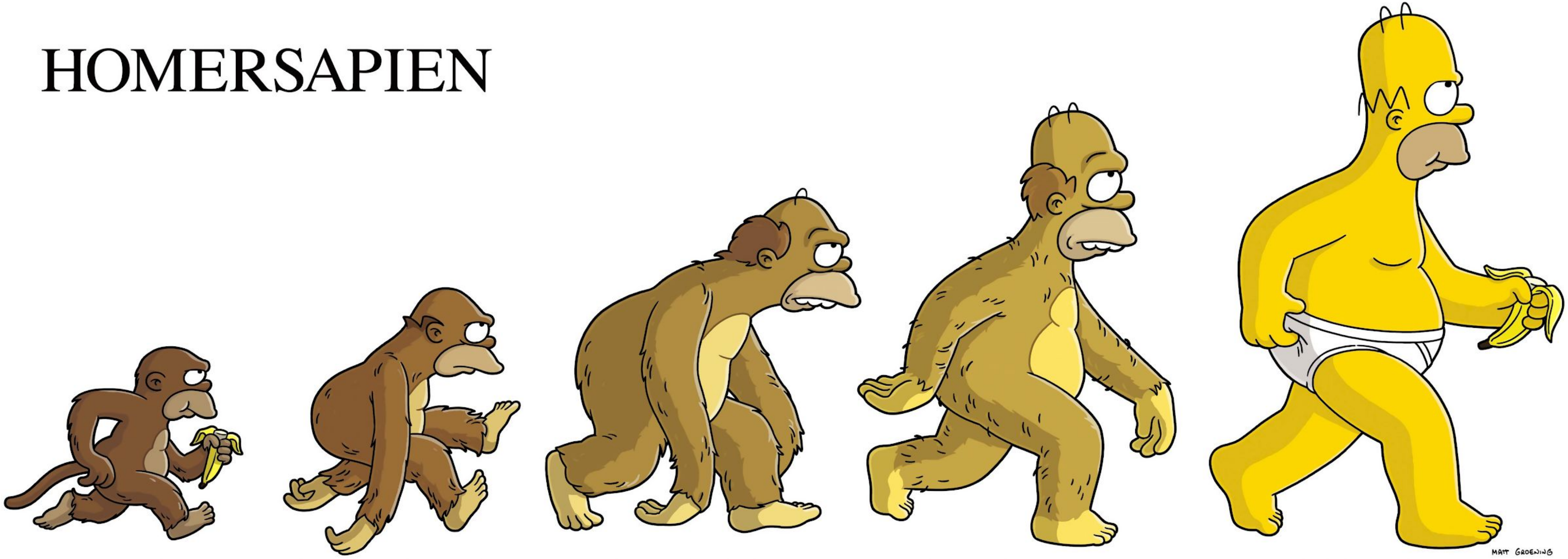
Spring Boot 2.0: Introduction to Reactive Programming

