GR Reference Models

GR Reference Models

The CPS solution stores session data in a document-oriented database. The key advantage is that the application layer responsible for transactional Session data is stored in MongoDB (document-oriented database). Data is replicated to help guarantee data integrity. MongoDB refers to replication configuration as replica sets as opposed to Master/Slave terminology typically used in Relational Database Management Systems (RDBMS).

Replica sets create a group of database nodes that work together to provide the data resilience. There is a primary (the master) and 1..n secondaries (the slaves), distributed across multiple physical hosts.

MongoDB has another concept called Sharding that helps scalability and speed for a cluster. Shards separate the database into indexed sets, which allow for much greater speed for writes, thus improving overall database performance. Sharded databases are often setup so that each shard is a replica set.

The replica set model can be easily extended to a Geo-redundant location by stretching the set across two sites. In those scenarios, an Arbiter node is required. The Arbiter is used as a non-data-processing node that helps decide which node becomes the primary in the case of failure. For example, if there are four nodes: primary, secondary1, secondary2 and the arbiter, and if the primary fails, the remaining nodes “vote” for which of the secondary nodes becomes the primary. Since there are only two secondaries, there would be a tie and failover would not occur. The arbiter solves that problem and “votes” for one node breaking the tie.

Without Session Replication

The following list provides information related to GR without session replication:

- If PCEF elements need to switch over clusters, the current Diameter session between the PCEF and PCRF will be terminated and a new session will need to be re-established.
- Simplifies architecture and reduces complexity.
- Quota data not reported. Currently, this is a limitation.

