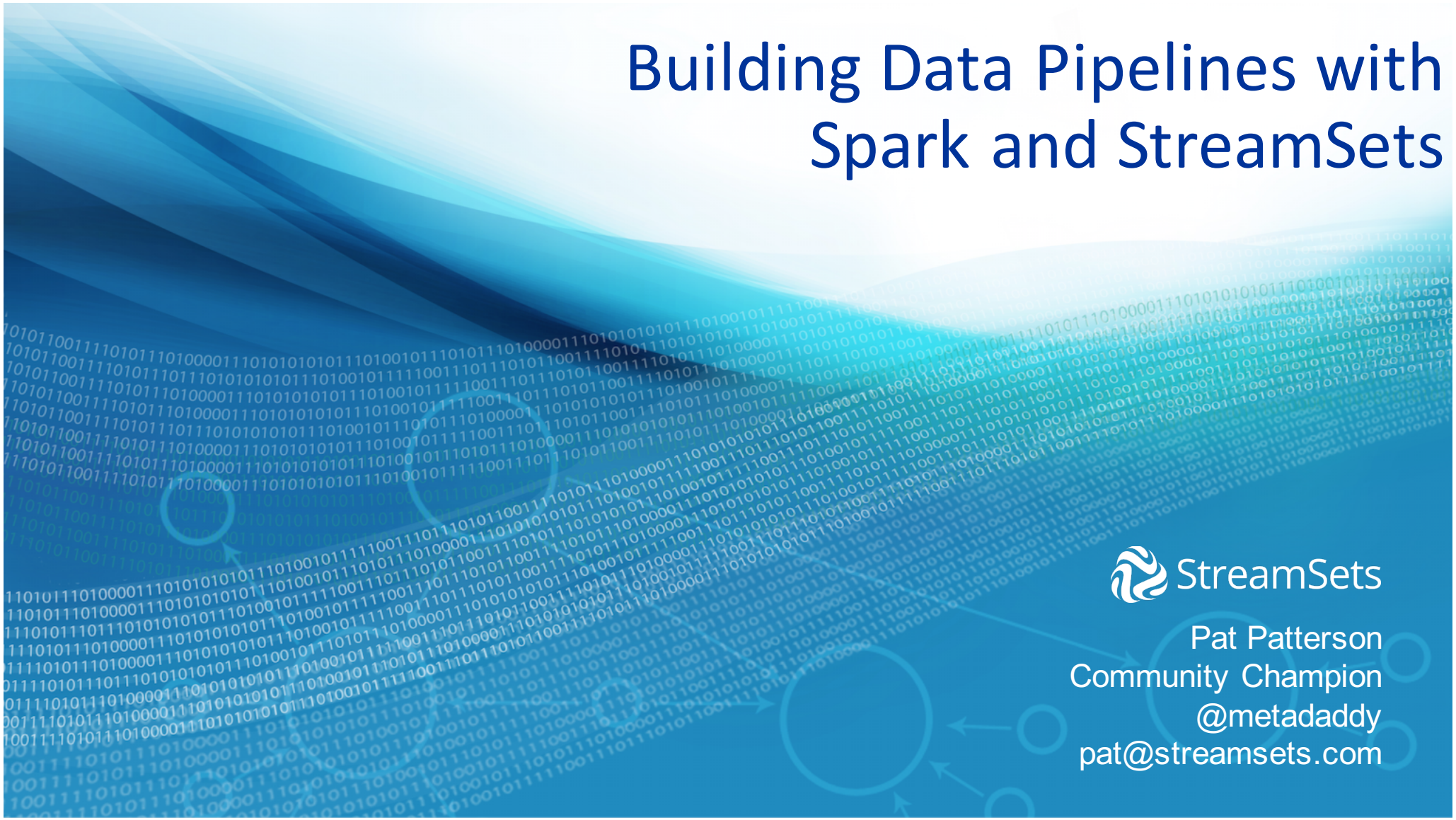
Building Data Pipelines with Spark and StreamSets



