

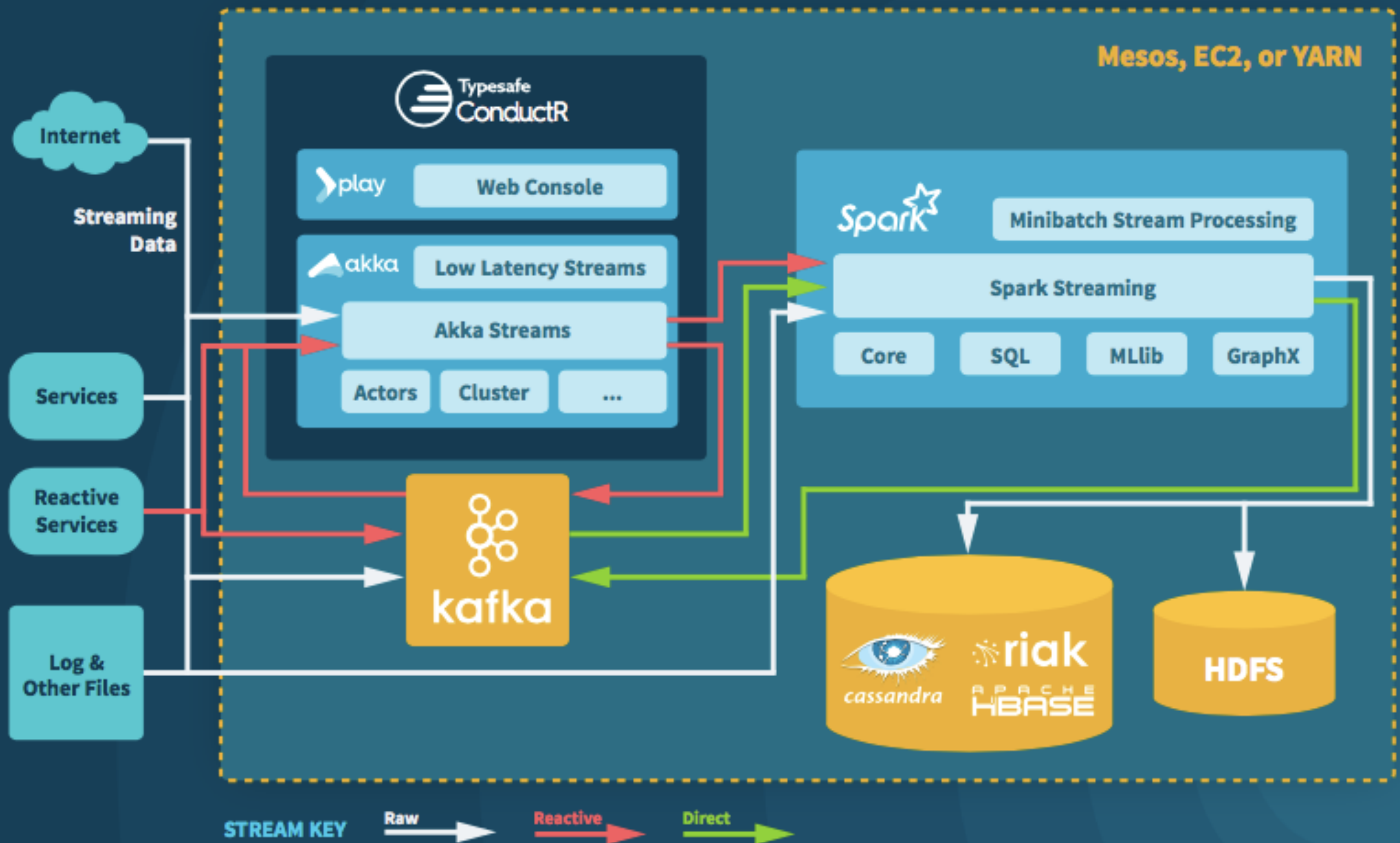
When things go *south*

Nilanjan Raychaudhuri
@nraychaudhuri

When things go *south*

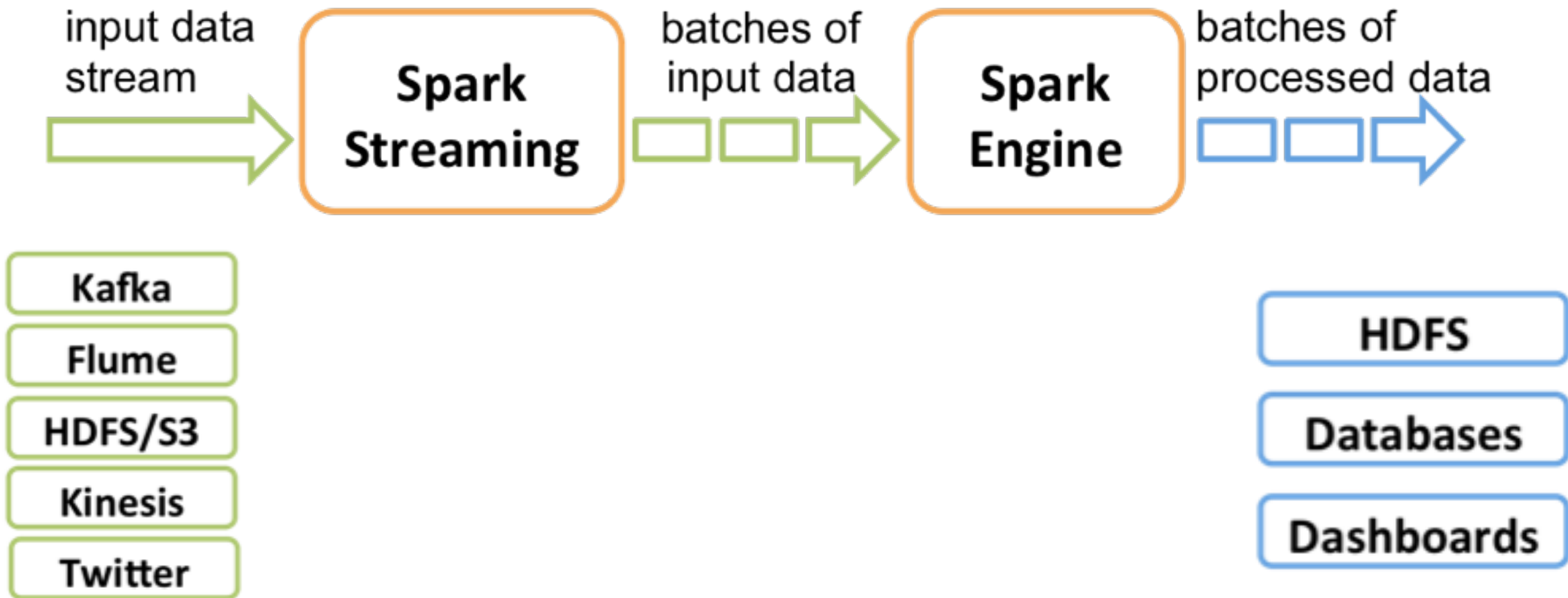
Nilanjan Raychaudhuri
@nraychaudhuri

Streaming landscape

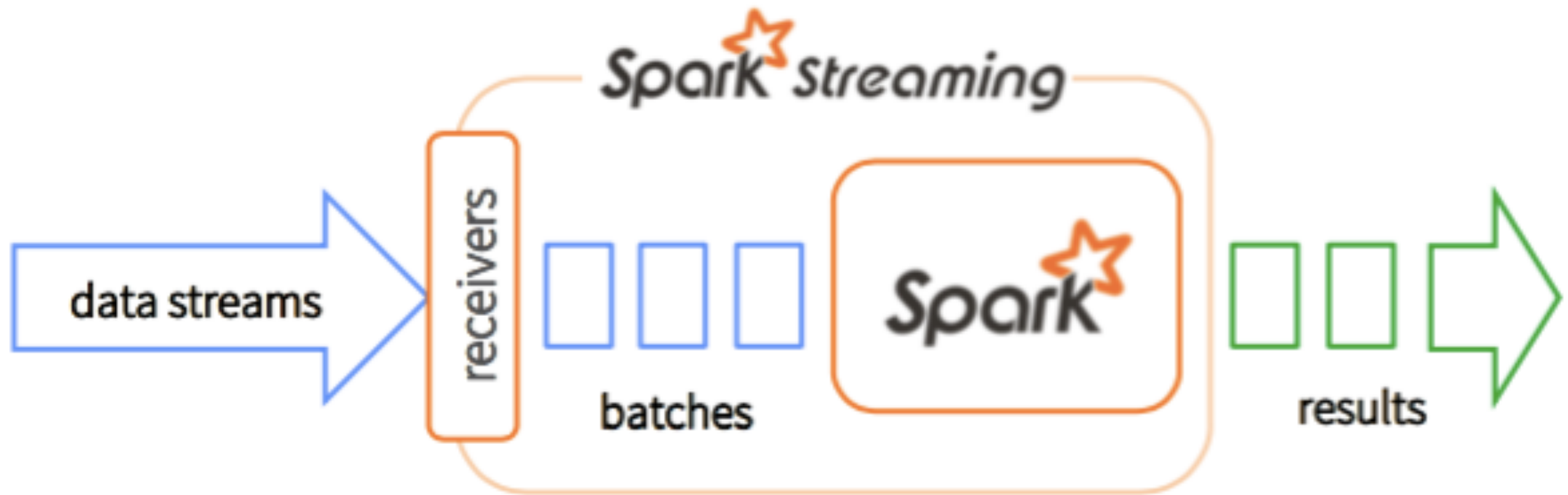


Spark streaming *101*

Spark Streaming



Spark Streaming

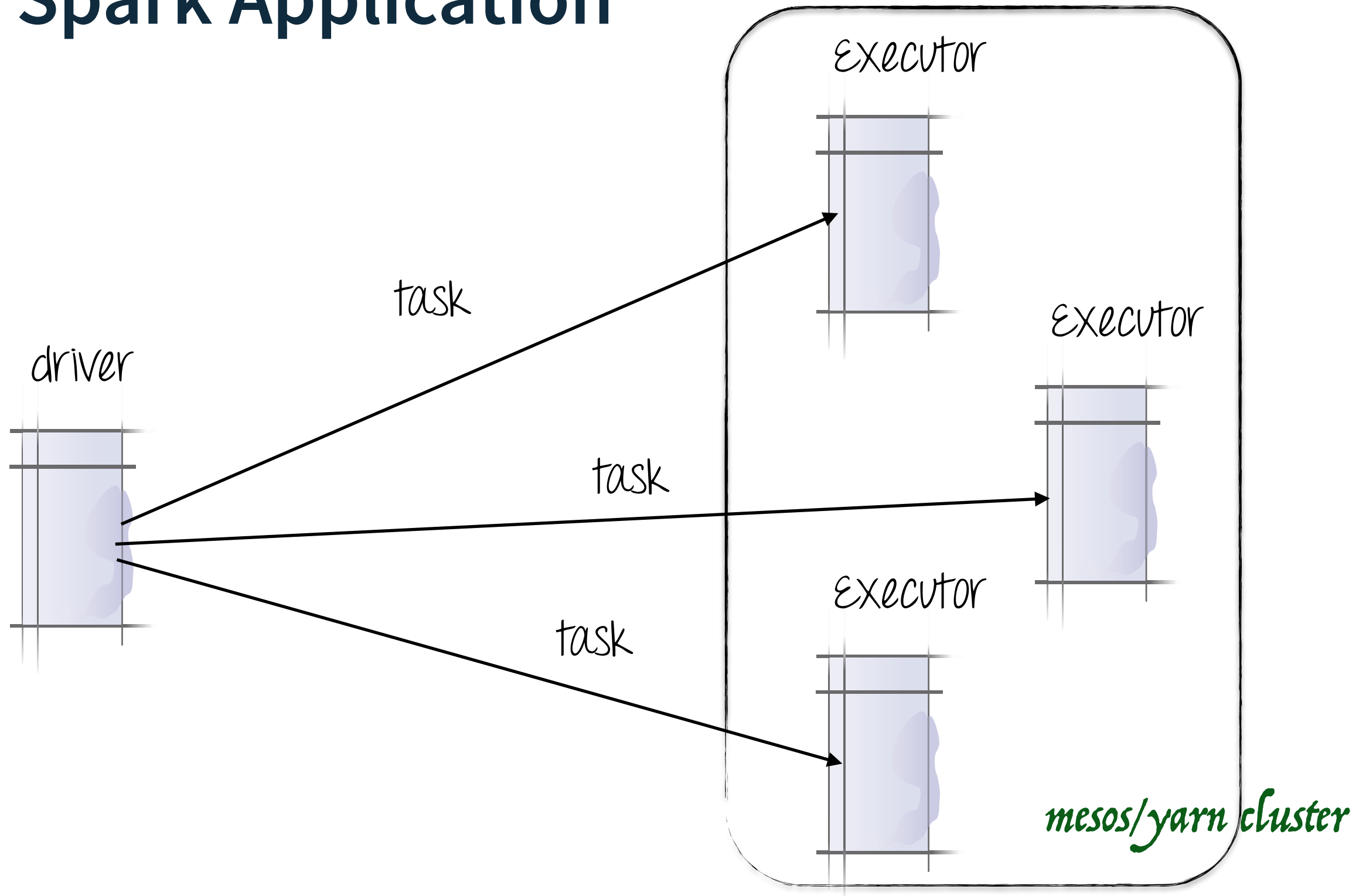


WordCount

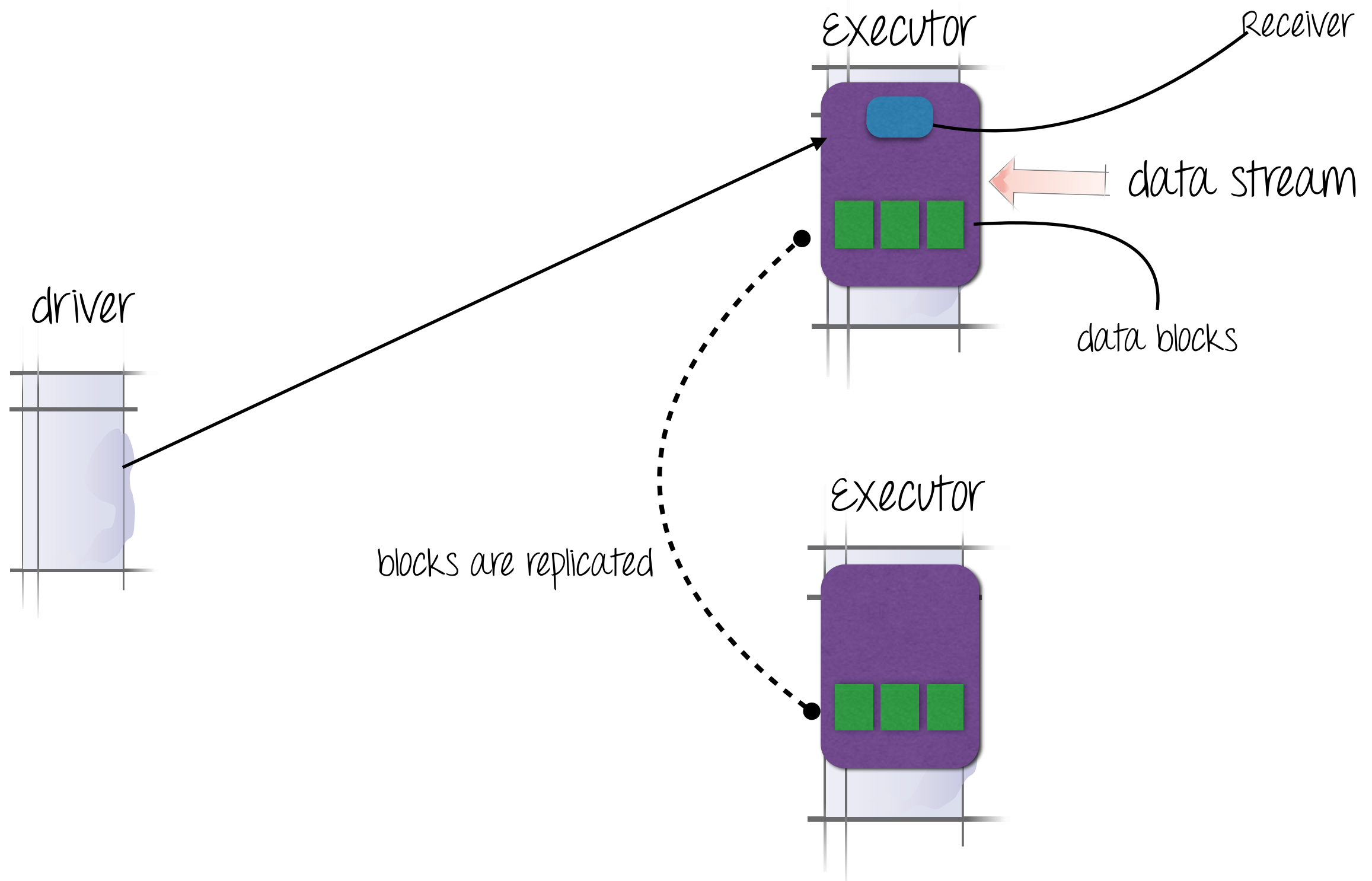
```
object WordCount {  
  def main(args: Array[String]) {  
    val context = new StreamingContext(new SparkConf(), Seconds(1))  
    val lines = KafkaUtils.createStream(context, ...)  
    val words = lines.flatMap(_.split(" "))  
    val wordCounts = words.map(x => (x,1)).reduceByKey(_ + _)  
    wordCounts.print()  
    context.start()  
    context.awaitTermination()  
  }  
}
```

