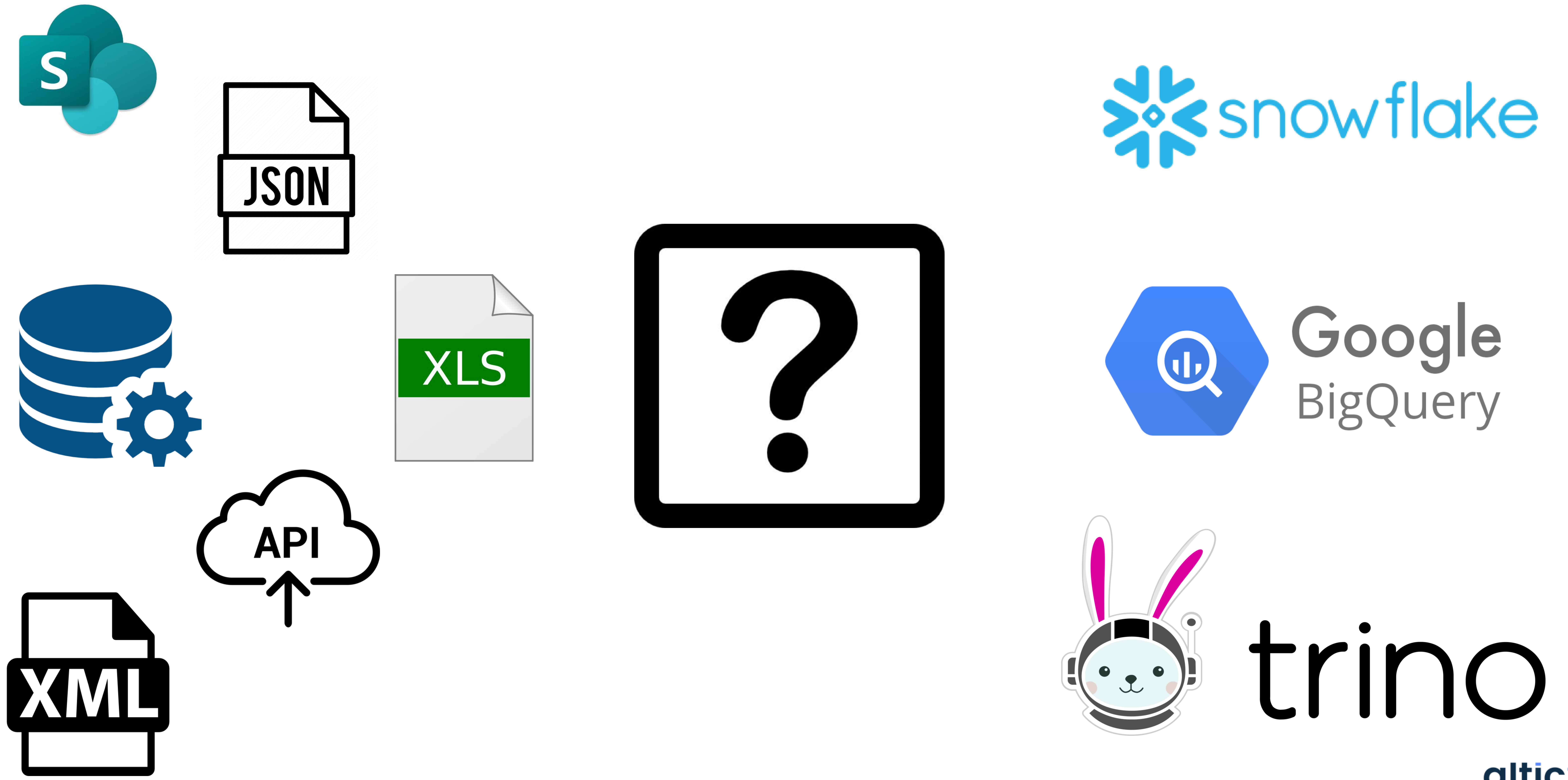
Serverless ETL pipelines with Apache Beam and GCP Dataflow

Romualdas Randamanskas (altic)

