Creating the Sample Application

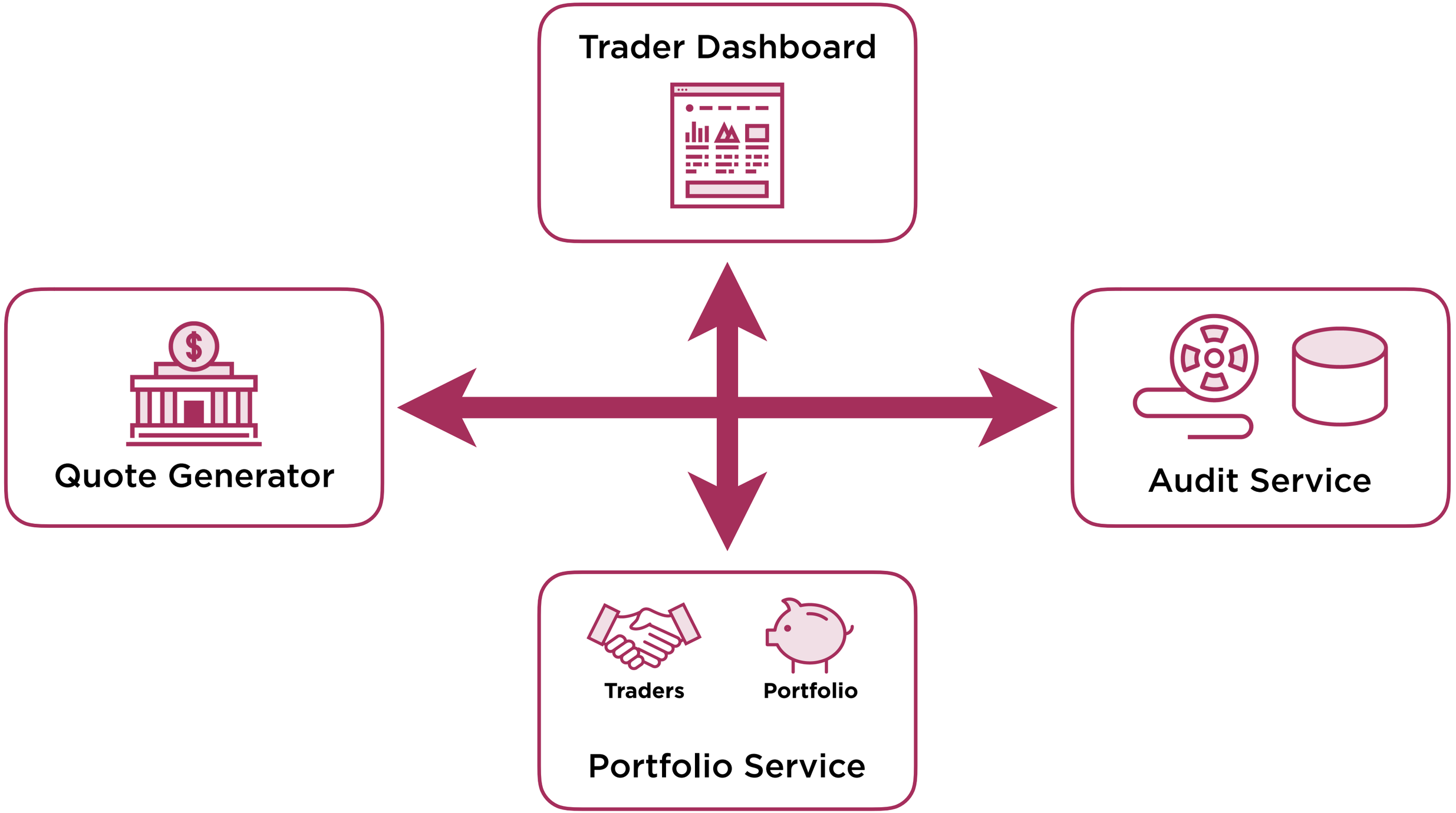
Justin Menga
FULL STACK TECHNOLOGIST
@jmenga pseudo.co.de

