Introduction of Oracle RAC

Oracle Real Application Clusters

By

Rajesh Kumar

DevOps@RajeshKumar.xyz

www.RajeshKumar.xyz

Oracle RAC is the technology that allows an Oracle database to provide increased levels of **high availability** and **big performance** benefits.

What is Oracle RAC?

Oracle Real Application Clusters (Oracle RAC) is considered to be one of the most advanced and capable technologies for enabling a highly available and scalable relational database. It is considered the default go-to standard for creating highly available and scalable Oracle databases.

What is Oracle RAC?

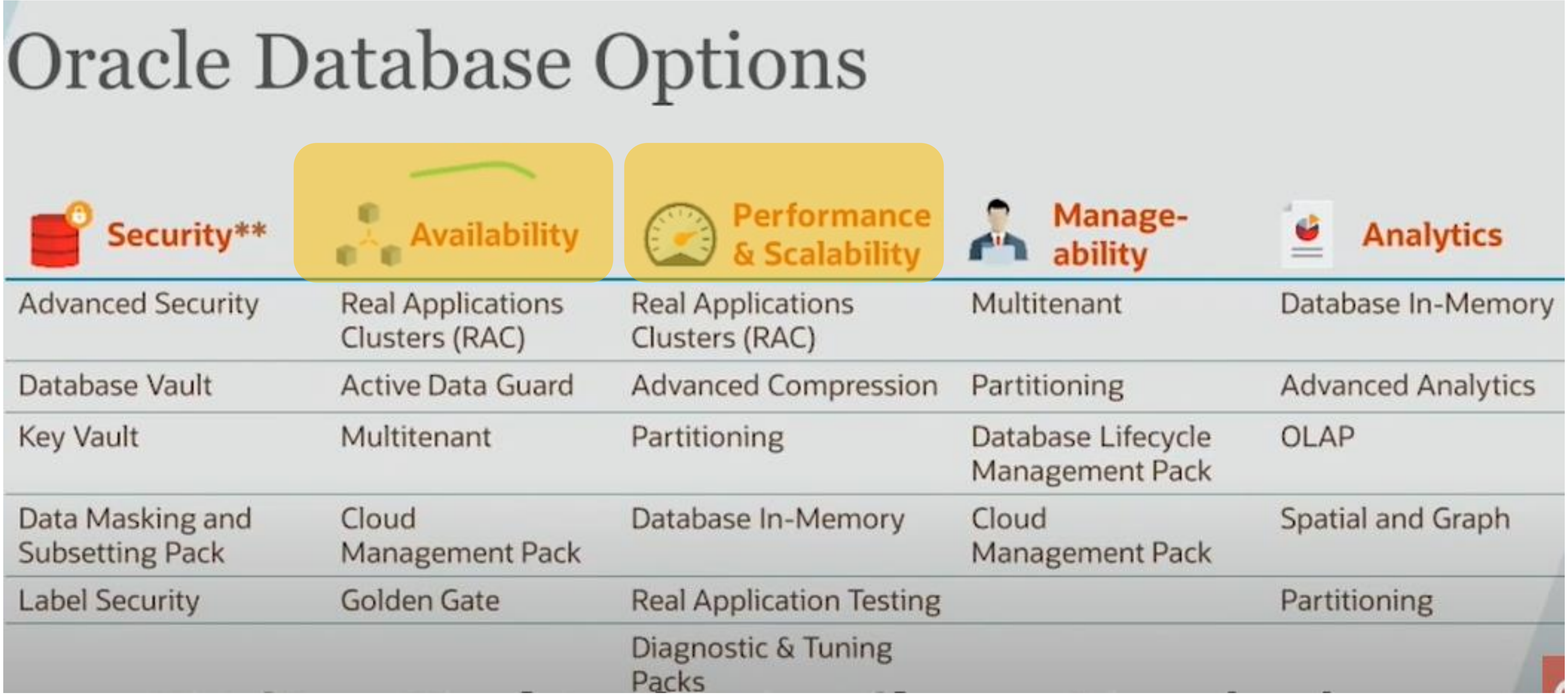
Oracle RAC uses an intricate end-to-end software stack developed by Oracle—including

- Oracle Clusterware and
- Grid Infrastructure,
- Oracle Automatic Storage Management (ASM),
- Oracle Net Listener, and the
- Oracle Database itself

Combined with enterprise-grade storage that enables a shared-everything database cluster technology.

What is Oracle RAC?

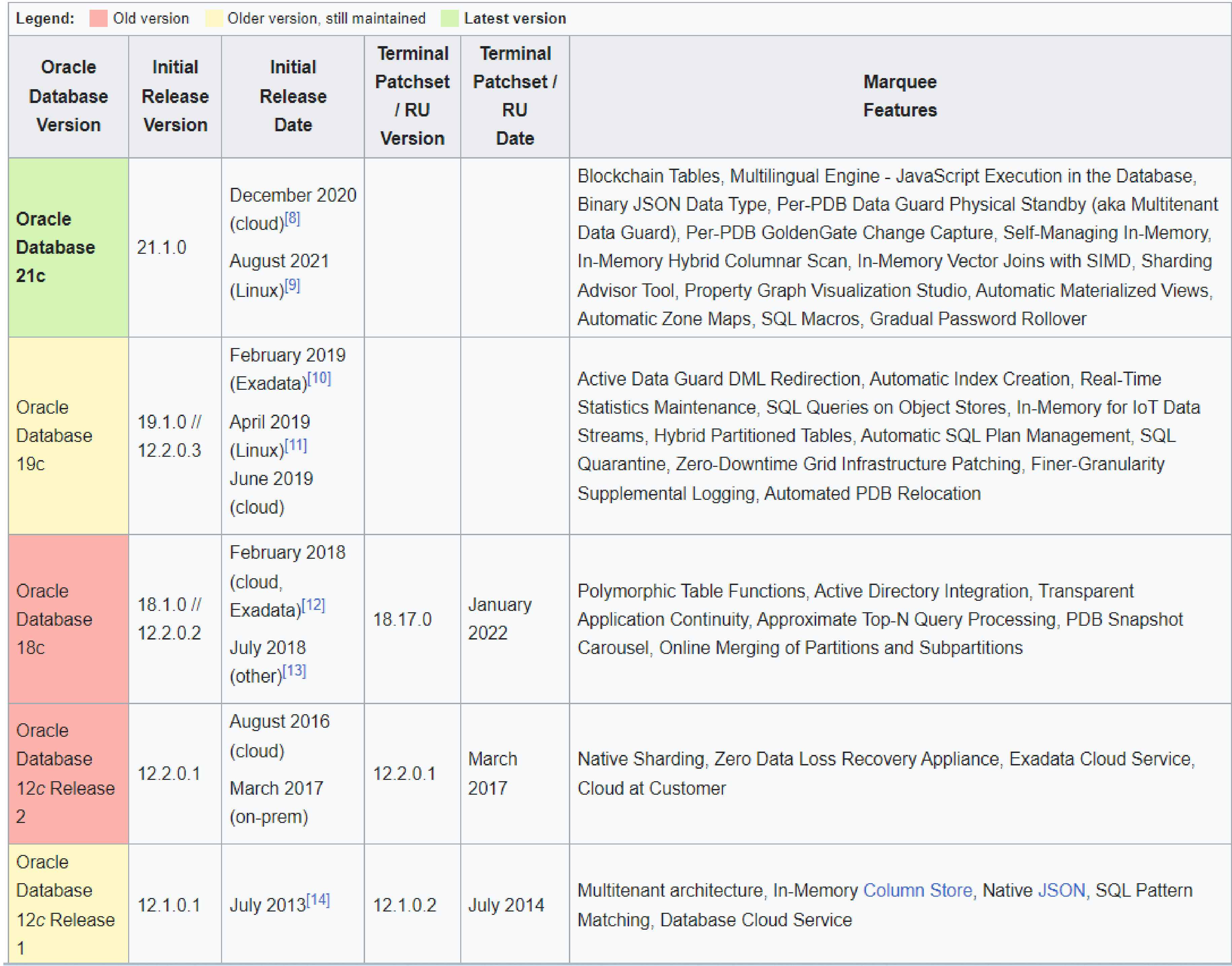
Oracle RAC is an **Active/Active** database cluster, where multiple Oracle database servers (running Oracle Database instances, which is a collection of in-memory processes and caches) access a shared storage device that contains a single set of disk-persistent database files.



Oracle Database Options

Security**	Availability	Performance & Scalability	Manageability	Analytics
Advanced Security	Real Applications Clusters (RAC)	Real Applications Clusters (RAC)	Multitenant	Database In-Memory
Database Vault	Active Data Guard	Advanced Compression	Partitioning	Advanced Analytics
Key Vault	Multitenant	Partitioning	Database Lifecycle Management Pack	OLAP
Data Masking and Subsetting Pack	Cloud Management Pack	Database In-Memory	Cloud Management Pack	Spatial and Graph
Label Security	Golden Gate	Real Application Testing		Partitioning
		Diagnostic & Tuning Packs		

Version



Legend: 🟥 Old version 🟨 Older version, still maintained 🟩 Latest version

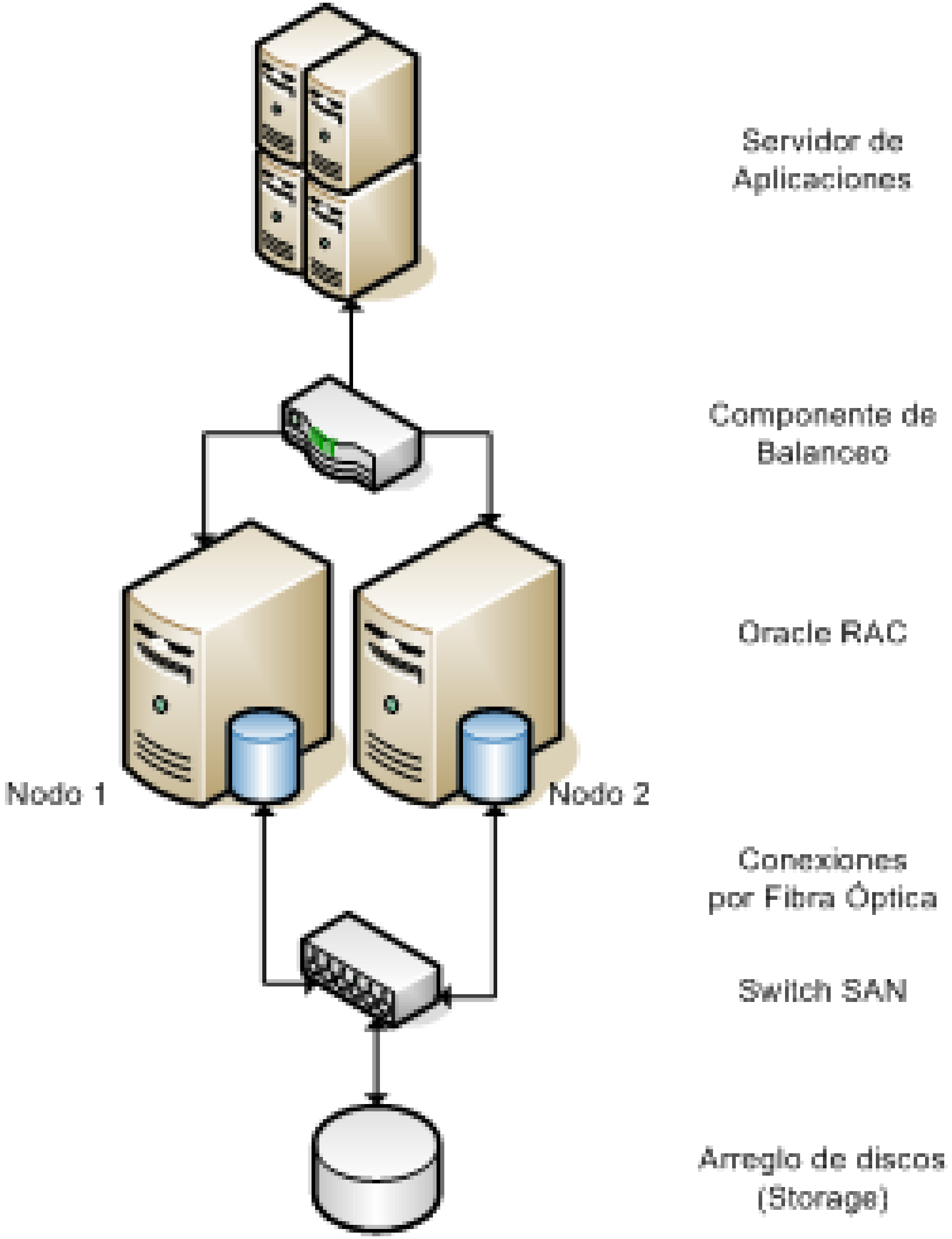
Oracle Database Version	Initial Release Version	Initial Release Date	Terminal Patchset / RU Version	Terminal Patchset / RU Date	Marquee Features
Oracle Database 21c	21.1.0	December 2020 (cloud)[8] August 2021 (Linux)[9]			Blockchain Tables, Multilingual Engine - JavaScript Execution in the Database, Binary JSON Data Type, Per-PDB Data Guard Physical Standby (aka Multitenant Data Guard), Per-PDB GoldenGate Change Capture, Self-Managing In-Memory, In-Memory Hybrid Columnar Scan, In-Memory Vector Joins with SIMD, Sharding Advisor Tool, Property Graph Visualization Studio, Automatic Materialized Views, Automatic Zone Maps, SQL Macros, Gradual Password Rollover
Oracle Database 19c	19.1.0 // 12.2.0.3	February 2019 (Exadata)[10] April 2019 (Linux)[11] June 2019 (cloud)			Active Data Guard DML Redirection, Automatic Index Creation, Real-Time Statistics Maintenance, SQL Queries on Object Stores, In-Memory for IoT Data Streams, Hybrid Partitioned Tables, Automatic SQL Plan Management, SQL Quarantine, Zero-Downtime Grid Infrastructure Patching, Finer-Granularity Supplemental Logging, Automated PDB Relocation
Oracle Database 18c	18.1.0 // 12.2.0.2	February 2018 (cloud, Exadata)[12] July 2018 (other)[13]	18.17.0	January 2022	Polymorphic Table Functions, Active Directory Integration, Transparent Application Continuity, Approximate Top-N Query Processing, PDB Snapshot Carousel, Online Merging of Partitions and Subpartitions
Oracle Database 12*c* Release 2	12.2.0.1	August 2016 (cloud) March 2017 (on-prem)	12.2.0.1	March 2017	Native Sharding, Zero Data Loss Recovery Appliance, Exadata Cloud Service, Cloud at Customer
Oracle Database 12*c* Release 1	12.1.0.1	July 2013[14]	12.1.0.2	July 2014	Multitenant architecture, In-Memory Column Store, Native JSON, SQL Pattern Matching, Database Cloud Service

