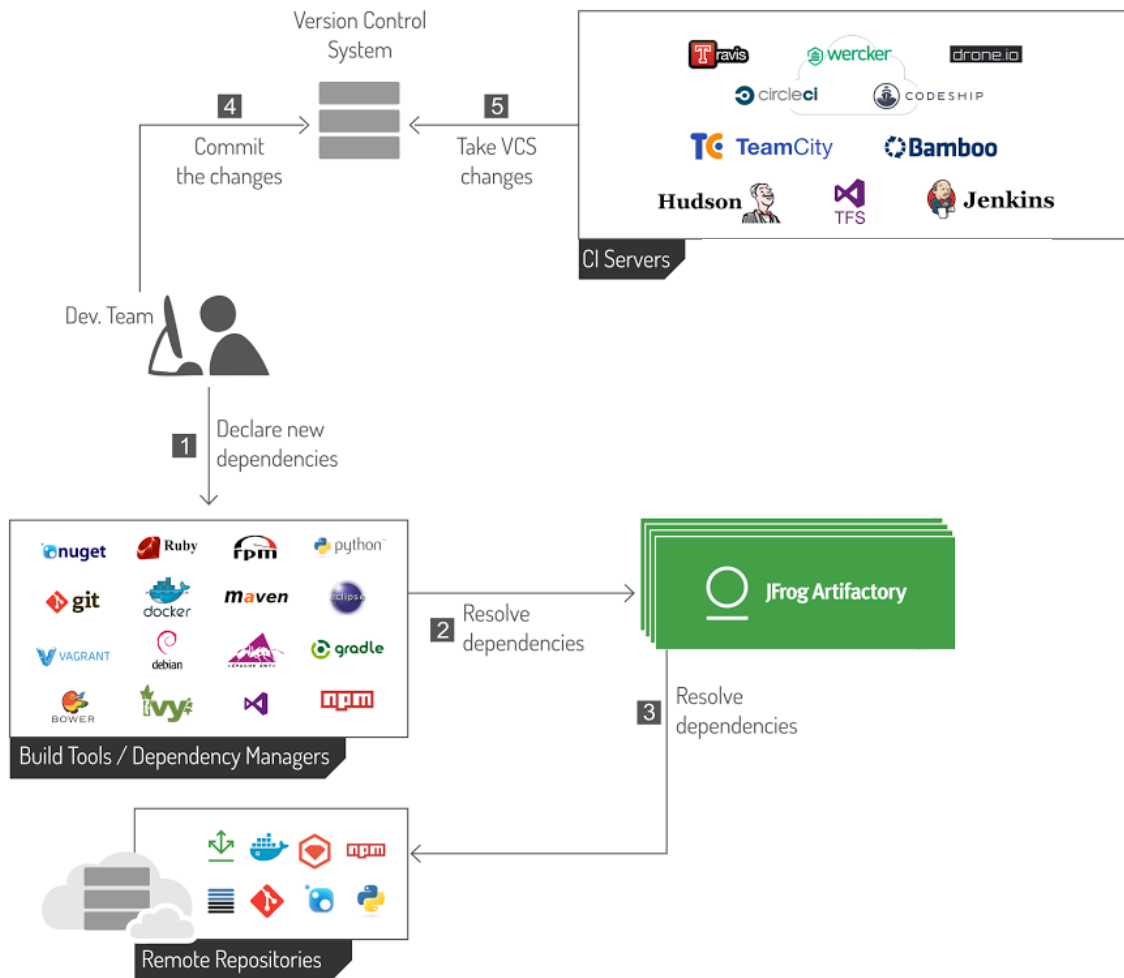


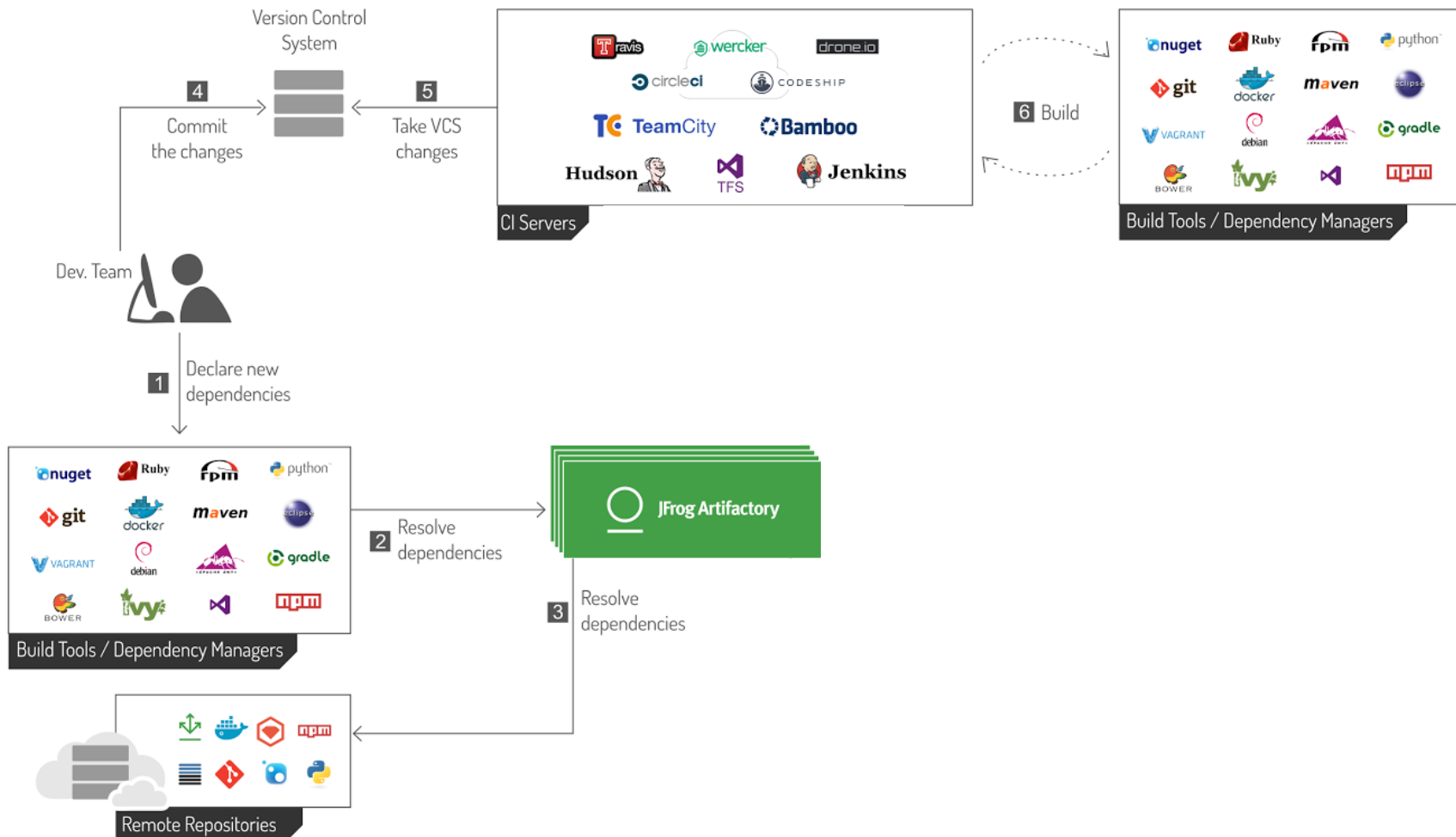
3

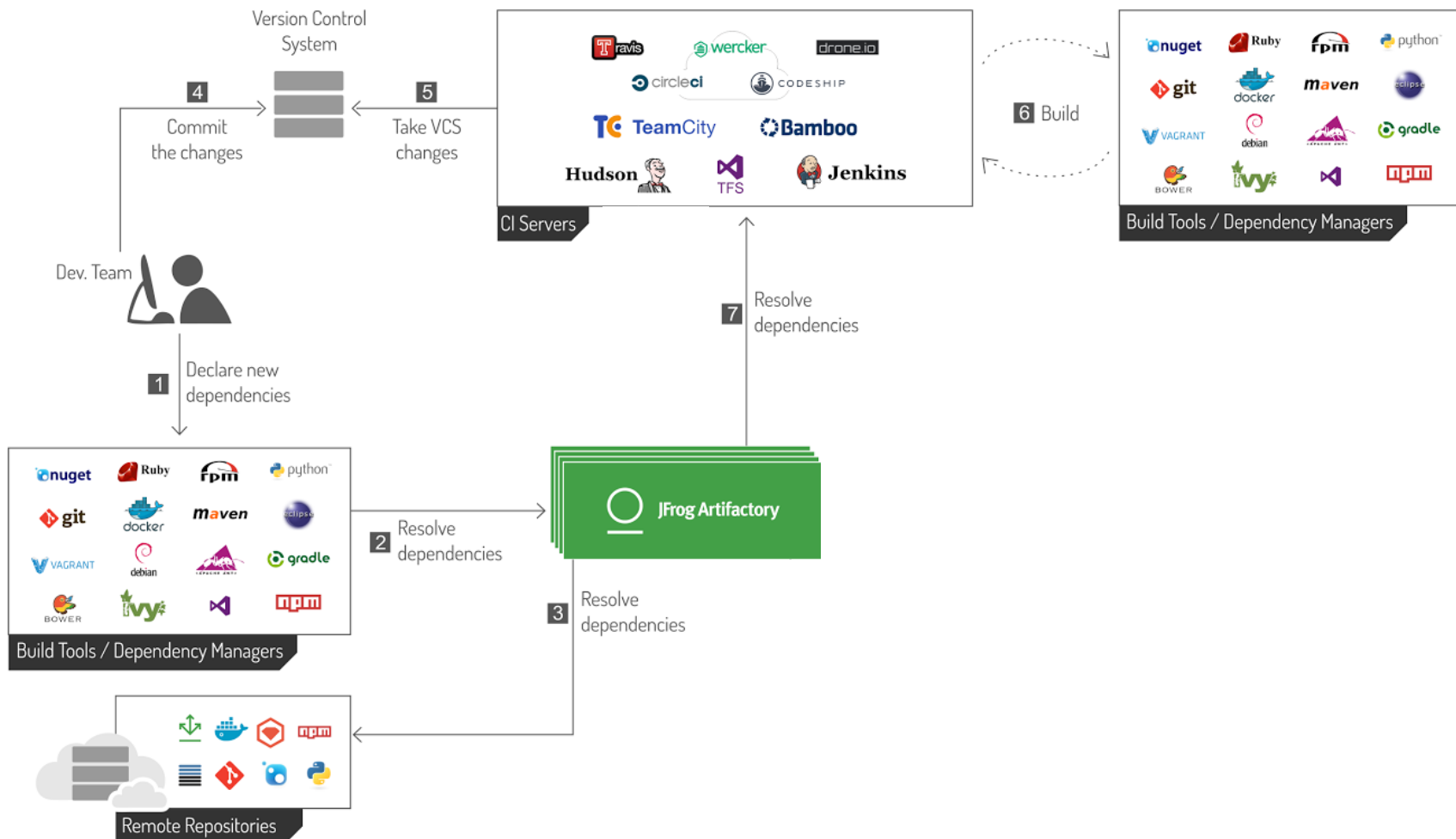
Resolve dependencies

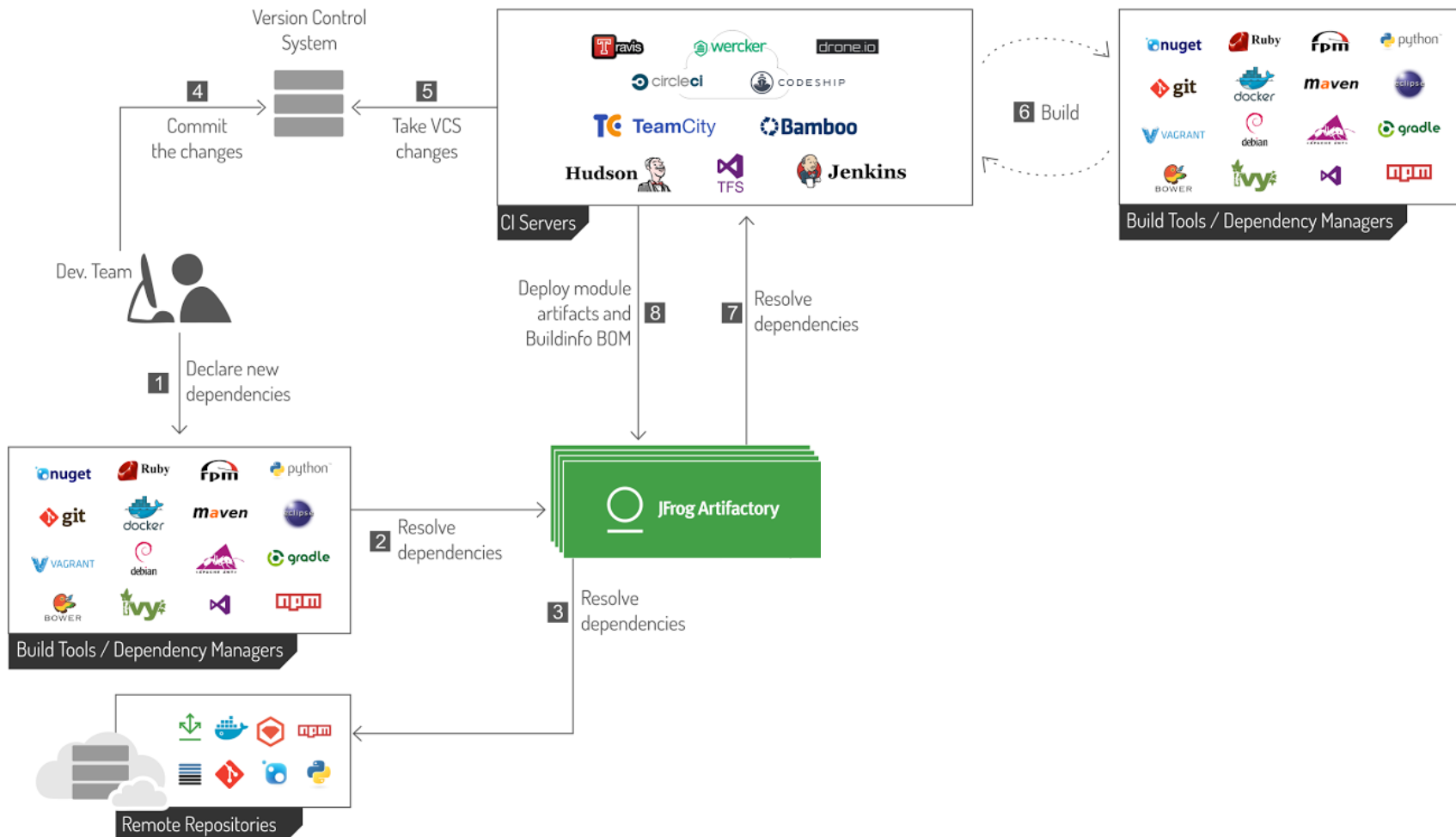


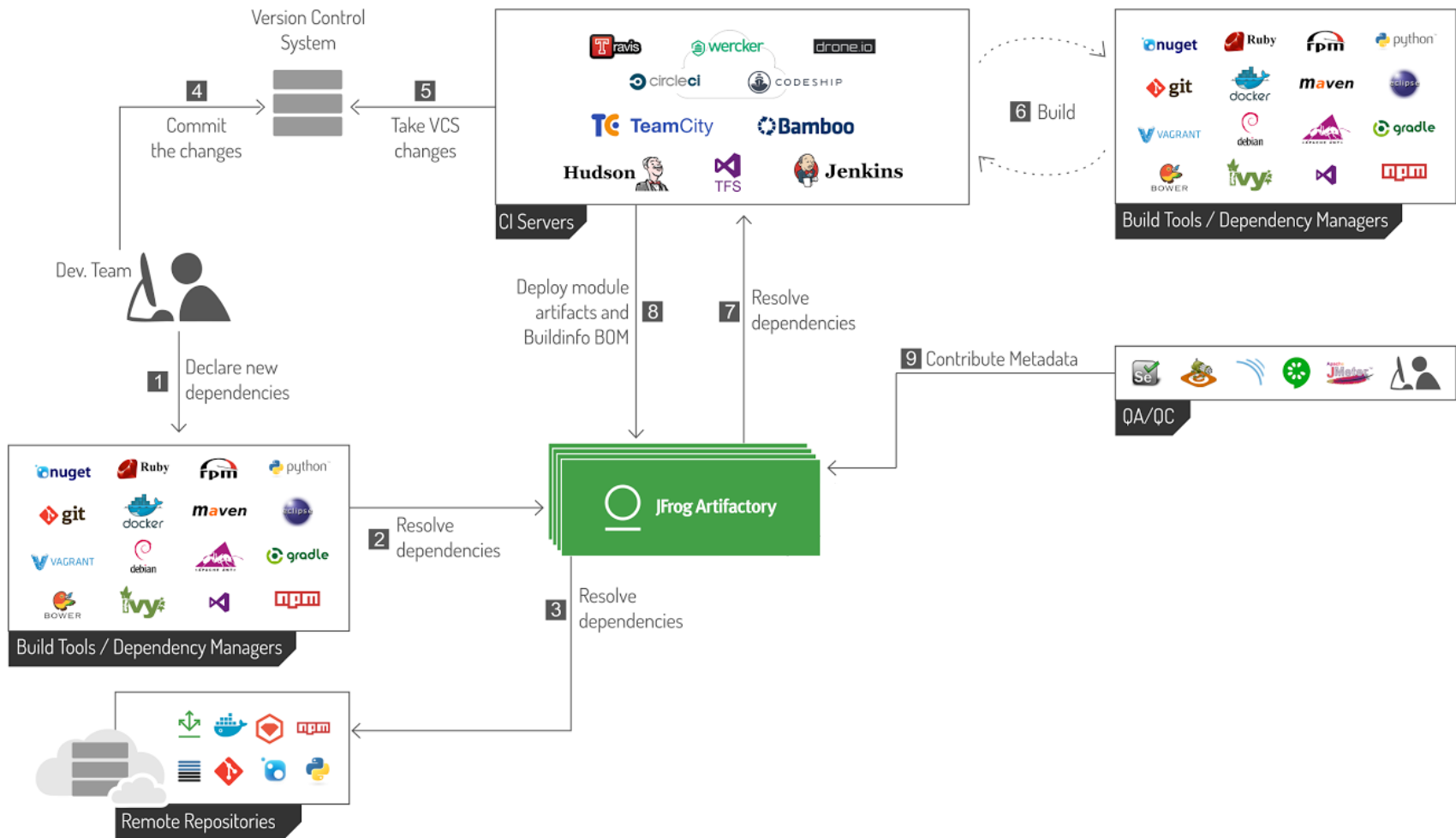


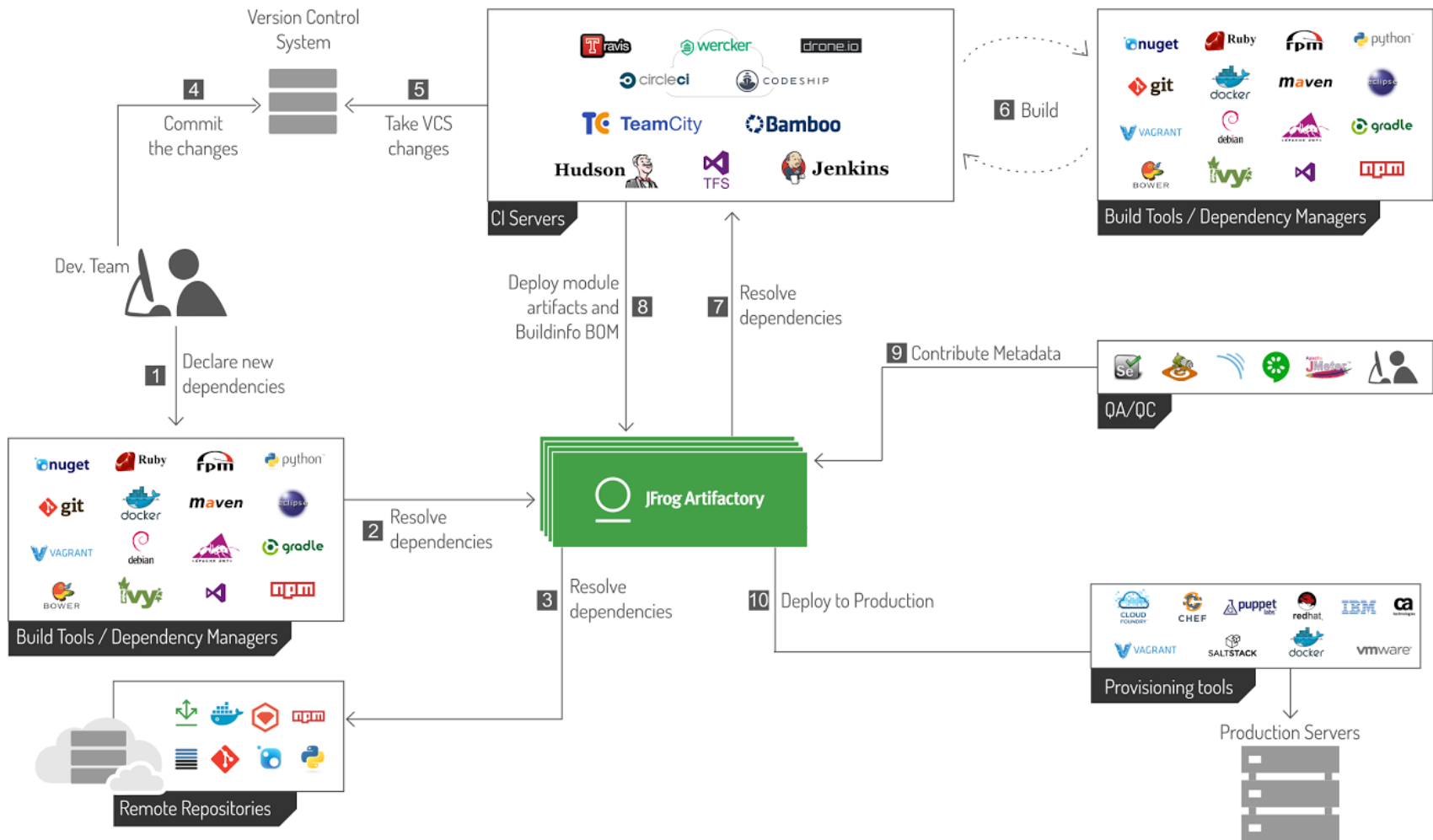


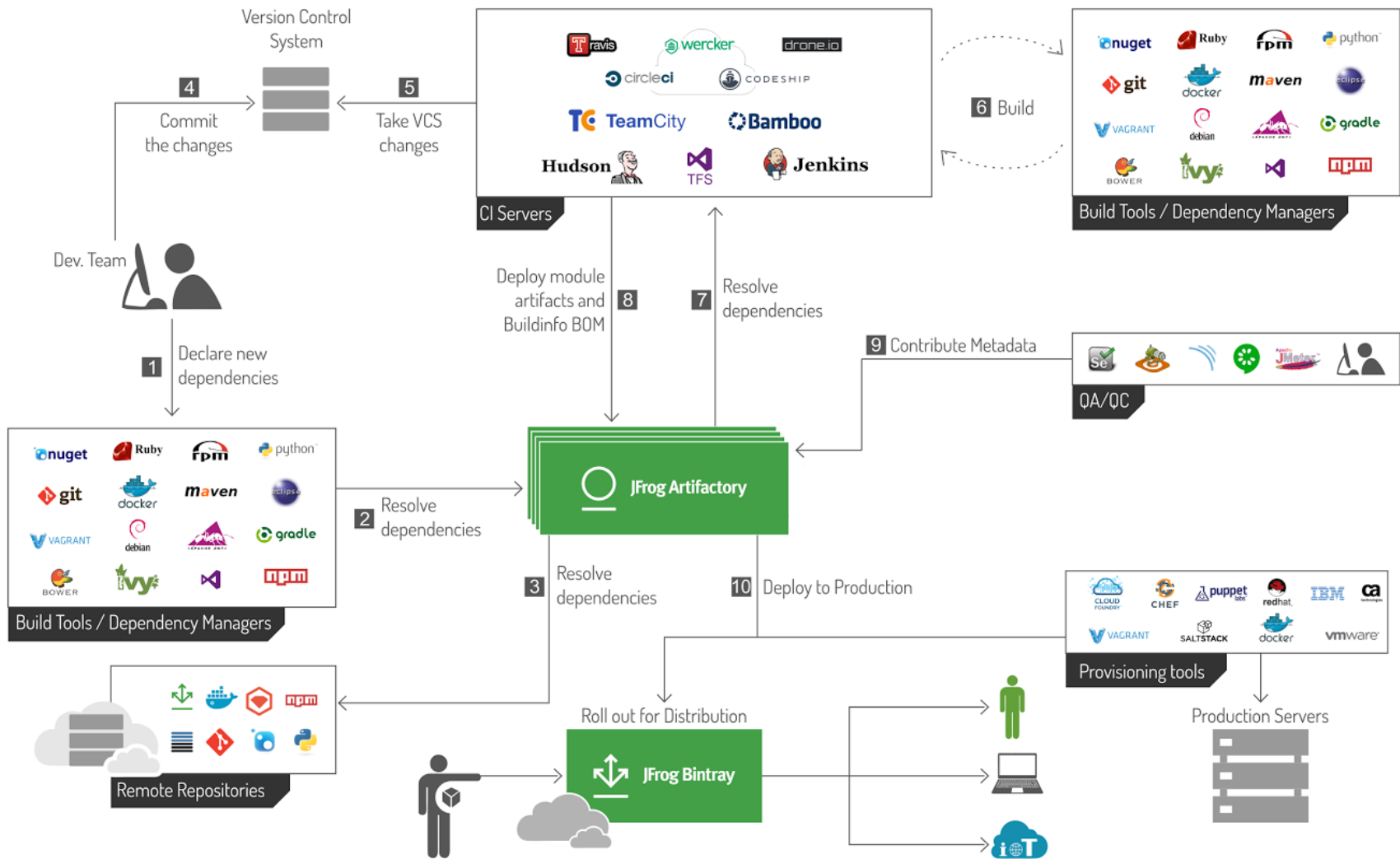








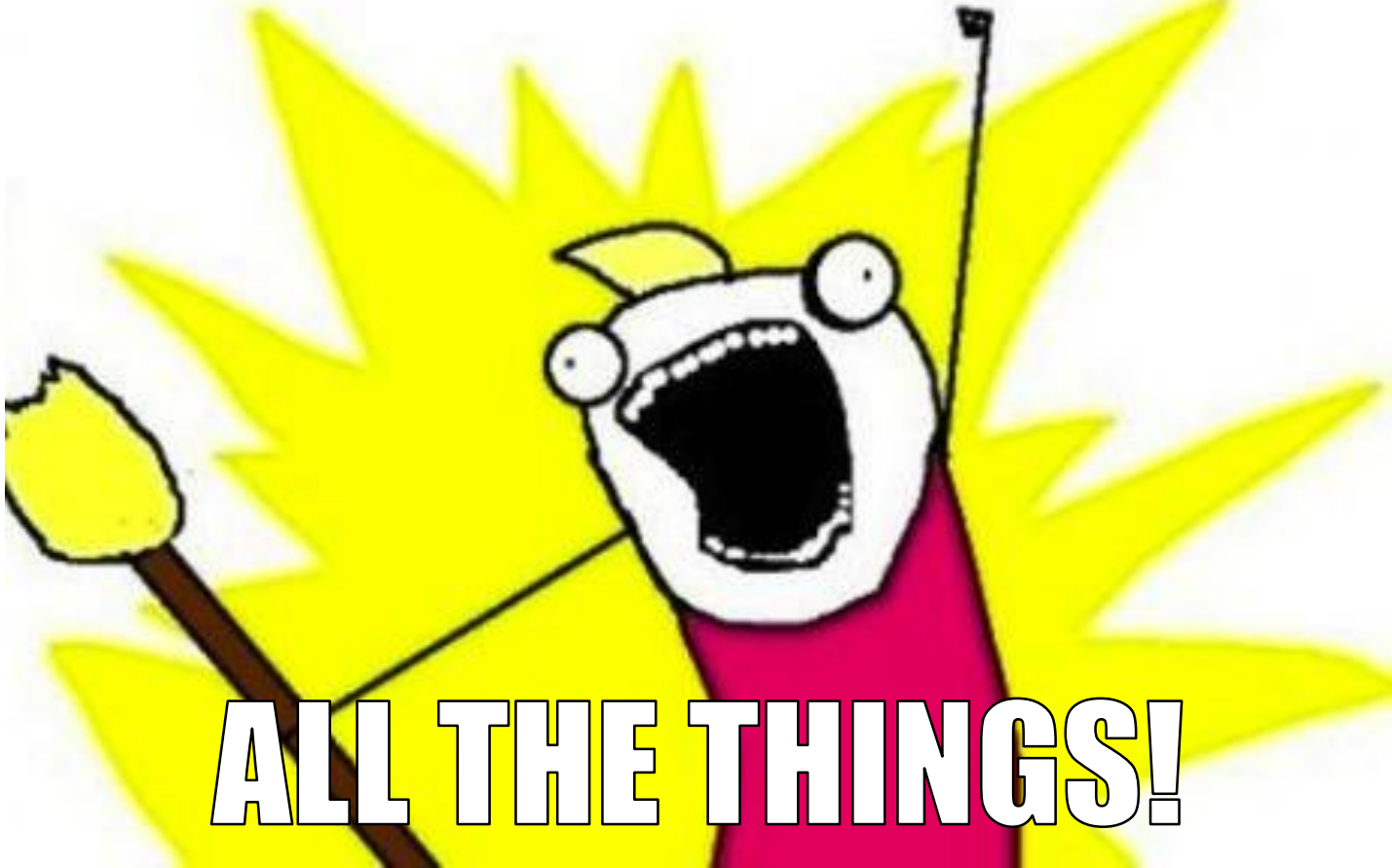




