

Data Streaming and Analytic Microservices

Stewart Bryson 

Stewart Bryson

Oracle ACE Director in BI/DI ♠

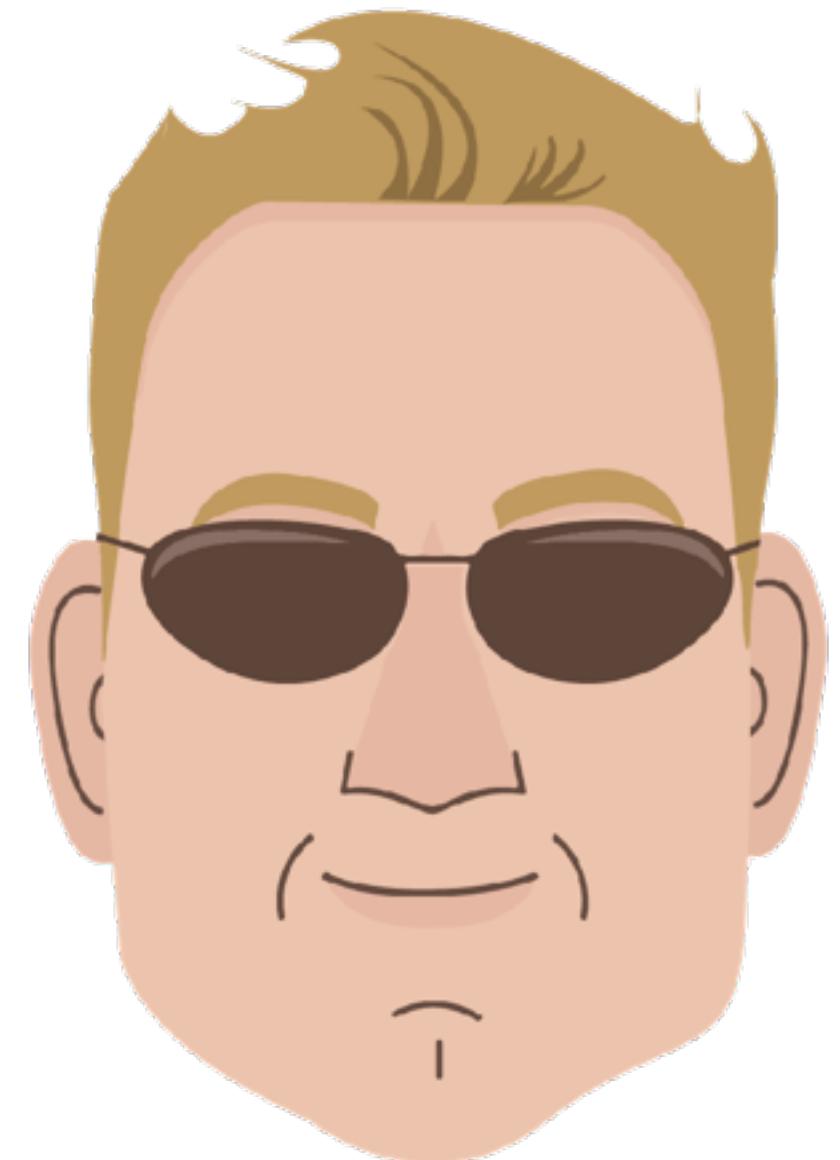
20 Years in Analytics, Business
Intelligence and Data Warehousing

Owner & Co-founder of Red Pill
Analytics

 @stewartbryson

 medium.com/@stewartbryson

 linkedin.com/in/stewartbryson



YOU CAN
CHOOSE
TO SEE
DATA
DIFFERENTLY

a RED
PILL
ANALYTICS



A complete DevOps solution for simplifying Oracle BI development

Our custom-built solution adds full source control, true multi-user and multi-workstream development, release build, unit testing, and migration automation to OBIEE. Our Capacity Analytics service uses Checkmate to optimize delivery, but Checkmate can also be implemented standalone on-premises or hosted by us. Checkmate has the following features:



Check-in & Automate

Continuous Integration

Automatically test, merge and deploy OBIEE content



True Multi-User Development

Stop struggling with multi-user development.



Hosted or On-Premise

License it on-premise or cloud-hosted.



Full Source Control

Git keeps track of every little change.



CAPACITY ANALYTICS

A new approach for efficiently and effectively delivering BI & Analytics

Let our team be your team. Our unique cloud approach and Agile methodology allows you to choose your development team size and begin delivering quickly, just like with Software as a Service.



BI Development as a Service

Choose small, medium, or large development capacity.



Continuous Integration

We use our own Checkmate offering to develop efficiently



Agile Development

Release new content every 4-6 weeks, not every 4-6 months.



Support

When you need expert help to fix production issues, give us a call.

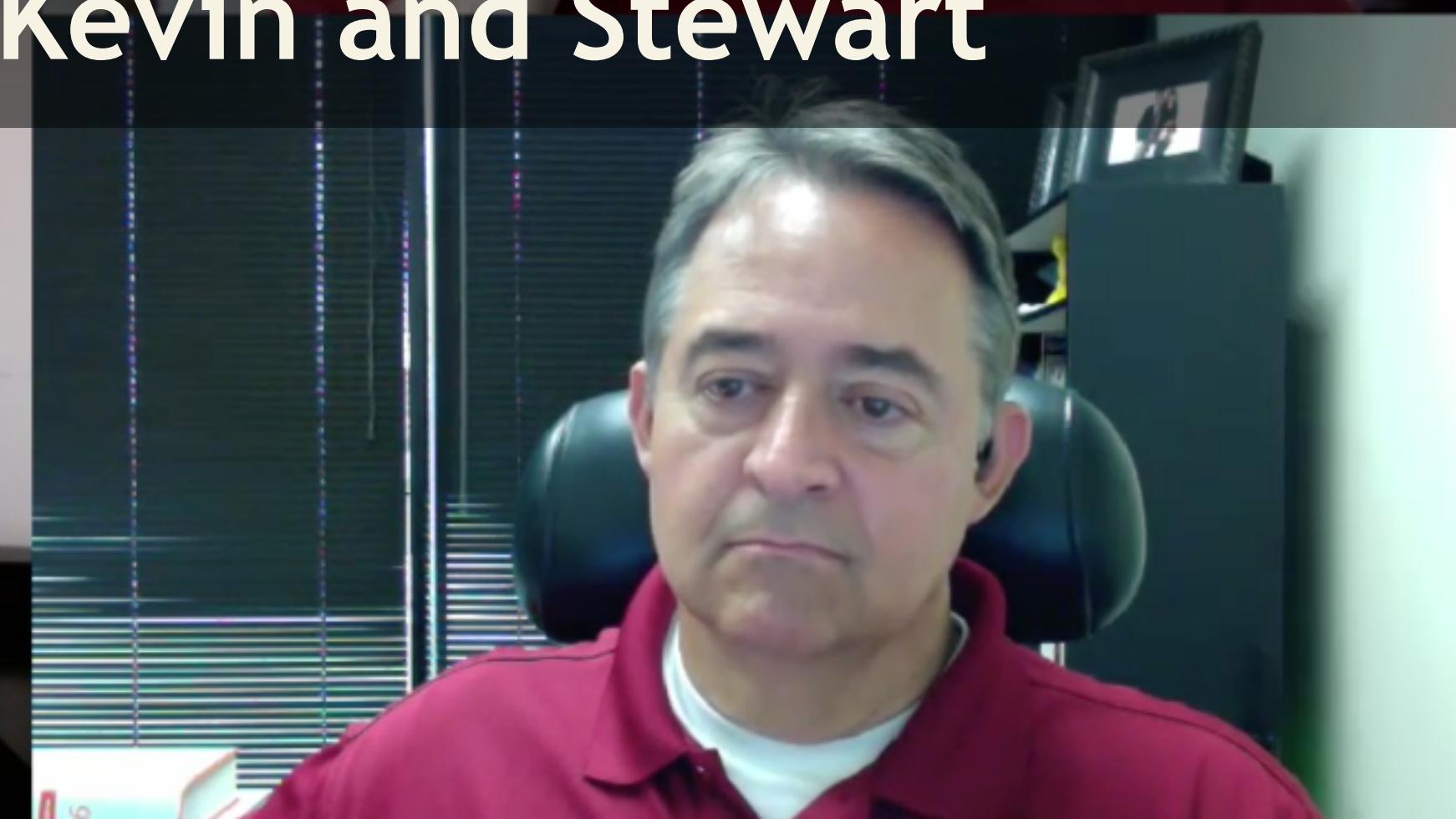


Cloud Enabled

Deliver faster with cloud-based development environments.

