# Self-Service Data Ingestion Using NiFi, StreamSets & Kafka

TECH17
UKOUG Technology Conference & Exhibition

Guido Schmutz – 6.12.2017


@gschmutz guidoschmutz.wordpress.com

# Guido Schmutz

Working at Trivadis for more than 21 years

Oracle ACE Director for Fusion Middleware and SOA

Consultant, Trainer Software Architect for Java, Oracle, SOA and Big Data / Fast Data

Head of Trivadis Architecture Board

Technology Manager @ Trivadis

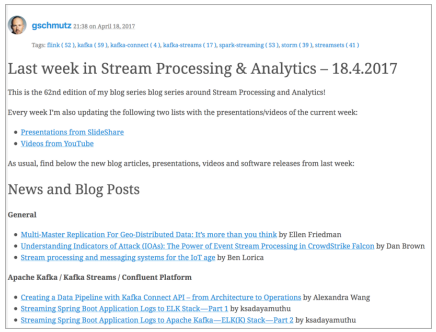
More than 30 years of software development experience


Contact: guido.schmutz@trivadis.com

Blog: http://guidoschmutz.wordpress.com

Slideshare: http://www.slideshare.net/gschmutz

Twitter: gschmutz

gschmutz 21:38 on April 18, 2017

Tags: flink ( 52 ), kafka ( 59 ), kafka-connect ( 4 ), kafka-streams ( 17 ), spark-streaming ( 53 ), storm ( 39 ), streamsets ( 41 )

Last week in Stream Processing & Analytics – 18.4.2017

This is the 62nd edition of my blog series blog series around Stream Processing and Analytics!

Every week I'm also updating the following two lists with the presentations/videos of the current week:

- Presentations from SlideShare
- Videos from YouTube

As usual, find below the new blog articles, presentations, videos and software releases from last week:

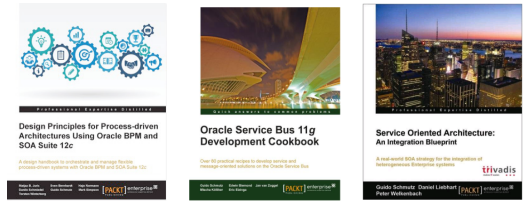
News and Blog Posts

**General**

- Multi-Master Replication For Geo-Distributed Data: It's more than you think by Ellen Friedman
- Understanding Indicators of Attack (IOAs): The Power of Event Stream Processing in CrowdStrike Falcon by Dan Brown
- Stream processing and messaging systems for the IoT age by Ben Lorica

**Apache Kafka / Kafka Streams / Confluent Platform**

- Creating a Data Pipeline with Kafka Connect API – from Architecture to Operations by Alexandra Wang
- Streaming Spring Boot Application Logs to ELK Stack—Part 1 by ksadayamuthu
- Streaming Spring Boot Application Logs to Apache Kafka—ELKK Stack—Part 2 by ksadayamuthu


trivadis
makes IT easier.

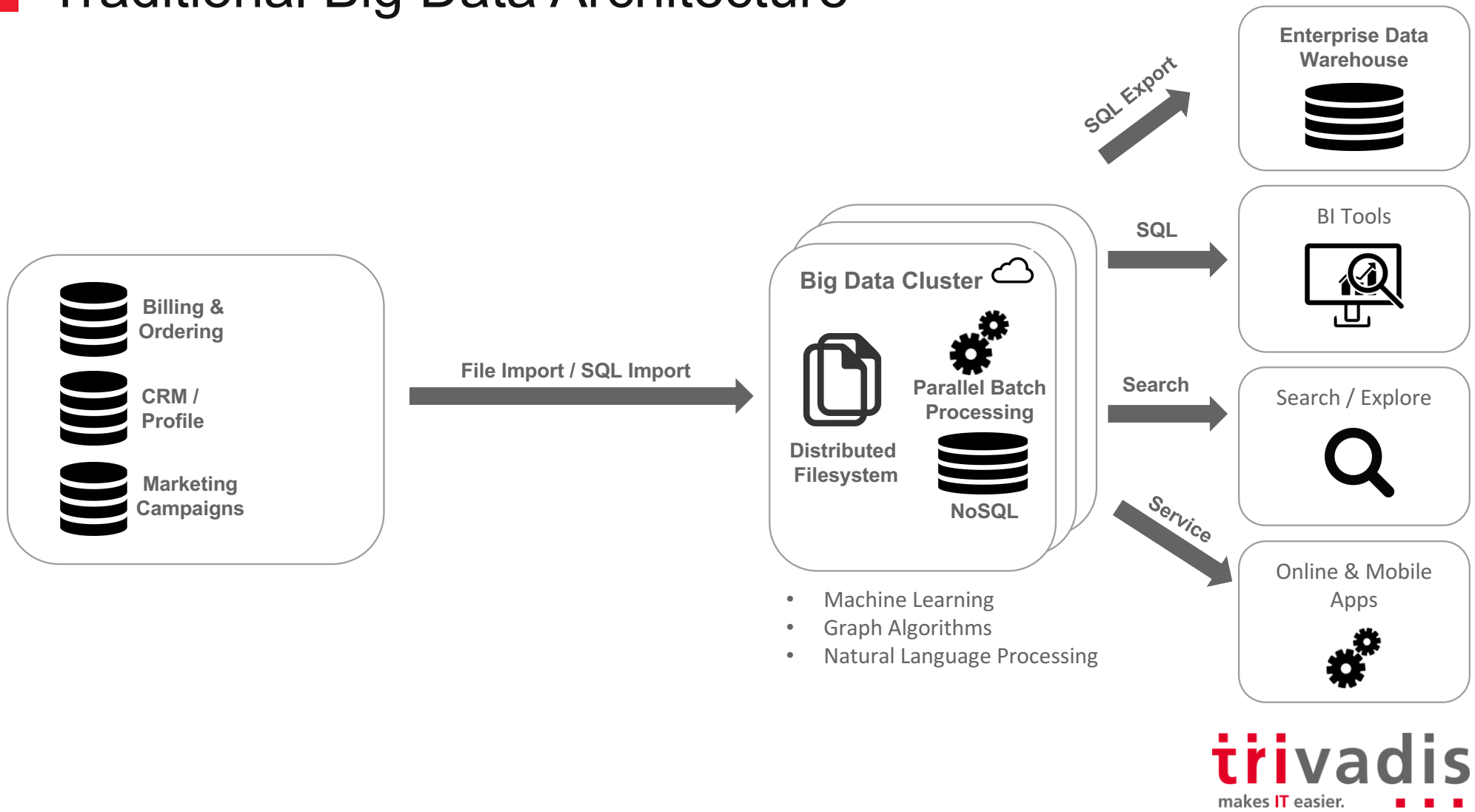
# Agenda

1. Data Flow Processing
2. Apache NiFi
3. StreamSets Data Collector
4. Kafka Connect
5. Summary

# Data Flow Processing

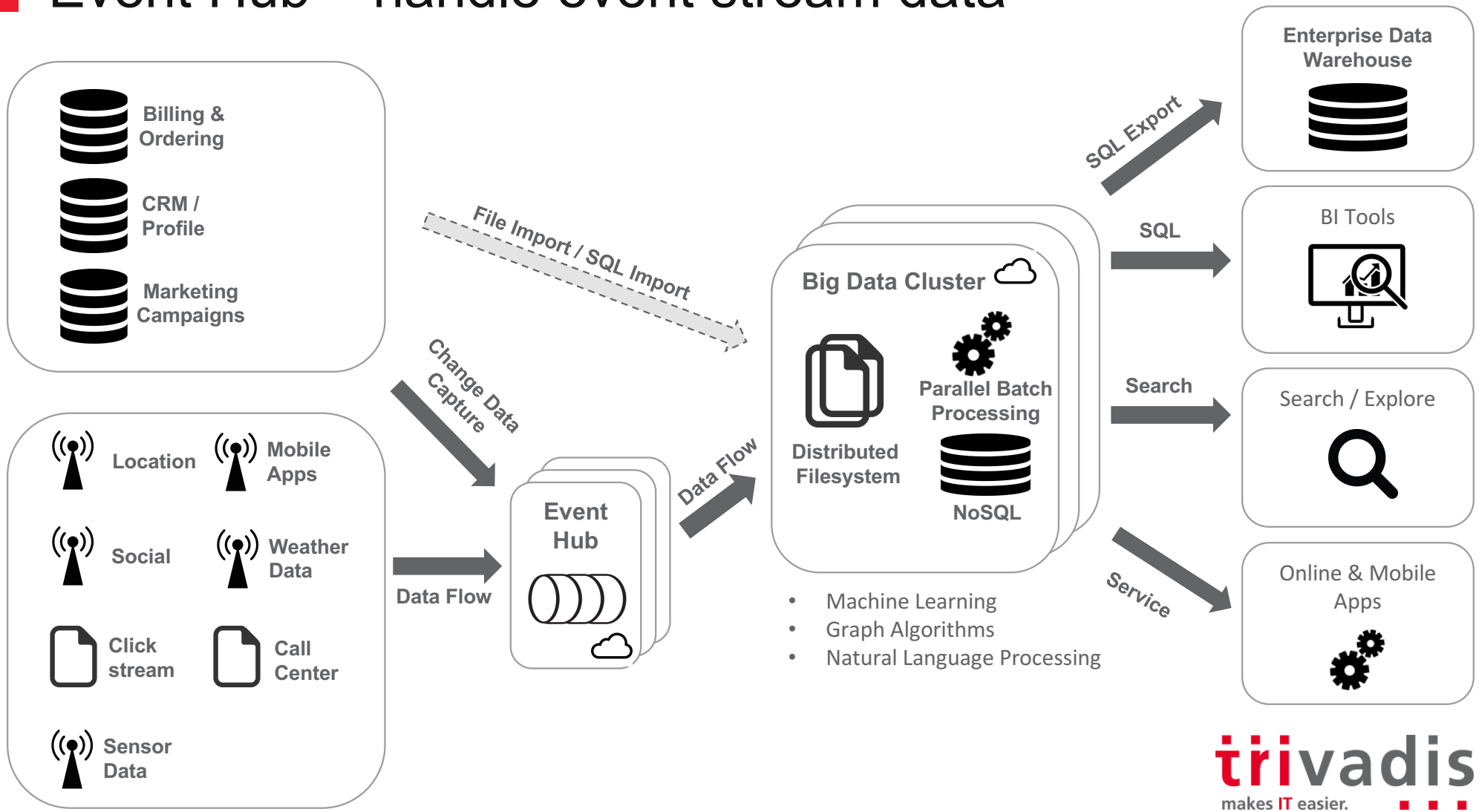
trivadis
makes IT easier.

# Traditional Big Data Architecture



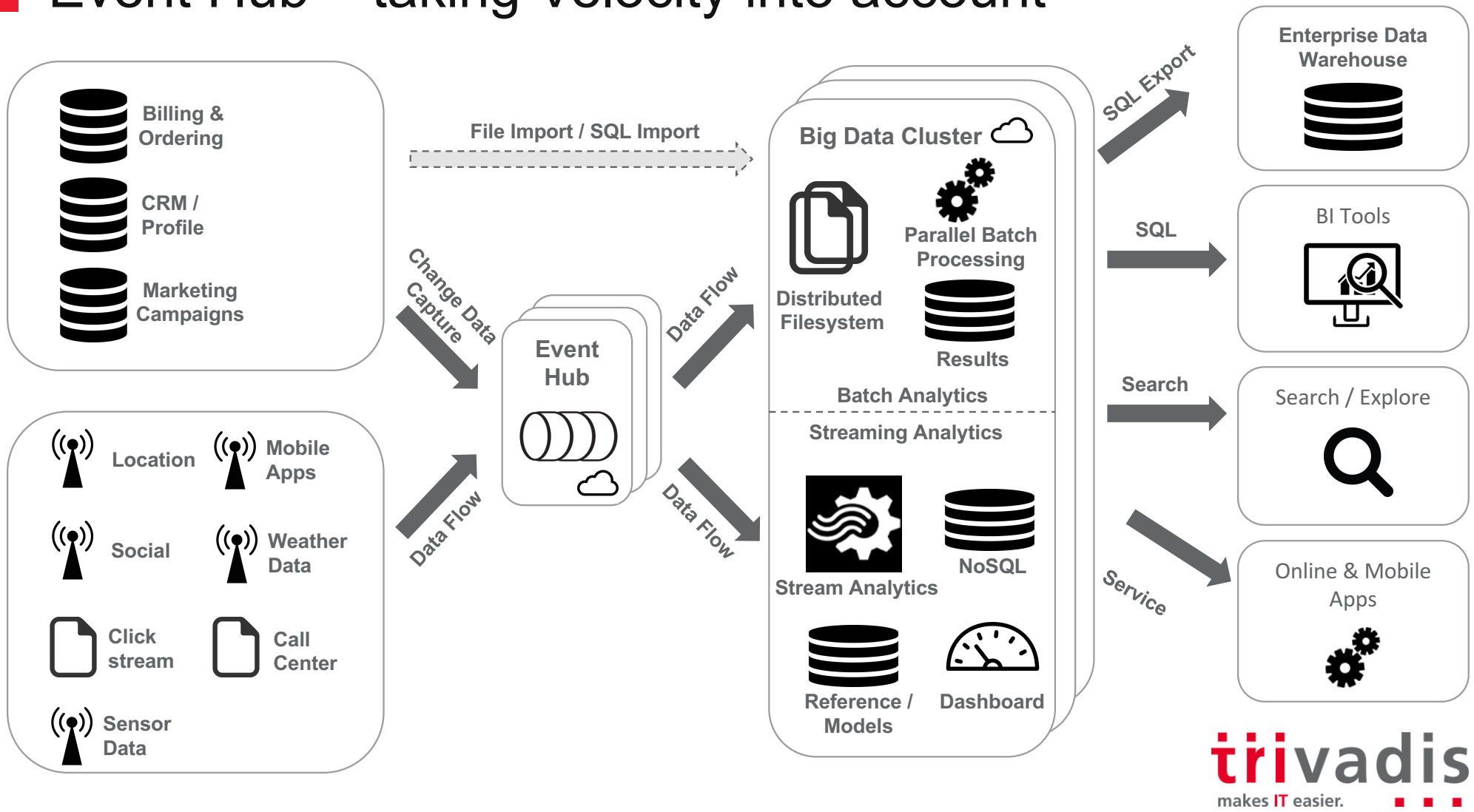

trivadis
makes IT easier.