Deeper view

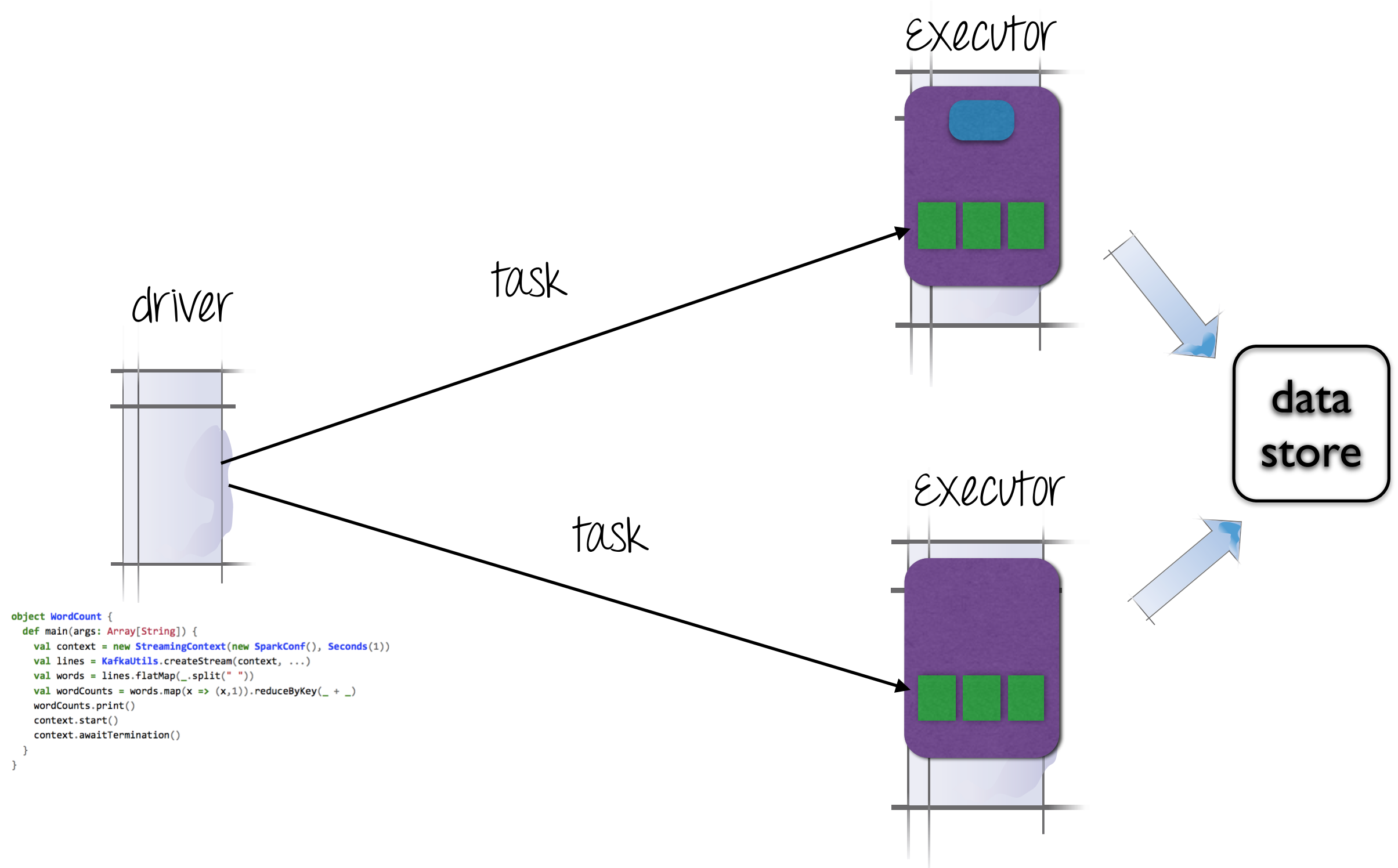
Any Spark Application



Spark Streaming: Receive data

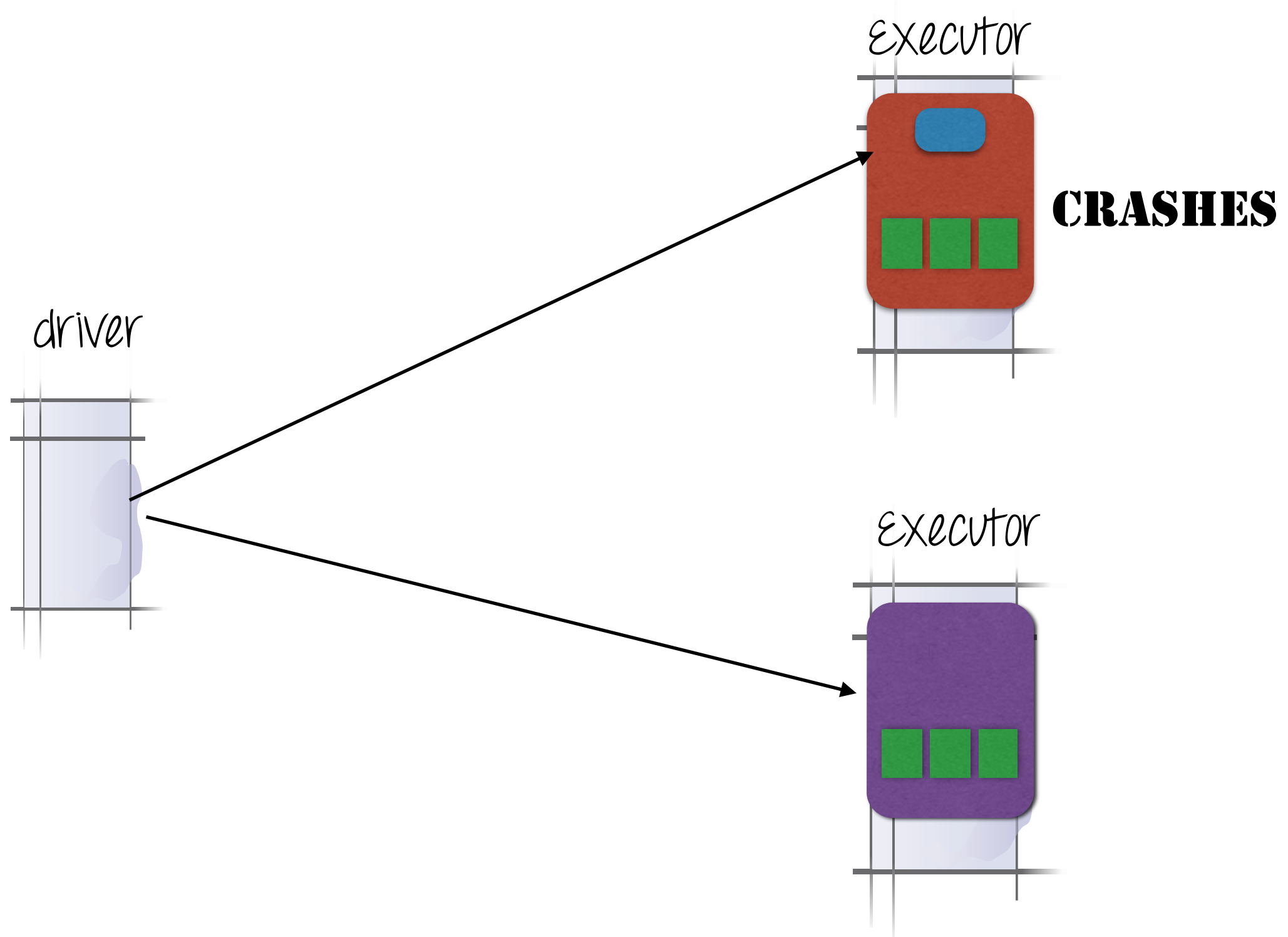


Spark Streaming: Process data

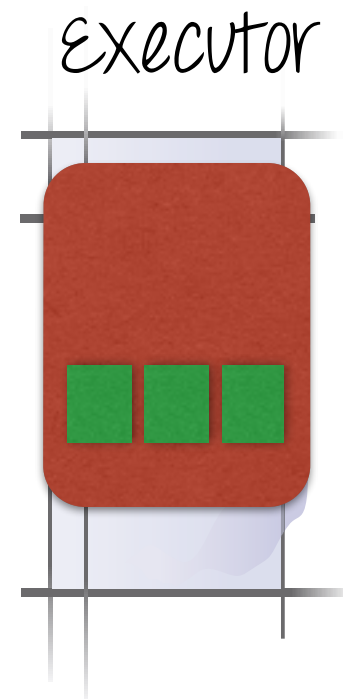
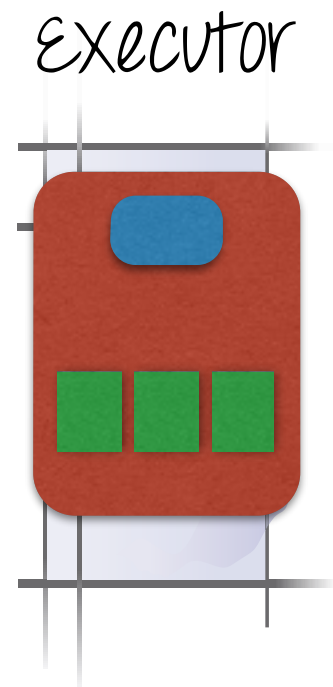
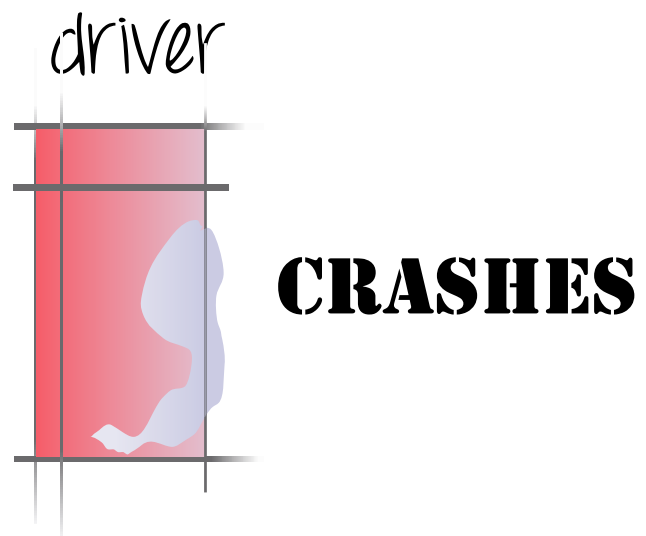




