

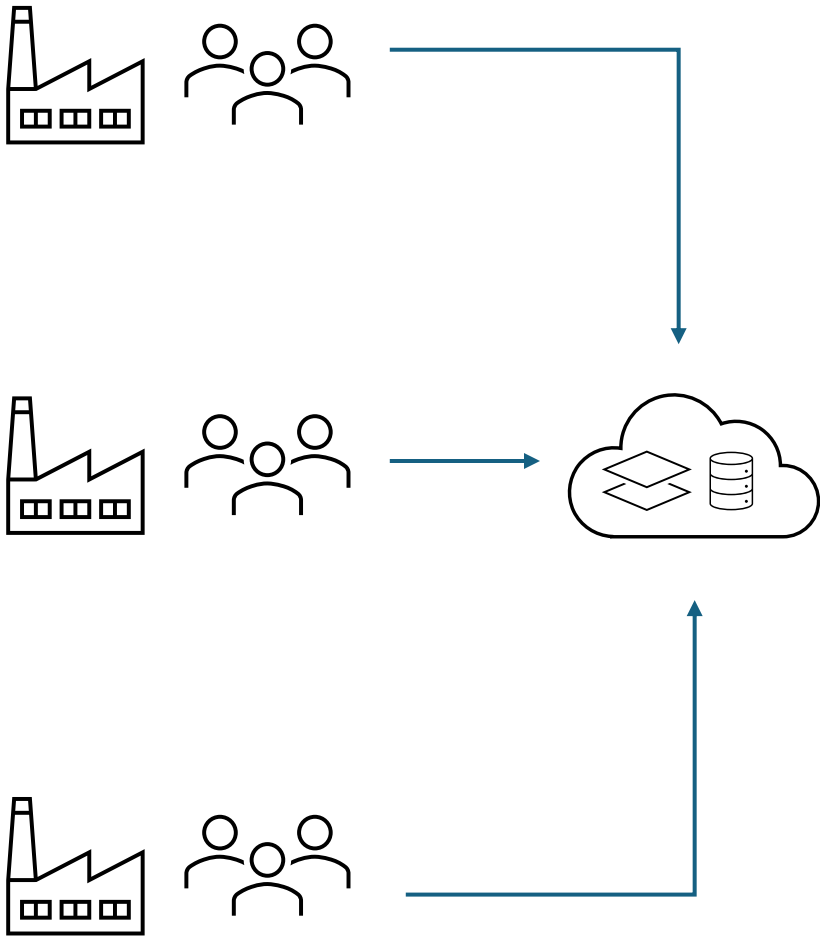
Exploring Multi-Tenant Architectures

Models, Benefits, and Challenges

by Marian Veteanu



Introduction to Multi-tenancy

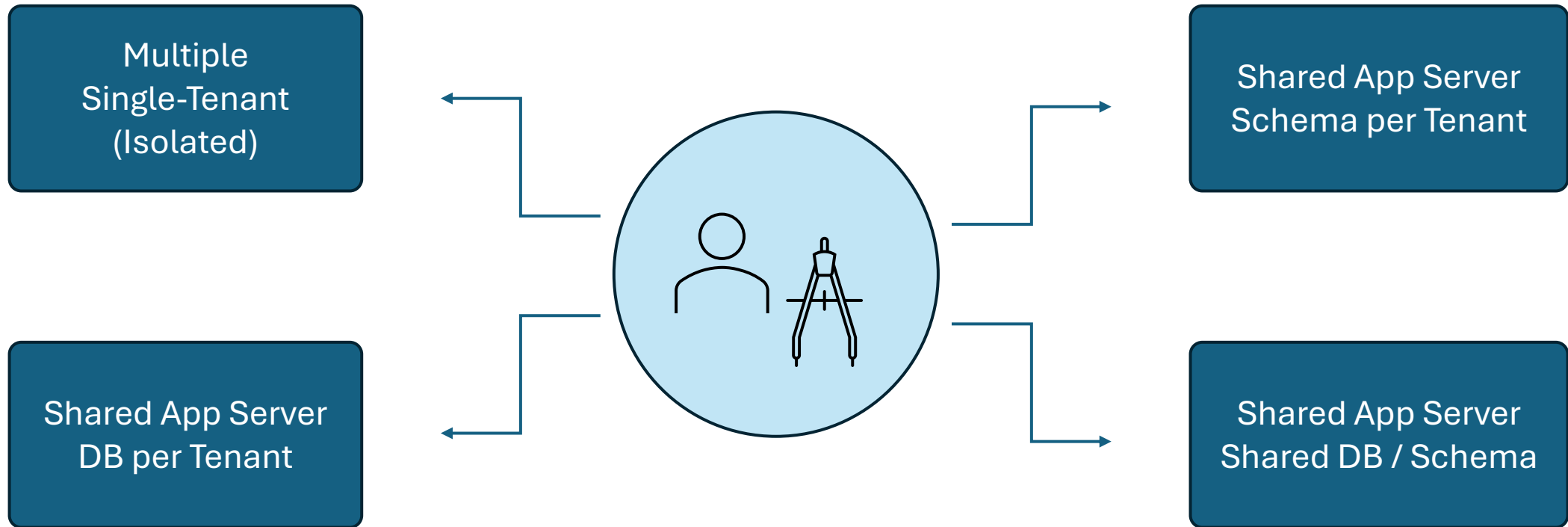


You're considering building a B2B SaaS application to meet a specific business need for organizations. In addition to scalability and security, one of the key requirements for your system is *multitenancy*, enabling you to efficiently serve multiple customers (or tenants) from your cloud-based platform.

Each tenant's data should be isolated, and the tenants should be unaware of the shared infrastructure.

How can we structure multi-tenancy for scalability and security?

