Troubleshooting RabbitMQ

and services that use it

Who am I?

- Staff Engineer, RabbitMQ @ Pivotal

Who am I?

- Staff Engineer, RabbitMQ @ Pivotal
- @michaelklishin, github.com/michaelklishin

The monolith problem

Monolith

S1

Monolith

S1

S2

Monolith

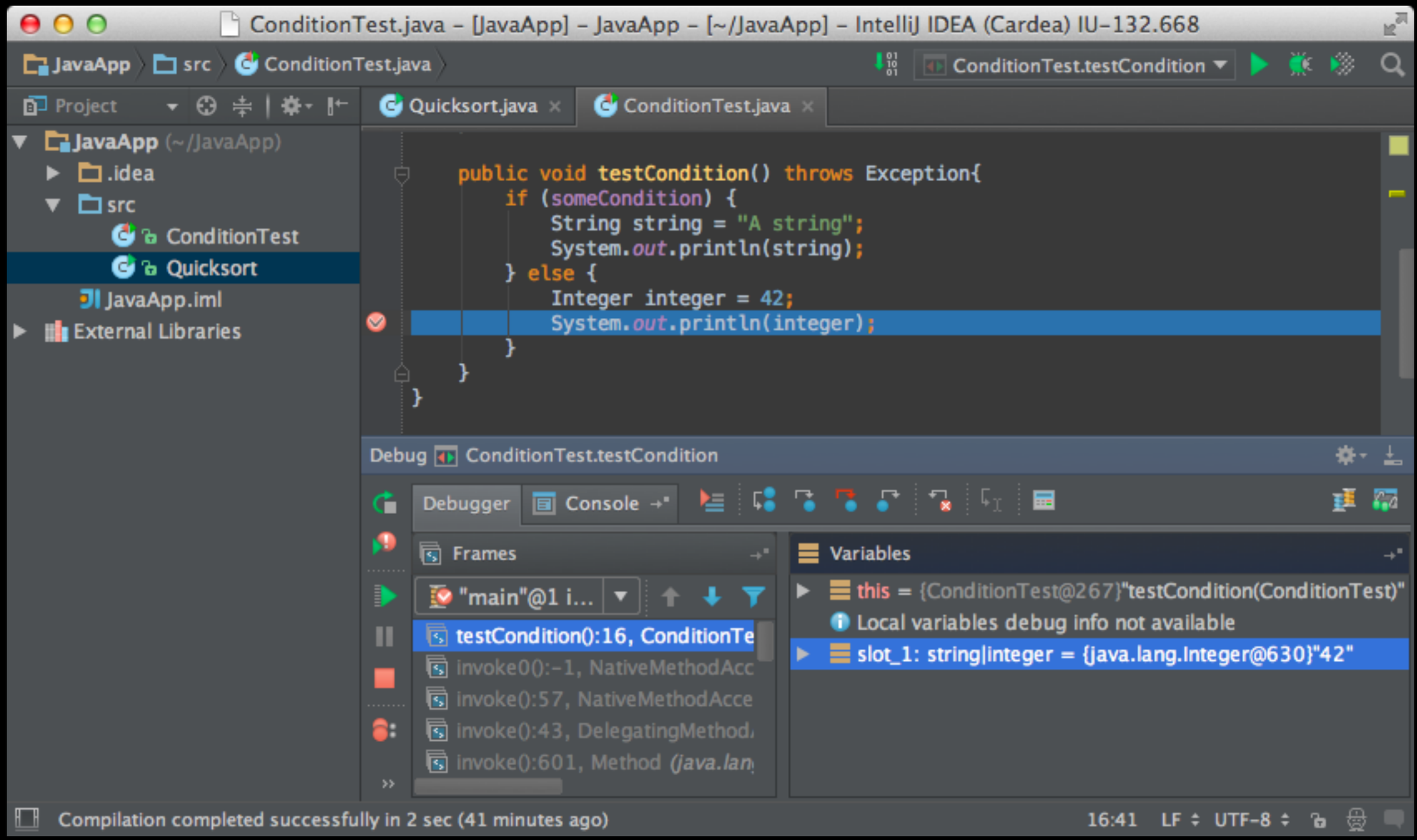
S1

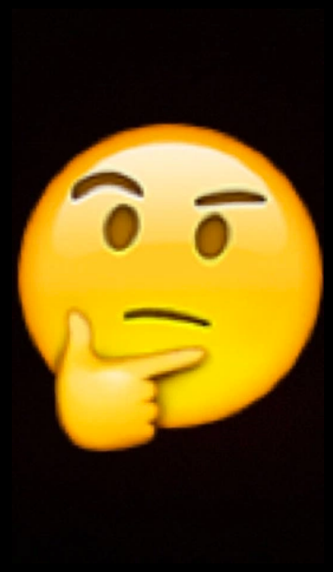
S2

Monolith

S3

S4


ConditionTest.java – [JavaApp] – JavaApp – [~/JavaApp] – IntelliJ IDEA (Cardea) IU-132.668
JavaApp
src
ConditionTest.java
ConditionTest.testCondition
Project
Quicksort.java
ConditionTest.java
JavaApp (~/JavaApp)
.idea
src
ConditionTest
Quicksort
JavaApp.iml
External Libraries
public void testCondition() throws Exception{
if (someCondition) {
String string = "A string";
System.out.println(string);
} else {
Integer integer = 42;
System.out.println(integer);
}
}
}
Debug
ConditionTest.testCondition
Debugger
Console
Frames
"main"@1 i...
testCondition():16, ConditionTe
invoke0():-1, NativeMethodAcc
invoke():57, NativeMethodAcce
invoke():43, DelegatingMethod
invoke():601, Method (java.lan
Variables
this = {ConditionTest@267}"testCondition(ConditionTest)"
Local variables debug info not available
slot_1: string|integer = {java.lang.Integer@630}"42"
Compilation completed successfully in 2 sec (41 minutes ago)
16:41
LF
UTF-8