Table of Contents

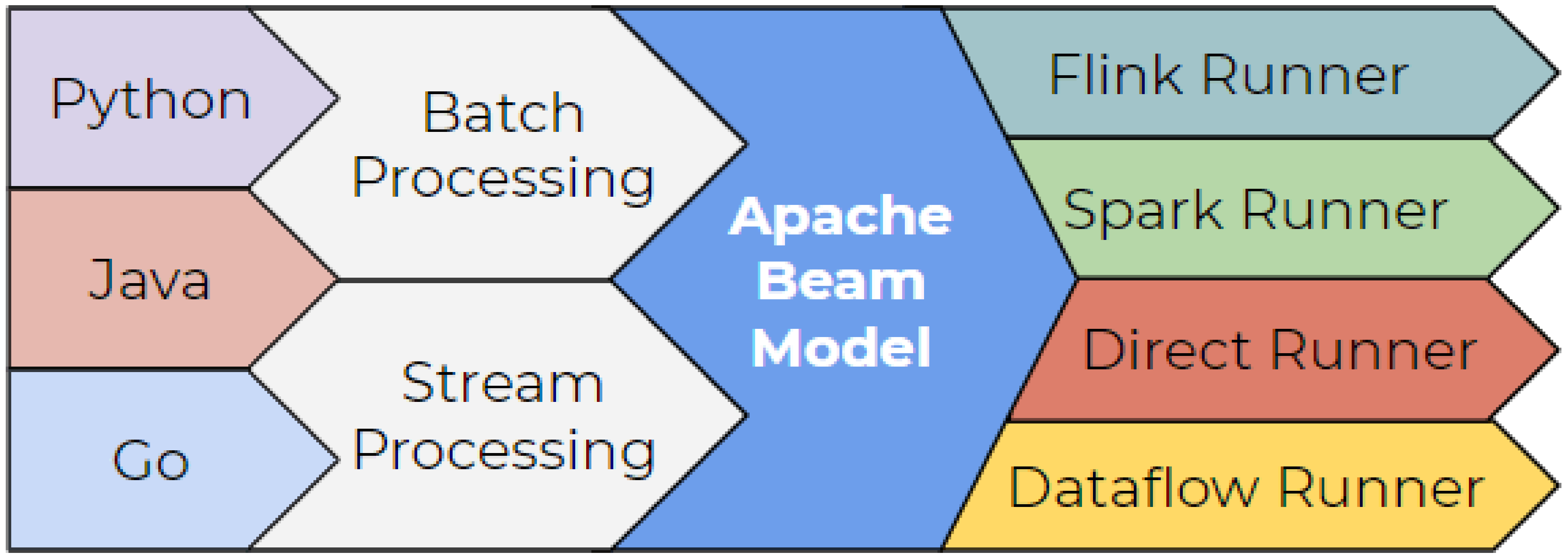
Problem



altic>_

Apache Beam Intro

- Unified programming model, Python, Java, Go SDKs
- Batch and Streaming Processing
- Open source, parallel data processing, IO connectors
- Different Runners, write one, deploy of you choice



https://blog.knoldus.com/apache-beam-overview/

beam

altic>_

Beam Abstractions

- **Pipeline** :- single program which includes input data, transforming that data and writing output data.
- **PCollection**:- A PCollection represents a distributed datasets that Beam pipeline operates on.
- **PTransform**:- A PTransform explains a data processing operation, or a step, in pipeline.
- **IOTransform**:- Beam contains huge number of IOs – library transforms so as to read or write data to external systems.

