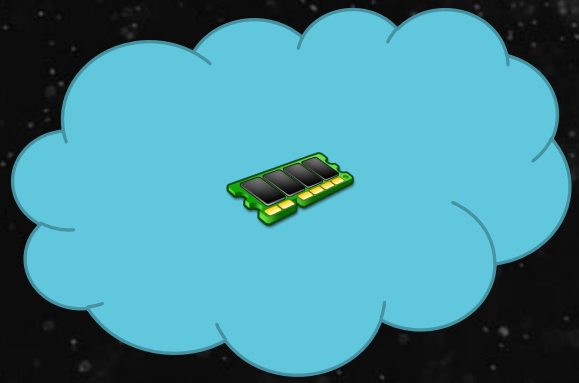
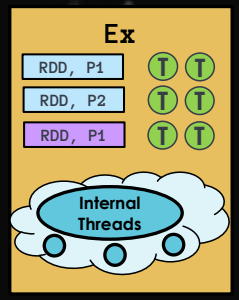


MEMORY AND PERSISTENCE

databricks

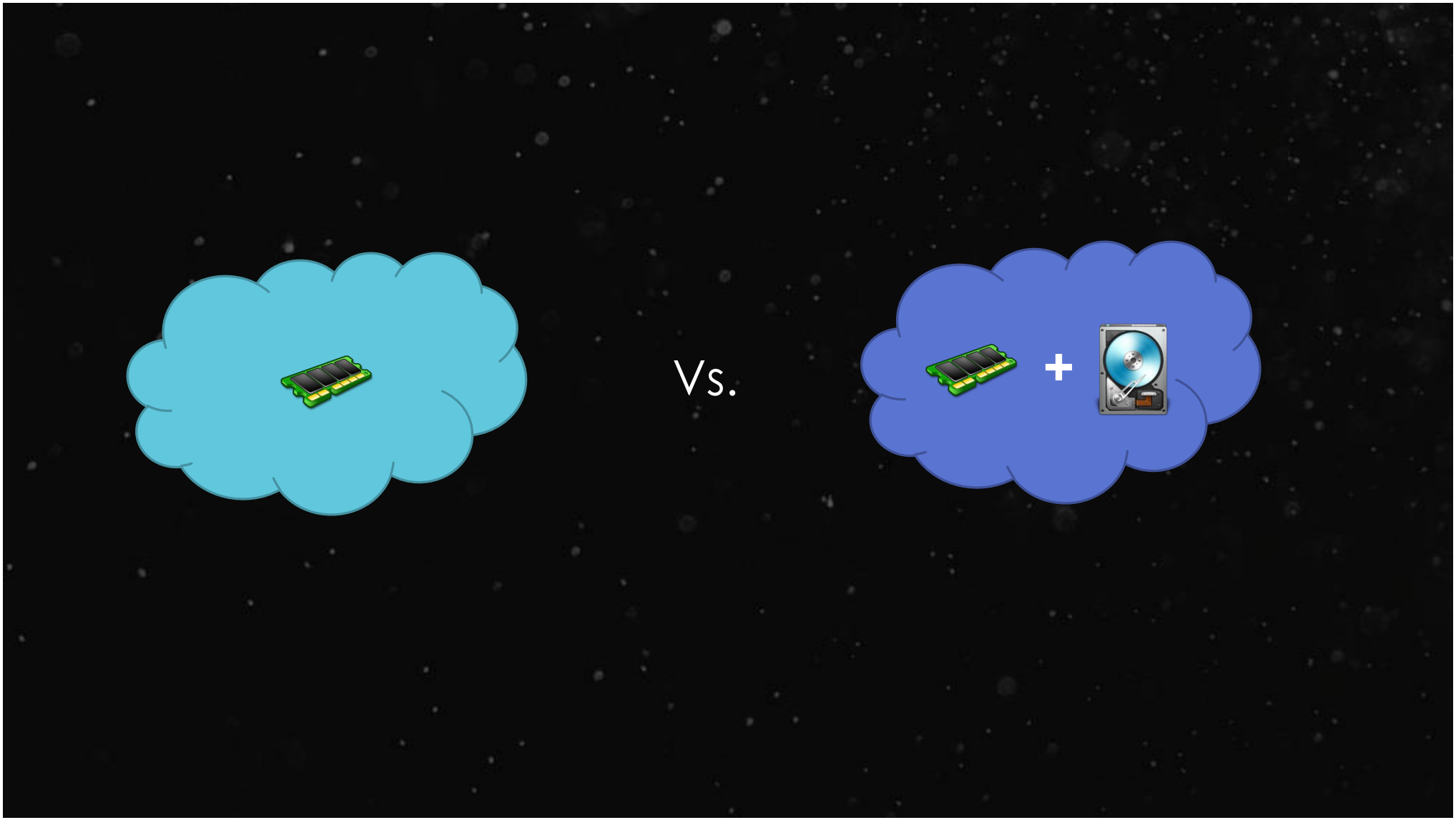


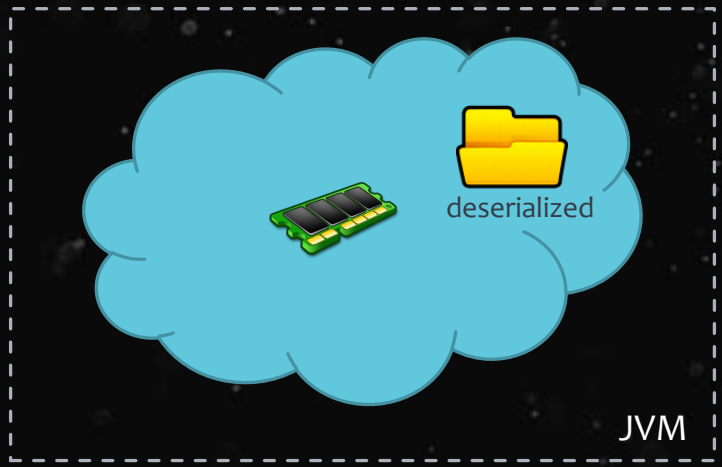
Recommended to use at most only 75% of a machine's memory for Spark

