



@rmoff

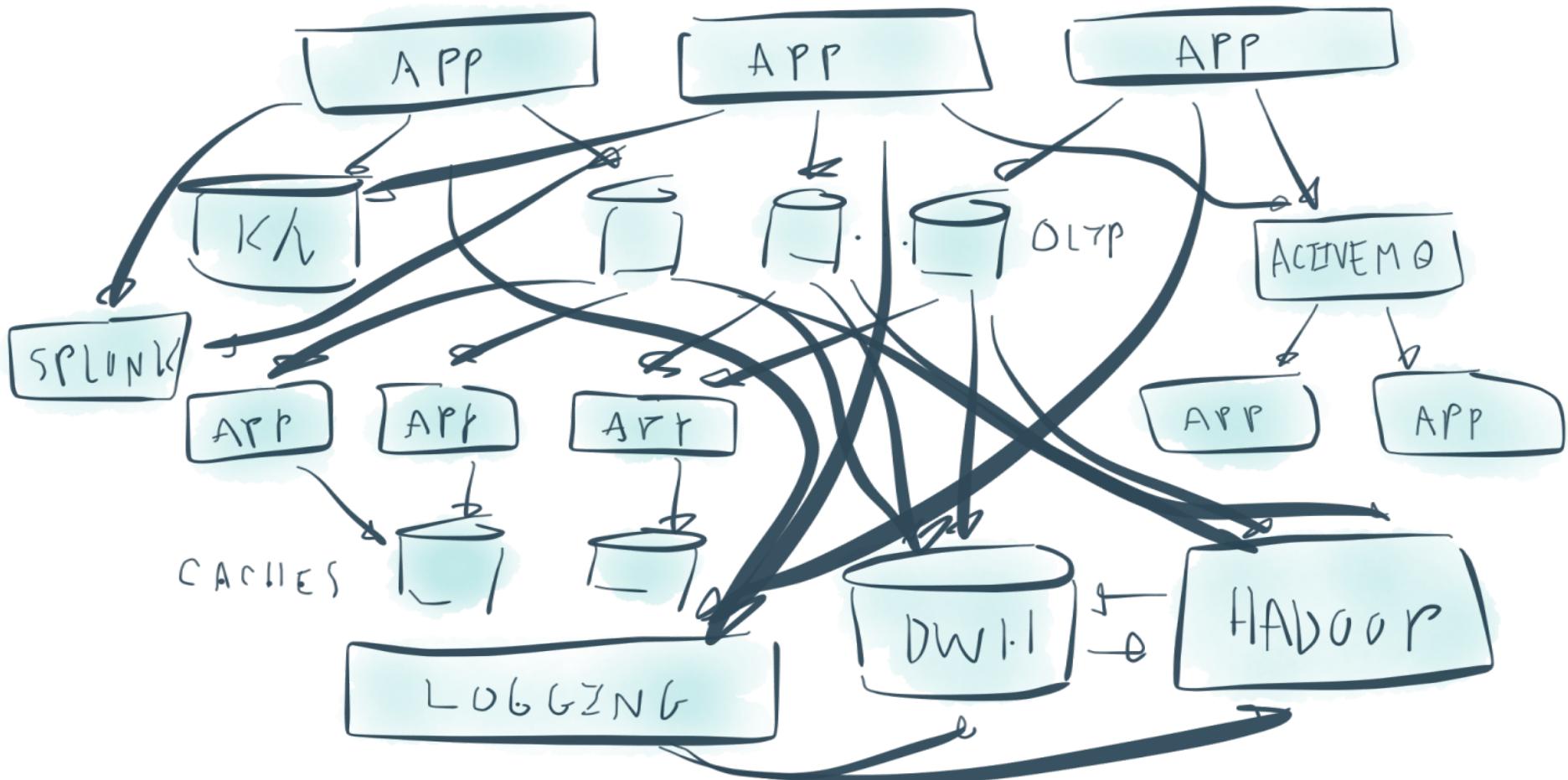
robin@confluent.io

Real-time Data Integration at Scale with Kafka Connect

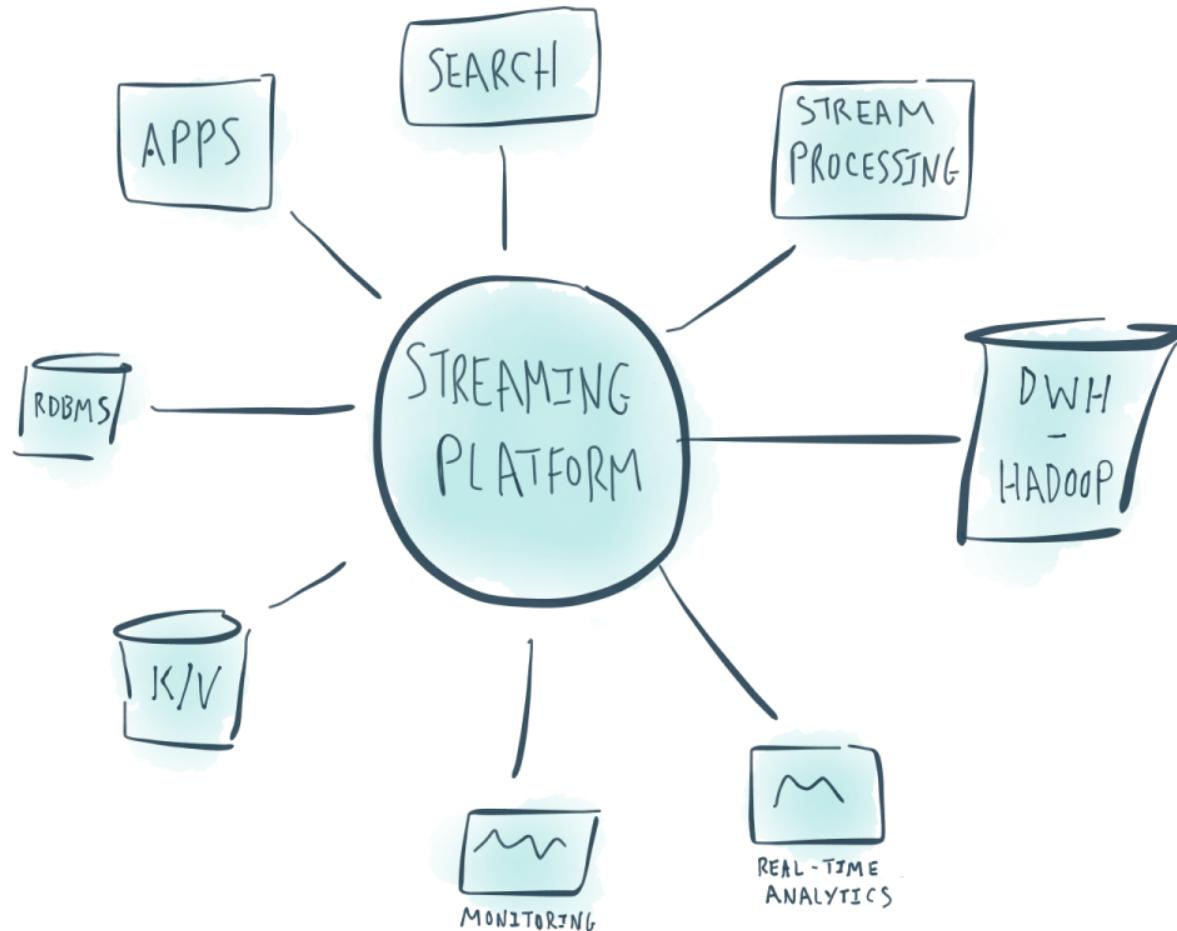
Robin Moffatt

Partner Technology Evangelist, EMEA @ Confluent

A GIANT MESS