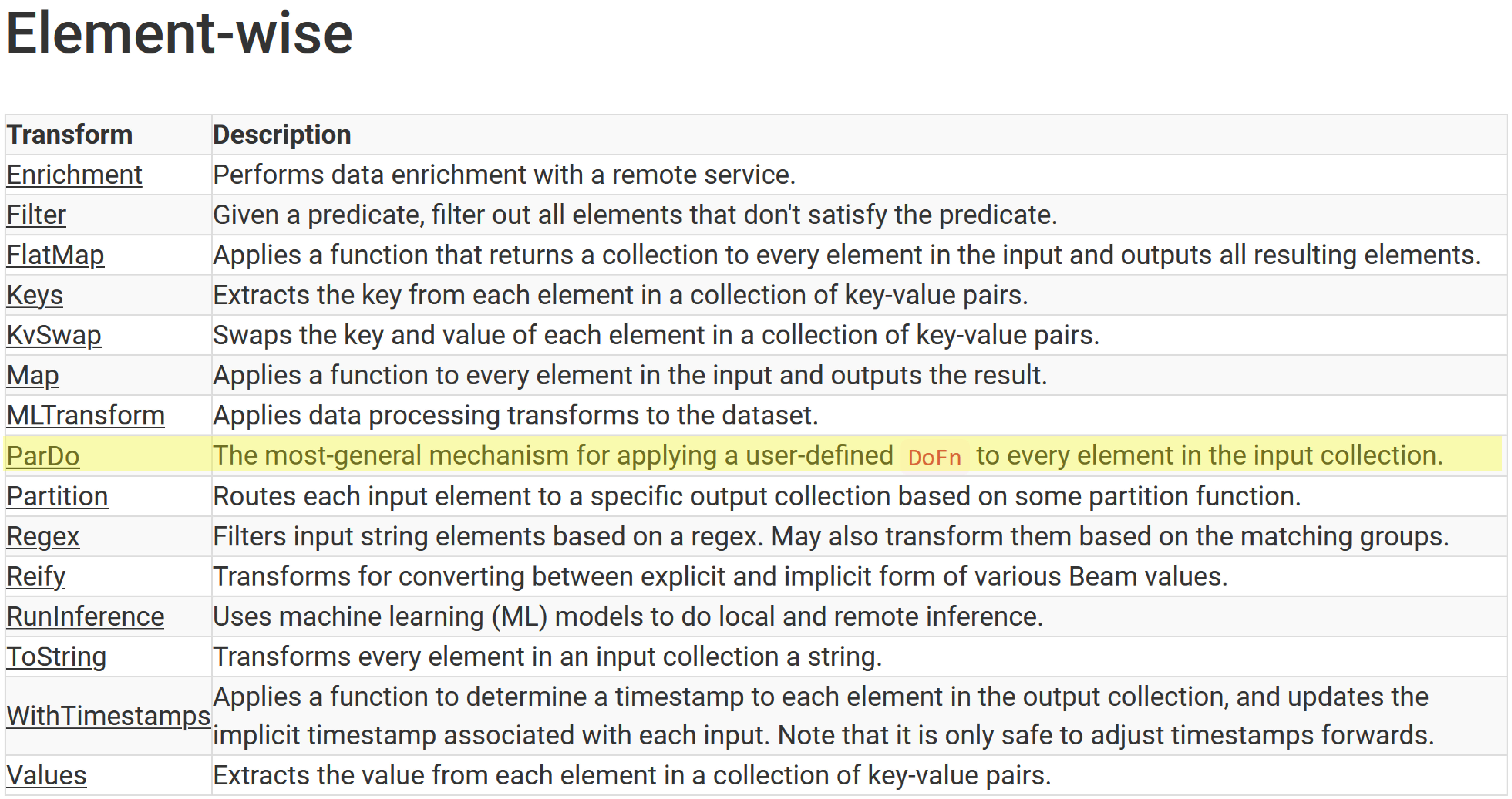
altic>_

Beam Program Workflow

- Firstly, create a pipeline object ans set Pipeline Options including Runner.
- Secondly, create an initial PCollection for pipeline data.
- Thirdly, apply PTransform to input PCollection.
- Lastly, use IOTransform to write the final output to external systems.
- At last, run the pipeline using the designated Pipeline runner.