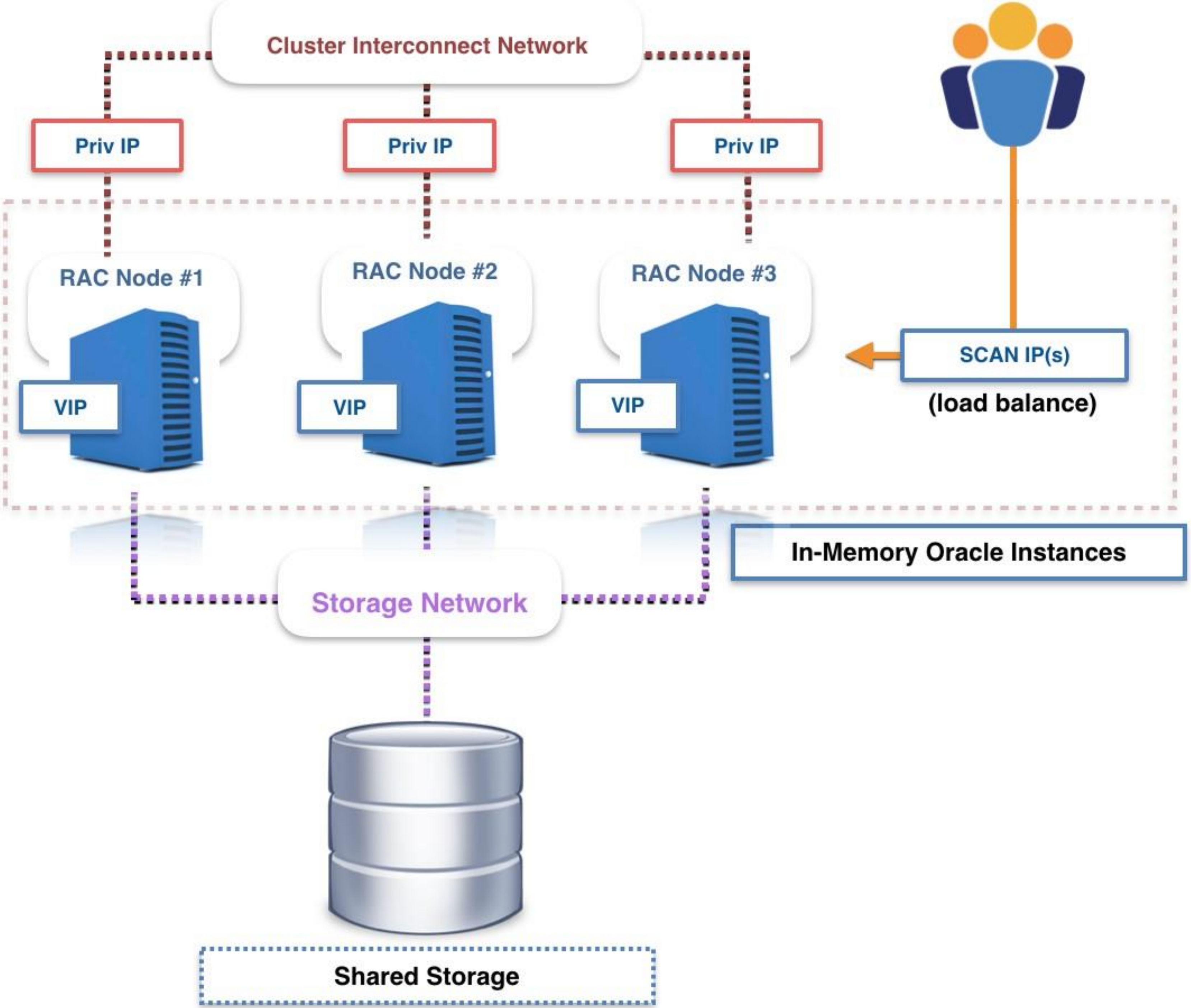
Major Benefits of Oracle RAC?

- Multiple database nodes within a single RAC cluster provide increased high availability. No single point of failure exists from the database servers themselves. However, the shared storage requires storage-based high availability or DR solutions.
- Multiple cluster database nodes allow for scaling-out query performance across multiple servers.

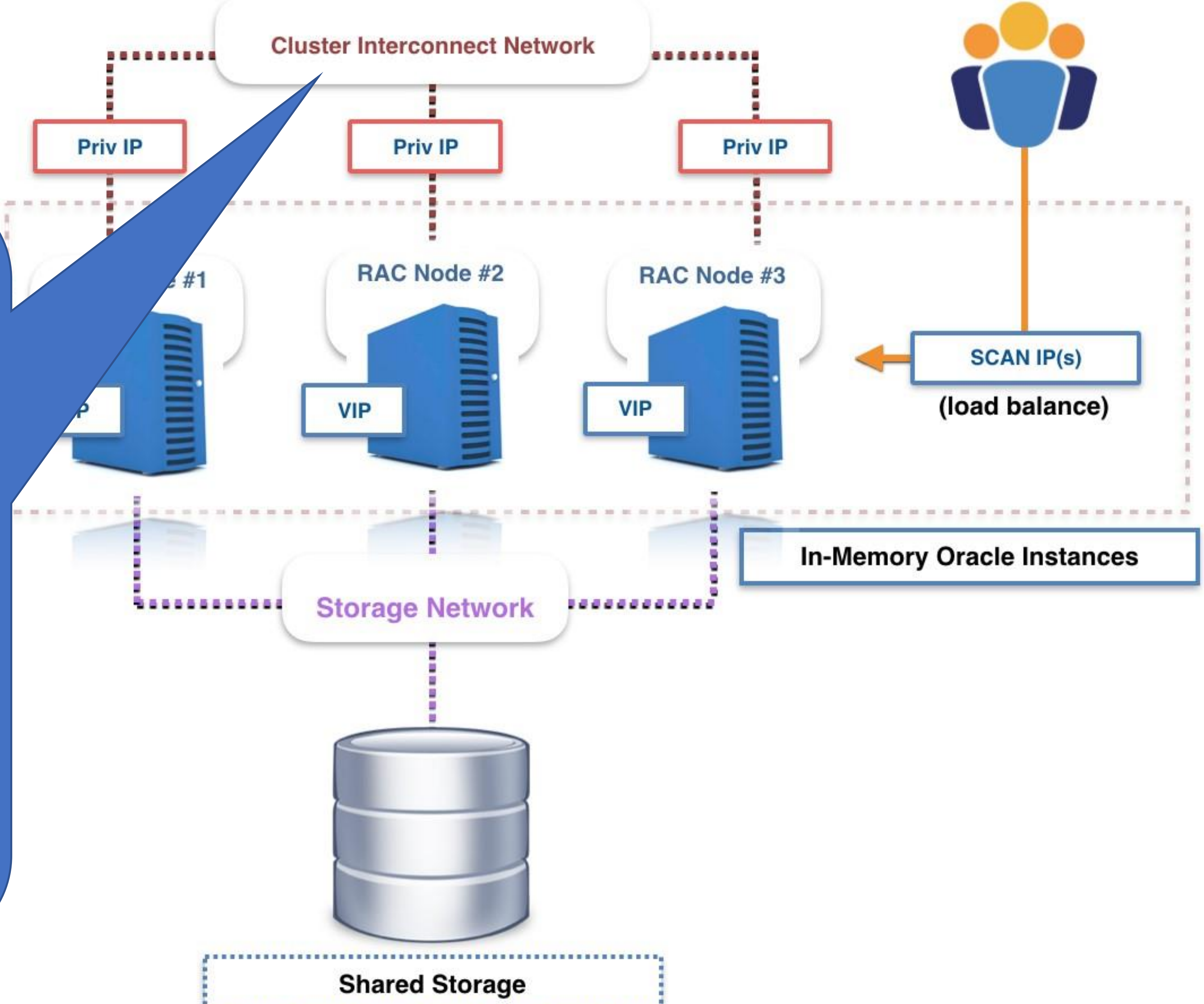
Oracle RAC Architecture

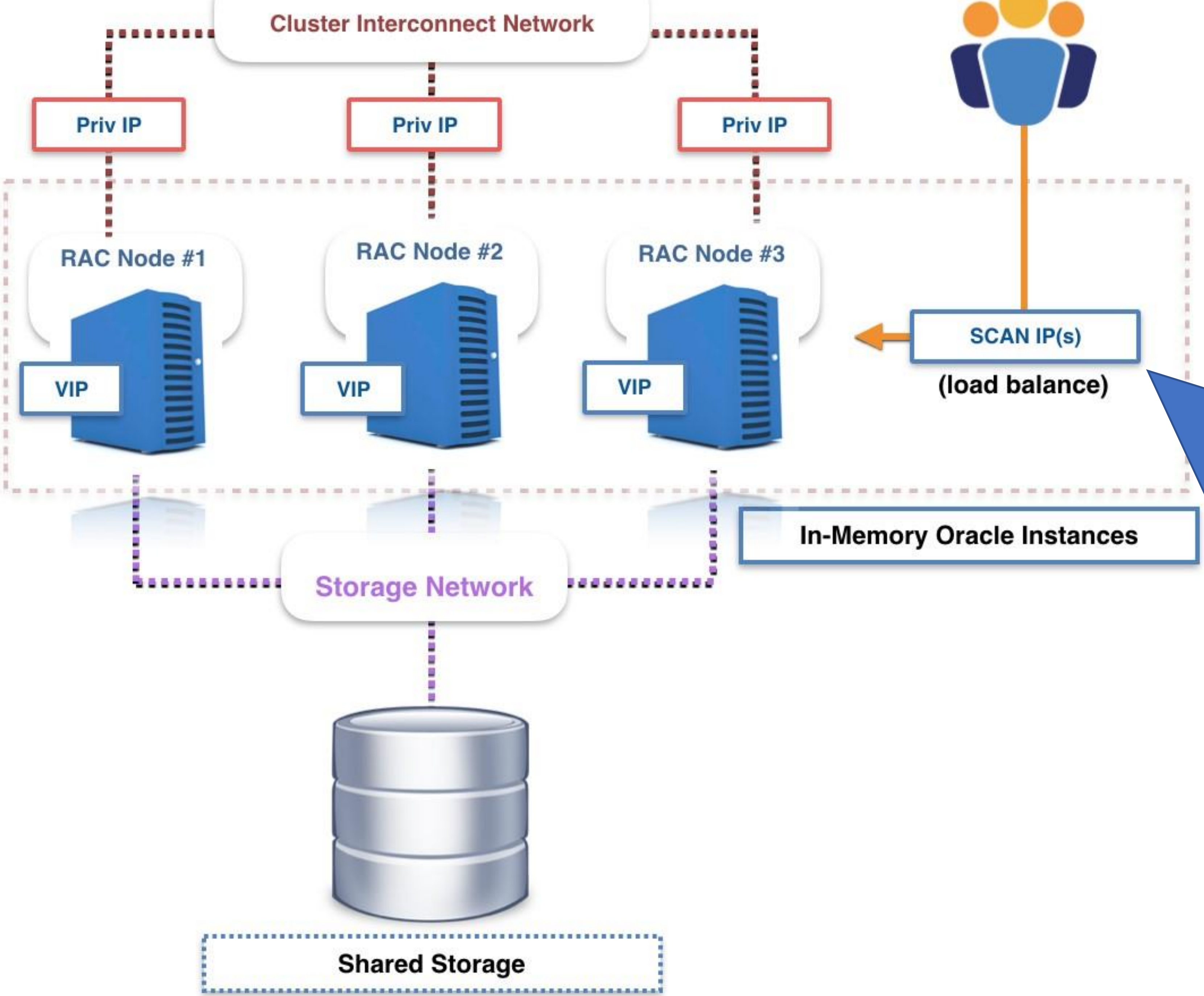
Components


Servidor de Aplicaciones
Componente de Balanceo
Oracle RAC
Nodo 1
Nodo 2
Conexiones por Fibra Óptica
Switch SAN
Arreglo de discos (Storage)

