

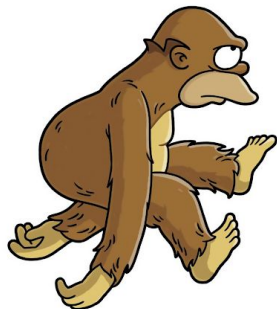
# Spring Boot 2.0: Introduction to Reactive Programming

@rob\_winch

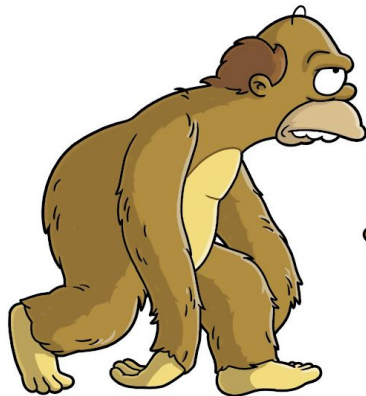
# HOMERSAPIEN



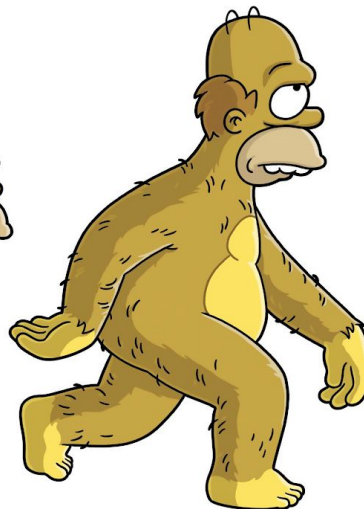
MONKIUS EATALOTIS



CHIMPUS IMBECILUS



APEIS STUPIDIUS



NEANDERSLOB

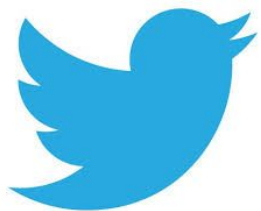


HOMERSAPIEN

MATT GROENING

# Problems with Threading

- Threads waste resources (memory)
- Context Switching Decreases Performance
- Threads underutilize resources (CPU)



