Being Ready for Apache Kafka: Today's Ecosystem and Future Roadmap

Michael G. Noll

@miguno

Developer Evangelist, Confluent Inc.

confluent

Apache: Big Data Conference, Budapest, Hungary, September 29, 2015

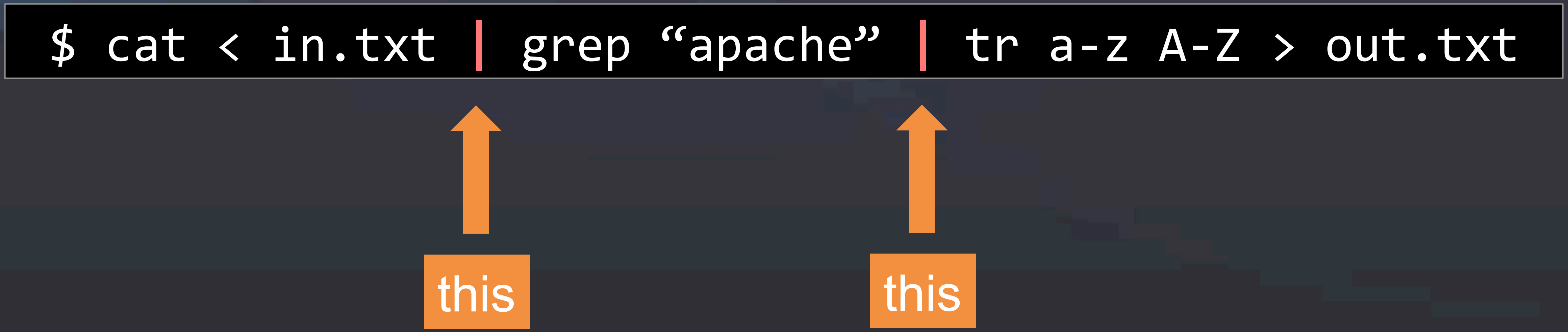
- Developer Evangelist at Confluent since August ‘15
- Previously Big Data lead at .COM/.NET DNS operator Verisign
- Blogging at http://www.michael-noll.com/ (too little time!)
- PMC member of Apache Storm (too little time!)
- michael@confluent.io

- Founded in Fall 2014 by the creators of Apache Kafka
- Headquartered in San Francisco bay area
- We provide a stream data platform based on Kafka
- We contribute a lot to Kafka, obviously ☺

kafka

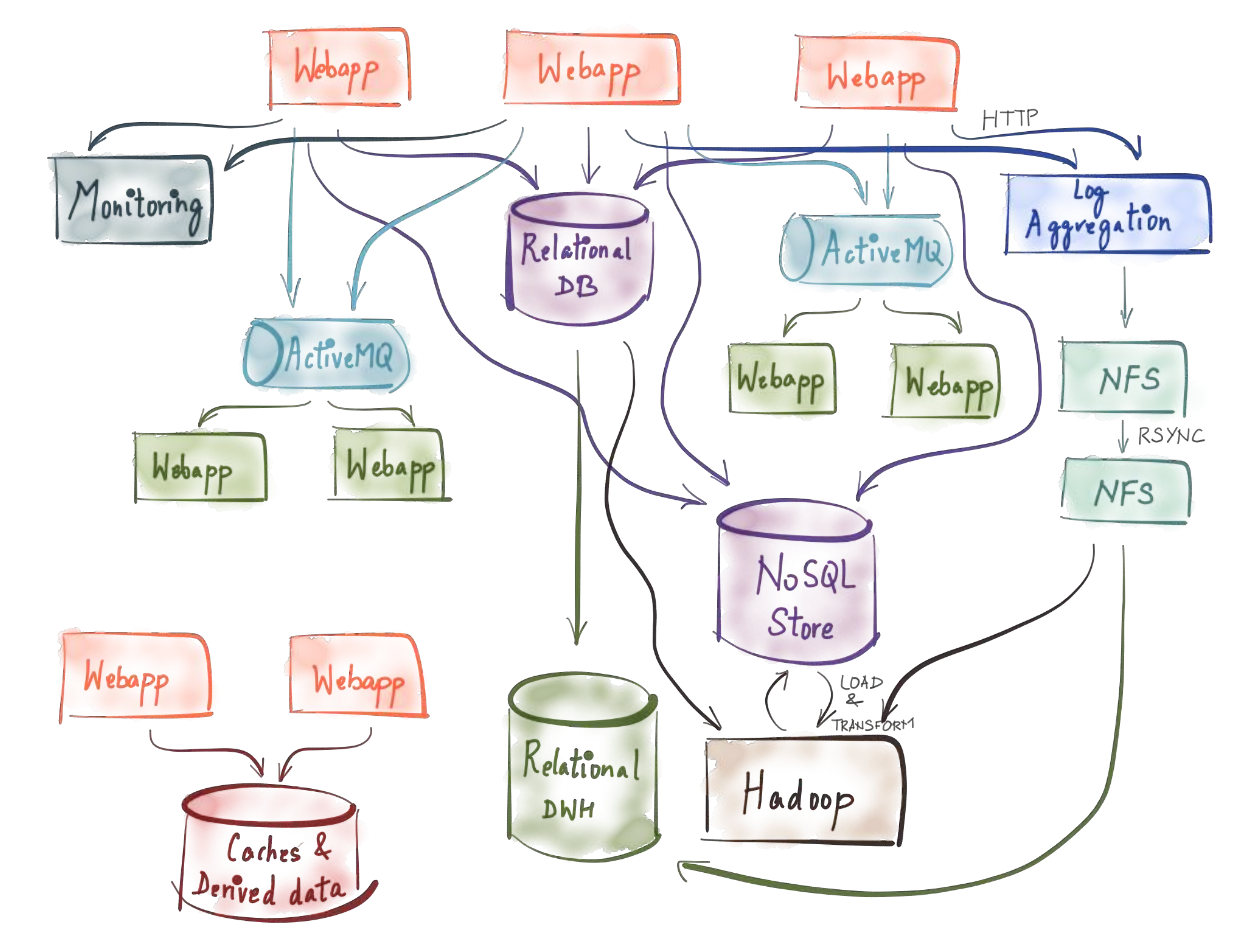
?

Apache Kafka is the distributed, durable equivalent of Unix pipes.
Use it to connect and compose your large-scale data apps.

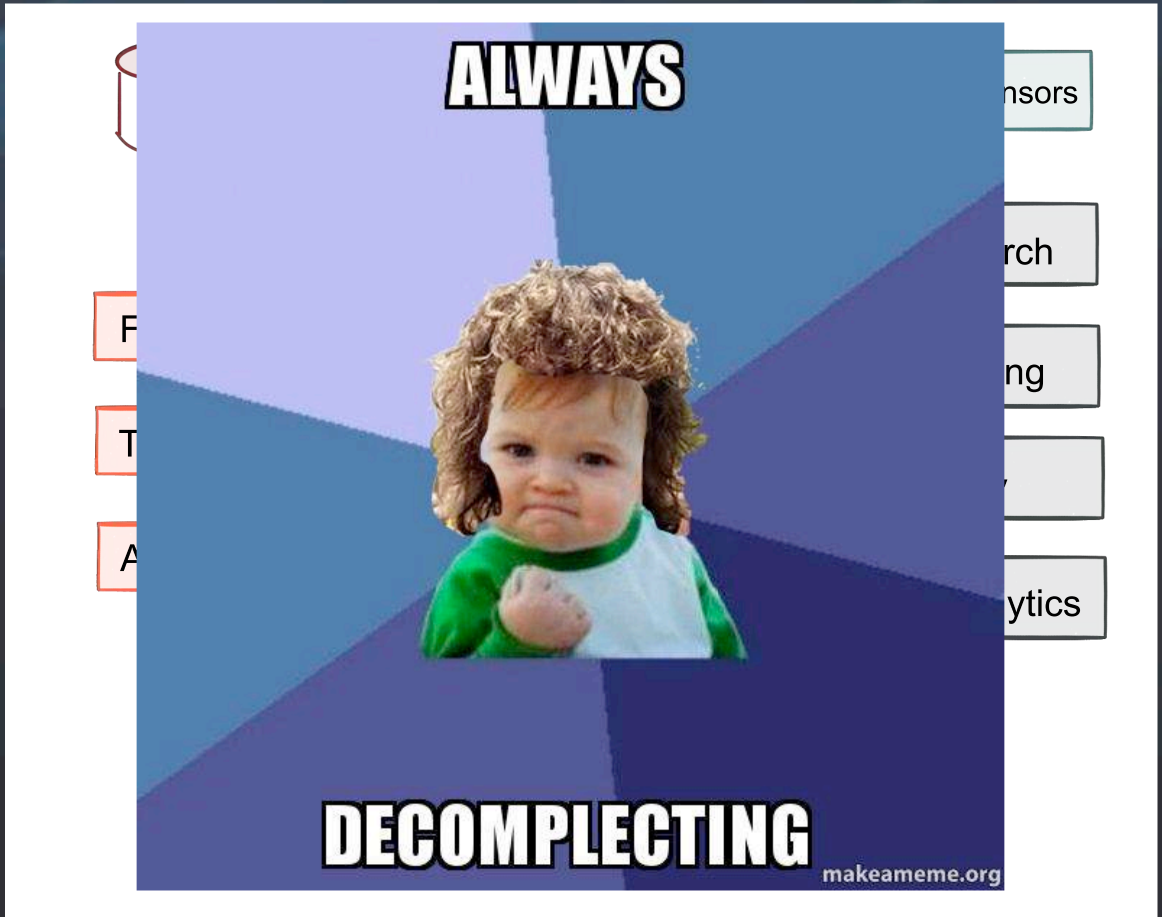


```
$ cat < in.txt | grep “apache” | tr a-z A-Z > out.txt
```

Example: LinkedIn before Kafka



Example: LinkedIn after Kafka

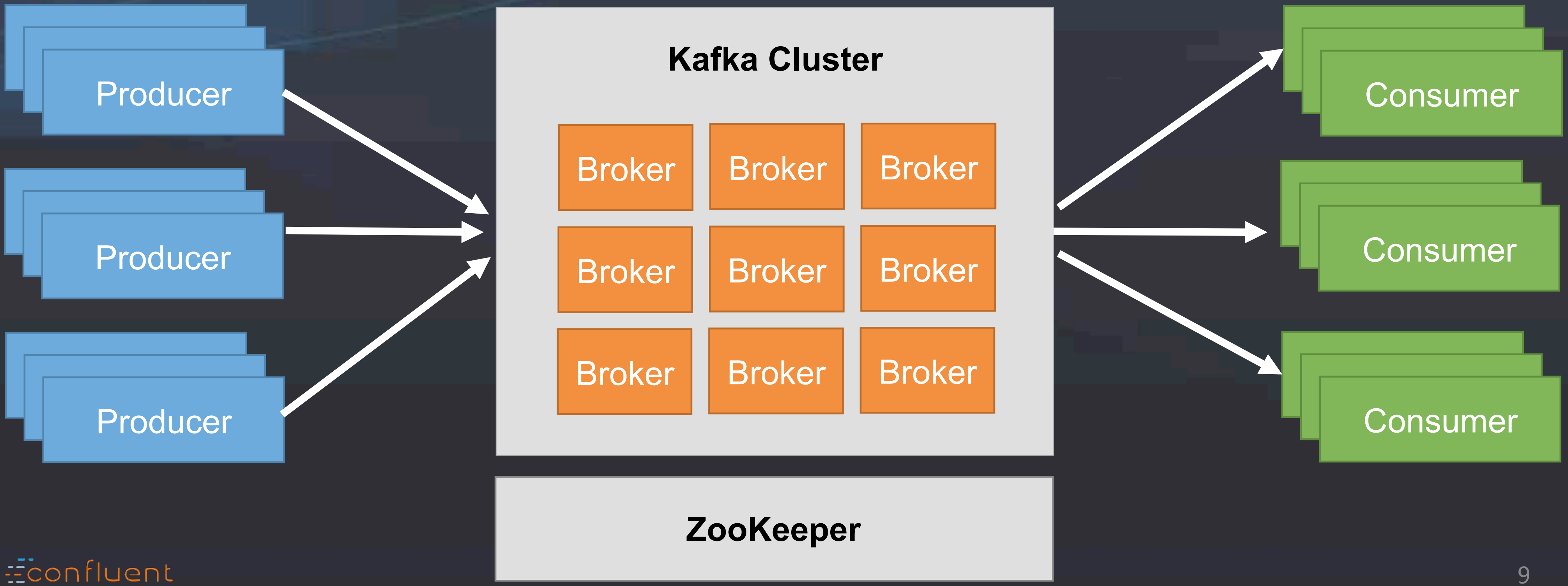


Apache Kafka is a high-throughput distributed messaging system.

