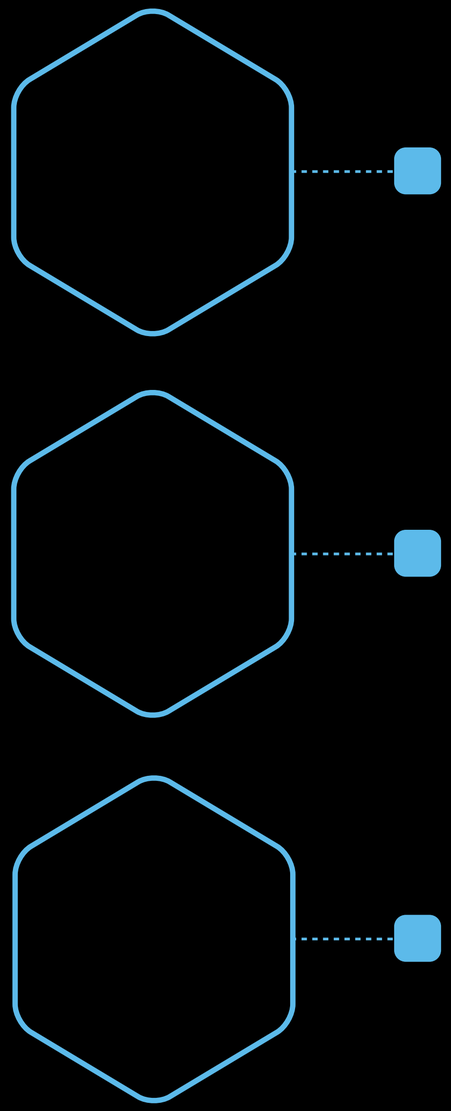
SERVERLESS JAVA WITH QUARKUS LIVE TRAINING

@PRPATEL

PRATIK PATEL

LEAD DEVELOPER ADVOCATE
@IBM
JAVA CHAMPION
JAVASCRIPT TROUBLEMAKER
PYTHON HACKER
FOUNDER, PERL RECOVERY GROUP
@PRPATEL



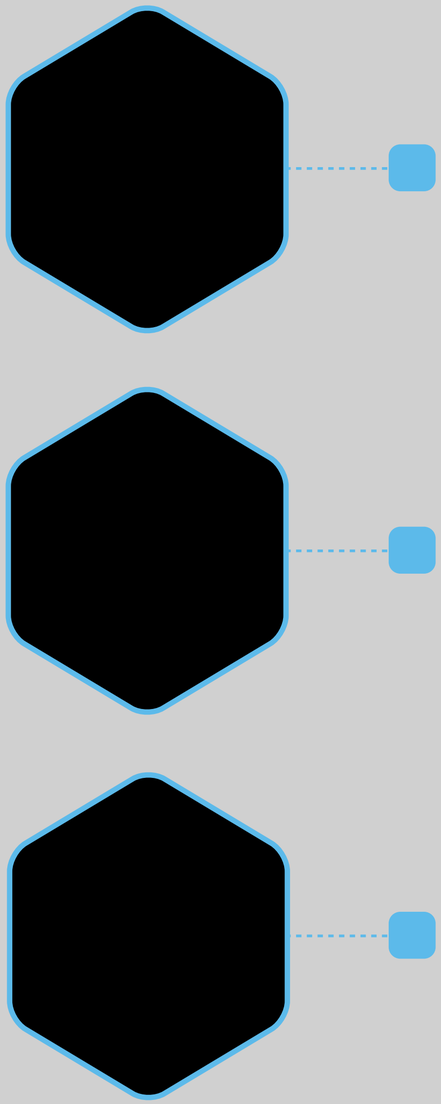
CLOUD NATIVE

JAVA AND SERVERLESS

WORKSHOP

@PRPATEL

CLOUD NATIVE

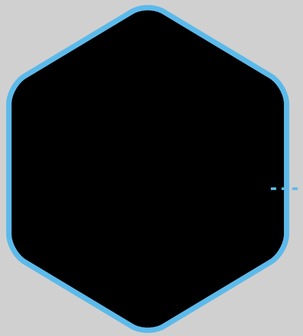


CONTAINERIZED

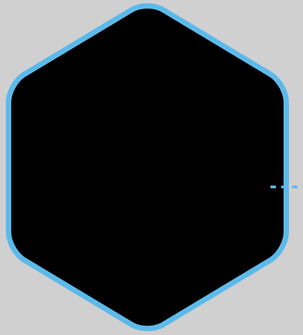
DYNAMICALLY ORCHESTRATED

MICROSERVICES-ORIENTED

CLOUD NATIVE

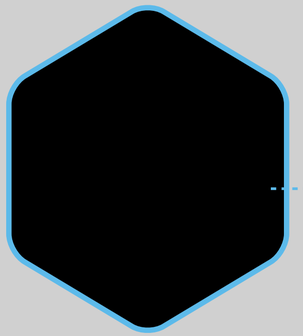


EACH APP IN OWN CONTAINER

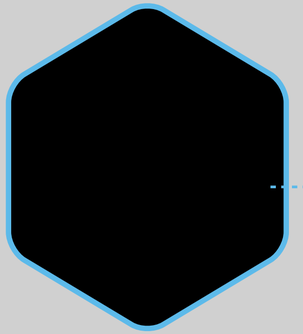


REPRODUCABILITY - TRANSPARENCY - ISOLATION

CONTAINERIZED

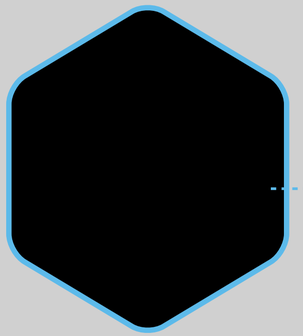


CONTAINERS ACTIVELY MANAGED

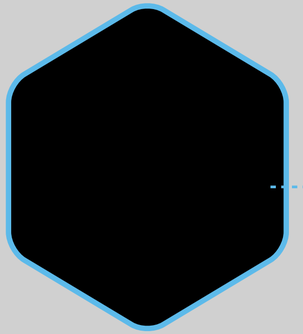


OPTIMIZE RESOURCE UTILIZATION

DYNAMIC ORCHESTRATION

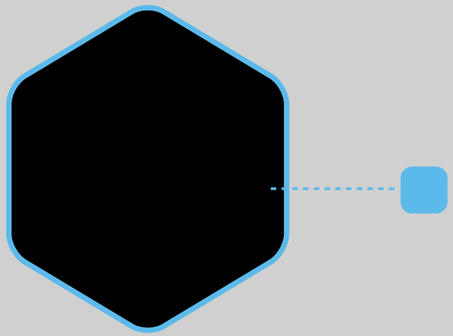


APP IS SEGMENTED INTO MICROSERVICES

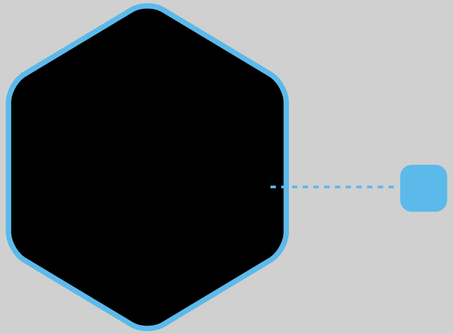


AGILITY - MAINTAINABILITY

MICROSERVICES

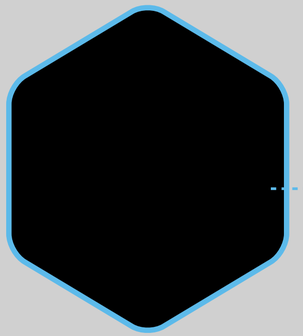


EASIER TO IMPLEMENT AND UNDERSTAND A SMALLER APPLICATION THAT PROVIDES ONE FUNCTIONALITY

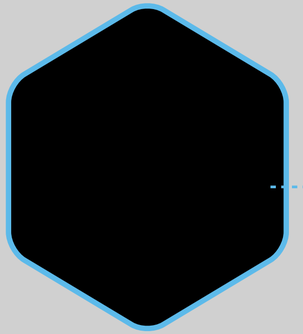


Speeds up development and makes it a lot easier to adapt the service to changed or new requirements