Different types of multitenant architectures

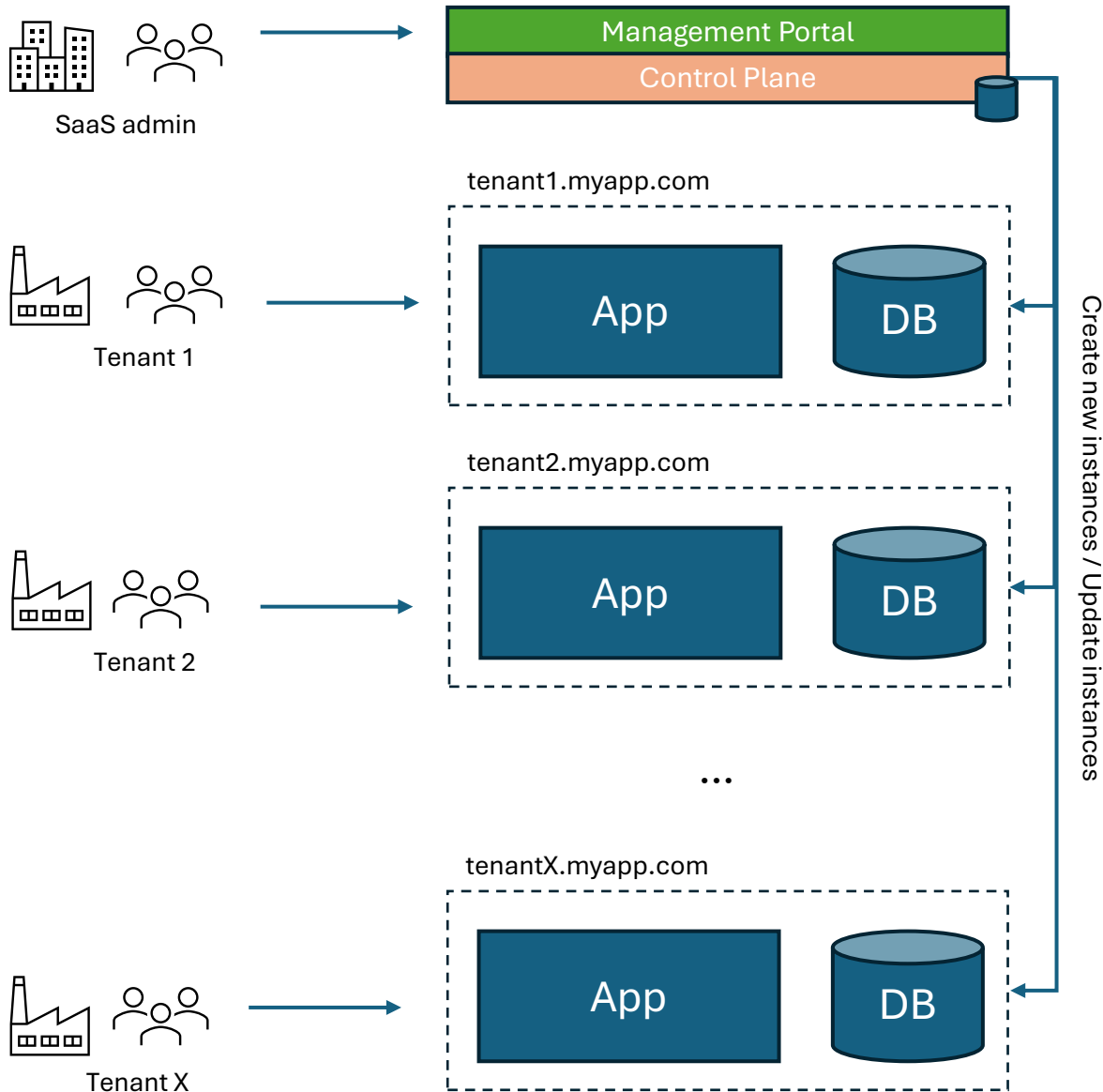


Reference:

<https://www.youtube.com/watch?v=UlbaPcyw4Hw>

<https://learn.microsoft.com/en-us/azure/architecture/guide/multitenant/approaches/control-planes>

Multiple single-tenant apps with single DBs



Benefits

Maximum Isolation: High security as each tenant's data and application are separated. Preferred in some regulated industries.