Minimum Executor heap size should be 8 GB

Max Executor heap size depends… maybe 40 GB (watch GC)

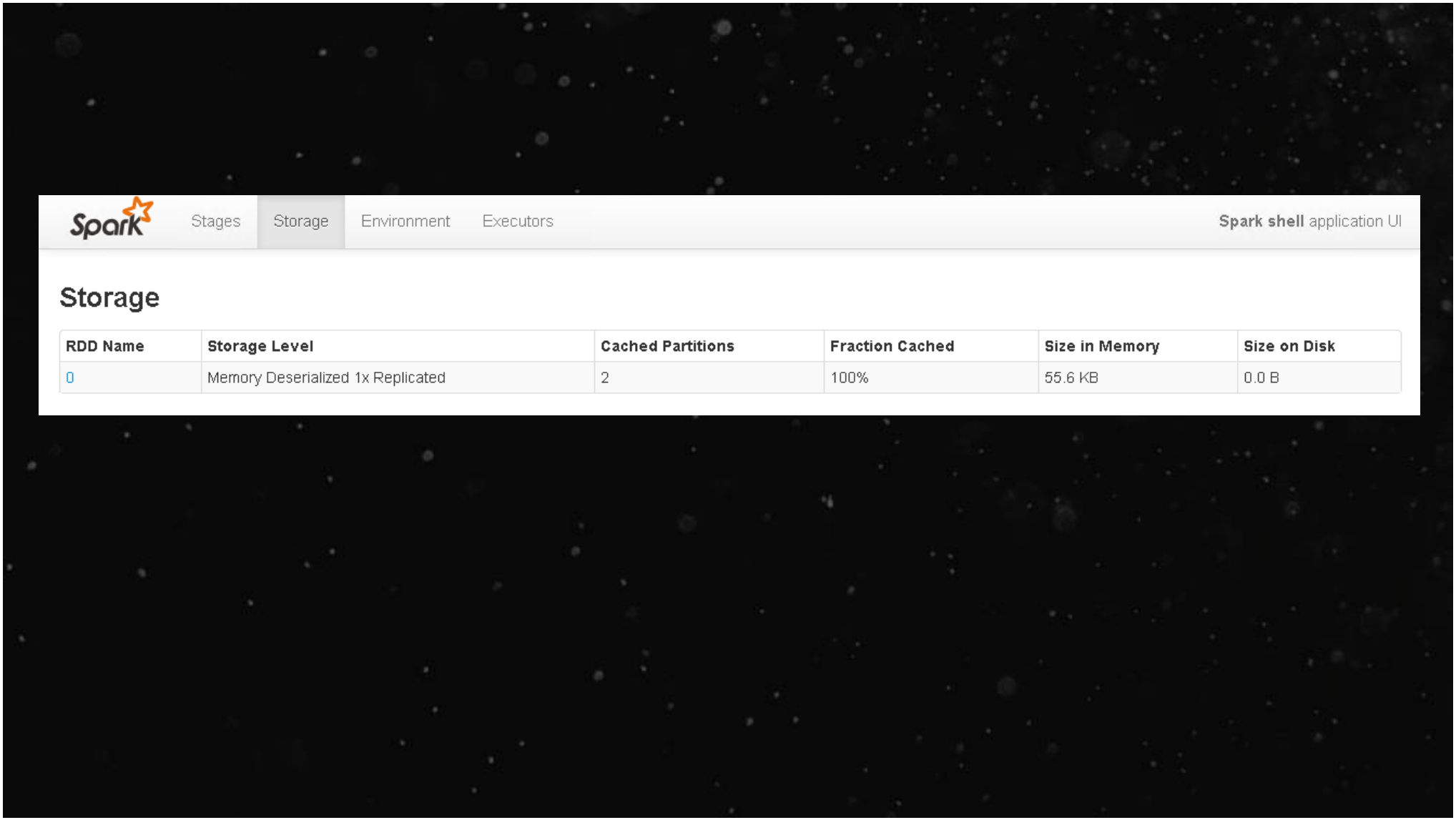
Memory usage is greatly affected by storage level and serialization format

