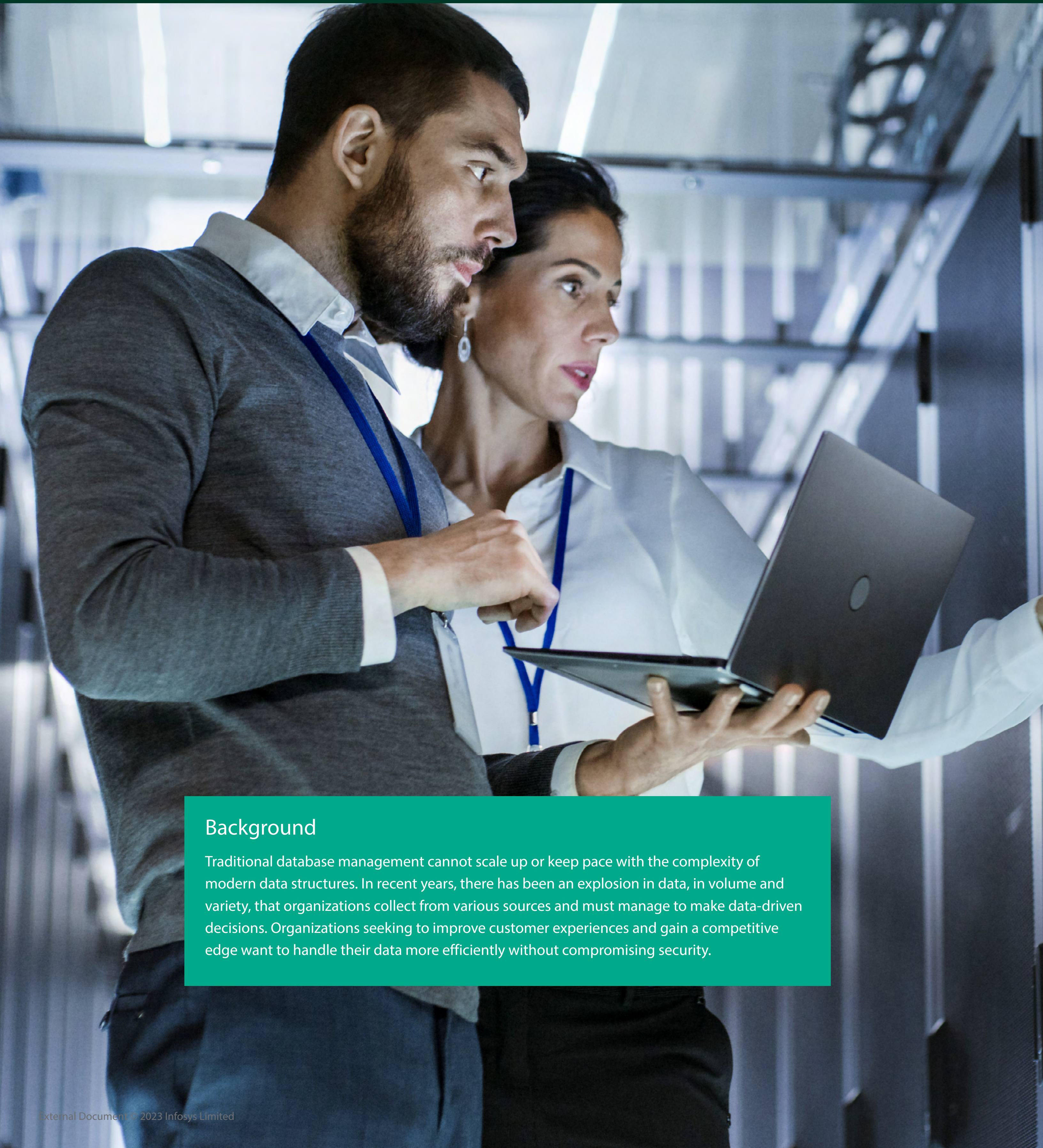
WHITE PAPER

BUILDING A MANAGED DATABASE SERVICE PLATFORM WITHIN A PRIVATE CLOUD ENVIRONMENT

Infosys®
Navigate your next

Introduction

This white paper explores the various nuances of leveraging managed database services within private cloud environments for optimizing data operations. It provides real-world use cases and insights with recommendations on the building blocks, design, architecture, and integrations required for setting up end-to-end managed database services.



Background

Traditional database management cannot scale up or keep pace with the complexity of modern data structures. In recent years, there has been an explosion in data, in volume and variety, that organizations collect from various sources and must manage to make data-driven decisions. Organizations seeking to improve customer experiences and gain a competitive edge want to handle their data more efficiently without compromising security.

Industry and Customer Needs

- Need for efficient database solutions:

  Organizations recognize the need for more efficient and scalable database solutions. The rationale for this white paper is the growing recognition that traditional database management methods are no longer adequate to meet modern data demands.

- Exploring private cloud as a viable solution:

  Private clouds are gaining popularity due to their ability to provide control, security, and customizability while maintaining the benefits of cloud computing. This white paper argues that private clouds offer an attractive environment for managed database services.

- Addressing key challenges:

  The white paper also highlights the intent to address key challenges faced by organizations such as scalability, security, and compliance. Managed database services within private clouds are seen as a solution to these challenges.

- Empowering decision-makers:

  By offering insights and best practices, the white paper seeks to empower decision-makers within organizations. It aims to provide the knowledge and guidance necessary to make informed decisions about adopting managed database services in a private cloud context.

Challenges in Database Management and Solutions Offered

Scalability and Performance

As data volumes expand, traditional databases struggle to scale seamlessly and maintain performance levels. Managed database services on private clouds can offer scalability by automatically provisioning resources as needed.

Security and Compliance

Unmanaged databases are vulnerable to security breaches and non-compliance with data protection regulations. With automated patch management, databases are patched with more recent security packs/hot fixes etc. Also, managed services provide features like zero trust security, access-controlled database, and audit logging to enhance security and meet compliance standards.

