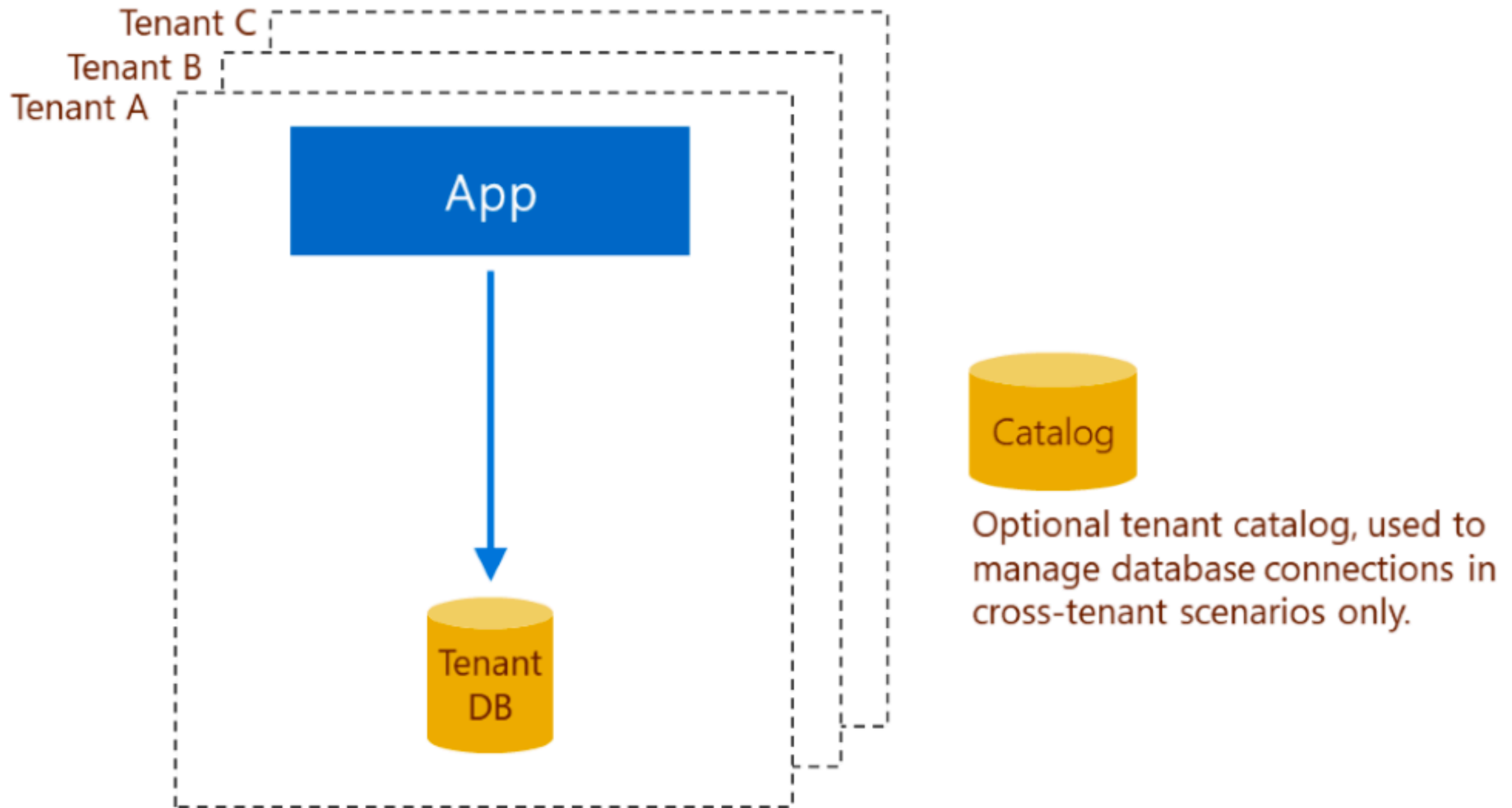


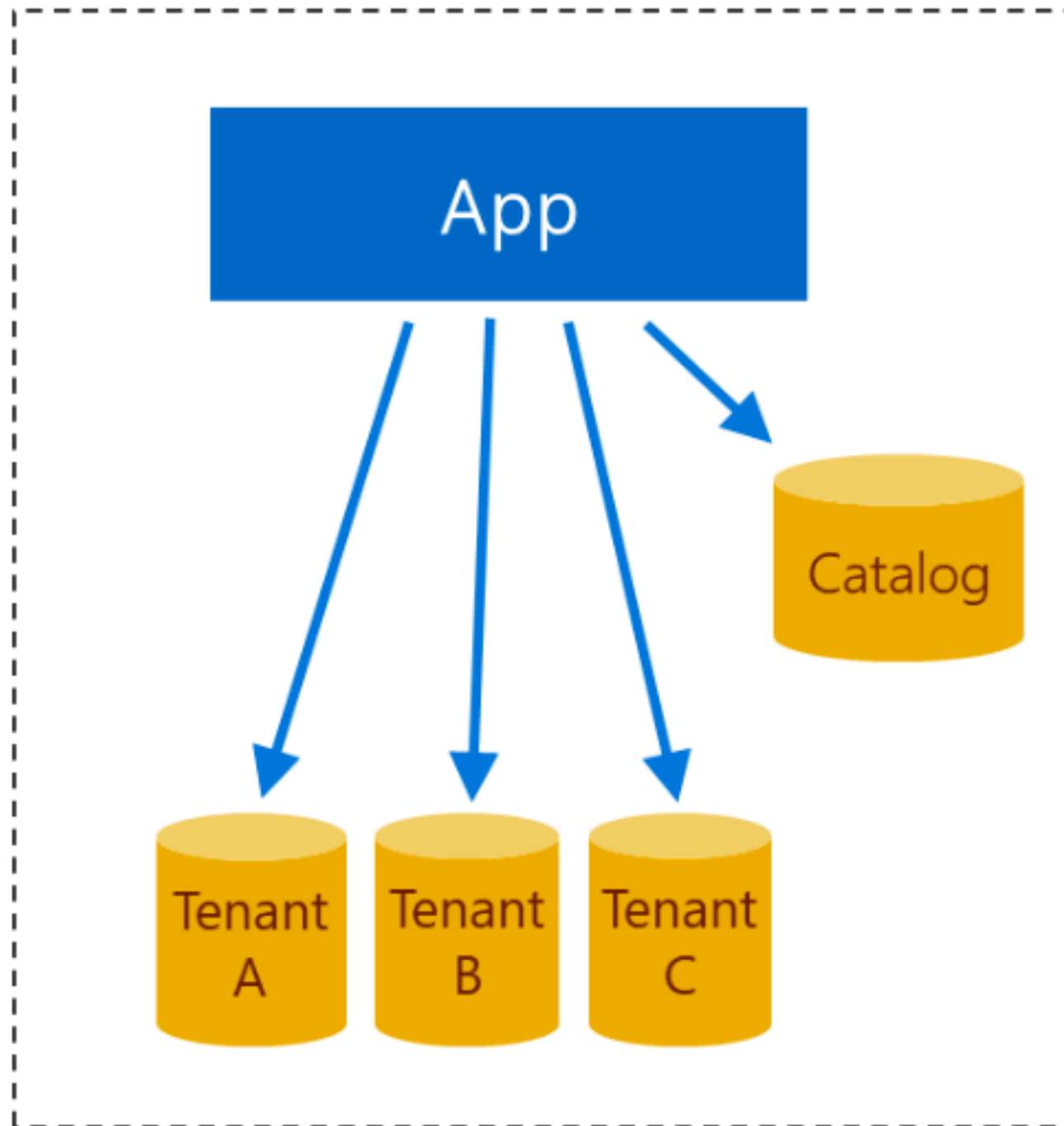
# Multi Tenant App

## Standalone single-tenant app with single-tenant database

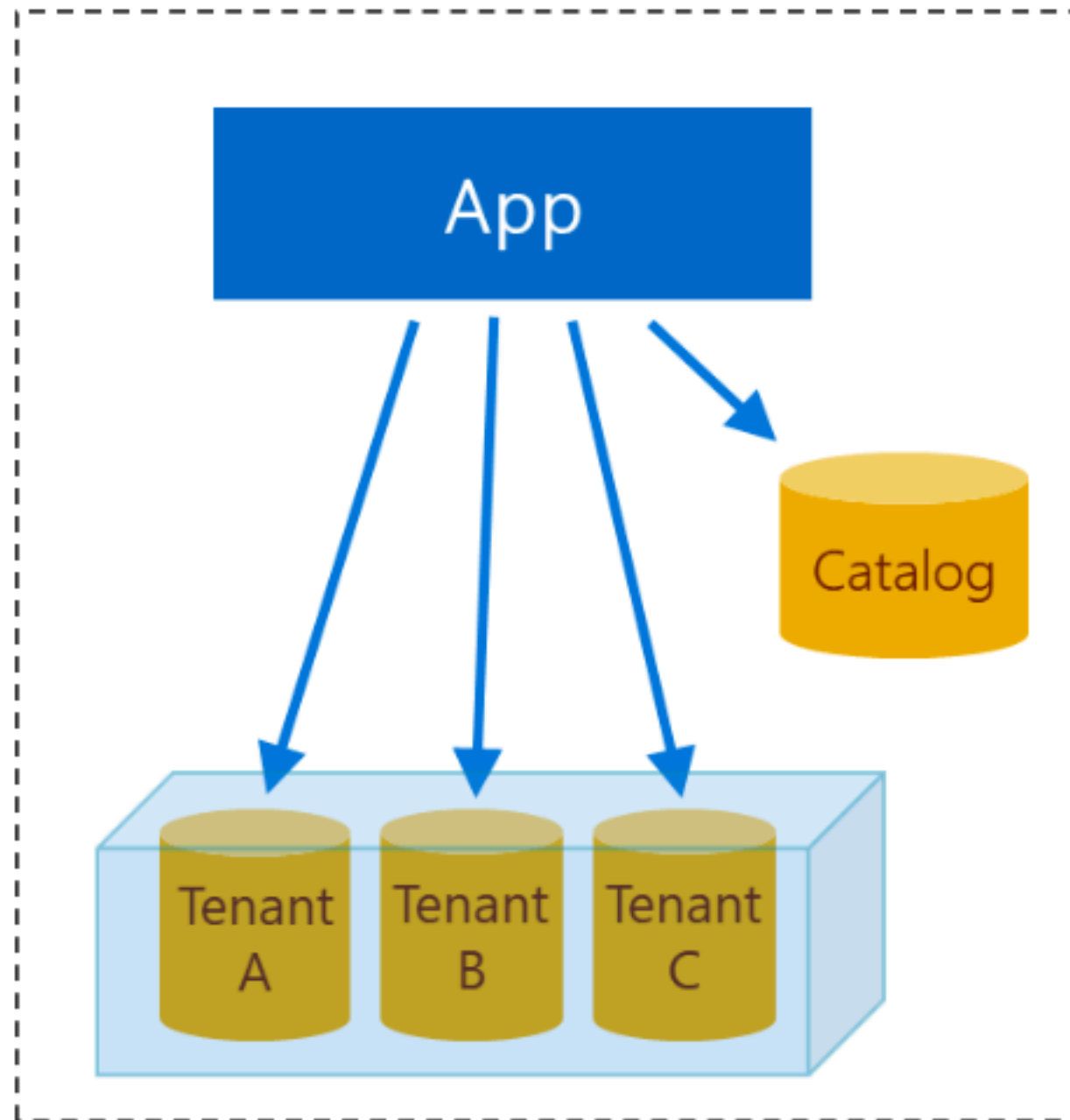