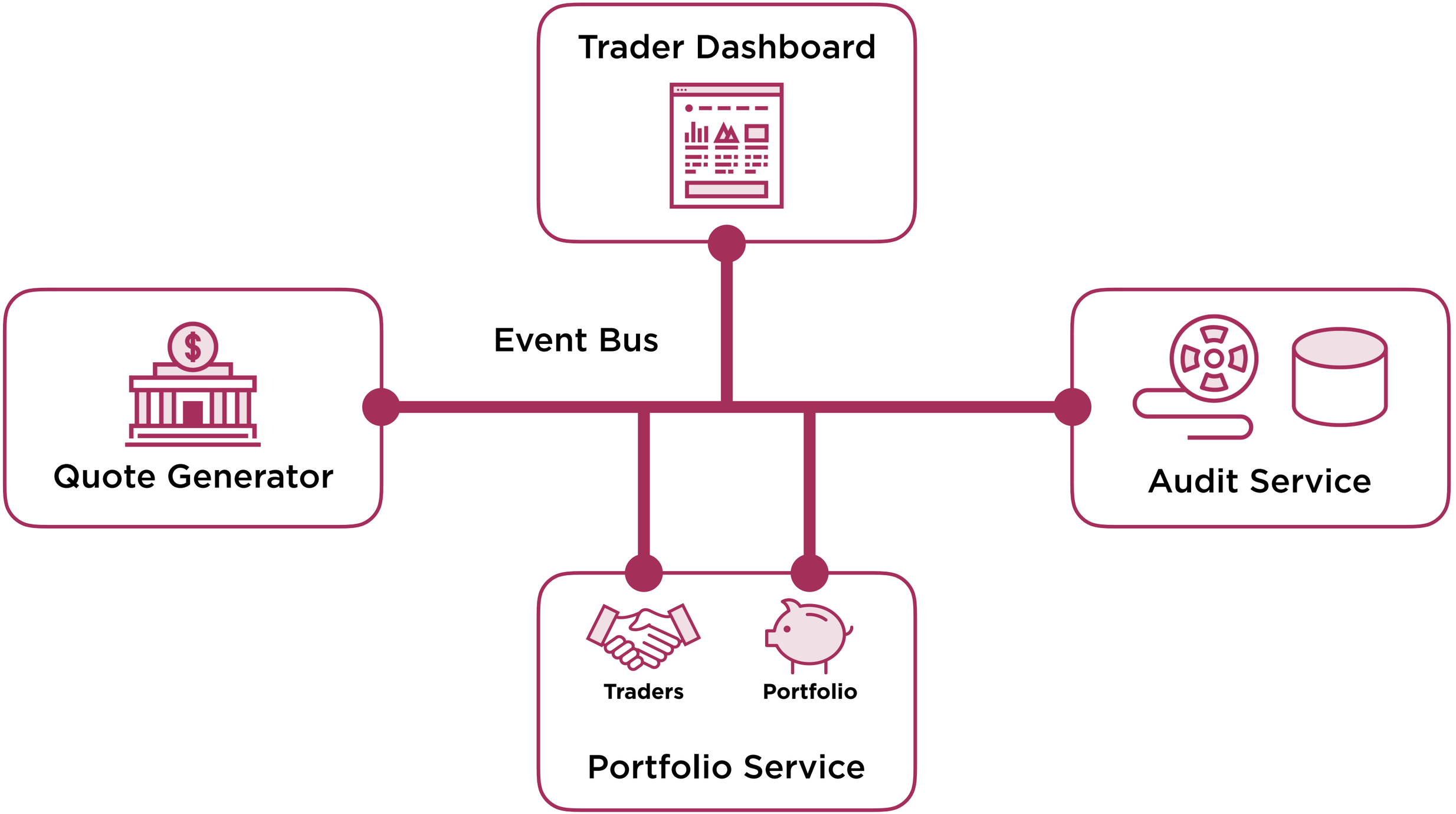
Introduction

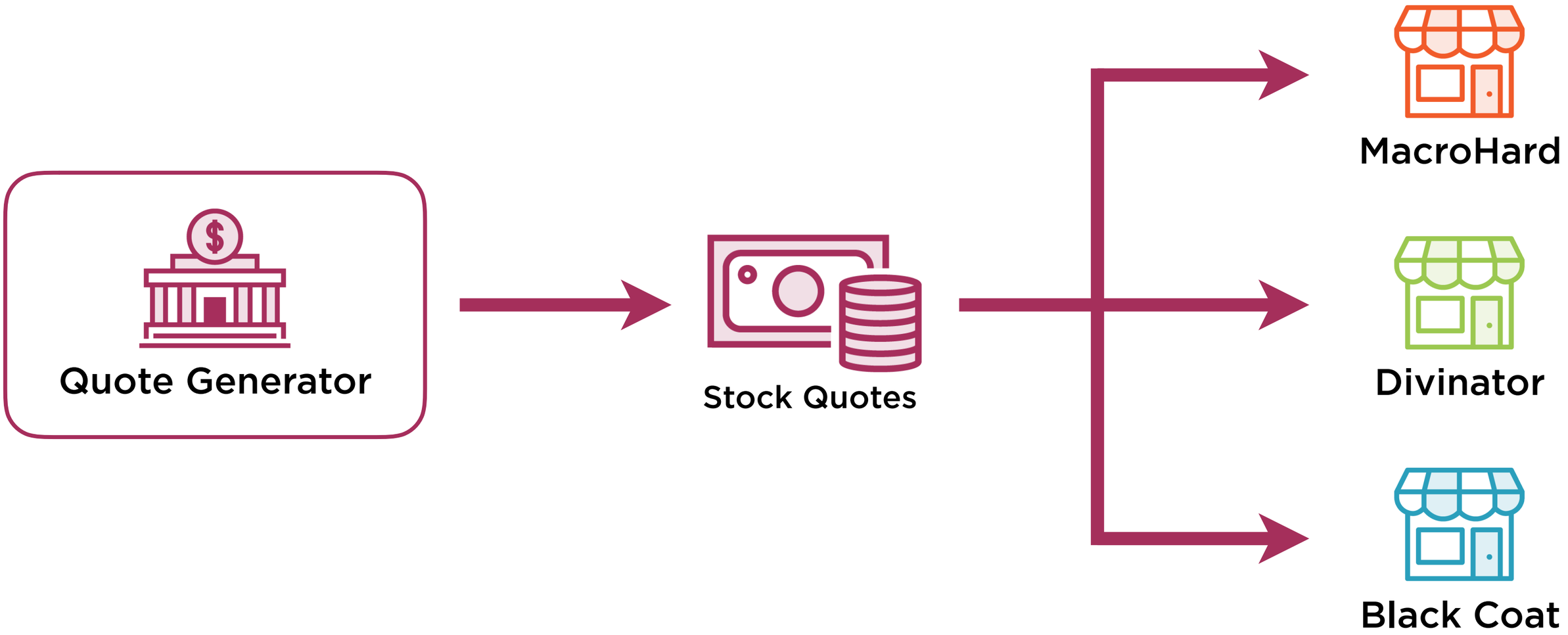
Introducing the Microtrader Application

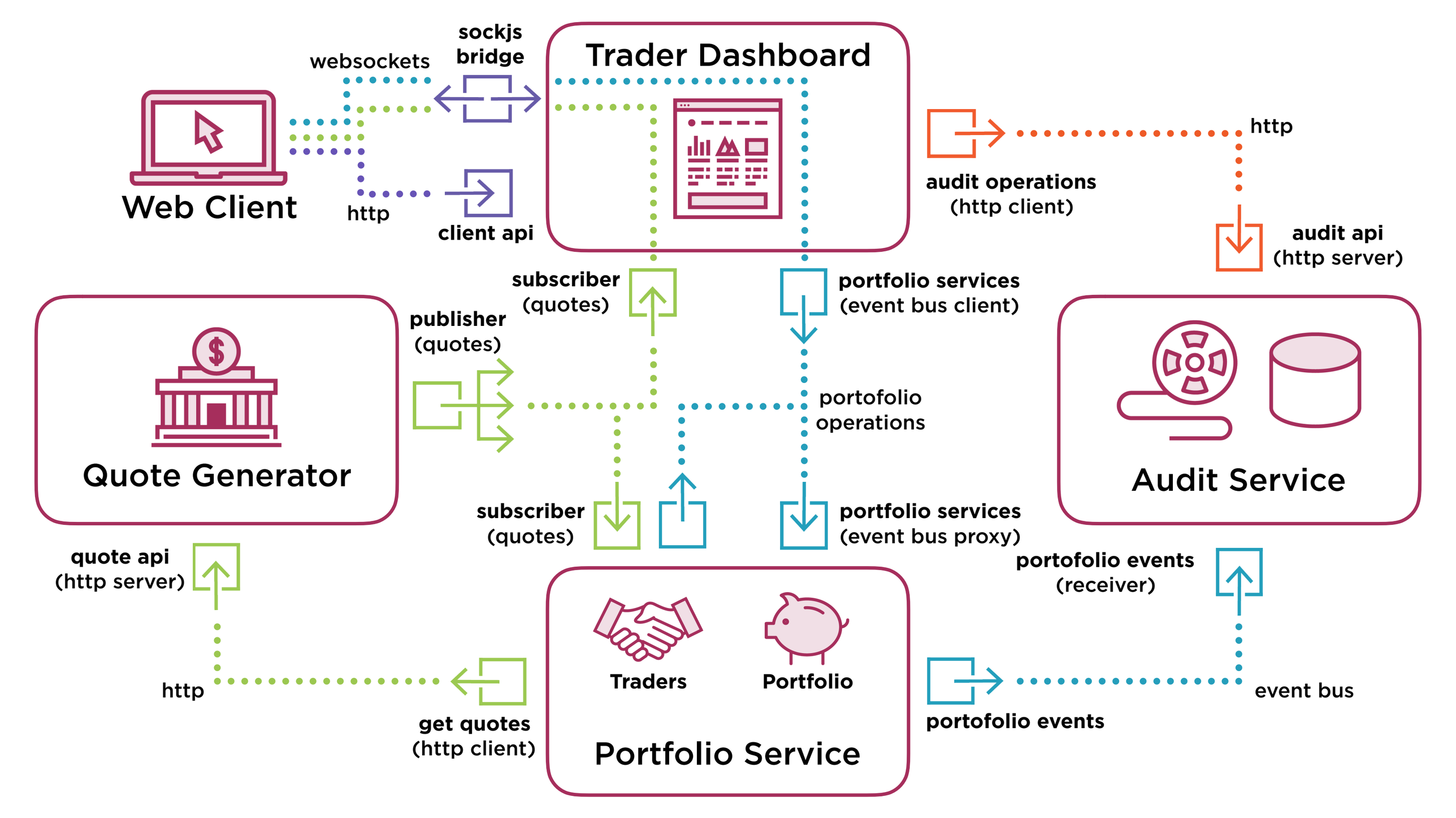
- Local development environment
- Microservices architecture
- Installing the application
- Building the application
- Running the application
- Testing the application

Application Architecture


Trader Dashboard
Quote Generator
Audit Service
Traders
Portfolio
Portfolio Service


Trader Dashboard
Event Bus
Quote Generator
Audit Service
Traders
Portfolio
Portfolio Service


Quote Generator
Stock Quotes
MacroHard
Divinator
Black Coat


Trader Dashboard
sockjs bridge
websockets
Web Client
http
client api