StreamSets

Pat Patterson
Community Champion
@metadaddy
pat@streamsets.com

Agenda

Data Drift

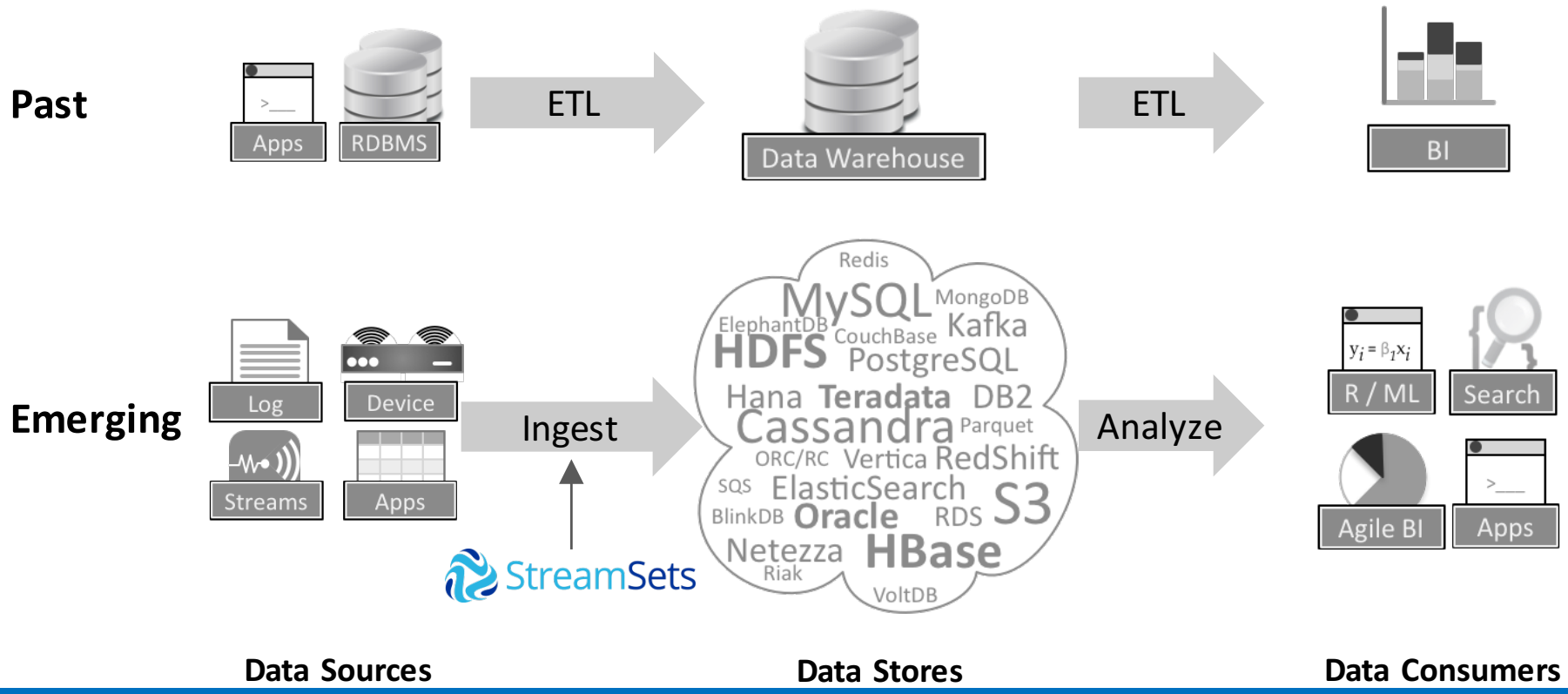
StreamSets Data Collector

Running Pipelines on Spark Today

Future Spark Integration

Demo

The Evolution of Data-in-Motion



Data Drift - a Data Engineering Headache

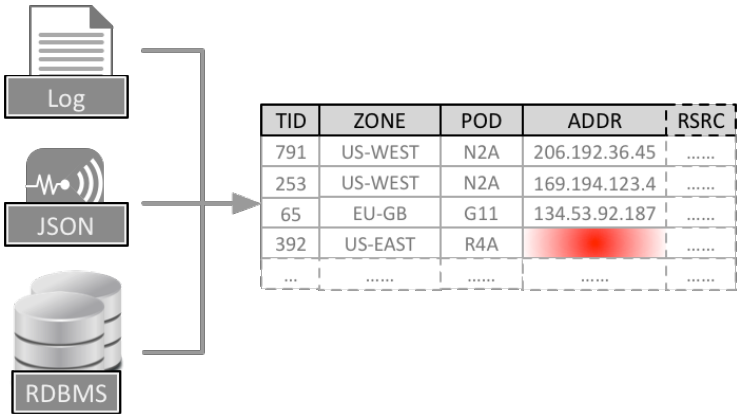
The unpredictable, unannounced and unending mutation of data characteristics caused by the operation, maintenance and modernization of the systems that produce the data