MICROSERVICES REVIEW

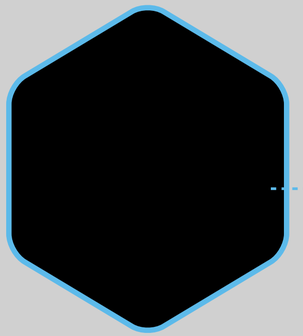


ALLOWS YOU TO SCALE MORE EFFICIENTLY

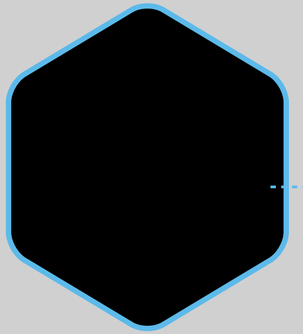


HORIZONTAL SCALING - STATELESS APPLICATION YOU CAN SEND THE NEXT REQ TO ANY APP INSTANCE

MICROSERVICES REVIEW

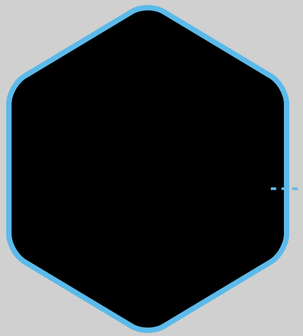


DISTRIBUTED MODEL -
COMPLEXITY AT SYSTEM LEVEL

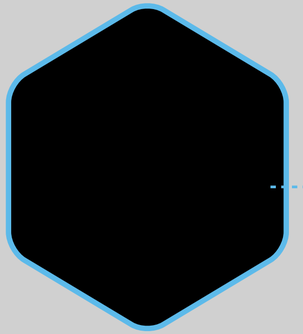


TRY TO AVOID ANY DEPENDENCIES
BETWEEN YOUR MICROSERVICES

MICROSERVICES REVIEW



HARDER TO MONITOR AND MANAGE YOUR SYSTEM IN PRODUCTION

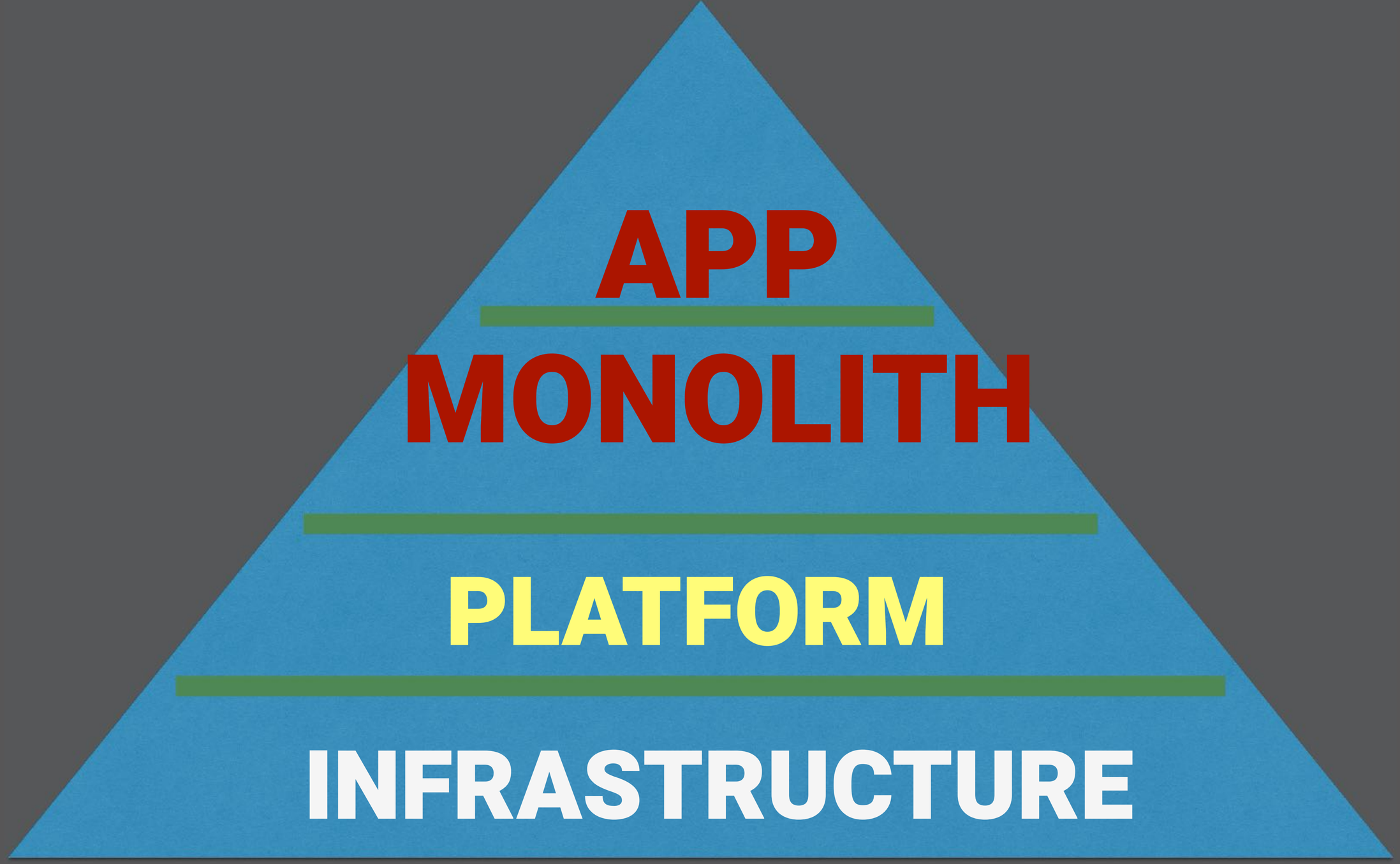


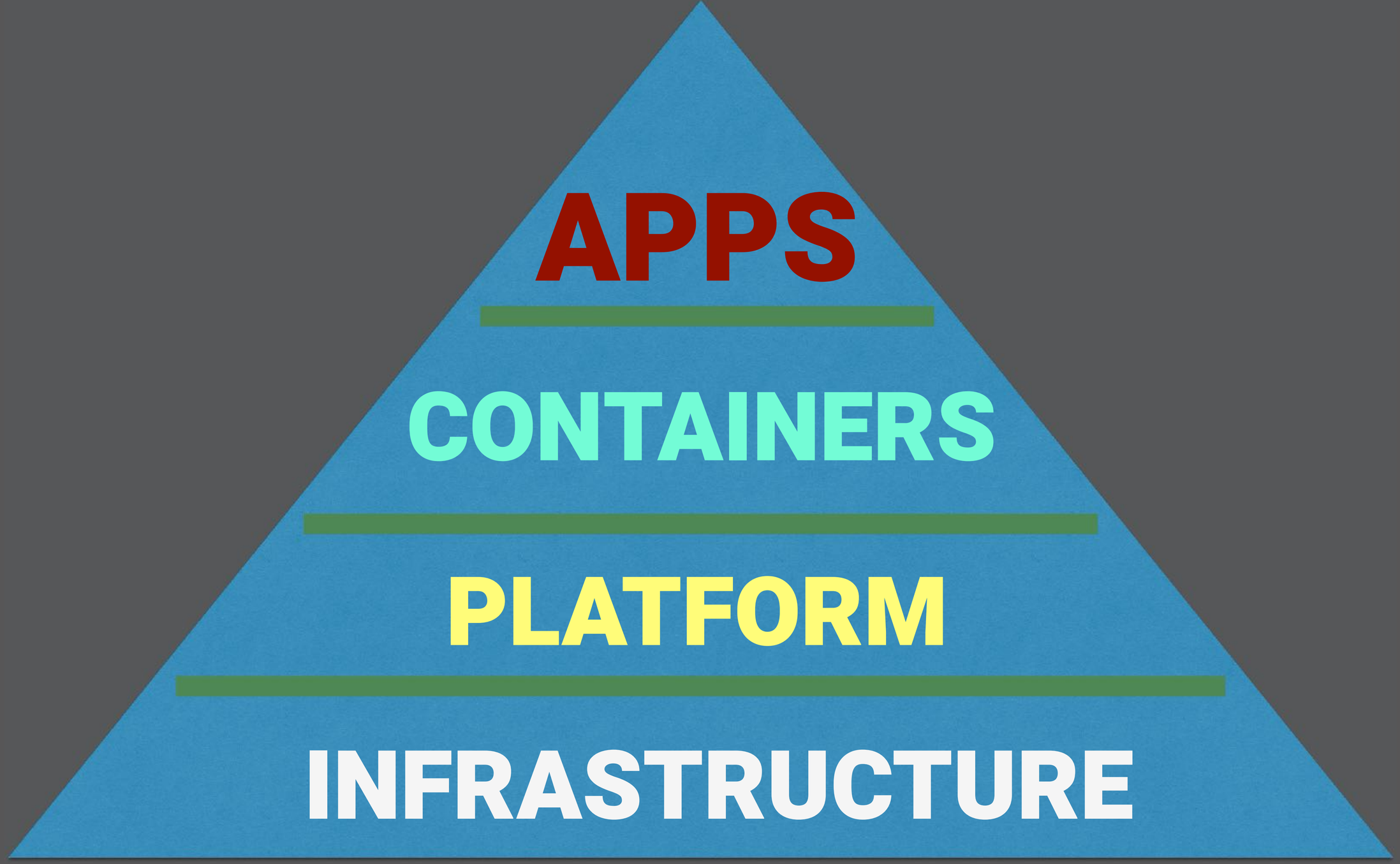
NEED TO MONITOR MUCH MORE APPLICATION INSTANCES AS YOU DID IN THE PAST

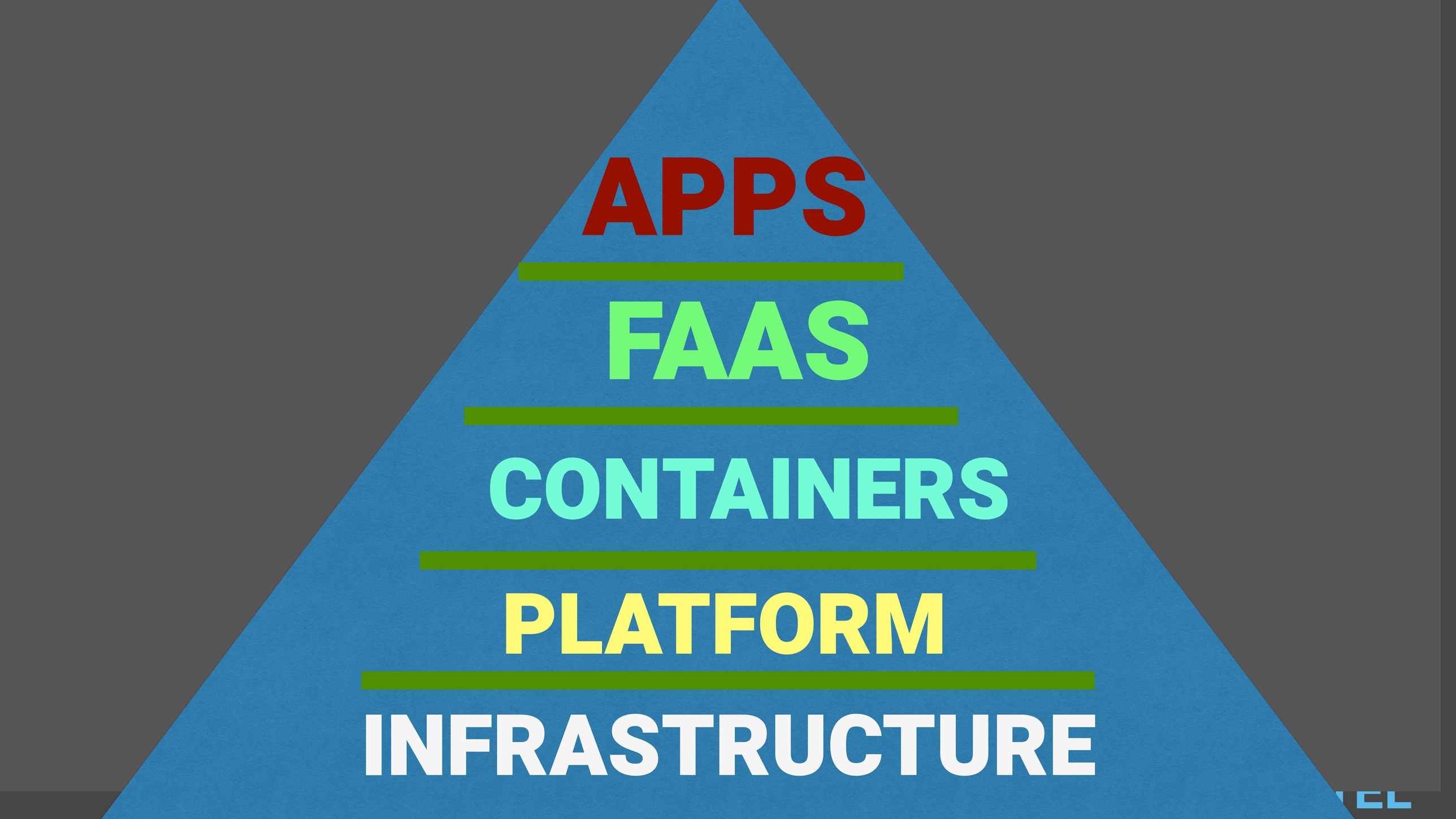
MICROSERVICES REVIEW

MONOLITH
MICROSERVICE
SERVERLESS
ARCHITECTURE

@PRPATEL

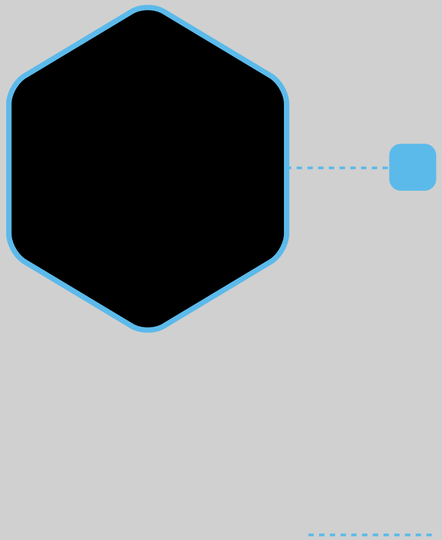

APP
MONOLITH
PLATFORM
INFRASTRUCTURE


APPS
CONTAINERS
PLATFORM
INFRASTRUCTURE


APPS
FAAS
CONTAINERS
PLATFORM
INFRASTRUCTURE

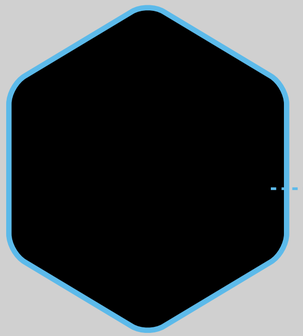
SERVERLESS

A VERY BAD NAME

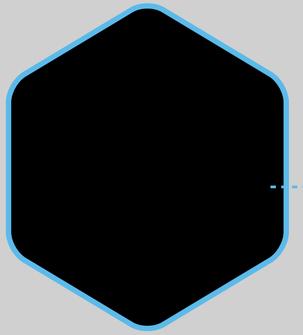


YES, THERE ARE STILL SERVERS

SERVER - LESS?



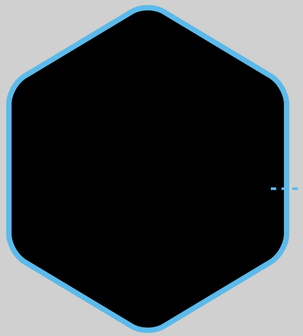
CHEAPER



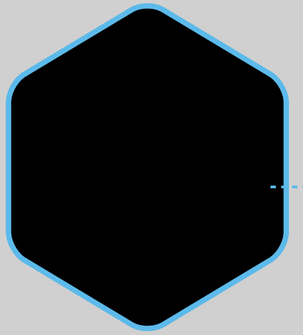
EASIER

WHAT’S THE POINT?

