# REDIS SOLUTIONS PROPOSAL

This document’s aim is to explain different redis solutions approaches and consider their advantages and disadvantages.

**Terms :**

**No SPOF** ( No Single Point of Failure ) : If one of the physical component of cluster system fails, the cluster system can still survive.

**CAP Theorem :** Consistency, Availability, Partition Tolerant

**CP :** Consistency, Partition Tolerant ( Horizontal Scalability )

**AP :** Availability, Partition Tolerant ( Horizontal Scalability )

**DC :** Data Center

**DR :** Disaster Recovery

**RACK :** Logical group of nodes or instances

**QUORUM VOTE :** Majority vote of cluster components.

**DC QUORUM :** Dynomite Data Center Level Quorum Consistency

**DC ONE :** Dynomite Data Center Level One Node Consistency

**DynoBiC :** Dynomite Cluster

**ReBiC :** Redis Cluster

**Sharding** : Breaking a single database into smaller, more manageable chunks, and distributing those chunks across multiple servers, in order to spread the load and maintain a high throughput.

**Partitioning :** A data partition is a mechanism for ensuring that data is evenly distributed across a cluster.

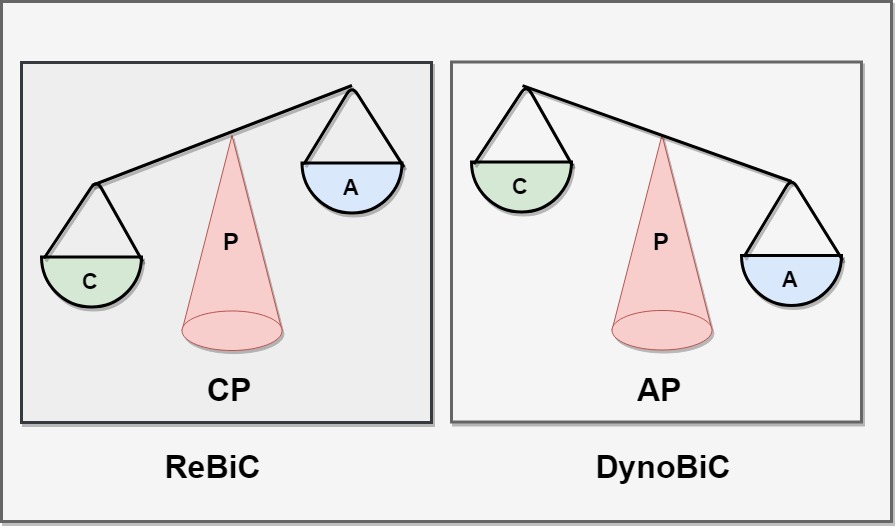
## Description Of The Situation

To support of requirements, we need very fast data management structures which can be scale horizontally. Redis seems to be good option to response this expectation. However, there are several Redis cluster solutions and all of them have some advantages over others besides disadvantages. Therefore, we should consider which solution approach is more appropriate for our project needs.

# THE CAP ISSUES

For our Perspective **Horizontal Scability ( Partition Tolerance )** is **MUST**

Hence, we have to choose **CP** or **AP** solution.