\$docker build

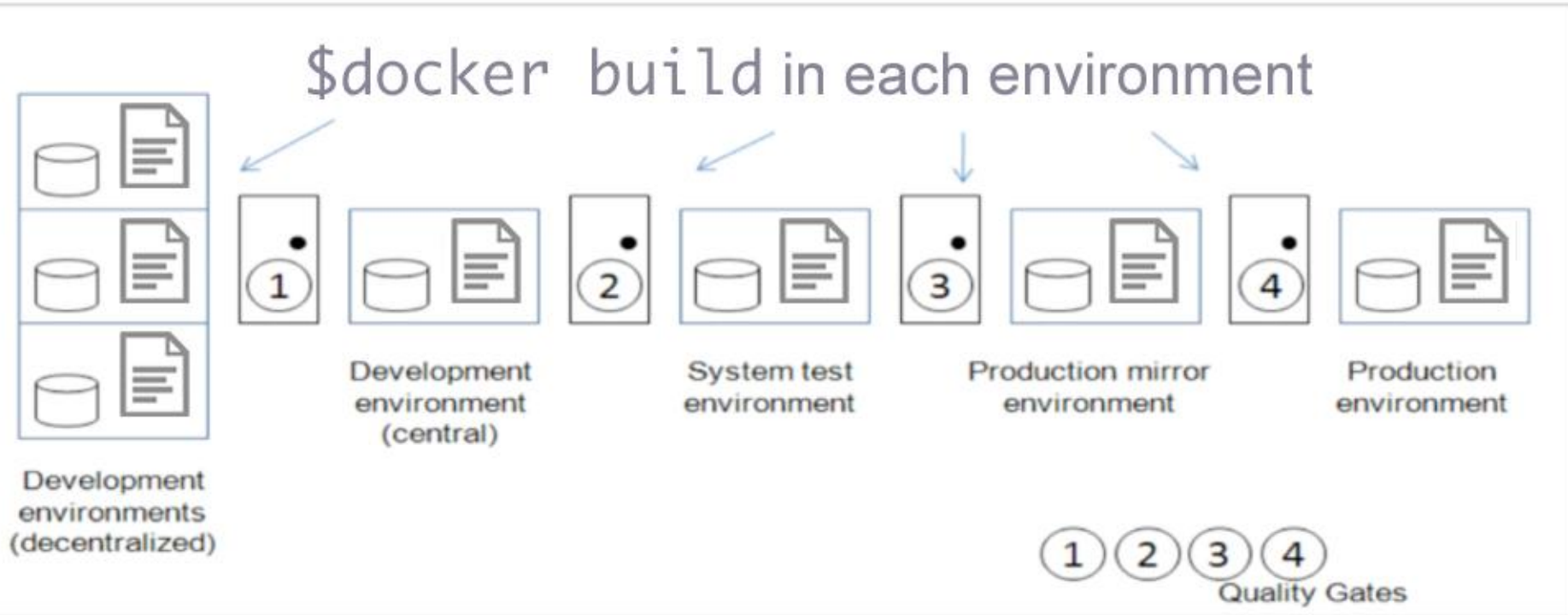


DOCKER BUILD



ALL THE THINGS!

Too easy!



FAST AND CHEAP BUILDS

NOT ALWAYS THE WAY TO GO



That's why.

```
1 FROM ubuntu ← Latest version
2
3 RUN apt-get install -y python-software-properties python → Latest version
4 RUN apt-get install -y nodejs
5 RUN mkdir /var/www ← Latest version
6
7 ADD app.js /var/www/app.js ← Latest version
8
9 CMD ["/usr/bin/node", "/var/www/app.js"]
```

Let's fix it!

```
1  FROM ubuntu:14.04
2
3  RUN apt-get install -y python-software-properties python
4  RUN apt-get install -y nodejs
5  RUN mkdir /var/www
6
7  ADD app.js /var/www/app.js
8
9  CMD ["/usr/bin/node", "/var/www/app.js"]
```



Better now?

Let's fix it (again)!

```
1 FROM ubuntu:0844055d30c0cad5ac58097597a94640b0102f47d6fa972c94b7c129d87a44b7
2
3 RUN apt-get install -y python-software-properties python
4 RUN apt-get install -y nodejs
5 RUN mkdir /var/www
6
7 ADD app.js /var/www/app.js
8
9 CMD ["/usr/bin/node", "/var/www/app.js"]
```



And now?

How about those?