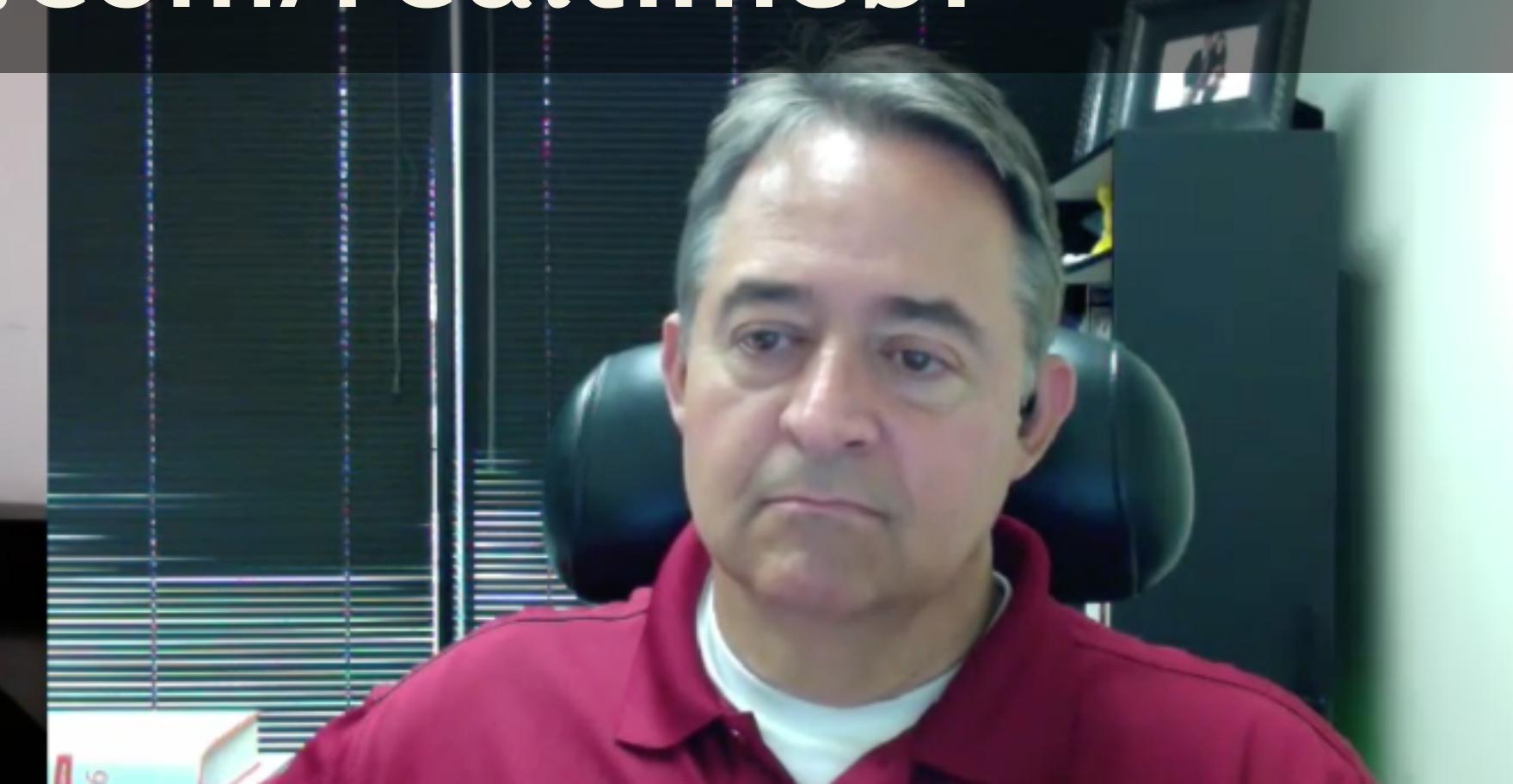


Realtime BI with Kevin and Stewart





www.youtube.com/realtimеби

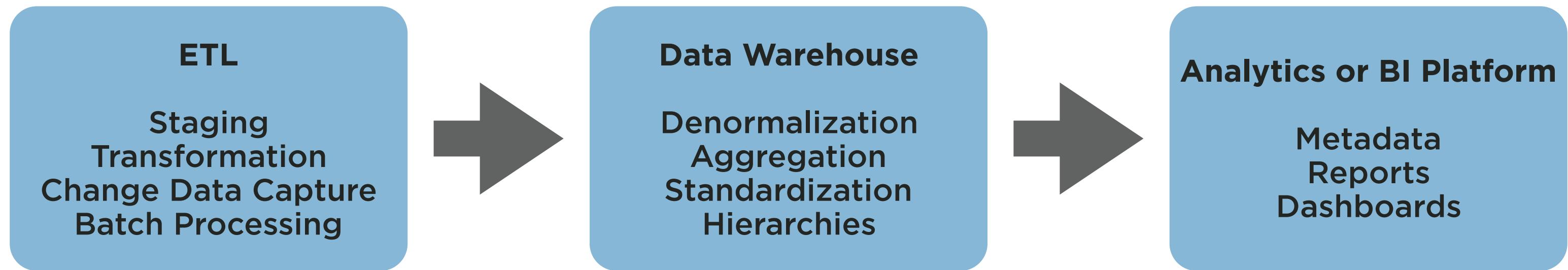


History Lesson





Traditional Data Warehouse



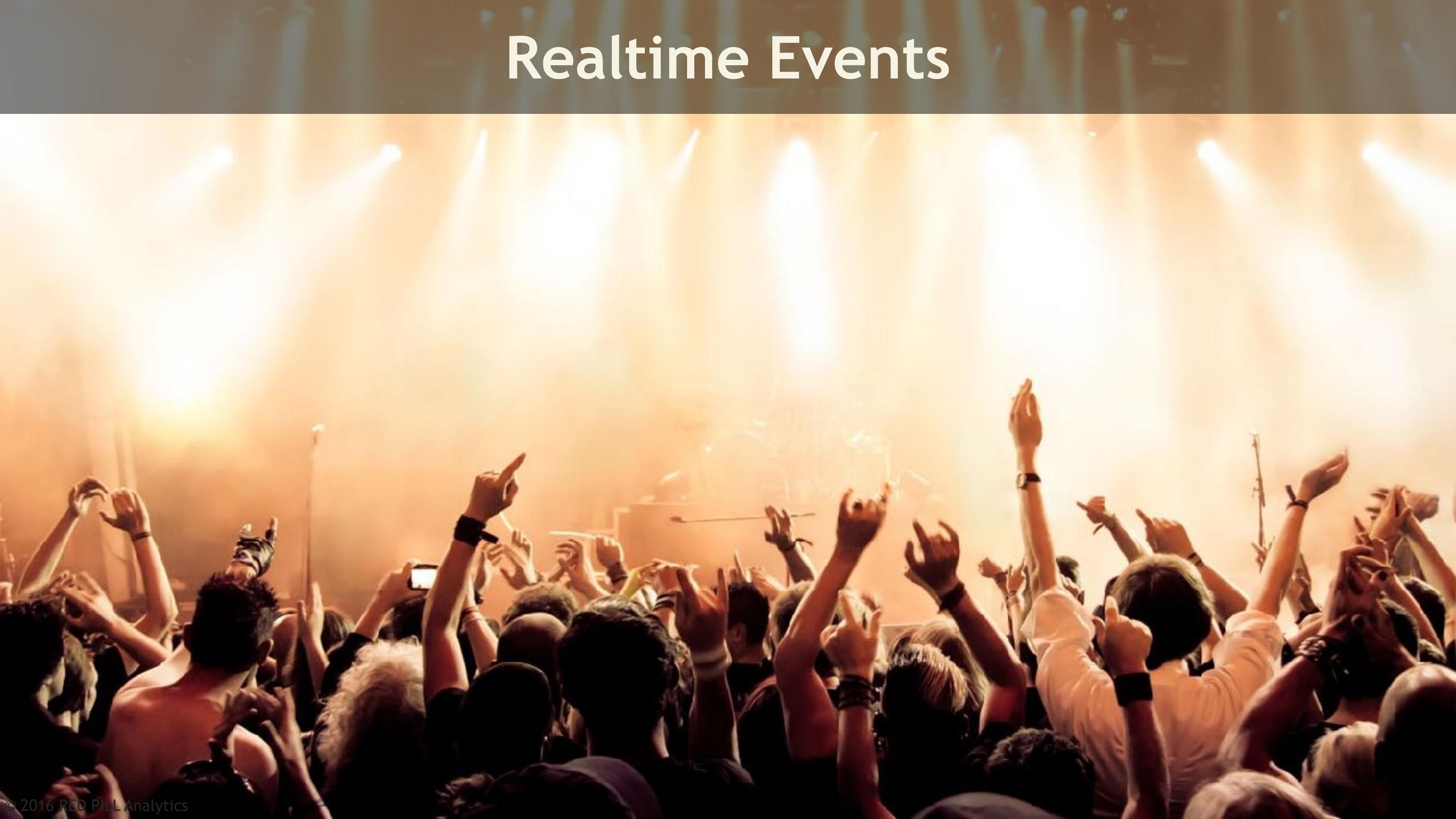


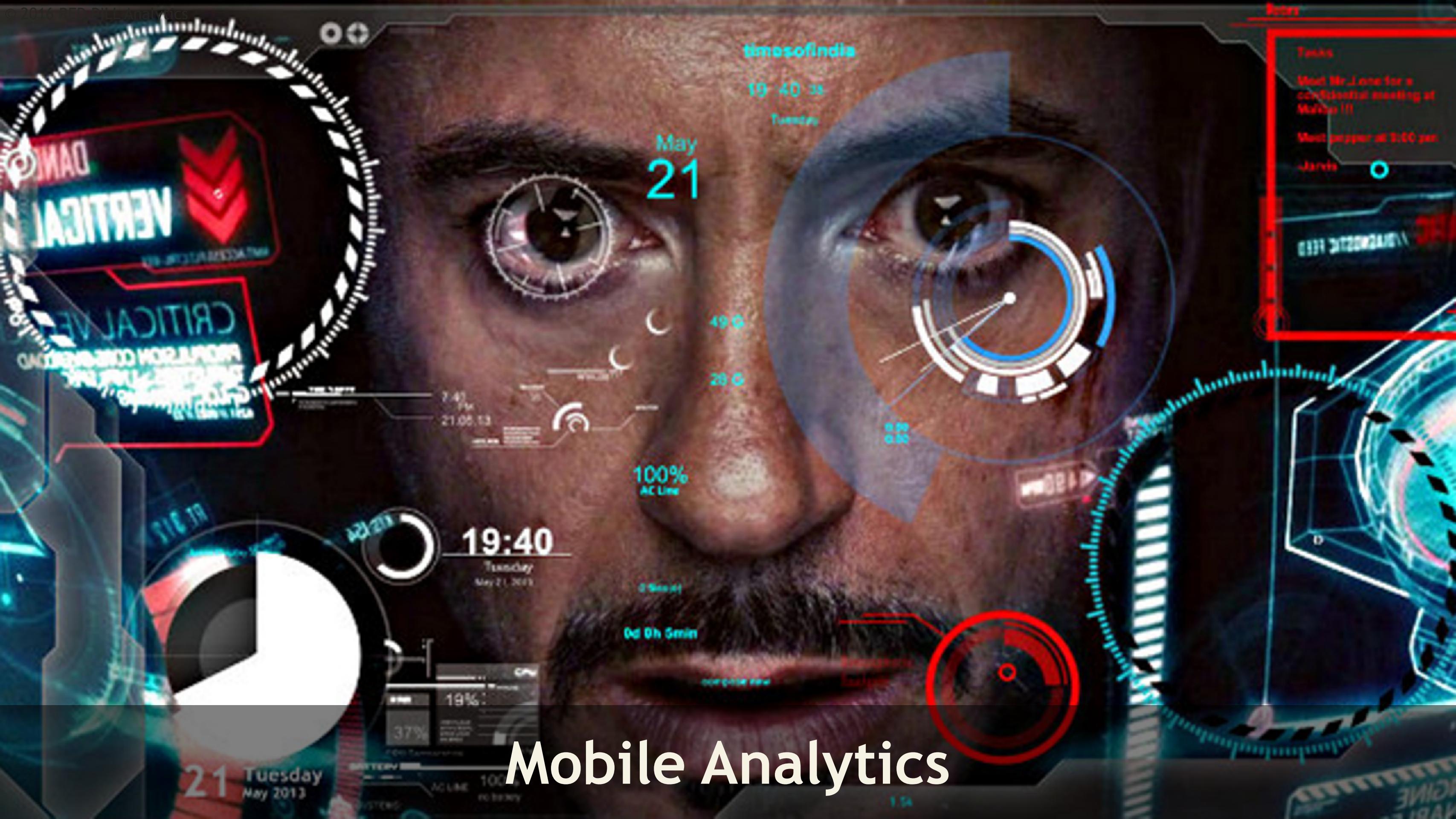
Resistance is Futile



**What are some use cases that
challenged this paradigm?**

Realtime Events

A photograph capturing a vibrant concert scene. The foreground is filled with the silhouettes of a large crowd, their hands raised high in the air, some pointing towards the stage. The stage itself is a brilliant, overexposed white light, suggesting intense spotlights or a firework display. The overall atmosphere is one of excitement and energy.



Mobile Analytics

Search





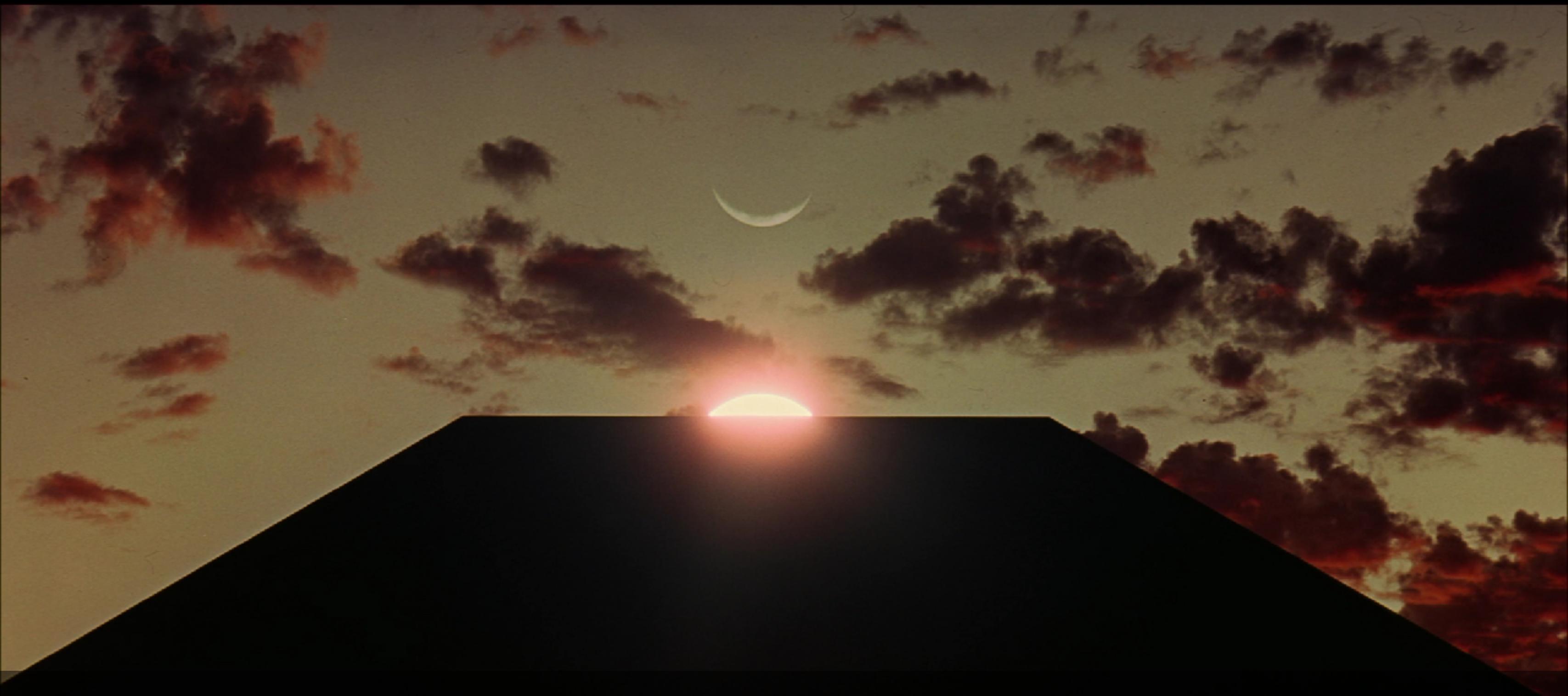
Machine Learning



Microservices

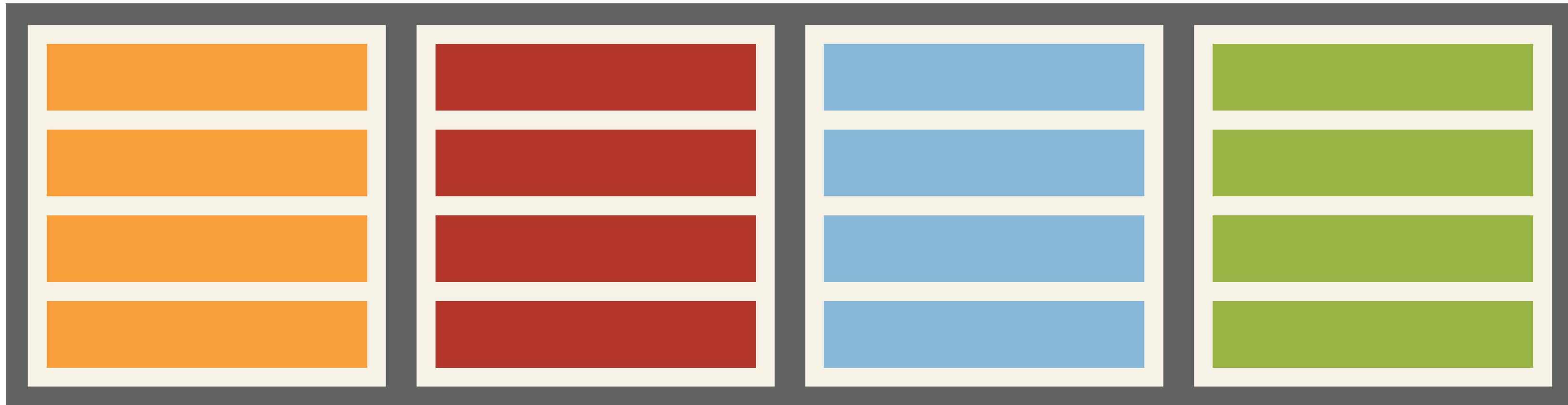
The *microservice* architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery.

