

BUILDING RESILIENT MICROSERVICES with APACHE QPID PROTON

Richard Li

Rafael Schloming

datawire.io

- **MICROSERVICES**
- **DESIGNING MICROSERVICES**
- **DEMO**
- **WRAP UP**



- Release any time
- You're responsible for reliability, availability, scalability, security
- You're also responsible for monitoring, billing, user admin, ...



Idiot proof deploy



Homogenous tech stack

Minimize upgrade frequency



Synchronized release

Easy for vendor to debug



ACID; 1 simultaneous release



Ship as fast as possible



Continuous delivery

Lots of functional
breadth



Design/build in parts

Reliability, availability,
security, scale



Resilient system design

Continuous delivery

Design/build in parts

Resilient system design

Microservices.

Componentization via Services
Organized around Business Capabilities
Products not Projects
Smart endpoints and dumb pipes
Decentralized Governance
Decentralized Data Management
Infrastructure Automation
Design for failure
Evolutionary Design

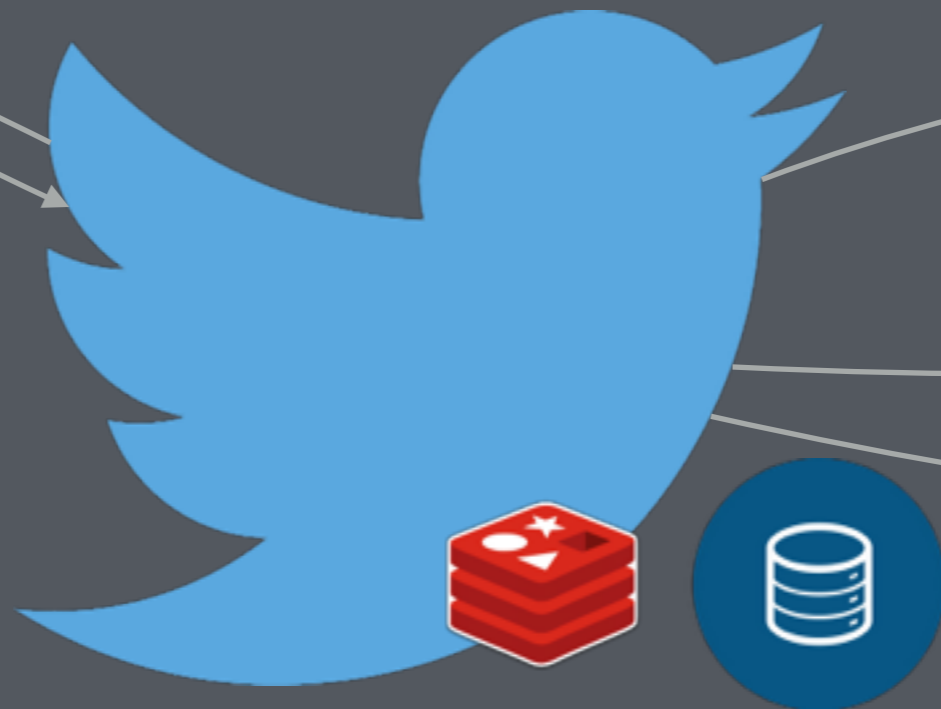


<http://martinfowler.com/articles/microservices.html>

DESIGNING MICROSERVICES

Monolith

1. Send a tweet.



2. Get followers.
3. Publish tweet.



App server

Three App Servers

1. Send a tweet.

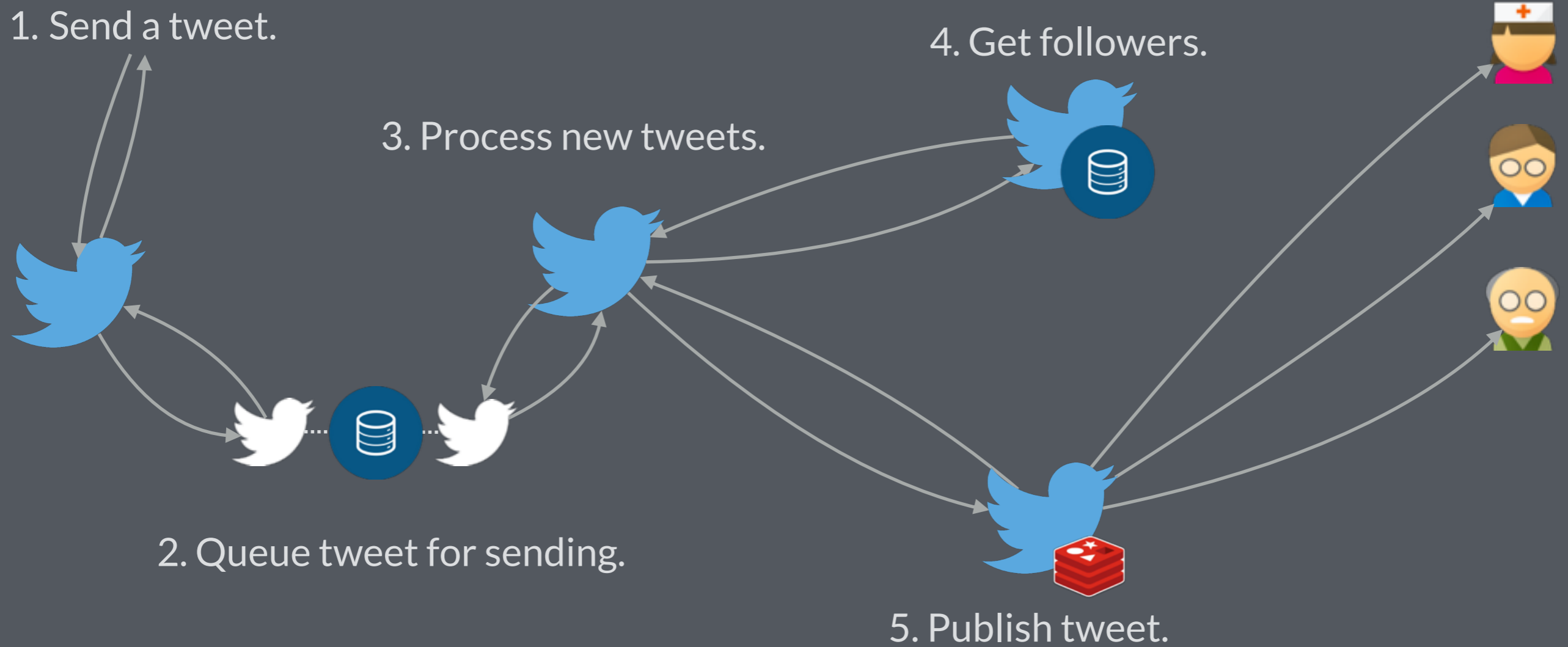
2. Get followers.