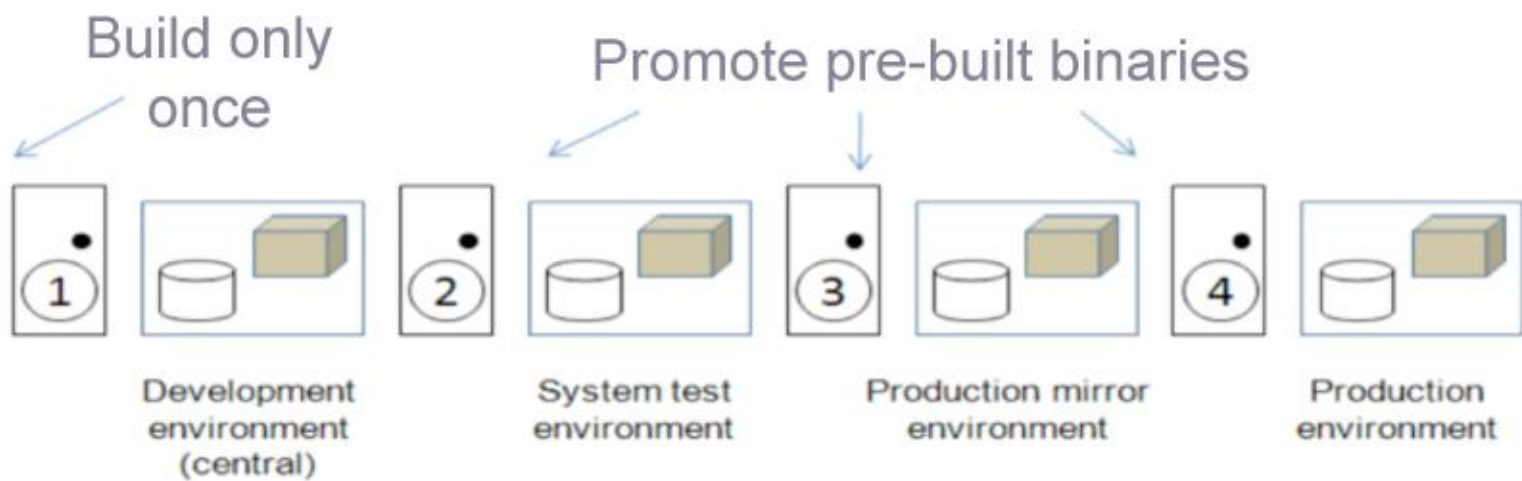
I DON'T ALWAYS BUILD PROMOTION PIPELINES



**BUT WHEN I DO, IT'S WITH
IMMUTABLE AND STABLE BINARIES**



Development environments (decentralized)



Trumped-up limitations



The Anatomy of Docker Tag

2r

Products

Customers

Community

Partners

tag

```
Usage: docker tag [OPTIONS] IMAGE[:TAG] [REGISTRYHOST/][USERNAME/]NAME[:TAG]
```

Tag an image into a repository

Wait a second, how can I
have more than one
repository per host now?!



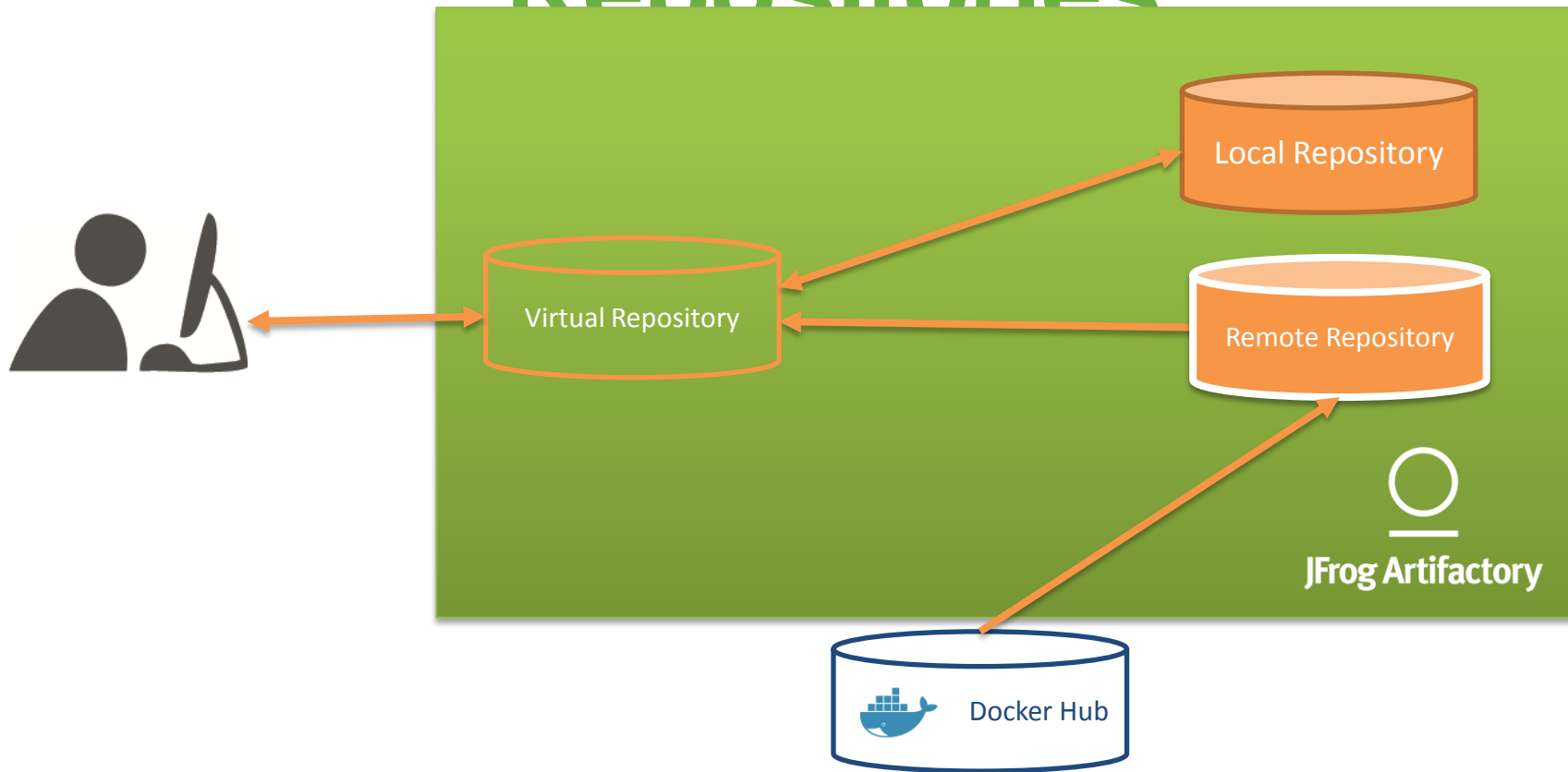
**“ONE REGISTRY PER HOST OUGHT
TO BE ENOUGH FOR ANYBODY.”**



Panic!



Artifactory Virtual Repositories



What we did?

What we did?

- Minimize number of repositories docker interact with

What we did?

- Minimize number of repositories docker interact with
- deploy to virtual (backed by dev repository)