audit operations (http client)
http
audit api (http server)
subscriber (quotes)
portfolio services (event bus client)
publisher (quotes)
portofolio operations
Quote Generator
Audit Service
subscriber (quotes)
portfolio services (event bus proxy)
quote api (http server)
portofolio events (receiver)
Traders
Portfolio
http
get quotes (http client)
portofolio events
event bus
Portfolio Service

Demo

Application Walkthrough

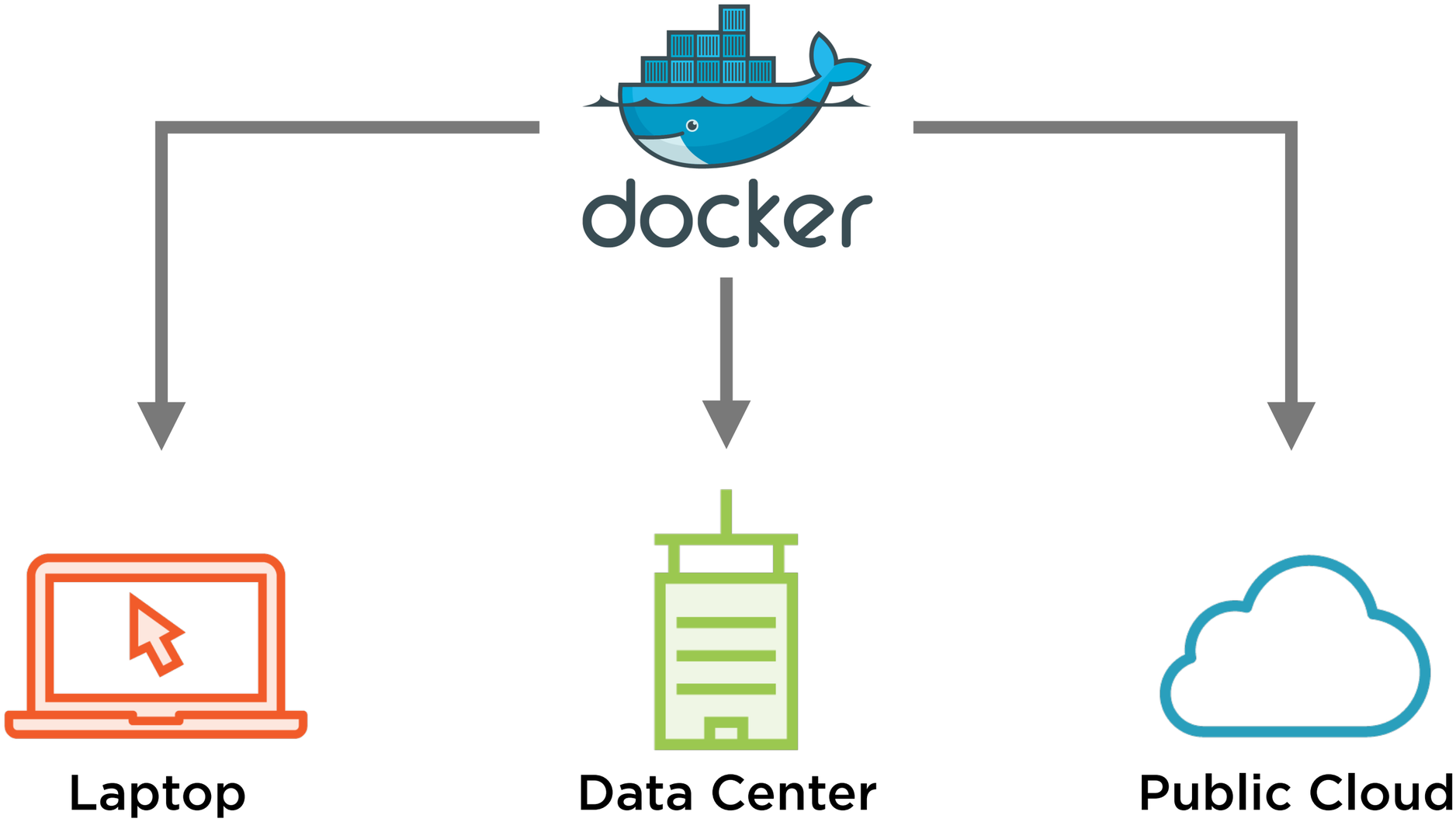
- Installing the application
- Building the application
- Running the application
- Testing the application

Installing the Application

Building the Application

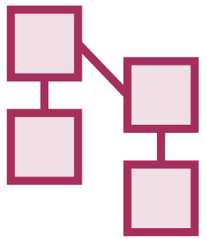
“But it worked on my machine!”

