An Architect's Guide to the Oracle Private Database Cloud

Reference Architecture Overview

ORACLE ENTERPRISE ARCHITECTURE WHITE PAPER | DECEMBER 2014

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Table of Contents

Executive Summary

Private Database Cloud Services or Database as a Service (DBaaS) is no longer a new idea. In fact, it is quickly becoming the de facto standard for development and testing environments both on premises and in the public cloud. And while there are many use cases and deployment options, overall database total cost of ownership and business agility have benefited from a standardized approach to workload management. Whether you are a DBA, an operations manager, or a CIO, you are well aware that business-driven interest in social, mobile, big data, and internet of things have caused an explosion of development, data, and database workloads. The justification for database operations to pool resources and standardize services has never been clearer – watch this customer story (TRT1:30).

Today's best practices in cloud architecture require server scalability, zero data loss resiliency, and most importantly workload security and isolation through multitenancy. For database services, the architecture would be incomplete if database operations did not also natively support multitenancy. Initial approaches for DBaaS were limited since they only relied upon virtual machines for workload isolation and database provisioning. As general technology containers, VMs had no intrinsic understanding of database operations, so they were unable to optimize performance, scalability, and resilience as well as simplify database administration efforts.

Today's best practice for database cloud services overcomes these limitations. The Oracle Private Database Cloud approach is both revolutionary and elegantly simple. By engineering multitenant capabilities throughout the Oracle database platform, the complete range of database operations and administration can now be natively managed *and* without the overhead of a virtual machine. Oracle's Private Database Cloud guarantees isolation and leverages Oracle's strengths in reliability, scalability, security, and systems management. Large database estates also benefit from a host of related capabilities, such as cost-recovery reporting, self-service management, and public cloud integration. You will find that Oracle database platform is ideal for a standardized enterprise deployment or cloud service, whether development/test or production – watch this customer story (TRT2:17).

This white paper presents a reference architecture overview and Oracle product mapping for DBaaS in a private cloud deployment model. The approach and guidance offered is the byproduct of hundreds of customer projects and highlights the decisions that customers faced in the course of their architecture planning and implementations. Oracle's advising architects work across many industries and government agencies and have developed standardized methodology based on enterprise architecture best practices. Oracle's enterprise architecture approach and framework are articulated in the Oracle Architecture Development Process (OADP) and the Oracle Enterprise Architecture Framework (OEAF).

Fundamental Concepts

What is a Private Database Cloud and Database as a Service?

Two terms have evolved to describe a cloud-based approach to managing databases: Private Database Cloud and Database as a Service (DBaaS). DBaaS describes an architectural and operational approach that enable IT providers to deliver database functionality as a service to one or more consumers. A Private Database Cloud is a term that refers to a database as a service architecture that is always under the strategic control of the enterprise whether or not it is deployed on or off premises. For this paper we will use the term DBaaS to describe the architectural and operational approach only.

The principles of a DBaaS architecture supports the following necessary capabilities:

- Consumer-based provisioning and management of database instances using on-demand, self-service mechanisms;
- Automated monitoring of and compliance with provider-defined service definitions, attributes and quality of service levels;
- Fine-grained metering of database usage enabling show-back reporting or charge-back functionality for each individual consumer

In addition to these required characteristics, it is expected that DBaaS architectures will naturally support granular service elasticity, secure multitenancy, access using a broad range of non-proprietary devices and mechanisms, automated resource management, and integrated capacity planning. While these architectural attributes are certainly important, they are equally essential for traditional database service architectures and therefore are not specific to what makes Database as a Service new and unique.

Why Consider Database as a Service?

Reduced Costs

Business and IT leaders understand the need for accurate and timely information when making decisions that impact their business. To ensure that the right information is available when it is needed, IT projects often spend a large portion of their time, resources and money to create what amounts to individual information silos, with distinct requirements, configurations, and support models. Historically, production environments have been configured for peak load, peak performance and uninterrupted business continuity. They have been shadowed by multiple development, testing, quality assurance environments that mirror various aspects of their production counterparts. The acquisition and ongoing maintenance and support costs associated with this approach are significant and are often repeated for each business application.

DBaaS affords organizations an opportunity to standardize and optimize on a platform that eliminates the need to deploy, manage and support dedicated database hardware and software for each project's multiple development, testing, production, and failover environments. DBaaS architectures are inherently designed for elasticity and resource pooling. DBaaS environments can deliver production and non-production database services that support average daily workload requirements while scaling-up to handle increased demand and scaling-down when that demand has subsided.

Improved Service Levels

The service orientation aspects of DBaaS architectures benefit both IT providers and consumers. Providers benefit from being able to develop and offer pre-defined services for their consumers to use – minimizing vendor, software version and configuration diversity. This reduced diversity supports business goals of agility, efficiency and improved quality of service through the development of standardized processes, common support mechanisms and focused skills development. Further, by focusing on a standardized and optimized platform, providers can also offer the illusion of unlimited capacity to their consumers through improvements in procurement, capacity planning, and resource management.

Consumers also benefit in a number of ways: (1) self-service provisioning and management means that consumers are able to more quickly and easily deploy new applications for the business; (2) service definitions allow consumers to understand exactly what capabilities and service levels they should expect from the provider; and (3) metering allows for greater transparency into utilization and IT costs allowing consumers to make more informed decisions about where to spend their time and money.

The use of industry and vendor recommended practices, reference architectures, standardized configurations, and repeatable IT processes enable organizations to foster a culture of continuous improvement while reducing risk and down-time. Reference architectures, in particular, can help organizations better understand how and what capabilities will be delivered, how security, reliability, availability, manageability, and recovery requirements will be supported, and what trade-offs must be considered in order to meet the performance, availability, regulatory compliance, security and other requirements of the business.

Enhanced Information Access and Rationalization

Another common practice within organizations stems from the misperception that information requirements are so unique that each line of business or region must maintain separate and dedicated database environments. This explosion and unnecessary duplication of information impacts the integrity and quality of data and results in the need to introduce additional capabilities (and supporting infrastructure) to the business. DBaaS offers organizations a better approach. Organizations can rationalize their information architecture controlling this information explosion using data virtualization and security controls to give data owners the ability to control how and when data is shared while also eliminating the need for redundant database instances.

What is Different about Database as a Service?

There are many approaches to delivering database services today, but Database as a Service differs from traditional architectures in its service-orientation and focus on customer self-service interaction models. Together, these two attributes enable DBaaS architectures to realize significant benefits in terms of increased security, efficiency and service levels while also providing greater transparency and flexibility with costs.