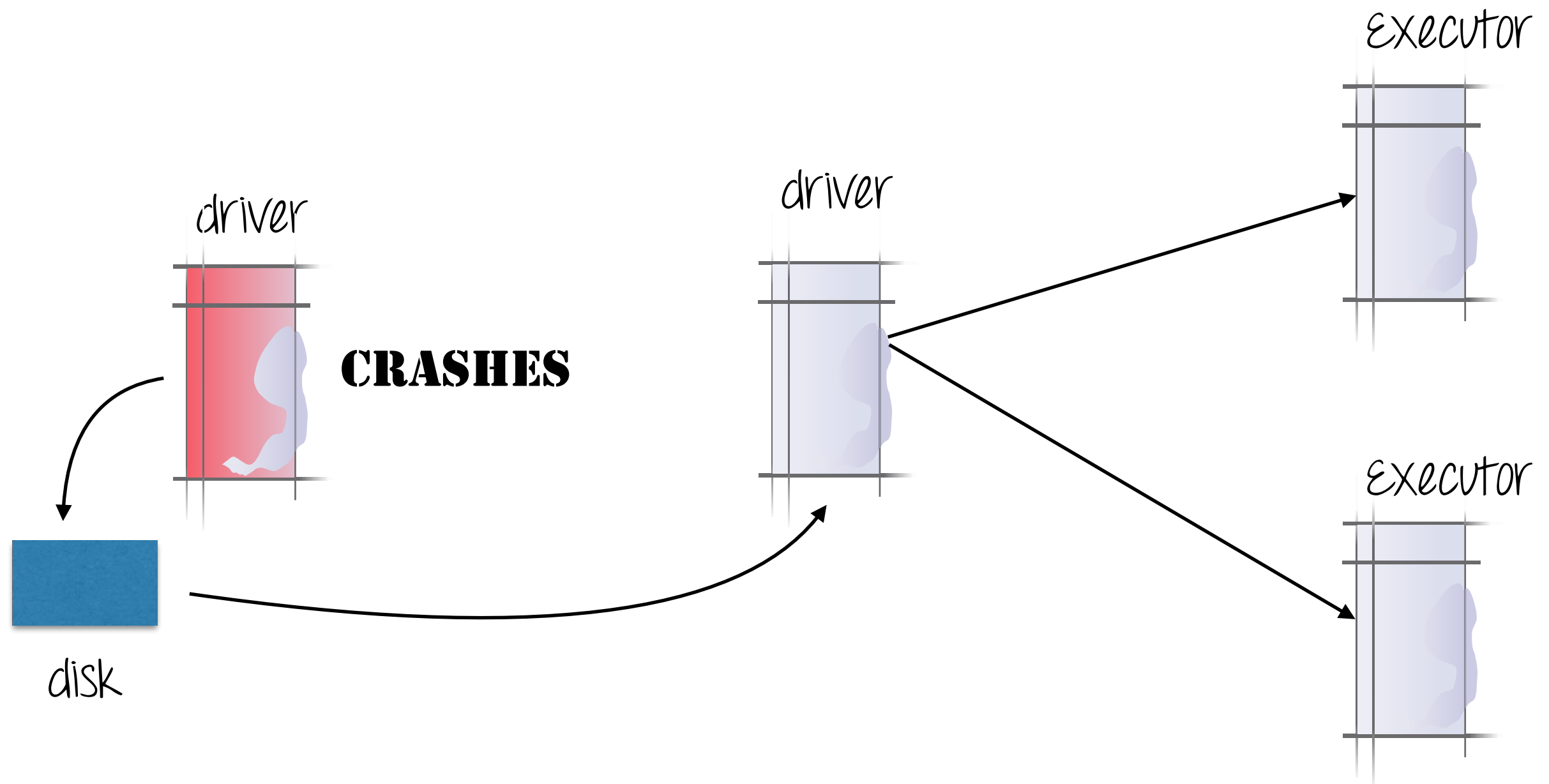
What if the executor fails



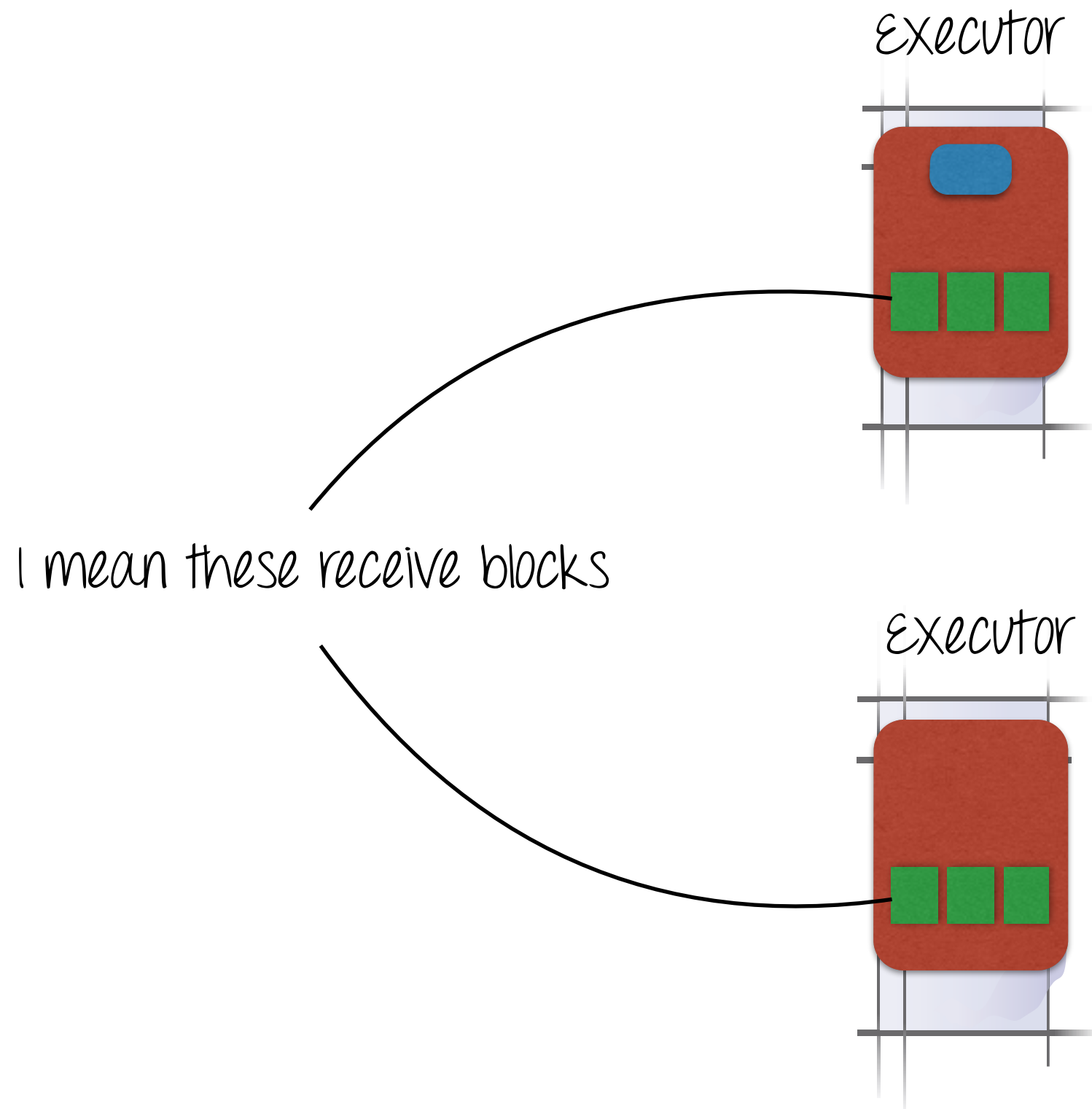
What if the driver fails