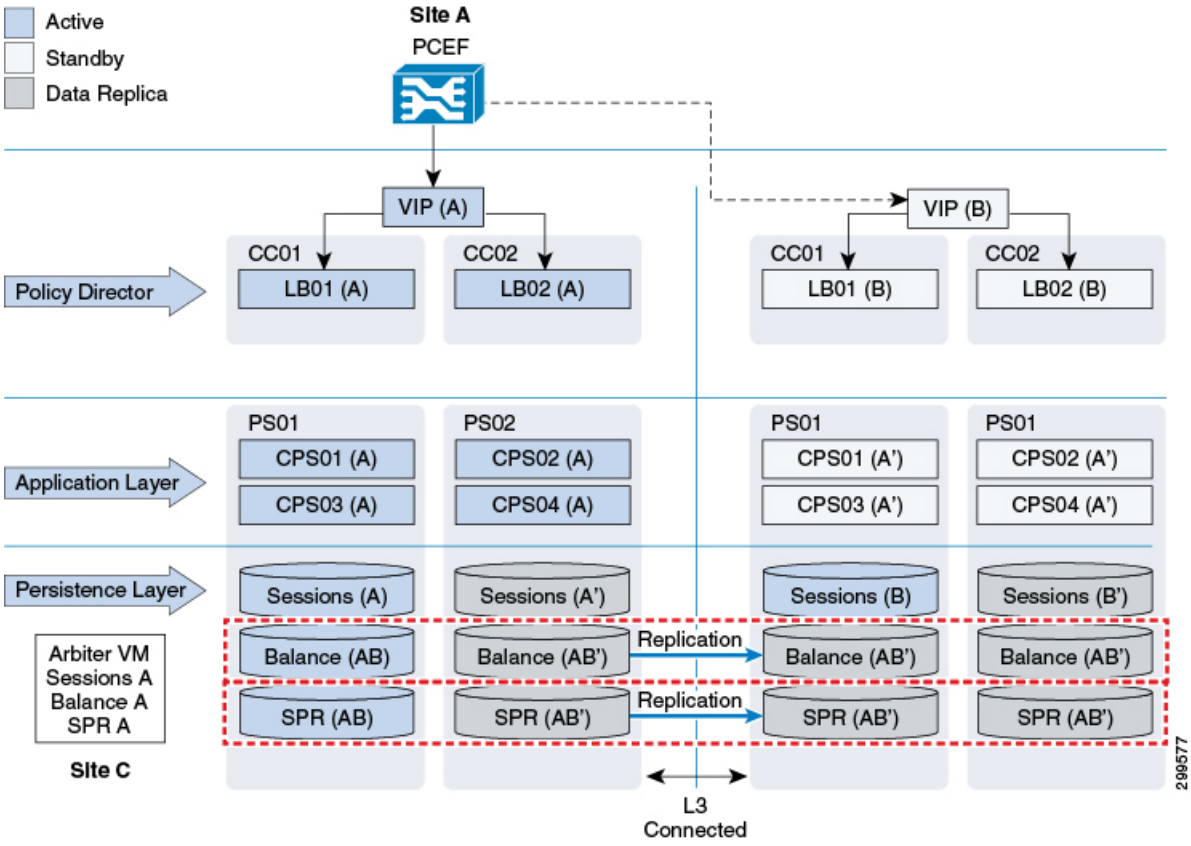
Active/Standby

In active/standby mode, one CPS system is active while the other CPS system, often referred to as the Disaster Recovery (DR) site, is in standby mode. In the event of a complete failure of the primary CPS cluster or the loss of the data center hosting the active CPS site, the standby site takes over as the active CPS cluster. All PCEFs use the active CPS system as primary, and have the standby CPS system configured as secondary.

The backup CPS system is in standby mode; it does not receive any requests from connected PCEFs unless the primary CPS system fails, or in the event of a complete loss of the primary site.

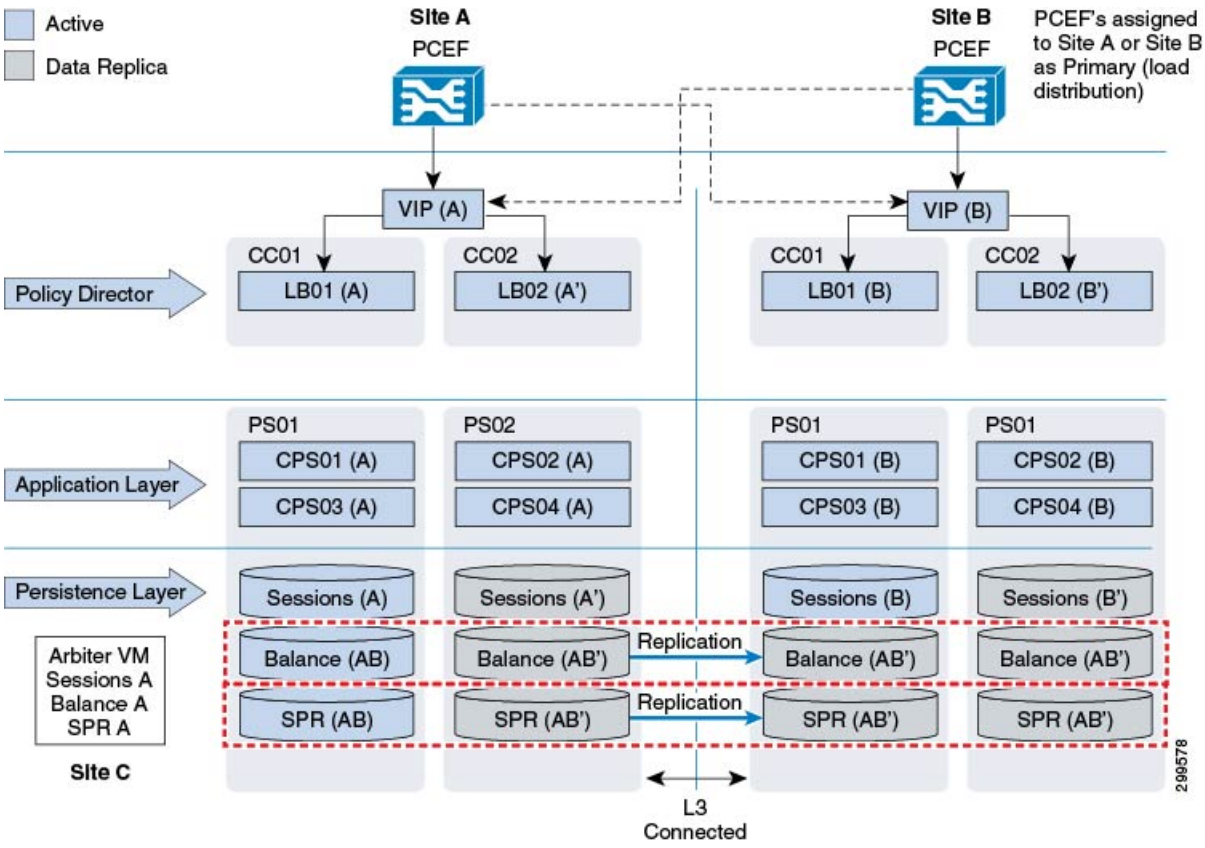
If an external load balancer or Diameter Routing Agent (DRA) is used, the CPS in the active cluster is typically configured in one group and the CPS in the standby cluster is configured in a secondary group. The load balancer/DRA may then be configured to automatically fail over from active to passive cluster.