Database as a Service architectures use a service catalog to define the database services, attributes and quality of service levels that are offered. The actual way in which the services are defined and grouped is often provider-specific, but one can easily envision a collection of database services such as:

DBaaS Service Catalog

DBaaS Service Name	CPU Size	Server Memory	Storage Redundancy	Service Availability	Security Options	Backup / Archival	DR
Small OLTP	4 CORES	6 GB	2-Way / 3-Way	Node / Server / Site	Encryption / Strong Auth	Backup / Archival	DR
Medium OLTP	8 CORES	12 GB	2-Way / 3-Way	Node / Server / Site	Encryption / Strong Auth	Backup / Archival	DR
Large OLTP	16 CORES	18 GB	2-Way / 3-Way	Node / Server / Site	Encryption / Strong Auth	Backup / Archival	DR
Small OLAP	2 CORES	12 GB	2-Way / 3-Way	Node / Server / Site	Encryption / Strong Auth	Backup / Archival	DR
Medium OLAP	4 CORES	24 GB	2-Way / 3-Way	Node / Server / Site	Encryption / Strong Auth	Backup / Archival	DR
Large OLAP	8 CORES	48 GB	2-Way / 3-Way	Node / Server / Site	Encryption / Strong Auth	Backup / Archival	DR

Figure 1: Sample DBaaS Service Catalog

It is important for DBaaS providers to develop well-defined, unambiguous service catalogs as well as comparable pricing models so that consumers can quickly and easily understand what is being offered and at what price. You can find further discussion and examples of DBaaS service catalogs – here.

Another unique aspect of DBaaS architectures is the strict provider-consumer relationship that impacts how consumers register with the service, perform account management operations, define payment methods, review service utilization, as well as provision and use database services. The need to support secure multitenancy means that consumer access to the privileged functions and even the underlying infrastructure must be limited. Providers must determine what functions and capabilities to expose to their consumers (using management portals or APIs) in order to offer the greatest value to their consumers while also managing risk and ensuring privacy.

Considering Moving to Database as a Service?

Business Considerations

DBaaS is only interesting when it delivers quantifiable benefits to the business, both internally and externally. To determine the benefits of DBaaS to an organization, consider quantifying the following business requirements:

» Business user benefit – DBaaS must be able to benefit the business directly or indirectly. For example, would a business user be able to accomplish their work better, faster or in a more cost-effective manner if DBaaS was available?

» Business processes benefit – DBaaS should provide a measurable benefit to business processes such as supply chain management, enterprise resource planning, or human capital management. For example, could DBaaS provide a more agile and optimized foundation to support key business processes that rely on database services?

» Contribution to strategic business goals and objectives – DBaaS may support strategic business objectives. For example, a standardized and optimized platform such as DBaaS could help reduce the time required to create, migrate or integrate new applications resulting from acquisitions or new lines of business.

IT Considerations

Strategy -- Organizations must understand their enterprise IT strategy when considering DBaaS. When considering DBaaS deployment options, organizations must understand their IT operating model, how it could impact the deployment and whether any changes needed. Coordinated and unified IT operating models are best suited to organization-wide deployments due to the inherent service-orientation of DBaaS architectures, although DBaaS architecture can be easily leveraged by most organizations to different degrees.

Another key component of an IT operating model is governance. When considering DBaaS architectures, organizations must consider how the service will be governed and how compliance with corporate standards, policies and external (industry or regulatory) mandates will be maintained. Further, the DBaaS architecture must also align with enterprise architecture (EA) and IT strategy, policy and plans.

People and Process – Another key factor to consider is how to "operationalize" DBaaS. The concepts of best practices, reference architectures, and service-oriented operating models will need structure and a framework for IT to deliver DBaaS to the consumer in a consistent, repeatable, and measurable way. Industry standards and frameworks such as ITIL® (IT Infrastructure Library), COBIT (Control Objectives of Information and related Technology), and OEAF (Oracle Enterprise Architecture Framework) are a few of the frameworks that have been successfully used by organizations deploying DBaaS.

Technology – Once a provider has defined its strategy and understands the frameworks that will be used to develop and "operationalize" their DBaaS deployment, they are free to begin consideration of technology options. Technology choices are usually provider specific as they are driven by a variety of factors including business and IT requirements, services to be offered, existing resources, assets and skills, architecture principles and technology standards.

Architectural Perspectives

DBaaS is a paradigm shift that impacts the EA landscape and in turn is affected by the ecosystem in which it is deployed. While DBaaS can be deployed independently, organizations will extract the greatest value when it is well integrated into the organization's overall enterprise architecture and IT strategy. Therefore, for successful adoption and sustainable execution, it is important to understand how DBaaS fits into the overall environment and what the enterprise-wide considerations are to correctly architect and actualize DBaaS.

Architecture Principles

Architecture principles are the core tenets that guide and constrain architectural decision making. The form in which these principles are articulated varies from terse maxims to highly detailed catalogues. Regardless of the format, however, all DBaaS architectural decisions should be traceable back to the set of DBaaS architecture principles. The following are a few examples of DBaaS architecture principles:

- Business Alignment. The DBaaS architecture must be in alignment with the business operating model of the enterprise.
- Value Tracking. The value provided by DBaaS architecture will be measured using established key value indicators (KVIs).
- Standardized Infrastructure. The DBaaS architecture must be built on a standardized portfolio of technologies and assets.
- Ongoing Architecture Improvement. The service catalogue and architecture will be continually rationalized to prevent service explosion.
- Cost Competitiveness. The charge-back rates of the database services will be competitive with external providers.
- Provider Consumer Collaboration. Database services, including definitions, attributes and quality of service shall be established and maintained in active collaboration with service consumers.

DBaaS Architecture Domains

Business Architecture

Business Architecture contextualizes the business aspects for DBaaS with the architecture typically mirroring the way that the Lines of Business (LoBs) are organized and run. Clearly, greater synergy between the LoBs allows for greater sharing and efficiency. A highly diversified company will likely have different requirements for database services when compared to a highly unified company. The following are a few of the key business requirements that can differ as a result of an organization's operating model:

- Different chargeback models or rate structure. Some LoB may need periodic reservations while others may prefer strict usage-based charge-back. In other cases, show-back reporting may be sufficient depending upon their overall funding model.
- Unique quality of service requirements. LOBs may have differing needs with respect to performance, scalability, security, high availability, backup and archive, and disaster recovery. While this is often factored into service definitions, it is important to understand that there are still limits that must be considered and negotiated.
- Unique variations in workflow. LOBs may have specific workflow requirements in order to comply with management, finance, security, audit or other requirements. Where possible, however, DBaaS providers should strive to limit this kind of diversity or provide ways to allow LOBs to develop and integrate their own workflow into a common framework to promote a standardized and optimized core platform.