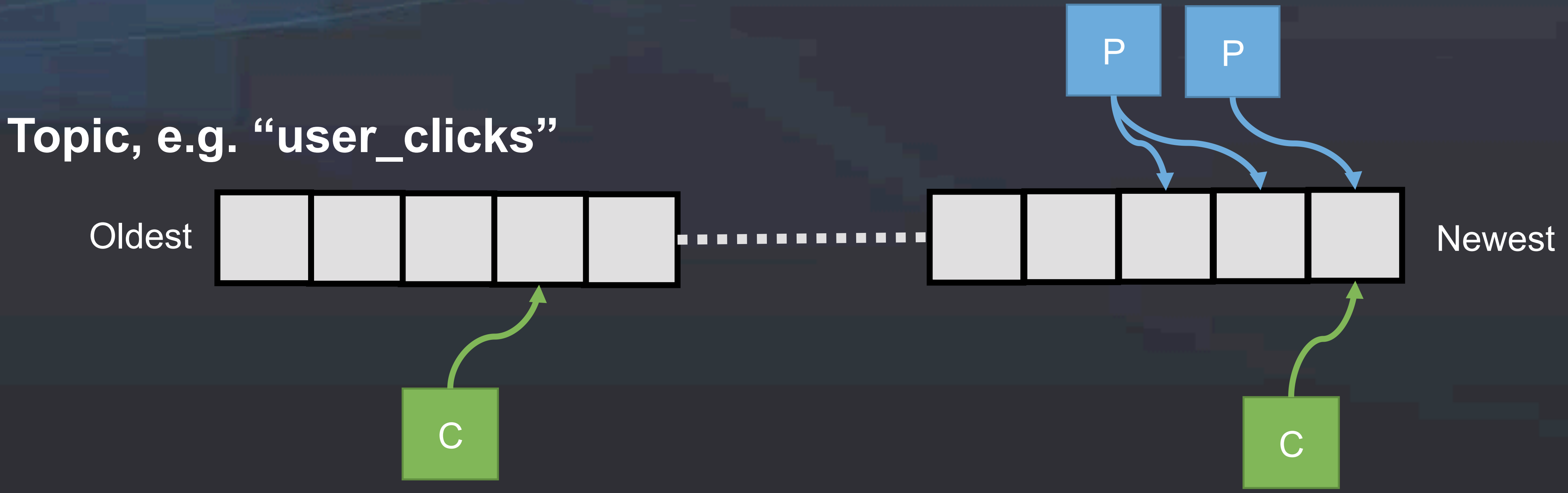
“**1,100,000,000,000 msg/day**, totaling 175+ TB/day” (LinkedIn)

= **3 billion** messages since the beginning of this talk

Apache Kafka is a publish-subscribe messaging rethought as a distributed commit log.

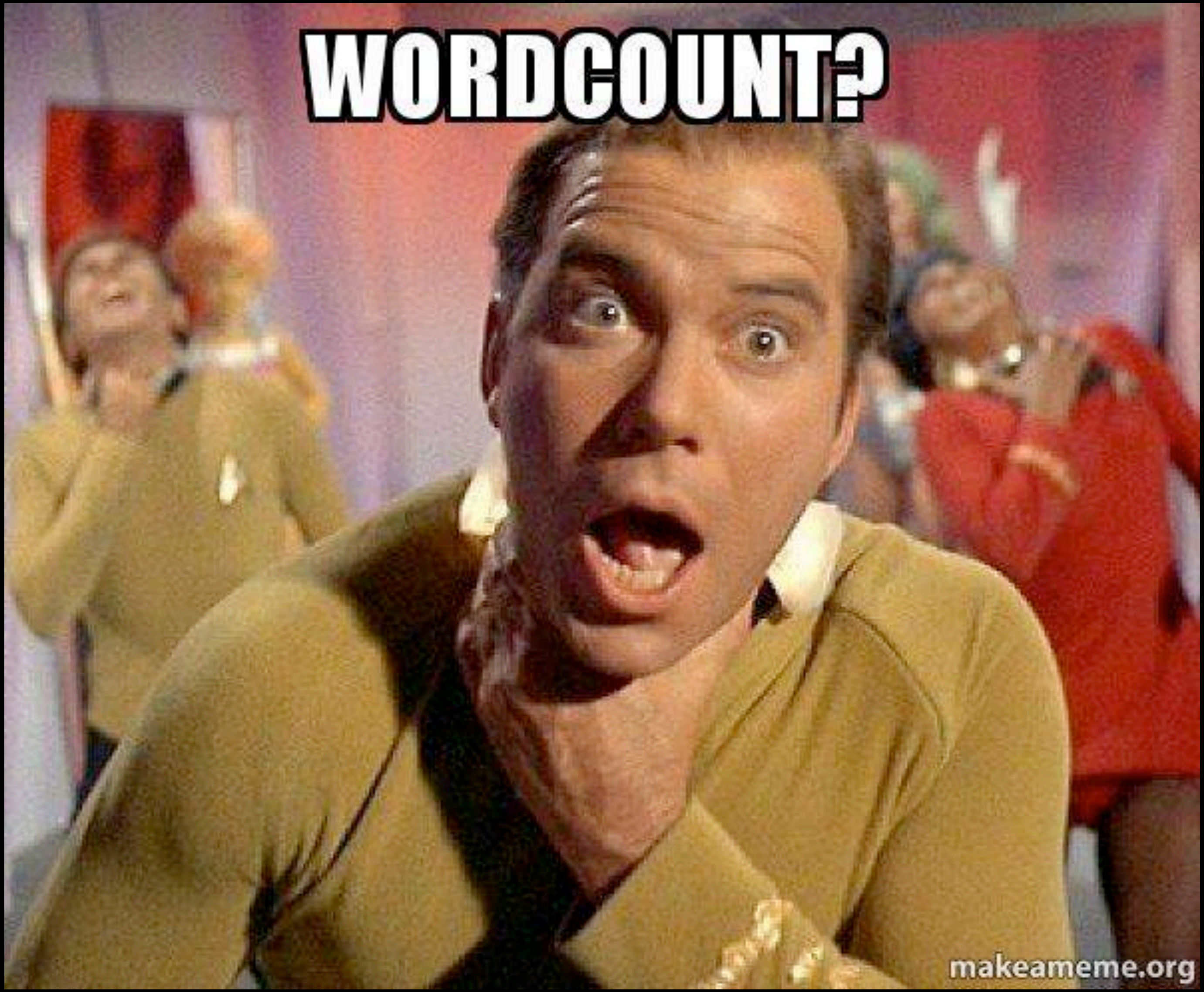


Apache Kafka is a publish-subscribe messaging rethought as a distributed commit log.

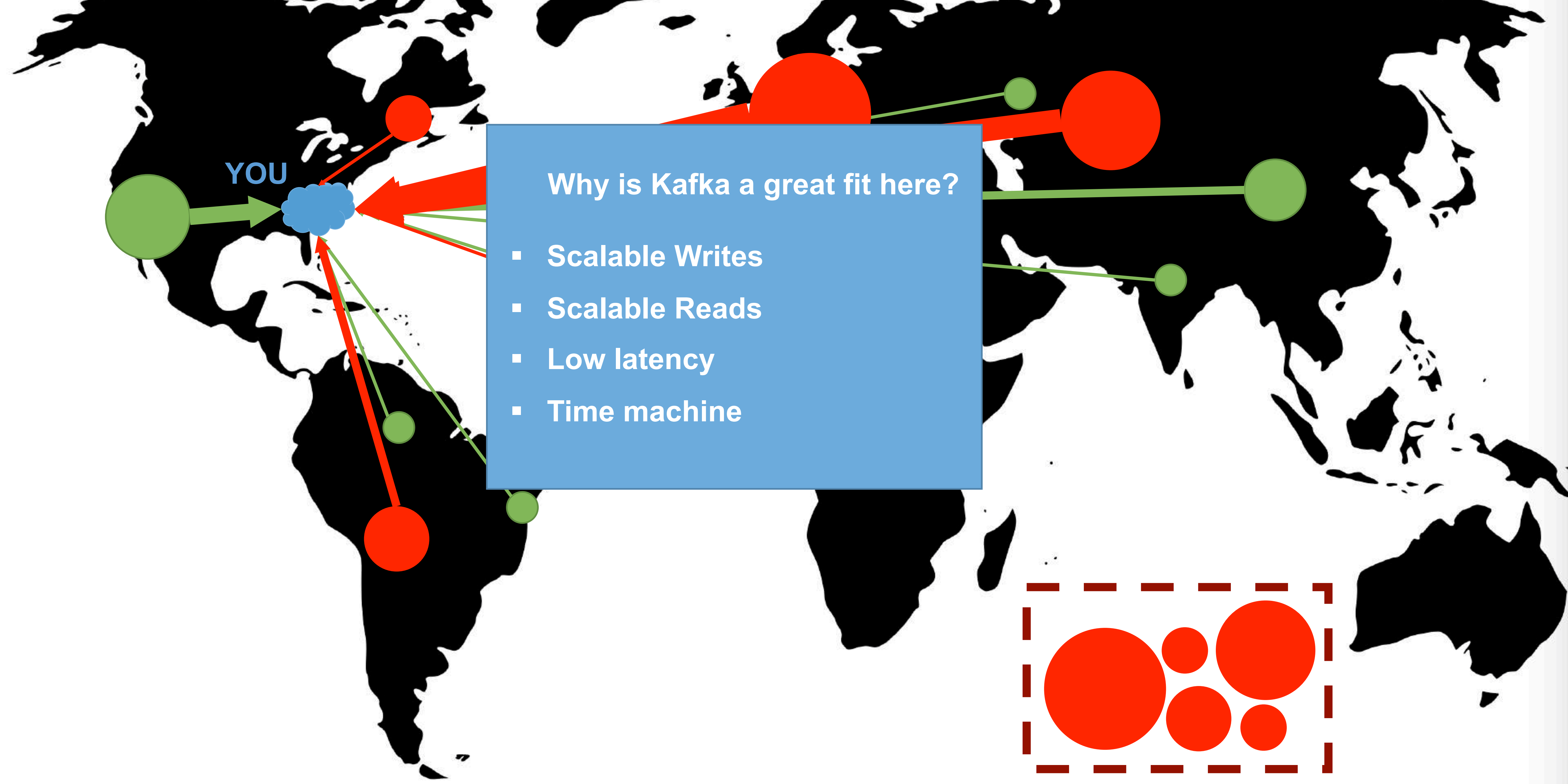


So where can Kafka help me?

