Introduction and Document Scope

This document is designed to a pre-read on the core differentiators for OCI as a public cloud when compared to the first three hyperscale clouds (AWS, Azure and GCP). It does not address the whole of Oracle cloud, including the breadth of services in the cloud. Gartner Group has a rather complete solution scorecard to compare the cloud-native services of public clouds where Oracle currently outscores GCP and is on par to Azure for cloud-native services. Additionally, this document doesn't cover the application services offered by Oracle that run on the Oracle Cloud Infrastructure, such as Fusion cloud, it's industry applications or Netsuite. The scope of this document is narrowly focused on what makes OCI different as a second-generation cloud and as an introduction to why OCI is material different in the approach to cloud.

The Early Cloud and it's approach to scale:

To understand the design goals and strategy behind Oracle Cloud Infrastructure as a cloud service, it is first important that we understand what is meant by "cloud computing" in its early form and the nature of the first three hyperscale clouds. By understanding how they are architected for scale as an application environment, as well as the limitations that this approach creates, we can better understand the impetus for the design of OCI.

What these hyperscale clouds had in common as a category was an architecture based on shared services connected by a shared network model with a hub and spoke topology, with router/switches at the top of the rack serving as a switch for traffic in the rack (north/south traffic) and as a router for traffic between racks (east/west traffic). Complimenting this compute and networking approach was the use of network attached storage and server attached block storage as a shared storage architecture. This 'shared everything' model meant that servers could be simple commodity servers and network and storage services could be radically simplified. This shared physical architecture was managed by incredibly intelligent software that could both manage the provisioning of new tenancies and services, but also manage the isolation of tenants through TCP/IP virtualization and rigid control plane management of services.

Amazon, Azure, and Google all had one core aspect in common – they already had an existing hyperscale architecture to support existing businesses and this architecture was each designed to support a new type of application that was built specifically to take advantage of this scale architecture, but also to work around its limitations. In the case of Amazon, it was initially shared compute services with the e-commerce architecture; in the case of Azure, it was Global Foundation Service, the compute/storage/network architecture behind Xbox Live, Bing, Office365 and Dynamics. With Google, GCP was built as a service layer on top of the Googleplex, the compute fabric that was designed to run YouTube, Search and Gmail. These hyperscale architectures were already operating with hyperscale services for a single customer – themselves. Making this a shared, public service was the non-trivial task of building the control plane and security to isolate and protect each customer and the services itself.

The early hyperscale clouds succeeded at an astounding level in being able to service several types of workloads that lent themselves to this architecture; departmental business applications could take advantage of new technology for developer operations and continuous integration/continuous delivery models that enabled applications to be updated with new logic as the business changed. Of course, the main use case for the initial hyperscale services were new companies built around scale web applications that could only effectively be run in a scale-capable cloud environment. Lastly, the rise of many SaaS providers was enabled by this new architecture where delivery of consistently updated application environments was a significant benefit to the customer and to the economics of the provider.

While the cloud-native approach to building applications enabled the ability to scale the application across many systems as needed, on demand, it has its limitations as clever software that could enable scale could not solve for the issues that arise with a completely shared architecture. Specifically, network collisions between tenants as they sought to move data between servers and between storage services would result in the need for stateless databases, where network latency and collisions would be mitigated by message broker models that would retry database actions until complete. Additionally, these new cloud services could horizontally scale (adding more systems of the same type) across worker nodes of an applications across many systems, but it struggled to manage performance when vertical scaling (adding more compute and memory resources to a system).

Unfortunately, while the new frontier of cloud-native applications was growing rapidly, the experience of moving enterprise applications to the cloud proved to be frustratingly difficult. These systems are often designed to perform at scale, leveraging cluster networking to pool compute, memory, and storage services to vertically scale the workload across distributed systems that needed low and predictable latency between systems.

Oracle's original approach to offering a cloud was identical to the approach used by Azure, AWS and GCP – shared virtual machines running on shared networks with container management software controlling applications that connect to stateless databases. This, as it turns out, was a terrible idea, as we combined the inefficiencies of hyperscale with the collisions and compromises of a shared architecture; more damning was that this approach struggled to run enterprise applications well, including Oracle's own application offerings. To succeed in the cloud marketplace, we needed to build a completely different type of cloud at a core infrastructure architecture level combined with the advances in cloud application development and services to address the unmet needs in the market – delivering a cloud that could natively and easily support not just cloud native applications, but enterprise applications at scale and delivering on a new approach to infrastructure in the cloud.

