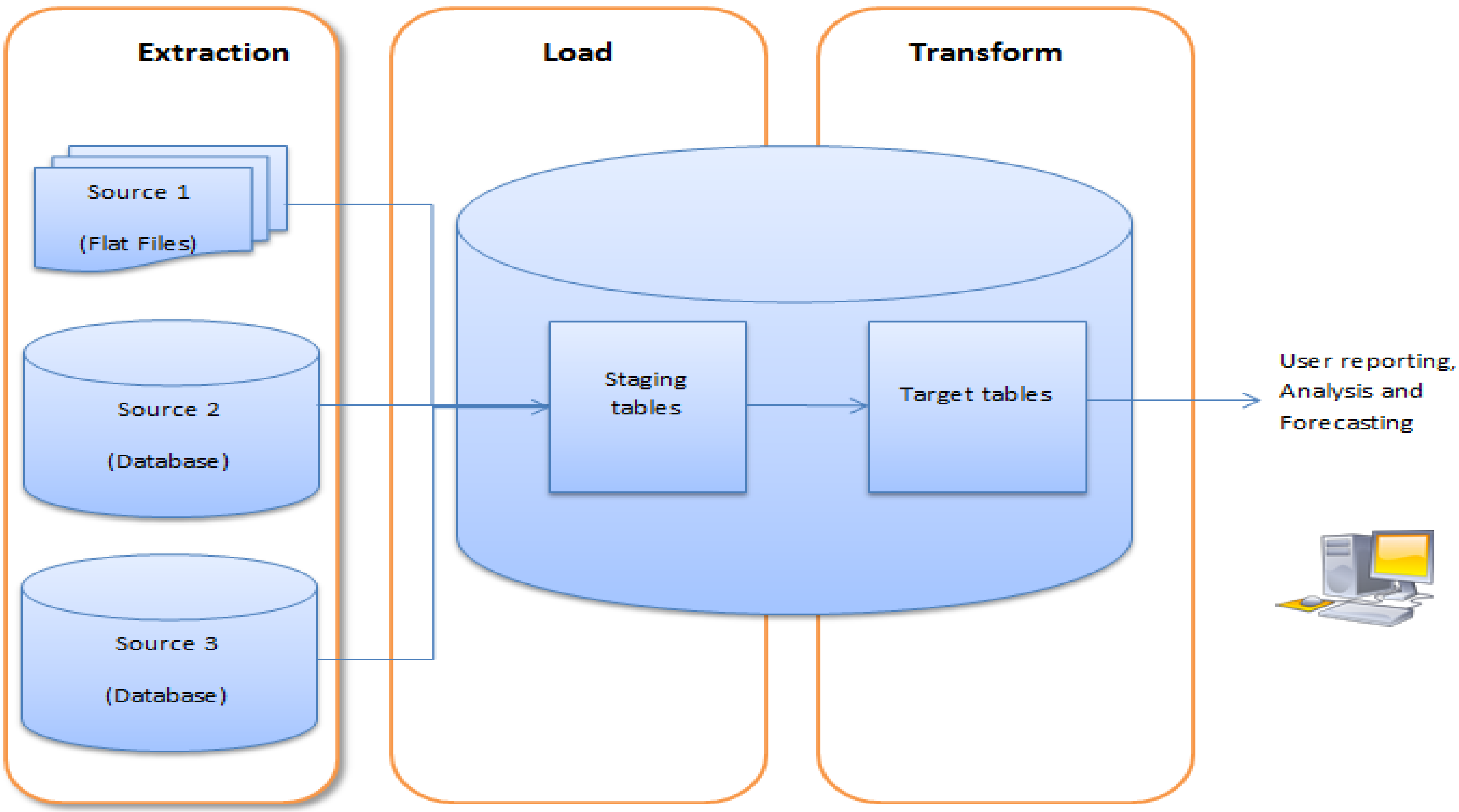
The ETL Process

Extract, Transform, and Load operations
for data warehouses

ETL (extract, transform, load)

ELT based Data Warehouse Architecture diagram



THE ETL PROCESS

- Capture/Extract
- Scrub or data cleansing
- Transform
- Load and Index

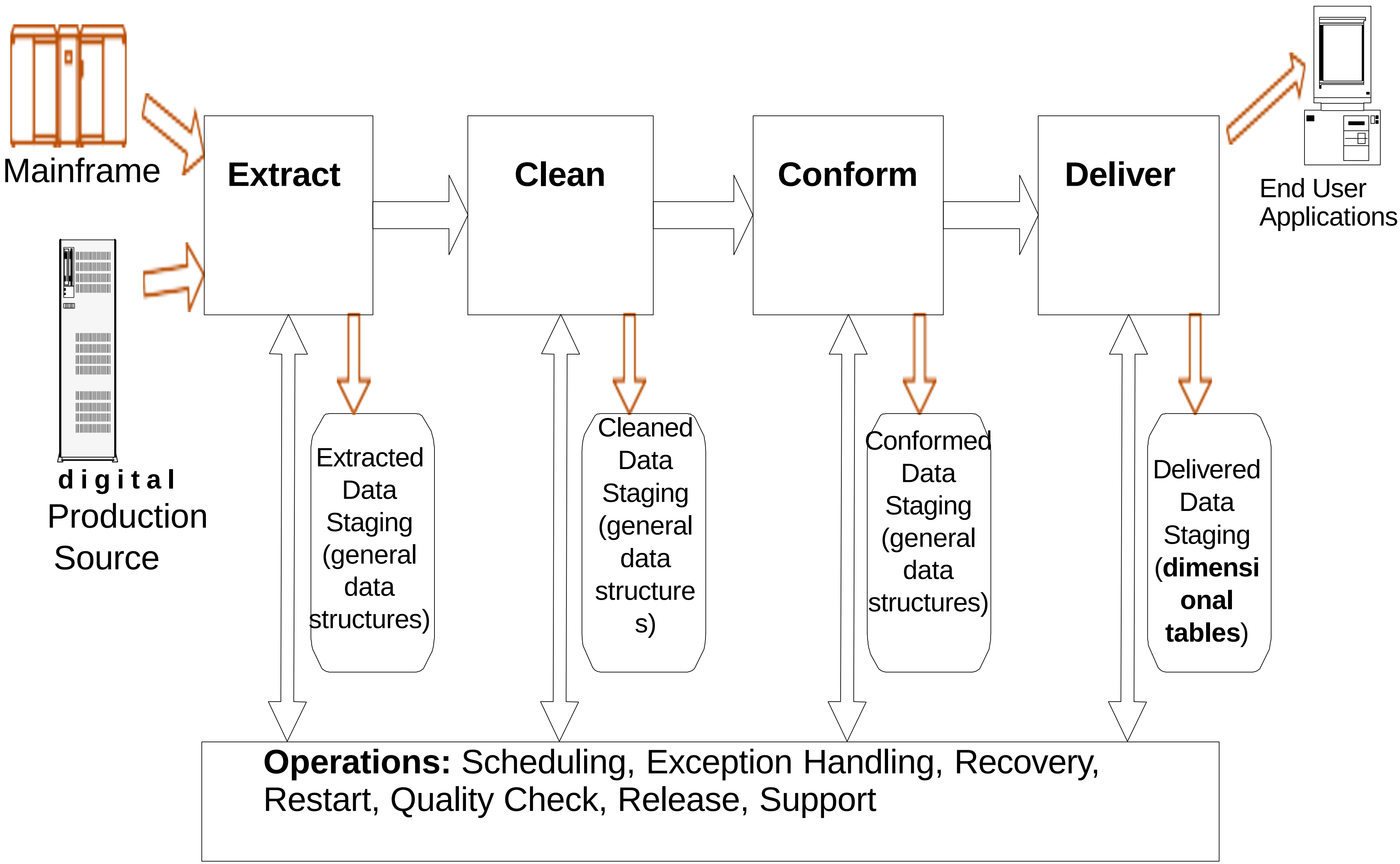
Pre ETL Tasks

- Profiling the data:
  - Know your data
  - Will give direct insight in the data quality of the source systems.
  - Find out how many rows have missing or invalid values, or what is the distribution of values in a specific column.
- This knowledge will help to specify rules (data standards / quality checks) in order to cleanse the data and to keep really bad data out of the repository.
- Doing data profiling before designing the ETL process helps you to better design a system that is correct, robust and has a clear structure.

ETL

- Data moved from source to target databases
- Focus: Preparing the data for reporting / analysis
- ETL = Extract -> Transform -> Load
  - Extract: Get the data from source(s) as efficiently as possible
  - Transform: Perform calculation / map data / clean data
  - Load: Load data to the target storage

The Four Staging Steps of a Data Warehouse



Source: Kimball, Ralph, and Joe Caserta. *The data warehouse ETL toolkit*. John Wiley & Sons, 2004.

Dirty Data

- Absence of Data / Missing Data
- Multipurpose Fields
- Cryptic Data
- Contradicting Data
- Violation of Data Rules
- Reused Primary Keys
- Non-Unique Identifiers
- Data Integration Problems

Data Cleaning: Parsing / Combining

- Parsing locates and identifies individual data elements in the source files and then *isolates* these data elements in the target files.
  - Examples include full name stored in one column

Name
Harry Johnson
Mary J. Blair
Dr. Chris Young
Martin K. Brown II

Title	First Name	Middle Name	Last Name	Suffix
	Harry		Johnson	
	Mary	J.	Blair	
Dr.	Chris		Young	
	Martin	K.	Brown	II

- Combining locates and identifies individual data elements in the source files and then *combines* these data elements in the target files
  - Examples include event date stored in different columns as date, month and year

Date	Month	Year
9	May	2016
06	04	2015

Date
2016-05-09
2015-04-06

Data Cleaning: Standardizing

- Standardizing applies conversion routines to transform data into its preferred (and consistent) format using both standard and custom data rules.

Sex
Male
F
M
Female
NULL
f
m

Sex
1
2
1
2
0
2
1

Data Cleaning: Matching

- Searching and matching records within and across the parsed, combined, corrected and standardized data based on predefined data rules to eliminate duplications, sequences

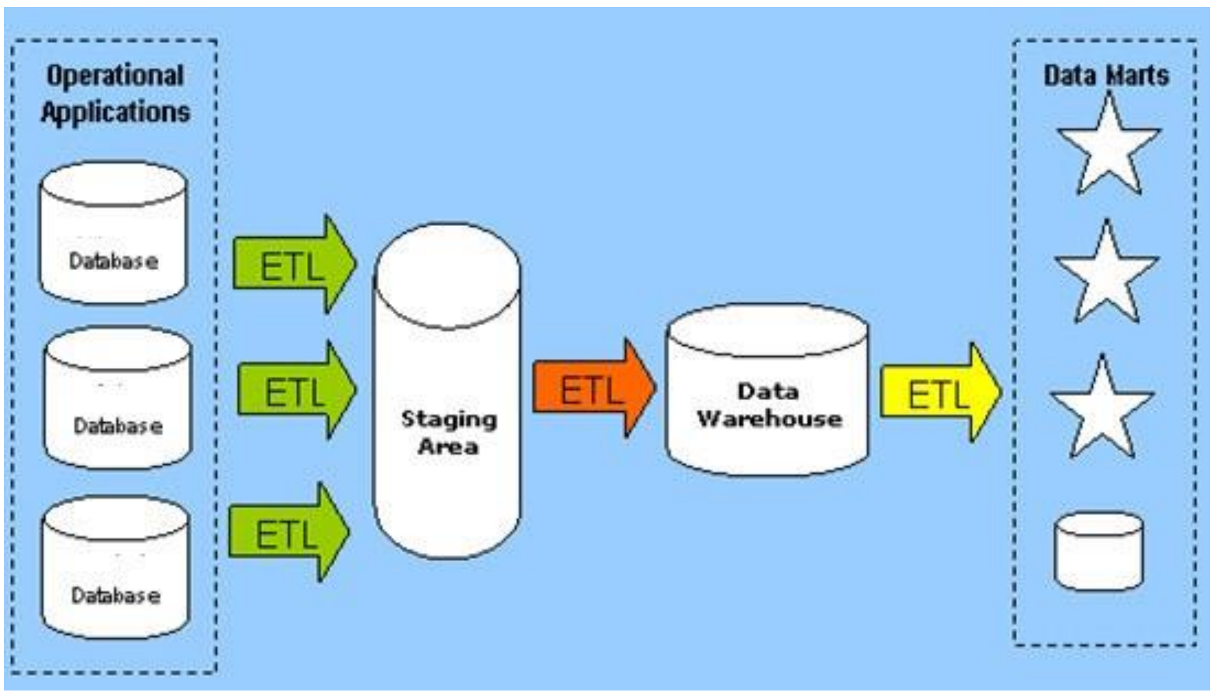
Pregnancy Number	Outcome Date	Outcome
1	2011-06-07	Twin

DoB	Name
2011-06-07	Child1
2011-06-07	Child2

Pregnancy Number	Birth Date	Name
1	2011-06-07	Child1
1	2011-06-07	Child2

Data Staging

- Used as an interim step between data extraction and later steps
- Accumulates data from asynchronous sources using native interfaces, flat files, FTP sessions, or other processes
- Data in the staging file is transformed and loaded to the warehouse
- There is no end user access to the staging file
- An operational data store may be used for data staging



