

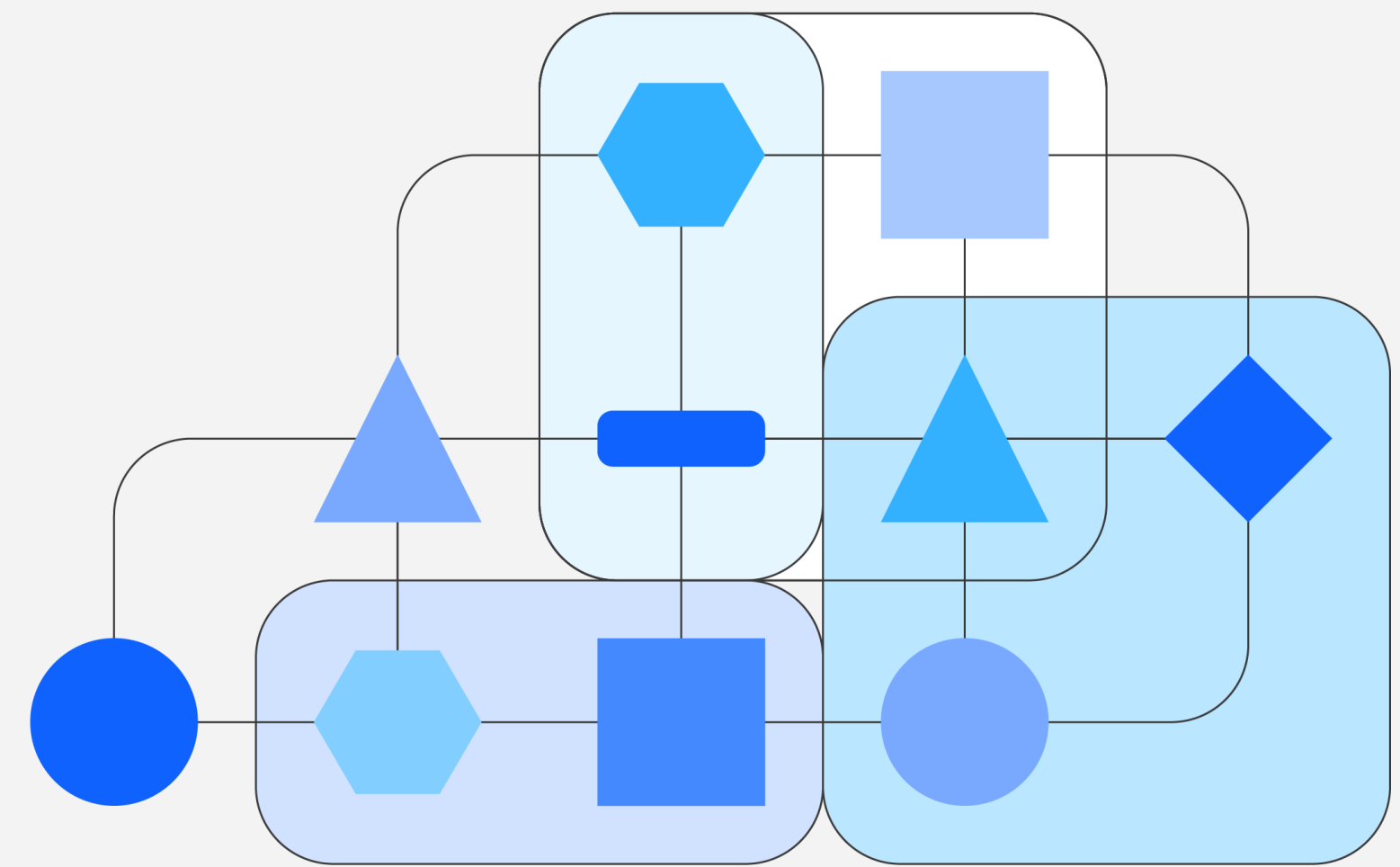
Data Quality in Streaming Data Pipelines

Eric Greisdorf

Eric.Greisdorf@ibm.com

Subject Matter Expert: StreamSets

IBM StreamSets

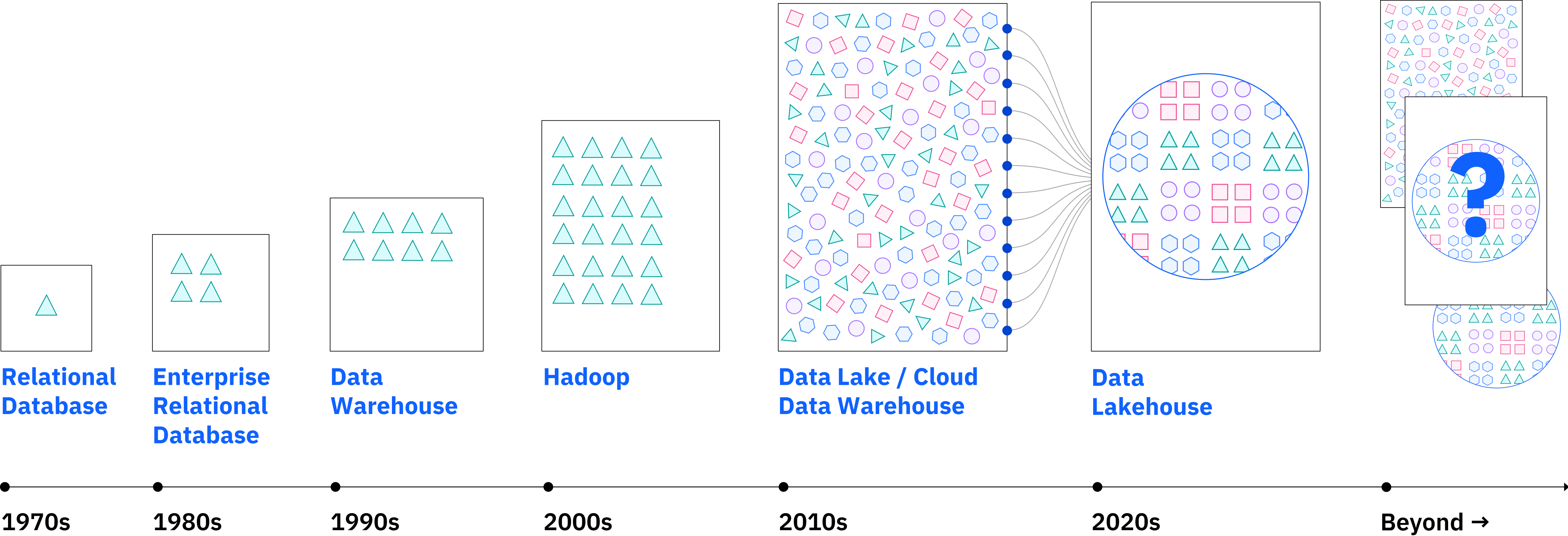


Agenda



- ❖ The Data and AI Continuum
- ❖ AI/ML Use Case Examples
- ❖ Streaming Data Defined
- ❖ Monitoring and Quality
- ❖ Wrap-Up / Next Steps

Relentless paradigm shifts in data storage have constantly disrupted data engineering—*and it will happen again*



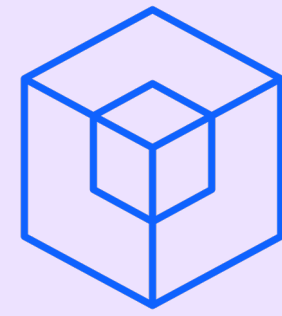
The continuum of AI Agents

Fixed Flow – Human Led



Chatbots

- Rules based
- Rigid
- Preprogrammed



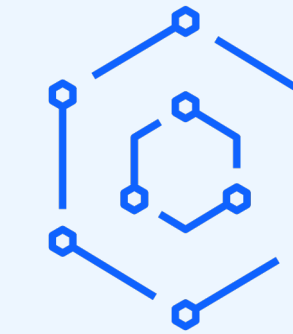
Virtual Assistants

- NLU powered
- Intent recognition
- ML and DL techniques
- Human created responses



AI Assistants + Automation

- Conversational AI: LLM powered intent recognition
- Knowledge grounded Q&A (RAG)
- Rules based Workflows (Automation)
- Connect to Enterprise Apps
- Call Gen AI tools (summarization, entity extraction, content generation)



AI Agents

- AI Orchestrator Agent (LLM) can reason, plan, and execute on a given task or problem
- Connected to multiple agents, assistants, data, tools and applications
- Understands Complex multi-threaded problems
- Autonomous action taking, self-correcting and self-reasoning
- Conversational or Non-Conversational

According to the latest estimates, **402.74 million terabytes** of data are created each day. Videos account for over half of internet data traffic.



Source: Exploding Topics <https://explodingtopics.com/blog/data-generated-per-day#how-much>

Unstructured Video and Images Streaming: AI/ML Example

```

recated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with amp.autocast(enabled=True):
Label: person - 0.83, Coordinates: (536.382080078125, 127.0810546875
, 678.4584350585938, 394.7198181152344)
Label: handbag - 0.43, Coordinates: (617.0795288085938, 254.29838562
01172, 667.2301025390625, 317.9366455078125)
Label: laptop - 0.27, Coordinates: (2.7062361240386963, 88.608390808
10547, 212.57131958007812, 478.2134704589844)
Label: handbag - 0.26, Coordinates: (536.6428833007812, 161.07661437
98828, 601.4154052734375, 241.31484985351562)
INFO: 127.0.0.1:51033 - "POST /detect/ HTTP/1.1" 200 OK
/Users/jmlegon/.cache/torch/hub/ultralytics_yolov5_master/models/com
mon.py:894: FutureWarning: `torch.cuda.amp.autocast(args...)` is dep
recated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with amp.autocast(enabled=True):
Label: person - 0.85, Coordinates: (547.2600708007812, 118.045333862
30469, 699.2008666992188, 387.5803527832031)
INFO: 127.0.0.1:51033 - "POST /detect/ HTTP/1.1" 200 OK
/Users/jmlegon/.cache/torch/hub/ultralytics_yolov5_master/models/com
mon.py:894: FutureWarning: `torch.cuda.amp.autocast(args...)` is dep
recated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with amp.autocast(enabled=True):
Label: person - 0.89, Coordinates: (545.1817016601562, 115.182975769
04297, 699.2396850585938, 388.80072021484375)
INFO: 127.0.0.1:51033 - "POST /detect/ HTTP/1.1" 200 OK

```

```

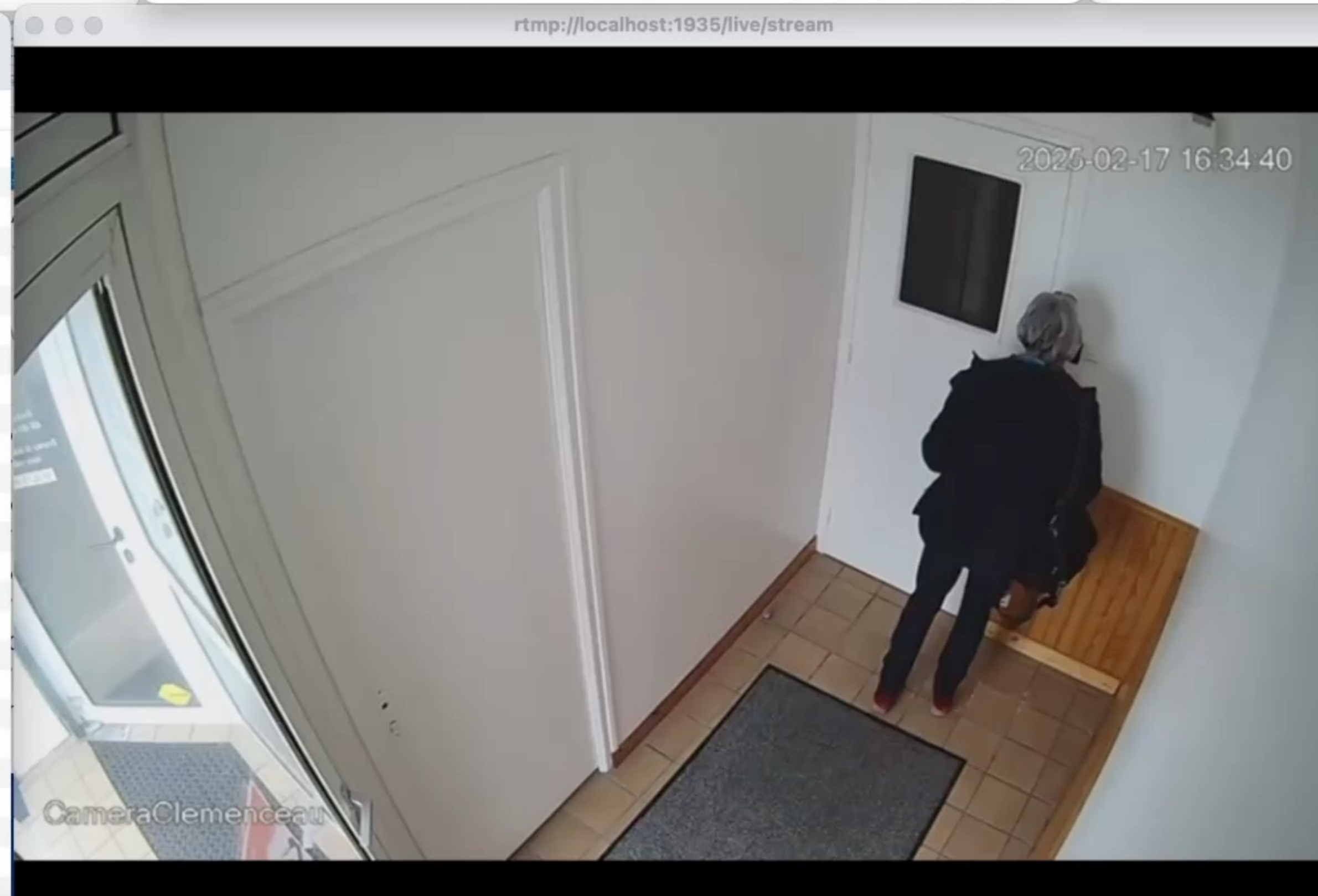
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
frame= 17 fps=0.0 q=10.3 size=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=368 speed=0.00
^C[in#0/flv @ 0x6000037fc300] Error during demuxing: Immediate exit requested
[out#0/image2 @ 0x600003ef8000] Video: 691KiB audio: 0KiB subtitle: 0KiB other streams: 0KiB g
lobal headers: 0KiB muxing overhead: unknown
frame= 17 fps=0.0 q=10.3 Lsize=N/A time=00:00:17.00 bitrate=N/A dup=0 drop=380 speed=0.0
0645x
Exiting normally, received signal 2.
(base) jmlegon@MacBook-Pro-de-jmlegon videos-to-files %

