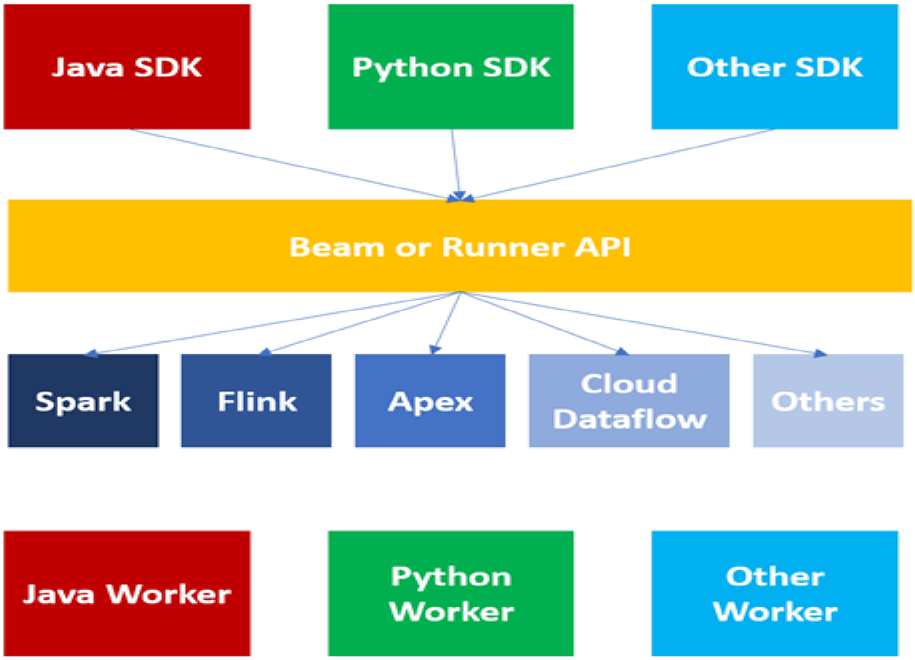
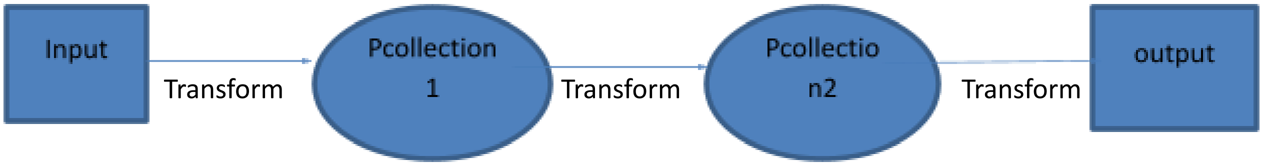
Apache Beam is a **unified model for defining both batch and streaming data-parallel processing that can build portable Big data pipelines**.

- **Apache beam**, the latest open source project of apache is a unified programming model for expressing efficient and portable big data processing pipelines.
  - Unified API to process both batch and streaming data.
  - **B**atch + Str**eam** -> Beam
  - Portable, beam pipeline once created in any language can be able to run on any execution frameworks like spark, flink, apex, cloud dataflow etc.,
  - Cloud Dataflow is fully managed service for creating and executing optimized parallel data processing pipelines.
  - Beam is a programming model whereas flink and spark are execution engines.
- **Languages:**
  - Java
  - Python
  - Go
- **Framework:**
  - Spark
  - Flink
  - Apex
  - Google dataflow
  - Samza
- **Architecture of Beam:**



- Beam or Runner API are the core of Apache Beam. If you want to run apache beam code in spark execution engine, so these beam or runner api will internally transfer apache beam to sparkcode to run on spark engine. Similarly for flink and others.
- Execution engines like spark, flink where apache code will actually run.

Flow of Beam Programming Model:



- o **Input:** text file, Big query, Avro files, database, stream (kafka, google pub/sub) etc.
- o **Output:** text file, database, hdfs, Google storage bucket, stream (kafka, google pub/sub) etc.

A typical Beam driver program works as follows:

- **Create a Pipeline object** and set the pipeline execution options, including the Pipeline Runner.
- **Create an initial PCollection** for pipeline data, either using the IOs to read data from an external storage system, or using a Create transform to build a PCollection from in-memory data.
- **Apply PTransforms** to each PCollection. Transforms can change, filter, group, analyze, or otherwise process the elements in a PCollection. A transform creates a new output PCollection without modifying the input collection.
- Use IOs to **write the final**, transformed PCollection(s) to an external source.
- **Run the pipeline** using the designated Pipeline Runner.

Basic Terminologies in Beam:

- **Pipeline -** A pipeline is a user-constructed graph of transformations that defines the desired data processing operations.
- **PCollection -** A PCollection is a data set or data stream. The data that a pipeline processes is part of a PCollection.
- **PTransform -** A PTransform (or transform) represents a data processing operation, or a step, in your pipeline. A transform is applied to zero or more PCollection objects, and produces zero or more PCollection objects.
- **Aggregation -** Aggregation is computing a value from multiple (1 or more) input elements.
- **User-defined function (UDF) -** Some Beam operations allow you to run user-defined code as a way to configure the transform.
- **Runner -** A runner runs a Beam pipeline using the capabilities of your chosen data processing engine.
    - o Direct runner
    - o Apache Flink runner
    - o cloud dataflow
    - o hazelcast jet runner

- Nemo runner
- Samza
- twister2 runner

Resources:

- https://beam.apache.org/documentation/runtime/model/
- https://beam.apache.org/documentation/sdks/python/
- https://beam.apache.org/documentation/basics/
- https://beam.apache.org/documentation/runtime/model/