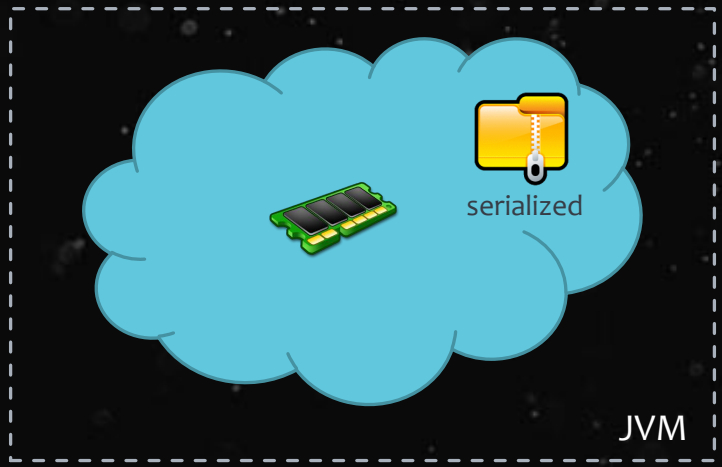

Vs.



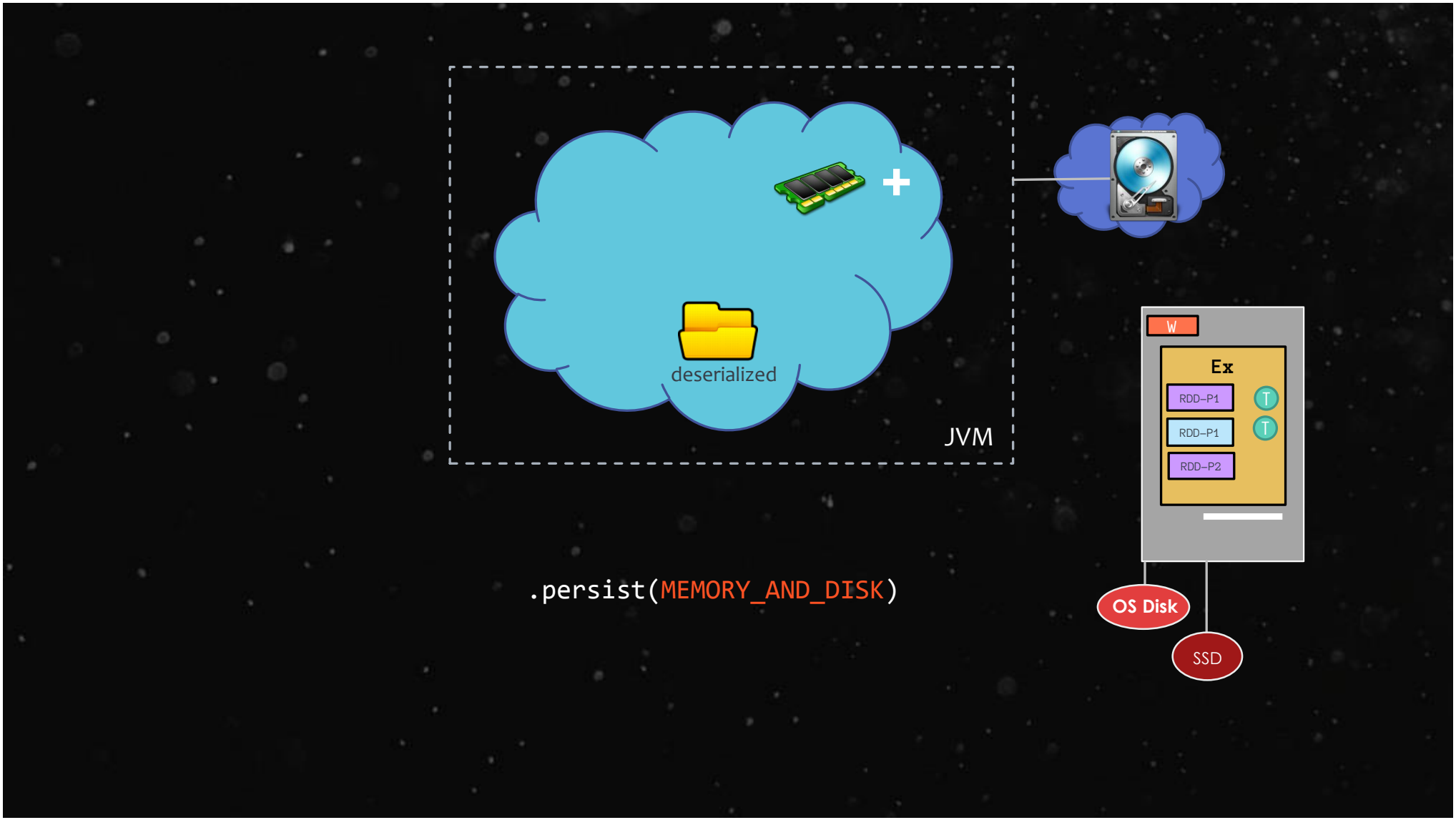

```
RDD.cache()  ==  RDD.persist(MEMORY_ONLY)
```

