

The background of the slide features a close-up of a lit sparkler against a dark night sky. The sparks are bright orange and yellow, creating a dynamic, starburst-like pattern. A large, semi-transparent green rectangle is overlaid on the left side of the image, serving as a backdrop for the title and subtitle.

Design Proposal for Data Pipelines

Data Ingestion- Design Proposal

Data processing requirements & objectives

System Use cases

- Understand & propose data processing platform which will handle following use cases:
- Support a REST API which will provide the following insights:
 - average weather
 - average market interest
 - accompanied data for a region
- daily weather data and daily market prices of the past 15 years
- run analyses on available data and store the results for consumption.

Requirement analysis & scope

- System requirements

Functional:

- Monitor & predict weather patterns
- Monitor & predict market interest

Nonfunctional:

- Maintainable : Minimal or no manual steps during daily operations
- Adaptable : Minimal rework to accommodate new sources of data
- Scalable : Scale up/down to handle future increases/decreases in data volumes
- Fault Tolerable: Minimal manual intervention to recover from failures.

- End Consumers of curated data

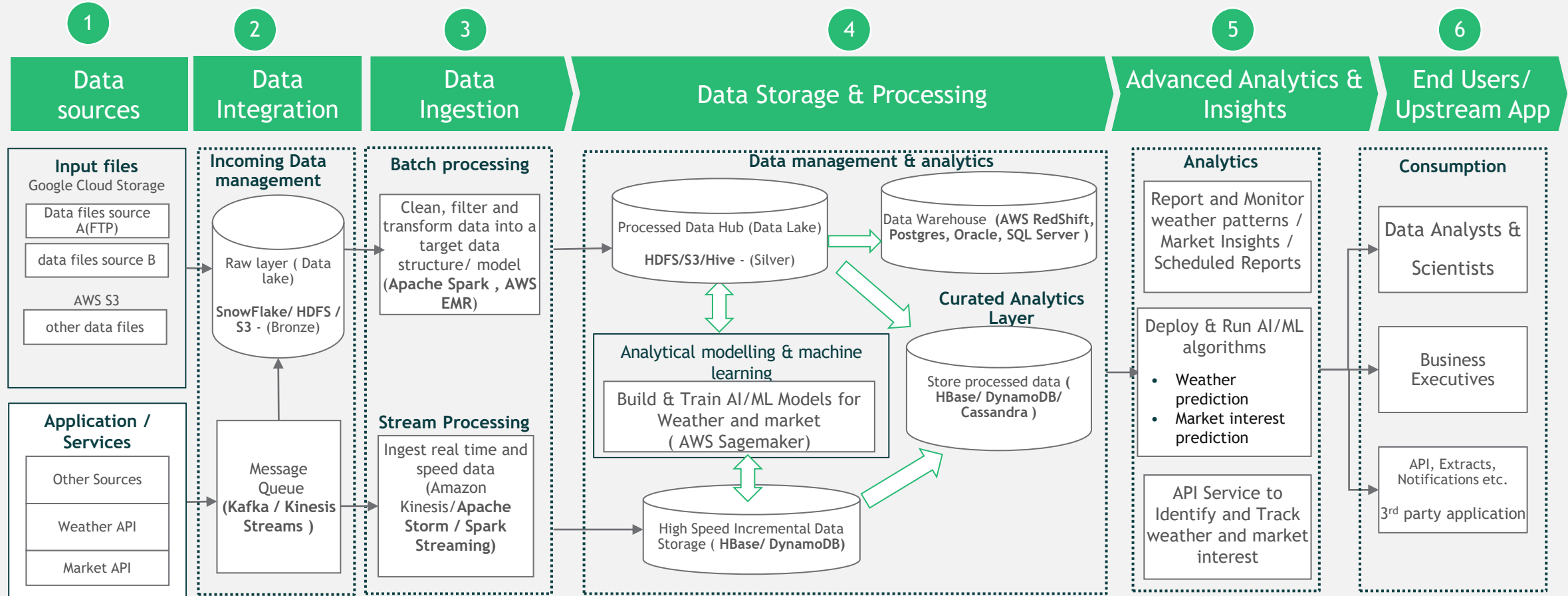
- Data analysts
- External users and 3rd party applications systems consume data through API service
- BI Executives

Data Processing Design & Technology

Solution design & tool stack

- Data Integration
- Data Ingestion
- Data Storage & Processing
- Advanced Analytics & Insights

Data sources will be leveraged to generate insights, reports, and predict via advanced analytical models



Proposed tools and technology

Storage layer

1. File based cloud storage: AWS S3, GCS
2. Relational data base service : RDS
3. Fast read /writes : HBase, DynamoDB
4. Data Lake : SnowFlake, S3, Apache Hudi




Data processing and management

1. Data engineering pipeline:
 - ETL processing : Apache Spark for processing data loads, AWS Glue
 - Orchestration tool of job automation and monitoring - Airflow, Oozie
2. ML mode building and training :
 - AWS Sagemaker for data Analyst to prepare, build, train , and deploy analytical models
 - Tensor flow as ML framework for data scientist to train and infer deep neural network
3. Version control and automated deployments of code base
 - GitLab, GitHub
 - Ocptopus, AWS codeCommit, CodeBuild, CodeDeploy

Weather Analytics Platform

Let's discuss and refine

All elements of architecture available as fully managed, public cloud services

Architecture element				
Batch/Serving layer	Data Ingestion	Kinesis	Cloud Pub/Sub	Data Factory
	Data lake	S3, EMR	Cloud Storage	Data Lake Store
	Batch processing	EMR (Spark)	Cloud Dataflow (~analogous Spark)	HDInsight (Spark)
	Batch view database	EMR, HBase, Redshift	BigQuery	SQL Data Warehouse, HDInsight
Stream layer	Data Ingestion	Kinesis	Cloud Pub/Sub	HDInsight, Event Hubs, IoT Edge/Hub
	Real-time view processing	Kinesis Analytics, Apache Storm	Cloud Dataflow (~analogous Spark)	HDInsight (Spark), Stream Analytics
	Real-time view database	EMR, HBase	BigQuery Streaming Insert	Data Factory, HDInsight

Weather Analytics Platform

Appendix

Data sources

1. Two data sources

data source A delivers data every day, with a 3-day delay via FTP

data source B delivers data every 14 days via FTP

2. weather source C

delivers either the current weather or a 7-day forecast via a JSON REST API

the data can be retrieved using two endpoints - one for the current weather and one for the forecast

the forecast endpoint takes two parameters: latitude and longitude

the current endpoint takes three parameters: latitude, longitude, and optionally a timestamp

3. market data source D

delivers near real-time market data, available every 15 seconds via a JSON API for streaming (e.g. like the Twitter streaming API)

each data set includes a coordinate with a latitude and longitude, where each coordinate represents a 50-mile market radius ("region").

Task

Design an architecture based on the given requirements. Provide an architecture diagram of the solution, along with any documentation you think relevant. We are looking primarily for a conceptual architecture rather than specific implementations. Whilst you are free to use concrete technologies in your diagram, we are not looking for any implementation details in the documentation like configuration or scripting.