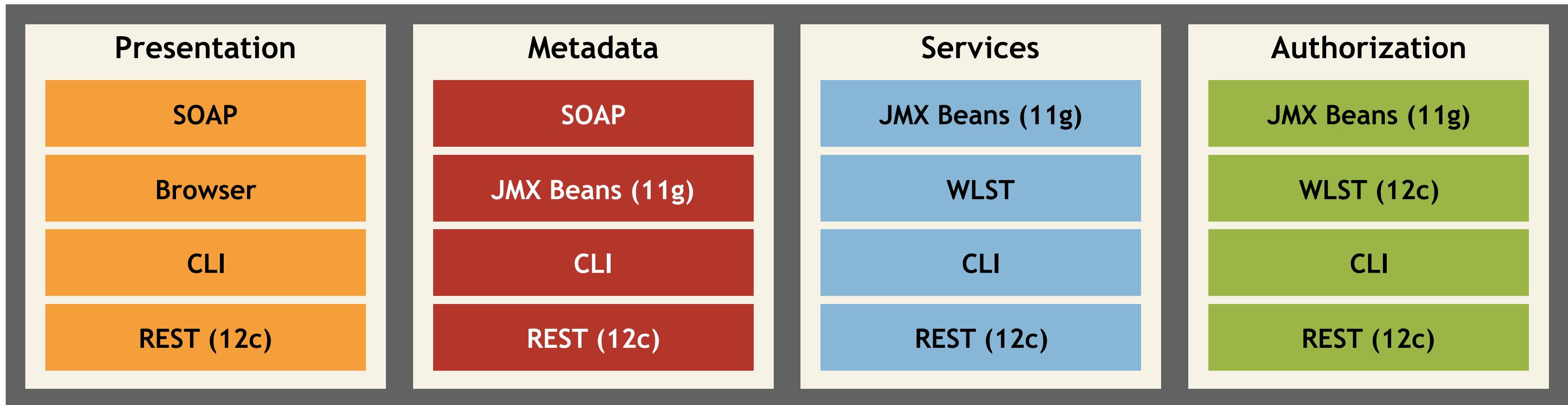
— Martin Fowler

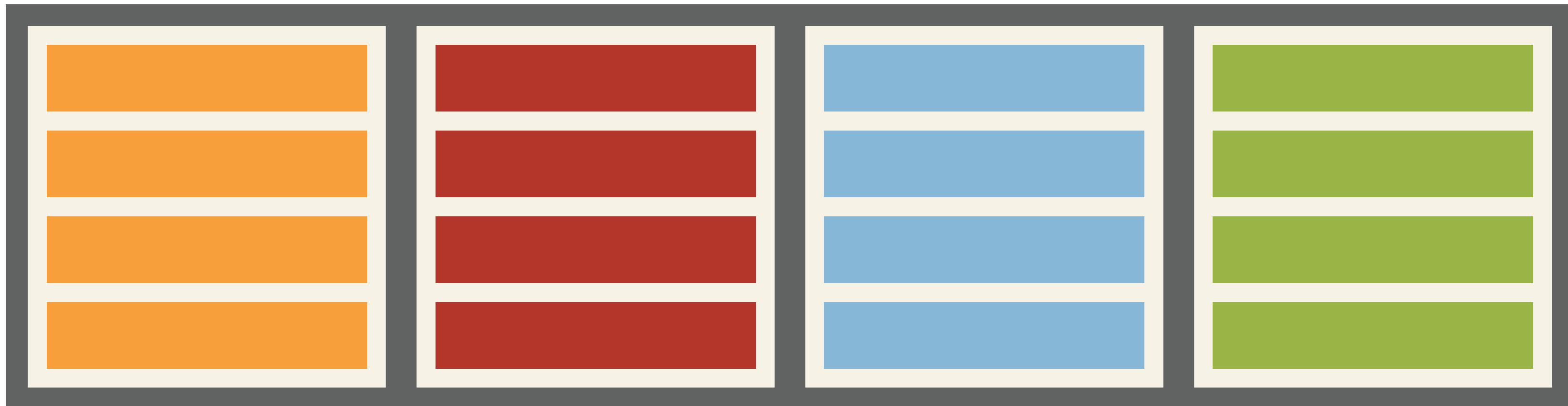


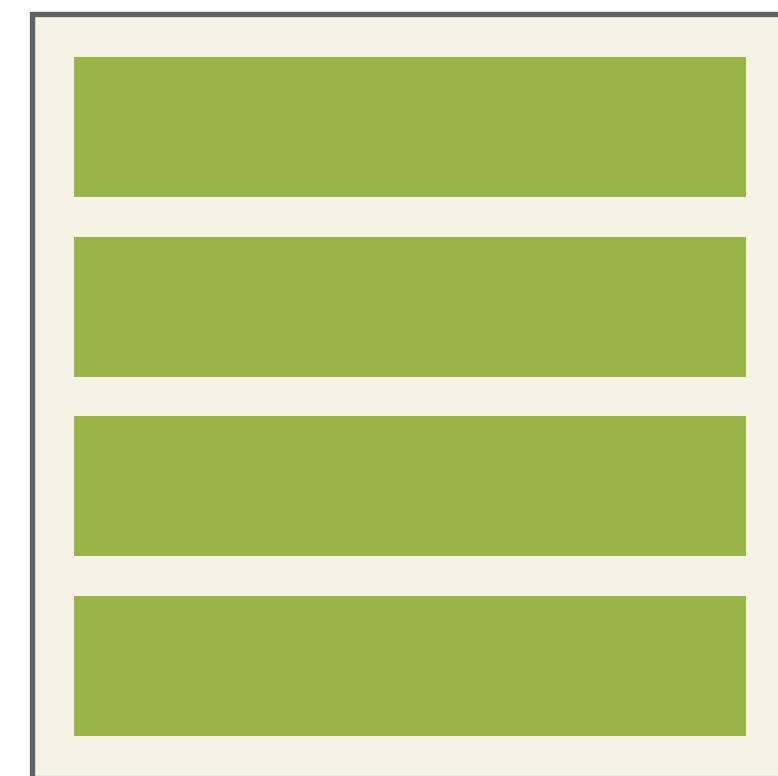
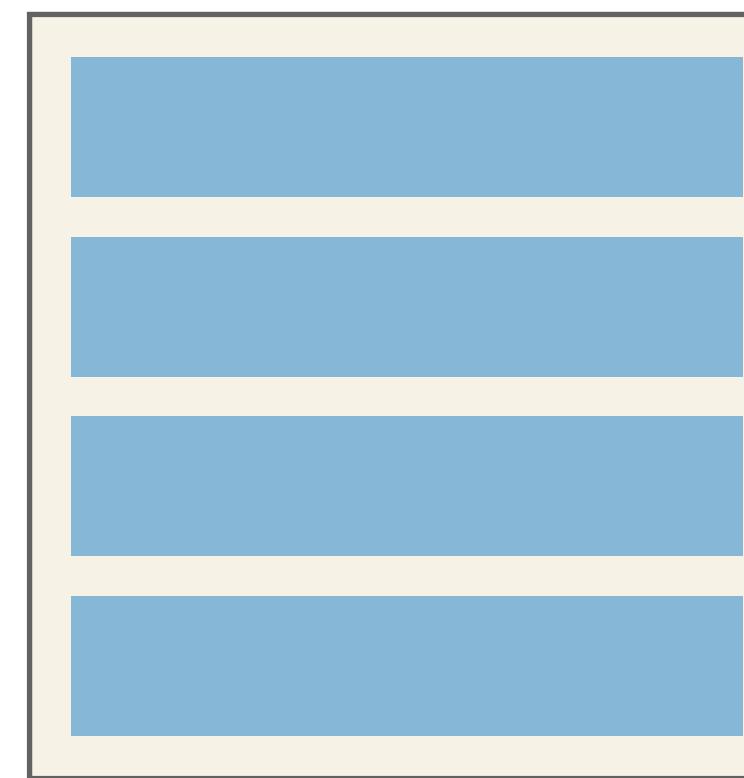
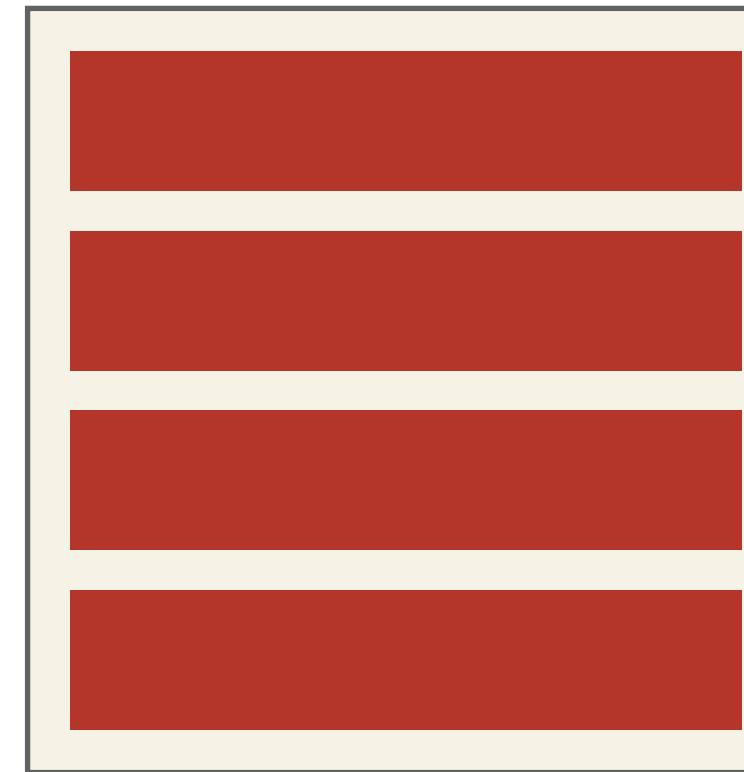
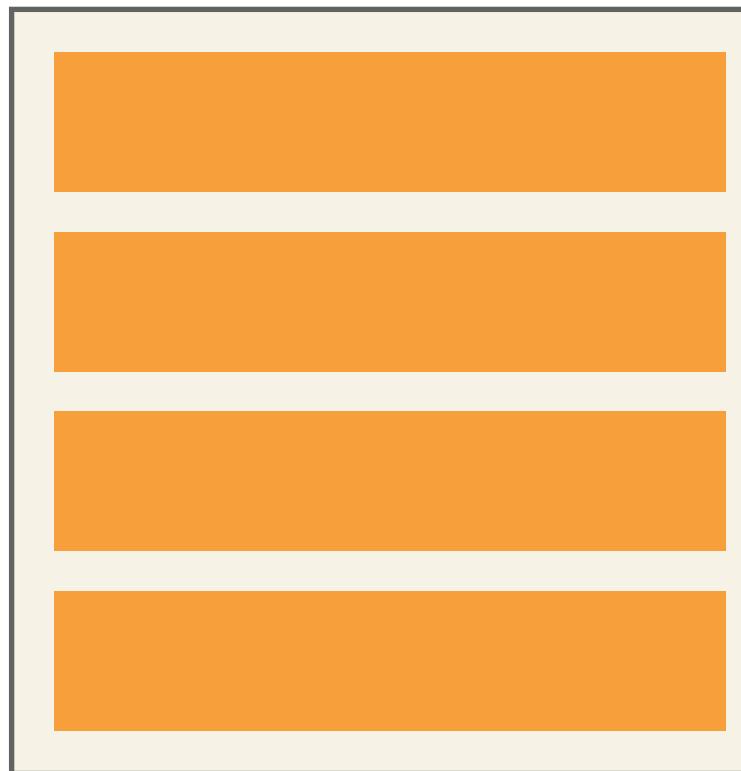
The Monolith