ALIASES
AKA



FUNCTION AS A SERVICE (FAAS)

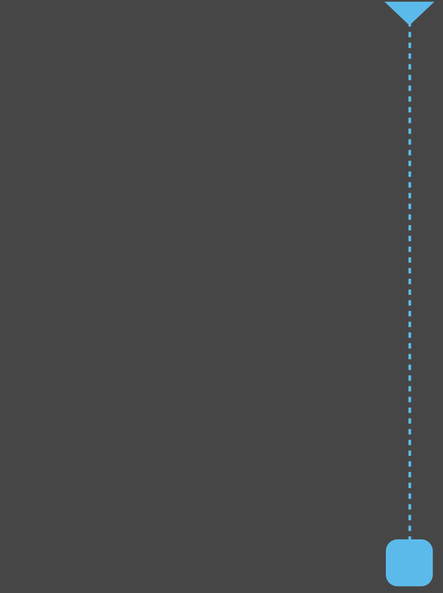


CLOUD FUNCTIONS

ALIASES

RUN CODE WITHOUT PROVISIONING OR MANAGING SERVERS

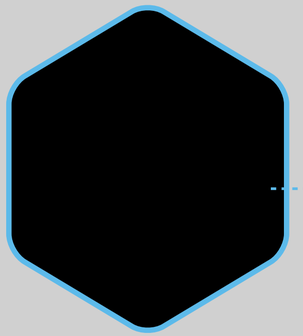
@PRPATEL



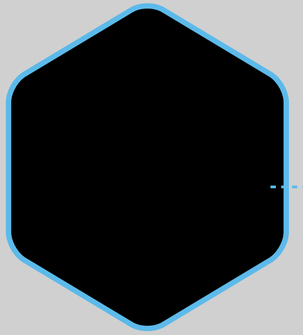
ALL THE DEV

NONE OF THE DEVOPS

@PRPATEL

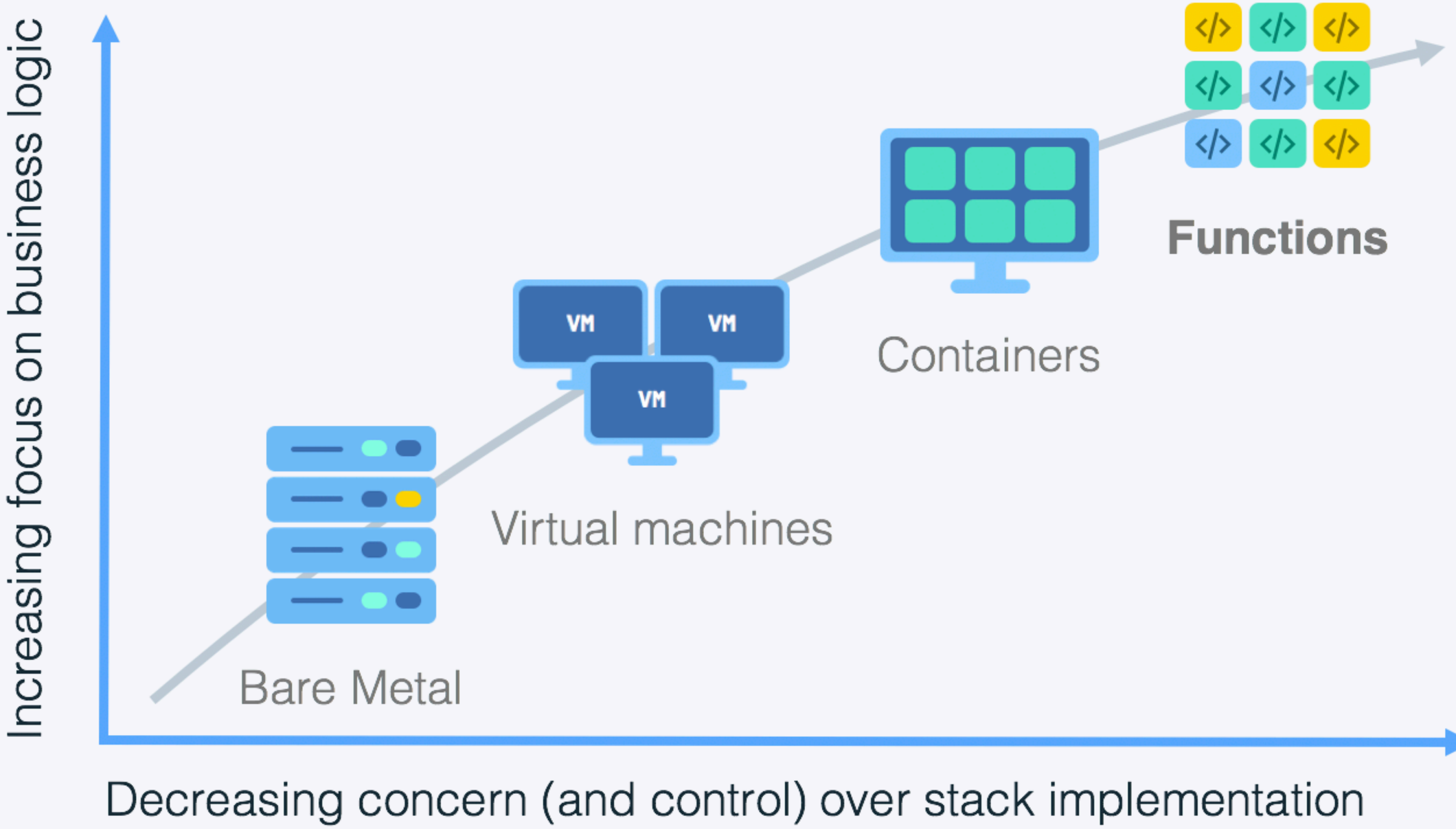


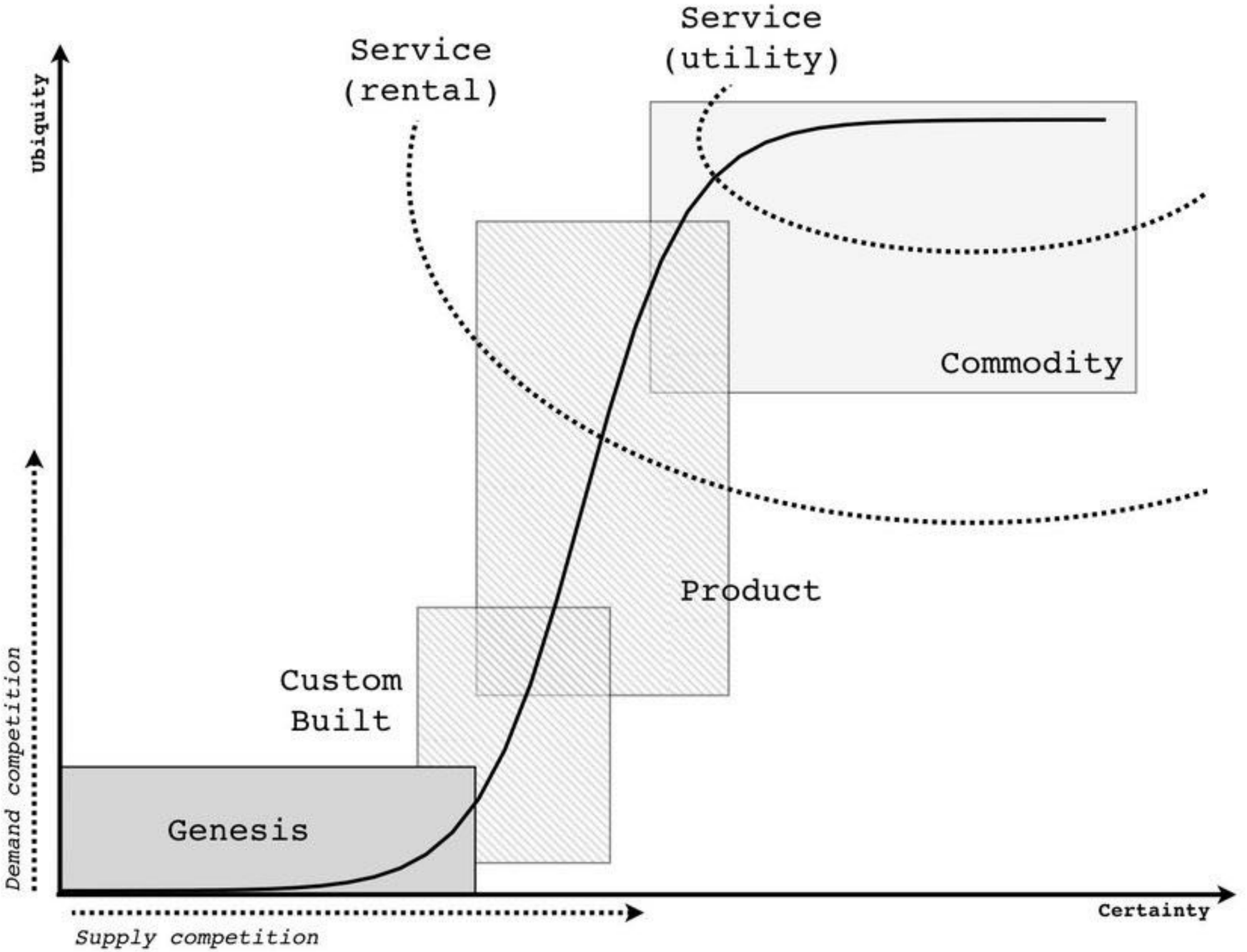
SCALABILITY

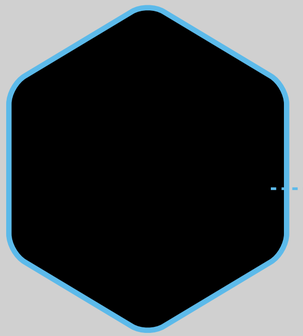


AVAILABILITY

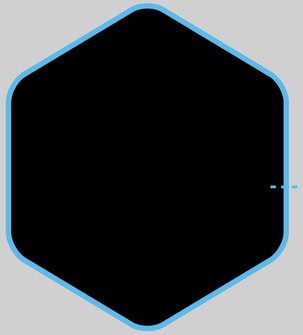
SERVERLESS TAKES CARE OF IT


Increasing focus on business logic
Functions
Containers
VM
VM
VM
Virtual machines
Bare Metal
Decreasing concern (and control) over stack implementation


Service
(rental)
Service
(utility)
Ubiquity
Commodity
Product
Custom
Built
Genesis
Demand competition
Supply competition
Certainty



KUBE - CONTAINER LEVEL, REQUIRES SETUP AND MAINTENANCE

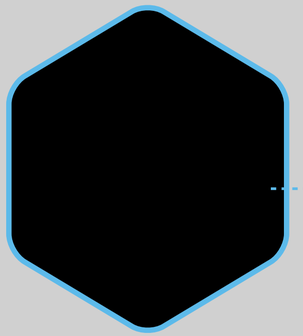


FAAS - CODE FOCUSED, LITTLE OR NO ADMIN

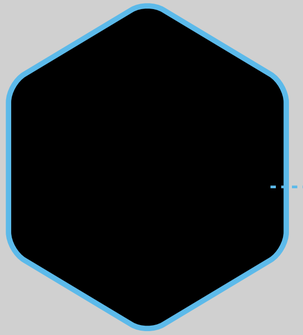
KUBERNETES OR SERVERLESS

COMPARE STANDARD MONOLITH JAVA APP WITH SERVERLESS PRINCIPLES

@PRPATEL



SMALL (A UNIT OF WORK)

