

Options Galore: From Source Code to Container Image

Matthias Haeussler, Novatec

@maeddes





Matthias Haeussler

Chief Technologist

Mobil: +49 175 222 5949

E-Mail: matthias.haeussler@novatec-gmbh.de

Twitter: @maeddes

Novatec Consulting GmbH

Bertha-Benz-Platz 1

D-70771 Leinfelden-Echterdingen

T. +49 711 22040-700

info@novatec-gmbh.de

www.novatec-gmbh.de

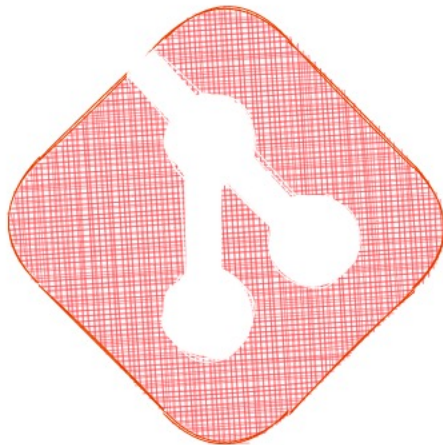




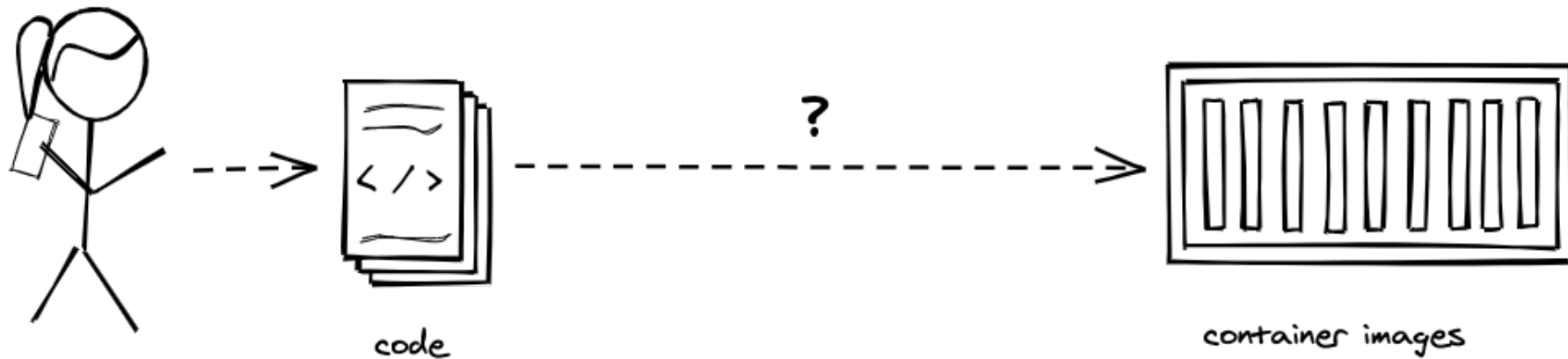
maeddes` experimental repo



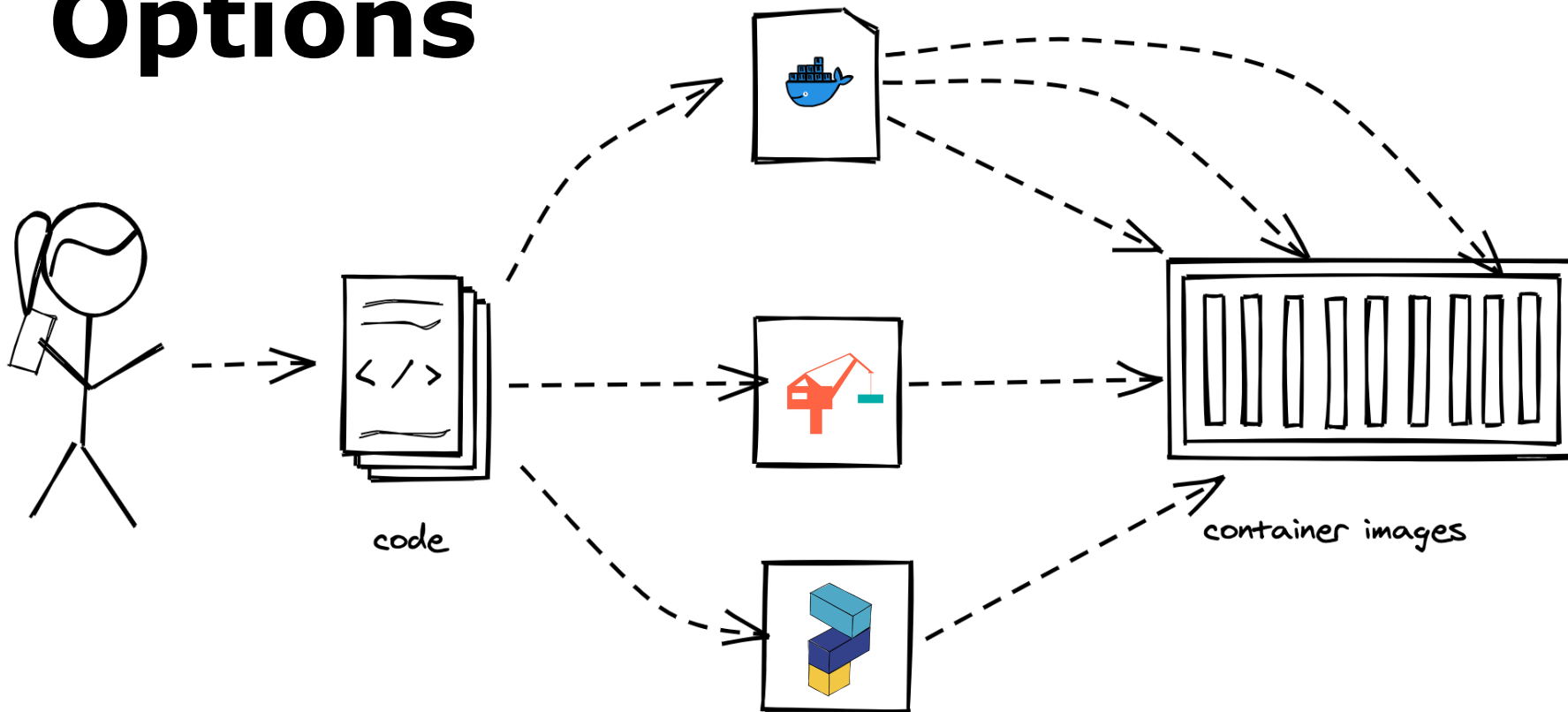
Scan the code for my
contact details



Situation



Options

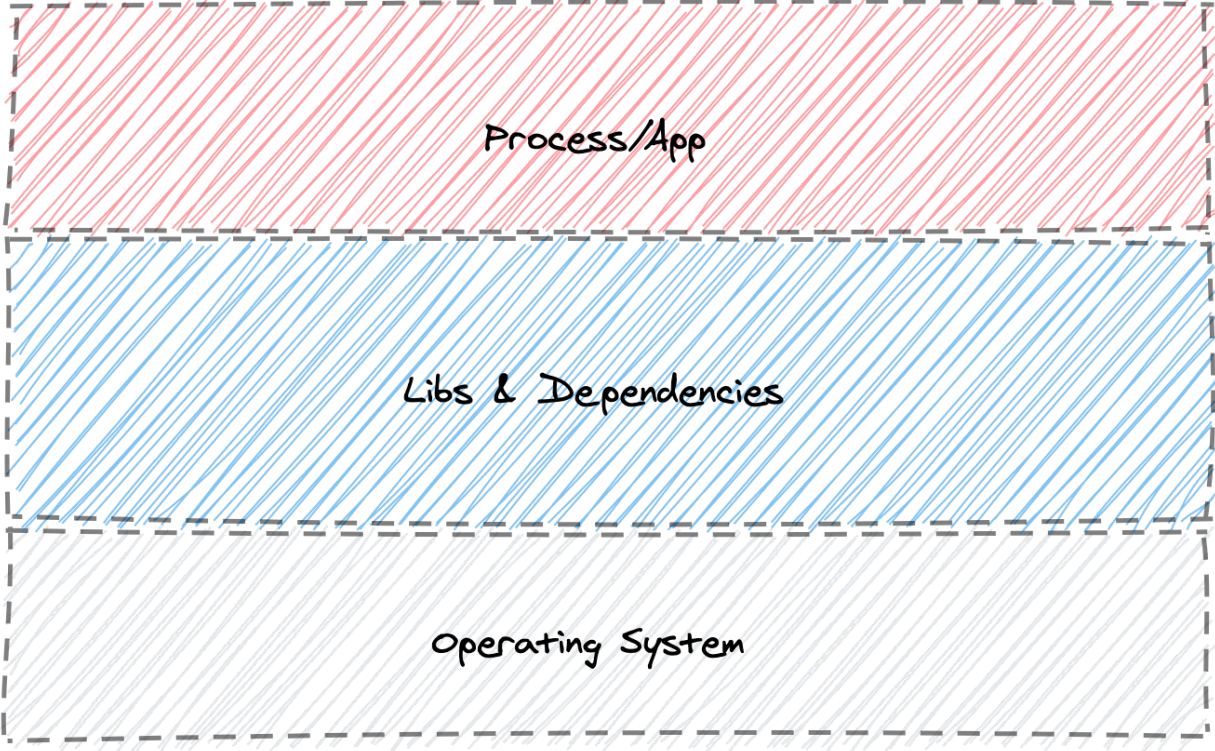


„What is the best Java base container image?“

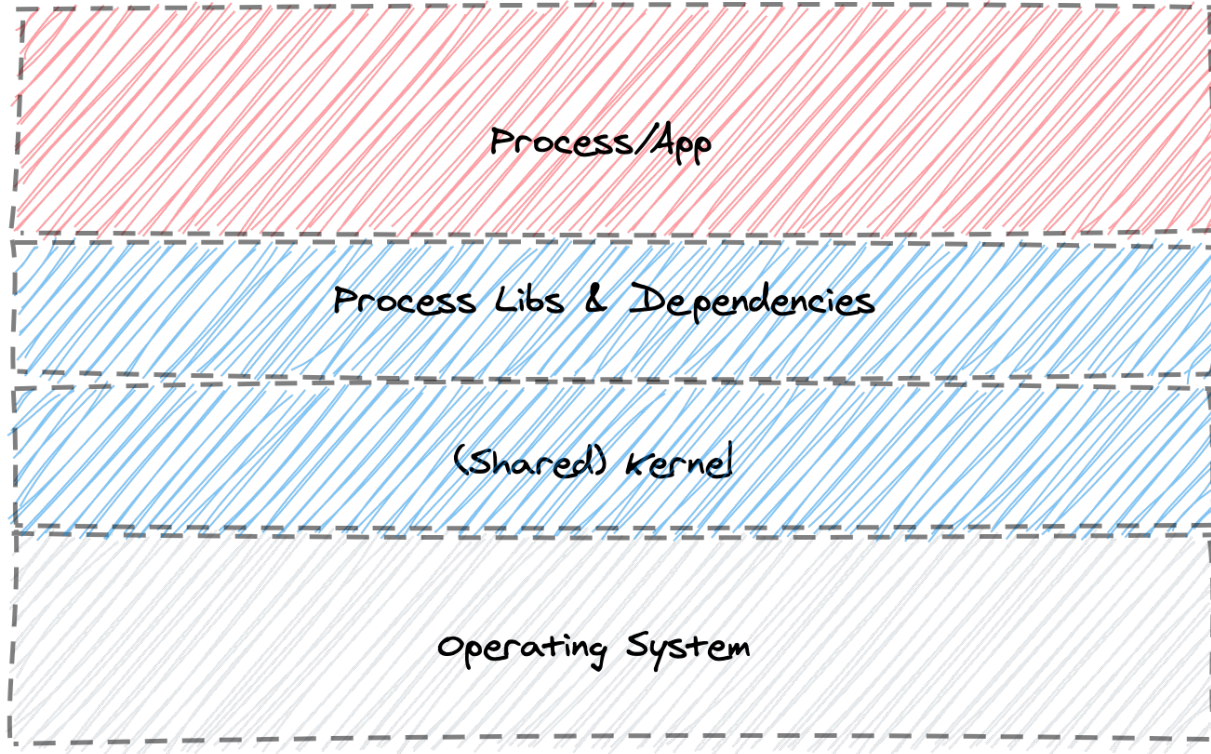
A developer, Novatec

**„What is the best
base container
image?“**

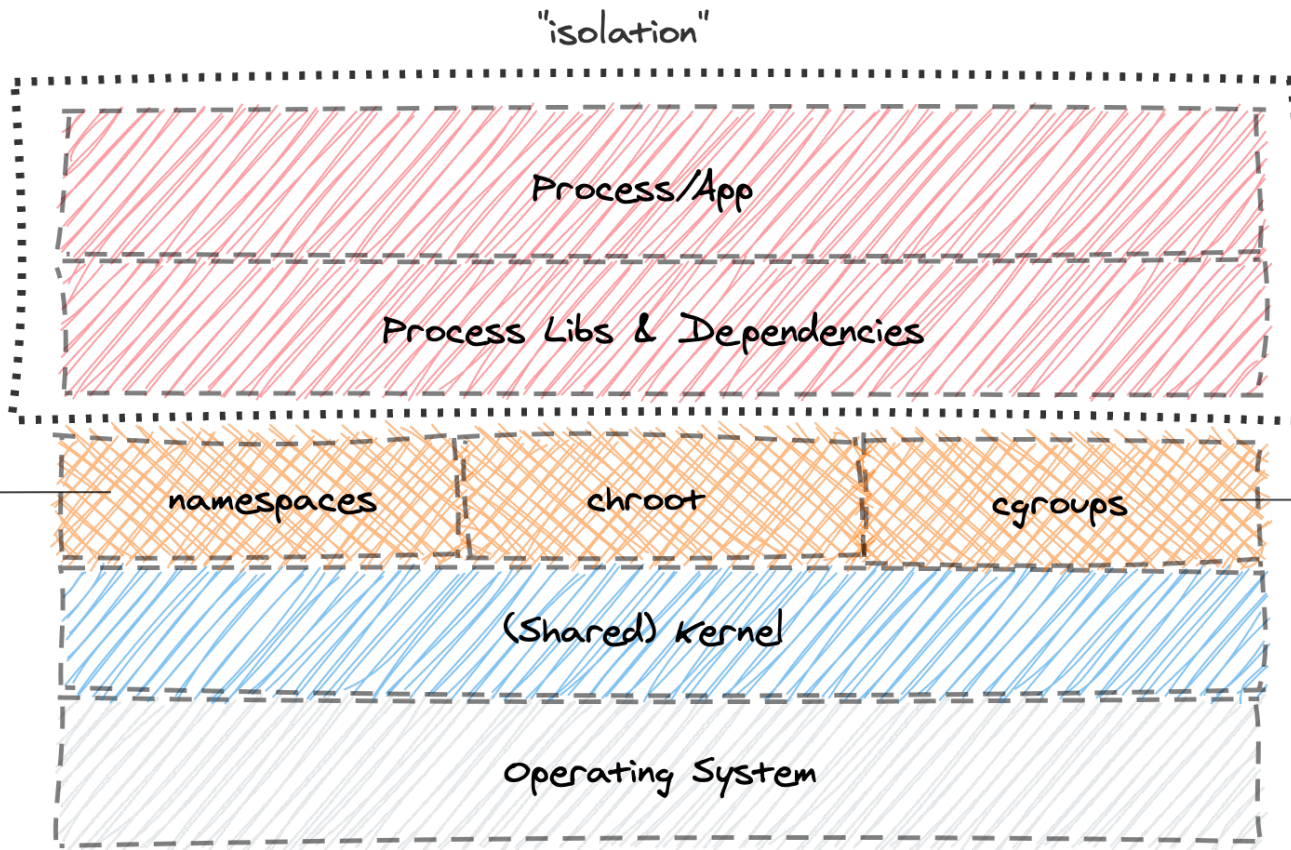
