

# The Secret Lives of Java Apps

Stories Told at Runtime



Chris Phillipson  
Copyright 2024



# Background

## Origin Story (circa 2018)

An aerospace company's platform team required some additional insight into applications that were running on Tanzu Application Service across several foundations. Commercial opt-in Telemetry did not exist at that time.



Chris Phillipson  
Copyright 2024

# Background

## What were they trying to achieve?

- Improving economics of operation at scale.
- Cultivating best practice operating principles.
- Better collaboration between the platform operations team and development and product teams.
- Manifesting a particular culture for adoption and consumption (e.g., “platform as a privilege” or “platform as a right”).

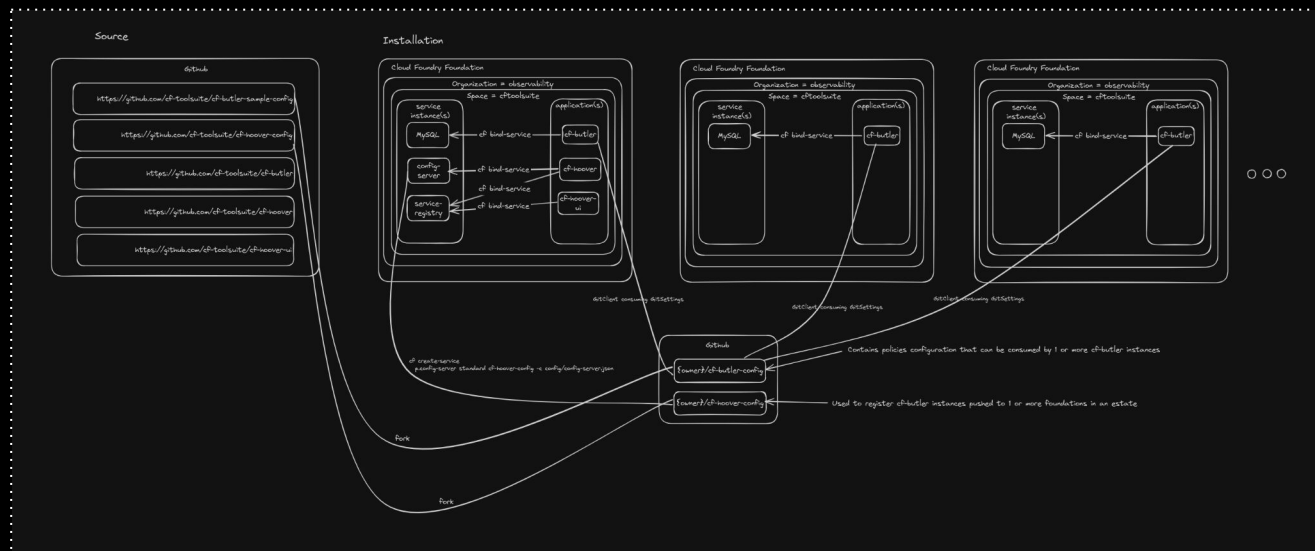


Chris Phillipson  
Copyright 2024

# Background

What evolved was a collection of purpose-built microservices

[cf-butler](https://github.com/cloudfoundry/cf-butler)  
[cf-hoover](https://github.com/cloudfoundry/cf-hoover)  
[cf-hoover-ui](https://github.com/cloudfoundry/cf-hoover-ui)  
[cf-archivist](https://github.com/cloudfoundry/cf-archivist)



Chris Phillipson  
Copyright 2024

# Background

## Frameworks

```
Project Reactor *  
Spring Boot 3.2 (Webflux, R2DBC)  
Cloud Foundry Java Client  
Java CFEnv  
Vaadin, Apex Charts
```

With much gratitude and respect to [Stephane Maldini](#)



Chris Phillipson  
Copyright 2024

# Background

## Outcomes

Platform teams and CISOs got runtime visibility into velocity of product teams and the operating characteristics of applications and service instances across entire estate.

Platform and product teams alike could define policies for governance and reporting purposes.



Chris Phillipson  
Copyright 2024

# Wait... hold up

What am I here to talk about?



