# Taking Spring Boot Reactive Apps for a Spin on Azure

Mark Heckler
Principal Technologist, Spring Developer Advocate
www.thehecklers.com
mark@thehecklers.com
mheckler@pivotal.io
@mkheck

with my buddy Bruno Borges!

Pivotal

# Who am I?

- Author
- Speaker
- Architect & Developer
- Java Champion
- Seeker of a better way

“In a nutshell reactive programming is about
**non-blocking, event-driven applications** that
**scale with a small number of threads** with
**backpressure as a key ingredient** that
aims to ensure producers
do not overwhelm consumers.”

*–Rossen Stoyanchev, Project Reactor team*

# Reactive Streams: 4 interfaces

- Publisher<T>
- Subscriber<T>
- Subscription
- Processor<T,R>

# Project Reactor: a quick overview

## REACTIVE CORE

Reactor is a **fully non-blocking** foundation with efficient demand management. It directly interacts with Java 8 *functional API*, *Completable Future*, *Stream* and *Duration*.

## TYPED [0|1|N] SEQUENCES

Reactor offers **2 reactive composable API** Flux [N] and Mono [0|1] extensively implementing Reactive Extensions.


## NON BLOCKING IPC

Suited for **Microservices** Architecture, Reactor IPC offers **backpressure-ready network engines** for HTTP (including Websockets), TCP and UDP. Reactive Encoding/Decoding is fully supported.

# Let’s code!

# Helpful resources

- http://www.reactive-streams.org
- https://projectreactor.io
- https://github.com/mkheck/FSRx
- (LATEST KOTLIN!: https://github.com/mkheck/kotlin-coffee-service)
- (R2DBC WIP!: https://github.com/mkheck/coffee-service-r2dbc)

# Taking it to Azure with Bruno ☁️

# Thanks for coming!

@mkheck
mark@thehecklers.com
mheckler@pivotal.io