**Every tenant has a connection pool**

## Multi-tenant app with database-per-tenant

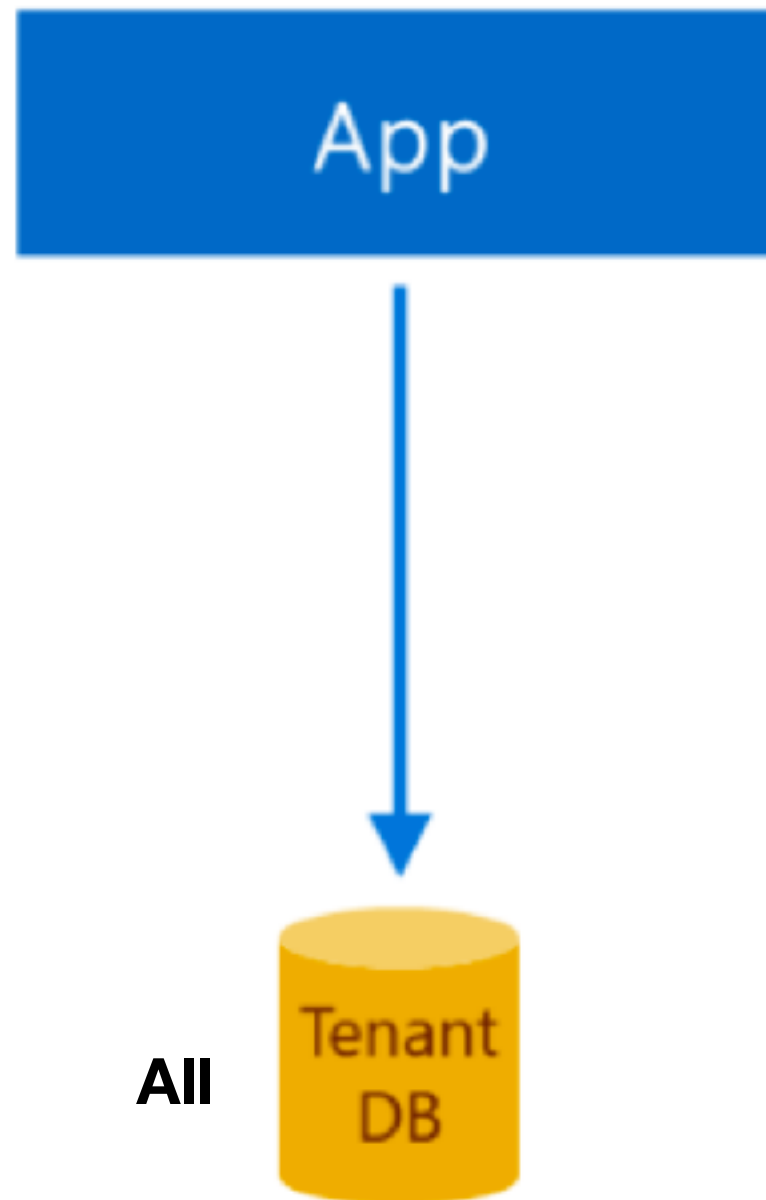