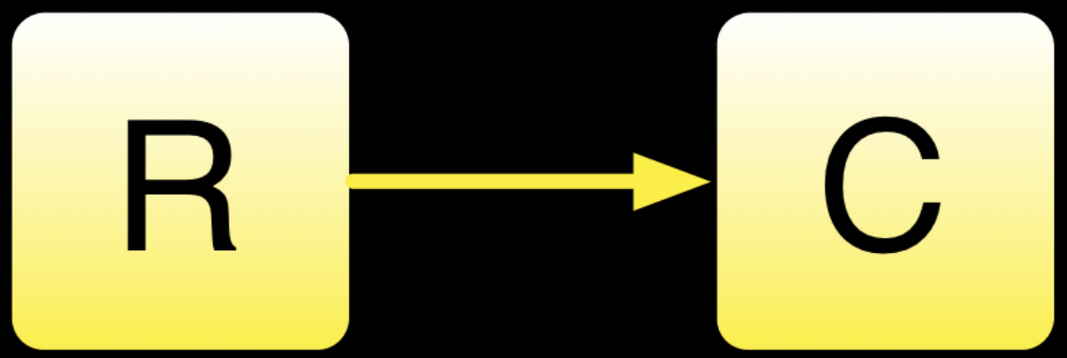

P
R
C

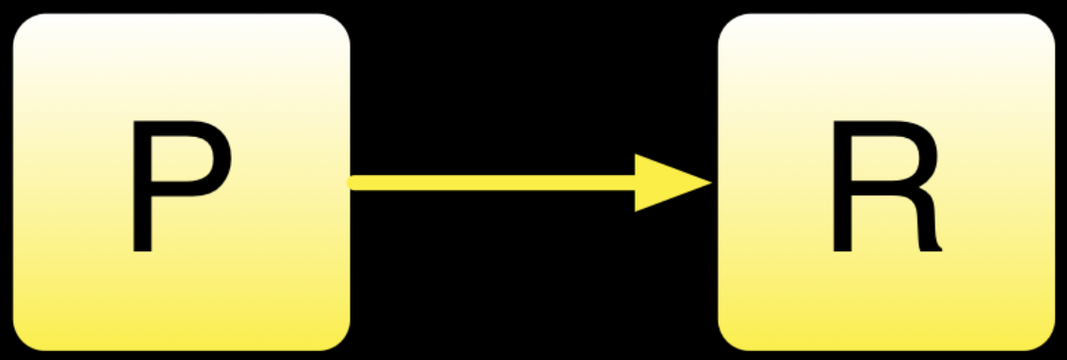

P
R
C


P
R
C

P

C


R
C


P
R


P
R
C

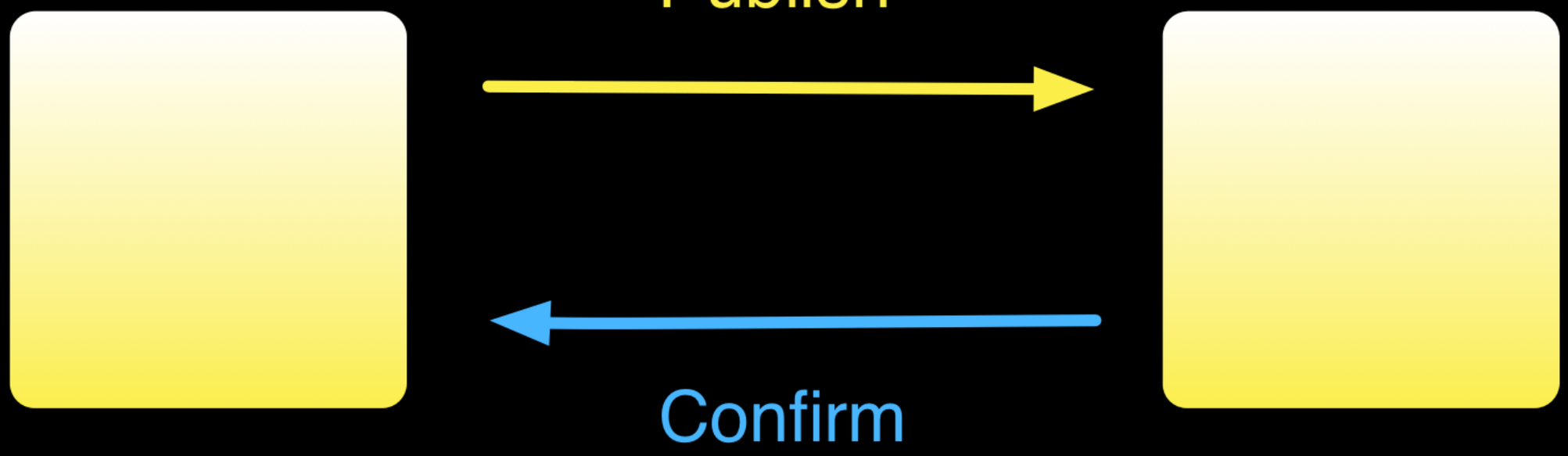
Troubleshooting publishers

Troubleshooting publishers

- I/O exceptions (shutdown handlers)

Troubleshooting publishers

- I/O exceptions (shutdown handlers)
- Publisher confirms


Publish
Confirm

When in doubt, borrow ideas from TCP

Troubleshooting publishers

- I/O exceptions (shutdown handlers)
- Publisher confirms
- Returned message handlers

Troubleshooting publishers

- I/O exceptions (shutdown handlers)
- Publisher confirms
- Returned message handlers
- Invalid payload (e.g. fails to deserialize or decrypt)

Troubleshooting publishers

- I/O exceptions (shutdown handlers)
- Publisher confirms
- Returned message handlers
- Invalid payload (e.g. fails to deserialize or decrypt)
- Identifying publisher instances