3. Publish tweet.



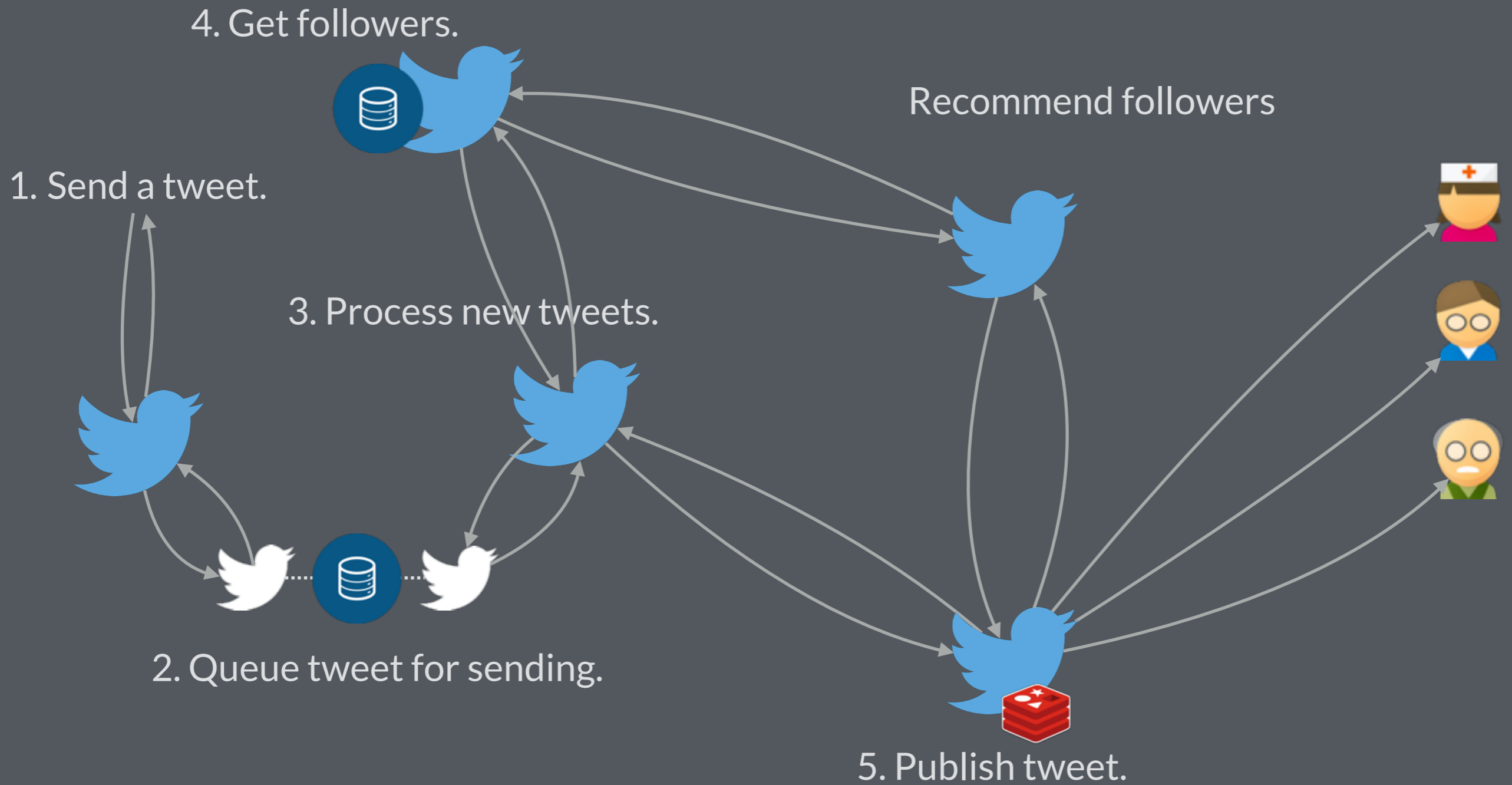
 App server

App Servers + Asynchronous Queue



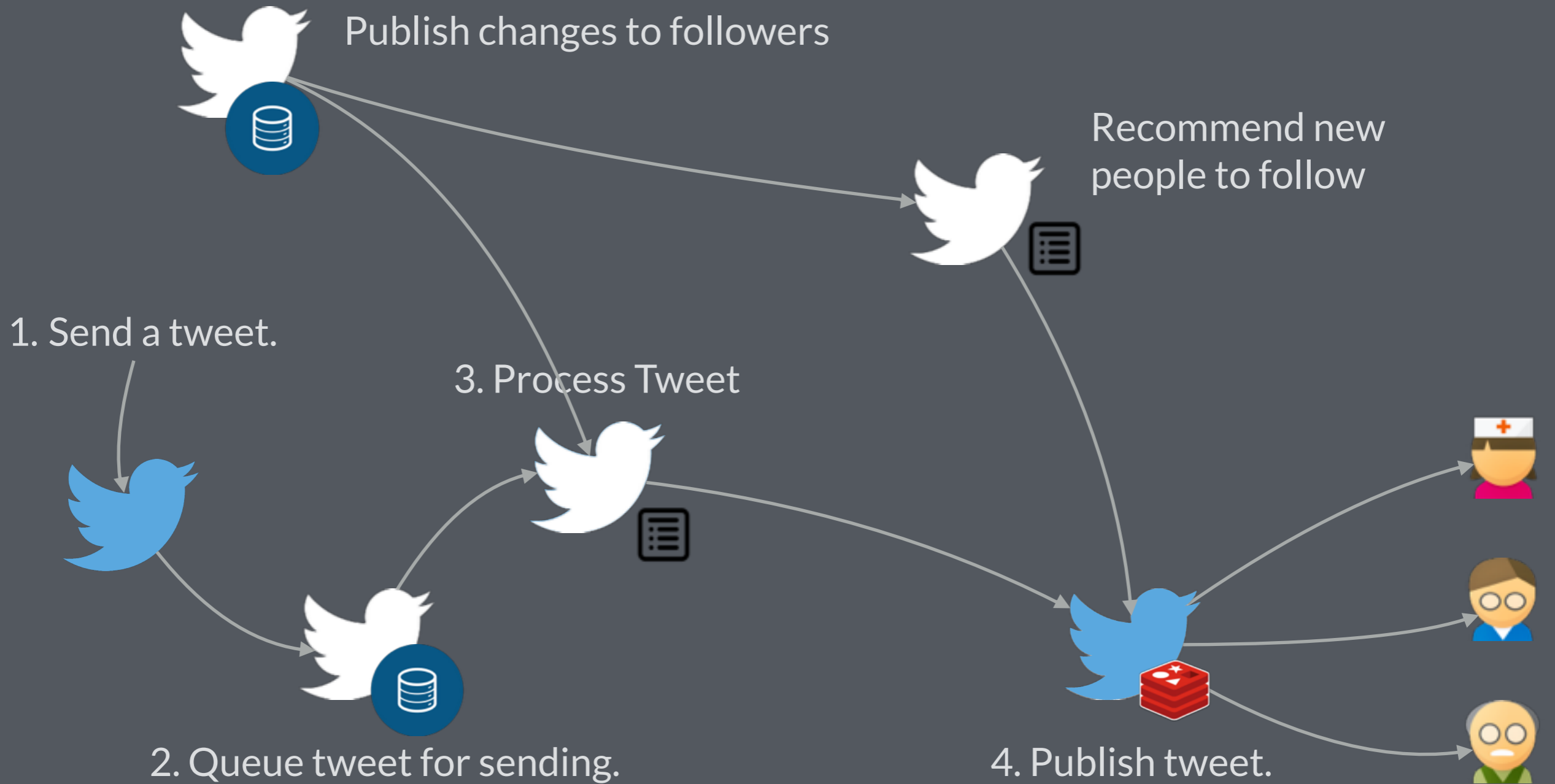
Not a typical app server

App Servers + Asynchronous Queue



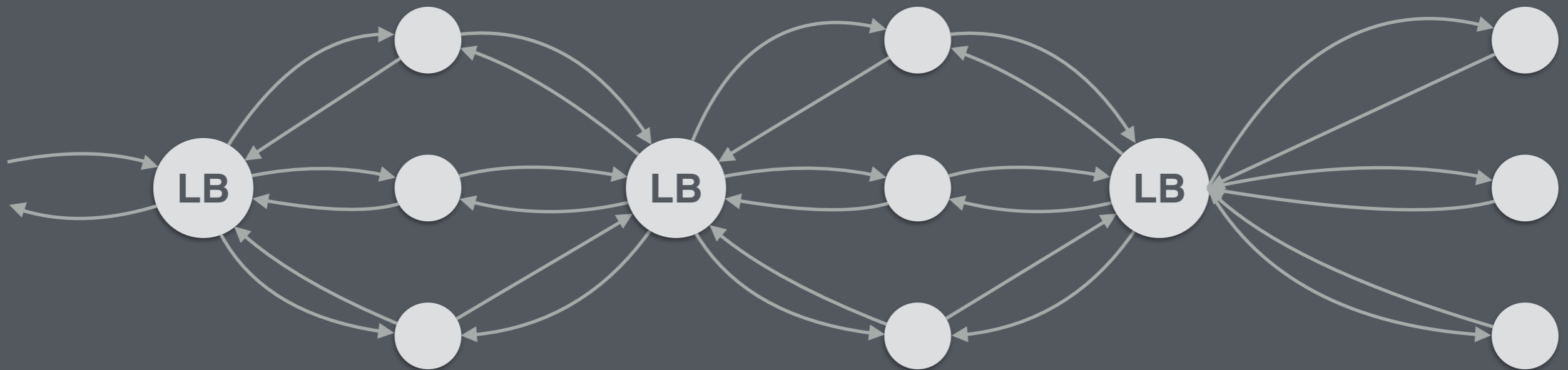
Not a typical app server

Fully Asynchronous