ETL Tools

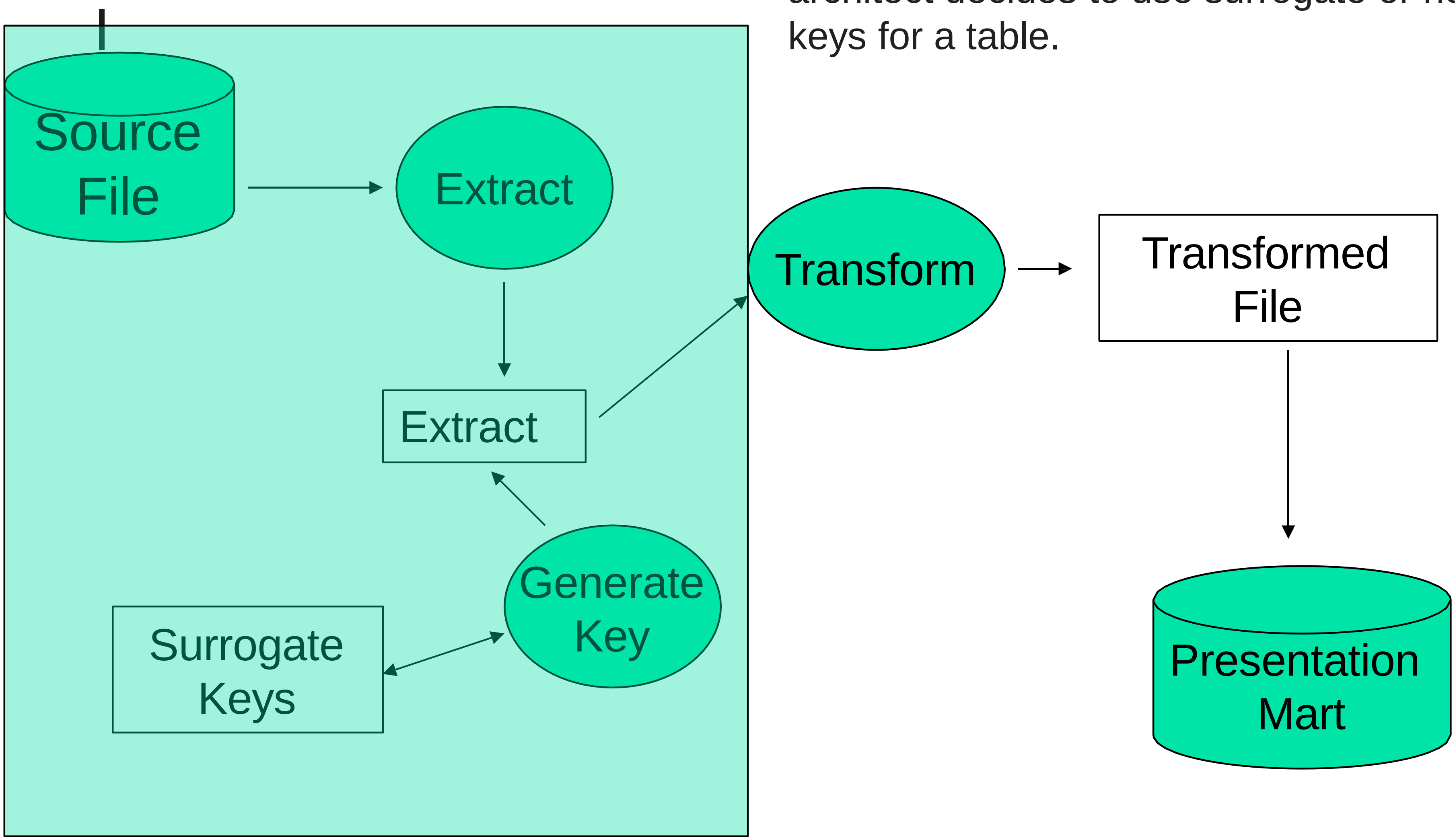
- Few popular commercial and freeware(open-sources) ETL Tools

- Commercial ETL Tools:
  - IBM Infosphere DataStage
  - Informatica PowerCenter
  - Oracle Warehouse Builder (OWB)
  - Oracle Data Integrator (ODI)
  - SAS ETL Studio
  - Business Objects Data Integrator(BODI)
  - Microsoft SQL Server Integration Services(SSIS)
  - Ab Initio

- Freeware, open source ETL tools:
  - Pentaho Data Integration (PDI)
  - Talend Integrator Suite
  - CloverETL
  - Jasper ETL

The Flow

A surrogate key is **a unique key for an entity in the client's business or for an object in the database.** Sometimes natural keys cannot be used to create a unique primary key of the table. This is when the data modeler or architect decides to use surrogate or helping keys for a table.



Add Surrogate Keys

in case we do not have a natural primary key in a table, then we need to artificially create one in order to uniquely identify a row in the table , this key is called the surrogate key or synthetic primary key of the table.

Example of Surrogate Key

registration_no	name	percentage
210101	Harry	90
210102	Maxwell	65
210103	Lee	87
210104	Chris	76

registration_no	name	percentage
CS107	Taylor	49
CS108	Simon	86
CS109	Sam	96
CS110	Andy	58

MERGE

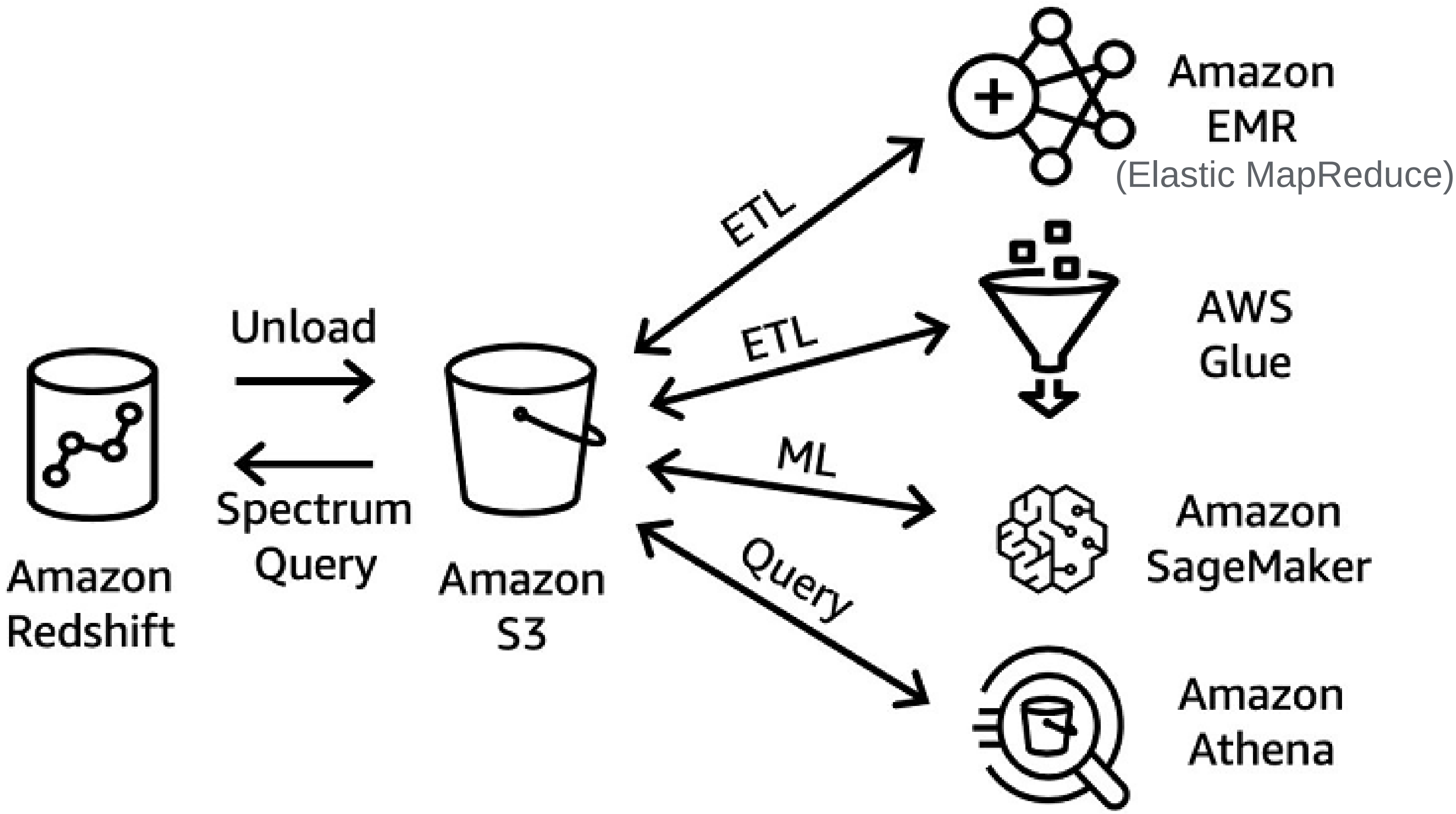
surr_no	registration_no	name	percentage
1	210101	Harry	90
2	210102	Maxwell	65
3	210103	Lee	87
4	210104	Chris	76
5	CS107	Taylor	49
6	CS108	Simon	86
7	CS109	Sam	96
8	CS110	Andy	58

`IDENTITY` keyword to perform an auto-increment feature.

Create a Surrogate Key Table

```
create table surrogate (
    AddressID int identity(1,1),
    NOT NULL,
    ...,
    PRIMARY KEY (AddressID)
    );
```

Amazon ETL



Amazon Web Services, Inc.

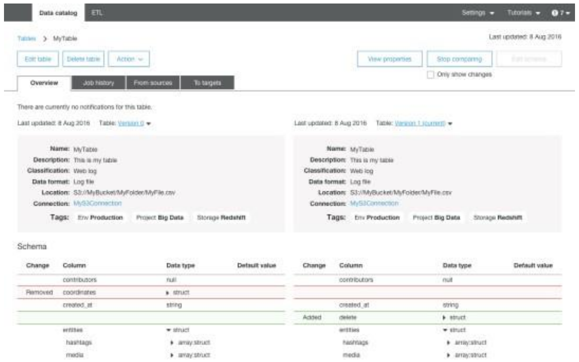
ETL on Redshift

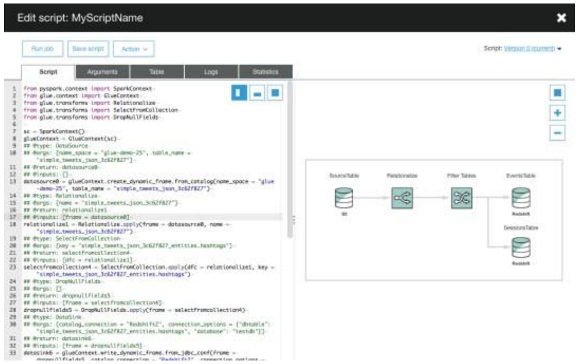
AWS Glue

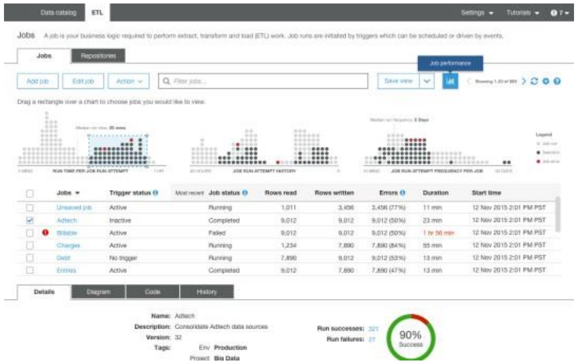
Fully-managed data catalog and ETL service

Integrated with:

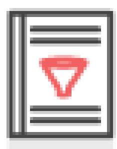
S3, RDS, Redshift & any
JDBC-compliant data store





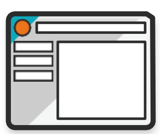


AWS Glue: Components



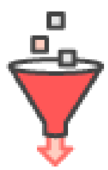
Data Catalog

- Hive Metastore compatible with enhanced functionality
- Crawlers automatically extracts metadata and creates tables
- Integrated with Amazon Athena, Amazon Redshift Spectrum



Job Authoring

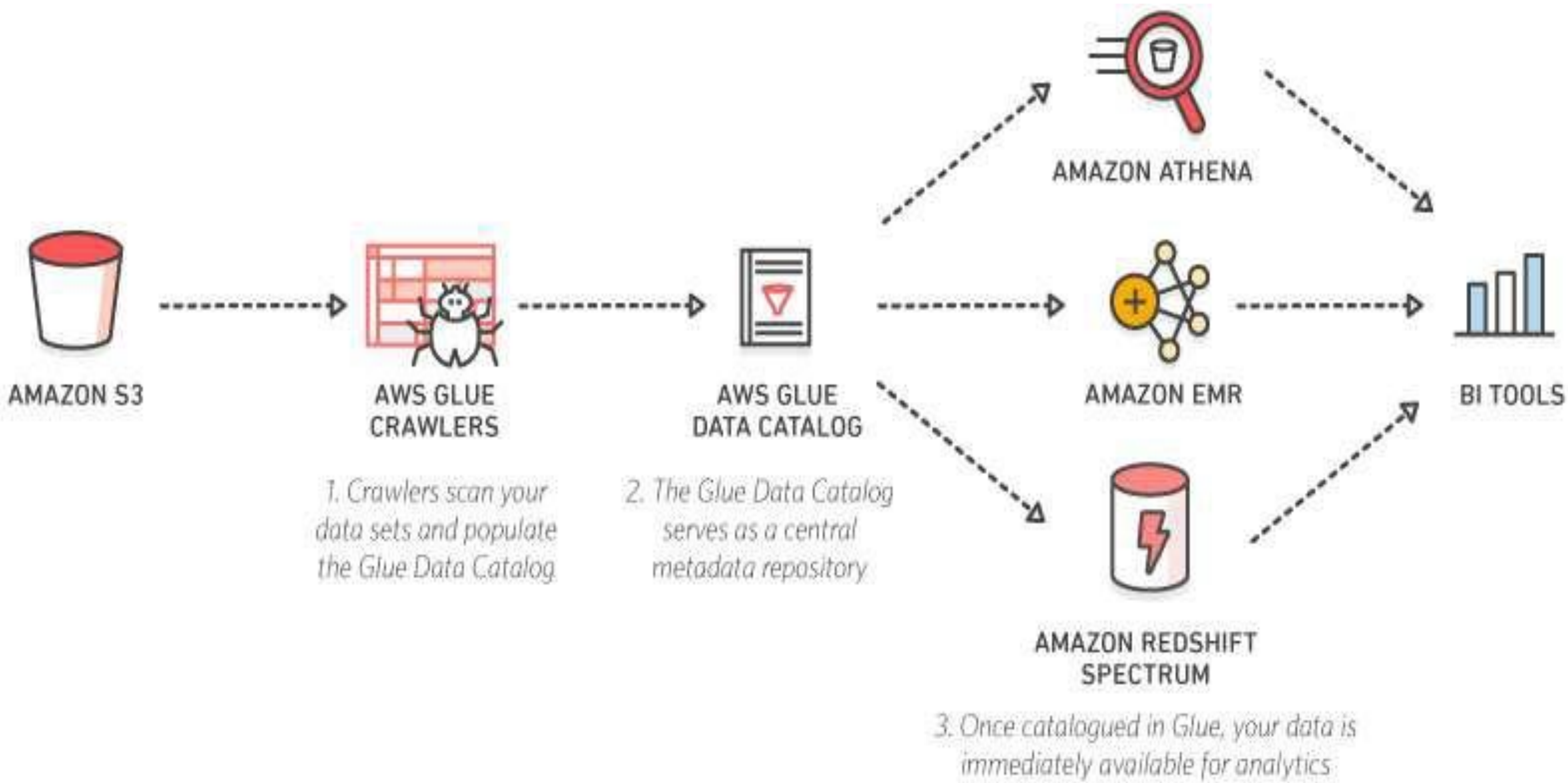
- Auto-generates ETL code
- Build on open frameworks – Python and Spark
- Developer-centric – editing, debugging, sharing



Job Execution

- Run jobs on a serverless Spark platform
- Provides flexible scheduling
- Handles dependency resolution, monitoring and alerting

Instantly query your data lake on Amazon S3



Amazon Web Services, Inc.

