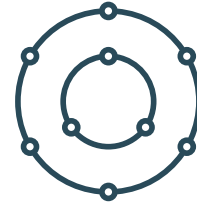


TECHNICAL BRIEF

Distributed High Availability

Active-Active PostgreSQL with deterministic consistency and autonomous failover



Solution overview

The world increasingly runs on real-time data. Contactless payments, AI-driven workflows, and agentic systems all depend on infrastructure that can process and deliver information with near-zero latency and never goes down.

Traditional PostgreSQL deployments weren't designed for this reality. The dominant architecture still relies on a single primary node that handles all writes, with passive standbys that exist only to take over if something fails. This model makes sense for simpler workloads, but it introduces hard limits on performance, scalability, and availability that modern systems quickly encounter.

Those limits are no longer just an inconvenience. As organizations push toward real-time data processing and AI, and agentic workloads multiply, the volume and velocity of database activity create gaps in high availability within the business. These are gaps that businesses no longer can afford.

Two structural problems stand in the way

Traditional PostgreSQL stumbles over two fundamental availability challenges that tuning and tooling can't fully resolve.

- **Failover in these systems is inherently disruptive:** When a primary node fails, the system must detect the failure, promote a standby, reconfigure clients to point to the new primary, and reconcile replication state, all in sequence, all under pressure. Even in well-optimized environments, recovery time objectives (RTOs) run from tens of seconds to several minutes. Recovery point objectives (RPOs) are harder to guarantee: Depending on the amount of replication lag at the time of failure, some data loss may occur. For organizations running real-time or AI-driven workloads, that window, however brief, is unacceptable.
- **High availability is a patchwork, not a system:** There is no single PostgreSQL high-availability solution; it is a composition of tools such as replication managers, failover orchestrators, monitoring systems, backup utilities, and upgrade mechanisms. Each is independently configured, and each has its own failure modes. Operators must understand how these pieces interact, maintain them separately, and diagnose problems that can span multiple layers. This fragmentation doesn't just increase operational complexity; it multiplies the surface area for things to go wrong and raises the workload on the teams responsible for keeping systems running.

