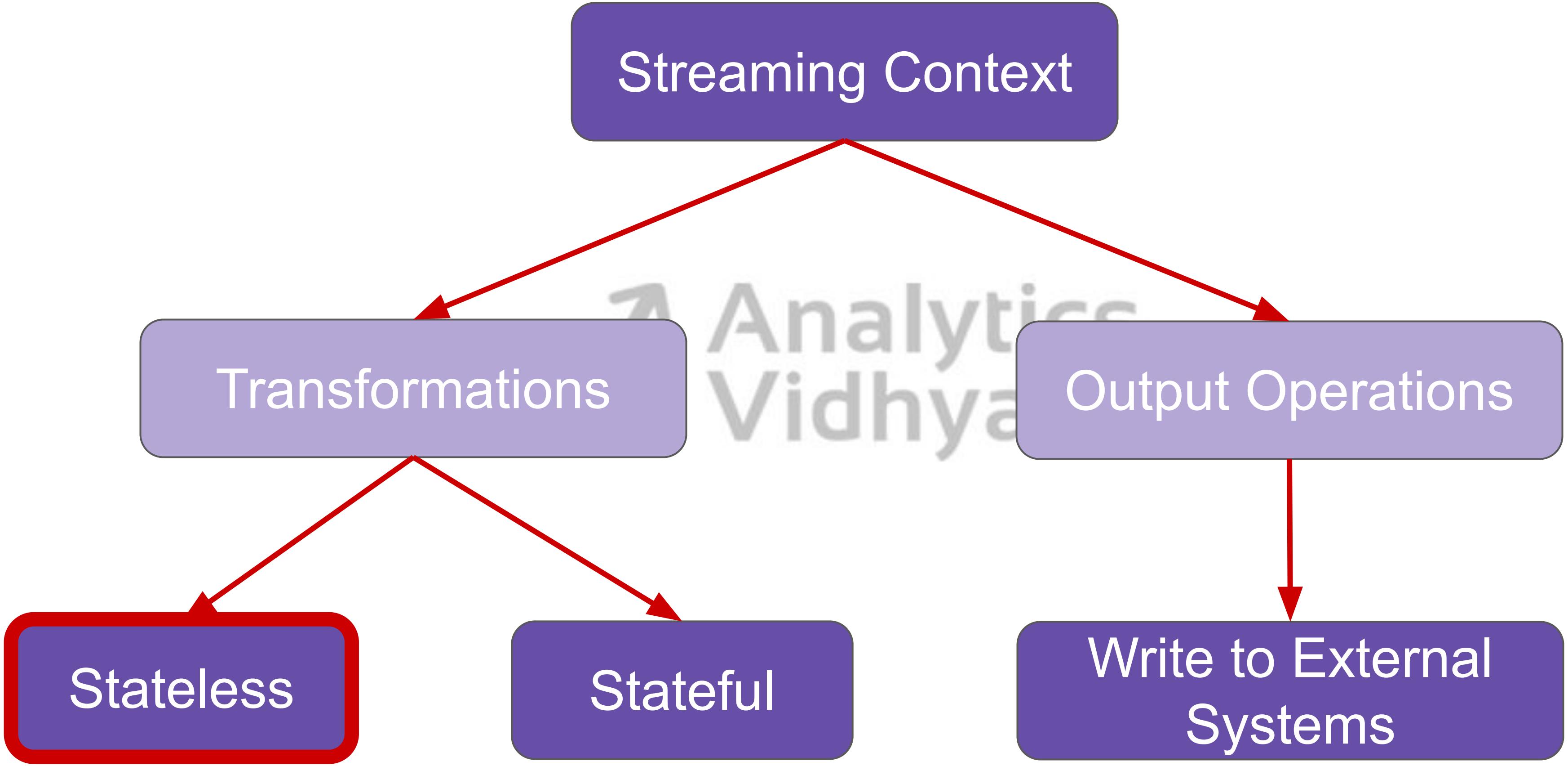
Stateless Transformations



Streaming Context

Transformations

Output Operations

Stateless

Stateful

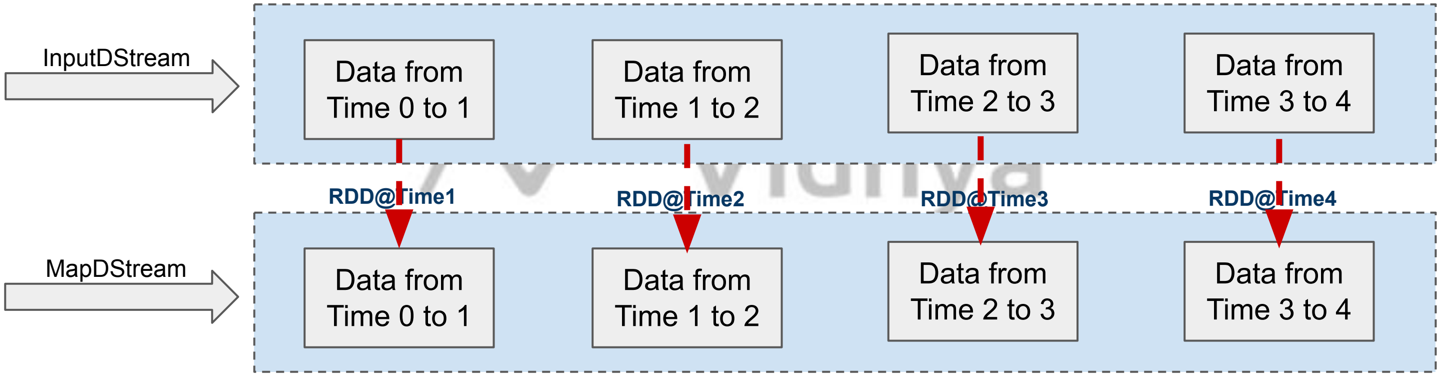
Write to External Systems

Stateless Transformations

- Processing of each batch has no dependency on the data of previous batches
- Similar to that of RDDs applied on every batch
  - **map()**, **filter()**, **reduceByKey()** etc
  - **cogroup()**, **join()**, **leftOuterJoin** etc

Stateless Transformations

- Performing ***operations*** on **DStreams** is equivalent to performing operations on underlying RDD operations on each batch



Map() each element to another

Transformations

map(func)	Return a new DStream by passing each element of the source DStream through a function func

Transformations

map(func)	Return a new DStream by passing each element of the source DStream through a function func
flatmap(func)	Similar to map, but each input item can be mapped to 0 or more output items

Transformations

map(func)	Return a new DStream by passing each element of the source DStream through a function func
flatmap(func)	Similar to map, but each input item can be mapped to 0 or more output items
filter(func)	Return a new DStream by selecting only the records of the source DStream on which func returns true

Transformations

map(func)	Return a new DStream by passing each element of the source DStream through a function func
flatmap(func)	Similar to map, but each input item can be mapped to 0 or more output items
filter(func)	Return a new DStream by selecting only the records of the source DStream on which func returns true
union(otherStream)	Return a new DStream that contains the union of the elements of the source DStream and the otherDStream

Transformations

map(func)	Return a new DStream by passing each element of the source DStream through a function func
flatmap(func)	Similar to map, but each input item can be mapped to 0 or more output items
filter(func)	Return a new DStream by selecting only the records of the source DStream on which func returns true
union(otherStream)	Return a new DStream that contains the union of the elements of the source DStream and the otherDStream
join(otherStream)	When called on two DStreams of (k,v) and (k,w) pairs, return a new DStream of (k,(v,w)) pairs with all pairs of elements for each key

Transformations

Few more Transformations:

- count()
- reduce()
- cogroup()
- countByValue()
- reduceByKey()
- transform()

Thank You