Customizability: Tenants can have their own configurations and custom features and are in control when updates are applied. Regulatory requirements, can be more easily managed.

Performance: A tenant can be run on dedicated hardware not shared with other tenants leading to maximum performance.

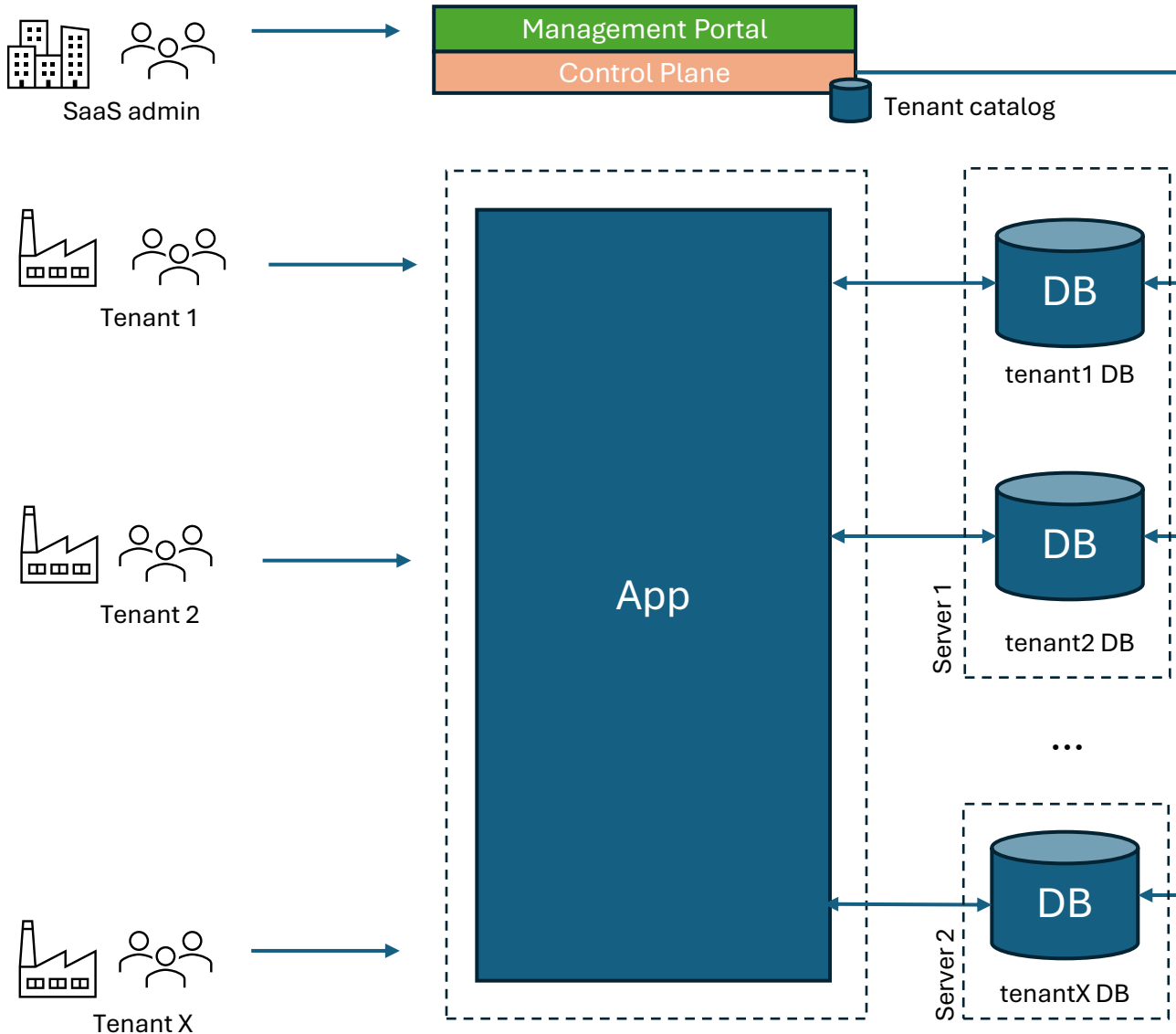
Lift and shift: This model offers a quick transition from on-prem or legacy apps to modern cloud-based SaaS systems.

