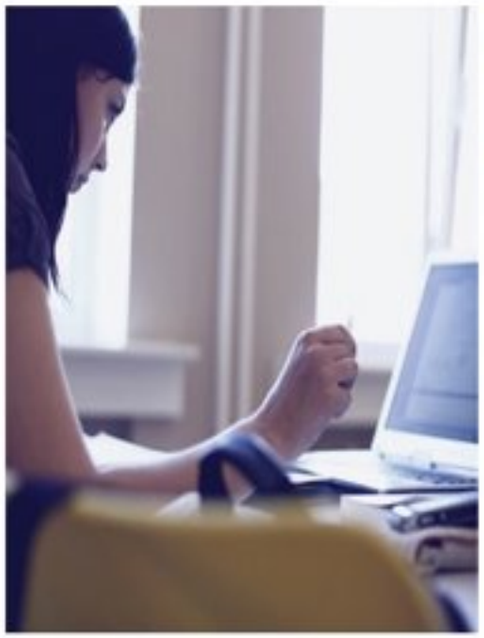
INF700
IT for Business & Management







BIS Infrastructure

Applications Portfolio Deployment & e-Business Applications

IT Infrastructure: Starts with Governance

IT Governance– Framework for decision rights and accountability to encourage desirable behavior in the use of IT.
Governance complements organizational structure to enable a firm to meet conflicting objectives. (*More - 15 September Lecture*)

Five key IT decisions need to be governed

Principles for IT	High level statements about how IT is to be used. Driven by business principles (e.g., operating model)
Enterprise Architecture	Organizing logic for data, applications, and infrastructure captured in a set of policies, relationships, and technical choices to achieve desired business and technical standardization and integration
IT Infrastructure Strategies	Strategies for shared IT capability (both technical and human) delivered as reliable services (e.g., network, help desk, shared data)
Business Application Needs	Specifying the business need for purchased or internally developed IT applications
IT Investment and Prioritization	Decisions about how much and where to invest in IT including project approvals and justification techniques

Operating Models & Infrastructure

Operating Model_– the desired level of business process integration and business process standardization for delivering goods and services to customers. It describes how a firm will profit and grow.