Structure Drift

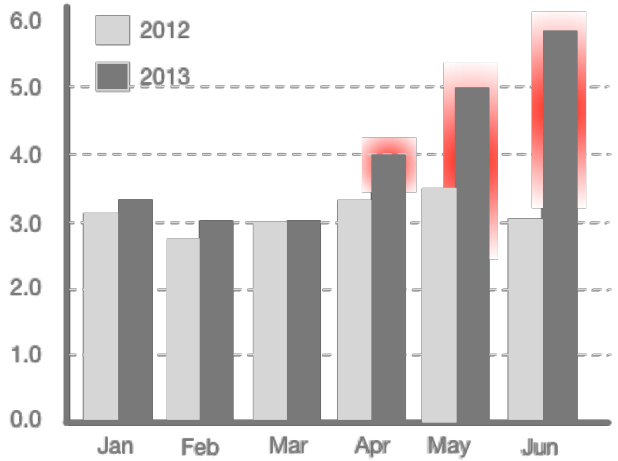
Semantic Drift

Infrastructure Drift

Example: Data Loss and Corrosion



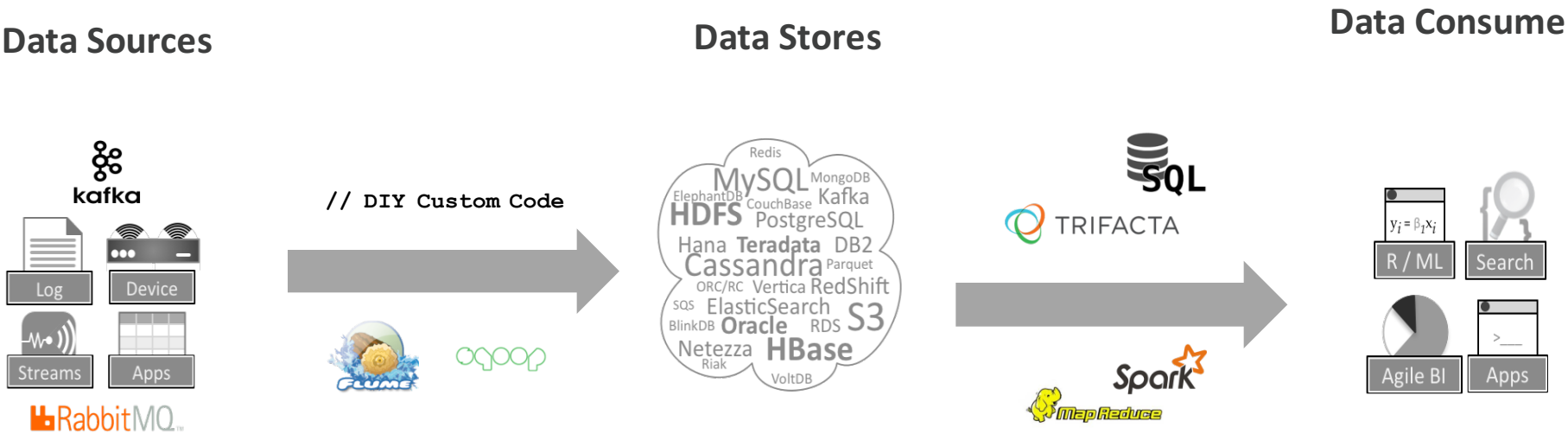
SQL on Hadoop (Hive)



Y/Y Click Through Rate

80% of analyst time is spent preparing and validating data,
while the remaining 20% is actual data analysis