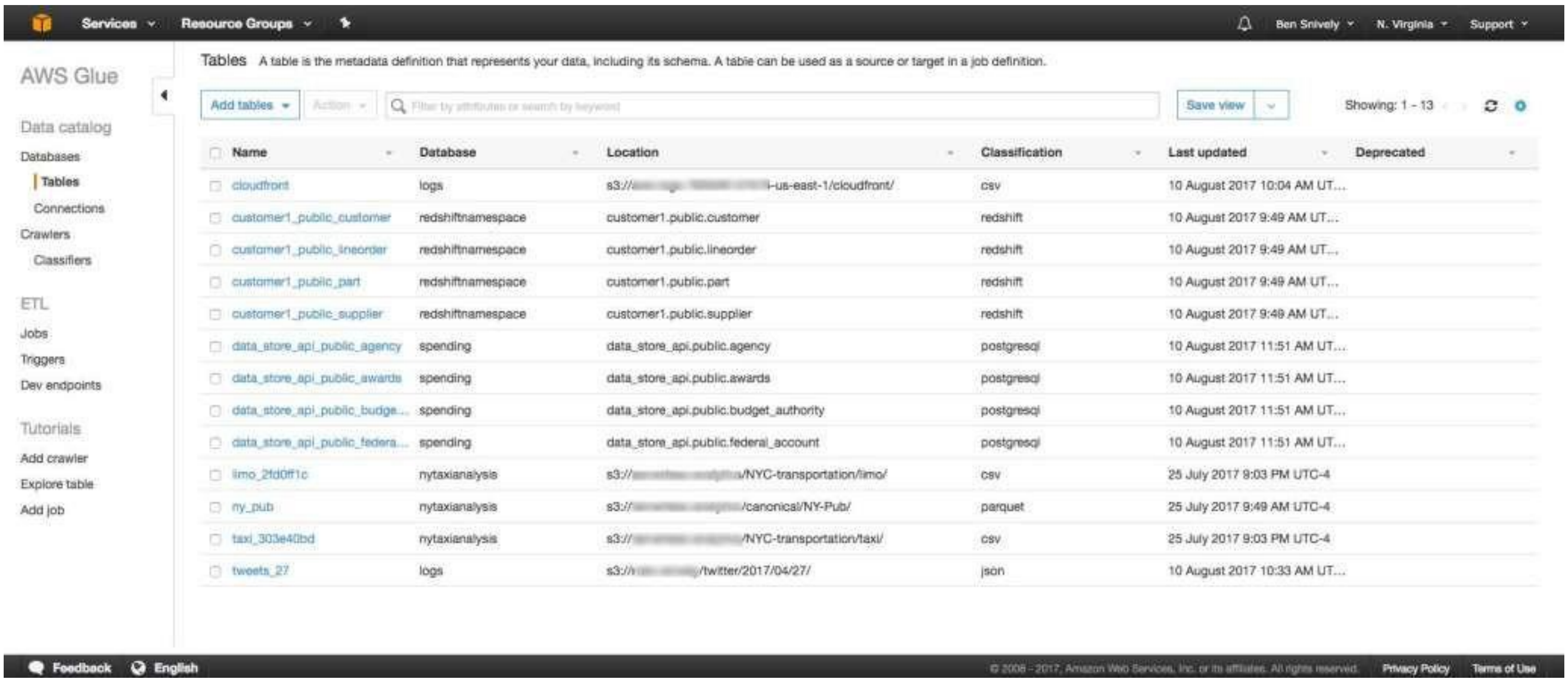
Data Catalog: Crawlers

Crawlers automatically build your Data Catalog and keep it in sync

- Automatically discover new data, extracts schema definitions
  - Detect schema changes and version tables
  - Detect Hive style partitions on Amazon S3
- Built-in classifiers for popular types; custom classifiers using Grok expressions
- Run ad hoc or on a schedule; serverless – only pay when crawler runs

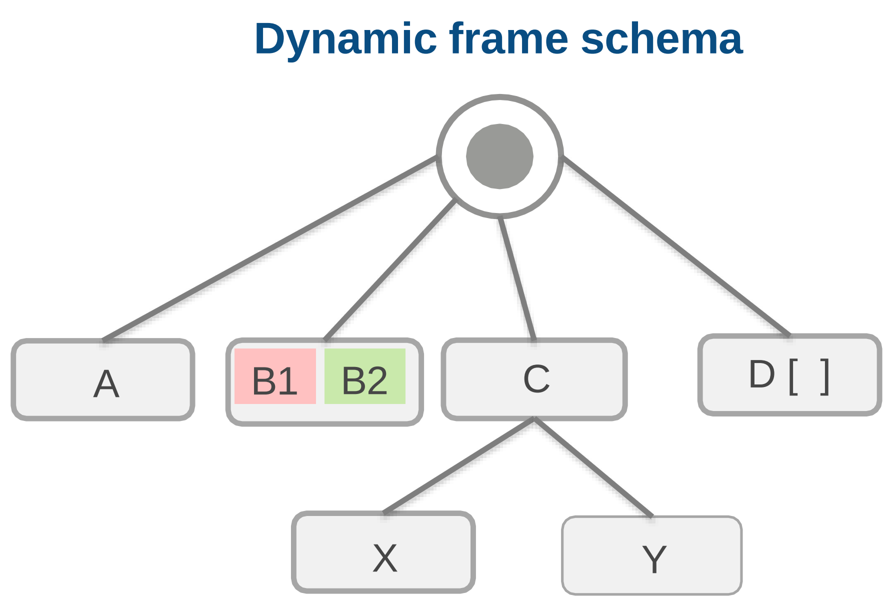
Bring in metadata from a variety of data sources (Amazon S3, Amazon Redshift, etc.) into a single categorized list that is searchable

AWS Glue



Amazon Web Services, Inc.

Job Authoring: Glue Dynamic Frames



Like Spark's Data Frames, but better for:

- Cleaning and (re)-structuring **semi-structured** data sets, e.g. JSON, Avro, Apache logs ...

No upfront schema needed:

- Infers schema on-the-fly, enabling transformations in a **single pass**

Easy to handle the unexpected:

- Tracks new fields, and inconsistent changing data types with **choices**, e.g. integer or string
- Automatically mark and separate error records

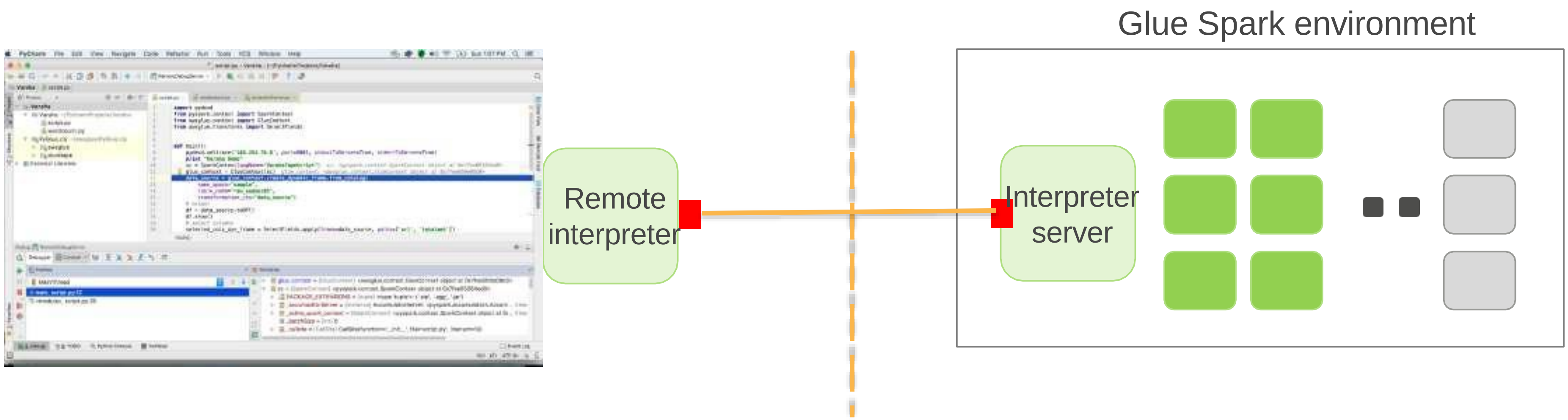
Job authoring: Glue transformations

Add transform

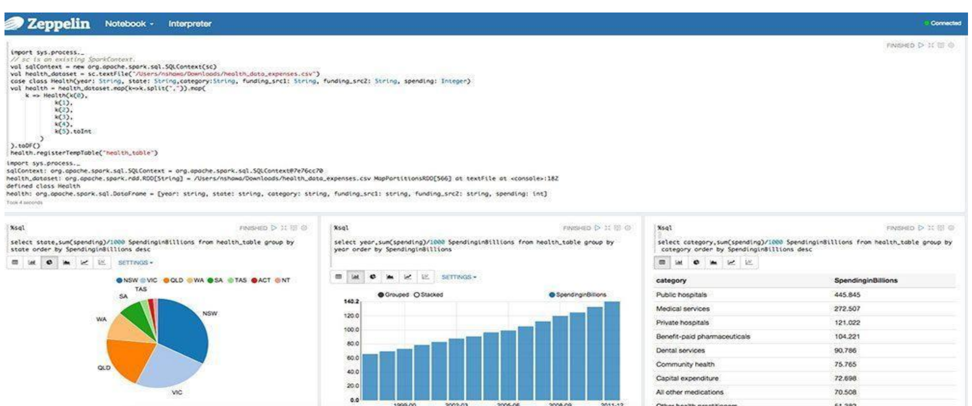
Name	Description
DropFields	Drop fields from a DynamicFrame
DropNullFields	DynamicFrame without null fields
Join	Join two DynamicFrames
MapToCollection	Apply a transform to each DynamicFrame in this DynamicFrameCollection
Relationalize	Flatten nested schema and pivot out array columns from the flattened frame
RenameField	Rename a field within a DynamicFrame
SelectFields	Select fields from a DynamicFrame
SelectFromCollection	Select one DynamicFrame from a DynamicFrameCollection
SplitFields	Split fields within a DynamicFrame
SplitRows	Split rows within a DynamicFrame based on comparators
Unbox	Unbox a string field

- **Prebuilt transformation:** Click and add to your job with simple configuration
- **Spigot** writes sample data from DynamicFrame to S3 in JSON format
- **Expanding…** more transformations to come

Job authoring: Developer endpoints



- Environment to **iteratively develop** and test ETL code.
- Connect your IDE or notebook (e.g. **Zeppelin**) to a Glue development endpoint.
- When you are satisfied with the results you can create an ETL job that runs your code.



https://www.cloudera.com/products/open-source/apache-hadoop/apache-zeppelin.html

Job execution: Scheduling and monitoring

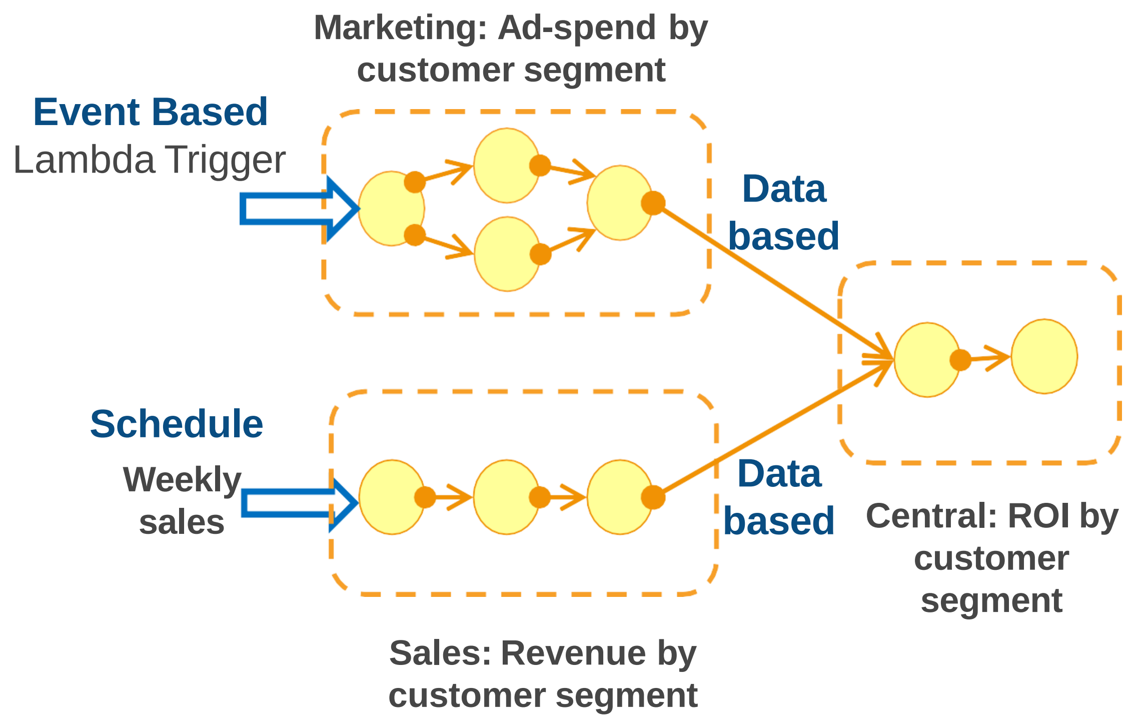
Compose jobs globally with event-based dependencies

- Easy to reuse and leverage work across organization boundaries

Multiple triggering mechanisms

- **Schedule-based:** e.g., time of day
- **Event-based:** e.g., job completion
- **On-demand:** e.g., AWS Lambda
- **More coming soon:** Data Catalog based events, S3 notifications and Amazon CloudWatch events

Logs and alerts are available in Amazon CloudWatch

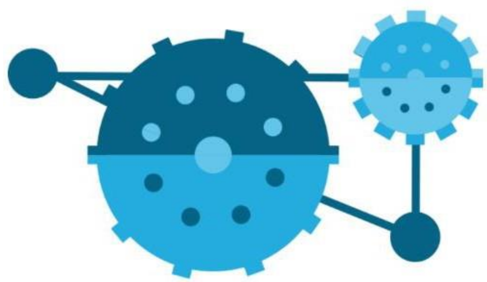


QuickSight for BI on Redshift

Amazon QuickSight allows everyone in your organization to understand your data by asking questions in natural language, exploring through interactive dashboards, or automatically looking for patterns and outliers powered by machine learning.