The decision on how best to standardize, structure and deploy DBaaS is highly contingent on the direction the executive leadership team and the organization's operating model. Considering these issues at the start will help avoid management conflicts, culture clashes and other conflicts later on.

Application Architecture

For most organizations, their application portfolio ranges from transactional applications to analytical applications; business critical to low value applications; compute centric to network centric and homegrown applications to COTS applications. At this stage, one of the first steps that an organization should take is to determine which applications are best suited to operate in a DBaaS environment. Application architecture, workload characteristics, and a host of other factors should be considered to evaluate applications and develop a prioritized list of those best suited for migration.

Clearly, not all applications are equal. There are several classes of workloads that are more appropriate for DBaaS architectures including:

- Applications with "spikey" workloads. Applications impacted by seasonal traffic or peaking at periodic or well-defined times. DBaaS environments can allow these applications to flex, expand and contract based upon demand. The effectiveness of this approach depends however on the applications ability to leverage the additional capacity made available by the DBaaS environment.
- Transient workloads. This self-service provisioning and metered usage of the DBaaS architecture make it ideal for transient workloads such as those often associated with development, testing, quality assurance, training and similar functions.

Beyond understanding workloads, organizations should also consider how the use of a DBaaS capability will impact the development, procurement, and use of new applications and services and whether changes to an organization's IT standards, procurement policies or application development life cycle processes may be in order.

Information Architecture

Organizations must take into consideration how their overall information lifecycle flows and integrates across DBaaS and non-DBaaS styles of architecture. In particular, organizations should be on guard against information fragmentation as consumers are given greater flexibility and control over where and how they store their information. Common vocabularies, definitions, semantic data models, and standard data exchange formats must not be abandoned just because consumers can provision their own database instances. Additional checks and balances are also needed to prevent database sprawl, information silos, and a proliferation of unused or unnecessary data stores.

Information should always be protected in a way commensurate with its business value. This does not change for DBaaS architectures. Highly valuable information should utilize service instances that provide the appropriate quality of service levels. Similarly, organizations should be careful to not deploy services with quality of service levels beyond their needs as they will end up paying for capabilities that they will not need or use. Ideally, enterprise information policies should be developed that provide guidance by linking information categories with DBaaS service definitions to help ensure that consumers are well informed as to which should be used when.

Finally, the DBaaS consumer-provider model creates an interesting twist for information integration and security. There may be requirements for data separation, co-location or near-location that may need to be supported by the DBaaS architecture for security or performance reasons. Organizations should work to understand the degree to which these types of requirements exist as they have a substantial impact on the underlying architecture and how instances are created and capacity is assigned to consumers. DBaaS providers may require additional insight from consumers regarding meta-information, information affinity and information dependencies to correctly configure the database service provisioning algorithms beyond simple resource utilization characteristics. The criterion governing the placement of information assets should be discussed and agreed early to mitigate reactive rearrangement of database services as an afterthought.

Technology Architecture

The capabilities and qualities provided by a DBaaS are highly dependent on the underlying technology architecture – which must be chosen judiciously, balancing the needs of the consumers with the overall maturity of the organization. Mission critical database services may require advanced pre-engineered systems while transient workloads could leverage virtualized commodity hardware. While transparent to the consumer, both of these models could co-exist in the same DBaaS environment providing distinct service levels (mapped to specific sets of service definitions and quality of service attributes).

By lowering barriers to entry, organizations will likely use their DBaaS to experiment and incubate new applications and ideas to help impact the business. While from the consumer point of view, the DBaaS environment has the illusion of infinite capacity, the reality is that there are limits to processing, memory and storage capacity as well as network bandwidth. Organizations should develop an agile capacity planning strategy as part of their DBaaS architecture. While there are a variety of models that can leveraged, what is important is that the model is right-sized for the business. That said, well-integrated capacity monitoring, procurement and provisioning processes (with known and reliable lead times) will certainly help to quickly correct capacity imbalances. Other approaches include the use of capacity throttling and even suspending less critical instances to offer that capacity to more business critical services.

Lastly, security plays a critical role in shaping the details of the DBaaS technical architecture. Organizations must develop a clear understanding of security requirements, trust boundaries, threat profiles. Further, organizations must understand how trust is established and propagated as well as how threats are mitigated across the DBaaS environment. Organizations need to understand where and how security policy is defined, how access is managed, and how audit and compliance requirements are met given the distinct capabilities that providers and consumer have in a DBaaS architecture.

Architecture Views

The following are a series of architecture views that define and refine the capabilities required to support a DBaaS strategy. The architecture views build from a foundation of conceptual capabilities to a sample physical implementation of the technology providing those capabilities.

» Conceptual Architecture. Illustrates the high-level relationships between the actors, capabilities and functions required to support a DBaaS deployment.
» Logical Architecture. Refines the conceptual architecture to identify the functional capabilities and their relationship to the processes supporting the implementation of a DBaaS deployment. For brevity, only the management capabilities and process overview is included in this document.
» Physical Architecture. Defines the physical implementation of services and systems required to support a DBaaS deployment.

Conceptual Architecture Overview

The DBaaS conceptual model contains the core capabilities supporting the delivery of database services to an organization through a service-oriented and self-service driven model. The following is an overview of these capabilities and their relationships:

