What's (new) with Spring Boot and Containers

Eva Maria Panadero Peris

Matthias Haeussler

Novatec - now a part of CGI

Who we are

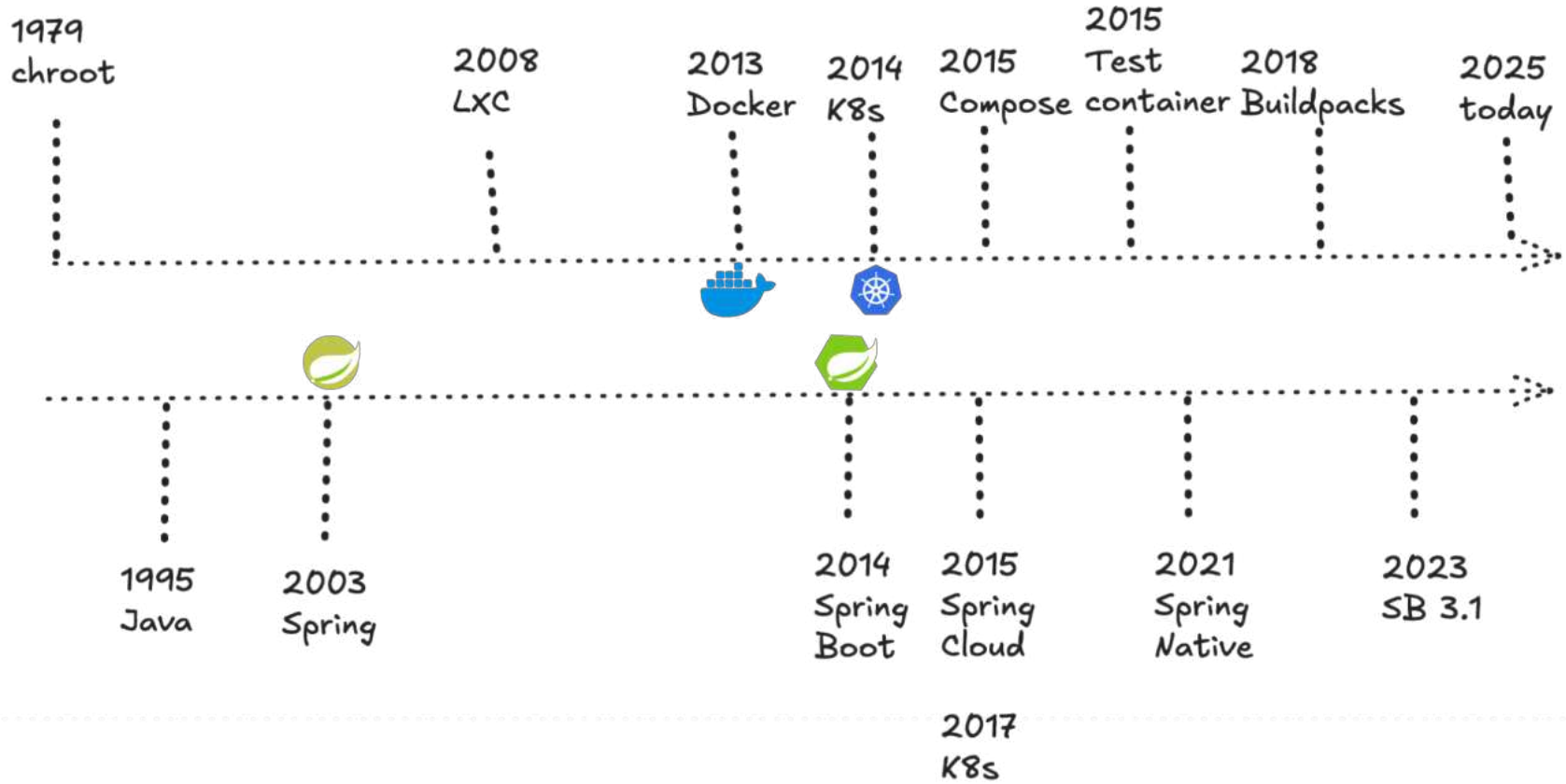
Eva Panadero
Senior Backend Engineer,
CGI/Novatec

Matthias Haeussler
Chief Technologist,
CGI/Novatec

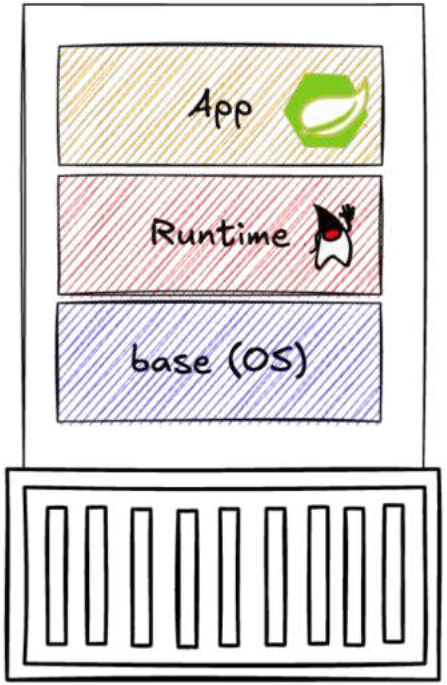
Why this talk?

Some Background …

History!

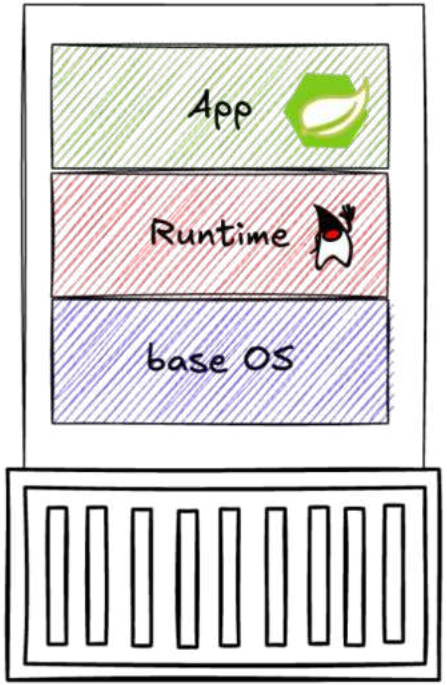

1979
chroot
2008
LXC
2013
Docker
2014
K8s
2015
Compose
2015
Test
container
2018
Buildpacks
2025
today
1995
Java
2003
Spring
2014
Spring
Boot
2015
Spring
Cloud
2017
K8s
2021
Spring
Native
2023
SB 3.1