Amazon Redshift

Achieve high concurrency by offloading end user queries to SPICE



Calculations can be done in SPICE reducing the load on the underlying database

https://aws.amazon.com/quicksight/

Apache Flume

What is Flume

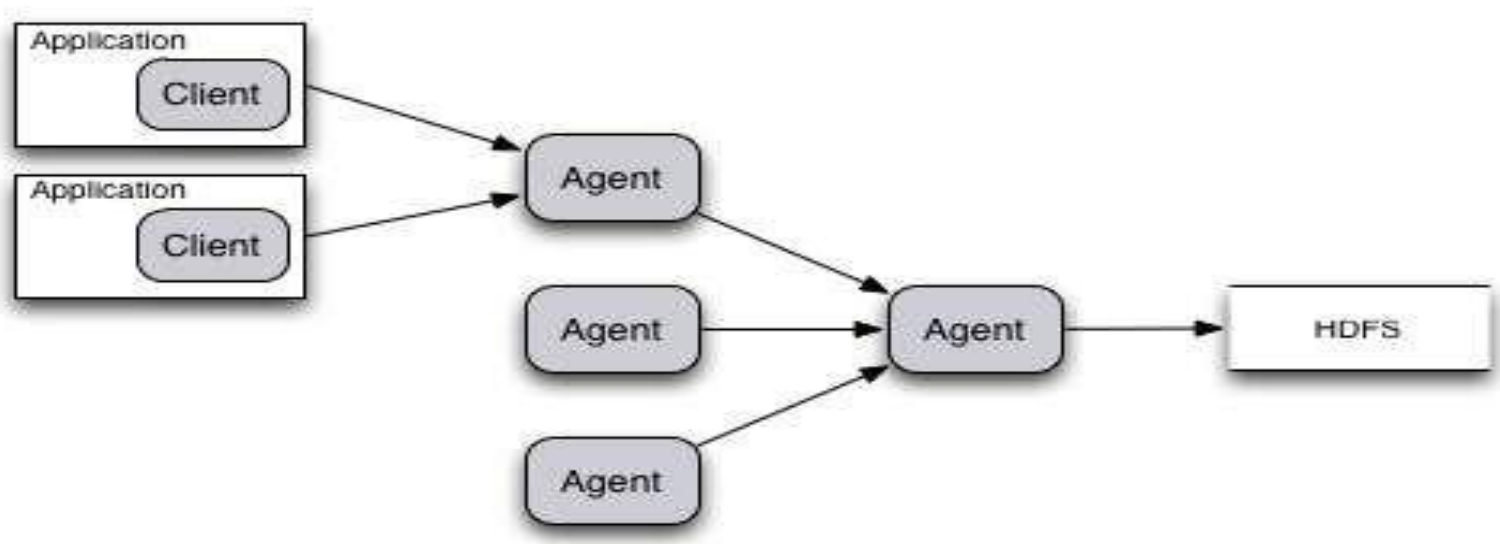
- Collection, Aggregation of streaming Event Data
  - Typically used for log data
- Significant advantages over ad-hoc solutions
  - Reliable, Scalable, Manageable, Customizable and High Performance
  - Declarative, Dynamic Configuration
  - Contextual Routing
  - Feature rich
  - Fully extensible

Core Concepts: Event

An Event is the fundamental unit of data transported by Flume from its point of origination to its final destination. Event is a byte array payload accompanied by optional headers.

- Payload is opaque to Flume
- Headers are specified as an unordered collection of string key-value pairs, with keys being unique across the collection
- Headers can be used for contextual routing

Typical Aggregation Flow



[Client]⁺ → Agent [→ Agent]* → Destination

CLOUDERA

Core Concepts: Channel

A passive component that buffers the incoming events until they are drained by Sinks.

- Different Channels offer different levels of persistence:
  - Memory Channel: volatile
    - Data lost if JVM or machine restarts
  - File Channel: backed by WAL implementation
    - Data not lost unless the disk dies.
    - Eventually, when the agent comes back data can be accessed.

Core Concepts: Sink

An active component that removes events from a Channel and transmits them to their next hop destination.

- Different types of Sinks:
  - Terminal sinks that deposit events to their final destination. For example: HDFS, HBase, Morphline-Solr, Elastic Search
  - Sinks support serialization to user's preferred formats.
  - HDFS sink supports time-based and arbitrary bucketing of data while writing to HDFS.
  - IPC sink for Agent-to-Agent communication: Avro, Thrift
- Require exactly one channel to function

Flume Agent

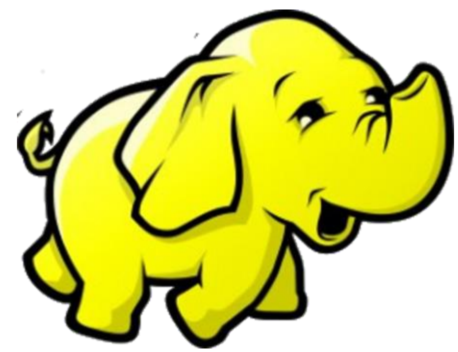
- Installed on each node
- Collects events