## Units of deployment

- Applications
- (Managed) Services

## Where could they run?

- Cloud Foundry
- Kubernetes
- [ECS / Fargate](#)
- [Google Cloud Run](#)
- [Azure Container Apps](#)

## Awareness and intimacy

- I want to know what's been deployed and its current state
- How frequently does my team (or the org at-large) deploy updates?
- Who deployed an application or service and when did they do that?
- What does my application depend upon?
- What dependency versions are prevalent?
- What critical vulnerabilities might I be exposed to?
- Which applications are affected?
- How might I pull latest information from source and affected artifacts forward?
- How easily may I (or other interested parties) obtain answers to these questions **at runtime**?



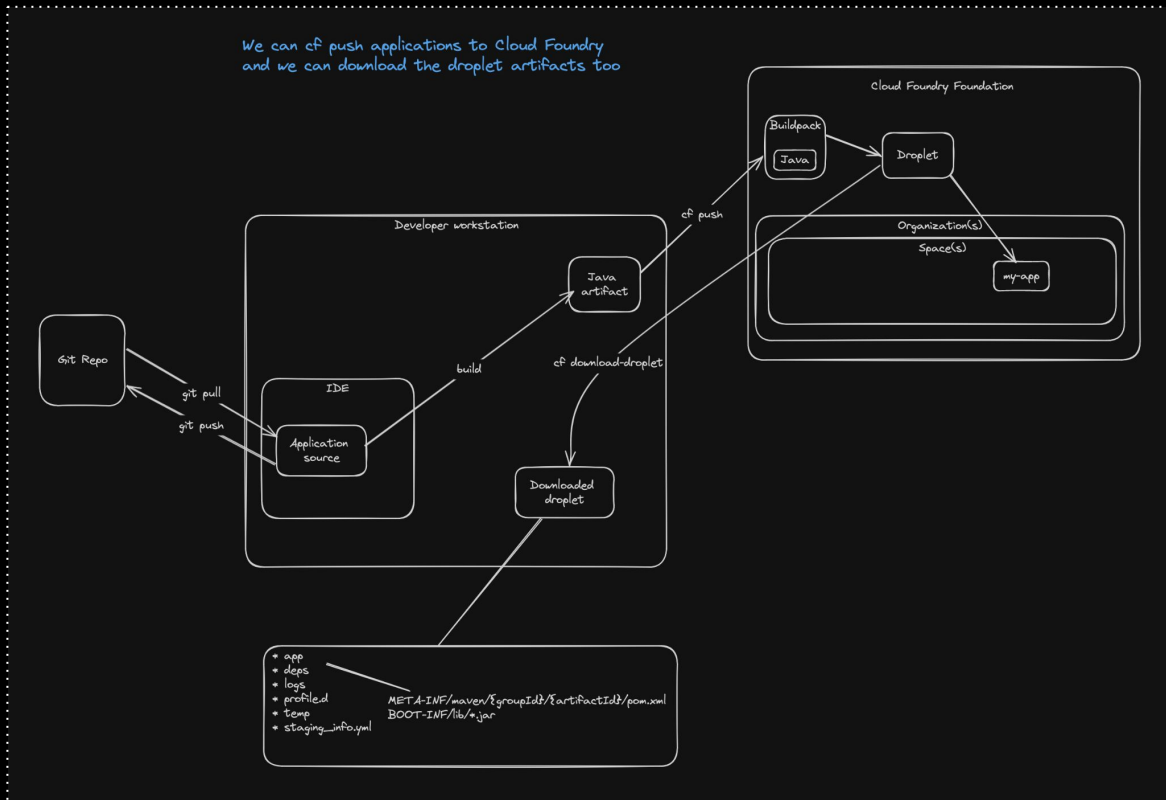
Chris Phillipson  
Copyright 2024

# An aside on Cloud Foundry

## Did you know?

### Commands

- `cf push`
- `cf create-service`
- `cf bind-service`
- `cf apps`
- `cf services`
- `cf download-droplet`



Chris Phillipson  
Copyright 2024



# We want to know more

## Domain

We want to collect details about the **applications** and **services**, right?