**Every tenant has a connection pool**



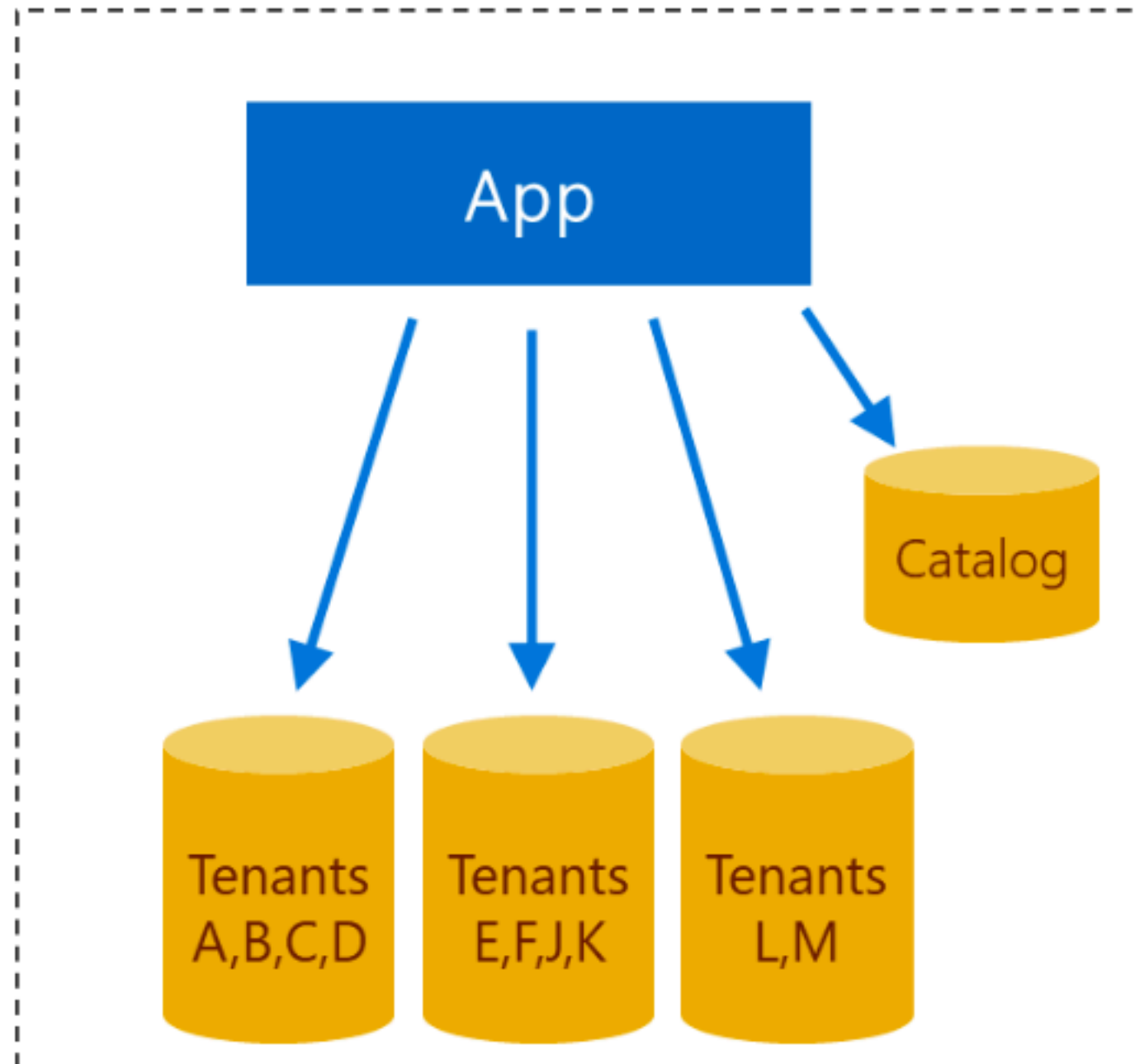
When databases are deployed in the same resource group, they can be grouped into elastic pools. The pools provide a cost-effective way of sharing resources across many databases. This pool option is cheaper than requiring each database to be large enough to accommodate the usage peaks that it experiences.