Figure 1: Active/Standby Without Session Replication



Active/Active

Figure 2: Active/Active Without Session Replication

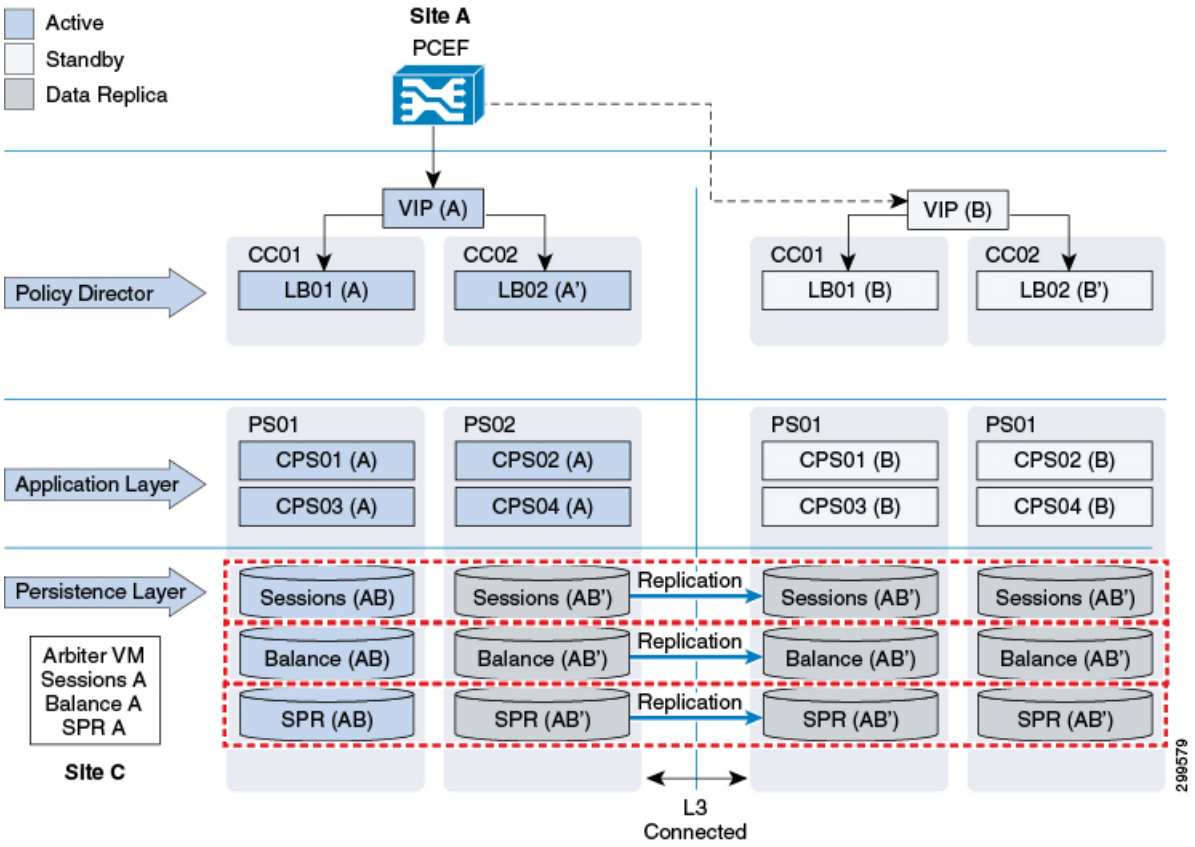


- Traffic from the network is distributed to two CPS clusters concurrently.
- PCEFs are divided within the Service Provider's network to have a 50/50% split based on traffic.
- Session data is not replicated across sites.
- SPR (subscriber information) data is replicated across Standby site.
- Balance data is replicated across Standby site.
- Diameter sessions need to be re-established if a failover occurs. Outstanding balance reservations will time out and be released.
- In case of a failure all traffic is routed to the remaining CPS site.