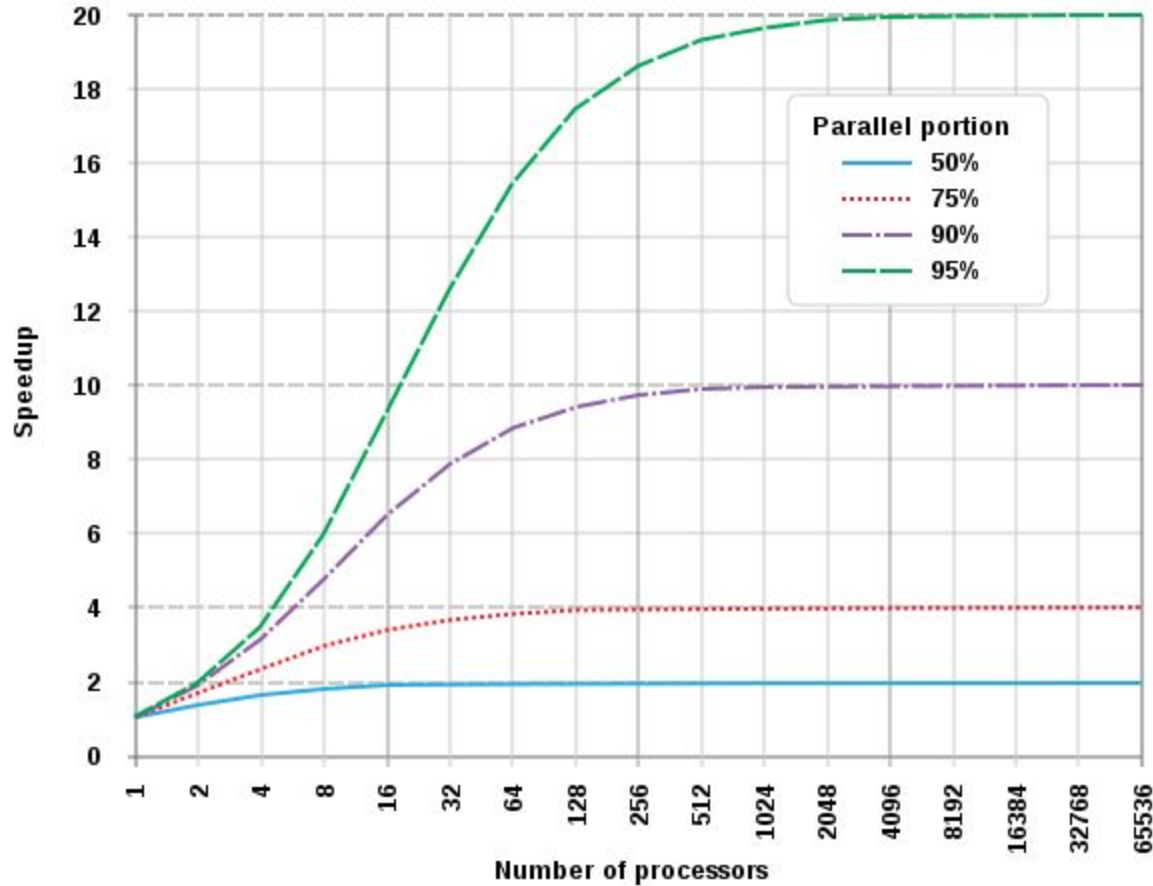
Google Cloud Platform





TOM  
FISH  
BURNE

## Amdahl's Law



Adding **blocking code** will impact the performance of your code

# Monolithic vs Microservices



Monolithic



Microservices



@alvaro\_sanchez

odobo

# What is Reactive?









# Reactive Manifesto

- Responsive
- Resilient
- Elastic
- Message Driven



# Reactive Streams



## Subscription



  request(long) void



  cancel() void

## Subscriber



  onSubscribe(Subscription) void

  onNext(T) void

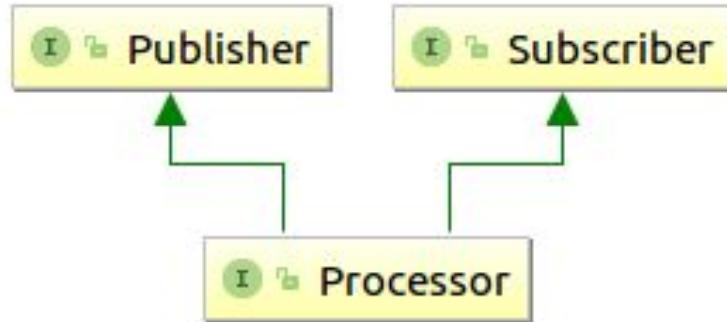
  onError(Throwable) void

  onComplete() void

## Publisher

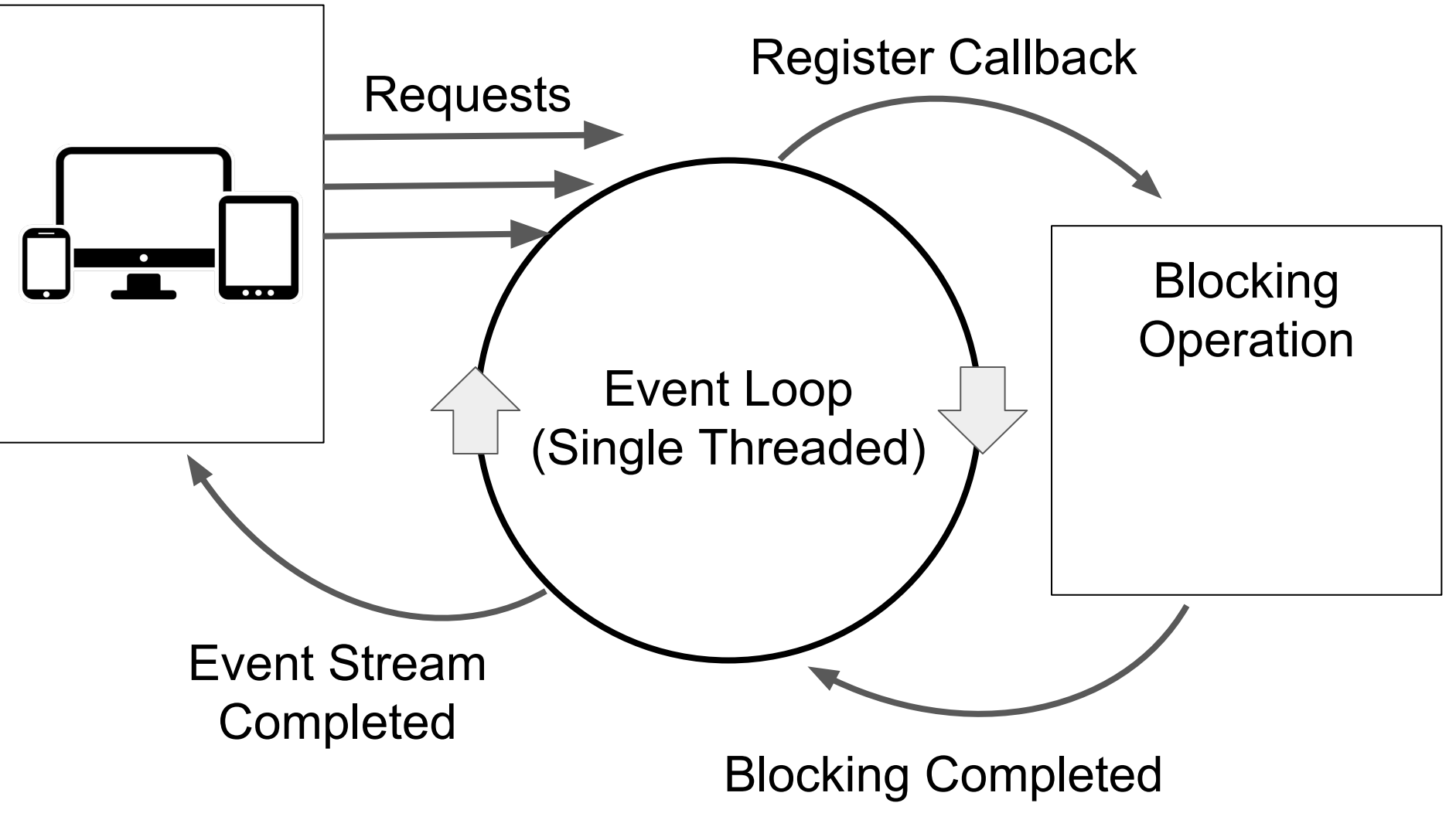
  subscribe(Subscriber<? super T>) void

# Reactive Streams



# Reactive Streams Implementations

- Akka
- MonogoDB
- Ratpack
- Reactive Rabbit
- Reactor
- RxJava
- Slick
- Vert.x



Demo