**„What is the best
base container
image?“**



"separation"



"what can be seen"



"what can be used"

"container"

Process/App

Process Libs & Dependencies

Container Daemon

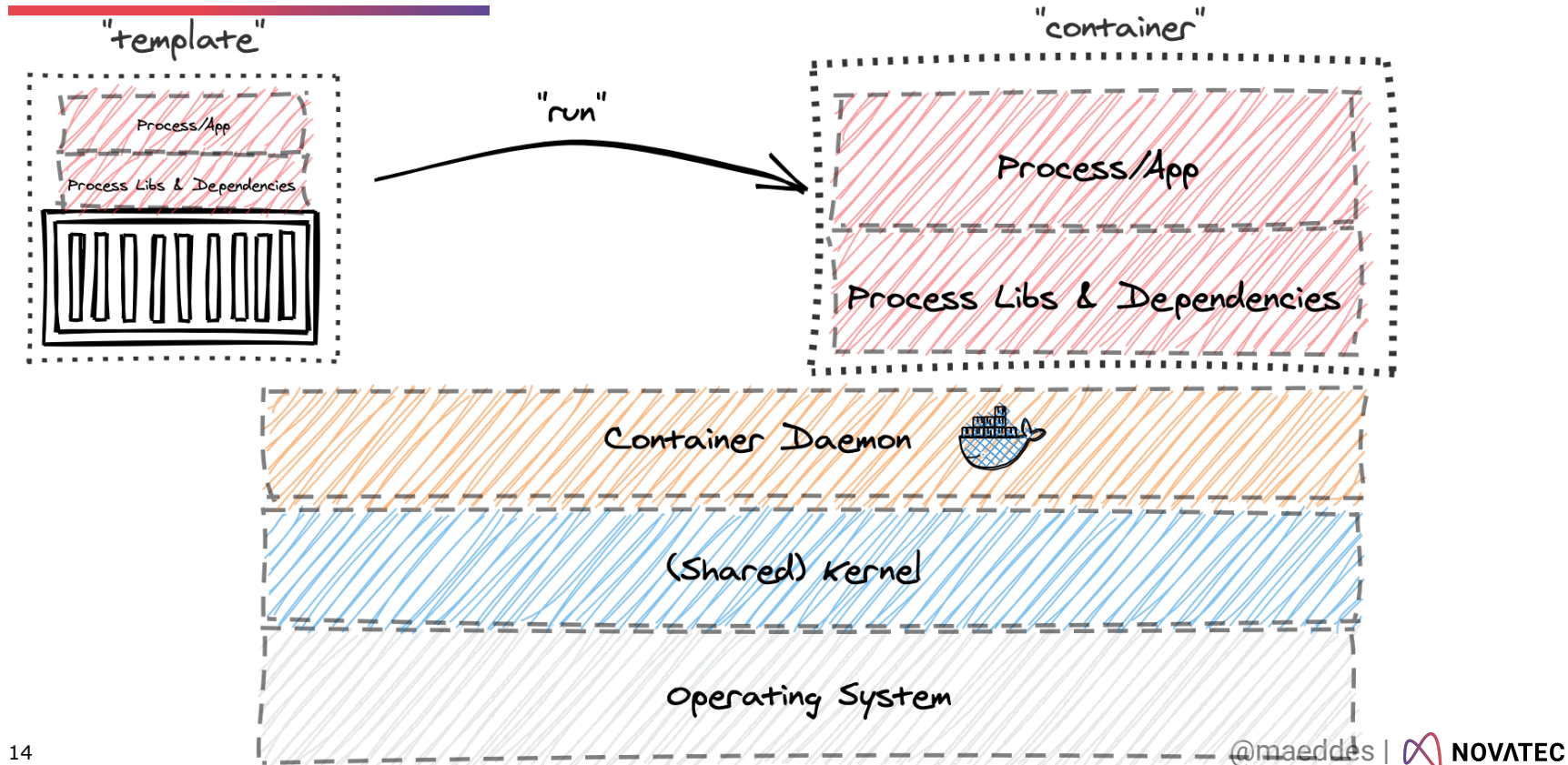


(Shared) kernel

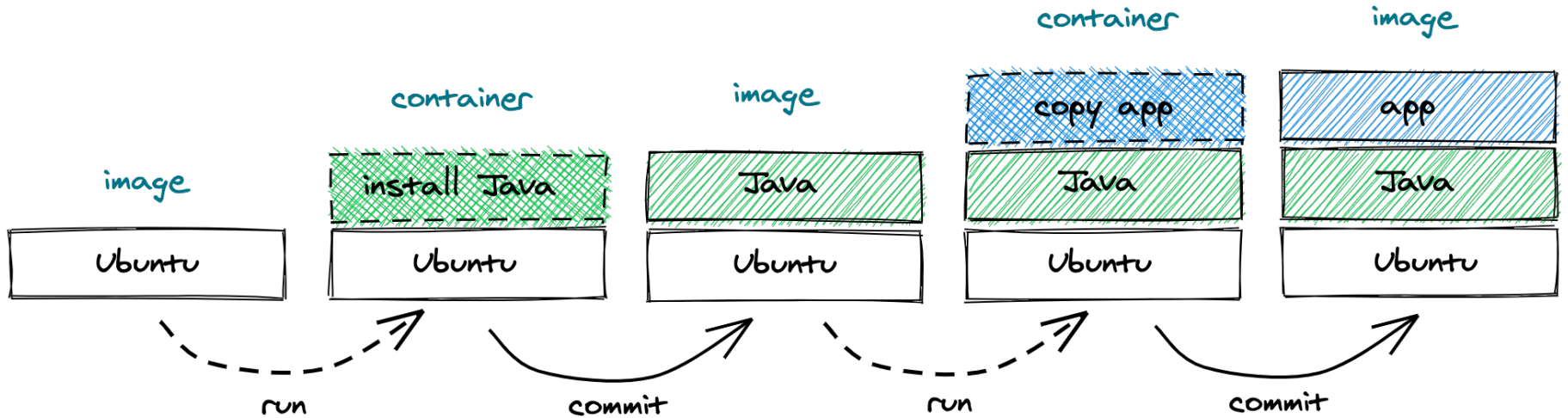
Operating System

**„What is the best
base container
image?“**

Container Images as Templates



Container, Images & Layers



Docker History

0.2.1 (2013-05-01)

- 'docker commit -run' bundles a layer with default runtime options: command, ports etc.
- Improve install process on Vagrant
- New Dockerfile operation: "maintainer"
- New Dockerfile operation: "expose"
- New Dockerfile operation: "cmd"
- Contrib script to build a Debian base layer
- 'docker -d -r': restart crashed containers at daemon startup
- Runtime: improve test coverage

