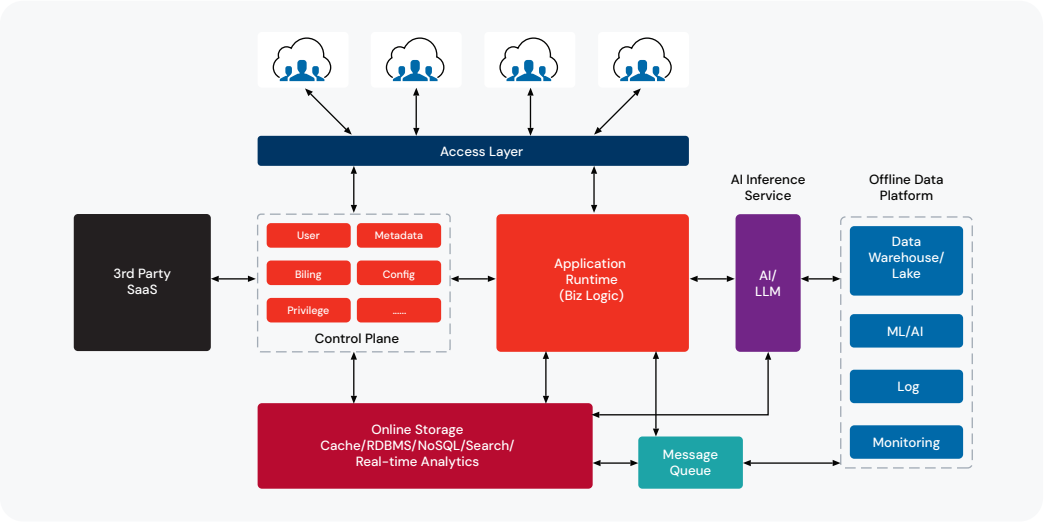
Scaling Multi-Tenant SaaS Platforms

Square PLAID Dify conga Catalyst

What is SaaS Platform Architecture?

SaaS platform architecture is a design pattern where each tenant requires a unique schema or individual tables for their data. SaaS platform architecture can dynamically onboard thousands of concurrent new tenants or introduce new features requiring data isolation while maintaining high availability for tenants to access when they need it.

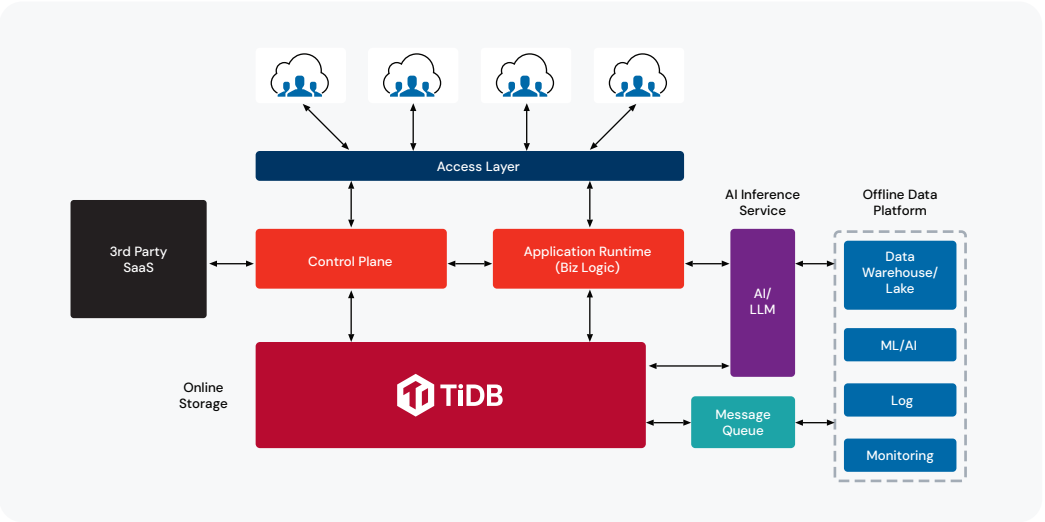


What are the Major Challenges in Scaling Multi-Tenant SaaS Platforms?

- **Noisy Neighbors & SLA Risk:** Tenant mix shifts cause p95/p99 spikes and unpredictable performance across shared resources.
- **Architecture Trade-offs:** Shared schema struggles with isolation/BYOK/residency, while database-per-tenant creates metadata/connection explosions and rising cost.
- **Fleet Sprawl & Ops Overhead:** Sharding, bin-packing, and per-tenant management inflate operational complexity, DR/backups, and on-call toil.
- **Change Velocity Without Downtime:** Continuous schema/ index updates and rolling upgrades are required, but legacy solutions force maintenance windows.
- **Compliance at Multi-Region Scale:** Enforcing access controls, audit, and data residency across products/regions—often with extensions/add-ons—adds heavy governance burden.

How Does TiDB Solve Multi-Tenant SaaS Platform Challenges?

TiDB is an open-source, distributed SQL database built for multi-tenant SaaS, delivering predictable latency, strong consistency, and high availability at scale. It's ideal for dynamic workloads that require scaling compute and storage separately while shipping changes without downtime.



- **Handle Growth with Steady P95/P99:** TiDB scales out compute and storage independently to keep latency predictable under millions of tenants, schemas, and connections, all without manual sharding.
- **Lower TCO with True Multi-Tenant Controls:** TiDB consolidates fleets while maintaining per-tenant isolation using Resource Groups and data placement policies; avoid over-provisioning and noisy-neighbor bleed.
- **Always-On Operations:** With TiDB, you can ship change with online DDL and rolling upgrades; automated failover and fast recovery (including PITR) reduce risk during incidents and releases.
- **Real-time Insight Without ETL Bloat:** TiDB serves OLTP and analytics together while streaming changes with TiCDC for risk, reporting, and personalization on fresh data.
- **Enterprise-Grade Governance at Scale:** TiDB enforces least-privilege access, auditing, and data-residency/BYOK patterns across regions and products without fragmenting your architecture.

TiDB Success Story

Atlassian

Atlassian runs one of the world's largest multi-tenant SaaS platforms (Jira, Confluence, Trello). As tenant count and data grew, its mix of RDS PostgreSQL and DynamoDB created fragmentation: thousands of siloed instances, rising costs, schema limits, and heavy ops toil to keep noisy neighbors and SLAs in check.

To break through, Atlassian designated TiDB as its primary relational database and consolidated diverse workloads—transactional, KV/document, FTS, and analytics—onto a single distributed SQL foundation across 13 regions. One TiDB cluster now accommodates 3M+ tables (e.g., third-party plugin ecosystems) with online DDL, resource controls, and HTAP capabilities, dramatically shrinking instance sprawl.

The impact: roughly 1% of the prior instance count, lower vCPU footprint, faster schema evolution (lock-free DDL), and streamlined operations (rolling upgrades, faster recovery) delivering 99.99% uptime. Engineering teams gained agility as platform work unified under one engine, while costs and operational risk dropped even as petabyte-scale data and millions of QPS continued to grow.