Monolith versus Microservice









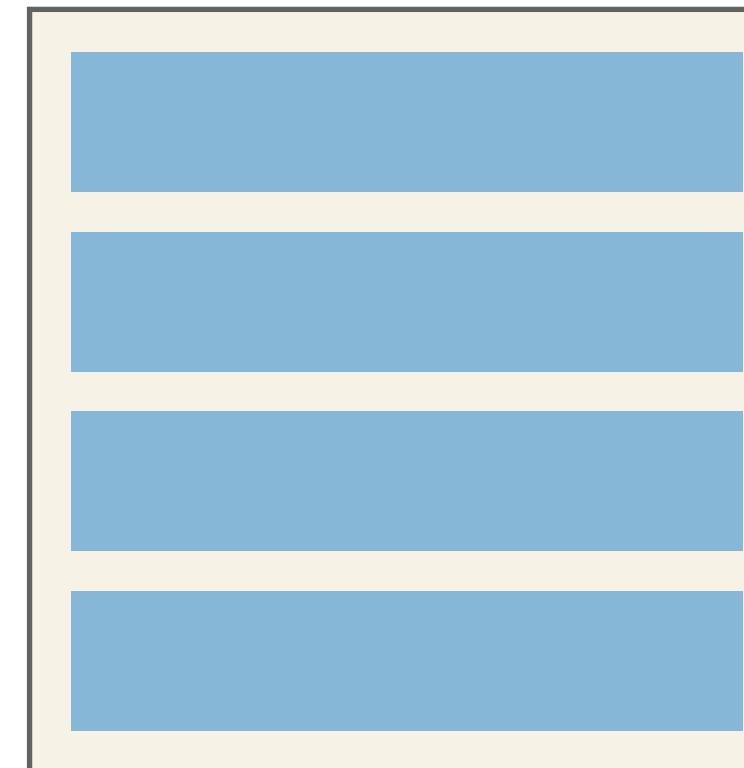
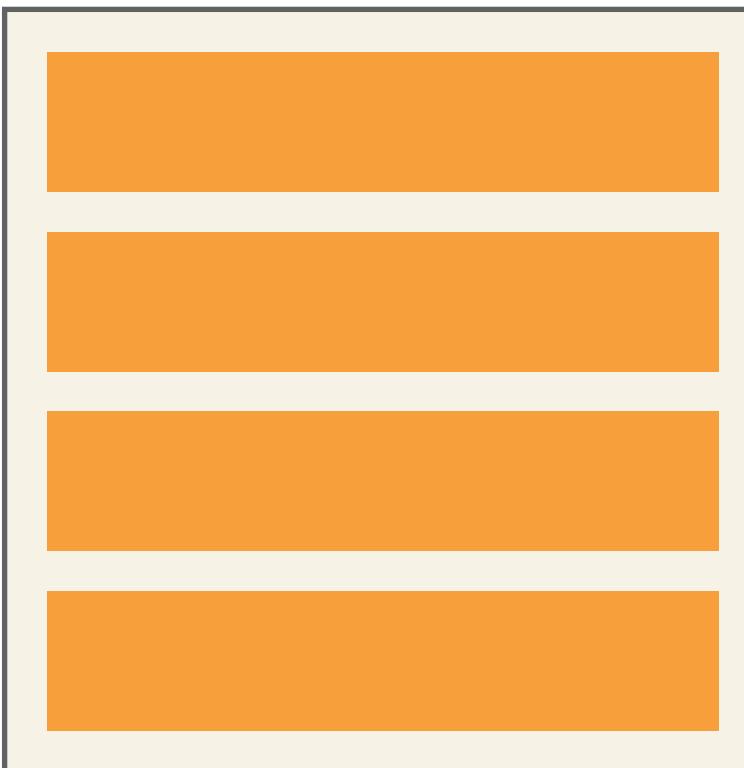
Organized Around Capabilities