@rob_winch

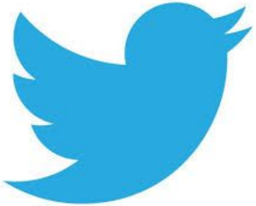
HOMERSAPIEN



MONKIUS EATALOTIS CHIMPUS IMBECILUS APEIS STUPIDIUS NEANDERSLOB HOMERSAPIEN

Problems with Threading

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



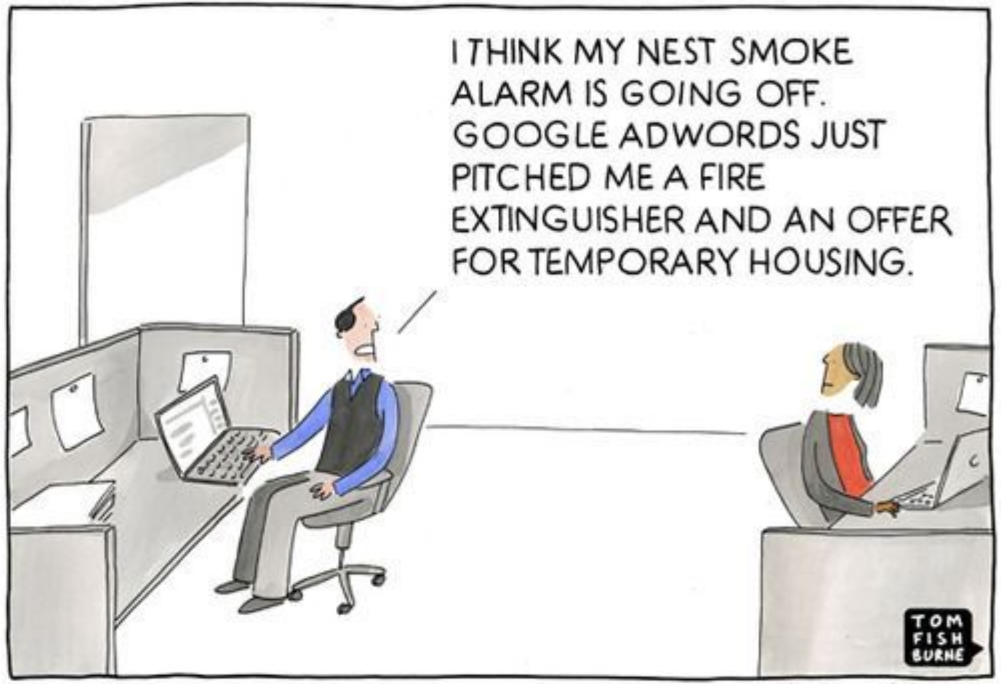