Docker works on any machine...

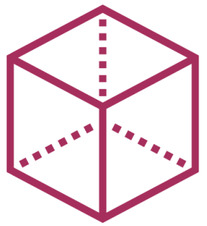


Fat JAR

Application Source



Dependencies



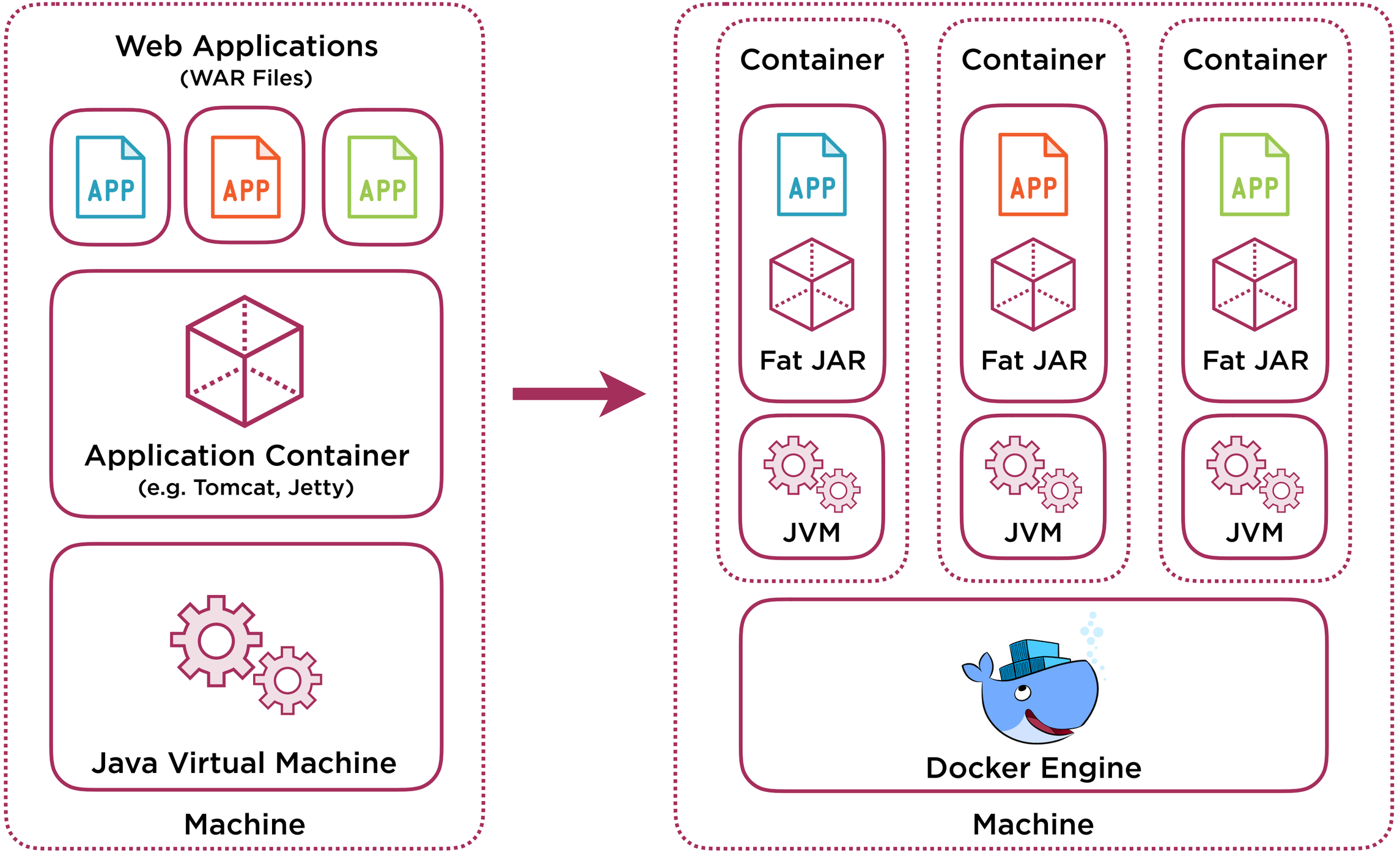
Application Container
(e.g. Jetty)