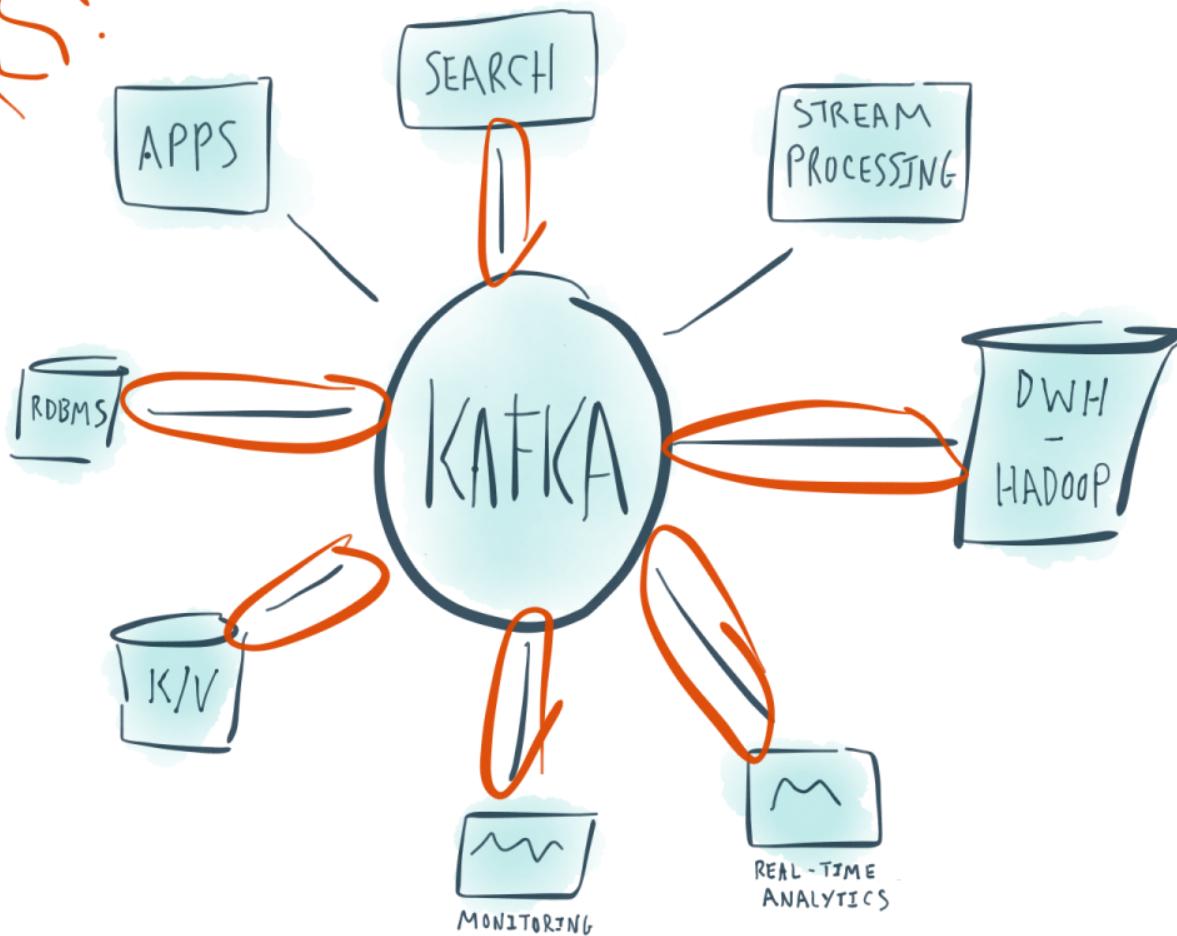


STREAMING PLATFORM

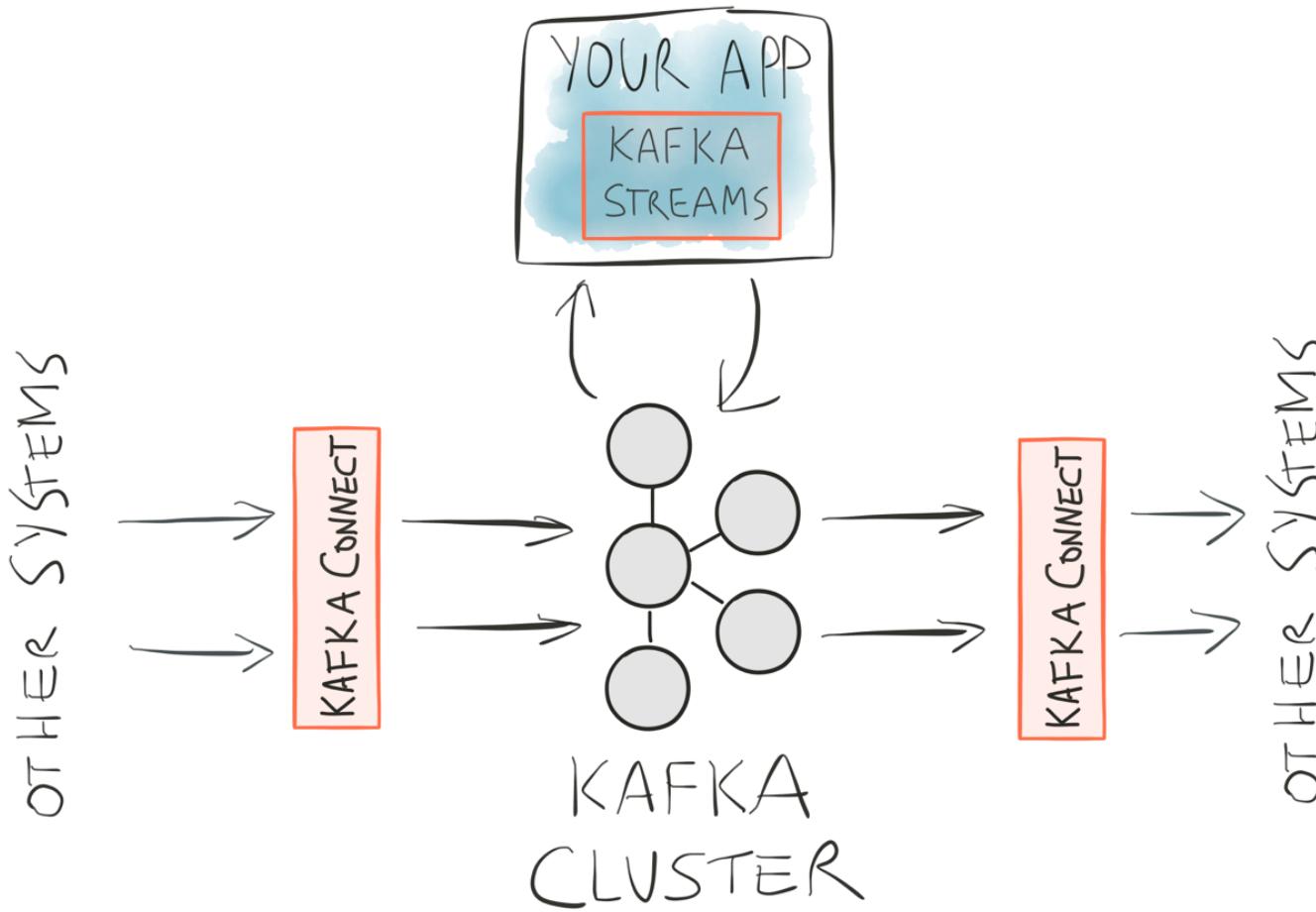


STREAMING PLATFORM

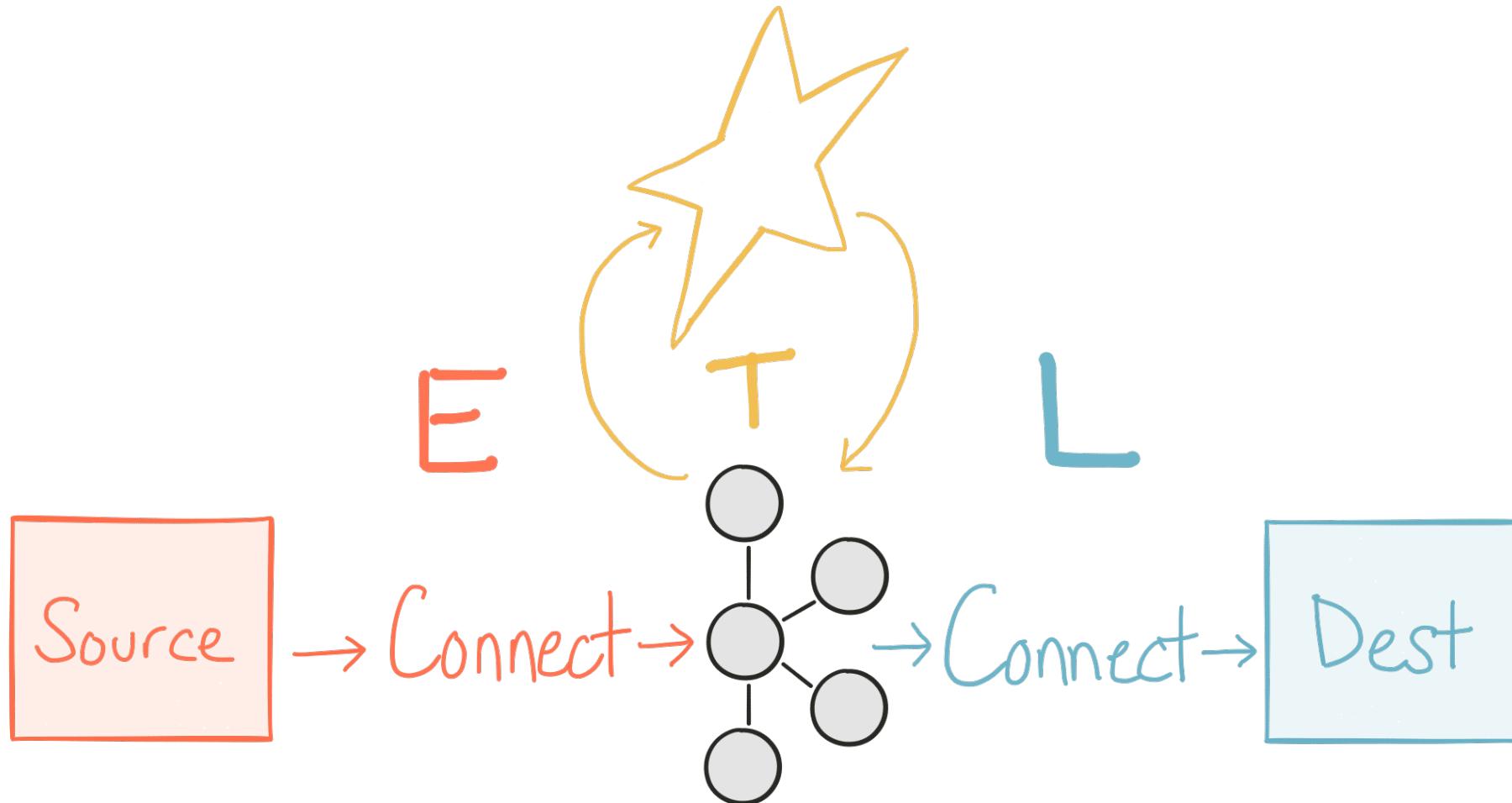
CONNECTORS:



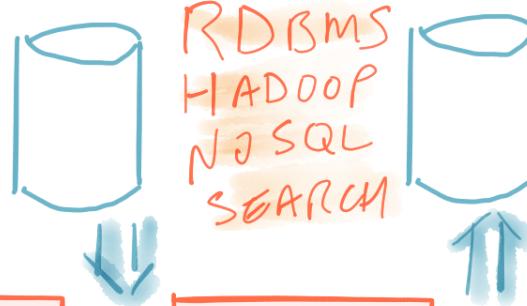
Kafka Connect in the Apache Kafka ecosystem