NETFLIX

f

Google Cloud Platform

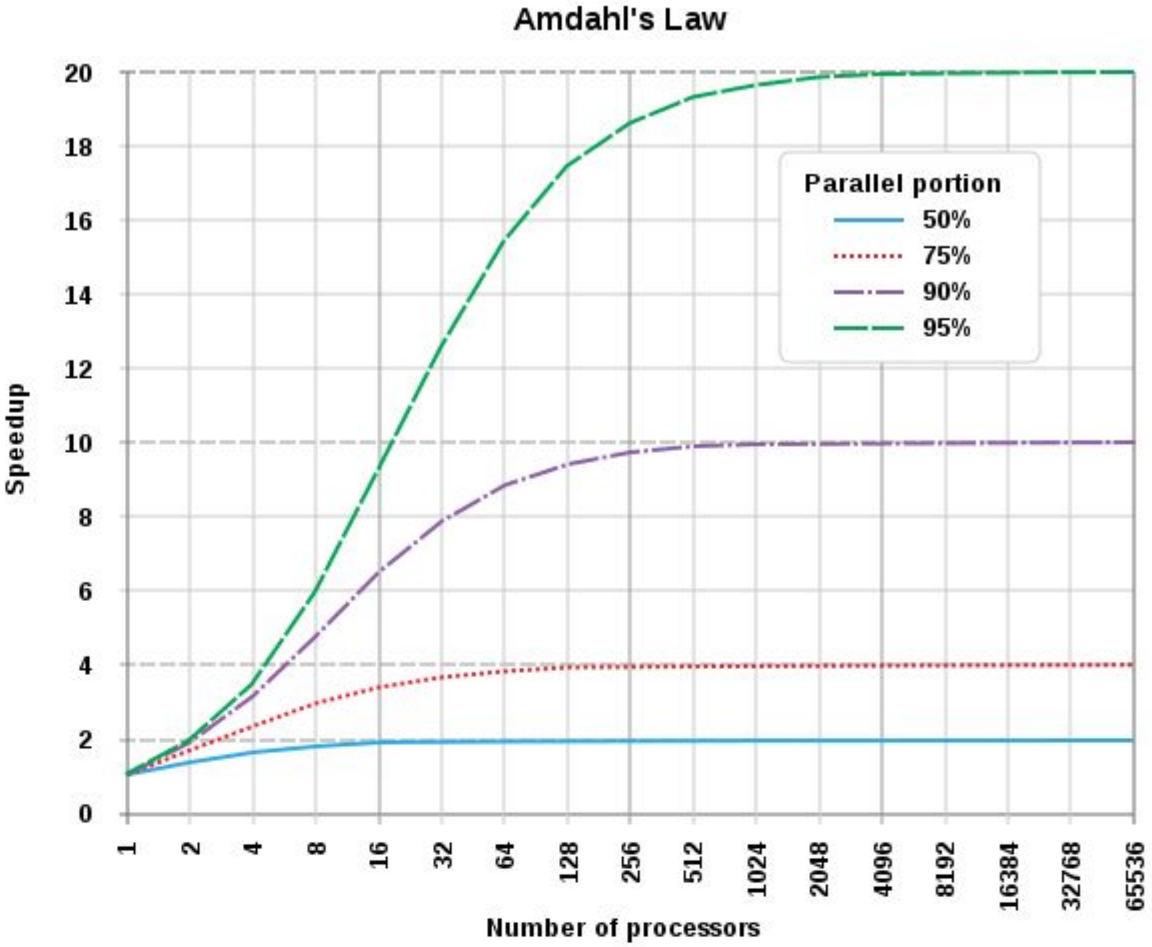
aws

Azure



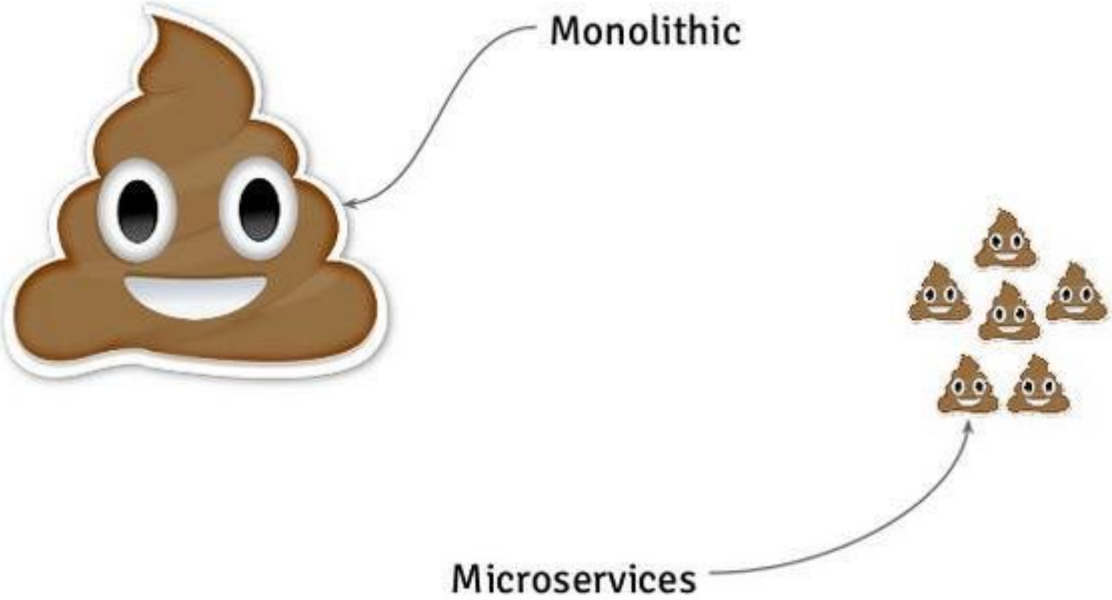
I THINK MY NEST SMOKE ALARM IS GOING OFF. GOOGLE ADWORDS JUST PITCHED ME A FIRE EXTINGUISHER AND AN OFFER FOR TEMPORARY HOUSING.
TOM FISH BURNE