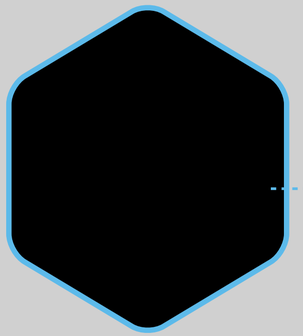


SINGLE-PURPOSE

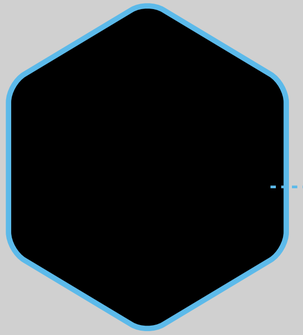
SERVERLESS - FAAS

- SMALL (A UNIT OF WORK)
- SI[illegible]SE

SERVERLESS - FAAS



SHORT RUNNING



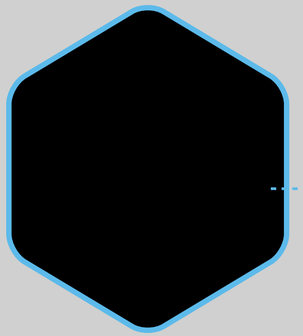
STATELESS

SERVERLESS - FAAS

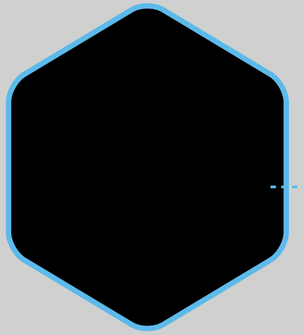
- SHORT RU

- STATELES

SERVERLESS - FAAS



SYNC



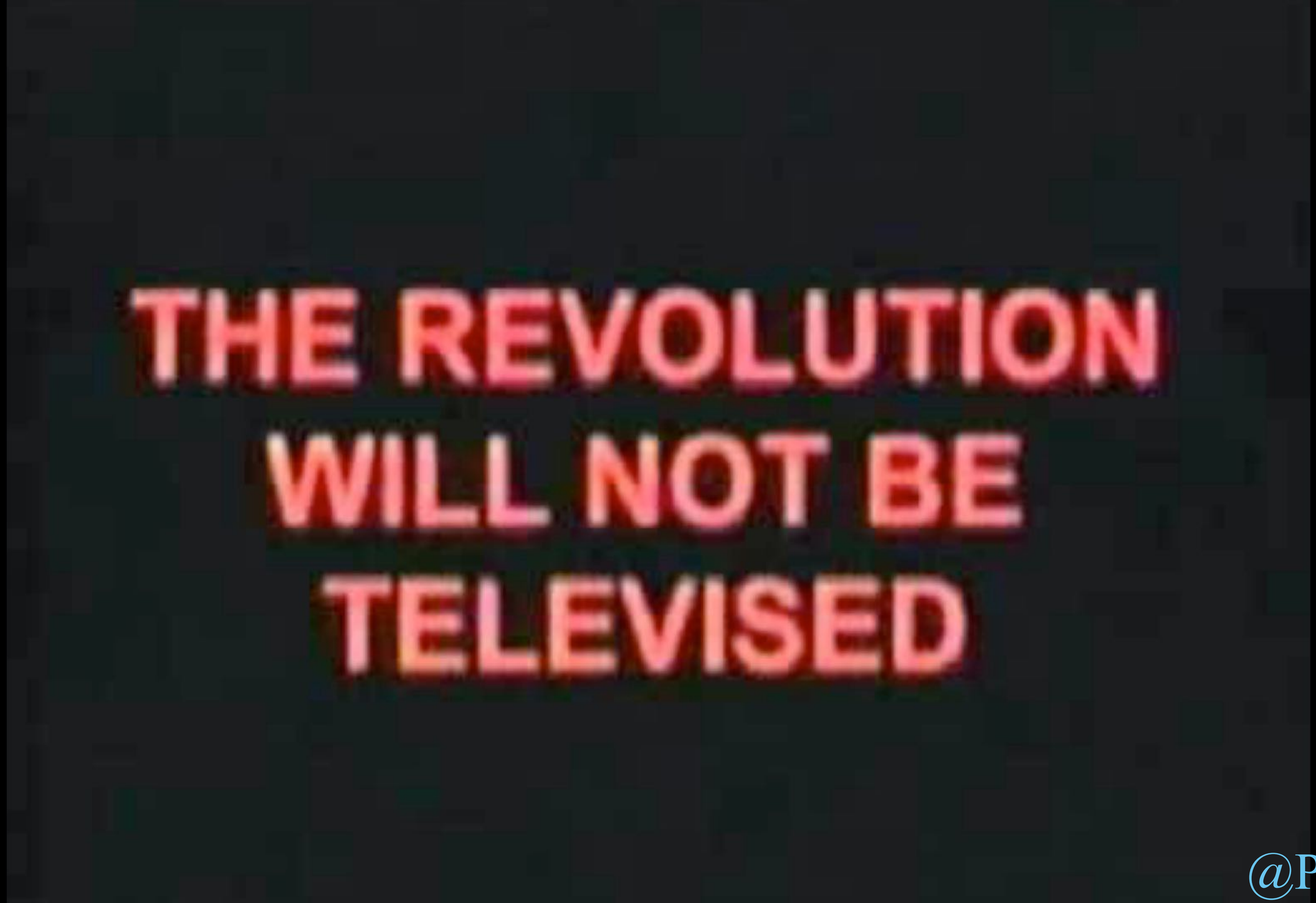
ASYNC

SERVERLESS - FAAS

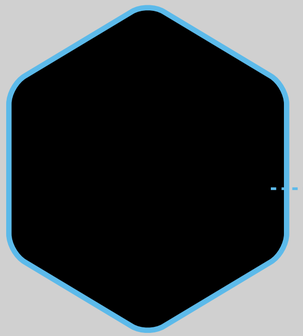
SYNC

ASYNC

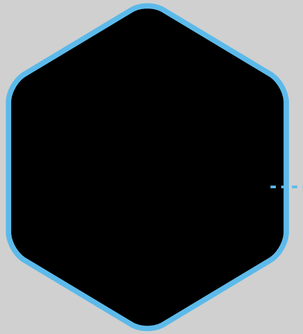
SERVERLESS - FAAS



JAVA + FRAMEWORKS == !SERVERLESS (?)



HEAVY - GEARED TOWARDS MONOLITHS



MEANT TO BE LONG-RUNNING, SLOW STARTUP OK FOR THIS ARCHITECTURE

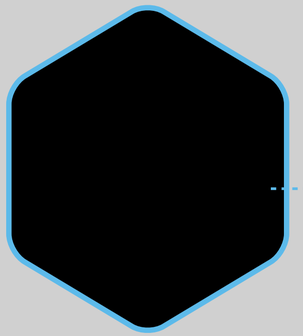
WHY NO FRAMEWORKS?


FROM MY COLD, DEAD HANDS!
makeameme.org

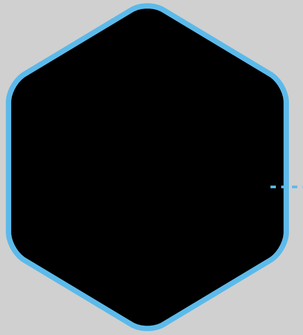
“SERVERLESS JAVA”

FAST STARTUP
LOW FOOTPRINT

@PRPATEL

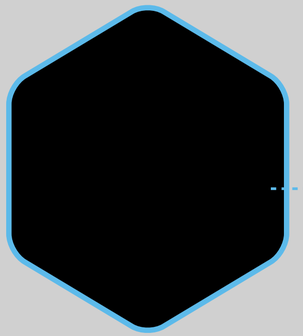


FASTER STARTUP TIMES

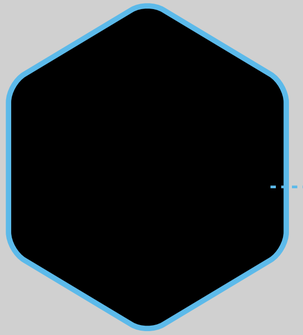


LOWER OVERHEAD

JAVA - NOW SERVERLESS FRIENDLY

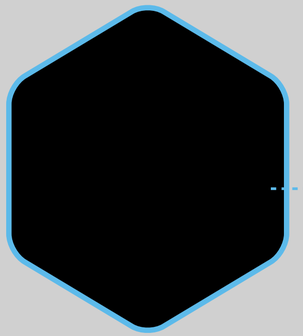


- JAVA LANG “MODERNIZED”