Solving Data Drift



Data Drift

Custom code

Fixed-schema

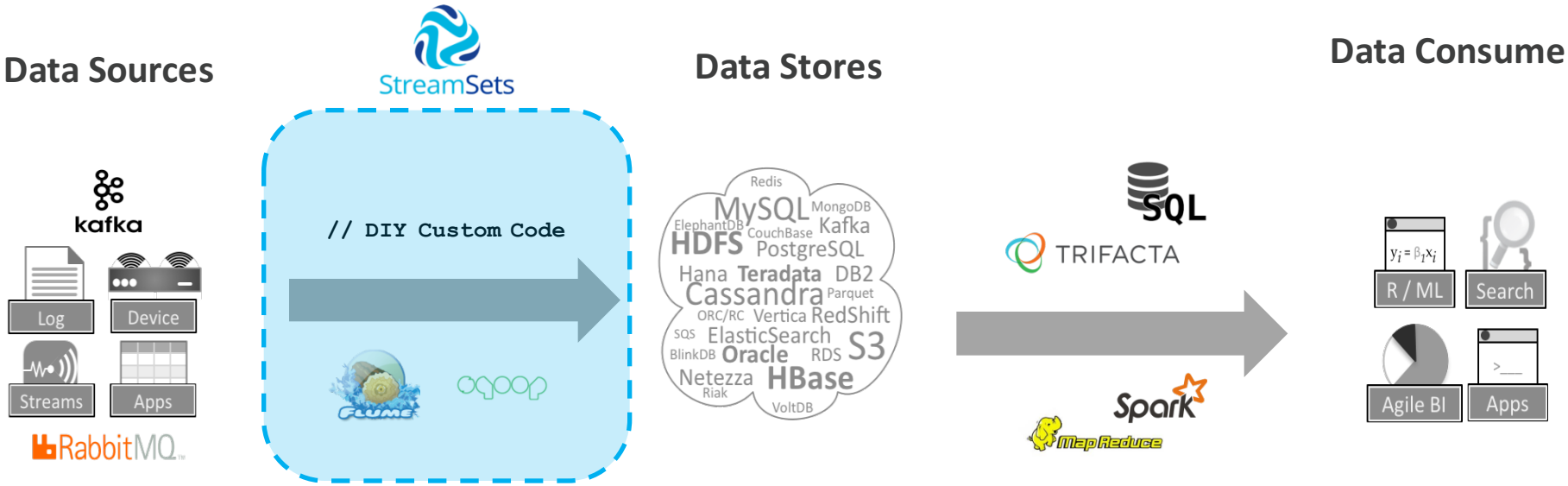
Poor Data Quality

Tools

Applications

Delayed and False Insights

Solving Data Drift



Data Drift

Intent-Driven

Drift-Handling

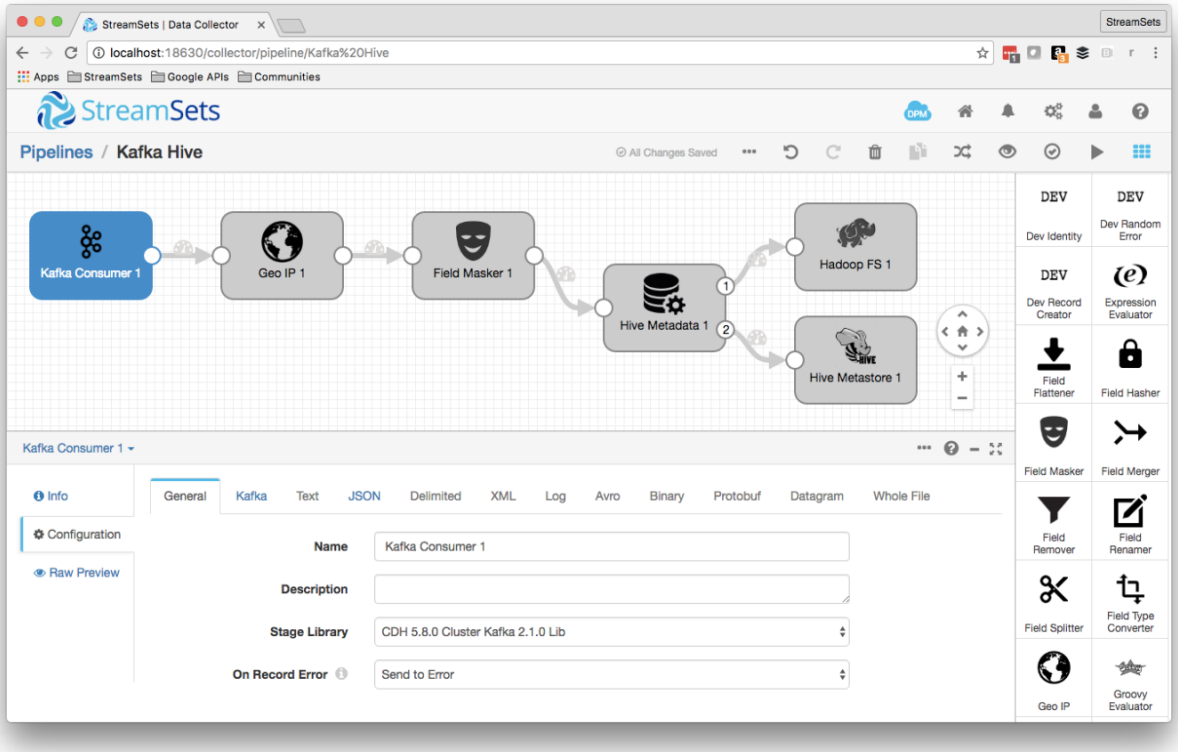
Data KPIs

Tools

Applications

Trusted Insights

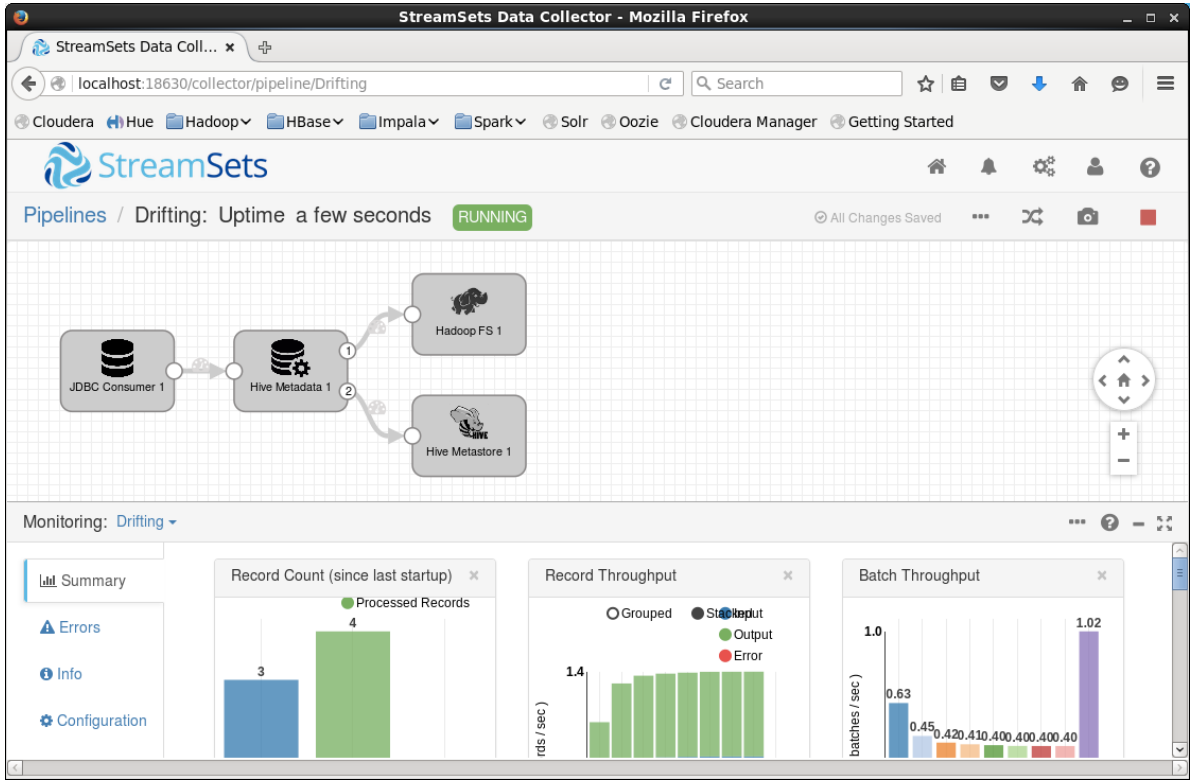
StreamSets Data Collector



Open source software for the rapid development and reliably operation of complex data flows.

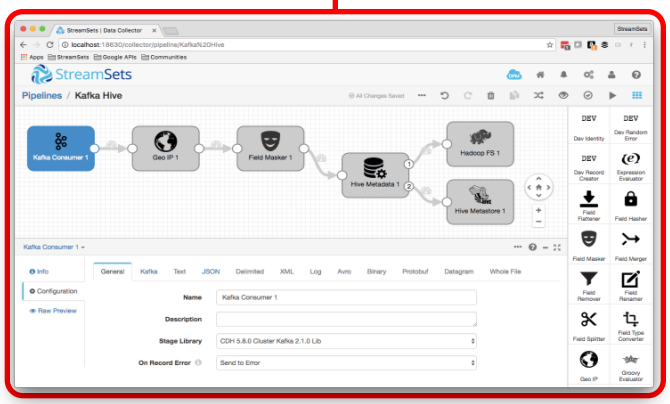
- Intent-driven