Standard case - Application Container



Standard case - Application Container

standardized

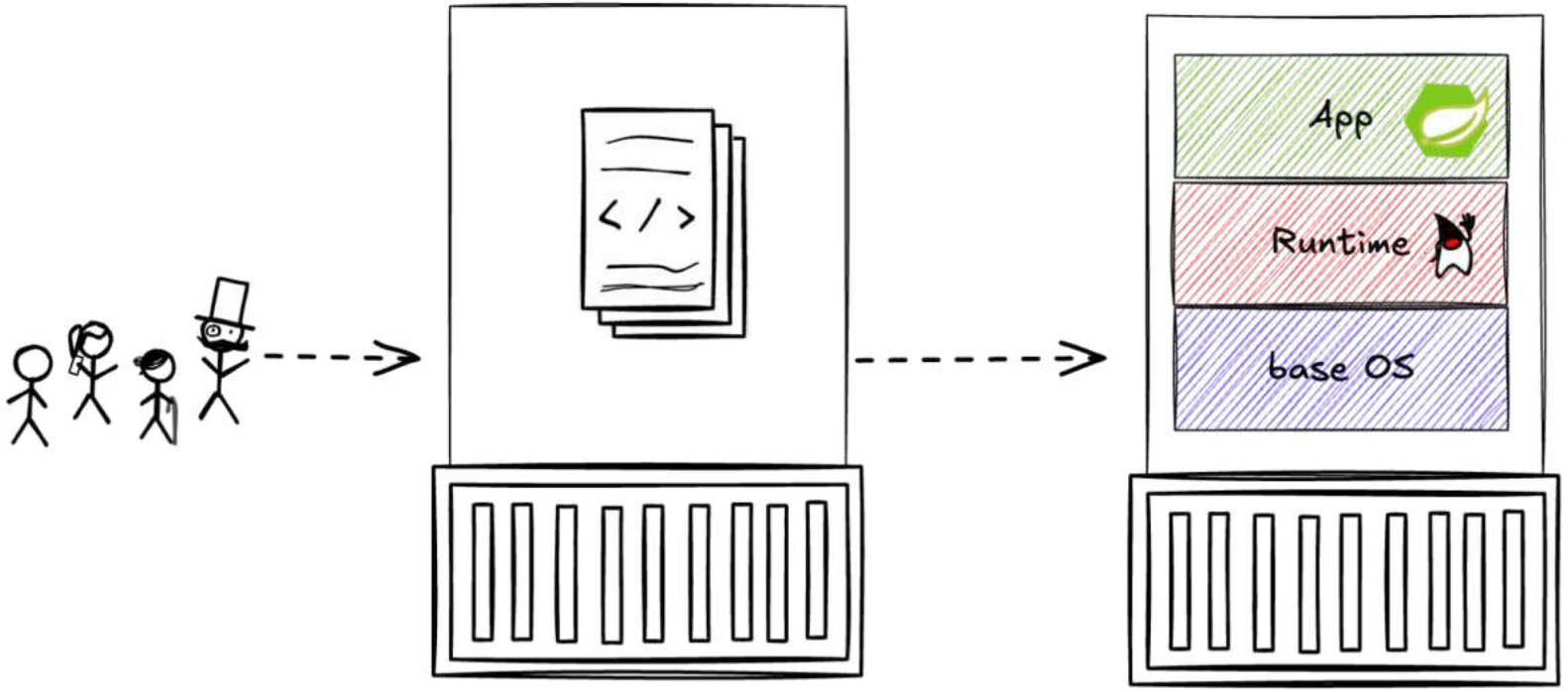


portable

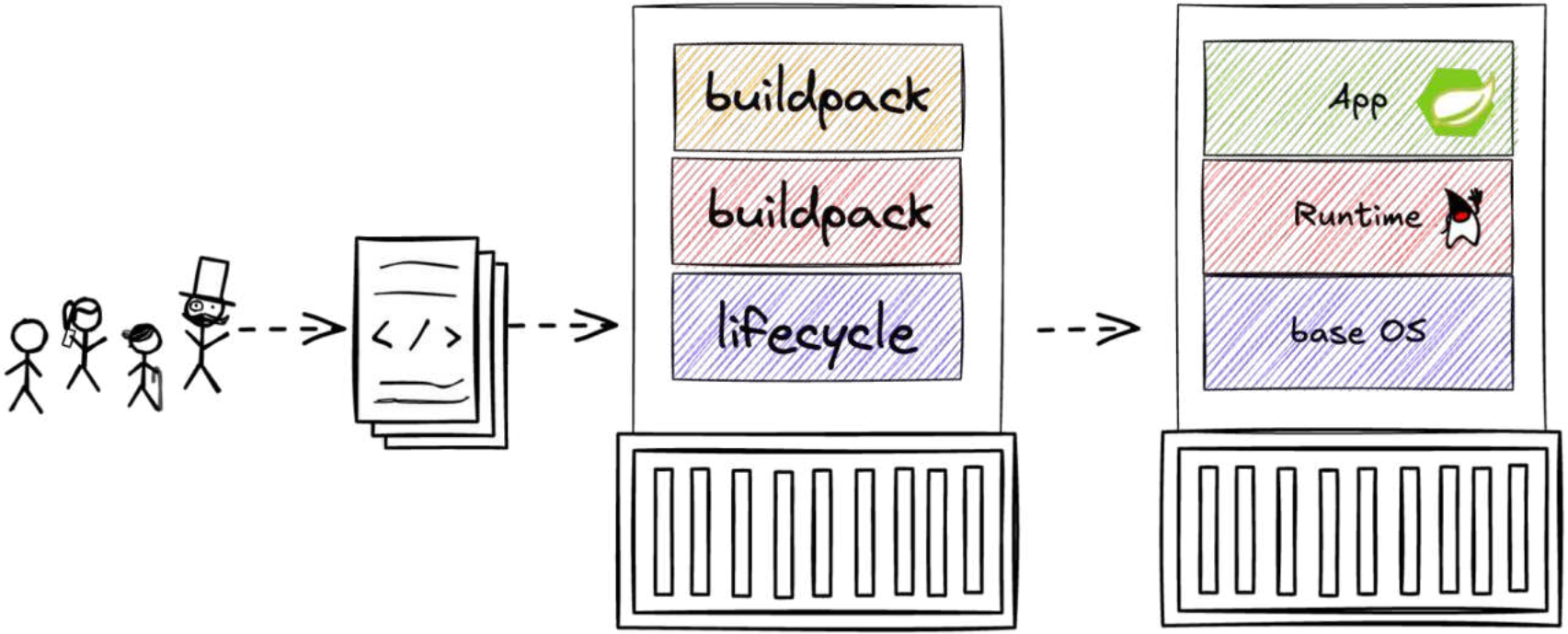
repeatable

consistent

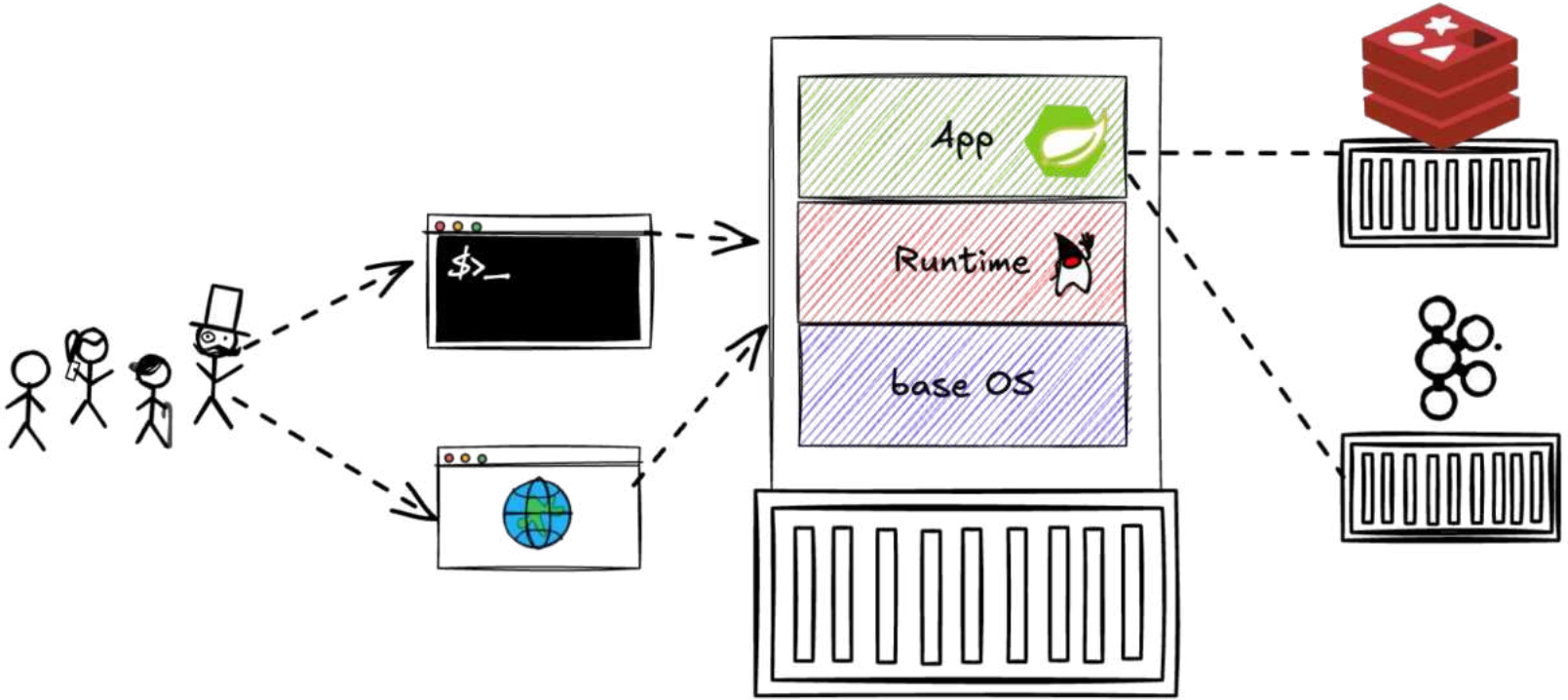
Container to DEVELOP



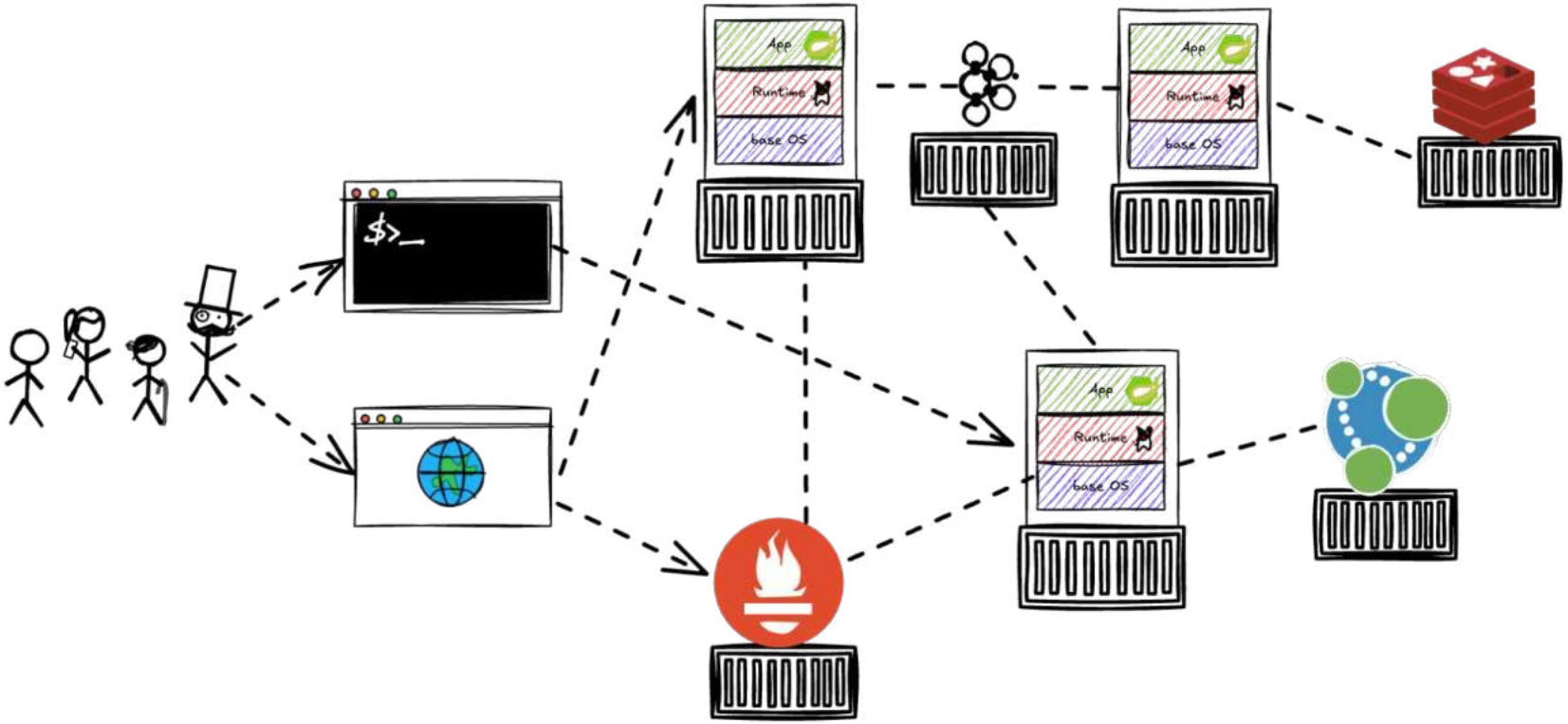
Container to BUILD



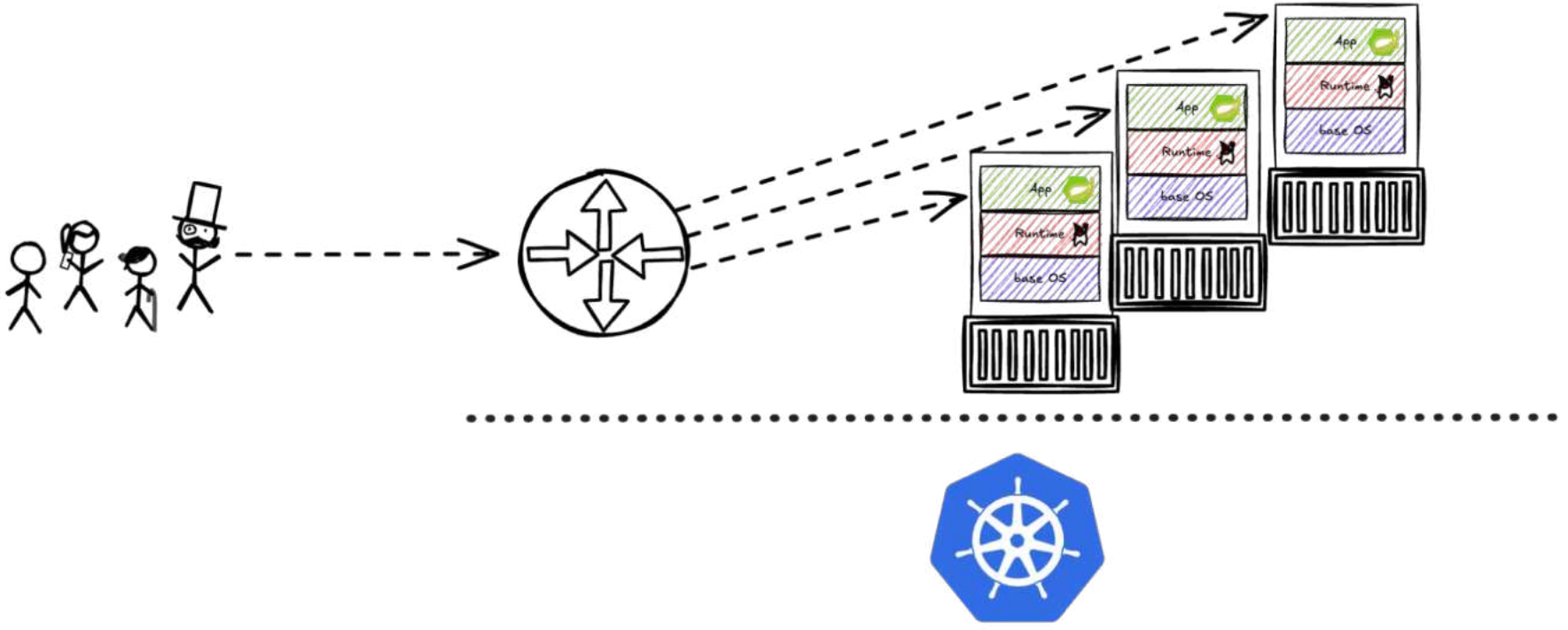
Container to TEST



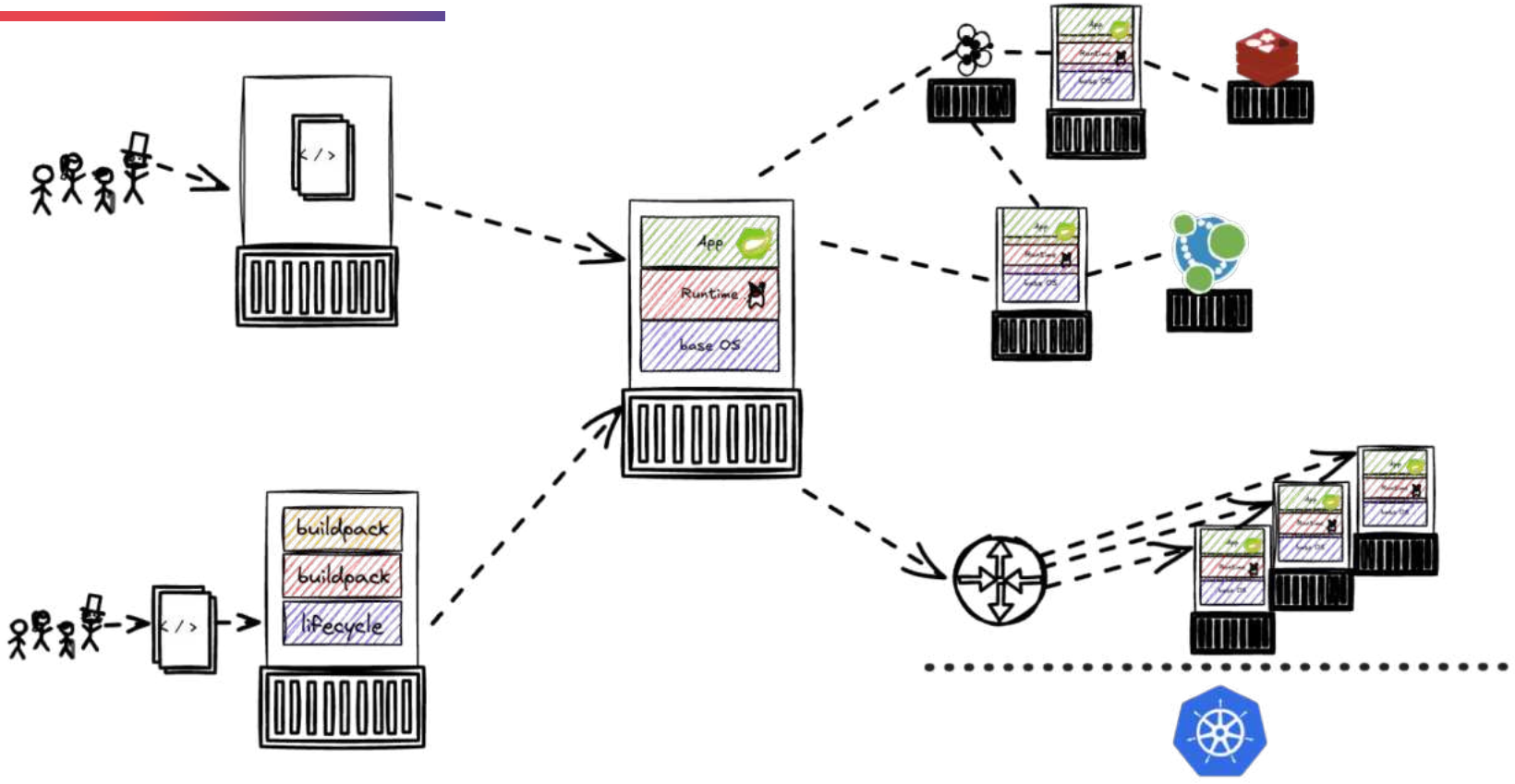