Challenges

High Cost for SaaS provider: Infrastructure costs scale linearly with the number of tenants.

Complex Maintenance: Need to maintain multiple instances. Can be mitigated by investing in management portal and control plane, or use infrastructure as a code to automate tenant installations and updates.

Shared App Server, Each tenant gets its own DB



Benefits

Good Data Isolation: Each tenant has its own database, reducing the risk of data leakage and making it easier to manage security and compliance.

Centralized Application Management: Since the application is shared, it simplifies updates, bug fixes, and feature rollouts across all tenants without needing multiple deployments.

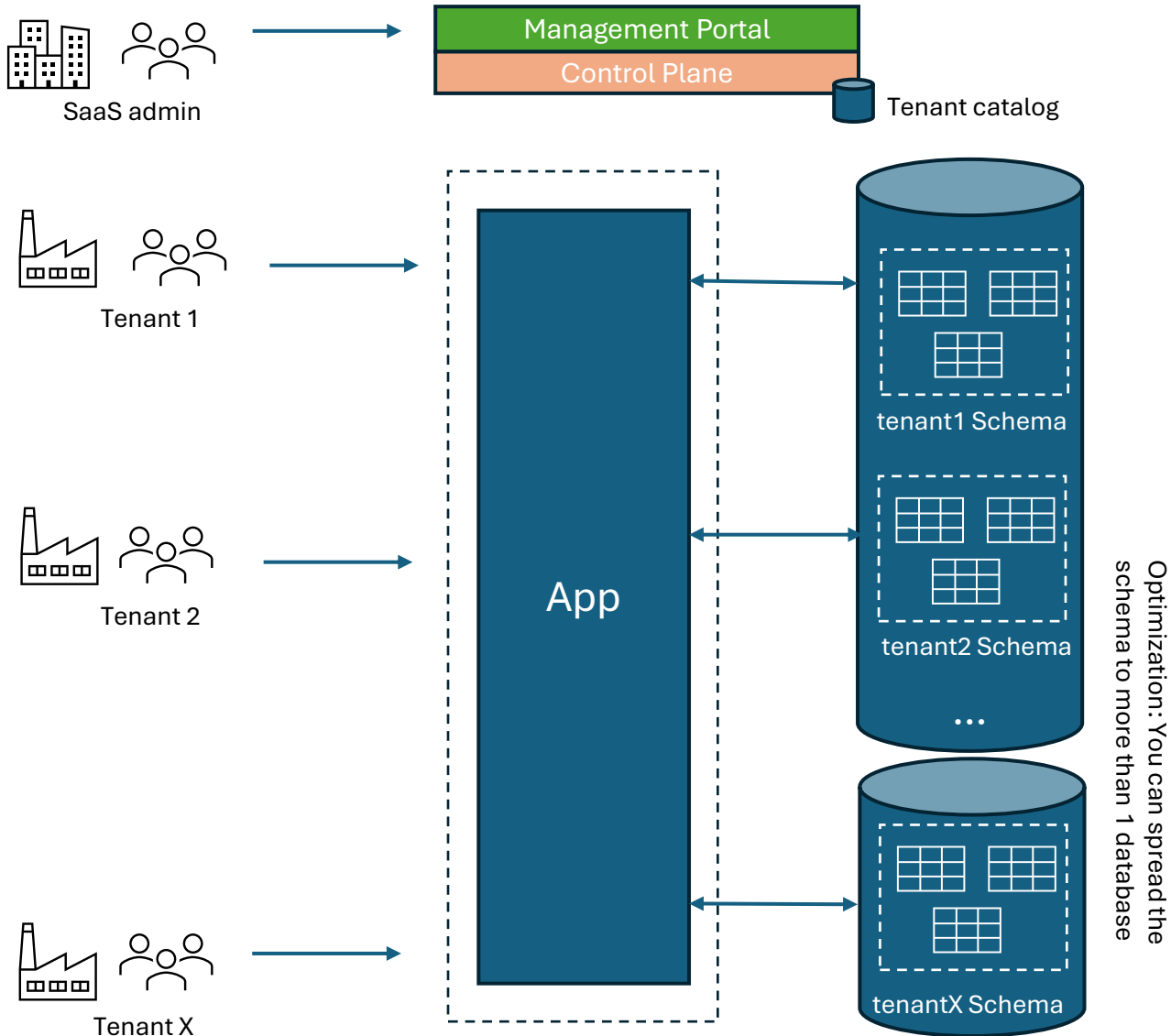
Easier Backup and Scaling: Individual databases can be backed up, restored, and scaled independently, offering flexibility in managing resources for each tenant.

Challenges

Higher Database Overhead: While application servers are shared, maintaining separate databases for each tenant increases database management overhead and cost.

Complex Database Management: As the number of tenants grows, managing numerous databases can become operationally complex.

Shared App, Each tenant gets its own schema



Tenants share a single database, but each tenant has its own schema.

Benefits

Balance of Isolation and Cost: Allows for a degree of isolation between tenants while still saving on infrastructure costs.

Easier Customization: Allows custom configurations for each tenant at the schema level.