Example, anyone?


WORDCOUNT?
makeameme.org

Example: Protecting your infrastructure against DDoS attacks



Why is Kafka a great fit here?

- Scalable Writes
- Scalable Reads
- Low latency
- Time machine

Kafka powers many use cases

- User behavior, click stream analysis
- Infrastructure monitoring and security, e.g. DDoS detection
- Fraud detection
- Operational telemetry data from mobile devices and sensors
- Analyzing system and app logging data
- Internet of Things (IoT) applications
- …and many more
  - Yours is missing? Let me know via michael@confluent.io !

Wikipedia
The Free Encyclopedia
PayPal
NETFLIX
Goldman Sachs
STRAVA
Wooga
YAHOO!
IFTTT
if this then that
ORACLE
Etsy
Pinterest
loggly

Diverse and rapidly growing user base across many industries and verticals.

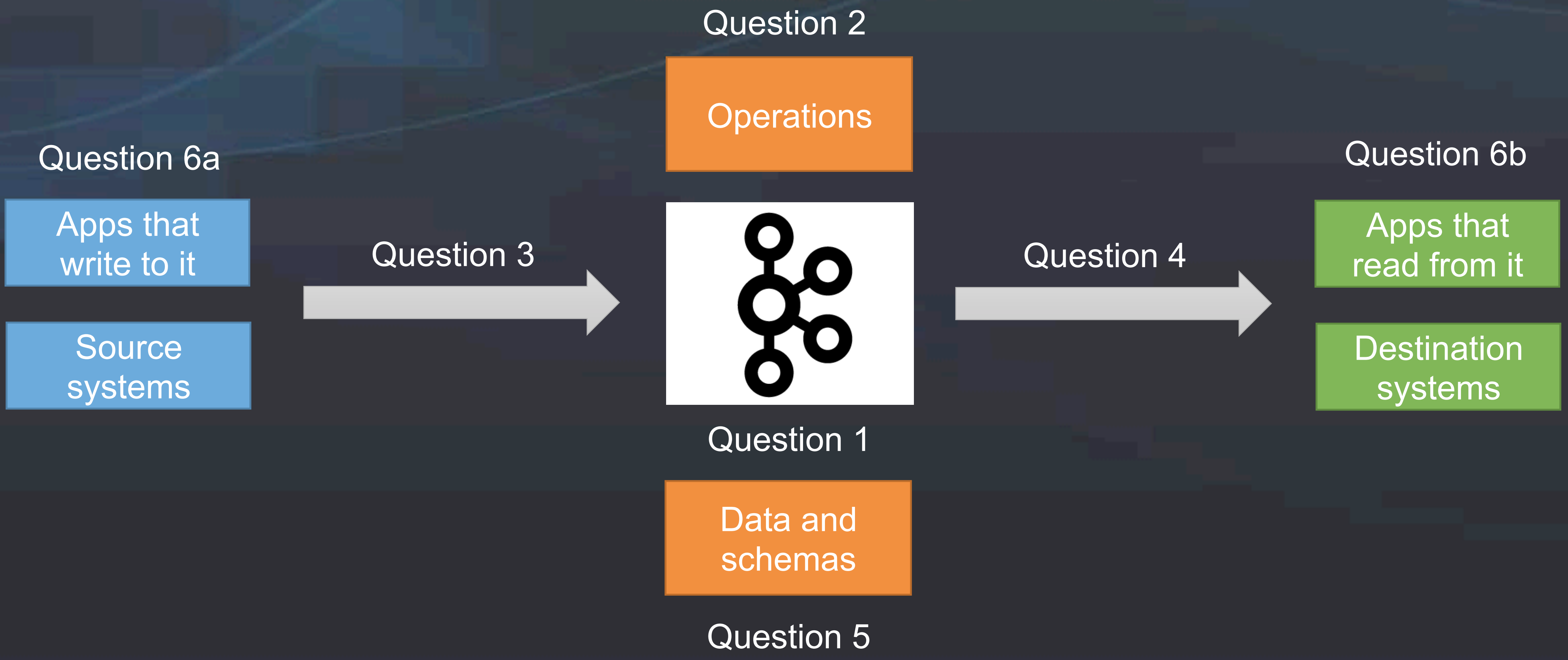
confluent

https://cwiki.apache.org/confluence/display/KAFKA/Powered+By

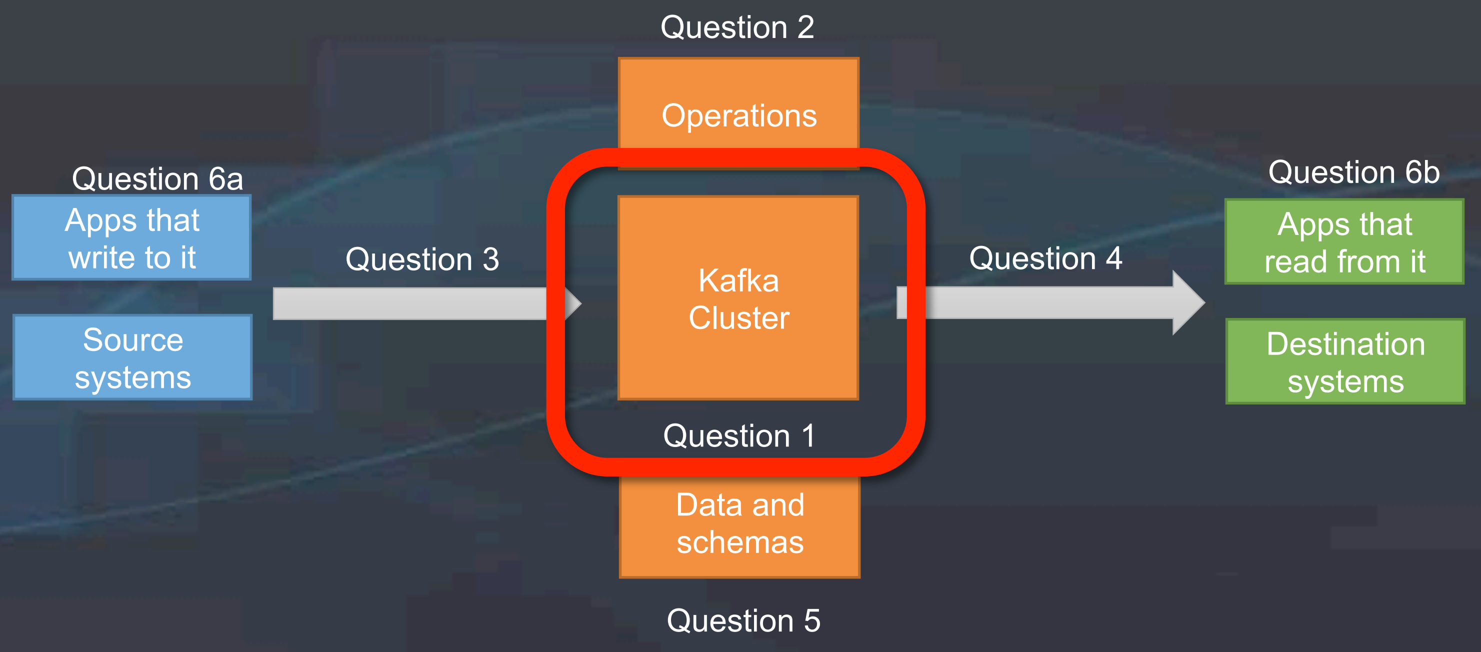
A typical Kafka architecture

Yes, we now begin to approach “production”

Typical architecture => typical questions



Wait a minute!



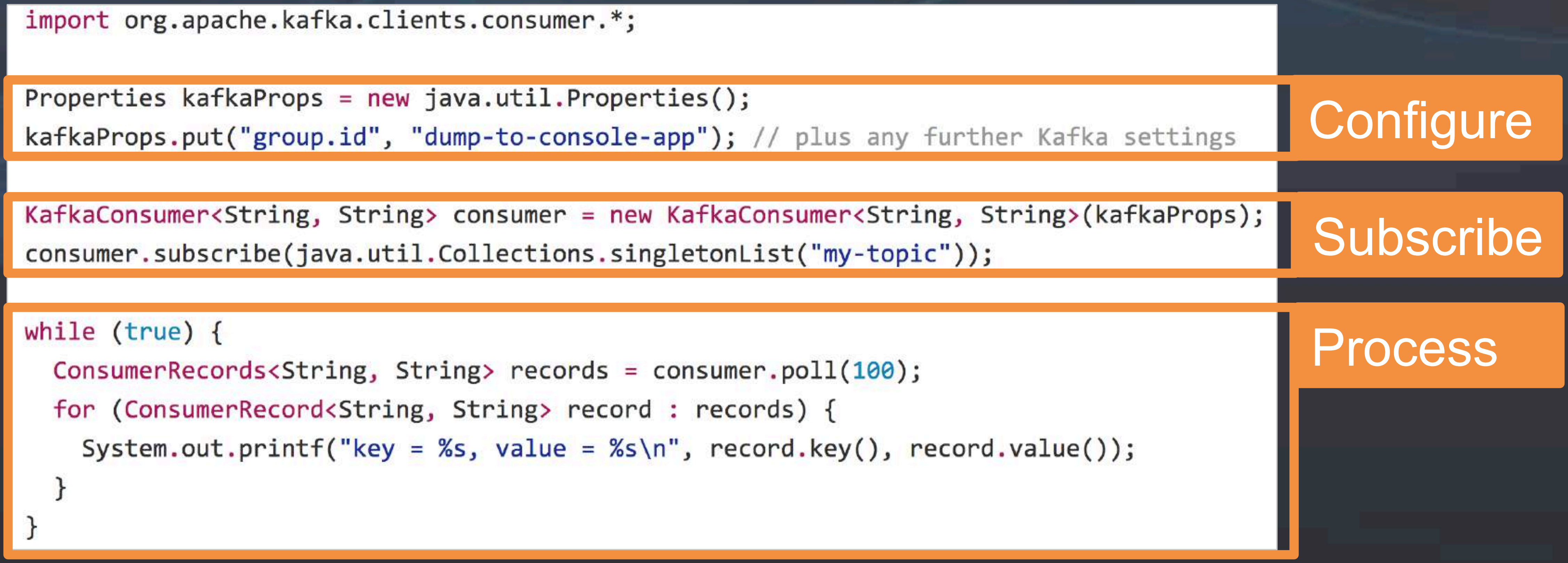
Kafka core

Question 1 or “What are the upcoming improvements to core Kafka?”