```

```

frame= 177 fps= 27 q=-1.0 size= 1257KiB time=00:00:07.08 bitrate=1454.0kbits/s
frame= 189 fps= 27 q=-1.0 size= 1318KiB time=00:00:07.56 bitrate=1427.7kbits/s
frame= 202 fps= 27 q=-1.0 size= 1426KiB time=00:00:08.08 bitrate=1446.2kbits/s
frame= 214 fps= 27 q=-1.0 size= 1486KiB time=00:00:08.56 bitrate=1422.6kbits/s
frame= 227 fps= 26 q=-1.0 size= 1608KiB time=00:00:09.08 bitrate=1451.2kbits/s
frame= 240 fps= 26 q=-1.0 size= 1683KiB time=00:00:09.60 bitrate=1436.1kbits/s
frame= 252 fps= 26 q=-1.0 size= 1791KiB time=00:00:10.08 bitrate=1455.8kbits/s
frame= 265 fps= 26 q=-1.0 size= 1857KiB time=00:00:10.60 bitrate=1435.1kbits/s
frame= 278 fps= 26 q=-1.0 size= 1974KiB time=00:00:11.12 bitrate=1454.1kbits/s
frame= 290 fps= 26 q=-1.0 size= 2042KiB time=00:00:11.60 bitrate=1442.3kbits/s
frame= 303 fps= 26 q=-1.0 size= 2160KiB time=00:00:12.12 bitrate=1460.2kbits/s
frame= 315 fps= 26 q=-1.0 size= 2224KiB time=00:00:12.60 bitrate=1445.8kbits/s
frame= 328 fps= 26 q=-1.0 size= 2339KiB time=00:00:13.12 bitrate=1460.5kbits/s
frame= 341 fps= 26 q=-1.0 size= 2400KiB time=00:00:13.64 bitrate=1441.3kbits/s
frame= 353 fps= 26 q=-1.0 size= 2482KiB time=00:00:14.12 bitrate=1439.8kbits/s
frame= 366 fps= 26 q=-1.0 size= 2510KiB time=00:00:14.64 bitrate=1404.7kbits/s
frame= 378 fps= 26 q=-1.0 size= 2584KiB time=00:00:15.12 bitrate=1399.9kbits/s
frame= 391 fps= 26 q=-1.0 size= 2618KiB time=00:00:15.64 bitrate=1371.5kbits/s
[flv @ 0x127605f10] Failed to update header with correct duration.
[flv @ 0x127605f10] Failed to update header with correct filesize.
[out#0/flv @ 0x6000014103c0] video:2627KiB audio:0KiB subtitle:0KiB other streams:
0KiB global headers:0KiB muxing overhead: 0.307499%
frame= 396 fps= 26 q=-1.0 Lsize= 2635KiB time=00:00:15.84 bitrate=1363.0kbits/
s speed=1.03x
(base) jmlegon@MacBook-Pro-de-jmlegon demo-drone %

```



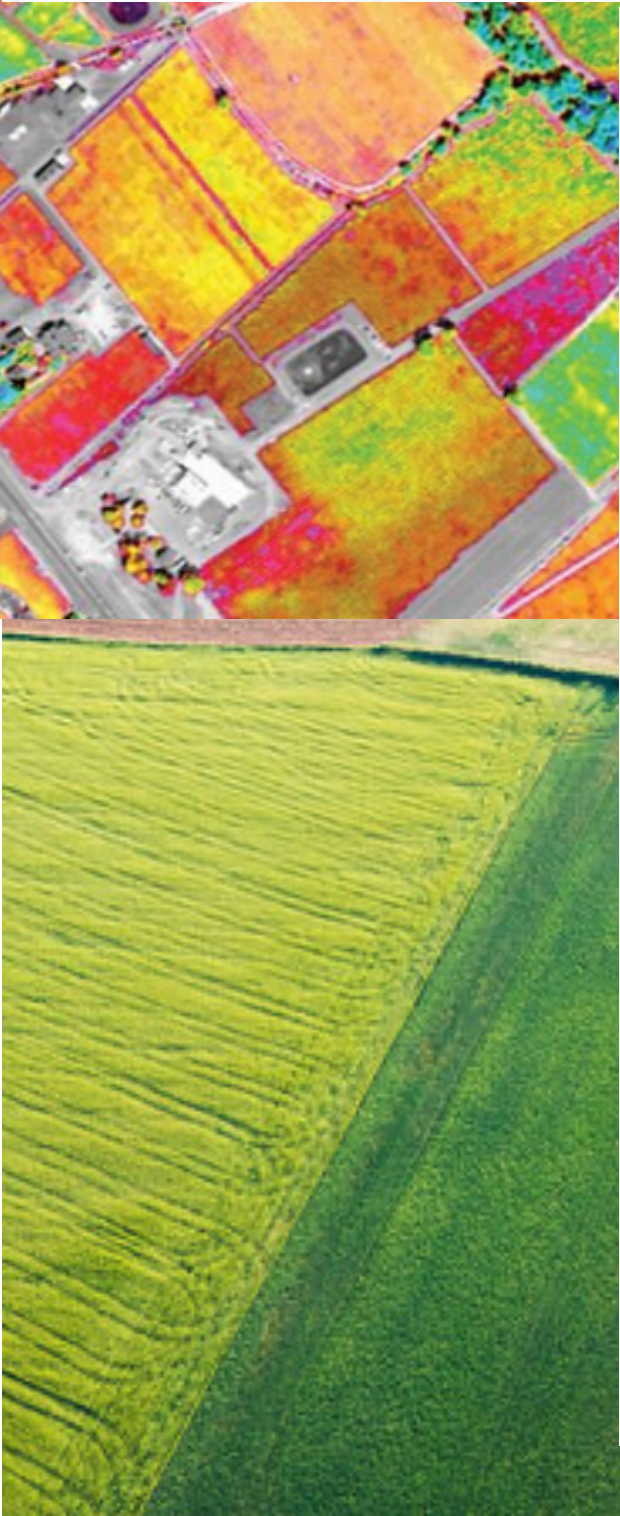
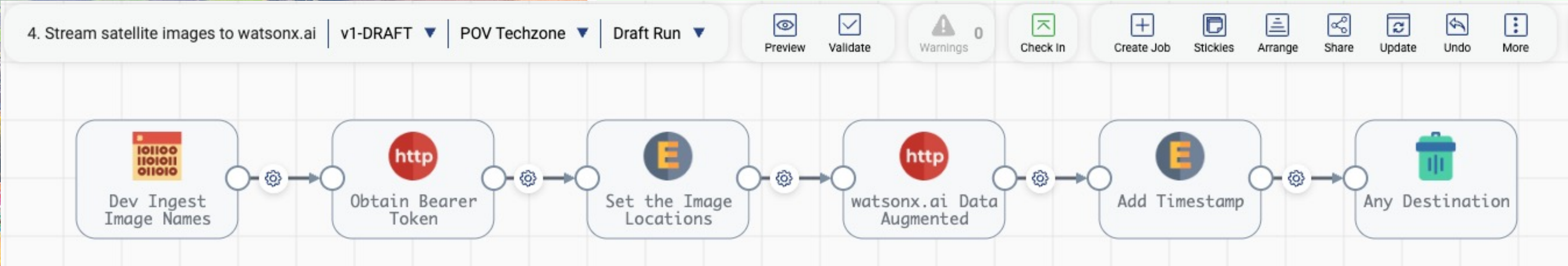
```
jmlongon — ffmpeg rtmp://localhost:1935/live/strea...
254953.46 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.49 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.52 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.56 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.59 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.62 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.65 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.69 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.72 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.75 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.79 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.82 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.85 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.88 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.92 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.95 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254953.98 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.02 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.05 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.08 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.11 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.14 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.18 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.21 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.24 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.27 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.31 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.34 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.37 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.41 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.44 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.47 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.50 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.54 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.57 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.60 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.64 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.67 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.70 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.74 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
254954.77 A-V:251754.943 fd= 170 aq=      0KB vq=      0KB sq=
```


Retrieval Augmented AI Prompts and Responses



generated_text : {STRING} "

The satellite image shows a diverse range of agricultural fields with varying crop types and health. Here's a descriptive analysis of the image:



Request Data ⓘ

```
1 {
2   "input": "describe the crops and health of the Agricultural fields in the satellite
3   image ${record:value('/url')}",
4   "parameters": {
5     "decoding_method": "greedy",
6     "max_new_tokens": 300,
7     "min_new_tokens": 0,
8     "stop_sequences": [],
9     "repetition_penalty": 1
10  },
11  "model_id": "meta-llama/llama-3-2-90b-vision-instruct",
12  "project_id": "272efb50-8524-4c28-85e7-0a26269db220"
13 }
```

plants still in the ve

brown color indica

color and irregular

lush green vegetati

show signs of stres
ht, nutrient deficien

3. **Waterlogged Fields**: A few

While data integration remains a priority for organizations, challenges remain

67% of data leaders believe prioritizing integrating data from various sources is a key priority¹

Data access

50% of data analyst time is spent working reactively or trying to find or get access to data²

Data drift

80% of data engineering time is spent on maintaining & fixing existing pipelines due to data drift, creating a costly maintenance burden³

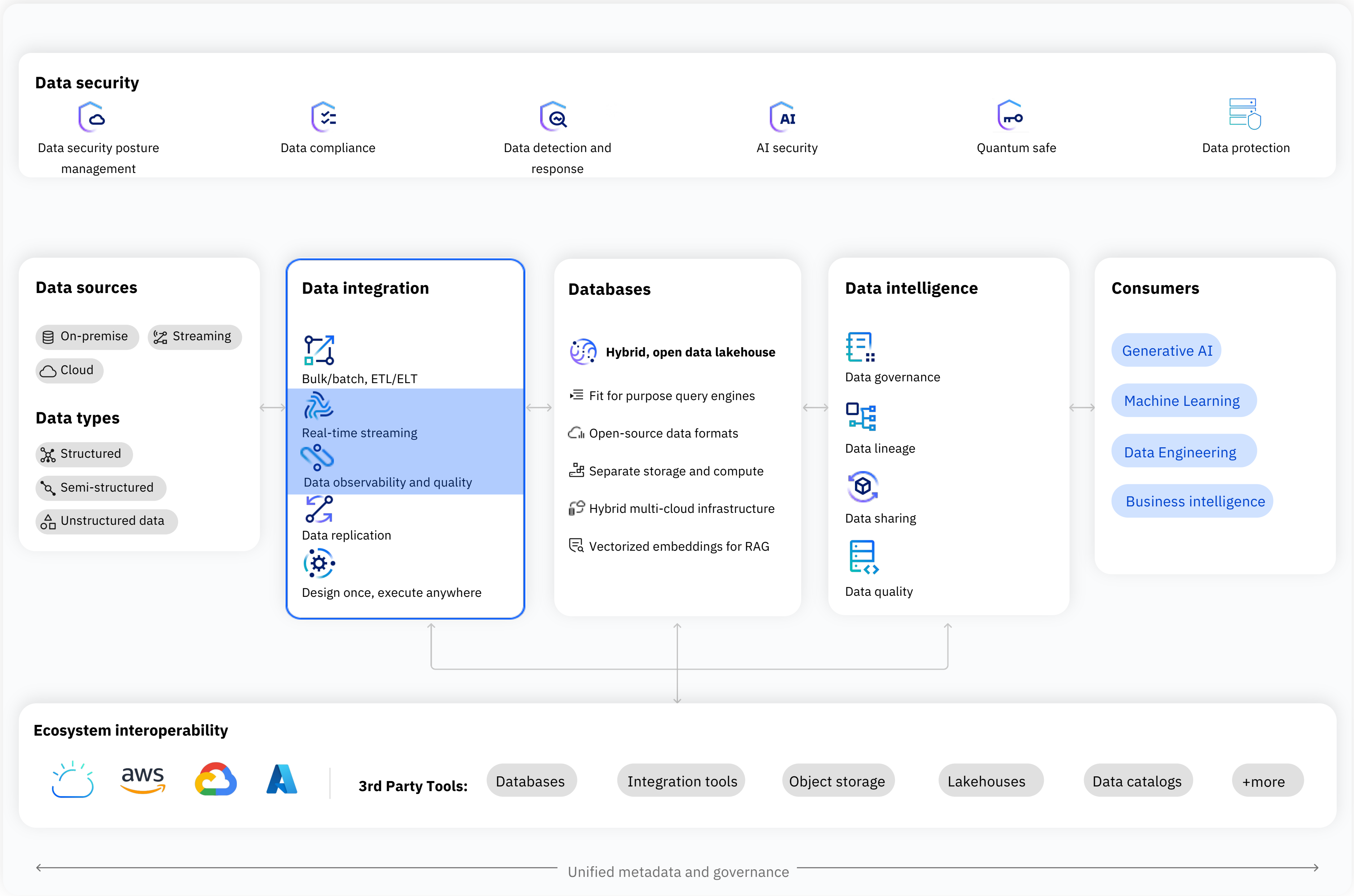
Data proliferation

With growing data volume and datasets organizations require scalability and faster response times