- UI Abstraction
- Extensible

Handling Drift with Hive



- Monitor data structure
- Detect schema change
- Alter Hive Metadata

Running Pipelines on Spark Today




```
spark-submit
--num-executors ...
--archives ...
--files ...
--jar ...
--class ...
```

- Container on Spark
- Leverage Kafka RDD
- Scale out for performance

SDC on Spark - Connectivity

Sources

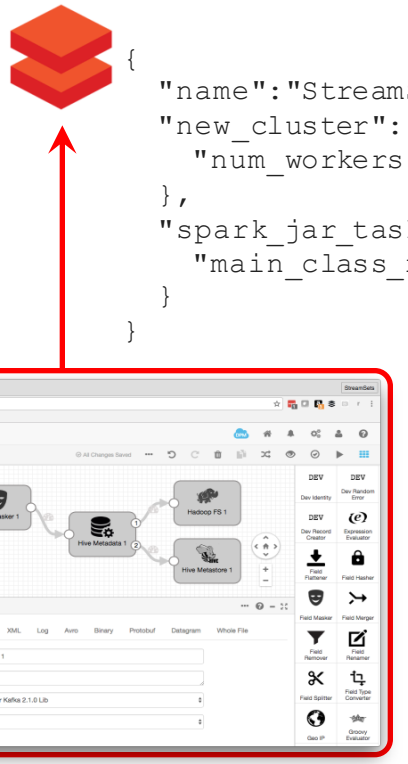
- Kafka

Destinations

- HDFS
- HBase
- S3
- Kudu
- MapR DB
- Cassandra
- ElasticSearch
- Kafka
- MapR Streams
- Kinesis
- etc, etc, etc!

