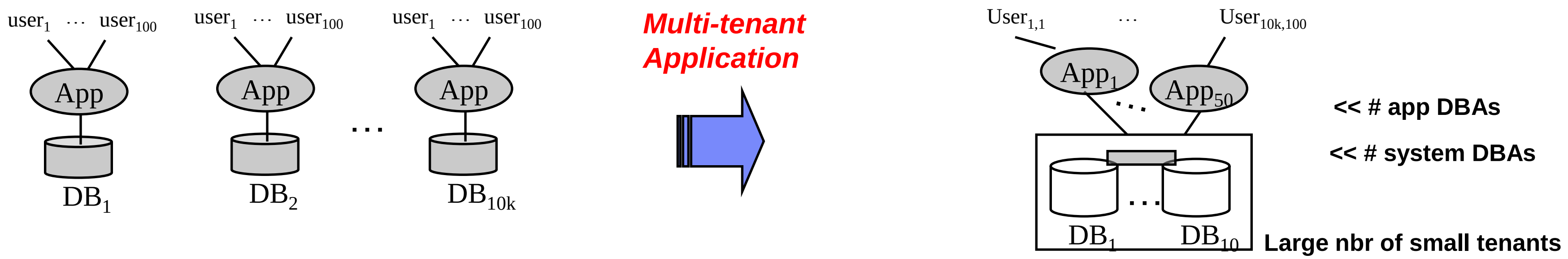
Multitenancy

Berthold Reinwald, IBM Almaden Research Center

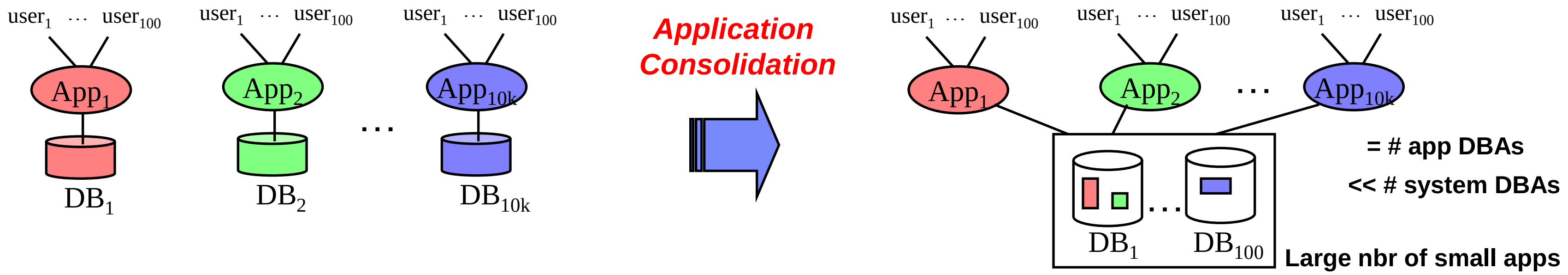
UW MSR Summer Institute, 2010

Two Use Cases for Multi-Tenancy

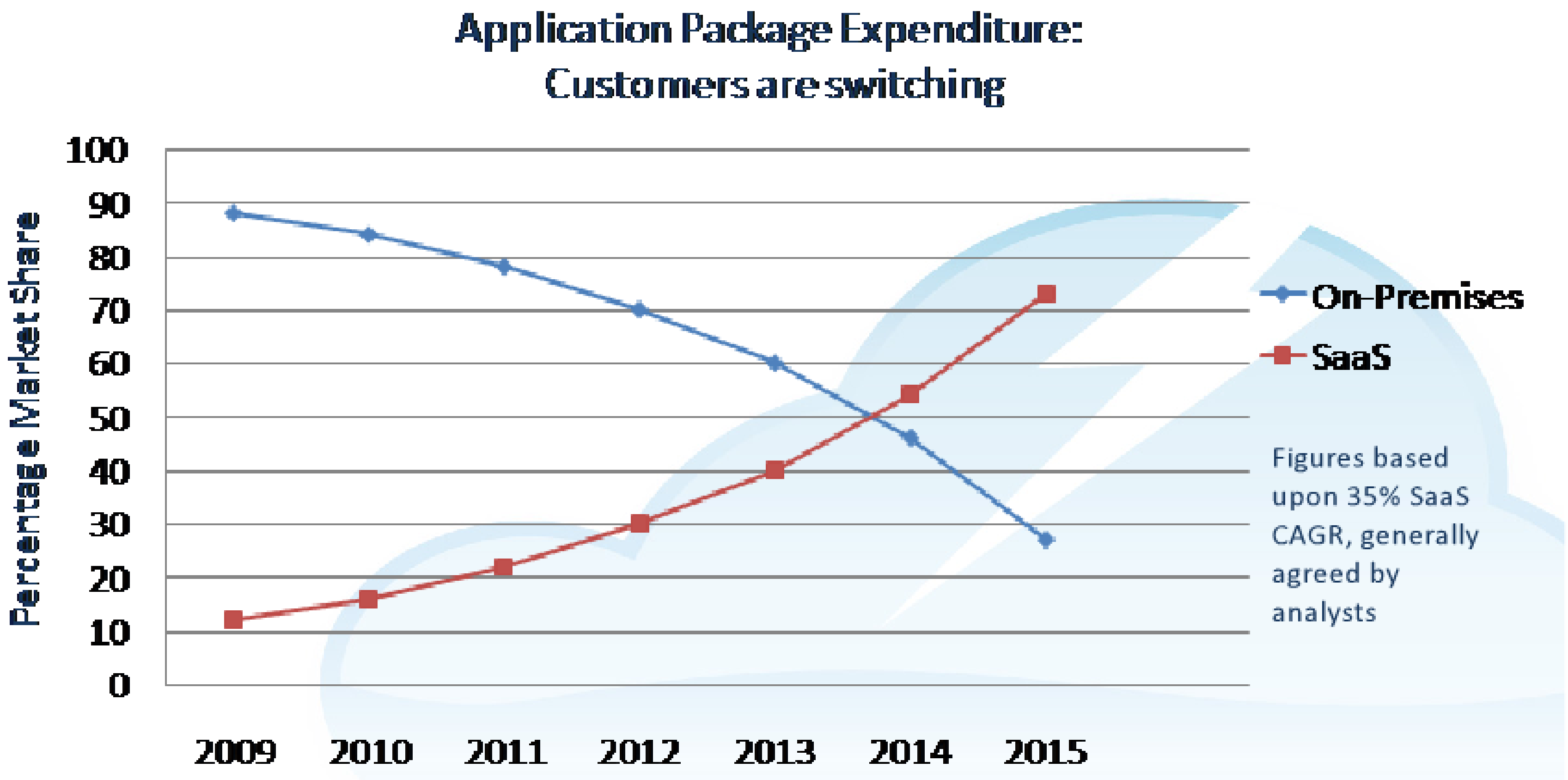
- **SaaS ISVs** (Multi-tenant Applications):
  - "Long tail of tenants"
  - very large number of small tenants using <u>same logical database schema</u>



- **Application/Database Consolidation**:
  - "Long tail of applications" (<u>with different schemas</u>)



