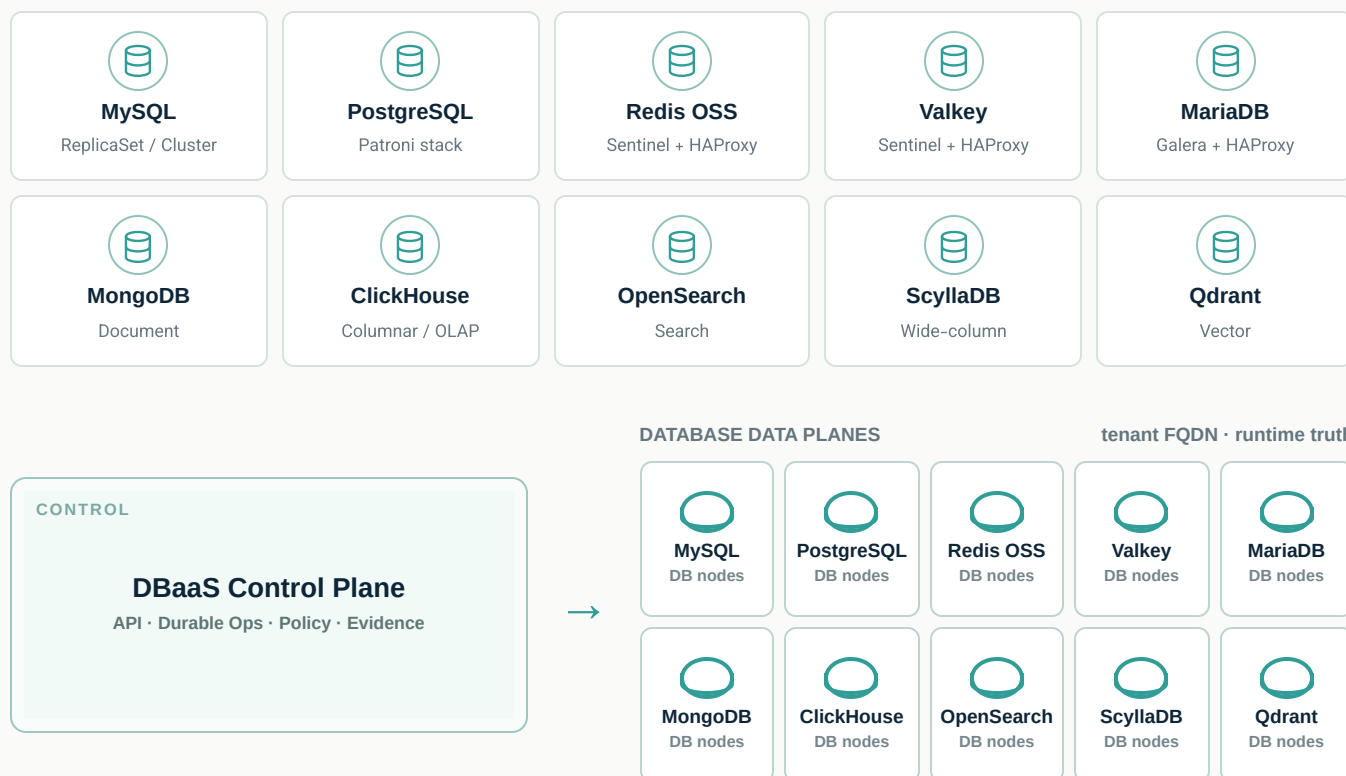


TECHNICAL BROCHURE · DBaaS ENTERPRISE

DBaaS Enterprise

Database lifecycle control, not just installation

Ten database engines under one shared yet Engine-aware Control Plane; an independent control plane for provisioning, capacity changes, maintenance, recovery and observability of databases, with durable operations, topology guardrails and Runtime read-back.



Clear Control Plane / Data Plane boundary

The Manager is not a member of the tenant topology, the quorum or the database service path.

Success through Runtime Read-back

Accepting a request does not finish the job; execution state must be confirmed from Runtime and Evidence.

Bring the servers and the engine; the platform turns the operational lifecycle into a repeatable experience.