Flume Agent

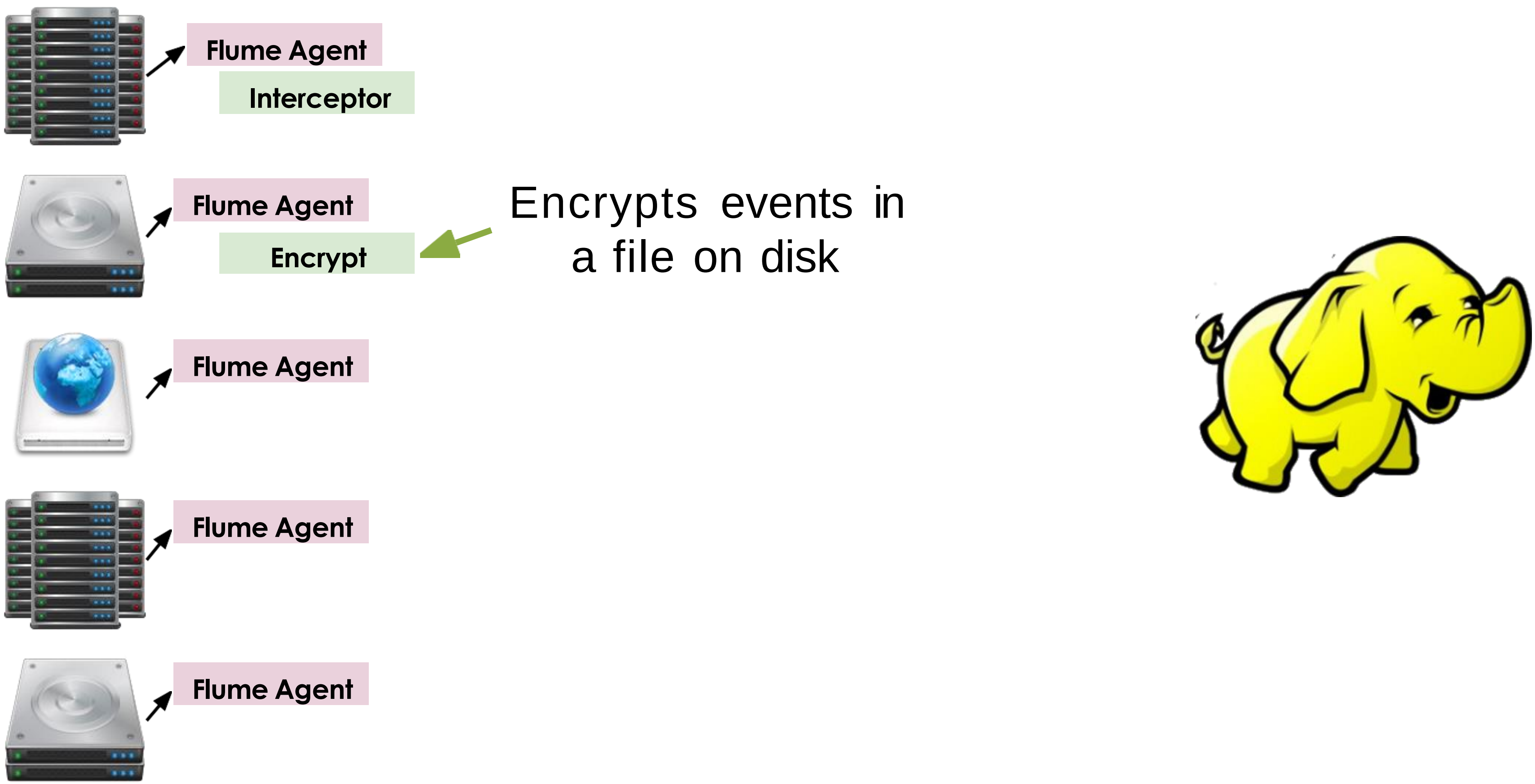
Flume Agent

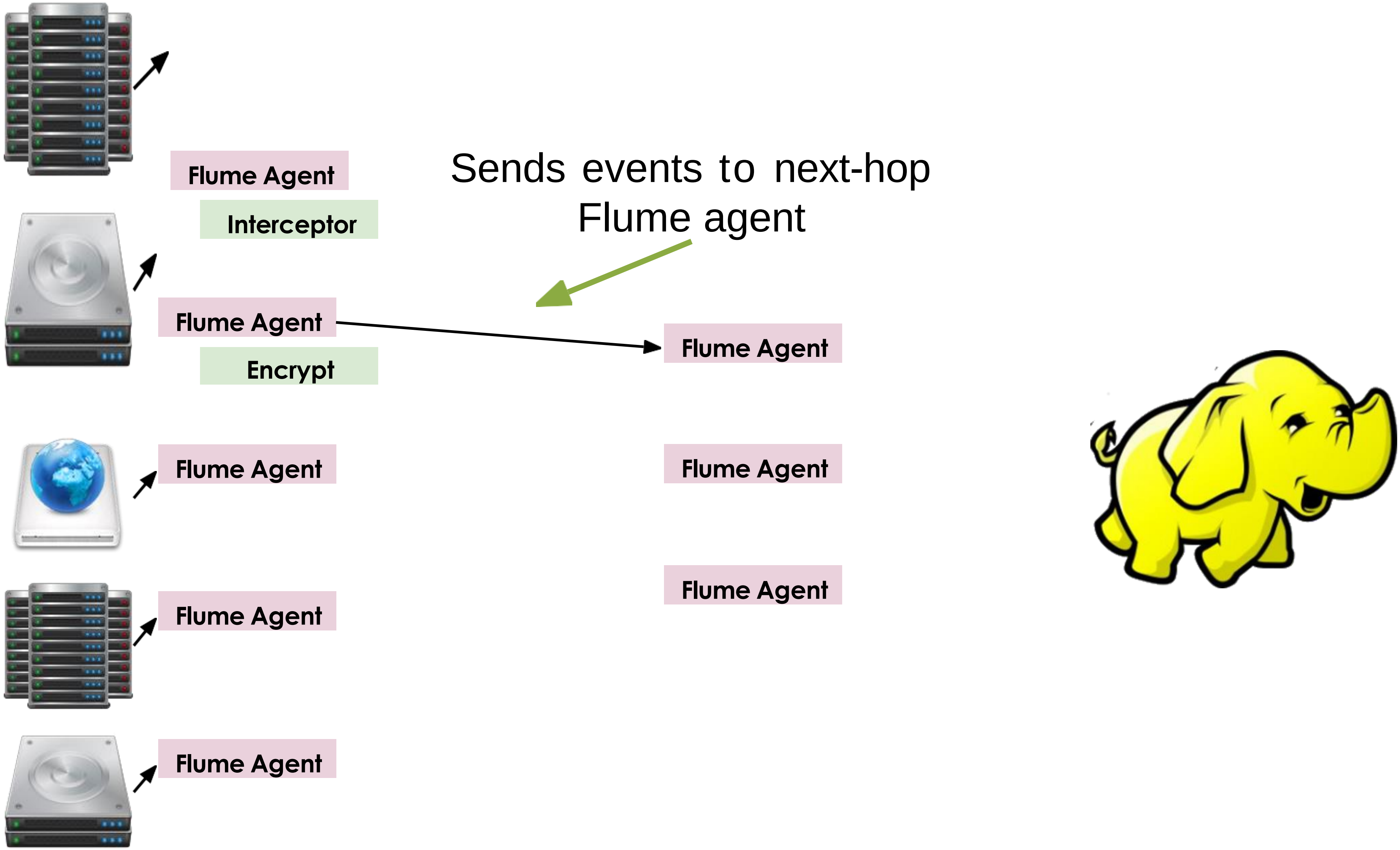
Flume Agent

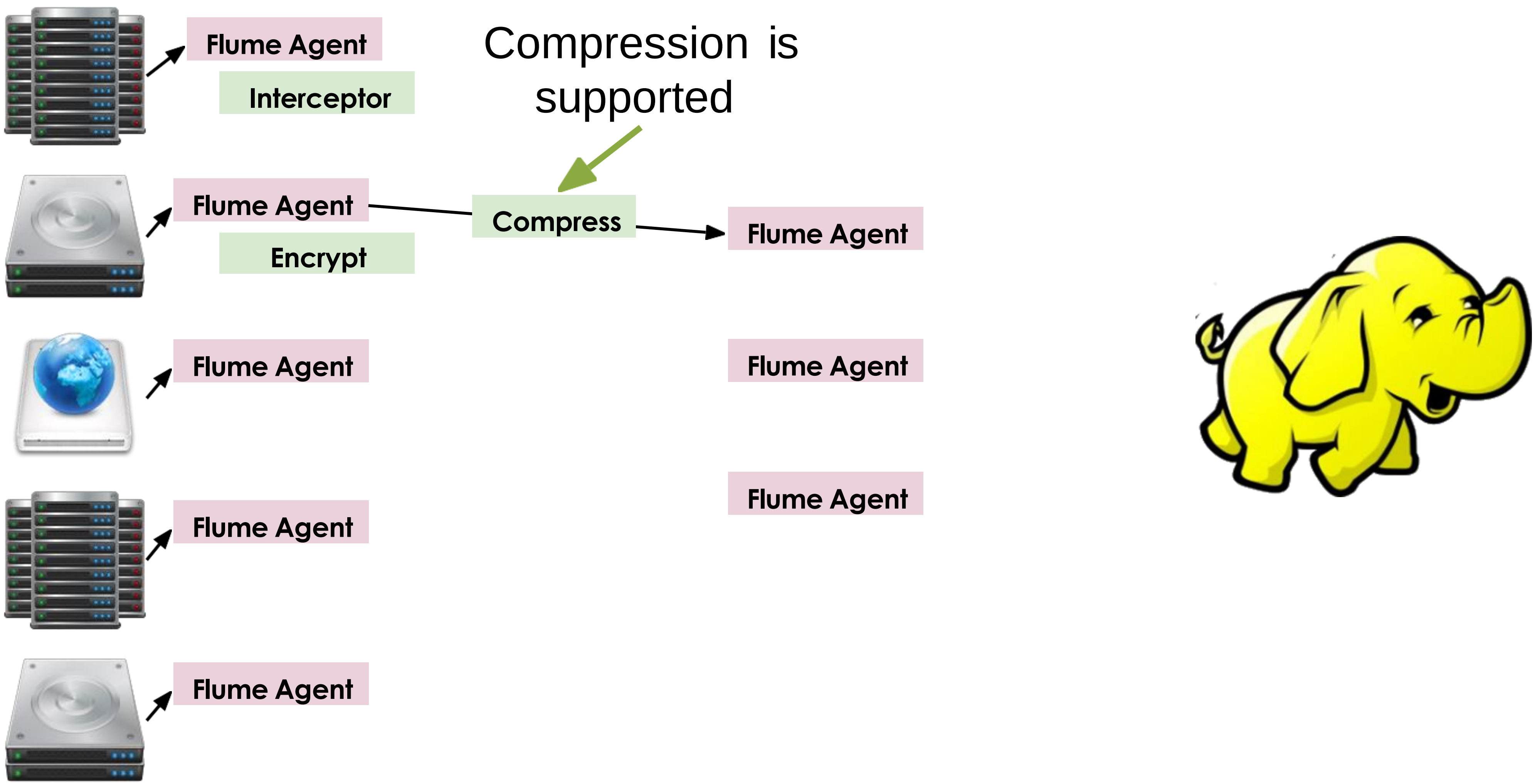
Flume Agent



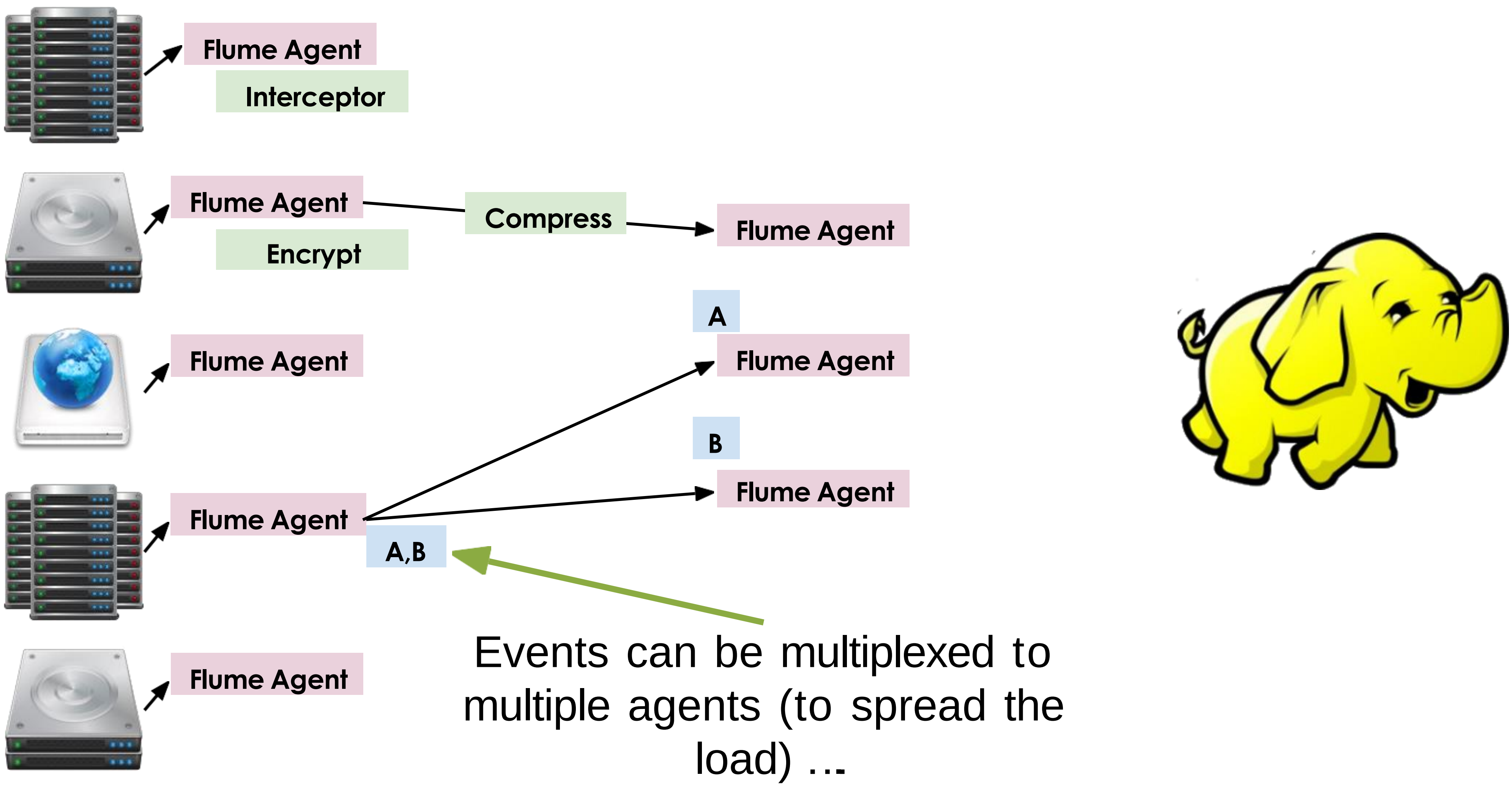
- Filters useless events
- Decorates events by adding metadata
  - e.g. timestamp, hostname, UUID, static markers

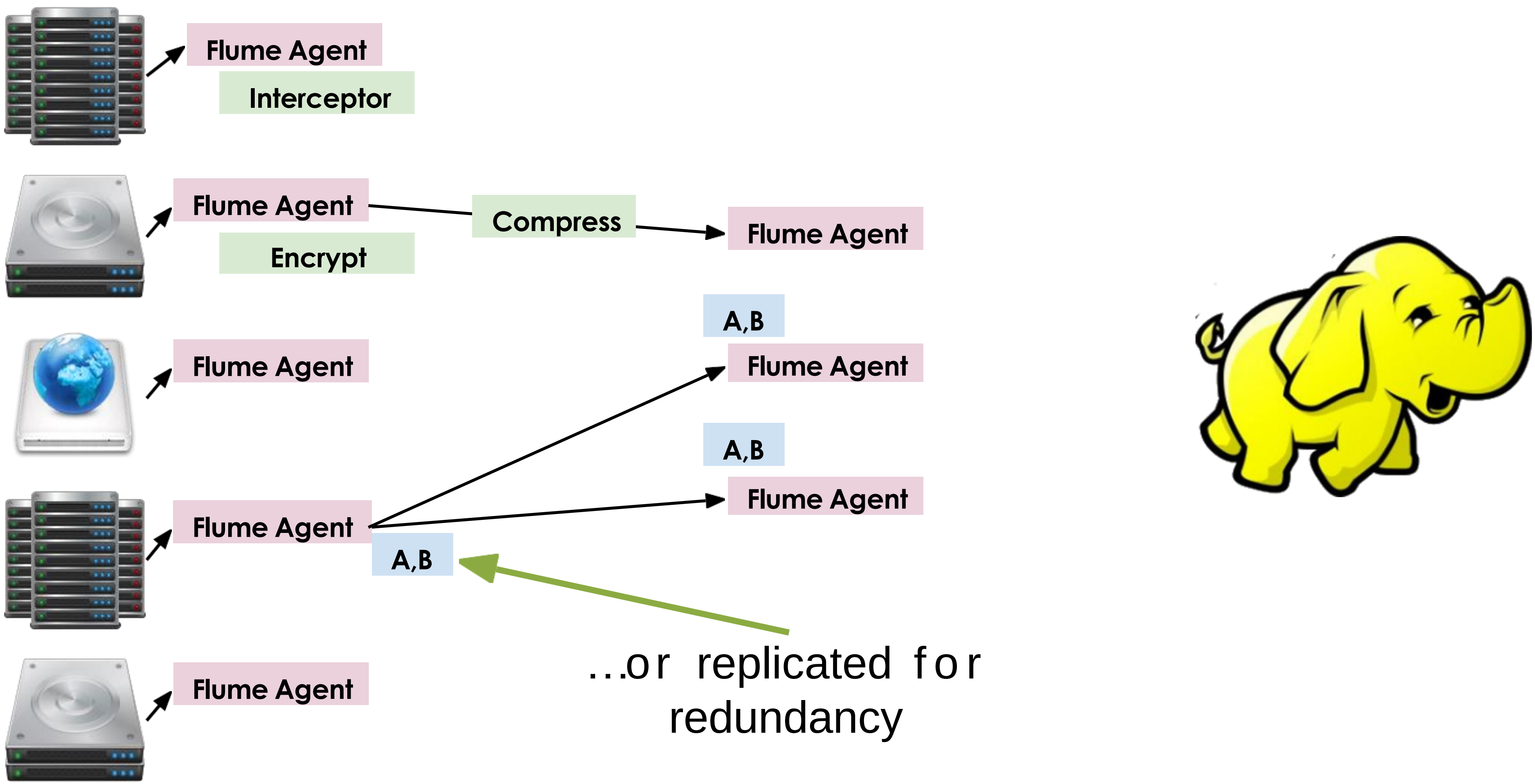

Flume Agent
Interceptor
Flume Agent
Encrypt
Encrypts events in a file on disk
Flume Agent
Flume Agent
Flume Agent


Sends events to next-hop Flume agent
Flume Agent
Interceptor
Flume Agent
Encrypt
Flume Agent
Flume Agent
Flume Agent
Flume Agent
Flume Agent
Flume Agent

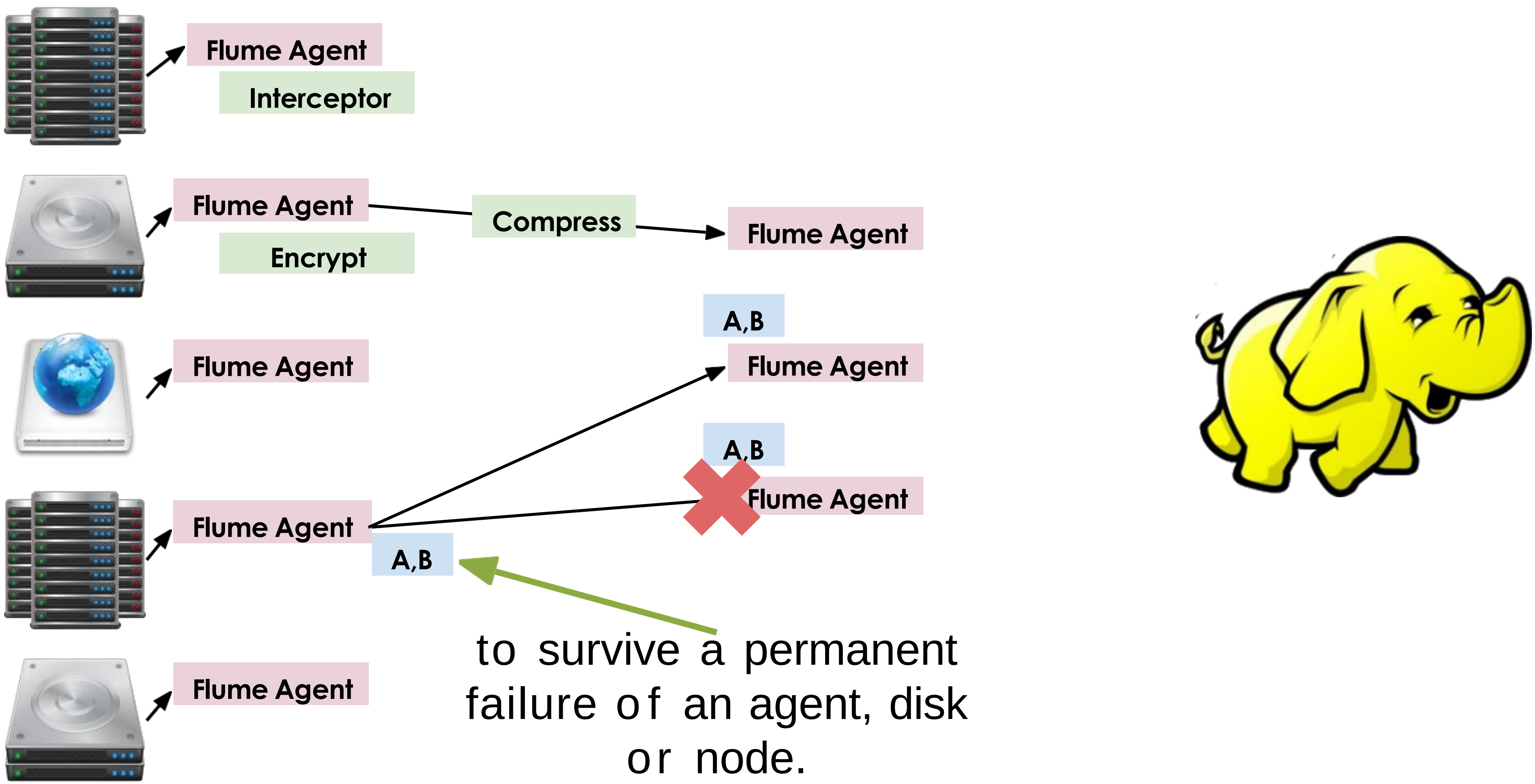

Flume Agent
Interceptor
Compression is supported
Flume Agent
Compress
Flume Agent
Encrypt
Flume Agent
Flume Agent
Flume Agent
Flume Agent
Flume Agent

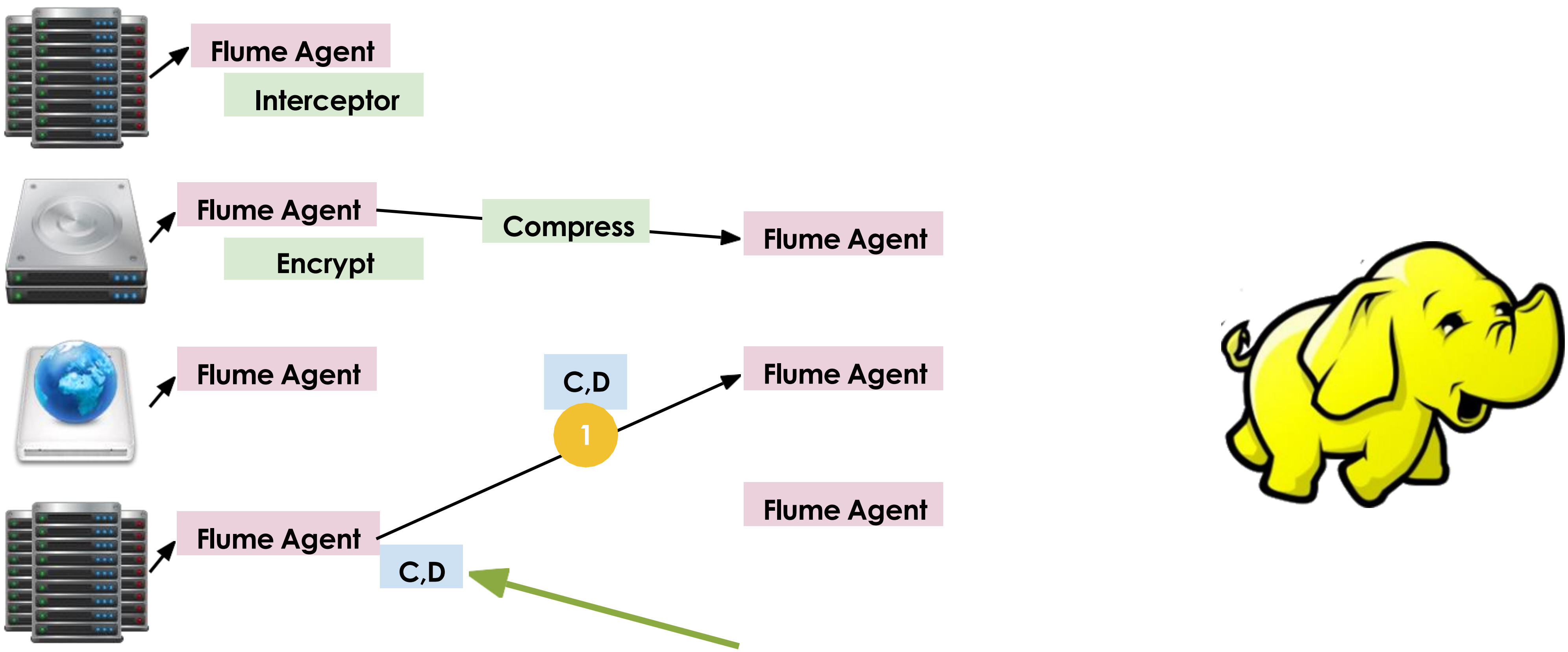
CLOUDERA


Flume Agent
Interceptor
Flume Agent
Encrypt
Compress
Flume Agent
Flume Agent
A
Flume Agent
B
Flume Agent
Flume Agent
A,B
Flume Agent
Events can be multiplexed to multiple agents (to spread the load) ...