Future Directions

Run Pipelines on Databricks



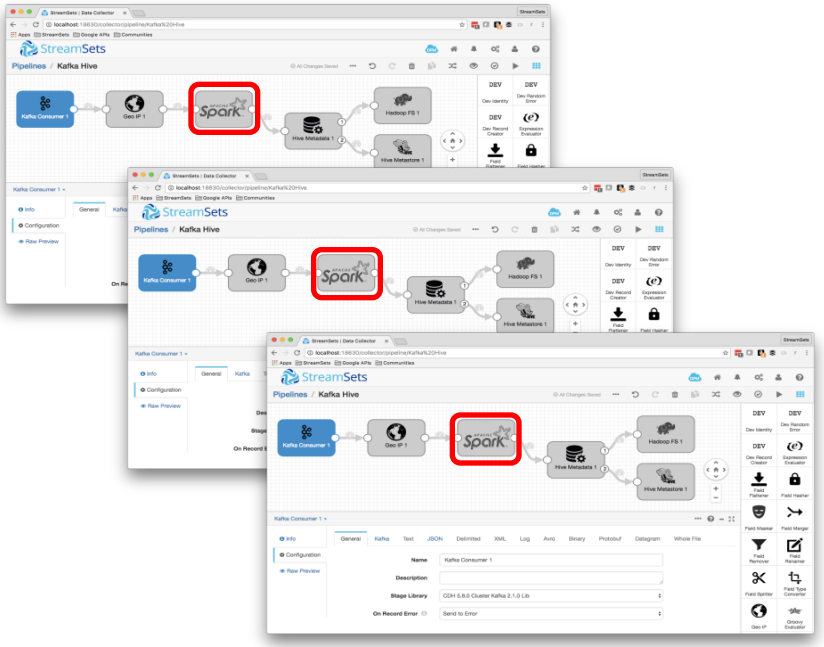
```
{
  "name":"StreamSets Data Collector",
  "new_cluster":{
    "num_workers":1
  },
  "spark_jar_task":{
    "main_class_name":"com.streamsets..."
  }
}
```

- Container on Databricks
- Leverage REST API
- Add S3 origin

Break Out Spark Processor

StreamSets

Apache Spark *Local Mode*



- Standalone containers, Spark processor
- Leverage Spark code
- Custom RDD
- Start local Spark job for each batch
- Example use cases: *running* image classification, sentiment analysis

Spark Processor - Connectivity

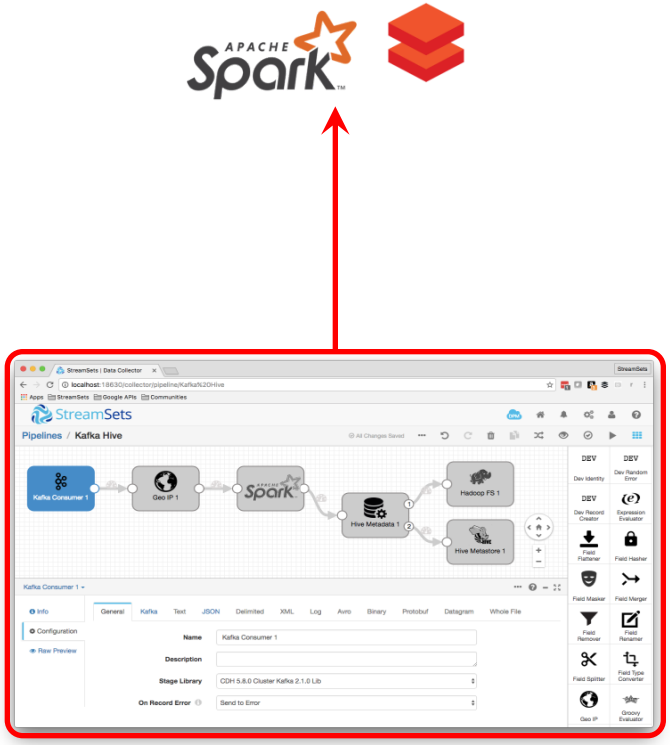
Sources

- Kafka
- S3
- MapR Streams
- JDBC
- MongoDB
- Local Filesystem
- Redis
- JMS
- HTTP
- UDP
- etc, etc, etc!