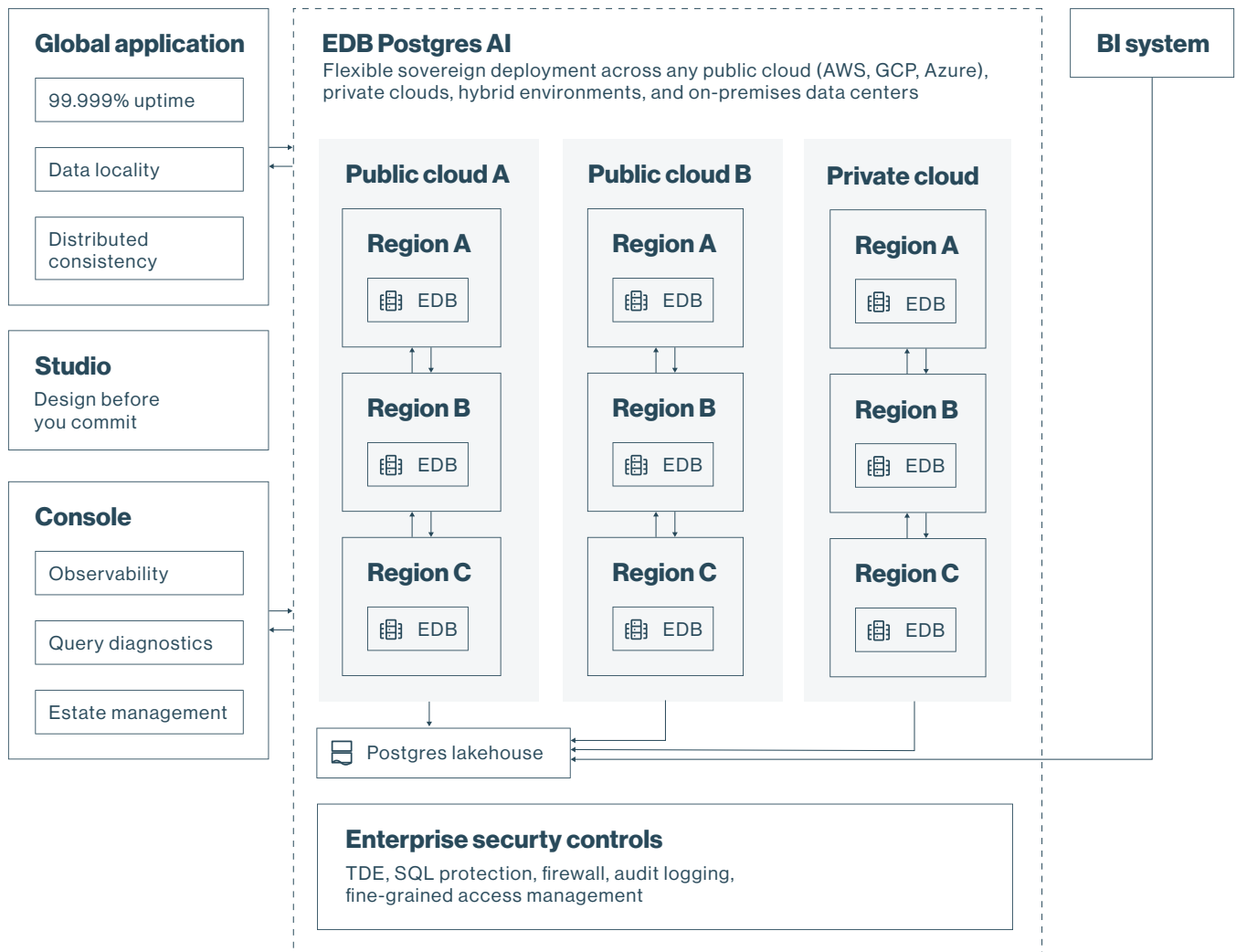
In combination, these two problems highlight the fact that traditional PostgreSQL high availability is neither as reliable nor as manageable as modern workloads demand.

A better path forward

EDB Postgres® AI (EDB PG AI) is built to solve the stumbling blocks outlined above. Instead of the traditional model in which one primary node handles writes while standby nodes wait, EDB PG AI distributes workloads across multiple active nodes simultaneously, ensuring that data stays consistent, failures are handled automatically, and everything is managed from a single control plane.

This isn't just about incremental improvement. EDB PG AI brings distributed systems principles to PostgreSQL without the operational overhead that typically comes with them. The result is a platform that eliminates unplanned downtime, guarantees data consistency, and reduces the sprawl of tools and configurations that make traditional high availability so difficult to operate at scale.

For technical teams, that translates to fewer moving parts, faster and more predictable recovery, and greater confidence in how the system behaves under pressure. For the business, it means a data foundation capable of supporting continuous, global operations even as the demands placed on it keep growing.



Distributed high availability, the new standard

Distributed high availability isn't a better version of traditional high availability. It's a fundamentally different approach, built for a world in which downtime is measured in dollars, and data loss is simply not an option.

This matters most in environments where the stakes leave no room for compromise. Financial systems in which every transaction must be accounted for. Payment networks that process millions of transactions across the global infrastructure around the clock. Energy and critical infrastructure, where an interruption doesn't just affect revenue, it affects people. Global SaaS platforms whose users in any region expect the same low-latency, always-on experience regardless of what's happening in the underlying infrastructure.

Distributed high availability is designed specifically for that threshold. It eliminates the trade-offs between resilience, performance, and operational simplicity that traditional architectures force you to accept.