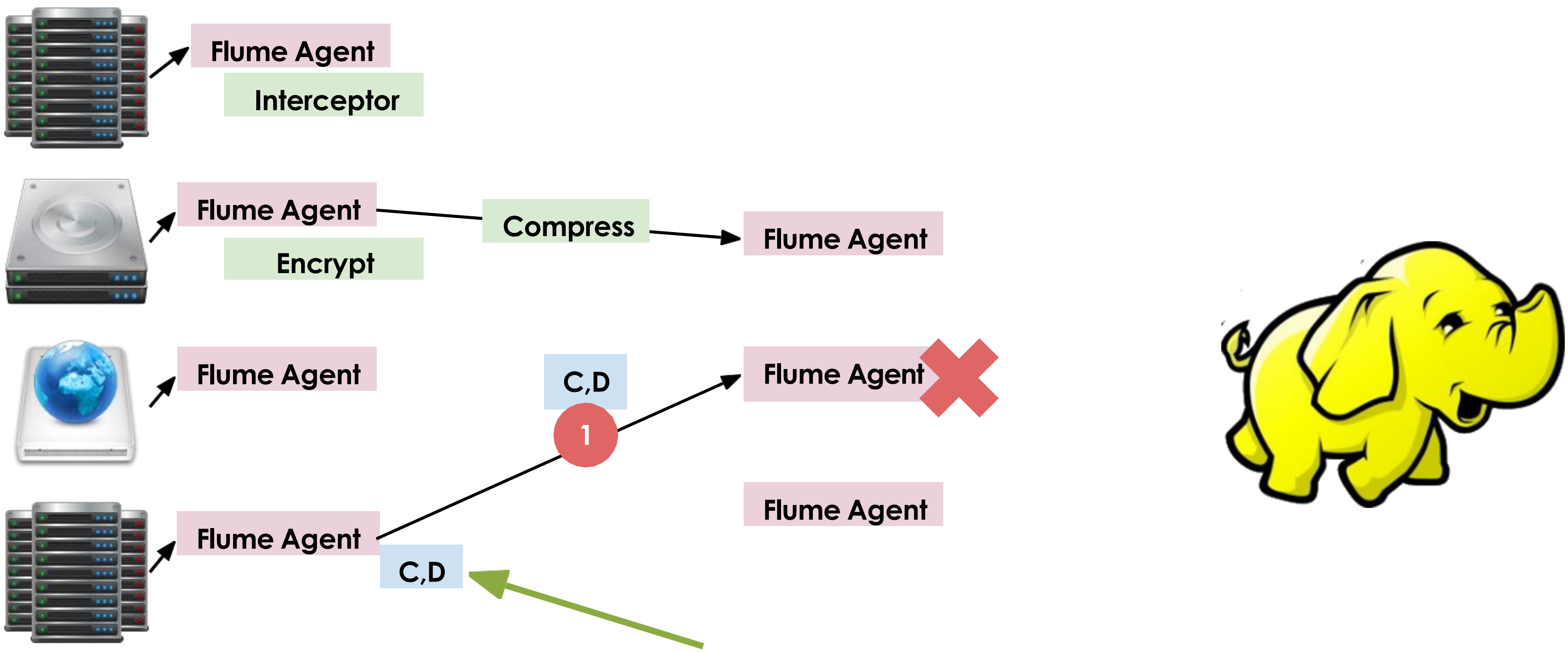


...or replicated for redundancy


Flume Agent
Interceptor
Flume Agent
Encrypt
Compress
Flume Agent
Flume Agent
A,B
Flume Agent
A,B
Flume Agent
Flume Agent
A,B
Flume Agent
to survive a permanent failure of an agent, disk or node.