# Event Hub – handle event stream data



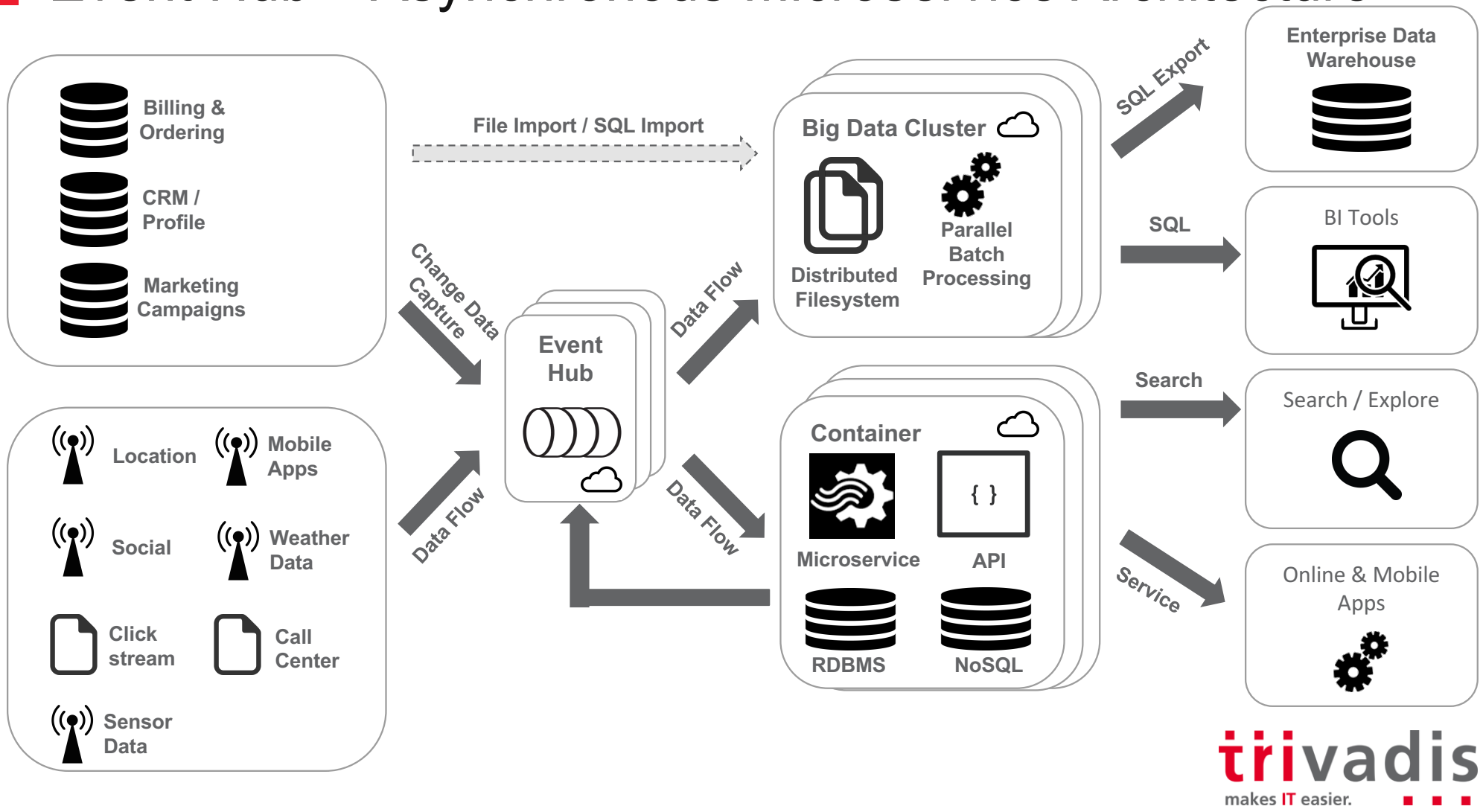

trivadis
makes IT easier.

# Event Hub – taking Velocity into account

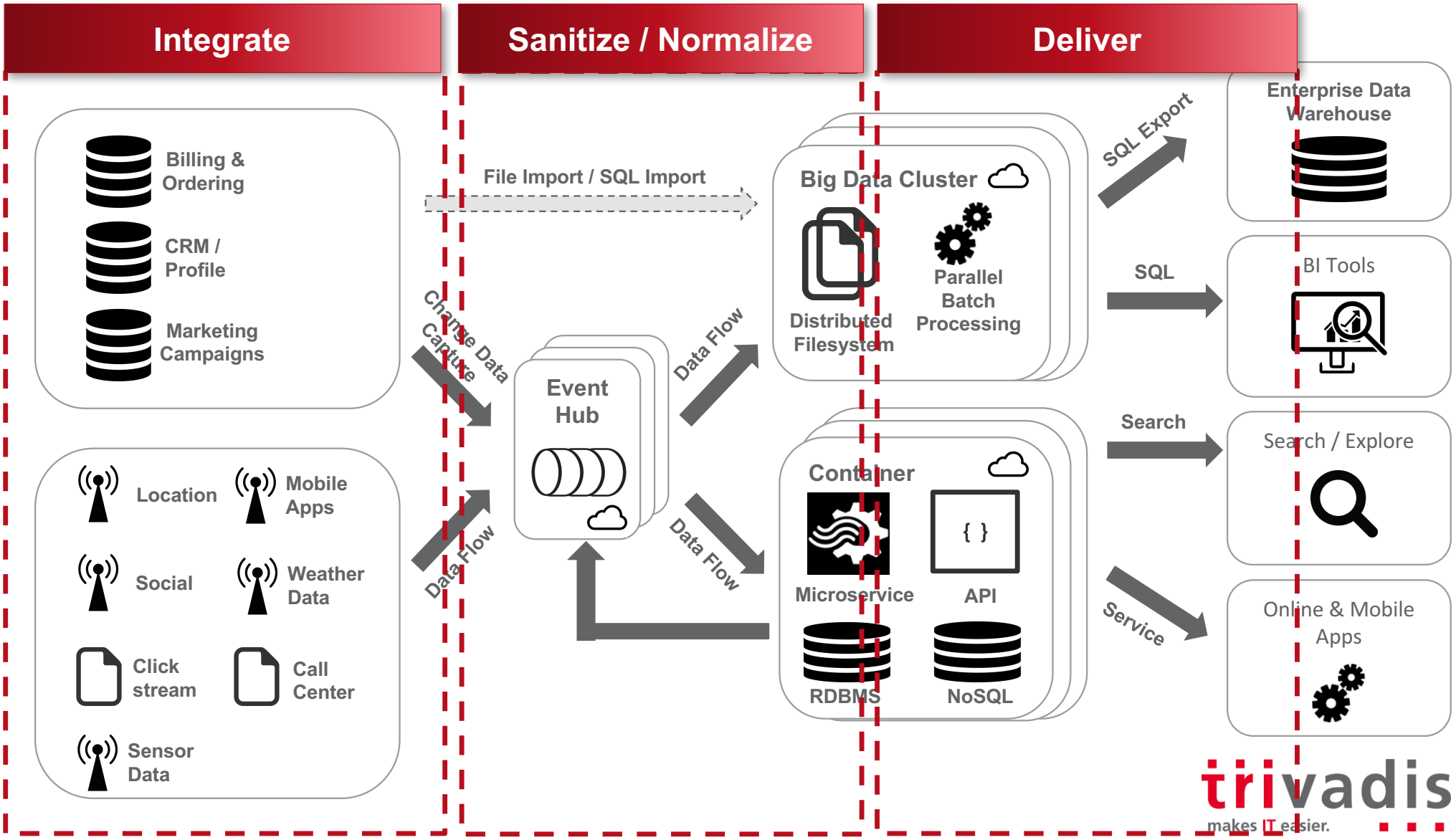



trivadis
makes IT easier.

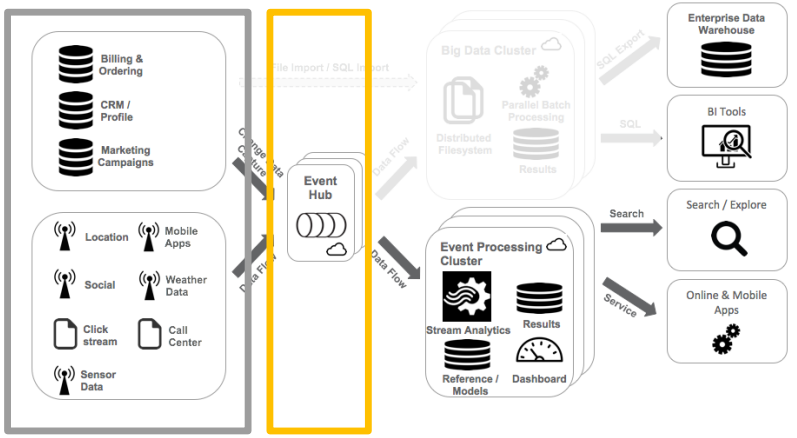
# Event Hub – Asynchronous Microservice Architecture




trivadis
makes IT easier.


Integrate
Sanitize / Normalize
Deliver
Billing & Ordering
CRM / Profile
Marketing Campaigns
Location
Mobile Apps
Social
Weather Data
Click stream
Call Center
Sensor Data
File Import / SQL Import
Change Data Capture
Data Flow
Data Flow
Data Flow
Event Hub
Big Data Cluster
Distributed Filesystem
Parallel Batch Processing
Container
Microservice
API
RDBMS
NoSQL
SQL Export
SQL
Search
Service
Enterprise Data Warehouse
BI Tools
Search / Explore
Online & Mobile Apps
trivadis
makes IT easier.

# Continuous Ingestion - DataFlow Pipelines





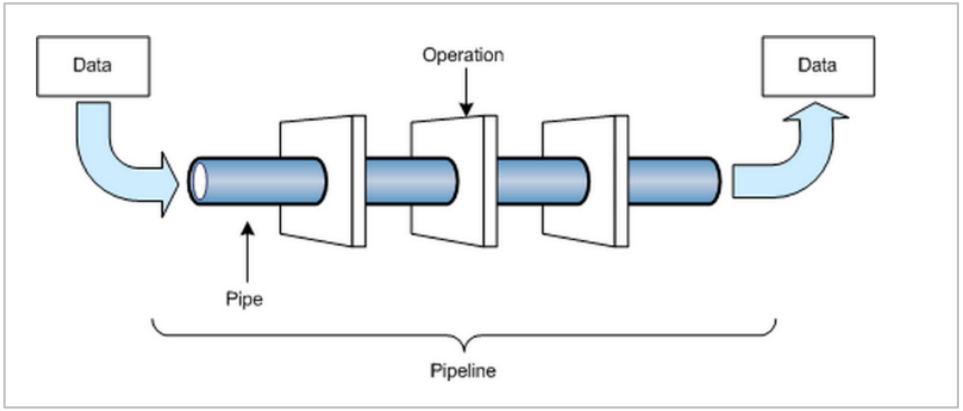


trivadis
makes IT easier.