A distributed architecture built for always-on operations

At the core of this solution is a geo-distributed, high availability, active-active cluster in which every node can process both reads and writes simultaneously. There is no single primary to protect, no standby waiting to be promoted, and no failover sequence to execute. When a node fails, the cluster continues operating because it was never dependent on that node in the first place.

What sets EDB PG AI distributed high availability apart

EDB PG AI's approach to high availability differs from traditional solutions in four meaningful ways:

- **Every node is a full participant:** In primary standby architectures, only one node accepts writes. In an active-active cluster, every node handles read and write operations. This eliminates write bottlenecks and makes better use of available infrastructure.
- **Consistency is built in, not bolted on:** A combination of quorum commit, conflict-free replicated data types (CRDTs), and automated conflict resolution ensures strong data consistency across nodes without manual intervention or custom application logic.
- **High availability exists in a single system:** All high-availability functions—replication, failover, monitoring, and recovery—are integrated into one unified platform. There is no need to assemble and maintain a stack of separate tools, each with its own configuration and failure modes.
- **Deployments adapt to your requirements:** Whether driven by regulation, data sovereignty, or operational preference, organizations can deploy the platform in the environments that meet their specific needs on premises, in the cloud, or across both.

These capabilities combine to create a PostgreSQL architecture that goes beyond merely surviving failure. It's designed so that failure becomes largely irrelevant to the systems and users that depend on it.

Under the hood: Core architectural components

These capabilities are made possible by five tightly integrated layers, each designed to eliminate a specific weakness in traditional high-availability architectures.

- **Active-active replication:** Every node participates in replication, receiving and applying transactions in near real time. There is no idle standby capacity because every node is working all the time, which means the cluster loses nothing when any single node drops out.
- **Quorum commit protocol:** A transaction is only confirmed once a majority of nodes have acknowledged it. This ensures that data is durable and consistent across the cluster before any write is considered complete, removing the risk of replication lag that makes RPO guarantees so difficult in traditional deployments.
- **Conflict detection and resolution:** When concurrent writes occur across regions, deterministic algorithms and CRDT-based mechanisms resolve them automatically. The cluster converges to a consistent state without manual intervention or application-level work-arounds.
- **Unified control plane:** A single management layer spans the entire cluster, providing topology visualization, configuration management, failover orchestration, and observability. Operators work from one place rather than context-switching across a stack of independent tools.
- **Intelligent traffic routing:** Client requests are dynamically routed to available nodes, so failover and load balancing happen beneath the application layer, making them invisible to users and requiring no client-side reconfiguration.

Together, these capabilities fundamentally change what high availability delivers. Rather than minimizing the impact of failures, EDB PG AI is designed to make them irrelevant, ensuring that node outages, network disruptions, and regional failures never reach the applications and users that depend on the system.

Consistency model: Strong guarantees in a distributed system

One of the primary challenges in active-active systems is maintaining data consistency across distributed nodes. EDB PG AI addresses this through a combination of quorum-based consensus, deterministic conflict resolution, and commit control mechanisms.

Quorum commit

With distributed high availability, EDB PG AI implements a quorum commit protocol in which a transaction is considered committed only after it has been acknowledged by a majority of nodes. This ensures:

- **Durability:** Data is replicated across multiple nodes before acknowledgment.
- **Consistency:** All nodes converge on the same state.
- **Fault tolerance:** The system can tolerate node failures without losing committed data.

This approach avoids the risks associated with asynchronous replication, in which data loss can occur during failover.

Conflict resolution

In any distributed system in which multiple nodes accept writes simultaneously, concurrent updates to the same data are inevitable. EDB PG AI handles this through a layered resolution approach that ensures the cluster always converges to a consistent state without manual intervention.

- **Deterministic resolution policies** apply business rules to resolve conflicts predictably, ensuring consistent, auditable outcomes.
- **Conflict-free replicated data types (CRDTs)** provide automatic convergence for data structures in which concurrent updates can be mathematically reconciled.

