BIG DATA STACK

L6

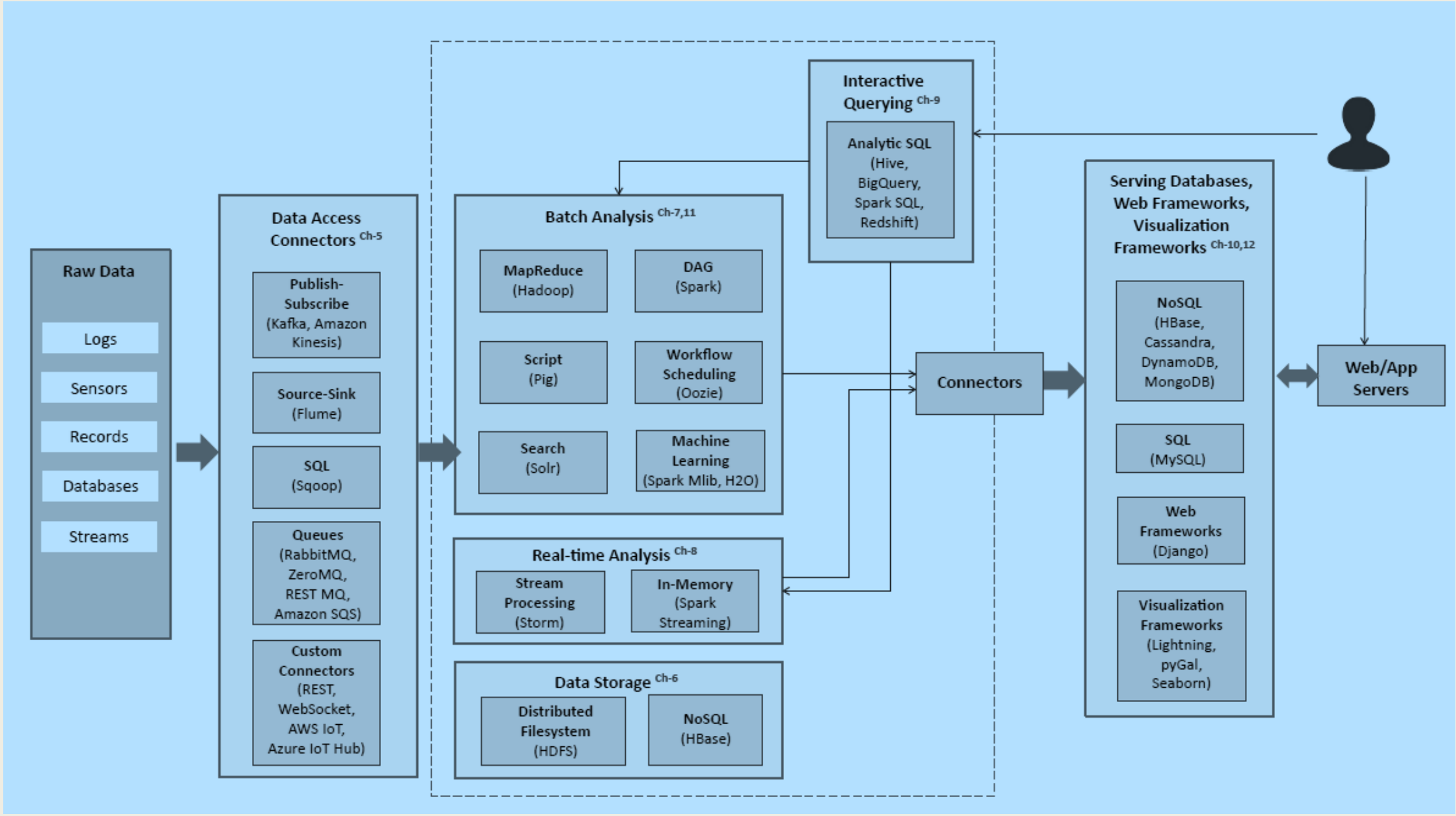
Motivation

- While the Hadoop framework has been one of the most popular frameworks for big data analytics, there are several types of computational tasks for which Hadoop does not work well.
- Hadoop is an open source framework for distributed batch processing of massive scale data using the Map Reduce programming model.
- The Map Reduce programming model is useful for applications in which the data involved is so massive that it would not fit on a single machine.
- Hadoop is an open source framework for distributed batch processing of massive scale data using the Map Reduce programming model.