Kafka Connect : Separation of Concerns

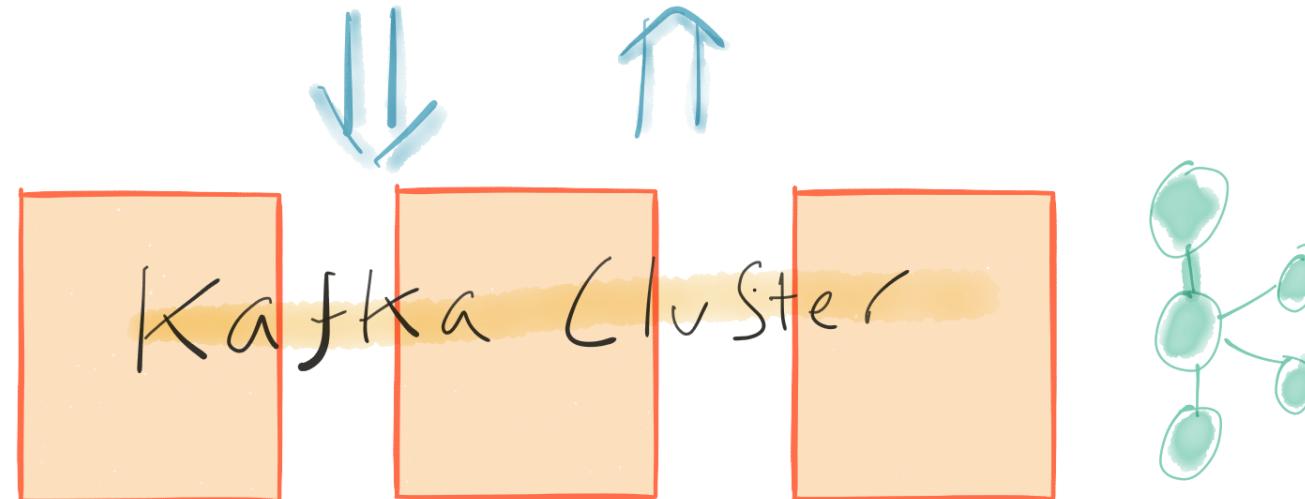


STREAM
FROM



RDBMS
HADOOP
NoSQL
SEARCH

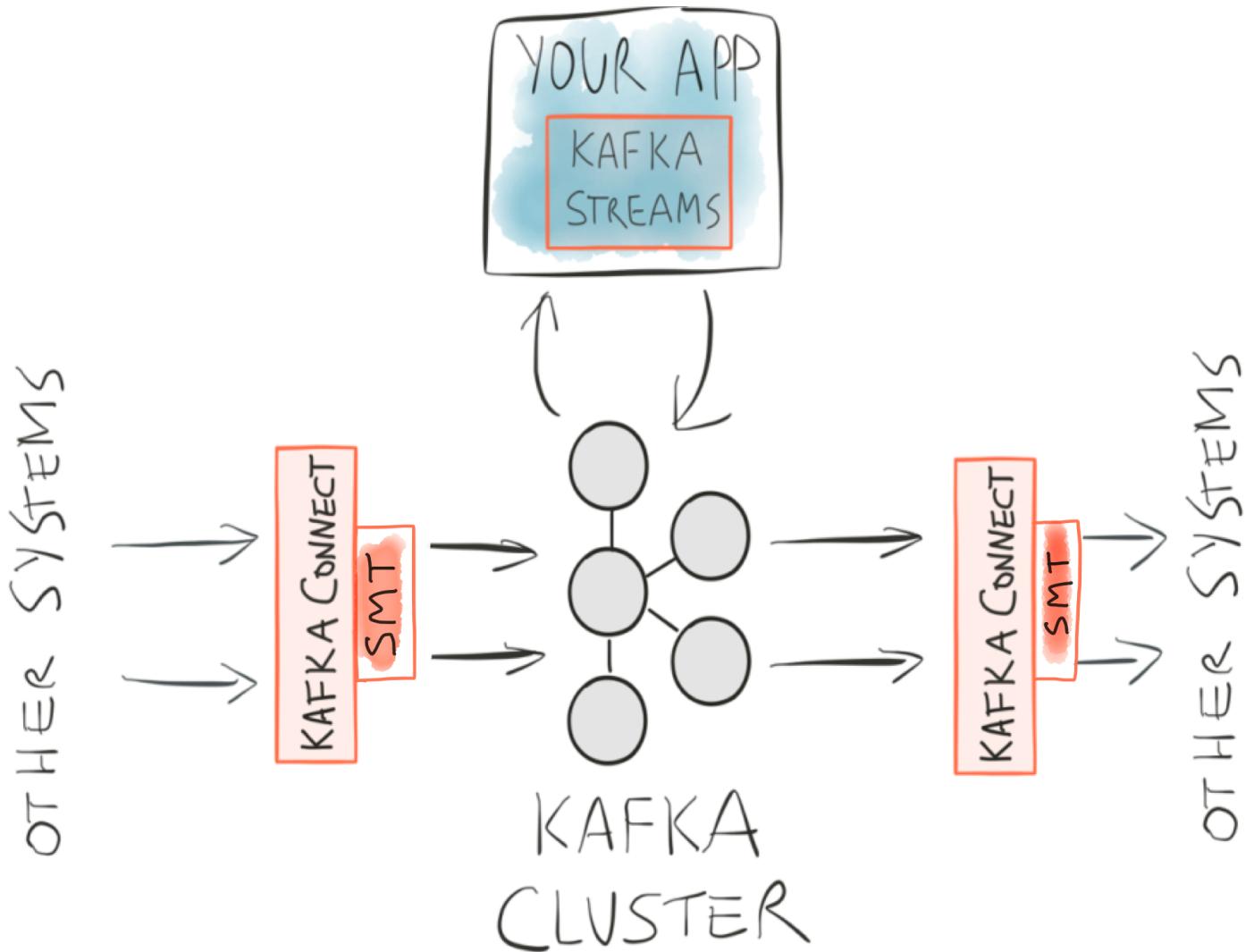
STREAM
TO



Build Data Pipelines

Single Message Transform (SMT) -- Extract, TRANSFORM, Load...

- Modify events *before storing* in Kafka:
 - Mask/drop sensitive information
 - Set partitioning key
 - Store lineage
- Modify events *going out* of Kafka:
 - Route high priority events to faster data stores
 - Direct events to different ElasticSearch indexes
 - Cast data types to match destination



Kafka Connect API Library of Connectors

Databases



Datastore/File Store



Analytics

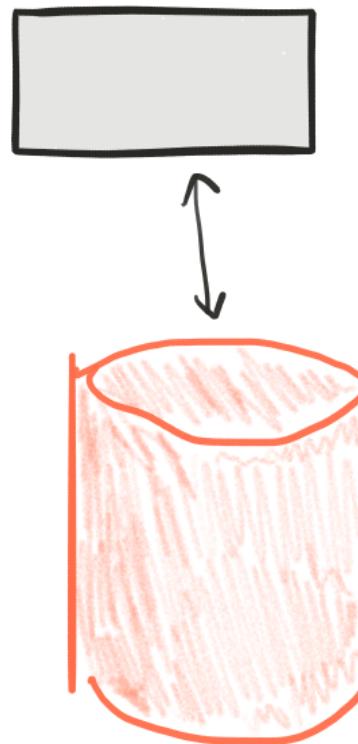


Applications / Other



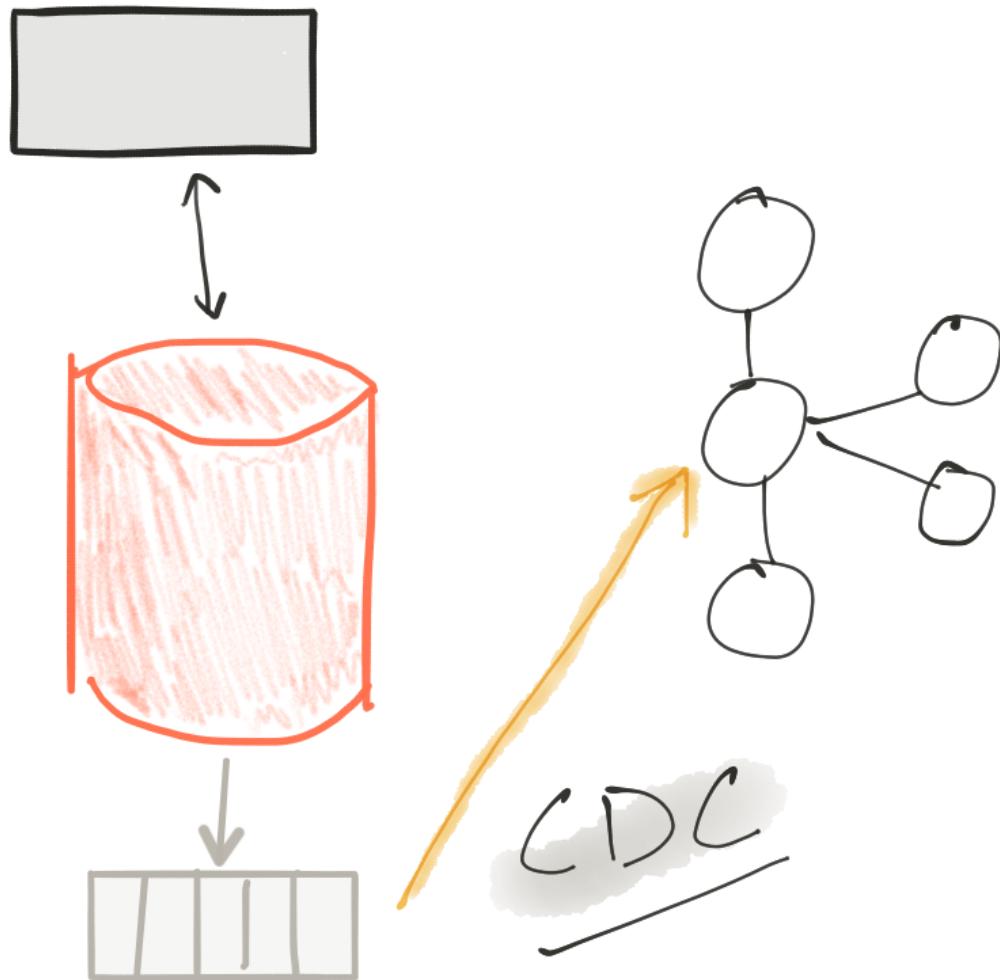
Streaming Application Data to Kafka

- Applications are rich source of events
- Modifying applications is not always possible or desirable
 - And what if the data gets changed within the database or by other apps?
- JDBC is one option for extracting data