Kafka core: upcoming changes in 0.9.0

- Kafka 0.9.0 (formerly 0.8.3) expected in November 2015
- ZooKeeper now only required for Kafka **brokers**
  - ZK dependency removed from clients = producers and consumers
  - Benefits include less load on ZK, lower operational complexity, user apps don’t require interacting with ZK anymore
- New, unified consumer Java API
  - We consolidated the previous “high-level” and “simple” consumer APIs
  - Old APIs still available and not deprecated (yet)

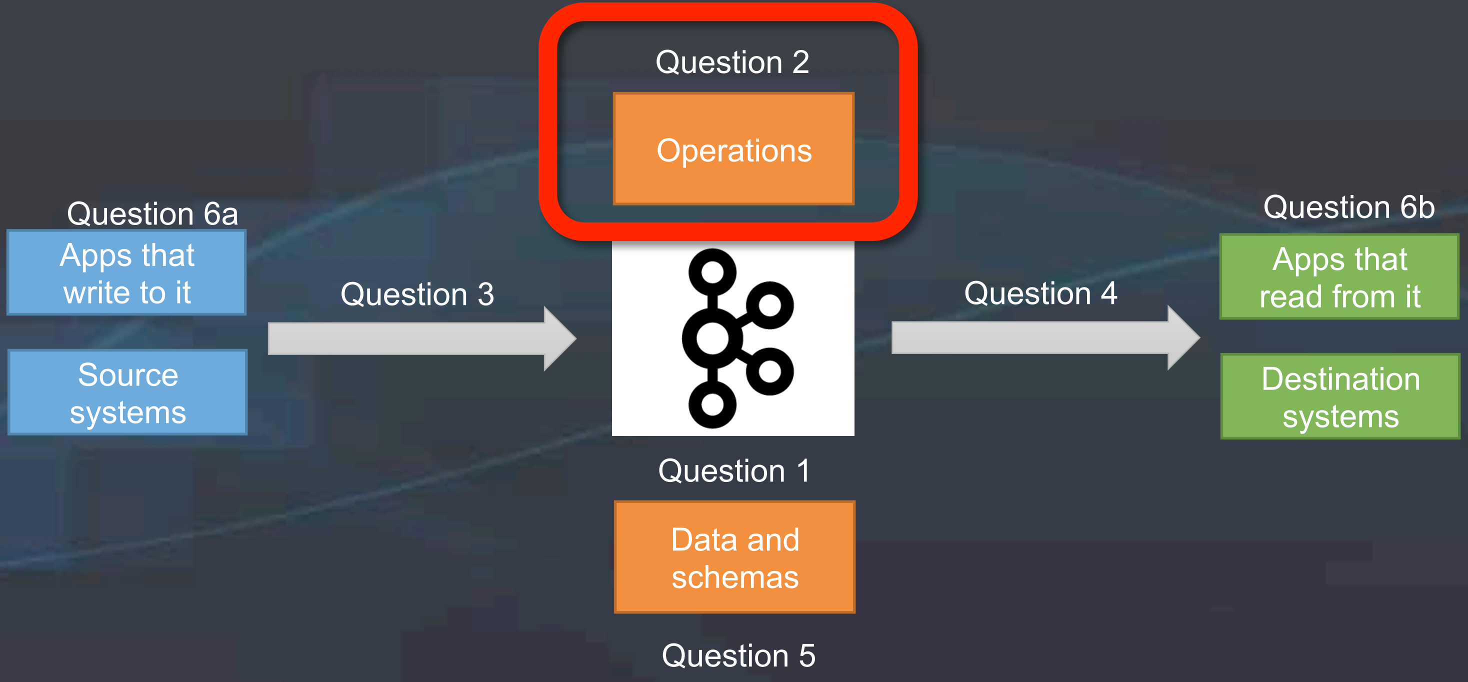
New consumer Java API in 0.9.0



```java
import org.apache.kafka.clients.consumer.*;

Properties kafkaProps = new java.util.Properties();
kafkaProps.put("group.id", "dump-to-console-app"); // plus any further Kafka settings

KafkaConsumer<String, String> consumer = new KafkaConsumer<String, String>(kafkaProps);
consumer.subscribe(java.util.Collections.singletonList("my-topic"));

while (true) {
  ConsumerRecords<String, String> records = consumer.poll(100);
  for (ConsumerRecord<String, String> record : records) {
    System.out.printf("key = %s, value = %s\n", record.key(), record.value());
  }
}
```

Kafka core: upcoming changes in 0.9.0

- Improved **data import/export** via Copycat
    - KIP-26: https://cwiki.apache.org/confluence/pages/viewpage.action?pageId=58851767
    - Will talk about this later
- Improved **security**: SSL support for encrypted data transfer
    - Yay, finally make your InfoSec team (a bit) happier!
    - https://cwiki.apache.org/confluence/display/KAFKA/Deploying+SSL+for+Kafka
- Improved **multi-tenancy**: quotas aka throttling for Ps and Cs
    - KIP-13: https://cwiki.apache.org/confluence/display/KAFKA/KIP-13+-+Quotas
    - Quotas are defined per broker, will slow down clients if needed
    - Reduces collateral damage caused by misbehaving apps/teams



Kafka operations

Question 2 or “How do I deploy, manage, monitor, etc. my Kafka clusters?”

Deploying Kafka

- Hardware recommendations, configuration settings, etc.
  - http://docs.confluent.io/current/kafka/deployment.html
  - http://kafka.apache.org/documentation.html#hwandos
- Deploying Kafka itself = DIY at the moment
- Packages for Debian and RHEL OS families available via Confluent Platform
  - http://www.confluent.io/developer
- Straight-forward to use orchestration tools like Puppet, Ansible
- Also: options for Docker, Mesos, YARN, …

Managing Kafka: CLI tools

- Kafka includes a plethora of CLI tools
  - Managing topics, controlling replication, status of clients, …
- Can be tricky to understand which tool to use, when, and how
- Helpful pointers:
  - https://cwiki.apache.org/confluence/display/KAFKA/System+Tools
  - https://cwiki.apache.org/confluence/display/KAFKA/Replication+tools
- KIP-4 will eventually add better management APIs

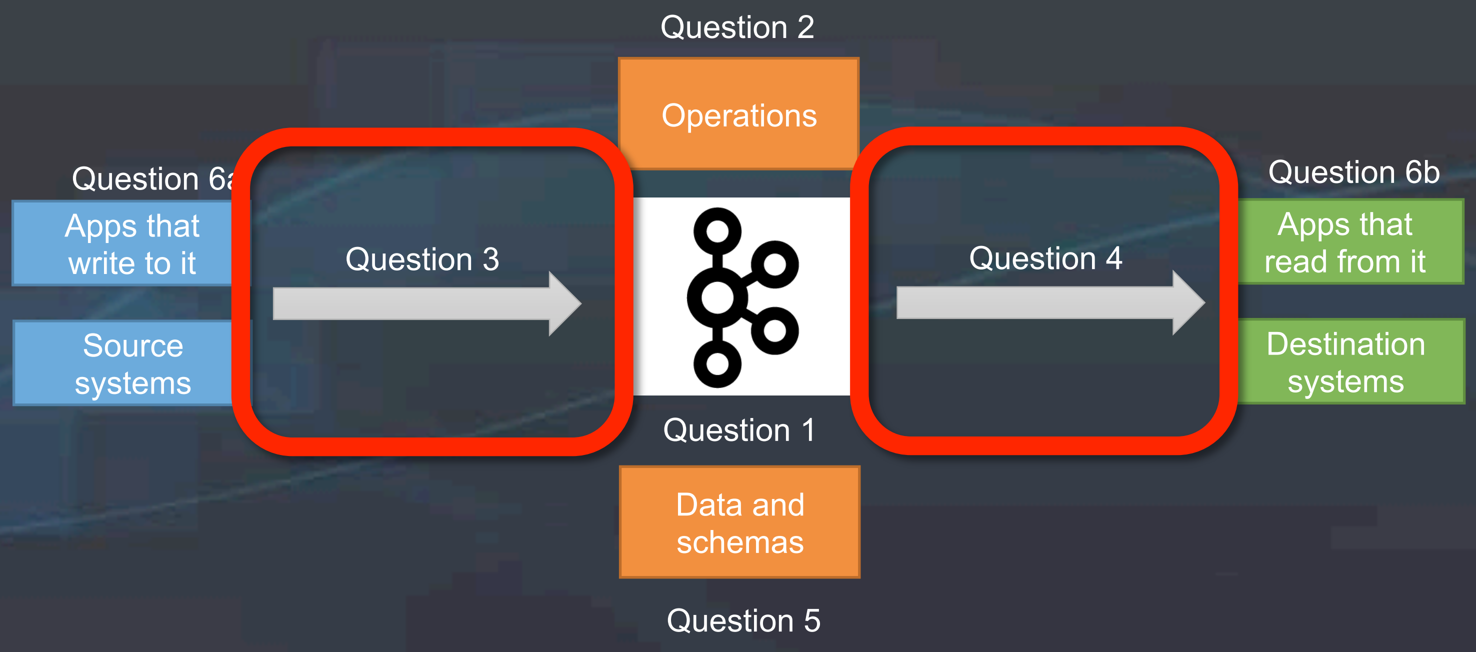
Monitoring Kafka: metrics

- How to monitor
  - Usual tools like Graphite, InfluxDB, statsd, Grafana, collectd, diamond
- What to monitor – some key metrics
  - Host metrics: CPU, memory, disk I/O and usage, network I/O
  - Kafka metrics: consumer lag, replication stats, message latency, Java GC
  - ZooKeeper metrics: latency of requests, #outstanding requests
- Kafka exposes many built-in metrics via JMX
  - Use e.g. jmxtrans to feed these metrics into Graphite, statsd, etc.

Monitoring Kafka: logging

- You can expect lots of logging data for larger Kafka clusters
- Centralized logging services help significantly
  - You have one already, right?
  - Elasticsearch/Kibana, Splunk, Loggly, …

- Further information about operations and monitoring at:
  - http://docs.confluent.io/current/kafka/monitoring.html
  - https://www.slideshare.net/miguno/apache-kafka-08-basic-training-verisign



Kafka clients #1

Questions 3+4 or “How can my apps talk to Kafka?”