Market Trend

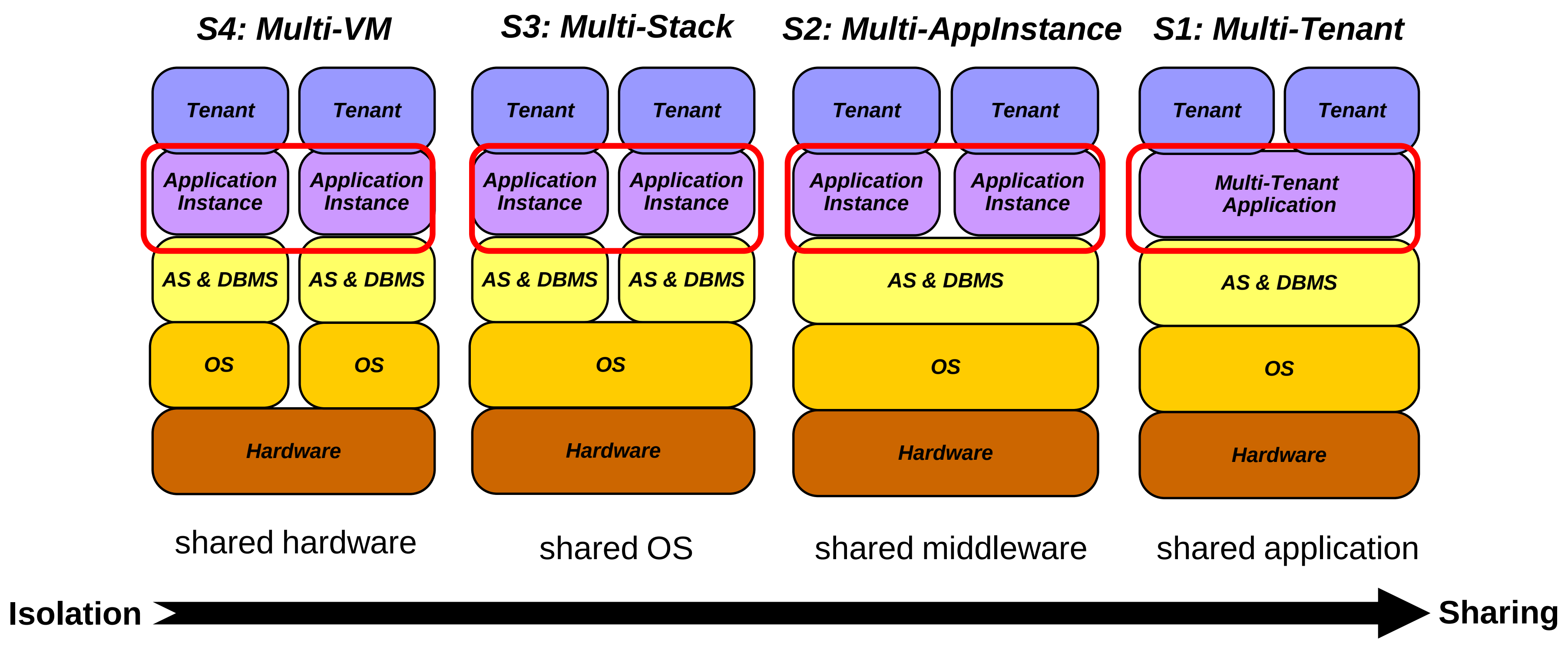


Figures based upon 35% SaaS CAGR, generally agreed by analysts

Some Observations regarding SaaS Applications

- Extremely low pricing for SMB-oriented Saas offerings
  - E.g. each food shop pays service fee $1/day, or very small amount of data
- Lower delivery costs
  - Tooling/automation/standardization in hosting center
  - By maximizing resource sharing
- Isolation
  - Data/Security, Performance (workload), system (failure), maintenance
- Customization
  - Several alternative, e.g. reserved fields, xml columns.
- SLAs
  - Ranging from app level to DB level
  - Tenants upgrade/downgrade SLAs.
- Lower developments costs for ISVs
  - transparent programming interface for multi-tenants apps
- Scalability & availability
  - Larger number of tenants (millions), incremental scale-out on low end machines w/o impacting existing service