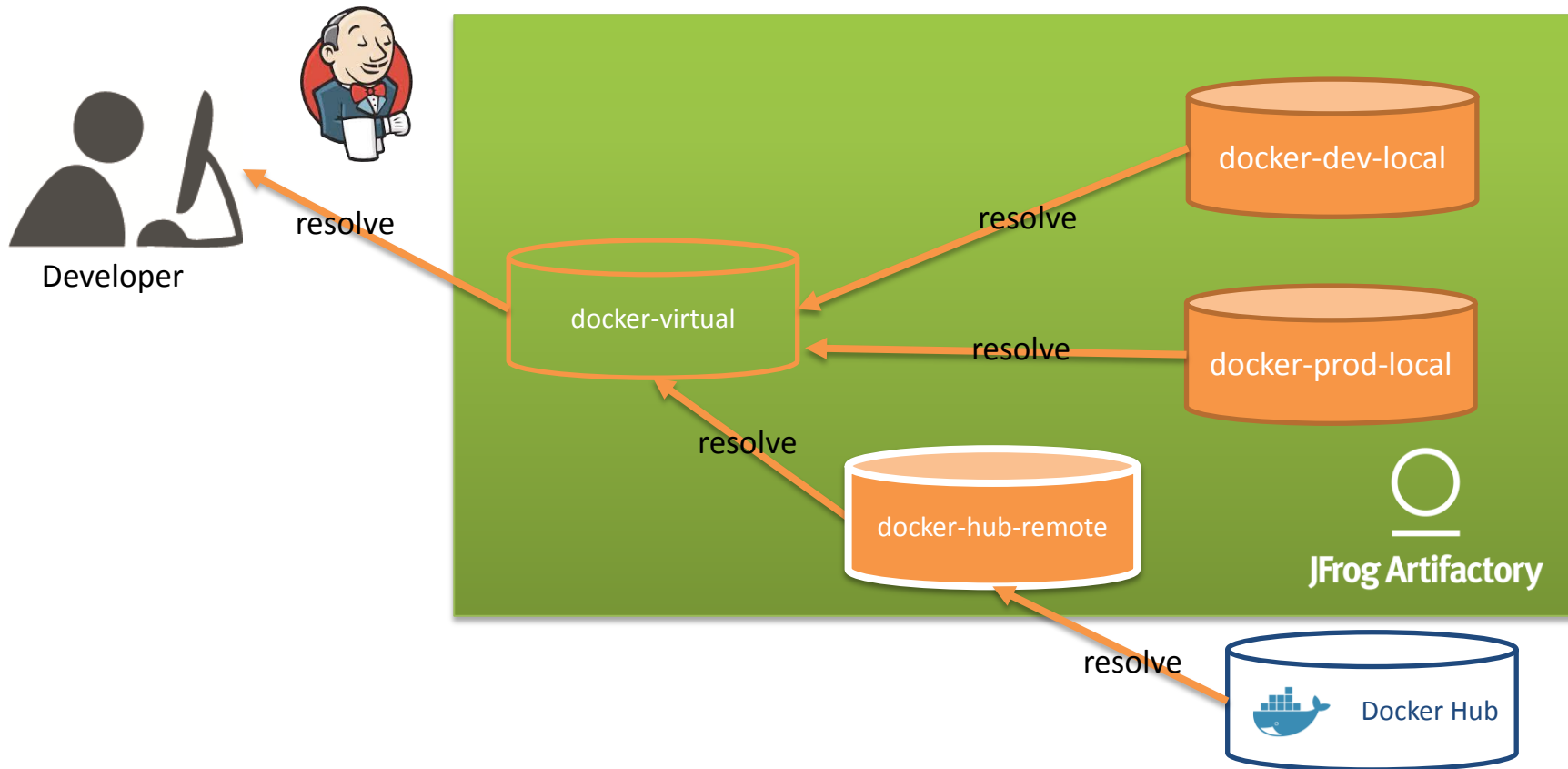
What we did?

- Minimize number of repositories docker interact with
- deploy to virtual (backed by dev repository)
- promote within artifactory

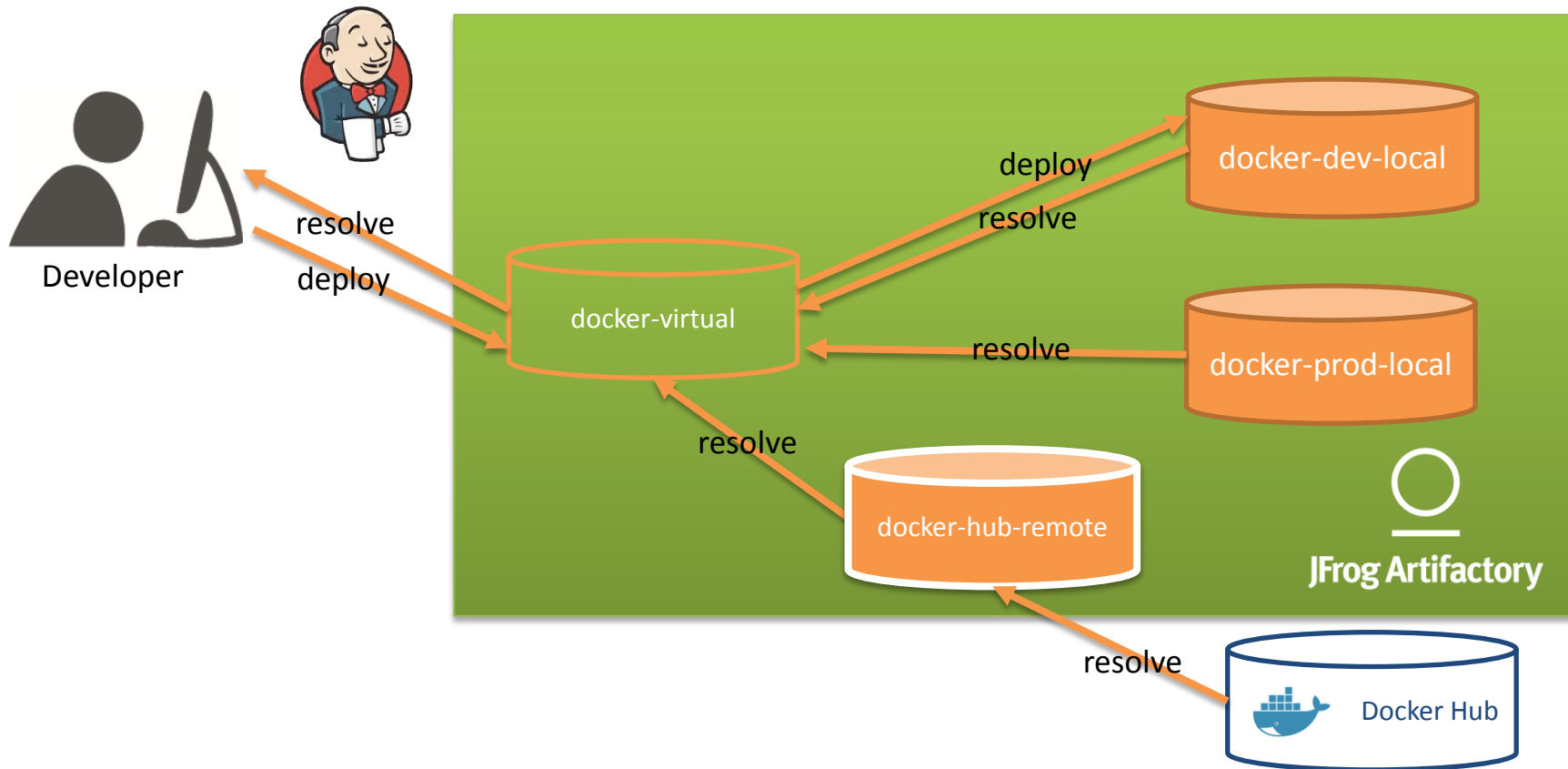
What we did?

- Minimize number of repositories docker interact with
 - deploy to virtual (backed by dev repository)
 - promote within artifactory
- Resolve from virtual (production-ready images)