Packages application and all dependencies

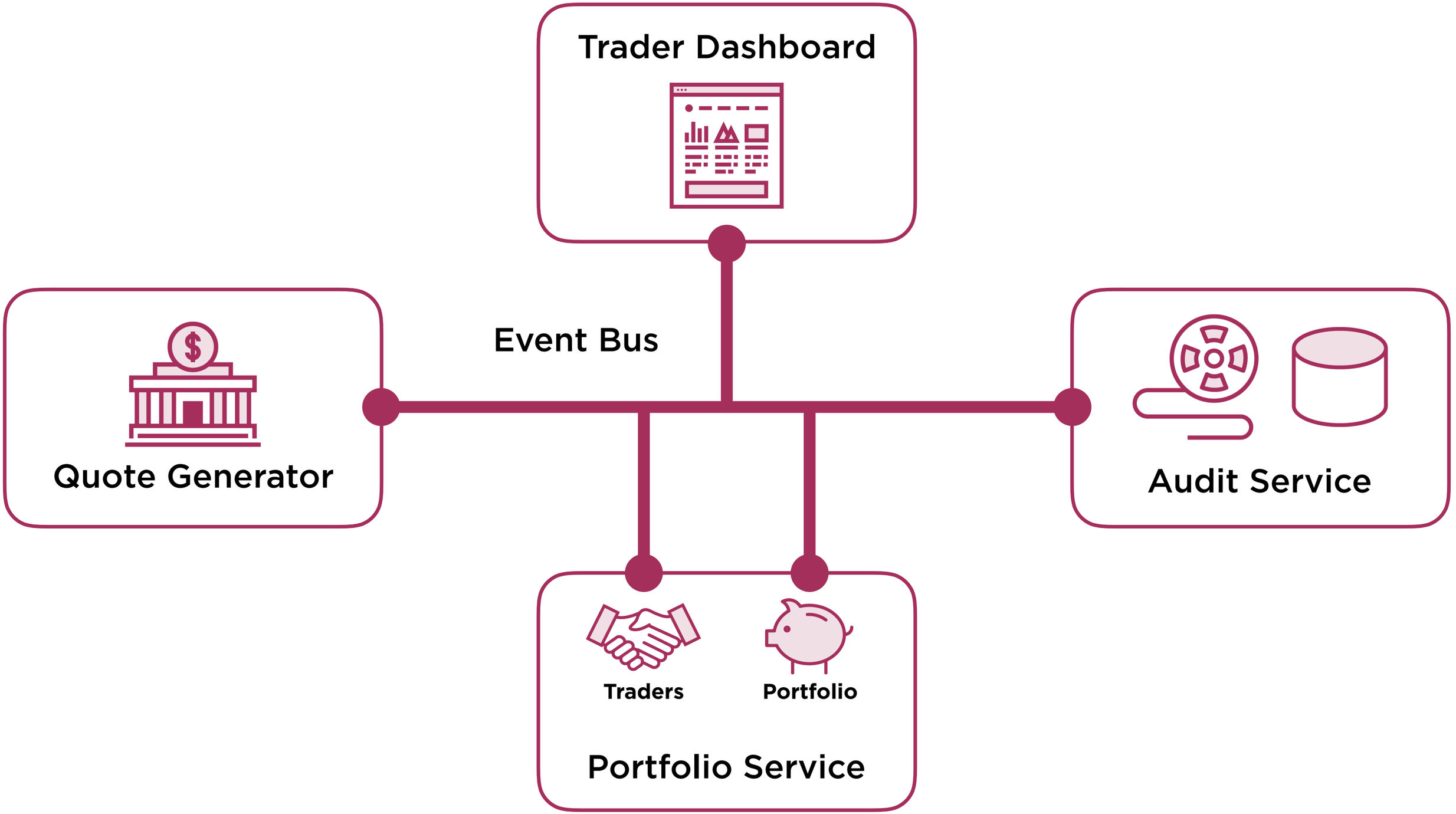
Self contained deployable artifact

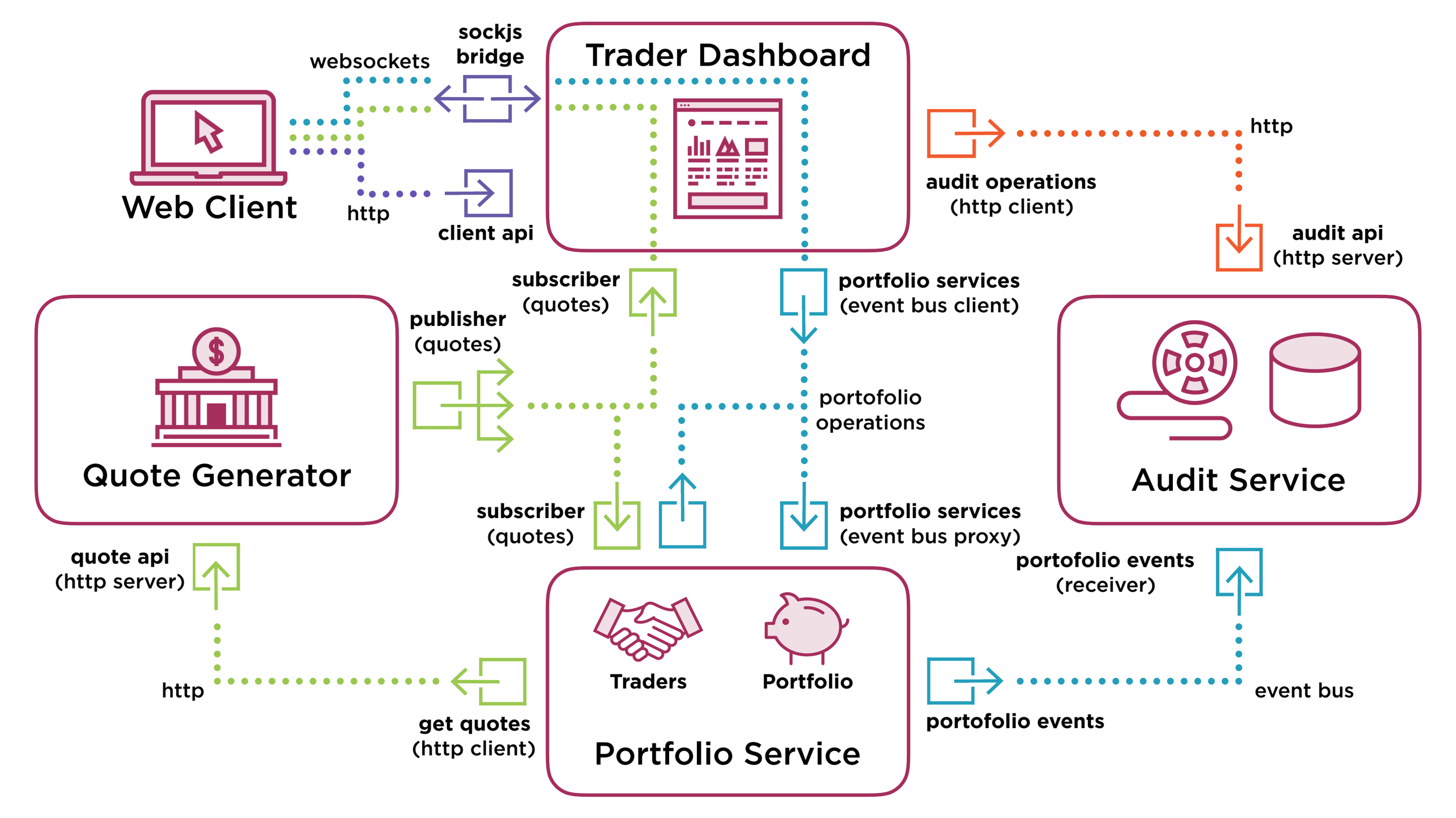
Single runnable artifact

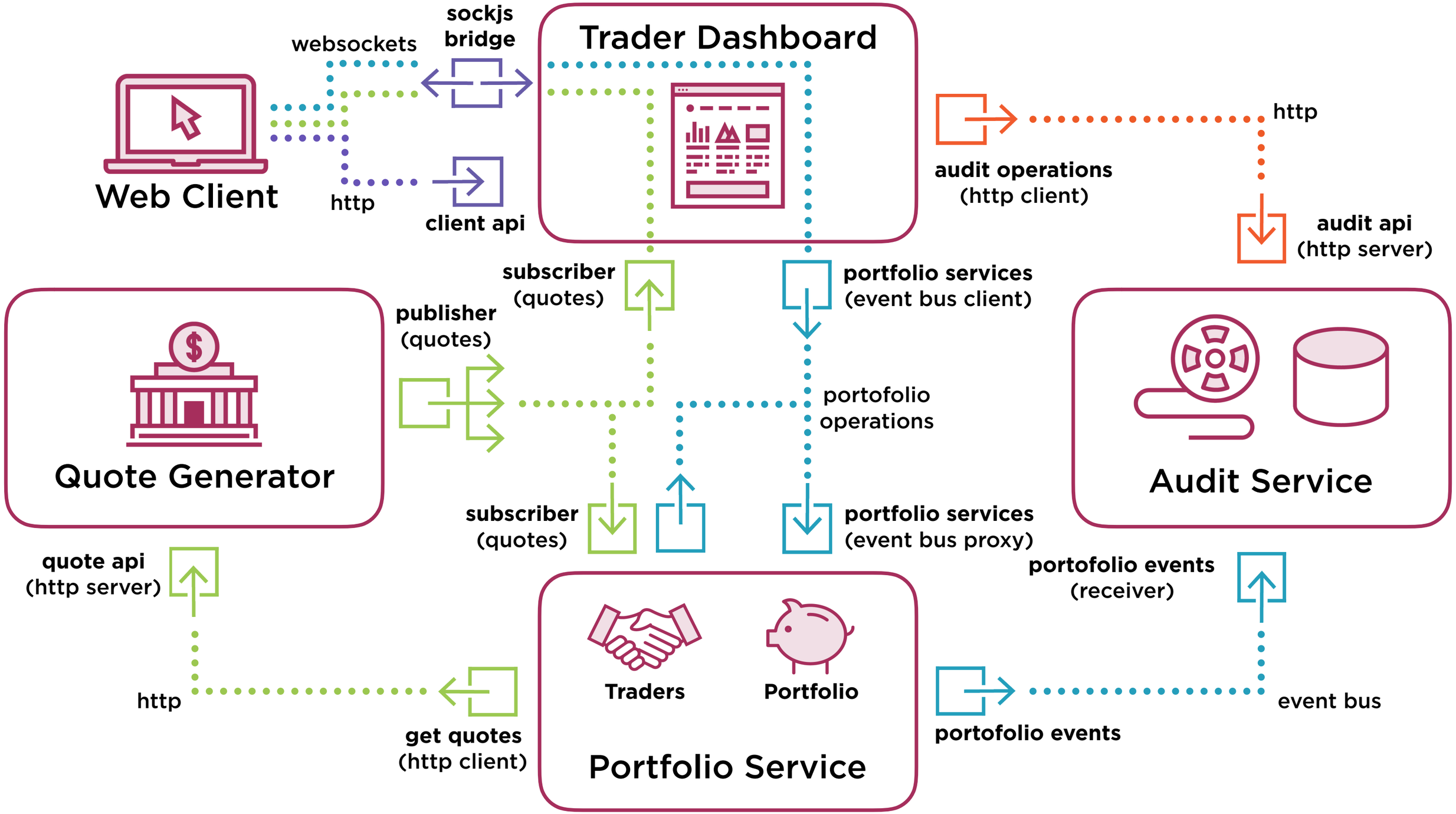
- e.g. $ java -jar my-fat-jar.jar

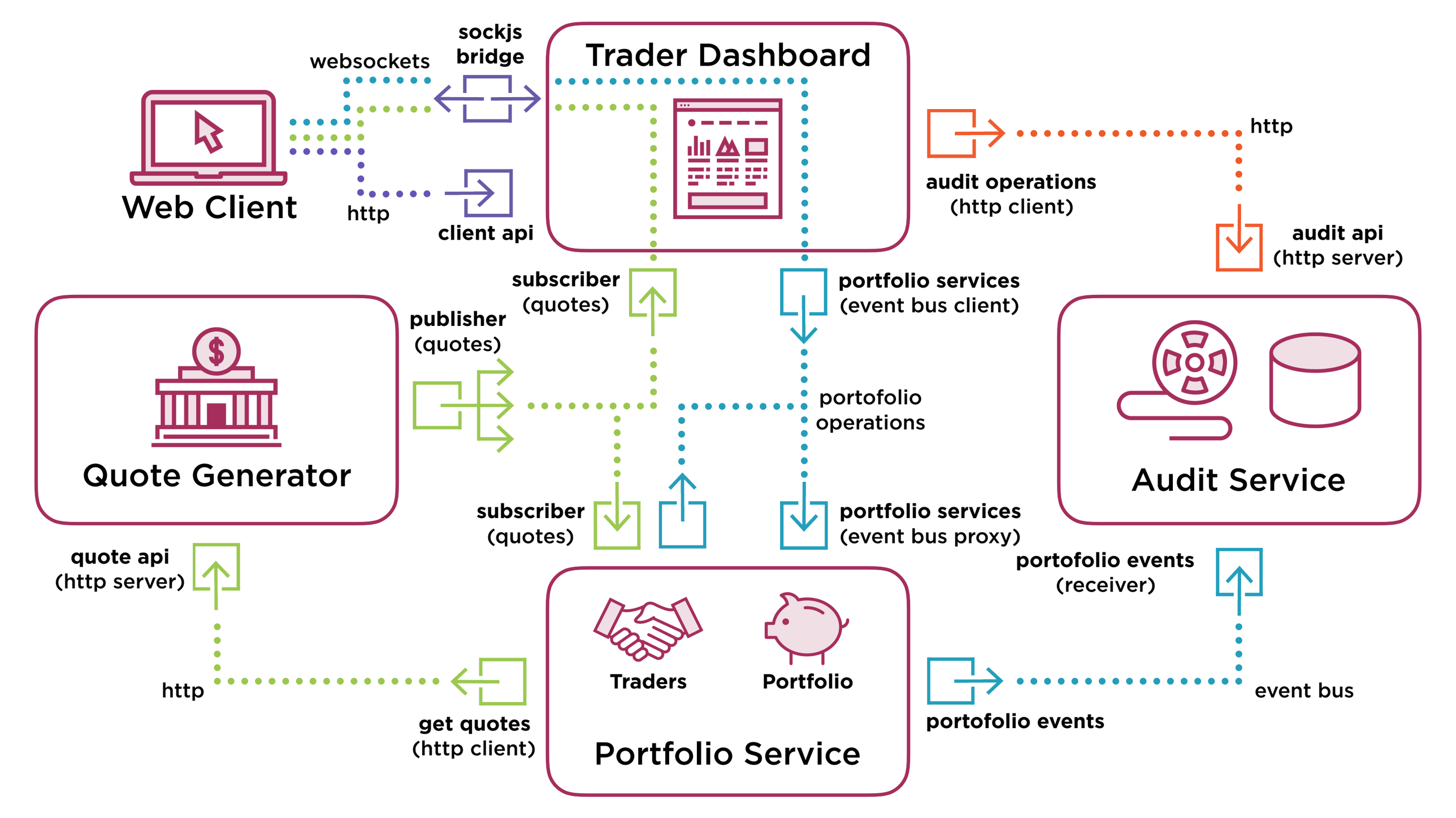

Web Applications
(WAR Files)
APP
APP
APP
Application Container
(e.g. Tomcat, Jetty)