Dockerfile


```
FROM ubuntu:20.10
```

```
RUN apt update && apt install openjdk-11-jre-headless -y
```

```
COPY target/simplecode-0.0.1-SNAPSHOT.jar /opt/app.jar
```

```
CMD ["java", "-jar", "/opt/app.jar"]
```

direction
of
processing




Alternative Dockerfile

```
FROM adoptopenjdk:11-jre-hotspot
```

```
COPY target/simplecode-0.0.1-SNAPSHOT.jar /opt/app.jar
```

```
CMD ["java", "-jar", "/opt/app.jar"]
```

direction
of
processing




Alternative Dockerfile

```
FROM openjdk:ill-jre-slim
```

```
COPY target/simplecode-0.0.1-SNAPSHOT.jar /opt/app.jar
```

```
CMD ["java", "-jar", "/opt/app.jar"]
```

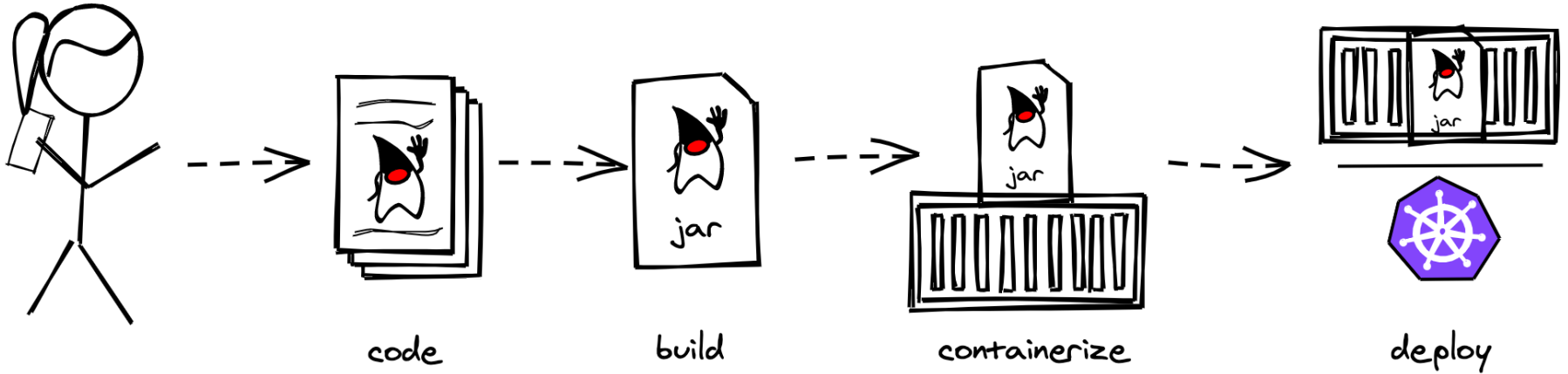
direction
of
processing



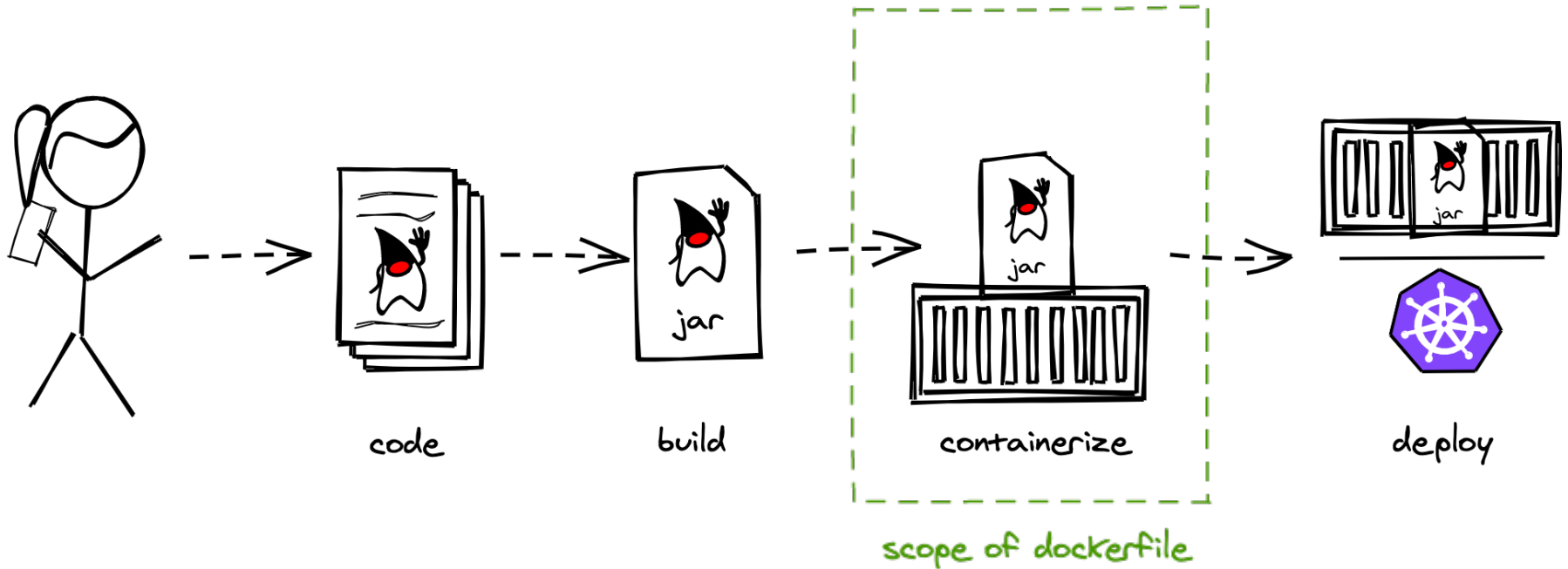
**„What is the best
Java base container
image?“**

**„How do you use
the base container
image?“**

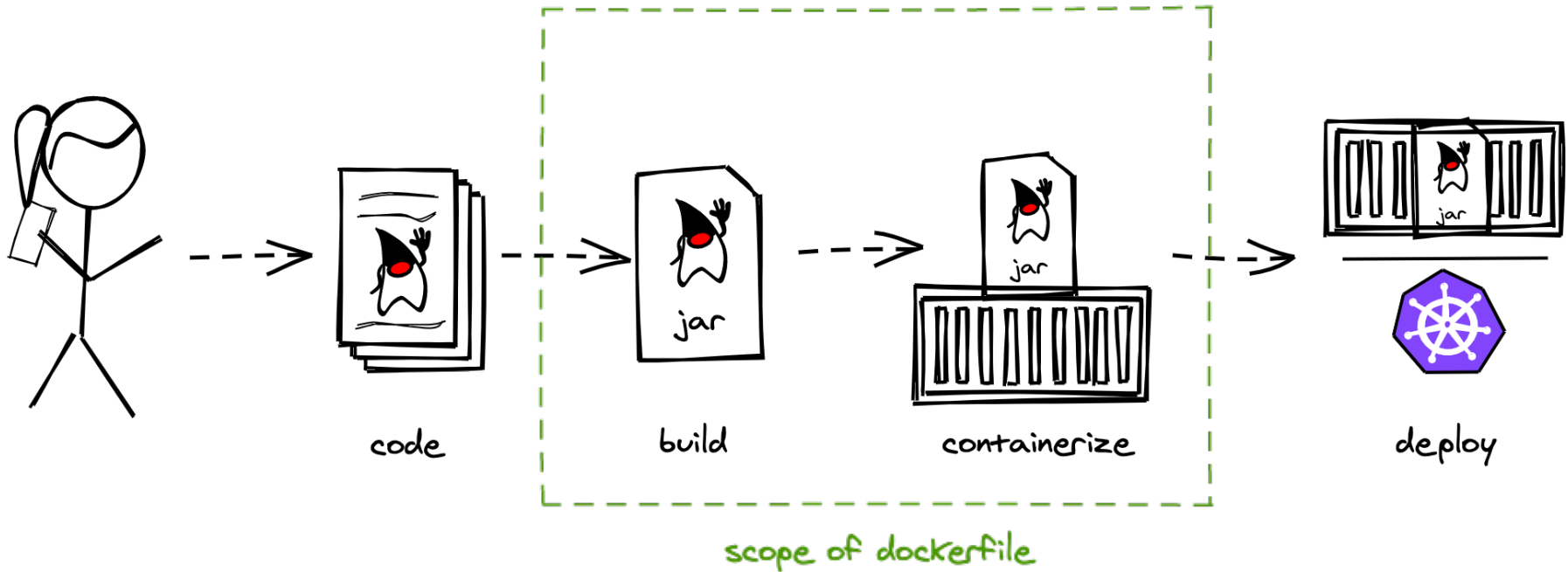
Code, Build, Container, Deploy



Code, Build, Container, Deploy

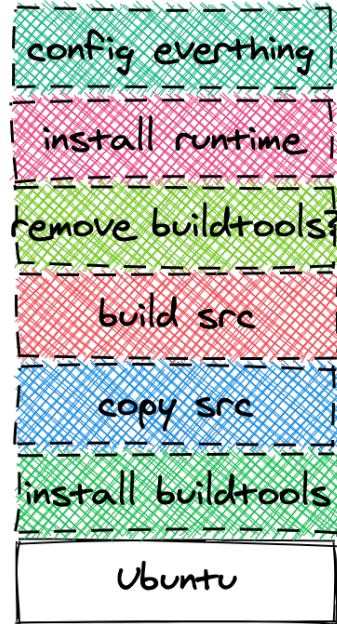
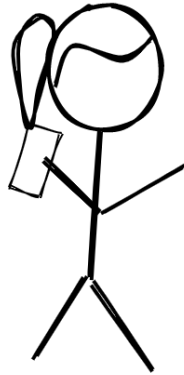


Code, Build, Container, Deploy



How?

Dockerfile???



