Drinking from the Stream
How to use messaging platforms for scalability & performance

Mark Heckler
Professional Problem Solver, Spring Developer & Advocate
www.thehecklers.com
mark@thehecklers.com
mheckler@pivotal.io
@mkheck
Distributed systems are easy...right?
Who am I?

• Author
• Architect & Developer
• Java Champion, Rockstar
• Professional Problem Solver
• Spring Developer & Advocate
• Creador y curador de
New book!
But you can’t buy it yet...

DISCLAIMER: artist’s rendition only, not the real cover

@mkheck www.thehecklers.com

Mark Heckler
Takeaways

- Why use messaging platforms/where do they fit in a distributed architecture?
- Examples of leading messaging platforms
- What is Spring Cloud Stream?
- Why use it?
Takeaways

- Why use messaging platforms/where do they fit in a distributed architecture?
- Examples of leading messaging platforms
- What is Spring Cloud Stream?
- Why use it?
Takeaways

- Why use messaging platforms/where do they fit in a distributed architecture?
- Examples of leading messaging platforms
- **What is Spring Cloud Stream?**
- Why use it?
Takeaways

- Why use messaging platforms/where do they fit in a distributed architecture?
- Examples of leading messaging platforms
- What is Spring Cloud Stream?
- Why use it?
Why use it?

We know where the magazine will be read. We need nets, rope and tranquilizer darts.

I'll have to ask engineering to build one of these space stations...

Phoot

Men
SCSt + \${messaging.platform} =
Scalability
Flexibility

Source → Processor → Processor → Sink → Sink → Sink
Versatility

Source → Processor → Processor → Sink

@mkheck  www.thehecklers.com
Stream Revisited: Legacy

Source → Processor → Sink
Stream Revisited: Evolution

Supplier -> Function -> Consumer
Let’s code!
Resources

- https://github.com/mkheck/drinking-from-the-stream
- https://cloud.spring.io/spring-cloud-stream/
- mark@thehecklers.com, mheckler@pivotal.io
- @mkheck on Twitter