Engineering a completely different approach to cloud computing:

To approach this problem, we started with rethinking the approach to physical infrastructure and addressing several problems that come from a shared services architecture. First, we saw the shared control plane that ran on the same servers and the client tenancy workload as a problem – it opened the door for a rogue actor to use a security flaw in the operating system or application in a tenancy to be used to escalate privileges to gain access to the shared control plane software. More critically, this architecture of running the control plane on the same hardware as the server meant that you could never give the full performance of a server to a client, nor could you encrypt the system in a way that gave the client control of the keys for the entire system, not just the VM.

To address this, we created an off-box control plane DPU that controls the provisioning of services on OCI. This enables us to deliver full bare-metal servers to customers as an automated service, but also allows customers to own their own encryption keys for the entire server. OF course, this same off-box control plane administers VMs, containers, serverless instances and other PaaS services in a secure manner as well.

Network latency in the cloud is the core reason why companies are compelled to move all applications to stateless database services – the collisions between tenancy traffic leads to a problem termed the “noisy neighbor” problem, which prevents the use of scale, performant stateful database services. We developed a proprietary, hardware-based virtualization device that isolates and virtualized traffic at the data frame layer (Layer 2 in the OSI model). We combined this with a folded leaf and spine topology for our network switches, creating a Clos architecture non-blocking network as the standard networking model for our entire estate and creating the first public cloud that can guarantee network performance with a financially backed service level agreement. Additionally, this approach to network isolation allows other layer 3 technologies, like VMware’s NSX, to run natively on our network without modification.

Lastly, our approach to rack design was to address the ability for systems to vertically scale through cluster networking. We built remote direct memory access (RDMA) over converged ethernet backplanes into our Intel and AMD server systems, enabling clustering between servers as an automated service.

The combination of these three innovations results in a very different kind of core infrastructure for cloud services; this approach enables complete “like for like” migration of systems, resulting in cloud migrations that take days and weeks instead of months and quarters. It enables customers to be able to not just move performance tuned systems like ERP or supply chain automation to the cloud quickly, but it yields significant benefits for cloud-native services by eliminating network contention between container nodes and speeds up the execution of serverless applications accessing stateless systems. The ability for our cloud to deliver clustered, bare metal services in a completely automated way combined with Layer 2 network virtualization enables us to offer the only full fidelity VMWare service with NSX, vRealize and vSAN running natively with support for third party tools and hypervisor versions, enabling complete compatibility with on-premises VMware environments and quick migrations.

In short, this approach created a cloud architecture that both addresses the needs of modern, horizontally scaling cloud-native systems at scale as well as the performance needs of highly tuned and integrated stateful enterprise applications and environments, all without compromise.

Accelerating Cloud Native services with a low learning curve

With this new approach to delivering services, as noted above we saw significant acceleration of cloud-native workloads as we controlled for network collisions and node failures more effectively. However, we knew there was much more to be improved in the experience with cloud native systems, so we focused on building systems that could scale automatically, not just on a horizontal basis like all other container systems where nodes are created as function demand increases in the app, but also on a vertical basis, allowing vm’s in the system to be tuned to exactly the size of processor and memory needed. This “Flexible Services” approach appears in several key services for cloud native scaling:

1. In our compute service, for virtual machines, you can specify exactly the processor platform, processor count and memory per core allocation to your exact specifics. If you want a VM with 7 arm cores and 18GB of memory per core, you can get exactly that. Additionally, if you see you are hitting a bottleneck related to processor utilization or memory consumption, you can increase the size of the VM as needed and shrink it back down on the fly.

2. For load balancing services, we of course automate the creation of additional load balancer services as needed based on your scaling policy, but additionally you can alter the throughput limits from 10GB to 100GB to 400GB automatically based on budget policy and scaling metrics
3. Additionally, you can increase block storage performance on the fly through the console or an API call, allowing storage performance to scale without reprovisioning and migration work

Combined with our network efficiency and compute architectural advantages, you can scale cloud native services more gracefully and cost effectively on OCI.