Motivation

- Map Reduce is best suited for descriptive analytics and the basic statistics computational tasks because the operations involved can be done in parallel (for example, computing counts, mean, max/min, distinct, top-N, filtering and joins).
- Many of these operations are completed with a single Map Reduce job. For more complex tasks, multiple Map Reduce jobs can be chained together.
- However, when the computations are iterative in nature, where a Map Reduce job has to be repeatedly run, Map Reduce takes a performance hit because of the overhead involved in fetching the data from HDFS in each iteration.
- For other types of analytics and computational tasks, there are other alternative frameworks which we will discuss as a part of the Big Data Stack.

Big Data Stack



Raw Data Sources

- In any big data analytics application or platform, before the data is processed and analyzed, it must be captured from the raw data sources into the big data systems and frameworks. Some of the examples of raw big data sources include:
  - *Logs*
  - *Transactional data*
  - *Social Media*
  - *Databases*
  - *Sensor Data*
  - *Clickstream Data*
  - *Surveillance Data*
  - *Healthcare Data*

Data Access Connectors

- The Data Access Connectors includes tools and frameworks for collecting and ingesting data from various sources into the big data storage and analytics frameworks.
- The choice of the data connector is driven by the type of the data source. Let us look at some data connectors and frameworks which can be used for collecting and ingesting data. These connectors can include both wired and wireless connections.
  - *Publish-Subscribe Messaging*
  - *Source-Sink Connectors*
  - *Database Connectors*
  - *Messaging Queues*
  - *Custom Connectors*

Data Storage

- The data storage block in the big data stack includes distributed filesystems and non-relational (NoSQL) databases, which store the data collected from the raw data sources using the data access connectors.
- Hadoop Distributed File System (HDFS), a distributed file system that runs on large clusters and provides high-throughput access to data.
- With the data stored in HDFS, it can be analyzed with various big data analytics frameworks built on top of HDFS.
- For certain analytics applications, it is preferable to store data in a NoSQL database such as HBase. HBase is a scalable, non-relational, distributed, column-oriented database that provides structured data storage for large tables.

Batch Analytics

- The batch analytics block in the big data stack includes various frameworks which allow analysis of data in batches. These include the following:
  - *Hadoop-MapReduce*
  - *Pig*
  - *Oozie*
  - *Spark*
  - *Solr*
  - *Machine Learning*

Real-time Analytics

- The real-time analytics block includes the Apache Storm and Spark Streaming frameworks. Apache Storm is a framework for distributed and fault-tolerant real-time computation.
- Storm can be used for real-time processing of streams of data. Storm can consume data from a variety of sources such as publish-subscribe messaging frameworks (such as Kafka or Kinesis), messaging queues (such as RabbitMQ or ZeroMQ) and other custom connectors.
- Spark Streaming is a component of Spark which allows analysis of streaming data such as sensor data, click stream data,web server logs, for instance.
- The streaming data is ingested and analyzed in micro-batches. Spark Streaming enables scalable, high throughput and fault-tolerant stream processing.

Interactive Querying

- Interactive querying systems allow users to query data by writing statements in SQL-like languages.
  - *Spark SQL*
  - *Hive*
  - *Amazon Redshift*
  - *Google BigQuery*