### AppDetail

- + pk
- + organization
- + space
- + appId
- + appName
- + buildpack
- + buildpackVersion
- + buildpackReleaseType
- + buildpackReleaseDate
- + buildpackLatestVersion
- + buildpackLatestUrl
- + image
- + stack
- + runningInstances
- + totalInstances
- + memoryUsed
- + diskUsed
- + memoryQuota
- + diskQuota
- + lastPushed
- + lastEvent
- + lastEventActor
- + lastEventTime
- + requestedState
- + routes

### Buildpack

- + id
- + name
- + position
- + enabled
- + locked
- + filename
- + stack

### ServiceInstanceDetail

- + pk
- + organization
- + space
- + serviceInstanceId
- + name
- + service
- + description
- + plan
- + type
- + applications
- + lastOperation
- + lastUpdated
- + dashboardUrl
- + requestedState

### JavaAppDetail

- + pk
- + organization
- + space
- + appId
- + appName
- + dropletId
- + pomContents
- + jars
- + springDependencies

### Organization

- + id
- + name

### Space

- + organizationId
- + organizationName
- + spaceId
- + spaceName

### AppRelationship

- + pk
- + organization
- + space
- + appId
- + appName
- + serviceInstanceId
- + serviceName
- + serviceOffering



Chris Phillipson  
Copyright 2024

# The building blocks



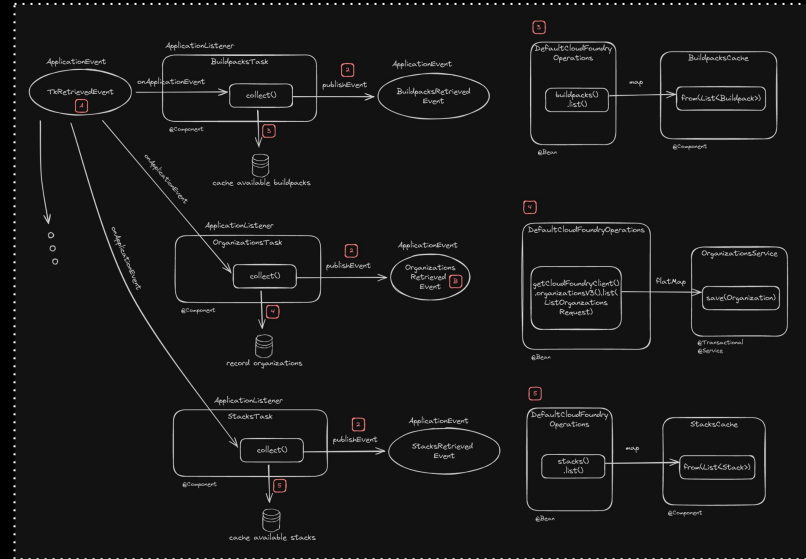
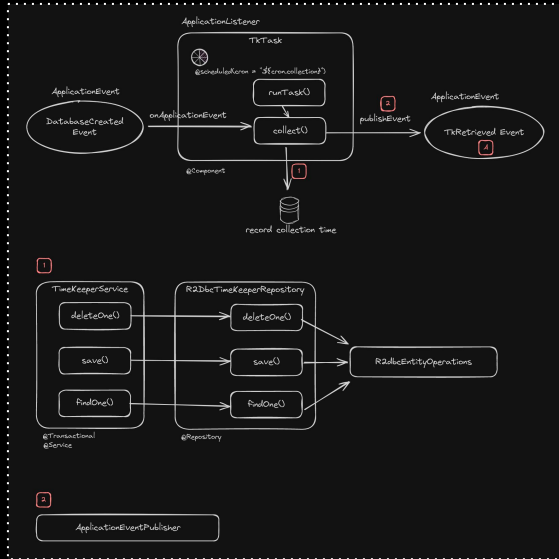
```
graph TD
    RS[Reactive Stack] --> SB1[org.springframework.boot  
spring-boot-starter-webflux]
    SB1 --> SB2[org.springframework.boot  
spring-boot-starter-data-r2dbc]
    SB2 --> CF1[org.cloudfoundry  
cloudfoundry-client-reactor]
    SB2 --> CF2[org.cloudfoundry  
cloudfoundry-operations]
    SB2 --> R2[asyncer-io  
r2dbc-mysql]
    SB2 --> R3[io.r2dbc  
r2dbc-postgres]
    SB2 --> R4[io.r2dbc  
r2dbc-h2]
    CF1 --> CF[Cloud Foundry]
    CF2 --> CF
    R2 --> MySQL[MySQL]
    R3 --> PG[Postgres]
    R4 --> H2[H2]
    R2 --> AC[org.apache  
commons-compress]
    R3 --> AC
    R4 --> AC
    AC --> IP[io.pivotal.cfenv  
java-cfenv-boot]
    IP --> SBRM[org.cftoolsuite  
spring-boot-starter-runtime-metadata]
```