Data latency

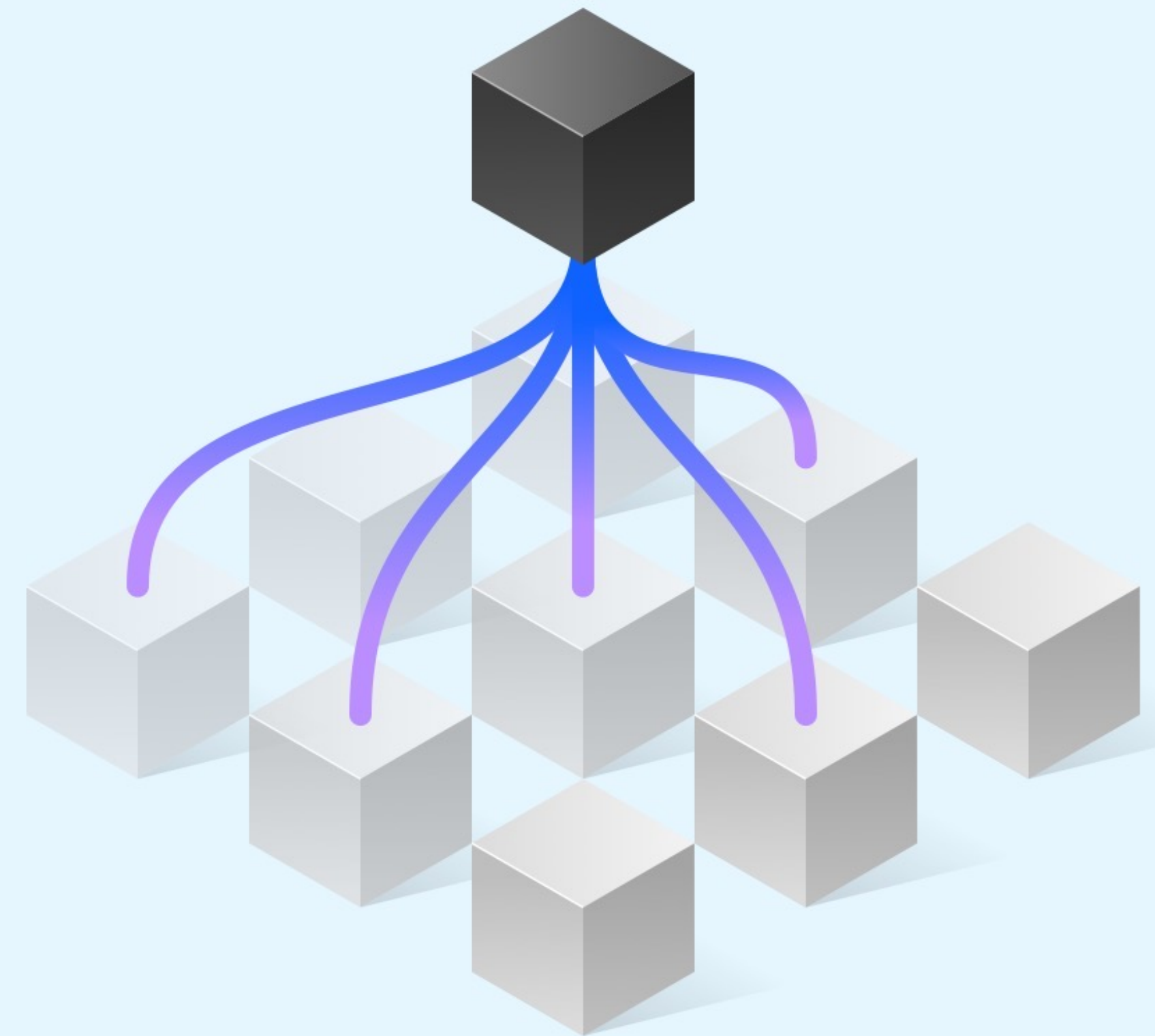
Need for processing data as it arrives to reduce delays in business decision and operations

Integrate,
access, govern,
and secure **all**
data types with
an open and
hybrid **data**
architecture



What is Real-Time Data Integration?

IBM defines real-time data integration as the **ability to ingest, process, and write data as soon as it's available** instead of on an intermittent or scheduled basis.

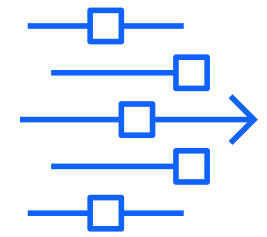


IBM StreamSets

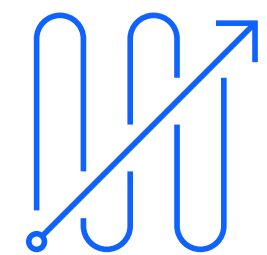
Benefits



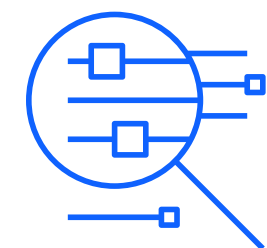
Real-time data integration solution for building streaming data pipelines to enhance real-time decision-making and mitigate risks



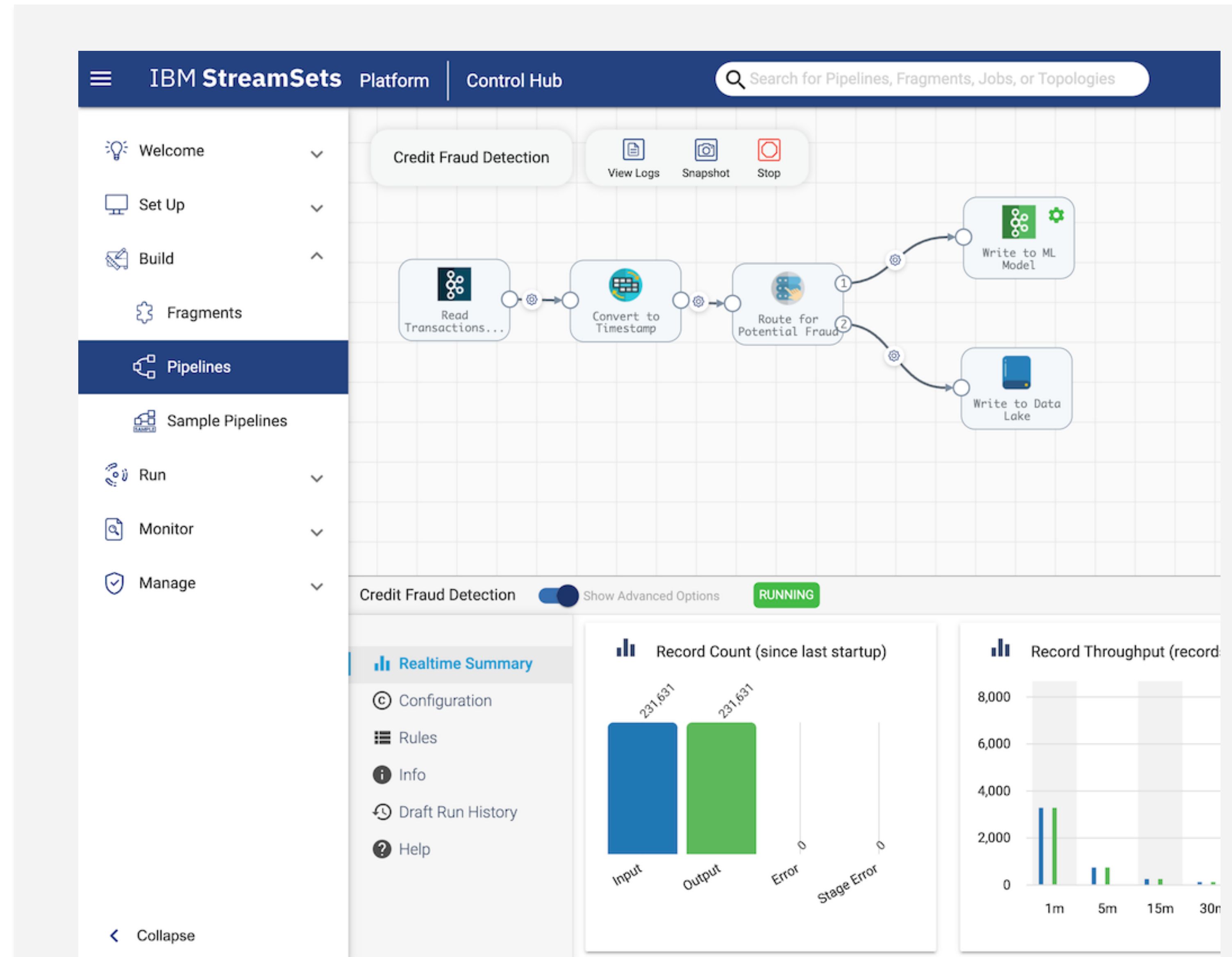
Enable real-time data ingestion at scale



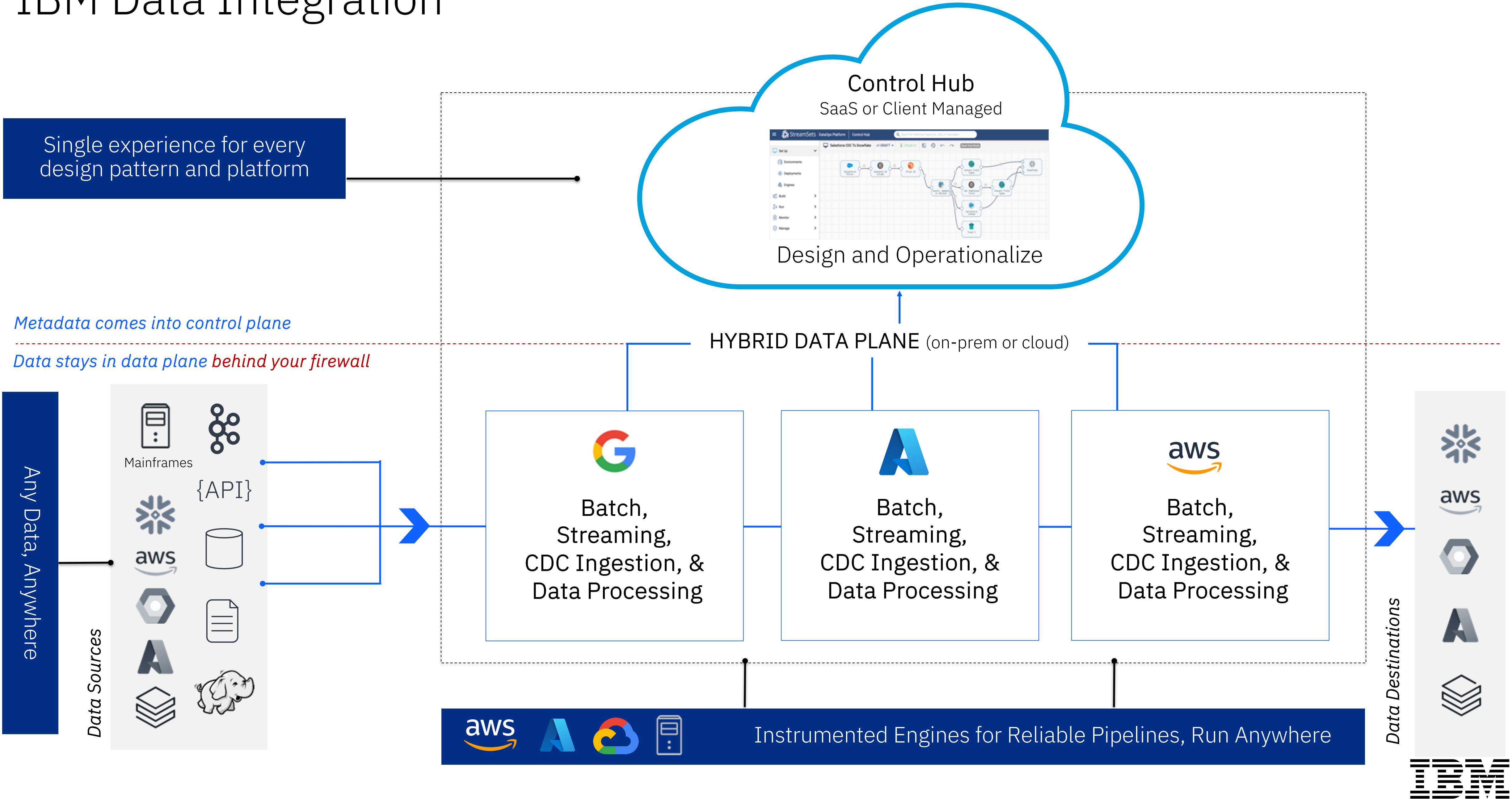
Reduce data drift with intelligent streaming data pipelines