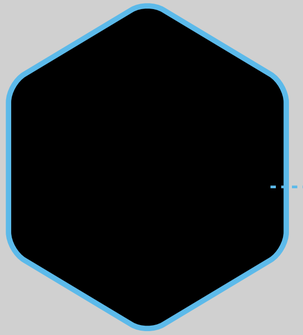


- RUNTIME CHOICE: OPENJ9

JAVA - NOW SERVERLESS FRIENDLY

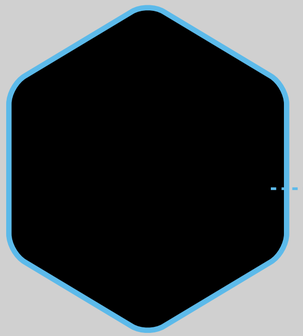


QUARKUS.IO

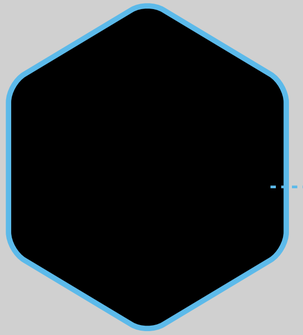


MICRONAUT

NEW FRAMEWORKS

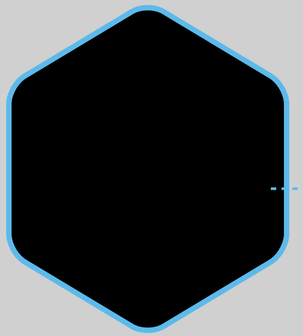


AOT COMPILATION

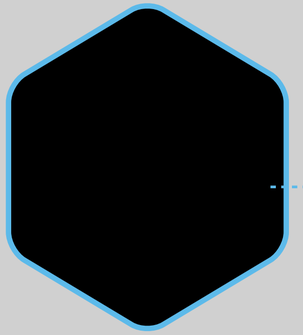


NO REFLECTION

FAST STARTUP / PERF

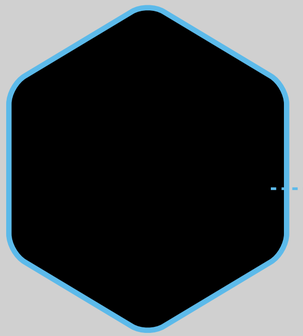


ANNOTATION SCANNING

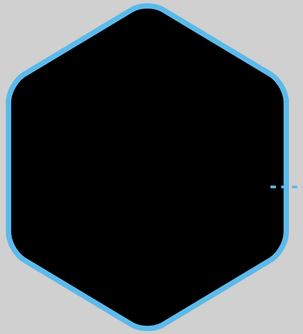


XML PARSING

BUILD TIME INSTEAD OF RUNTIME

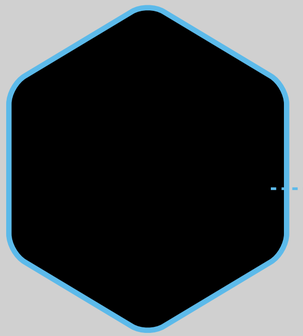


REDUCES STARTUP TIME

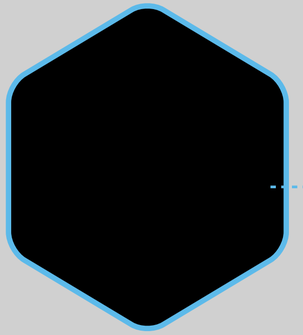


REDUCES RUNTIME MEMORY USAGE

BUILD TIME INSTEAD OF RUNTIME



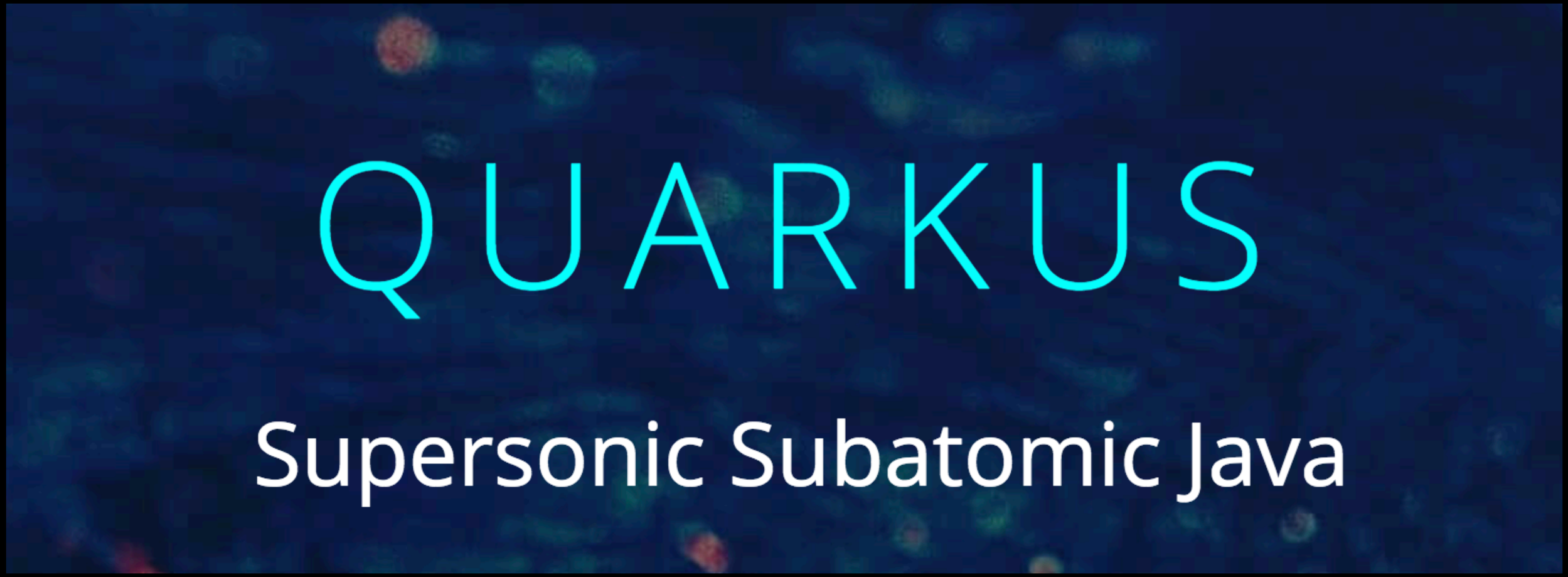
NEAR INSTANT STARTUP



LIMITING - NO REFLECTION, OTHER TRADEOFFS

GRAALVM

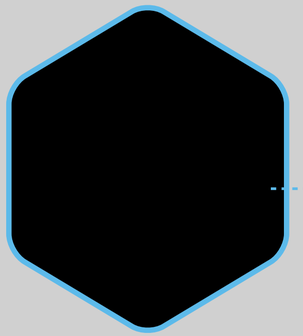
QUESTIONS?



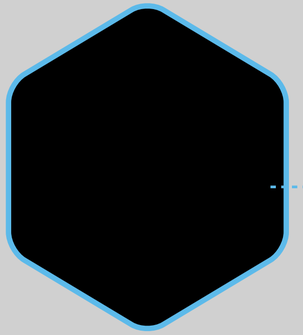
@prpatel

CODING THAT SPARKS JOY!

@PRPATEL

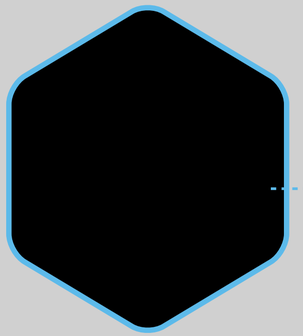


RAPID DEVELOPMENT

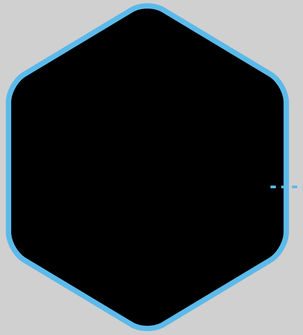


GREAT PRODUCTIVITY

WHY QUARKUS?

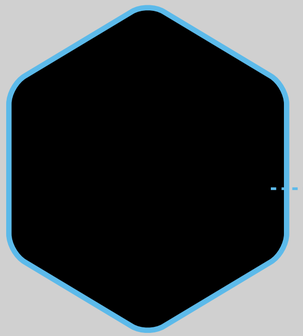


REDUCED BOILERPLATE

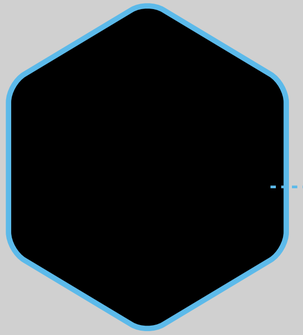


CONVENTION OVER CONFIG

WHY QUARKUS?



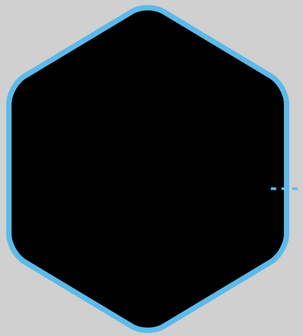
CODE -> TEST / NO RESTART



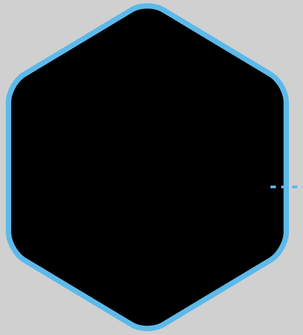
REFLECTIVE DEFAULTS -> LESS CODE

WHY QUARKUS?

CLOUD NATIVE

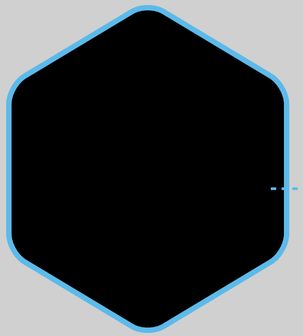


LOW FOOTPRINT

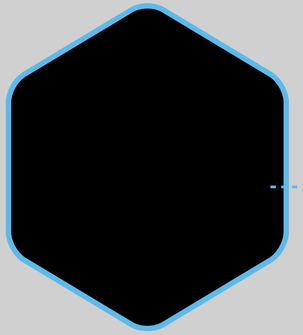


FAST STARTUP

WHY QUARKUS?

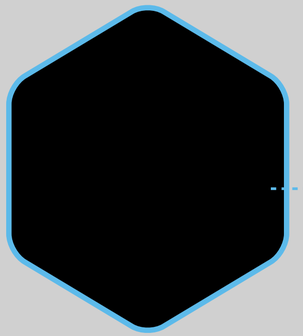


BASED ON STANDARDS

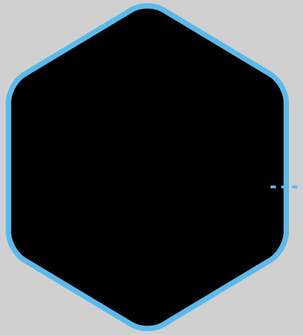


PROVEN, PRODUCTION-READY CLOUD-READY

WHY QUARKUS?

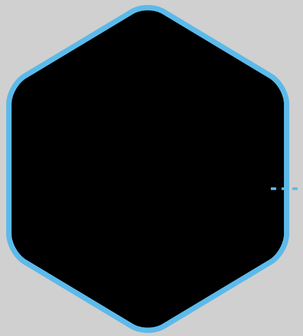


STANDARDS BASED

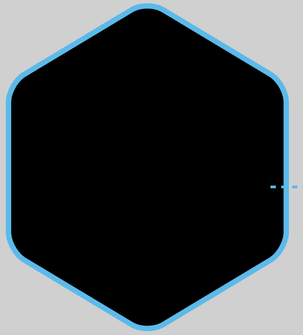


LIBS REQUIRE ‘FIXES’

QUARKUS

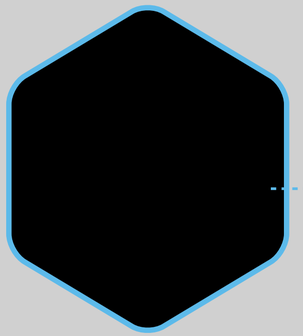


HAS CDI - BUT LIMITED

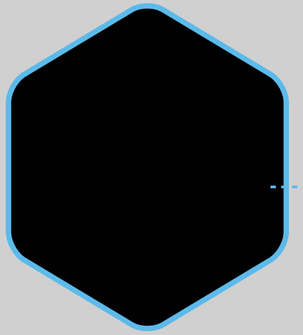


NOT SPEC COMPLIANT

LIMITATIONS

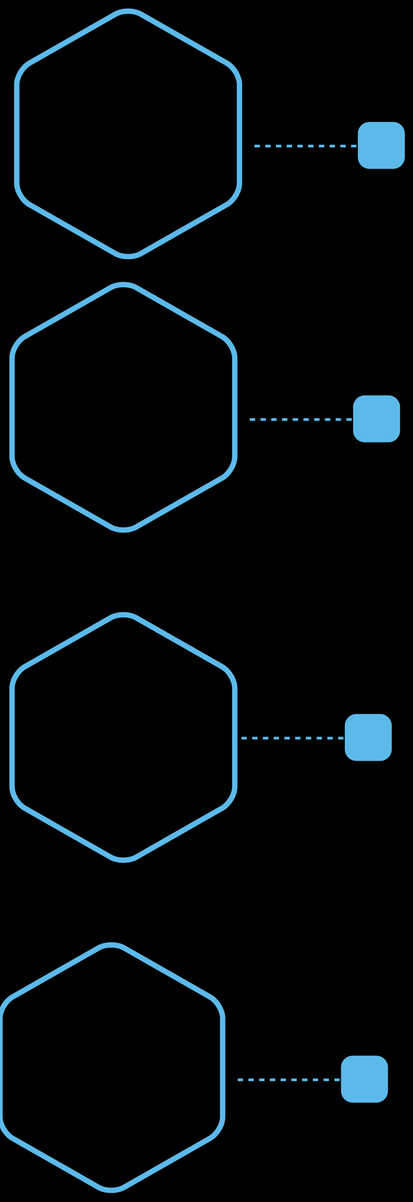


GRAALVM



SUBSTRATE VM

FAST STARTUP



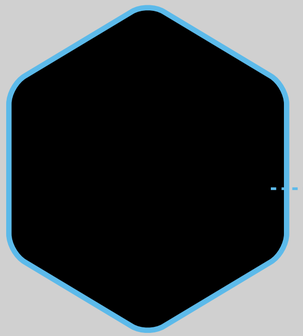
HTTP MICROSERVICES

REACTIVE APPLICATIONS

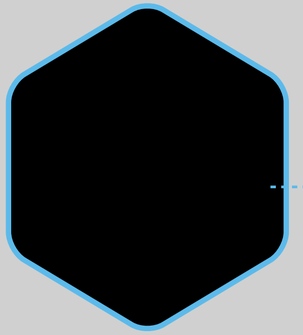
MESSAGE-DRIVEN MICROSERVICES AND

SERVERLESS

@PRPATEL

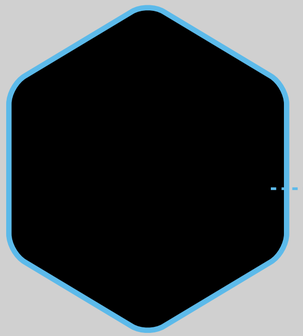


STANDARDS BASED

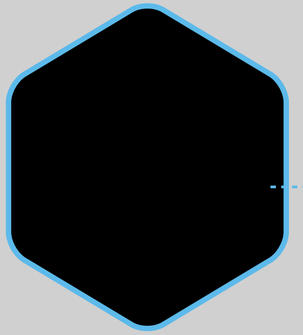


LIBS REQUIRE ‘FIXES’