Recommended* Kafka clients as of today

Polyglot Ready (tm)

Language	Name	Link
Java	**<built-in>**	**http://kafka.apache.org/**
C/C++	librdkafka	http://github.com/edenhill/librdkafka
Python	kafka-python	https://github.com/mumrah/kafka-python
Go	sarama	https://github.com/Shopify/sarama
Node	kafka-node	https://github.com/SOHU-Co/kafka-node/
Scala	reactive kafka	https://github.com/softwaremill/reactive-kafka
…	…	…

confluent

*Opinionated! Full list at https://cwiki.apache.org/confluence/display/KAFKA/Clients

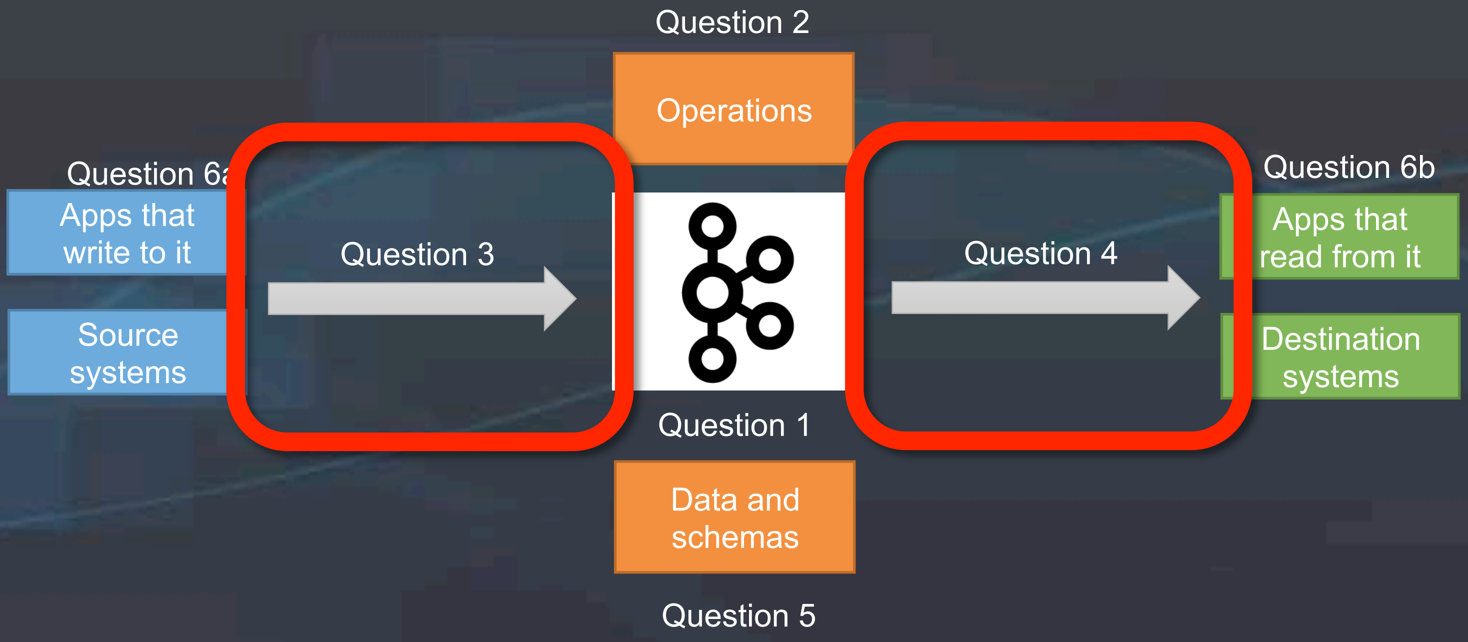
Kafka clients: upcoming improvements

- Current problem: only **Java client** is officially supported
  - A lot of effort and duplication for client maintainers to be compatible with Kafka changes over time (e.g. protocol, ZK for offset management)
  - Wait time for users until “their” client library is ready for latest Kafka
- Idea: use **librdkafka** (C) as the basis for Kafka clients and provide bindings + idiomatic APIs per target language
- Benefits include:
  - Full protocol support, SSL, etc. needs to be implemented only once
  - All languages will benefit from the speed of the C implementation
  - Of course you are always free to pick your favorite client!

Confluent Kafka-REST

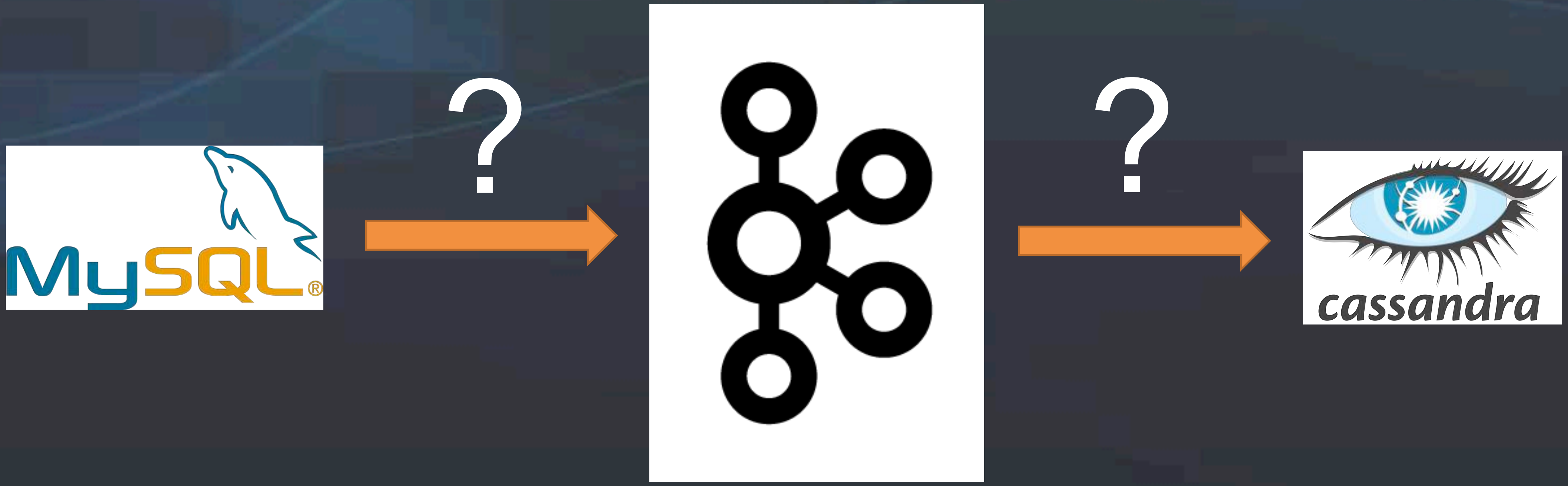
- Open source, included in Confluent Platform
  https://github.com/confluentinc/kafka-rest/
- Alternative to native clients
- Supports reading and writing data, status info, Avro, etc.

```
# Get a list of topics
$ curl "http://rest-proxy:8082/topics"

[ { "name":"userProfiles",    "num_partitions": 3 },
  { "name":"locationUpdates", "num_partitions": 1 } ]
```



Kafka clients #2

Questions 3+4 or “How can my systems talk to Kafka?”

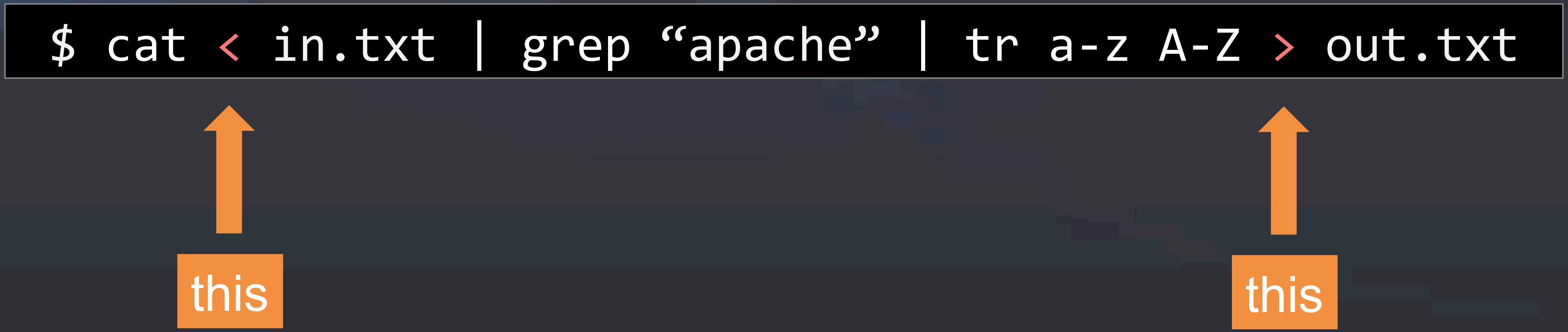

MySQL
?
?
cassandra

Data import/export: status quo

- Until now this has been **your** problem to solve
  - Only few tools available, e.g. LinkedIn Camus for Kafka → HDFS export
  - Typically a DIY solution using the aforementioned client libs

- Kafka 0.9.0 will introduce **Copycat**

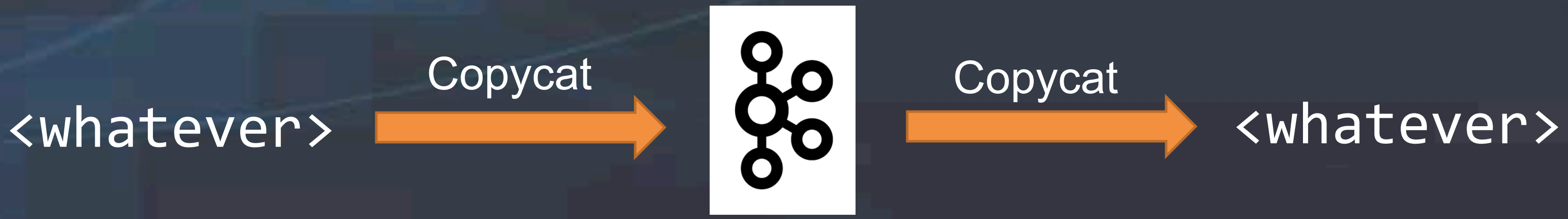
Copycat is the I/O redirection in your Unix pipelines.
Use it to get your data into and out of Kafka.



```
$ cat < in.txt | grep “apache” | tr a-z A-Z > out.txt
```

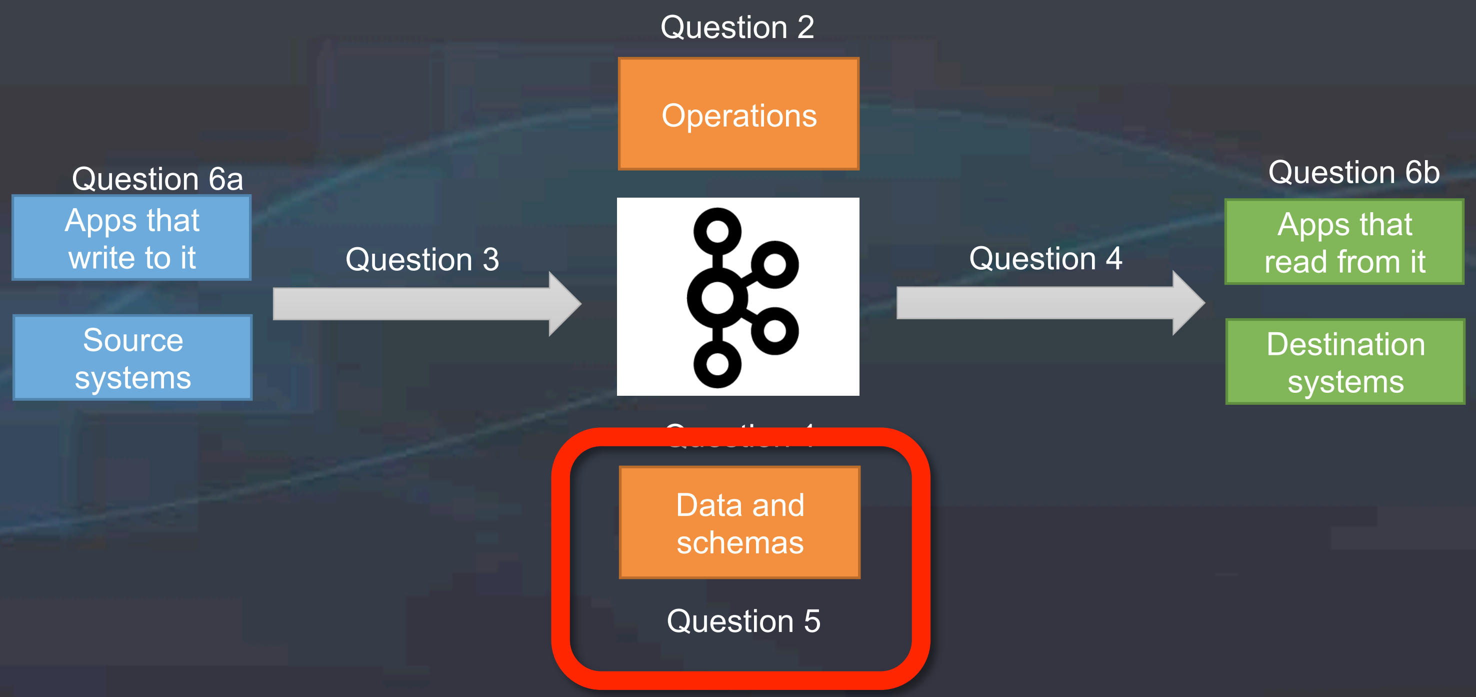
Data import/export via Copycat

- **Copycat** is included in upcoming Kafka 0.9.0



```
<whatever>
<whatever>
```