Chris Phillipson  
Copyright 2024

# Architecture

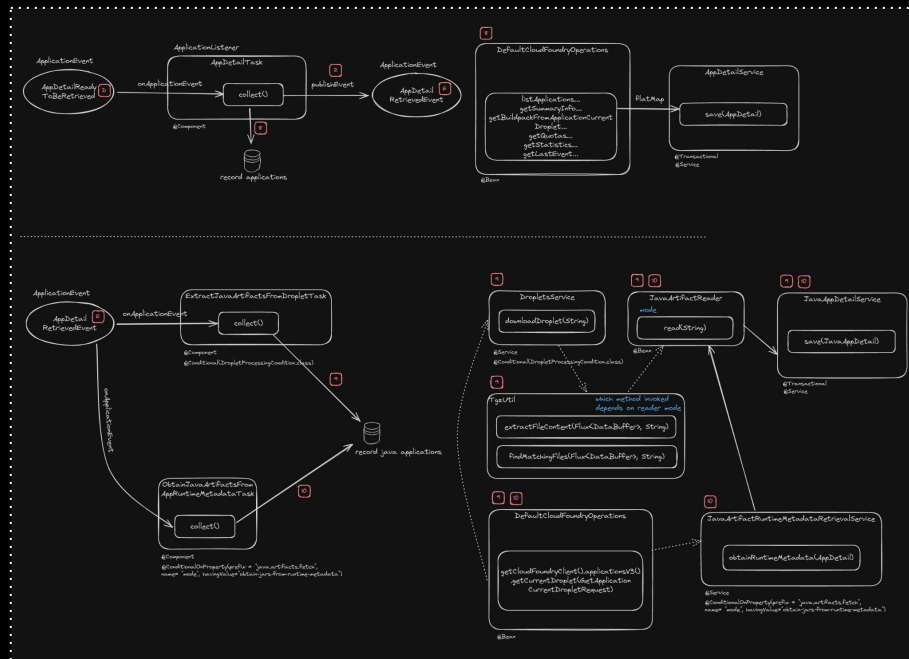
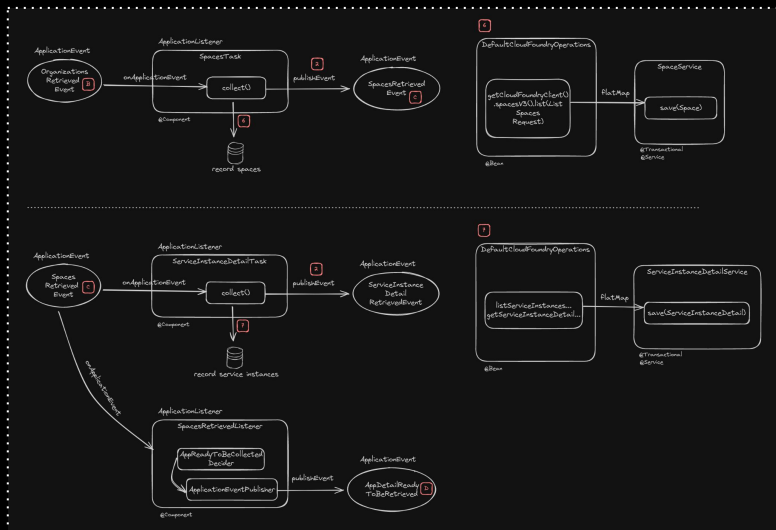
## Pubsub pattern



Chris Phillipson  
Copyright 2024

# Architecture

## Pubsub pattern (continued)



Chris Phillipson  
Copyright 2024

Varying approaches to collecting dependencies

# Reporting on dependencies

## Contrasting approaches

### Unpack Maven POM contents in droplet

- unpacks and filters application droplets in-memory for a pom.xml file, return the contents
- only returns results if application was built with Maven