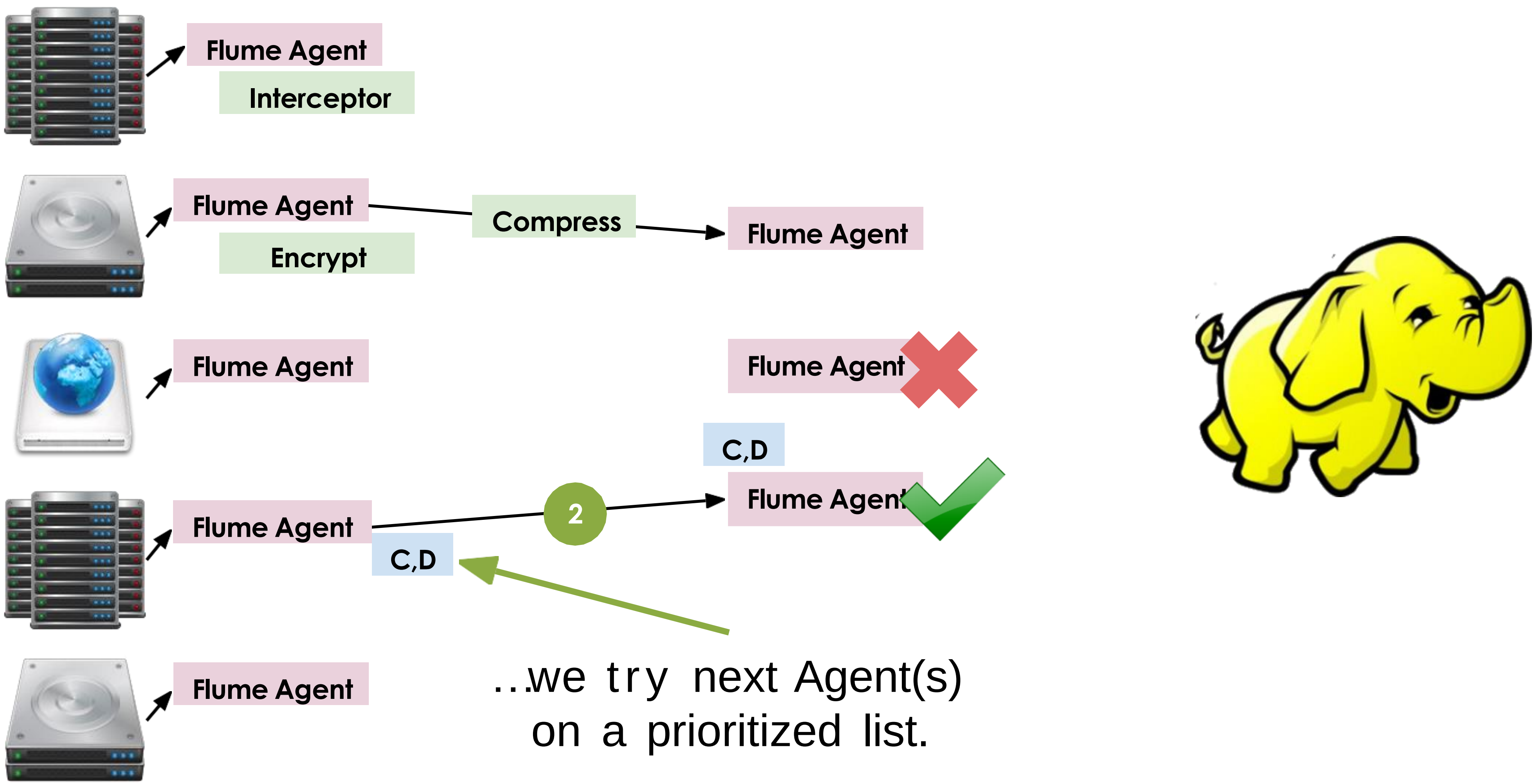


Events can be also delivered in “failover” mode where …

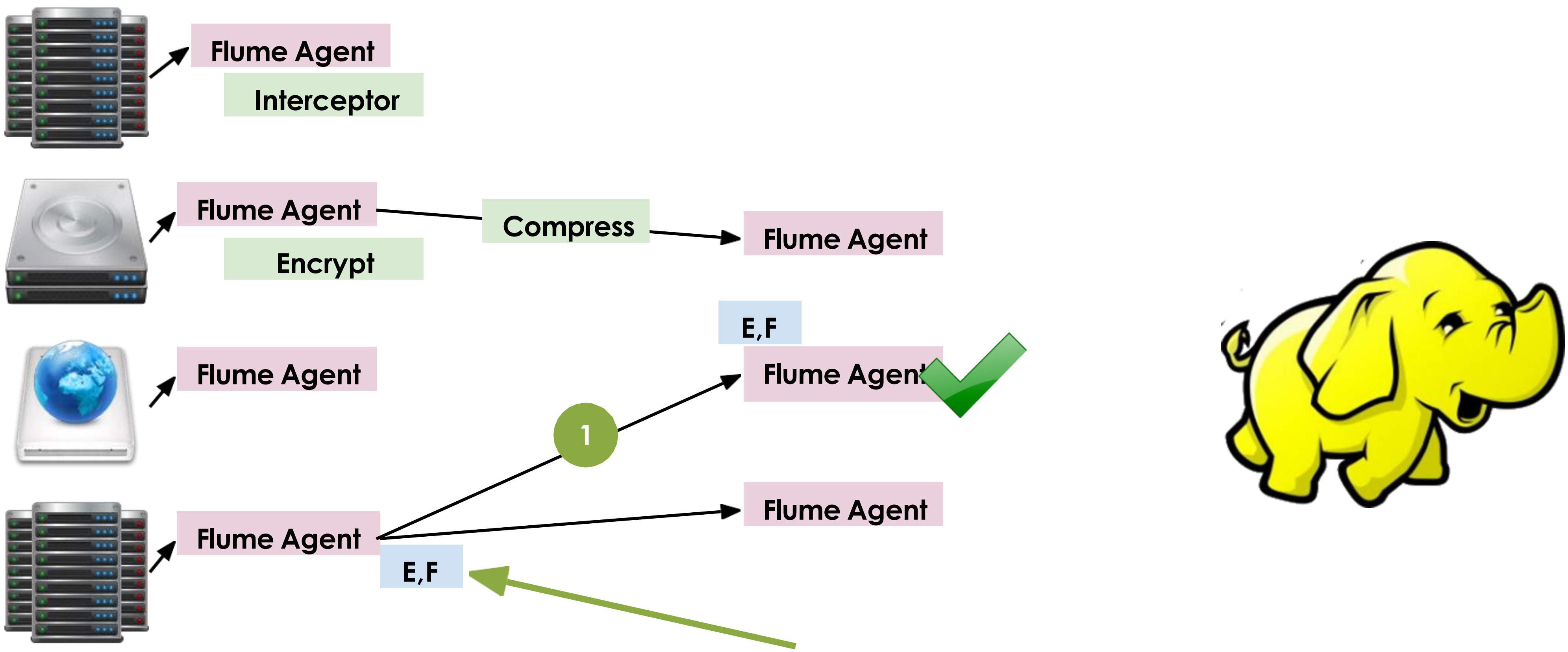


… in case of a failure of the next-hop agent …

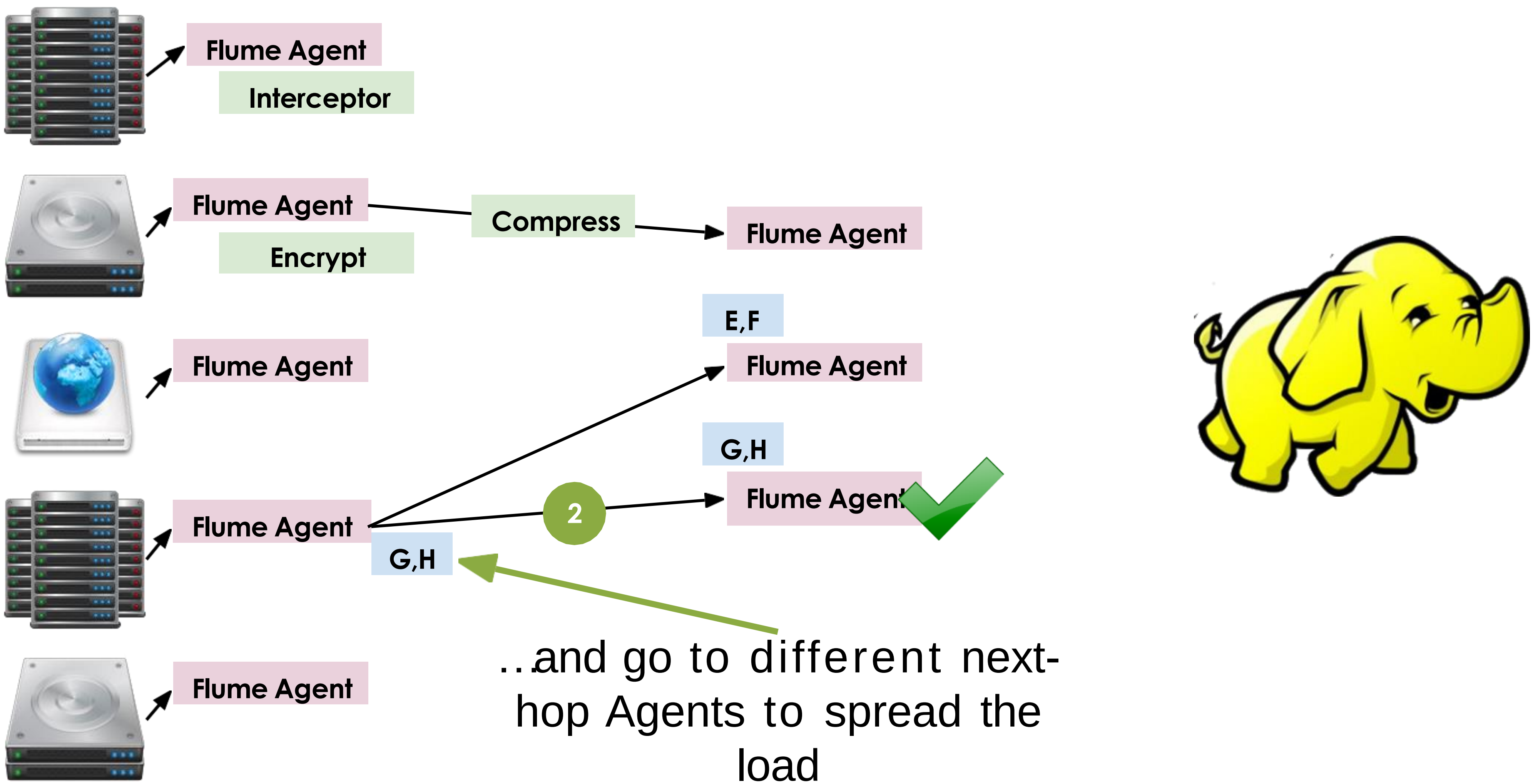
CLOUDERA

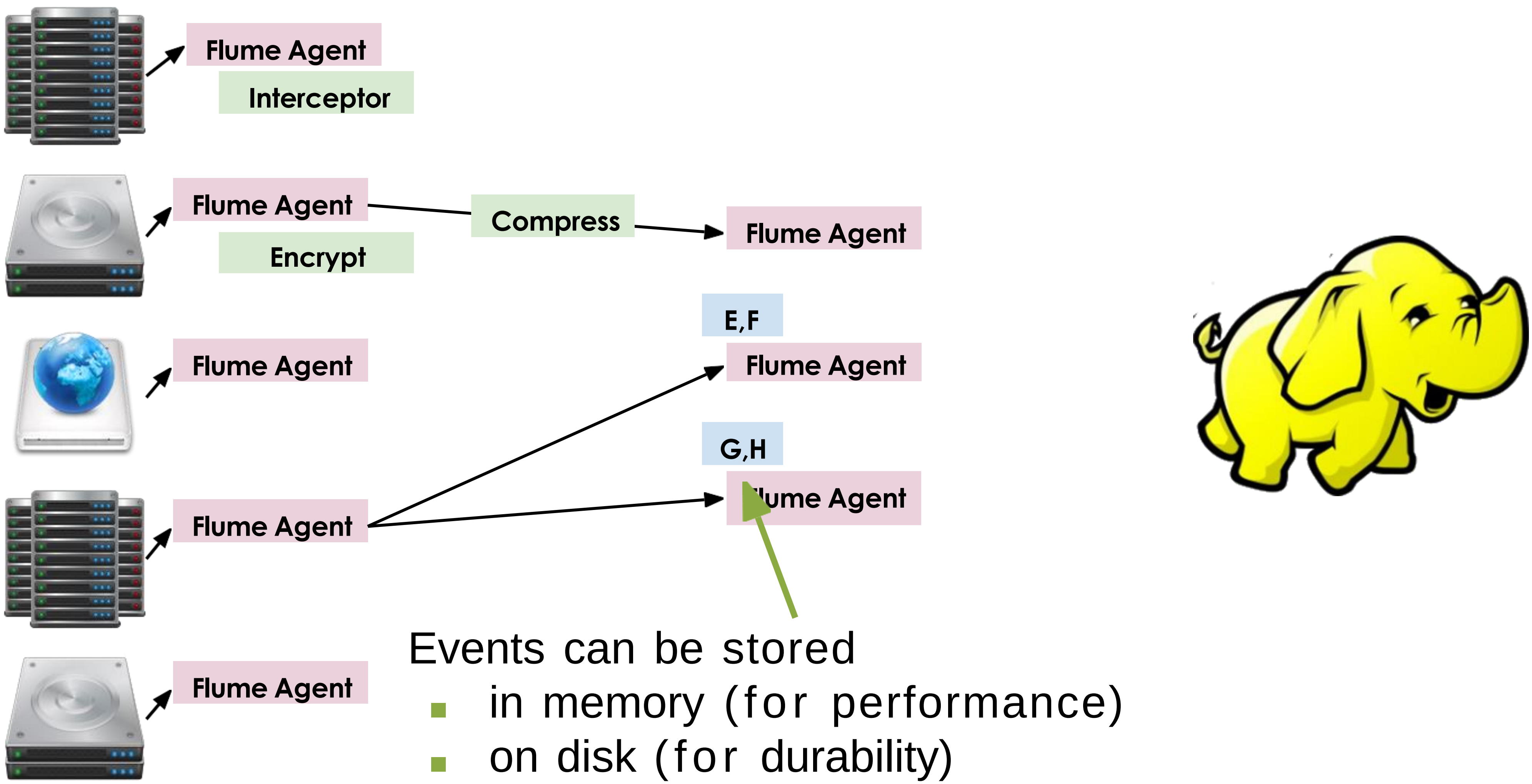

Flume Agent
Interceptor
Flume Agent
Encrypt
Compress
Flume Agent
Flume Agent
Flume Agent
C,D
Flume Agent
2
Flume Agent
C,D
Flume Agent
...we try next Agent(s)
on a prioritized list.

CLOUDERA



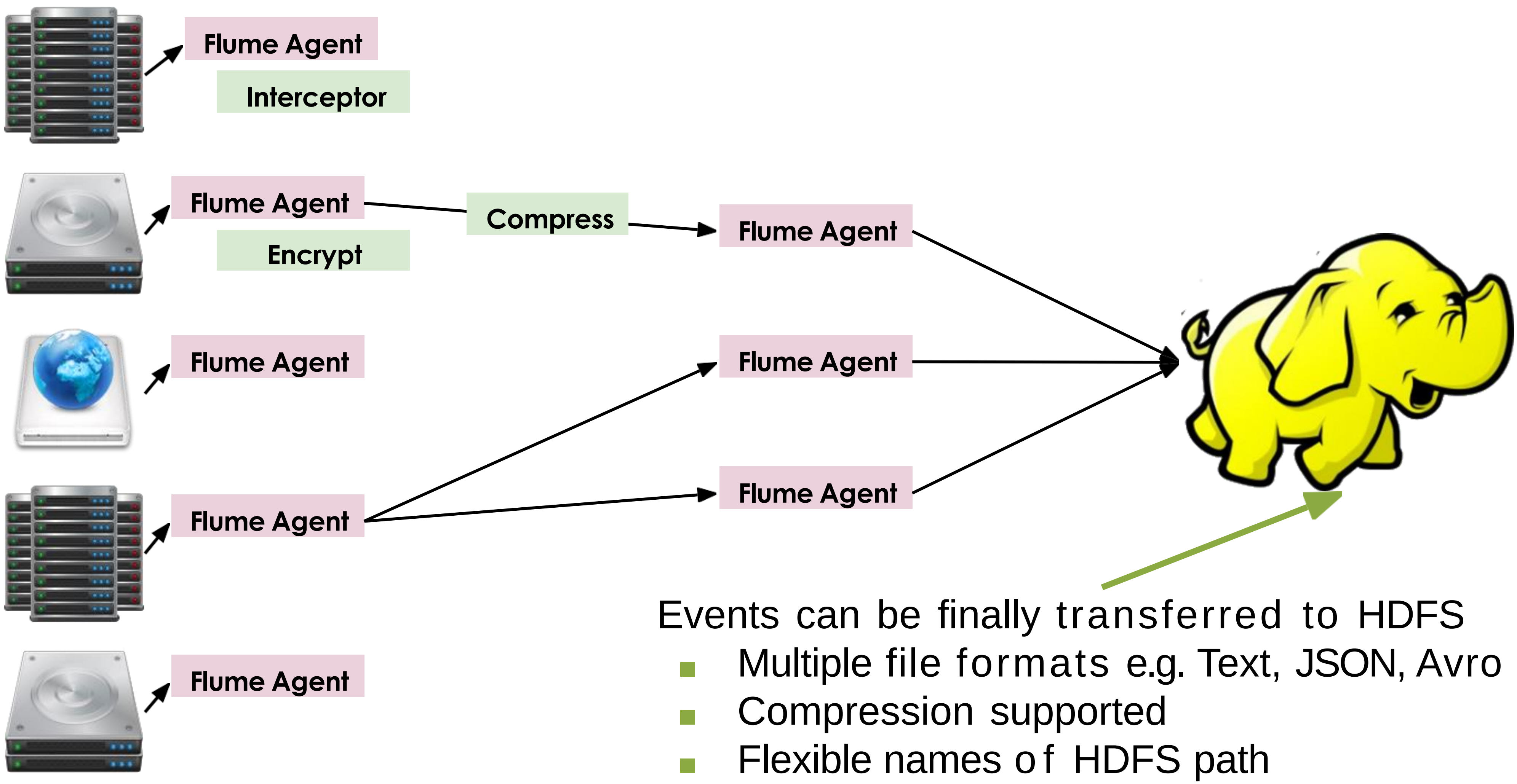
Events can be also load-balanced (round robin, random and custom) …


Flume Agent
Interceptor
Flume Agent
Encrypt
Compress
Flume Agent
Flume Agent
E,F
Flume Agent
G,H
Flume Agent
2
Flume Agent
G,H
Flume Agent
...and go to different next-hop Agents to spread the load



Events can be stored

- in memory (for performance)
- on disk (for durability)

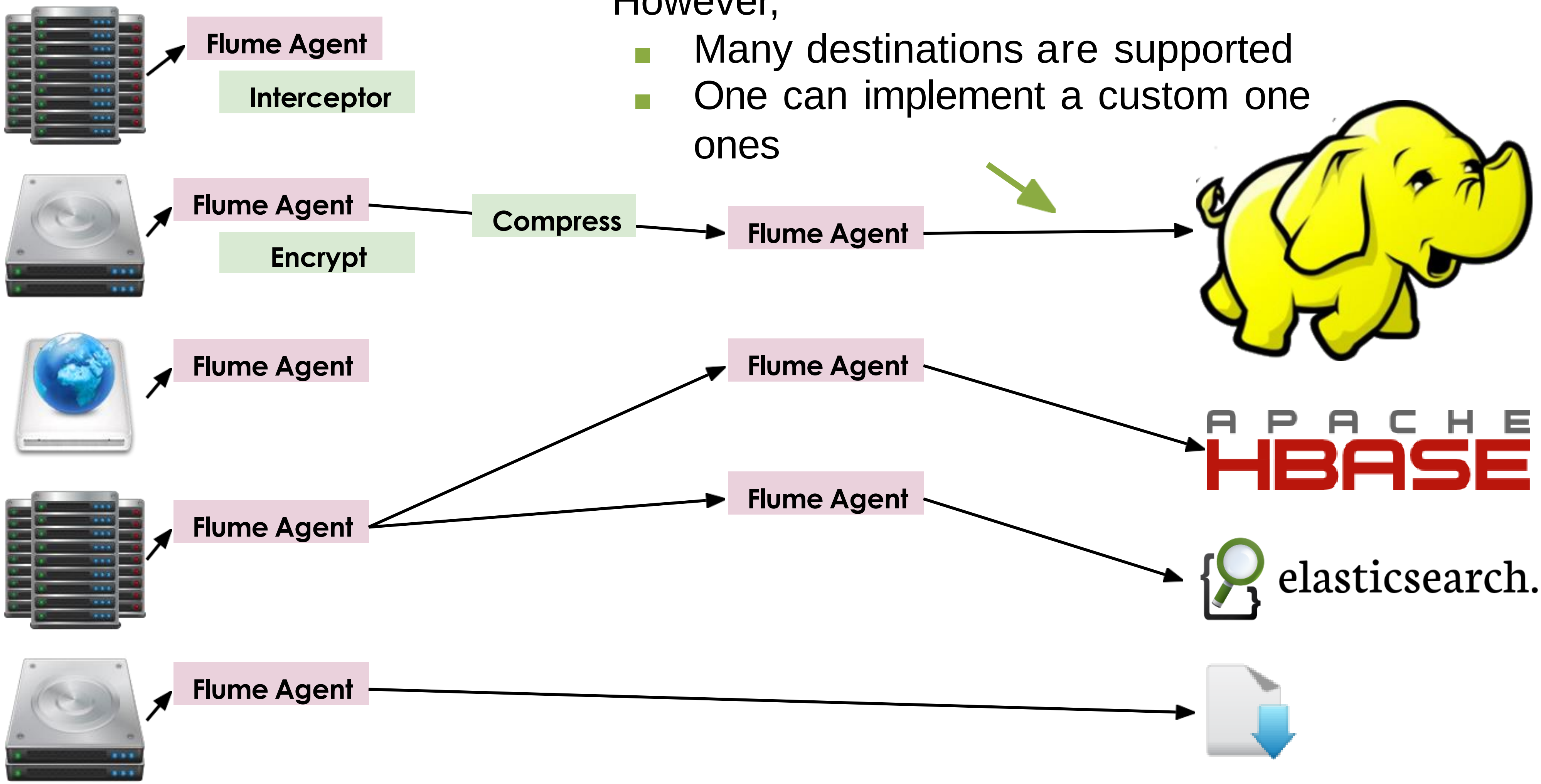


Events can be finally transferred to HDFS

- Multiple file formats e.g. Text, JSON, Avro
- Compression supported
- Flexible names of HDFS path

However,

- Many destinations are supported
- One can implement a custom one ones



Flume

- **Distributed**
  - Agents installed on many machines
- **Scalable**
  - Add more machines to transfer more events
- **Reliable**
  - Durable storage, failover and/or replication
- **Manageable**
  - Easy to install, configure, reconfigure and run

Flume

- **Nicely integrated with the Hadoop Ecosystem**
  - Various destinations e.g. HDFS, HBase
  - Various file formats e.g. Avro, SequenceFile
- **Extensible**
  - Possibility to add new functionality e.g. source and destination for events