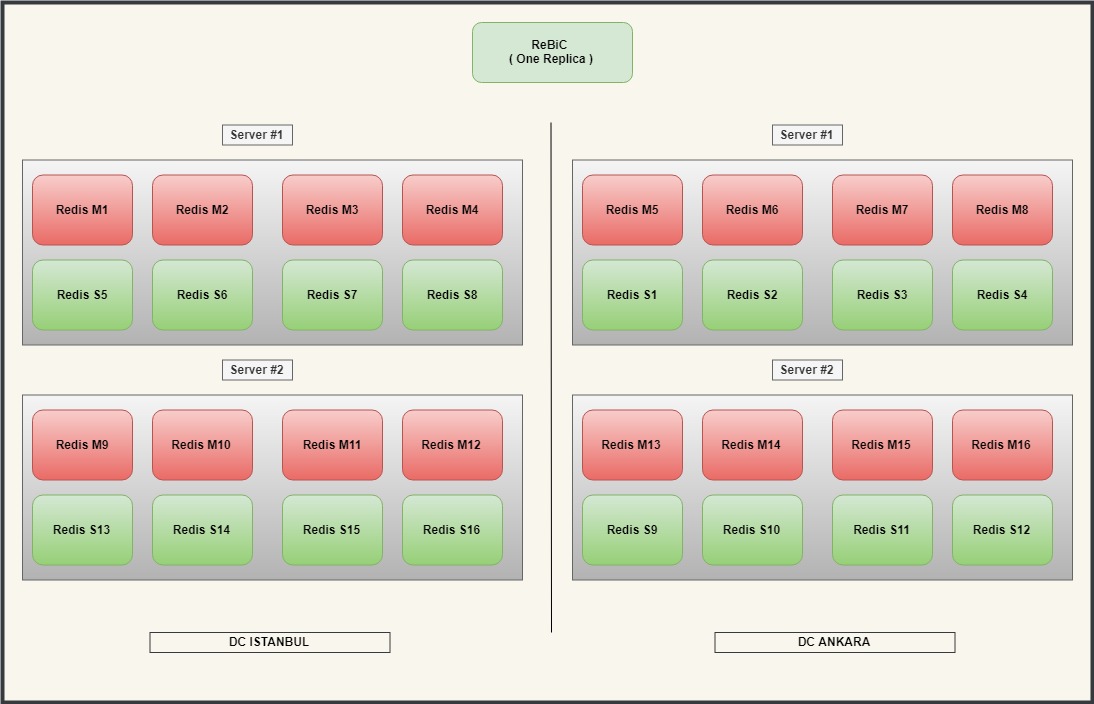


**ReBiC :** Pure Redis sharding base cluster solution.

**DynoBiC :** Dynomite partition base ( Dynamo logic ) redis proxy cluster solution.

Both Two Solutions are **No SPOF**

## ReBiC ( One Replica-Slave, Two DC Redis Cluster )



**Advantages:**

* Enhance consistency
* Low latency in same dc
* High throughput in same dc
* Online upgrade
* Comparatively more memory capacity
* Comparatively simple management
* Easy Scaling

**Disadvantages:**

* Higher latency over different dc
* Low throughput over different dc
* Comparatively less high availability level ( If DC which has majority of master nodes of the Cluster is down, Cluster is down )

## DynoBiC - DC\_QUORUM ( Dynomite Redis Cluster, 6 Replica)



**Advantages:**

* Enhance availability.
* Enhance multi-DC DR ( Disaster Recovery ) capacity
* Low latency for multi-DC environment
* High throughput for multi-DC environment

**Disadvantages:**

* Has consistency issues.
* Comparatively less memory capacity ( 1/6 )
* More difficult management
* More difficult Scaling and node replacement
* Higher latency in same dc operations
* Low throughput in same dc operations

## DynoBiC - DC\_ONE ( Dynomite Redis Cluster, 4 Replica)



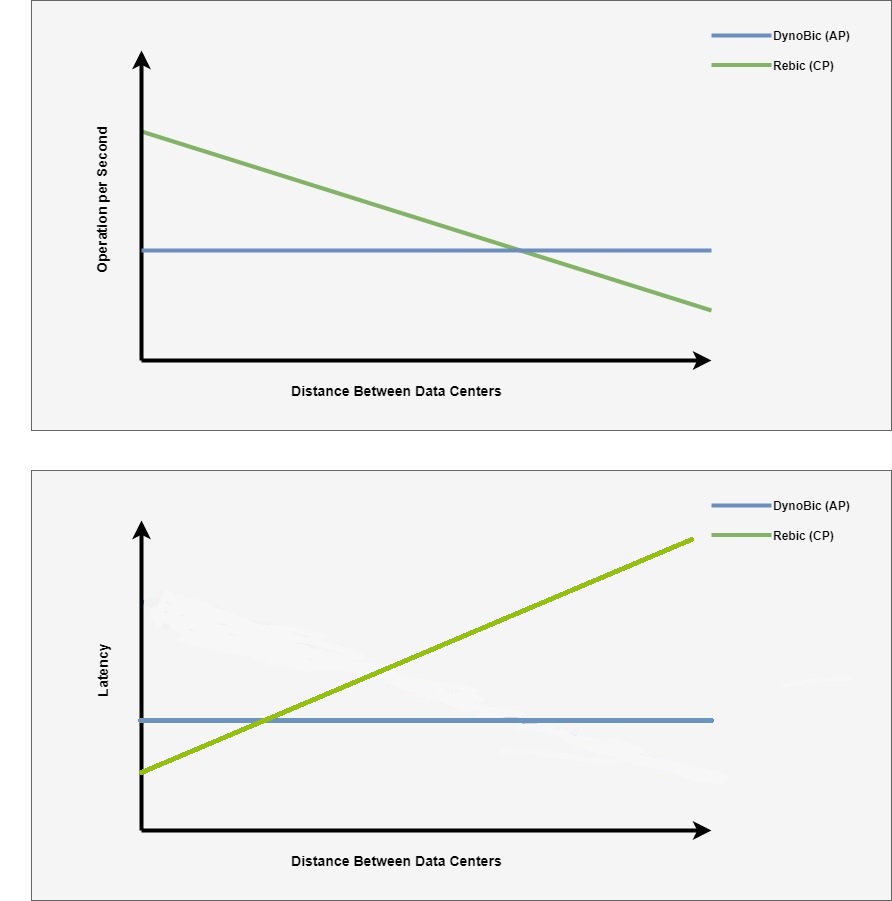
**Advantages, according to DC\_QUORUM :**