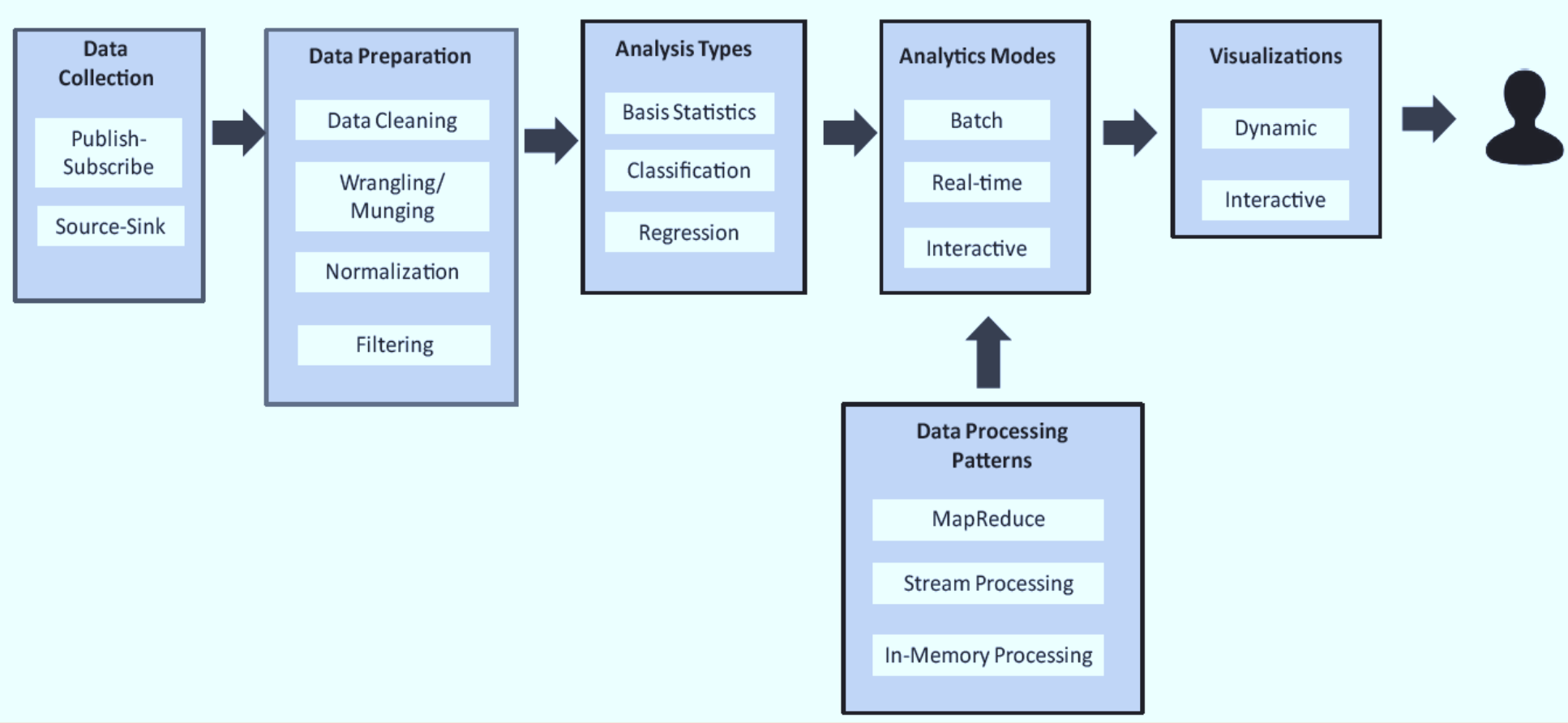
Serving Databases, Web & Visualization Frameworks

- While the various analytics blocks process and analyze the data, the results are stored in serving databases for subsequent tasks of presentation and visualization.
- These serving databases allow the analyzed data to be queried and presented in the web applications.
  - *MySQL*
  - *Amazon DynamoDB*
  - *Cassandra*
  - *MongoDB*