Elastic for
Search

Cassandra for
Mobile Analytics

Spark for
Machine Learning

Spark for
Realtime Events

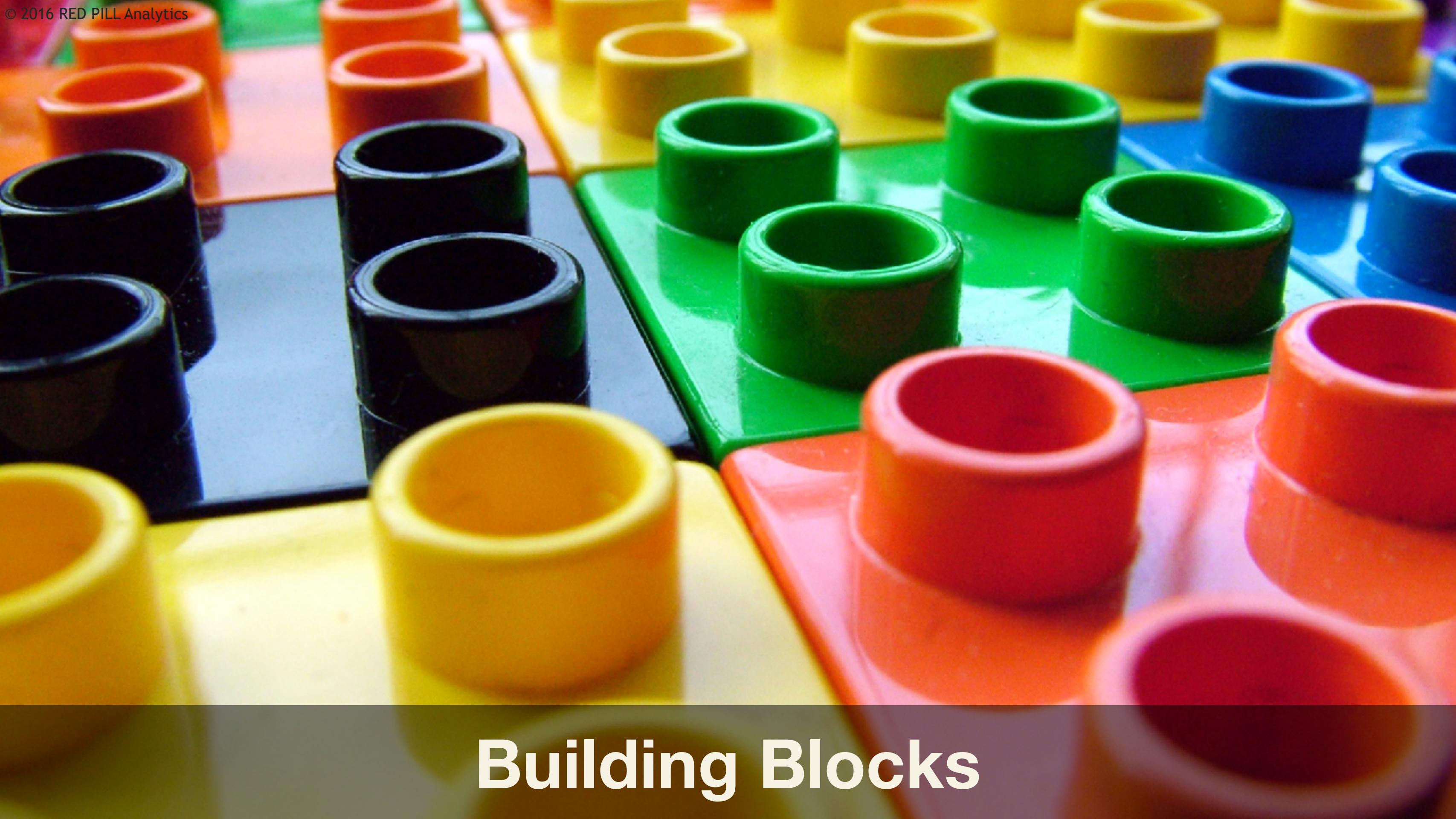




Separate



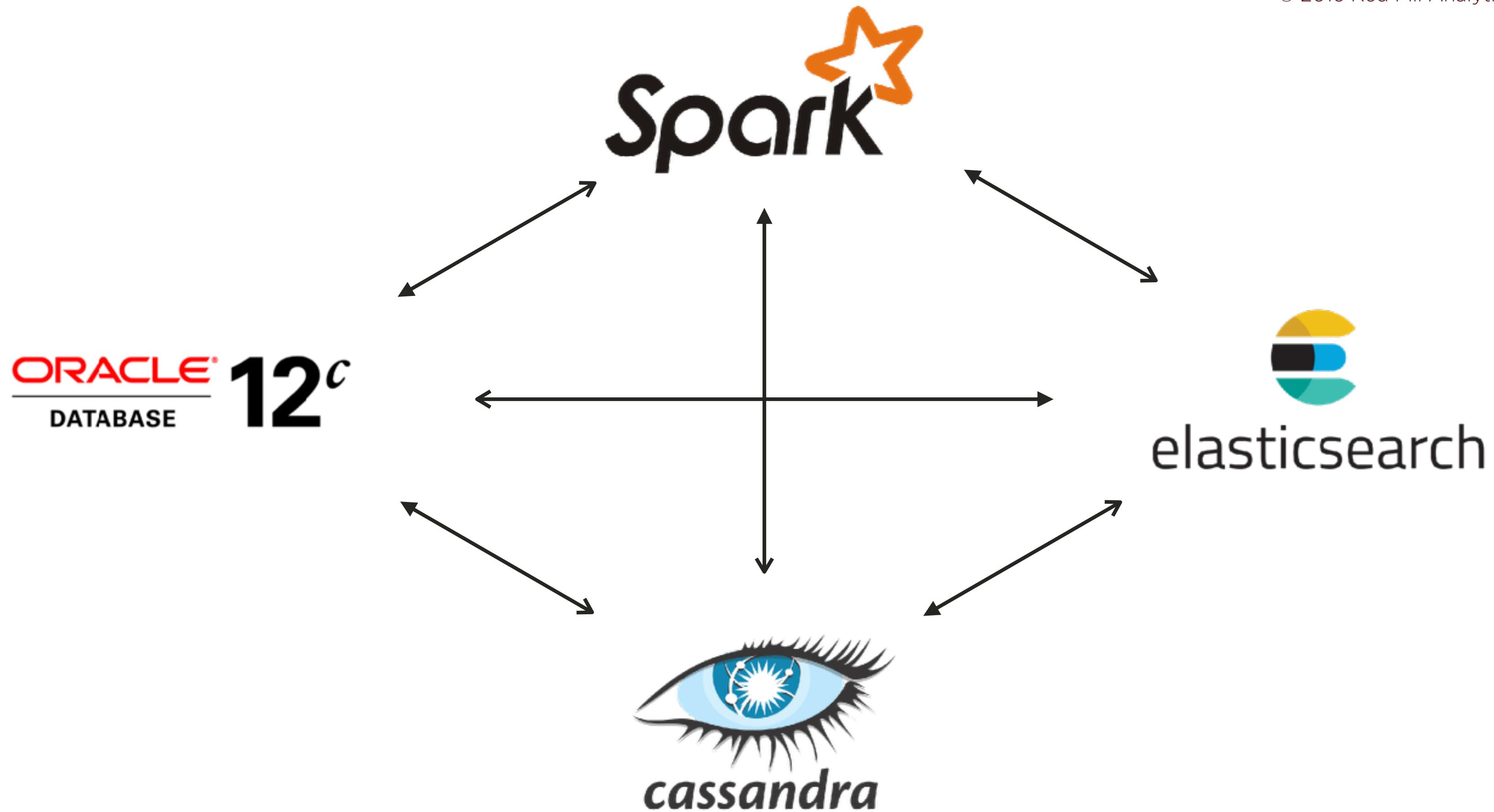
Connected

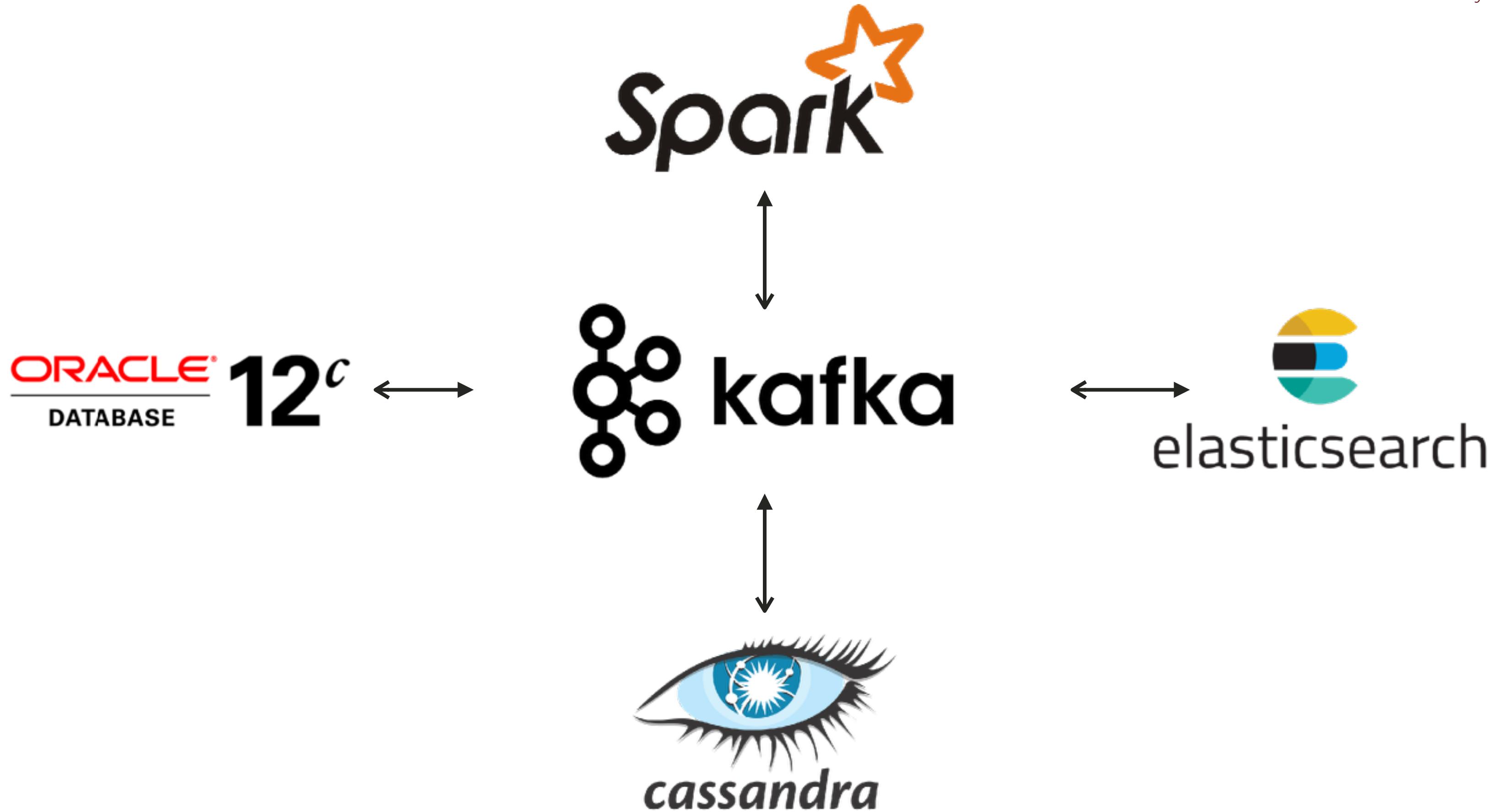


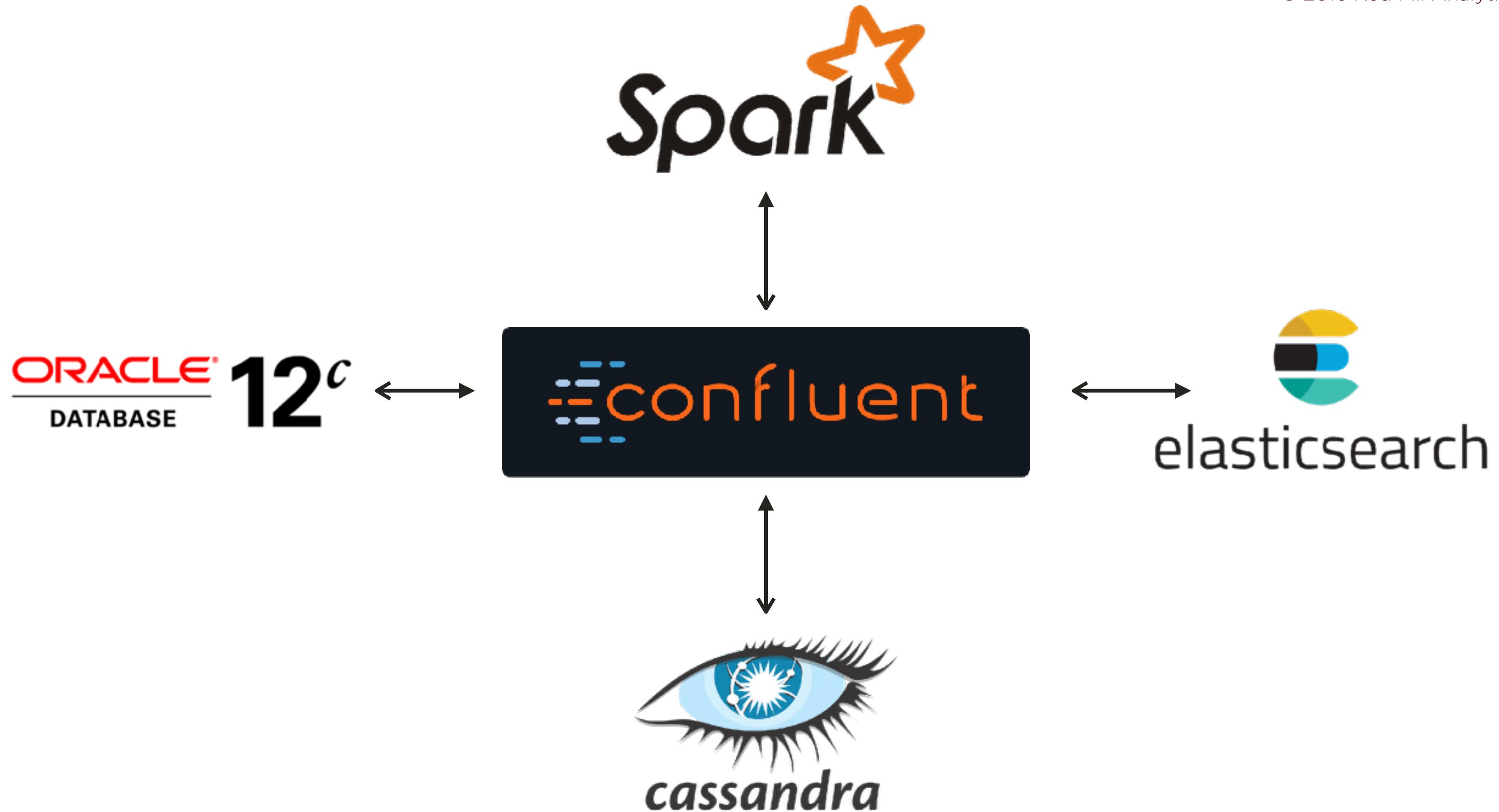
Building Blocks



Why Is This Relevant?









Apache Kafka



Managed Apache Kafka in the Cloud.

Oracle Event Hub Cloud Service delivers the power of Kafka as a managed streaming data platform integrated with the rest of Oracle's Cloud.



Fast.

Instant creation, scaling of Topics – ready for action in seconds.

Managed.

Oracle manages the Kafka infrastructure, while you leverage the power and simplicity of the platform.

Flexible.

Integrate either using REST APIs or Native-Kafka APIs.

Cost Effective.

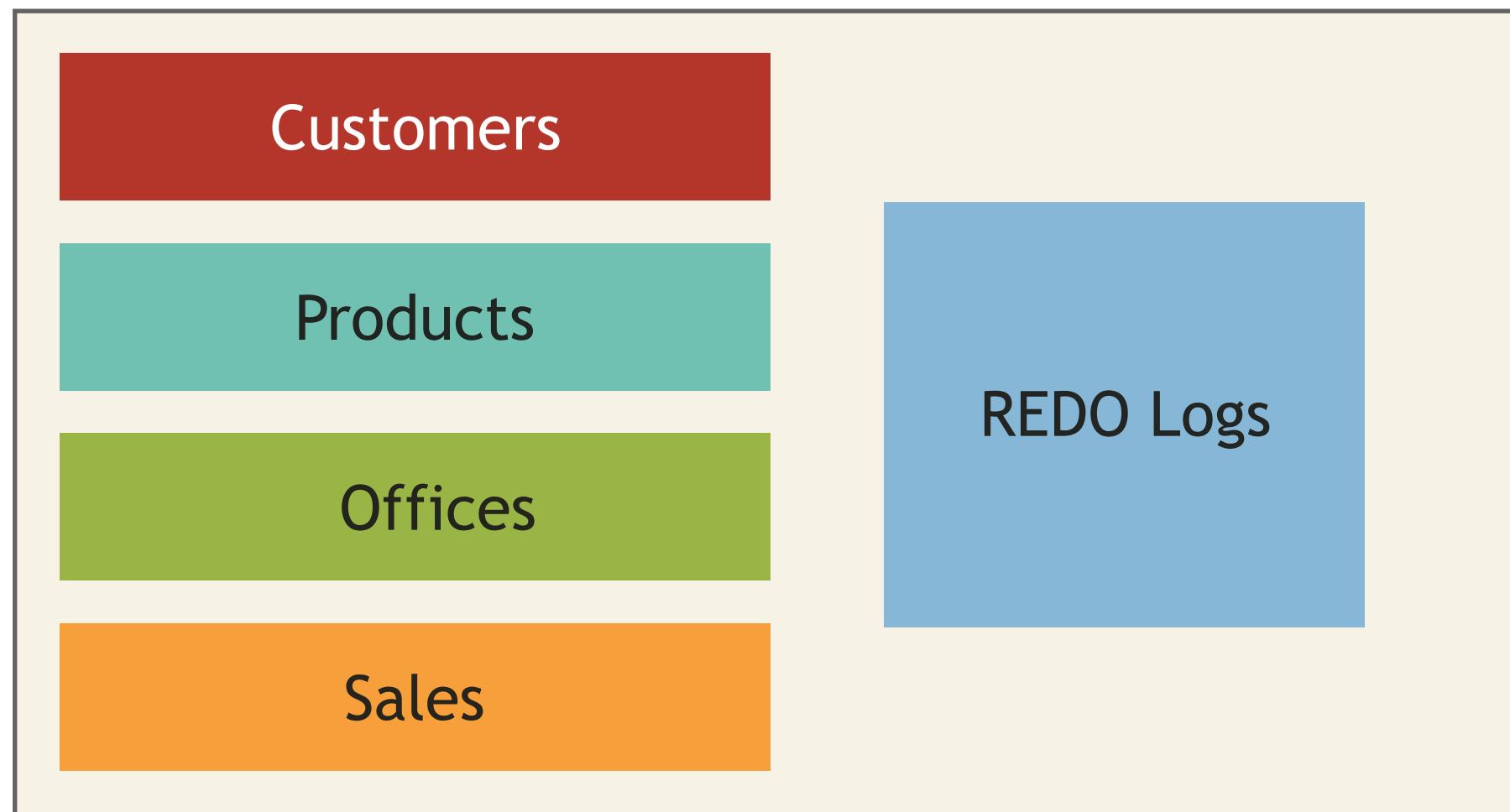
Scale from a few hundred operations a second to a few million operations per second.

Commit Log



Commit Log

ORACLE®
DATABASE **12^c**



Commit Log

| NAME | TICKER_SYMBOL | OWNERSHIP |
|--------------------|---------------|-----------|
| Red Pill Analytics | | Private |

| NAME | TICKER_SYMBOL | OWNERSHIP | SCN | COMMIT_DATE | DML_TYPE |
|--------------------|---------------|-----------|---------|---------------------|----------|
| Red Pill Analytics | | | 2992758 | 06/01/2014 12:00 AM | INSERT |

Commit Log

| NAME | TICKER_SYMBOL | OWNERSHIP |
|-------------------------|---------------|-----------|
| Red Pill Analytics, LLC | | Private |

| NAME | TICKER_SYMBOL | OWNERSHIP | SCN | COMMIT_DATE | DML_TYPE |
|-------------------------|---------------|-----------|---------|---------------------|----------|
| Red Pill Analytics | | | 2992758 | 06/01/2014 12:00 AM | INSERT |
| Red Pill Analytics, LLC | | Private | 2992760 | 07/14/2014 12:00 AM | UPDATE |

Commit Log

| NAME | TICKER_SYMBOL | OWNERSHIP |
|--------------------------|---------------|-----------|
| Red Pill Analytics, Inc. | RPAI | Public |

| NAME | TICKER_SYMBOL | OWNERSHIP | SCN | COMMIT_DATE | DML_TYPE |
|--------------------------|---------------|-----------|---------|---------------------|----------|
| Red Pill Analytics | | | 2992758 | 06/01/2014 12:00 AM | INSERT |
| Red Pill Analytics, LLC | | Private | 2992760 | 07/14/2014 12:00 AM | UPDATE |
| Red Pill Analytics, Inc. | RPAI | Public | 2992762 | 02/04/2017 12:00 AM | UPDATE |