Java Virtual Machine
Machine
Container
APP
Fat JAR
JVM
Container
APP
Fat JAR
JVM
Container
APP
Fat JAR
JVM
Docker Engine
Machine

Running the Application

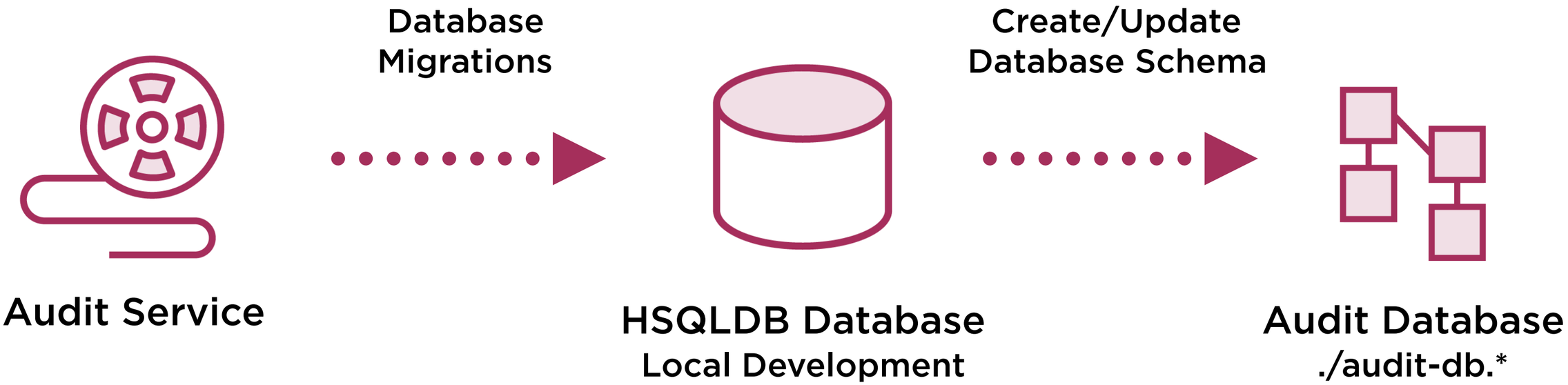

Trader Dashboard
Event Bus
Quote Generator
Audit Service
Traders
Portfolio
Portfolio Service


Trader Dashboard
websockets
sockjs bridge
Web Client
http
client api
audit operations (http client)
http
audit api (http server)
subscriber (quotes)
portfolio services (event bus client)
publisher (quotes)
portofolio operations
Quote Generator
Audit Service
subscriber (quotes)
portfolio services (event bus proxy)
quote api (http server)
portofolio events (receiver)
Traders
Portfolio
http
get quotes (http client)
portofolio events
event bus
Portfolio Service