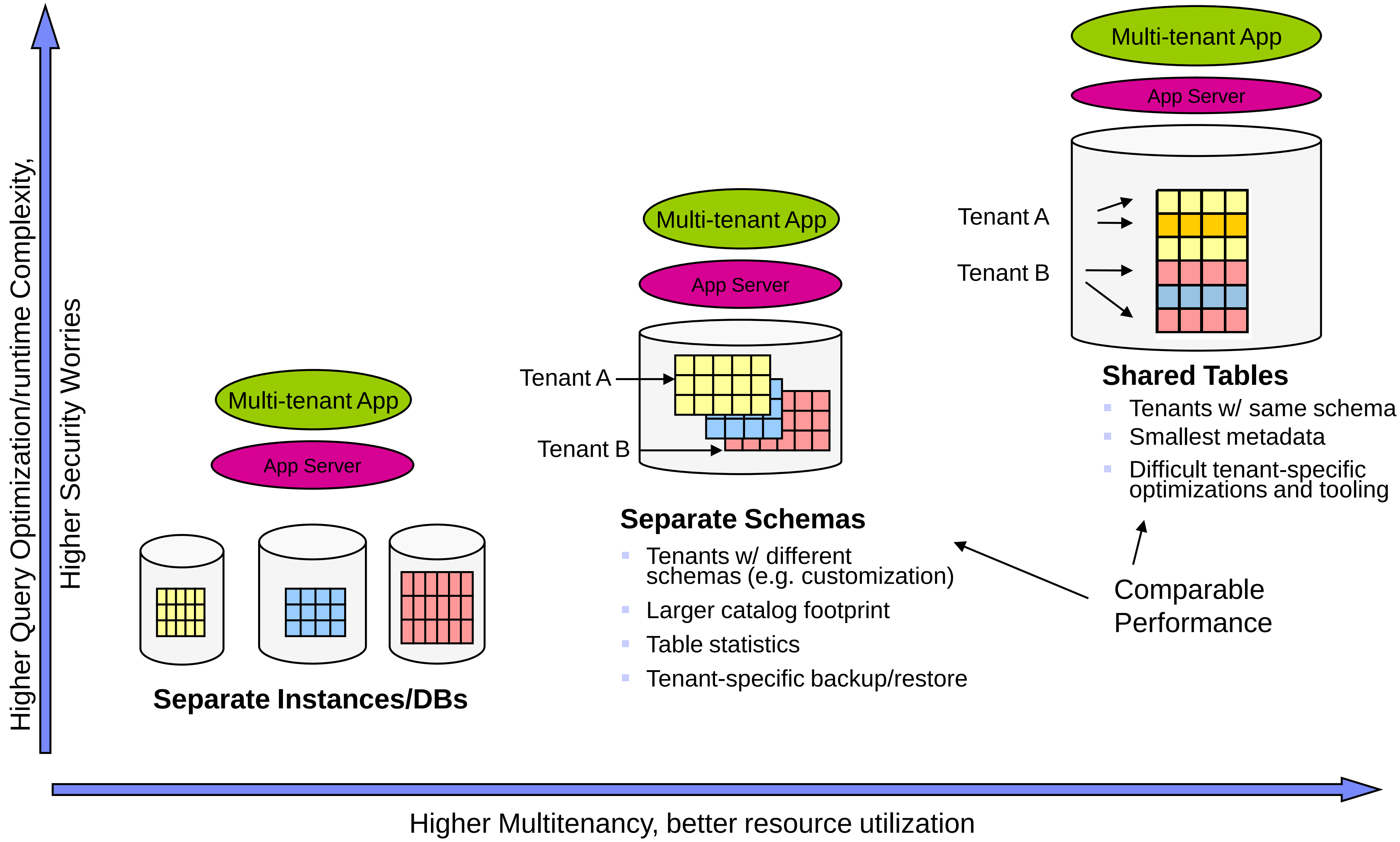
Multi-instance single-tenant applications vs. single-instance multi-tenant applications



From Single-tenant to Multi-tenant:

- ❑ Isolation and customization
- ❑ Application Time-to-Market
- ❑ Economy of multi-tenancy

Database Multi-Tenancy Models



Separate Instances/DBs

Separate Schemas

- Tenants w/ different schemas (e.g. customization)
- Larger catalog footprint
- Table statistics
- Tenant-specific backup/restore