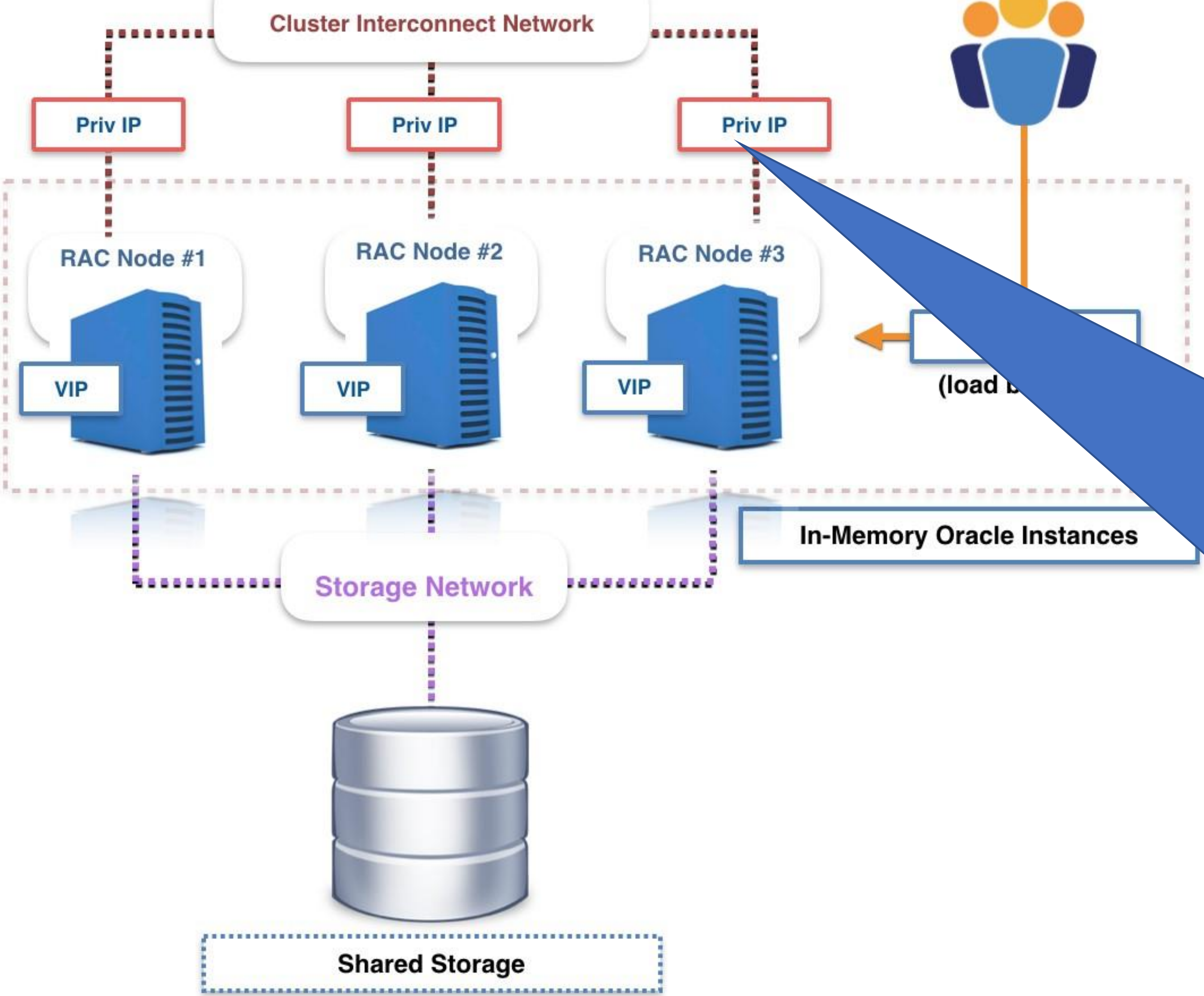

Cluster Interconnect Network
Priv IP
Priv IP
Priv IP
RAC Node #1
RAC Node #2
RAC Node #3
VIP
VIP
VIP
SCAN IP(s)
(load balance)
In-Memory Oracle Instances
Storage Network
Shared Storage

All the database nodes coordinate with one another using both a dedicated network-based communication channel (known as the **cluster interconnect**) and a set of disk-based files.





Public access to the cluster (from incoming applications, SQL queries, users, etc.) is performed using a set of SCAN IPs that are used to load-balance incoming sessions.



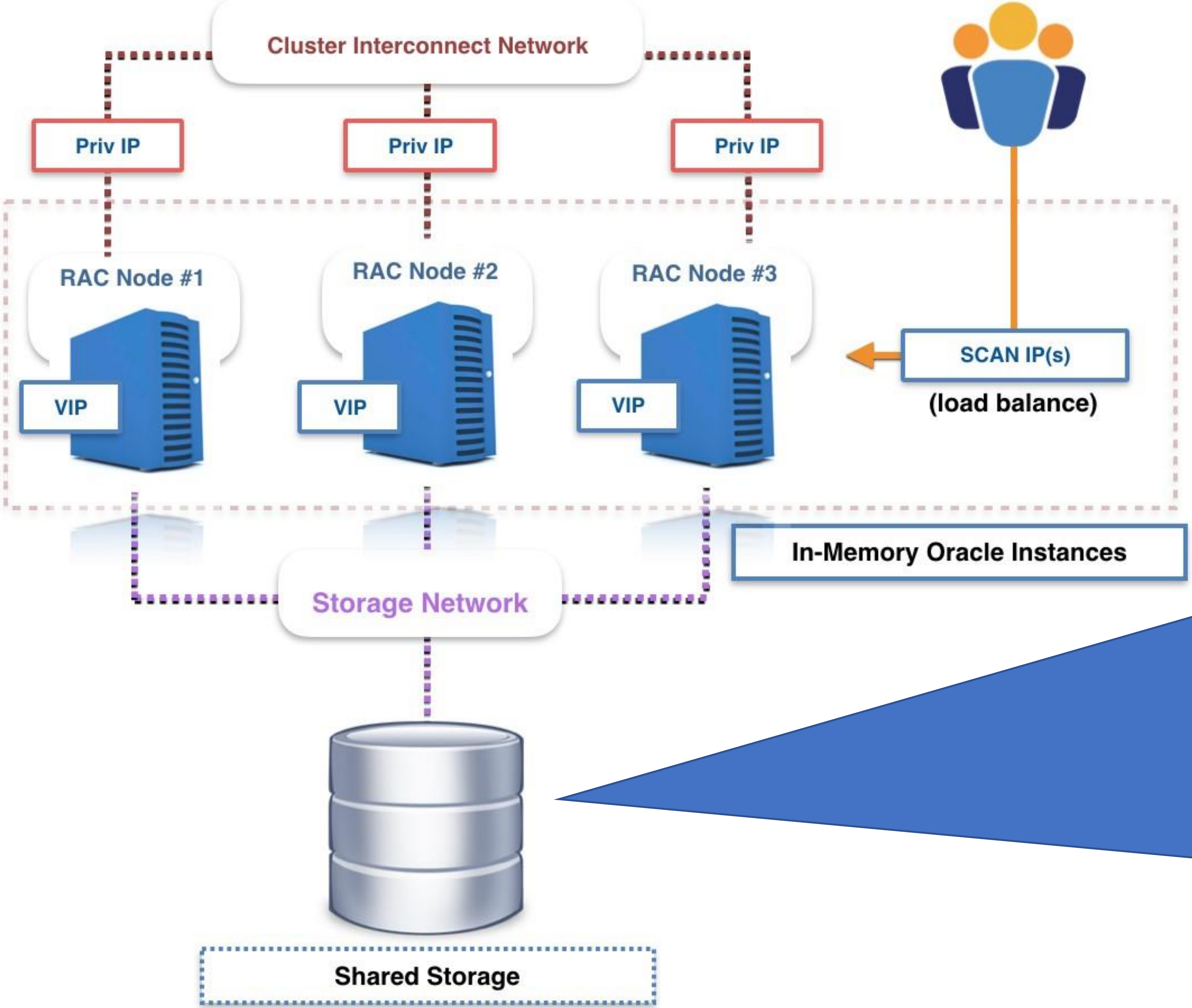
each RAC cluster node has its own physical and virtual IP addresses that can be used to open a connection directly to a specific node.

Because of the shared nature of the RAC cluster architecture—specifically, having all nodes write to a **single set of database data files on disk**—the following mechanisms were implemented to ensure that the Oracle database objects and data maintain **ACID compliance**:

- GCS
- GES

These services, which run as background processes on **each cluster node**, are essential to serialize access to shared data structures in the Oracle database.

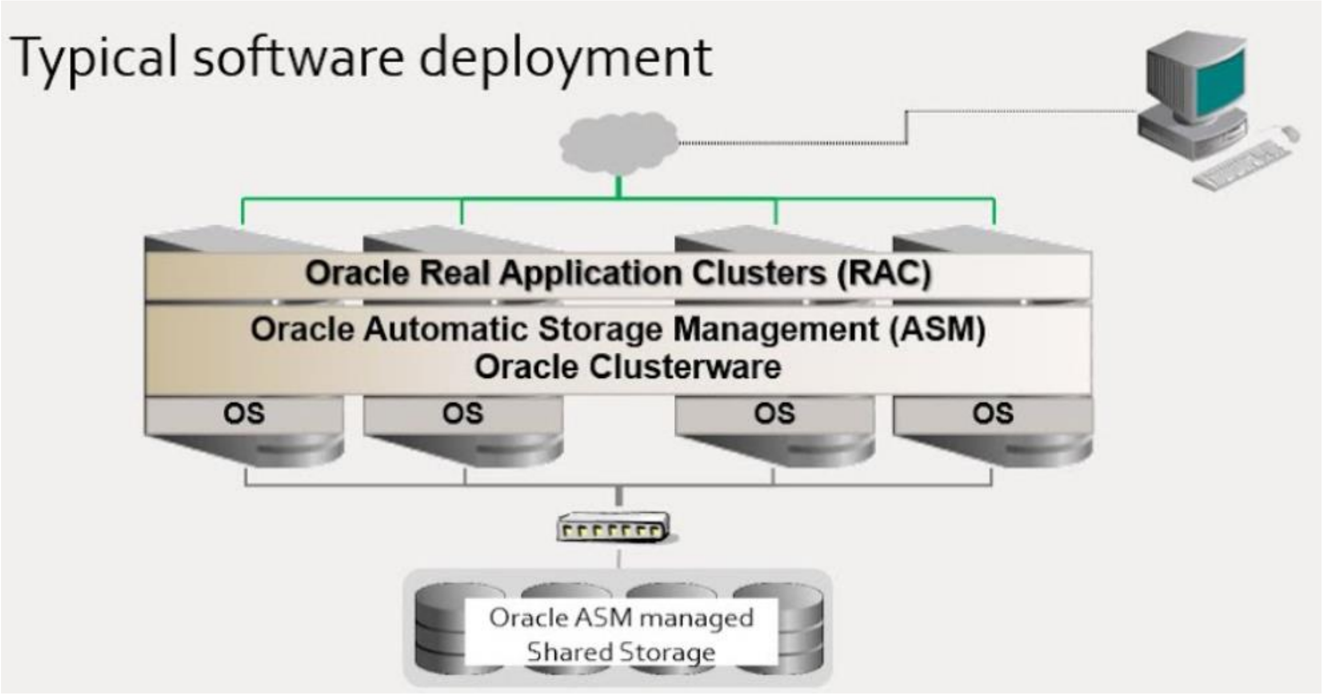
- **GCS (Global Cache Services)** tracks the location and the status of the database data blocks and helps guarantee data integrity for global access across all cluster nodes.
- **GES (Global Enqueue Services)** performs concurrency control across all cluster nodes, including cache locks and the transactions.



Shared storage is another essential component in the Oracle RAC architectures. All cluster nodes read and write data to the same physical database files stored in a disk that is accessible by all nodes. Most customers rely on high-end storage hardware to provide the shared storage capabilities required for RAC.

ASM

Oracle provides its own software-based storage/disk management mechanism called **Automatic Storage Management, or ASM**. ASM is implemented as a set of special **background processes** that run on **all cluster nodes** and allow for **easier management of the database storage layer**.


Typical software deployment
Oracle Real Application Clusters (RAC)
Oracle Automatic Storage Management (ASM)
Oracle Clusterware
OS
OS
OS
OS
Oracle ASM managed
Shared Storage

Summary

So, to recap, the main components of an Oracle RAC architecture include the following:

- **Cluster nodes:** Set of one or more servers running Oracle instances, each with a collection of in-memory processes and caches.
- **Interconnect network:** Cluster nodes communicate with one another using a dedicated “interconnect” network.
- **Shared storage:** All cluster nodes access the same physical disks and coordinate access to a single set of database data files that contain user data. Usually handled by a combination of enterprise-grade storage with Oracle’s ASM software layer.
- **SCAN (Single Client Access Name):** “Floating” virtual hostname/IPs providing load-balancing capabilities across cluster nodes. Naming resolution of SCAN to IP can be done via DNS or GNS (Grid Naming Service).
- **Virtual IPs (and Physical IPs):** Each cluster node has its own dedicated IP address.

Oracle RAC: Performance & Scale Out

With Oracle RAC, you can **add new nodes to an existing RAC cluster without downtime**. Adding more nodes to the RAC cluster increases the level of high availability that’s provided and also enhances performance.