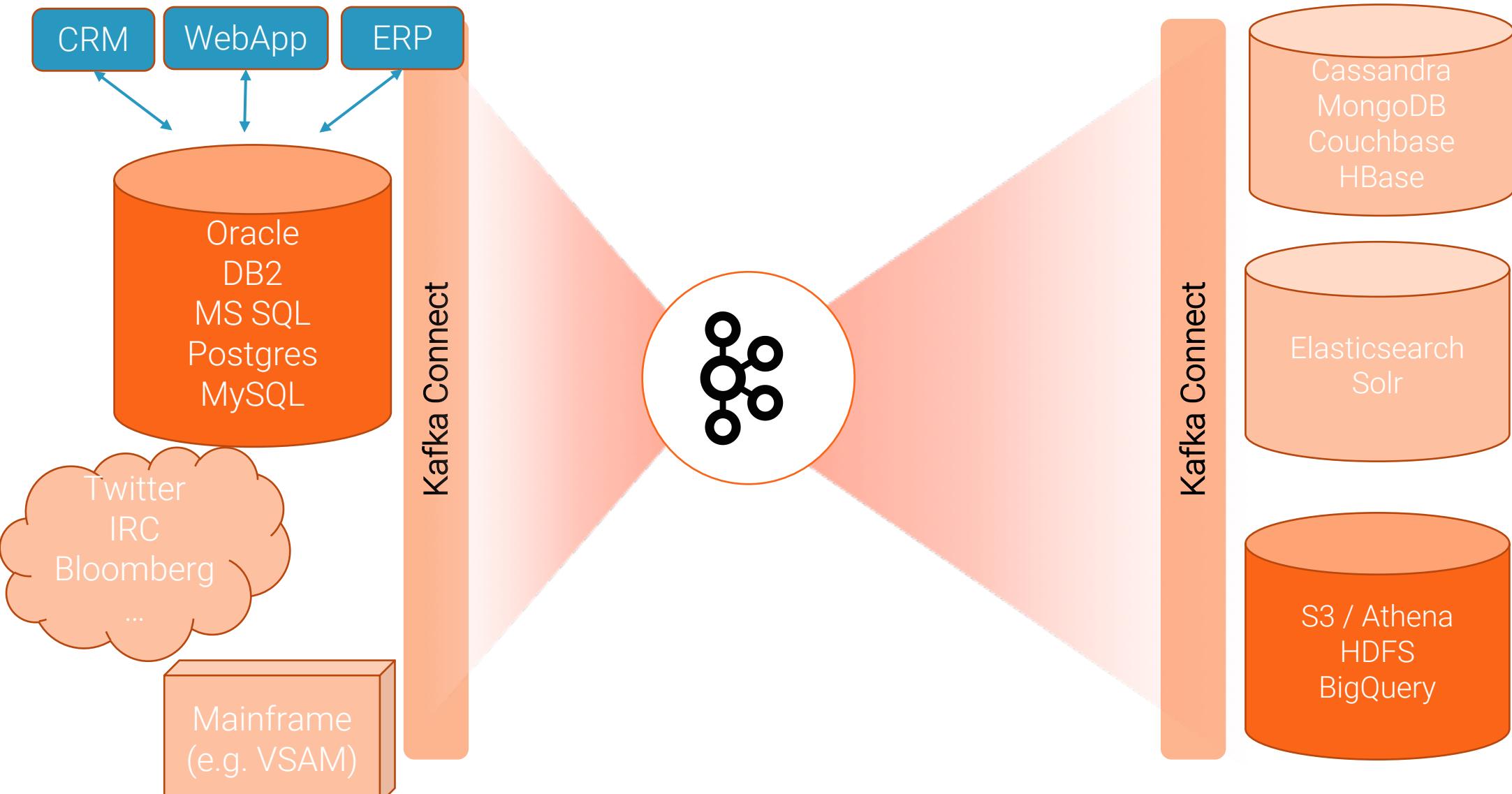


Liberate Application Data into Kafka with CDC

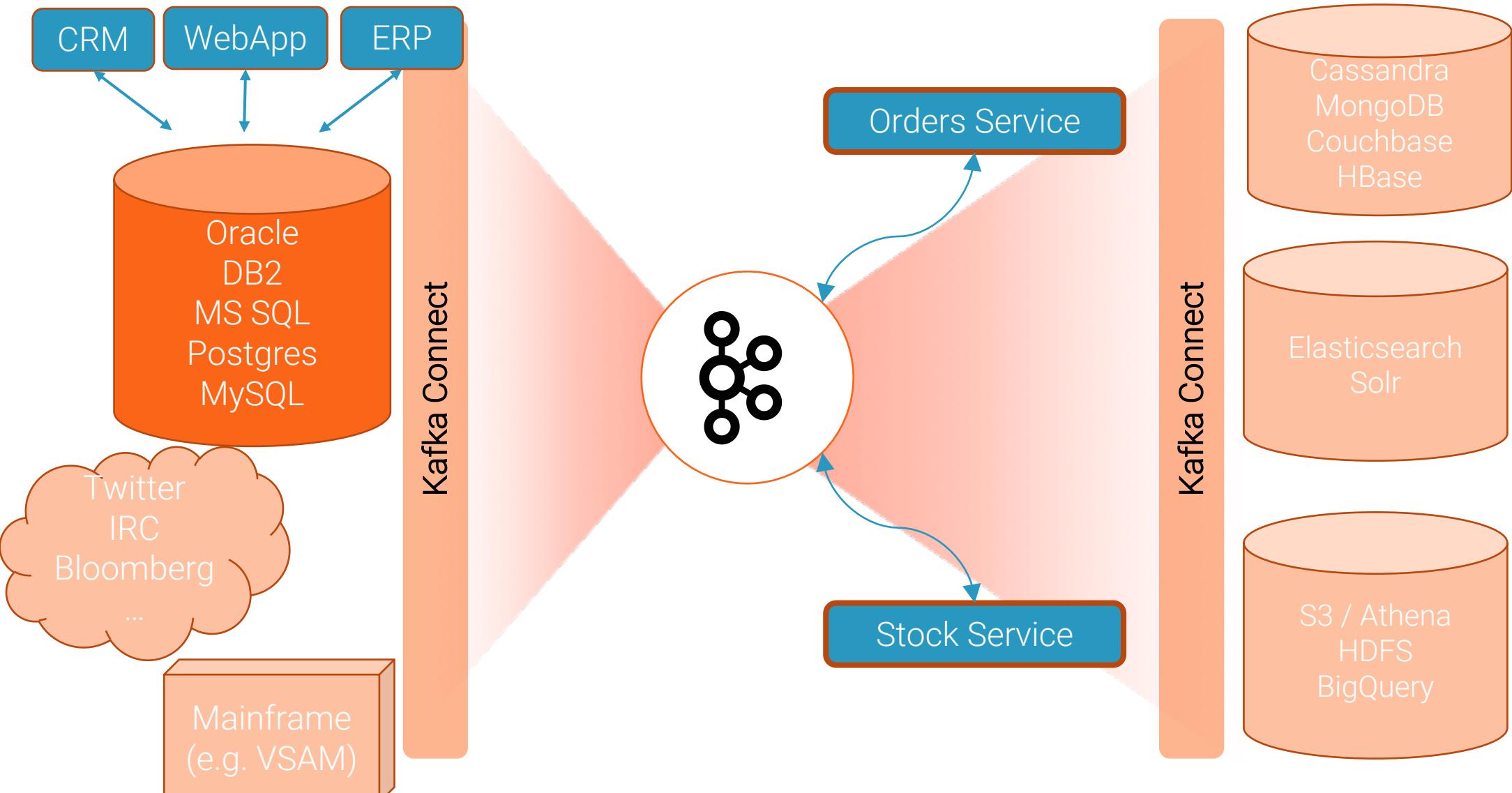
- Relational databases use transaction logs to ensure Durability of data
- Change-Data-Capture (CDC) mines the log to get raw events from the database
- CDC tools that integrate with Kafka Connect include:
 - Debezium
 - DBVisit
 - GoldenGate
 - Attunity
 - + more



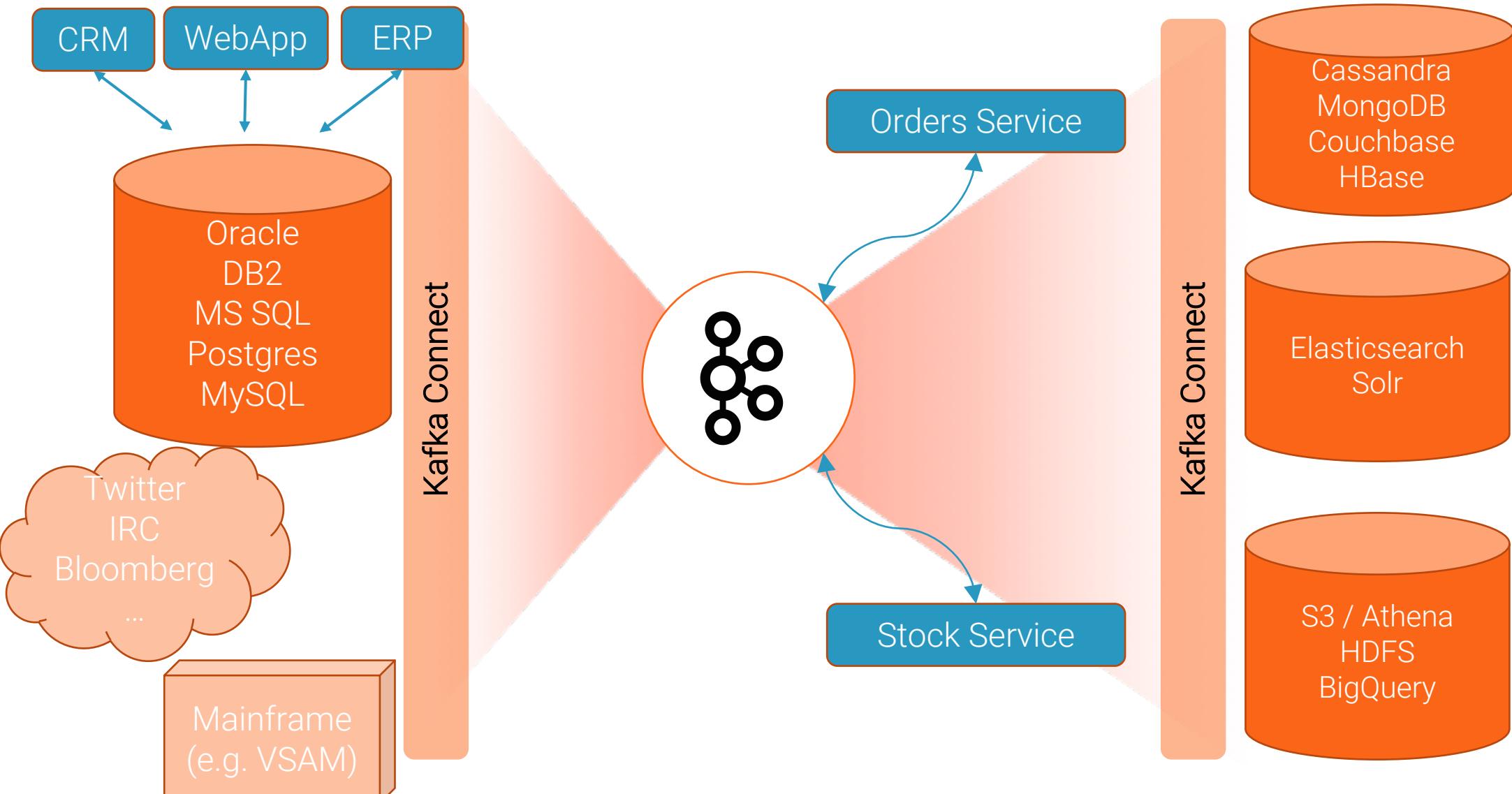
Common Patterns – Data Integration into Data Lake for batch analytics