QUARKUS

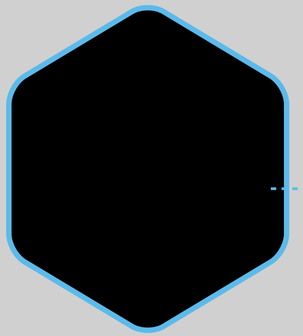


HAS CDI - BUT LIMITED

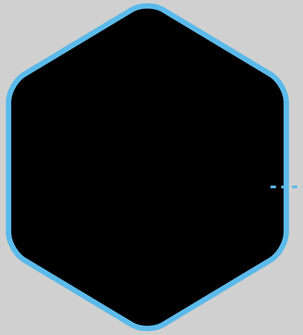


NOT SPEC COMPLIANT

LIMITATIONS



GRAALVM



SUBSTRATE VM

FAST STARTUP


MakeAGIF.com

CDI

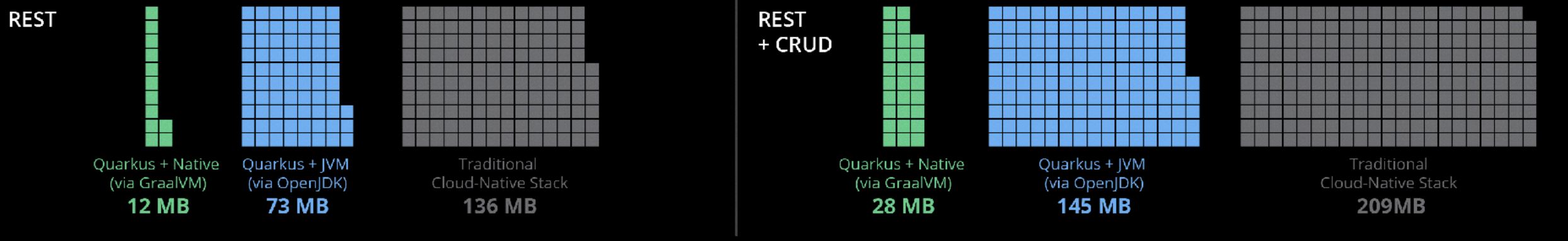
JPA + PANACHE

VERT.X - REACTIVE BY DESIGN

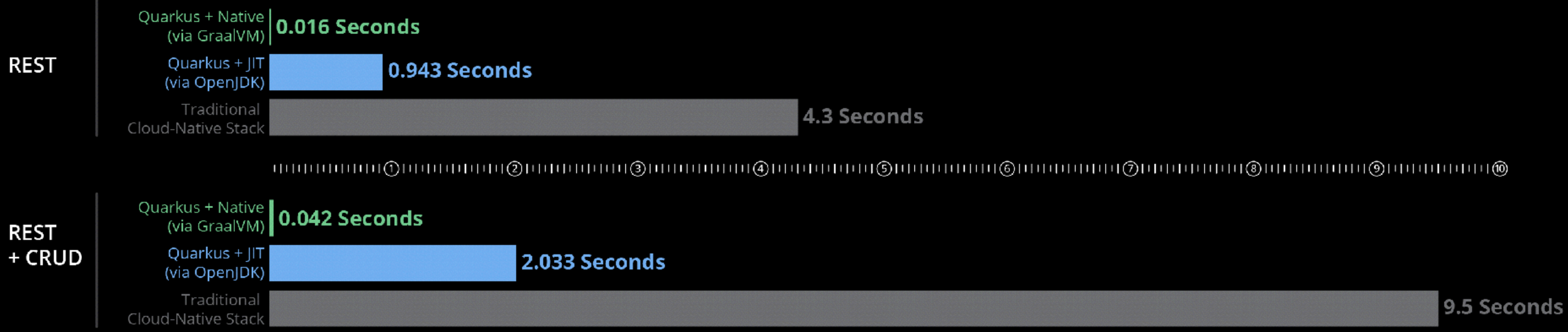
QUARKUS

Memory (RSS) in Megabytes*

*Tested on a single-core machine

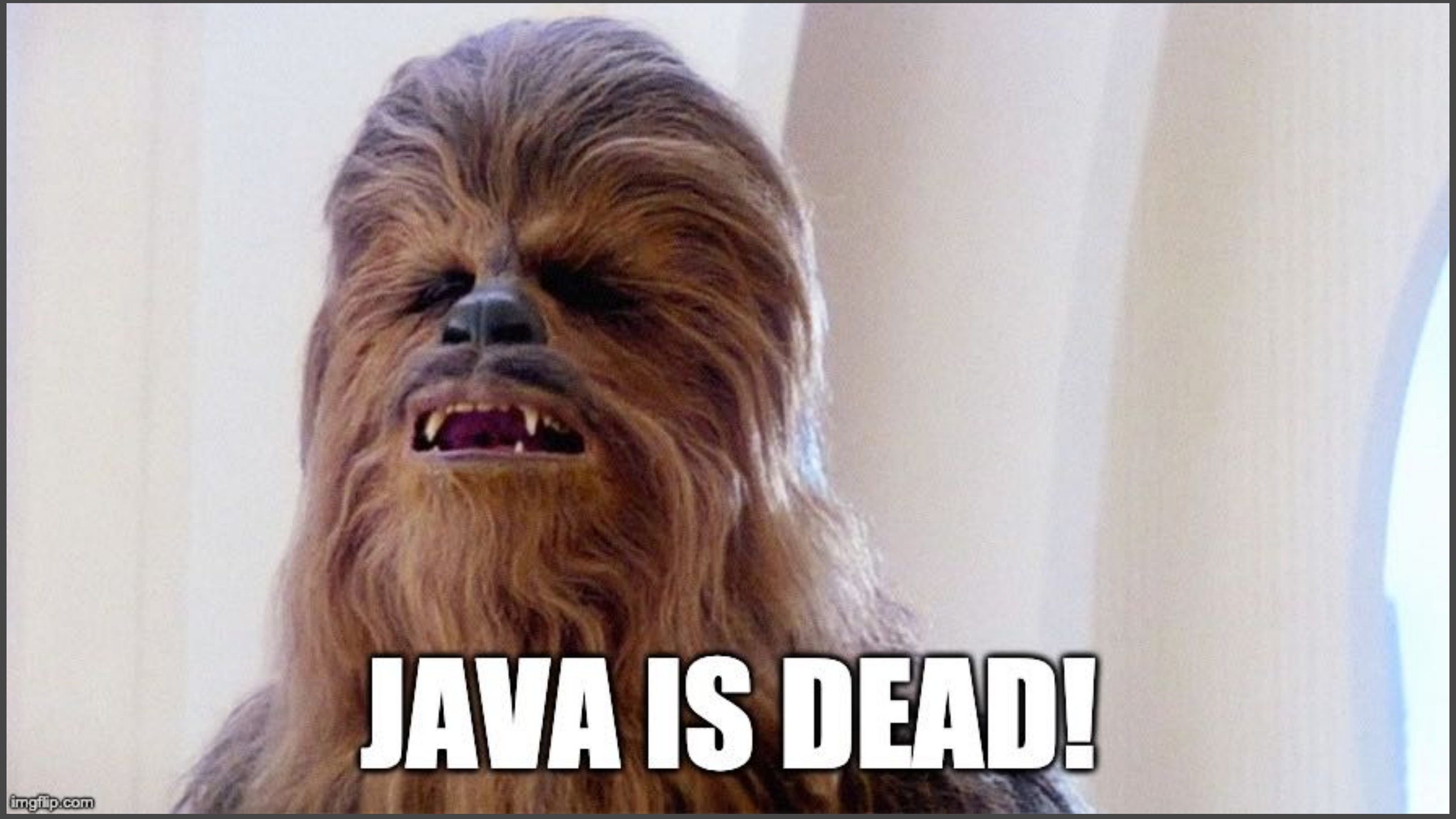


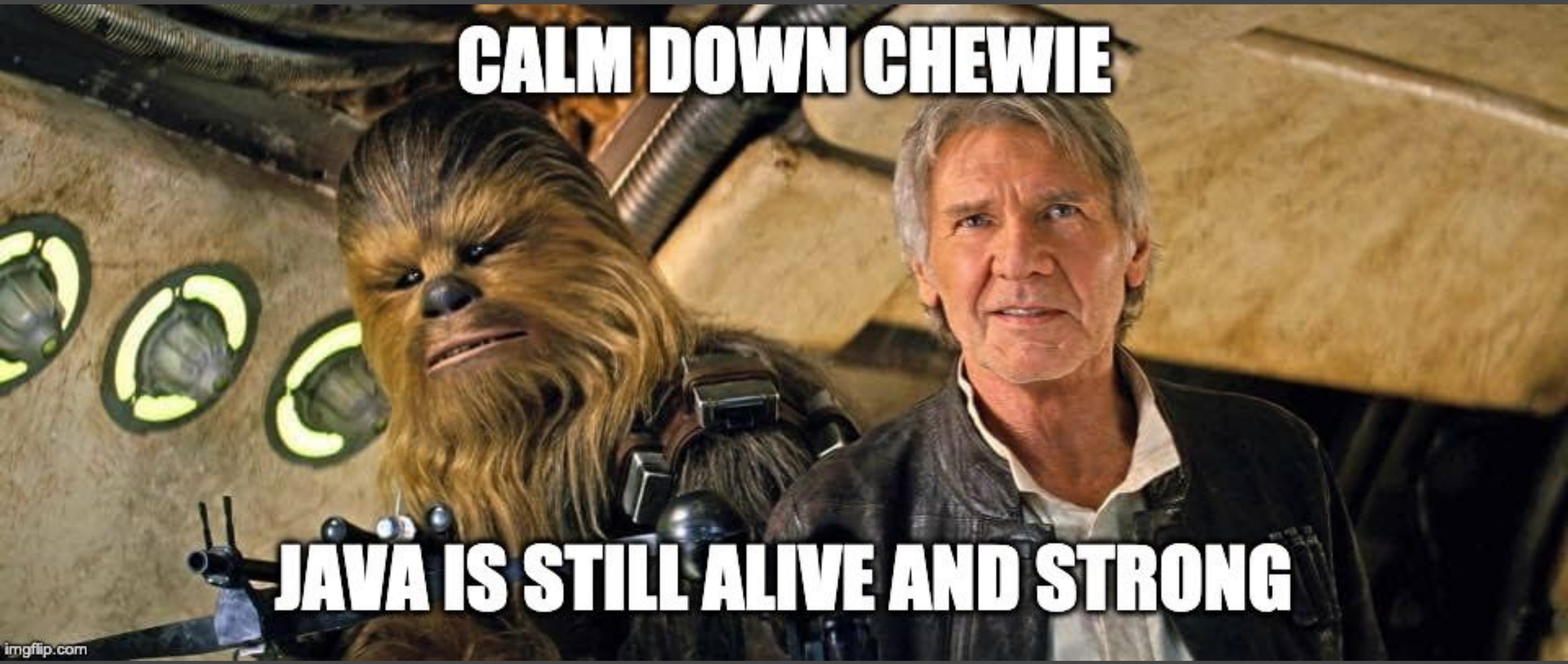
BOOT + First Response Time



@PRPATEL


RT.X
REST
Eas


JAVA IS DEAD!
imgflip.com


CALM DOWN CHEWIE
JAVA IS STILL ALIVE AND STRONG
imgflip.com

READY TO EXPLORE QUARKUS?

WORKSHOP SETUP (GET STARTED NOW)

Step 1: Lab repo (please clone AND open in browser):
https://github.com/prpatel/quarkus-workshop

Step 2: Make sure you have Java 8, Docker, IDE installed

Step 3: Check your env:
java -version (should be Java 8)
mvn -version (recent version)

Step 4: Open cloned project in IDE

HAVE QUESTIONS NOW OR WHILE WORKING ON THE WORKSHOP?

PLEASE ASK!

FIRST STEPS WITH QUARKUS

@PRPATEL

TIME FOR A DEMO!