Multi-Stage

Docker Engine 17.05 release notes

17.05.0-ce

2017-05-04

Builder

- Add multi-stage build support #31257 #32063
- Allow using build-time args (ARG) in FROM #31352
- Add an option for specifying build target #32496
- Accept `-f -` to read Dockerfile from `stdin` , but use local context for building #31236

Multi-Stage Dockerfile

```
FROM maven:3.6.3-jdk-11 AS builder
```

```
RUN mkdir -p /opt/app/src
```

```
COPY src /opt/app/src
```

```
COPY pom.xml /opt/app
```

```
RUN mvn -f /opt/app/pom.xml clean package
```

```
FROM adoptopenjdk:11-jre-hotspot
```

```
COPY --from=builder /opt/app/target/simplecode-0.0.1-SNAPSHOT.jar /opt/app.jar
```

```
CMD ["java", "-jar", "/opt/app.jar"]
```

One
Dockerfile

Buildkit

18.06.0-ce

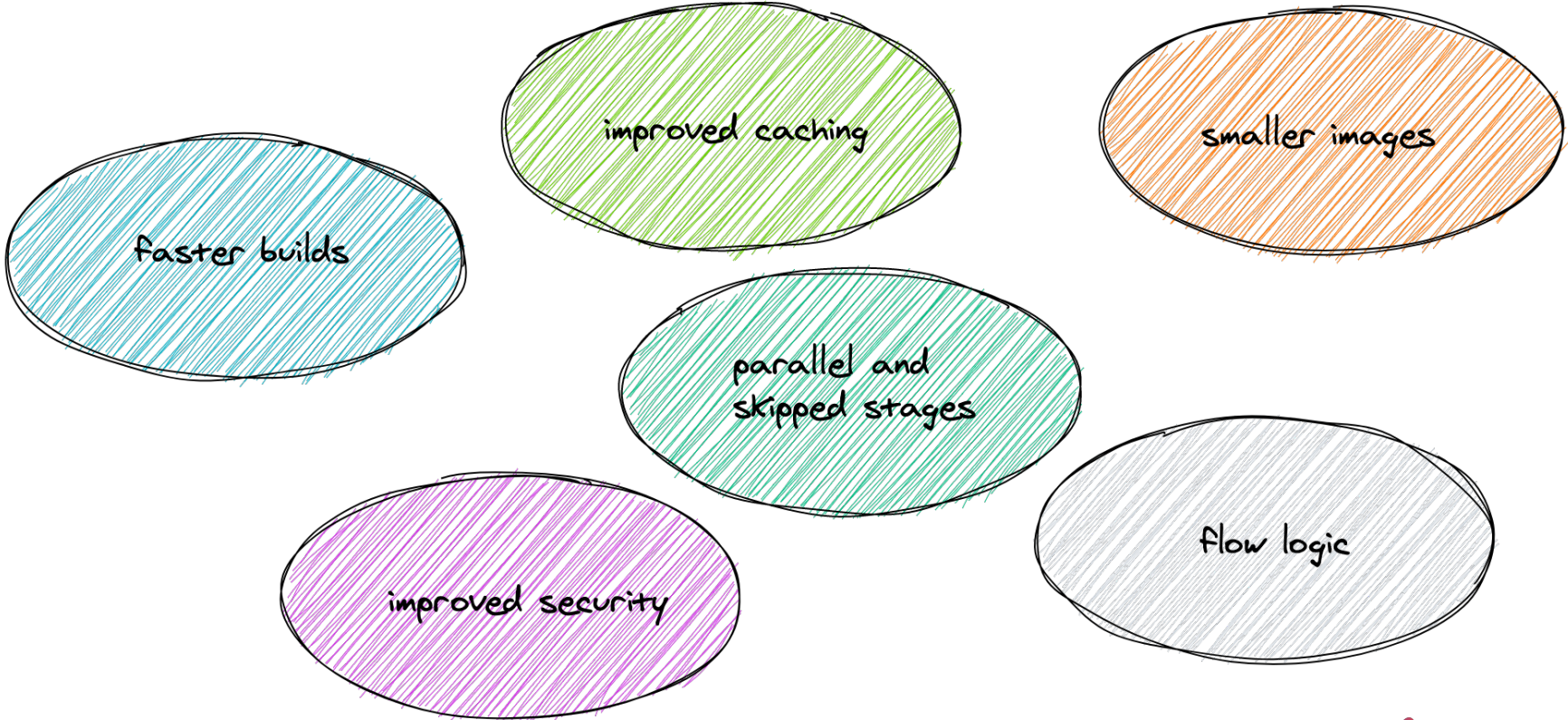

2018-07-18

Important notes about this release

- Docker 18.06 CE will be the last release with a 4-month maintenance lifecycle. The planned Docker 18.09 CE release will be supported for 7 months with Docker 19.03 CE being the next release in line. More details about the release process can be found [here](#).

Builder

- Builder: fix layer leak on multi-stage wildcard copy. [moby/moby#37178](#)
- Fix parsing of invalid environment variable substitution. [moby/moby#37134](#)
- Builder: use the arch info from base image. [moby/moby#36816](#) [moby/moby#37197](#)
- New experimental builder backend based on BuildKit. To enable, run daemon in experimental mode and set `DOCKER_BUILDKIT=1` environment variable on the docker CLI. [moby/moby#37151](#) [docker/cli#1111](#)
- Fix handling uppercase targets names in multi-stage builds. [moby/moby#36960](#)



faster builds

improved caching

smaller images

parallel and
skipped stages

improved security

flow logic

FROM debian AS **Java-Base**
RUN install JRE

FROM ubuntu AS **Gradle-Base**
RUN install JDK
RUN install gradle

FROM **Gradle-Base** AS **builder**
COPY src
RUN gradle

FROM **Java-Base**
COPY jar from **builder**
CMD java -jar app.jar

"parallel and sequential processing"

Experimental – Mount Cache

```
# syntax=docker/dockerfile:experimental
```

```
FROM maven:3.6.3-jdk-11 AS build
```

```
RUN mkdir -p /opt/app/src
```

```
COPY src /opt/app/src
```

```
COPY pom.xml /opt/app
```

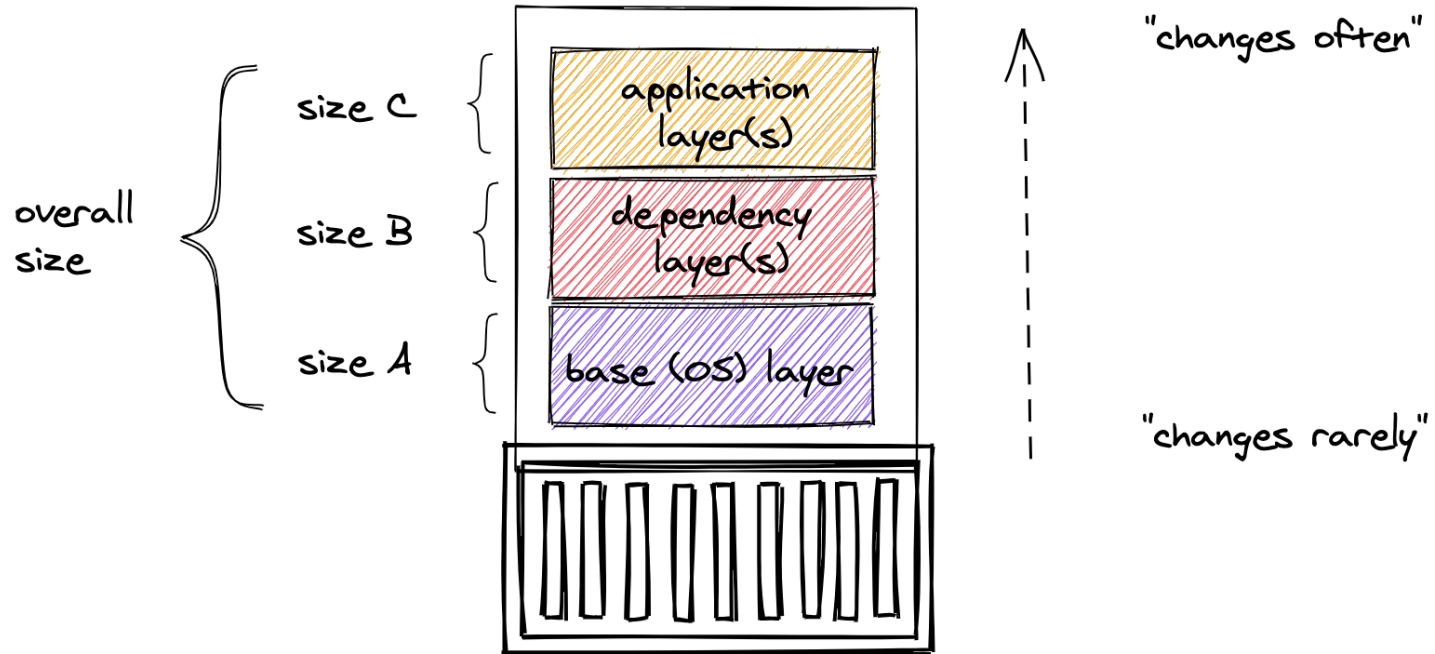
```
RUN --mount=type=cache,target=/root/.m2 mvn -f /opt/app/pom.xml clean package
```

**„What makes a
(Java) container
image *good*?“**

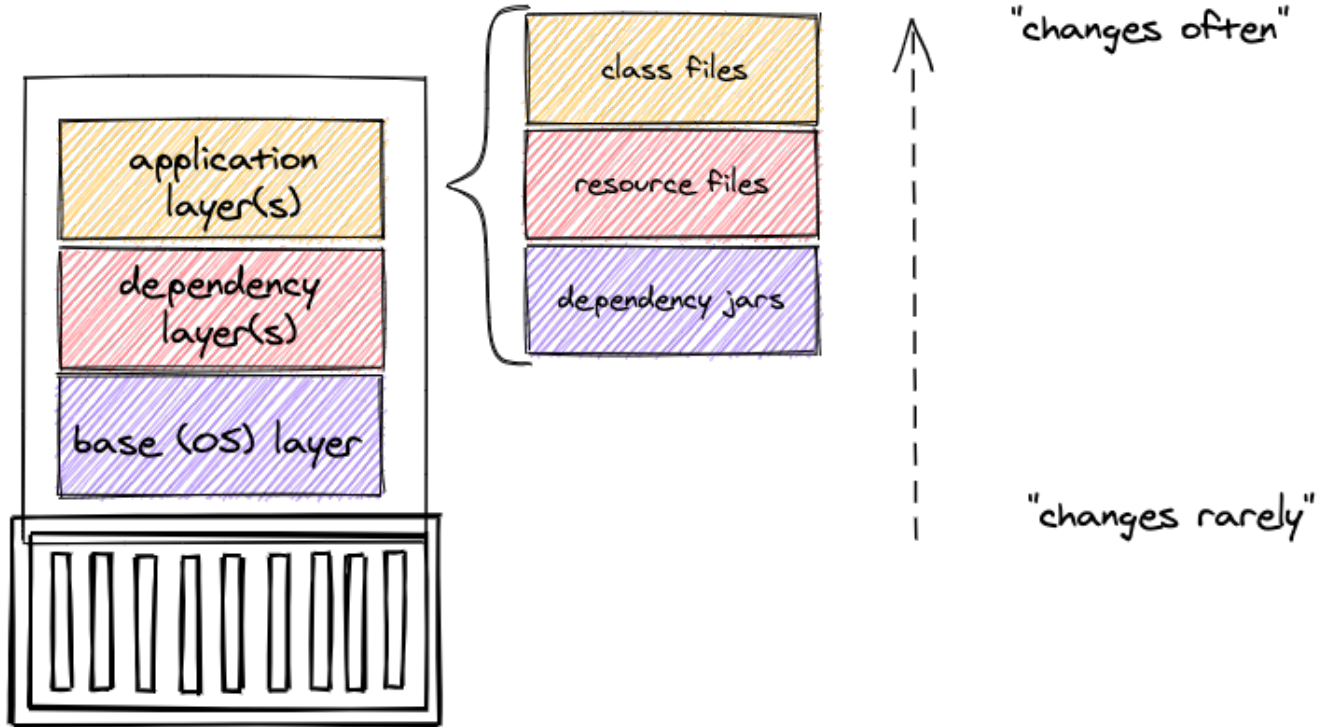
Criteria

- **Speed**
- **Size**
- **Structure**
- **Standardization**

Considerations



Considerations for Java



Using layered jars

```
FROM adoptopenjdk:11-jre-hotspot as builder
```

```
WORKDIR application
```

```
COPY --from=maven /opt/app/target/simplecode-0.0.1-SNAPSHOT.jar application.jar
```

```
RUN java -Djarmode=layertools -jar application.jar extract
```

```
FROM adoptopenjdk:11-jre-hotspot
```

```
WORKDIR application
```