Commit Log



ORACLE®
DATABASE **12^c**



Distributed Commit Log



Topic

Topic

Topic

Partition

Partition

Partition

Partition

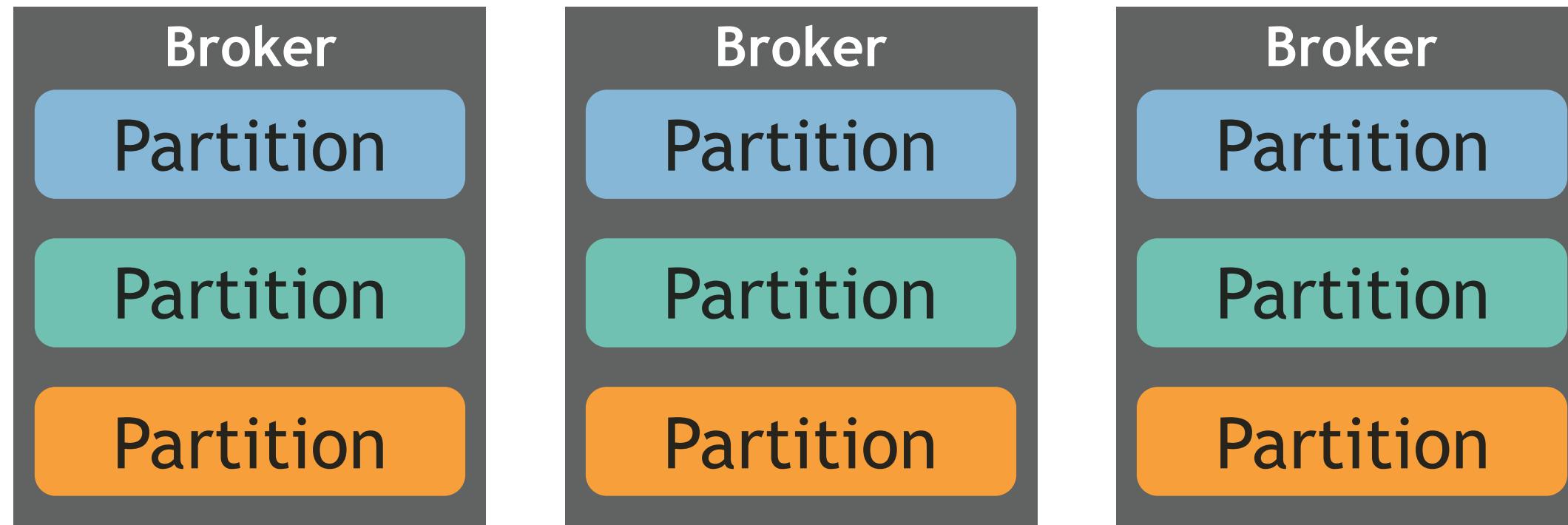
Partition

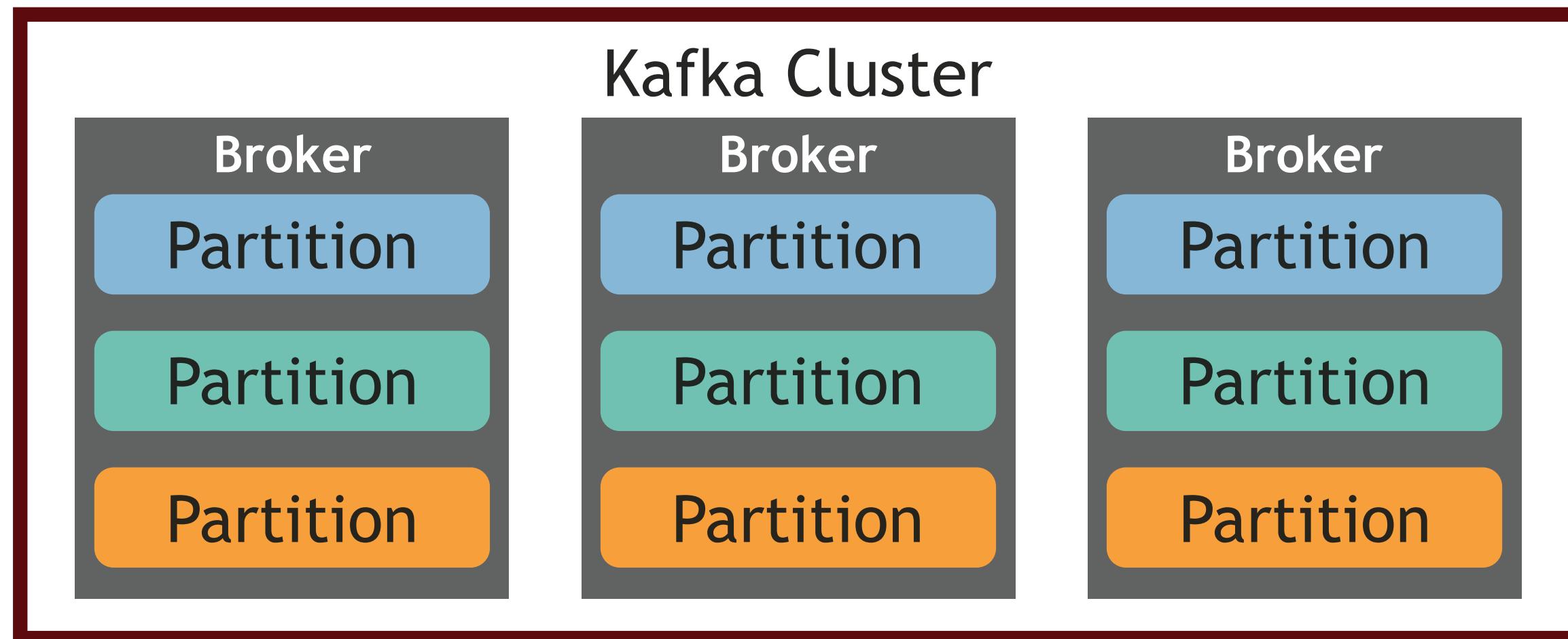
Partition

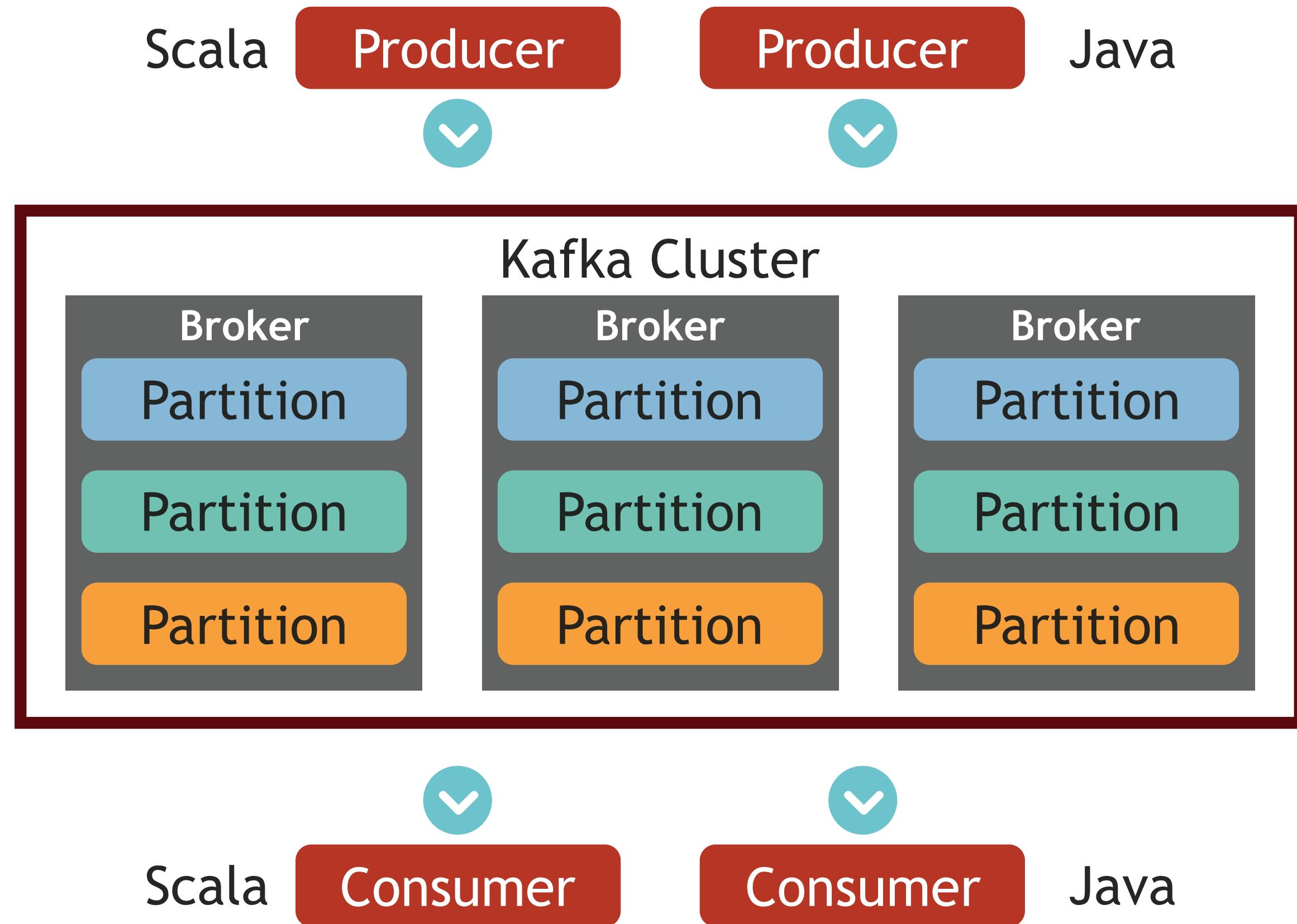
Partition

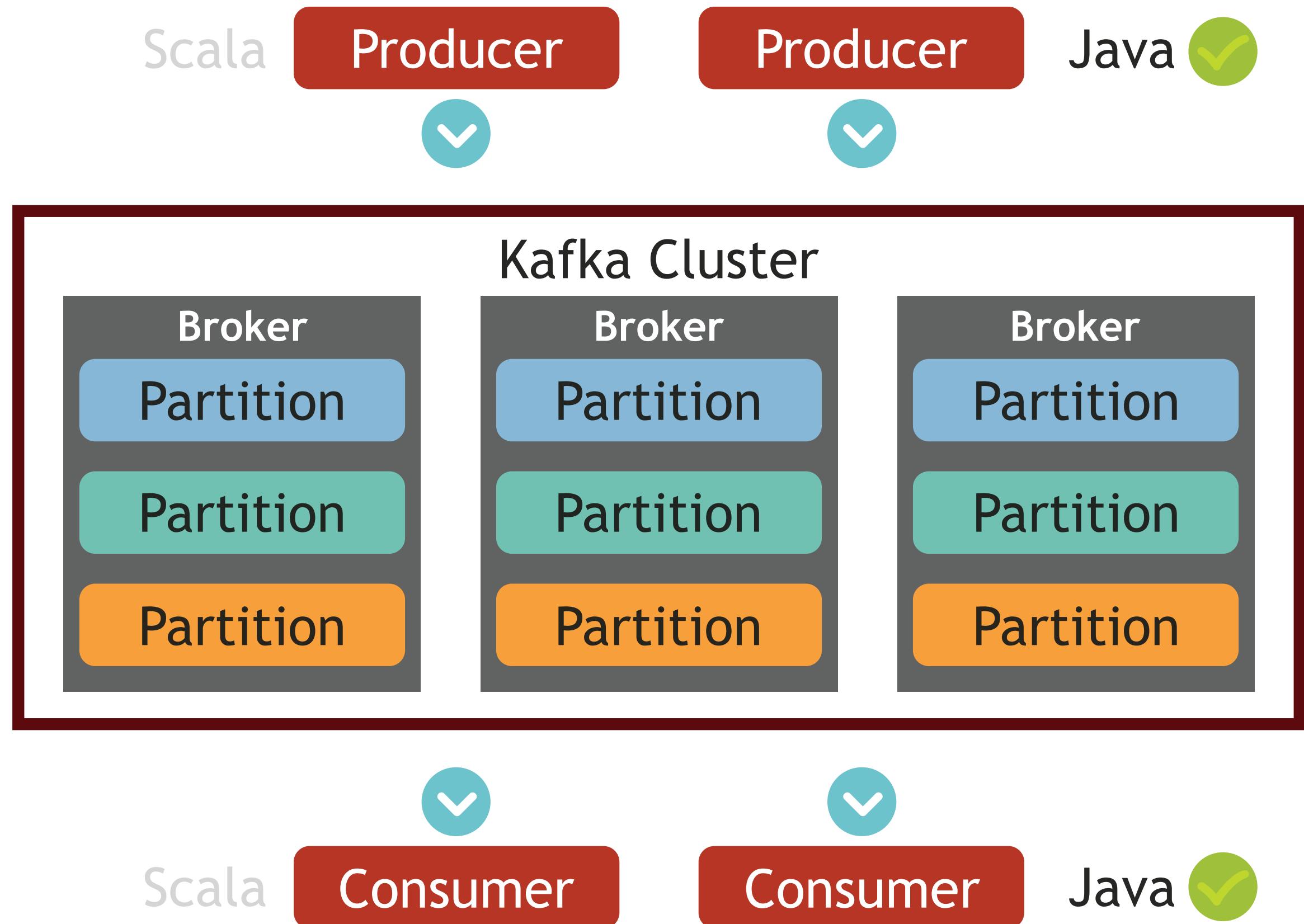
Partition

Partition









Kafka Connect (Source)

Producer

Producer

Java



Kafka Cluster

Broker

Partition

Partition

Partition

Broker

Partition

Partition

Partition

Broker

Partition

Partition

Partition

Consumer

Consumer

Java



Kafka Connect (Sink)

Kafka Connect (Source)

Producer

Producer

Java 



Kafka Cluster

Broker

Partition

Partition

Partition

Broker

Partition

Partition

Partition

Broker

Partition

Partition

Partition

Defined with
Connector **class**

Spawns Tasks and
Workers

Seamless
management of
offsets

Consumer

Consumer

Java 

Kafka Connect (Sink)



Standalone or
Distributed

What Goes in a Kafka Topic?