Container to RUN



Container to SCALE



Sooo many integrations

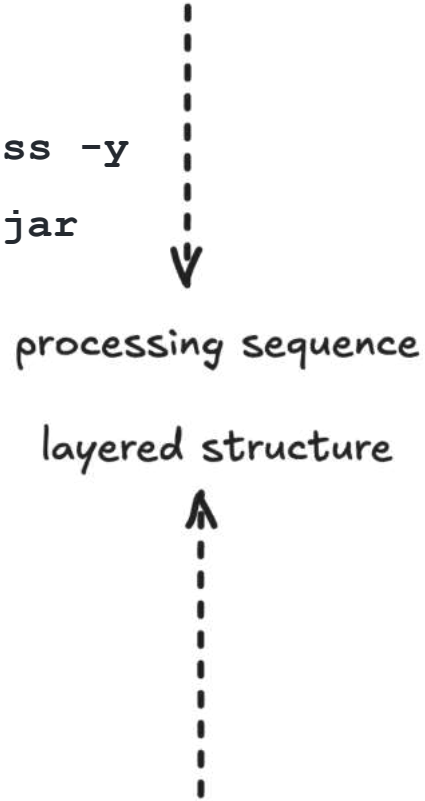


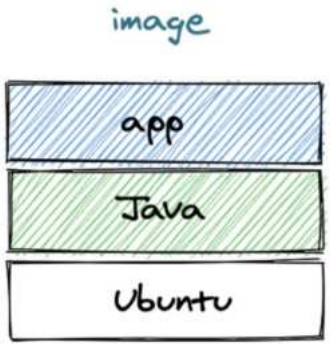
develop
(later)

build

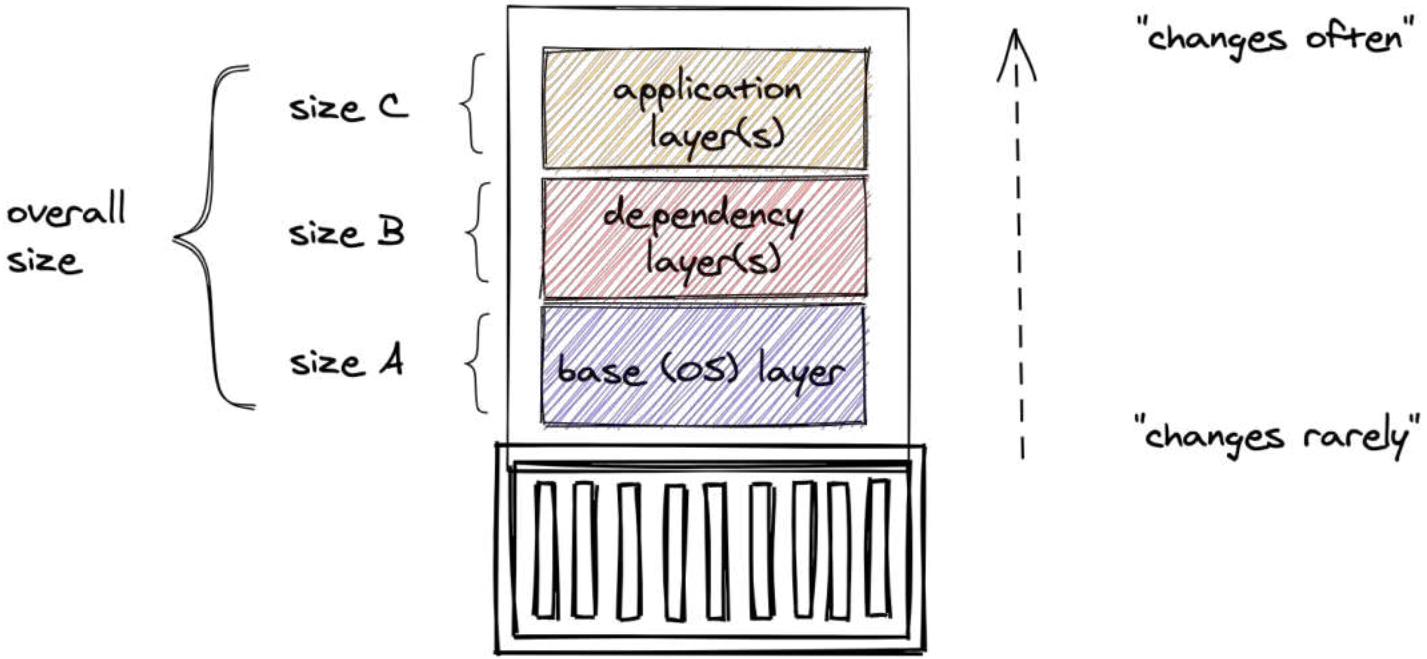
Dockerfile (simple)

```
FROM ubuntu:24.04
RUN apt update && apt install openjdk-21-jre-headless -y
COPY target/simplecode-0.0.1-SNAPSHOT.jar /opt/app.jar
CMD ["java","-jar","/opt/app.jar"]
```