## Multi-tenant app with single multi-tenant databases



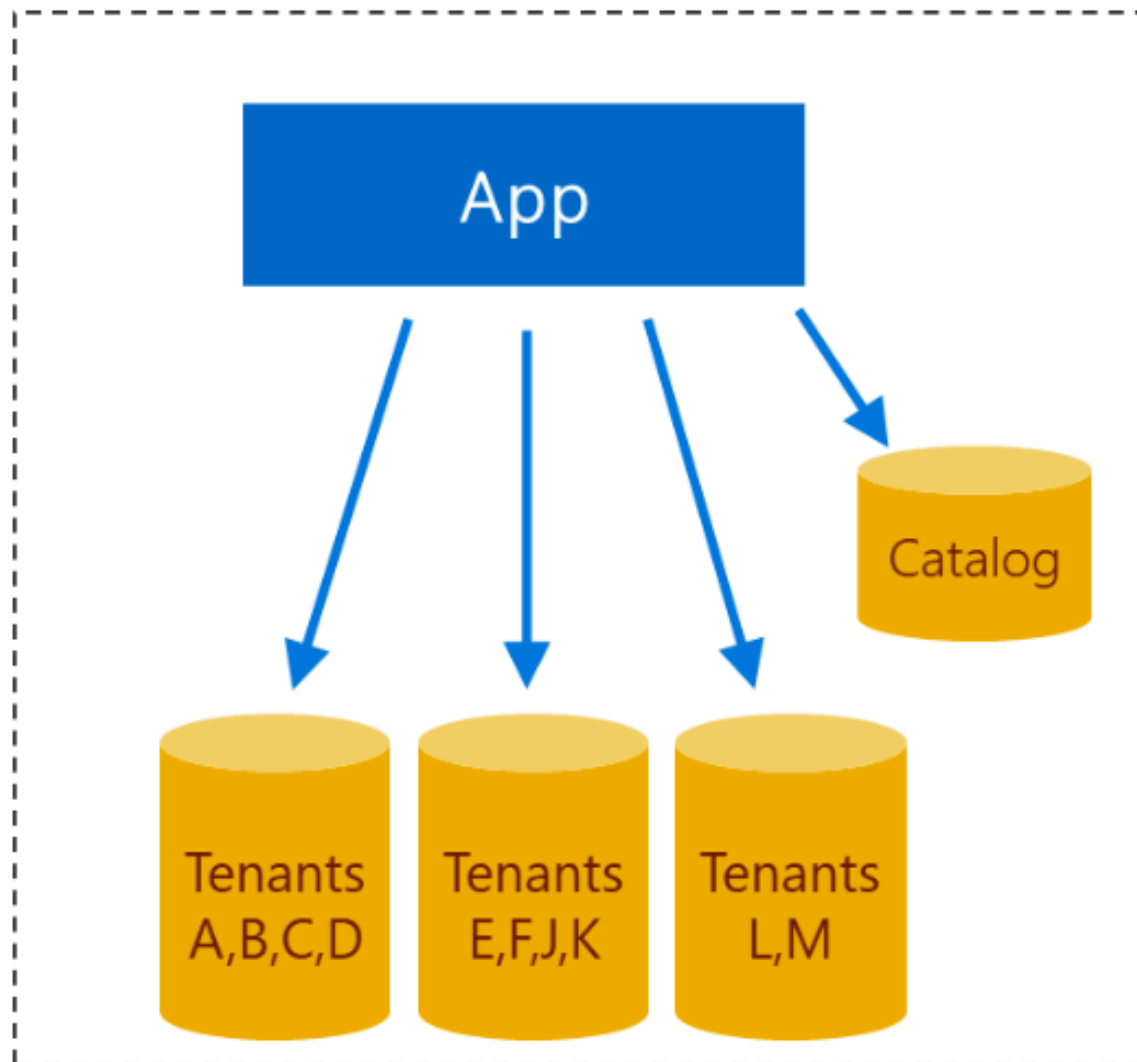
**tenant share a connection pool**