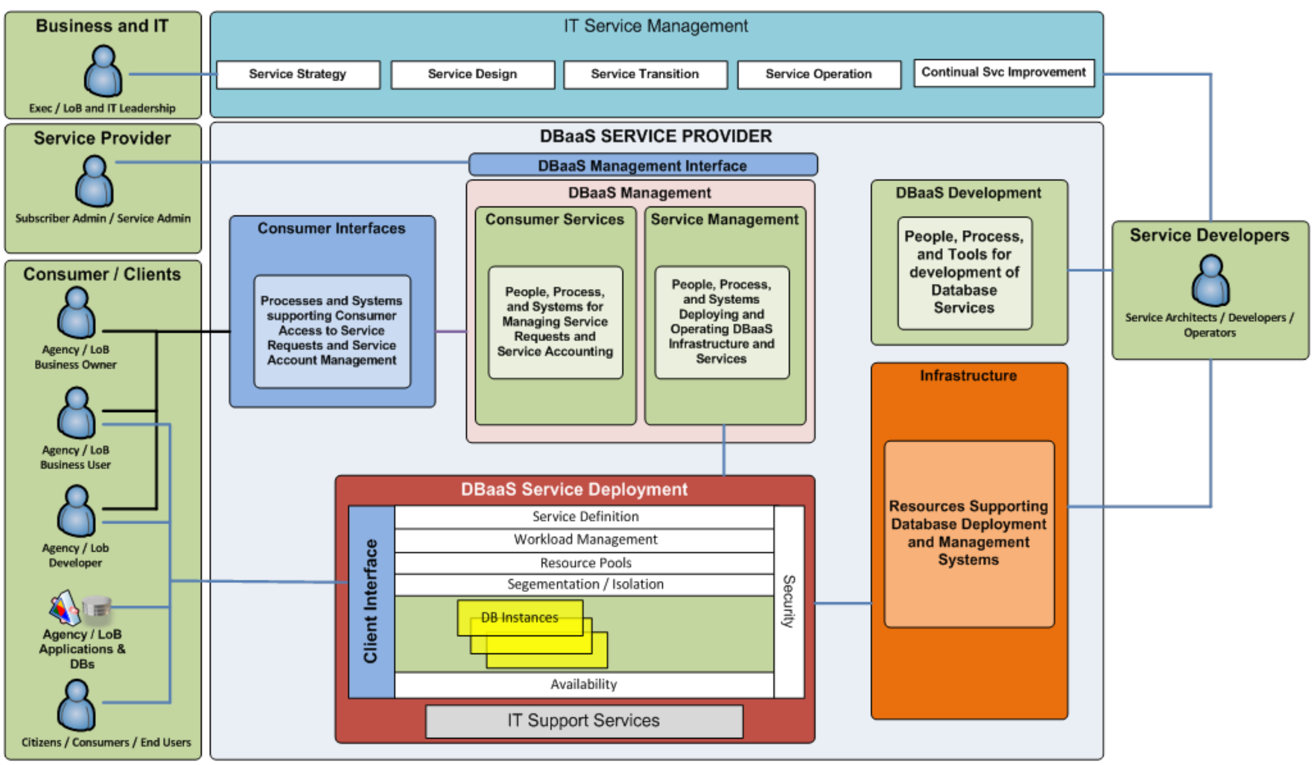

Figure 2: DBaaS Conceptual Model

DBaaS Development – The people, process and tools used to define the DBaaS service offerings, the design and implementation of the infrastructure required to support the DBaaS services offered, and the design and development of the systems and services required to deploy and administer the services within the DBaaS deployment architecture. DBaaS Development is responsible for defining the service offerings and management of the service catalog.

DBaaS Management – The people, process and systems supporting the organization's ability to request, manage, operate, and account for database services and their utilization. Within DBaaS Management there are two sets of sub-capabilities: Subscriber Services which supports the interactions between the subscriber and delivery of database services, and Service Management which implements database services and manages the resources and systems supporting their delivery.

DBaaS Service Deployment – The instantiated physical resources and their configuration in support of the DBaaS service offerings, and the interfaces required to manage the deployment, monitoring, and management of the services. This would include the configuration of servers, networking, and software to support the specific database deployment models and database instances within the shared resource pool.

Infrastructure – The physical resources required to support the systems and services supporting the management and deployment of the DBaaS architecture. This would include servers, networking, software and facilities.

IT Service Management – Is the framework for service definition, design and operational practices and policies. IT Service Management provides capabilities for defining and managing the services required by the organization, the design process, change management process definition, service operation policy structure, and process improvement framework.

Subscriber Interfaces – The methods, systems and procedures for interacting with the DBaaS management capabilities as well as the database service instances deployed for the specific subscriber.

DBaaS Management Capabilities and Process Overview

The Logical Architecture refines and elaborates on the capabilities required to support DBaaS deployments. There is no requirement that all capabilities must be implemented or automated before an organization may begin to reap the benefits of their DBaaS deployment. Typically, DBaaS deployments begin gradually, building from a well-defined foundation. The adherence to core architecture principles, and the capabilities and process definition, allow for the consistent and sustainable development of the DBaaS architecture using a phased approach. The capabilities, sub-capabilities and processes act as building blocks for the development of the management, operational, run-time services that will be developed supporting the DBaaS implementation.

The following table is a subset of the capabilities and processes that would typically be defined and implemented to support a DBaaS architecture. These capabilities are realized through application services and processes implemented by the physical architecture.

DATABASE AS A SERVICE CAPABILITY AND PROCESS MATRIX

Capability	Sub-Capability	Processes	Actors
Subscriber Services			
	Customer Relationship Management	Account Creation Define Account Roles Capture Billing Information	Subscriber – Provider Line of Business and Subscriber Services
	Order Management	Account Authorization Service Selection Service Request Approval Service Request Tracking	Subscriber – Provider Line of Business and Subscriber Services
Service Management			
	Operations Management	Service Monitoring Service SLA Analysis Perform Service Backups Restore Service Backups	Provider Subscriber Services and Service Management
	Provisioning and Automation	Process Service Request Deploy Service Offering Define Service Interface Manage Service State Change Service	Provider Subscriber Services and Service Management
	Service Repository	Create Service Offering Update Service Offering Change Service Offering	Provider Subscriber Services and Service Management
DBaaS Development			
	Product Management	Define Service Requirements Define Service Demand Define Service Offerings Manage Release Cycle	Line of Business, IT Management – DBaaS Development
	Service Management	Define Service Strategy Define and Collect KPIs Validate Service Strategy	Line of Business, IT Management, Service Development
	Product Development	Design Service Architecture Design Deployment Models Create Deployable Entities	Product Management and Product Development

DBaaS Physical Architecture

The physical architecture provides an understanding of the major technology components and their relationships in support of the DBaaS capabilities, processes and services. In the Logical Architecture, examples focused on DBaaS Management Capabilities. In this section, examples will focus on the fundamental DBaaS capabilities related to supporting the deployment of database instances. The deployment and management services must be implemented carefully to ensure that they are secure, reliable, scalable, and support all availability and service continuity requirements.

DBaaS Deployment Platform – Exadata Database Machine X4-2

Core to the delivery of DBaaS is the implementation of the database service deployment platform. There are a variety of technologies that can be deployed together to provide the foundation for the deployment platform.

One such set of technologies is the Exadata Database Machine X4-2 architecture. Oracle Exadata provides an exceptional deployment platform for DBaaS through its balanced system design and support for multiple deployment models. The ability to operate multiple workloads combined with flexibility in database deployment configurations allows differing service levels to be supported on the same platform. Further, optimizations in storage, storage-compute interconnects, and I/O resource management allows for the delivery of flexible tiered server and storage resource configurations.

Resource segmentation both at the compute and storage tiers are enhanced through the Exadata software. Resource allocation can be applied to define specific database service offerings through instance caging and IO Resource Management providing for predictable performance and isolation from resource contention when the databases are deployed on shared servers.