Expensive  
Not comprehensive  
Not Portable

### List jars in droplet

- unpacks and filters application droplets in-memory returning list of embedded .jar files

Expensive  
Comprehensive  
Not portable

### Obtain jars from runtime metadata

- expects applications to expose /actuator/jars endpoint from which .jar files are obtained

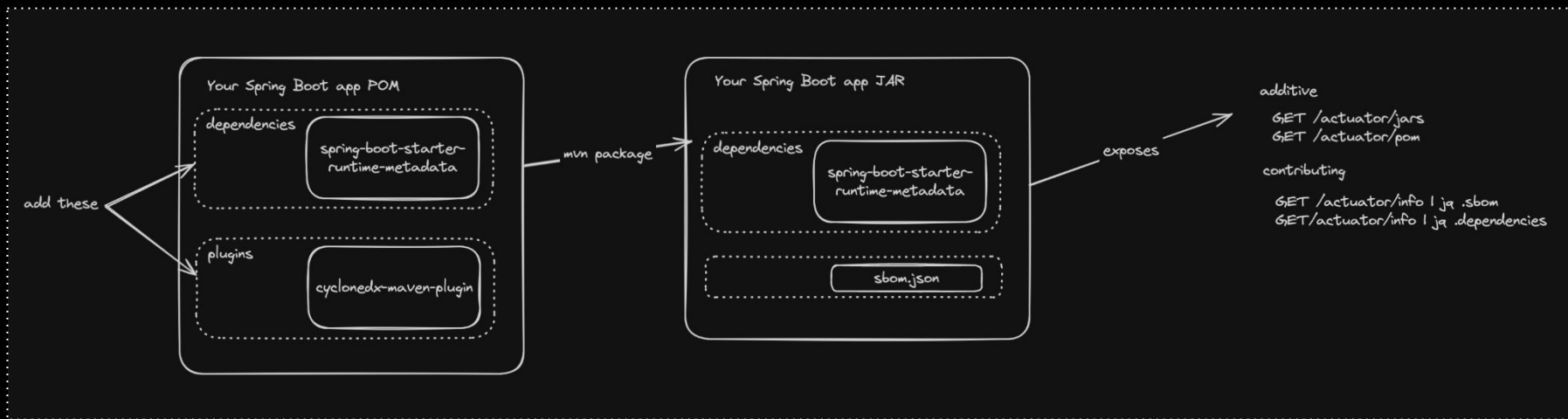
Light-weight  
Not comprehensive  
Portable



Chris Phillipson  
Copyright 2024

# Let's look at that last option

## Extending the Spring Boot Actuator info endpoint



Chris Phillipson  
Copyright 2024

Coming soon in Spring Boot 3.3  
a dedicated sbom endpoint

# Demo



Chris Phillipson  
Copyright 2024

# For further research

<https://projectreactor.io/>

<https://spring.io/projects/spring-boot>

<https://spring.io/projects/spring-cloud>

<https://docs.spring.io/spring-framework/reference/web/webflux.html>

<https://spring.io/projects/spring-data-r2dbc>

<https://cyclonedx.org/>

<https://github.com/CycloneDX/cyclonedx-gradle-plugin>

<https://github.com/CycloneDX/cyclonedx-maven-plugin>

<https://github.com/CycloneDX/cdxgen>

<https://www.endorlabs.com/learn/how-cyclonedx-vex-makes-your-sbom-useful>

<https://github.com/CycloneDX/bom-examples/tree/master/VEX>

<https://github.com/cf-toolsuite/home>

<https://github.com/cf-toolsuite/cf-butler>

<https://github.com/cf-toolsuite/cf-hoover>

<https://github.com/cf-toolsuite/cf-archivist>

<https://github.com/cf-toolsuite/spring-boot-starter-runtime-metadata>

<https://github.com/cloudfoundry/cf-java-client>

<https://github.com/kubernetes-client/java>

<https://github.com/kubernetes-client/java/blob/master/examples/samples-release-18/src/main/java/io/kubernetes/client/examples/SpringControllerExample.java>

<https://github.com/googleapis/google-cloud-java/tree/main/java-run>



Chris Phillipson  
Copyright 2024