Challenges with Scale out

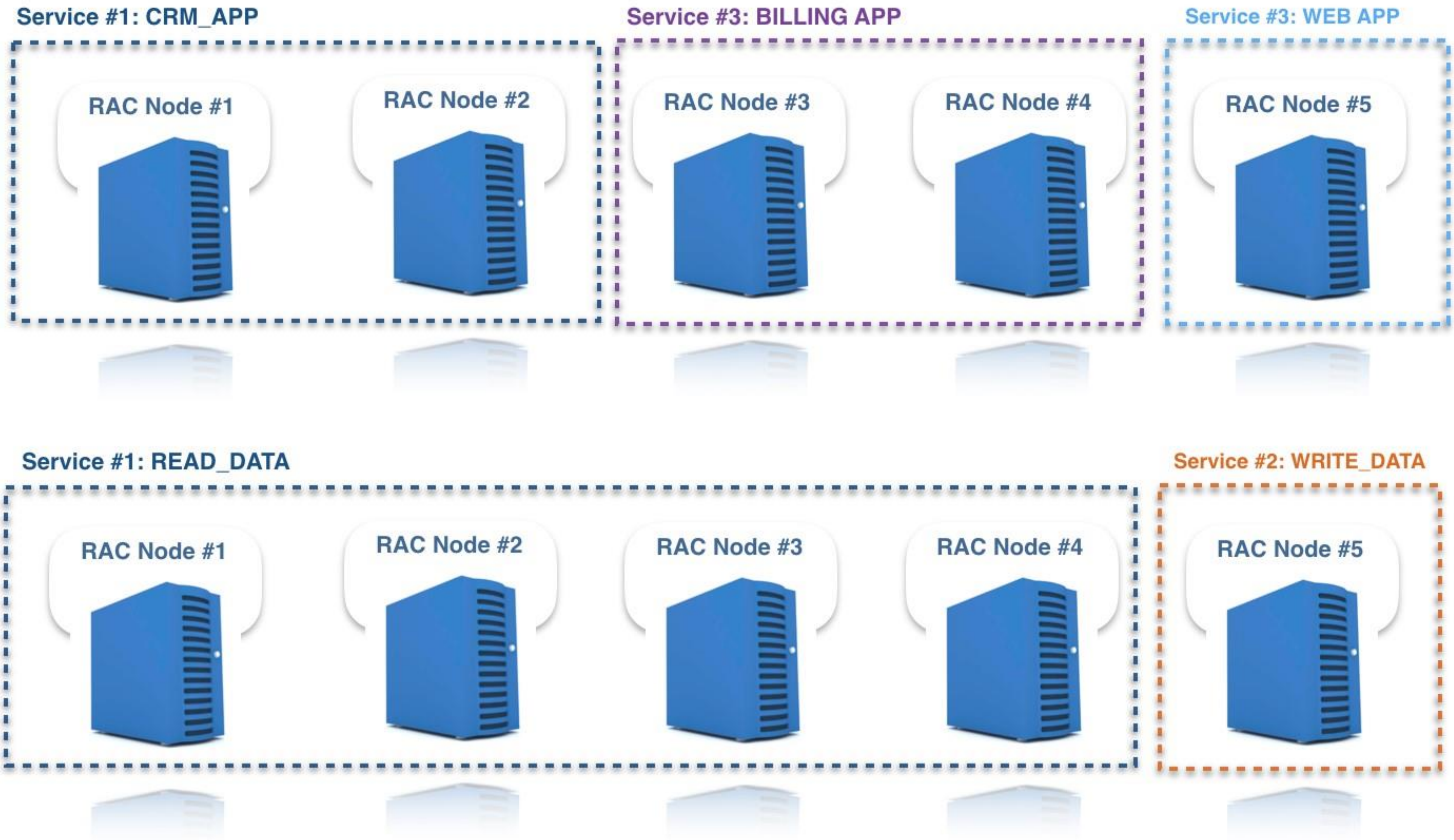
Although you can scale read performance easily by adding more cluster nodes, scaling write performance is a more complex subject. **Technically, Oracle RAC can scale writes and reads together when adding new nodes to the cluster, but attempts from multiple sessions to modify rows that reside in the same physical Oracle block (the lowest level of logical I/O performed by the database) can cause write overhead for the requested block and affect write performance.**

Challenges with Scale out

Concurrency is also one of the reasons and why RAC implements a “**smart mastering**” mechanism which attempts to reduce write-concurrency overhead. The “**smart mastering**” mechanism enables the database to determine which service causes which rows to be read into the buffer cache and master the data blocks only on those nodes where the service is active. **Scaling writes in RAC isn’t as straightforward as scaling reads.**

Challenges with Scale out

With the limitations for pure write scale-out, many Oracle RAC customers choose to split their RAC clusters into multiple “**services,”** which are logical groupings of nodes in the same RAC cluster. By using services, you can use Oracle RAC to perform direct writes to specific cluster nodes.



Service #1: CRM_APP

RAC Node #1

RAC Node #2

Service #3: BILLING APP

RAC Node #3

RAC Node #4

Service #3: WEB APP

RAC Node #5

Service #1: READ_DATA

RAC Node #1

RAC Node #2

RAC Node #3

RAC Node #4

Service #2: WRITE_DATA

RAC Node #5

1. Splitting writes from different individual “modules” in the application (that is, groups of independent tables) to different nodes in the cluster. This is also known as “application partitioning” (not to be confused with database table partitions).
2. In extremely un-optimized workloads with high concurrency, directing all writes to a single RAC node and load-balancing only the reads.

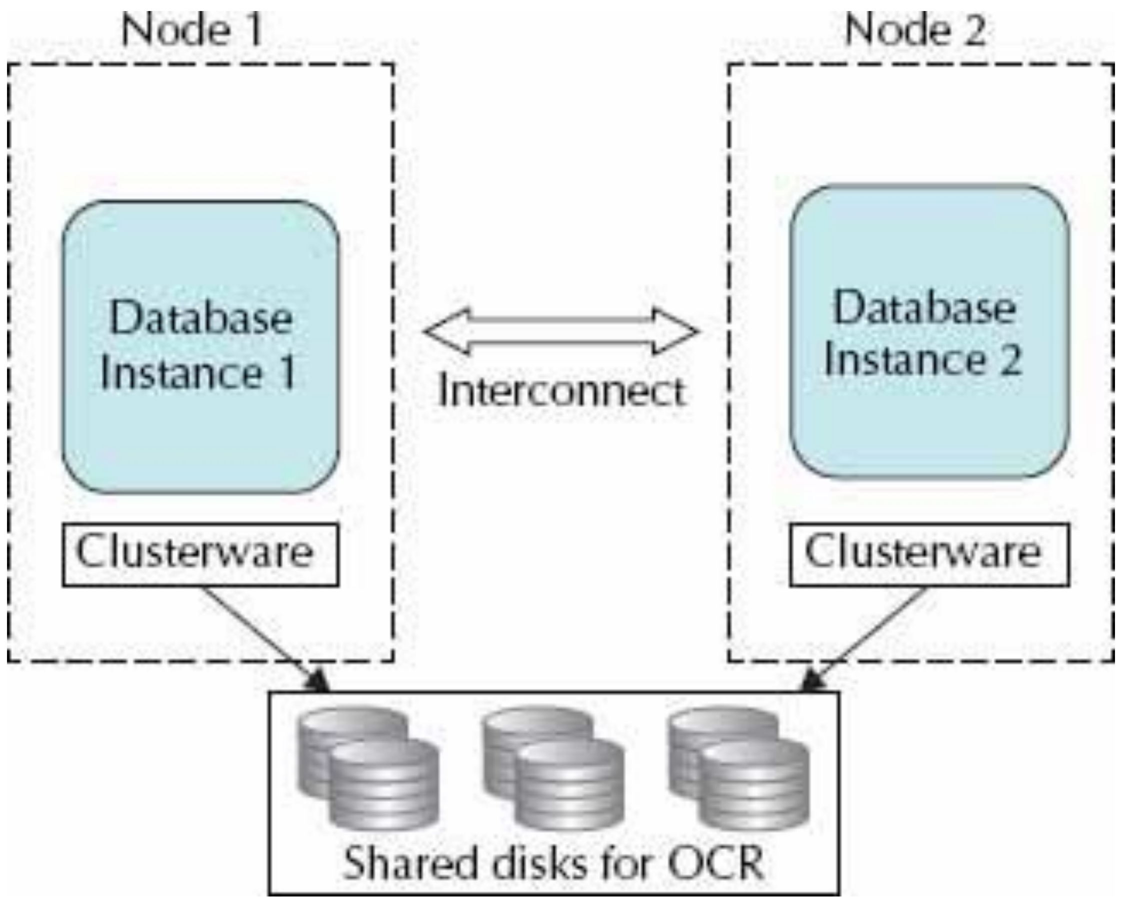
Oracle Clusterware

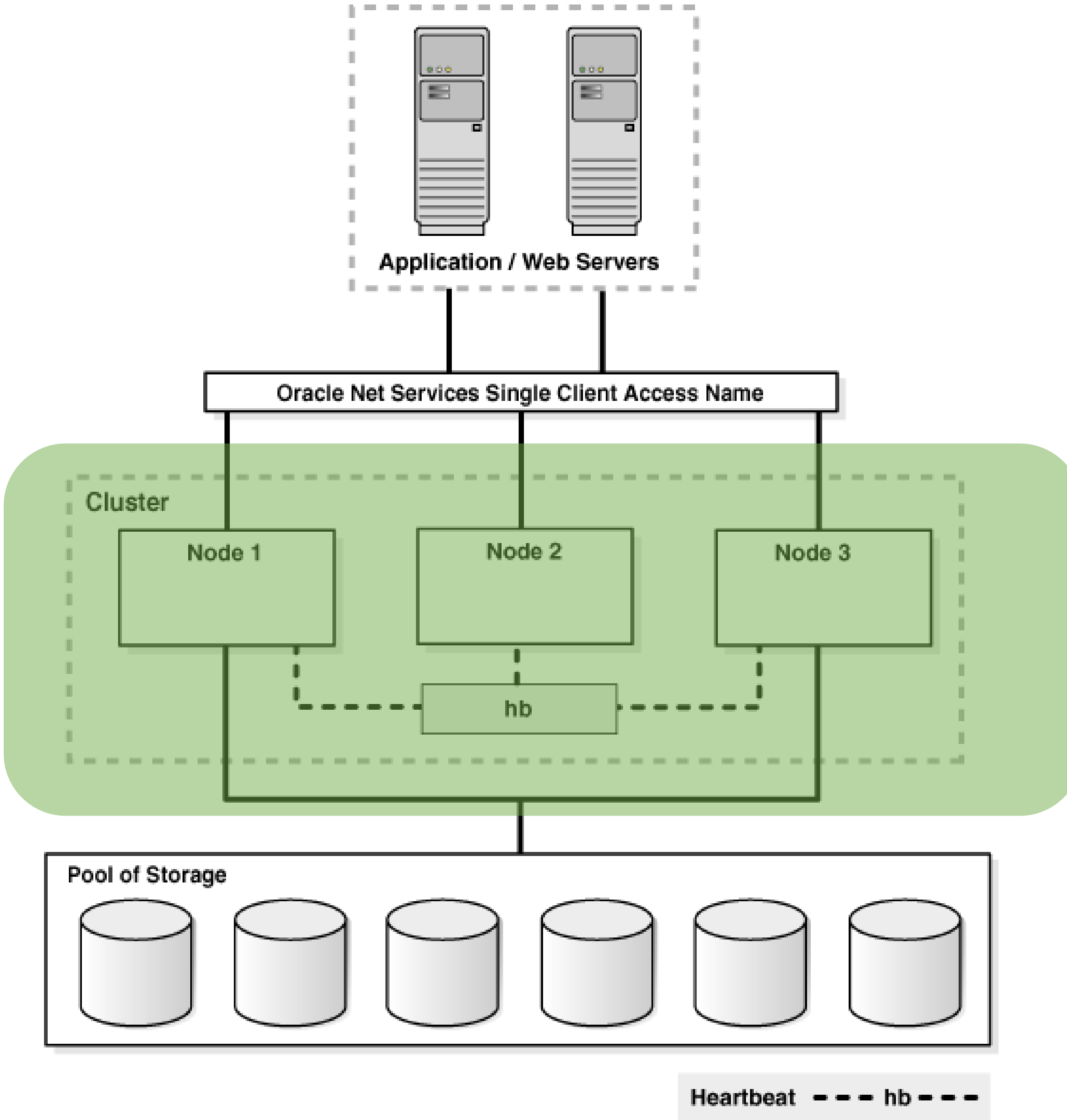
How to implement Oracle RAC?

Oracle Clusterware

Oracle **Clusterware** provides the infrastructure necessary to run Oracle Real Application Clusters (Oracle RAC). Oracle Clusterware also manages resources, such as virtual IP (VIP) addresses, databases, listeners, services, and so on.

Oracle **Clusterware enables servers to communicate with each other**, so that they appear to **function as a collective unit**. This combination of servers is commonly **known as a cluster**. Although the servers are standalone servers, each server has additional processes that communicate with other servers. In this way the separate servers appear as if they are one system to applications and end users.


Node 1
Node 2
Database Instance 1
Interconnect
Database Instance 2
Clusterware
Clusterware
Shared disks for OCR


Application / Web Servers
Oracle Net Services Single Client Access Name
Cluster
Node 1
Node 2
Node 3
hb
Pool of Storage
Heartbeat - - - hb - - -

Oracle Clusterware

Voting Disks

Oracle Clusterware uses voting disk files to determine which nodes are members of a cluster. You can configure voting disks on Oracle ASM, or you can configure voting disks on shared storage.