High Availability and Disaster Recovery

Downtime and data loss disrupt operations and affect customer trust. Along with multiple database copies availability across sites, automated failover for various failure scenarios ensures high availability of the database. Managed database services also offer integrated backup strategies to ensure data recovery with point-in-time recovery.

Enablers of Effective Data Management

Private Cloud

A private cloud is a computing services environment dedicated to an organization and catering to its specific data governance, regulation and compliance needs.

Private clouds provide the organization complete control over infrastructure, security, and data locality. It keeps data within an organization's network perimeter, thus reducing exposure to external threats. Security measures like firewalls, intrusion detection, and network segmentation are used to bolster data protection. It offers the flexibility to customize environments while ensuring data privacy, which allows tailoring of the infrastructure to specific database needs. Custom configurations enhance database performance and enable the adoption of required features and tools in line with business requirements.

With the range of features and benefits it offers, private cloud is not only a basic requirement but also lends itself to an organization looking to build a managed database service platform.

Managed Database Services

A managed database service is a cloud-based offering built on enterprise private cloud. It is designed to handle many of the operational aspects of database management, allowing organizations to offload routine database maintenance tasks to experts while providing a scalable and highly available database environment.

Managed database services alleviate the administrative burden by handling administrative tasks such as provisioning, patching, backup, and monitoring. These services free organizations from routine maintenance, allowing them to focus on application development. Moreover, these enhance security, compliance, and scalability while optimizing performance and resource utilization.

In addition, managed database services offer a host of benefits as they can:

- Support a variety of database engines including SQL databases like MSSQL, MySQL, PostgreSQL, Oracle, DB2 and NoSQL databases like MongoDB, Cassandra etc. This versatility caters to diverse application needs.
- Offer predictable pricing models, helping organizations manage their database-related costs effectively. Organizations pay for the resources they use, thus avoiding overprovisioning.
- Provide automation scripts and tools that automate routine tasks like database provisioning, scaling, and backup. This reduces manual intervention and minimizes the risk of human error.
- Enable self-service portals that allow consumers to request and manage database services themselves, promoting agility and reducing IT administrative overheads.

Use Cases and Benefits of Managed Database Services

A secure, scalable, and cost-efficient data platform for the automotive industry that helps:

- Scale with the growing number of connected vehicles without a corresponding increase in infrastructure costs.
- Harness the power of data to provide enhanced customer experience, and innovation while maintaining data security.

Scaling for seasonal peaks in e-commerce platforms by:

- Dynamically scaling resources during peak shopping seasons.

Ensuring data privacy and compliance in healthcare systems by:

- Enabling encryption, access controls, and audit trails to protect sensitive patient information.

High availability and disaster recovery capabilities for financial institutions in the form of:

- Data replication, failover mechanisms, and backup strategies to minimize service disruptions and data loss.

Benefits

It becomes both easier and convenient for the consumer to plan and meet their business and application requirements of data hosting, securing, redundancy, resiliency, performance, scalability without worrying about the intricacies of the underlying infrastructure setup, design, build, management, and maintenance of the database platform.

The organization can expect to benefit from:

- **Efficient resource utilization:** Optimized resource allocation reduces waste and saves costs.
- **Scalability:** Automated scaling ensures databases can handle varying workloads.
- **Reduced operational burden:** Automation and managed services alleviate the administrative workload.
- **Security and compliance:** Robust security measures and compliance features protect data.
- **Flexibility:** Consumers can choose the database type and scale resources as needed.
- **Data protection:** Automated backups and recovery mechanisms ensure data durability.
- **Pay as you use model:** Consumers just pay for the resources consumed. No capex is involved.

Key Considerations for Implementing Managed Database Services

Functional Considerations

The metrics listed below play a vital role in building and designing the managed database service platform, and therefore organizations should:

- Clearly define Availability SLA, Recovery Point Objective (RPO) and Recovery Time Objective (RTO) – essential objectives for data protection, KPIs and costing mechanisms.
- Assess factors pertaining to compatibility, required features, and vendor support.
- Evaluate the ability of the platform to meet specific business and consumer requirements.

Technical Considerations

The managed database service platform's design depends on the factors below to meet different service class requirements and for successful implementation:

- Type of datacenter and its availability
- Network bandwidth, latency, and vicinity of paired datacenters
- Underlying compute, storage and network design
- Licensing model of the software being used (OS, DB etc.)
- Available tools and integration interfaces
- Best-practice recommendations from vendors

Security, Compliance, and Regulatory Considerations

To secure the managed database service platform, the factors that must be taken into consideration are listed here:

- Data governance and compliance with various regulations (e.g., GDPR, HIPAA), which require rigorous policies, auditing, logging, and reporting
- Management of data retention policies and the secure disposal of data
- Encryption requirements and encryption key management solutions to safeguard data
- Environment and platform hardening benchmarks