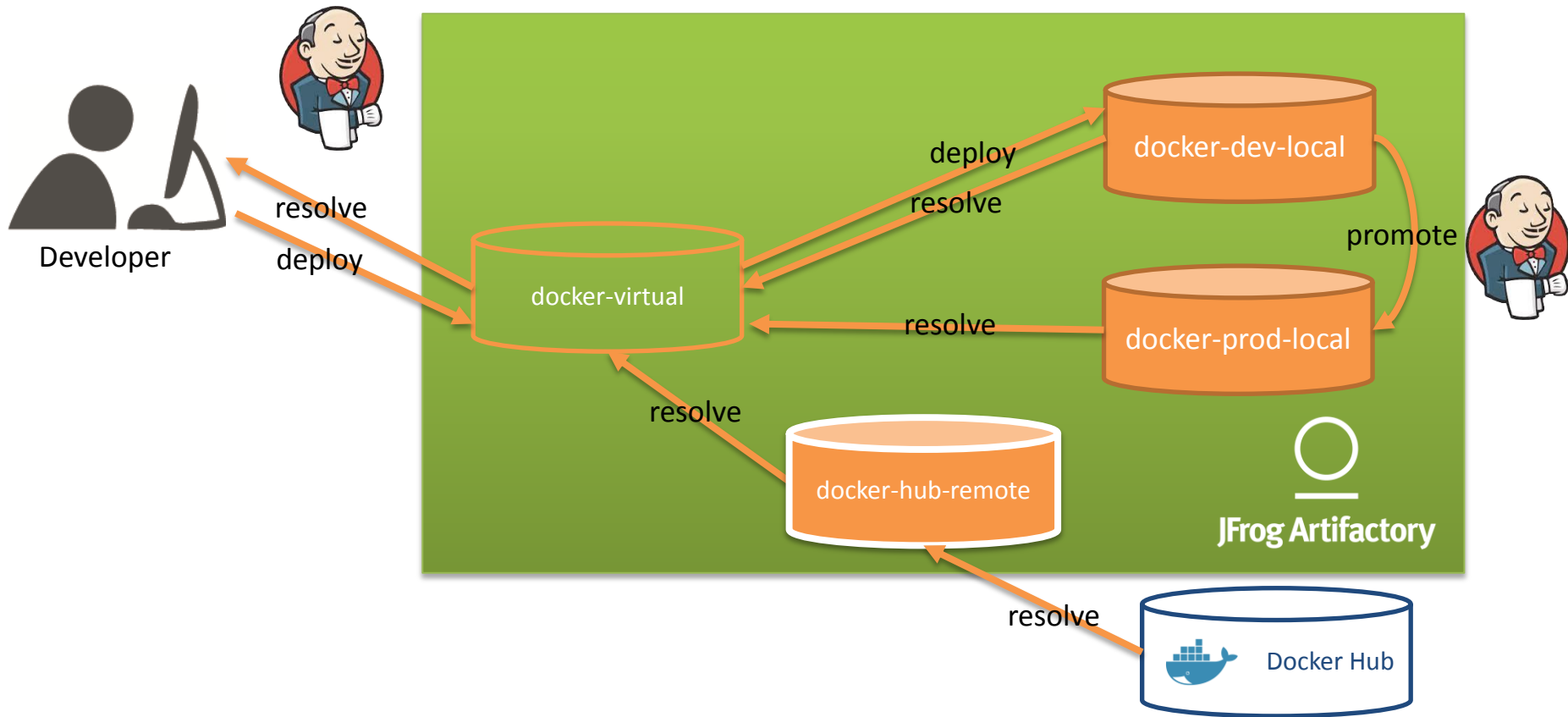
Virtual Repositories FTW



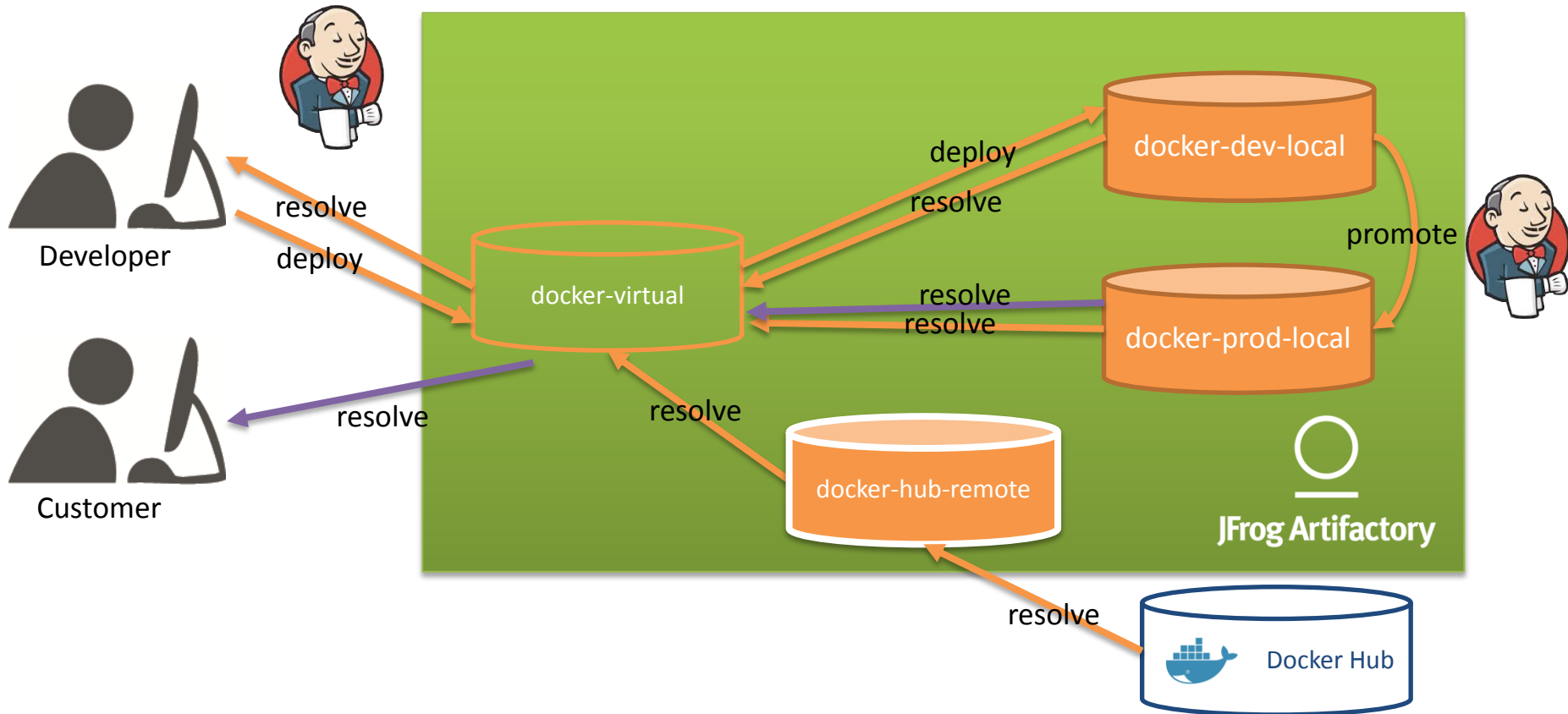
Virtual Repositories FTW



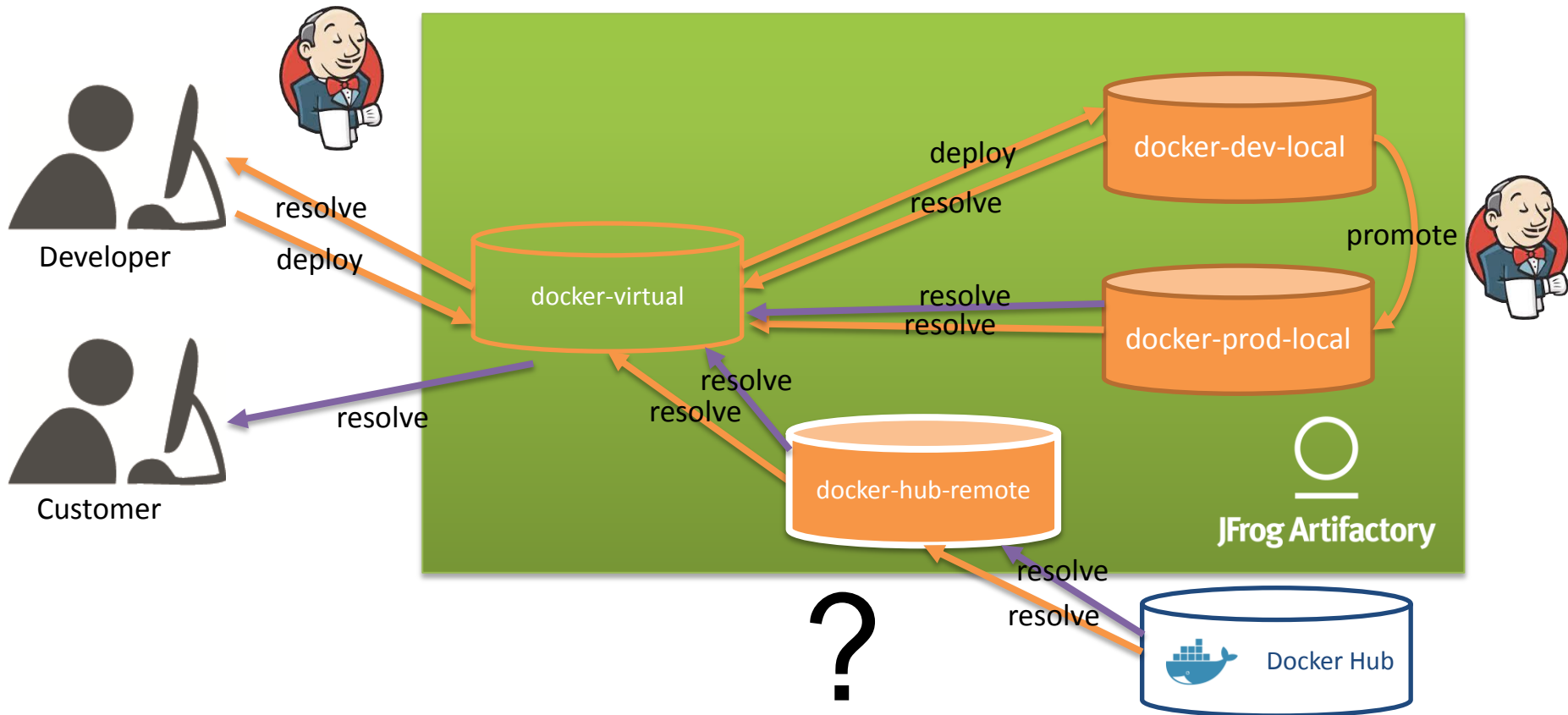
Virtual Repositories FTW



Virtual Repositories FTW



Virtual Repositories FTW





**Question: context name? repo
name?**

Virtual repository name

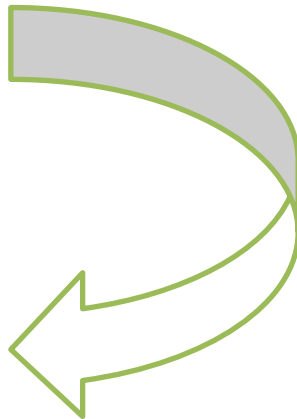
Tag name

Context name

`https://host:8081/artifactory/docker/busybox`

`docker tag host:port/busybox`

How to convert?