Stream any data from any source

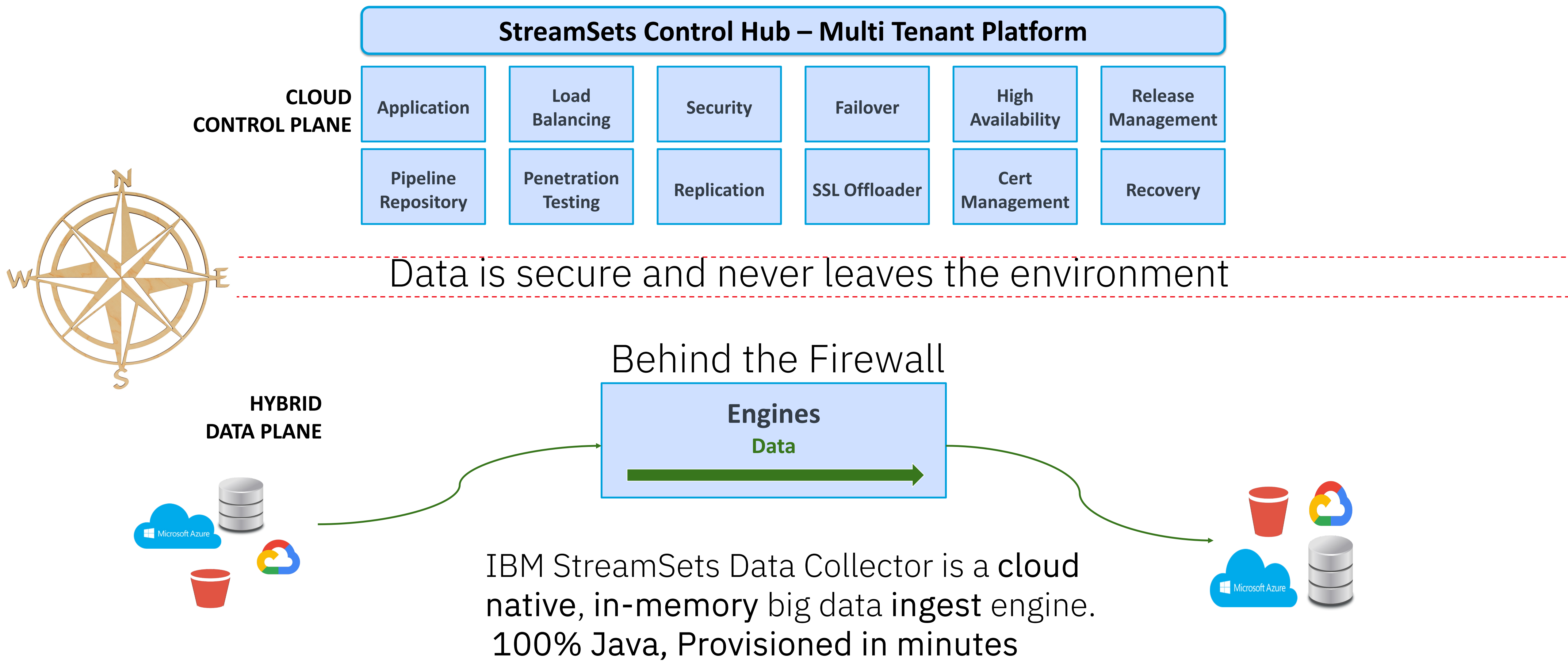


IBM Data Integration



IBM StreamSets Engines

SaaS or Cartridge



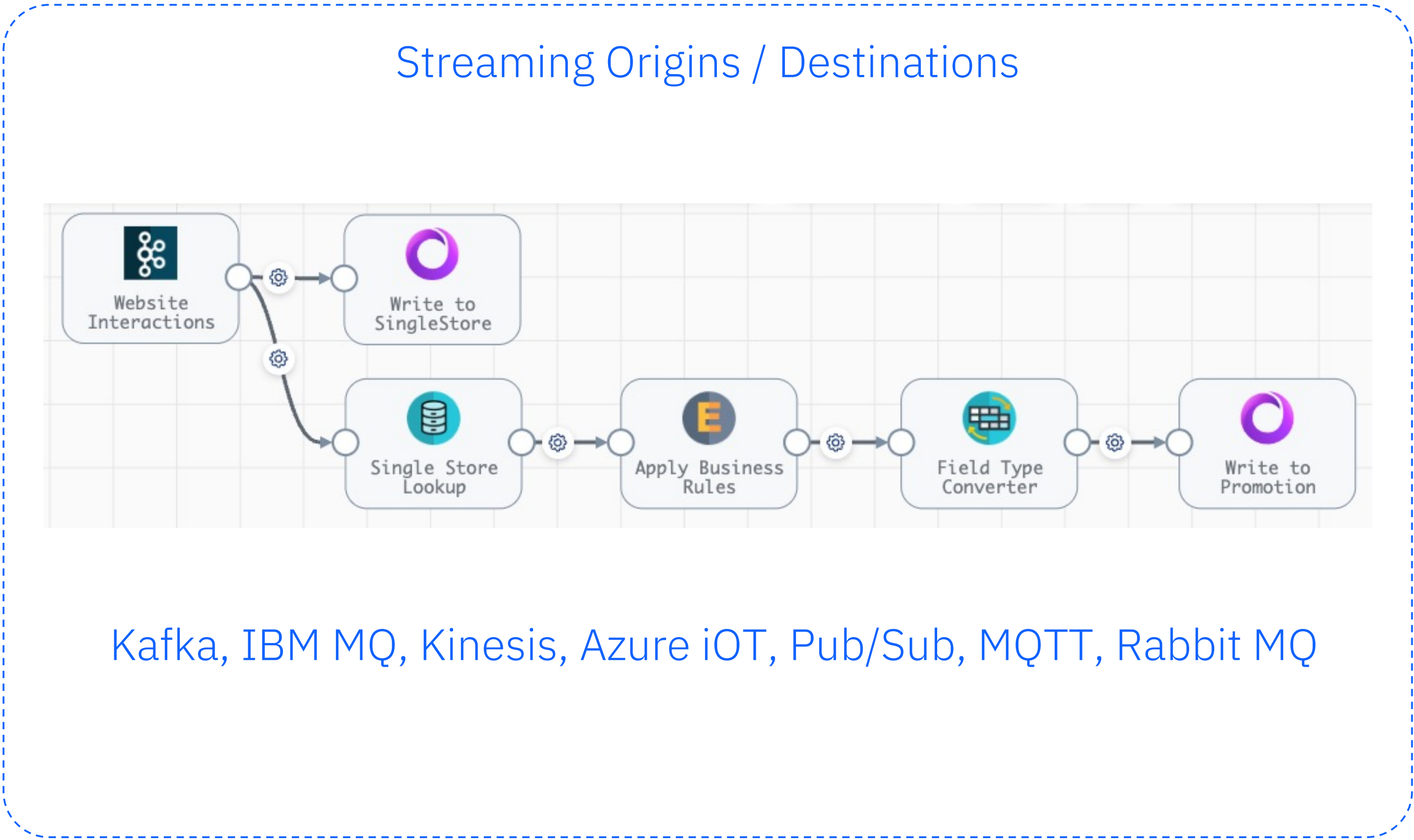
Streaming Ingest for Flash Promotions and Personalization

Data sources

- On-premise
- Cloud
- Streaming

Data types

- Structured
- Semi-structured
- Unstructured data

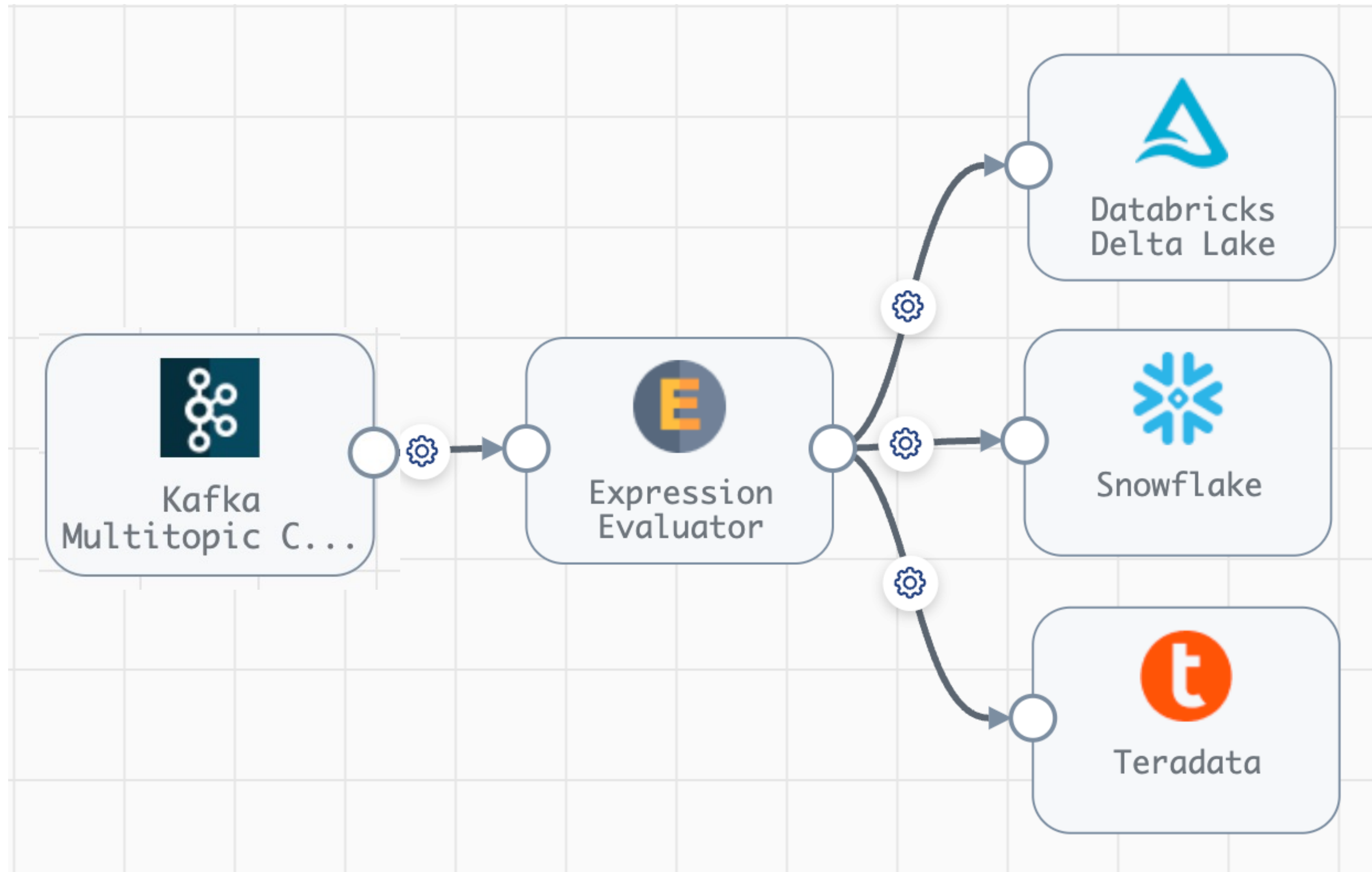


Includes:

- Stream critical data and events in real-time
- Leverage broker capabilities
- Highly performant and resilient
- Real-time Alerting