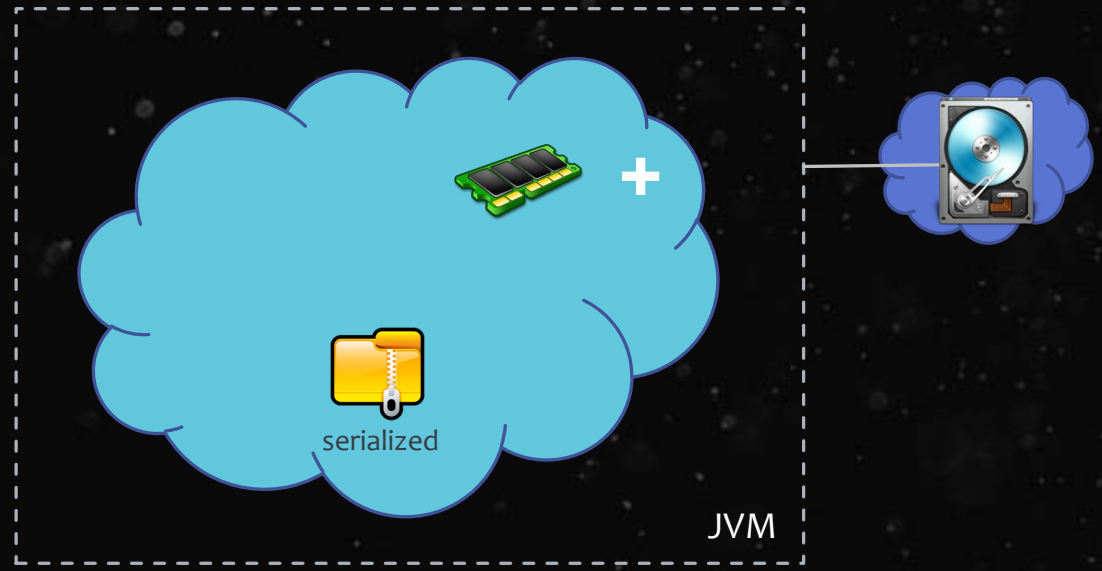
most CPU-efficient option


Spark
Stages
Storage
Environment
Executors
Spark shell application UI
Storage
RDD Name
Storage Level
Cached Partitions
Fraction Cached
Size in Memory
Size on Disk
0
Memory Deserialized 1x Replicated
2
100%
55.6 KB
0.0 B