Combined, these mechanisms enable concurrent writes across regions and promote a consistent, trustworthy state regardless of where or how data is written.

Commit scope control

EDB PG AI allows fine-grained control over commit behavior, enabling applications to balance consistency, latency, and availability. For example, control can include:

- **Global commit scope** for strong consistency across regions
- **Local commit scope** for low-latency operations with eventual consistency

This flexibility allows architects to optimize system behavior based on application requirements.

Failover and recovery: No promotion, no interruption

Traditional high availability systems are built around state transitions. When a primary node fails, a standby must be detected, promoted, and reconfigured before the system can resume normal operation. The sequence takes time and introduces risk at every step. EDB PG AI's distributed high availability eliminates that sequence entirely. Because every node is already active and already processing transactions, there is no promotion to trigger and no reconfiguration to execute. When a node fails, the remaining nodes continue without pause. The cluster doesn't switch states; it simply continues operating with one fewer participant.

How failover works in practice

The process reduces to two steps:

- Continuous health monitoring detects node or network failures as soon as they occur.
- Client traffic is immediately rerouted to healthy nodes.

Because every node was already active and synchronized, there is nothing to promote and nothing to reconcile. The result is a failover that is, in effect, just a routing decision and produces no service interruption visible to applications or users.

Zero data loss

Transactions are confirmed only after a majority of nodes have acknowledged them; no committed data is lost, even if a node fails at the worst possible moment. There is no replication lag to worry about, no async gap where data exists on one node but not yet on others.

When data integrity is nonnegotiable, as in financial transactions, payment records, and audit trails, this means RPO effectively reaches zero, because every write that the system confirms has already been committed across the cluster. This gives organizations a guarantee that their infrastructure will never lose confirmed data.

Operational model: Unified control, less complexity

EDB PG AI consolidates all high-availability functions into a single control plane, replacing the fragmented tool stacks that make traditional PostgreSQL environments difficult to operate and scale.

Unified management

All operational functions, topology visualization, cluster configuration, monitoring, alerting, failover orchestration, and upgrade workflows are accessible from a single place. Operators no longer need to context-switch across independent tools or reconcile information from multiple systems. The entire cluster is visible and manageable from a single interface.

Zero-downtime upgrades

Rolling upgrades are applied incrementally across nodes while the cluster continues serving traffic. There are no maintenance windows to schedule, no downtime to communicate to users, and no trade-off between keeping systems current and keeping them available.

Faster deployment

Automated configuration and simplified workflows reduce the time required to stand up a distributed cluster by up to 75%—getting production-grade infrastructure in place faster and with less room for manual error.

Global distribution and data sovereignty

EDB PG AI is designed for organizations operating across regions that can't afford to choose between performance and compliance. A write-local, read-global architecture ensures that users interact with their nearest node for the lowest possible latency, while data is replicated globally to maintain consistency. Read operations can be served from any node, distributing load naturally across the cluster. Where data residency requirements apply, selective replication and row-level filtering give organizations precise control over where specific data lives. This supports compliance with GDPR, FedRAMP, and other regional regulations without requiring separate infrastructure or architectural compromises.

The result is a platform that delivers global performance and local control simultaneously and without complexity.

Comparative architecture analysis

Distributed Database Architecture Comparison

Capability	EDB PG AI	Commercial DBs including Oracle	Cloud DBaaS (Aurora Global DB)	Distributed database alternatives
Replication model	Active-active	Active-passive/RAC	Primary-secondary	Active-active
Consistency model	Quorum + deterministic resolution	Strong but proprietary	Eventual/async	Strong consistency
Conflict resolution	Built-in (CRDT, deterministic)	Limited	Not applicable	Built in
Failover model	Routing based	Promotion based	Promotion based	Automatic
RTO	<30 seconds	Minutes	Seconds–minutes	Seconds
RPO	Zero	Possible loss	Possible loss	Near zero

*Competitive comparisons are based on publicly available information and are subject to change as vendor offerings evolve and new information is made available. All product names, trademarks, and registered trademarks are the property of their respective owners.