Destinations

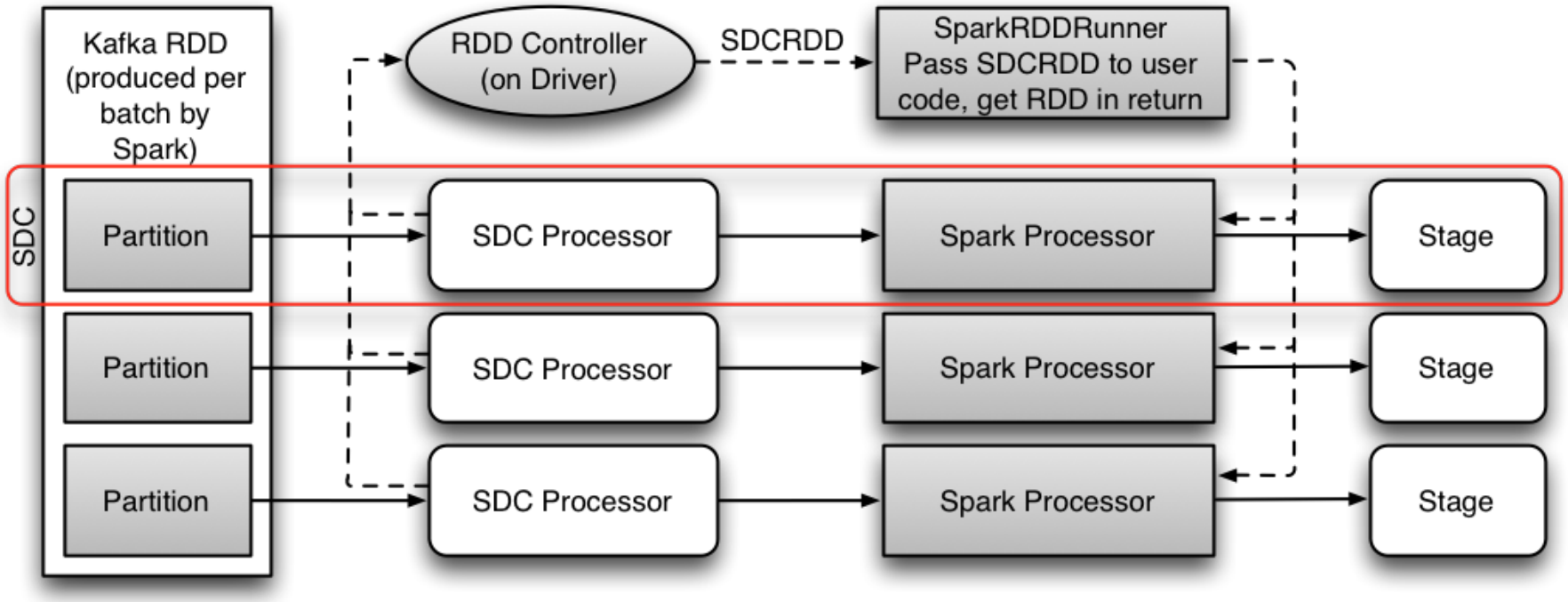
- HDFS
- HBase
- S3
- Kudu
- MapR DB
- Cassandra
- ElasticSearch
- Kafka
- MapR Streams
- JDBC
- etc, etc, etc!

Deepen Spark Integration



- Container on Spark, Spark processor
- Leverage Spark code
- Custom RDD
- Start Spark job 'on cluster' for each pipeline
- Example use cases: *training* image classification, sentiment analysis

Spark Integration Architecture



SDC on Spark - Connectivity Tomorrow

Sources

- Kafka
- *S3*
- *MapR Streams*
- *JDBC*
- *MongoDB*
- *Redis*
- *JMS*
- *HTTP*
- *UDP*
- ...any partitionable data source...

Destinations

- HDFS
- HBase
- S3
- Kudu
- MapR DB
- Cassandra
- ElasticSearch
- Kafka
- MapR Streams
- JDBC
- etc, etc, etc!

Demo

Conclusion

StreamSets Data Collector brings a UI abstraction to Spark

Standalone container + local Spark Processor bring wide connectivity to Spark code

Spark Container + Spark Processor allow iterative Spark code in pipelines

Resources

Download StreamSets Data Collector

https://streamsets.com/opensource

Contribute Code

https://github.com/streamsets/datacollector

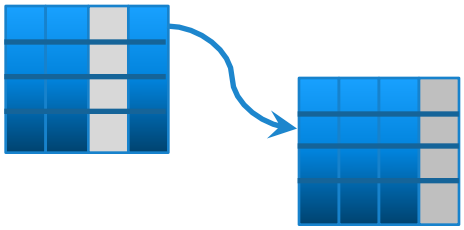
Get Involved

https://streamsets.com/community

Thank You!

Backup Slides

Structure Drift



107.3.137.195 → fe80::21b:21ff:fe83:90fa

```
{
 “first“: “jon”
 “last“: “smith”
 “email“: “jsmith@acme.com”
 “add1“: “123 Washington”
 “add2“: “”
 “city“: “Tucson”
 “state“: “AZ”
 “zip“: “85756”
}
```

→

```
{
 “first“: “jane”
 “last“: “smith”
 “email“: “jane@earth.net”
 “add1“: “456 Fillmore”
 “add2“: “Apt 120”
 “city“: “Fairfield”
 “state“: “VA”
 “zip“: “24435-1001”
 “phone”: “401-555-1212”
}
```