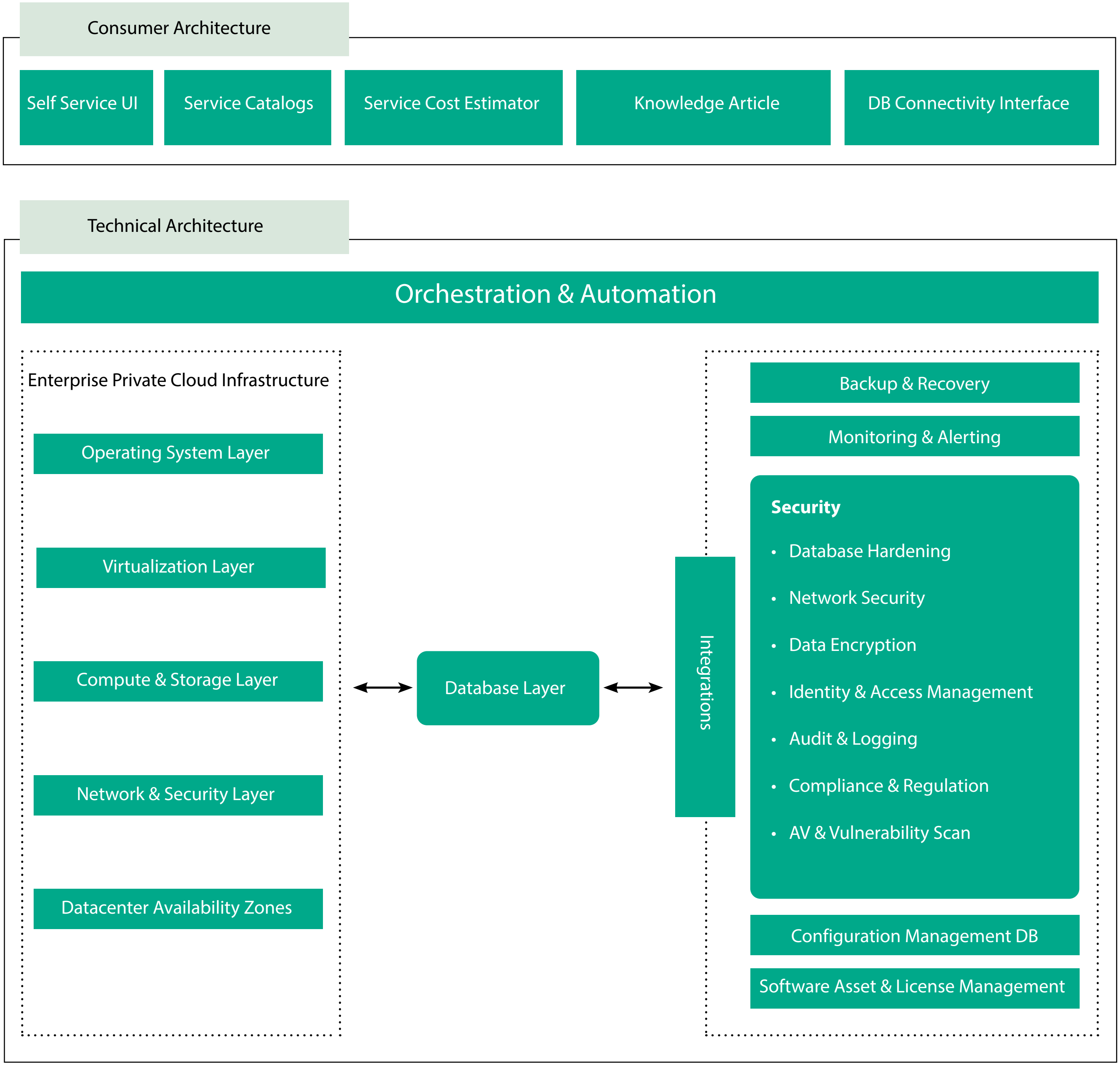
Architecture of the Managed Database Service Platform

Architecture Overview

The architecture of the managed database service platform can be broadly classified into:

- Consumer architecture which focuses on the consumer-centric approach and customer experience
- Technical architecture which outlines the required technical solution design
- Operational architecture which provides the support framework for the service

The diagram below depicts the high-level architecture along with important components of each category of the entire managed database service platform.



Operational Architecture

ITSM Integration

Support & Operations

- Database Support
- OS\ Server Support
- Infrastructure Support
- Network Support
- Tools & Integrations Suppor

- L1/L2/L3 support
- ITIL process adherence
- Knowledge management
- Vendor management
- Capacity management
- Patching & life cycle management

- New service inclusions
- Custom request & demand management
- Continuous improvement & optimization
- Feedback mechanism