Common Patterns – Event-Driven microservices



Common Patterns – Event-Driven microservices & audit/search/storage

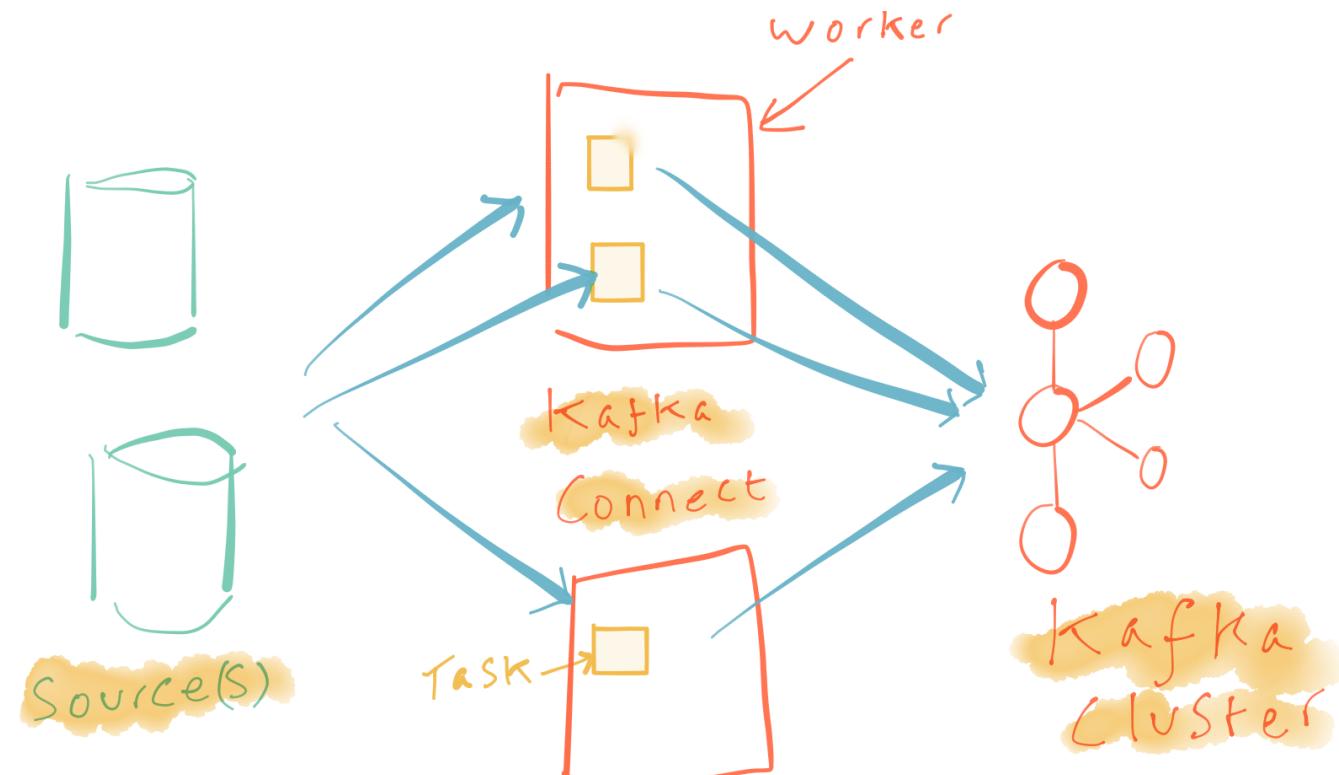


The Numerous Benefits of Kafka Connect

- Restart capabilities (offset management)
- Distributed workers
 - Parallelism (for throughput)
 - Load balancing
 - Fault tolerance
- Schema preservation
- Data serialisation
- Centralised management and configuration

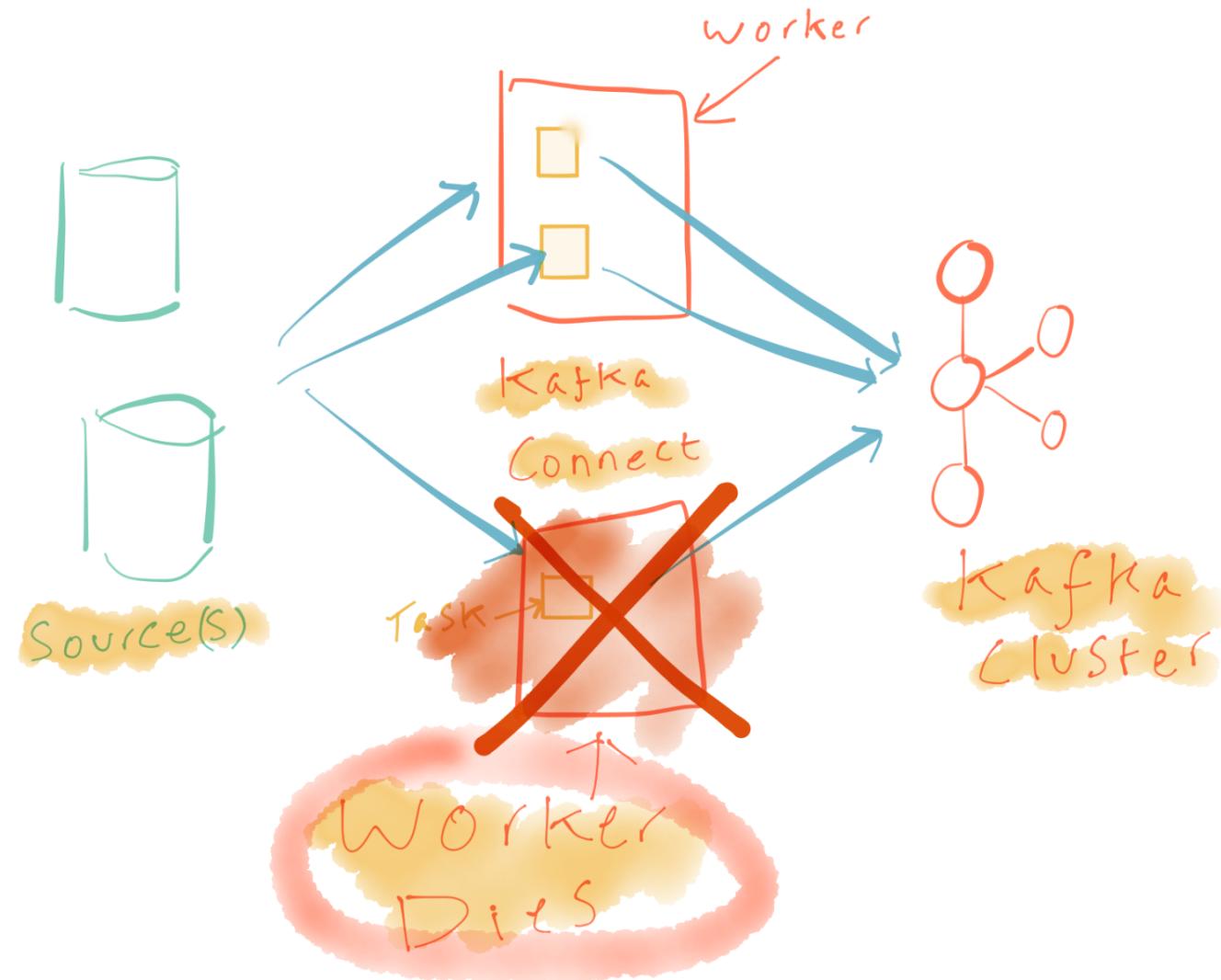
Kafka Connect – under the covers

- Each Kafka Connect node is a **worker**
- Each worker executes one or more **tasks**
- Tasks do the actual work of pulling data from sources / landing it to sinks
- Kafka Connect manages the distribution and execution of tasks
- Parallelism, fault-tolerance, load balancing all handled automatically