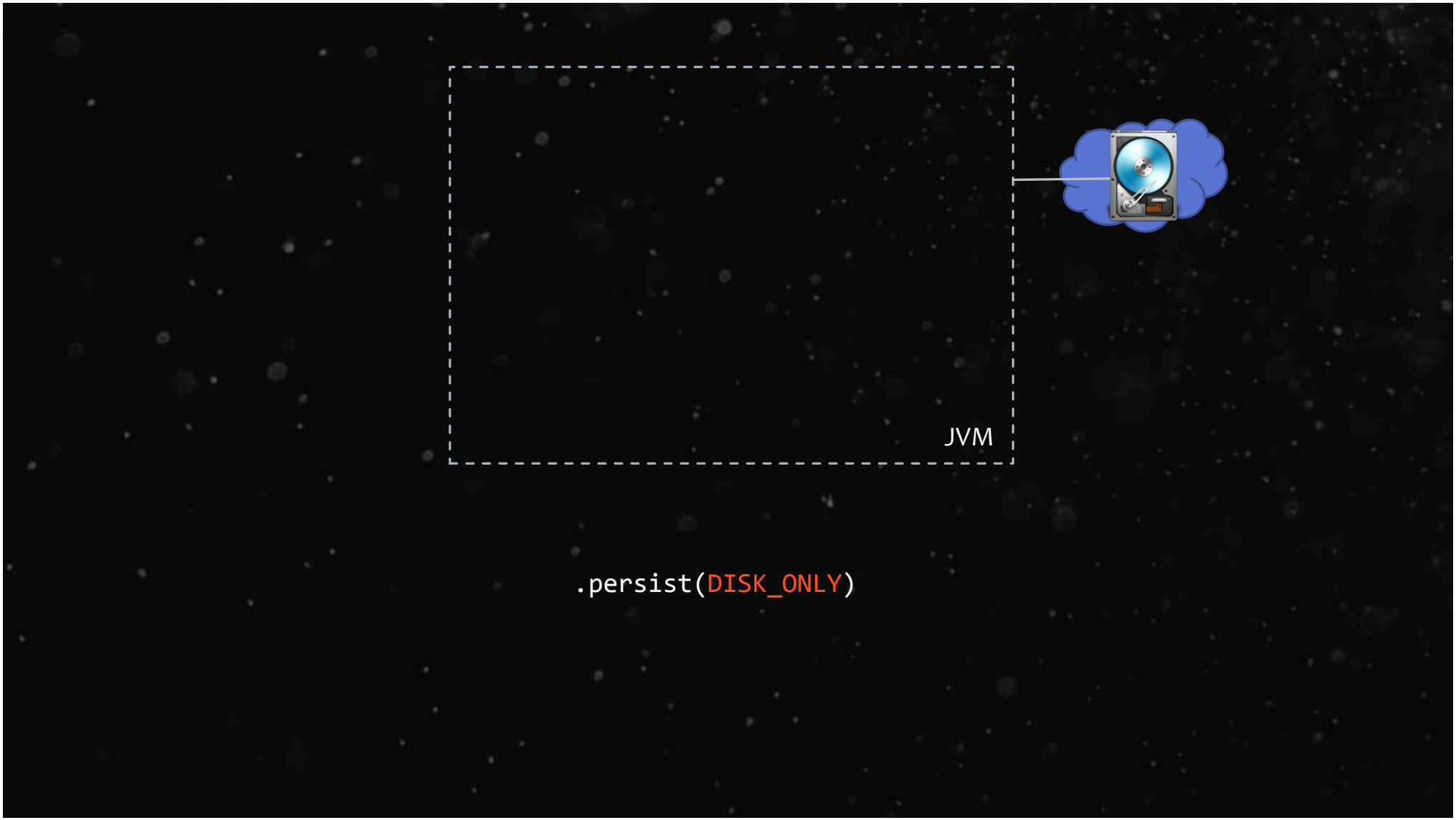


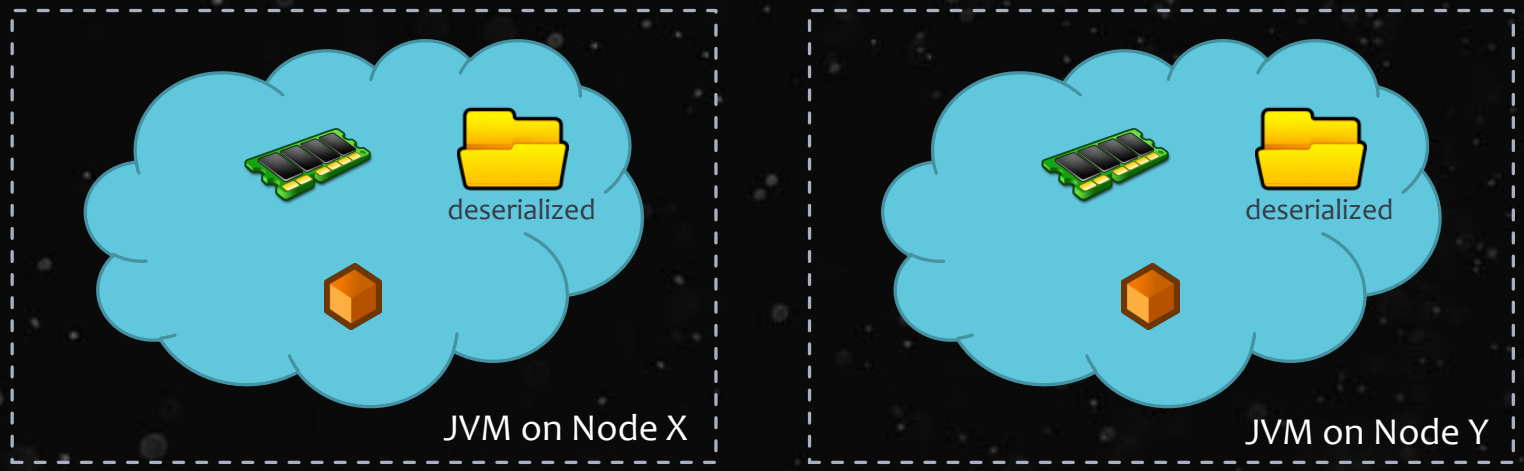

```
RDD.persist(MEMORY_ONLY_SER)
```


deserialized
JVM
W
Ex
RDD–P1
T
RDD–P1
T
RDD–P2
OS Disk
SSD
.persist(MEMORY_AND_DISK)