CAPABILITIES SNAPSHOT

Capabilities Overview

Engines and Topology

**MySQL**1/2-node ReplicaSet, three-node Group Replication

**PostgreSQL**Patroni + etcd + PgBouncer + HAProxy

**Redis OSS / Valkey**Replication + Sentinel + HAProxy

**MariaDB**Galera multi-primary + HAProxy

**+5 Engines**MongoDB, ClickHouse, OpenSearch, ScyllaDB and Qdrant with each engine's native semantics

Platform Model

**Product API**Single entry point for the Resource model and operations

**PostgreSQL Authority**State and Durable Operation in one Authority

**Outbox + Queue**Transactional outbox and RabbitMQ/Celery

**Runtime Read-back**Operations closed with Evidence and actual state

Lifecycle

01

Provision / Validate

Admission, preconditions and bounded execution

02

Scale / Upgrade

Capacity and version changes with topology awareness

03

Failover / Recover

Only on topologies that meet HA conditions

04

Backup / Restore

Recovery verified before service is restored

Operational Guardrails

**Topology-aware**Every action bound to topology and engine semantics

**Execution Lease**Durable execution ownership that prevents conflicts

**Source Manifest Fence**Execution against a specific, provable source

**Evidence**Job-scoped result and operational read-back

**Design goal:** one shared Control Plane for a consistent operational experience, without forcing different engines into the same semantics.

Capabilities Snapshot

TOPOLOGY & TRANSITION MATRIX

Topology, Quorum and Capacity-change Paths

A node count alone does not mean HA; each engine's semantics decide which operations are allowed and which paths are deliberately restricted.

Engine	1 Node	2 Nodes	3 Nodes	HA / Failover semantics
MySQL	ReplicaSet	ReplicaSet	Cluster / GR	HA at 3 data members
PostgreSQL	Patroni	quorum-limited	Patroni HA	etcd + leader election
Redis OSS	single	auto-FO disabled	Sentinel HA	Sentinel-managed
Valkey	single	auto-FO disabled	Sentinel HA	Sentinel-managed
MariaDB	single	no quorum HA	Galera quorum	multi-primary

MongoDB · ClickHouse · OpenSearch · ScyllaDB · Qdrant

Document, columnar/analytical, search, wide-column and vector; topology, HA and Data Protection follow each engine's native semantics under the same Control Plane and lifecycle.

Capacity-change paths



HA actions only on a suitable topology
Automatic failover or a quorum-sensitive action is not shown decoratively on a topology without a majority; the Guardrail restricts it.