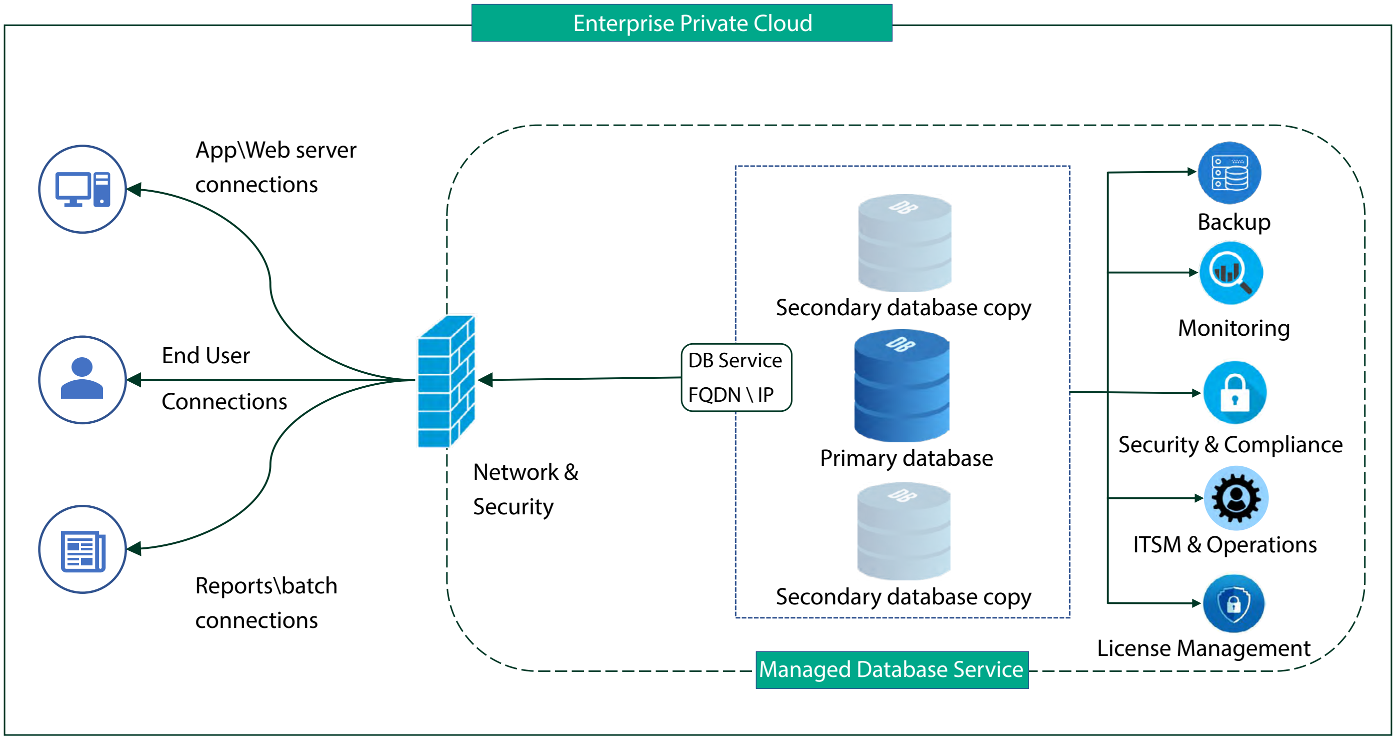
Consumer Architecture

In a consumer-centric architecture for a managed database platform, the focus is on providing a seamless and user-friendly experience to the consumers of the database services. This architecture should prioritize meeting the needs of various stakeholders including developers, database administrators, and business users. It must foster a user-centric approach by empowering consumers to manage their database services efficiently, enhancing productivity, scalability, and security.

Components of Consumer Architecture:

- **Self-service user interface:** A key principle of consumer architecture is self-service. Consumers can perform routine tasks like database creation, data migration, and scaling without relying heavily on manual intervention from administrators. The architecture begins with a user-friendly and intuitive user interface. This could be a web-based portal or a command-line interface (CLI) that allows consumers to interact with the managed database platform.
- **Service catalog:** The UI provides access to a service catalog, which lists the various managed database services available. Consumers can choose the type of database service they need based on their requirements such as SQL or NoSQL databases, Availability SLA, required configuration, access levels etc. Sample service catalog is provided in the following table.
- **Service cost estimator:** A comprehensive cost list for the configurations selected by the consumer must be displayed so that consumers are aware of the kind of services for which they are charged.
- **Knowledge based articles:** Technical and process write ups must be provided so that consumers know how to consume the service, inclusions, and exclusions in the service, connect to the service, and order day-2 services. Knowledge articles with extensive FAQs should give complete information to help consumers understand the service and leverage self-service options with utmost clarity.
- **User connectivity interface:** There needs to be an interface or a tool for consumers to connect to the database. Consumers in need of connecting to the database directly to query or perform any other data operations inside the database should be able to do so. A graphical user interface and/or a command line utility should be able to meet this requirement. Vendor-provided tools like SSMS, MongoDB Compass, pgAdmin, Db2 CLI tool etc., can also be hosted/integrated to the user connectivity interface. The architecture should also offer APIs that allow consumers to programmatically interact with the managed database services. This enables seamless integration with existing applications and workflows.

	Service Class - Platinum | Gold | Silver | Bronze
Deployment & Server Type	Standalone VM | Multi node VM within site | Multi node VMs across sites
Operating System	Windows | SUSE Linux | RHEL | CentOS | Ubuntu
OS Version	Windows Server 2022 | SUSE 15 | RHEL 9
Database Product	MSSQL Server | Oracle | IDM DB2 | PostgreSQL | MongoDB | MySQL
Database Version	SQL Server 2022 | Oracle 19c | DB2 11.5 | Postgres 15 | MongoDB 6
Database Edition	Enterprise | Standard | Community | ASE
Database Collation	Latin1_CP1_CI_AS | UTF 8 |
Database Access Rights	Owner, Writer, Reader
T-Shirt Sizes	XS, S, M, L, XL, XXL, XXXL
Database Backup	Full | Log | Incremental | Differential
Database Encryption	TDE | Transport encryption
Availability SLA	99.99% | 99% | 98% | 95%
RPO	Min Level | Expected Level (in mins/hours)
RTO	Min Level | Expected Level (in mins/hours)
Support Hours	24*7 | 24*5 | 16*5 | 8*5



Consumer View of the Managed Database Service

Technical Architecture

Once a required database service is selected by consumer, the respective database along with requested configuration must be provisioned. The provisioning of the selected database service including resource allocation, configuration setup and deployment, security configuration, integration with backup, monitoring, and several other interconnects must be done automatically.

This section talks about the technical architecture and outlines how the various layers, and its components interact to provide efficient and reliable database services.

Components of the Technical Architecture

Orchestrator and Automation

- The architecture begins with the orchestration layer which ensures seamless coordination of various components for building the database platform.
- The orchestrator coordinates with the user interface, infrastructure layer, and all the interconnecting tools for provisioning, scaling, integrating, deploying, and management of database services.

The end-to-end provisioning workflow must be automated. Automation streamlines all the tasks thus ensuring standardized environment, faster provisioning, and deployment. This reduces manual intervention and minimizes the risk of human error.

Private Cloud Infrastructure

The enterprise private cloud infrastructure including datacenter availability zones, physical servers, storage, and network resources form the base infrastructure layer. This layer should enable efficient resource utilization, isolation of workloads, securing the service perimeter and communication channel between different layers and interconnects.

The network architecture, including subnets, and firewall rules, must be designed so as to isolate and secure the database environment. Micro segmentation enables zero trust security architecture.

The network architecture that facilitates communication between clients, application servers, and the database server includes protocols (e.g., TCP/IP), load balancers, firewalls, and other networking components. The Compute layer provides computing resources like CPU and memory. Data storage must be designed to include the primary data storage and any additional storage for backups, replicas, or caching. The architecture must consider the type of storage (e.g., SSD, HDD) and its scalability to accommodate growing data volumes.

The Virtualization layer built on top of base infrastructure layer abstracts the underlying hardware, allowing multiple virtual machines (VMs) to run on the same physical hardware. The OS layer forms the basis for installing the database system. It is necessary to ensure the right configurations that support database binaries, creating clusters, and hosting the database system. It's also important for the architecture to support easy scaling based on demand, allowing for rapid scaling of resources to accommodate varying workloads.

Additionally, the private cloud infrastructure can be divided into resource pools, each containing compute, storage, and network resources. These resource pools when allocated to different managed database services based on their requirements should help achieve faster provisioning, easy maintenance and management, and on-demand scaling and performance benefits.