@prpatel

EXERCISE 1

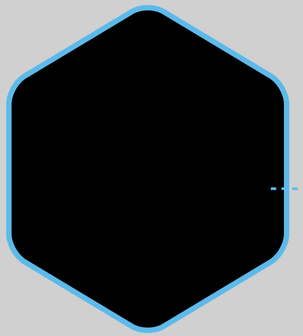
* Lab repo (please clone AND open in browser):
https://github.com/prpatel/quarkus-workshop

* exercise1/ folder

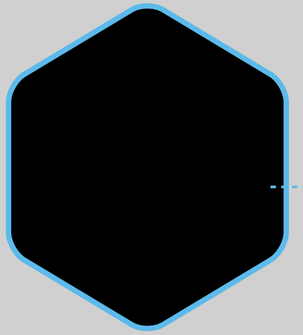
QUESTIONS?

BUILD A SIMPLE ENDPOINTS USING ANNOTATIONS AND RESTEASY

@PRPATEL

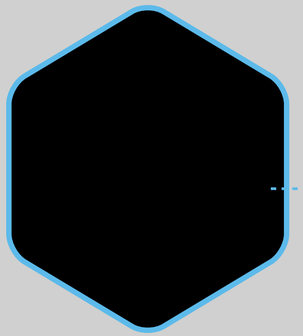


- CDI, JPA, JAX-RS, JTA

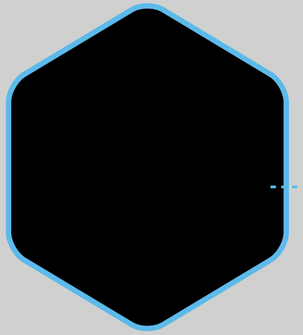


- BASED ON STANDARDS

QUARKUS - STANDARDS



RESTEASY



JSON-B

QUARKUS - LIBS

TIME FOR A DEMO!

@prpatel

EXERCISE 2

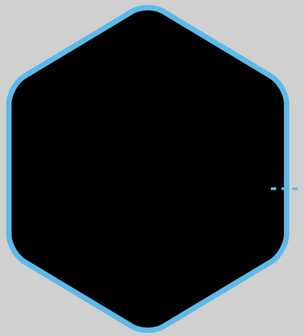
* Lab repo (please clone AND open in browser):
https://github.com/prpatel/quarkus-workshop

* exercise2/ folder

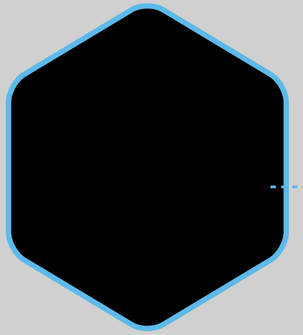
QUESTIONS?

BUILD FULL REST ENDPOINTS

@PRPATEL

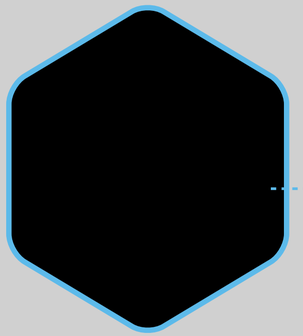


CDI, JPA, JAX-RS, JTA

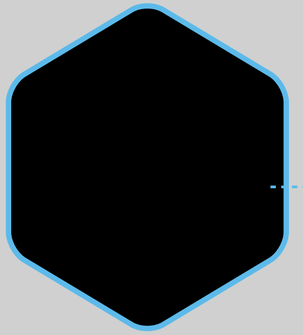


ANNOTATIONS

QUARKUS - ANNOS



JAVAX.WS.RS.CORE .RESPONSE



STANDARD LIBS

QUARKUS - LIBS

HOW ARE OBJECT SERIALIZED TO/FROM JSON?

@prpatel

TIME FOR A DEMO!

@prpatel

EXERCISE 3

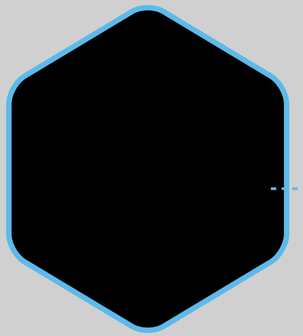
* Lab repo (please clone AND open in browser):
https://github.com/prpatel/quarkus-workshop

* exercise3/ folder

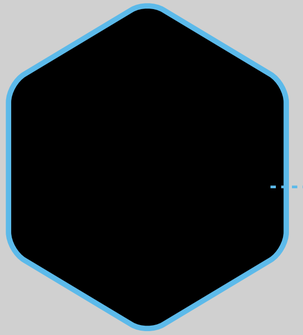
QUESTIONS?

QUARKUS MAKES TESTING CODE EASY

@PRPATEL

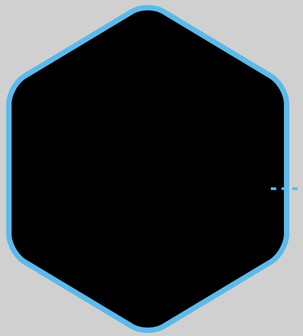


- REST-ASSURED

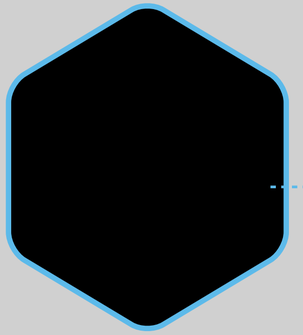


- JUNIT 5 (CAN USE 4)

QUARKUS - TESTING



QUARKUS RUNNING IN BACKGROUND



@QUARKUSTEST

QUARKUS - TESTING

TIME FOR A DEMO!

@prpatel

EXERCISE 4

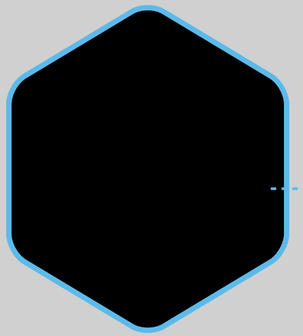
* Lab repo (please clone AND open in browser):
https://github.com/prpatel/quarkus-workshop

* exercise4/ folder

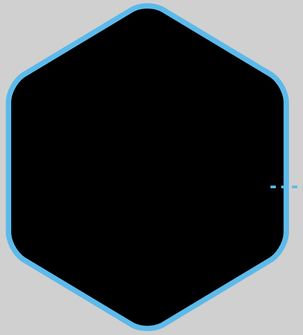
QUESTIONS?

QUARKUS DATABASE ACCESS WITH PANACHE

@PRPATEL

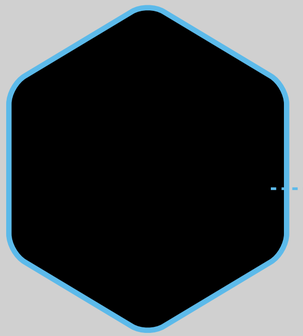


COMPACT

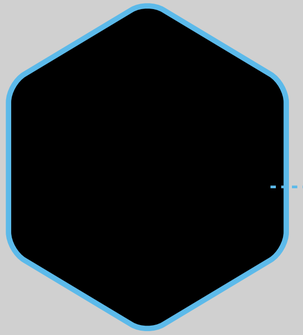


READABLE

QUARKUS - PANACHE

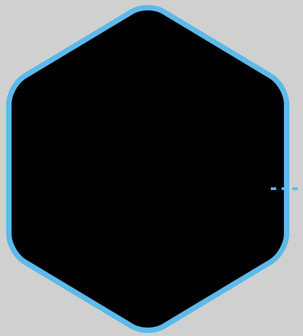


LESS BOILERPLATE

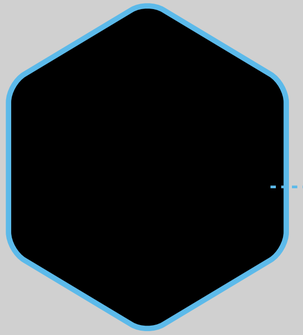


LESS ‘HELPER’ OBJECTS

QUARKUS - PANACHE

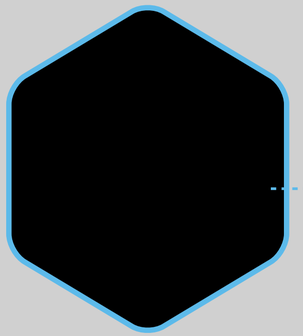


- BASED ON HIBERNATE

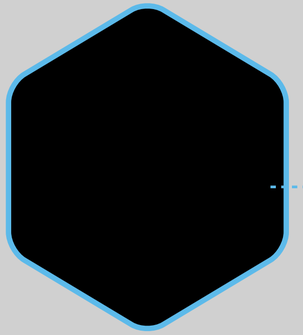


- SUPPORTS EVERYTHING

QUARKUS - PANACHE



CAN DO NORMAL JPA



USEFUL FOR MIGRATING EXISTING APP

QUARKUS - JPA

TIME FOR A DEMO!

@prpatel

EXERCISE 5

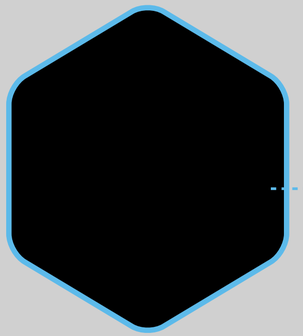
* Lab repo (please clone AND open in browser):
https://github.com/prpatel/quarkus-workshop

* exercise5/ folder

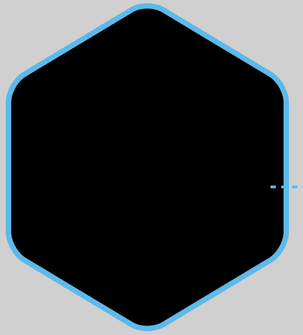
QUESTIONS?

QUARKUS JPA / PANACHE TESTING WITH TESTCONTAINERS

@PRPATEL

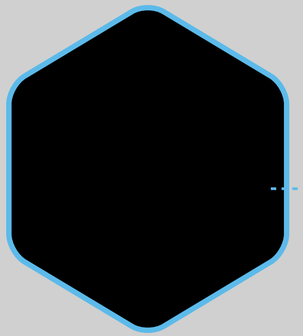


SERVICES WITH TIGHT JUNIT INTEGRATION

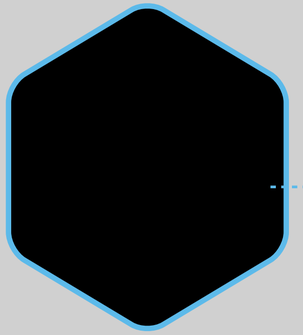


POSTGRES, KAFKA, ETC

TEST CONTAINERS



USES DOCKER



CONFIGURE USING CLASSES OR YML

TESTCONTAINERS

TIME FOR A DEMO!

@prpatel

EXERCISE 6

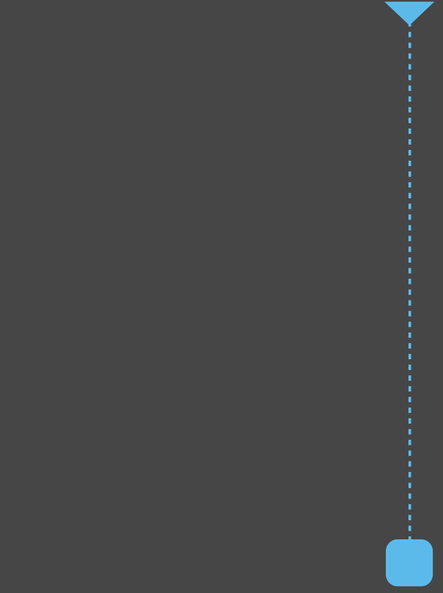
* Lab repo (please clone AND open in browser):
https://github.com/prpatel/quarkus-workshop

* exercise6/ folder

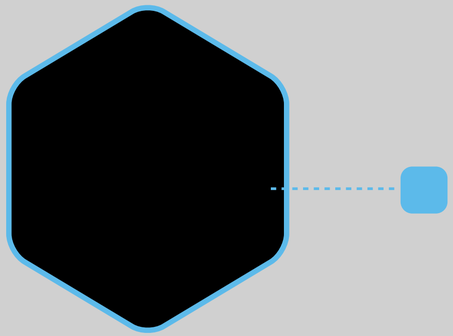
QUESTIONS?