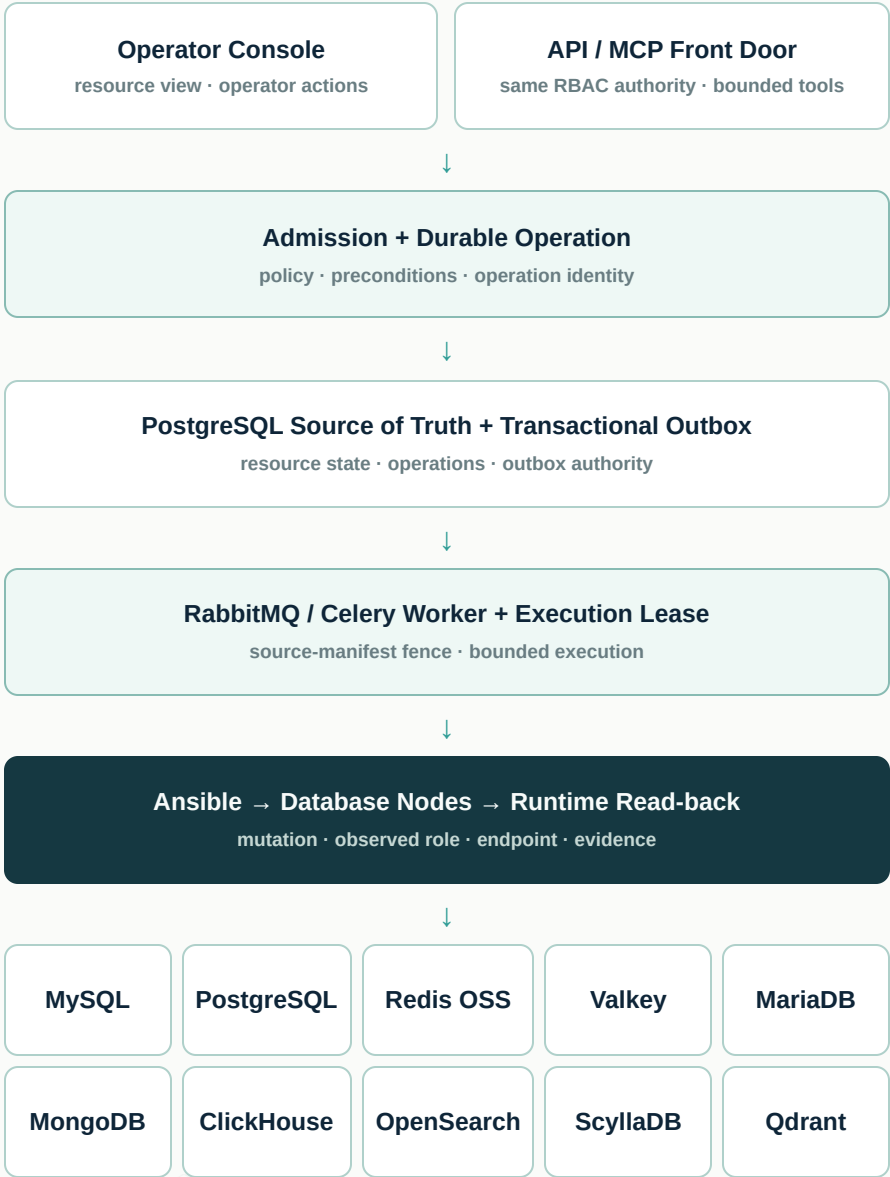
Stable identity independent of nodes
The client identity is the tenant FQDN; the service path points to the active role and the Manager never enters the tenant data path.

Manager is Control Plane only
The Manager is not a witness, not a quorum member, not a tenant Router/HAProxy and not a failover target.

Transitions close with Runtime Read-back
After a scale or failover, membership, role and endpoint are re-read from Runtime and the result is attached to the operation.

ARCHITECTURE OVERVIEW

Control Plane Architecture and Durable Execution



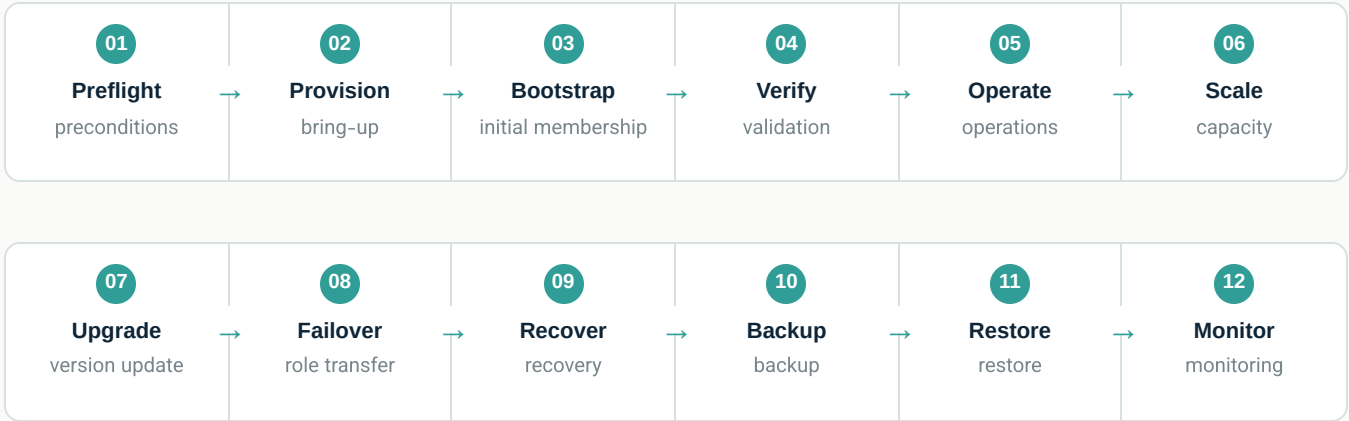
- 01 Single Authority**
Console, API and worker act on the same Resource/Operation authority; the UI never builds an independent source of truth.
- 02 The Data Plane is separate**
The Control Plane has a management role and is not a member of replication, quorum or the tenant service path.
- 03 Execution is fenced**
Lease and source-manifest fence prevent duplicate execution or a mutation against the wrong input.
- 04 Evidence closes the result**
Success does not come from the job's internal status alone; runtime state and post-checks are attached to the result.
- 05 Remote MCP adds no new RBAC model**
OAuth 2.1 + PKCE is the entry point; permission authority stays the existing PostgreSQL model.