Troubleshooting publishers

- identifying blocked (throttled) publishers

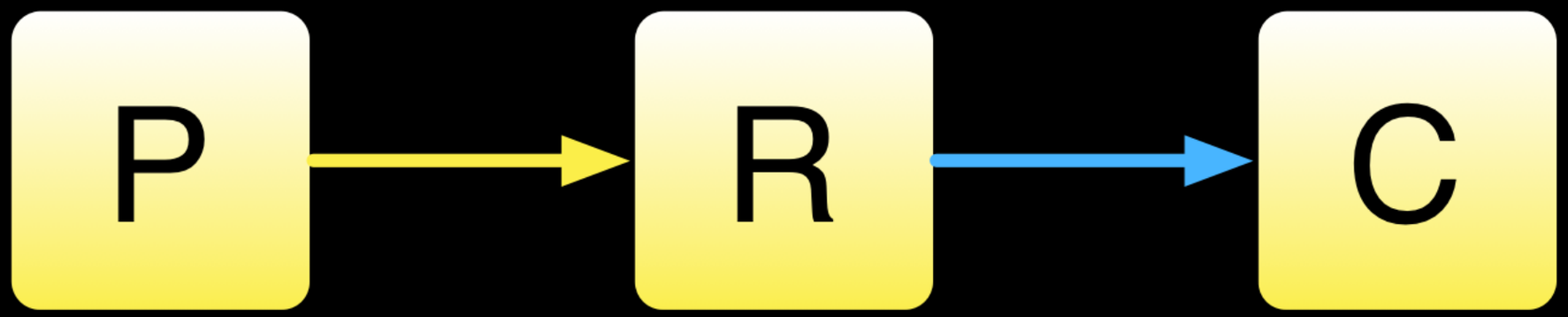
Client-provided connection names in RabbitMQ 3.6.3+

Troubleshooting publishers

- identifying blocked (throttled) publishers
- retries

Troubleshooting publishers

- spring-amqp can cover all of the above


P
R
C

Troubleshooting consumers

Troubleshooting consumers

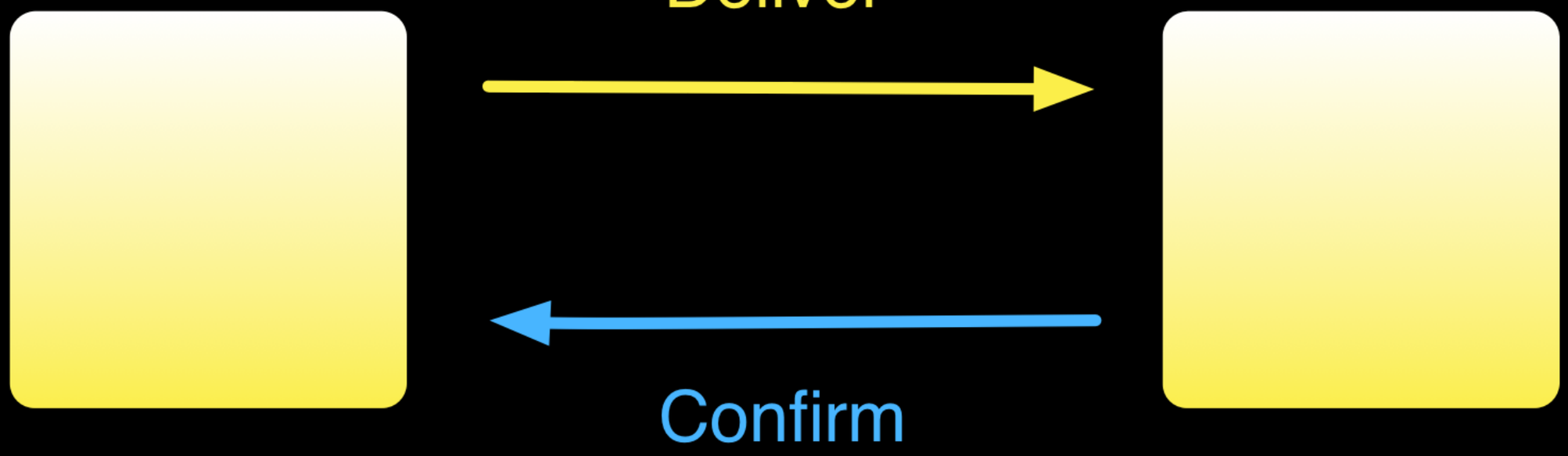
- I/O exceptions

Troubleshooting consumers

- I/O exceptions
- Inadequate delivery QoS

Troubleshooting consumers

- I/O exceptions
- Inadequate delivery QoS
- Lack of confirmations; double-confirming