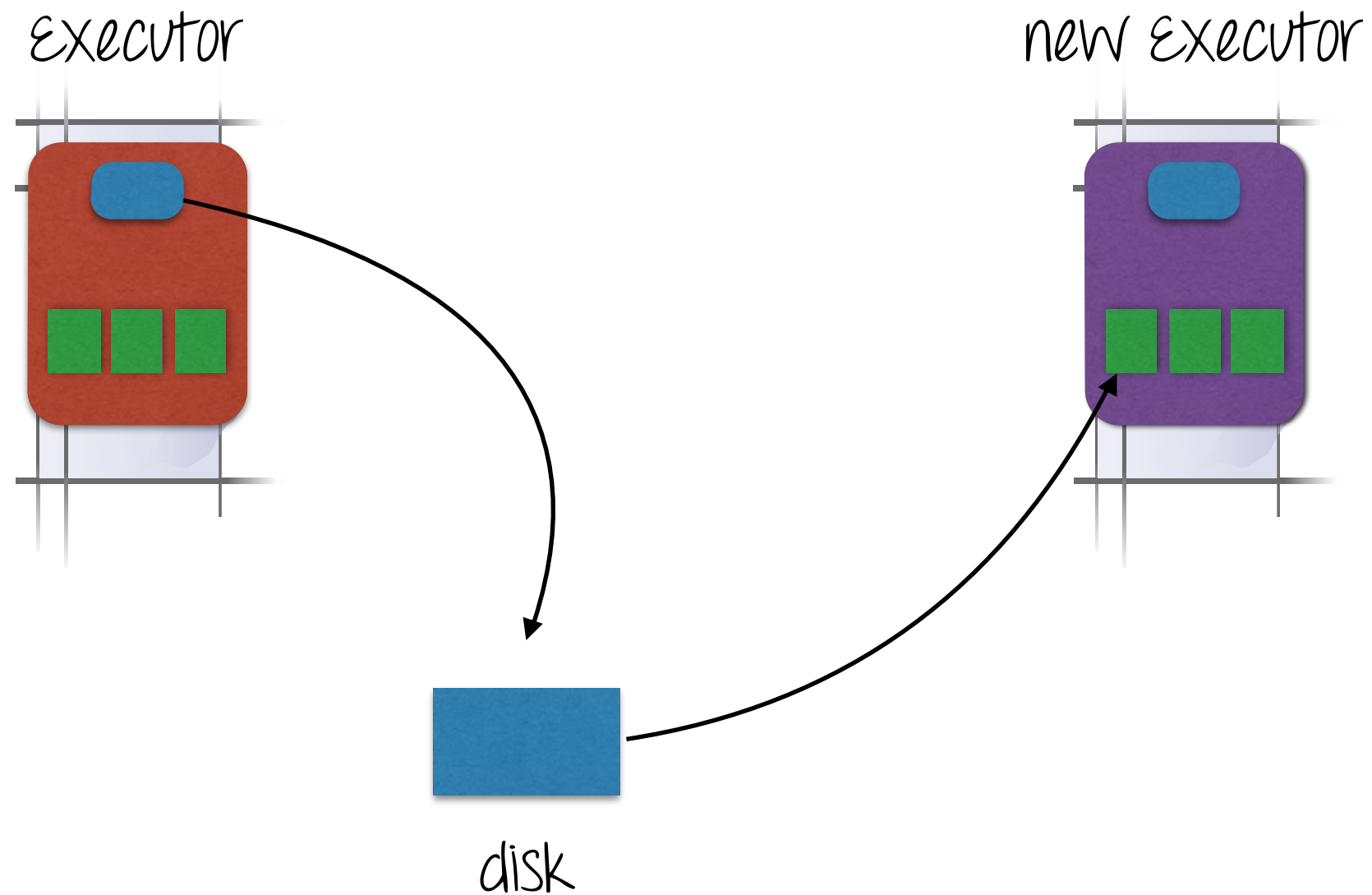
What if the driver fails - checkpointing



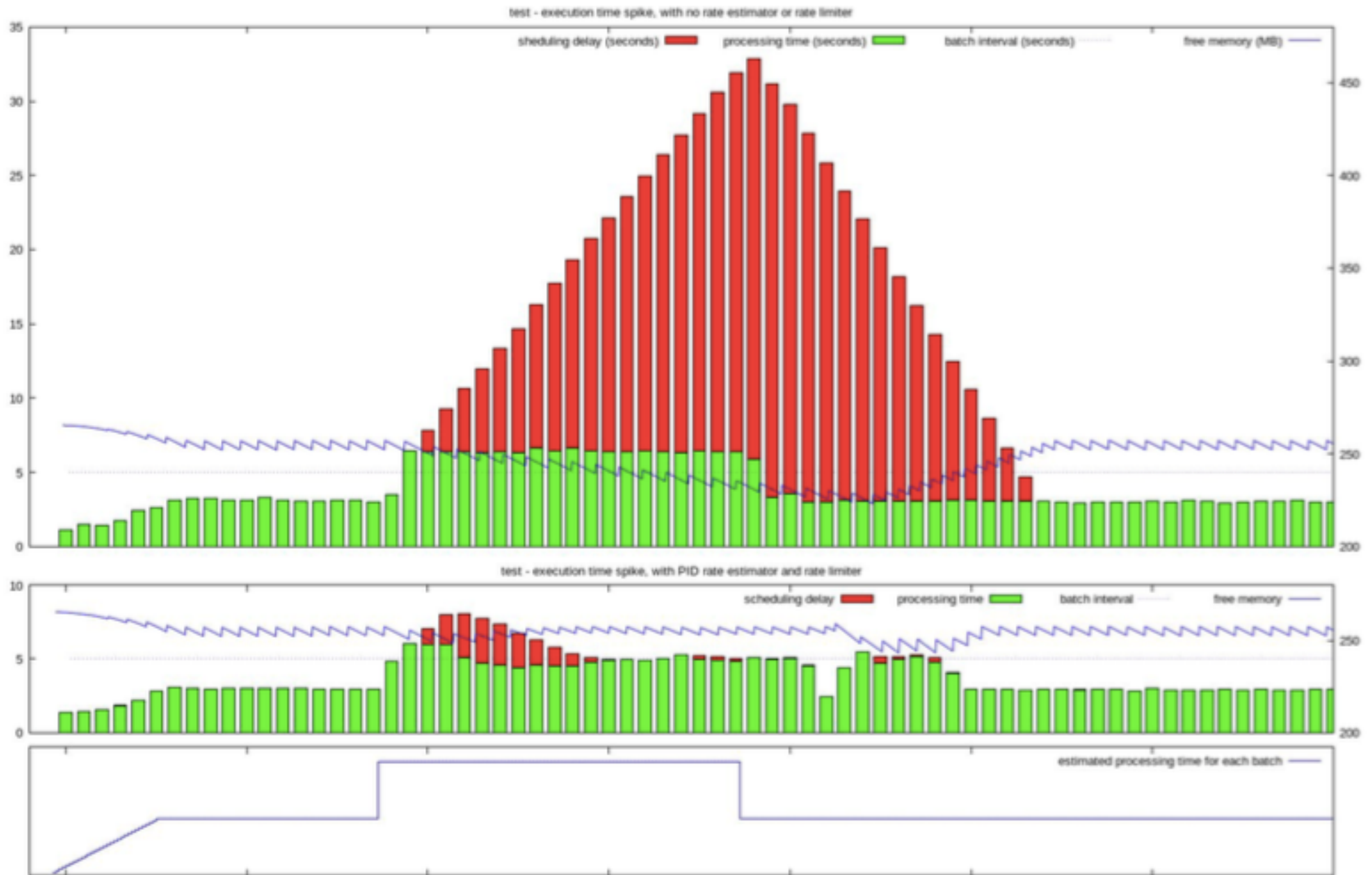
What if the data is lost?