© marketoonist.com



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

Monolithic vs Microservices



Monolithic

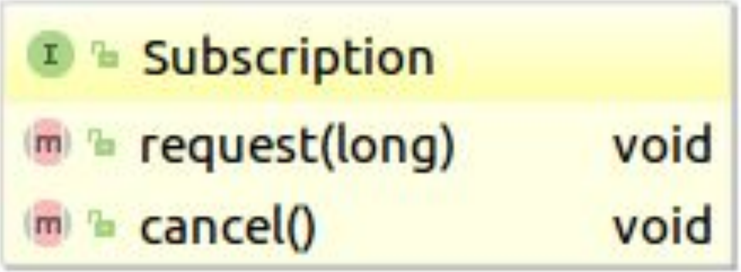
Microservices

What is Reactive?

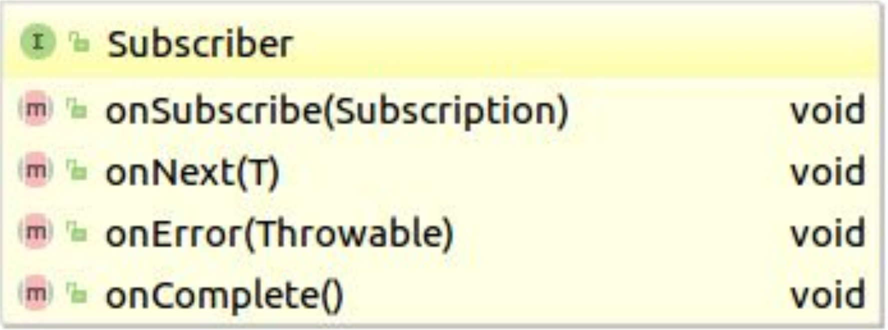
Reactive Manifesto

- Responsive
- Resilient
- Elastic
- Message Driven

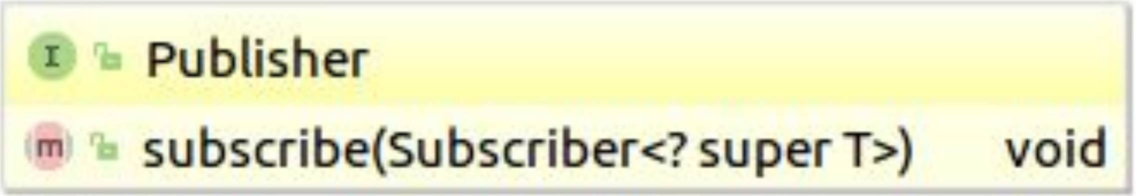
Reactive Streams



Subscription	
request(long)	void
cancel()	void

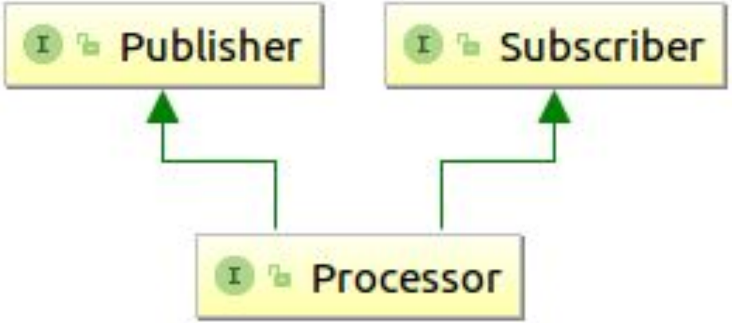


Subscriber	
onSubscribe(Subscription)	void
onNext(T)	void
onError(Throwable)	void
onComplete()	void