Deliver
Confirm

Troubleshooting consumers

- I/O exceptions
- Inadequate delivery QoS
- Lack of confirmations; double-confirming

Troubleshooting consumers

- I/O exceptions
- Inadequate delivery QoS
- Lack of confirmations; double-confirming
- Redelivery metrics

Troubleshooting consumers

- I/O exceptions
- Inadequate delivery QoS
- Lack of confirmations; double-confirming
- Redelivery metrics
- Identifying consumer instances

Troubleshooting consumers

- Consumer utilization (reported by HTTP API)

Troubleshooting consumers

- spring-amqp can help with some of the above

“In God we trust, all others must bring data…”

— W. Edwards Deming

“In God we trust, all others must bring data…”

— W. Edwards Deming

— What do you do for a living?

— What do you do for a living?

— Tell people to read the logs.

Sources of data useful for debugging

Sources of data useful for debugging

- Metrics

Sources of data useful for debugging

- Metrics
- Your logs

Sources of data useful for debugging

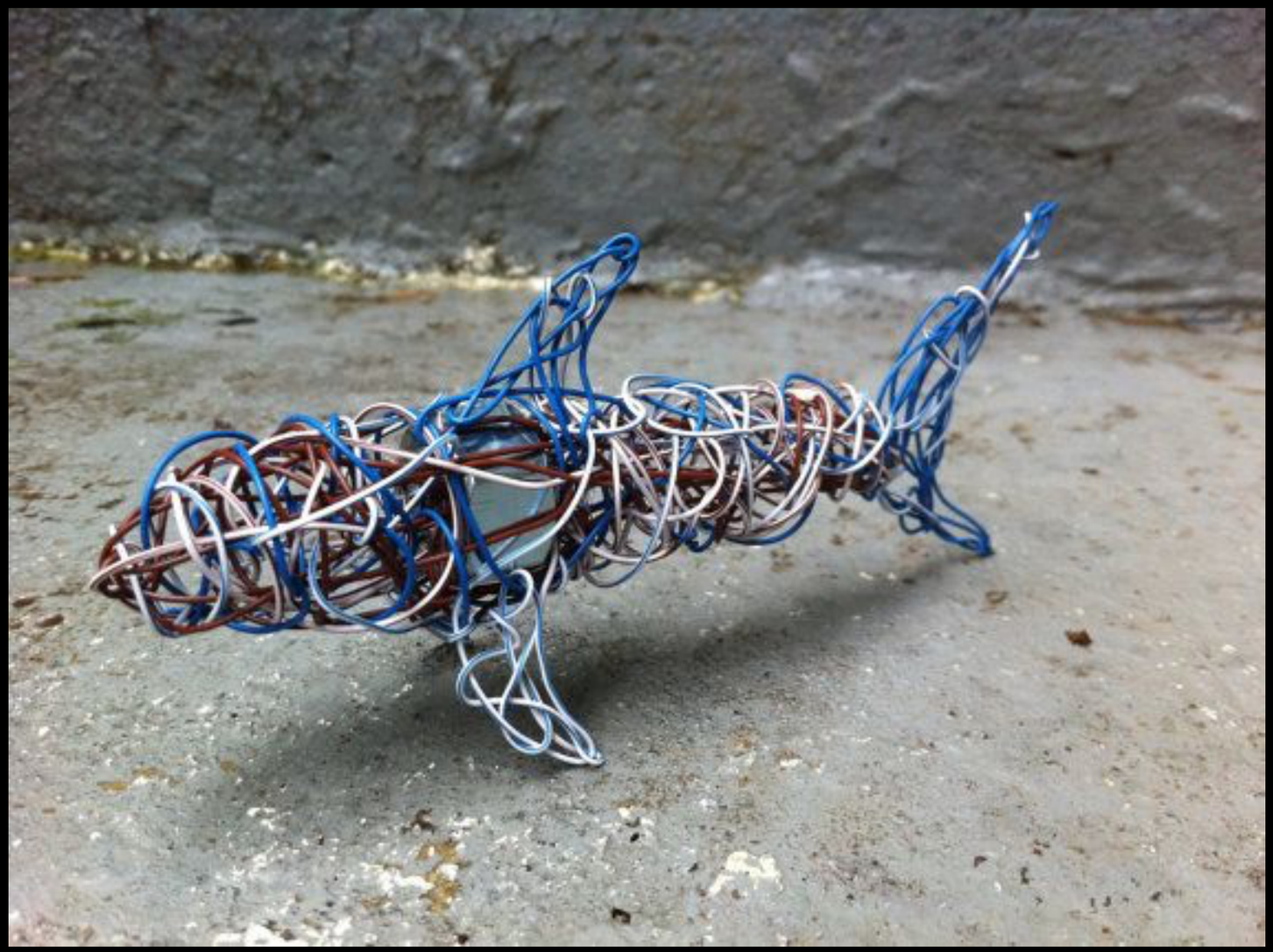
- Metrics
- Your logs
- Someone else's logs