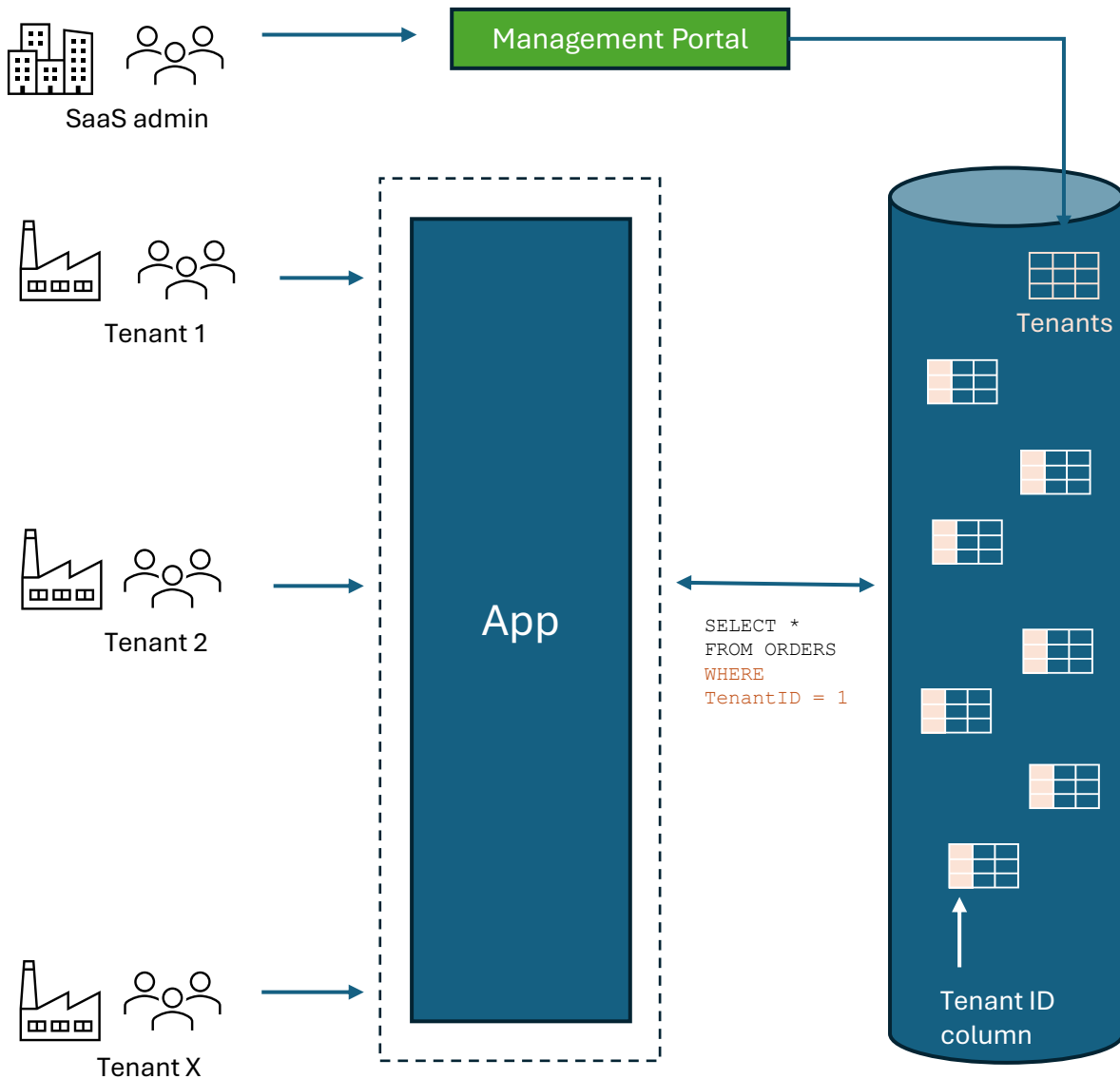
Challenges

Moderate Complexity: Managing multiple schemas adds a layer of complexity, particularly in scaling.

Database Size: The size of the database grows as more schemas are added. Note: You can spread the schema across multiple databases.

Model is good for applications needing a higher degree of isolation but that don't want to maintain separate databases for each tenant.

Shared App, Shared DB / Schema



Tenants share a single database and the same schema, with tenant-specific data distinguished by a unique tenant identifier.

Benefits

Cost Efficiency: Only one database and one schema to manage, reducing infrastructure costs.

Simplified Operations: A single version of the application simplifies upgrades and maintenance.