Oracle Cluster Registry

Oracle Clusterware uses the Oracle Cluster Registry (OCR) to store and manage information about the components that Oracle Clusterware controls, such as Oracle RAC databases, listeners, virtual IP addresses (VIPs), and services and any applications. The OCR stores configuration information in a series of key-value pairs in a tree structure. To ensure cluster high availability, Oracle recommends that you define multiple OCR locations (multiplex)

Oracle Clusterware

Oracle Clusterware Network Configuration: GNS

When you are using Oracle RAC, all of the clients must be able to reach the database. This means that the VIP addresses must be resolved by the clients. This problem is solved by the addition of the Grid Naming Service (GNS) to the cluster. GNS is linked to the corporate Domain Name Service (DNS) so that clients can easily connect to the cluster and the databases running there. Activating GNS in a cluster requires a DHCP service on the public network.

Single Client Access Name (SCAN)

Oracle RAC 11g release 2 (11.2) introduces the Single Client Access Name (SCAN). The SCAN is a single name that resolves to three IP addresses in the public network. When using GNS and DHCP, Oracle Clusterware configures the VIP addresses for the SCAN name that is provided during cluster configuration. The node VIP and the three SCAN VIPs are obtained from the DHCP server when using GNS. If a new server joins the cluster, then Oracle Clusterware dynamically obtains the required VIP address from the DHCP server, updates the cluster resource, and makes the server accessible through GNS.

Benefits of Clusterware

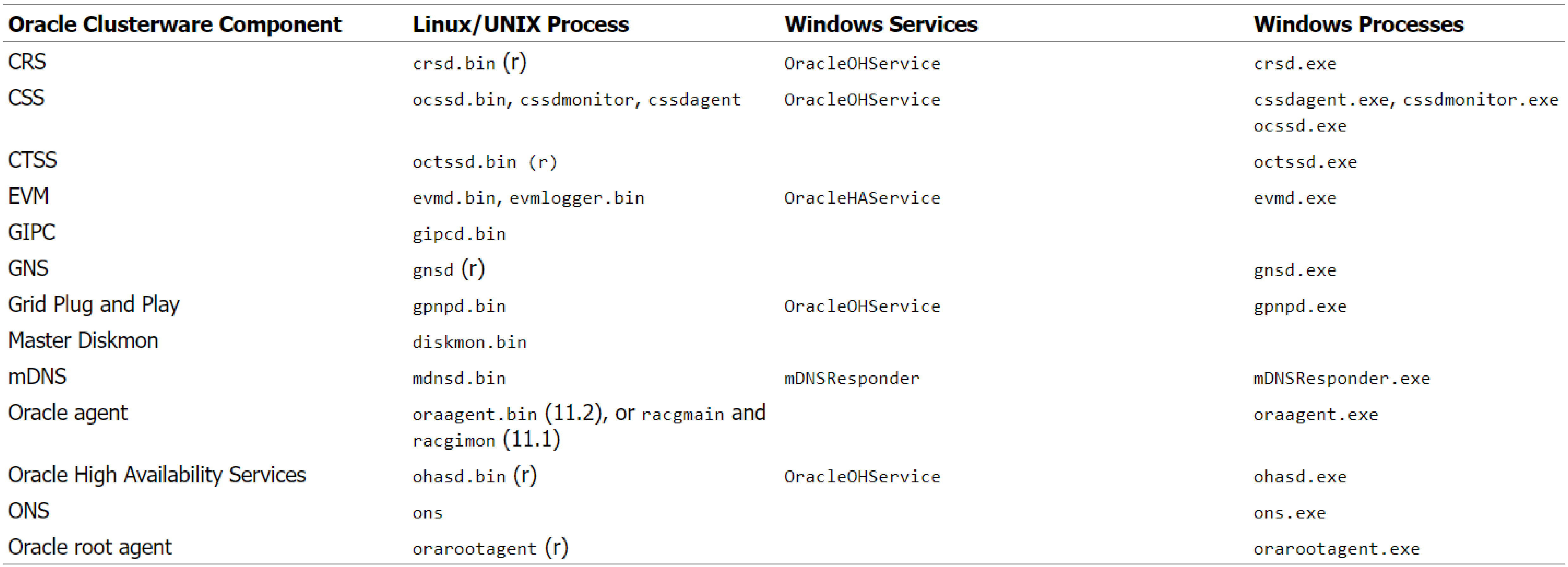
- Scalability of applications
- Use of less expensive commodity hardware
- Ability to fail over
- Ability to increase capacity over time by adding servers
- Ability to program the startup of applications in a planned order that ensures dependent processes are started
- Ability to monitor processes and restart them if they stop

Advantage of Clusterware over others

- Eliminate unplanned downtime due to hardware or software malfunctions
- Reduce or eliminate planned downtime for software maintenance
- Increase throughput for cluster-aware applications by enabling the applications to run on all of the nodes in a cluster
- Increase throughput on demand for cluster-aware applications, by adding servers to a cluster to increase cluster resources
- Reduce the total cost of ownership for the infrastructure by providing a scalable system with low-cost commodity hardware

Oracle Clusterware

Table 1-1 List of Processes and Services Associated with Oracle Clusterware Components



Oracle Clusterware Component	Linux/UNIX Process	Windows Services	Windows Processes
CRS	`crsd.bin` (r)	`OracleOHService`	`crsd.exe`

CSS	`ocssd.bin`, `cssdmonitor`, `cssdagent`	`OracleOHService`	`cssdagent.exe`, `cssdmonitor.exe`
			`ocssd.exe`

CTSS	`octssd.bin (r)`		`octssd.exe`
EVM	`evmd.bin`, `evmlogger.bin`	`OracleHAService`	`evmd.exe`
GIPC	`gipcd.bin`		
GNS	`gnsd` (r)		`gnsd.exe`
Grid Plug and Play	`gpnpd.bin`	`OracleOHService`	`gpnpd.exe`
Master Diskmon	`diskmon.bin`		
mDNS	`mdnsd.bin`	`mDNSResponder`	`mDNSResponder.exe`
Oracle agent	`oraagent.bin` (11.2), or `racgmain` and `racgimon` (11.1)		`oraagent.exe`
Oracle High Availability Services	`ohasd.bin` (r)	`OracleOHService`	`ohasd.exe`
ONS	`ons`		`ons.exe`
Oracle root agent	`orarootagent` (r)		`orarootagent.exe`

The Cluster Ready Services Stack

Cluster Ready Services (CRS): The primary program for managing high availability operations in a cluster. The CRS daemon (crsd) manages cluster resources based on the configuration information that is stored in OCR for each resource. This includes start, stop, monitor, and failover operations.

The crsd process generates events when the status of a resource changes. When you have Oracle RAC installed, the crsd process monitors the Oracle database instance, listener, and so on, and automatically restarts these components when a failure occurs.

Cluster Synchronization Services (CSS): Manages the cluster configuration by controlling which nodes are members of the cluster and by notifying members when a node joins or leaves the cluster. If you are using certified third-party clusterware, then CSS processes interfaces with your clusterware to manage node membership information.

The cssdagent process monitors the cluster and provides I/O fencing. This service formerly was provided by Oracle Process Monitor Daemon (oprocd), also known as OraFenceService on Windows. A cssdagent failure results in Oracle Clusterware restarting the node.

The Cluster Ready Services Stack

Oracle ASM: Provides disk management for Oracle Clusterware.

Cluster Time Synchronization Service (CTSS): Provides time management in a cluster for Oracle Clusterware.

Event Management (EVM): A background process that publishes events that Oracle Clusterware creates.

Oracle Notification Service (ONS): A publish and subscribe service for communicating Fast Application Notification (FAN) events.

Oracle Agent (oraagent): Extends clusterware to support Oracle-specific requirements and complex resources. Runs server callout scripts when FAN events occur. This process was known as RACG in Oracle Clusterware 11g release 1 (11.1).

Oracle Root Agent (orarootagent): A specialized oraagent process that helps crsd manage resources owned by root, such as the network, and the Grid virtual IP address.

The Oracle High Availability Services Stack

The list in this section describes the processes that comprise the Oracle High Availability Services stack. The list includes components that are processes on Linux and UNIX operating systems, or services on Windows.

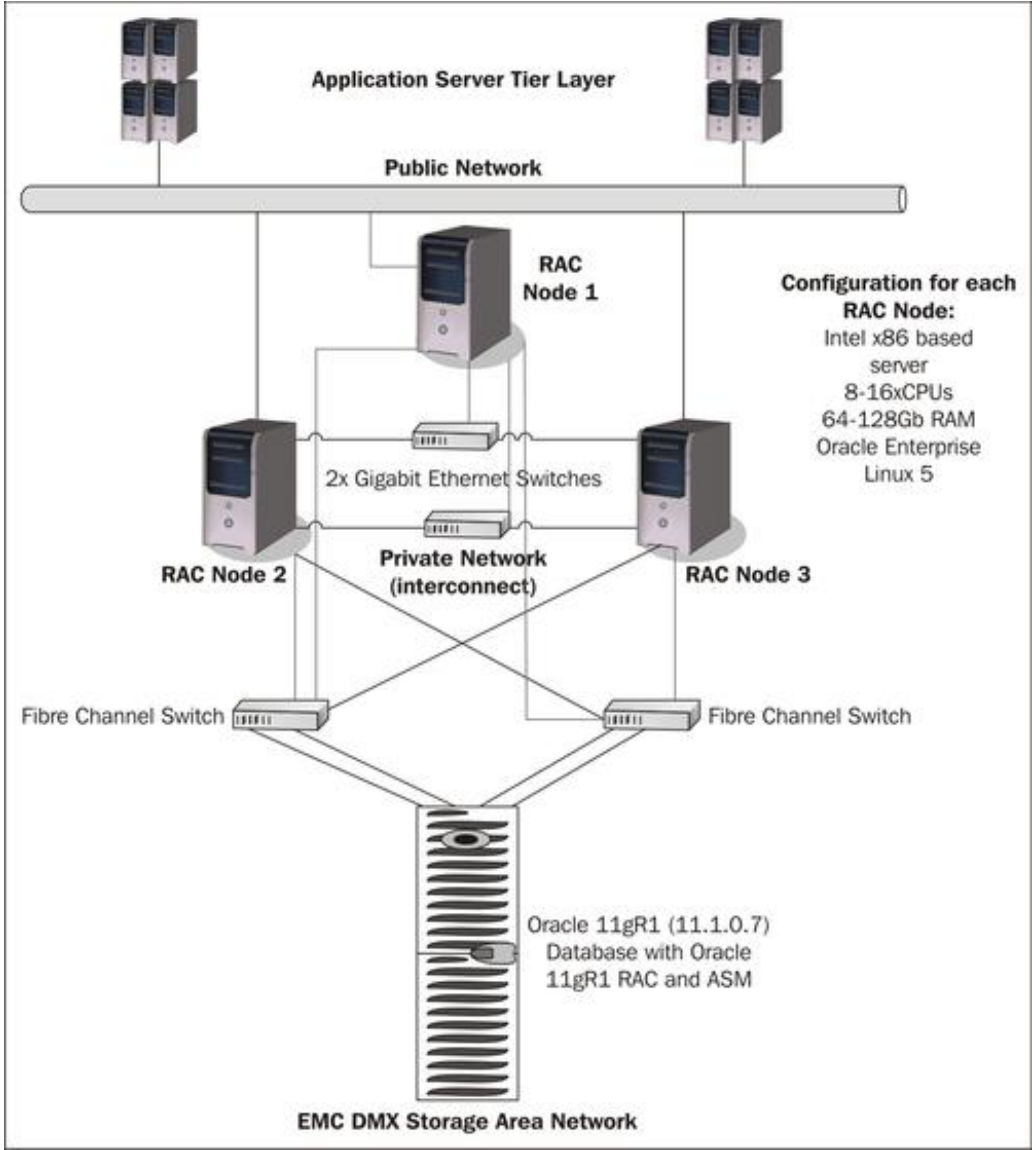
Grid Plug and Play (GPNPD): GPNPD provides access to the Grid Plug and Play profile, and coordinates updates to the profile among the nodes of the cluster to ensure that all of the nodes node have the most recent profile.