- Federated Copycat “connector” development for e.g. HDFS, JDBC
- Light-weight, scales from simple testing and one-off jobs to large-scale production scenarios serving an entire organization
- Process management agnostic, hence flexible deployment options
  - Examples: Standalone, YARN, Mesos, or your own (e.g. Puppet w/ supervisord)



Question 2

Operations

Question 6a

Apps that write to it

Source systems

Question 3

Question 4

Question 6b

Apps that read from it

Destination systems

Question 1

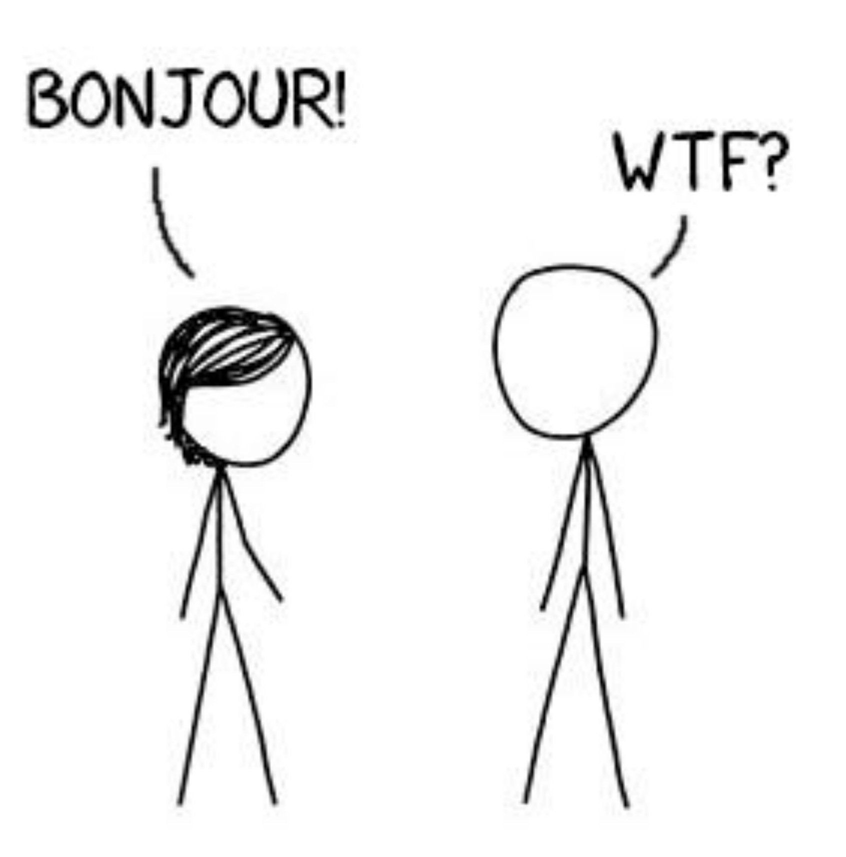
Data and schemas

Question 5

Data and schemas

Question 5 or “Je te comprends pas”

ALL I SAID WAS
LET'S AGREE ON A DATA FORMAT
memegenerator.net

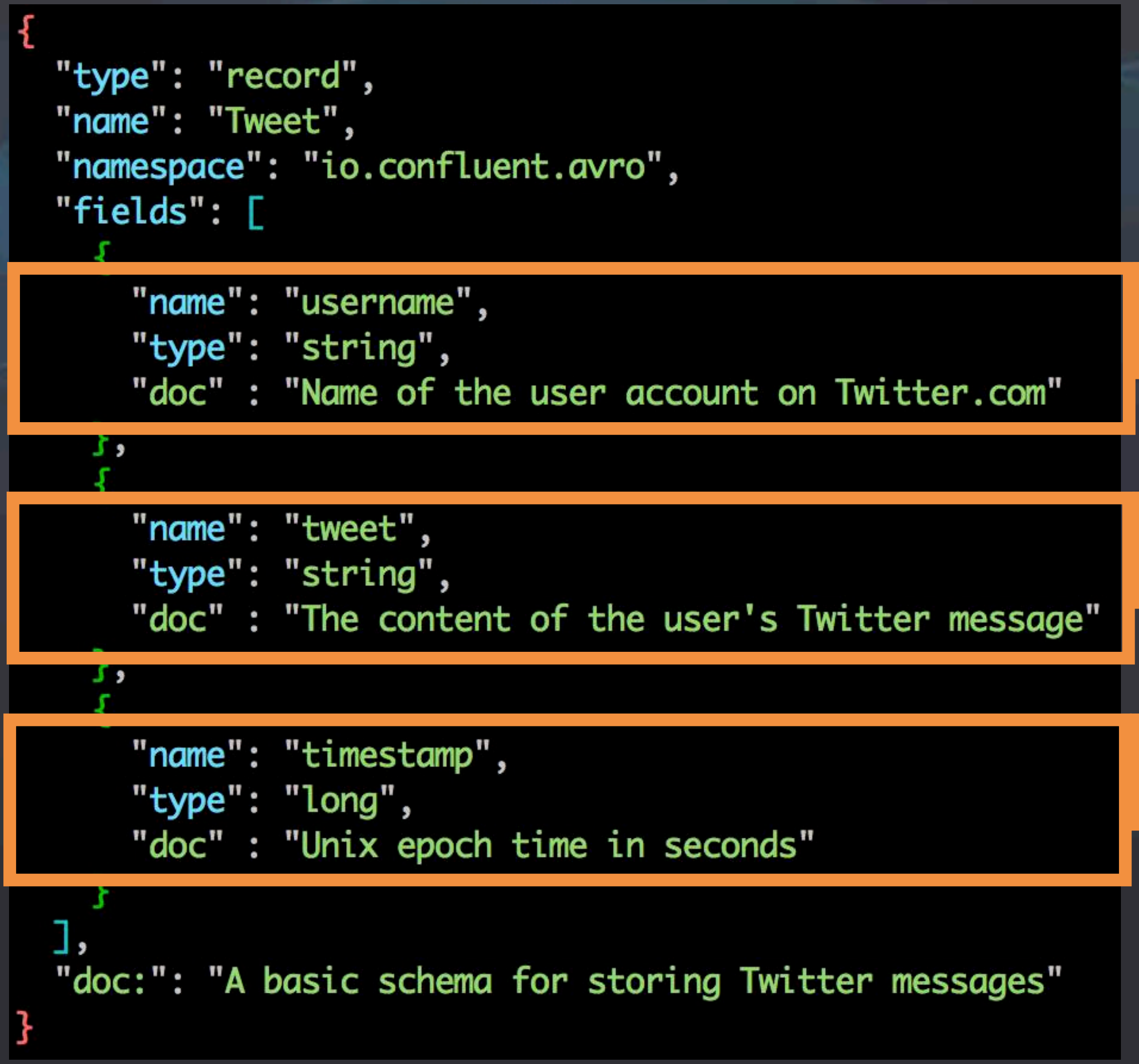

BONJOUR!
WTF?

Data and schemas

- Agree on **contracts for data** just like you do for, say, APIs
  - Producers and consumers of data must understand each other
  - Free-for-alls typically degenerate quickly into team deathmatches
  - Benefit from clear contract, schema evolution, type safety, etc.
- Organizational problem rather than technical
  - Hilarious /facepalm moments
    - Been there, done that ☺
- Take a look at **Apache Avro**, Thrift, Protocol Buffers
  - Cf. Avro homepage, https://github.com/miguno/avro-hadoop-starter

“Alternative” to schemas

Example: Avro schema for tweets



```
{
  "type": "record",
  "name": "Tweet",
  "namespace": "io.confluent.avro",
  "fields": [
    {
      "name": "username",
      "type": "string",
      "doc" : "Name of the user account on Twitter.com"
    },
    {
      "name": "tweet",
      "type": "string",
      "doc" : "The content of the user's Twitter message"
    },
    {
      "name": "timestamp",
      "type": "long",
      "doc" : "Unix epoch time in seconds"
    }
  ],
  "doc:": "A basic schema for storing Twitter messages"
}
```