Database Layer

This is the core of the architecture providing different types of databases like SQL and NoSQL. It involves the creation of a

structured framework that governs the storage, organization, retrieval, and management of data within a database system. This process ensures that the database efficiently serves the needs of the intended applications while adhering to essential principles of data integrity, security, and performance.

Database architecture pertains to the overall structure, components, and arrangement of the database system, encompassing hardware, software, protocols, and the interactions between them.

Implementing database replication for fault tolerance and data redundancy will help ensure high availability and data durability. Replication mechanisms like master-slave or multi-master replication with synchronous or asynchronous replication of data to standby nodes in different zones or regions can be considered.

Here are some additional considerations to be made:

- Utilize a multi availability zone deployment approach to replicate the database across multiple availability zones within a region, ensuring high availability and resilience to zone failures.
- Build high availability and disaster recovery (HADR) at a database level or database instance level in accordance with the contractual agreements to achieve the Availability SLAs, RPO & RTO defined for each service class.
- Ensure the HADR architecture supports automatic failovers for various failure scenarios without any manual intervention.
- Ensure operating system, database instance and database level configurations are standardized and in accordance with database security hardening guidelines and considering vendor best practice recommendations for better performance and stability.
- Consider tools and processes for database optimization including query optimization, performance tuning, data archiving, log recycle, index and statistics management. This ensures optimal database performance for various workloads.
- Build interfaces and APIs that allow seamless integration with applications and services. This includes SDKs, connectors, and protocols that applications use to interact with the managed database.

Interconnects and Integrations

Backup and Recovery

The architecture includes integrating backup solutions with the managed database service and underlying infrastructure. Automated backup and mechanisms for point-in-time recovery are crucial to ensure data durability and availability.

A regular backup schedule must be configured (full database backups and transaction/archive log backups) with options for speedy recovery in case of failures or data corruption to meet RPO and RTO requirements. VM snapshot and disk backups must be considered for a comprehensive backup strategy. Moreover, backups must be encrypted and securely stored to meet data security guidelines.

Monitoring and Alerting

A crucial aspect of the architecture is real-time monitoring. A comprehensive monitoring system that tracks system failures, performance, resource usage, and overall health of the managed database is essential. It should provide real-time insights and trigger alerts for potential issues or breaches of defined thresholds. This would entail the integration of monitoring tools to continuously collect, track, and alert systems to notify of potential issues and breaches of defined thresholds.

Database Security

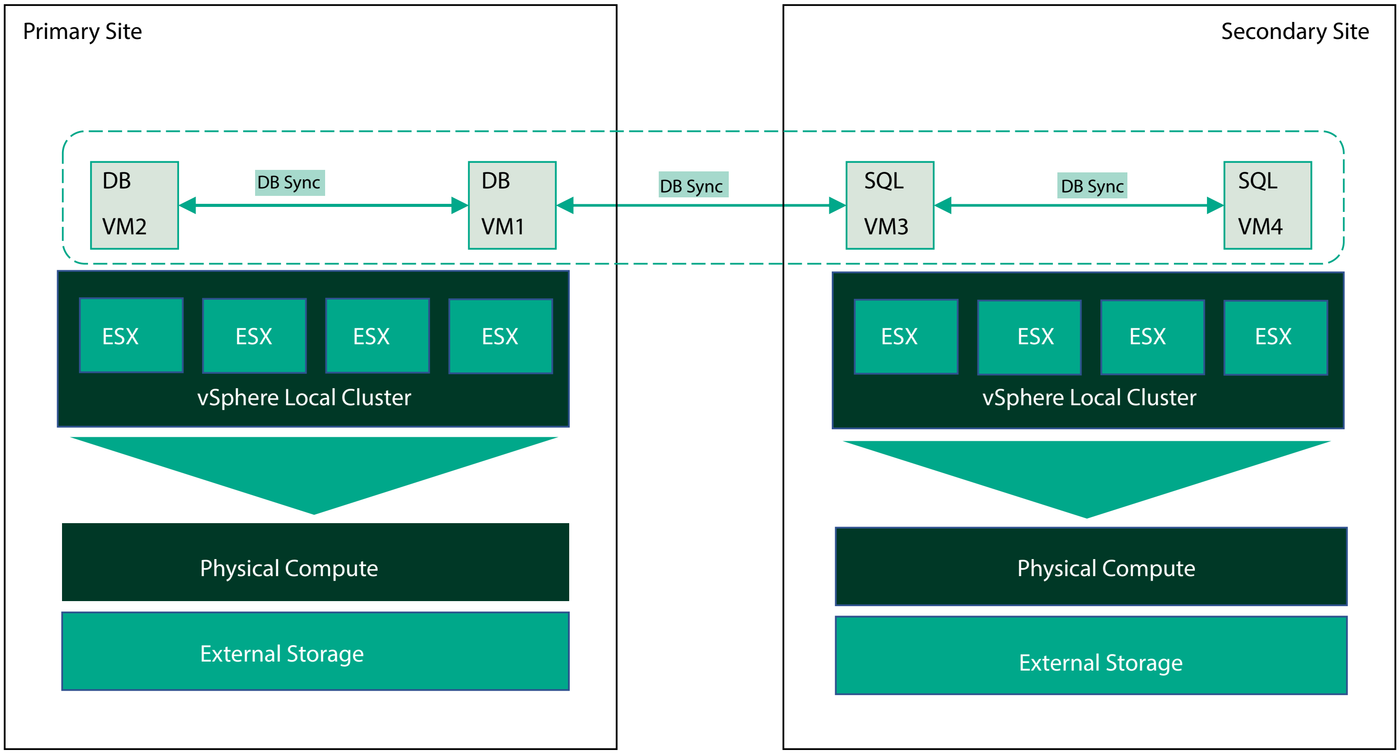
Ensuring robust security measures is paramount for protecting sensitive data. Aside from the regular security patches and updates provided by the database service provider to address known vulnerabilities and enhance the security posture of the managed database, here is a structured approach to database security within a managed database service.