Ecosystem interoperability | Snowflake, Databricks, Teradata

IBM StreamSets Multi-Destination – Structured Streaming



Propagate changes
to multiple destinations

OFFSET / SCN tracking

At least once or at
most once delivery

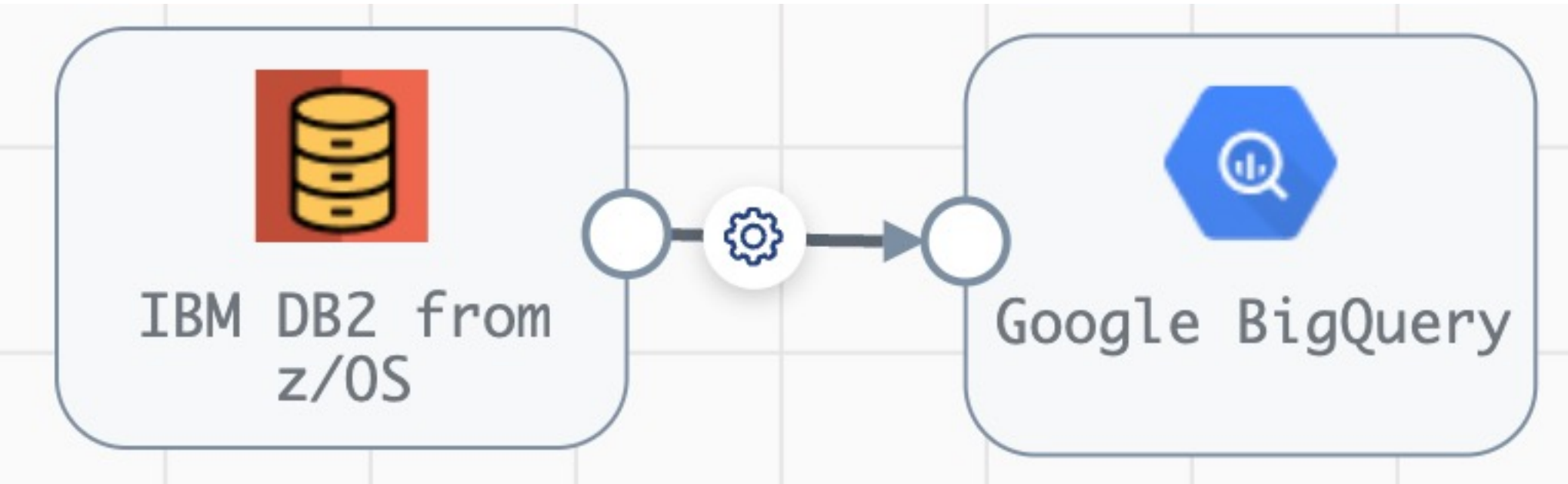
Automatic
resume / recovery

Table auto-create

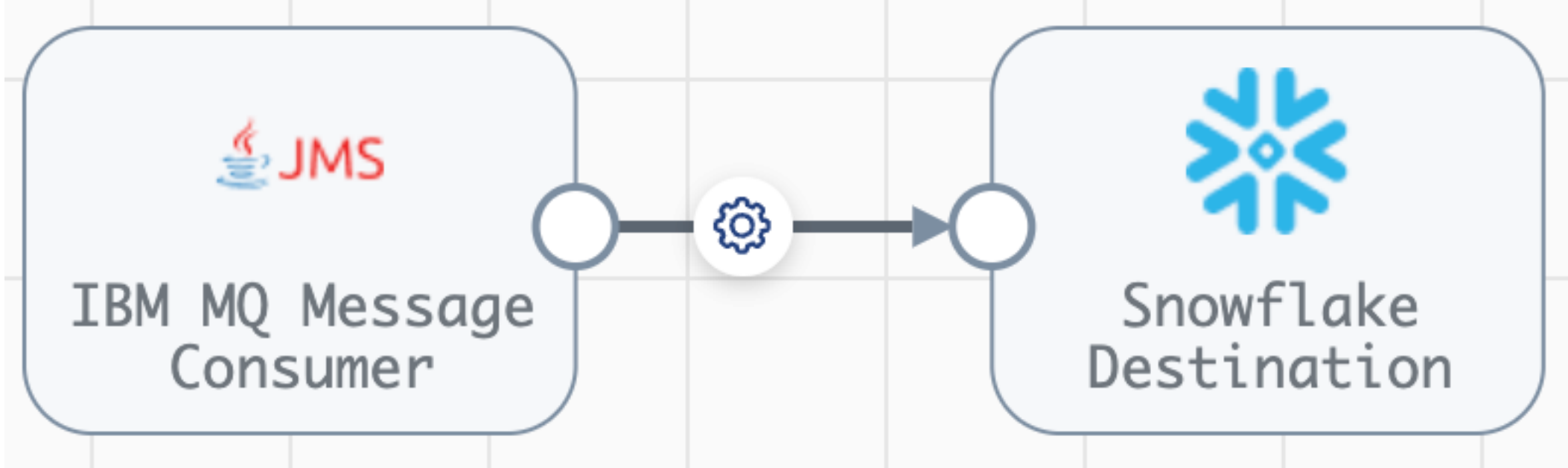
Schema evolution

StreamSets Structured and Semi-Structured

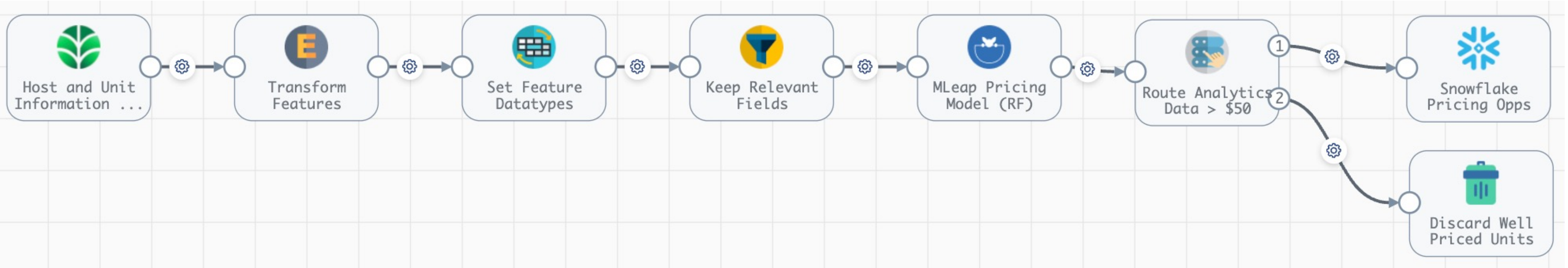
Streaming from DB2 via offsets



Streaming from IBM MQ



Streaming MongoDB to AI/ML Managed Snowflake Ingest



What about data quality?

IBM StreamSets Example Security Features



Log In

Email Address*

eric.greisdorf@ibm.com

Password*


.....


Log In

[Forgot Password?](#)

OR LOG IN WITH SSO

 Sign in with IBMid

 Continue with Google

 Continue with Microsoft

 Sign in with SSO SAML

Need an account? [Sign up](#)

Secure By Design

Strict separation of North/South (management) and East/West (data) planes

Integration with identity providers and SSO

Integration with credential stores for connections

Integration with AD groups for Role Based Access Controls

Engines heartbeat fully encrypted outbound calls only



Origins/Sources

Data sources

On-premise

Cloud

Streaming


Data types

Structured

Semi-structured

Unstructured data

70+ Sources/Origins



Aurora PostgreSQL CDC Client

Processes Write-Ahead Logging (WAL) data to generate change data capture records for

+



CONNX

Reads data from CONNX Server.

+



Amazon S3

Reads files from Amazon S3

-



Amazon SQS Consumer

Reads messages from Amazon SQS

-



CONNX CDC

Reads change data from CONNX Server.

+



Couchbase

Reads data from Couchbase

+



Azure Blob Storage

Reads data from Azure Blob Storage

-



Azure Data Lake Storage Gen2

Reads data from Azure Data Lake Storage Gen2

-



Dev Synthetic Data Generator

Generates records with synthetic data in the specified field names based on their defined

+



Elasticsearch

Read data from an Elasticsearch cluster

+



Azure Data Lake Storage Gen2 (Legacy)

Reads data from Azure Data Lake Storage

-



Azure IoT/Event Hub Consumer

Reads data from Azure Event Hub

-



IBM Db2

Reads data from IBM Db2 using Flight Service

+



Jira

Jira Origin

+



CoAP Server

Listens for requests on a CoAP endpoint

-



Cron Scheduler

Generates a record with the current datetime based on a cron expression

-



JMS Consumer

Reads data from a JMS source.

+



Jython Scripting

Produces record batches using Jython script

+



Dev Data Generator

Generates records with the specified field names based on the selected data type. For

-



Dev Random Record Source

Generates records with the specified field names, using Long data. For development

-



Kinesis Consumer

Reads data from Kinesis

+



MapR DB CDC Consumer

Reads MapR DB CDC data from MapR Streams

+



Dev Raw Data Source

Add Raw data to the source.

-



Dev Snapshot Replaying

Play snapshots as source records

-

Consumers

Regulatory and compliance reporting

Business intelligence and reporting

Self service analytics

Generative AI

Data quality programs

Ecosystem interoperability



Snowflake, Databricks, Teradata

IBM Data Integration 2025

19

Destinations/Targets

Data sources

On-premise

Cloud

Streaming

Data types

Structured

Semi-structured

Unstructured data

60+ Targets/Destinations

Aerospike Client

Writes data to Aerospike

+

Couchbase

Writes data to Couchbase

+

Google Bigtable

Writes data to Google Cloud Bigtable

+

Hive Metastore

Updates the Hive Metastore.

+

IBM watsonx.data

Writes data to IBM watsonx.data using Flight Service

+

Jira

Jira Destination

+

Kinesis Firehose

Writes data to Amazon Kinesis Firehose

+

Cassandra

Writes data to Cassandra

+

Elasticsearch

Upload data to an Elasticsearch cluster

+

HBase

Writes data to HBase

+

IBM Db2

Writes data to IBM Db2 using Flight Service

+

InfluxDB 2.x

Writes data to InfluxDB 2.x

+

JMS Producer

Write data to a JMS MQ.

+

Kinesis Producer

Writes data to Amazon Kinesis

+

Amazon S3

Writes to Amazon S3

-

Azure Data Lake Storage Gen2

Writes data to Azure Data Lake Storage Gen2

-

Azure IoT Hub Producer

Writes data to Azure IoT Hub

-

CoAP Client

Uses a CoAP client to write data

-

Google BigQuery

Writes data to BigQuery

-

Google Pub Sub Publisher

Publishes messages to Google Pub/Sub

-

HTTP Client

Uses an HTTP client to write data.

-

Azure Blob Storage

Writes data to Azure Blob Storage

-

Azure Event Hub Producer

Writes data to Azure Event Hub

-

Azure Synapse SQL

Loads data to Azure Synapse SQL

-

Databricks Delta Lake

Writes data to Databricks Delta Lake tables

-

Google Cloud Storage

Writes to google cloud storage.

-

Hadoop FS

Writes to a Hadoop file system

-

JDBC Producer

Insert, update, and delete data to a JDBC

-

Consumers

Regulatory and compliance reporting

Business intelligence and reporting

Self service analytics

Generative AI

Data quality programs

Ecosystem interoperability









Snowflake, Databricks, Teradata

Transformative Capabilities + Route, Regex, Groovy, Jython, Jolt, Fragments, etc.

Data sources

On-premise

Cloud

Streaming

Data types

Structured

Semi-structured

Unstructured data

70+ Processors/Executors

Couchbase Lookup

Performs Couchbase lookups to enrich records

Encrypt and Decrypt Fields

Encrypts or decrypts field values

Base64 Field Decoder

Decodes a Base64 encoded Byte Array field

Base64 Field Encoder

Encodes a Byte Array field into a Base64 encoded Byte Array

HBase Lookup

Performs KV lookups to enrich records

Hive Metadata

Generates Hive metadata and write information for HDFS

Control Hub API

Calls a Control Hub API

Data Generator

Serializes records to various different data formats.

Jython Evaluator

Processes records using Jython

Kaitai Struct Parser

Parser for binary data based on Kaitai Struct

Data Parser

Parses a field with data

Delay

Allows you to delay any records passing through it by a given number of milliseconds

Kudu Lookup

Performs KV lookups to enrich records

MongoDB Lookup

Performs KV lookups to enrich records

Dev Identity

It echoes every record it receives without changing, other than stage header

Dev Random Error

Generates error records and silently discards records as specified.

TensorFlow Evaluator

Uses TensorFlow models to generate predictions or classifications of data

Web Client

Processor to execute HTTP/S requests

Dev Record Creator

It creates 2 records from each original record

Expression Evaluator

Performs calculations on a field-by-field basis

Whole File Transformer

Transforms whole file data to a different data format

Field Flattener

Flattens nested structures.