DBaaS Deployment Platform – Oracle SuperCluster Deployment Platform

Oracle SuperCluster shares many of the same characteristics and supports much of the same technology that made the Exadata X4-2 Database Machine a particularly strong candidate for hosting a private database cloud. As with Exadata, the SuperCluster platform is optimized through the use of the Exadata Database Software and the integration of the Oracle Exadata Storage Cells to support a highly consolidated, mixed workload, and extremely scalable platform for database services. The Oracle SuperCluster platform is supported by the SPARC operation system and its highly flexible set of server virtualization options.

The SPARC virtualization technologies add additional flexibility in the deployment and operations of the DBaaS resource pools. Through server virtualization, the service offerings that can be deployed on the SuperCluster platform can be expanded beyond a single version of the Oracle Database Software, or the SPARC operating system itself. Where organizations require support for multiple database versions, the ability to virtualize server instances allows for the deployment of cloud services which are fully isolated from one another with resource segmentation at the virtual server level, as opposed to the database software level typically applied to the Exadata X4-2 platforms.

Additionally, the use of server virtualization can expand the service catalog to support operating system kernel level isolation for those applications requiring this level of isolation due to security or regulatory requirements.

However, just as with the Oracle Exadata Platform, Oracle SuperCluster supports Oracle Database 12*c* Multitenant deployments for the highest level of resource efficiency. This can be applied as a physical deployment to the T5-8 server node, or through a virtualized server instance.

DBaaS Deployment Platform – Generic Platform

Consolidation of database infrastructure is both a desired objective of a move to the private cloud database, as well as an outcome of the deployment. Where the engineered systems provide examples of optimized performance in the private cloud, many organizations will also desire a lower cost private cloud infrastructure solution for systems of lesser strategic value. The ability to create services based upon different cost and performance objectives provides the enterprise with greater flexibility in their allocation of resource dollars. As such, creating different tiers of infrastructure – those of optimized performance and those of minimized costs provides the service developers the ability to develop a single private database cloud architecture that meets multiple objectives that lowers investment risk, while still meeting the requirements for systems of different workloads and strategic impact.

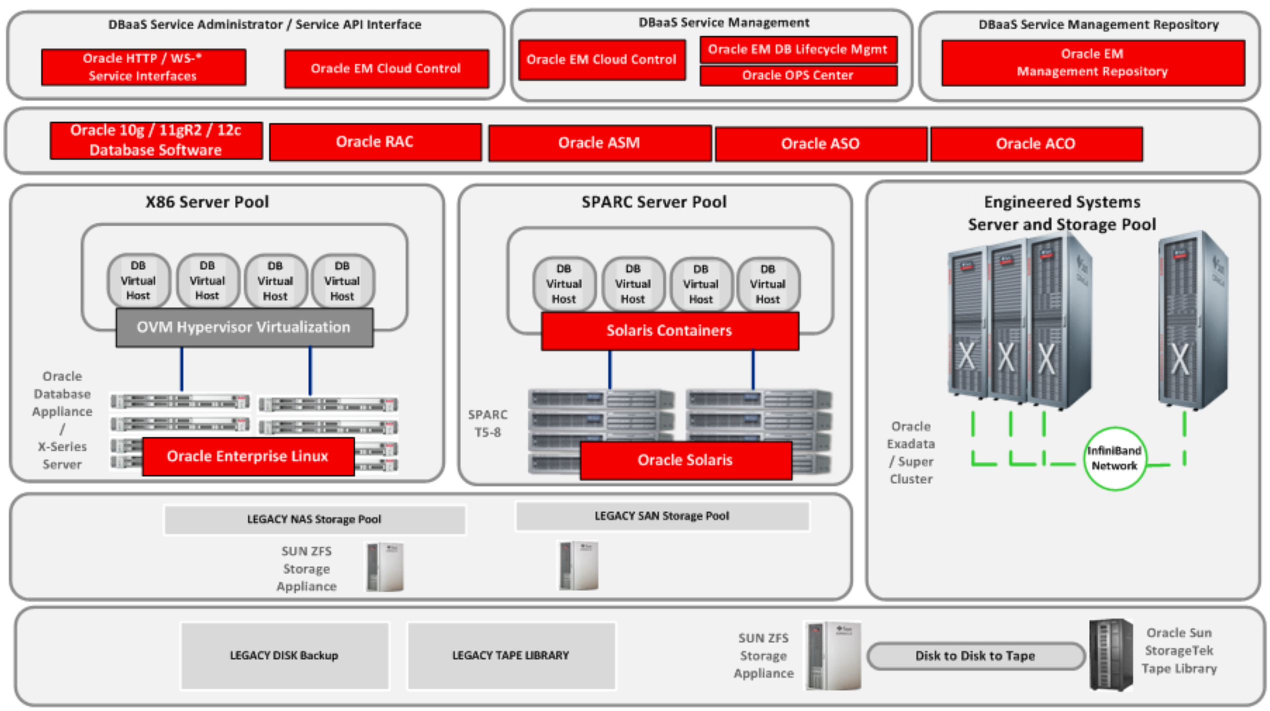

Figure 3: DBaaS Multi-Tier Infrastructure Model

The combination of consolidation and service differentiation allows for the inclusion of a tier of resources based upon the reuse of existing assets, such as traditional SPARC or Intel servers, and the acquisition of these individual servers to make up private cloud database resource pools. In order to gain the greatest operating and administrative efficiency, as well as support the simplicity of automated provisioning, these groups of resources are to be deployed as homogeneous logical groups. These groups, or resource pools – provides the foundation for the management of a large number of independent servers through a single management interface and control plane.

Once established, the resources, such as CPU and Memory, become the logical and physical target for the deployment of the private database cloud instances. Within the Oracle Private Database Cloud architecture, Oracle Enterprise Manager Cloud Control is this common management service, just as with the Exadata and Super Cluster deployments.

Oracle T-Series and X-Series server families provide for extreme cost performance in a rack based deployment package. The ability to incrementally adding compute resources to the pool allows for just in time growth as the private database cloud services expands. By logically grouping these servers of common resources and configuration, the cloud designer and operator is provided with a building block approach to pool development. Each server sharing the same capabilities and performance characteristics allows for addition of resources without concern for the deployment processes or packages. This standardization supports the goals of automation for self service and operating efficiency in administration.

In addition to the management capabilities, and the resource pools, the private cloud database will require the ability to control how the resources from the pool are allocated to the individual database service instances, and how they maintain isolation from other instances running within the pool. Oracle Virtual Machine for either Linux or SPARC based solutions allows for the virtualization of the individual servers within the resource pool. The OVM Logical Host or Guest Instance becomes a container for the database service to reside. The OVM Guest controls the resources assigned to each database instance, as well as maintaining resource and runtime segregation of each of the databases. As within the management of the resource pool, the management and provisioning of the OVM Guest is administered and operated through the Oracle Enterprise Management Cloud Control services.