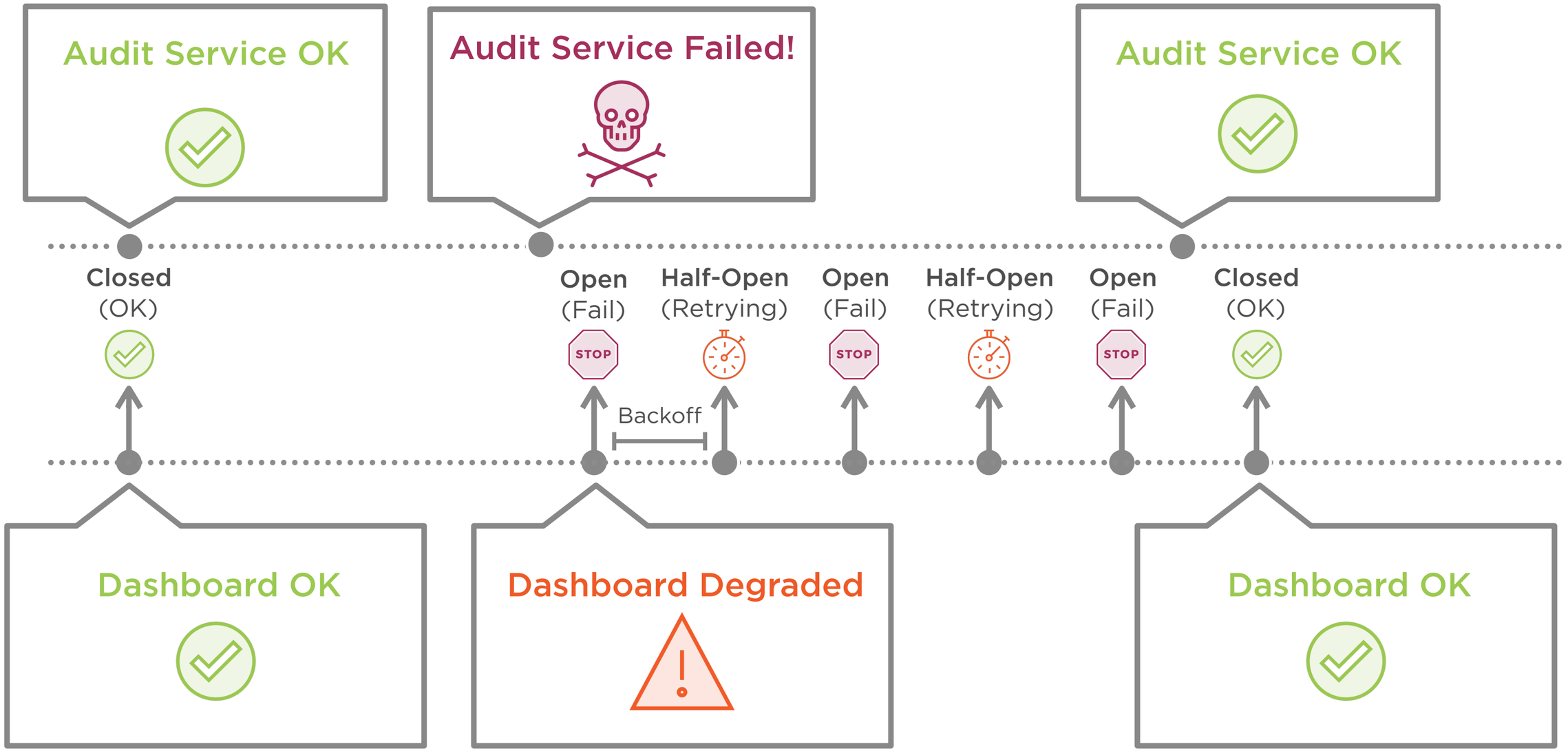

Trader Dashboard
sockjs bridge
websockets
Web Client
http
client api
audit operations (http client)
http
audit api (http server)
subscriber (quotes)
portfolio services (event bus client)
publisher (quotes)
portofolio operations
Quote Generator
Audit Service
subscriber (quotes)
portfolio services (event bus proxy)
quote api (http server)
portofolio events (receiver)
Traders
Portfolio
http
get quotes (http client)
portofolio events
event bus
Portfolio Service


Trader Dashboard
websockets
sockjs bridge
Web Client
http
client api
audit operations (http client)
http
audit api (http server)
subscriber (quotes)
portfolio services (event bus client)
publisher (quotes)
portofolio operations
Quote Generator
Audit Service
subscriber (quotes)
portfolio services (event bus proxy)
quote api (http server)
portofolio events (receiver)
Traders
Portfolio
http
get quotes (http client)
portofolio events
event bus
Portfolio Service

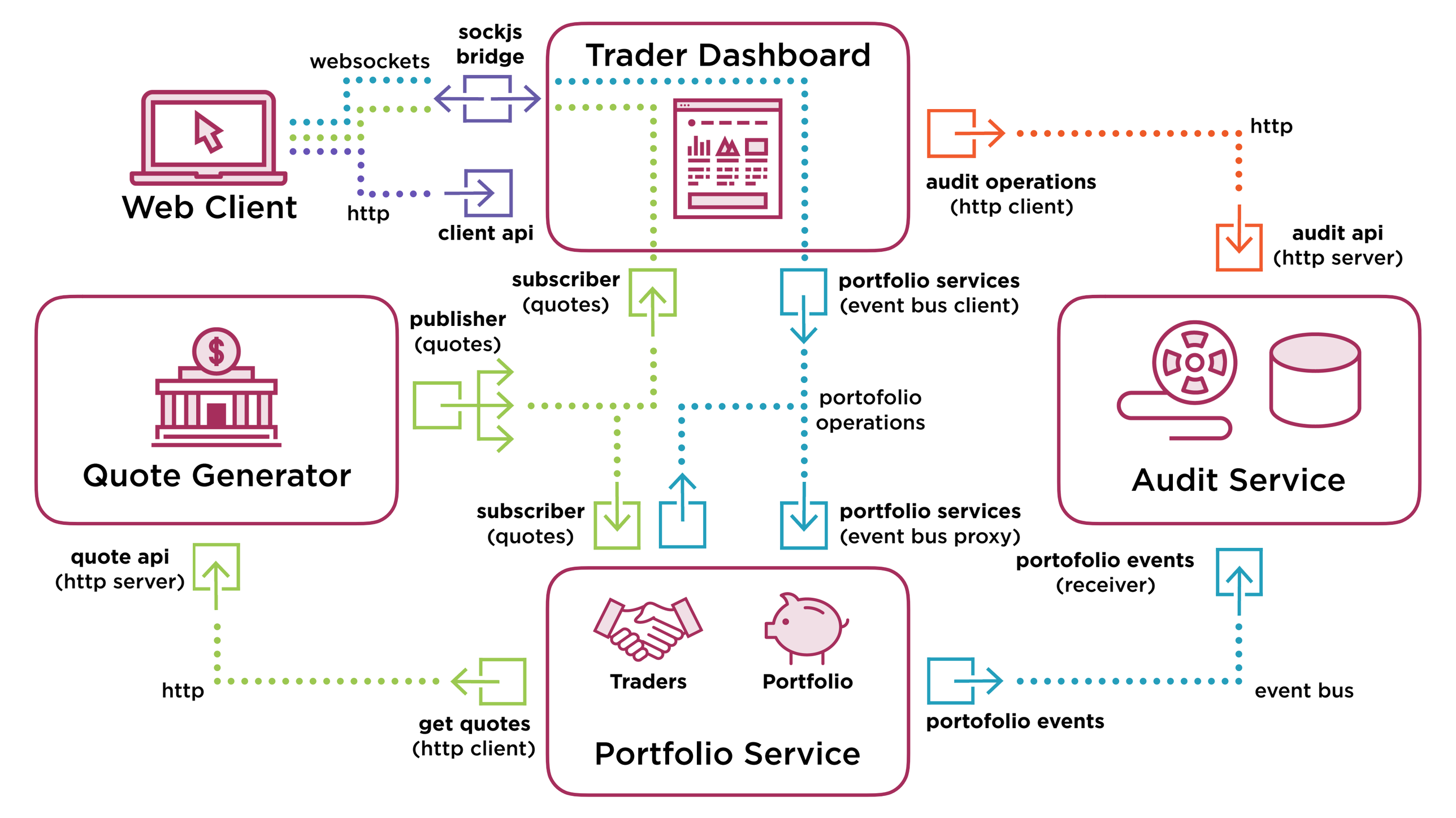
Audit Database Migrations



Circuit Breaker Pattern



Testing the Application


Trader Dashboard
websockets
sockjs bridge
Web Client
http
client api
audit operations (http client)
http
audit api (http server)
subscriber (quotes)
portfolio services (event bus client)
publisher (quotes)
portofolio operations
Quote Generator
Audit Service
subscriber (quotes)
portfolio services (event bus proxy)
quote api (http server)
portofolio events (receiver)
Traders
Portfolio
http
get quotes (http client)
portofolio events
event bus
Portfolio Service

Summary

Creating the Sample Application

- Microservices architecture
- Gradle multi-project build
- Fat JARs for each application service
- Vert.x clustering for distributed communications
- Acceptance tests verify the application as a whole