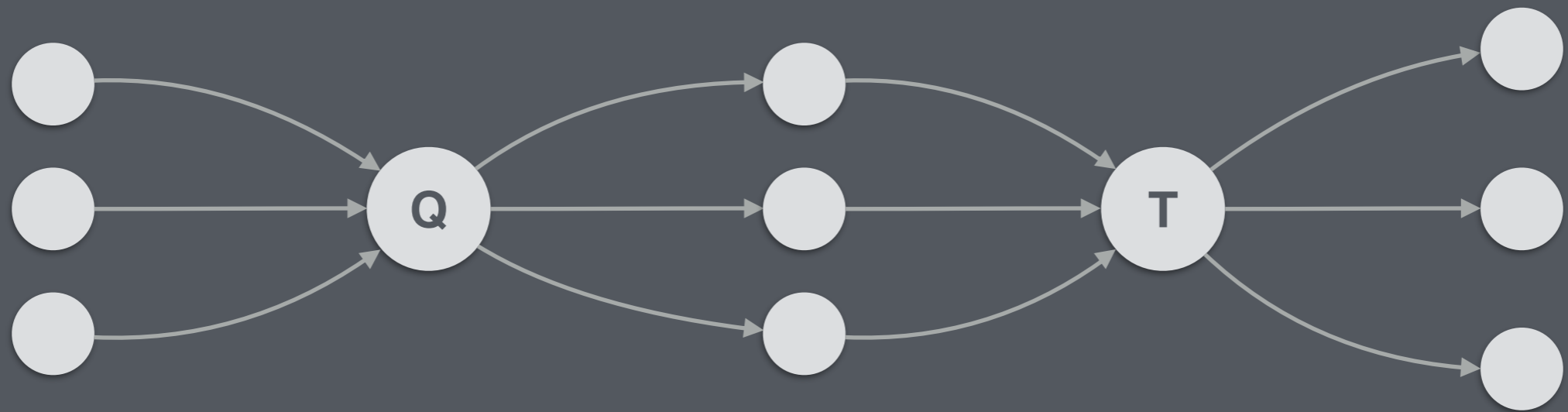


MESSAGING

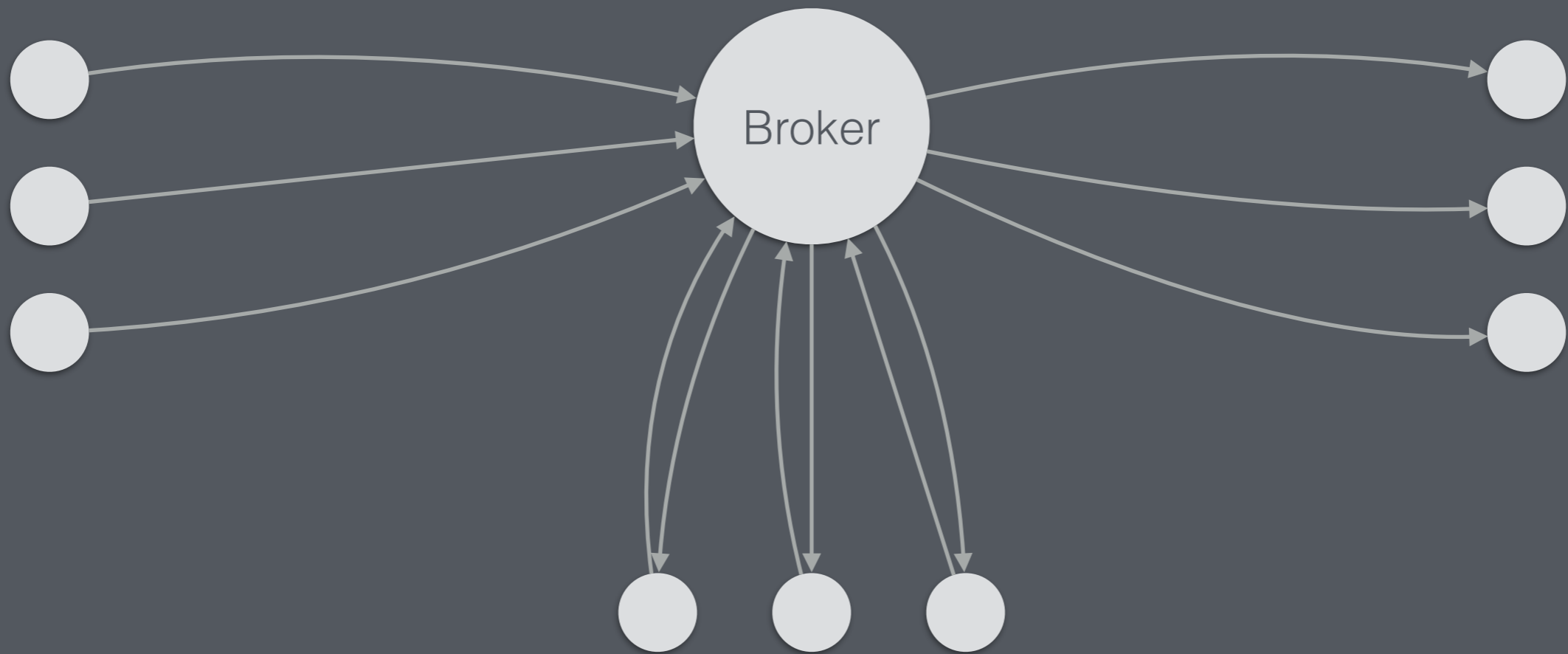
HTTP



Asynchronous Messaging

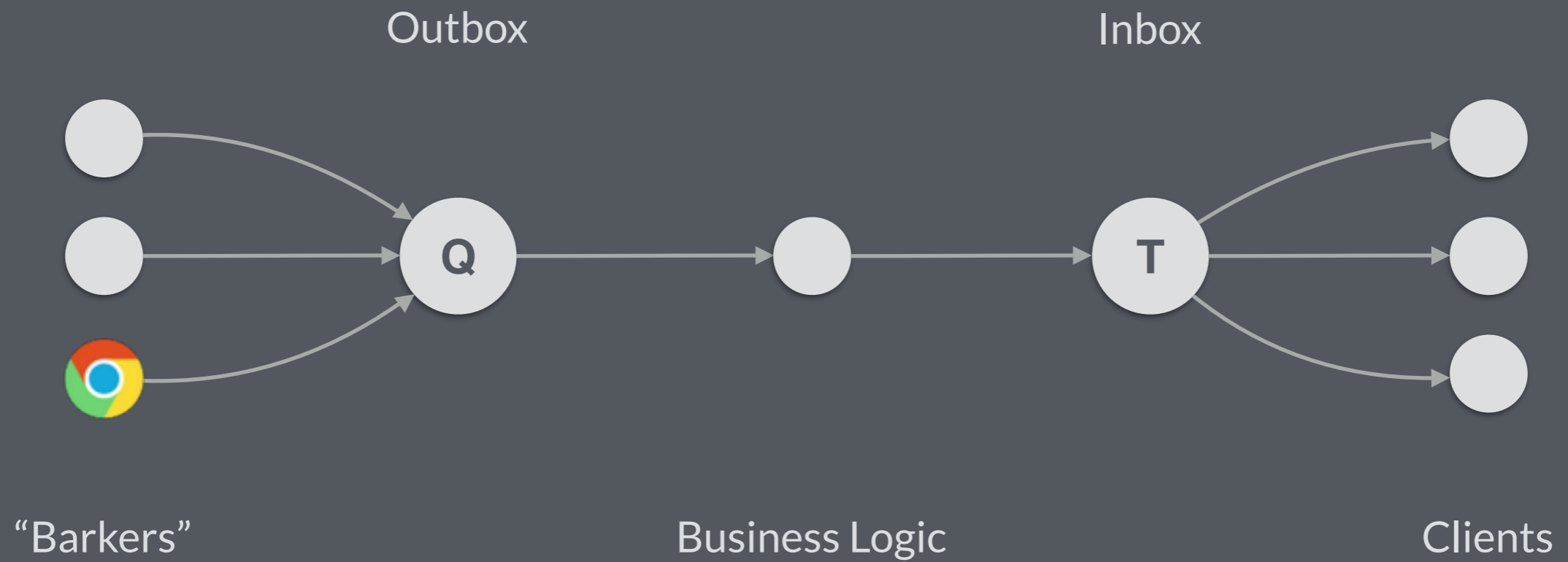


Asynchronous Message Broker



DEMO

Smart endpoints with Proton



Practical Proton Details

- Native AMQP 1.0 protocol engine
- Core engine implemented in C
- Language bindings in Python, JavaScript, Ruby, PHP, Perl, Java, Go, C++
- Part of Apache Qpid
- Used by a number of Qpid projects, including the C++ brokers, JMS Client, Qpid Dispatcher
- Also used by ActiveMQ, HornetQ, Microsoft Azure, IBM MQLite, and many organizations

<http://qpid.apache.org/proton>

SUMMARY

- Microservices are a natural paradigm for cloud-delivered software
- Microservices need to be loosely coupled
- Asynchronous messaging is the key to loose coupling
- Proton provides a simple, powerful async messaging engine

THANK YOU!

richard@datawire.io

rhs@datawire.io

<http://qpuid.apache.org/proton>