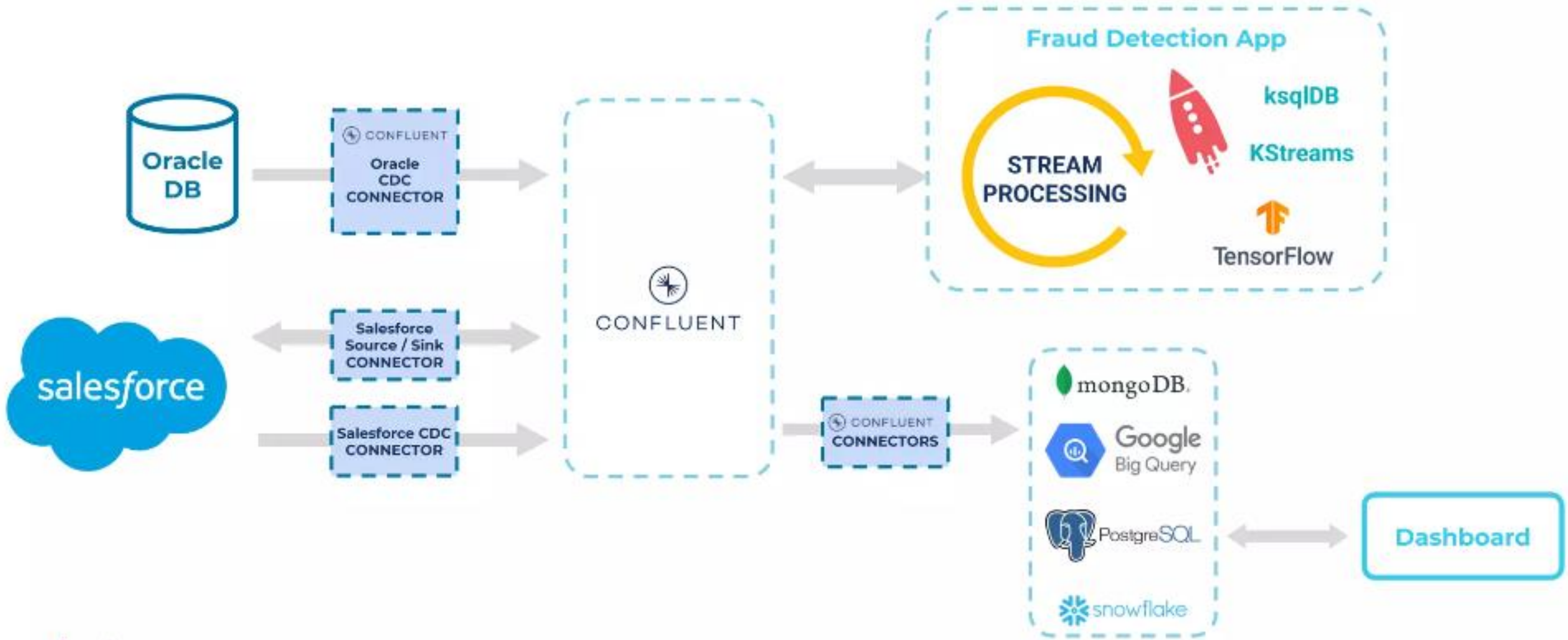
An Event Streaming Platform

The Underpinning of Data in Motion

Kai Wahner, Technology Evangelist at Confluent

Example Architecture for Data in Motion

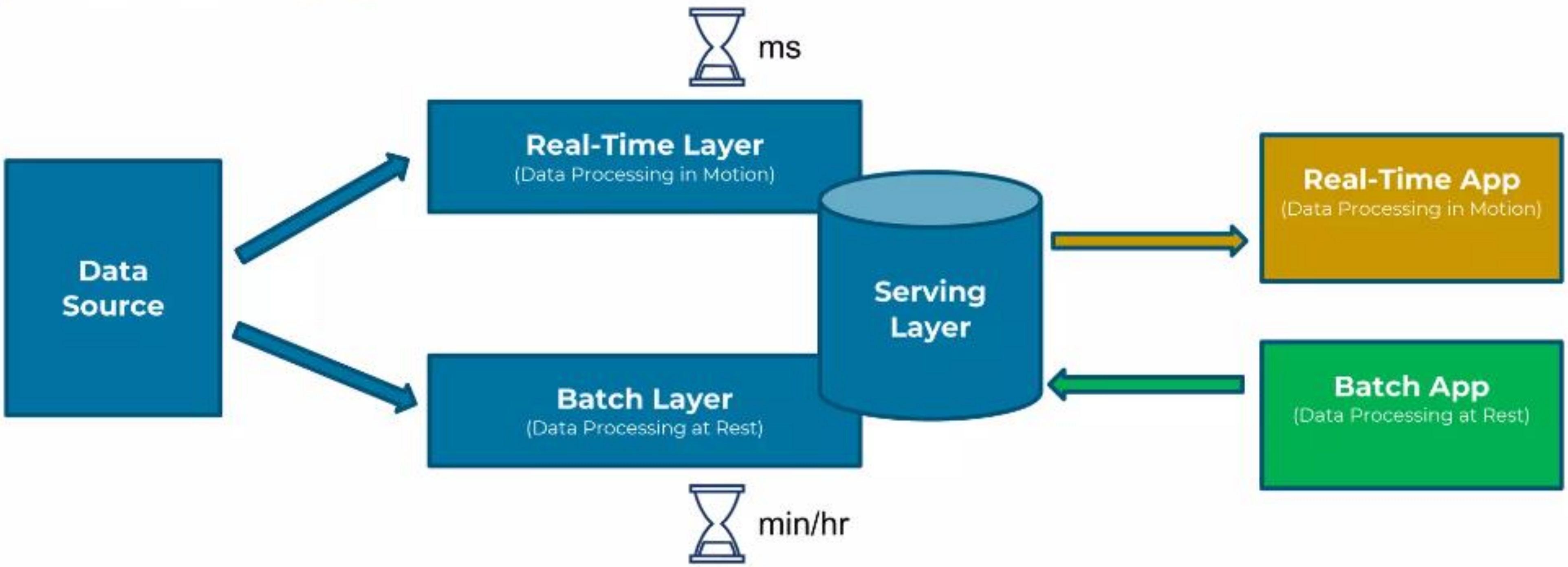
Real-time decision making for claim processing and fraud detection



Kai Wahner, Technology Evangelist at Confluent

Lambda Architecture

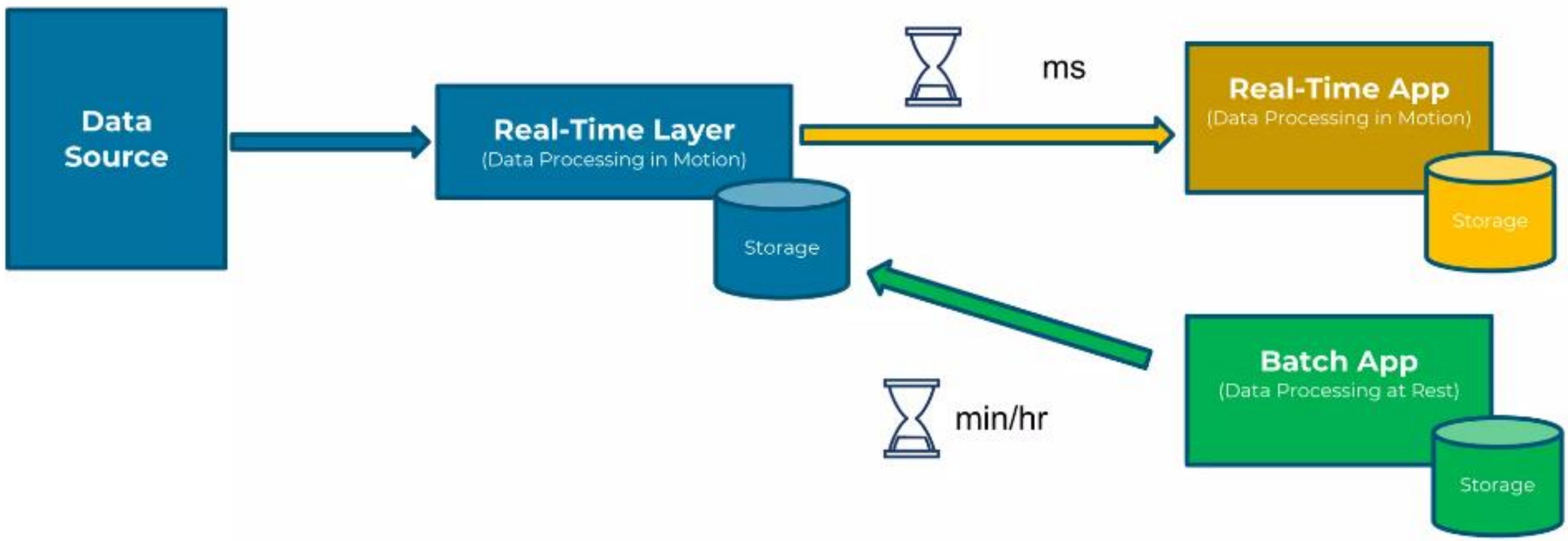
Option 1: Unified serving layer



Kai Wahner, Technology Evangelist at Confluent

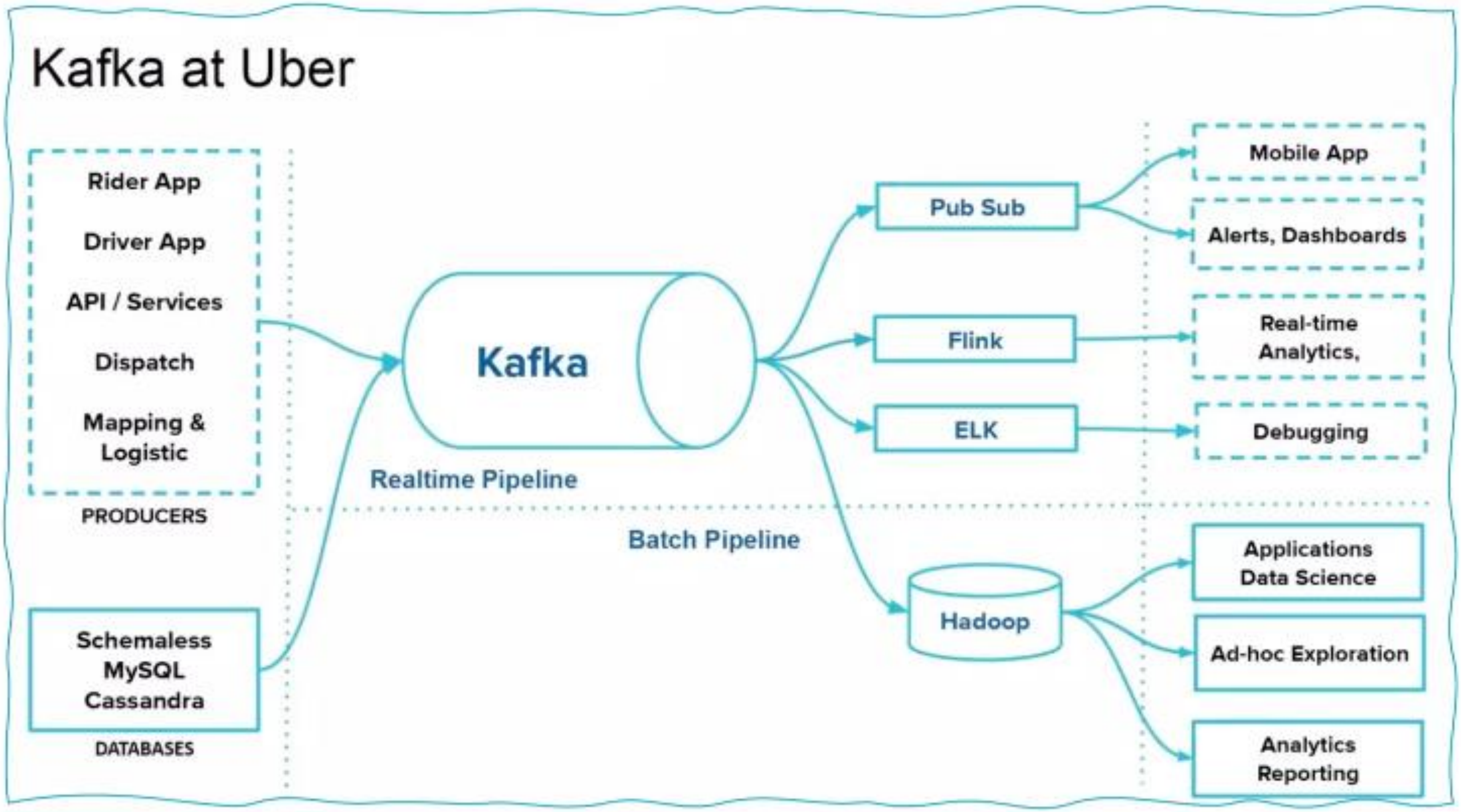
Kappa Architecture

One pipeline for real-time and batch consumers



Kai Wahner, Technology Evangelist at Confluent

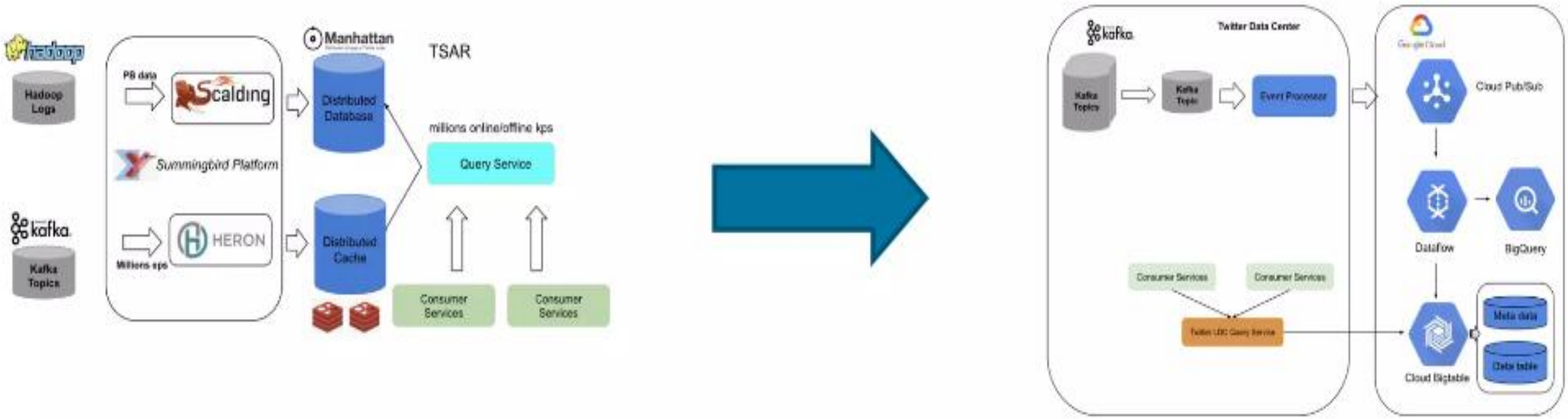
Kappa @ Uber



Flink ML is a library which provides machine learning (ML) APIs and infrastructures that simplify the building of ML pipelines

Kai Wahner, Technology Evangelist at Confluent

Kappa @ Twitter

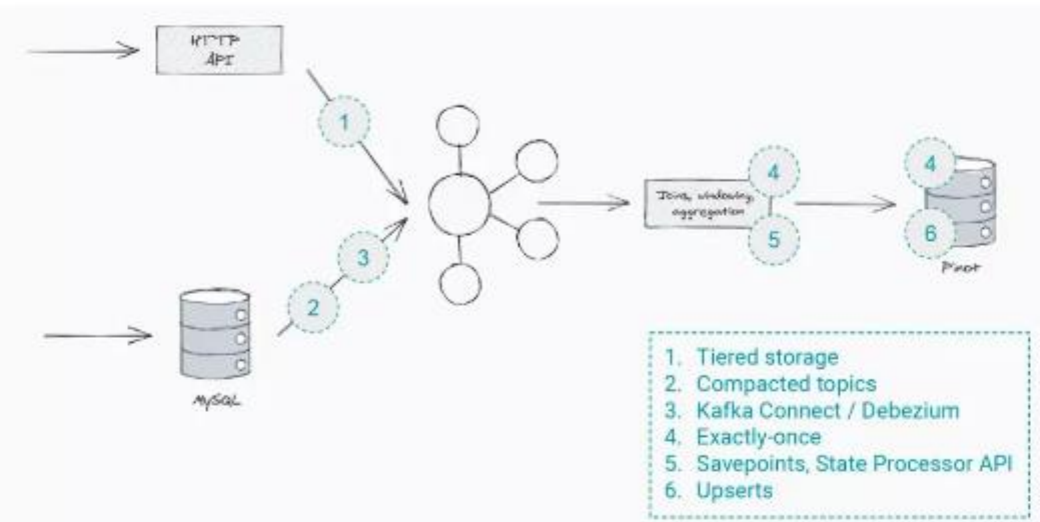


Migration from Hadoop and Kafka to a hybrid architecture on both Twitter data center and Google Cloud Platform with Kafka and GCP, Twitter is able to process billions of events in real-time and achieve low latency, high accuracy, stability, architecture simplicity, and reduced operation cost

Kai Wahner, Technology Evangelist at Confluent

Kappa @ Shopify

shopify



Kappa Building Blocks

The Log (Kafka)

Durability with Topic Compaction and Tiered Storage
Consistency via Exactly-Once Semantics (EOS)
Data Integration via Kafka Connect
Elasticity via dynamic Kafka clusters

Streaming Framework (Kafka Streams / Flink)

Reliability and scalability
Fault tolerance
State management

Sinks

Update/Upsert for simplified design:
RDBMS, NoSQL, Compacted Kafka Topics
Append-only: Regular Kafka Topics, Time Series

Kai Wahner, Technology Evangelist at Confluent