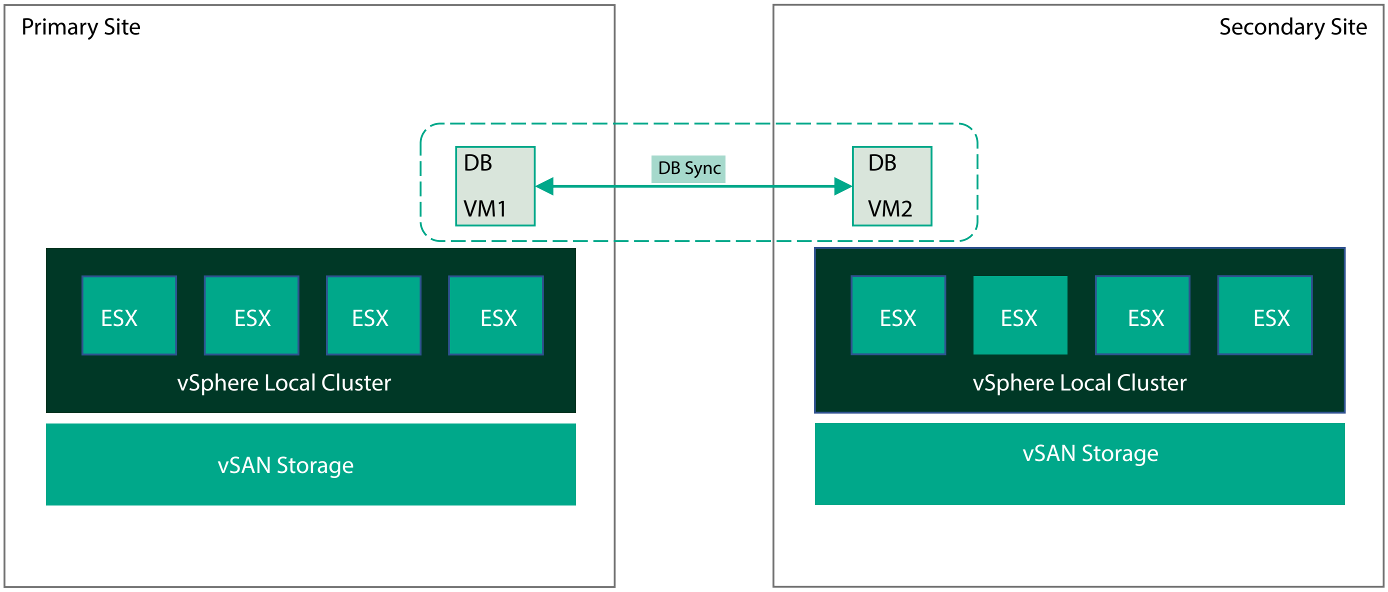
- Database Hardening – Adherence to industry standards like CIS benchmark guidelines will help harden the database environment.
- Network Security – Using virtual private network configurations helps isolate the database from unauthorized access. Firewalls and network security groups help control inbound and outbound traffic.
- Data Encryption – Implementation of encryption mechanisms for data at rest and data in transit helps safeguard sensitive information. The use of industry-standard encryption algorithms and key management practices is recommended.
- Identity and Access Management – This involves the implementation of stringent access control policies, limiting access to authorized users only. Using a role-based access control (RBAC) mechanism to assign appropriate permissions based on roles and responsibilities is a recommended practice. To ensure strong authentication, mechanisms such as multi-factor authentication (MFA) and enforcing strong password policies will help. Additionally, implementing authorization controls will restrict access to specific database objects based on user roles.

- Compliance and Regulations – Adherence to industry-specific compliance standards (e.g., HIPAA, GDPR) is achieved by configuring the database to meet compliance requirements. Conducting regular audits to ensure ongoing compliance is recommended here.
- Auditing and Logging – Auditing and logging features must be made available to track and monitor database access, modifications, and potential security breaches. In additional, regular reviews of logs for any suspicious activities and security violations will be crucial.
- AV & Vulnerability Scan – The majority of database vendors recommend excluding database files from antivirus scan – hence, implementing the AV exclusion policy for DB file extensions and binaries makes sense. Database and underlying systems should be regularly scanned for vulnerabilities to make sure that the necessary hardening is implemented properly. Further, security incident and event management must be implemented to identify security issues.
- Configuration Management DB (CMDB) – A CMDB tool must be integrated with the database service to maintain a comprehensive inventory of the complete infrastructure of the platform.
- Software Asset and License Management – A software asset management (SAM) tool must be integrated which deals with the management and maintenance of all IT assets. SAM is mainly helpful in maintaining and managing software licenses.