Mutation path: Admission → PostgreSQL Durable Operation → Transactional Outbox → RabbitMQ/Celery → Lease/Fence → Ansible → Runtime Read-back → Job-scoped Evidence

LIFECYCLE OPERATIONS

The Complete Database Operations Lifecycle

Actions are not a set of independent commands; every change follows a defined path from Admission to read-back and recovery.



Core Guardrails

Topology-aware

Whether an action is allowed comes from topology and engine semantics, not from a generic button.

Durable & resumable

Operations carry identity and a lease so retry or resume never becomes a parallel mutation.

Policy gates

Admission and execute-time checks define the boundary for high-risk changes.

Observed success

Service role, membership and endpoint are re-read from runtime after each change.



ENGINE PROFILES

Each Engine Keeps Its Own Architecture

MY MySQL	InnoDB ReplicaSet → InnoDB Cluster / Group Replication1 / 2 / 3 nodes RoutingMySQL Router on DB nodesHA3-member Group Replication ProvisionScaleUpgradeFailoverBackupRestore
PG PostgreSQL	Patroni + Streaming Replication + etcd + PgBouncer + HAProxydatabase-node colocated HA stack LeaderPatroni / etcdRoutingHAProxy + PgBouncer ProvisionScaleSwitchoverBackupRestoreMonitor
RD Redis OSS	Replication + Sentinel + HAProxy2-node auto failover disabled ElectionSentinel at HA topologyRoutingHAProxy follows active master ProvisionScaleFailoverBackupRestore
VK Valkey	Replication + Sentinel + HAProxysame safety boundary as Redis ElectionSentinel-managed at HA topology2 nodesautomatic failover disabled ProvisionScaleFailoverBackupRestore
MA MariaDB	Galera (wsrep) + HAProxymulti-primary · quorum at 3 members ReplicationGalera / wsrepRoutingHAProxy on DB nodes ProvisionScaleUpgradeBackupRestore