Hmm, but how the
--registry-mirror
fits in?..



Registry-mirror

Registry-mirror

- instructs docker to ignore the host in tag

Registry-mirror

- instructs docker to ignore the host in tag
- or lack of it

Registry-mirror

- instructs docker to ignore the host in tag
- or lack of it
- It's still a single URL

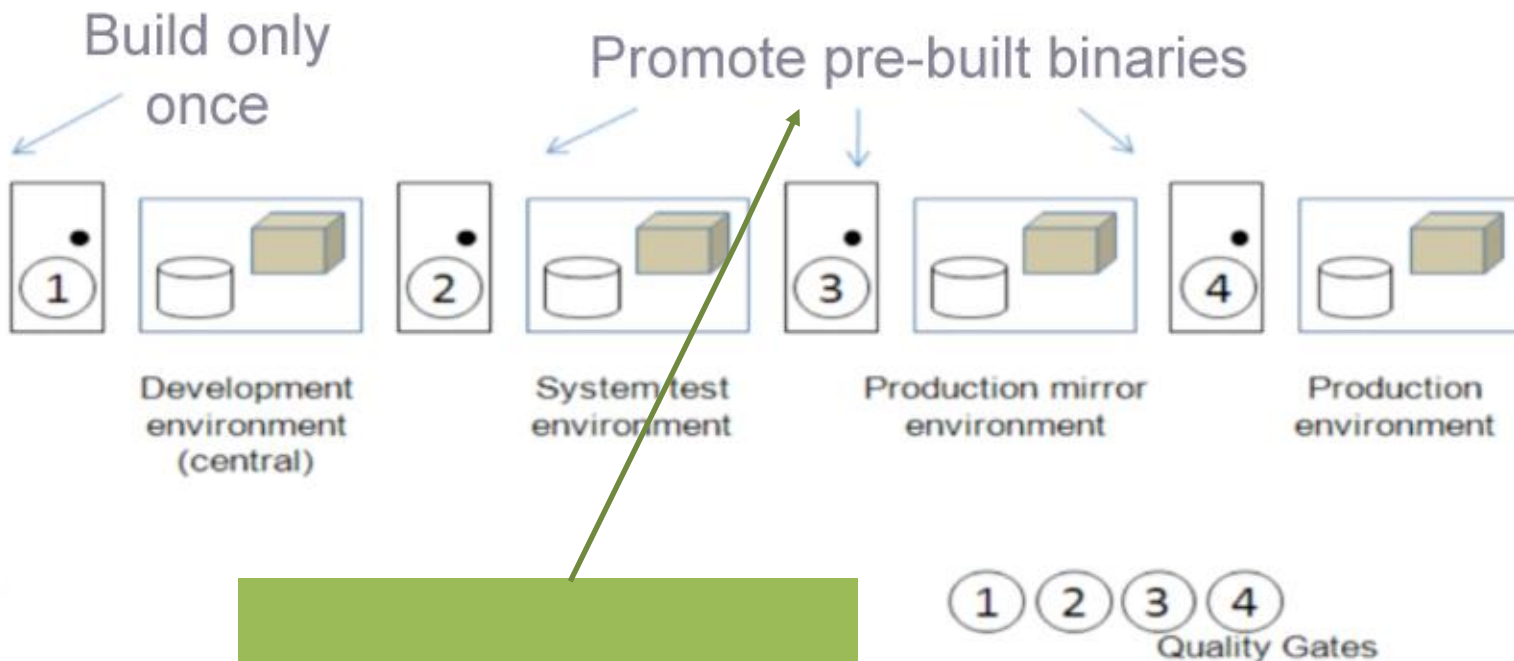
Registry-mirror

- instructs docker to ignore the host in tag
- or lack of it
- It's still a single URL
- and it is set per daemon (client)





Development environments (decentralized)



Why?

Finding the needle in haystack of binaries

Finding the needle in haystack of binaries

- What deployed where
- Tracing binaries to sources

Finding the needle in haystack of binaries

- What deployed where
- Tracing binaries to sources
- Cherry pick the one to prod

Adding Metadata



[Support](#) [Training](#) [Docs](#) [Blog](#) [Dc](#)

[Products](#) [Customers](#) [Community](#) [Partners](#) [C](#)

Install



Docker Fundamentals



Use Docker



[About Docker](#)

[Apply custom metadata](#)

[Docker Deprecated Features](#)

[Understand the architecture](#)

Provision & set up Docker hosts



Create multi-container applications



Apply custom metadata

You can apply metadata to your images, containers, or daemons via labels. Labels serve a wide range of uses, such as adding notes or licensing information to an image, or to identify a host.

A label is a `<key>` / `<value>` pair. Docker stores the label values as *strings*. You can specify multiple labels but each `<key>` must be unique or the value will be overwritten. If you specify the same `key` several times but with different values, newer labels overwrite previous labels. Docker uses the last `key=value` you supply.

Note: Support for daemon-labels was added in Docker 1.4.1. Labels on containers and images are new in Docker 1.6.0

Label keys (namespaces)



Consuming Metadata



Artifactory / Welcome to Artifactory / Artifactory Pro

Artifactory Query Language

📄 Created by Rami Honig, last modified by Mark Galpin on Jan 15, 2016

Overview

Artifactory Query Language (AQL) is specially designed to let you uncover any data related to the artifacts and builds stored within Artifactory. Its syntax offers a simple way to formulate complex queries that specify any number of search criteria, filters, sorting options, and output parameters. AQL is exposed as a RESTful API which uses data streaming to provide output data resulting in extremely fast response times and low memory consumption. Currently, AQL can only extract data that resides in your instance of Artifactory, so it runs on **local repositories** and **remote repository caches**.

Here are a few simple examples:

```
// Return all artifacts of the "artifactory" build.
items.find({"@build.name":{"$eq":"artifactory"}})

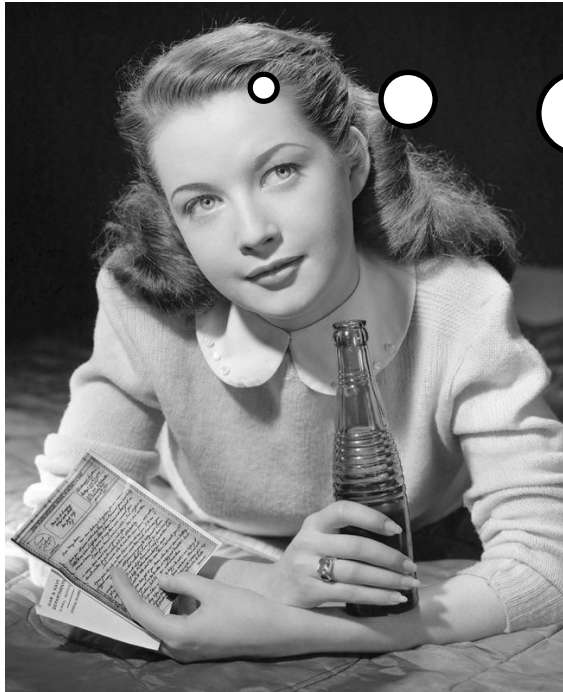
// Return all builds that have a dependency with a license that is not Apache.
builds.find({"module.dependency.item@license":{"$nmatch":"Apache-*"}})

// Return all archives containing a file called "org/artifactory/Main.class".
items.find({"archive.entry.name":{"$eq":"Main.class"} , "archive.entry.path":{"$eq":"org/artifactory"}})
```

YMMV



Hook your own logic in promotion



I wish I could
script it in
Groovy...

**CAN YOUR DOCKER
REGISTRY DO THIS?**





```
promotions {  
  myCustomPromotion(version: '1.0', description: 'desc', params: [customParamName: 'customValue'])  
    { buildName, buildNumber, params ->  
      //script your own promotion logic here  
      message = 'promotionMessage'  
      status = 201  
    }  
}
```

Summary: Automate Everything!



One More Thing!

[HOME](#)[ABOUT](#)[SPEAKERS](#)[SPONSORS](#)[AGENDA](#)[REGISTER](#)[LOCATION](#)[CONTACT US](#)[JFrog Artifactory](#)[JFrog Mission Control](#)[JFrog Bintray](#)

MAY 23-24, 2016

JFROG USER CONFERENCE

The Napa Experience

[REGISTER](#)[PROPOSE A TALK](#)

LOCATION
Napa, California



DATE
MAY 23-24, 2016

That's all, Folks!