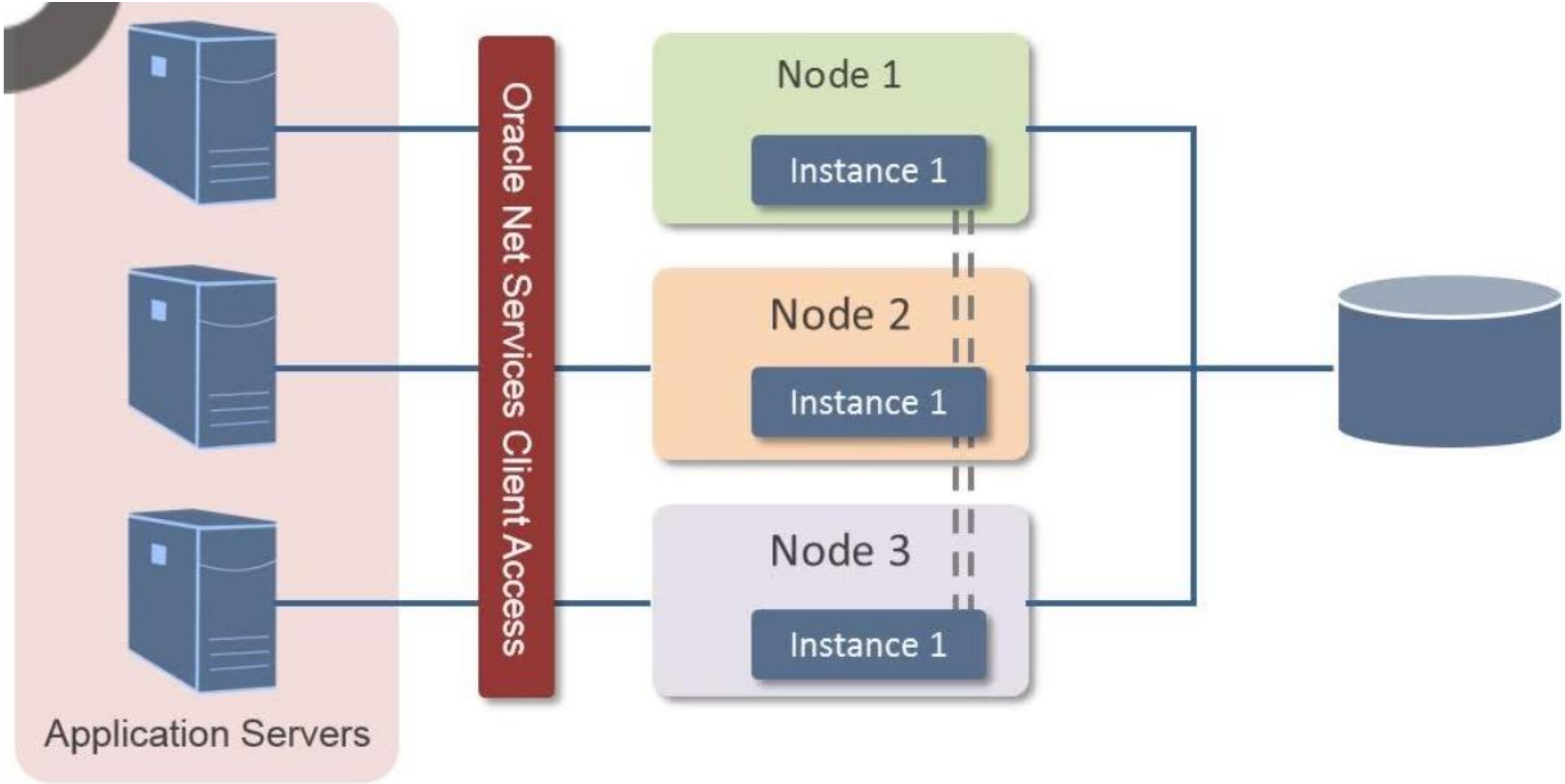
Grid Interprocess Communication (GIPC): A helper daemon for the communications infrastructure. Currently has no functionality; to be activated in a later release.

Multicast Domain Name Service (mDNS): Allows DNS requests. The mDNS process is a background process on Linux and UNIX, and a service on Windows.

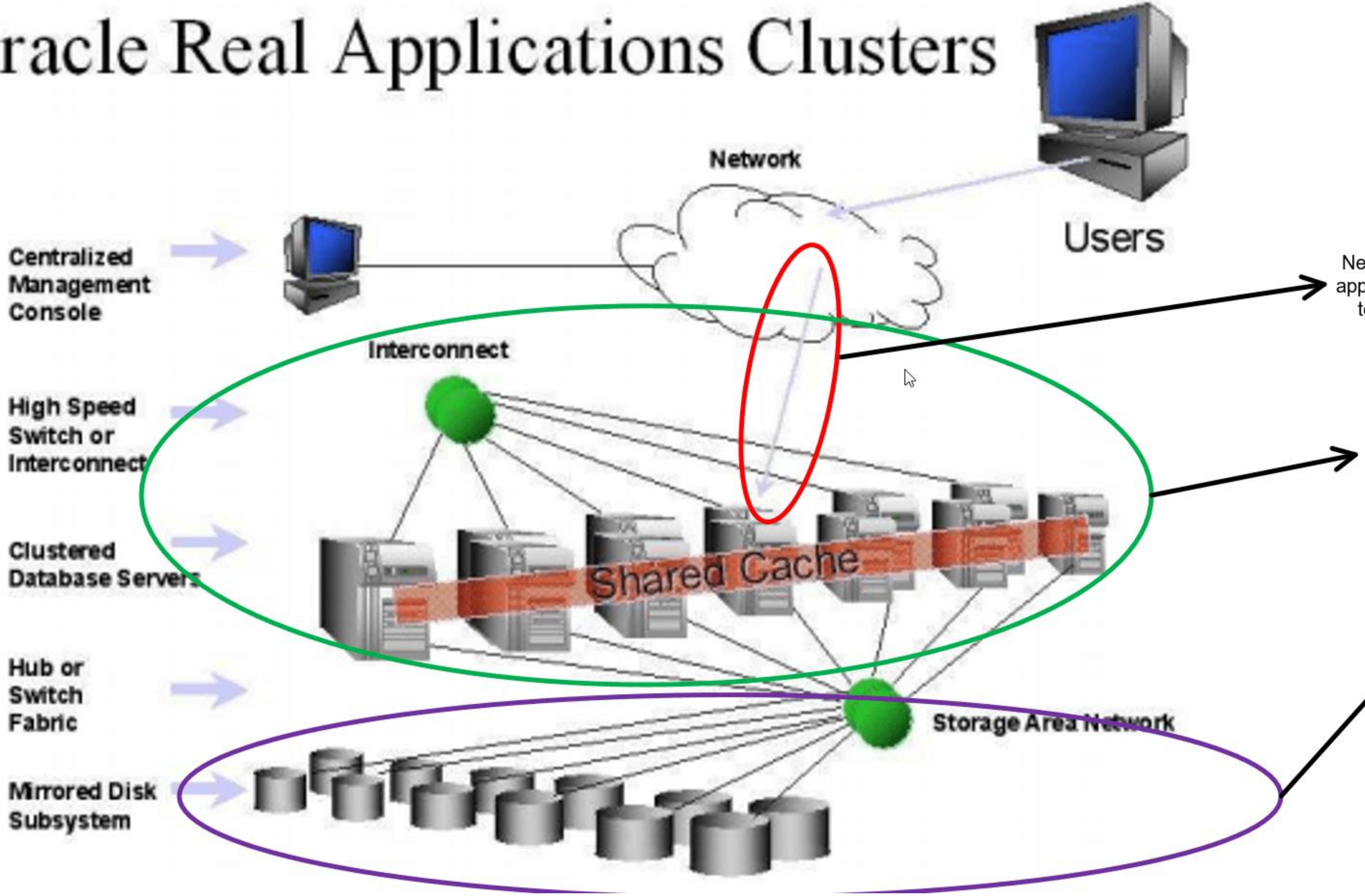
Oracle Grid Naming Service (GNS): A gateway between the cluster mDNS and external DNS servers. The gnsd process performs name resolution within the cluster.

Architecture Diagram Reference


Application Server Tier Layer
Public Network
RAC
Node 1
Configuration for each
RAC Node:
Intel x86 based
server
8-16xCPUs
64-128Gb RAM
Oracle Enterprise
Linux 5
2x Gigabit Ethernet Switches
Private Network
(interconnect)
RAC Node 2
RAC Node 3
Fibre Channel Switch
Fibre Channel Switch
Oracle 11gR1 (11.1.0.7)
Database with Oracle
11gR1 RAC and ASM
EMC DMX Storage Area Network


Oracle Net Services Client Access
Node 1
Instance 1
Node 2
Instance 1
Node 3
Instance 1
Application Servers

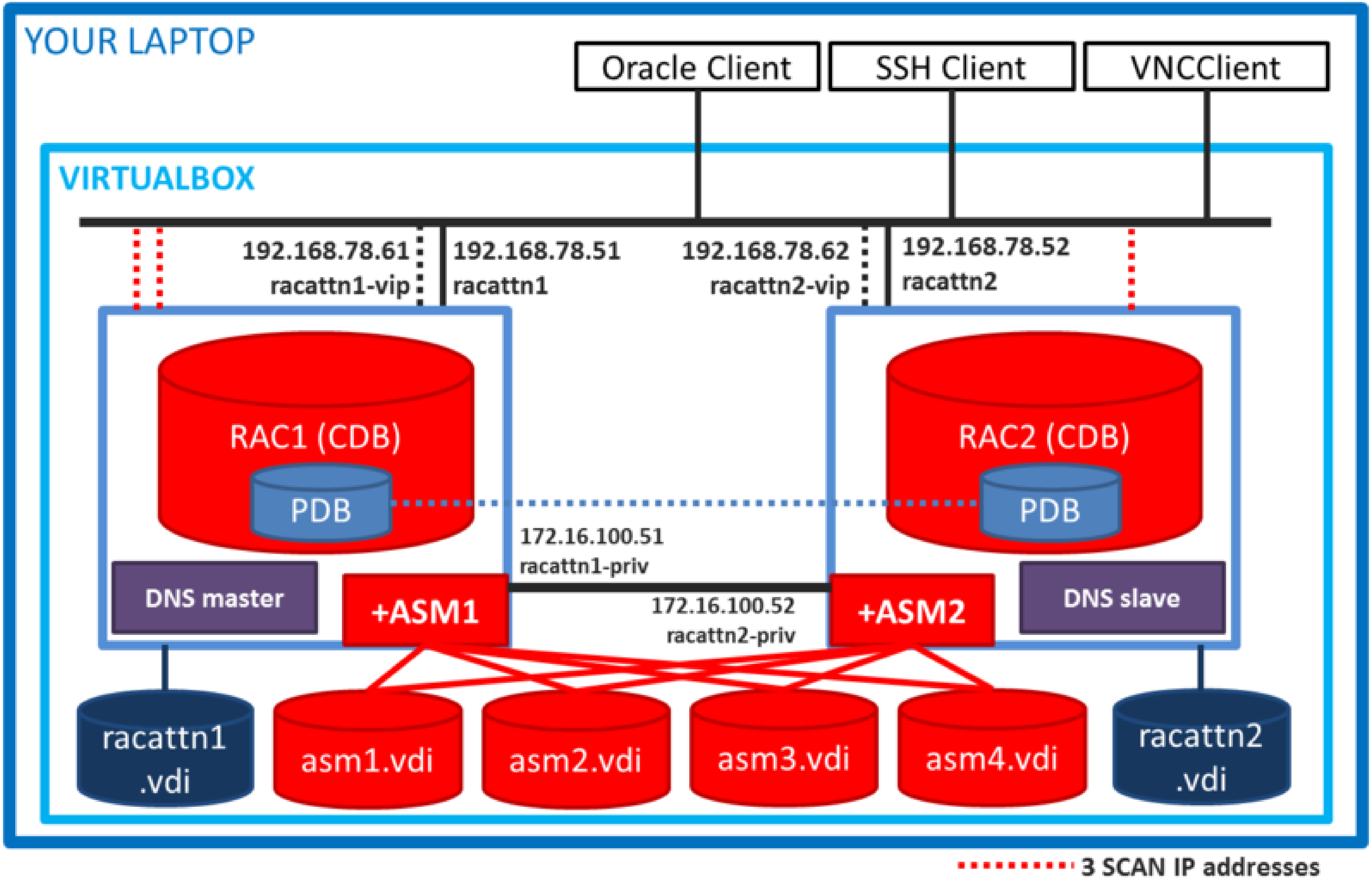
Oracle Real Applications Clusters

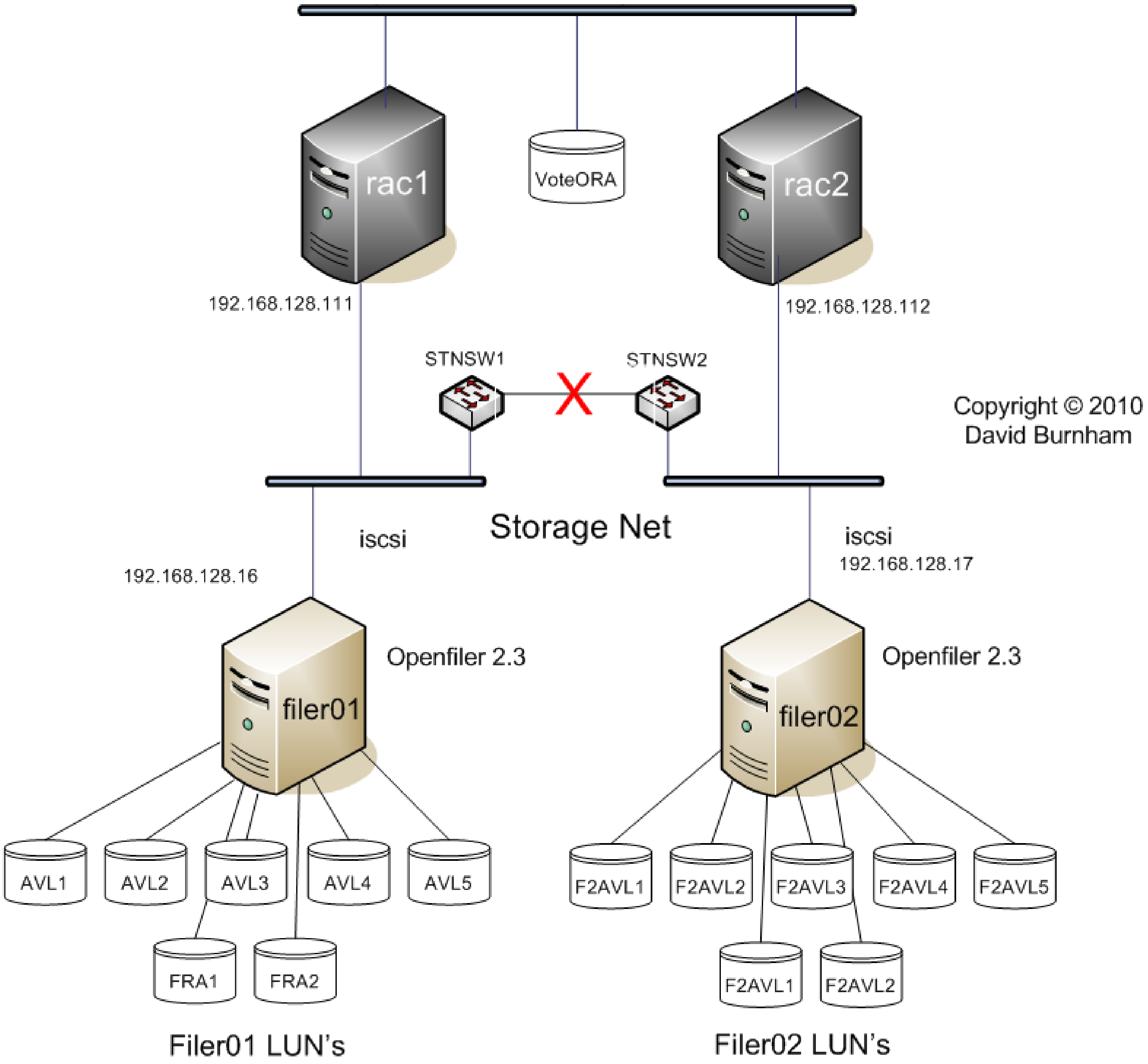


Public Network
Network which users and application severs connect to access data on the database server