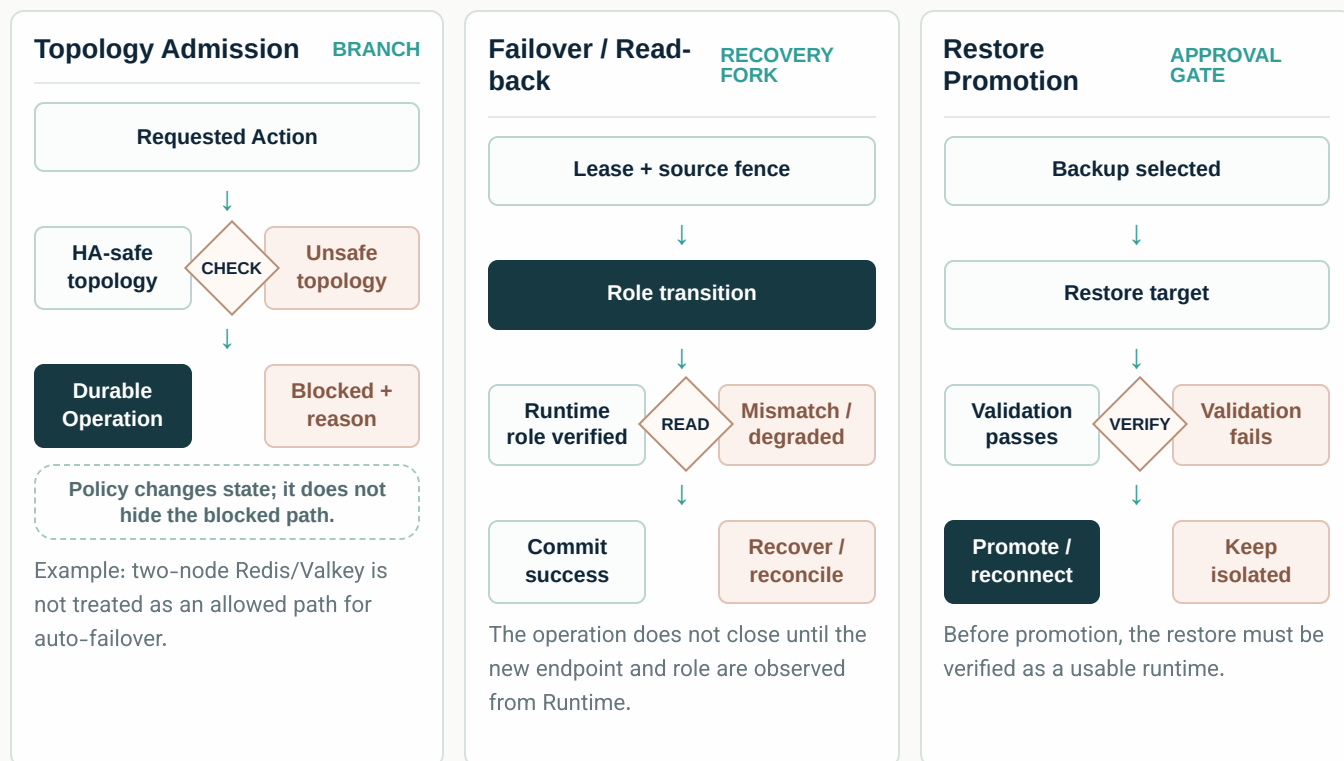
 MongoDB Document	 ClickHouse Columnar / OLAP	 OpenSearch Search	 ScyllaDB Wide-column	 Qdrant Vector
---	---	--	---	--

DEEP TECH WORKFLOWS

Three Paths That Are Not “Linear Cards”

Gates, forks and reconciliation carry meaning, not decoration.

Each path has a defined failure/recovery boundary.



Action families

Provision Validate Scale Upgrade Switchover Failover
Backup Restore Monitor Diagnose Credential lifecycle
Endpoint read-back

Shared principle

The action catalog must stay engine-aware; an action's name alone does not mean it is allowed on every topology.

ACTION & OPERATIONS CATALOG

Operations, Security, Evidence and Recovery

D/V Discovery & Validation

preflight validate-access runtime-status
primary/role endpoint-status config-check drift

Goal: topology and runtime truth are observable before and after every mutation.

L/C Lifecycle & Capacity

provision scale-out scale-in upgrade restart
rejoin destructive guard

Capacity or membership changes close with lease, engine semantics and post-check.

HA HA & Recovery

switchover failover backup restore recover
topology gate

HA operations on an unsuitable topology are blocked or restricted instead of run as "best effort".

SEC Identity, Access & MCP

RBAC credential lifecycle OAuth 2.1 PKCE
mcp_clients bounded tools

MCP is another entry point into the existing authority; it creates no second, independent permission model.

What does this brochure demonstrate?

DBaaS is not a multi-engine installer; it is a lifecycle control plane that preserves the differences between engines while sharing Durable Ops, Policy and Evidence.

01 Control Plane ≠ Data Plane

02 10 engine profiles

03 Topology-aware actions

04 Runtime read-back

05 Durable / fenced execution

06 Evidence-linked operations

Choose the engine. Define the topology. Operate the lifecycle.

The product does not remove expertise; it turns sensitive operational paths into transparent, repeatable and observable workflows.