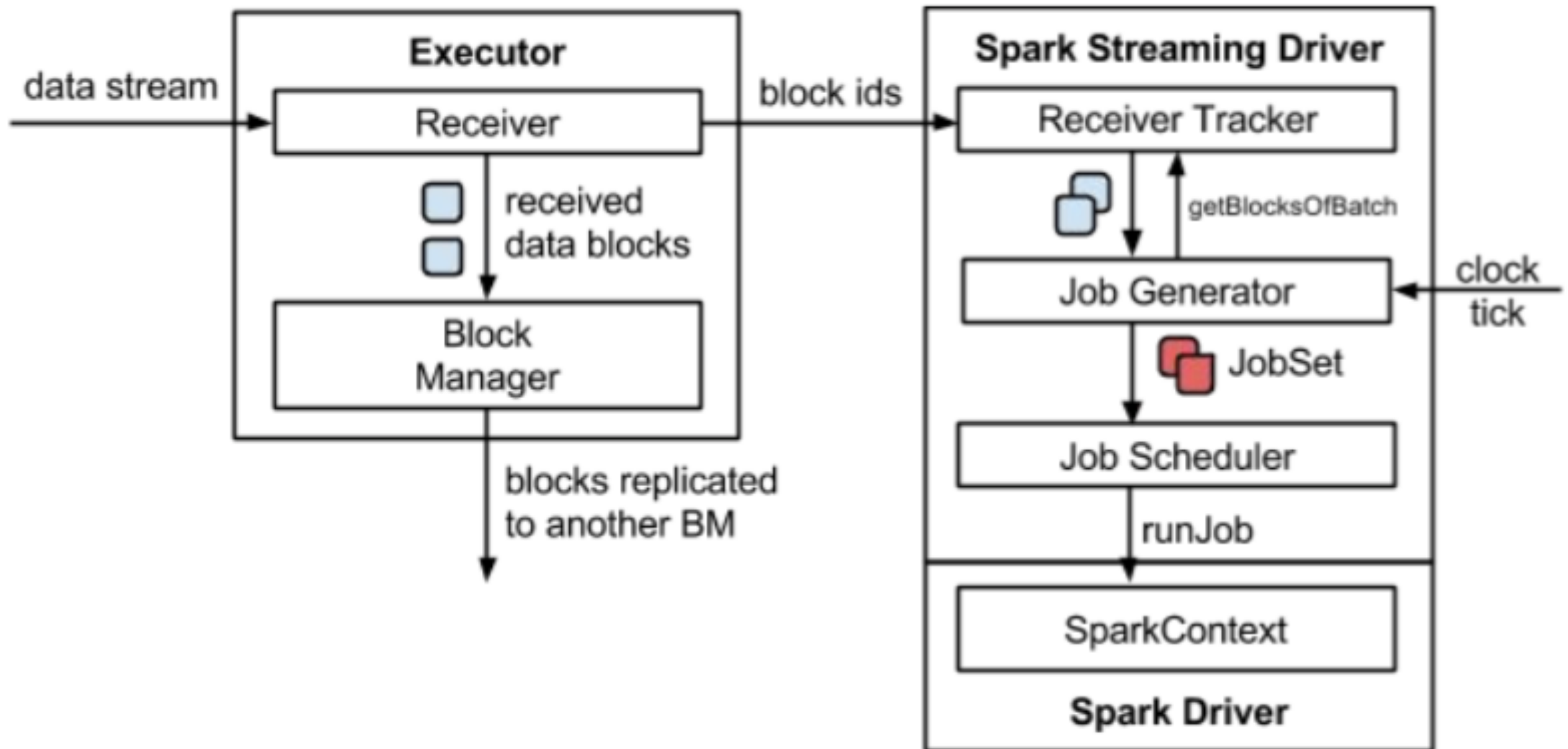
WAL - Write Ahead Log



What if there is a spike?