With Session Replication

Active/Standby

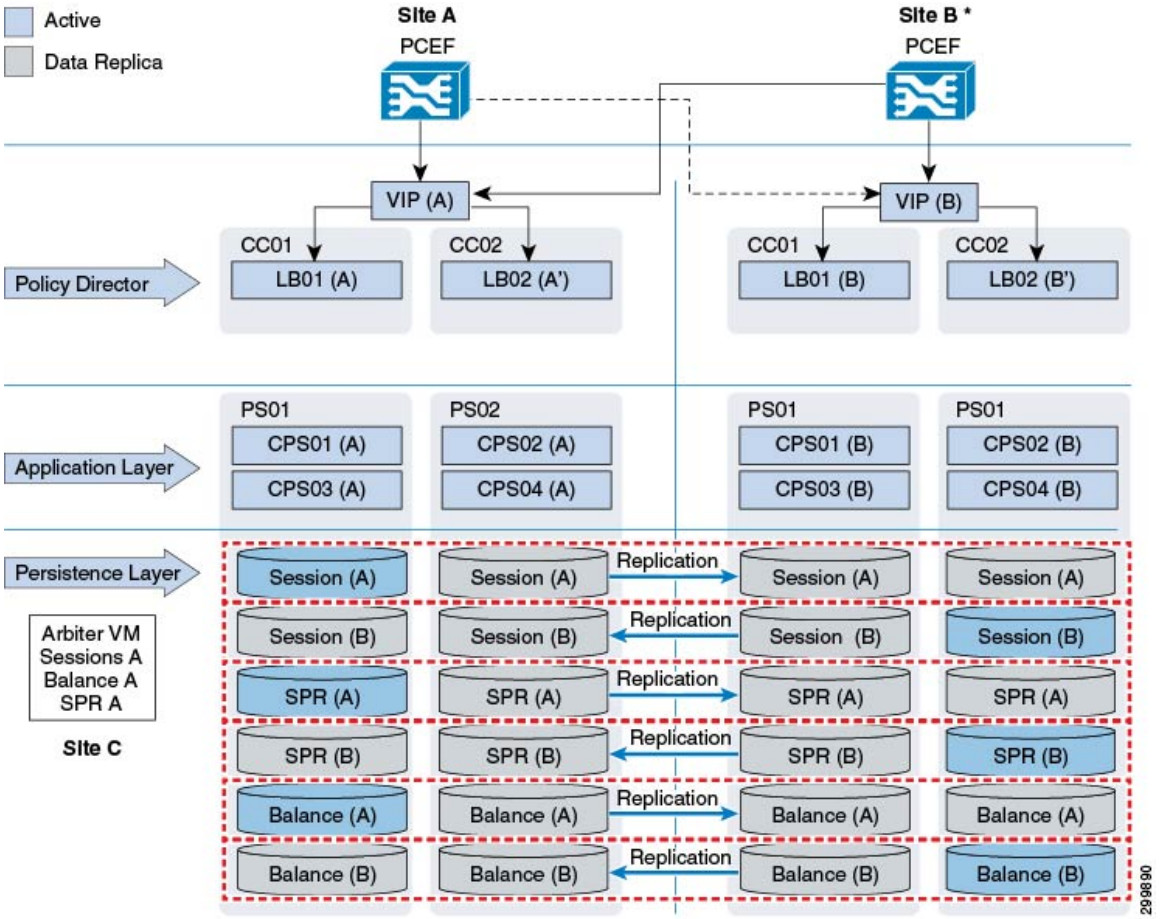
Figure 3: Active/Standby With Session Replication



- Solution protects against complete site outage as well as link failure towards one or more PCEF sites.
- If PCEF fails over to Secondary site while Primary site is still active (for example, link failure):
  - SPR data is retrieved from local SPR replica members at Secondary site.
  - Session and Balance data is read/written across from/to Primary site.
- Complete Outage of Policy Director Layer results in database failover to Secondary site.
- On recovery from a failure, a CPS node does not accept traffic until databases are known to be in a good state.

Active/Active

Figure 4: Active/Active With Session Replication



* - Site A PCEF will connect to Site A as Primary and Site B as Secondary and similary Site B PCEF will connect to Site B as Primary and Site A as Secondary

- Traffic from the network is distributed to two clusters concurrently.
- PCEFs are divided within the Service Provider's network to have a 50/50% split based on traffic.
- Session data is replicated across sites (two way replication).
- SPR (subscriber information) data is replicated across Standby site.
- Balance data is replicated across Standby site.
- Diameter session does not need to be re-established if a failover occurs. No loss of profile or balance information.
- Load balancer VMs use only local VMs for traffic processing.
- In case of a failure all traffic is routed to the remaining site.

Advantages and Disadvantages of GR Models

The following table provides a comparison based on advantages and disadvantages for different GR models described in GR Reference Models, on page 1.