DBaaS Service Management Framework – Oracle Enterprise Manager Cloud Control

DBaaS requires the ability to manage the environment and resources as well as the individual database service deployed and operated on behalf of the subscribers to the service. Oracle Enterprise Manager provides for the deployment of pre-defined database templates and profiles representing the provider's service offerings. This same centrally-managed service allows for the configuration, management, and monitoring of the resource pools and the service deployments utilizing them.   Oracle Enterprise Manager Cloud Control is the foundational technology for the integration and delivery of the Service Management capabilities described in the logical architecture.

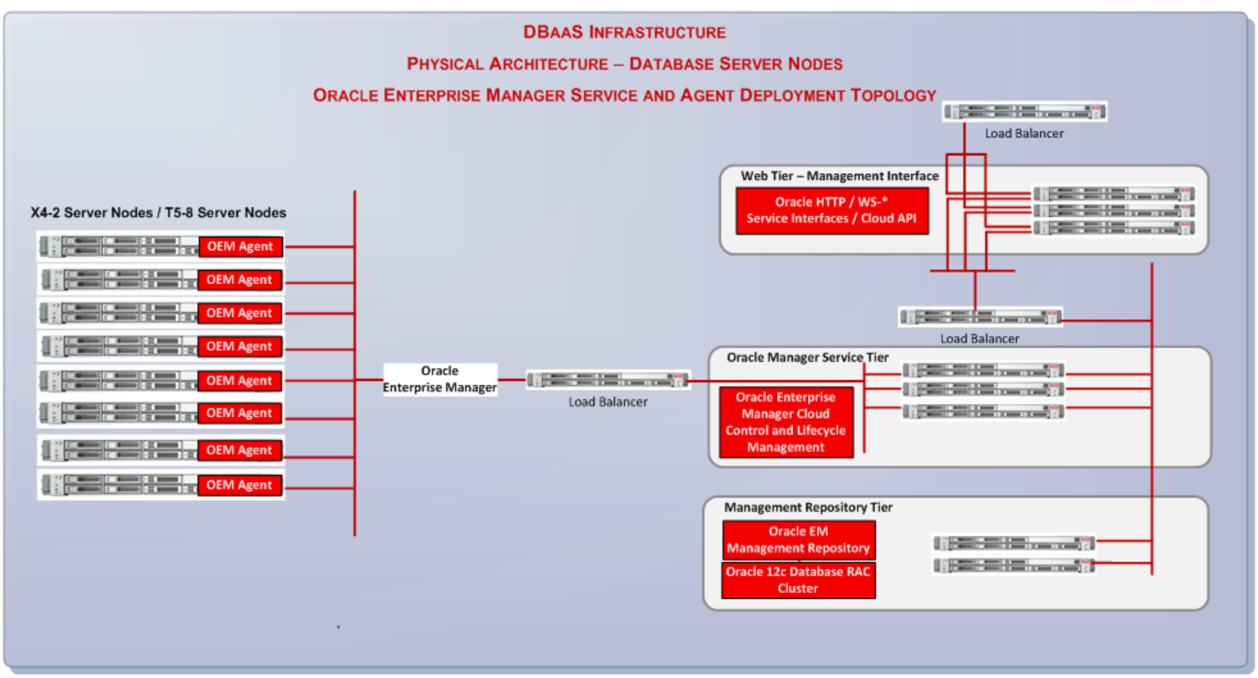

Figure 4: DBaaS Physical Architecture featuring Oracle Enterprise Manager Agents

While the X4-2 / T5-8 platforms and Oracle Enterprise Manager Cloud Control are specifically highlighted, it should be noted that there are a variety of other physical architecture technology components that will be leveraged by organizations building DBaaS architectures, including:

» Web 2.0 Technology – Provide multiple channels for subscriber interactions with management framework
» Business Workflow and Process Orchestration – Apply business process logic to subscriber and service management tasks and orchestration of management and delivery systems
» Security and Identity Management Services – Manage and control access to resources and services
» Content Management Database – Management, retention, and control over service offerings, the deployable entities and their versions
» Business Intelligence – Analysis and reporting of service and resource performance, as well as their adherence to service level and quality metrics
» Backup and Archival Storage - Facilities and resources for management and storage of provider and subscriber data
» Networking – Subscriber, management systems, inter-site communication; in addition to the Exadata InfiniBand fabric

DBaaS Multitenancy – Oracle Database 12*c*

With Oracle Database 12*c*, a multitenant database environment is now delivered at the database software level simplifying the creation, management, and operations of a consolidated pool of database resources.

Oracle Multitenant Database can be applied on a variety of platforms including the Oracle Exadata and Oracle SuperCluster engineered systems, as well as the Sun family of servers and non-Oracle platforms.

Oracle Multitenant supports a new architecture that allows for the deployment of pools of database deployments referred to as the Container Database (CDB) which acts as the operational and runtime home of subordinate Pluggable Database (PDB) instances. The CDB provides a common physical home for the operations of the individual database instances. Updates and management can be applied to the pool of databases greatly reducing the administrative tasks associated with running a highly consolidated database environment.

In addition, the CDB shares resources amongst the PDB instances deployed, reducing the duplication of common database tasks and processes reducing the overall resource utilization when running consolidated database instances on a single server.

Business Value	Virtual Machines	Many databases on one Cluster	Pluggable Databases
Implementation	Easy	Easy	Easy
Application Suitability	Some	All	All
Isolation	Highest	High	High
Availability	High	Highest	Highest
Scalability	Limited	Excellent	Excellent
Performance	Limited	Native	Native
Consolidation Density	Low	High	Highest
ROI	Low	High	Highest

Figure 5: Database Consolidation Approaches

The Container and Pluggable Database architecture allows for other key characteristics of Cloud computing, specifically that of elasticity. The ability to move the database instance from pool to pool via the container model, allows for the refinement of resource utilization as demands for database capacity, performance, or availability changes over time.