QUARKUS REACTIVE MESSAGING

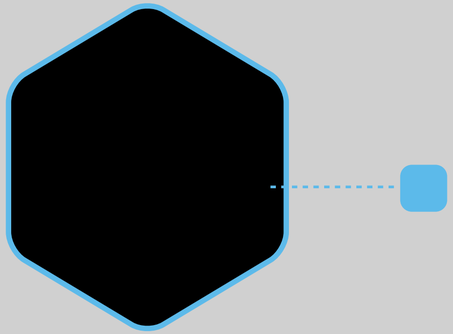
@PRPATEL



REMEMBER THAT QUARKUS HAS REACTIVE BAKED IN!

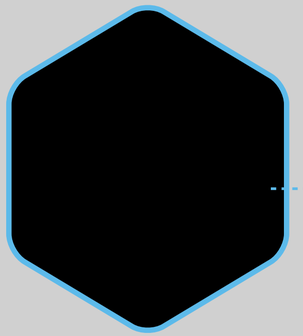


VERT.X UNDER THE HOOD

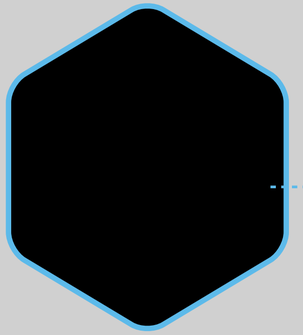


REACTIVE DRIVERS FOR POSTGRES, KAFKA

QUARKUS - REACTIVE



EASY INTEGRATION WITH KAFKA



MICROPROFILE REACTIVE MESSAGING

QUARKUS - KAFKA

TIME FOR A DEMO!

@prpatel

EXERCISE 7

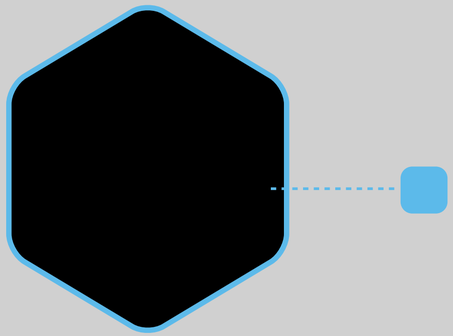
* Lab repo (please clone AND open in browser):
https://github.com/prpatel/quarkus-workshop

* exercise7/ folder

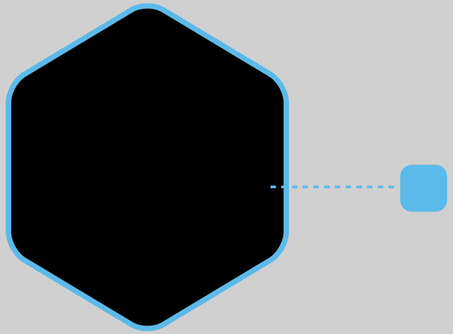
QUESTIONS?

DEPLOYING QUARKUS TO A CLOUD FUNCTION PROVIDER

@PRPATEL

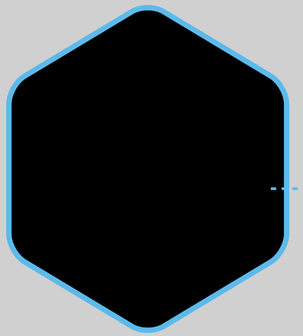


SERVERLESS CLOUD FUNCTIONS

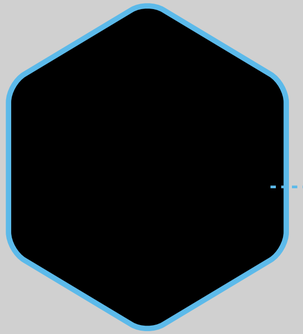


AMAZON, MICROSOFT, IBM CLOUD FNS

DEPLOYING TO CLOUD

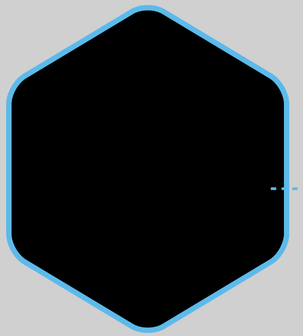


APACHE OPENWHISK - 100% OPEN SOURCE

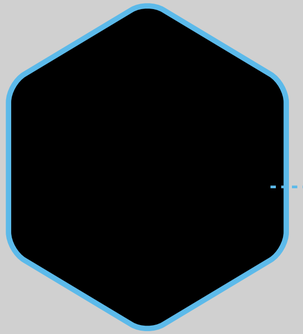


IBM CLOUD FUNCTIONS

DEPLOYING TO CLOUD



SIMILAR TO CLOUD FNS - ELASTIC SCALING



LOWER MEMORY AVAILABLE

KUBERNETES

TIME FOR A DEMO!

@prpatel

EXERCISE 8

* Lab repo (please clone AND open in browser):
https://github.com/prpatel/quarkus-workshop

* exercise8/ folder

QUESTIONS?

WRAP UP

QUARKUS

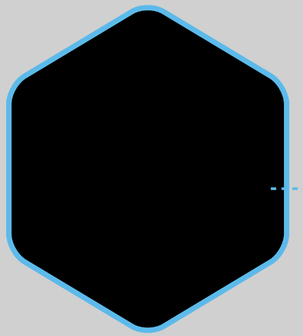
Supersonic Subatomic Java

@prpatel

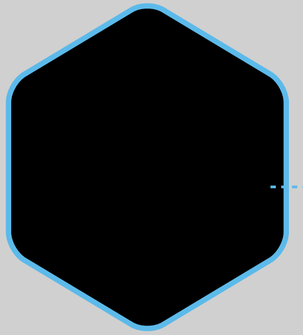
“SERVERLESS JAVA”

FAST STARTUP
LOW FOOTPRINT

@PRPATEL



FASTER STARTUP TIMES

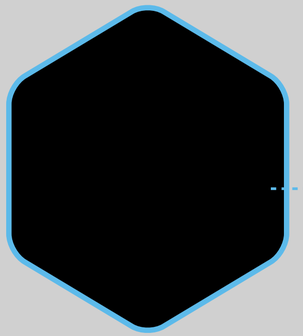


LOWER OVERHEAD

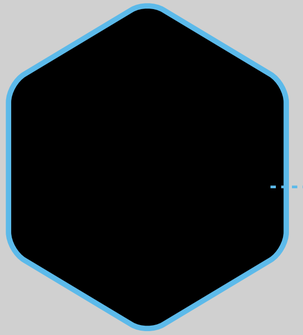
JAVA - NOW SERVERLESS FRIENDLY

CODING THAT SPARKS JOY!

@PRPATEL

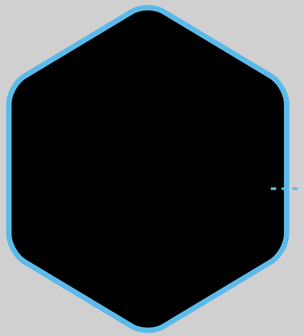


RAPID DEVELOPMENT

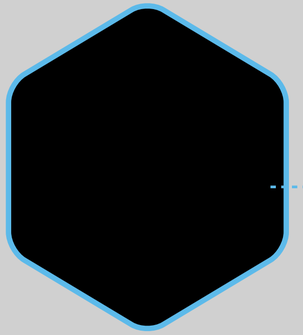


GREAT PRODUCTIVITY

WHY QUARKUS?

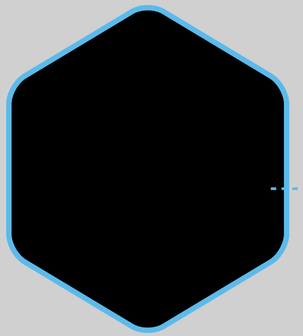


REDUCED BOILERPLATE

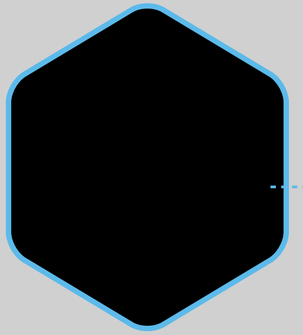


CONVENTION OVER CONFIG

WHY QUARKUS?

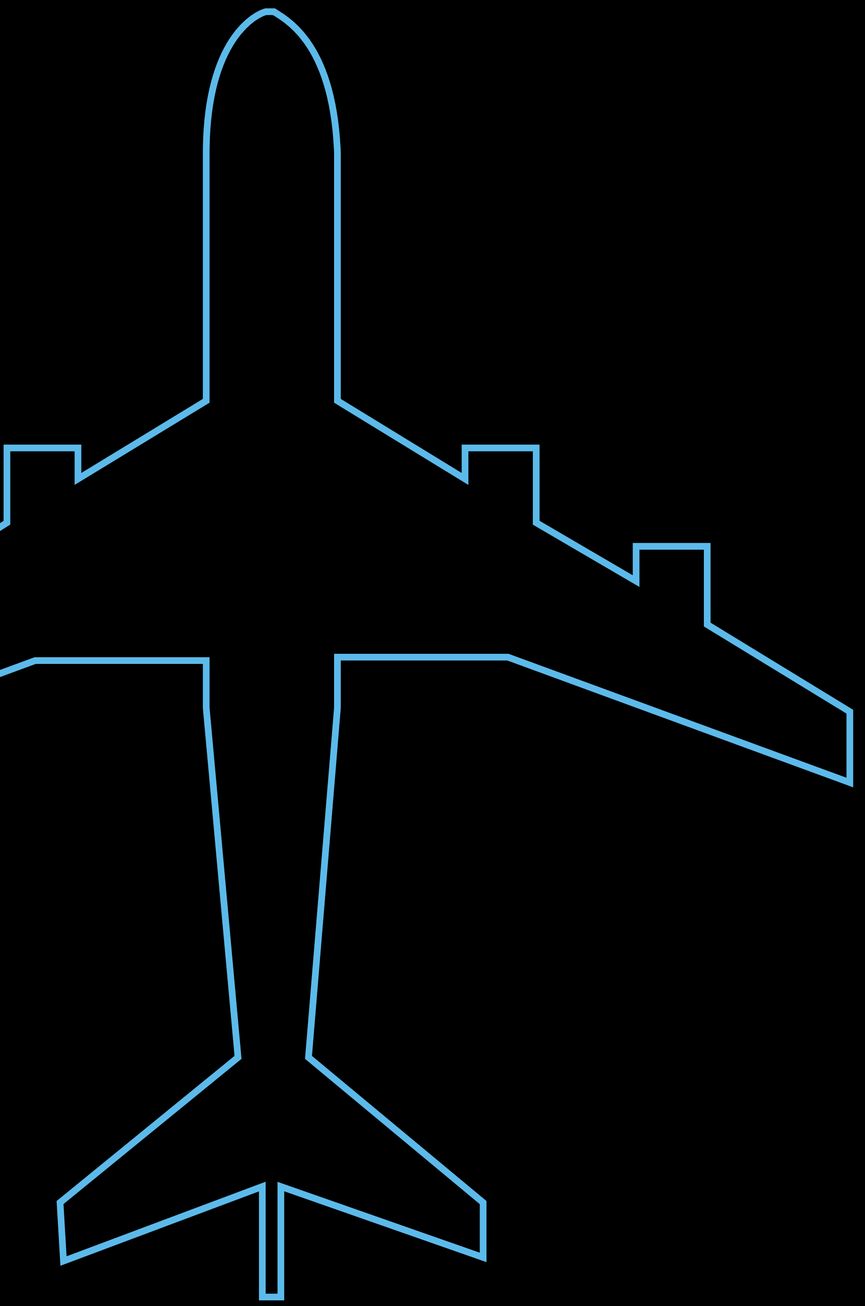


LOW FOOTPRINT



FAST STARTUP

WHY QUARKUS?



THANK YOU

FOLLOW ME ON TWITTER:
@PRPATEL

WANT THIS WORKSHOP IN-PERSON?
GET IN TOUCH!
PRATIK.R.PATEL@GMAIL.COM