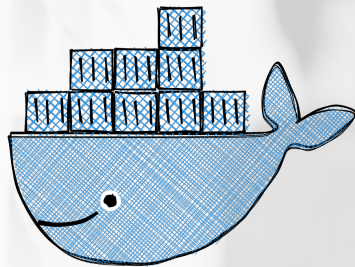
```
COPY --from=builder application/dependencies/ ./
```

```
COPY --from=builder application/spring-boot-loader/ ./
```

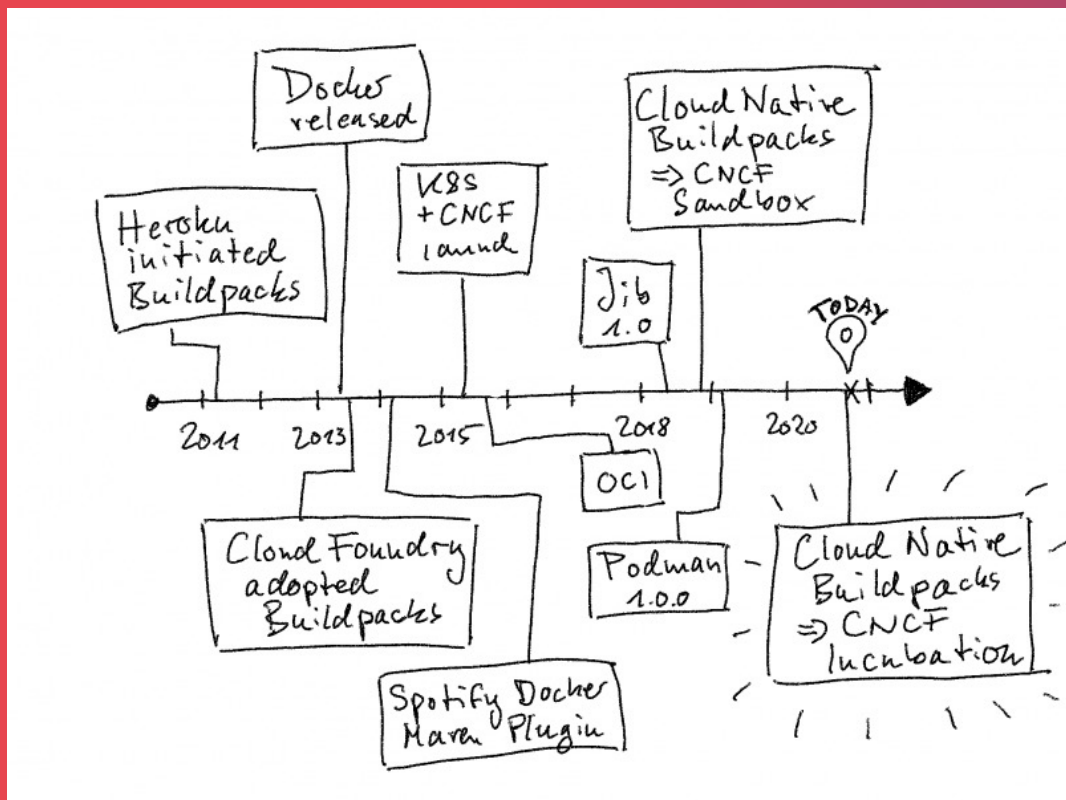
```
COPY --from=builder application/snapshot-dependencies/ ./
```

```
COPY --from=builder application/application/ ./
```

```
ENTRYPOINT ["java", "org.springframework.boot.loader.JarLauncher"]
```

& more ...?

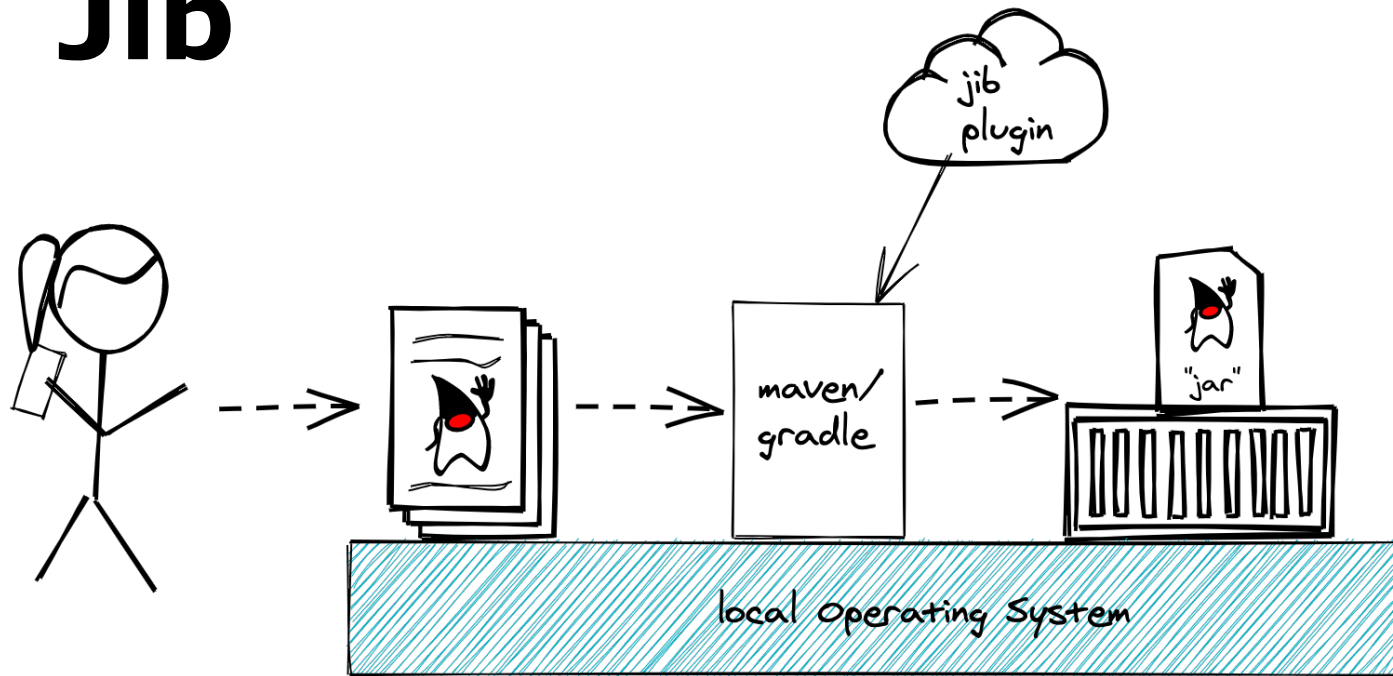




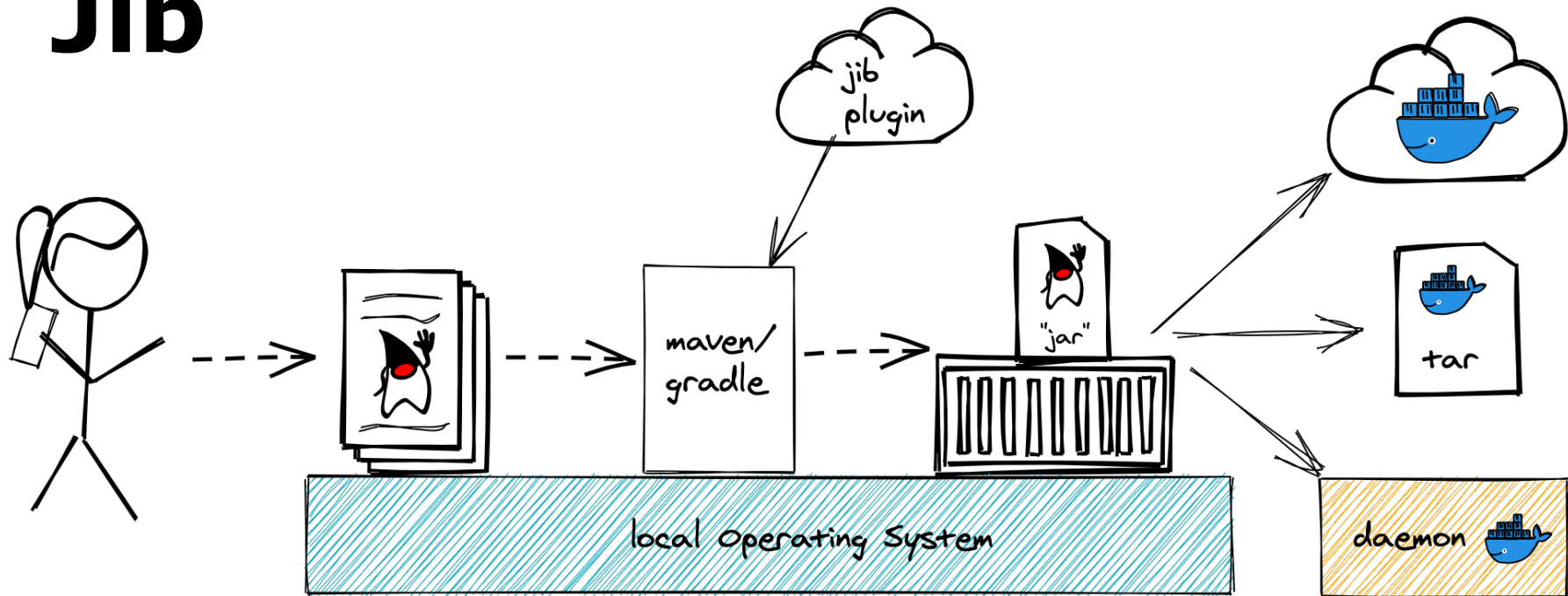
Jib

Containerize your Java application.