Table 1: Advantages and Disadvantages of GR Models

GR Model	Session	Other Databases	Advantages	Disadvantages
Active/Standby	Replicated	SPR and Balance replicated	Protection against complete site outage as well as link failure towards one or more PCEFs. Session Continuation, diameter sessions do not need to be re-established, hence VoLTE friendly.	Session replication demands bandwidth. In case there is network latency or high TPS, the hardware requirement increases as we are required to split the incoming traffic across multiple virtual machines to achieve high speed replication and recovery.
Active/Standby	Not replicated	SPR and Balance replicated	Protection against complete site outage as well as link failure towards one or more PCEFs.	Sessions do not continue after failover, hence, they need to be re-established, NOT VoLTE friendly.
Active/Active	Replicated	SPR and Balance replicated, they are separate to each site	Protection against complete site outage as well as link failure towards one or more PCEFs. Session Continuation, diameter sessions do not need to be re-established, hence VoLTE friendly.	Session replication demands bandwidth. The hardware requirement increases significantly as we need additional load balancers and session cache virtual machines.

GR Model	Session	Other Databases	Advantages	Disadvantages
Active/Active	Not replicated	SPR and Balance replicated, they are separate to each site	Protection against complete site outage as well as link failure towards one or more PCEFs. Low bandwidth and significantly low hardware requirements.	Sessions do not continue after failover, hence, they need to be re-established, NOT VoLTE friendly.

SPR/Balance Considerations

SPR Considerations

The following list provides the information related to SPR considerations:

- SPR data is read from secondary replica members:
  - MongoDB tag sets can be used to target read operations to local replica members, that way avoiding cross-site SPR reads.
- SPR data is always written to primary database:
  - Profile updates are broadcasted to other sites to trigger policy updates if/as required for sessions established through remote sites.
- SPR updates that happen while primary site is isolated are only enforced after session update (once primary site is available again).

Balance Considerations

The following list provides the information related to balance considerations:

- Balance data is always read from primary database unless primary database is not available (for example, in the event of site isolation).
- Balance data is always written to primary database.
- Balance database design options:
  - Single database across two GR sites: cross-site balance read/writes from site without primary database.
- CDR Balance Records Reconciliation
  - During site isolation, debits are written to backup CDR balance database for reconciliation when connectivity is restored.

- No thresholds or caps are enforced during site isolation.
- Policies associated with any threshold breaches during isolation are enforced at the time of balance reconciliation.
- Potential for balance leakage if balance consumed during isolation is greater than user‘s remaining allowance.

Data Synchronization

Geo-redundancy is achieved by synchronizing data across the site(s) in the cluster. Three types of data are replicated across sites:

- Service and policy rule configuration
- Subscriber data is stored in the SPR component
- Balance data stored in the MsBM component

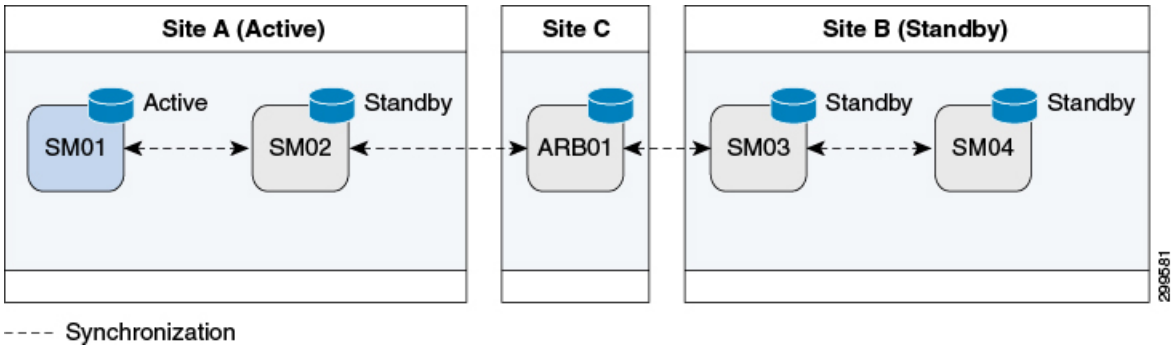
In addition, active session data stored in the Session Manager component may also be synchronized across sites when network conditions permit. Active session data is the most volatile data in CPS and has the most stringent synchronization requirements.

CPS utilizes a unicast heartbeat between sites in the geographic redundant solution. The heartbeat allows the session manager components to know which is the currently active component and protects against a split-brain scenario where data is accepted at more than one session manager component (possibly causing data corruption).

An additional external component called an “arbiter” provides a tie-breaking vote as to which of the session managers is the current master. This external component is required to reside on a separate site from the primary and secondary sites and must be routable from both sites. This is used to ensure that if one of the sites is lost, the arbiter has the ability to promote the standby sites session manager to be the master.