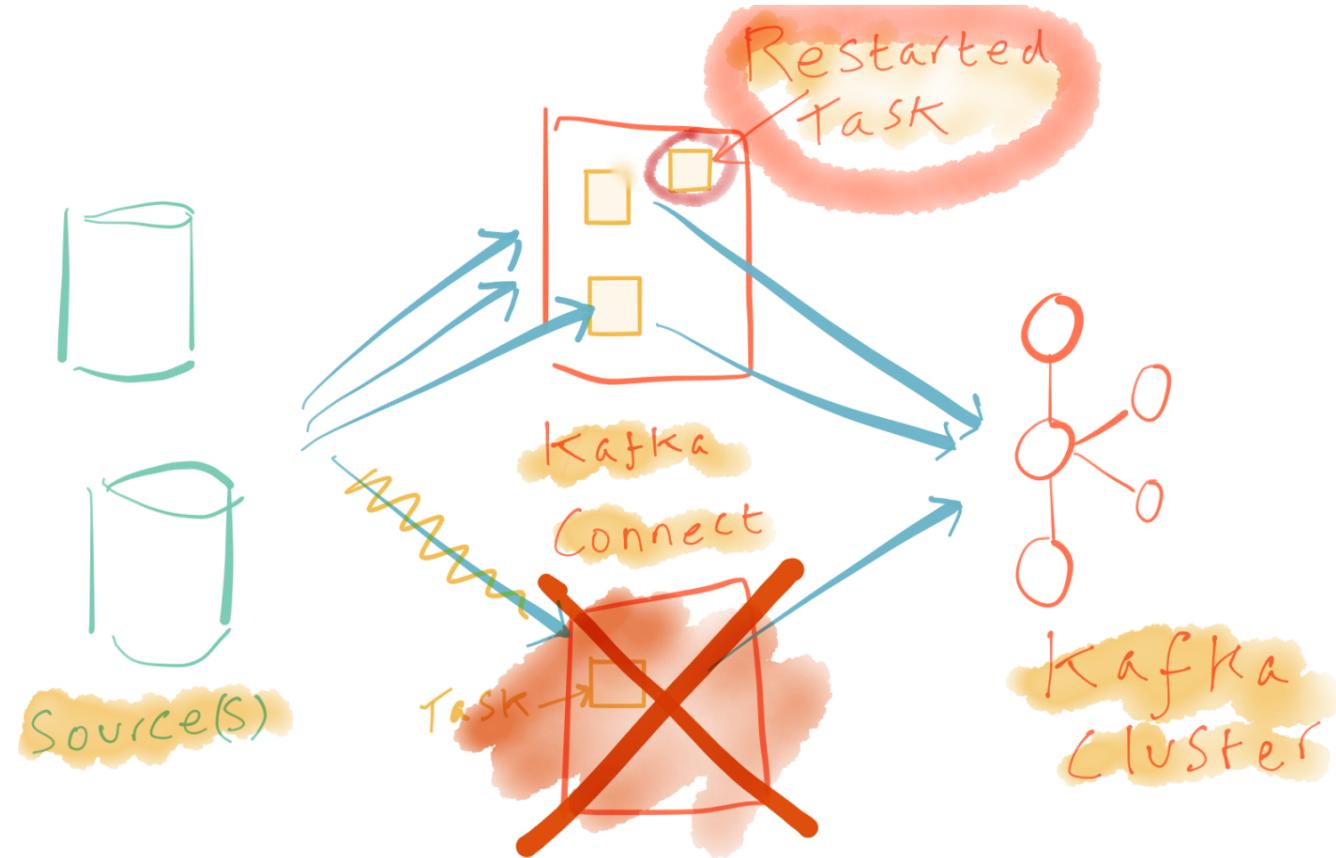
Kafka Connect – under the covers

- Each Kafka Connect node is a **worker**
- Each worker executes one or more **tasks**
- Tasks do the actual work of pulling data from sources / landing it to sinks
- Kafka Connect manages the distribution and execution of tasks
- Parallelism, fault-tolerance, load balancing all handled automatically



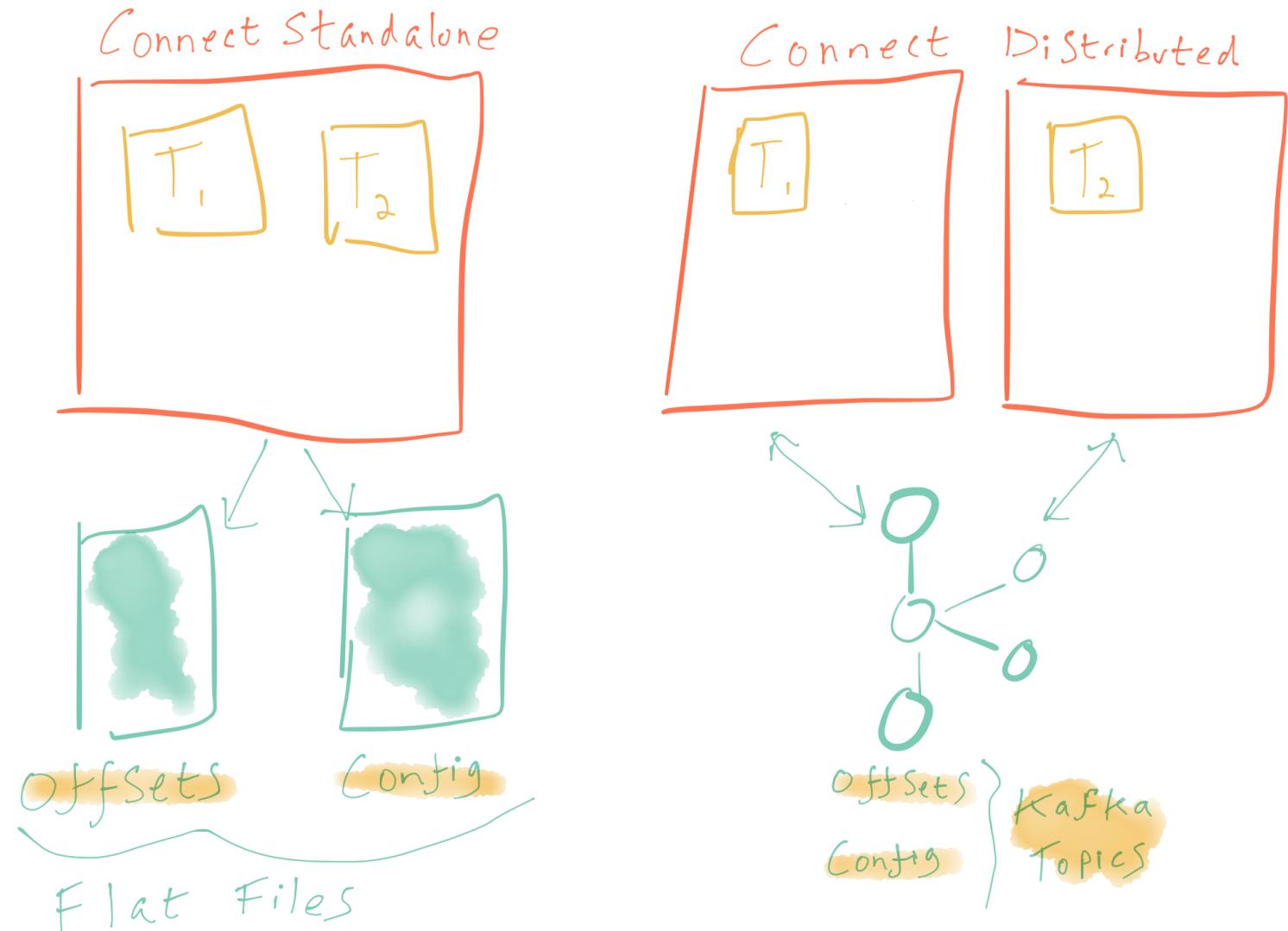
Kafka Connect – under the covers

- Each Kafka Connect node is a **worker**
- Each worker executes one or more **tasks**
- Tasks do the actual work of pulling data from sources / landing it to sinks
- Kafka Connect manages the distribution and execution of tasks
- Parallelism, fault-tolerance, load balancing all handled automatically



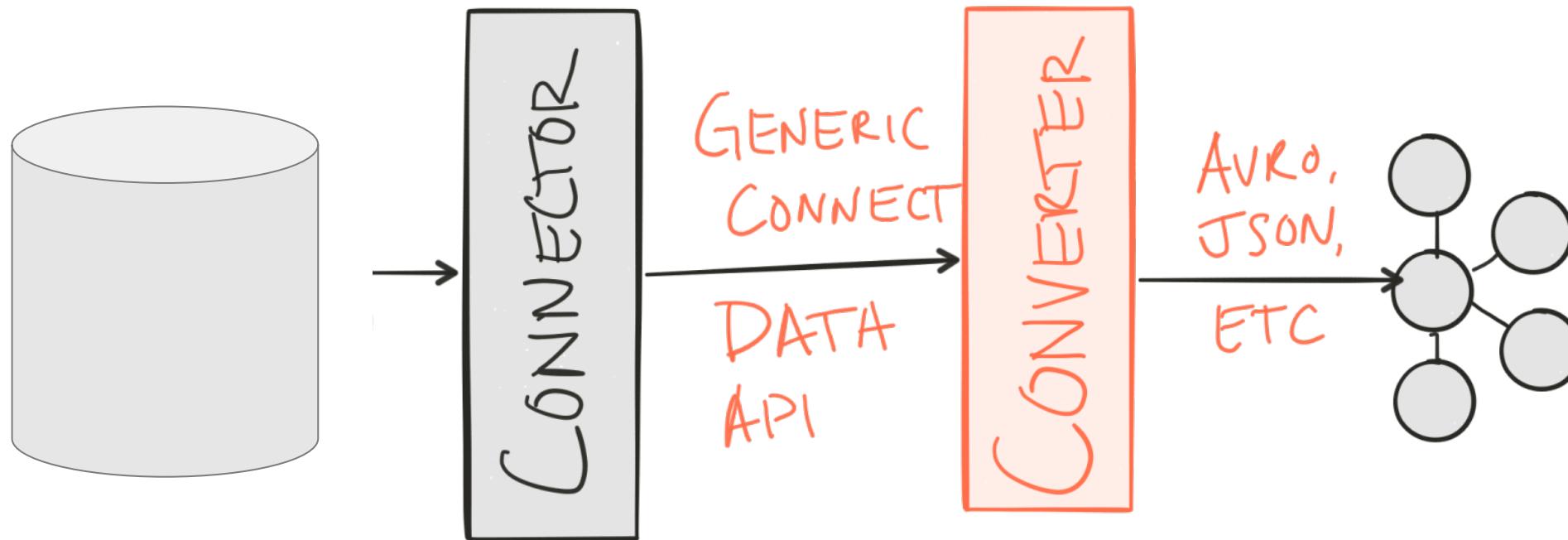
Kafka Connect – Standalone vs Distributed

- Kafka Connect has two modes **standalone or distributed**
- Distributed - Scaleout & fault tolerance easy – just add more workers
 - Can run on one node!
- Standalone - Useful for where data source is machine-specific (e.g. single-node log files)