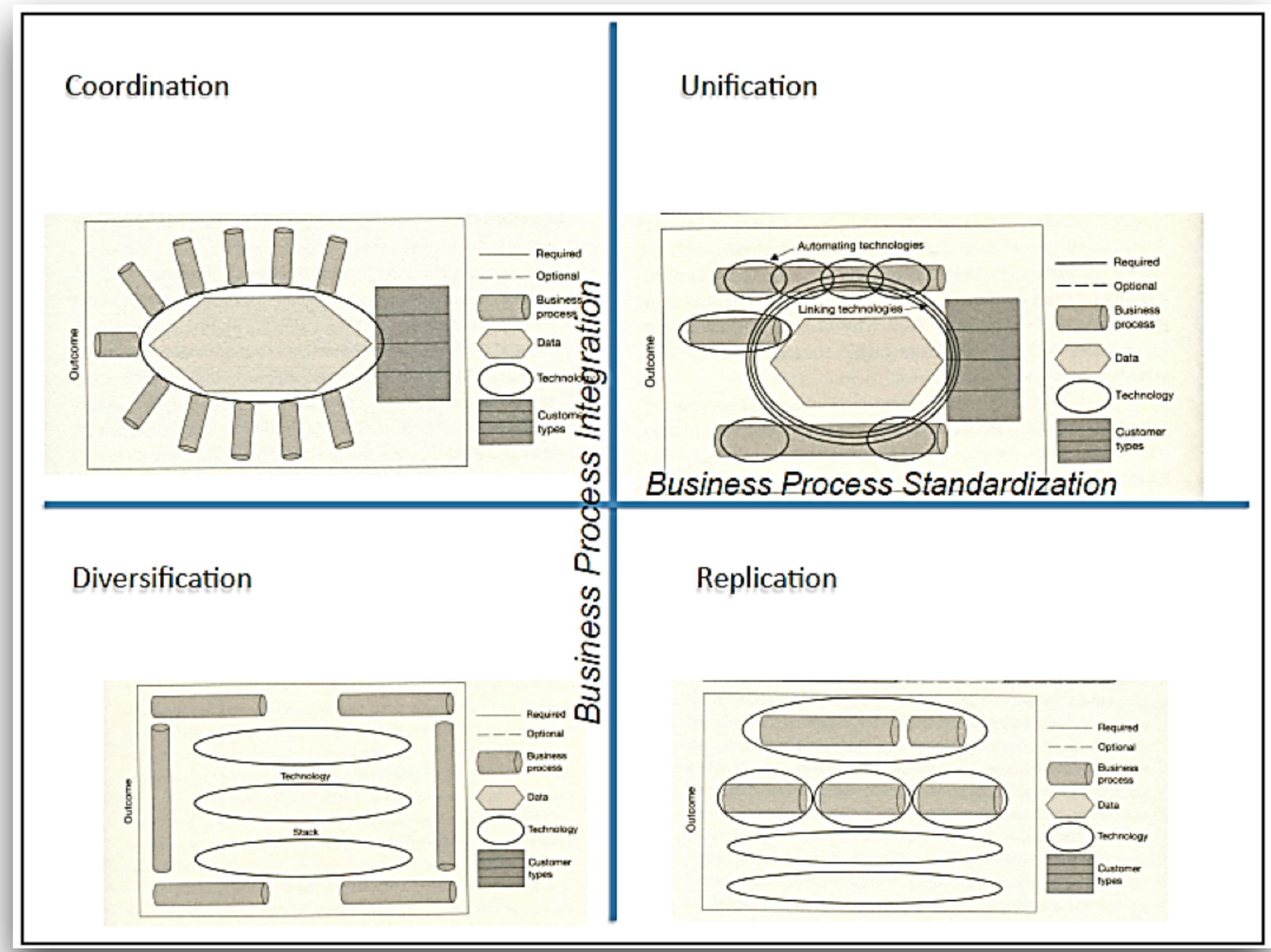
Figure 1: Characteristics of Four Operating Models

Business Process Integration		Business Process Standardization: Low	Business Process Standardization: High
	High	***Coordination*** ▪ Shared customers, products or suppliers ▪ Impact on other business unit transactions ▪ Operationally unique business units or functions ▪ Autonomous business management ▪ Business unit control over business process design ▪ Shared customer/supplier/product data ▪ Consensus processes for designing IT infrastructure services; IT application decisions are made in business units	***Unification*** ▪ Customers and suppliers may be local or global ▪ Globally integrated business processes often with support of enterprise systems ▪ Business units with similar or overlapping operations ▪ Centralized management often applying functional/process/business unit matrices ▪ High-level process owners design standardized process ▪ Centrally mandated databases ▪ IT decisions made centrally
	Low	***Diversification*** ▪ Few, if any, shared customers or suppliers ▪ Independent transactions ▪ Operationally unique business units ▪ Autonomous business management ▪ Business unit control over business process design ▪ Few data standards across business units ▪ Most IT decisions made within business units.	***Replication*** ▪ Few, if any, shared customers ▪ Independent transactions aggregated at a high level ▪ Operationally similar business units ▪ Autonomous business unit leaders with limited discretion over processes ▪ Centralized (or federal) control over business process design ▪ Standardized data definitions but data locally owned with some aggregation at corporate ▪ Centrally mandated IT services