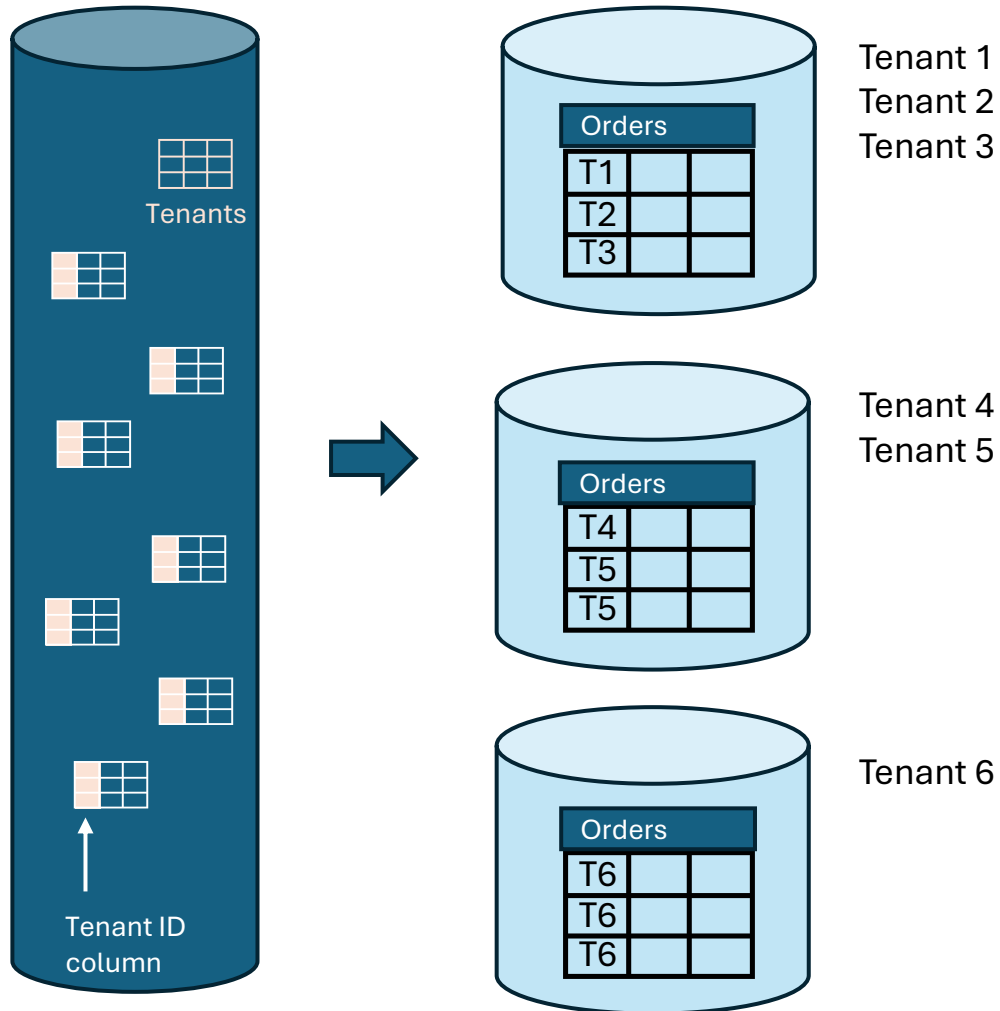
Challenges

Data Security Concerns: Data needs to be carefully partitioned at the application level.

Scalability Limits: As the number of tenants grows, the database could become a bottleneck.

Suitable for SaaS providers who prioritize low costs and fast onboarding of new customers.

Optimized Shared Schema Model (using Sharding)



An optimization to the “Shared DB / Schema model” is to partition data across multiple databases (or shards) to distribute the load. The application routes tenant requests to the correct shard based on a predefined sharding key (e.g., tenant ID)

Benefits

Scalability: Sharding allows for horizontal scaling by distributing tenants across multiple databases, which helps handle larger workloads and high traffic.