Cloud native with stateful services

One of the other advantages of our low latency network is that customers begin to realize that they cloud-native work they have been doing has been both a blessing and a curse. The blessing is the move to microservices architecture and CI/CD orchestration and delivery models – this approach to development of application logic has enabled a massive increase in productivity of developers and created a much easier model for application maintenance and management. The curse, however, was the nature of cloud networks in a massively shared, hyperscale approach of the original three clouds meant that application data stores had to move to a stateless model. For complex enterprise applications, this meant rebuilding 100,000's of tables in modern ERP, customer experience and other applications. While possible, the level of effort to do this often left customers wondering if the entire effort was worth it, versus remaining on premises.

With OCI, it became possible to have a high speed, low latency network supporting a stateful database system. This meant that customers could move their entire application to the cloud as-is, then modify the application code to microservices architecture over time while maintaining a high performance, stateful database model.

Data Management in the cloud

With the low latency and high throughput network model, the ability to cluster systems and the automation of bare metal infrastructure with an off-box control plane results in the Oracle cloud being capable of running stateful and stateless databases equally well. Because of this innovation in infrastructure, Oracle offers its own database as several configurations, in addition to open source and other database services. For Oracle Database, we offer autonomous database as a managed service where the database is automatically patched, managed, and optimized for customers without intervention which can be run as a PaaS service, as a bare metal instance, a RAC cluster or on our Exadata cloud service platform that offers our database-optimized appliance as a service. Oracle Database is delivered as a multi-modal, converged architecture that allows the DB to process relational, transactional, JSON, spatial and graph data models equally well within the same service. This enables customers to migrate single use databases to a consolidated cloud service architecture, enabling the creation of data management structures at scale. We also offer our MySQL service with the Heatwave analytics engine, enabling web-scale transaction processing with MySQL with zero ETL work needed to build analytic structures in the secondary Heatwave engine within the same database structure.

Additionally, Oracle offers modern OSS cloud databases, including OpenSearch, NoSQL, Hadoop, and other cloud models, including the ability to run any OSS database in the customer tenancy. Our commitment to OSS is to maintain canonical compatibility with the OSS project and never change APIs in a way that would prevent application portability to any other OSS provider environment.

More than these core transactional and analytical data structures, Oracle offers a complete suite of enterprise data management tools, including Data Catalog for discovery and collation of data sources, Cloud SQL for object store queries, Golden Gate for data replications services and many others that are part of building a 'lakehouse' approach to data consolidation and management from many disparate, single use database models prevalent in cloud deployments to a single source of the truth, across multiple clouds and on-premises systems.

Building cloud native workloads on Oracle Cloud tooling

In designing our approach to a differentiated cloud with innovation in the hardware and software levels, we recognized the significant advances in tooling in the cloud – code development to deployment systems, monitoring and observability, infrastructure as code services, etc. have all rapidly advanced in maturity and adoption by the market. Rather than creating a new set of tools for customers to struggle with a new learning curve, we focused on adopting what the market is already using. As a result, most of the infrastructure and development processes that support CI/CD, security, observability, management, and other operating functions are built on the same tooling customers currently use in the 3 existing hyperscalers, such as ServiceNow, PagerDuty, Grafana, Terraform and others. Generally, the approach we have taken is to let the market speak on tool choice and to support what the market is using, reducing the learning curve for OCI significantly. We support Jenkins, GitLab and GitHub integrations into the cloud natively and we support an array of security, observability, and management tools from third party providers. We also support Landing Zone architectures and other security constructs to enable safe, scaled use of the cloud by the user base.

Delivering on Hyperscale compute and GPU services

Scaling GPU compute is a critical path for building modern AI and ML-supported solutions. Oracle designed our server racks to have a separate, converged ethernet service to enable remote direct memory access between workloads and the entire rack of GPUs and Memory, without going through the CPU bus. We paired this with a cross-block network service designed as a secondary Clos non-blocking network model to enable clustering of 10's of thousands of GPUs into a single service instance. Additionally, we built intelligence into this layer by providing applications with "placement hints" on where to effectively deploy workloads in the cluster and intelligent placement algorithms to reduce fiber physical distance.