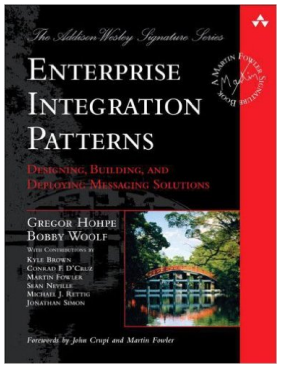
# DataFlow Pipeline

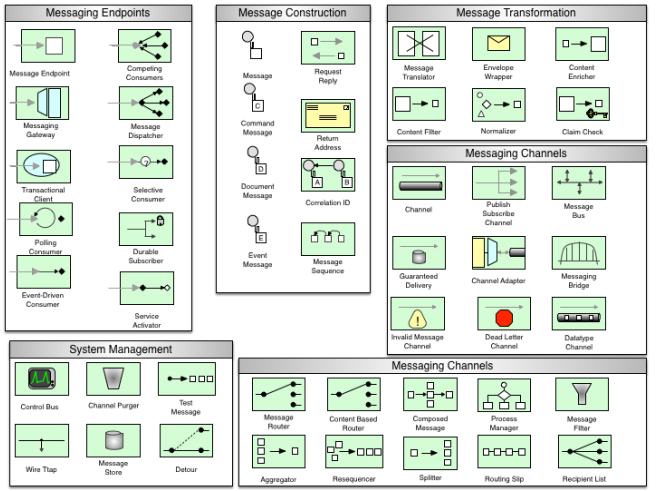
- Flow-based ”programming”
- Ingest Data from various sources
- **E**xtract – **T**ransform – **L**oad
- High-Throughput, straight-through data flows
- Data Lineage
- Batch- or Stream-Processing
- Visual coding with flow editor

- Event Stream Processing (ESP) **but not** Complex Event Processing (CEP)



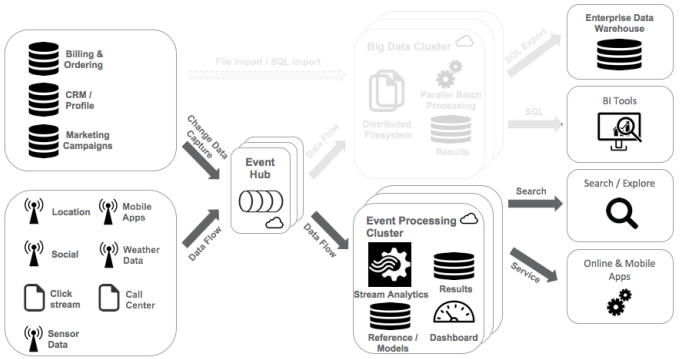

Source: Confluent



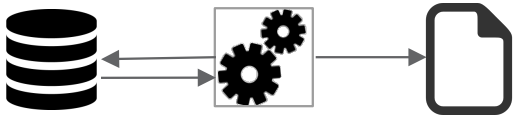




trivadis
makes IT easier.

# Continuous Ingestion – Integrating data sources

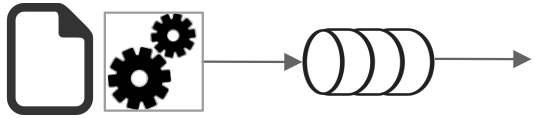



SQL Polling

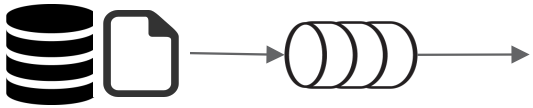


Sensor Stream

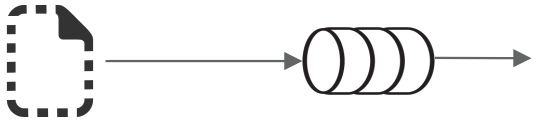
File Polling (File Tailing)



Change Data Capture (CDC)

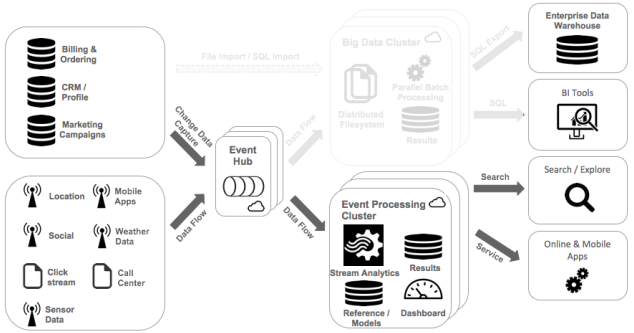


File Stream (Appender)



trivadis
makes IT easier.

# Ingestion with/without Transformation?




## Zero Transformation

- No transformation, plain ingest, no schema validation
- Keep the original format – Text, CSV, …
- Allows to store data that may have errors in the schema

## Enrichment Transformation

- Add new data to the message
- Do not change existing values
- Convert a value from one system to another and add it to the message

## Format Transformation

- Prefer name of Format Translation
- Simply change the format
- Change format from Text to Avro
- Does schema validation

## Value Transformation

- Replaces values in the message
- Convert a value from one system to another and change the value in-place
- Destroys the raw data!

trivadis
makes IT easier.

# Why is Data Ingestion Difficult?

## Infrastructure Drift

Physical and Logical Infrastructure changes rapidly

**Key Challenges:**

Infrastructure Automation
Edge Deployment

## Structure Drift

Data Structures and formats evolve and change unexpectedly

**Key Challenges:**

Consumption Readiness
Corruption and Loss

## Semantic Drift

Data semantics change with evolving applications

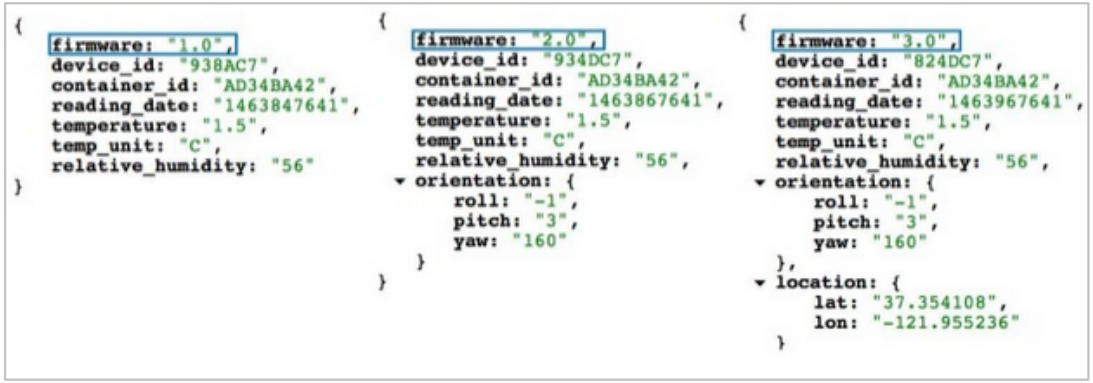
**Key Challenges**

Timely Intervention
System Consistency

Source: Streamsets

# Challenges for Ingesting Sensor Data

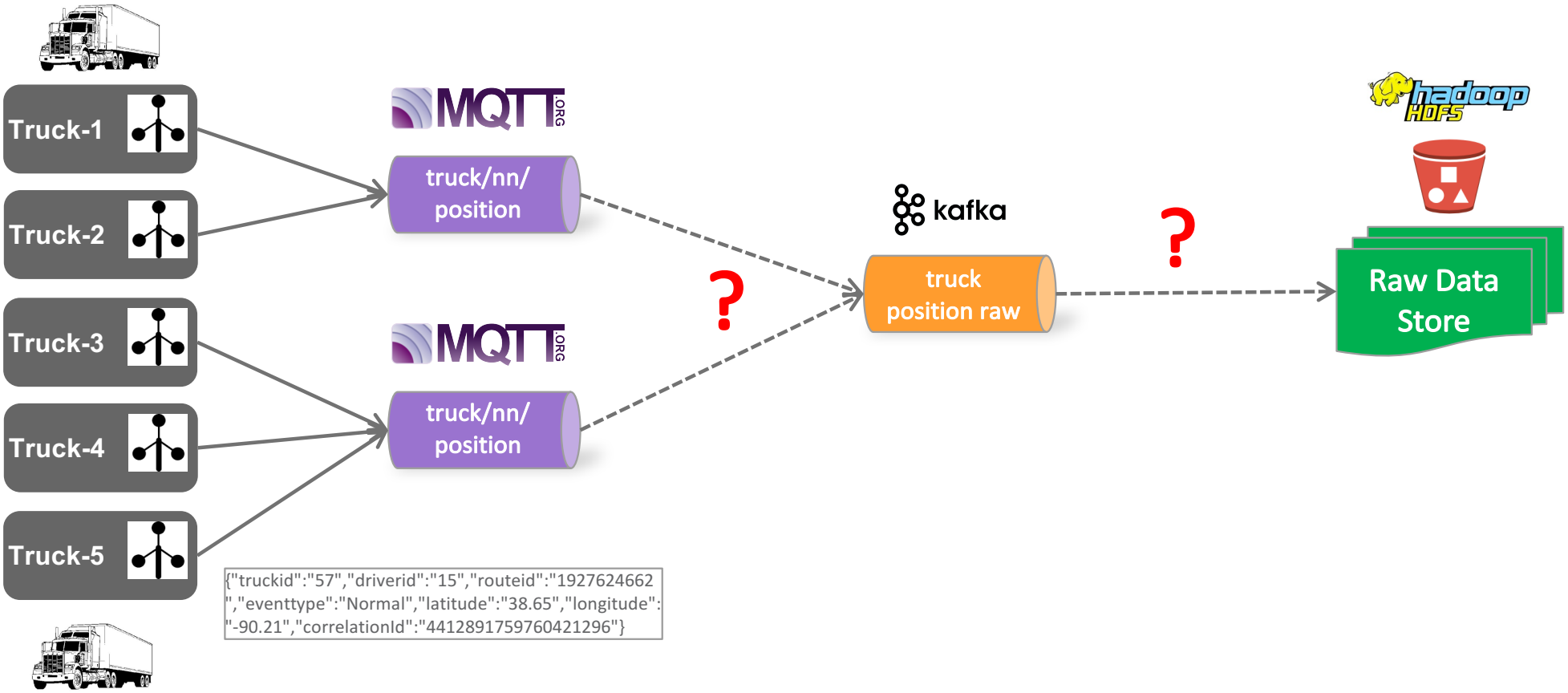
- Multitude of sensors
- Real-Time Streaming
- Multiple Firmware versions
- Bad Data from damaged sensors
- Regulatory Constraints
- Data Quality




Source: Cloudera

trivadis
makes IT easier.

# Demo Case



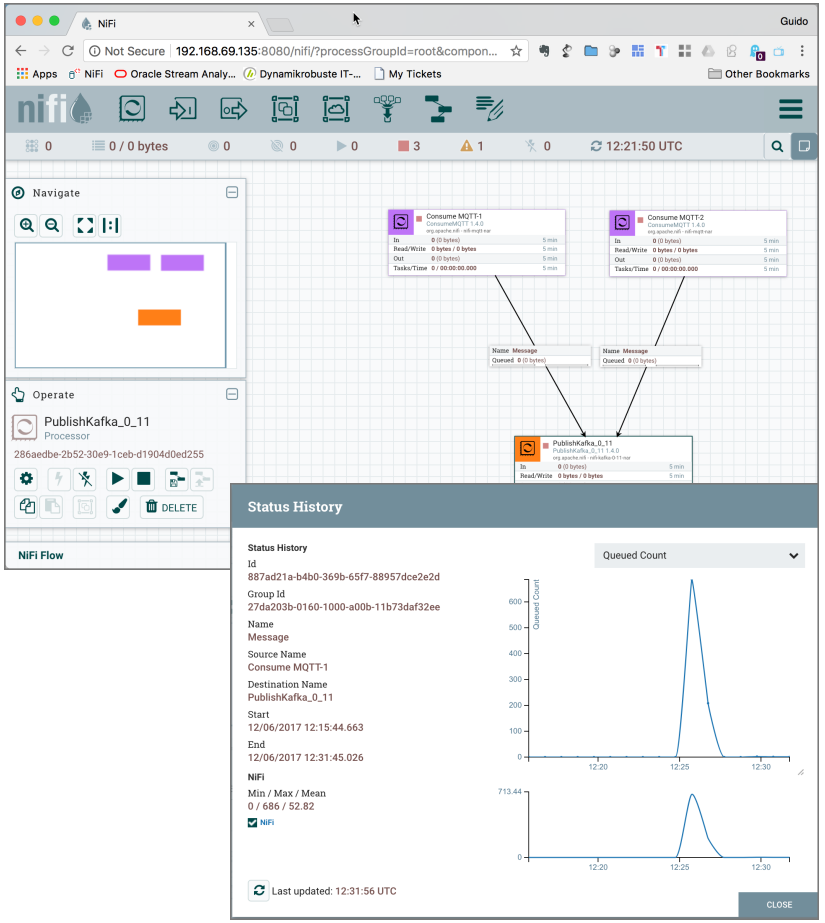

trivadis
makes IT easier.