Kafka Connect - Converters

- Data from source system is in its own format (e.g. RecordSet from JDBC)
- Kafka Connect's Converters provide reusable functionality to serialise data into JSON or Avro
 - The Confluent Schema Registry is used to stores schemas of ingested data



<http://docs.confluent.io/current/connect/concepts.html#converters>

Configuring Kafka Connect - REST API

- Configure & control Kafka Connect through REST API
 - Validate connector configuration
 - Create connectors
 - List available plugins
 - Query connector & task state
 - Pause, resume, restart connectors + tasks
- Configuration is persisted through a Kafka topic
- Reference :
<http://docs.confluent.io/current/connect/restapi.html>

↑ POST http://localhost:8083/connectors/

Description	Headers	URL Params	Body	Auth	Options
			Text	JSON	Form URL-Encoded
			Multipart		File

```
1 {
2   "config": {
3     "connector.class": "JdbcSourceConnector",
4     "connection.url": "jdbc:oracle:thin:soe/soe@192.168.56.101:1521/orcl",
5     "table.whitelist": "ORDERS",
6     "topic.prefix": "soe2-",
7     "mode": "incrementing",
8     "incrementing.column.name": "ORDER_ID",
9     "transforms": "InsertKey,ExtractId",
10    "transforms.InsertKey.type": "org.apache.kafka.connect.transforms.ValueToKey",
11    "transforms.InsertKey.fields": "CUSTOMER_ID",
12    "transforms.ExtractId.type": "org.apache.kafka.connect.transforms.ExtractField$Key",
13    "transforms.ExtractId.field": "CUSTOMER_ID"
14  },
15  "name": "jdbc-swingbench-source-orders"
16 }
```

</>

```
1 ## Create jdbc-swingbench-source-orders
2 curl -X "POST" "http://localhost:8083/connectors/" \
3   -H "Content-Type: application/json" \
4   -d ${'
5   "name": "jdbc-swingbench-source-orders",
6   "config": {
7     "transforms.InsertKey.type": "org.apache.kafka.connect.transforms.ValueToKey",
8     "transforms.ExtractId.field": "CUSTOMER_ID",
9     "topic.prefix": "soe2-",
10    "table.whitelist": "ORDERS",
11    "mode": "incrementing",
12    "connector.class": "JdbcSourceConnector",
13    "transforms.InsertKey.fields": "CUSTOMER_ID",
14    "transforms": "InsertKey,ExtractId",
15    "transforms.ExtractId.type": "org.apache.kafka.connect.transforms.ExtractField$Key",
16    "incrementing.column.name": "ORDER_ID",
17    "connection.url": "jdbc:oracle:thin:soe/soe@192.168.56.101:1521/orcl"
18  }
19 }'
```

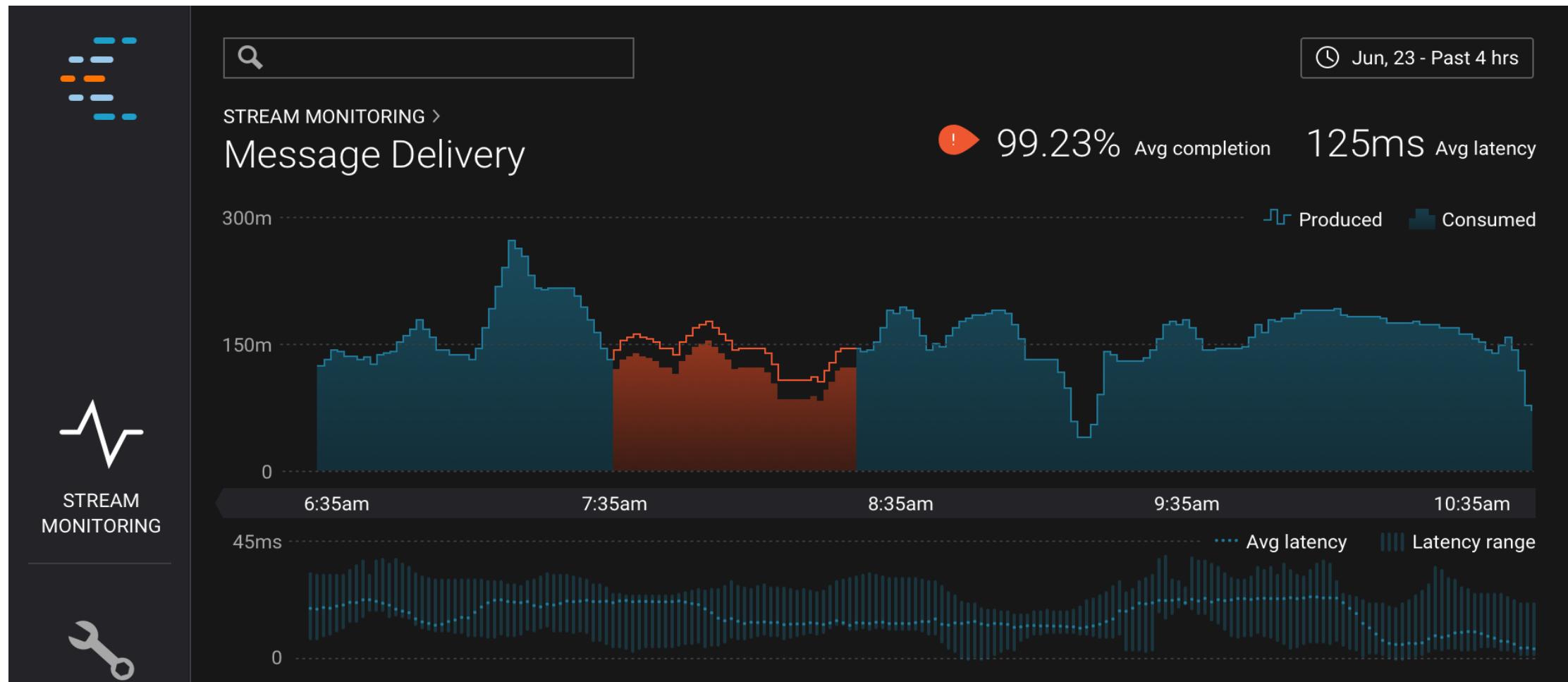
cURL | ↴ | 🔒

Configure Kafka Connect with Confluent Control Center

The screenshot shows the Confluent Control Center interface for managing Kafka Connect sources. The left sidebar includes sections for MONITORING (System health, Data streams), MANAGEMENT (Kafka Connect, Clusters, Topics), and ALERTS (Overview, Integration). The main content area is titled "Edit Source" under "MANAGEMENT > KAFKA CONNECT > SOURCES". The "SOURCES" tab is selected. A configuration card for a JDBC source connector is displayed, showing the following fields:

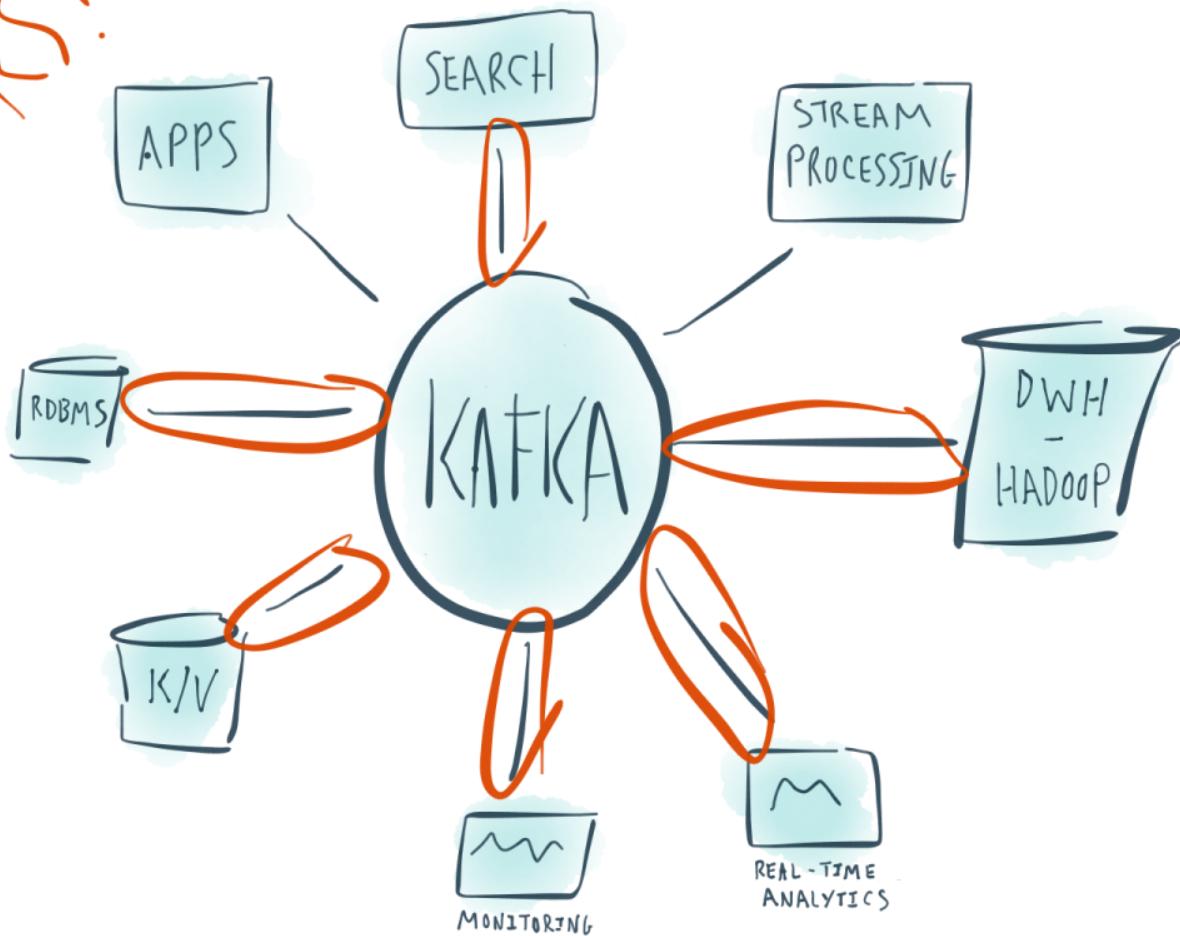
Connector Class*	io.confluent.connect.jdbc.JdbcSourceConnector	Name*	jdbc_source_sqlite_foo
Common		Key converter class ⓘ	io.confluent.connect.avro.AV
Tasks max ⓘ		Value converter class ⓘ	io.confluent.connect.avro.AV
Transforms			
Transforms ⓘ			
Database			
JDBC URL* ⓘ	jdbc:sqlite:/Users/Robin/cp/confluent-3.3.0-	JDBC User ⓘ	
JDBC Pass... ⓘ	Table Whitelist ⓘ	foo x	Map Numeri... ⓘ
Table Blacklist ⓘ		Schema pattern ⓘ	