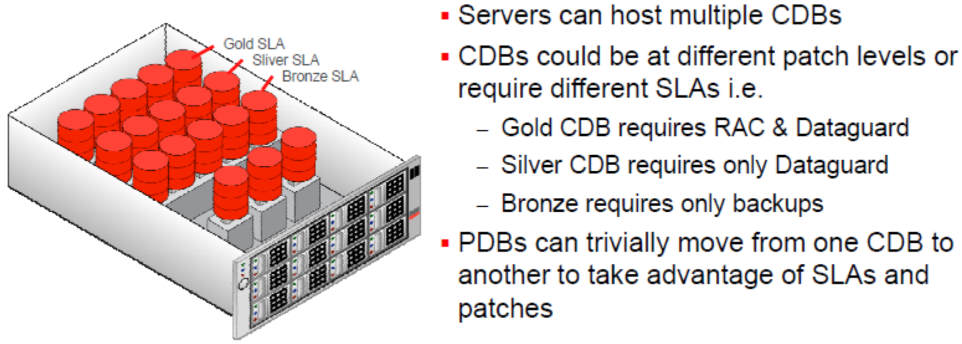

Figure 6: Service Tiers through Oracle Multitenant Database

With the inclusion of the Oracle Database 12*c* capabilities, the automation supporting self-service database provisioning is also enhanced. Database cloning is now simplified, as well as database configurations in support of multiple database service tiers. The database workload and performance tuning configuration can be applied at the Container Level, as databases are cloned from existing reference models and deployed to a specific container, the new database inherits the service levels associated with the target container. This greatly simplifies service management and reduces the time required to implement new database instances in support of expansions in capacity or new IT initiatives.

DBaaS Scalability

Scalability is a fundamental characteristic of DBaaS architectures given their support for self-service, elasticity and multitenancy. The Exadata Database Machine X4-2 and the Oracle Database Enterprise Edition 12c provide multiple planes of scalability in support of delivering database services, such as:

- Database Service Instance. Oracle RAC Clustering provides horizontal process scaling as service demands increase.
- Database Server Resources. Exadata Database Machine X4-2 provides server scaling through clustering of multiple Exadata Database Machine frames.
- Database Storage Resources. Exadata Database Machine storage pools can be expanded through the addition of Exadata Storage Expansion frames.
- Service Management Resources. Oracle Enterprise Manager provides a scalable deployment model where management nodes can be added as the number of resource components under management grows.
- Scaleable Resource Pooling: Oracle Database 12*c* provides a scalable multitenant architecture allowing for the deployment of shared database containers supporting pluggable database instances supporting a higher level of consolidation requiring less compute resources.

Oracle Enterprise Manager provides the management framework and capabilities to plan, monitor and administer DBaaS resource capacity and allocation across the various scalability planes. In conjunction with the Oracle Exadata Storage and Oracle Database software, resources can be applied to the DBaaS resource pool without disrupting existing deployed services.

DBaaS Availability

Not all services require the same level of availability which is why DBaaS service definitions include quality of service characteristics. Specifying what levels of availability are needed will enable the provisioning, monitoring and management processes to ensure that instances are deployed on the right platforms and in the correct configuration. The Exadata Database Machine X4-2 and the Oracle Exadata Software components support the ability to define various levels of availability within a standard set of technologies.

Service disruptions of both a planned and unplanned nature are compensated for through the combination of software, hardware, networking and deployment options. Technology such as Exadata, Oracle Real Application Clusters, Oracle RAC One Node and Oracle Grid Control provide the foundation for delivering these capabilities.

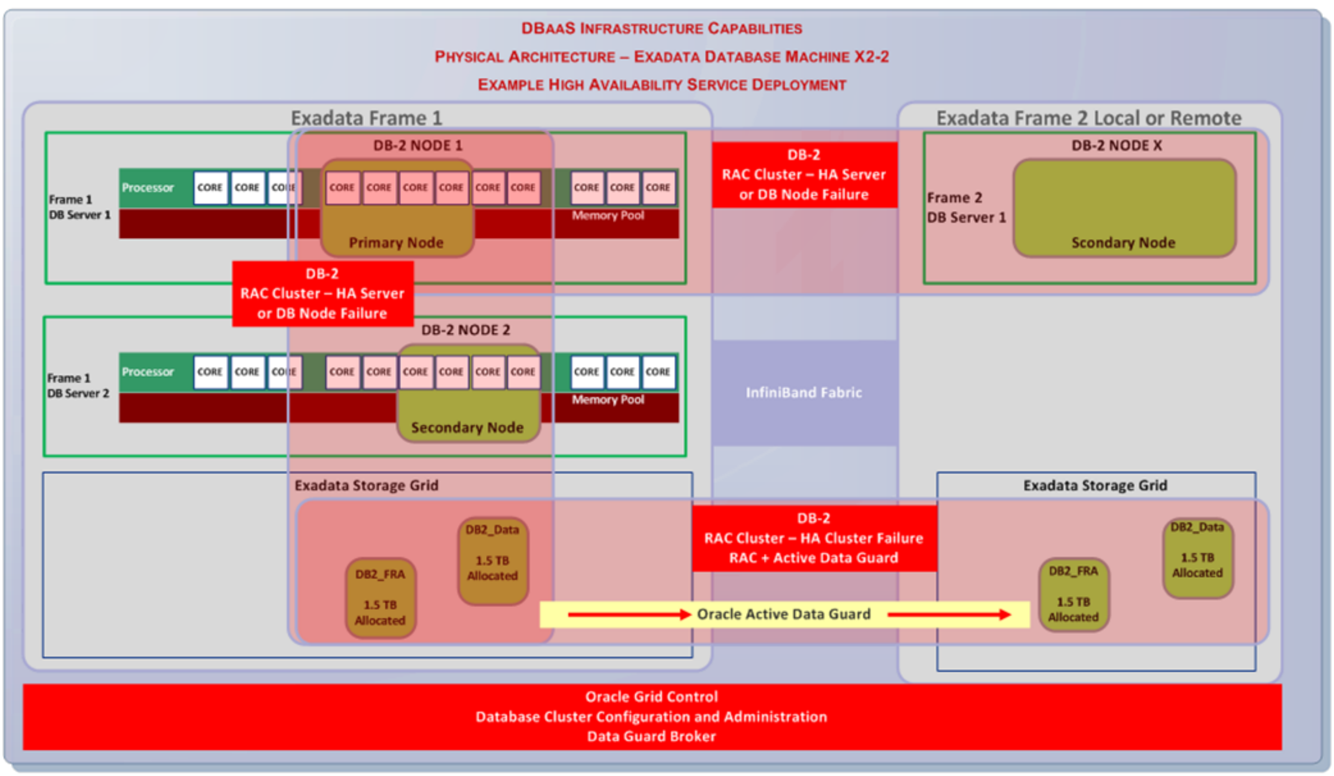

Figure 7: DBaaS Physical Architecture featuring Oracle Exadata in a high availability configuration

The delivery of high availability database services developed with the Exadata Database Machine and the Oracle Database12c software can be achieved through a variety of deployment options utilizing existing and long tested best practices. The following model and descriptions highlight some of the implementations and the scope of their support:

» Database Instance and Server High Availability (HA). Exadata and Oracle RAC Clusterware allow for the deployment of database instances with multiple processing instances spanning multiple servers within and across Exadata frames over a high-speed, low-latency InfiniBand network.