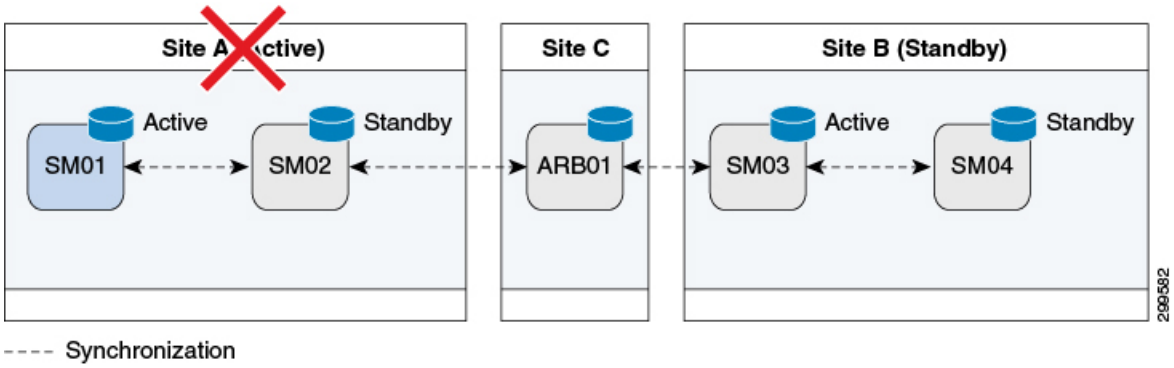
The following example shows a detailed architecture of the data synchronization for subscriber, balance and session data:

Figure 5: Data Synchronization for Subscriber, Balance and Session Data



In the case of Site A failure, Site B's session manager will become master as shown in the following example:

Figure 6: In Case of Site A Failure



Data Synchronization in MongoDB

In short, replication is achieved through a replica set where there are multiple members of a set. These members have only one primary member and others are secondary members. Write operations can occur only in primary, and read operations can happen from Primary and Secondary members. All the data written to Primary is stored in form of operation logs, that is, oplogs on primary database and secondaries fetch that to synchronize and remain up to date. In CPS, `/etc/broadhop/mongoConfig.cfg` file defines replica members, replica sets and therefore, defines databases that will be replicated and not replicated.

For more information on data synchronization in MongoDB, refer to http://docs.mongodb.org/manual/core/replica-set-sync/

CPS GR Dimensions

The GR dimensions such as databases, number of sites, arbiter considerations, and shards are dependent on each other. Only the deployment style is not dependent on other dimensions. Rest of the dimensions are inter-related and,

- Arbiter typically decides models for shard.
- Number of sites impacts the decision to have database in common.
- Common database style impacts decision to have shard.

Different Databases

CPS has three databases that have subscriber critical data such as subscriber database (SPR), balance, and sessions. Different deployment models exist depending upon how the user wants to have the databases configured. Some users might want a database common across different sites (typically this can happen for SPR), or individual instances at each site (most of the times this would be with sessions database and balance database). Typically, the databases that are updated more frequently (such as sessions and balance) would be maintained locally and replicated across sites whereas databases that are updated rarely can be kept common across sites (with some limitations).

Number of Sites

Typical deployment is expected to be two sites. However, there might be cases where multiple combinations might come up with respect to database redundancy, common database across multiple sites, general redundancy across multiple sites and so on. Since this is a highly variable factor, for each deployment model here, we need to understand various network requirements.

Arbiter Considerations

Typically the Arbiter needs to be located at a third independent site. However, depending upon customer needs and limitations, different deployment models come up where arbiter can be placed at one of the sites, creating limitations in the model.

The location of the Arbiter is an important factor in the design. Having the Arbiter located on the same site as that of Primary or Secondary poses various issues. The following table describes the issues:

Table 2: Issues Posed by Location of the Arbiter

Distribution of Arbiter	Impact on System
Arbiter at Active site	When Active site goes down, database on Secondary is supposed to become Primary. However, since it does not have required votes as Arbiter is also down at Primary site, role change cannot take place and we face downtime.
Arbiter on Secondary site	In this case, if Secondary site goes down, we do not have arbiter available. Due to this, database on Primary site does not have majority of votes, and database steps down. That way, we face downtime on system unless there is manual intervention. Additionally, if there is a split brain situation, since arbiter is on secondary site, database role changeover starts from Primary to Secondary, which is unnecessary.
Arbiter on third site	This is the best and recommended way of placing an arbiter. In any case, either Primary failure or Secondary failure, a proper failover happens as there are always majority of votes available to select a Primary.