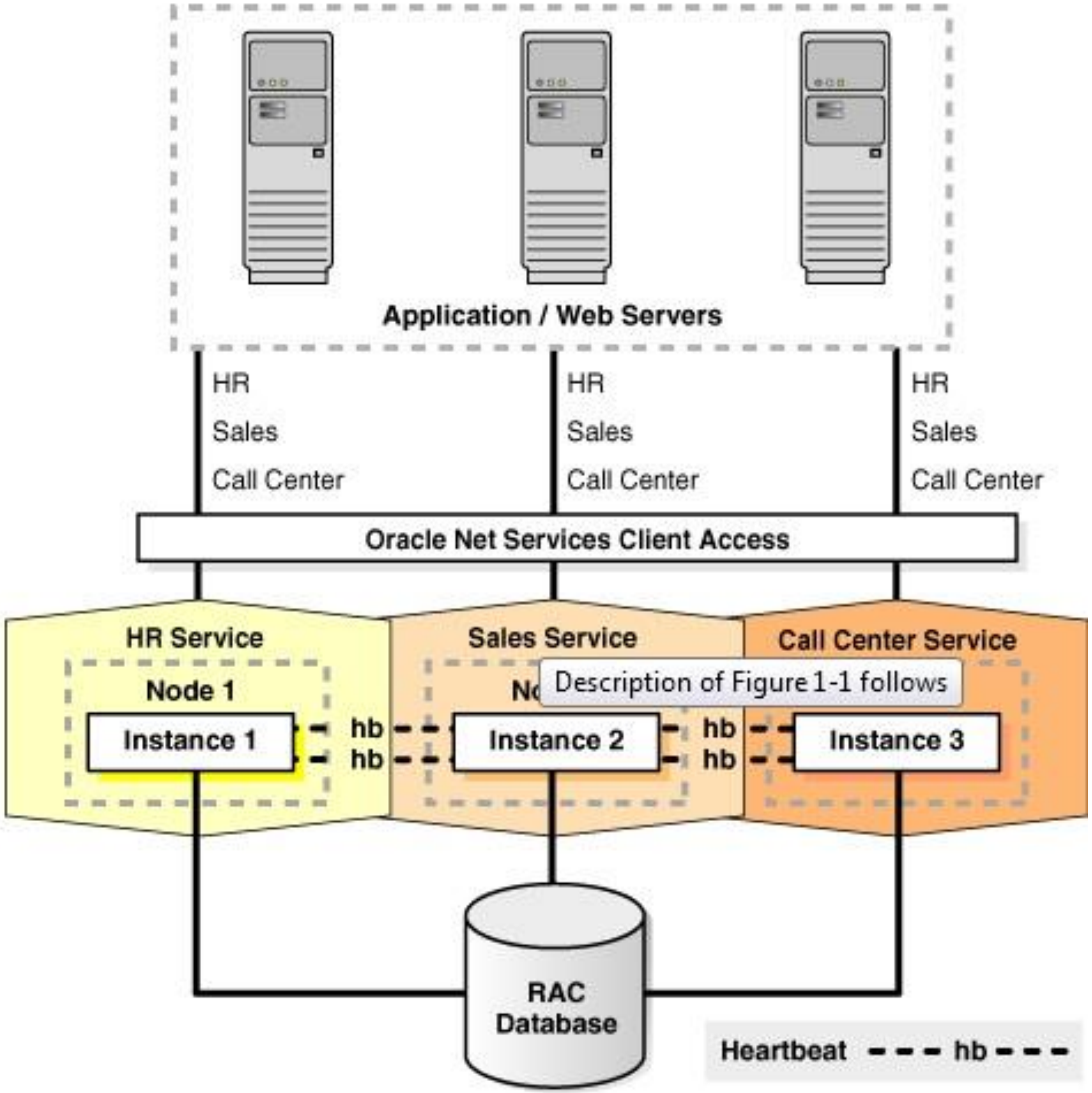
Private Network
Network used for internode communication

Storage Area Network
High-seed specialized network that provides block-level network access to storage

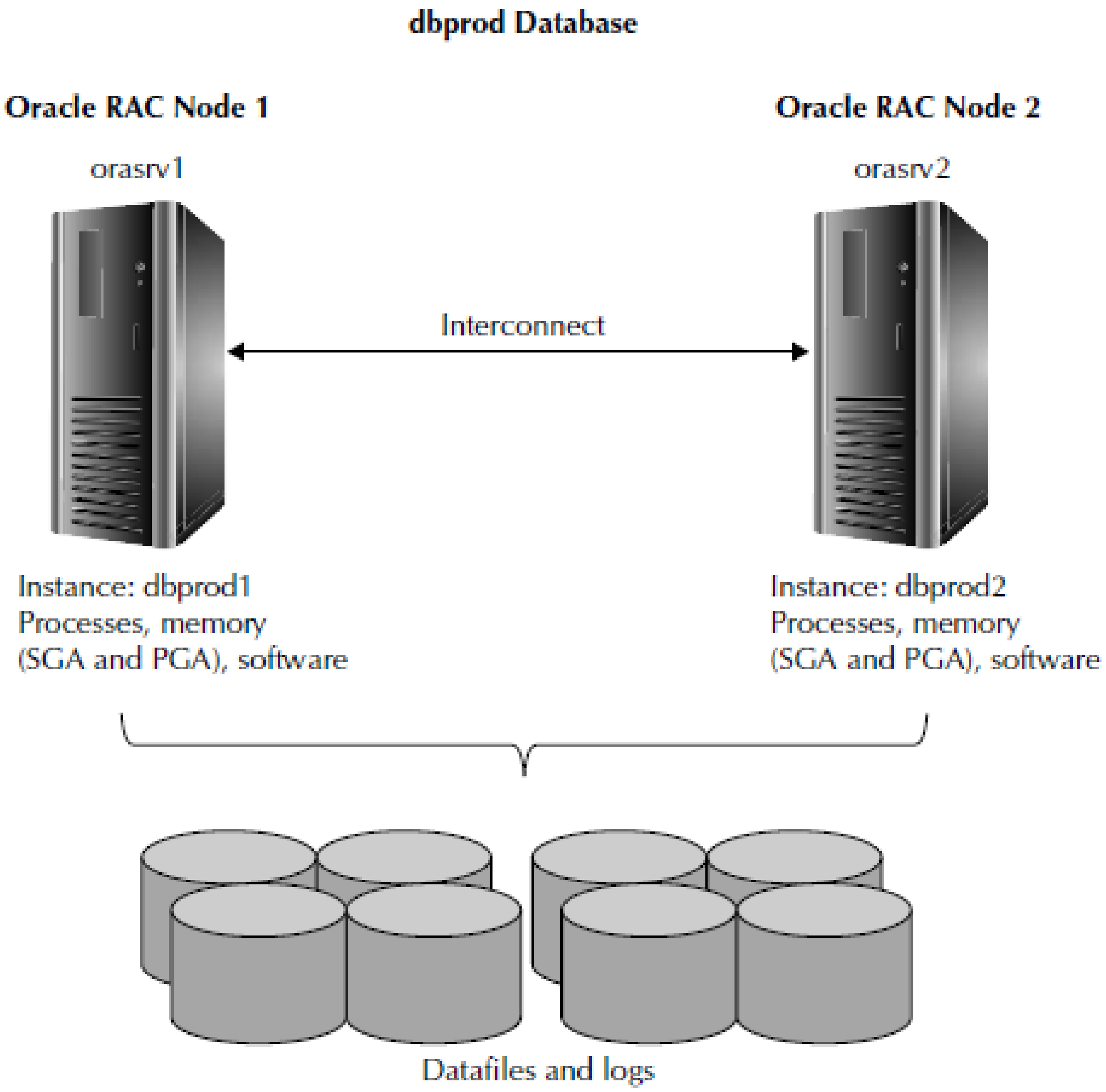

YOUR LAPTOP
Oracle Client
SSH Client
VNCClient
VIRTUALBOX
192.168.78.61
racattn1-vip
192.168.78.51
racattn1
192.168.78.62
racattn2-vip
192.168.78.52
racattn2
RAC1 (CDB)
PDB
RAC2 (CDB)
PDB
172.16.100.51
racattn1-priv
172.16.100.52
racattn2-priv
DNS master
+ASM1
+ASM2
DNS slave
racattn1
.vdi
asm1.vdi
asm2.vdi
asm3.vdi
asm4.vdi
racattn2
.vdi
3 SCAN IP addresses