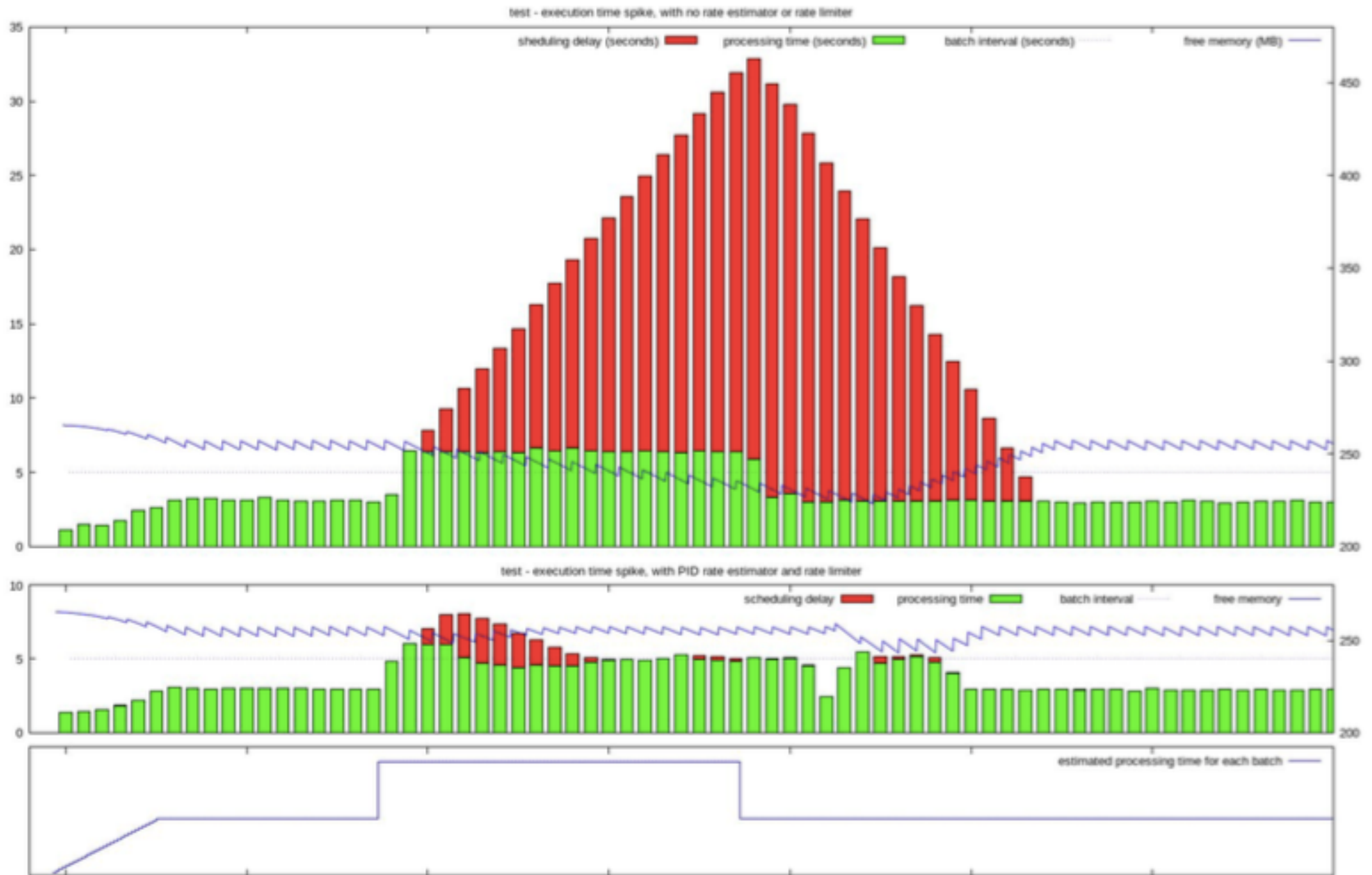


Going even deeper

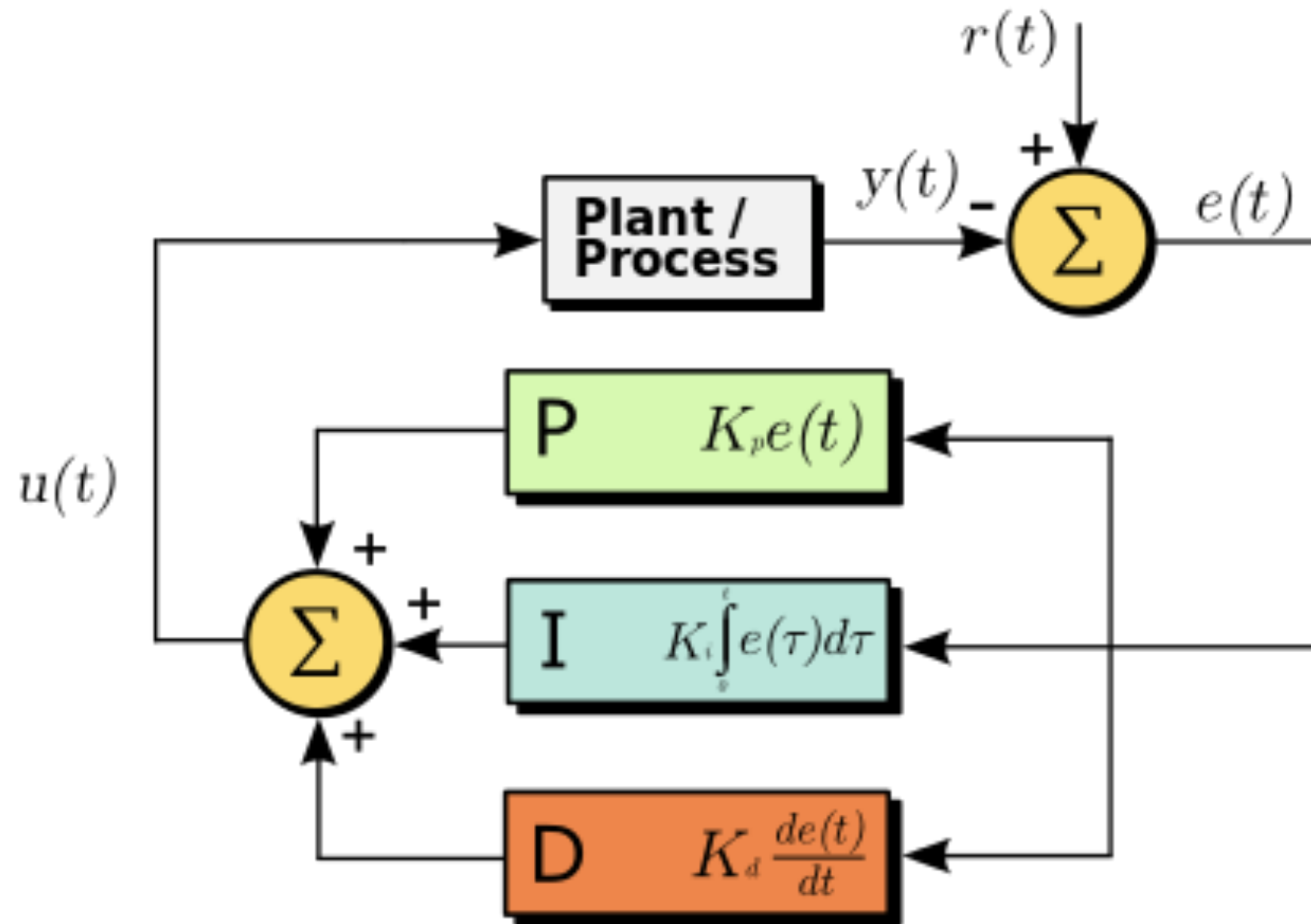


Static state limiter

What if there is a spike?

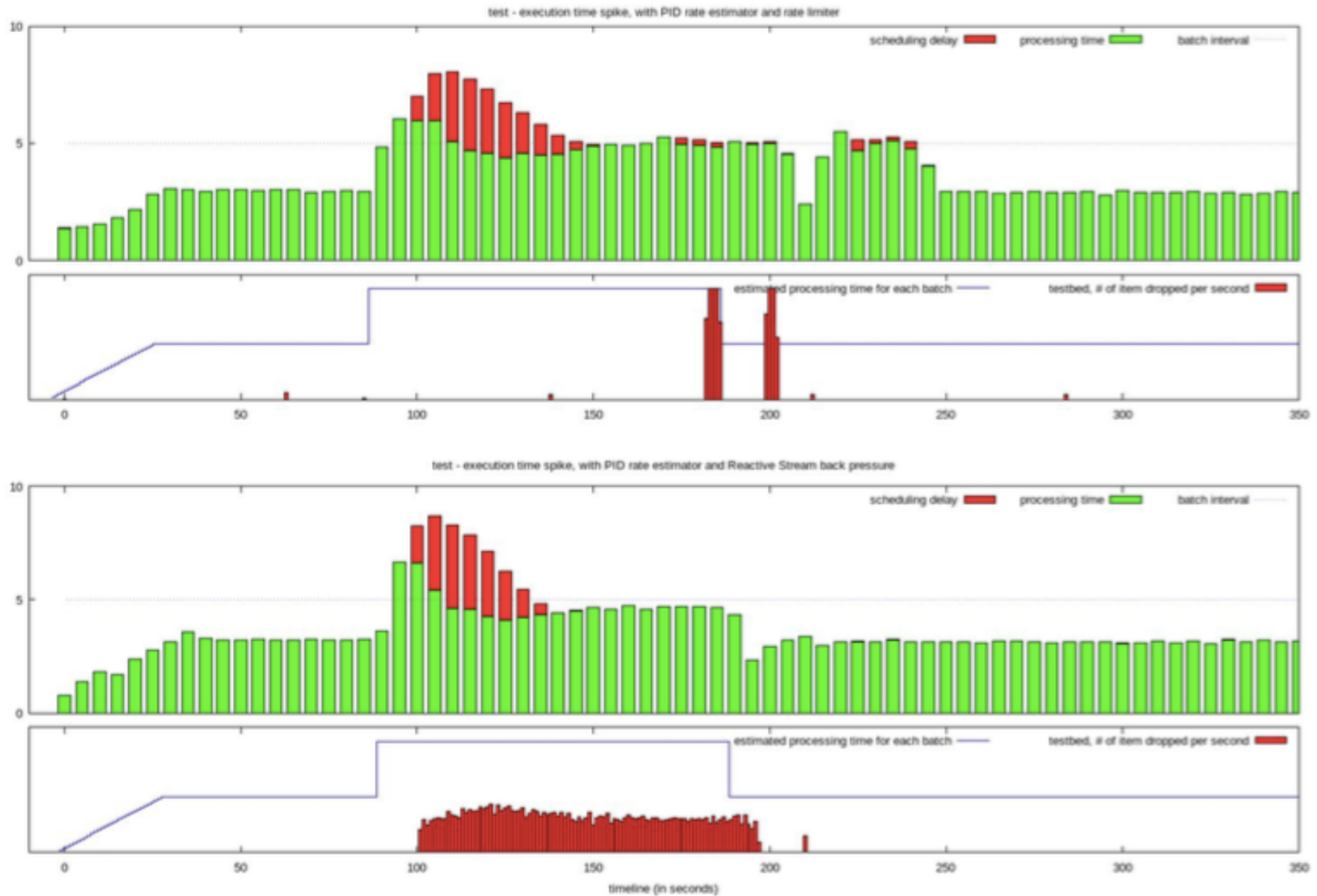


Dynamic rate control



https://en.wikipedia.org/wiki/File:PID_en_updated_feedback.svg

Going further - Reactive Streams





**KEEP
CALM
AND
STREAM
ON**

Q & Option[A]

@nraychaudhuri