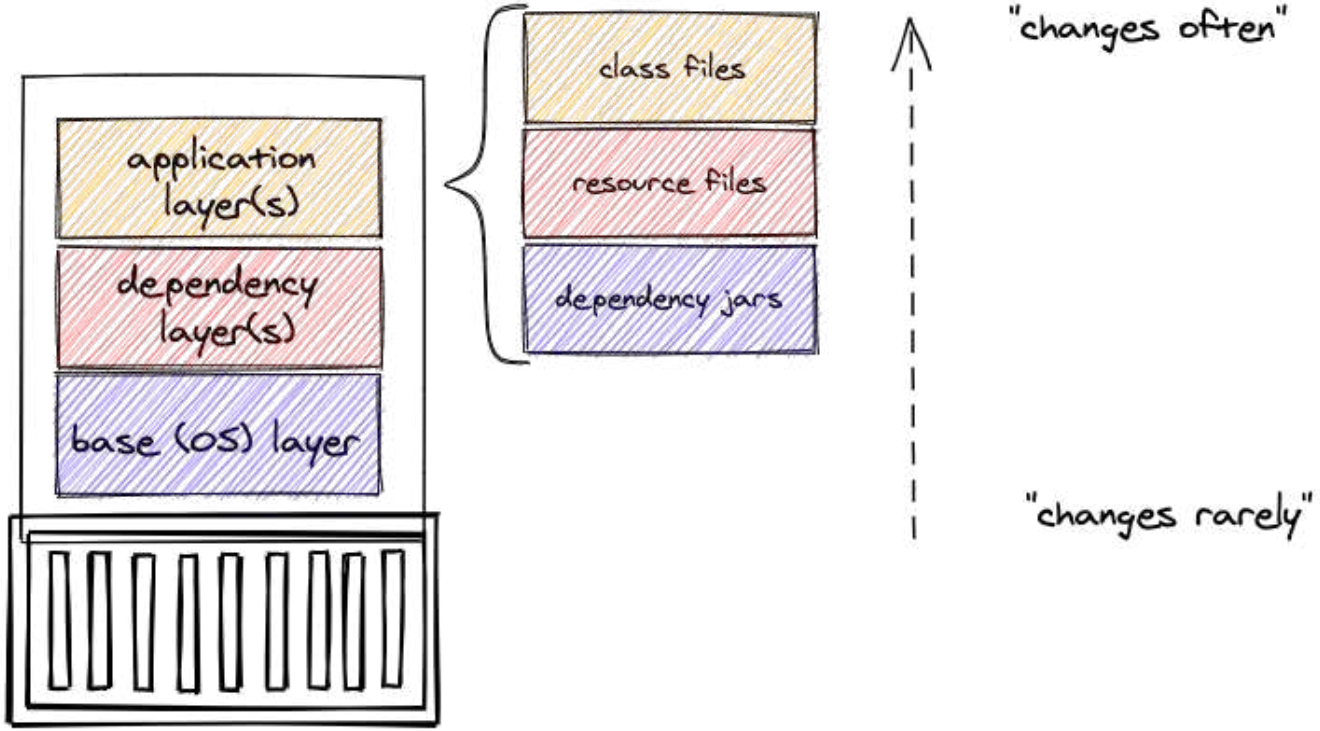




Considerations for structure and size



Considerations for Java



Layered Jar

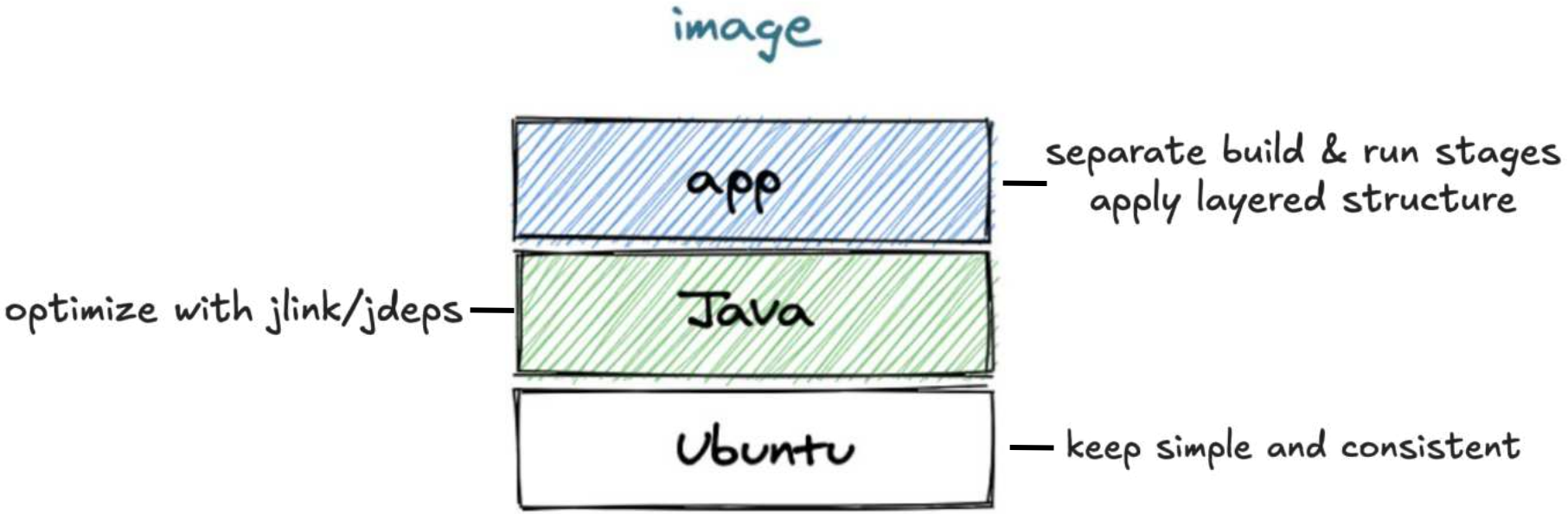
```
FROM eclipse-temurin:21-jre AS builder
COPY --from=maven /opt/app/target/simplecode-0.0.1-SNAPSHOT.jar application.jar
RUN java -Djarmode=layertools -jar application.jar extract

FROM eclipse-temurin:21-jre
COPY --from=builder application/dependencies/ ./
COPY --from=builder application/spring-boot-loader/ ./
COPY --from=builder application/snapshot-dependencies/ ./
COPY --from=builder application/application/ ./
ENTRYPOINT ["java","org.springframework.boot.loader.JarLauncher"]
```

jlink and jdeps

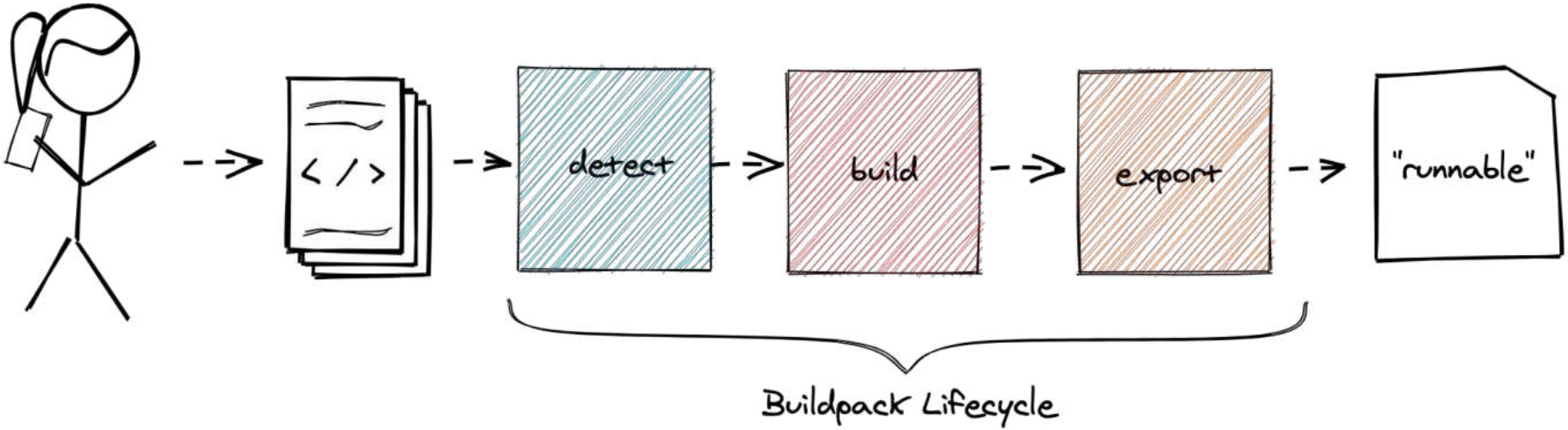
```
FROM maven:3-eclipse-temurin-21 AS build
RUN --mount=type=cache,target=/root/.m2 mvn package
RUN jdeps --ignore-missing-deps -q  \
    --recursive  \
    --multi-release 21  \
    --print-module-deps  \
    --class-path 'BOOT-INF/lib/*'  \
    target/simplecode-0.0.1-SNAPSHOT.jar > deps.info
RUN jlink \
    --add-modules $(cat deps.info) \
    --strip-debug \
    --compress 2 \
    --no-header-files \
    --no-man-pages \
    --output /myjre

FROM debian:bookworm-slim
COPY --from=build /myjre $JAVA_HOME
COPY --from=build /usr/src/project/target/simplecode-0.0.1-SNAPSHOT.jar /project/
```

```
FROM maven:3-eclipse-temurin-21 AS build
RUN mkdir /opt/app
COPY src /opt/app/src
COPY pom.xml /opt/app
WORKDIR /opt/app
RUN --mount=type=cache,target=/root/.m2 mvn package -DskipTests
RUN jar xf target/simplecode-0.0.1-SNAPSHOT.jar
RUN jdeps --ignore-missing-deps -q  \
   --recursive  \
   --multi-release 21  \
   --print-module-deps  \
   --class-path 'BOOT-INF/lib/*'  \
   target/simplecode-0.0.1-SNAPSHOT.jar > deps.info
RUN jlink \
   --add-modules $(cat deps.info) \
   --strip-debug \
   --compress 2 \
   --no-header-files \
   --no-man-pages \
   --output /myjre

FROM eclipse-temurin:21-jre AS extractor
RUN mkdir /opt/app
WORKDIR /opt/app
COPY --from=build /opt/app/target/simplecode-0.0.1-SNAPSHOT.jar application.jar
RUN java -Djarmode=layertools -jar application.jar extract

FROM ubuntu:jammy
ENV JAVA_HOME /opt/java/jdk21
ENV PATH $JAVA_HOME/bin:$PATH
COPY --from=build /myjre $JAVA_HOME
RUN mkdir /opt/app
WORKDIR /opt/app
COPY --from=extractor /opt/app/dependencies/ ./
COPY --from=extractor /opt/app/spring-boot-loader/ ./
COPY --from=extractor /opt/app/snapshot-dependencies/ ./
COPY --from=extractor /opt/app/application/ ./
ENTRYPOINT ["java","-XX:+UseParallelGC","-XX:MaxRAMPercentage=75","org.springframework.boot.loader.JarLauncher"]
```