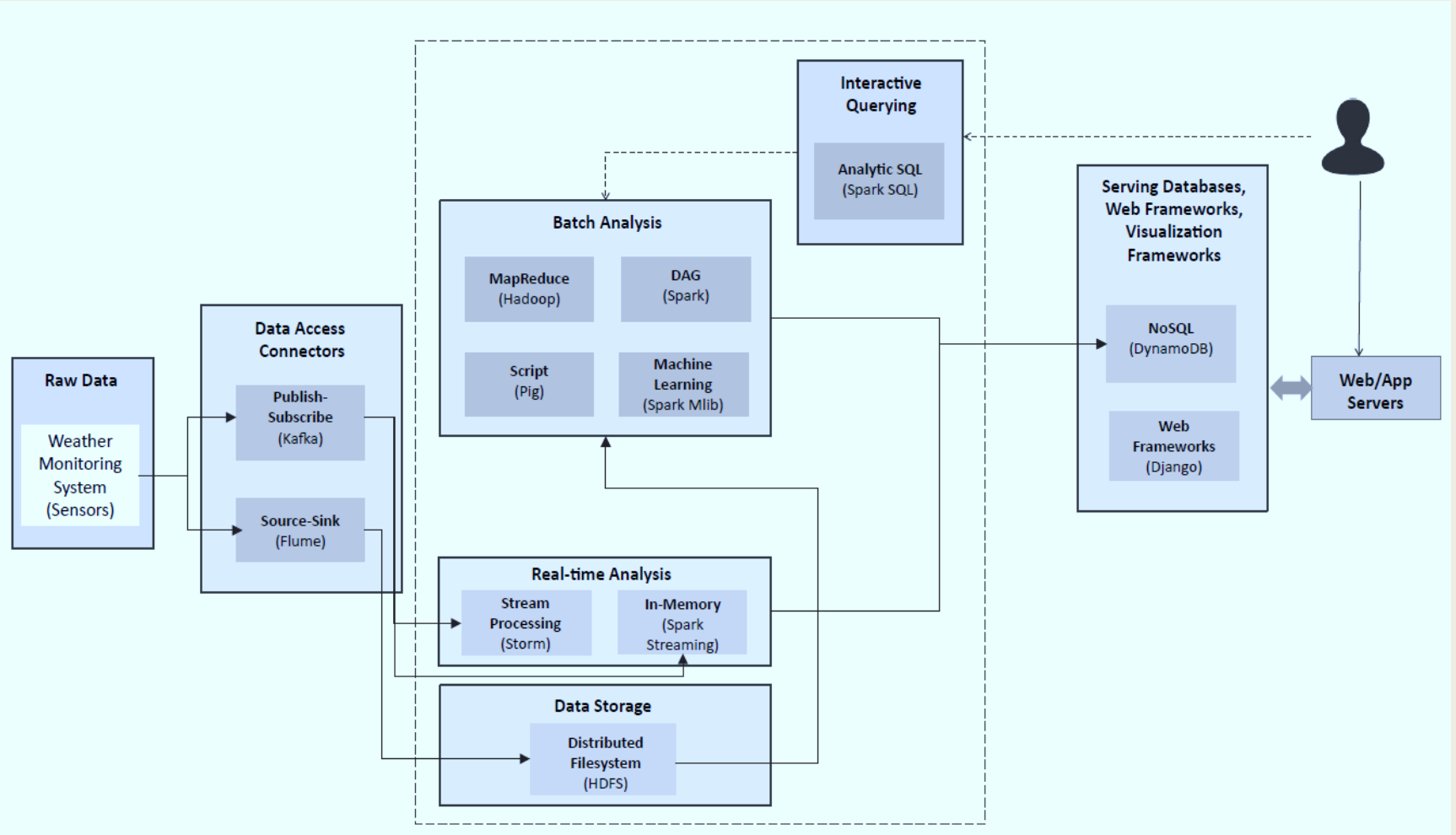
Serving Databases, Web & Visualization Frameworks

- Django, which is an open source web application framework for developing web applications in Python. Django is based on the Model-Template-View architecture and provides a separation of the data model from the business rules and the user interface.
- While web applications can be useful for presenting the results, specialized visualizing tools and frameworks can help in understanding the data, and the analysis results quickly and easily. Some of the tools are:
- Lightning
- Pygal
- Seaborn

Case Study: Weather Data Analysis



Case Study: Weather Data Analysis



Reference

- Big Data Analytics_ A Hands-On Approach.by Arshdeep Bahga and Vijay Medisetti