username

text

timestamp

confluent

<data> = <definition>

<data> = <definition>

<data> = <definition>

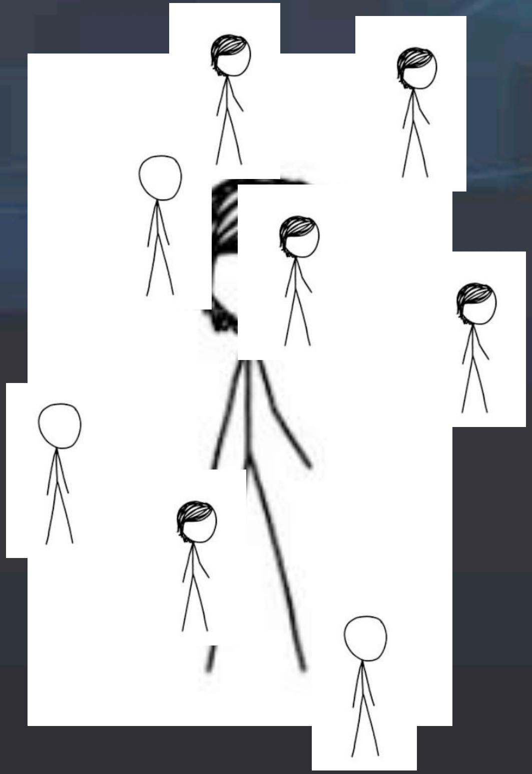
<data> = <definition>

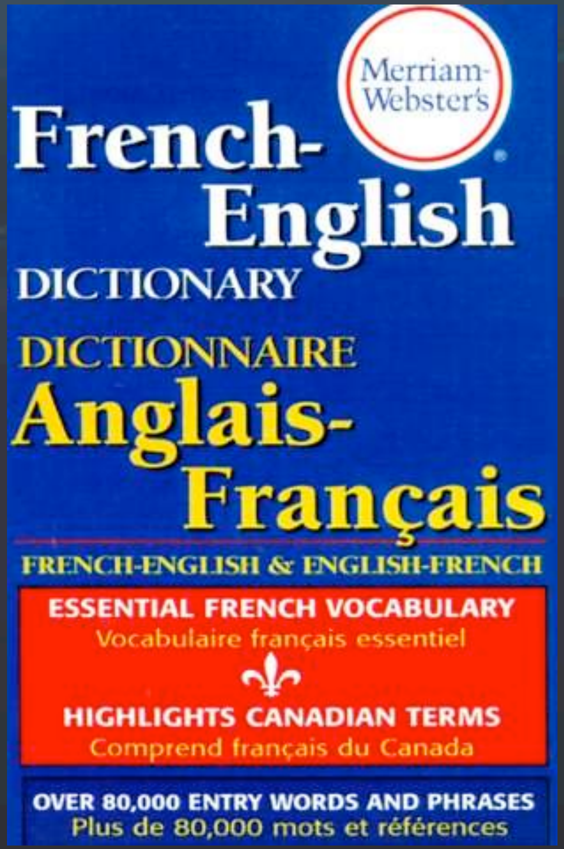
<data> = <definition>

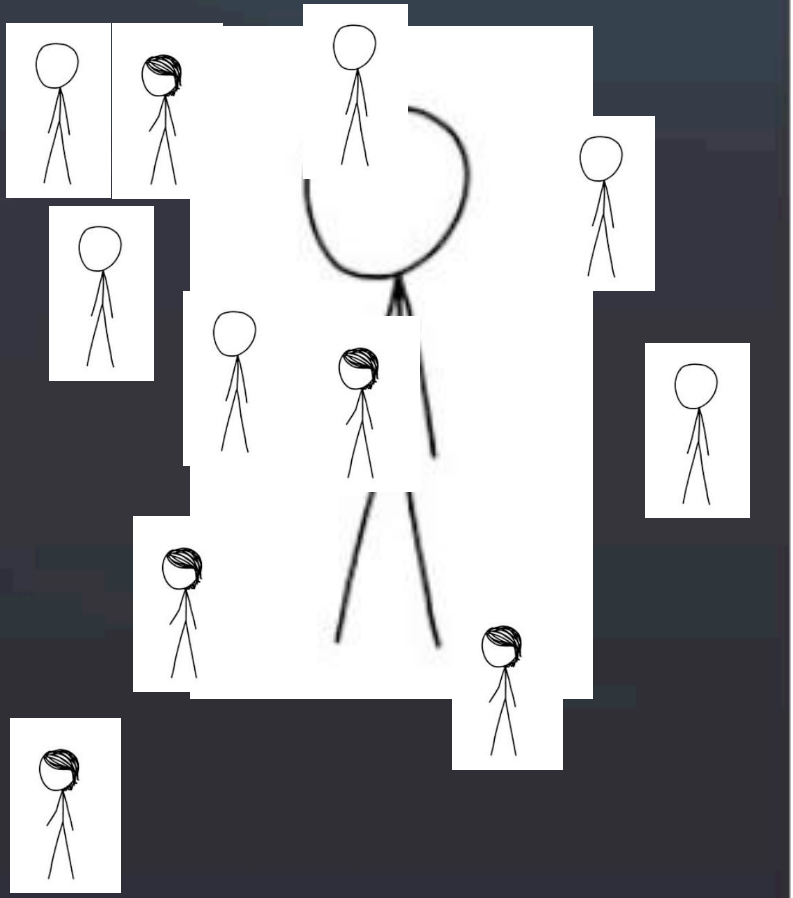
“UserProfile” = <definition>

“Tweet” = <definition>

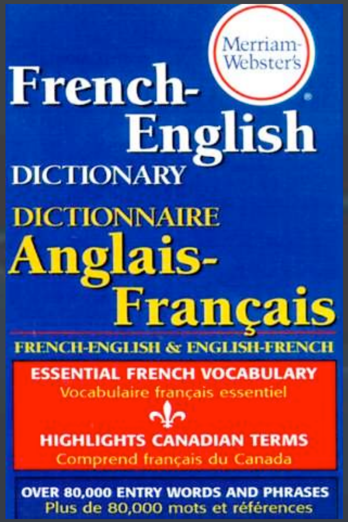
“Alert” = <definition>




Merriam-Webster's
French-English
DICTIONARY
DICTIONNAIRE
Anglais-Français
FRENCH-ENGLISH & ENGLISH-FRENCH
ESSENTIAL FRENCH VOCABULARY
Vocabulaire français essentiel
HIGHLIGHTS CANADIAN TERMS
Comprend français du Canada
OVER 80,000 ENTRY WORDS AND PHRASES
Plus de 80,000 mots et références



Schema registry


Merriam-Webster's
French-English DICTIONARY
DICTIONNAIRE Anglais-Français
FRENCH-ENGLISH & ENGLISH-FRENCH
ESSENTIAL FRENCH VOCABULARY
Vocabulaire français essentiel
HIGHLIGHTS CANADIAN TERMS
Comprend français du Canada
OVER 80,000 ENTRY WORDS AND PHRASES
Plus de 80,000 mots et références

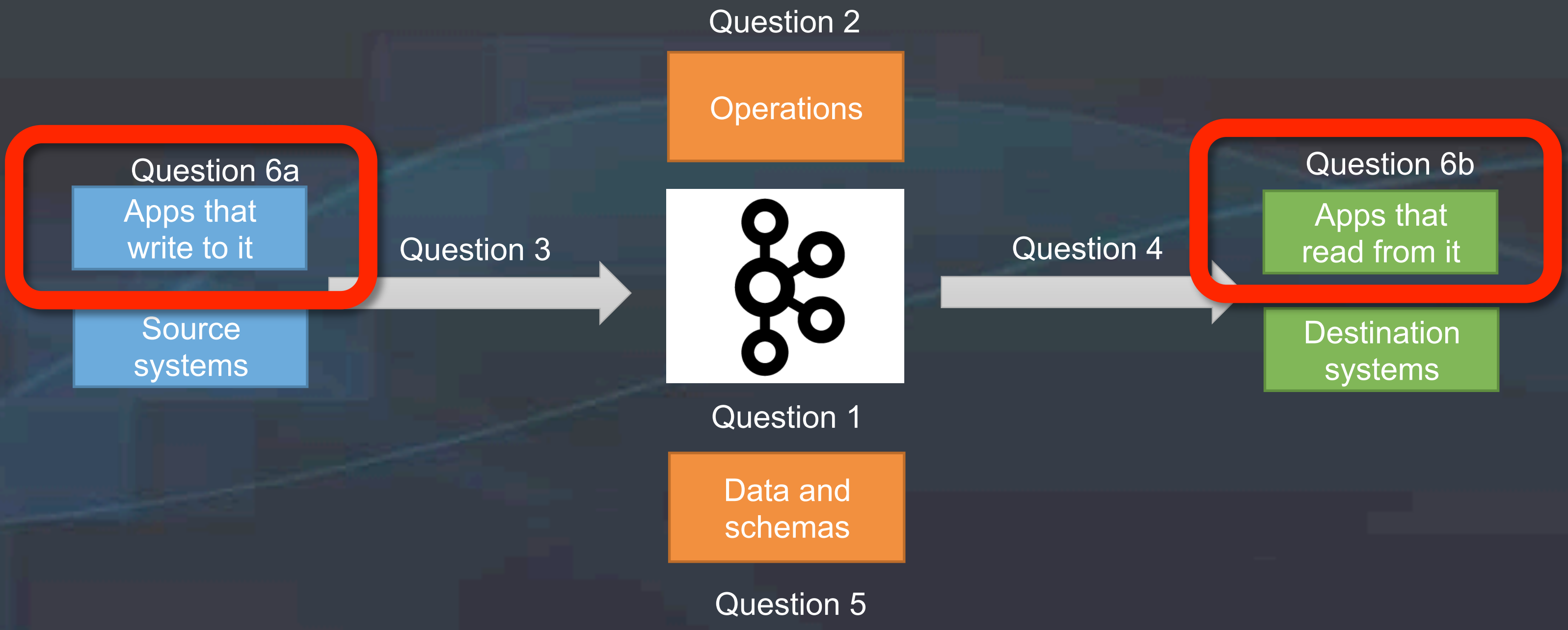
- Stores and retrieves your schemas
- Cornerstone for building resilient data pipelines
- Viable registry implementation missing until recently
  - AVRO-1124 (2012-2014)
  - So everyone started to roll their own schema registry
    - Again: been there, done that ☺
- There must be a better way, right?

Confluent Schema Registry

- Open source, included in Confluent Platform https://github.com/confluentinc/schema-registry/
- REST API to store and retrieve schemas etc.

```
# List all schema versions registered for topic "foo"
$ curl -X GET -i http://registry:8081/subjects/foo/versions
```

- Integrates with Kafka clients, Kafka REST, Camus, ...
- Can enforce policies for your data, e.g. backwards compatibility
- Still not convinced you need a schema registry?
  - http://www.confluent.io/blog/schema-registry-kafka-stream-processing-yes-virginia-you-really-need-one



Stream processing

Question 6 or “How do I actually process my data in Kafka?”

confluent

Stream processing

- Currently three main options
    - Storm: arguably powering the majority of production deployments
    - Spark Streaming: runner-up, but gaining momentum due to “main” Spark
    - DIY: write your own using Kafka client libs, typically with a narrower focus

Some people, when confronted with a problem to process data in Kafka, think “I know, I’ll use [Storm | Spark | …].”

Now they have *two* problems.

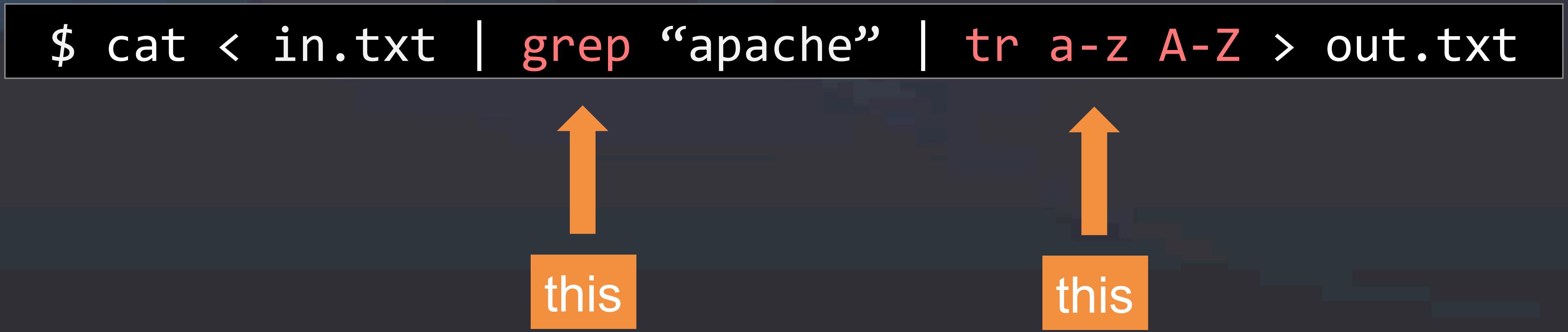
Stream processing

Four!