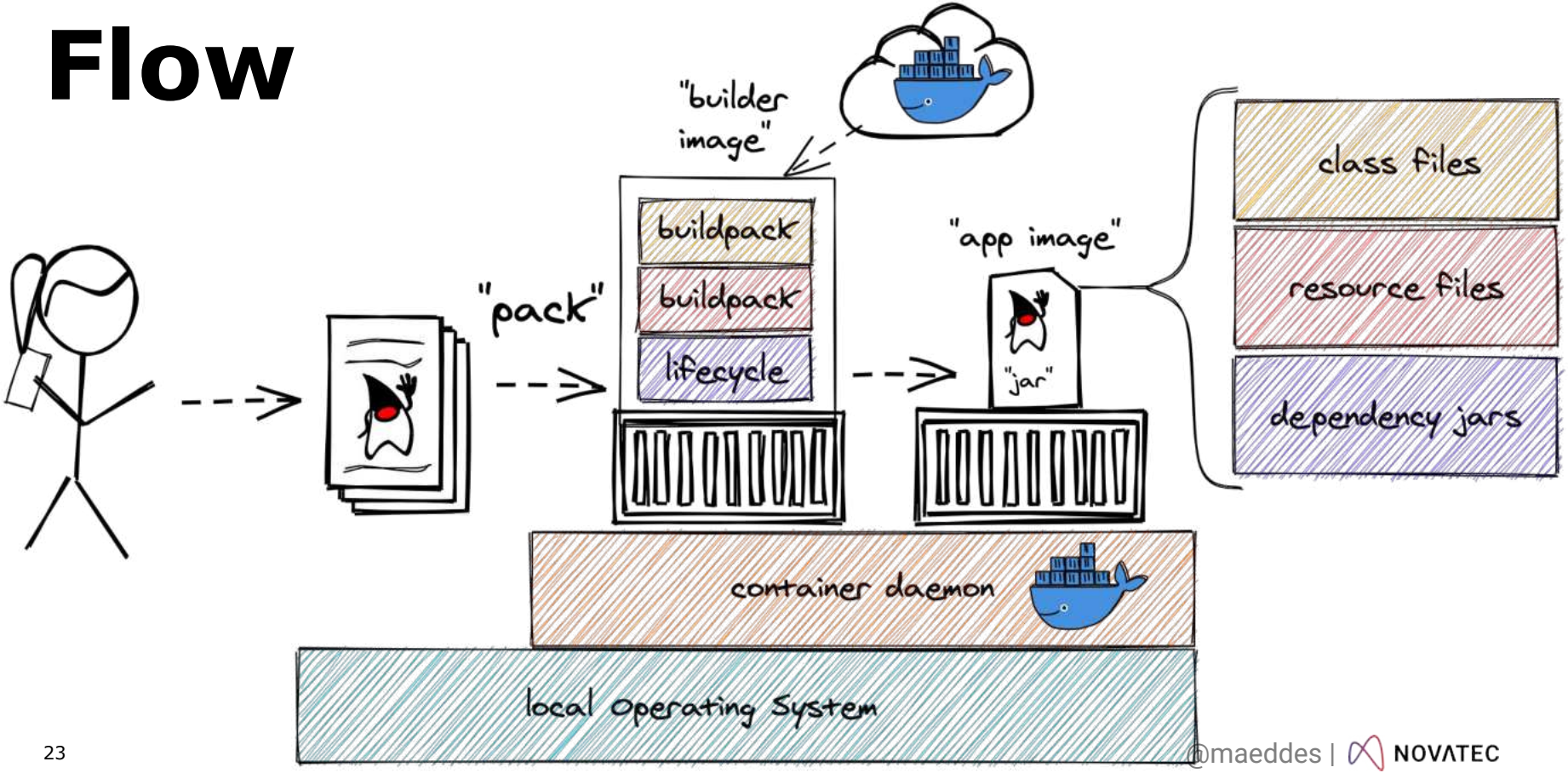
Learnings from optimized Dockerfiles



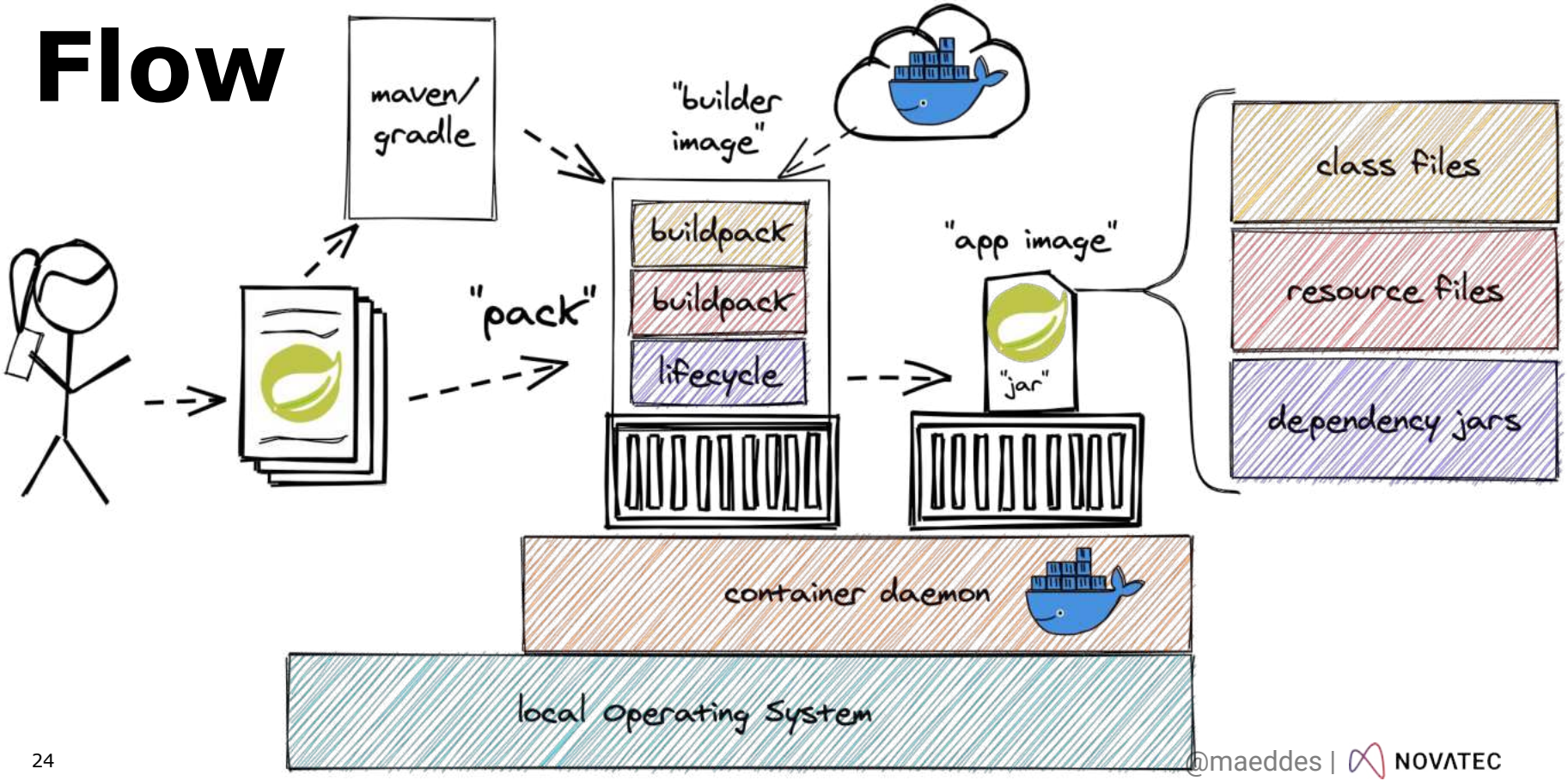
Buildpacks idea



Flow

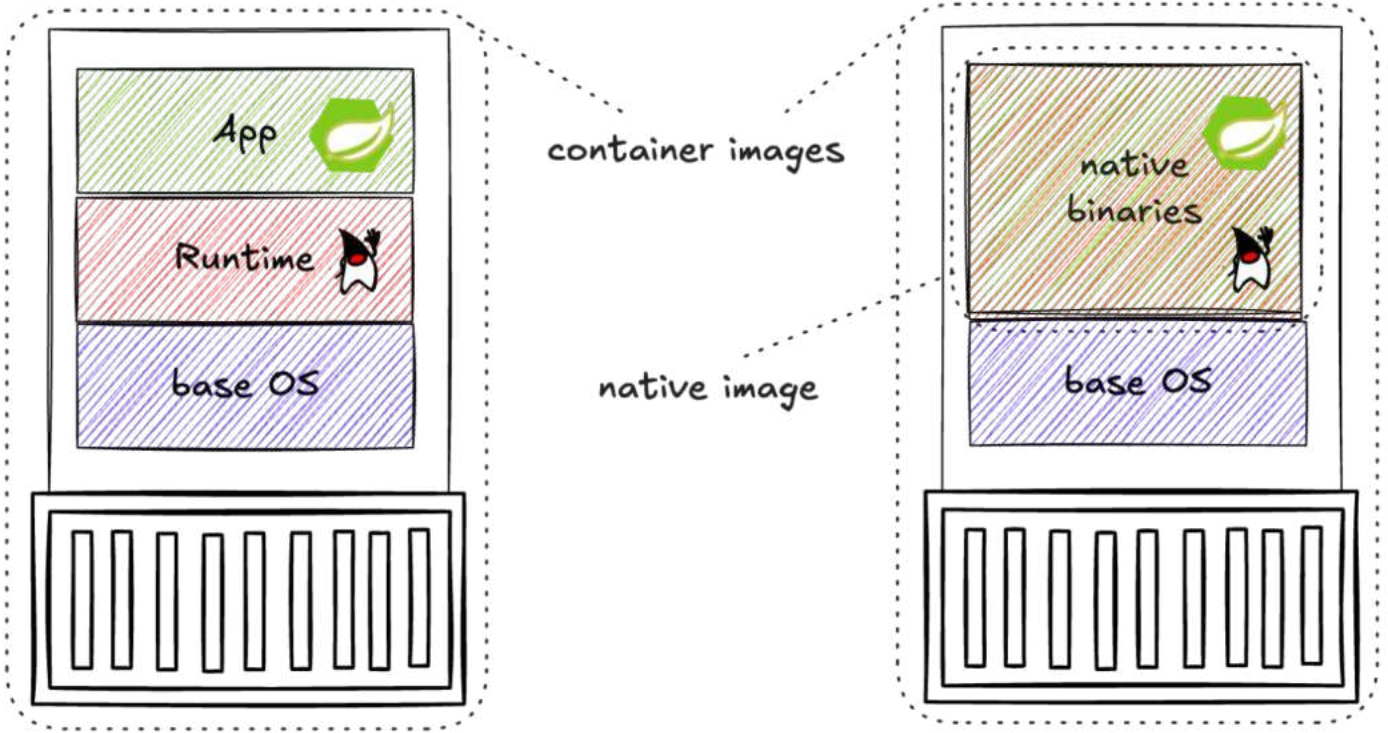

"builder image"
buildpack
buildpack
lifecycle
"pack"
"app image"
"jar"
class files
resource files
dependency jars
container daemon
local operating System