## Multi-tenant app with single sharded multi-tenant databases



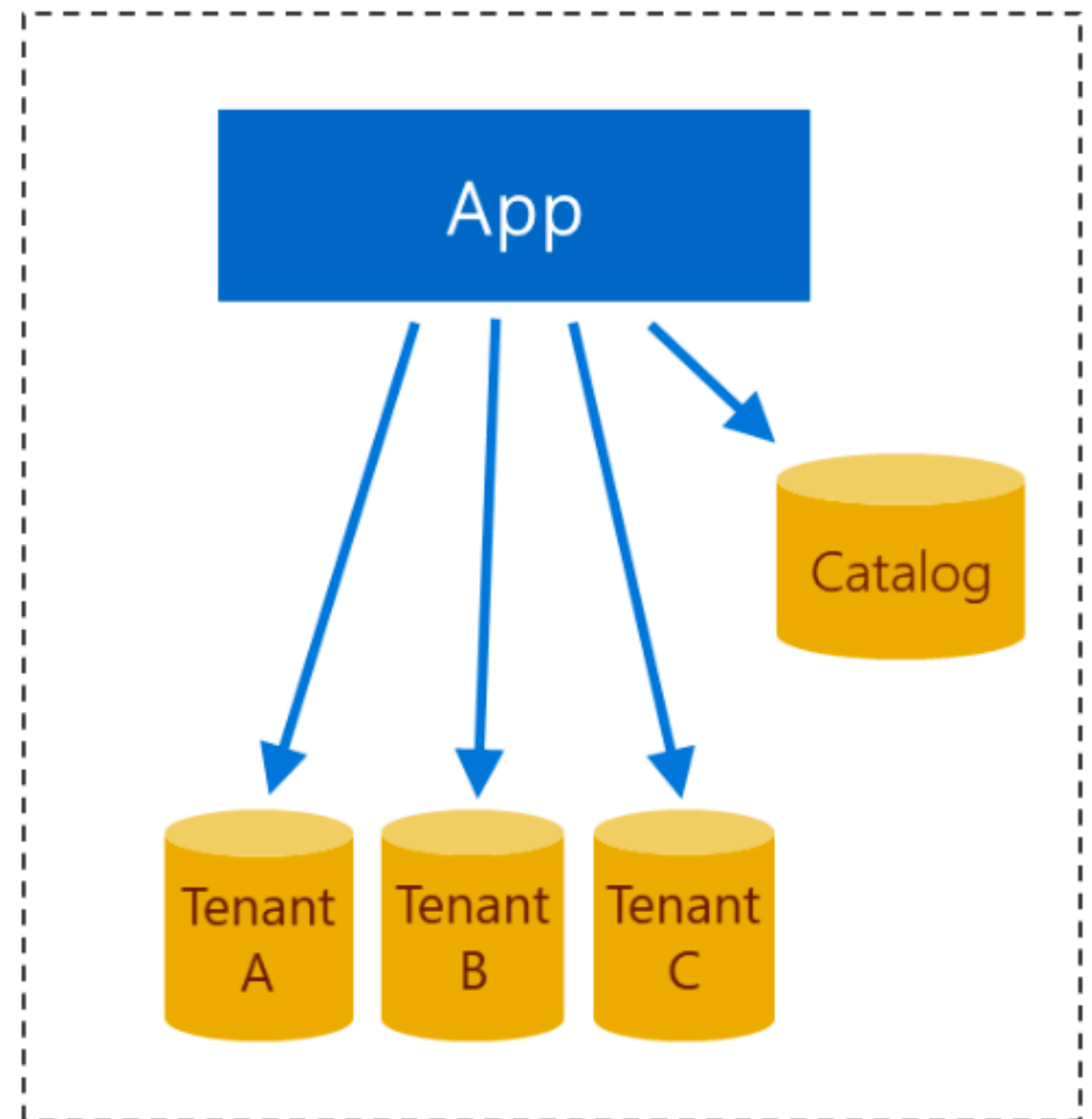
**A set tenants share a connection pool**