Q2





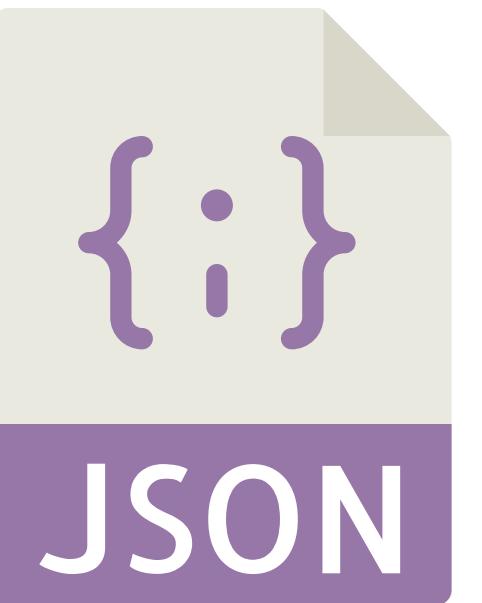
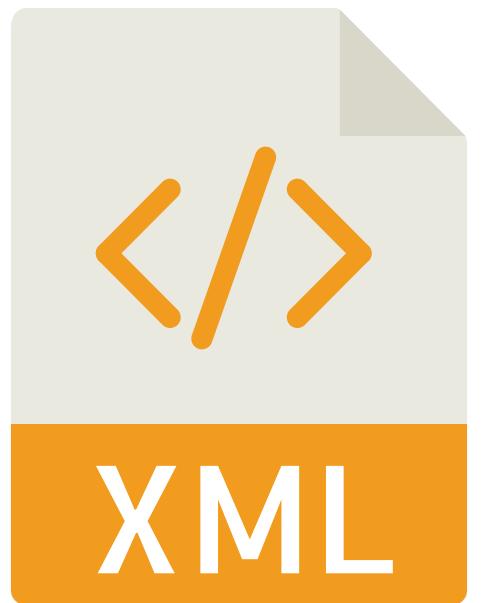
Schema on Read

Schema on Write



What format is my data in?

Any format you want!







Rich data structures

Compressible file format

Integrated with many
programming languages

Data structure (schema)
is stored with the data



Confluent Platform

What is a Streaming Platform?



Build streaming applications



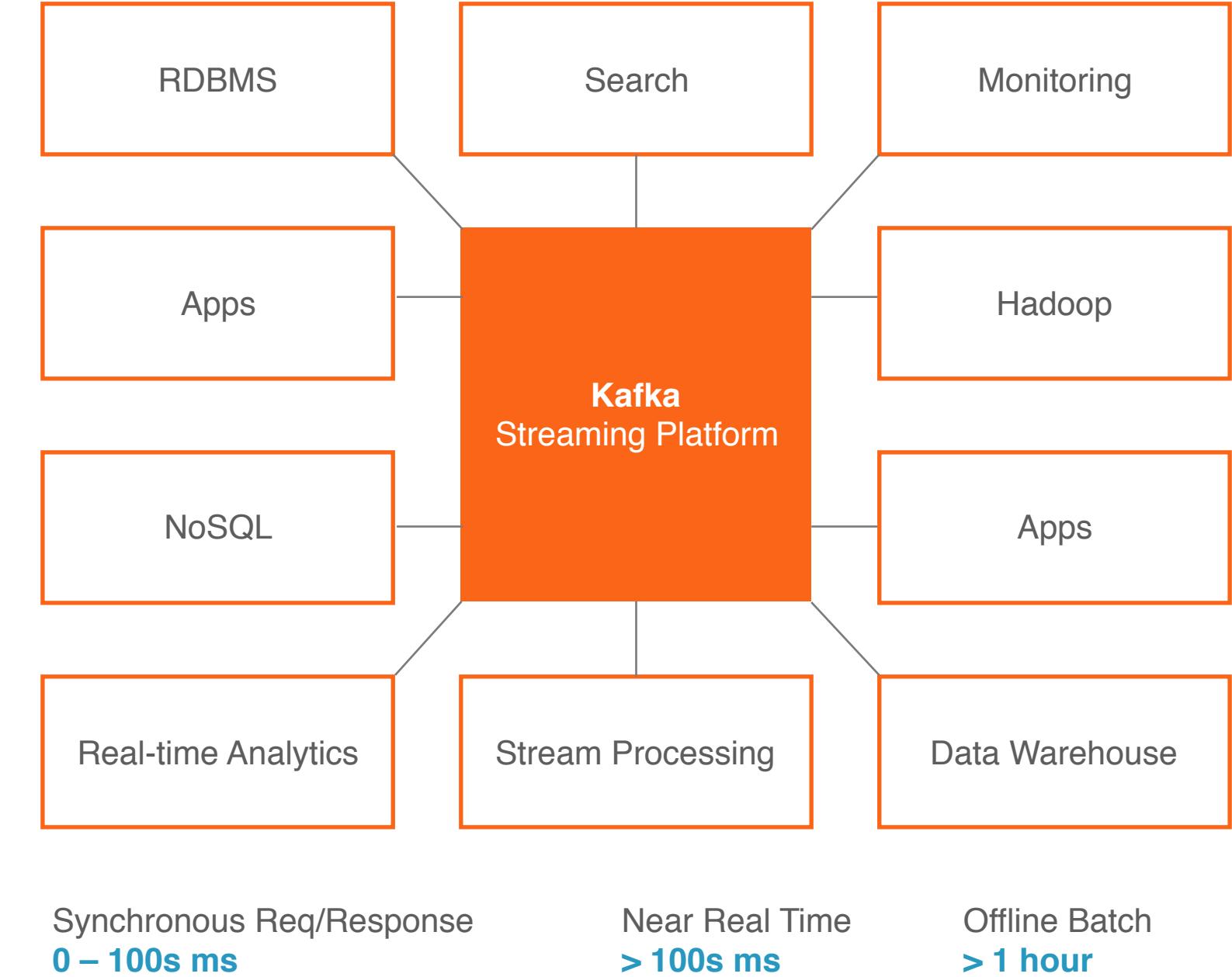
Deploy streaming applications at scale



Monitor and manage streaming applications

Common Kafka Use Cases

- Log data
- Database changes
- Sensors, device, IoT data
- Monitoring streams
- Call data records
- Real-time Monitoring
- Asynchronous applications
- Fraud and security
- Bridge to Cloud

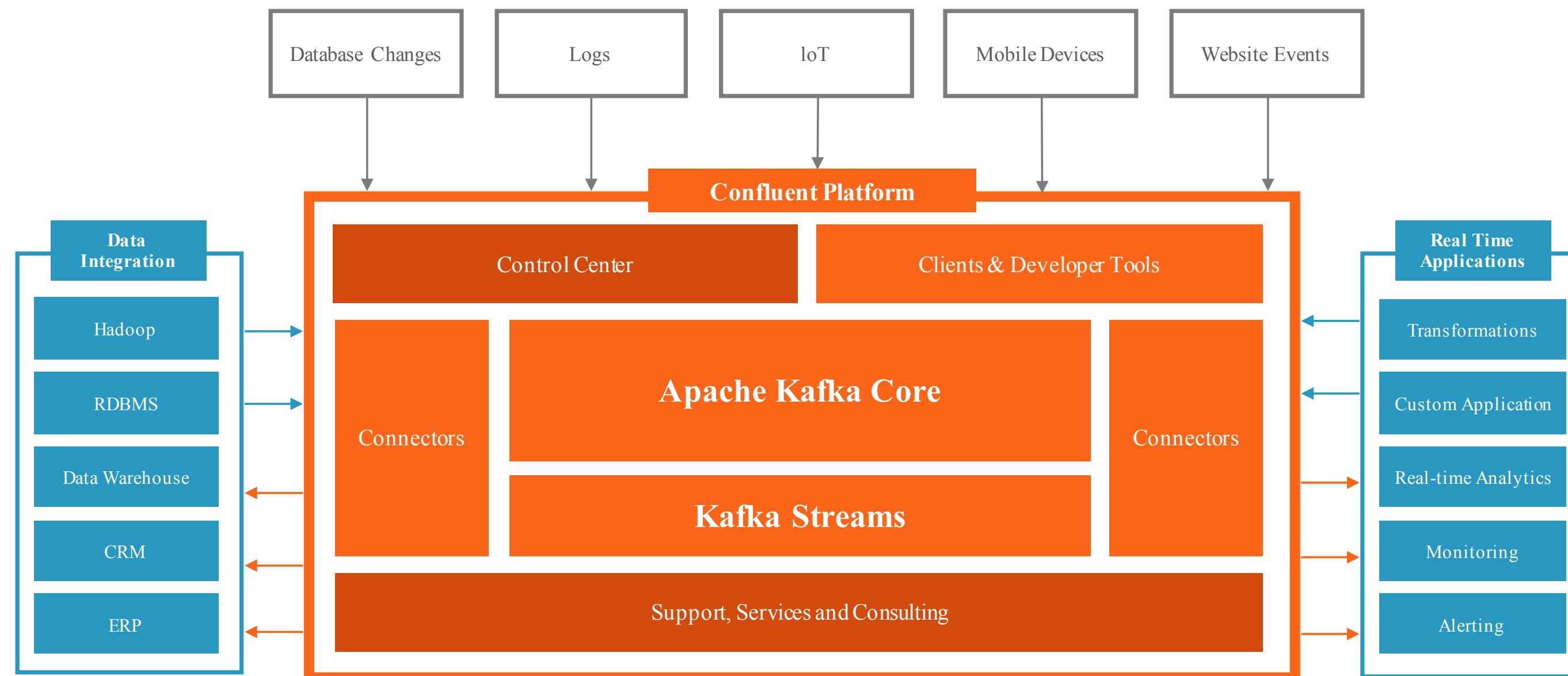


What is the Confluent Platform?

Confluent Platform

Confluent Platform Enterprise

External Product





Schema

When a new Kafka topic was added that data would automatically flow into Hadoop and a corresponding Hive table would be created using the event schema. When the schema evolved that metadata was propagated into Hadoop.

- Jay Kreps, Confluent CEO
- Describing implementation at LinkedIn

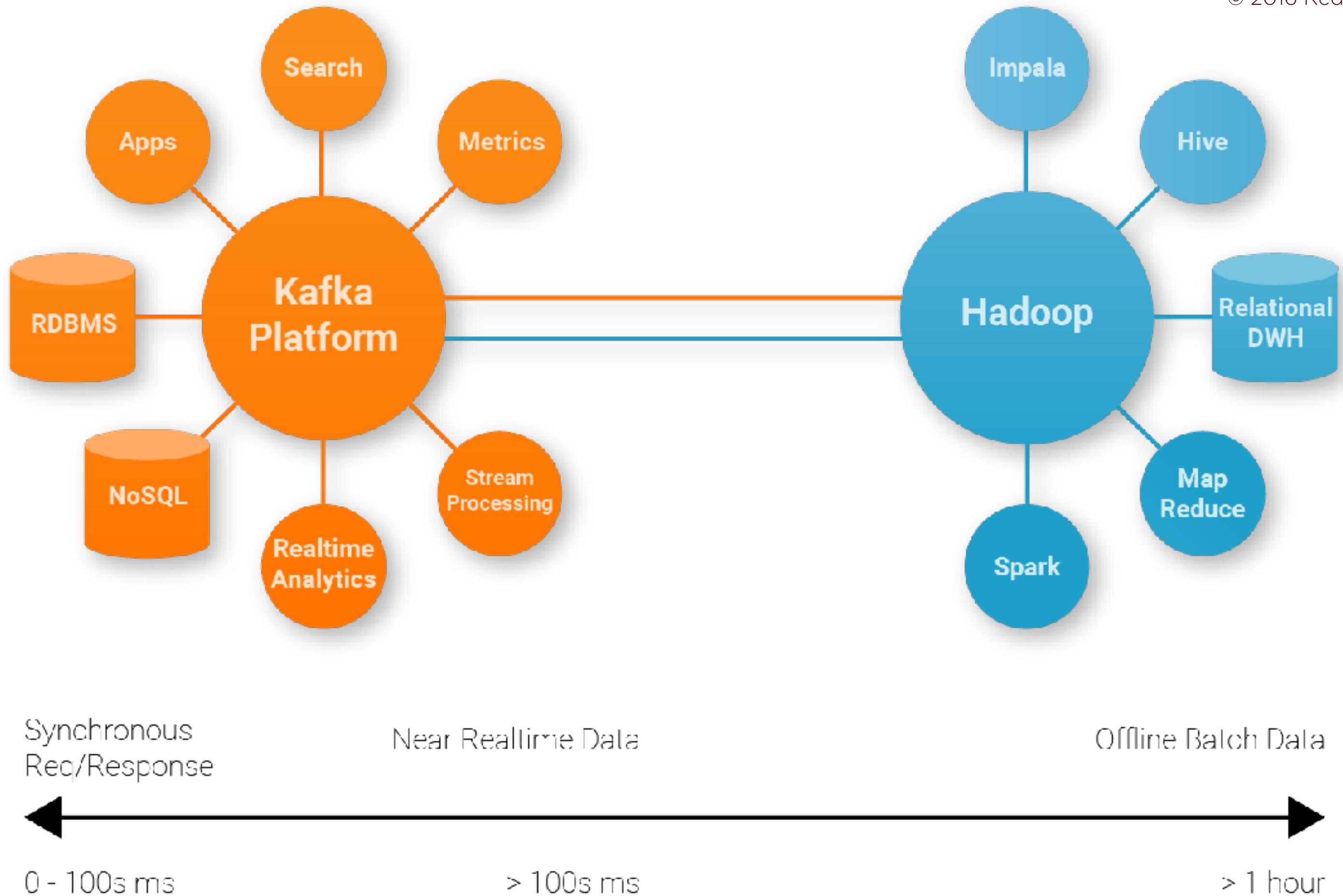
Schema Registry provides a serving layer for your metadata. It provides a RESTful interface for storing and retrieving Avro schemas. It stores a versioned history of all schemas, provides multiple compatibility settings and allows evolution of schemas according to the configured compatibility setting.