Field Hasher

Uses an algorithm to hash field values

Field Mapper

Maps fields in records based on expressions. Operates on field paths names

Field Masker

Masks field values

Consumers

Regulatory and compliance reporting

Business intelligence and reporting

Self service analytics

Generative AI

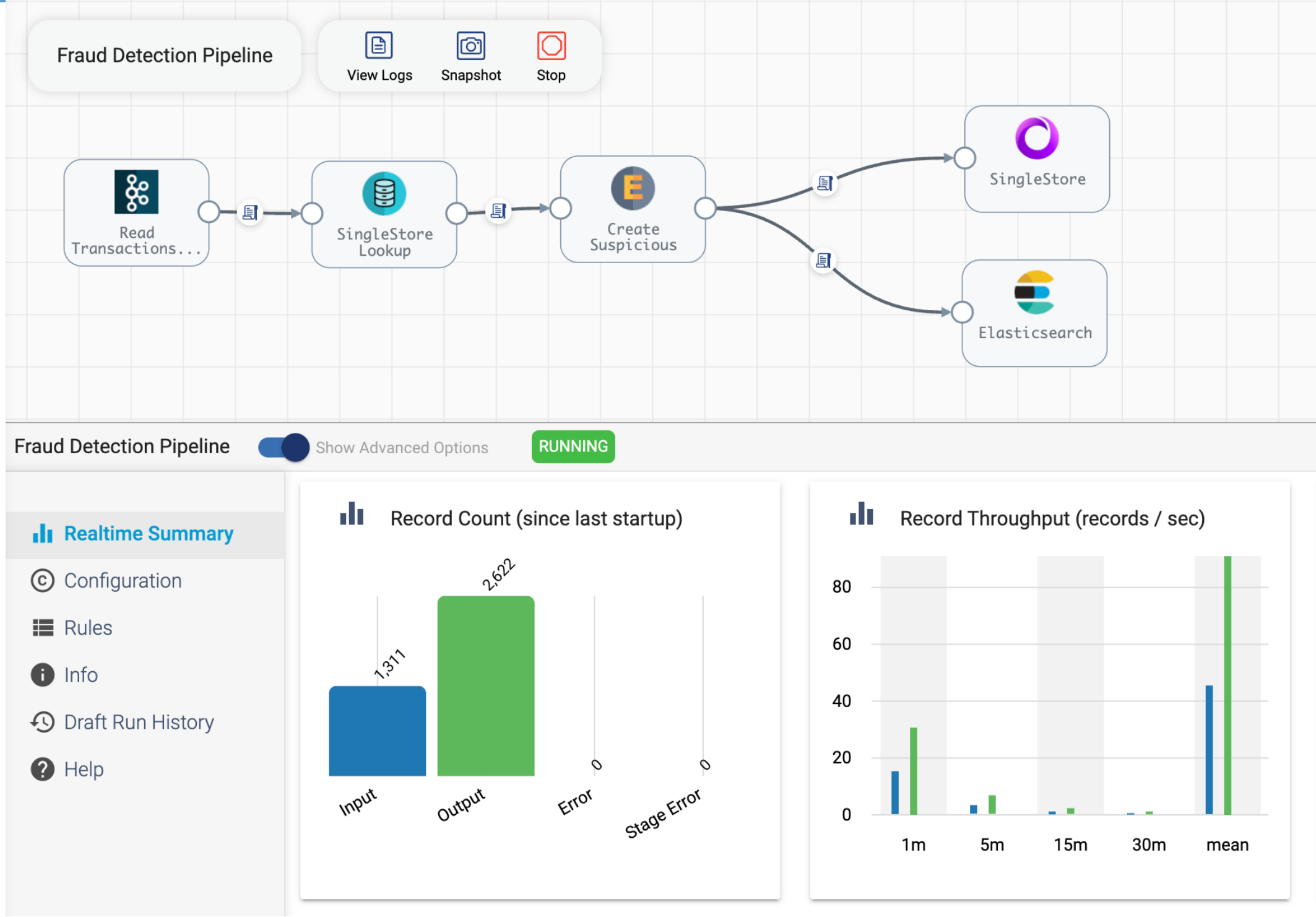
Data quality programs

Ecosystem interoperability

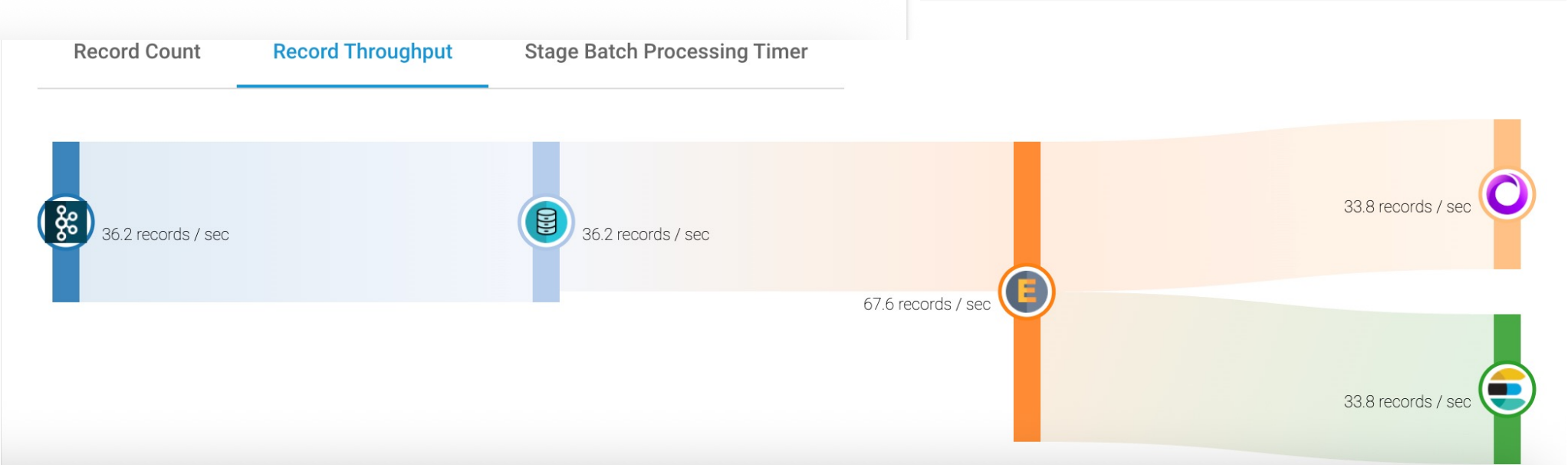
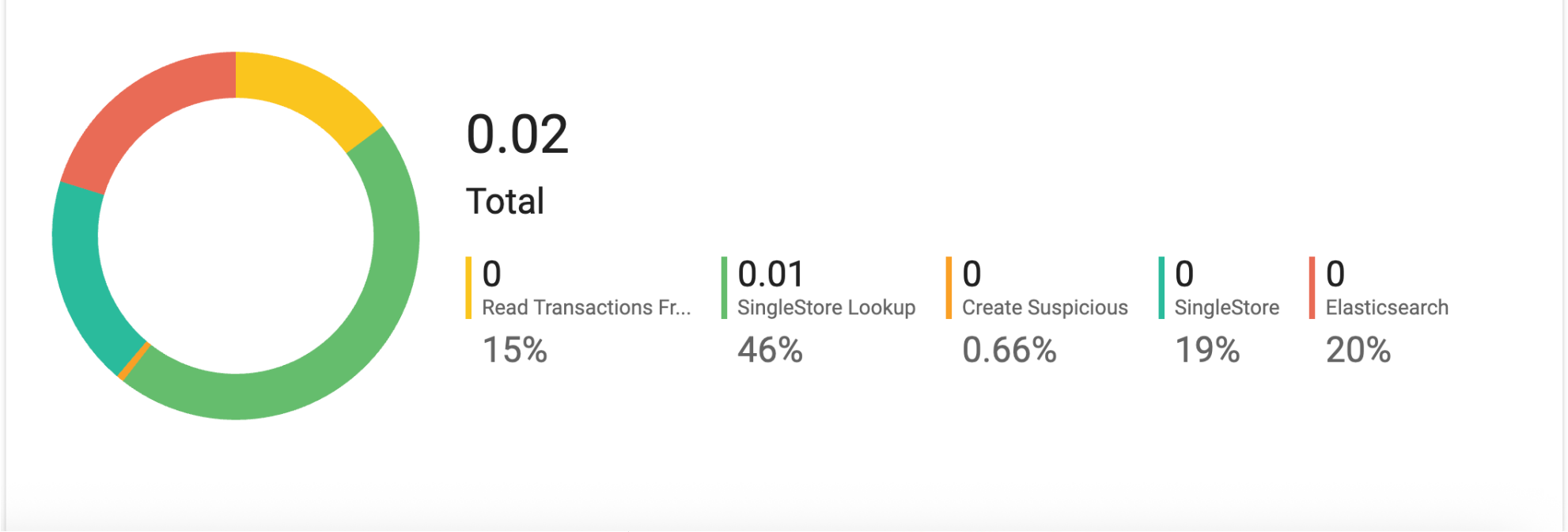


Snowflake, Databricks, Teradata

IBM StreamSets Operate and Monitor Pipelines



Monitor Record Throughput and Error Count (Real-Time, Historical, REST, SDK)



☒ ☐ **Kafka Multitopic Consumer 1 Output Stream 1** Drift in # of Columns

Condition
`${drift:size('/', false)}`

Sampling Records to retain
10

Alert Text
The size of the schema has changed! Call the boss or the data scientists!

Send Email
true

Joint Notifications

Sampling Percentage
100


Automatic Failover, Resume and Recovery

URL	Labels	Accessible	Version
<input type="radio"/> System Data Collector ↗		<input checked="" type="checkbox"/>	4.0.2
<input checked="" type="radio"/> https://172.20.10.184:18632 ↗	Casino #1, Casino #2, Casino #3, Casino #4, all	<input checked="" type="checkbox"/>	5.2.0

IBM StreamSets Templates


Sample Pipelines (14)

☐



Name ↑











☐



Tutorial

Origins

Destinations

<input type="checkbox"/>		Date Conversions	Click to view	Dev Raw Data Source	Trash
<input type="checkbox"/>		HDFS to ADLS Gen2	Click to view	Hadoop FS Standalone	ADLS Gen2
<input type="checkbox"/>		Oracle 19 CDC To Databricks Delta Lake	Click to view	Oracle CDC Client	Databricks Delta Lake
<input type="checkbox"/>		Oracle 19 To Databricks Delta Lake	Click to view	JDBC Multitable Consumer	Databricks Delta Lake
<input type="checkbox"/>		Oracle CDC To Snowflake	Click to view	Oracle CDC Client	Snowflake
<input type="checkbox"/>		Parse Twitter Data To JSON	Click to view	Dev Raw Data Source	Local FS
<input type="checkbox"/>		Parse Web Logs To JSON & Avro	Click to view	Dev Raw Data Source	Local FS
<input type="checkbox"/>		PostgreSQL CDC To Snowflake	Click to view	PostgreSQL CDC Client	Snowflake
<input type="checkbox"/>		Retail Inventory - Join, Aggregation, Reparti...	Click to view	File	Local FS
<input type="checkbox"/>		Salesforce CDC To Snowflake	Click to view	Salesforce	Snowflake

Filter Pipelines



☐

Keep Filter Persistent

Filter by Engine

☒ Show All

☐ Data Collector

☐ Transformer

☐ Transformer for Snowflake

Filter by Sample Type

☒ System Samples

☐ User Samples (pipelines with label 'templates')

Out of the box examples and user defined templates help ensure consistency.

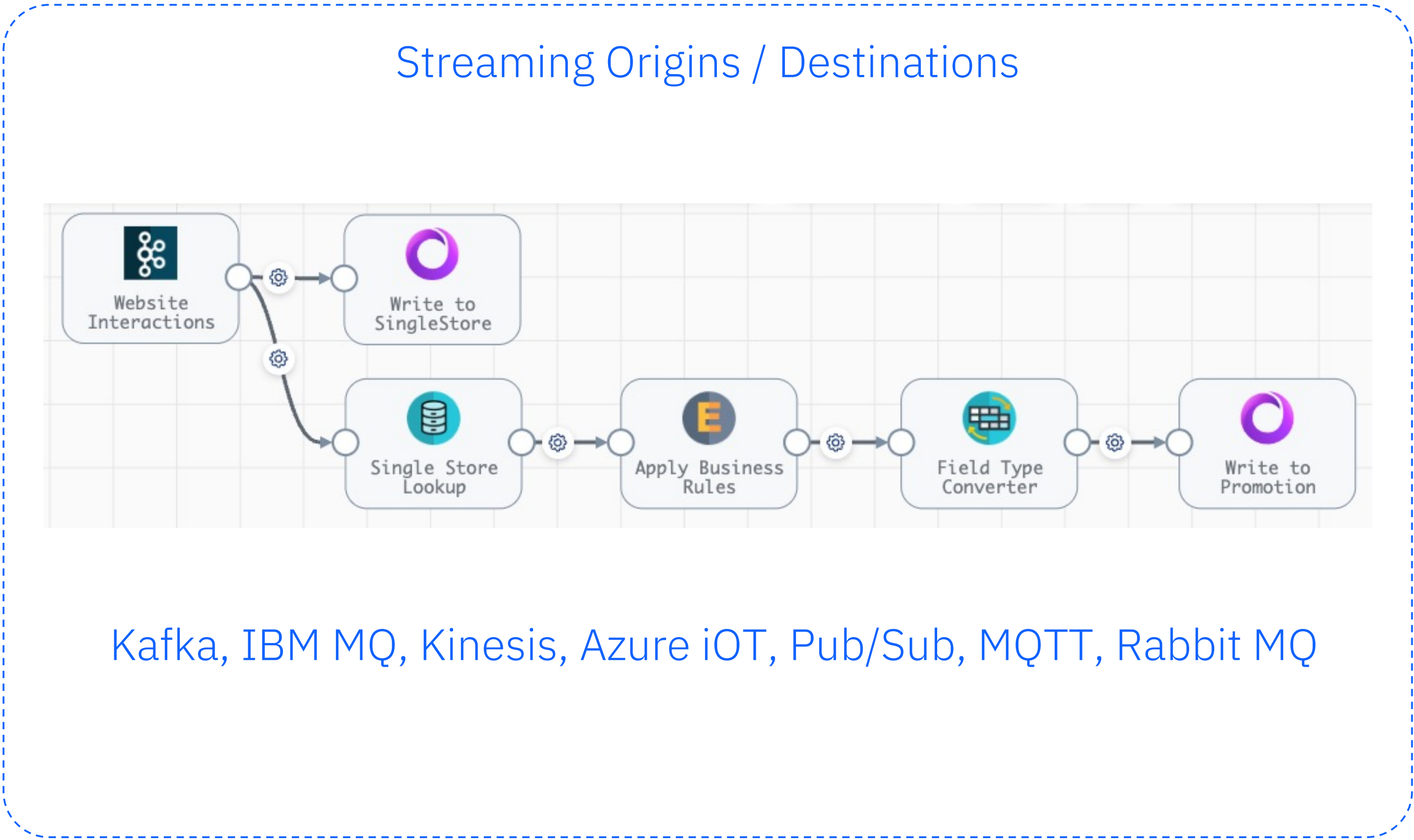
Streaming

Data sources

- On-premise
- Cloud
- Streaming

Data types

- Structured
- Semi-structured
- Unstructured data



Includes:

- Stream critical data and events in real-time
- Leverage broker capabilities
- Highly performant and resilient
- Real-time Alerting

Ecosystem interoperability     | Snowflake, Databricks, Teradata

IBM StreamSets

A no-code/low-code streaming data integration tool
for engineers who also *want the ability to code*