- Currently ~~three~~ main options
  - Storm: arguably powering the majority of production deployments
  - Spark Streaming: runner-up, but gaining momentum due to “main” Spark
  - DIY: write your own using Kafka client libs, typically with a narrower focus

- Kafka 0.9.0 will introduce **Kafka Streams**

Kafka Streams is the commands in your Unix pipelines.
Use it to transform data stored in Kafka.



```
$ cat < in.txt | grep “apache” | tr a-z A-Z > out.txt
```

this

this

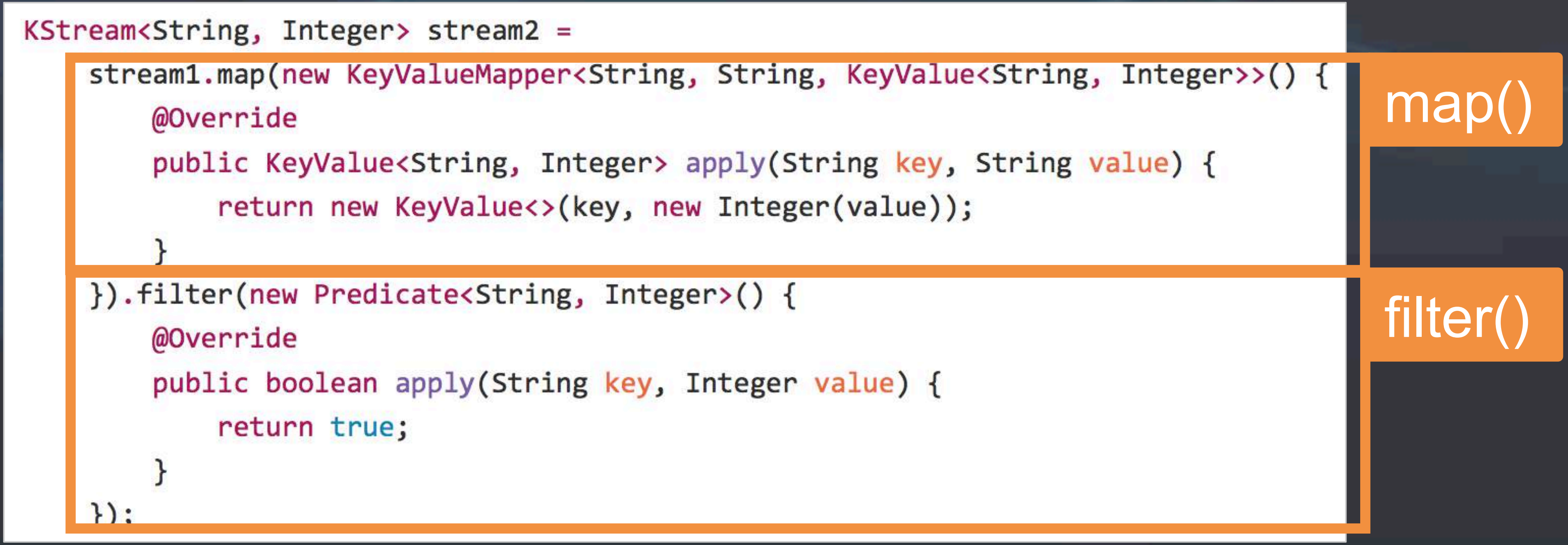
Kafka Streams

- Kafka Streams included in Kafka 0.9.0
    - KIP-28: https://cwiki.apache.org/confluence/display/KAFKA/KIP-28+-+Add+a+processor+client
- No need to run another framework like Storm alongside Kafka
    - No need for separate infrastructure and trainings either
- **Library** rather than framework
    - Won’t dictate your deployment, configuration management, packaging, …
    - Use it like you’d use Apache Commons, Google Guava, etc.
    - Easier to integrate with your existing apps and services
- 100% compatible with Kafka by definition

Kafka Streams

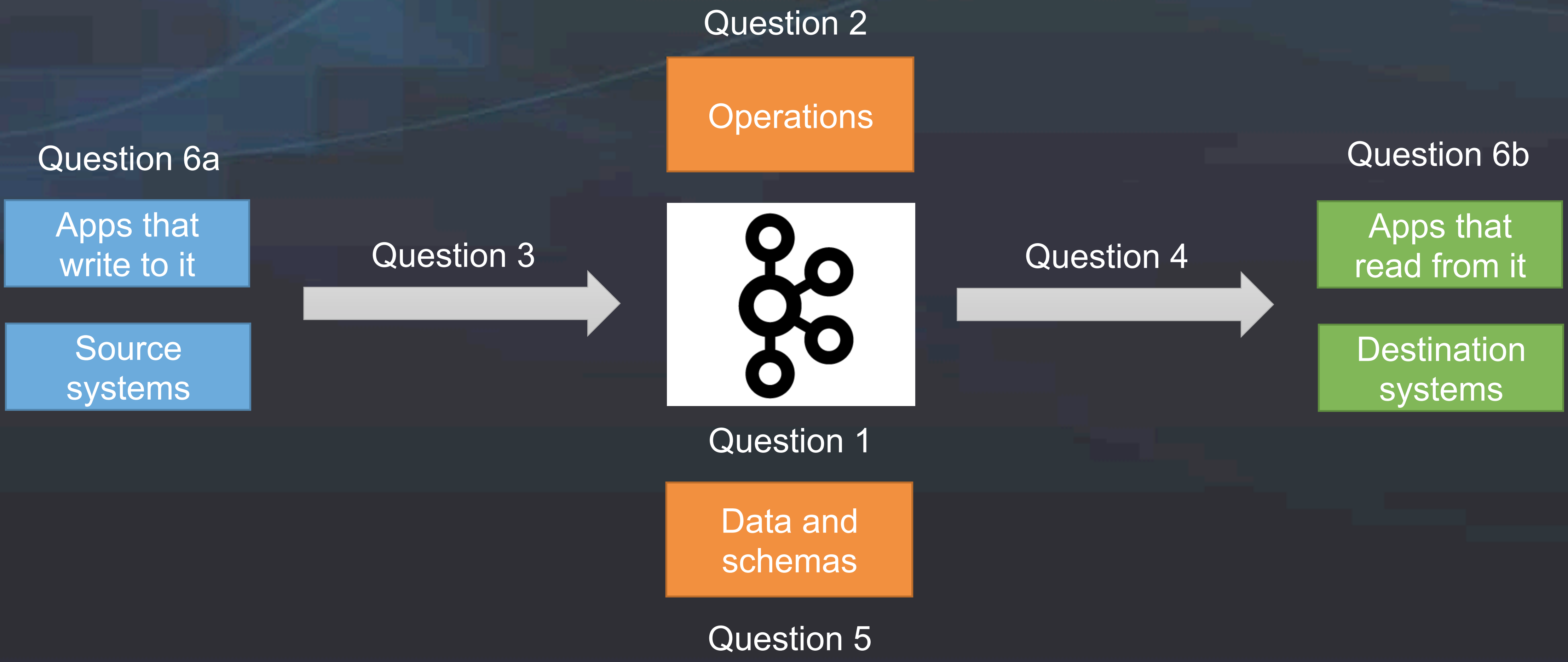
- Initial version will support
  - Low-level API as well as higher-level API for Java 7+
  - Operations such as `join/filter/map/`…
  - Windowing
  - Proper time modeling, e.g. event time vs. processing time
  - Local state management with persistence and replication
  - Schema and Avro support
- And more to come – details will be shared over the next weeks!

Example of higher-level API (much nicer with Java 8 and lambdas)



```
KStream<String, Integer> stream2 =
    stream1.map(new KeyValueMapper<String, String, KeyValue<String, Integer>>() {
        @Override
        public KeyValue<String, Integer> apply(String key, String value) {
            return new KeyValue<>(key, new Integer(value));
        }
    }).filter(new Predicate<String, Integer>() {
        @Override
        public boolean apply(String key, Integer value) {
            return true;
        }
    });
```

Phew, we made it!



Question 2

Operations

Question 6a

Apps that write to it

Source systems

Question 3

Question 1

Data and schemas

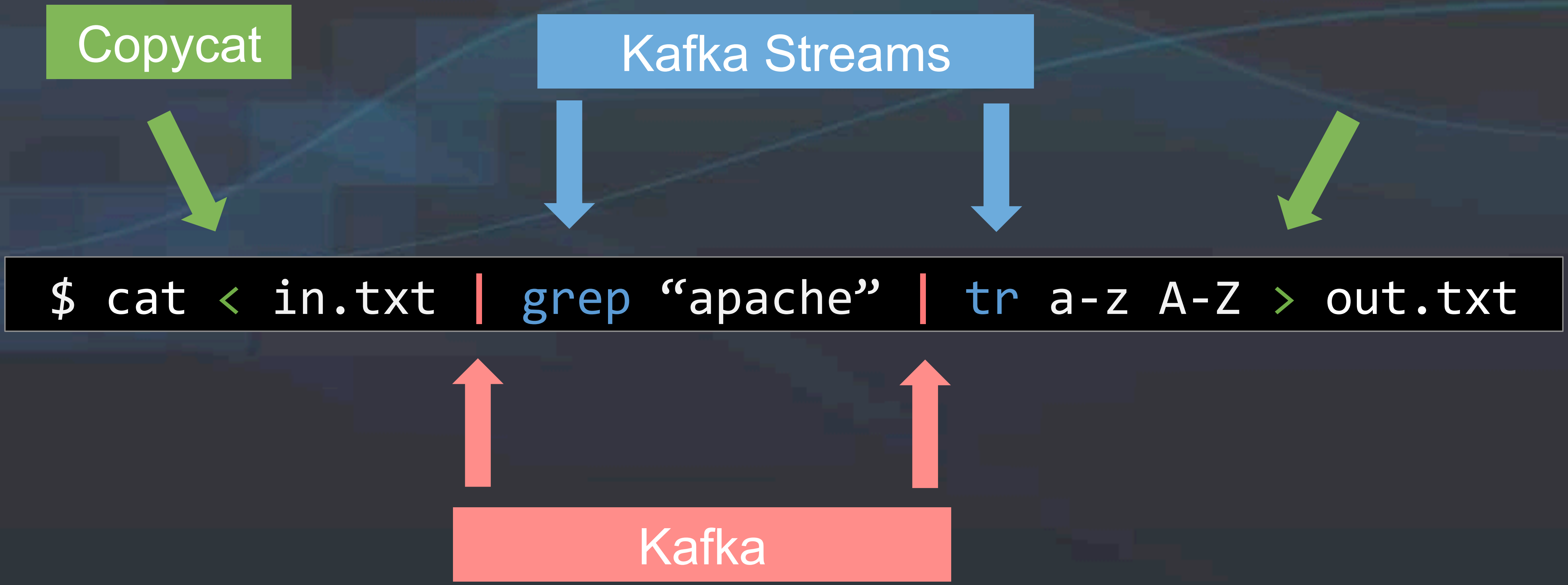
Question 5

Question 4

Question 6b

Apps that read from it

Destination systems



Copycat

Kafka Streams

```
$ cat < in.txt | grep “apache” | tr a-z A-Z > out.txt
```

Kafka

Want to contribute to Kafka and open source?

Join the Kafka community
http://kafka.apache.org/

…in a great team with the creators of Kafka
and also getting paid for it?

Confluent is hiring ☺
http://confluent.io/

Questions, comments? Tweet with #ApacheBigData and /cc to @ConfluentInc