# Apache NiFi

trivadis
makes IT easier.

# Apache NiFi

- Originated at NSA as Niagarafiles – developed behind closed doors for 8 years
- Open sourced December 2014, Apache Top Level Project July 2015
- Look-and-Feel modernized in 2016
- Opaque, “file-oriented” payload
- Distributed system of processors with centralized control
- Based on *flow-based programming* concepts
- Data Provenance and Data Lineage
- Web-based user interface



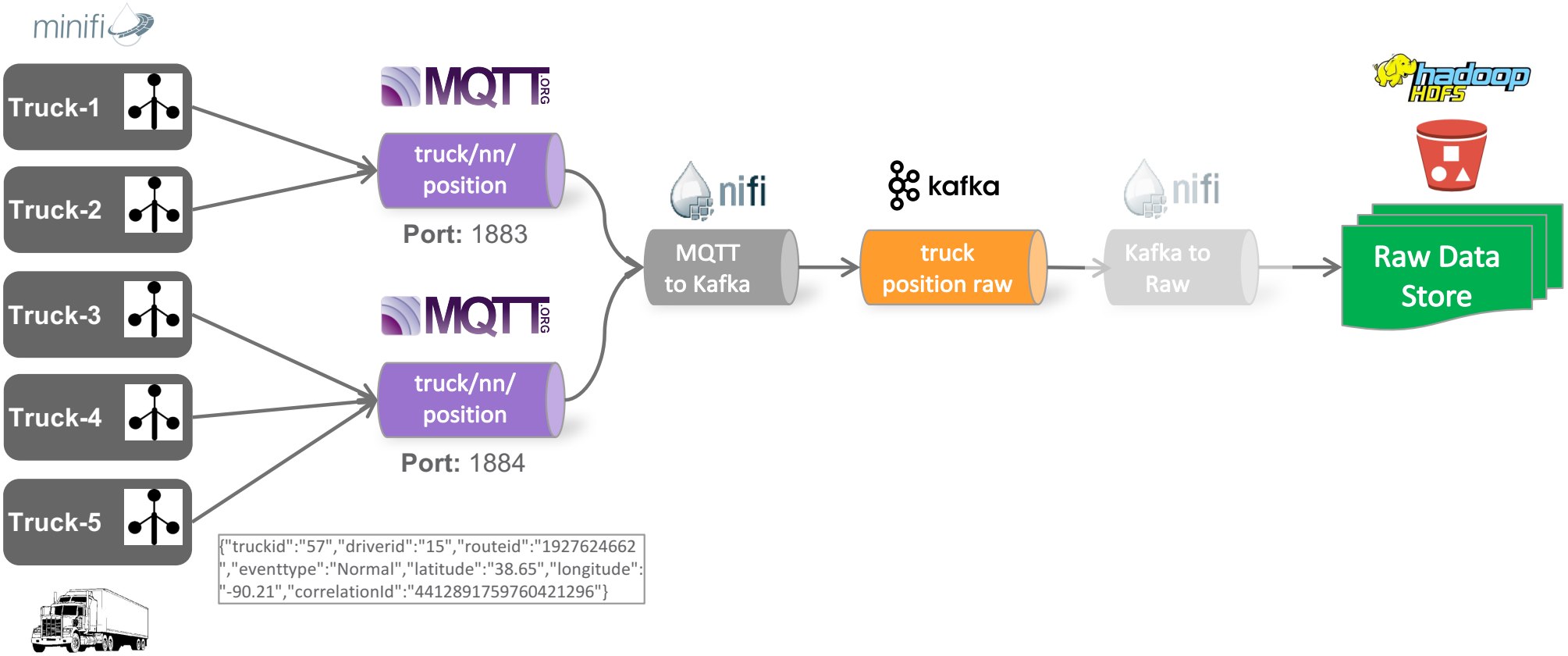
# Processors for Source and Sink


- **ConsumeXXXX** (AMQP, EWS, IMAP, JMS, Kafka, MQTT, POP3, …)
- **DeleteXXXX** (DynamoDB, Elasticsearch, HDFS, RethinkDB, S3, SQS, ...)
- **FetchXXXX** (AzureBlobStorage, ElasticSearch, File, FTP, HBase, HDFS, S3 ...)
- **ExecuteXXXX** (FlumeSink, FlumeSource, Script, SQL, ...)
- **GetXXXX** (AzureEventHub, Couchbase, DynamoDB, File, FTP, HBase, HDFS, HTTP, Ignite, JMSQueue, JMSTopic, Kafka, Mongo, Solr, Splunk, SQS, TCP, ...)
- **ListenXXXX** (HTTP, RELP, SMTP, Syslog, TCP, UDP, WebSocket, ...)
- **PublishXXXX** (Kafka, MQTT)
- **PutXXXX** (AzureBlobStorage, AzureEventHub, CassandraQL, CloudWatchMetric, Couchbase, DynamoDB, Elasticsearch, Email, FTP, File, Hbase, HDFS, HiveQL, Kudu, Lambda, Mongo, Parquet, Slack, SQL, TCP, ....)
- **QueryXXXX** (Cassandra, DatabaseTable, DNS, Elasticserach)

# Processors for Processing

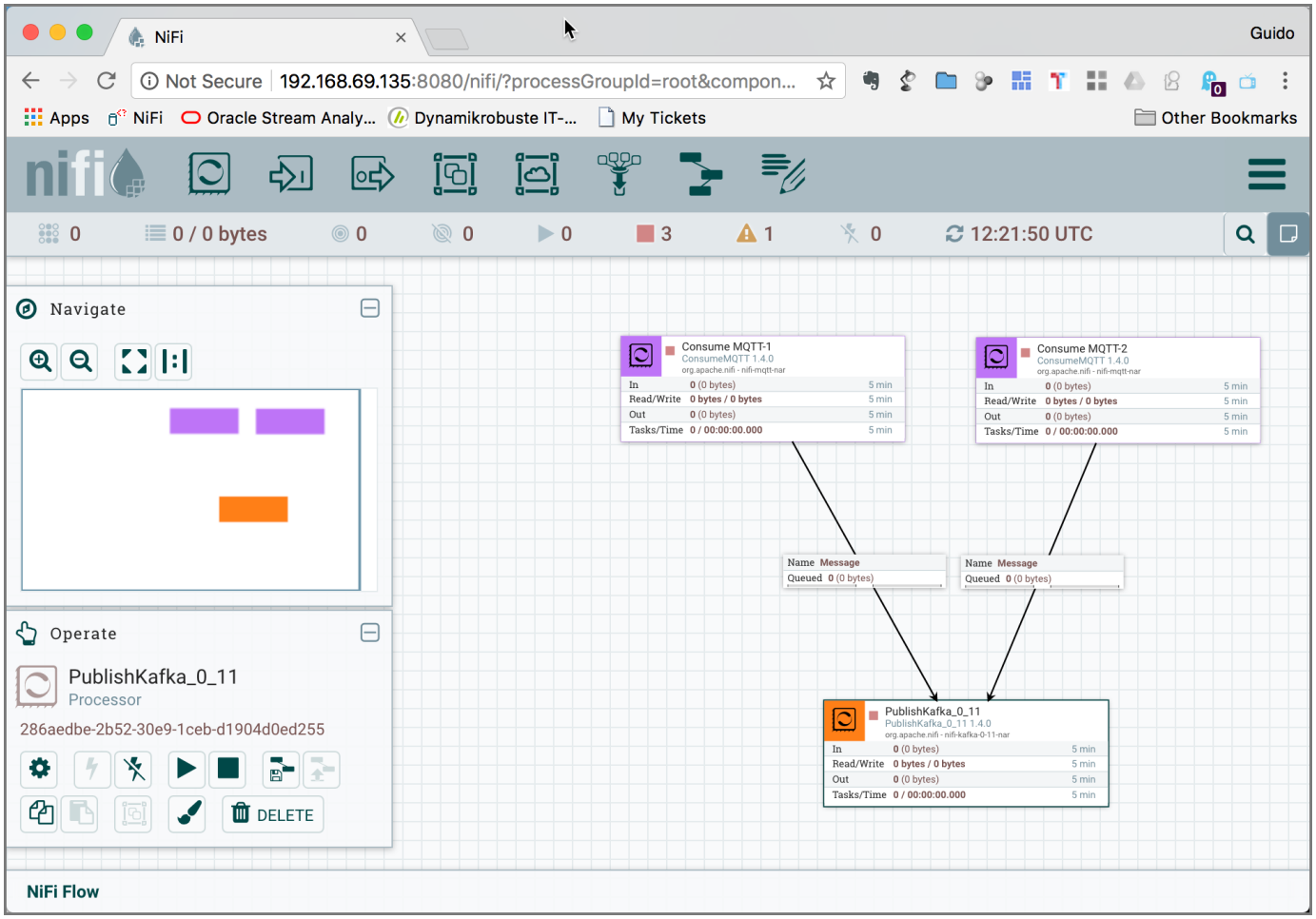

- **ConvertXxxxToYyyy**
- **ConvertRecord**
- **EnforceOrder**
- **EncryptContent**
- **ExtractXXXX** (AvroMetdata, EmailAttachments, Grok, HL7Attributes, ImageMetadata, ...)
- **GeoEnrichIP**
- **JoltTransformJSON**
- **MergeContent**
- **ReplaceText**
- **ResizeImage**
- **SplitXXXX** (Avro, Content, JSON, Record, Xml, ...)
- **TailFile**
- **TransformXML**
- **UpdateAttribute**

# Demo Case




trivadis
makes IT easier.