altic>_

Beam Transformations

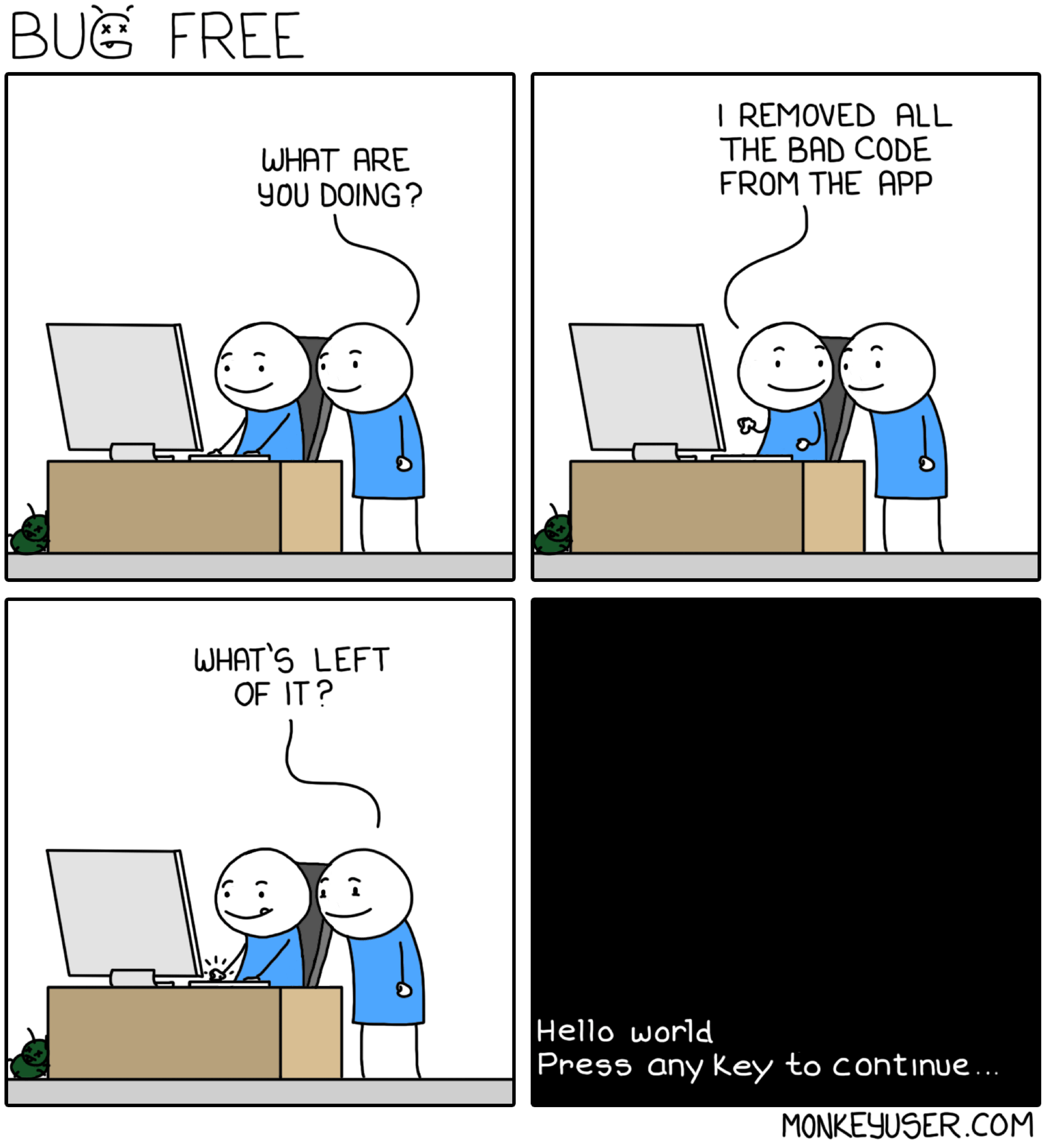


Element-wise

Transform	Description
Enrichment	Performs data enrichment with a remote service.
Filter	Given a predicate, filter out all elements that don't satisfy the predicate.
FlatMap	Applies a function that returns a collection to every element in the input and outputs all resulting elements.
Keys	Extracts the key from each element in a collection of key-value pairs.
KvSwap	Swaps the key and value of each element in a collection of key-value pairs.
Map	Applies a function to every element in the input and outputs the result.
MLTransform	Applies data processing transforms to the dataset.
ParDo	The most-general mechanism for applying a user-defined `DoFn` to every element in the input collection.
Partition	Routes each input element to a specific output collection based on some partition function.
Regex	Filters input string elements based on a regex. May also transform them based on the matching groups.
Reify	Transforms for converting between explicit and implicit form of various Beam values.
RunInference	Uses machine learning (ML) models to do local and remote inference.
ToString	Transforms every element in an input collection a string.
WithTimestamps	Applies a function to determine a timestamp to each element in the output collection, and updates the implicit timestamp associated with each input. Note that it is only safe to adjust timestamps forwards.
Values	Extracts the value from each element in a collection of key-value pairs.