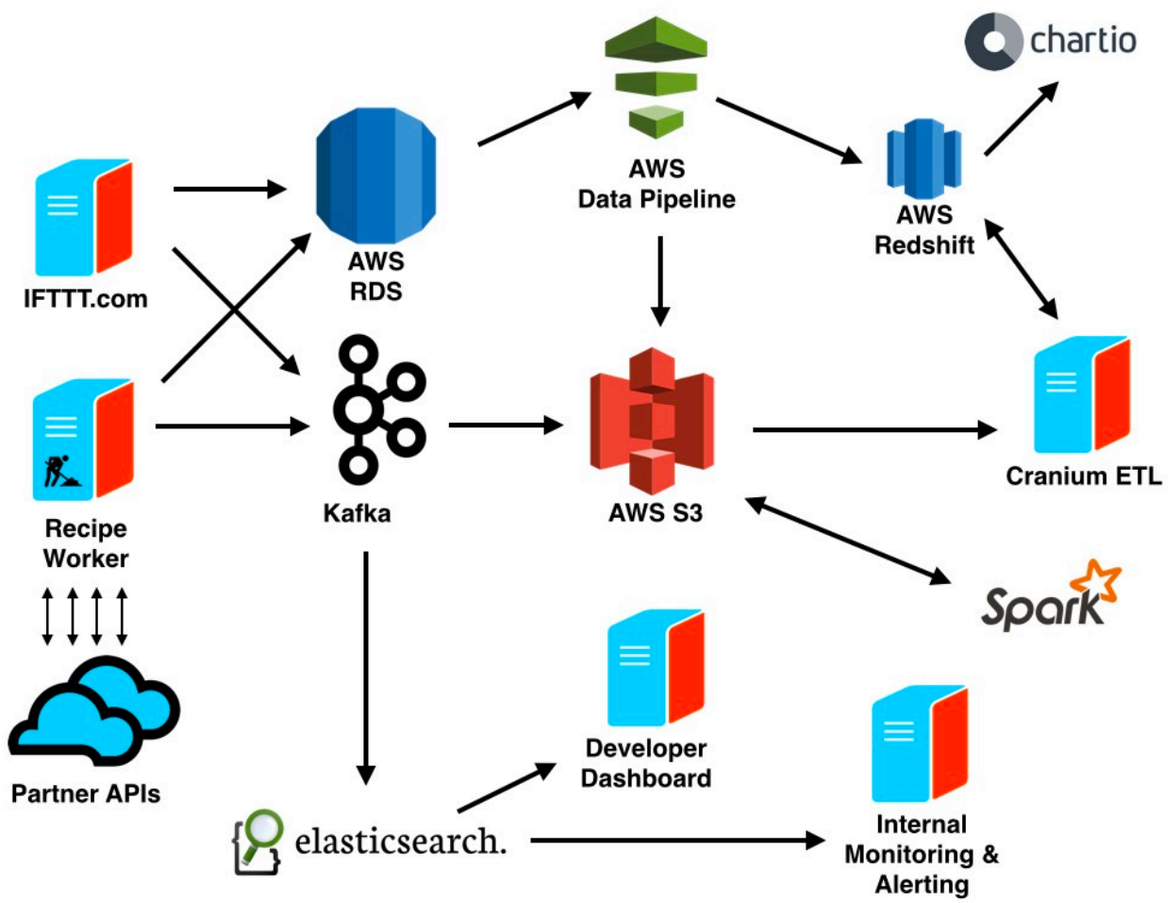
IT Infrastructure vs Operational Model

Implementing the Operational Model via Enterprise Architecture

Article: **Enterprise Architecture As Strategy**



Setting Up Data Infrastructure



The Data Infrastructure Ecosystem

Q1: What are the right tools for a business, especially a small one?

The Apache Foundation has listed 38 projects in the “Big Data” section and these tools have tons of overlaps on the problems they claim to address:

E.g **Distributed stream processing:**

- Flink
- Samza
- Storm
- Spark Streaming

Batch & Stream Processing:

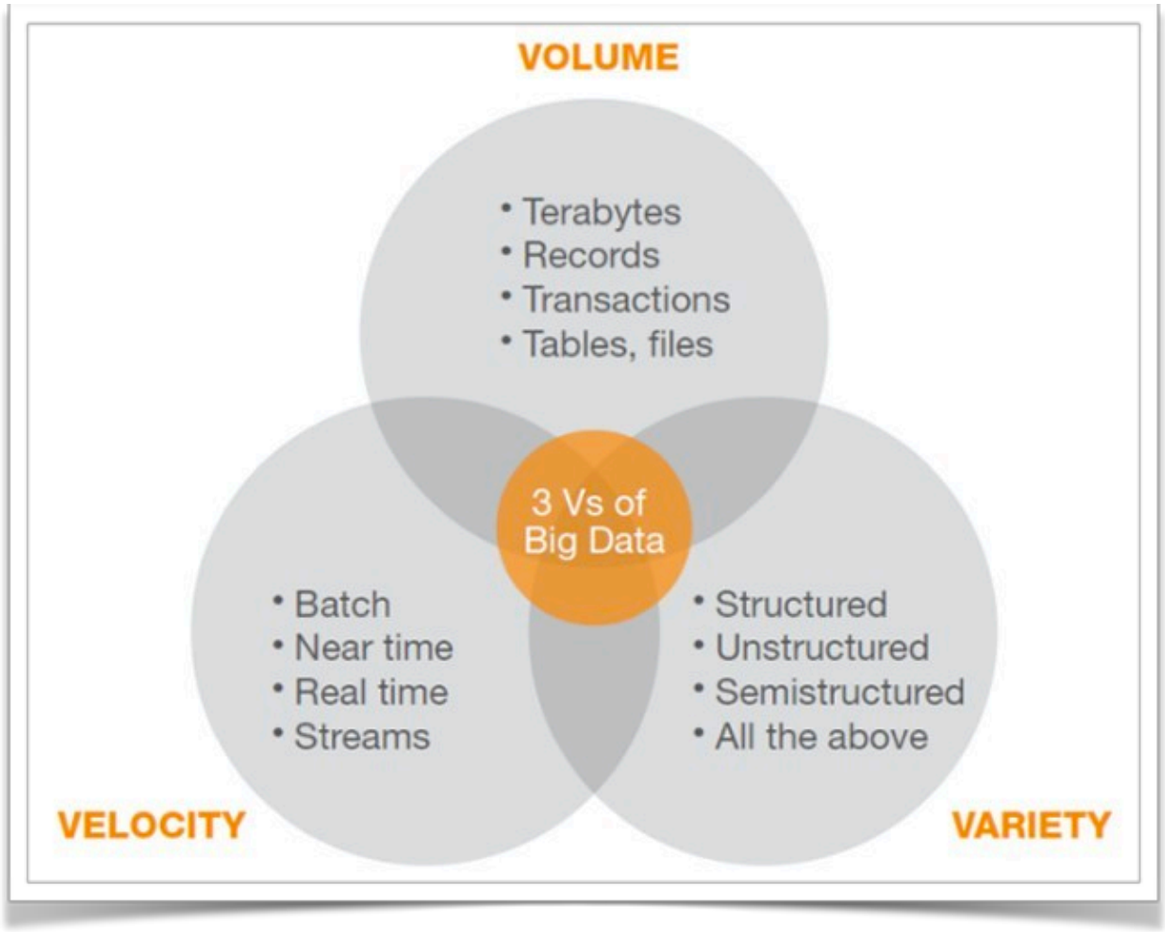
- Apex
- Beam

The Data Infrastructure Ecosystem

Stream processing and Big data

Stream-based applications such as
- Trading
- Social networks
- IoTs
- System monitoring, etc.

Involve i**mmense amounts of data (big data)** that have to be processed fast from a rapidly growing set of disparate data sources.



Big data are often characterised as:
Volume + Velocity + Variety

Volume describes the quantity of data
Variety describes range of data types and sources.
Velocity refers to the speed of generation of data or how fast the data is generated and processed to meet business demands.