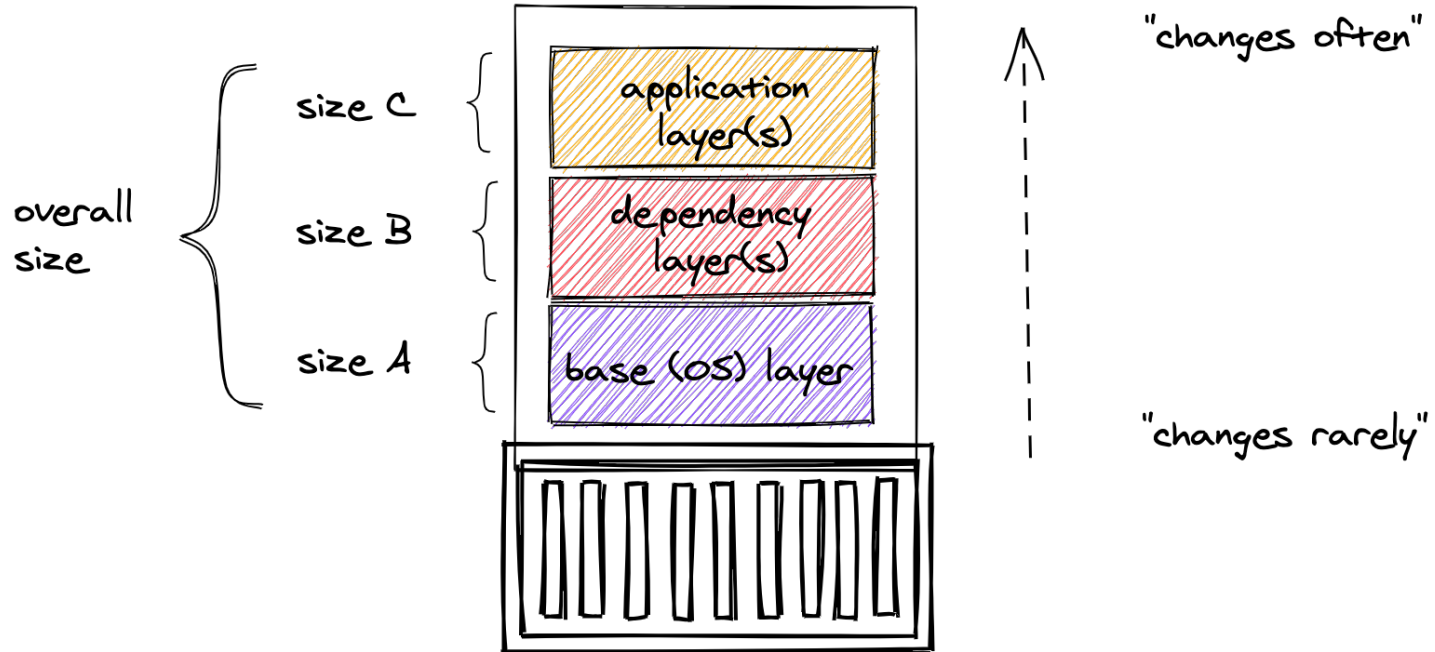
Jib



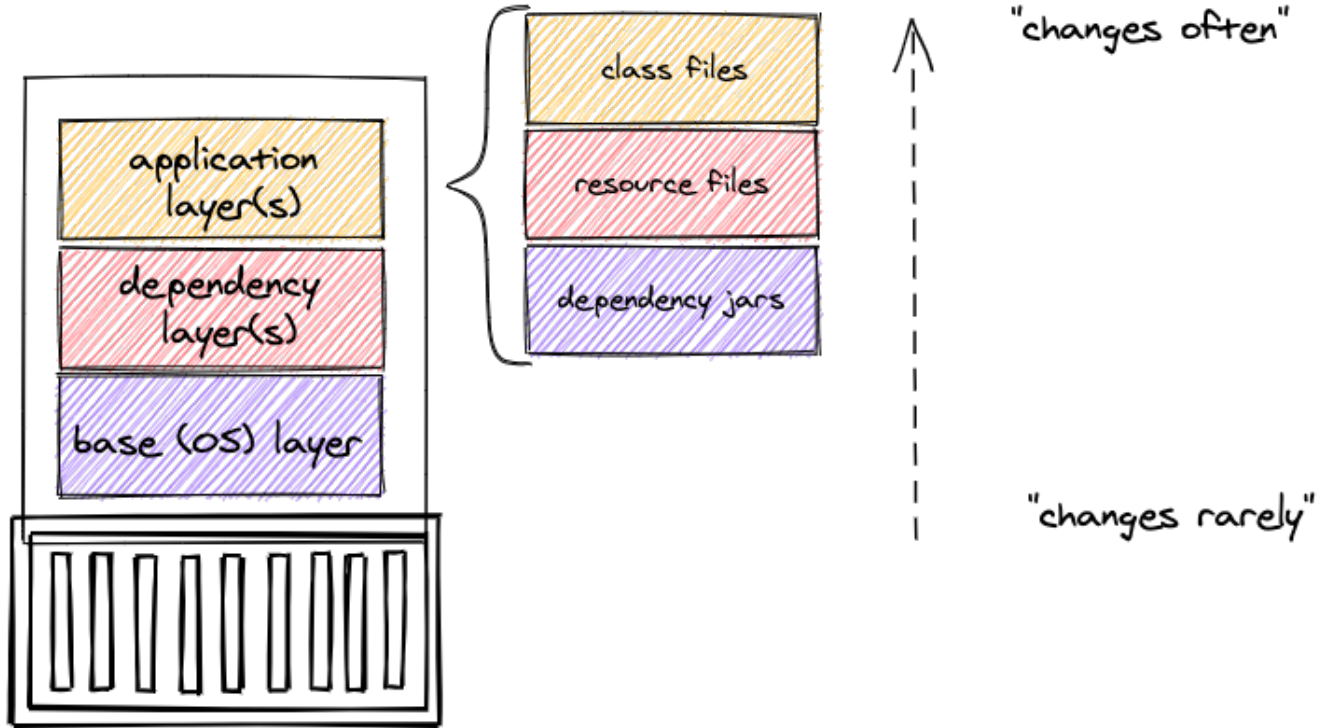
Jib



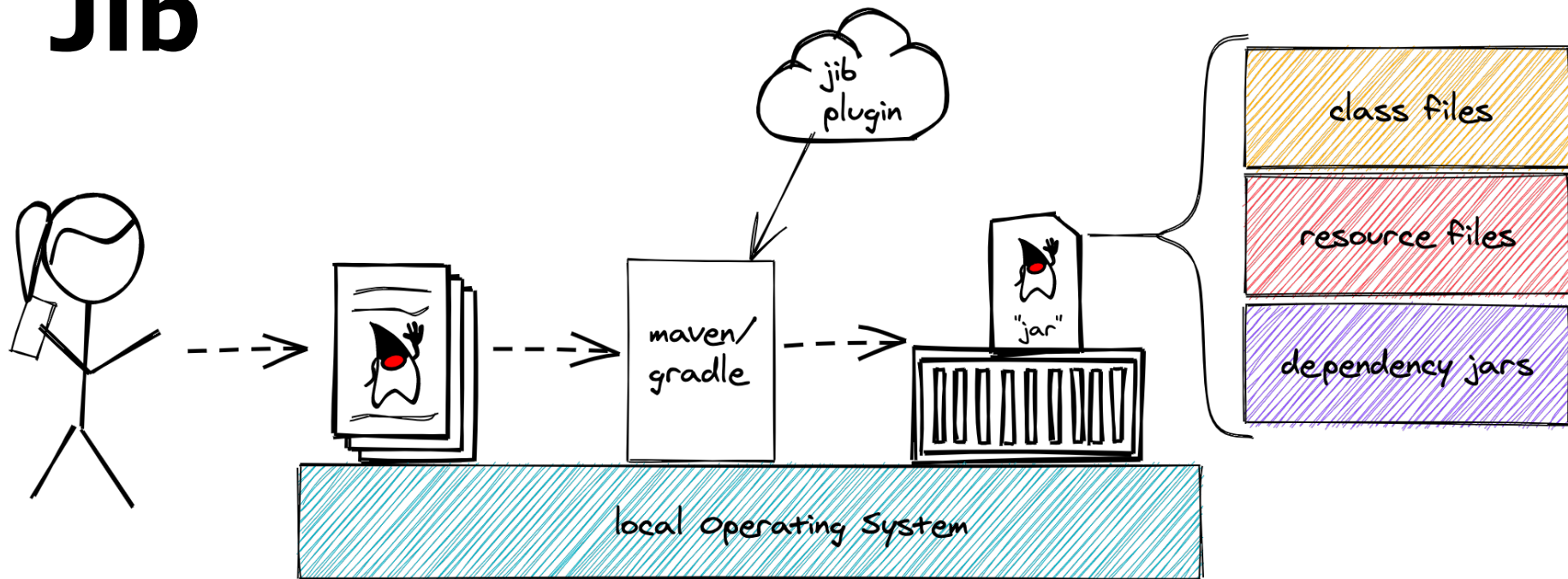
Considerations - Recap



Continued strategy



Jib

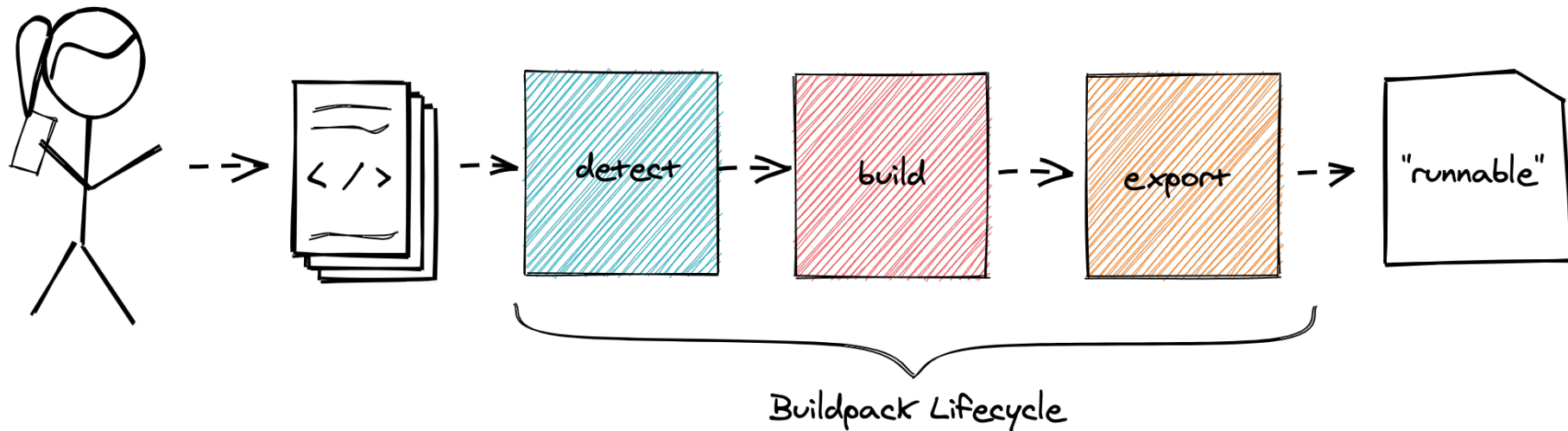




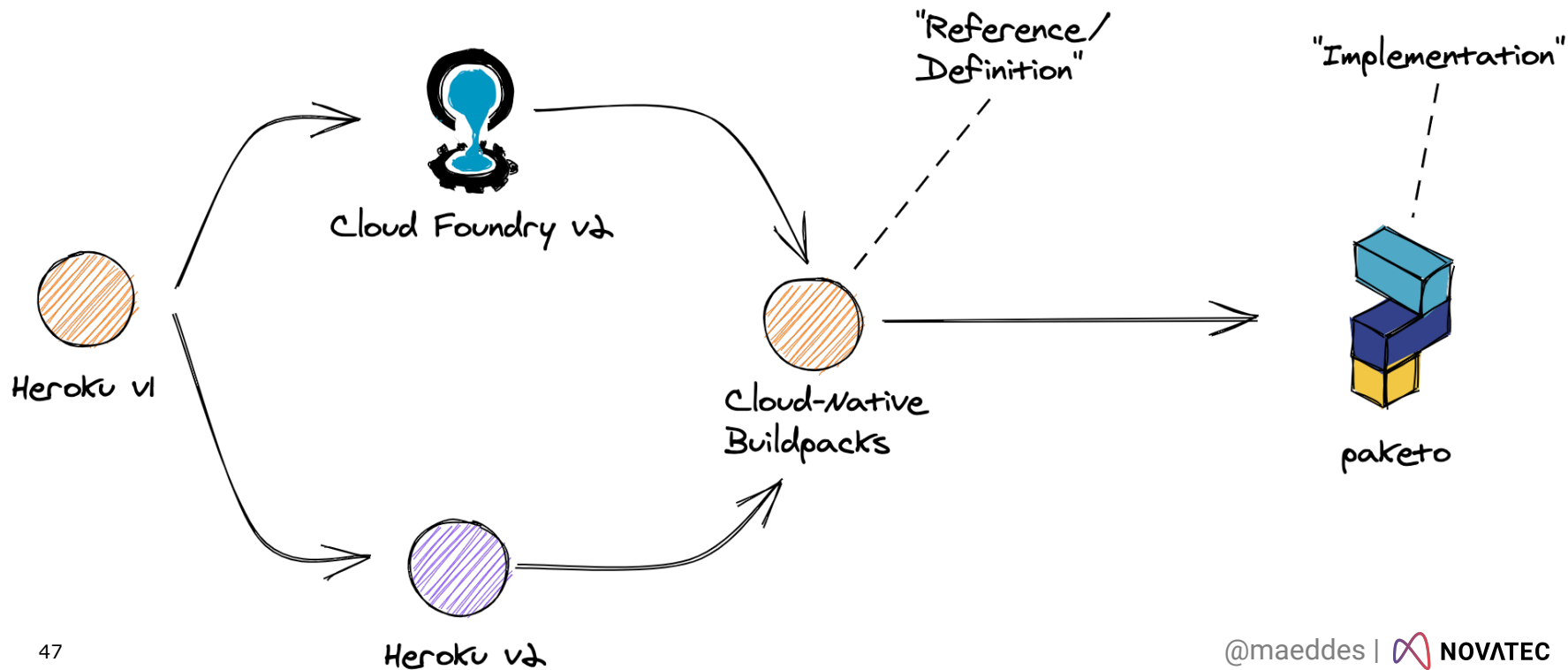
Cloud-Native Buildpacks

buildpacks.io

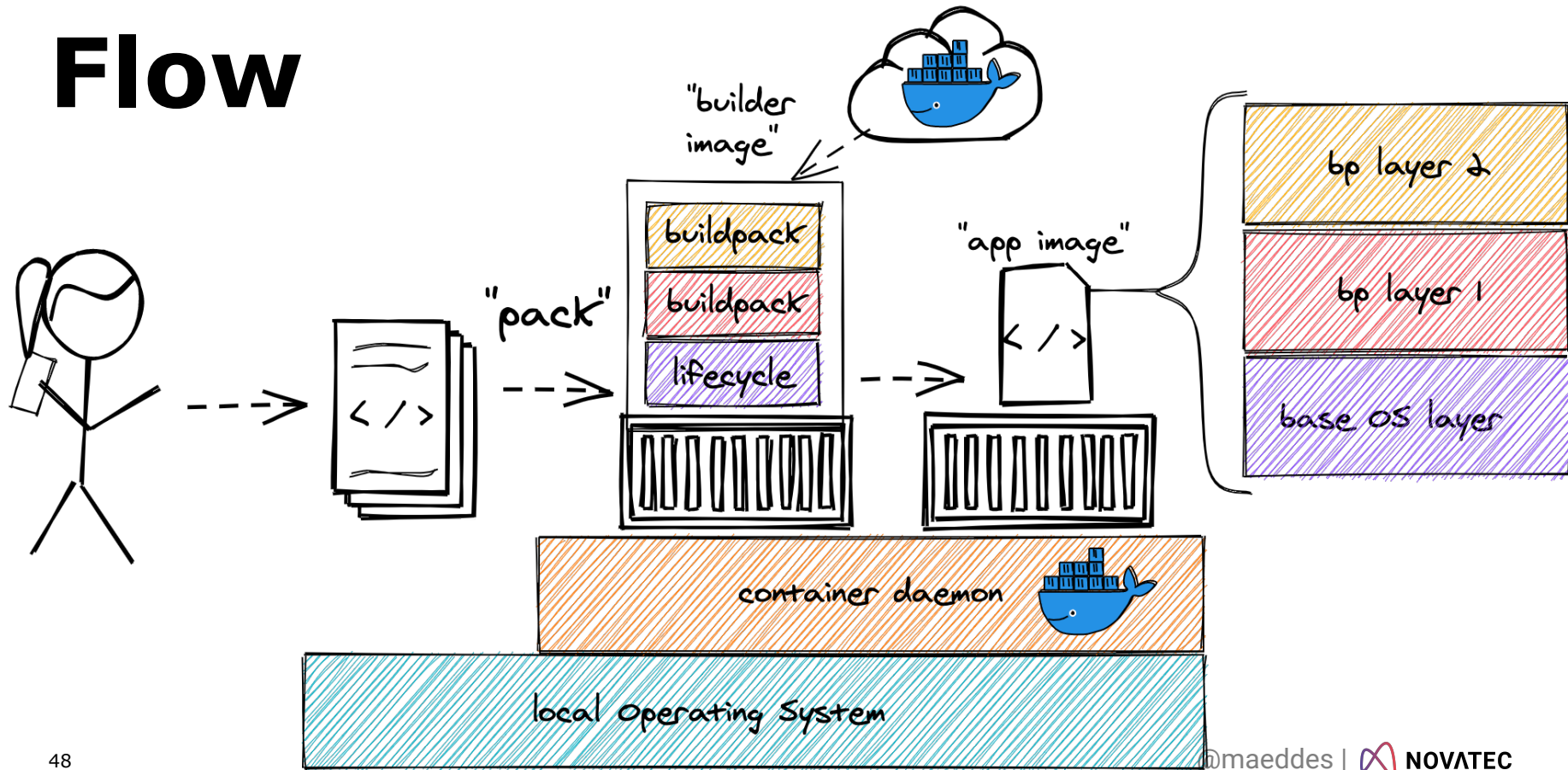
Buildpacks idea



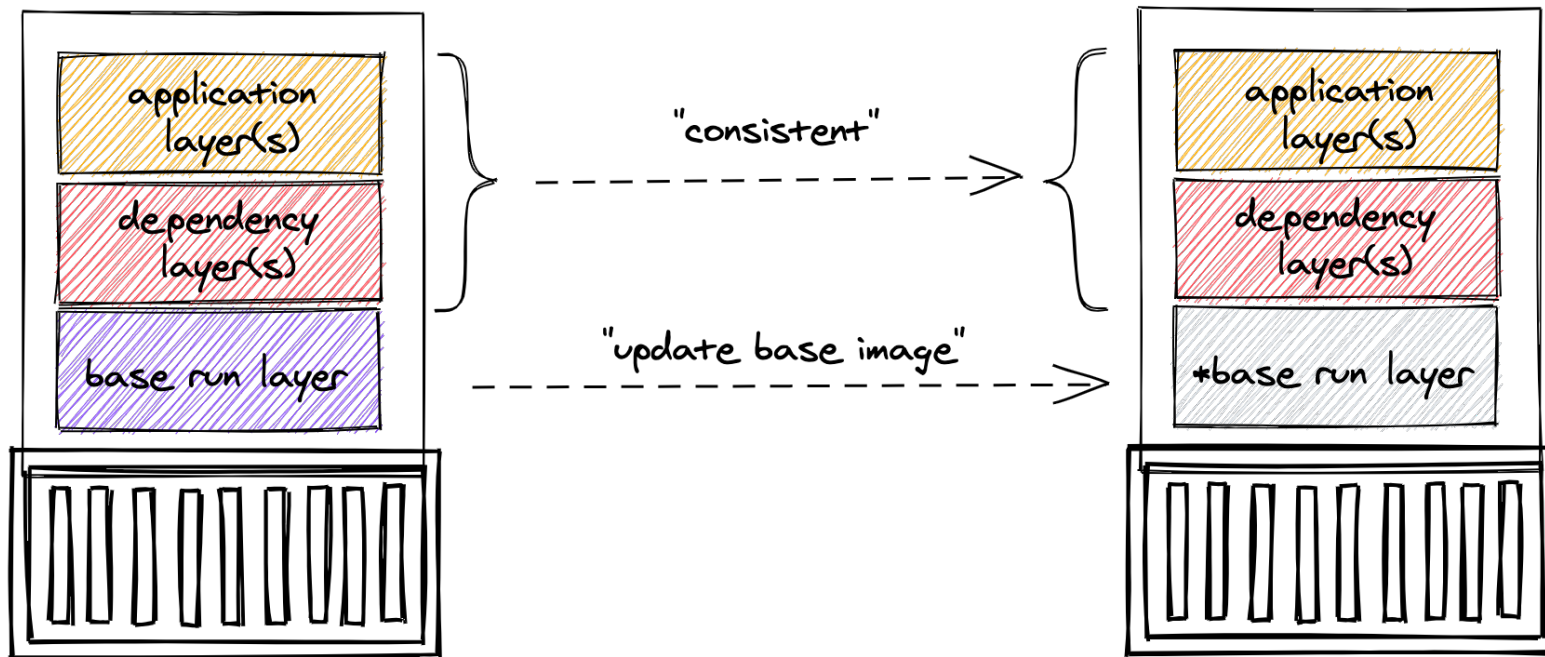
History

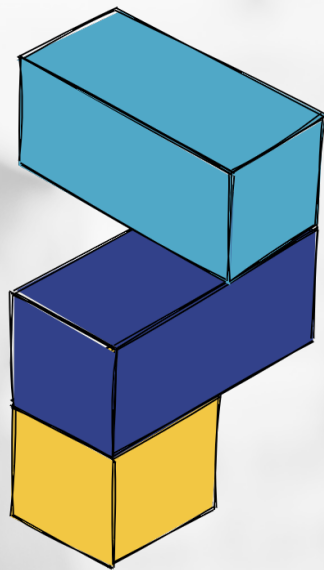


Flow



Rebase



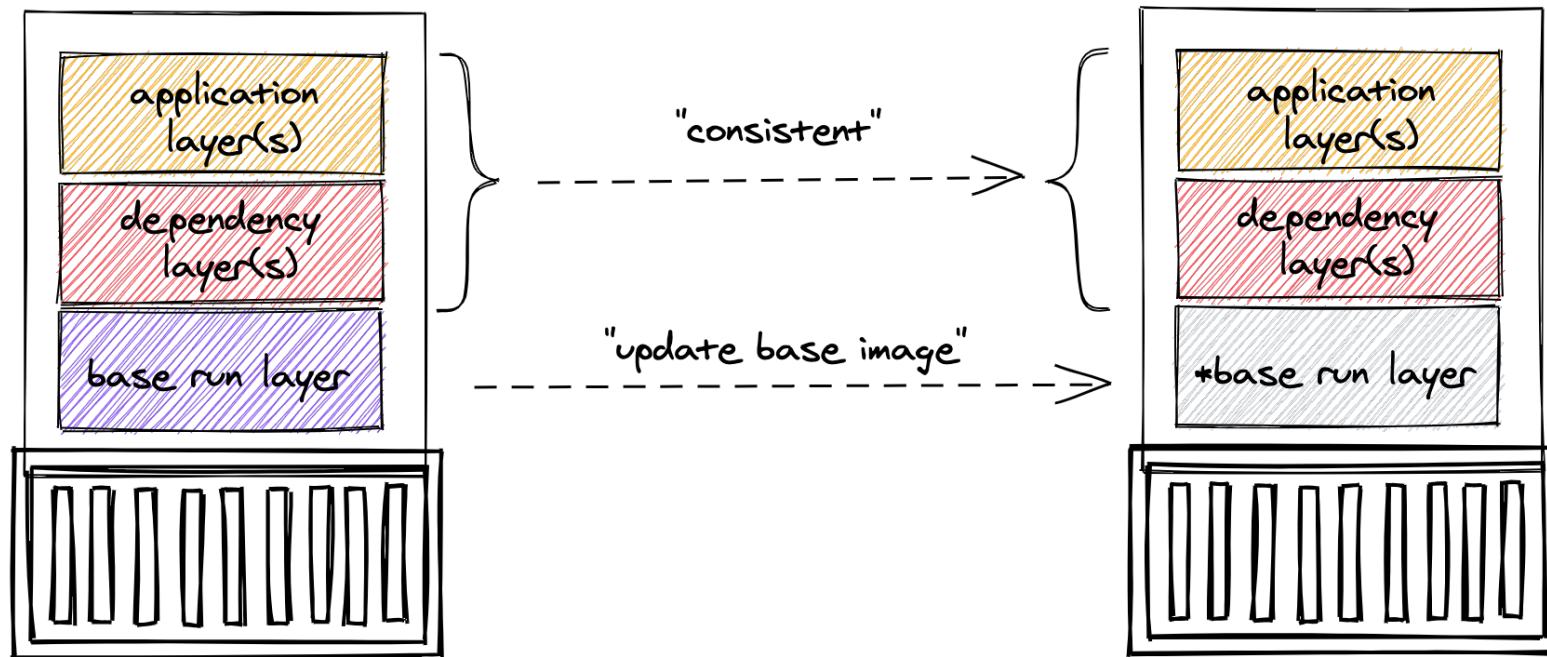


paketo.io