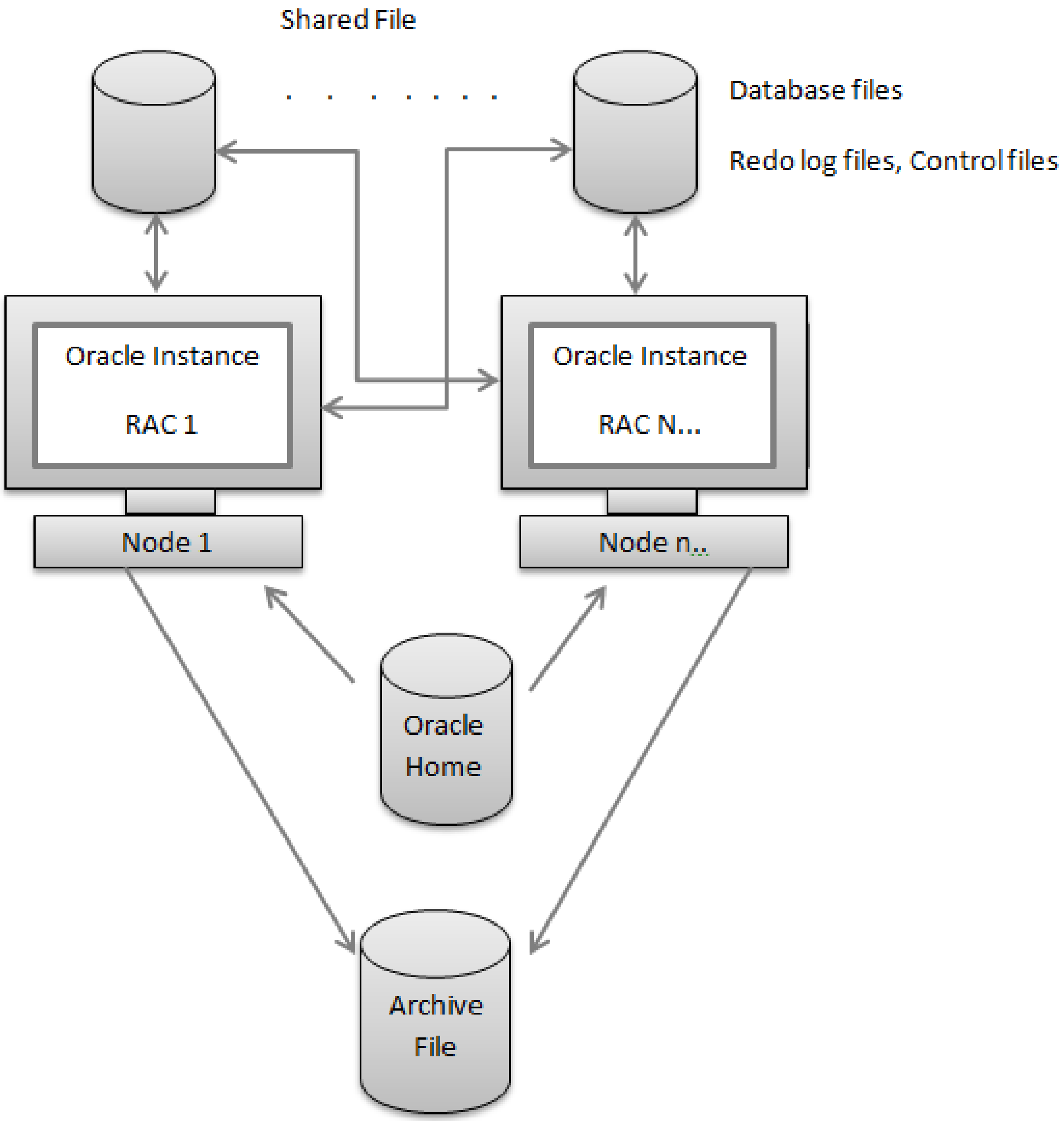


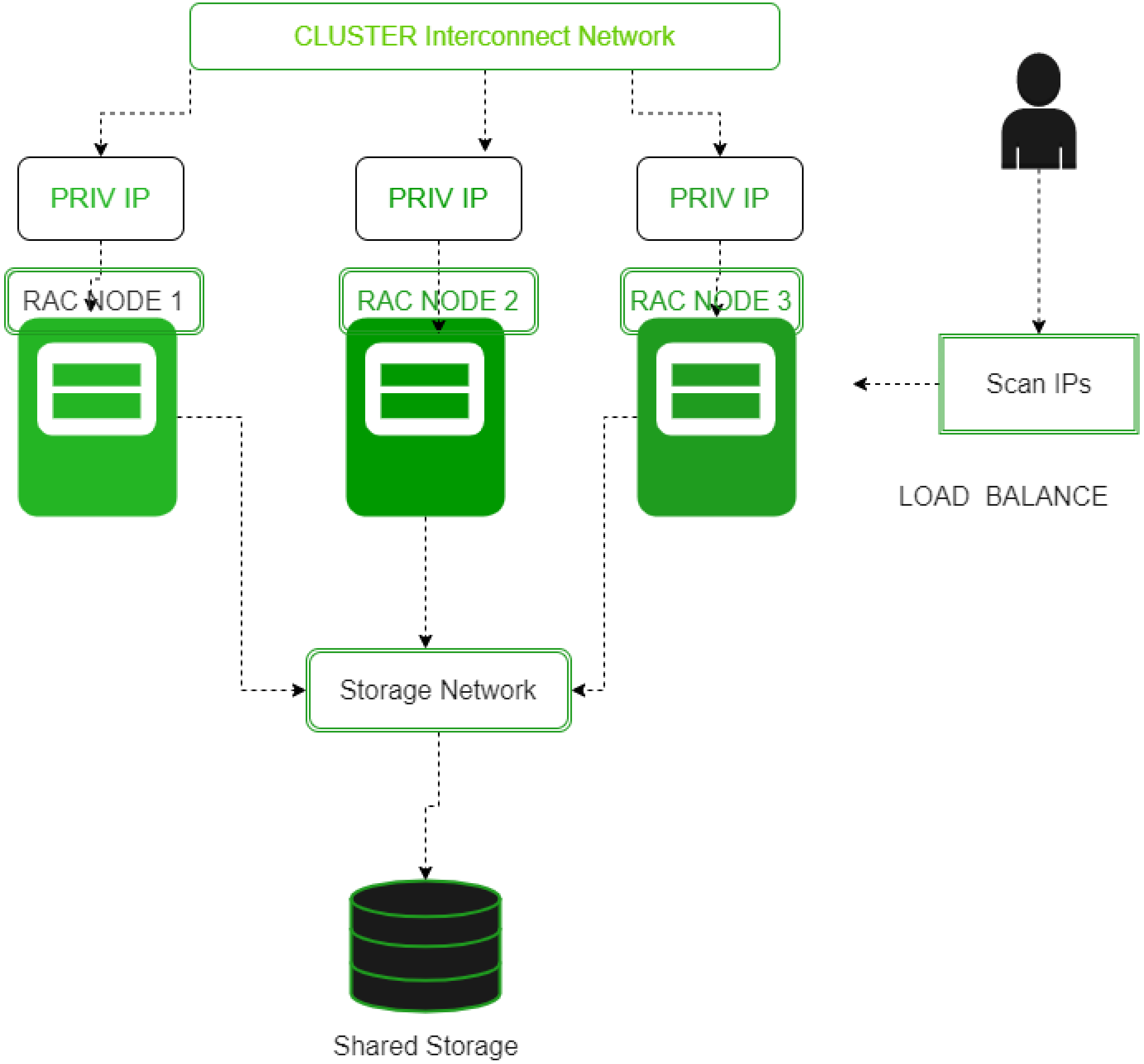
Storage layout on Extended RAC Test Cluster

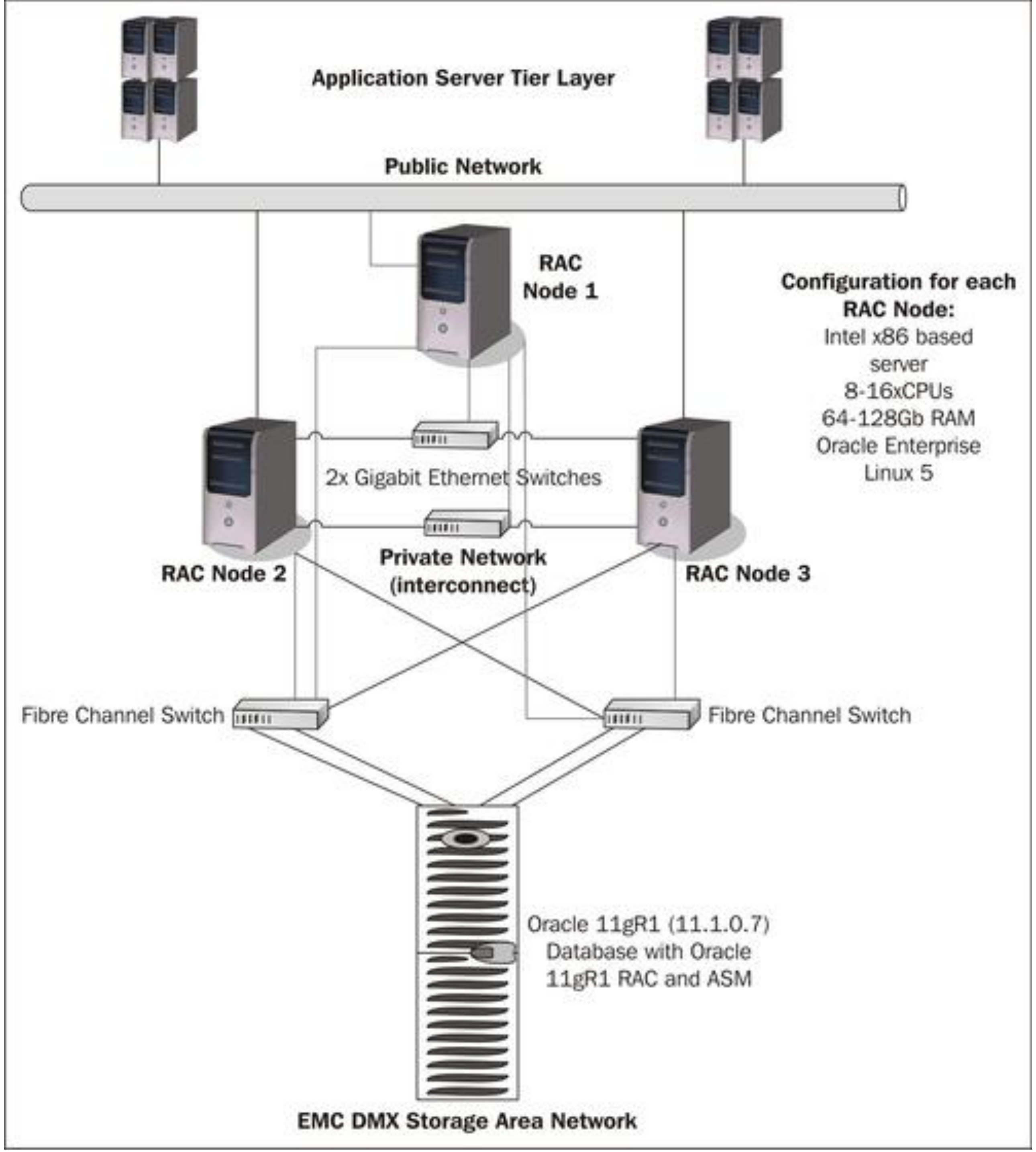


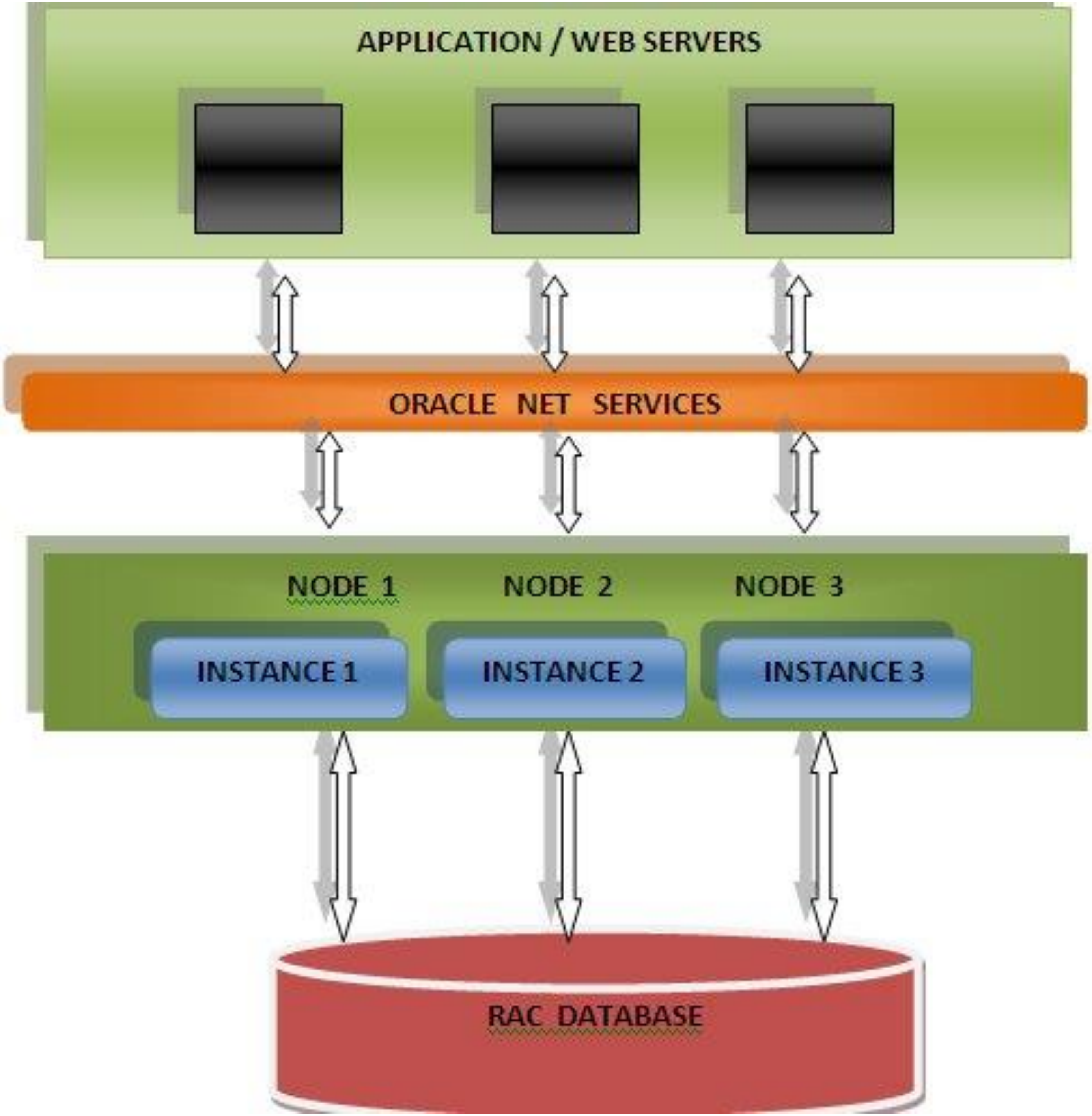
Description of Figure 1-1 follows


dbprod Database
Oracle RAC Node 1
orasrv1
Oracle RAC Node 2
orasrv2
Interconnect
Instance: dbprod1
Processes, memory
(SGA and PGA), software
Instance: dbprod2
Processes, memory
(SGA and PGA), software
Datafiles and logs


Shared File
. . . .
Database files
Redo log files, Control files
Oracle Instance
RAC 1
Oracle Instance
RAC N...
Node 1
Node n..
Oracle Home
Archive File


CLUSTER Interconnect Network
PRIV IP
PRIV IP
PRIV IP
RAC NODE 1
RAC NODE 2
RAC NODE 3
Scan IPs
LOAD BALANCE
Storage Network
Shared Storage


Application Server Tier Layer
Public Network
RAC Node 1
Configuration for each RAC Node:
Intel x86 based server
8-16xCPUs
64-128Gb RAM
Oracle Enterprise Linux 5
2x Gigabit Ethernet Switches
RAC Node 2
Private Network (interconnect)
RAC Node 3
Fibre Channel Switch
Fibre Channel Switch
Oracle 11gR1 (11.1.0.7) Database with Oracle 11gR1 RAC and ASM
EMC DMX Storage Area Network


APPLICATION / WEB SERVERS
ORACLE NET SERVICES
NODE 1
NODE 2
NODE 3
INSTANCE 1
INSTANCE 2
INSTANCE 3
RAC DATABASE

DevOps@RajeshKumar.xyz

www.RajeshKumar.xyz