Monitor Your Data Pipeline from End to End with Confluent Control Center

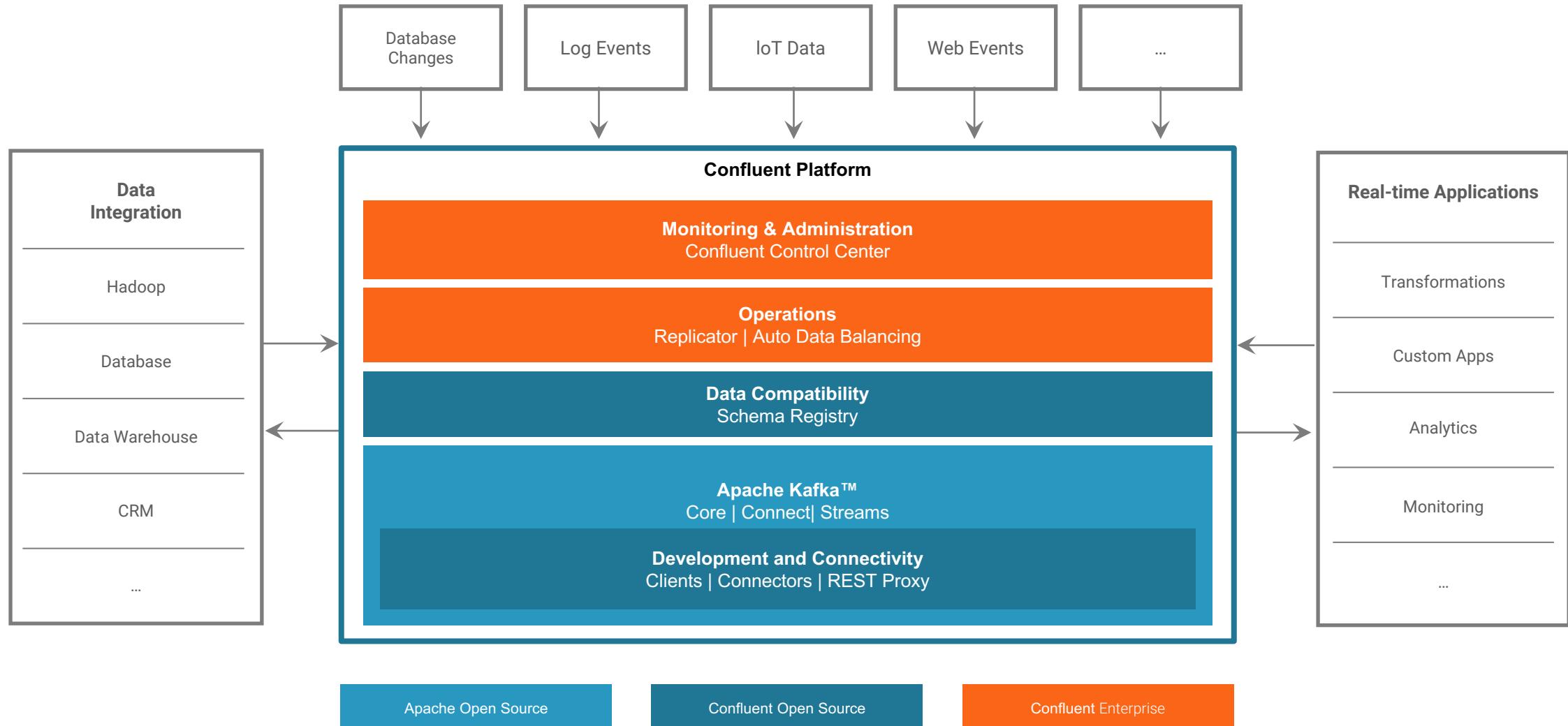


STREAMING PLATFORM

CONNECTORS:

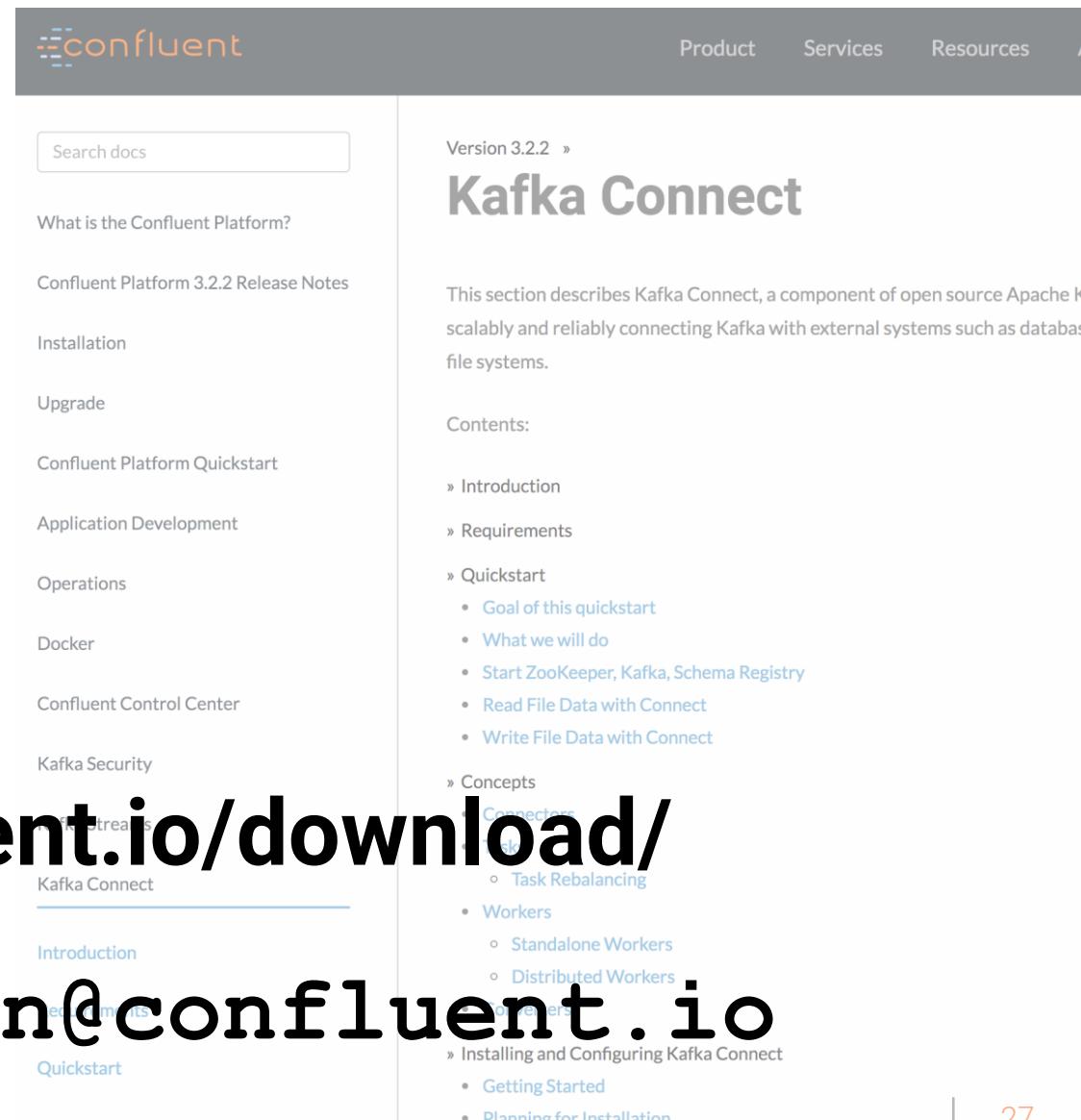


Confluent: a Streaming Platform based on Apache Kafka™



Kafka Connect – Getting Started

- Docs : <http://docs.confluent.io/current/connect/>
- Includes **Quickstart** and full Connect documentation including Architecture + Internals
- Official Confluent Platform Docker images available
 - <http://docs.confluent.io/current/cp-docker-images/docs/quickstart.html#kafka-connect>
- List of connectors
 - <https://www.confluent.io/product/connectors/>
 - Also search on github
<https://github.com/search?q=kafka-connect>



The screenshot shows the Confluent Kafka Connect documentation page. At the top, there's a navigation bar with the Confluent logo, 'Product', 'Services', 'Resources', and other links. Below the header, a breadcrumb trail says 'Version 3.2.2 » Kafka Connect'. The main content area has a sidebar with links like 'Search docs', 'What is the Confluent Platform?', 'Confluent Platform 3.2.2 Release Notes', 'Installation', 'Upgrade', 'Confluent Platform Quickstart', 'Application Development', 'Operations', 'Docker', 'Confluent Control Center', 'Kafka Security', and 'Connectors' (with sub-links for 'Task Rebalancing', 'Workers', 'Standalone Workers', 'Distributed Workers', and 'Workers'). The main content area also lists 'Contents' with sections like 'Introduction', 'Requirements', 'Quickstart' (with sub-links for 'Goal of this quickstart', 'What we will do', 'Start ZooKeeper, Kafka, Schema Registry', 'Read File Data with Connect', and 'Write File Data with Connect'), 'Concepts' (with sub-links for 'Connectors', 'Workers', 'Task Rebalancing', and 'Workers'), and 'Installing and Configuring Kafka Connect' (with sub-links for 'Getting Started' and 'Planning for Installation').

<https://www.confluent.io/download/>

@rmoff



robin@confluent.io