With this architecture, OCI can offer scale GPU compute to serve significant AI and ML workloads, as well as other computation tasks that benefit from scale GPU access. We believe this cross-block, massive scale supercluster capability paired with a non-blocking network model and intelligent placement hints to be a significant differentiator for customers looking at scale AI services in the public cloud. As a testament to this, Oracle cloud has been noted as winning customers at well-funded AI startups at scale.

Building the modern cloud on modern infrastructure

With a differentiated architecture designed, we also thought about what this approach to architecture would mean for our physical infrastructure models as the presumption in the prior cloud native models was that you needed hyperscale in each location to make the delivery of cloud native services efficient. With this new architecture of a non-blocking network and off-box control planes, it enabled us to deliver cloud services in smaller physical units of work, shrinking the size of the required real estate for data centers and allowing us to place data center locations much closer to the metro areas where customers are centered, reducing the latency and transit costs for customer use as well as enabling multiple data centers with the same sovereign political framework, so that customers could effectively manage their DR/Availability all within the same region.

But more than this, we realized that we could deliver cloud services in different form factors and in a distributed model, leading to our Dedicated region, Compute and Exadata Cloud@Customer offers, as well as roving edge devices.

The cloud comes to you

With Dedicated region, we can build an entire OCI region with all services inside of a customer's datacenter, reducing latency to on-premises systems to effectively line-speed. This is not customer hardware with our software loaded on, like Azure Stack Hub or an Azure Arc-enabled server, or a limited set of cloud container/compute features for remote execution of cloud native apps, like Google Anthos or AWS outposts. Our approach is a full and complete Oracle Cloud Infrastructure region, delivered as a full availability zone with three fault domains, with every OCI service available for customers to use. It is delivered at the same cost as the public regions, billed on consumption only, with no cost for the hardware or management of the region. Oracle maintains responsibility for the cloud service and the customer commits to a 4 year spend of at least $2.5m per year in consumed cloud services and the customer can order physical hardware changes to the environment as needed to plan for additional capacity, greater GPU services or more arm services, as an example.

Near edge and far edge services

With our approach to the cloud being a differentiated architecture, we are also able to offer a set of near edge (regional data center or customer datacenter) services as well as far-edge (remote and disconnected use cases) for cloud services. With our Compute Cloud@Customer and Exadata Cloud@Customer offers, we deliver an OCI-managed and provisioned hardware appliance configured either for Exadata database use or customer defined compute structures. This allows distributed applications to move work closer to the systems needed, such as enabling low latency connections between on-premises applications and cloud or the ability to distribute work regionally to address latency across the application delivery footprint. These services are part of the customer tenancy and operate with the same security, network policies and user access as the cloud regions, just with distributed compute operations.

For far-edge use cases, we offer Roving Edge Devices, which offer 80 or 8 processor cores in a ruggedized appliance for use in remote, disconnected or extremely low latency use cases, such as operating in a remote oil field or at extreme low latency in a manufacturing floor.

Oracle Cloud in a multi-cloud world

One of the realities of being 15 years late to the cloud marketplace is while you enjoy significant 'fast-follower' advantages of being able to engineer a new approach based on the learnings in the market, you also have to contend with the realities of customer investments in their existing cloud providers. We recognize that the world has already become multi-cloud, with the need for customers to connect on-premises to cloud, SaaS to cloud and cloud to cloud in a low friction, low-cost way. By partnering with Equinix, we offer a bi-directional, 40gbps, <2ms connection between Azure and OCI tenancies in 12 data center locations, as well as Equinix data fabric services to other public and SaaS cloud environments. In the case of the Azure interconnect, customers enjoy network connectivity without any egress fees and with extremely low latency in either direction for a set, fixed monthly cost. This approach sets the standard for intercloud interoperability with customers extending their Azure policies and Identities to OCI in a seamless way and moving workloads to the most efficient and effective cloud for the customer.

Architected for Enterprise and Scale Use Cases at an Efficient Cost

It is important to also understand our design goal for OCI was to service enterprise and scale cloud native customers as our primary customer base. This might seem obvious, but it has non-obvious implications to the business model and service delivery model for the cloud that we examined closely and make some design choices to benefit enterprise customers specifically.