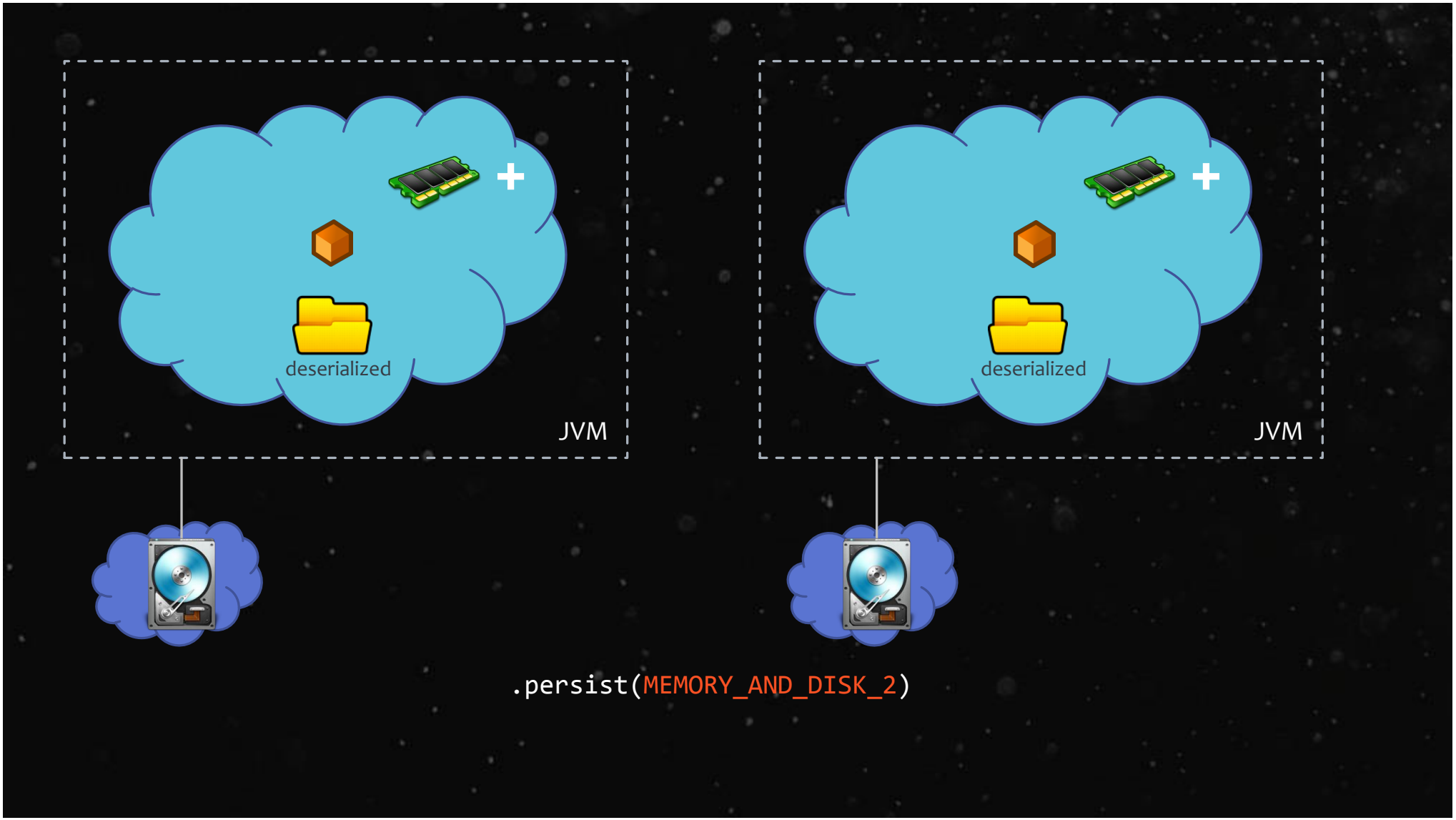


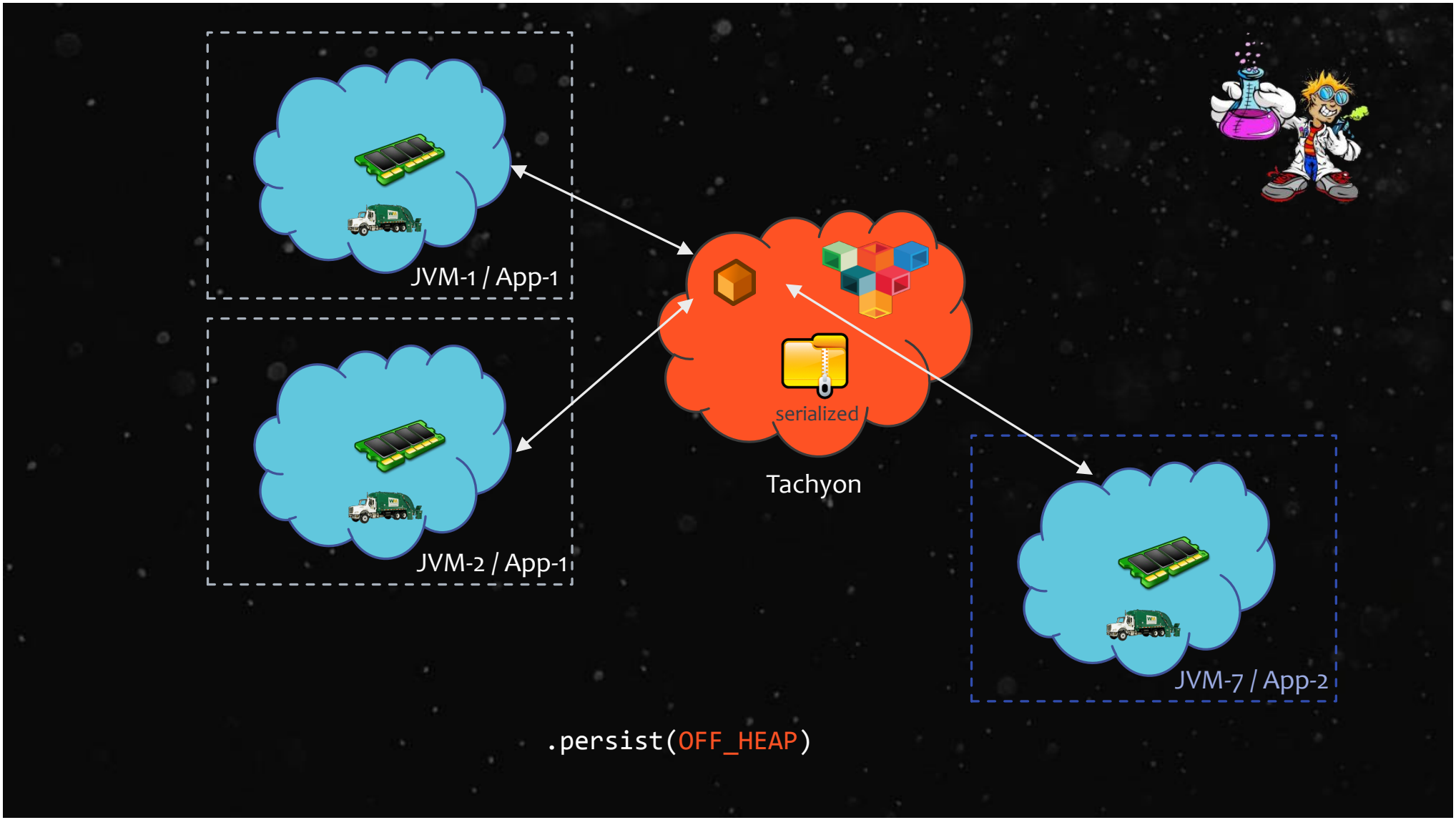

```