## Hybrid sharded multi-tenant database model

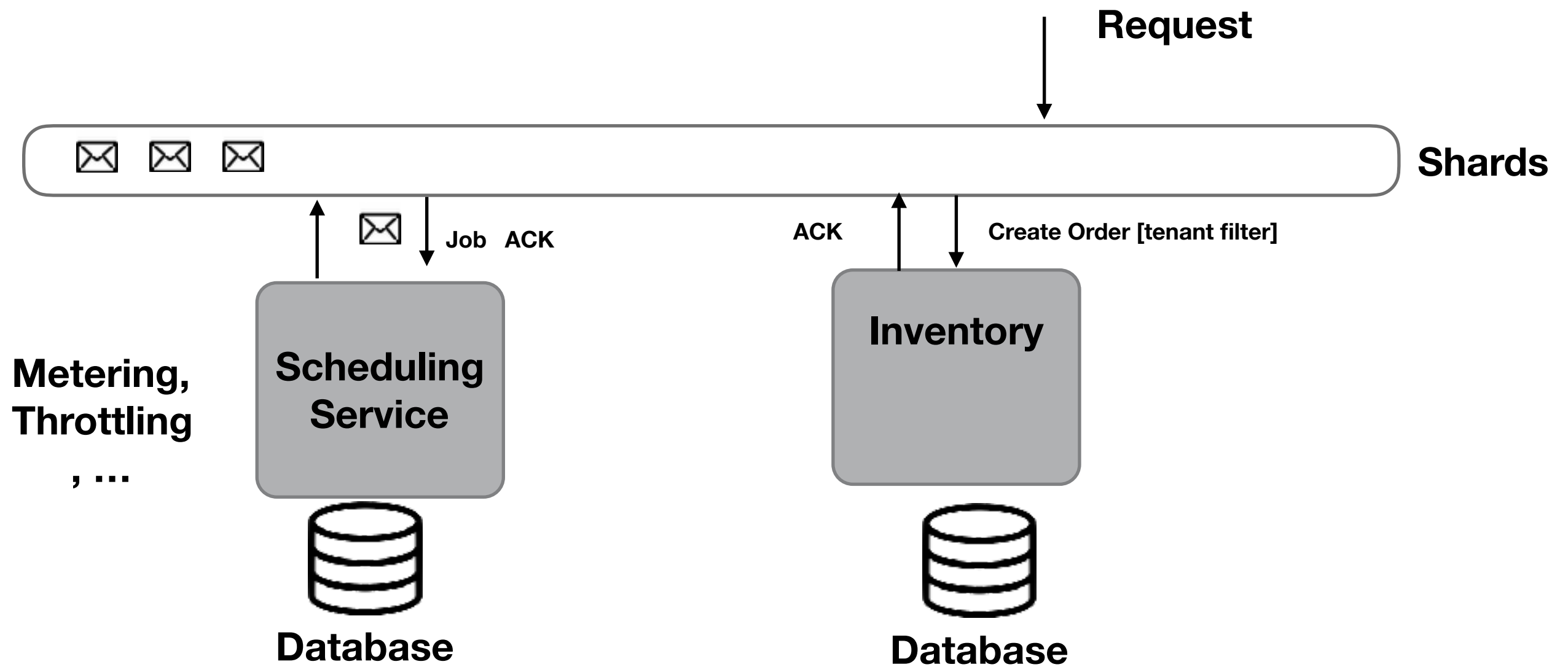


**A set tenants share a connection pool**

+

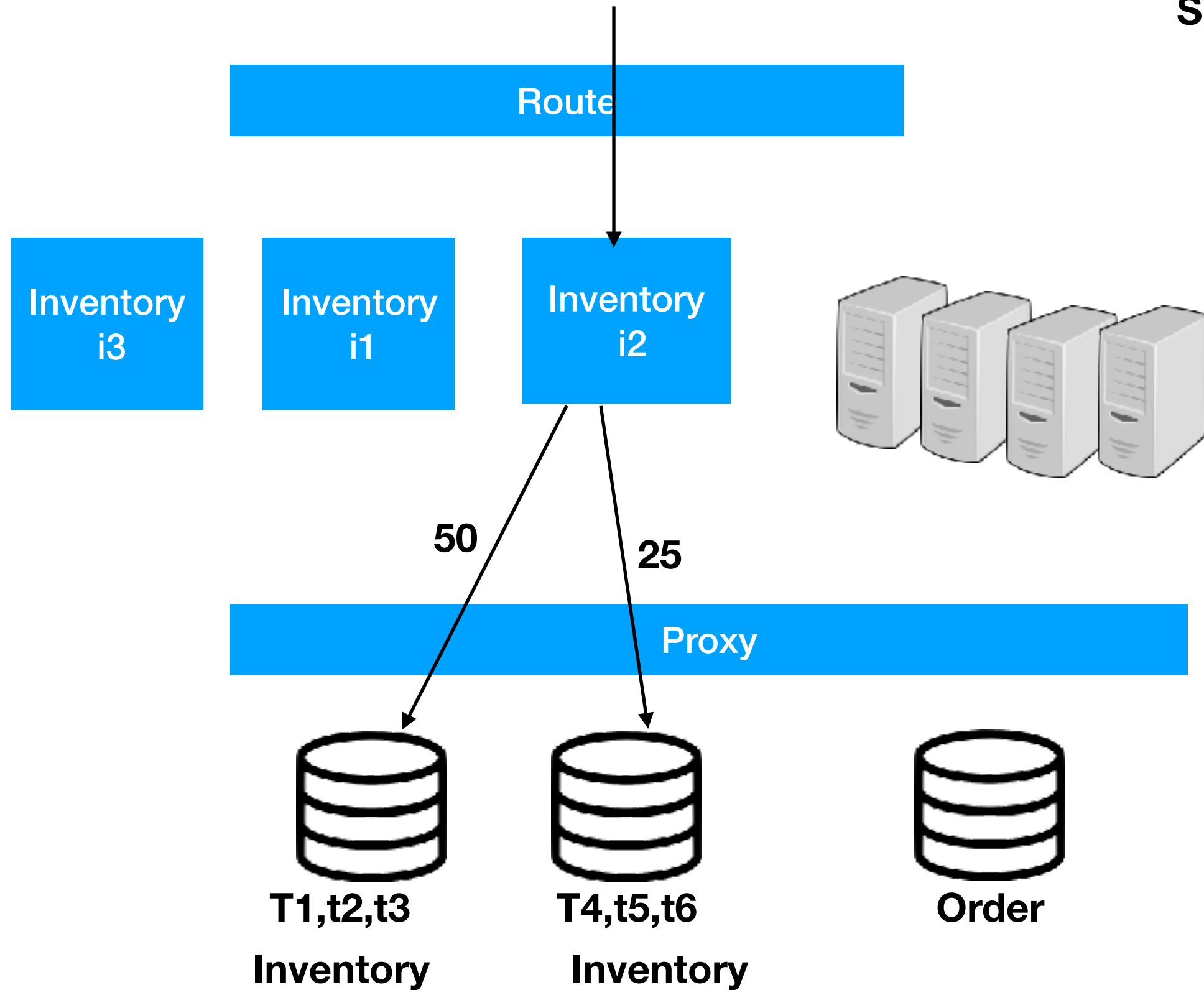


**Every tenant has a connection pool**





**Silver: max 10 users**  
**Gold : 100 users**





**T1,t2,t3**



**T4,t5,t6**



**T7,t8,t9**

# Multi Tenancy

Measurement	Standalone app	Database-per-tenant	Sharded multi-tenant
Scale	Medium 1-100s	Very high 1-100,000s	Unlimited 1-1,000,000s
Tenant isolation	Very high	High	Low; except for any single tenant (that is alone in an MT db).
Database cost per tenant	High; is sized for peaks.	Low; pools used.	Lowest, for small tenants in MT DBs.
Performance monitoring and management	Per-tenant only	Aggregate + per-tenant	Aggregate; although is per-tenant only for singles.
Development complexity	Low	Low	Medium; due to sharding.
Operational complexity	Low-High. Individually simple, complex at scale.	Low-Medium. Patterns address complexity at scale.	Low-High. Individual tenant management is complex.