https://beam.apache.org/documentation/transforms/python/overview/

Beam Demo



https://www.monkeyuser.com/2019/bug-free/

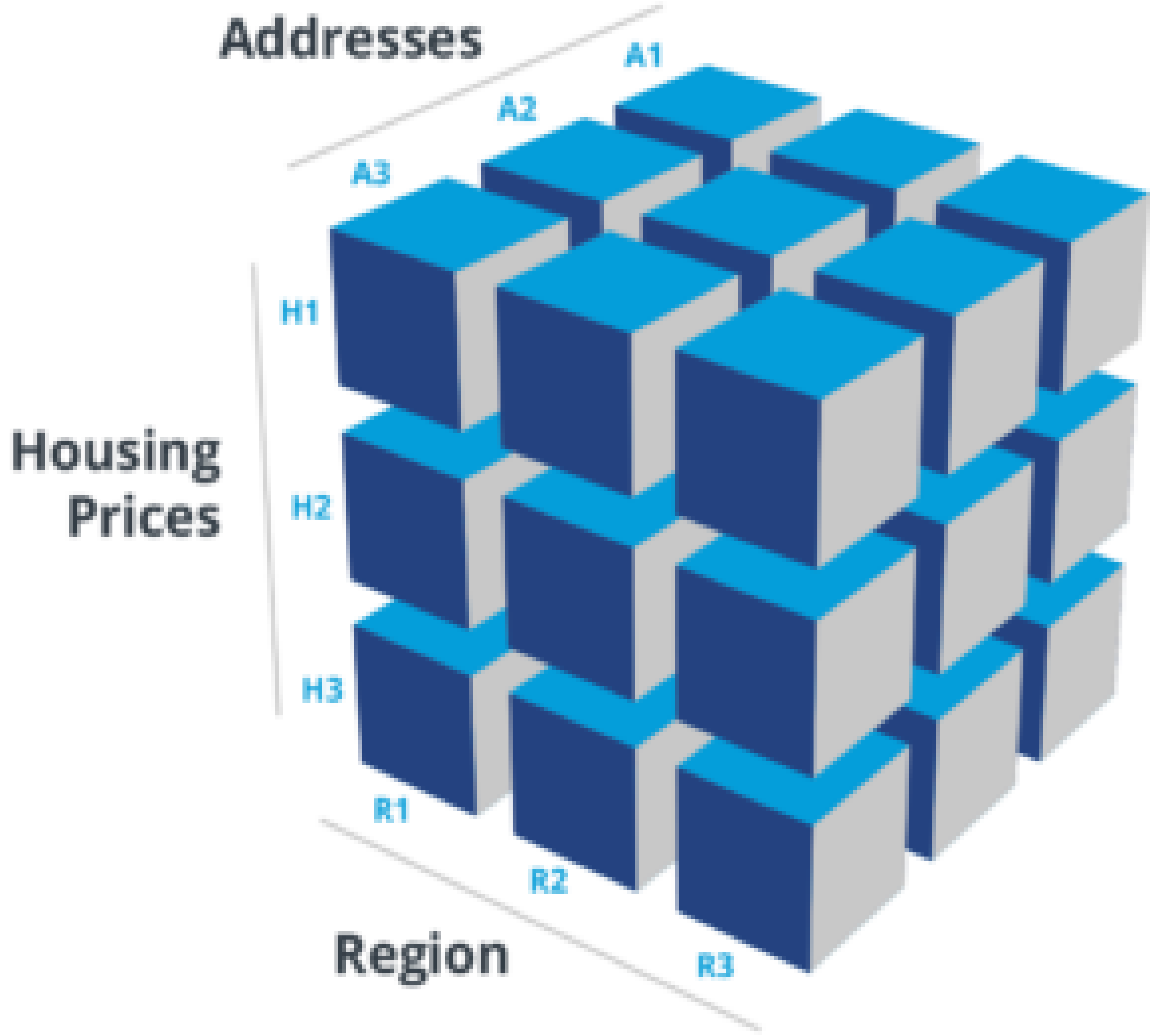
altic>_

OLAP (Online Analytical Processing)