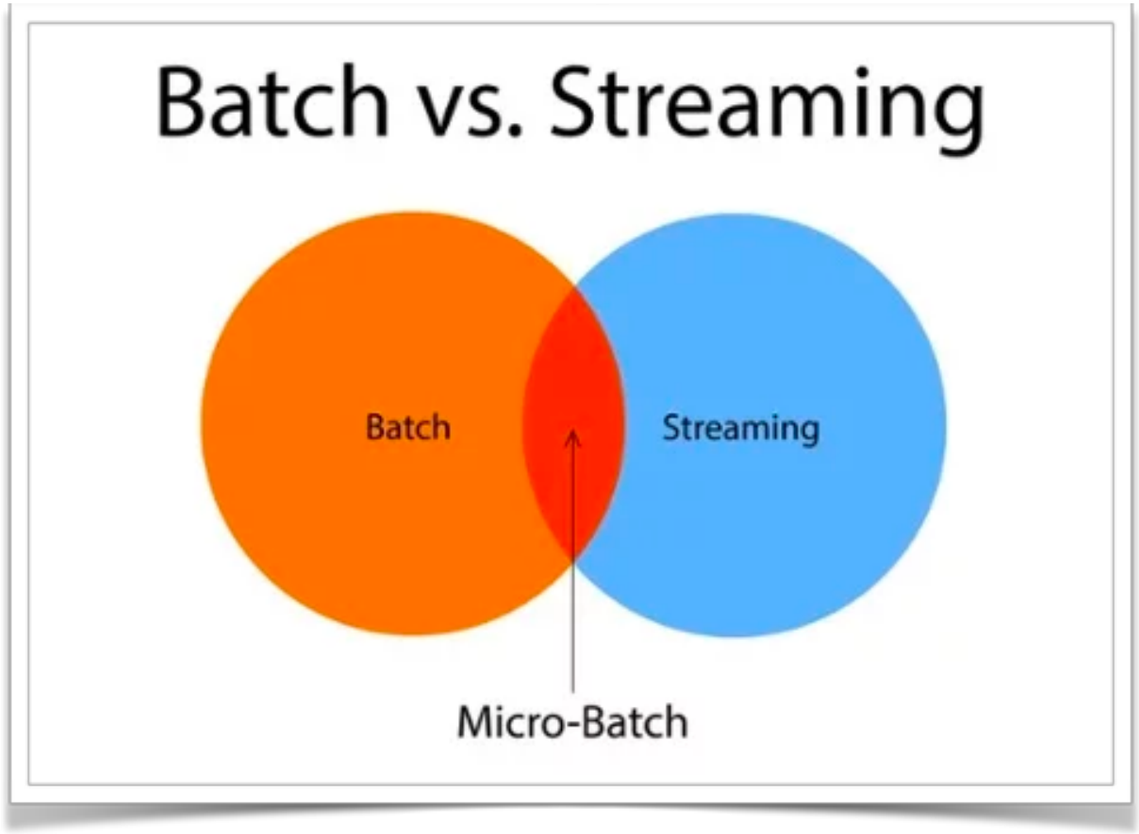
The Data Infrastructure Ecosystem

What is Distributed Stream processing

Distributed Stream Processing is a stateless, straight through processing of incoming data in a distributed fashion using ‘continuous queries’

There are three main approaches to processing data -

- **Batch Processing**,
- **Stream Processing** and
- **Micro-Batching**.



We can characterize **Batch Processing** systems as following:

- generally familiar concept of processing data en masse
- has access to all data
- might compute something big and complex
- more concerned with throughput that latency
- higher latencies (even in minutes)
- typical example is Hadoop’s MapReduce

Stream Processing system has following properties:

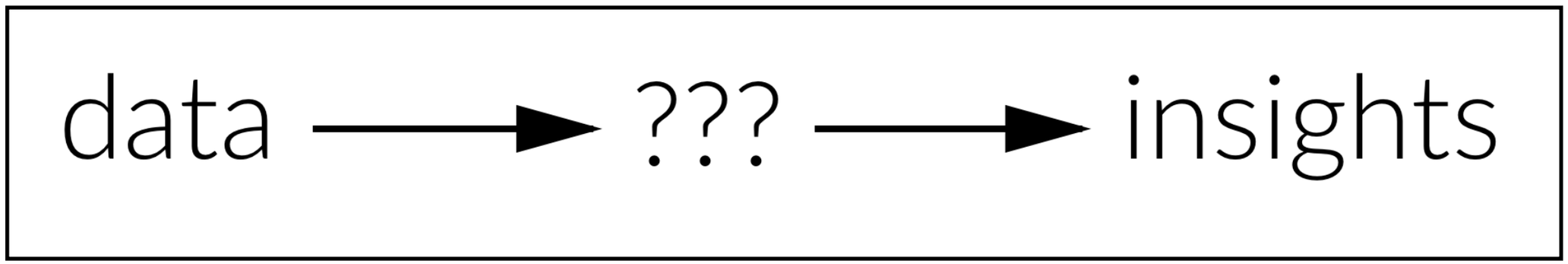
- a one-at-a-time processing model
- data are processed immediately upon arrival
- computations are relatively simple and generally independent
- sub-second latency
- difficult to maintain state efficiently
- typical example is Apache Storm Core