Sources of data useful for debugging

- Metrics
- Your logs
- Someone else's logs
- Tracing data

Sources of data useful for debugging

- Metrics
- Your logs
- Someone else's logs
- Tracing data
- Wireshark (tcpdump, libpcap)



P

C

Collecting data from RabbitMQ

Collecting data from RabbitMQ

- Logs

Collecting data from RabbitMQ

- Logs
- rabbitmqctl status

Collecting data from RabbitMQ

- Logs
- rabbitmqctl status
- rabbitmqctl environment

Collecting data from RabbitMQ

- Logs
- rabbitmqctl status
- rabbitmqctl environment
- rabbitmq-top (ships with RabbitMQ as of 3.6.3)

Collecting data from RabbitMQ

- Logs
- rabbitmqctl status
- rabbitmqctl environment
- rabbitmq-top (ships with RabbitMQ as of 3.6.3)
- HTTP API (lots of metrics)

http://{hostname}:15672/api

```
curl -u guest:guest http://127.0.0.1:15672/api/overview | python -m json.tool
curl -u guest:guest http://127.0.0.1:15672/api/nodes/{node} | python -m json.tool
curl -u guest:guest http://127.0.0.1:15672/api/queues | python -m json.tool
```

Collecting data from RabbitMQ

- Logs
- rabbitmqctl status
- rabbitmqctl environment
- rabbitmq-top (ships with RabbitMQ as of 3.6.3)
- HTTP API (lots of metrics)
- Message tracing ("firehose")

Collecting data from RabbitMQ

- HTTP API (lots of metrics)
- Message tracing ("firehose")
- Infrastructure metrics

Common theme?

Common theme?

- Collect logs system-wide

Common theme?

- Collect logs system-wide
- Collect metrics system-wide

Common theme?

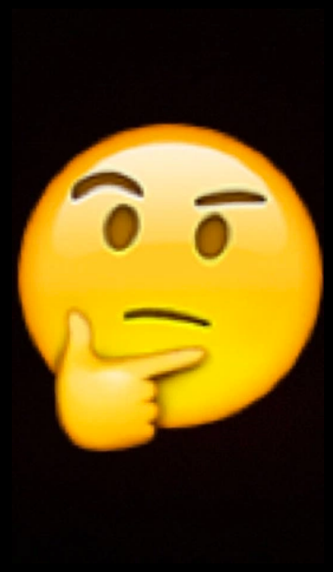
- Collect logs system-wide
- Collect metrics system-wide
- Collect exceptions system-wide

Common theme?

- Collect logs system-wide
- Collect metrics system-wide
- Collect exceptions system-wide
- Trace requests (e.g. with Zipkin)

Common theme?

- Collect logs system-wide
- Collect metrics system-wide
- Collect exceptions system-wide
- Trace requests (e.g. with Zipkin)
- Analyze



Common theme?

- Collect logs system-wide
- Collect metrics system-wide
- Collect exceptions system-wide
- Trace requests (e.g. with Zipkin)
- Analyze
- Sounds like something a structured platform can help with!

Distributed system
debugging is a problem
far from being solved.

Thank you

Thank you

- @michaelklishin

Thank you

- @michaelklishin
- github.com/michaelklishin

Thank you

- @michaelklishin
- github.com/michaelklishin
- mklishin@pivotal.io