* More memory capacity ( 1/4 )
* More write and read availability
* Slightly less latency

**Disadvantages, according to DC\_QUORUM :**

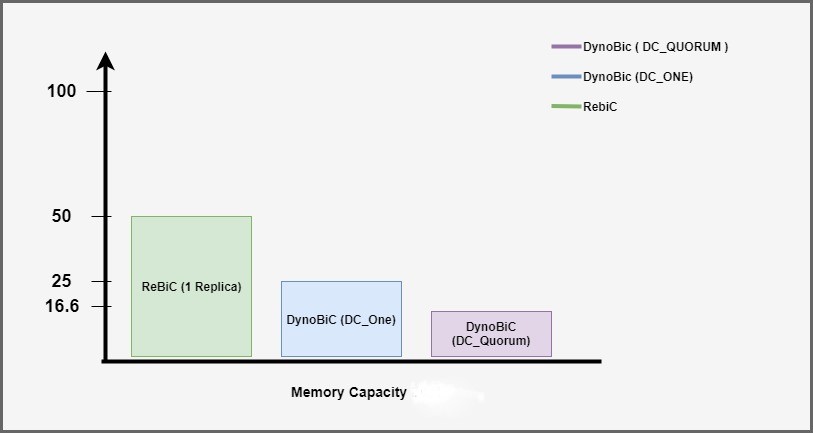
* More inconsistency ( might be exposed inconsistency inside same data center ) .

## Performans



* **Initially** ReBic has more performance
* While distance between data center increases, DynoBiC performance **will not** be affected.

## Memory Capacity



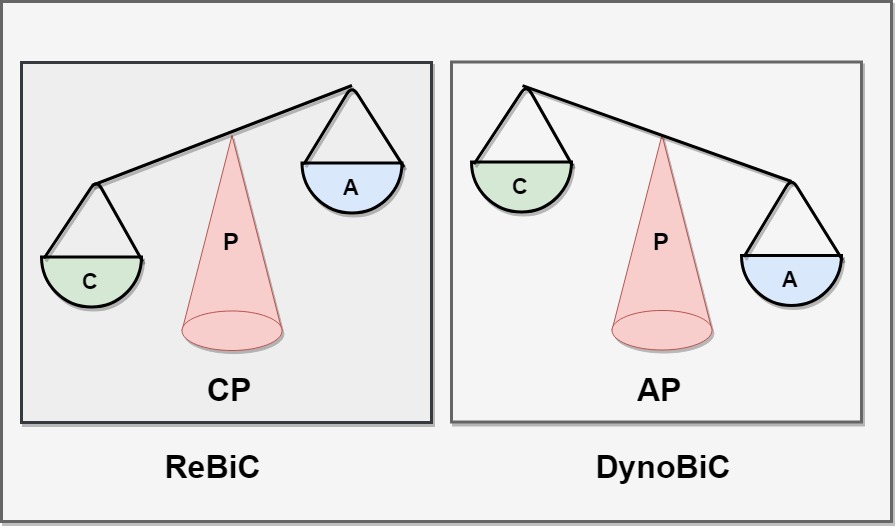
ReBiC ( One Replica-Slave ) **%50**

DynoBiC - DC\_ONE ( 4 Replica ) **%25**

DynoBiC - DC\_QUORUM ( 6 Replica ) **%16.6**

## Conclusions

We have to choose **CP** or **AP** solution.



But, We **DO NOT** have to choose **ReBic** or **DynoBiC**

We can use **BOTH** **ReBic** and **DynoBiC** **at the same time**, For different data sets

***This document is produced by Mustafa YAVUZ***