Finally, we will describe **Micro-Batching** as follows:

- a special case of batch processing with very small batch sizes
- mix between batching and streaming
- latency in seconds
- easier windowing and stateful computation
- typical example is Spark Streaming

The Data Infrastructure Ecosystem

Q2: How to navigate the options?



A well known challenge is mapping a goal such as "**get insights from data**" with a **specific set of technologies to deploy**.

Stage 1

You have small data

An article written by Chris Stucchio in 2013 Dont Use Hadoop.

If you have less than 5TB of data, start small to save you operational headaches with maintaining systems you don't need.

1. **Make your data querable in SQL**
2. **Choose a Business Intelligence (BI) tool**

1. Everything in SQL

This step unlocks data for the entire organisation

- **SQL is easy to use**. Providing SQL access enables the entire company to become self-serve analyst. This means more time for your engineering or technical team.

- Make **PostgreSQL**, **MySQL** your primary datastore and provision access.

- If you have a **NoSQL** database like ElasticSearch, MongoDB, or DynamoDB, you will need to do more work to convert your data and put it in a SQL database. This is an Extract, Transform, Load (**ETL)** pipeline.

- Depending on your existing infrastructure, there are many cloud ETL providers (e.g. **Segment**) that you can leverage.

- If you **need to build your own data pipelines**, you have to keep them simple at first **using simple scripts** to periodically dump updates from your database into SQL.

Business Intelligence (BI) Tool

A “perfect” Business Intelligence (BI) tool is a must….

Tools such as:

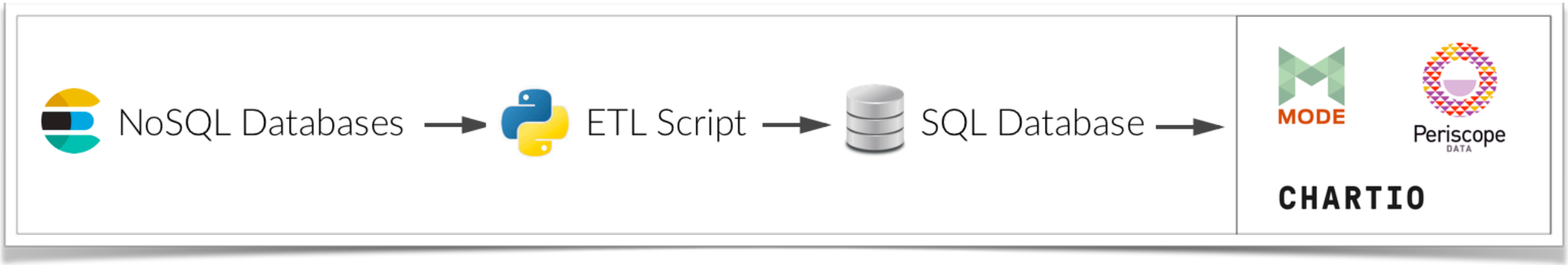
- **Chartio**
- **Mode Analytics**
- **Periscope Data**

Work super well to get your analytics started.

You can point these tools directly at your SQL database with a **quick configuration** so that you can dive right into creating **dashboards**

Our Small Data Pipeline

All together, we now have this set-up



At this stage,

- You may have **several datastores** and a **mix of SQL and NoSQL backends**,
- Maybe, a handful of **third parties** you are gathering data from.
- Probably **multiple levels in your ETL pipelines**, some dependencies, etc.

How to Fix This?

Workflow Management & Automation

To manage the ETL pipelines, set up (e.g. **Airflow** from Airbnb)

- Airflow will enable you to schedule jobs at regular intervals and express both temporal and logical dependencies between jobs. It is also a great place in your infrastructure to add job retries, monitoring & alerting for task failures

There is also: **Luigi** from Spotify

Pinball from Pinterest

How to Fix This?