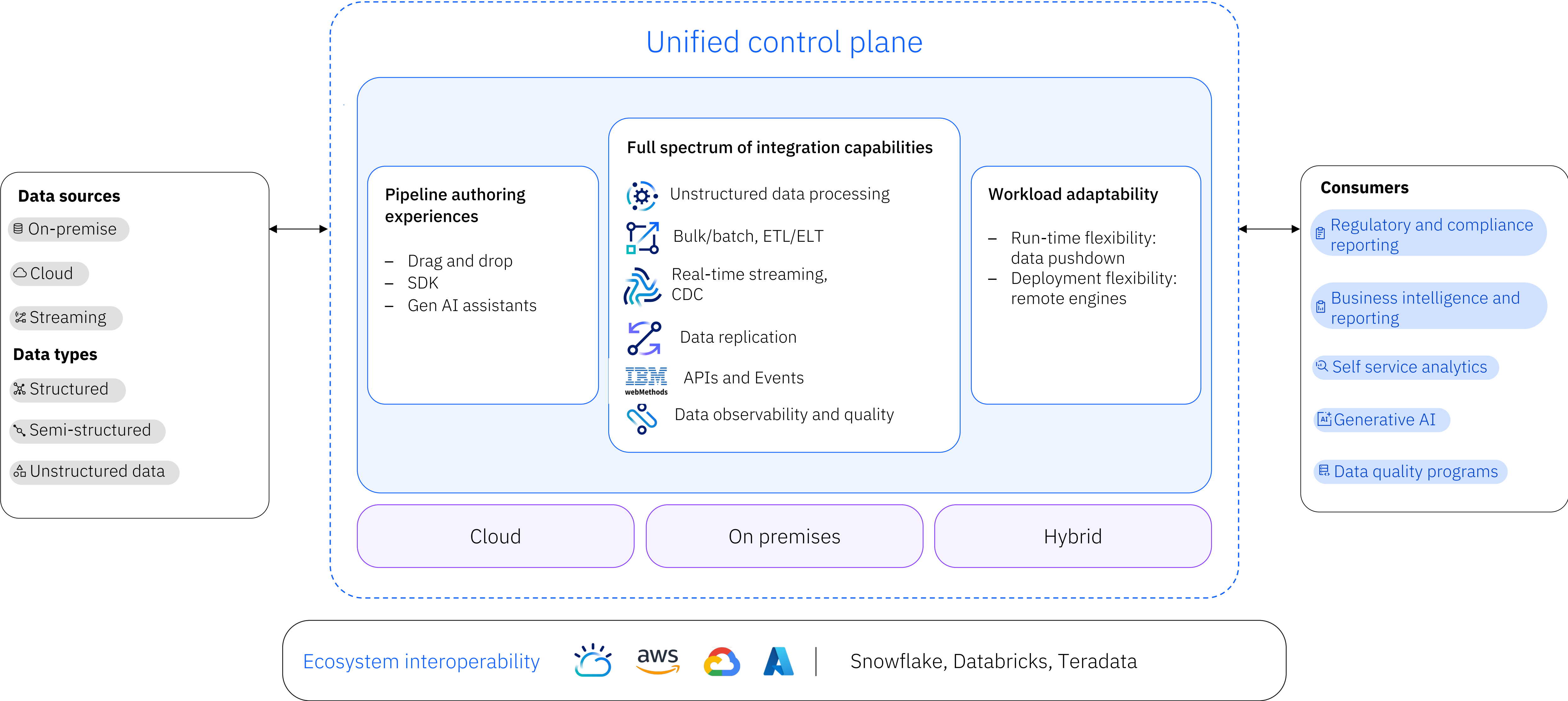


Best-in-class
developer
experience

Capabilities without
compromise

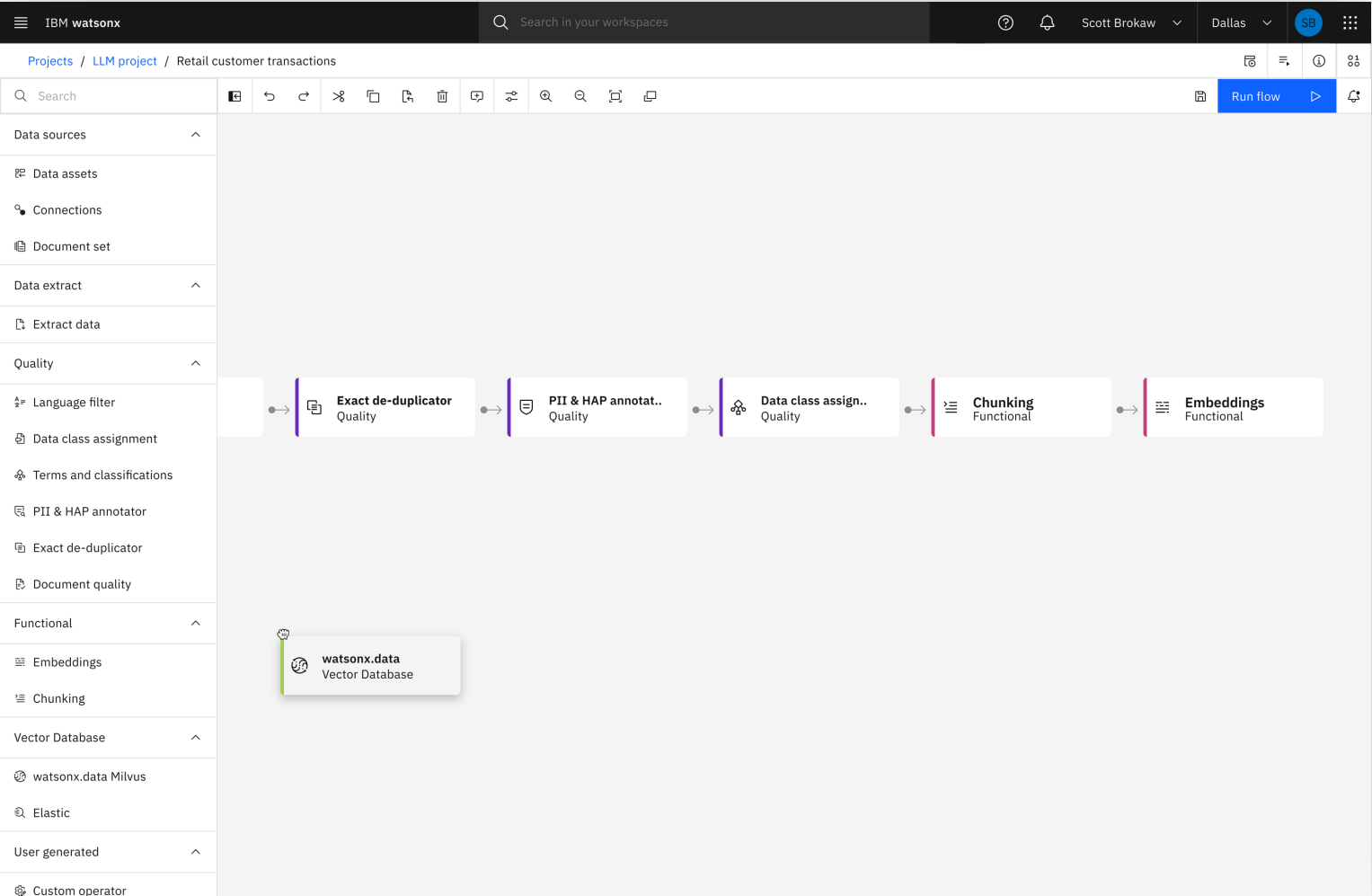
Built for enterprise
scale

Roadmap: IBM data integration

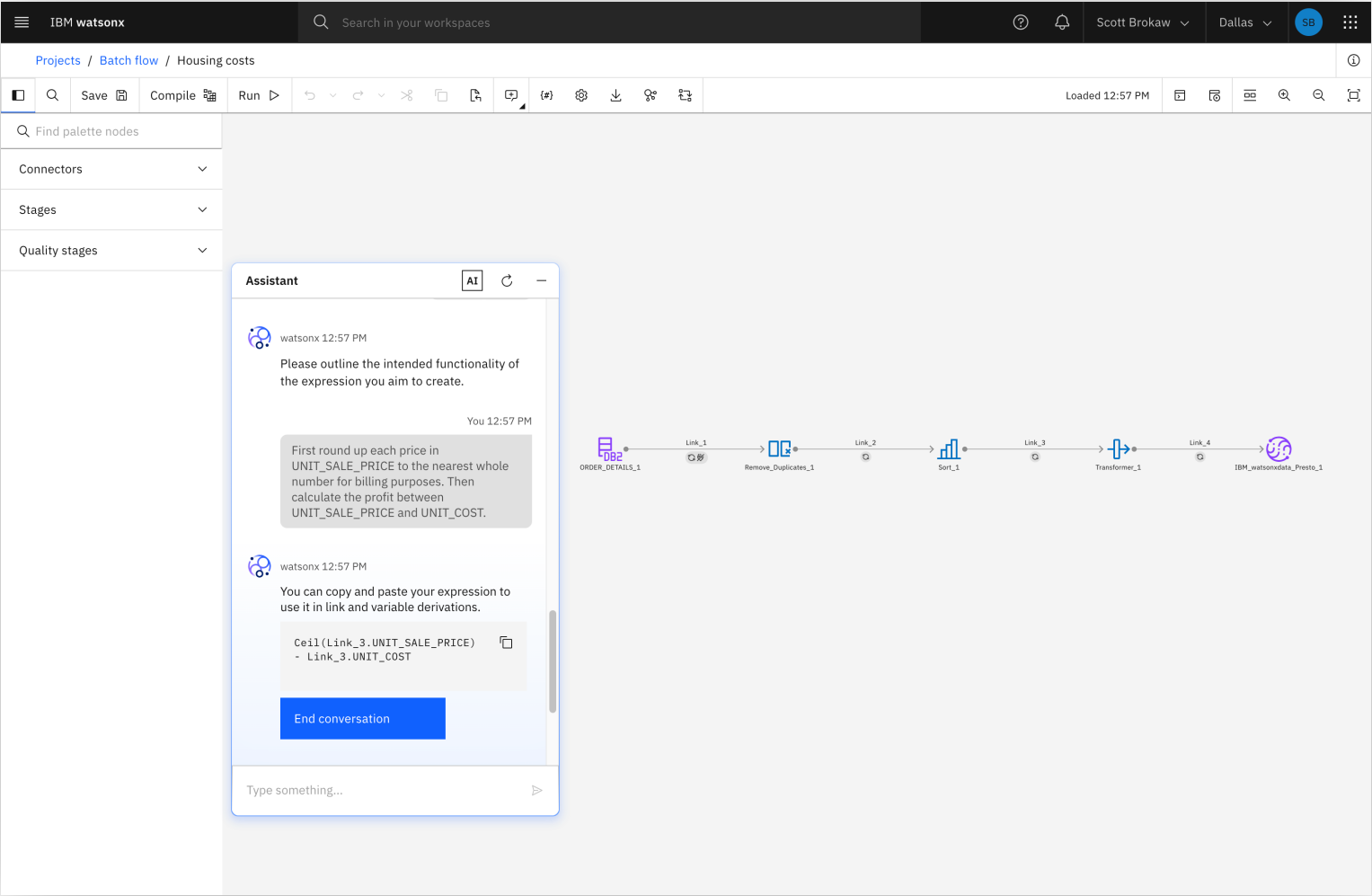


Pipeline authoring experiences

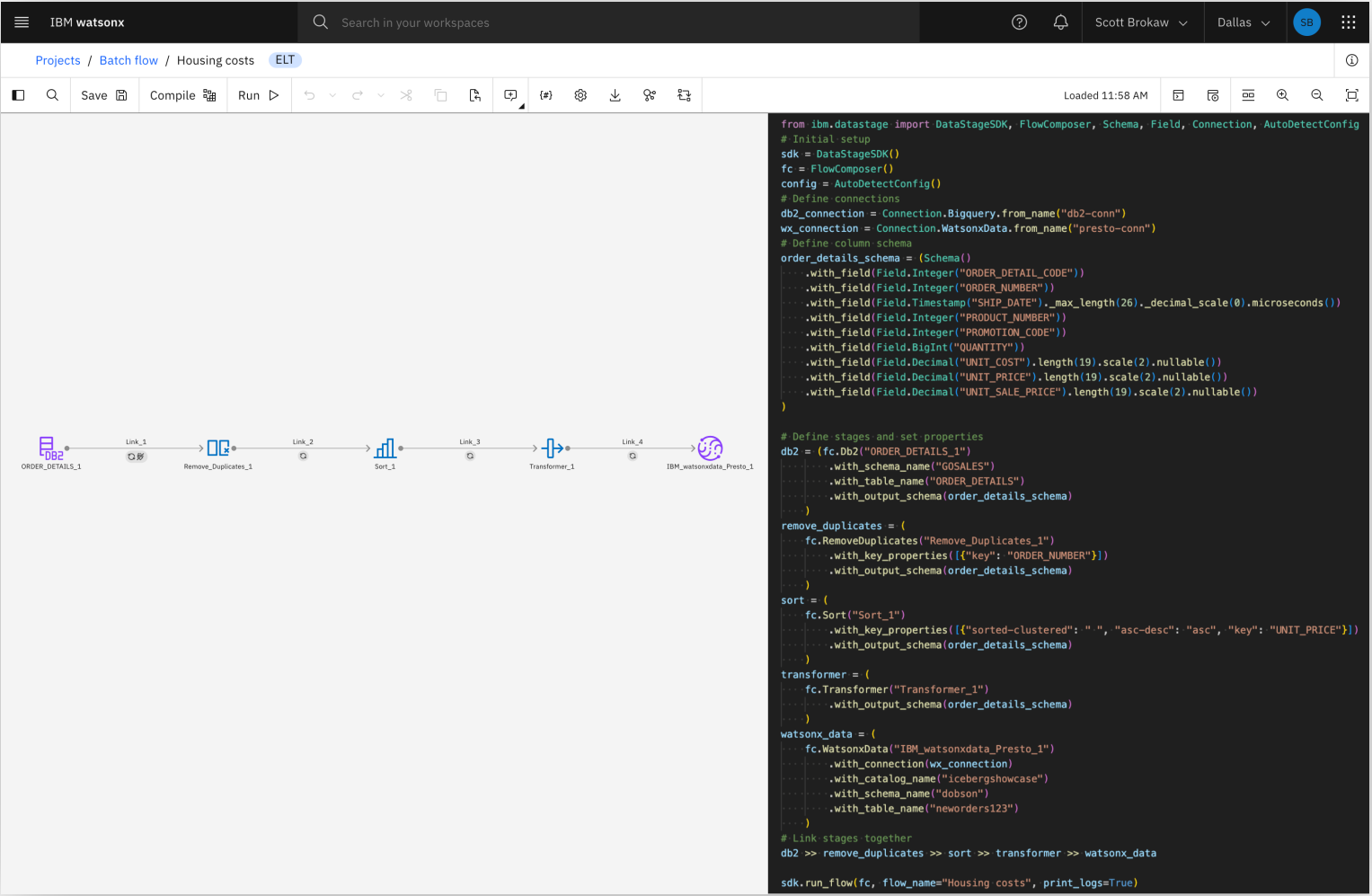
Canvas drag and drop



AI assistant

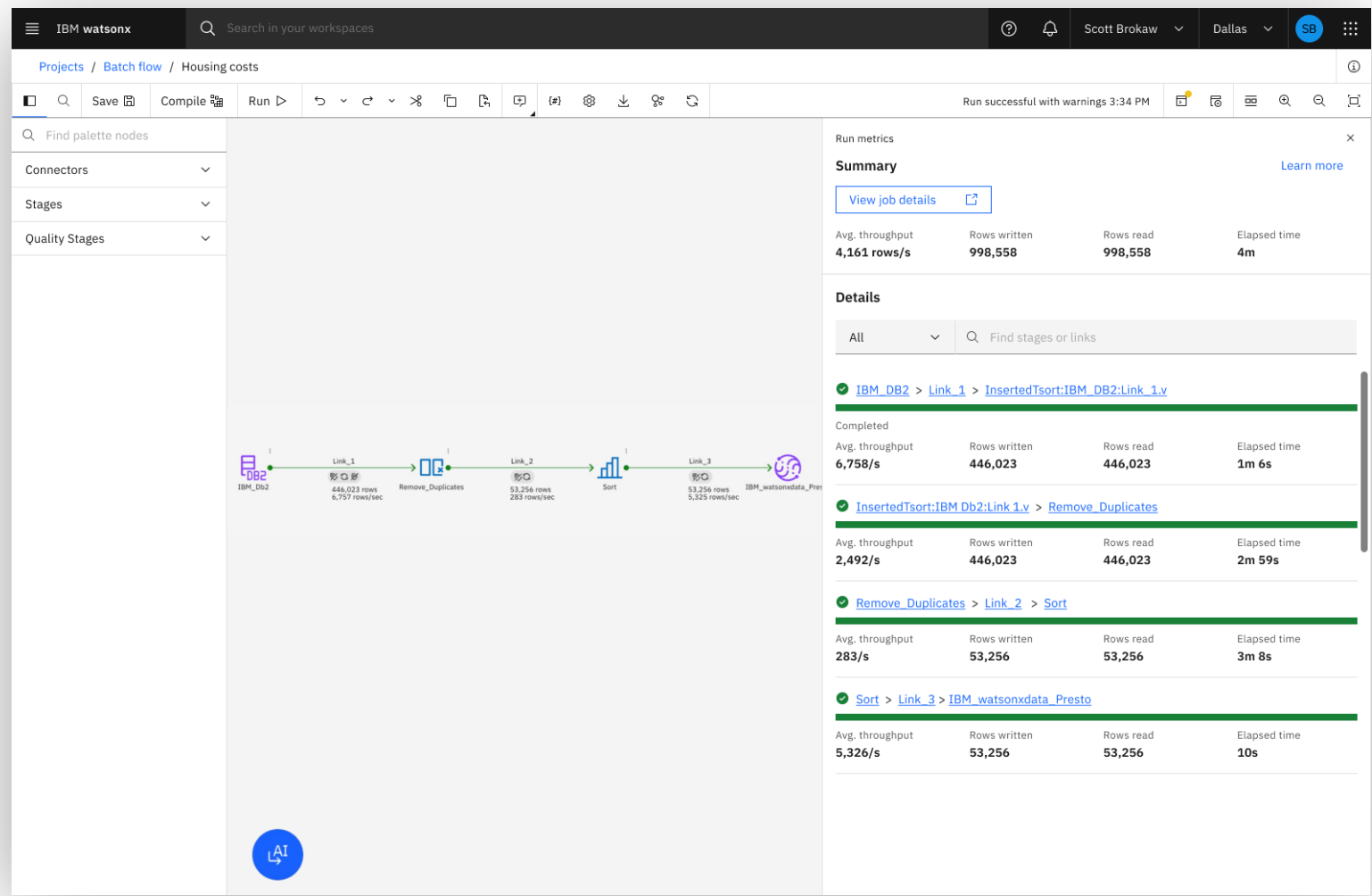


SDK

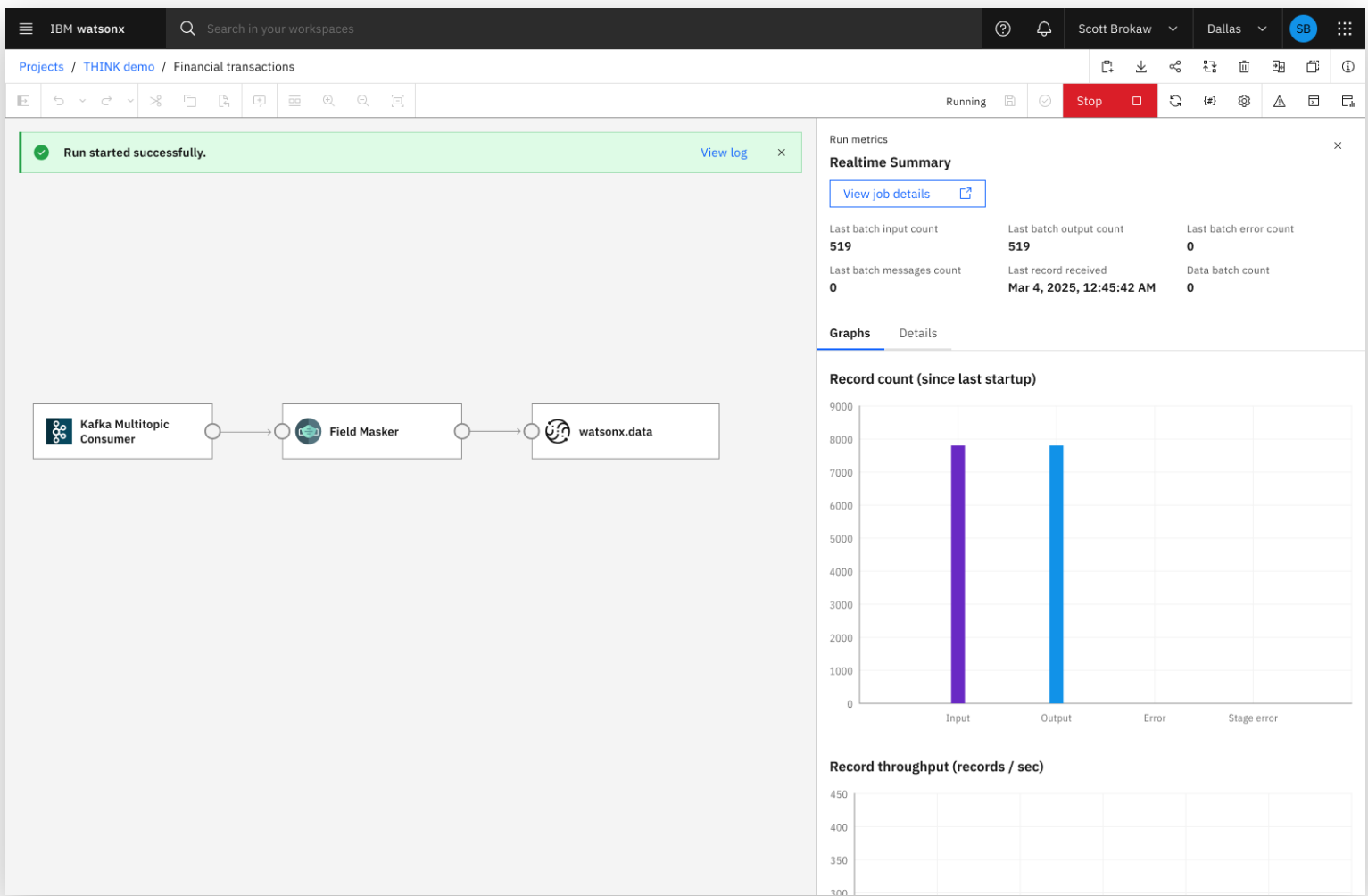


Full spectrum of integration capabilities