Flow


maven/
gradle
"builder
image"
buildpack
buildpack
lifecycle
"pack"
"app image"
"jar"
class files
resource files
dependency jars
container daemon
local operating System

native images/ GraalVM

Concept


App
Runtime
base OS
container images
native image
native
binaries
base OS

GraalVM - Building Types

Build options

```
#build JAR file
mvn clean package

#build Native Image
mvn -Pnative native:compile

#build JAR inside a Docker Image
mvn spring-boot:build-image

#build Native Image inside a Docker Image
mvn spring-boot:build-image -Pnative
-Dspring-boot.build-image.imageName=graalvm-demo:native

#build with Pack
pack build graalvm-native-images \
  --builder paketobuildpacks/builder:tiny \
  --env BP_NATIVE_IMAGE=true
```

GraalVM - Run Applications

Run options

```
#run JAR file
java -jar target/graalvm-0.0.1-SNAPSHOT.jar

#run Native Image
./target/graalvm

#run JAR inside a Docker Image
docker run graalvm:0.0.1-SNAPSHOT

#run Native Image inside a Docker Image
docker run graalvm-demo:native

#run the Buildpack Image
docker run graalvm-native-images
```

GraalVM - Setup

Adding the dependencies

```
<parent>
     <groupId>org.springframework.boot</groupId>
     <artifactId>spring-boot-starter-parent</artifactId>
     <version>3.4.5</version>
</parent>

…
<plugin>
     <groupId>org.graalvm.buildtools</groupId>
     <artifactId>native-maven-plugin</artifactId>
</plugin>
```

GraalVM - Setup

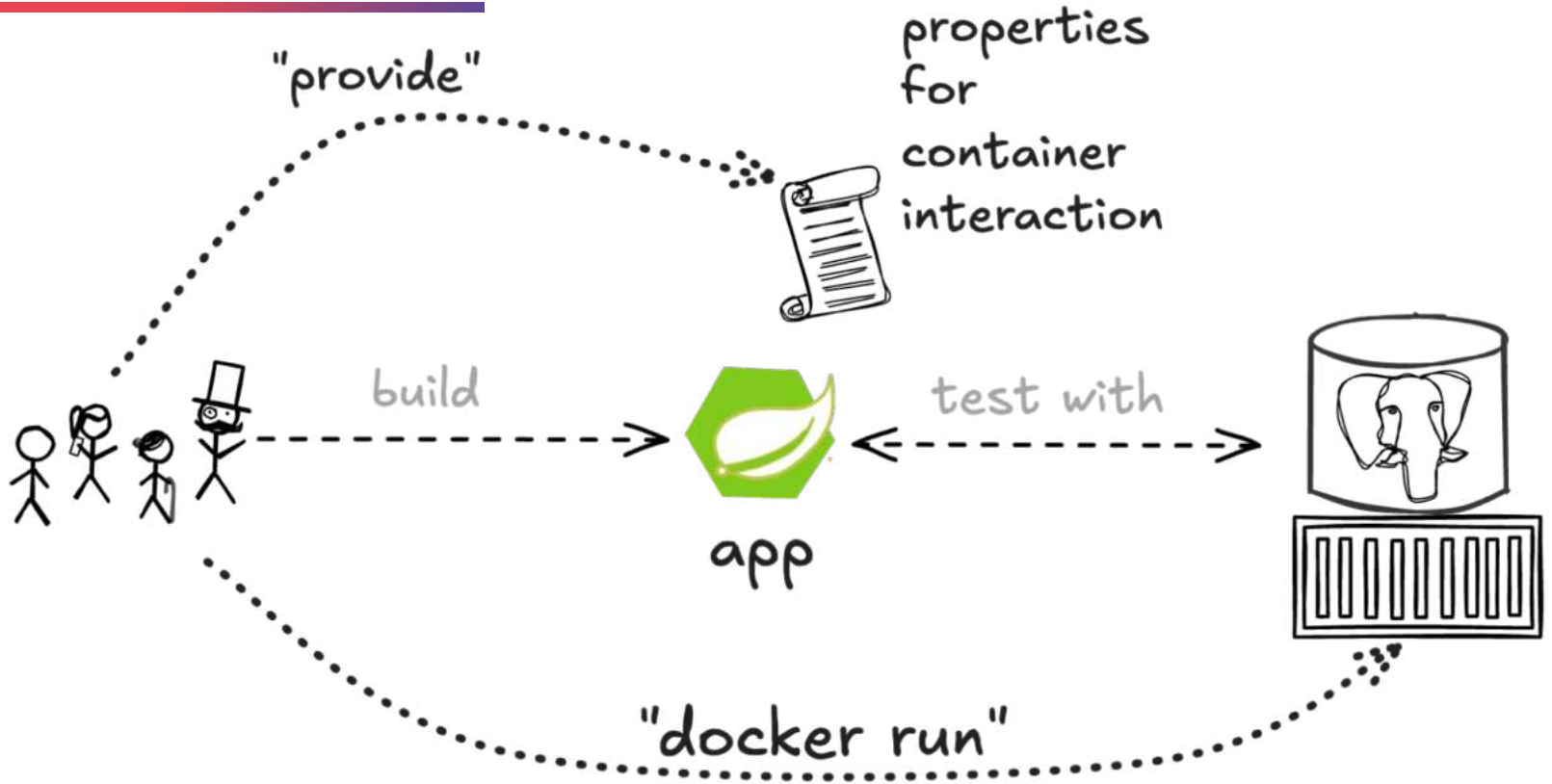
Spring Boot useful Links:

- https://docs.spring.io/spring-boot/reference/packaging/native-image/introducing-graalvm-native-images.html
- https://docs.spring.io/spring-boot/how-to/native-image/developing-your-first-application.html
- https://www.graalvm.org/latest/getting-started/
- https://buildpacks.io/docs/for-platform-operators/how-to/integrate-ci/pack/