– Confluent Documentation

Schema Registry REST API

```
curl -X GET -i http://localhost:8081/subjects
```

```
[  
    "REP-SOE.LOGON-value",  
    "REP-SOE.ADDRESSES-value",  
    "REP-SOE.ORDERS-value",  
    "REP-SOE.CARD_DETAILS-value",  
    "REP-SOE.CUSTOMERS-value",  
    "REP-SOE.INVENTORIES-value",  
    "REP-SOE.ORDER_ITEMS-value",  
    "REP-SCOTT-TX.META-value"  
]
```

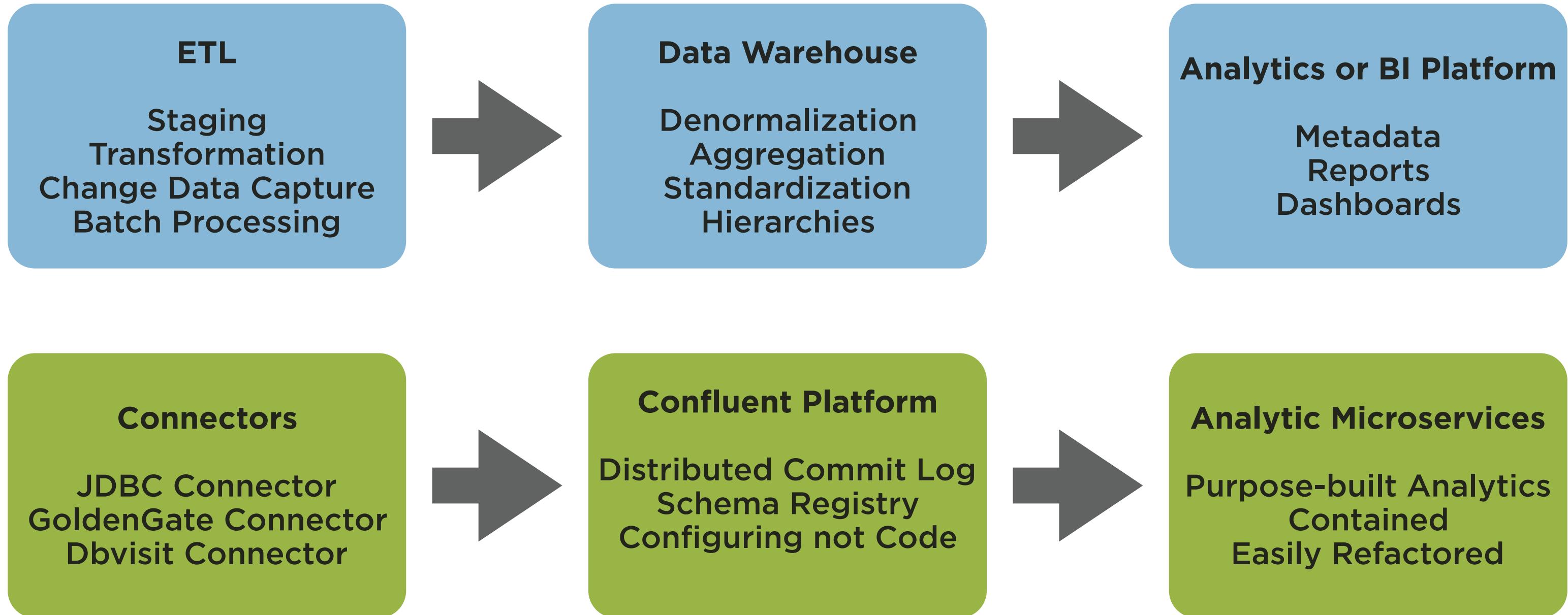


| CONNECTOR | TAGS | DEVELOPER/SUPPORT | DOWNLOAD |
|-----------------------|----------------------------------|-------------------|------------------|
| HDFS (Sink) | HDFS, Hadoop, Hive | Confluent | Confluent |
| JDBC (Source) | JDBC, MySQL | Confluent | Confluent |
| Elastic Search (Sink) | search, Elastic, log, analytics | Confluent | Confluent |
| DataStax (Sink) | Cassandra, DataStax | Data Mountaineer | Data Mountaineer |
| Attunity (Source) | CDC | Attunity | Attunity |
| Couchbase (Source) | Couchbase, NoSQL | Couchbase | Couchbase |
| GoldenGate (Source) | CDC, Oracle | Oracle | Community |
| JustOne (Sink) | Postgress | JustOne | JustOne |
| Striim (Source) | CDC, MS SQLServer, Oracle, MySQL | Striim | Striim |
| Syncsort DMX (Source) | DB2, IMS, VSAM, CICS | Syncsort | Syncsort |
| Syncsort DMX (Sink) | DB2, IMS, VSAM, CICS | Syncsort | Syncsort |
| Vertica (Source) | Vertica | HP Enterprise | HP Enterprise |
| Vertica (Sink) | Vertica | HP Enterprise | HP Enterprise |

Connecting the Enterprise

A detailed view of the Star Trek USS Enterprise NCC-1701 from the side, set against a dark, star-filled background. The ship's hull is a light grey with dark blue highlights, and the name 'NCC-1701' is visible on the side. The saucer section is prominent on the right, and the bridge and upper deck area are visible on the left. The ship is angled slightly, showing its length and the complex engineering of its exterior.

Side by Side (To Start)



JDBC Connector (Kafka Connect)

Kafka Connect Property File

```
name=SugarCRM
connector.class=io.confluent.connect.jdbc.JdbcSourceConnector
connection.url=jdbc:oracle:thin:sugarcrm/welcome1@localhost:1521:orcl
mode=timestamp+incrementing
incrementing.column.name=ID
timestamp.column.name=DATE_MODIFIED
topic.prefix=sugarcrm-
validate.non.null=false
```

GoldenGate Connector (Kafka Connect)

REPLICAT Parameter File

```
REPLICAT conf
TARGETDB LIBFILE libggjava.so SET property=dirprm/conf.props
REPORTCOUNT EVERY 1 MINUTES, RATE
GROUPTRANSOPS 1000
MAP ORCL.SUGARCRM.* , TARGET orcl.sugarcrm.*;
```

GoldenGate Connector (Kafka Connect)

Custom Handler Properties File

```
bootstrap.servers=localhost:9092  
  
value.serializer=org.apache.kafka.common.serialization.ByteArraySerializer  
key.serializer=org.apache.kafka.common.serialization.ByteArraySerializer  
schema.registry.url=http://localhost:8081  
  
value.converter=org.apache.kafka.connect.json.JsonConverter  
key.converter=org.apache.kafka.connect.json.JsonConverter  
internal.value.converter=org.apache.kafka.connect.json.JsonConverter  
internal.key.converter=org.apache.kafka.connect.json.JsonConverter
```

Dbvisit Connector (Kafka Connect)

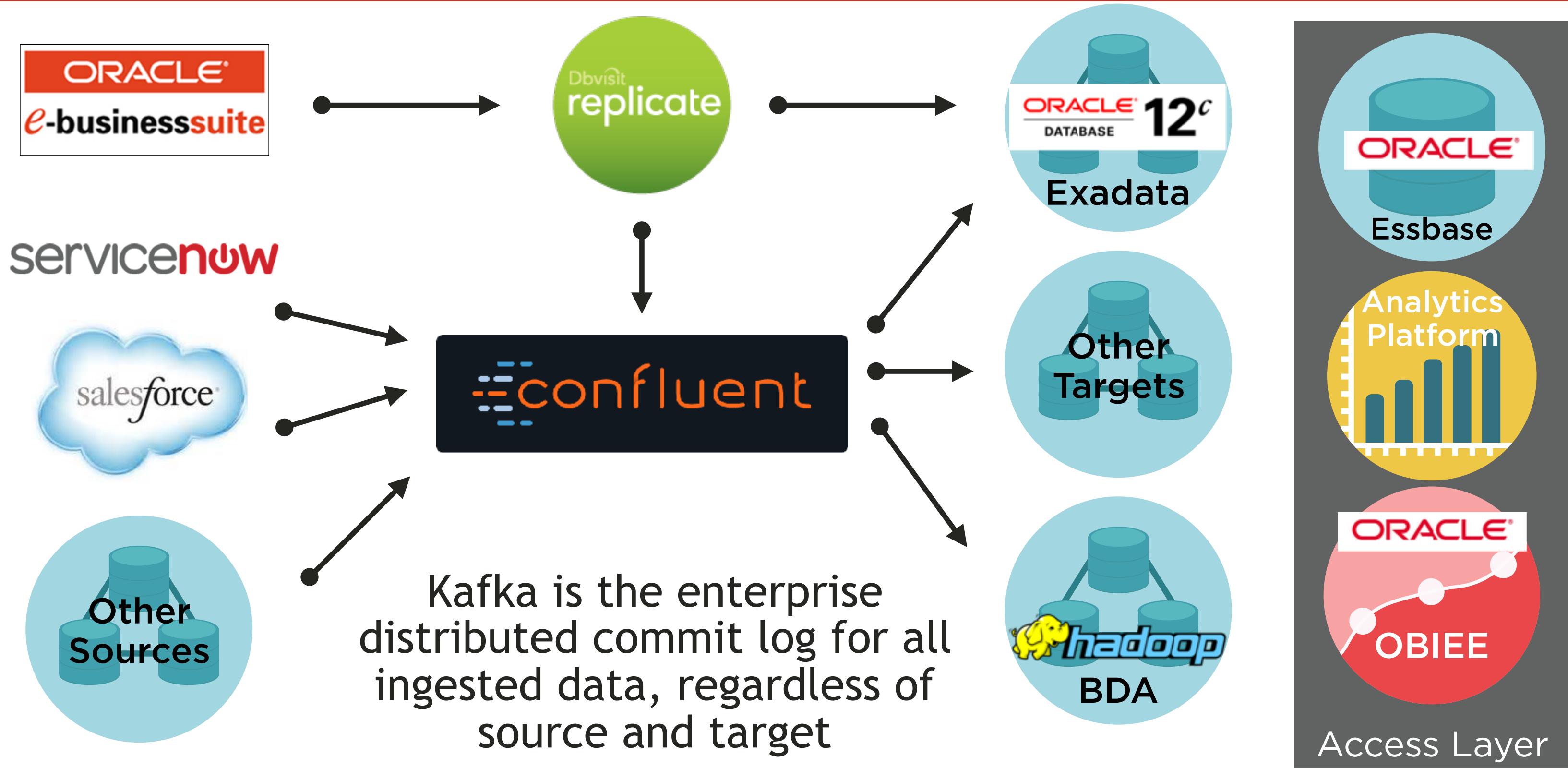
Kafka Connect Property File

```
name=replicate-test-file
connector.class=com.dbvisit.replicate.kafkaconnect.ReplicateSourceConnector
tasks.max=16
project.version=1.0
topic.prefix=REP-
plog.location.uri=file:/home/oracle/REPCON/mine
plog.data.flush.size=1
plog.interval.time.ms=500
plog.scan.interval.count=5
plog.health.check.interval=10
plog.scan.offline.interval=1000
topic.name.transaction.info=SCOTT-TX.META
```

A photograph of four business professionals in a meeting. In the foreground, a man with dark hair and a blue suit looks directly at the camera. Behind him, a man in a grey suit and a woman in a dark blazer are looking down at a document. To the right, a woman with grey hair and a white blouse is smiling and holding a pen over a document. They are in an office environment with large windows in the background.

Customer Case Study

The Kafka-Driven Data Lake



YOU CAN
CHOOSE
TO SEE
DATA
DIFFERENTLY

a RED
PILL
ANALYTICS