Operational Model Comparison

Function	EDB PG AI	Traditional high availability	Cloud high availability
Failover	Instant, automated	Manual/scripted	Automated but delayed
Monitoring	Unified control plane	Multiple tools	Vendor managed
Upgrades	Zero downtime	Requires downtime	Maintenance windows
Complexity	Low	High	Medium

*Competitive comparisons are based on publicly available information and are subject to change as vendor offerings evolve and new information is made available. All product names, trademarks, and registered trademarks are the property of their respective owners.

Deployment and Control Comparison

Capability	EDB PG AI	Commercial DBs including Oracle	Cloud DBaaS (Aurora Global DB)	Distributed database alternatives
Deployment	Hybrid/on-prem/multi-cloud	Limited	Cloud only	Cloud native
Sovereignty	Full control	Limited	Limited	Limited
Vendor lock-in	Low (Postgres)	High	Medium	Medium
Migration complexity	Low	High	Low	High

*Competitive comparisons are based on publicly available information and are subject to change as vendor offerings evolve and new information is made available. All product names, trademarks, and registered trademarks are the property of their respective owners.

What makes EDB PG AI different

EDB PG AI's approach to high availability is distinguished by four capabilities that, taken together, address what traditional PostgreSQL architectures have never fully solved.

- **Every node is a full participant:** Unlike primary-standby architectures, in which only one node accepts writes, every node in an active-active cluster handles both reads and writes. This removes the write bottleneck that limits throughput in traditional deployments and ensures that no single node is a point of failure.
- **Consistency is guaranteed, not approximated:** Quorum commit, CRDTs, and automated conflict resolution work together to ensure strong data consistency across the entire cluster without manual intervention, custom application logic, or post-failure reconciliation.
- **Use one platform, not a tool stack:** All high-availability functions—replication, failover, monitoring, recovery, and upgrades—are integrated into a single system. This reduces the configuration surface, lowers operational overhead, and gives teams a clear, unified view of how the system is behaving.
- **Deployments meet your requirements:** Whether driven by regulation, data sovereignty, or infrastructure preferences, organizations can deploy EDB PG AI in the environments that work for them on premises, in the cloud, or across both, without architectural compromise.

Built for what's next

The demands on data infrastructure are only accelerating. Real-time workloads, AI-driven systems, and global operations require a foundation that is resilient by design, not by intervention. EDB PG AI delivers exactly that: a distributed high-availability architecture that gives technical teams fewer tools to manage, faster recovery without the risk of data loss, and a single platform built to operate continuously under any conditions.

For the business, it means the data infrastructure that supports revenue, compliance, and customer experience will simply keep running whether a node fails, a region goes dark, or the workload doubles overnight. In a world where downtime is measured in dollars and data integrity is nonnegotiable, EDB PG AI provides the foundation organizations need to operate without limits.

EDB Postgres AI: The sovereign data and AI platform for the agentic enterprise

EDB PG AI brings together a unified data layer, governance, sovereign control and orchestration, and an agent runtime environment, giving enterprises a trusted foundation for AI on infrastructure they own and control. The platform unifies transactional, analytical, and AI workloads in a single Postgres-based architecture—eliminating ETL, data movement, and operational fragmentation. And you choose where and how to deploy: on-premises, cloud, managed, or certified appliance.

The outcome: production-ready sovereign AI in days or weeks, not months.



EDB Postgres® AI (EDB PG AI) is the sovereign data and AI platform for the agentic enterprise. Built on Postgres, the world's leading open source database, EDB PG AI unifies transactional, analytical, and AI workloads in a single governed architecture, on-premises and across clouds. To learn more, visit www.enterprisedb.com.

© EnterpriseDB Corporation 2026. All rights reserved.