testing

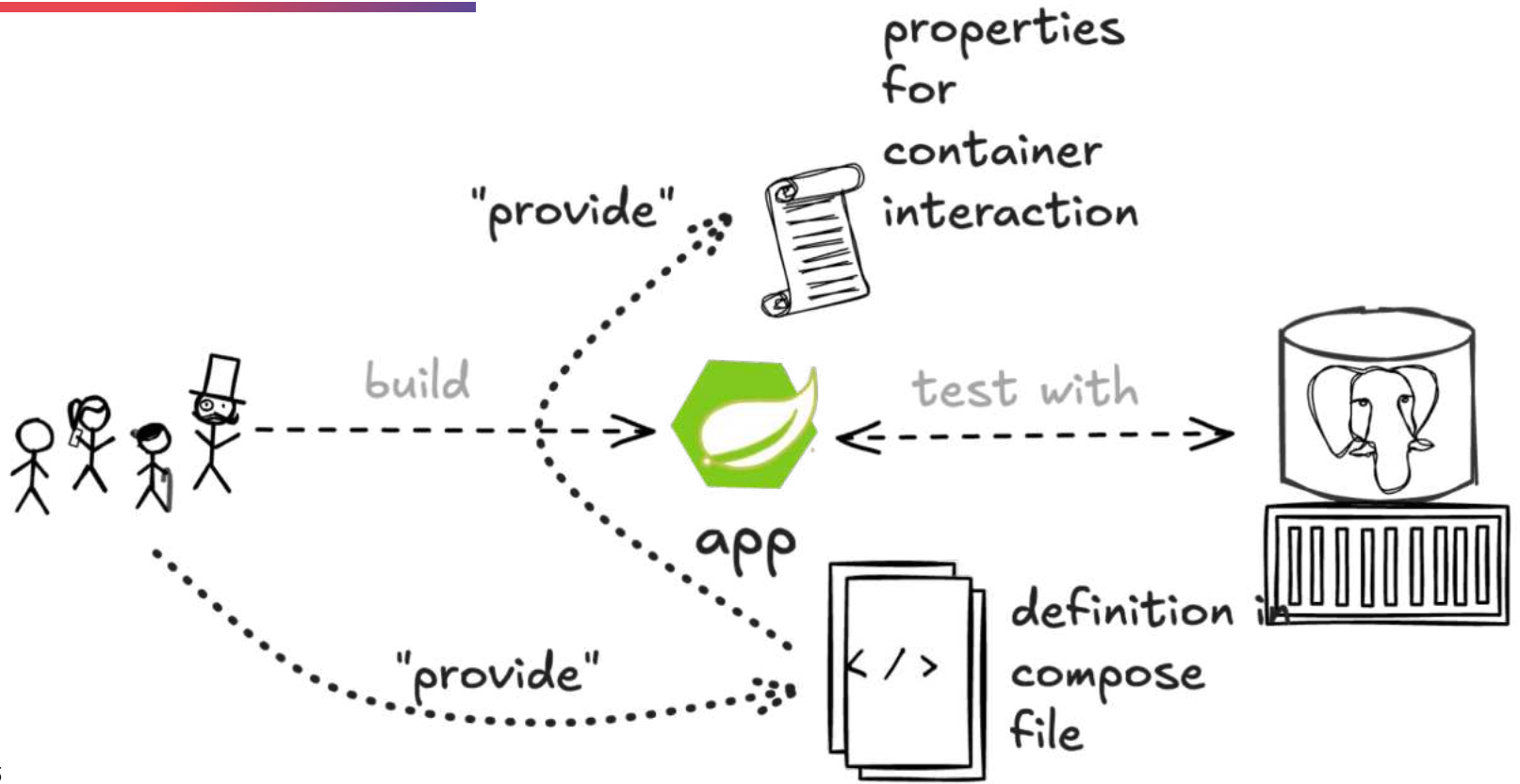
Basic principle

Basic principle



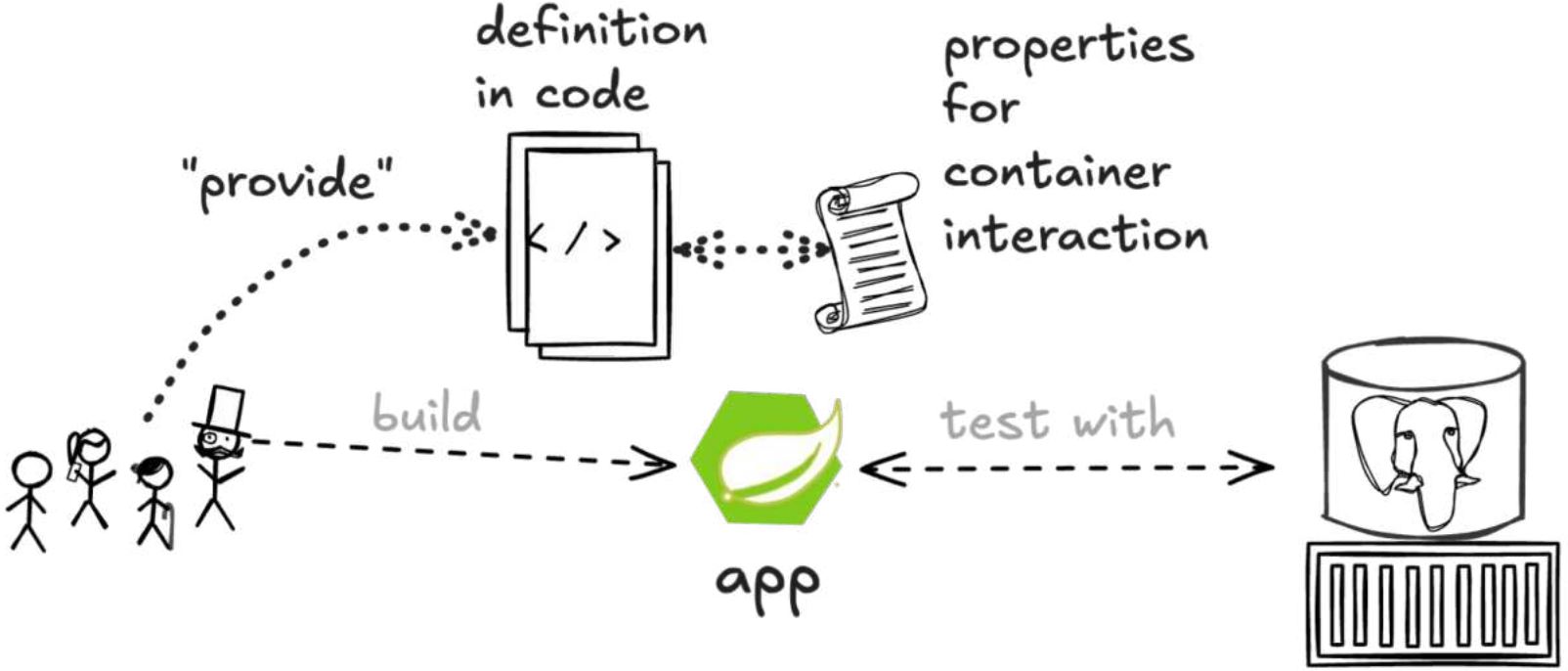
docker compose

Integration principle



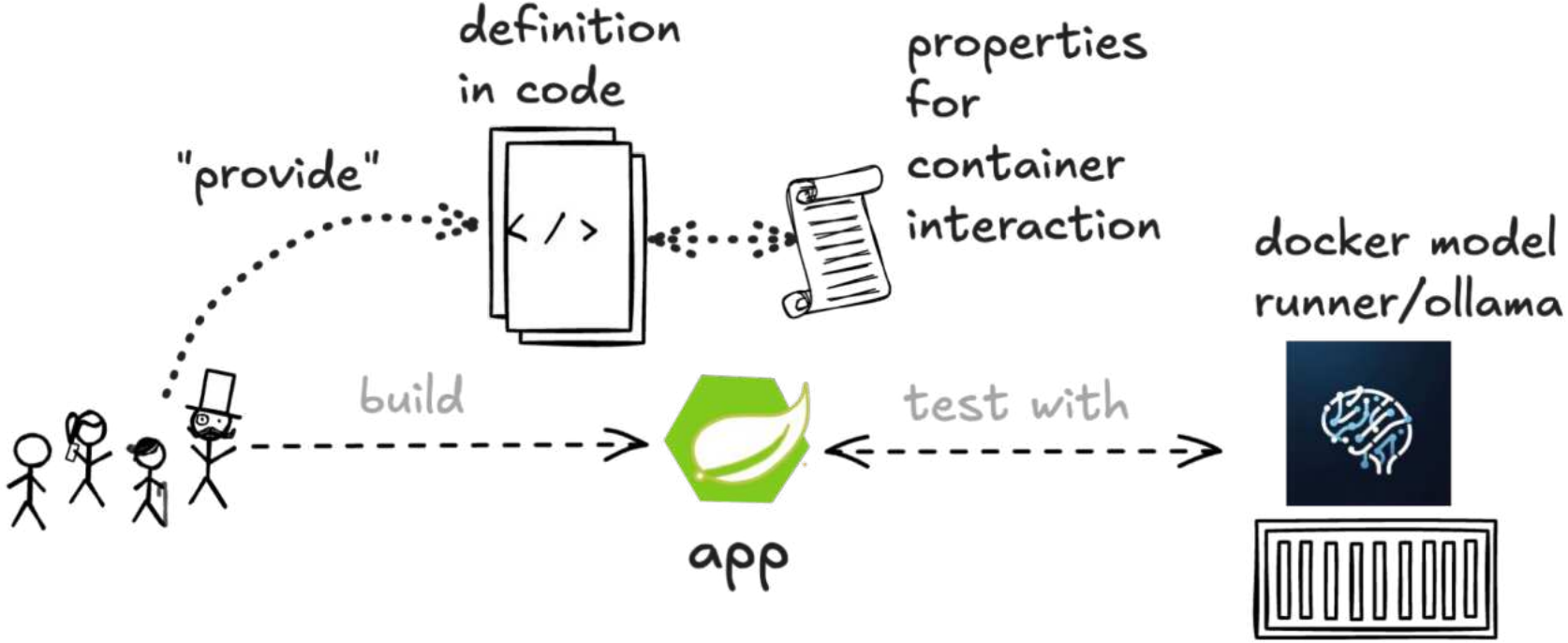
testcontainers

Integration principle



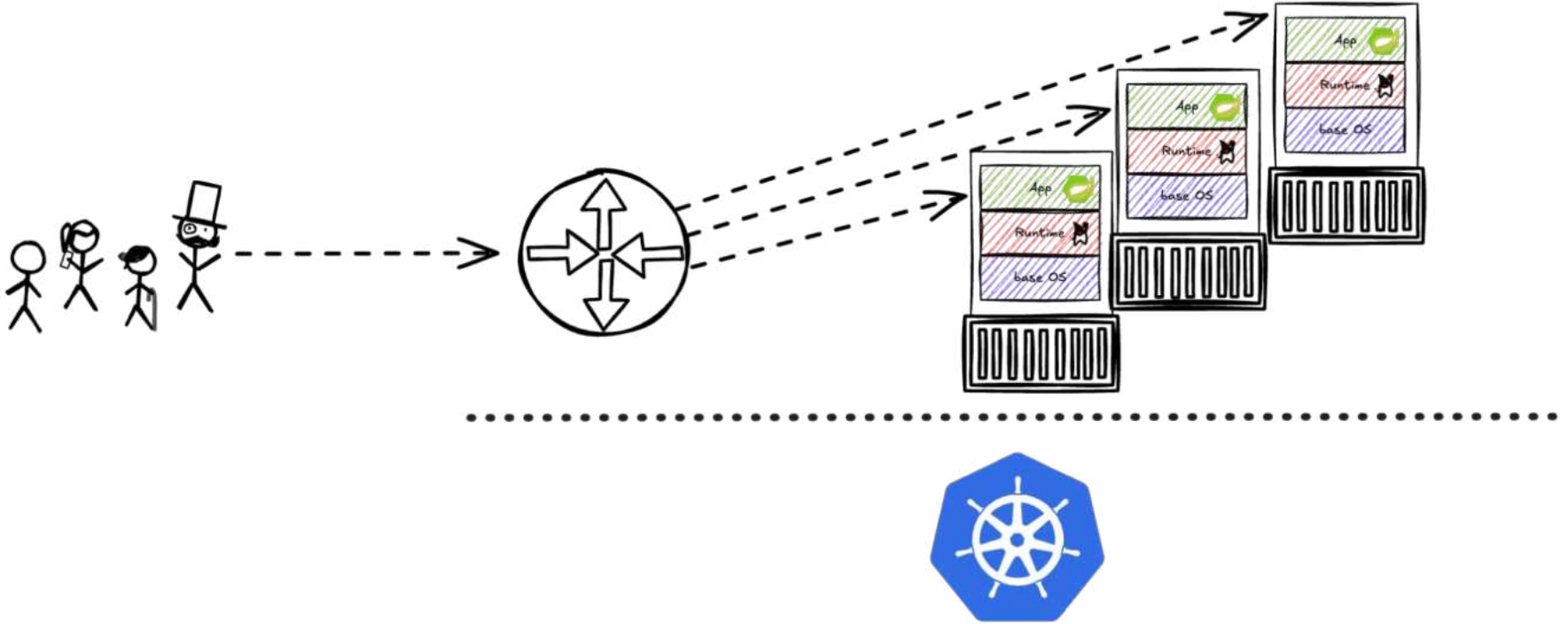
AI 

Integration principle

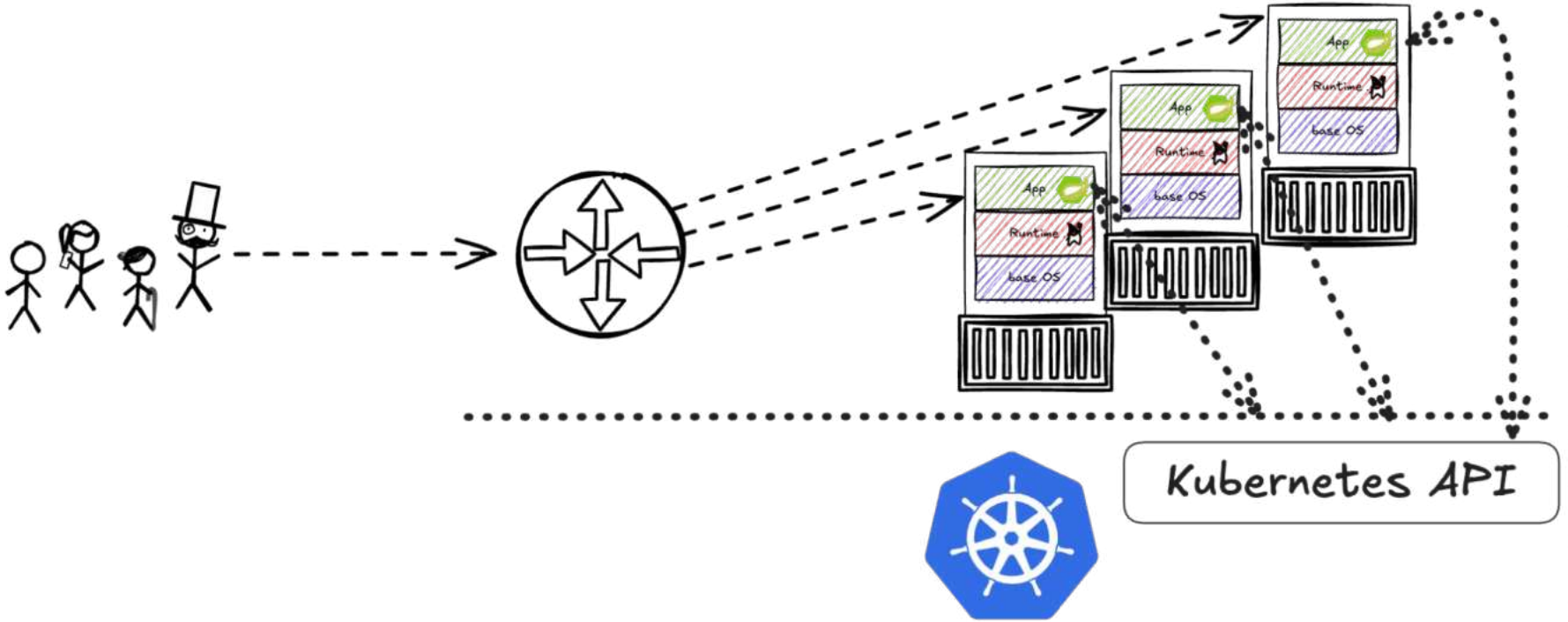