polars

pandas

dask



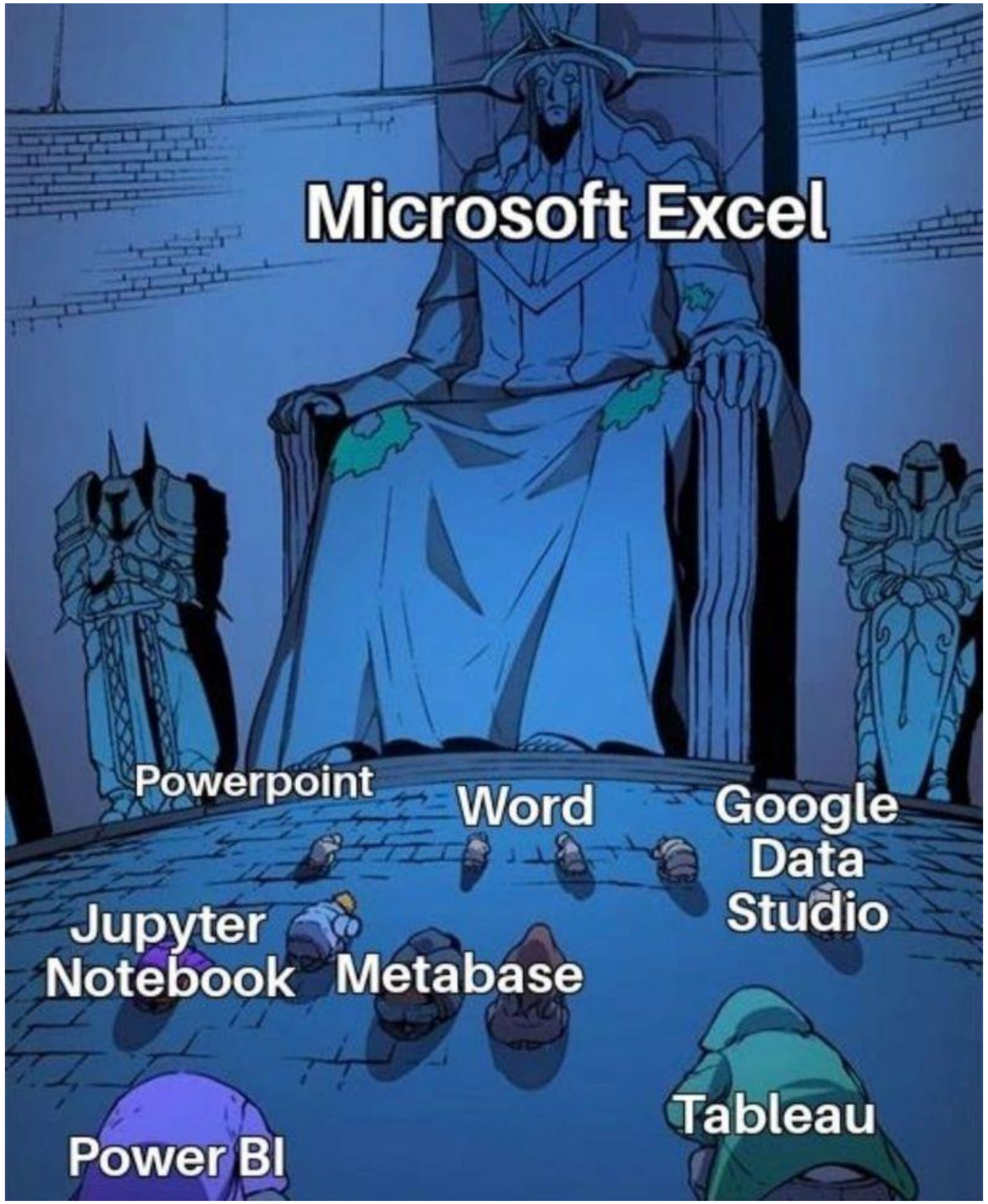
altic>_

Summary

Pros	Cons
Can run on multiple runners like Apache Flink, Apache Spark, and Google Cloud Dataflow.	Running jobs for certain runners (like Google Cloud Dataflow) might lead to vendor lock-in.
Designed to handle large-scale data processing tasks.	Requires careful tuning and resource management.
Easy migration from local/dev to cloud/prod	Steeper learning curve due to its strict DAG creation rules.
CLI integrates with other platforms and CI/CD tools	Specifics to run both batch and stream processing using a single model.
Many custom transformations and IO connectors.	

altic>_

Thanks!



altic