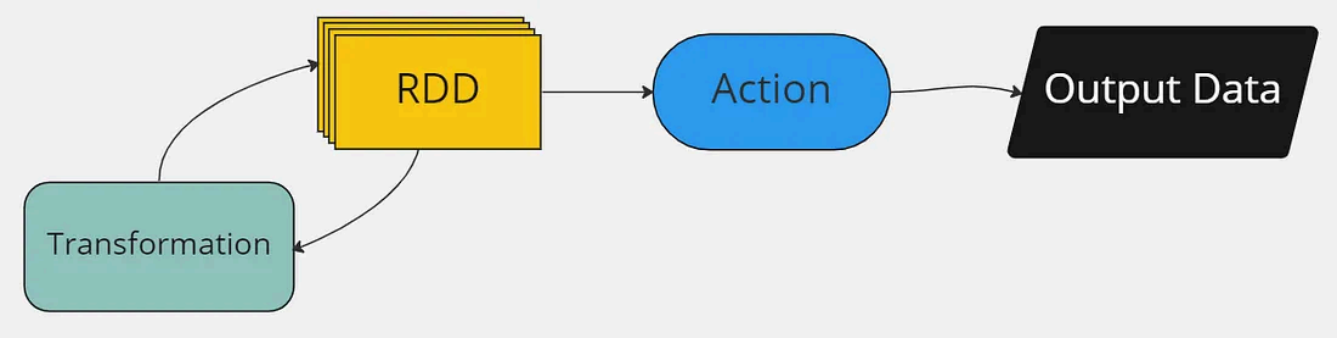
Spark Data Frame API Categories

Transformations

- Create RDDs from each other
- Used for transforming data



- **Narrow Transformations:**
  Narrow transformations are operations that can be performed on a single partition of a DataFrame without needing to shuffle the data across multiple partitions.
  - Faster and more efficient than wide transformations.
  - The lineage of RDDs is straightforward with narrow transformations. Each RDD depends on only one parent RDD or a set of parent RDDs that belong to the same partition.
    - One input partition results in one output partition
  - Performed in parallel on data partitions
  - Preserve the number of rows or reduce it.
  - Can be executed in a pipelined fashion without shuffling data.
  - Example: select(), filter(), withColumn(), drop()
  - For example filter() does not need to understand other data present on other worker nodes.
- **Wide Transformations** | `Shuffle` :
- These are the operations that require shuffling data across partitions. This means that the data needs to be moved between executor or worker nodes. Some examples of wide transformations in Spark include eg. Joins, repartitioning, groupBy, etc.
  worker nodes need to transfer (shuffle) data across the network to complete the required task.
  - Performed after grouping data from multiple partitions
  - Example: groupBy(), join(), and agg()
  - `join()` we need to collect data from across the cluster to complete a join of two datasets