It is important to understand the placement of arbiter and its implications. In Geographic Redundancy, failover is expected when a site goes down completely. There are many possibilities for a site to go down and based on these possibilities, we can decide the location of arbiter.

Database Partitioning - Shards

When the database size grows large, it is good to have it partitioned, in terms of MongoDB. The partitioning is done by creating shards for the database. MongoDB has some limitations for creating shards and depending upon deployment model, shard considerations come in picture. When shards come in picture, we need to also

consider the configuration servers for those shards. The configuration server decides which partition/shard contains what data. It has the keys based on which data distribution and lookup works.

Placement of these configuration servers also plays an important role in performance of databases. During site failures, if we have less number of configuration servers available, the performance of database is degraded. Hence, it is important to place the configuration servers in such a way that maximum of them are available always. Typically, the configuration servers are placed in line with database that is, one at primary, another at secondary and third at the arbiter. MongoDB supports a maximum of three configuration servers.

Session Shard Considerations

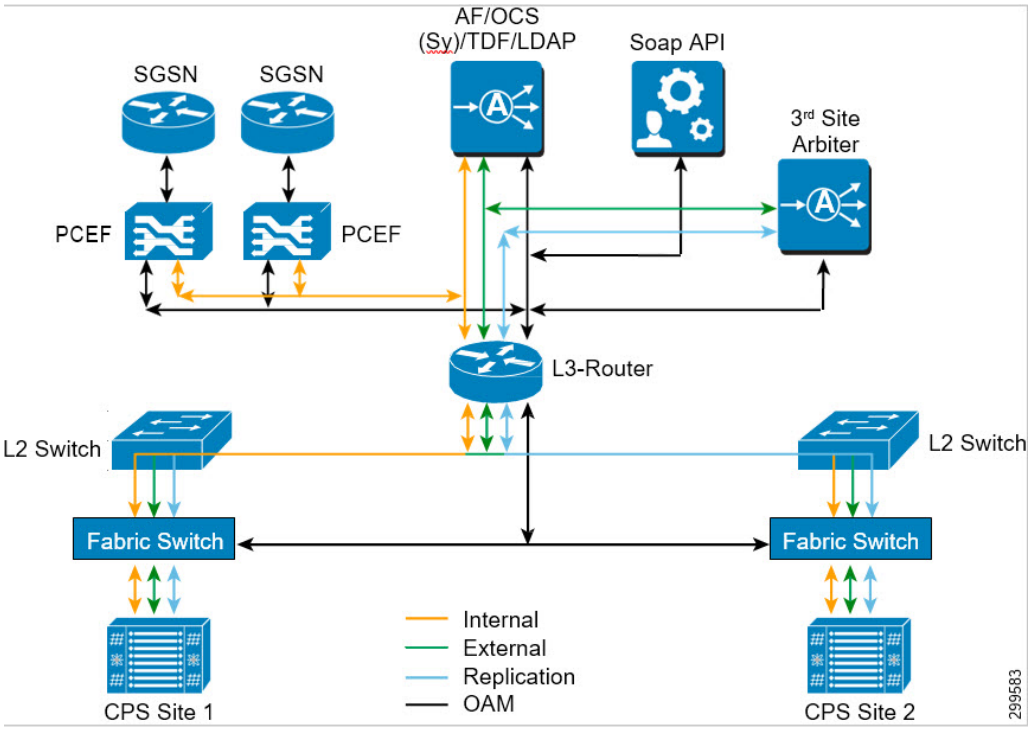
For sessions, we define internal shards. Currently, we create four internal shards per session database so that we see four internal databases. This helps to achieve parallel writes to same database thereby increasing write/read efficiency and achieve higher performance. Typically, for higher TPS, we might be required to create multiple shards across different virtual machines. In that case, an additional session replica set is created that contains four more shards. The admin database contains information for all such shards so that the Policy Server (QNS) processing engines route session calls to appropriate shards based on internal hashing algorithm. The actual number of shards required can be obtained from the dimensioning guide.

Network Diagrams

High Level Diagram including other Network Nodes

The following is an example of a high-level diagram showing various network modules connected to CPS for GR setup. This diagram can be different for different deployment scenarios. Contact your Cisco Technical Representative for high level diagram specific to your deployment scenario.