Building ETL Pipelines

As y**our business grows, your ETL pipeline requirements will change a lot**. You will need to **build more scalable infrastructure** and not a single script any longer.

You will have to expand from SQL access to converting your ETL scripts to run as a **distributed system, in clusters**.

- **Apache Sparks**, is a good start. It scales well and is fairly easy to set up and get running.
- Spark can be run on:
  - AWS using **EMR**
  - Google Cloud using **Cloud Dataproc**; or
  - **Apache Sqoop**

At this stage, you will get the following set up:

- **Extract** data from sources
- **Transform** data into standardized formats on persistent storage;
- **Load** into SQL-querable datastore

How to Fix This?

Setting Up a Data Warehouse

Now, you have to start building a data warehouse:

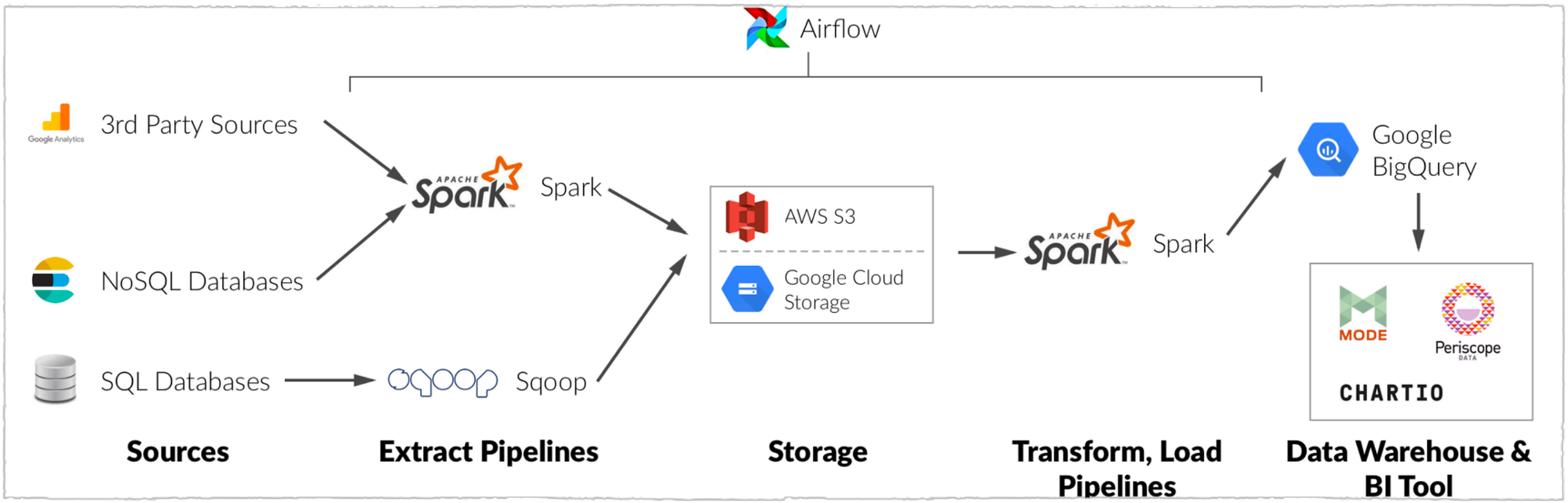
- Google's BigQuery is a good start (records can be loaded as JSON format) since it supports complex data types, is fully managed and serverless; reducing infrastructure to maintain. No admin needed; Your analysts can focus on analysing data to find insights using SQL.
- RedShift on AWS is another good example
- On-Premises Presto, as well

When setting up your data warehouse, you can adopt a 2-stage model:

- Unprocessed data is landed directly in a set of tables;
- A second tier that post-processes this data into "cleaner" tables for analytics. This second tier is also important because you can include metrics/KPI that you use to analyse each entity (e.g sign up time, number of purchases, geolocation of user, etc.)

How to Fix This?

Your infrastructure should now look like this



What is Data Virtualisation

https://www.denodo.com/en

Questions?