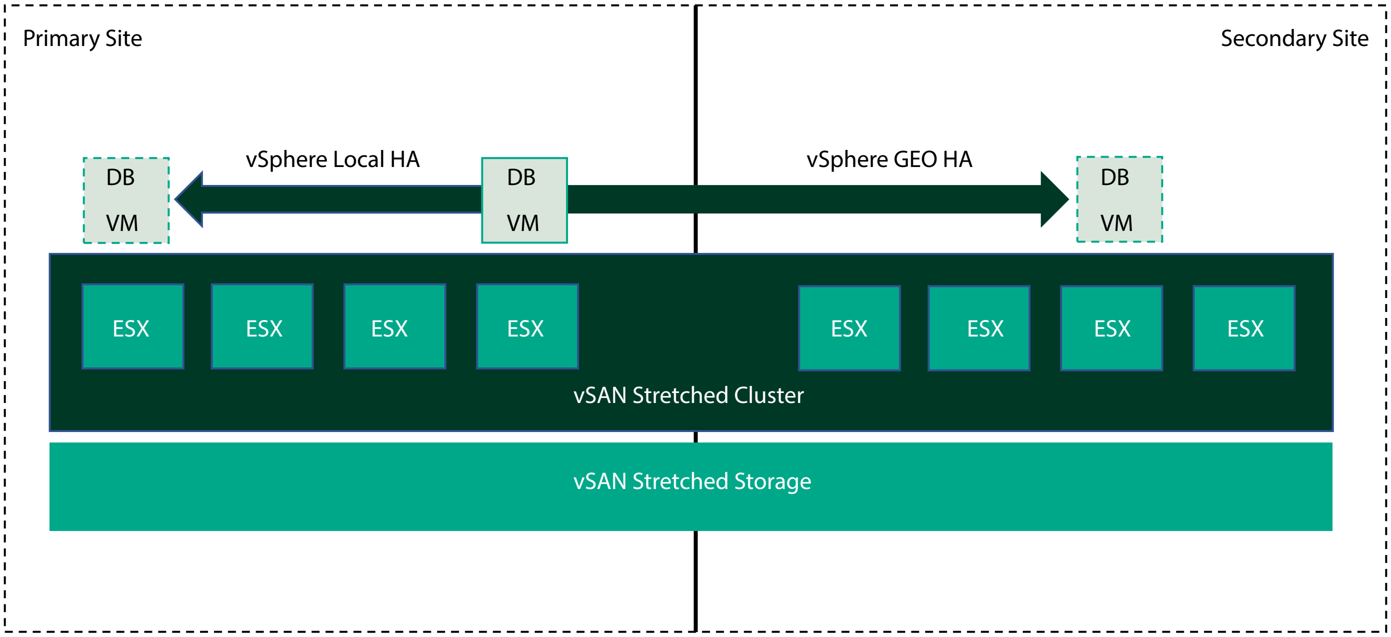
Sample Architectures

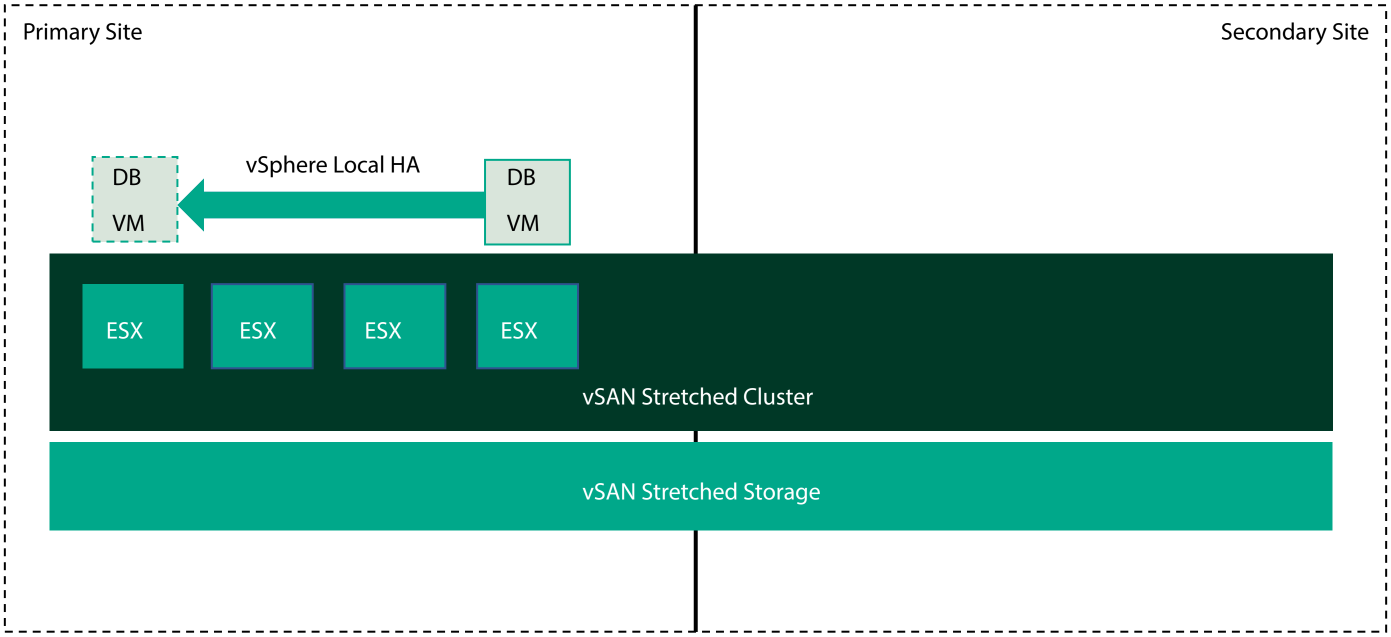
The architecture illustrated below is based on multi-site and multi-node deployment and database level replication/sync to achieve very high DB level SLAs.





Architectures illustrated below are based on storage level replication and leveraging vSphere HA for VM level SLAs.