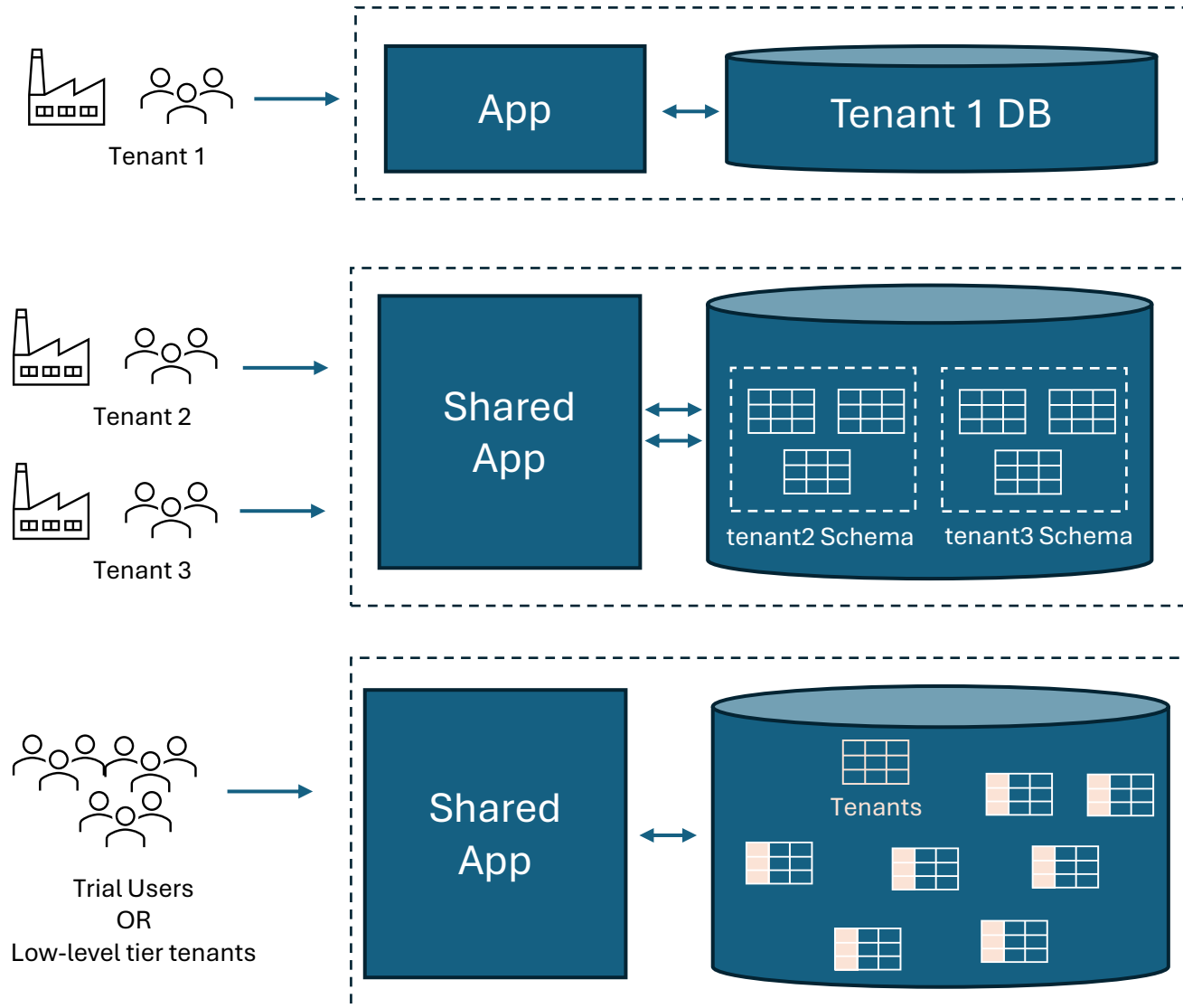
Performance Optimization: Sharding reduces the burden on any single database, improving query performance and enabling faster data access for tenants.

Challenges

Complex Sharding Logic: Determining the right sharding key and implementing logic to route requests can add complexity to the application architecture.

Data Movement: As tenants grow, data may need to be re-sharded, which can be complex and introduce downtime or migration overhead.

Hybrid Approach



Combining elements from different models, the hybrid approach allows flexibility. For instance, high-value tenants might have isolated databases, while lower-value tenants share resources.

Benefits

Flexibility: The hybrid model can adapt based on tenant size, value, or security requirements.

Cost Efficiency with Customization: Allows you to control costs while providing customization options for key clients.

Challenges

Management Complexity: Managing multiple models adds complexity in both design and maintenance.

The hybrid model works well for SaaS platforms with diverse customer needs and varying levels of customization or security requirements.

Comparing models

Model	Cost	Security	Customizability	Complexity	Best Use Case
Single-Tenant (Isolated)	High	High	High	High	High-security apps, custom needs
DB per Tenant	Medium	Medium	Medium	Medium	Balanced needs (medium size SaaS)
Schema per Tenant	High	Medium-High	Medium	Medium	Good data isolation (medium size SaaS)
Shared DB / Schema	Low	Medium	Low	Low	SaaS with many small tenants
Hybrid	Variable	Variable	Variable	High	Platform with diverse customer base

Marian Veteanu

Technology Architect and Product Leader

Looking to see how I can add value to your organization? Message me!

<https://www.linkedin.com/in/mveteanu/>
<https://x.com/mveteanu>