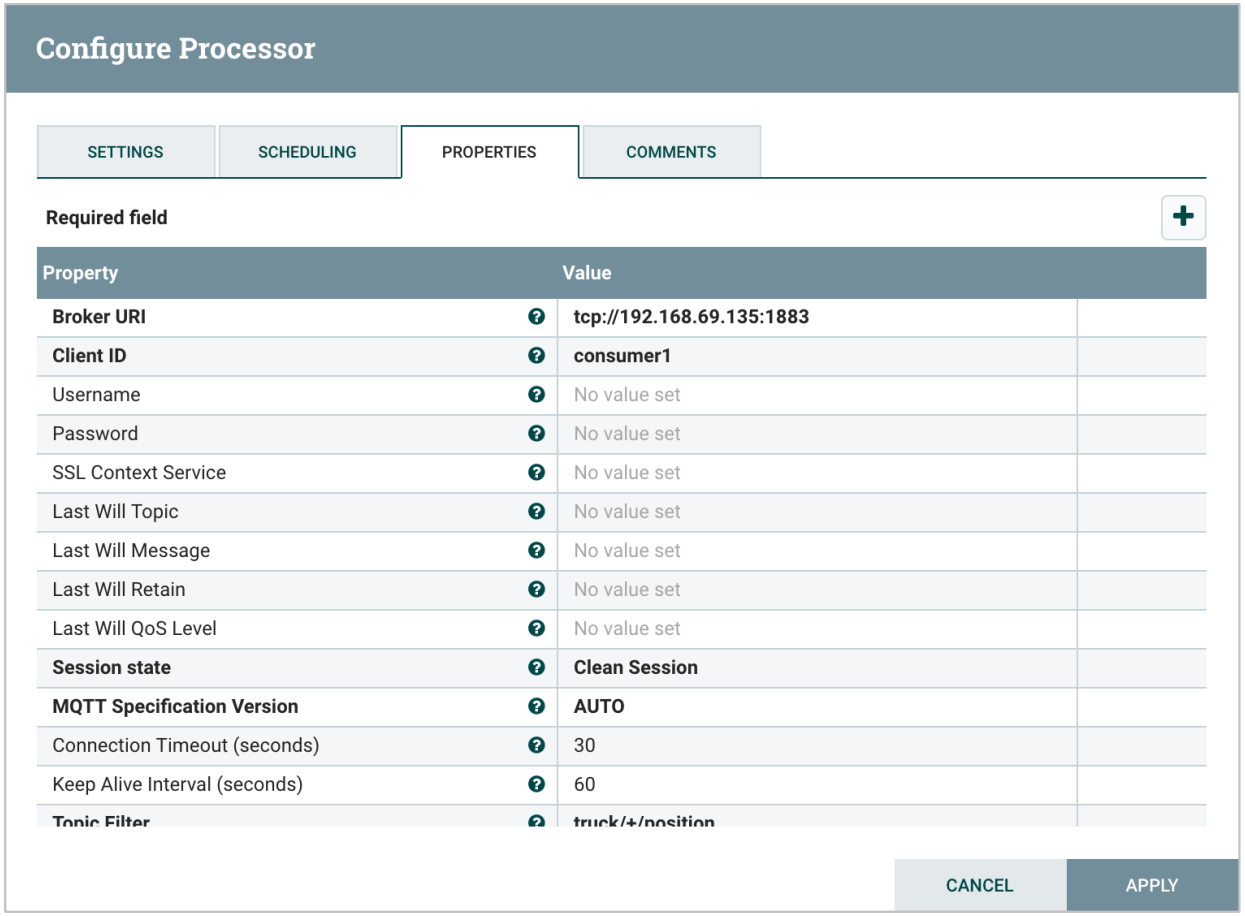
# Demo: Dataflow for MQTT to Kafka



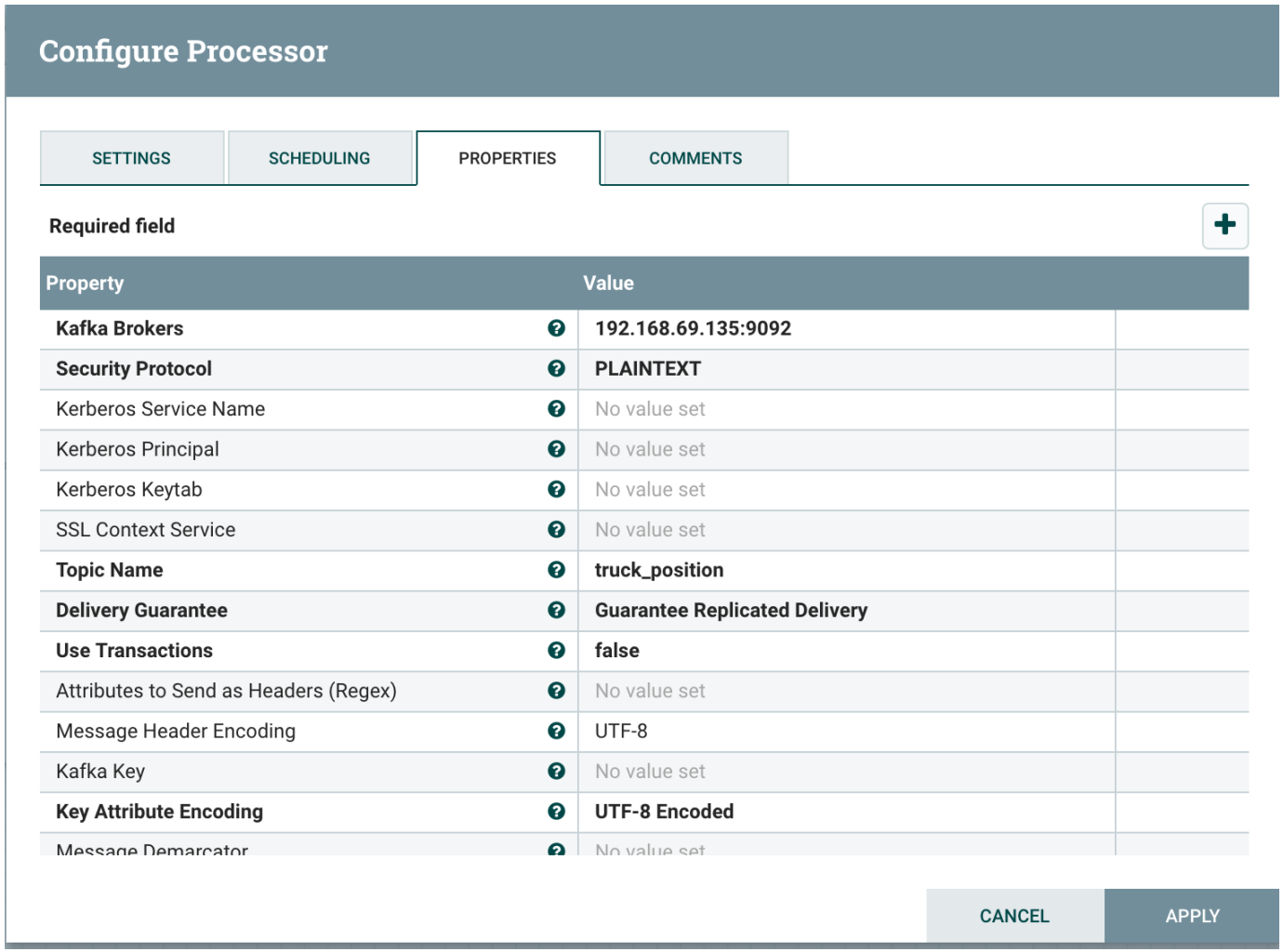

nifi

trivadis
makes IT easier.

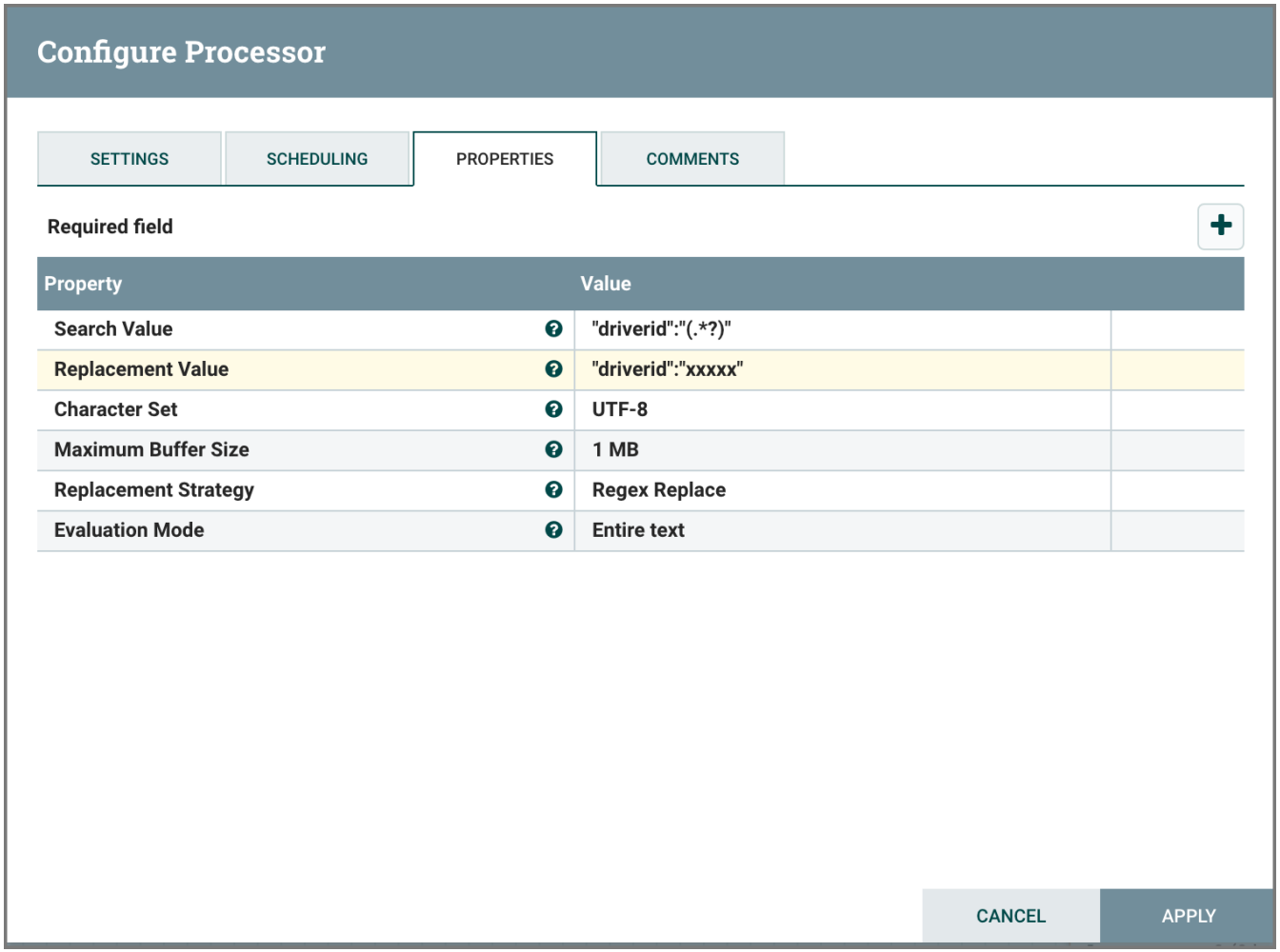
# Demo: MQTT Processor



# Demo: Kafka Processor



# Demo: Masking Field with ReplaceText Processor



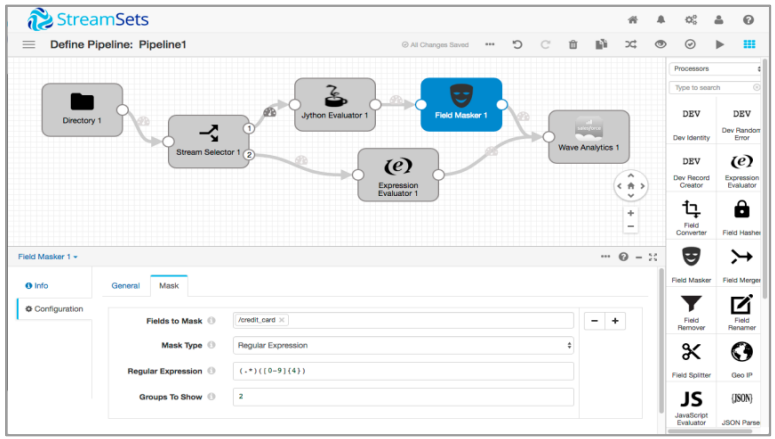
# StreamSets

trivadis
makes IT easier.

# StreamSets Data Collector

StreamSets

- Founded by ex-Cloudera, Informatica employees
- Continuous open source, intent-driven, big data ingest
- Visible, record-oriented approach fixes combinatorial explosion
- Batch or stream processing
  - Standalone, Spark cluster, MapReduce cluster
- IDE for pipeline development by ‘civilians’
- Relatively new - first public release September 2015
- So far, vast majority of commits are from StreamSets staff





trivadis
makes IT easier.

# StreamSets Origins

StreamSets

An origin stage represents the source for the pipeline. You can use a single origin stage in a pipeline

Origins on the right are available out of the box

API for writing custom origins


Source: https://streamsets.com/connectors

trivadis makes IT easier.

# StreamSets Processors

StreamSets

A processor stage represents a type of data processing that you want to perform

use as many processors in a pipeline as you need

Programming languages supported

- Java
- JavaScript
- Jython
- Groovy
- Java Expression Language (EL) Spark

Some of processors available out-of-the-box:

- Expression Evaluator
- Field Flattener
- Field Hasher
- Field Masker
- Field Merger
- Field Order
- Field Splitter
- Field Zip
- Groovy Evaluator
- JDBC Lookup
- JSON Parser
- Spark Evaluator
- …

trivadis
makes IT easier.