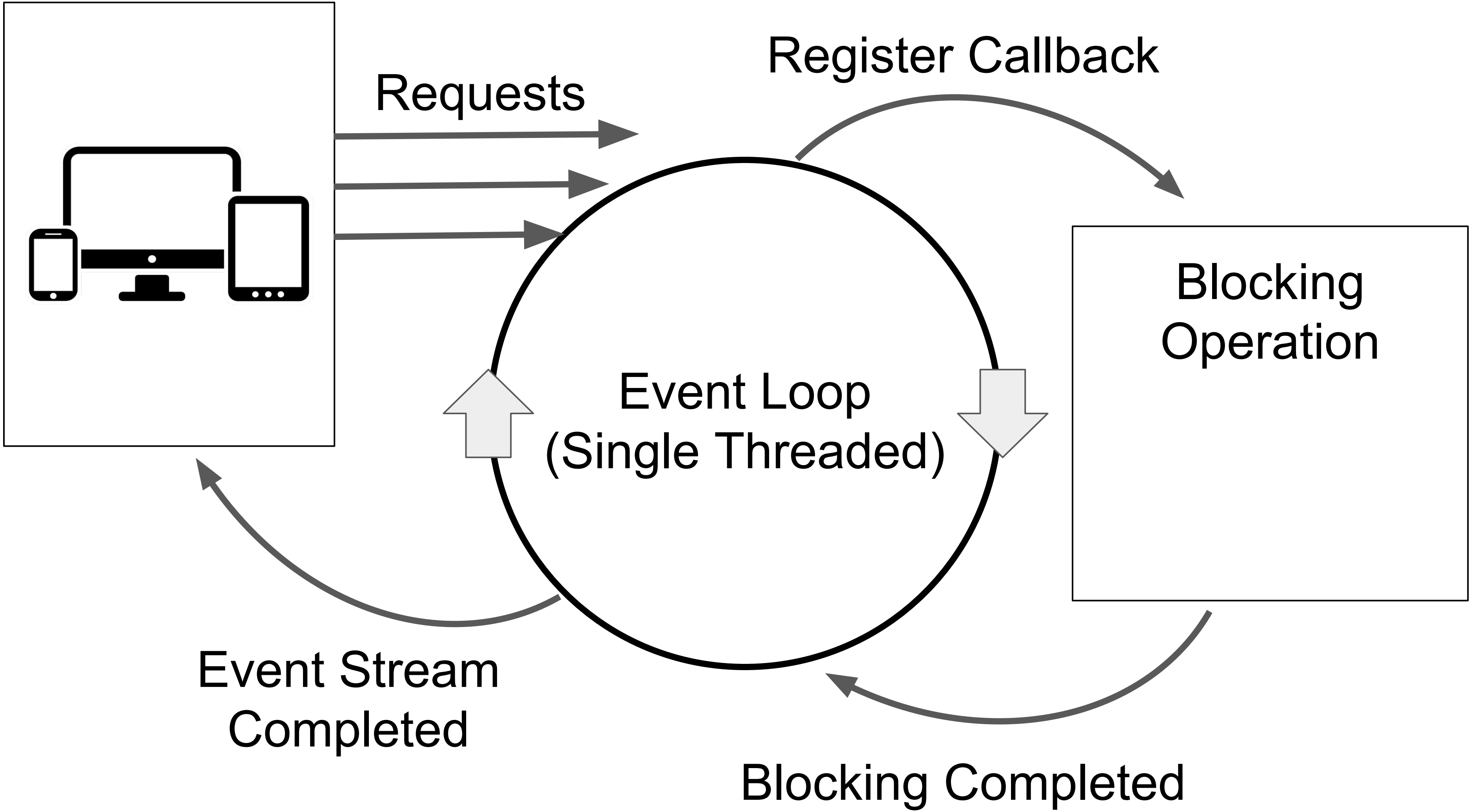
Publisher	
subscribe(Subscriber<? super T>)	void

Reactive Streams



Reactive Streams Implementations

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


Register Callback
Requests
Blocking Operation
Event Loop (Single Threaded)
Event Stream Completed
Blocking Completed

Demo

https://github.com/rwinch/spring-reactive-intro