Kubernetes

Integration principle

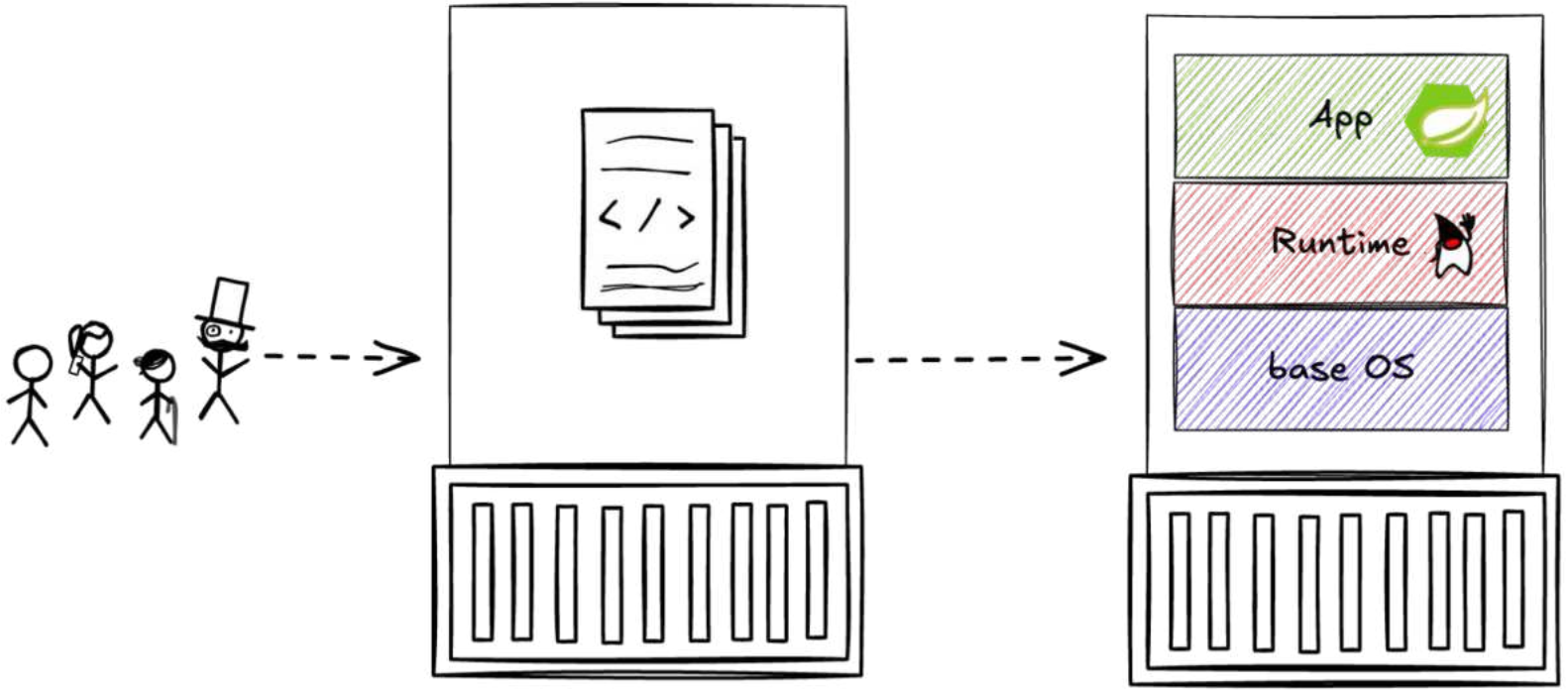


Integration principle



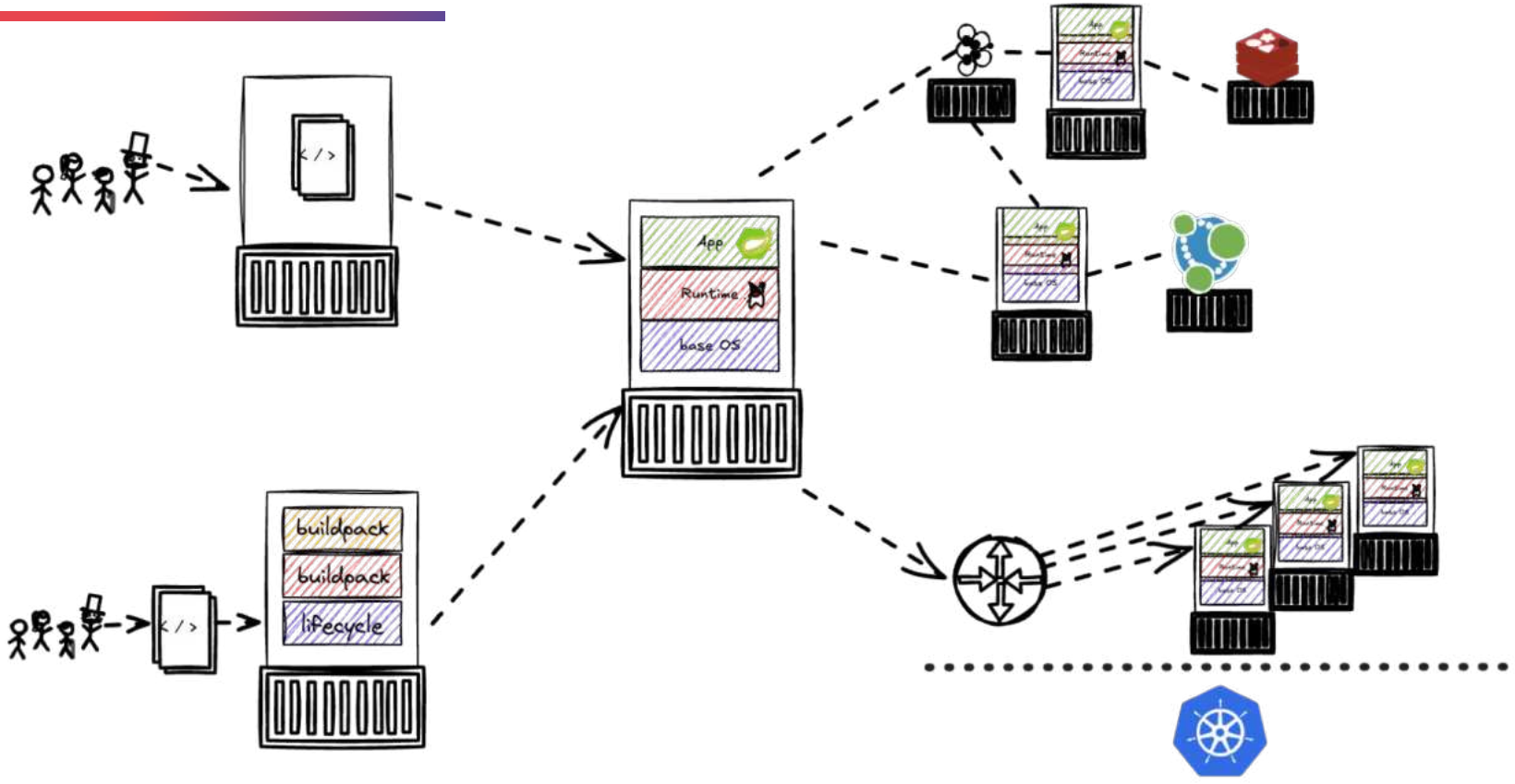
develop
(finally)

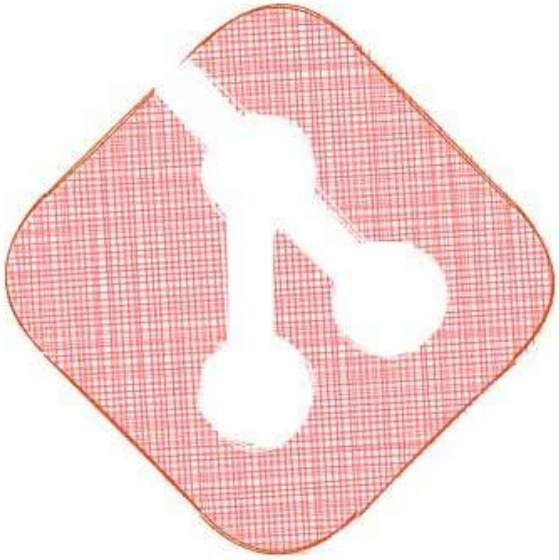
Container to DEVELOP



Summary

Sooo many integrations





Thanks for listening!

Eva

Matthias Häußler
Chief Technologist, Novatec
@maeddes