# StreamSets Destinations

StreamSets

A destination stage represents the target for a pipeline. You can use one or more destinations in a pipeline

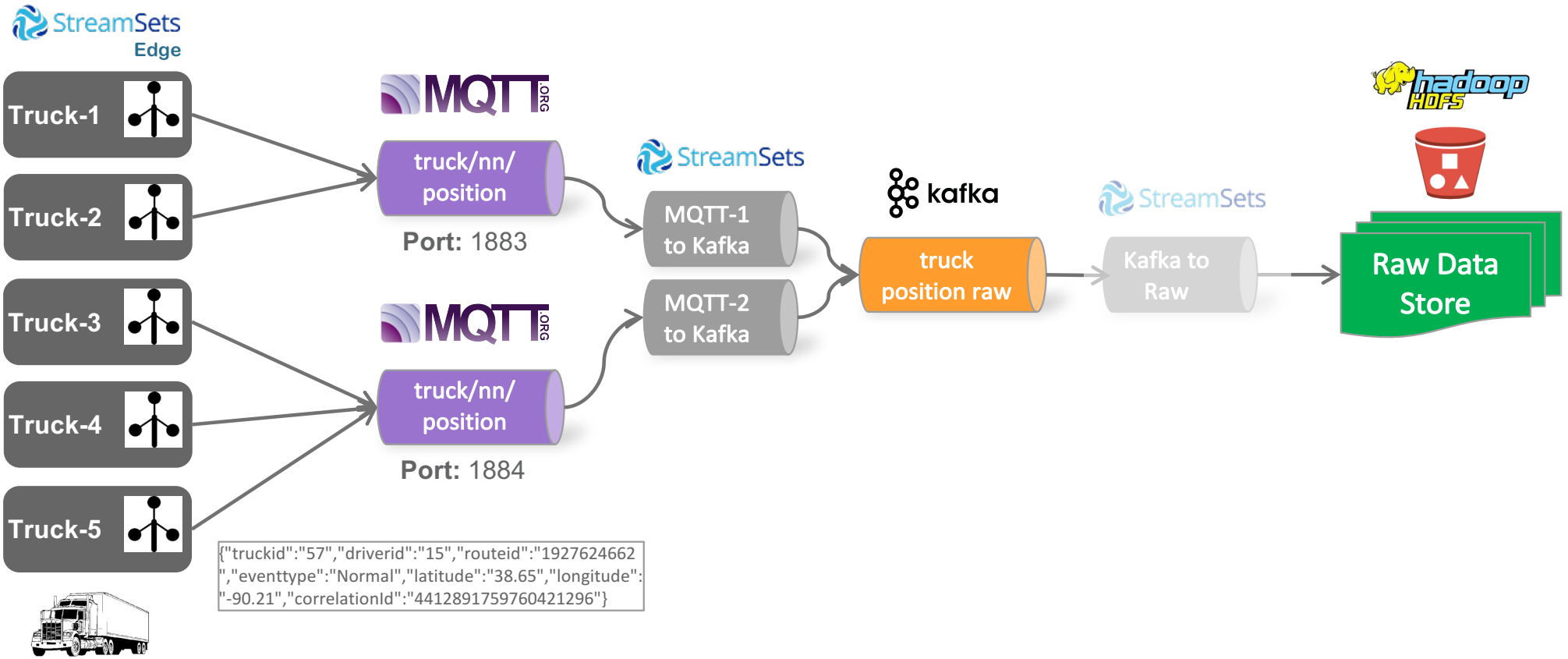
Destinations on the right are available out of the box

API for writing custom origins


Source: https://streamsets.com/connectors

trivadis makes IT easier.

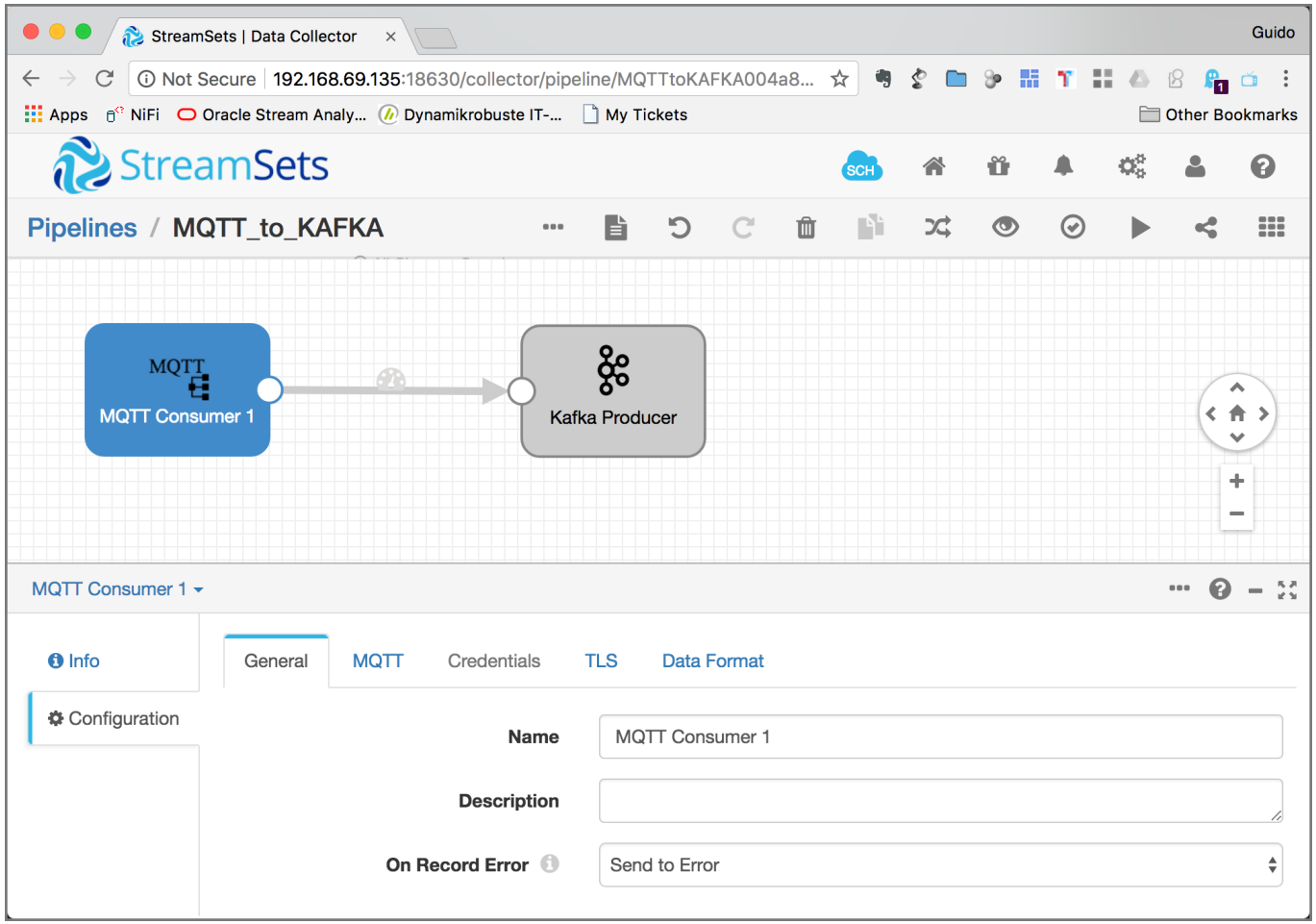
# Demo Case




trivadis
makes IT easier.

# Demo: Dataflow for MQTT to Kafka

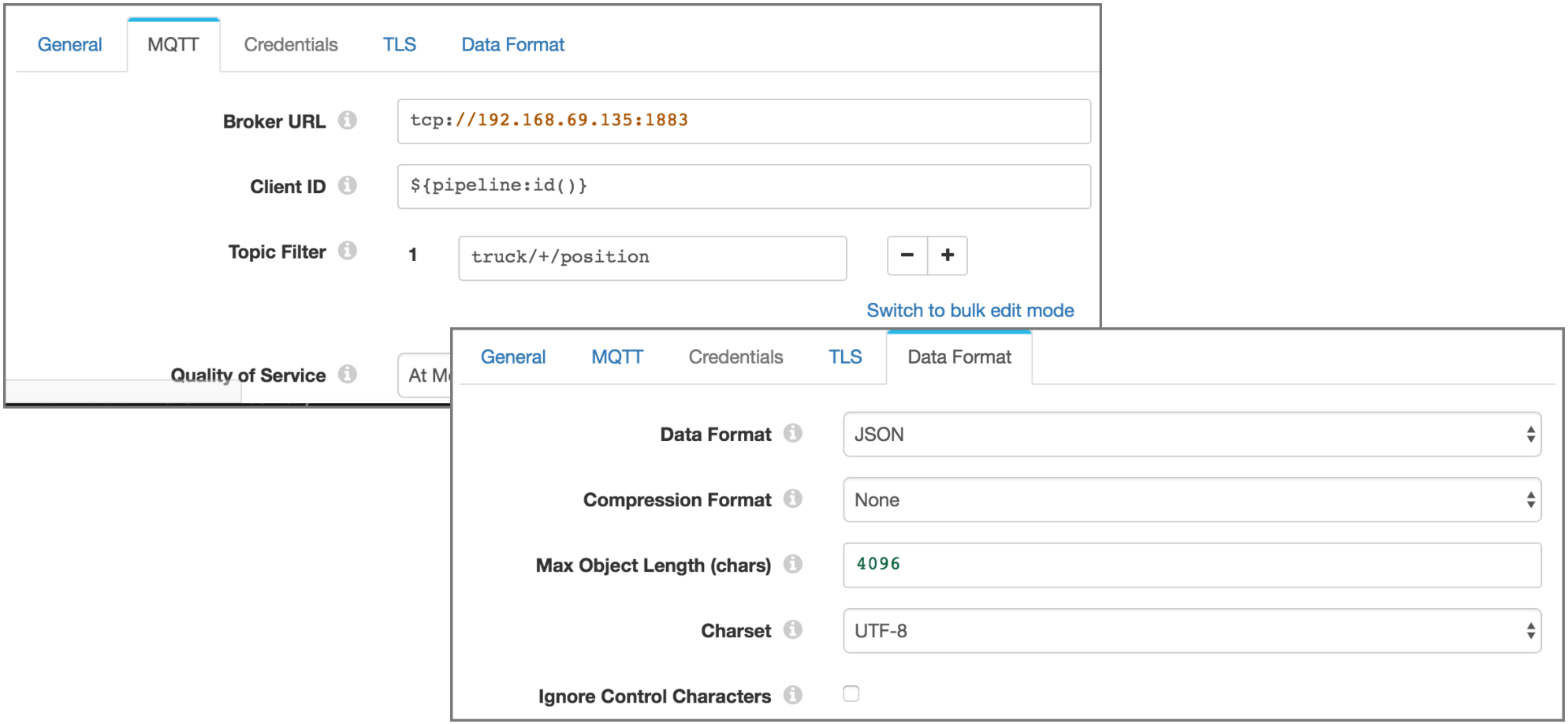
StreamSets




trivadis
makes IT easier.

# Demo: MQTT Source

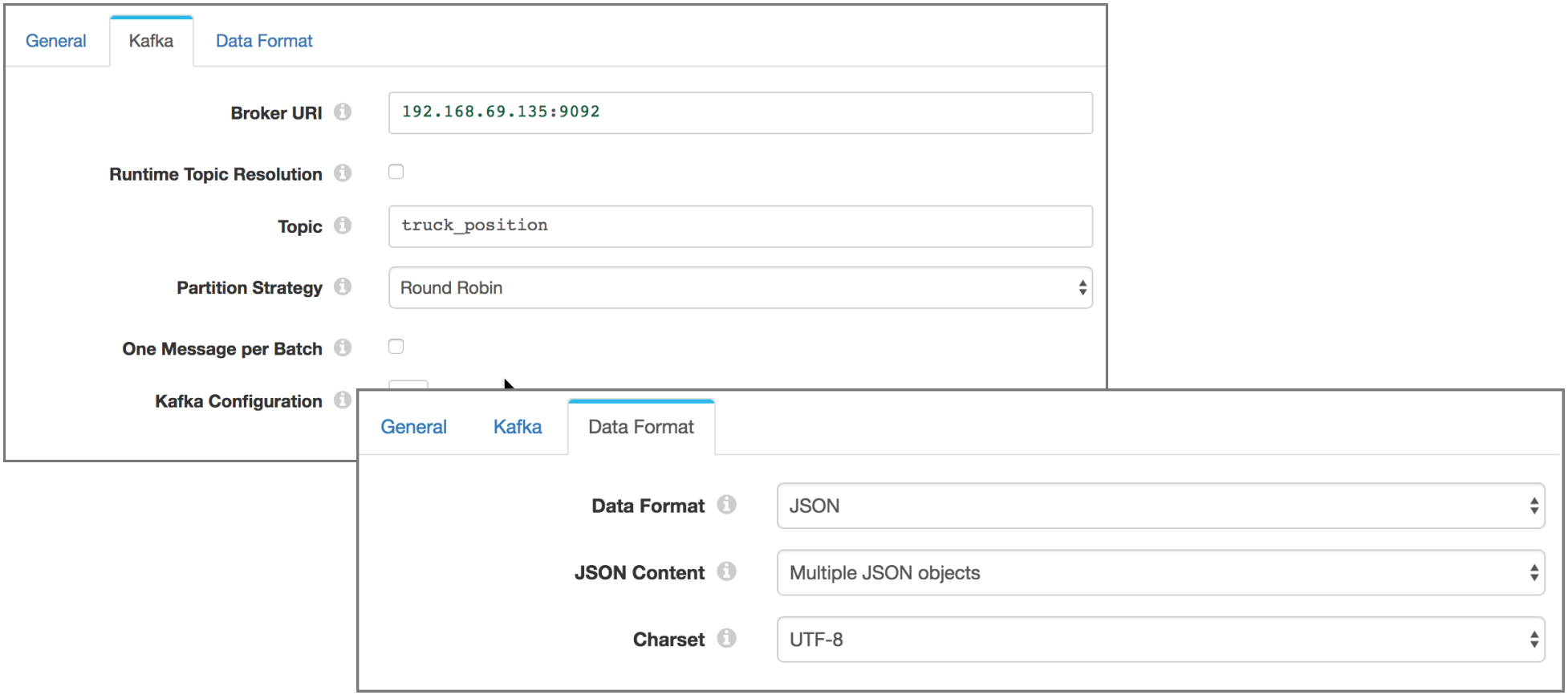
StreamSets




trivadis
makes IT easier.