First, we recognized that the pricing models for the existing hyperscale cloud providers is designed for startups – it is relatively easy and inexpensive to spin up a virtual machine, connect to a container management service and a shared storage service and begin creating an application inexpensively. However, when you move from development to production, the need to a raft of other, expensive services becomes apparent – you need managed NAT gateways, intrusion prevention services, logging and observability services, key management services, etc. to really run in the application in a public facing, visible way. Often, the consumption of these services can significantly increase the cost of the tenancy, often doubling the running cost from the development side.

We observed that zero enterprise customers will ever deploy an application or workload in the public cloud without these security, observability, and other management services, so we included this tooling for free in the cost of the tenancy services, with customers only paying for the compute and storage costs of running these tools in most of these services. This materially reduces the complexity of cloud bills, but also materially reduces the absolute cost of services when comparing OCI to other clouds.

Second, another area of significant cost and complexity in cloud computing is network transit costs. In the early days of the cloud, it was feared that customers would use the flexibility of being able to deploy multiple VM services to test applications against the various versions of databases, client OS and other systems as part of their development work, but that the production systems would be deployed on premises. To discourage this, the three early hyperscale clouds all implemented an artificially high egress cost, charging a penalty for moving large datasets off their cloud back to on-premises services. This initial strategy has become a juggernaut of cost explosion for customers, making the economics of cloud often in doubt relative to private data center work. Additionally, this approach to artificially high egress costs and transit costs acts as a significant friction on multi-cloud integration, including SaaS to public cloud, public cloud to on-premises or public cloud to public cloud interconnections. In building OCI, we intentionally designed our pricing and services to reflect the reality of a multi-cloud world, so our network egress costs are typically 10% of the other cloud providers, representing a material savings in operating costs.

Additionally, the cost of moving data between regions in public clouds in prohibitively high relative to the actual costs and not reflective of the customer need to replication production systems and data between active DR nodes within geographically separated areas, typically only allowing 10GB of transfer before charging significant costs for this service. With Oracle cloud, we designed our network transfer costs to start at 10TB, not 10GB, of data transit.

Our core compute and storage costs are designed to be market leading in performance and price. Our Ampere arm compute service set an industry milestone with a $.01 per core/hour cost, but more than this, our block storage can scale from balanced to 300k IOPS performance with a simple console or API change while being 90% cheaper than performance storage on Azure and other clouds. Across almost all services, OCI represents a significant cost savings relative to competitive offers.

Pricing for Oracle Cloud services are also globally consistent, with the cost of compute in Ashburn being identical to the cost of compute in Singapore, Johannesburg, or Sao Paolo. This globally consistent pricing is possible due to the efficient engineering of our approach to data centers and automation, as the only costs that vary by region are the cost of labor and power, which are minimal as a cost in each region relative to competitors.

We also recognize the significant cost to our customers for Oracle Database License Support – for many of our customers, this is a significant pain point in our relationship and a blocker to adoption of Oracle cloud at scale. To address this, we offer our customers Support Rewards, a program that allocates $.25 for every $1 spent on OCI services to credits to offset the cost of their database support bill. In many cases, this results in the support cost going to zero as they scale up their Oracle Cloud consumption, which also enjoying significant direct savings on cloud expense relative to other clouds.

Bringing it all together and making it easy to own

Oracle works closely with our enterprise cloud partners to design, deliver, and operate cloud solutions for our customers. Our network of skilled partners offers the advisory, design, development, and operating services to bring all of Oracle, from apps to core cloud, into a single solution. Oracle recognizes that getting to the cloud is a strategic priority for our customers, yet the experience in the cloud has been one where there is little value until the application is completely ready to run as a stateless, cloud native application. Because of the nature of our bare metal services, network topology and performance, we can deliver savings to customers by migrating them to the cloud first, then modernize their infrastructure to a microservices model over time. This notion of immediate savings on on-premises labor and vendor costs combined with a naturally more cost-efficient cloud often means that customers can significantly lower their costs of operations while accelerating their digital transformation with Oracle and its partners much faster than the traditional, cloud-native early hyperscale clouds.