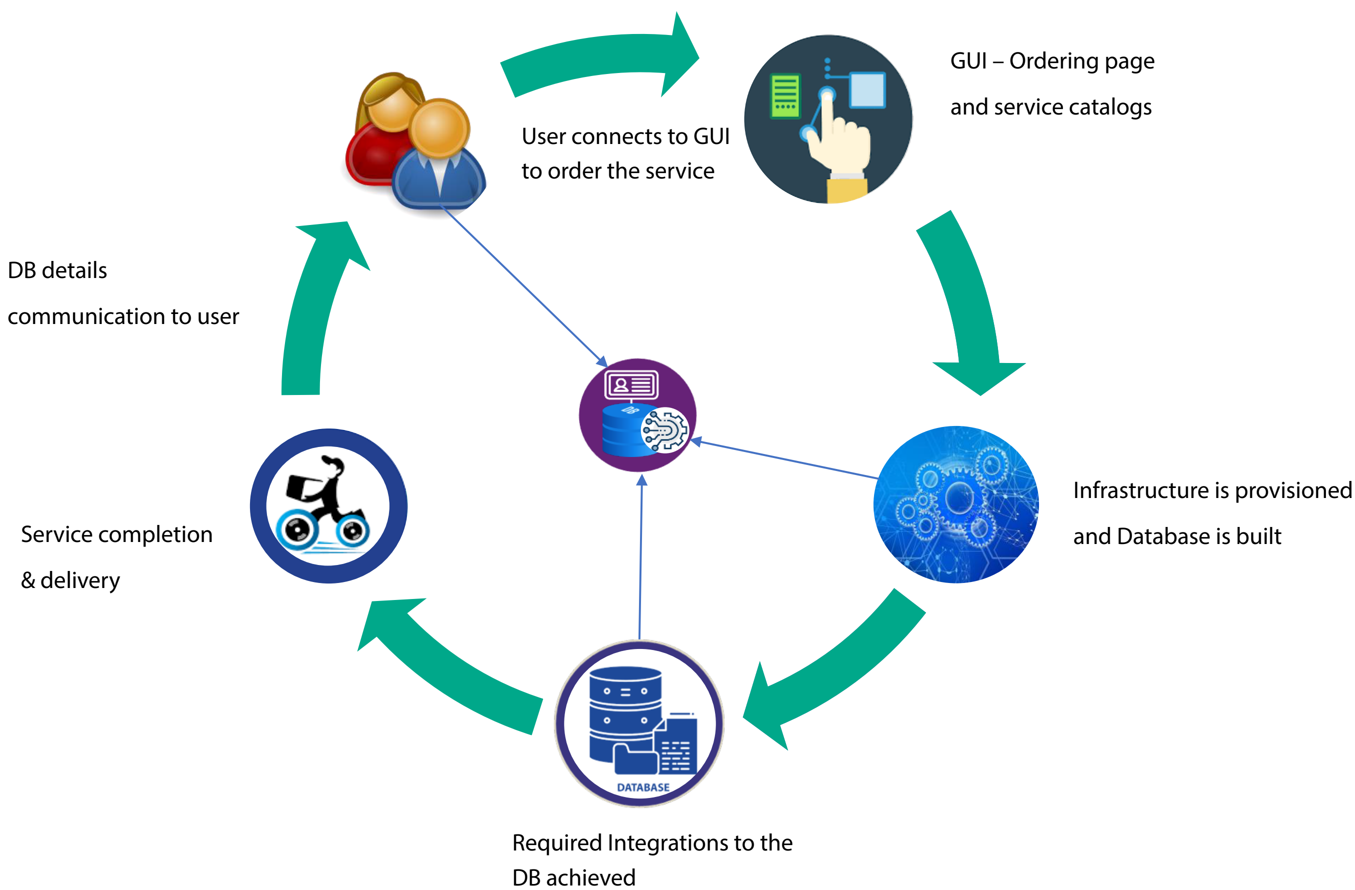
Operational Architecture

The operational architecture for a managed database platform outlines the way in which the various components interact to provide efficient and reliable database services. Integrating the managed database platform with the ITSM framework is crucial to ensure uptime, provide continuous service support, service improvement, as well as manage and maintain the database platform. The important aspects of regular support and operations to ensure zero downtime and seamless service availability are as follows:

- End-to-end service support and operations at each layer must be offered to cater to day 2 activities along with regular maintenance and management.
- Clear roles, responsibilities, and communication channels are essential. A clearly defined process must be in place to ensure seamless communication between all the teams involved. Collaboration between the IT teams, database administrators, customer and business stakeholders improves the customer experience.

- A well-defined patching and lifecycle management process must be defined to offer up-to-date and secure database environment.
- Automation and using scripts for regular maintenance and administration activities reduce the likelihood of human error.
- Capacity management helps understand the limitation and capabilities thus helping create an effective budget.
- Continuous improvement and optimization can reduce bugs, which helps build an error-free environment. It enhances the customer experience and reduces the probability of incidents occurring again. It can also reduce the cost, thereby benefitting the customer.
- Knowledge management ensures that proper documentation of all the activities is available. Regularly updating the SOPs, process documents, and knowledge-based articles increases effectiveness and reduces redundancies in effort.
- Implementing a feedback mechanism is another key aspect which ensures closing of the loop by directly establishing a communication channel with customers. Feedback from customers can enable insights and lead to improving the service.
- Finally, a process for demand management will help serve customized requests by the customer.

Service Workflow



Future Trends and Innovations

AI-driven Database Management and Maintenance

AI-driven tools optimize performance, predict failures, and automate routine tasks. In the future, these can prove particularly significant in:

- Anomaly detection and self-healing capabilities
- Predictive auto scaling
- Enhanced performance diagnostics and predictive maintenance
- Real-time analytics of monitoring and intelligent incident analysis to suggest corrective future actions
- Answering consumer queries and service requests via chat-bots

Multi-Cloud and Hybrid Strategies

Advances in this area are expected to be in the form of:

- Integration with Public cloud databases
- Hyperscaler tools being used for database management
- A single window for managing multi-cloud databases
- Easy and seamless data movement between cloud databases

Container Databases to Support Agile and Microservices Architectures

Containerizing databases abstract infrastructure resources from applications and is expected to result in efficient resource utilization and cost savings.

Container databases can run smoothly across multi cloud database platforms and can integrate with third party HADR solutions. They are best suited for micro service architecture which provides resource isolation for application multi-tenancy and thus provide better performance.

Conclusion

Building a managed database service platform requires a meticulous approach to designing the architecture at different layers by considering various facets such as business requirements, security, performance, scalability, and integration. By integrating the database service platform with comprehensive security architecture, organizations can ensure continuous availability of data, protect against security threats, and maintain regulatory compliance. These measures are critical for any database handling sensitive or critical information. After all, the larger goal is to create a secure, flexible, resilient, and efficient database solution that aligns with the organization's objectives and technical needs and helps consumers to cut the overheads associated with designing, building, and managing the database platform.

Authors

Siddalingaswamy D

Principal Consultant - Infrastructure Management, Cloud, Infrastructure & Security Services, Infosys

Siddalingaswamy has 16+ years of experience in IT infrastructure services as a database architect and administrator. In his current role as the Principal Architect for database technologies, he is responsible for architecting, designing, and automating managed database service platform to help transform the database landscape of a global automotive major.

Soma Sekhar Pamidi

AVP - Senior Prinicipal Technology Architect, Cloud, Infrastructure & Security Services, Infosys

Soma has 25+ years of experience in Performance Engineering and Hybrid Cloud Transformations. In his current role as CTO of a large datacenter transformation program for a global automotive major, he is responsible for architecting, designing, and delivering services to business users consistently on hybrid cloud landscape that spread across private cloud and major hyperscaler platforms.