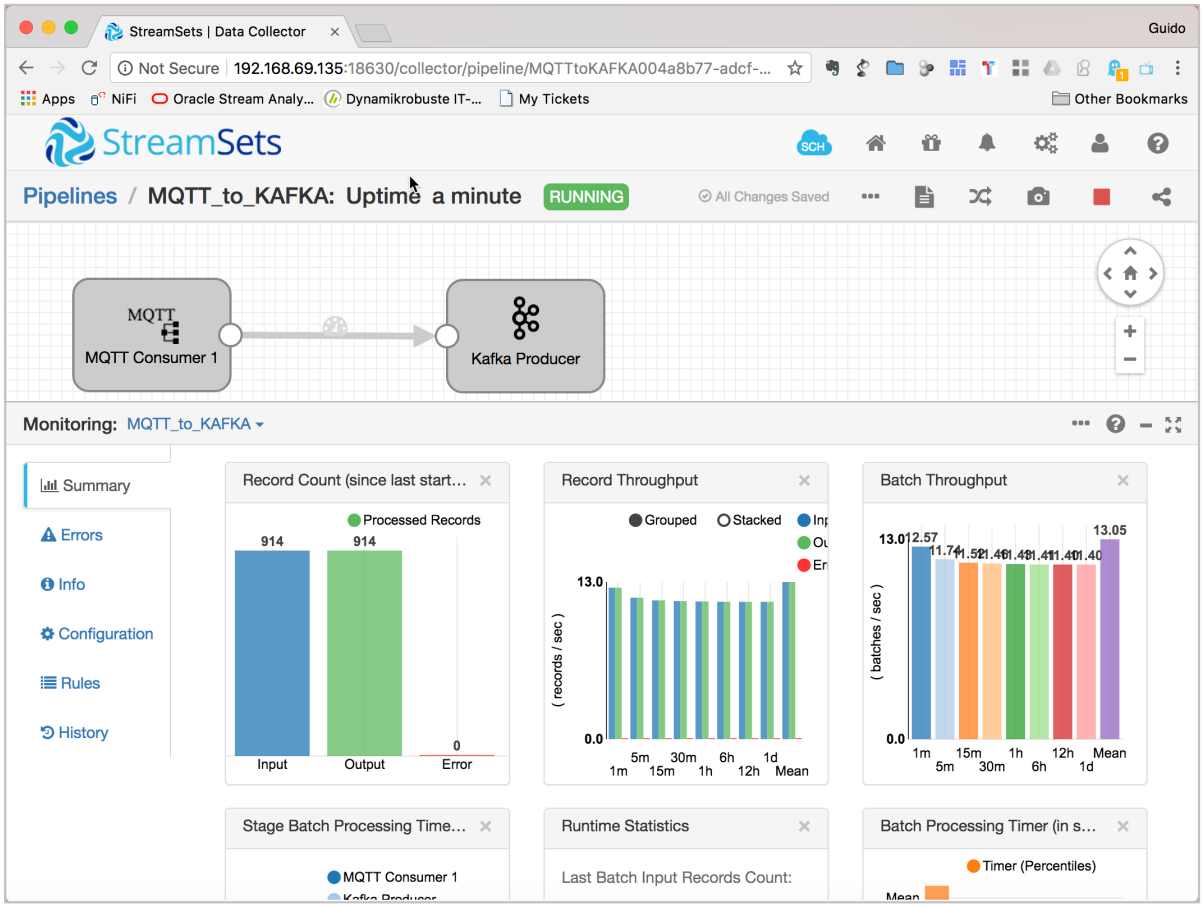
# Demo: Kafka Sink

StreamSets




trivadis
makes IT easier.

# Demo: Dataflow for MQTT to Kafka

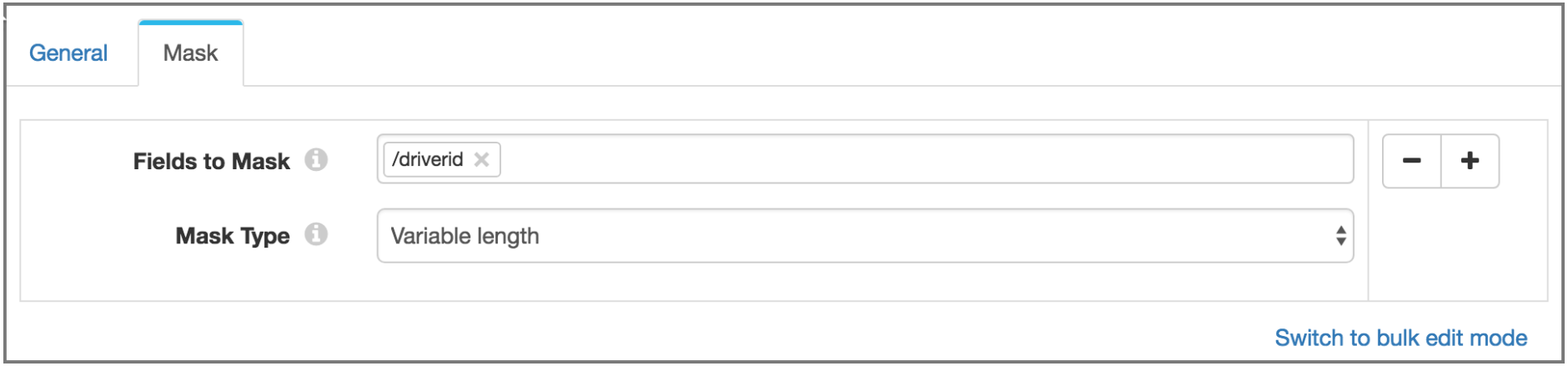



StreamSets

trivadis
makes IT easier.

# Demo: Masking fields

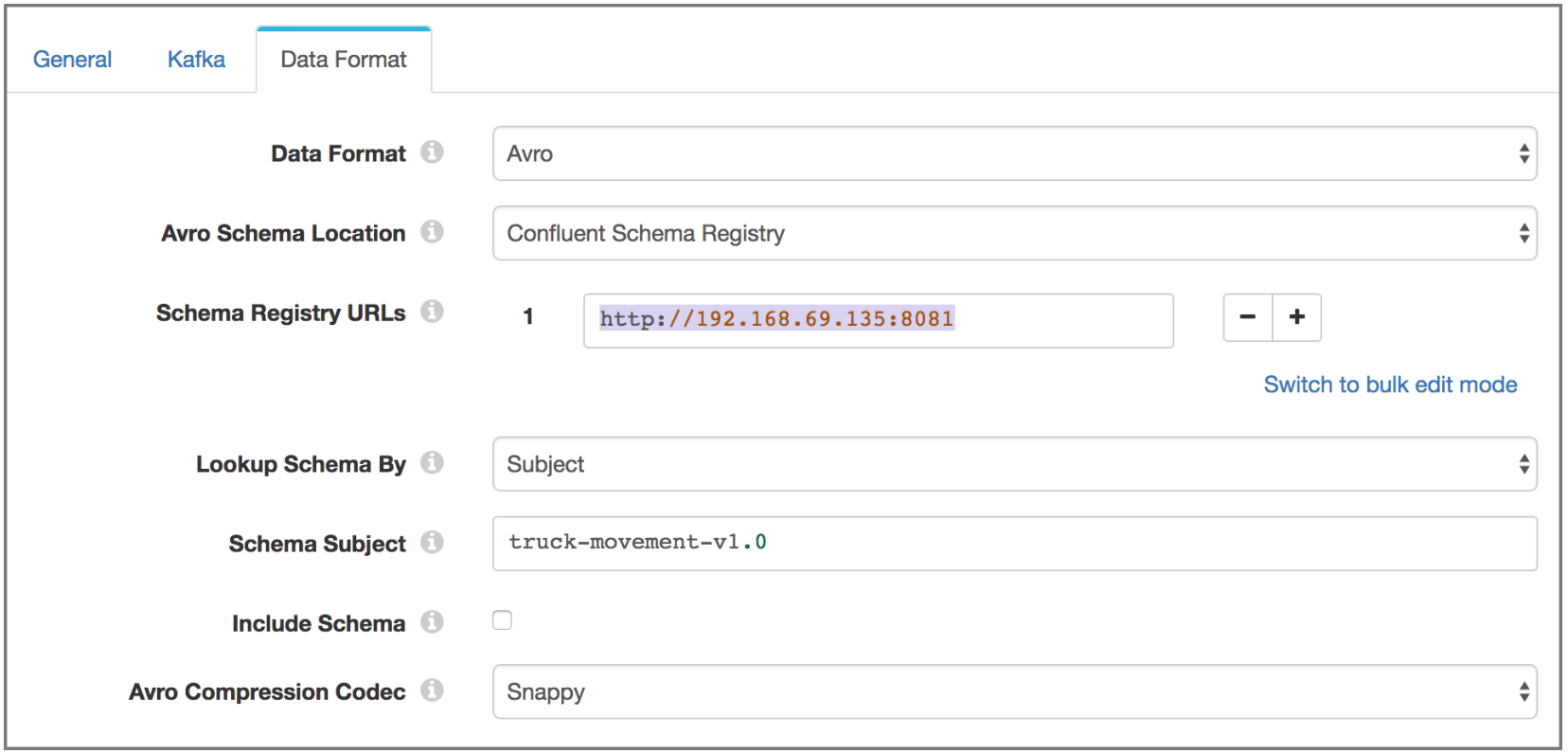
StreamSets




trivadis
makes IT easier.

# Demo: Sending Message to Kafka in Avro

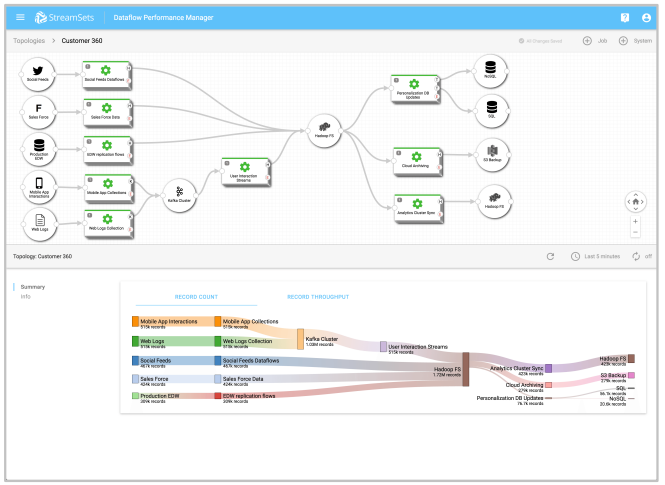
StreamSets

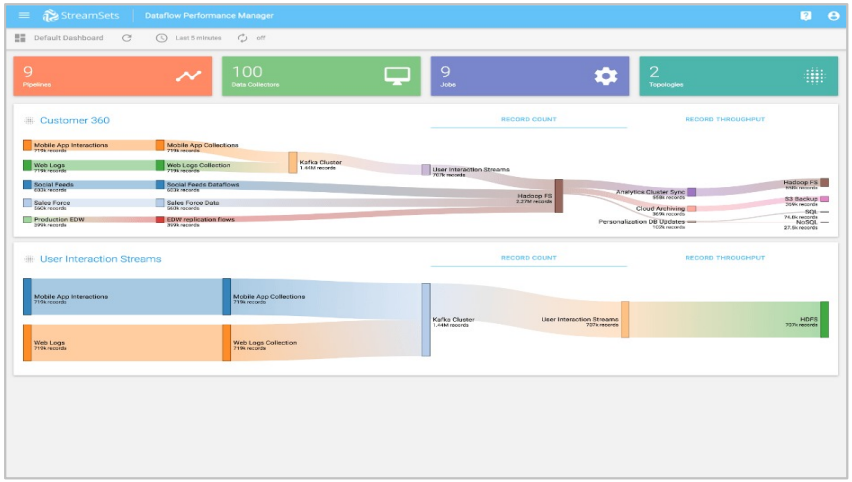



trivadis
makes IT easier.