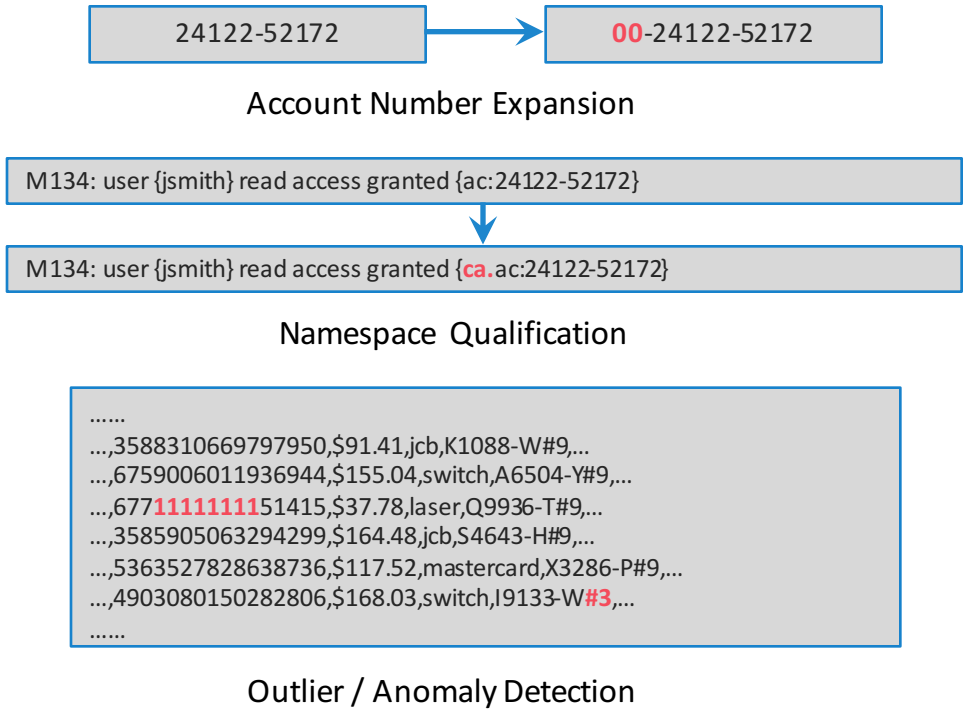
Data Structure Evolution

Structure Drift

Data structures and formats evolve and change unexpectedly

Implication:
Data Loss
Data Squandering

Semantic Drift

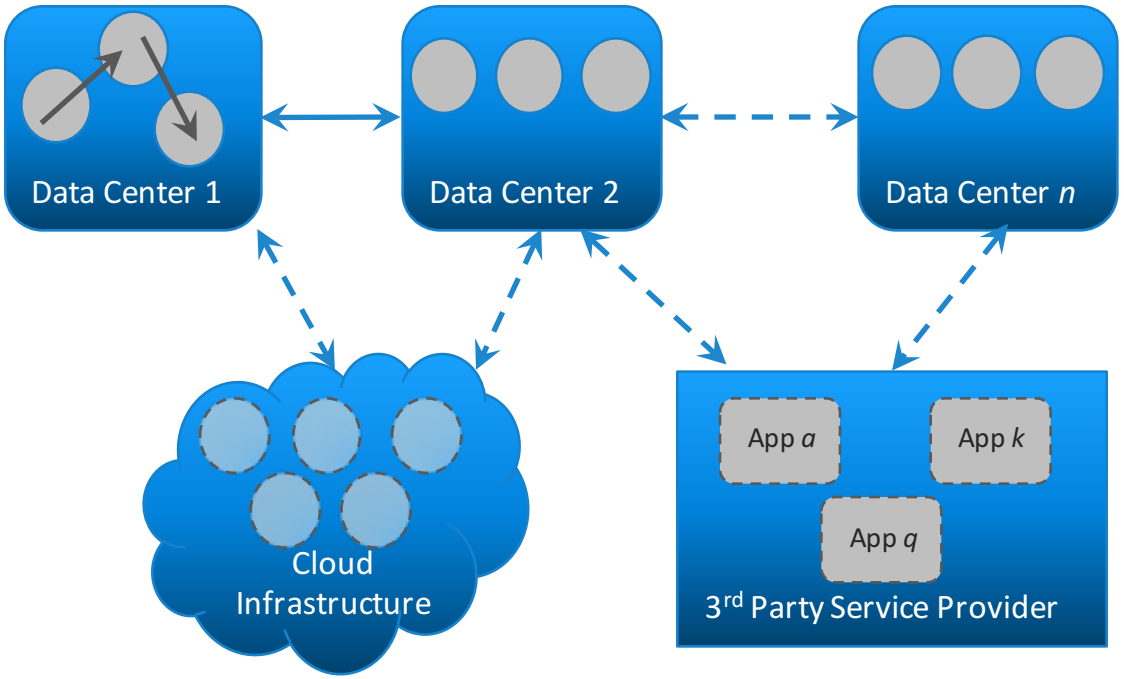


Semantic Drift

Data semantics change with evolving applications

Implication:
Data Corrosion
Data Loss

Infrastructure Drift



Infrastructure Drift

Physical and Logical Infrastructure changes rapidly

Implication:
Poor Agility
Operational Downtime