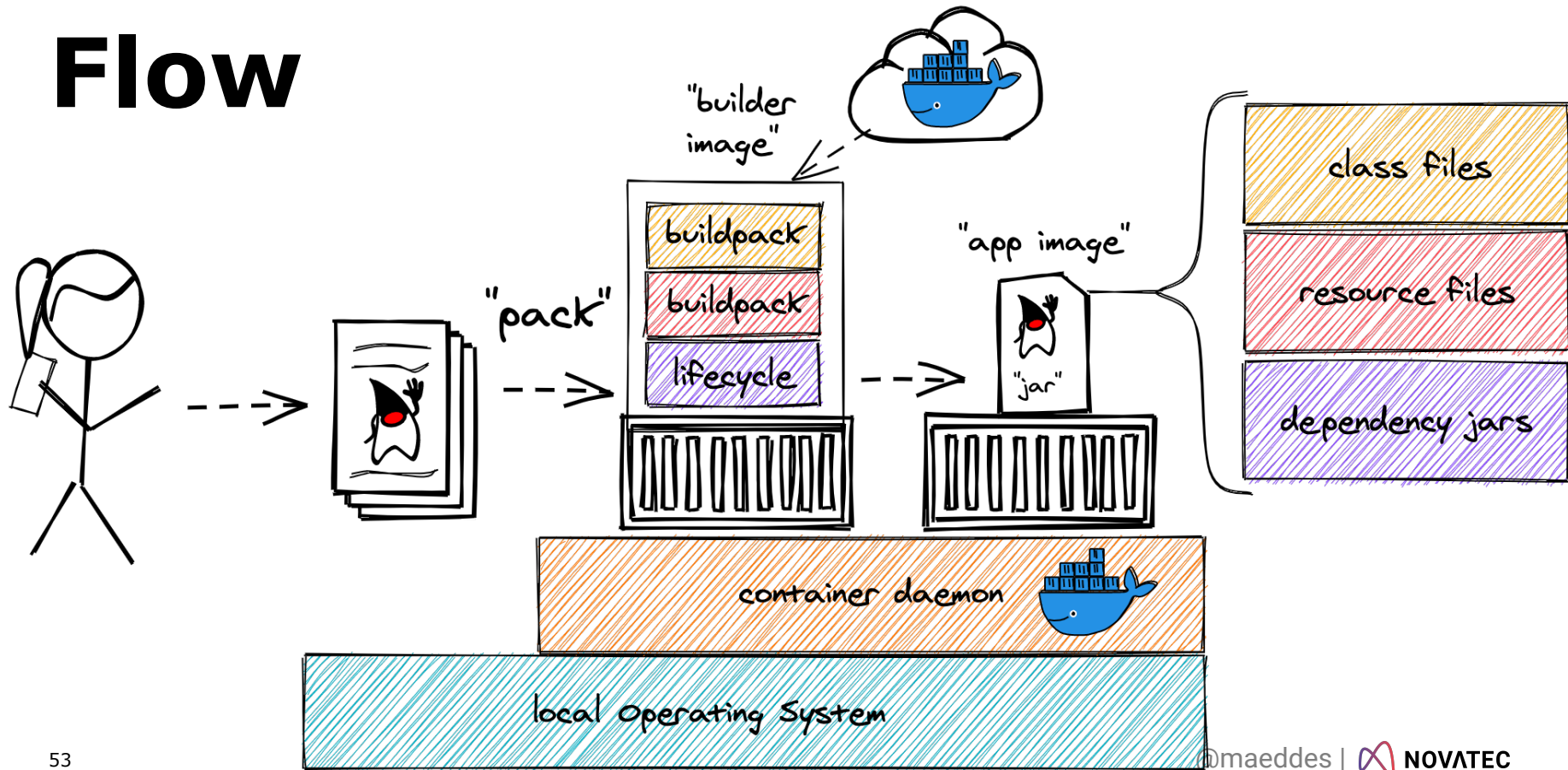
Multi Runtime/Language Support

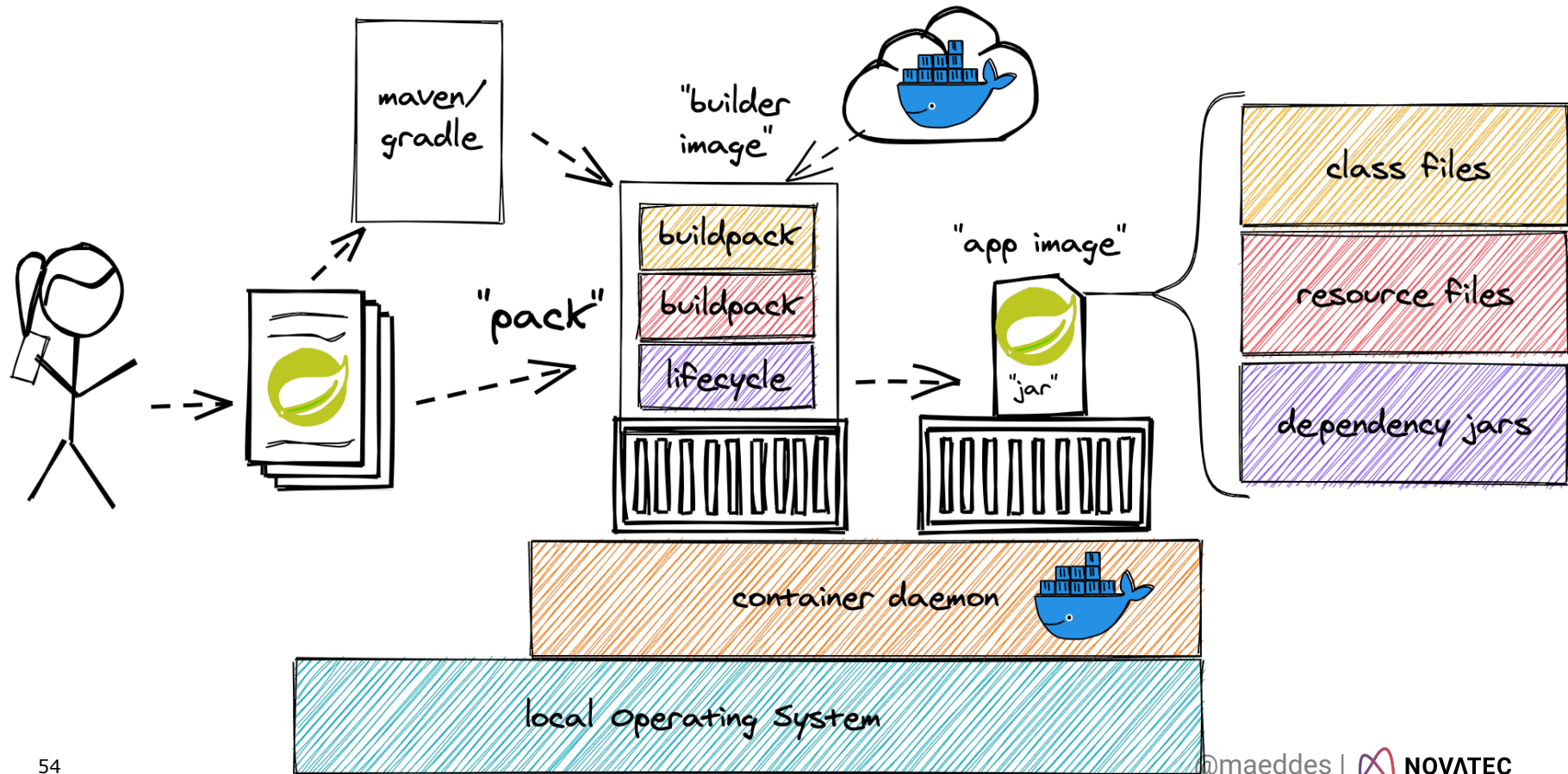


Rebase



Flow





**„What is the best
Java base container
image?“**

**„Who owns the
image?“**

**„Who owns the
build process?“**

Demo



Matthias Haeussler

Chief Technologist

Mobil: +49 175 222 5949

E-Mail: matthias.haeussler@novatec-gmbh.de

Twitter: @maeddes

Novatec Consulting GmbH

Bertha-Benz-Platz 1

D-70771 Leinfelden-Echterdingen

T. +49 711 22040-700

info@novatec-gmbh.de

www.novatec-gmbh.de

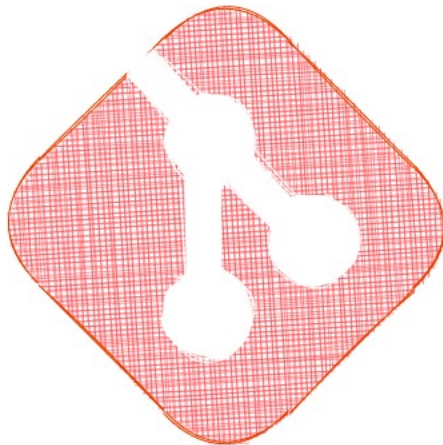




maeddes` experimental repo



Scan the code for my
contact details



Sources

- <https://www.excalidraw.com>
- [1] <https://docs.docker.com/engine/release-notes/prior-releases/>
- [2] <https://blog.codecentric.de/en/2020/11/buildpacks-spring-boot/>
- [3] <https://paketo.io/>
- <https://buildpacks.io/features/>
- <https://github.com/GoogleContainerTools/jib>
- <https://www.baeldung.com/docker-layers-spring-boot>
- <https://spring.io/blog/2020/01/27/creating-docker-images-with-spring-boot-2-3-0-m1>
- <https://github.com/openshift/source-to-image>