# StreamSets Dataflow Performance Manager

- Map dataflows to topologies, manage releases & track changes
- Measure KPIs and establish baselines for data availability and accuracy
- Master dataflow operations through Data SLAs





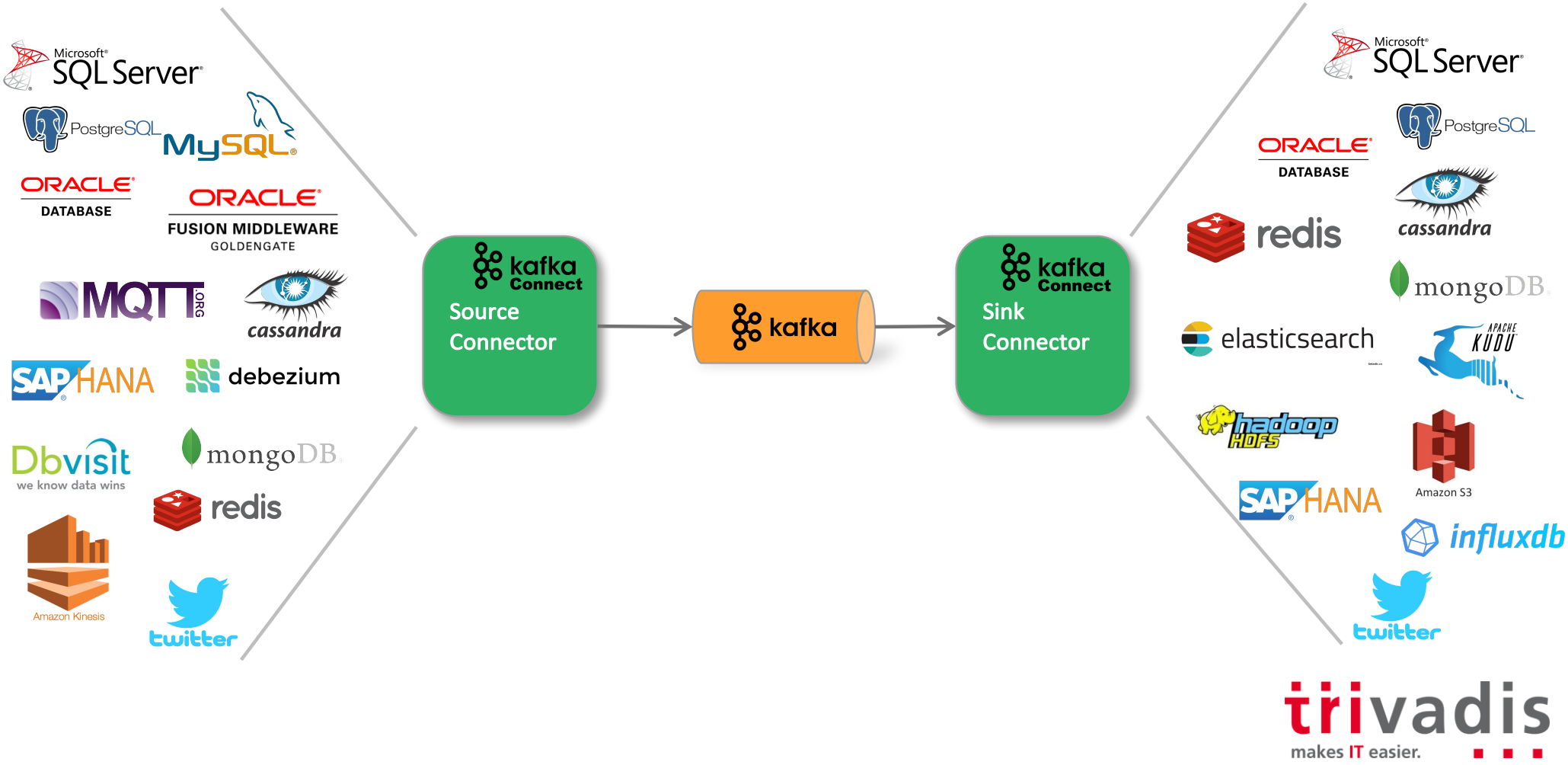


Source: https://streamsets.com/connectors

# Kafka Connect

trivadis
makes IT easier.

# Kafka Connect - Overview




trivadis
makes IT easier.

# Kafka Connect – Single Message Transforms (SMT)

Simple Transformations for a single message

Defined as part of Kafka Connect
- some useful transforms provided out-of-the-box
- Easily implement your own

Optionally deploy 1+ transforms with each connector
- Modify messages produced by source connector
- Modify messages sent to sink connectors

Makes it much easier to mix and match connectors

Some of currently available transforms:
- `InsertField`
- `ReplaceField`
- `MaskField`
- `ValueToKey`
- `ExtractField`
- `TimestampRouter`
- `RegexRouter`
- `SetSchemaMetaData`
- `Flatten`
- `TimestampConverter`

# Kafka Connect – Many Connectors

60+ since first release (0.9+)

20+ from Confluent and Partners

## Confluent supported Connectors

| CONNECTOR | TAGS | DEVELOPER/SUPPORT | DOWNLOAD |
|---|---|---|---|
| Amazon S3 (Sink) | S3 | Confluent | Confluent |
| Elasticsearch (Sink) | search, Elastic, log, analytics | Confluent | Confluent |
| HDFS (Sink) | HDFS, Hadoop, Hive | Confluent | Confluent |
| JDBC (Source) | JDBC, MySQL | Confluent | Confluent |
| JDBC (Sink) | JDBC, MySQL | Confluent | Confluent |

## Certified Connectors

| CONNECTOR | TAGS | DEVELOPER/SUPPORT | DOWNLOAD |
|---|---|---|---|
| Attunity (Source) | CDC | Attunity | Attunity |
| Azure IoTHub (Source) | IoT, messaging | Microsoft | Azure |
| Couchbase (Source) | Couchbase, NoSQL | Couchbase | Couchbase |
| Couchbase (Sink) | Couchbase, NoSQL | Couchbase | Couchbase |
| DataStax (Sink) | Cassandra, DataStax | Data Mountaineer | Data Mountaineer |
| Dbvisit Replicate (Source) | CDC, Oracle | Dbvisit | Dbvisit |
| IBM Data Replication (Source) | CDC | IBM | IBM |
| JustOne (Sink) | Postgres | JustOne | JustOne |
| Kinetica (Source) | GPU, RDBMS | Kinetica | Kinetica |
| Kinetica (Sink) | GPU, RDBMS | Kinetica | Kinetica |
| SAP HANA (Sink) | HANA, RDBMS | SAP | Community |
| SAP HANA (Source) | HANA, RDBMS | SAP | Community |
| 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 |
| VoltDB (Sink) | VoltDB, NewSQL | VoltDB | VoltDB |

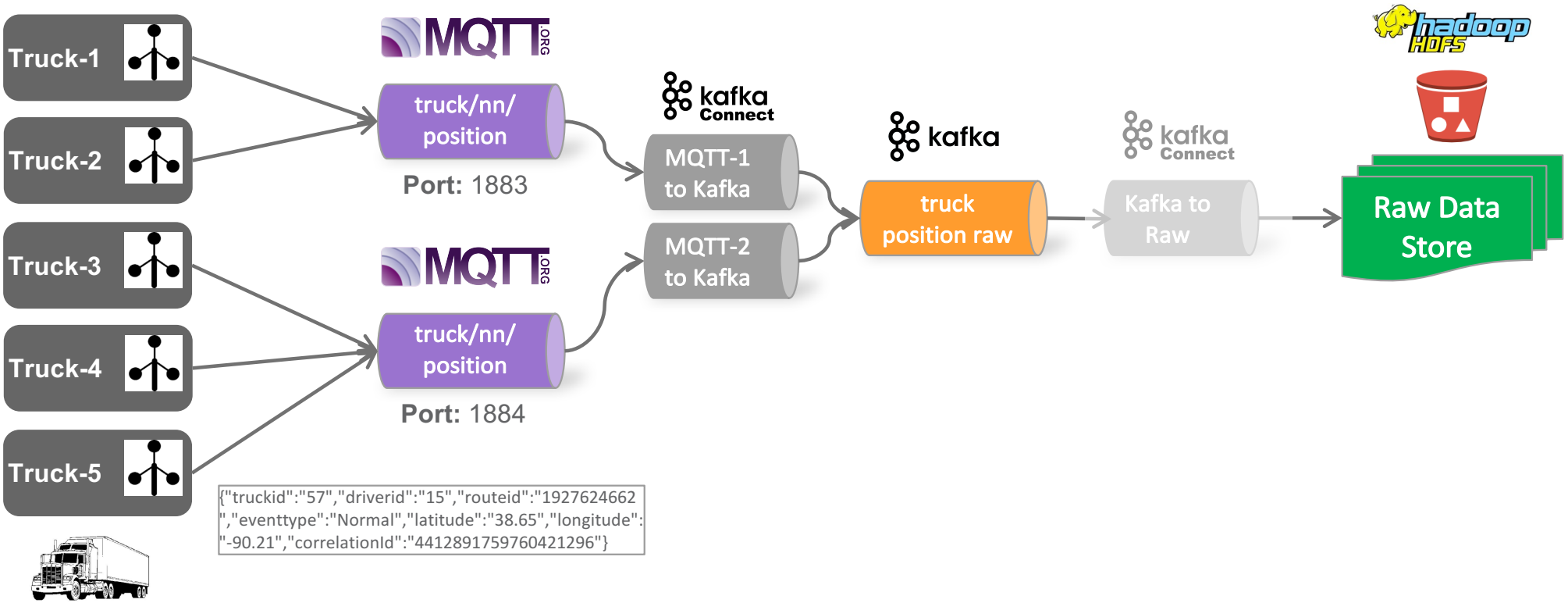
## Community Connectors

| CONNECTOR | TAGS | DEVELOPER/SUPPORT | DOWNLOAD |
|---|---|---|---|
| Apache Ignite (Source) | File System | Community | Community |
| Apache Ignite (Sink) | File System | Community | Community |
| Azure DocumentDB (Sink) | DocumentDB, Azure, NoSQL | Community | Community |
| Blockchain (Source) | Bitcoin, Blockchain | Community | Community |
| Bloomberg Ticker (Source) | Application feed | Community | Community |
| Cassandra (Source) | Cassandra | Community | Community 1 |
| Cassandra (Sink) | Cassandra | Community | Community |
| CoAP (Source) | Constrained Application Protocol, CoAP | Community | Community |
| DynamoDB | Dynamo, NoSQL | Community | Community |
| Elasticsearch (Sink) | Elastic, search, log, analytics | Community | Community 1<br>Community 2<br>Community 3 |
| Files/Directories (Source) | File System, Directories, Logs | Community | Community 1<br>Community 2 |
| FileSystem Connector (Source) | File System, S3, HDFS | Community | Community |
| FTP (Source) | File System | Community | Community 1<br>Community 2 |
| Github (Source) | Application Feed | Community | Community |
| Google BigQuery (Sink) | Analytics, Data Warehouse | Community | Community |

Source: http://www.confluent.io/product/connectors

trivadis
makes IT easier.

# Demo Case




trivadis
makes IT easier.

# Demo: Dataflow for MQTT to Kafka

```
#!/bin/bash
curl -X "POST" "http://192.168.69.138:8083/connectors" \
     -H "Content-Type: application/json" \
     -d $'{
"name": "mqtt-source",
"config": {
  "connector.class":
"com.datamountaineer.streamreactor.connect.mqtt.source.MqttSourceConnector",
  "connect.mqtt.connection.timeout": "1000",
  "tasks.max": "1",
  "connect.mqtt.kcql":
       "INSERT INTO truck_position SELECT * FROM truck/+/position",
  "name": "MqttSourceConnector",
  "connect.mqtt.service.quality": "0",
  "connect.mqtt.client.id": "tm-mqtt-connect-01",
  "connect.mqtt.converter.throw.on.error": "true",
  "connect.mqtt.hosts": "tcp://mosquitto-1:1883"
  }
}'
```

# Summary

trivadis
makes IT easier.

# Summary

## Apache NiFi 


- visual dataflow modelling
- very powerful – “with power comes responsibility”
- special package for Edge computing
- data lineage and data provenance
- supports for backpressure
- no transport mechanism (DEV/TST/PROD)
- custom processors
- supported by Hortonworks

## StreamSets 


- visual dataflow modelling
- very powerful – “with power comes responsibility”
- special package for Edge computing
- data lineage and data provenance
- no transport mechanism
- custom sources, sinks, processors
- supported by StreamSets

## Kafka Connect 


- declarative style data flows
- simplicity - “simple things done simple”
- very well integrated with Kafka – comes with Kafka
- Single Message Transforms (SMT)
- use Kafka Streams for complex data flows
- custom connectors
- supported by Confluent

trivadis
makes IT easier.

# Technology on its own won't help you.

## You need to know how to use it properly.


trivadis
makes IT easier.