» Exadata Storage HA. Oracle Automatic Storage Management in conjunction with Exadata Storage Software provides storage cell level high-availability through multiple levels of disk redundancy allowing for the development of tiered storage based upon level and cost of availability.

» Exadata Cluster-wide HA. Oracle Exadata, Oracle Database, and Oracle Grid Control work in conjunction to provide the deployment of frame level service availability through the implementation of various stand-by configurations. The combination of Oracle RAC HA deployments utilizing Oracle Active Data Guard capabilities allows for the implementation of database instance, server, and storage high availability configurations supporting frame and site level failover capabilities.

» Service Management HA. Oracle Enterprise Manager is based upon a state-less management server architecture allowing for management services to be supported across multiple nodes. Utilizing Oracle RAC technology in support of the management repository allows for redundancy within the data tier, and through the use of Oracle Data Guard, the ability to replicate the management data across data center boundaries.

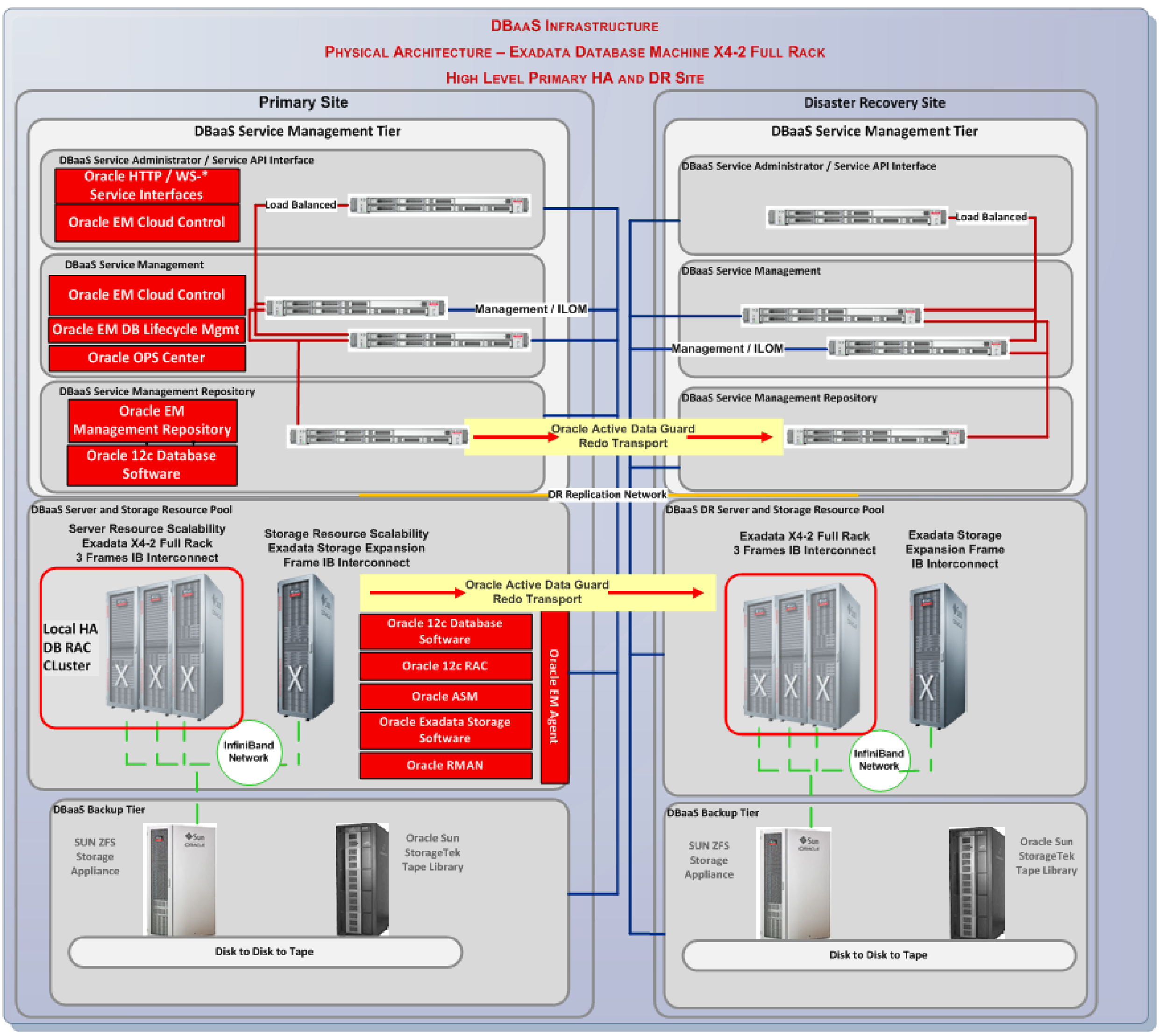

Figure 8: DBaaS Physical Architecture featuring Oracle Exadata in a high availability and disaster recovery configuration

The DBaaS Deployment Platform is of course only one of the sets of capabilities to be defined within the physical architecture. Any physical architecture must account for all of the technology and technology services supporting the operations, management, service delivery, service accounting and service development processes as well. Further, while scalability and availability were covered, it is important to understand that there are many other systemic qualities that must be considered including security, reliability, and recovery.

Conclusion

Database as a Service architectures offer organizations new and unique ways to offer, use and manage database services. The fundamental differences related to service-orientation and discrete consumer-provider roles challenge conventional models yet offer the potential for significant cost savings, improved service levels and greater leverage of information across the business. As discussed in this white paper, there are a variety of issues, considerations and choices that organizations must understand before embarking on a DBaaS project.

This white paper introduced you to Oracle DBaaS products, architecture, and the nature of Oracle's one-on-one architecture guidance services. To learn more about Oracle private cloud DBaaS products and service catalogs, visit Oracle.com and OTN. To learn about Oracle's public cloud database service, Oracle Database Cloud Service, visit cloud.oracle.com. To read an executive overview about Oracle private DBaaS, click here. To learn about Oracle Consulting cloud architecture services, click here. To delve deeper into the Oracle DBaaS reference architecture consisting of the artifacts, tools and samples, contact your local Oracle sales representative and ask to speak to Oracle's Enterprise Architects.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

blogs.oracle/enterprisearchitecture

facebook.com/OracleEA

twitter.com/oracleEAs

oracle.com/EA

Hardware and Software, Engineered to Work Together

December 2014
An Enterprise Architecture White Paper on the Oracle Private Database Cloud — Reference Architecture Overview
Author: Al Kiessel,
Contributing Author: Peter Heller