.persist(MEMORY_AND_DISK_SER)
```

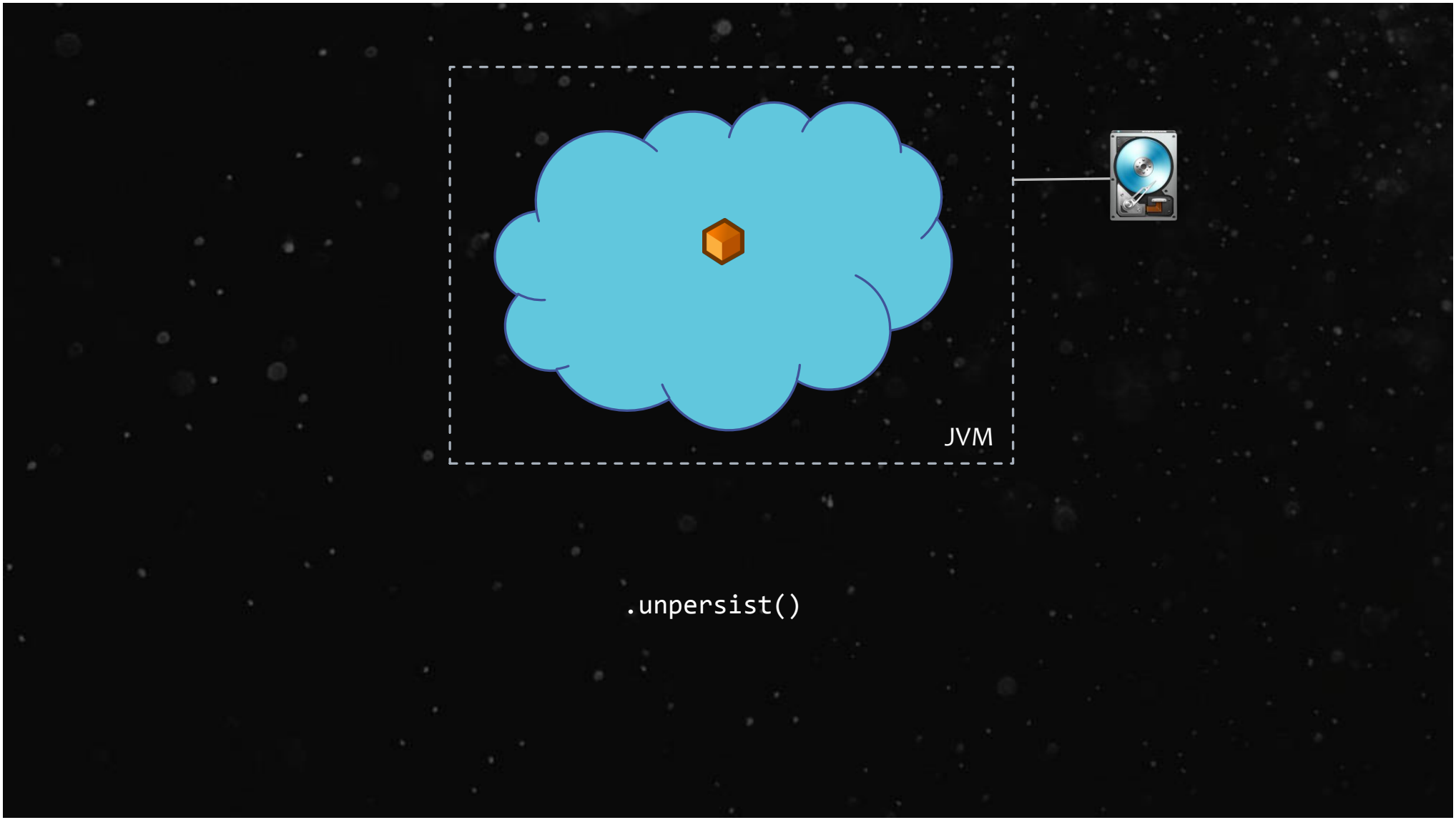

JVM
.persist(DISK_ONLY)