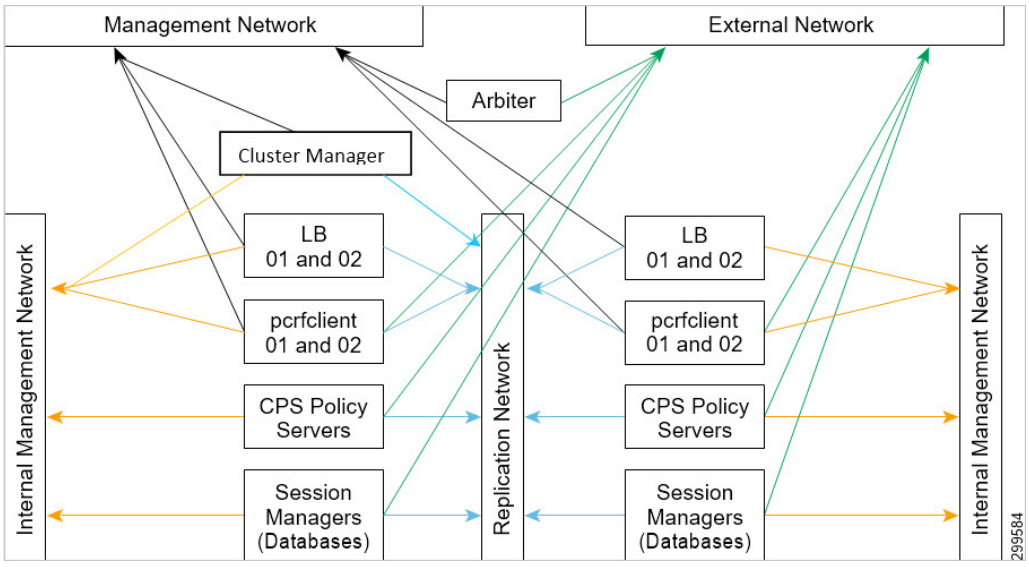
Figure 7: High Level Diagram including other Network Nodes



CPS Level Network Diagram

The following network diagram explains various requirements for GR setup:

Figure 8: CPS Level Network Diagram



The following sections describe the interface requirements for GR setup:

Management Network Interface

This interface is used for traffic to CPS, unified API, portal (not shown here), and for login to CPS machines through pcrfclient, and to access Policy Builder and Control Center web interfaces.

The following VMs need this network interface:

- Load balancers
- pcrfclient01
- Arbiter
- Cluster Manager

External and Routable Network Interface

This interface is used for communication between with any entity that is external to CPS HA System. Since the MongoDB configuration servers reside on pcrfclient01 of both sites, a separate network is needed for both to communicate with each other over an interface other than replication network. If the replication network fails, communication would still be needed between the arbiter and session managers, and between arbiter and pcrfclient01 so that the arbiter is able to determine the appropriate primary for databases, and make more than one configuration servers available. If this is not done, and if the arbiter is configured to communicate with databases over the replication network, if the replication network fails, a split brain situation occurs since the arbiter would be disconnected from both sites.

When there are no shards configured for databases, no configuration servers are needed. The pcrfclient01 at both sites still needs external network connectivity with arbiter as scripts on pcrfclient need to communicate with the arbiter (such as `get_replica_status.sh`).

In GR, we need to connect the Policy Server (QNS) to arbiter. During failover, Policy Server (QNS) gets all the available members' list and tries to see if they are reachable. In case arbiter is not reachable from Policy Server (QNS), it hangs there.

The following VMs need this network interface:

- pcrfclient01
- Arbiter
- Session managers
- Policy Server (QNS)

Replication Network Interface

Typically referred as Signaling Network, this network carries the data replication in a Geo-HA environment across two sites. Also, Policy Servers (QNS) on one site communicate with databases on another site using the same interface. The same network should be used to exchange messages between two sites.

The following VMs need this network interface:

- Policy Director (lbs)
- pcrfclient

- Policy Server (QNS)
- Session managers (databases)
- Cluster Manager

Note In Geo-HA environment, if you want to execute the platform scripts (such as, `diagnostics.sh`) from Cluster Manager, it must be connected to other site's Session Manager VM. Hence, Cluster Manager must be a part of external or replication network.

Internal Network

This network is used for internal communication of virtual machines of the same site.

All the CPS VMs need this network interface.

Summary

The following table provides a summary of the different VM network requirements:

Table 3: Summary

VM Name	Management IP	Signaling IP/Replication	Internal IP	External Non-Management IP
Cluster Manager	Yes	Yes	Yes	No
pcrfclient01/02	Yes	Yes	Yes	Yes
lb01/lb02	Yes	Yes	Yes	No
qns01-n	No	Yes	Yes	Yes
sessionmgrs	No	Yes	Yes	Yes
arbiter	Yes	No	No	Yes

Network Requirements

Bandwidth and latency are to be obtained from Cisco Technical Representative depending upon your deployment model.