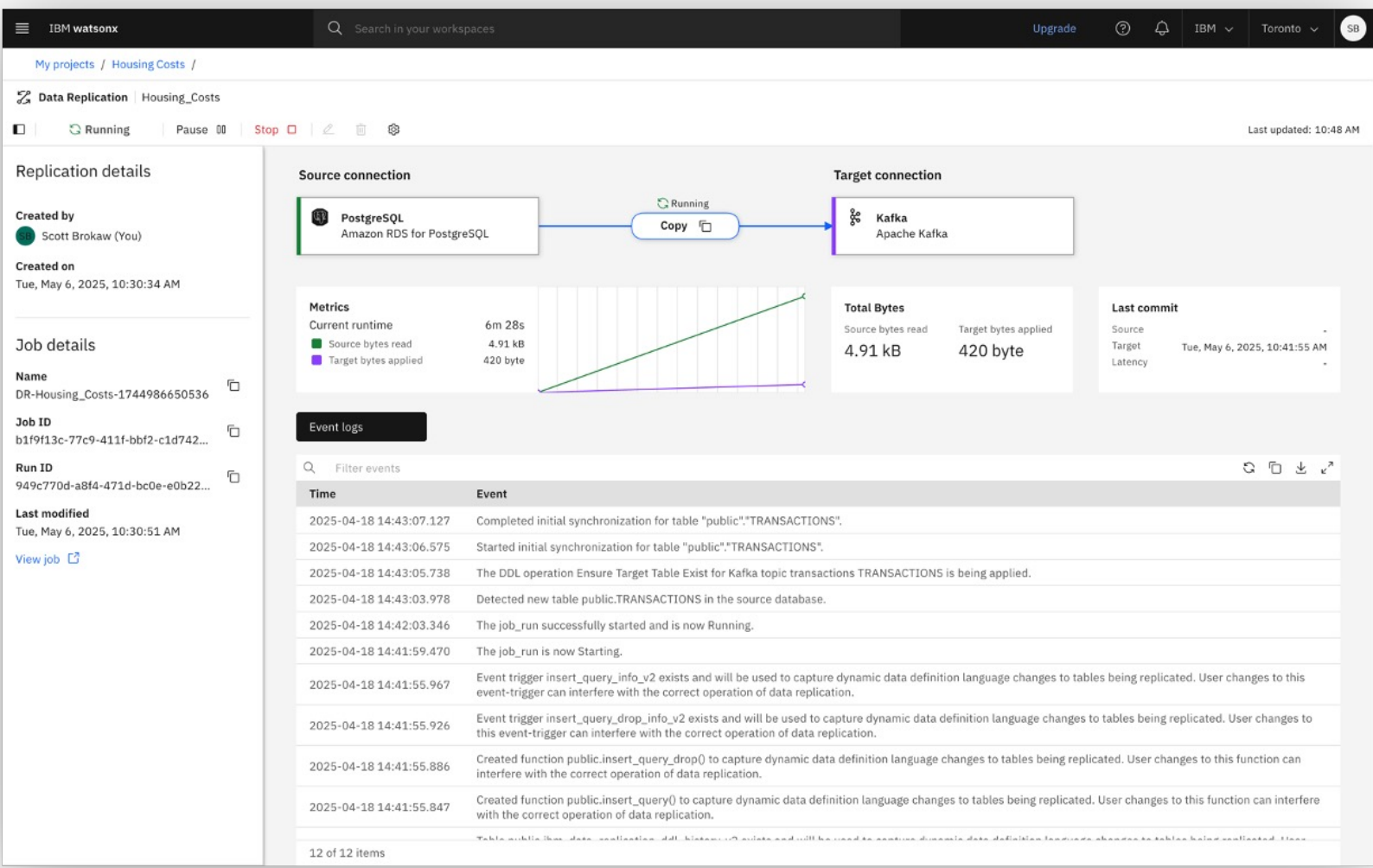
Batch



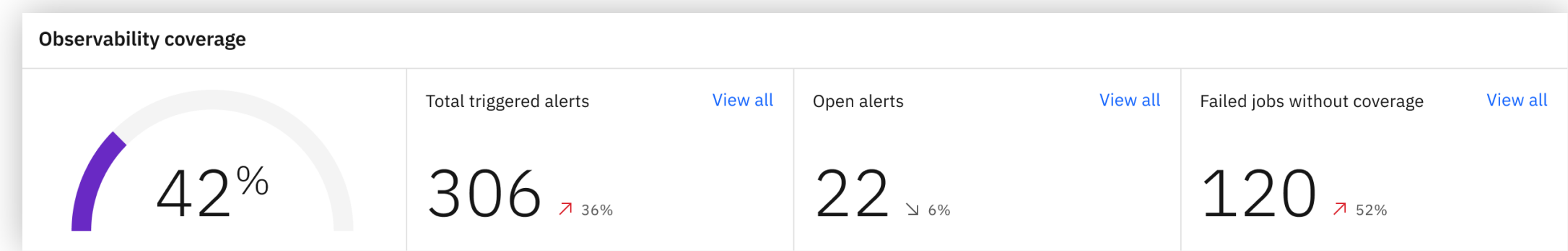
Streaming



Data replication



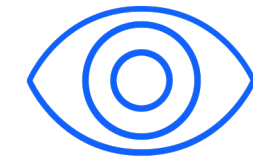
Data observability



Q&A Time



Three ways to get started with IBM data integration today



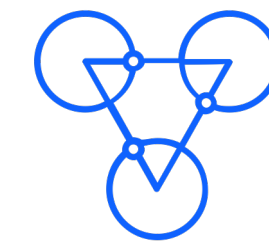
Want to read more about IBM data integration?

[Visit the IBM data integration website →](#)



See how it works for you.

[Start a data integration trial →](#)



Kickstart your project with the IBM technical experts

[Book here →](#)