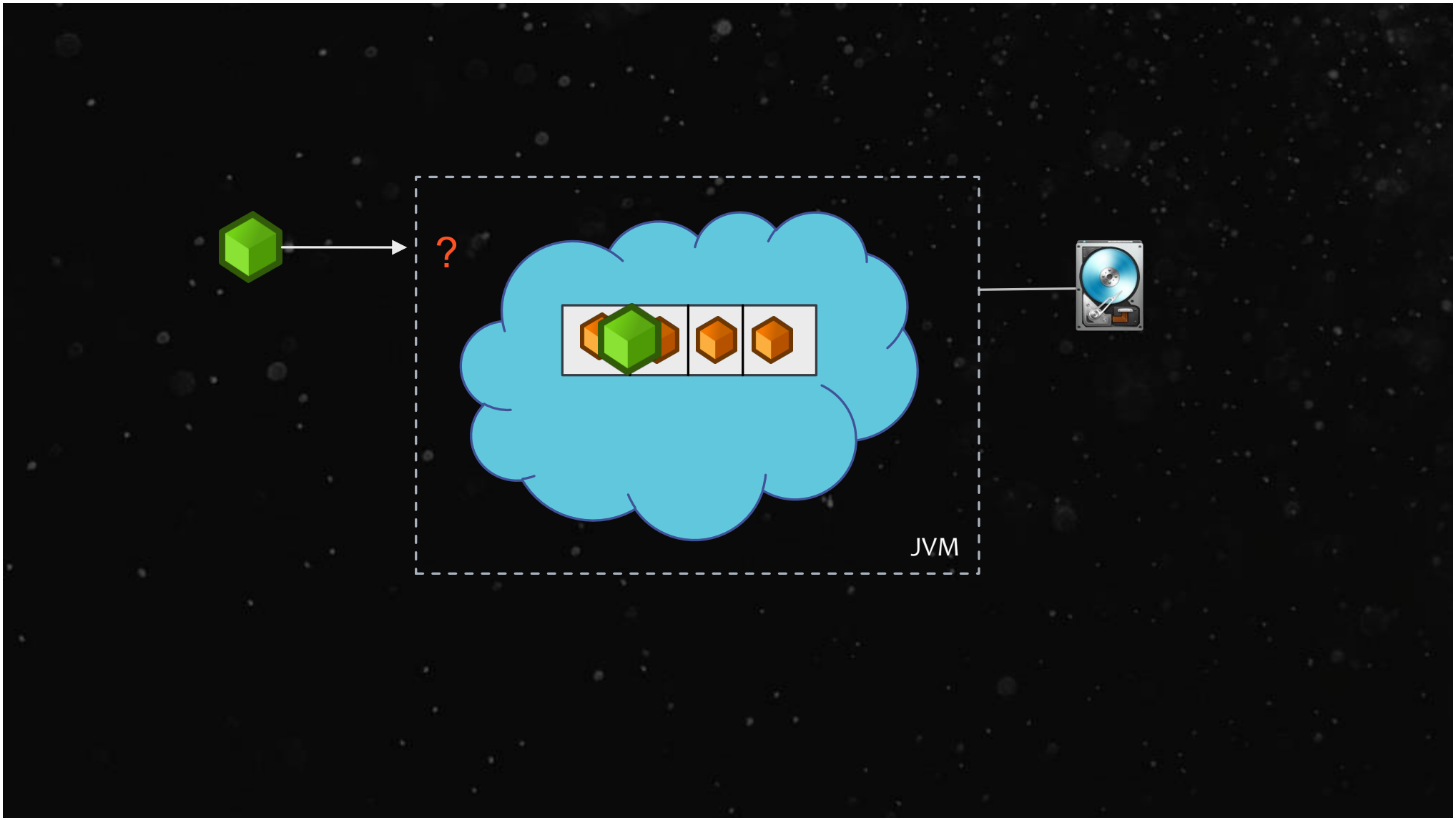



```
RDD.persist(MEMORY_ONLY_2)
```


deserialized
JVM
deserialized
JVM
.persist(MEMORY_AND_DISK_2)


JVM-1 / App-1
JVM-2 / App-1
serialized
Tachyon
JVM-7 / App-2
.persist(OFF_HEAP)


JVM
.unpersist()


?
JVM

?

- If RDD fits in memory, choose MEMORY_ONLY
- If not, use MEMORY_ONLY_SER w/ fast serialization library
- Don't spill to disk unless functions that computed the datasets are very expensive or they filter a large amount of data. (recomputing may be as fast as reading from disk)
- Use replicated storage levels sparingly and only if you want fast fault recovery (maybe to serve requests from a web app)

Remember!

Intermediate data is automatically persisted during shuffle operations

PySpark: stored objects will always be serialized with Pickle library, so it does not matter whether you choose a serialized level.