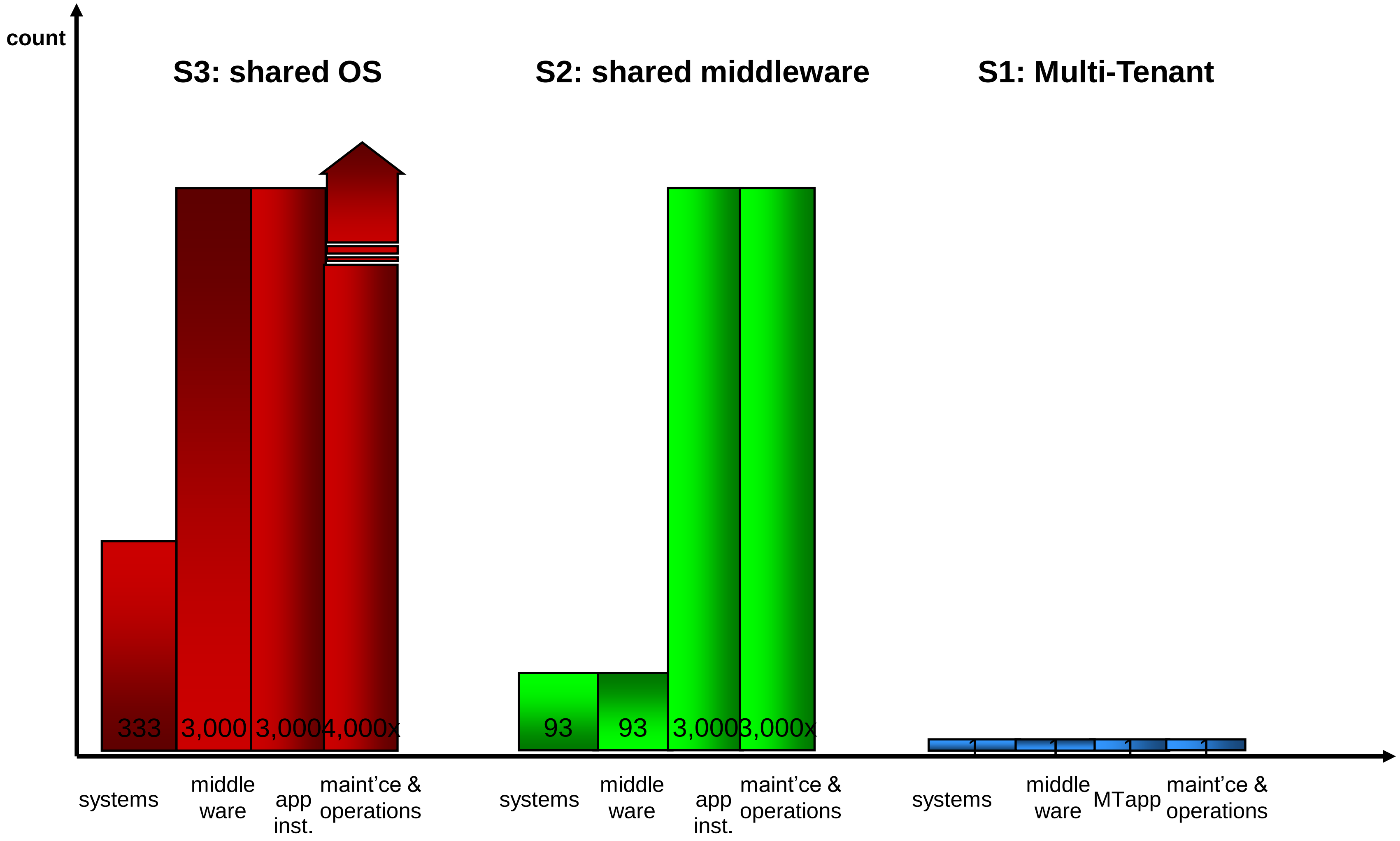
Shared Tables

- Tenants w/ same schema
- Smallest metadata
- Difficult tenant-specific optimizations and tooling

Economy of Multi-Tenancy - Scale-Out Quantification -

Tenant Characteristics	100 registered users per tenants 5% active user ratio (5 users)		
Active tenant ratio	10%		
	S3: Shared OS	**S2: Shared Middleware**	**S1: Multi-Tenant**
# concurrent tenants (footprint/tenant)	**9** → due to memory footprint	**32** → due to memory footprint	**300** → due to performance bottlenecks
# registered tenants	**9** → inactive tenants <u>consume</u> runtime resources	**32** → inactive tenants <u>consume</u> runtime resources	**3,000** →inactive tenants <u>don't consume</u> runtime resources
Scaling Nbr of Tenants	1x	3x	**300x**

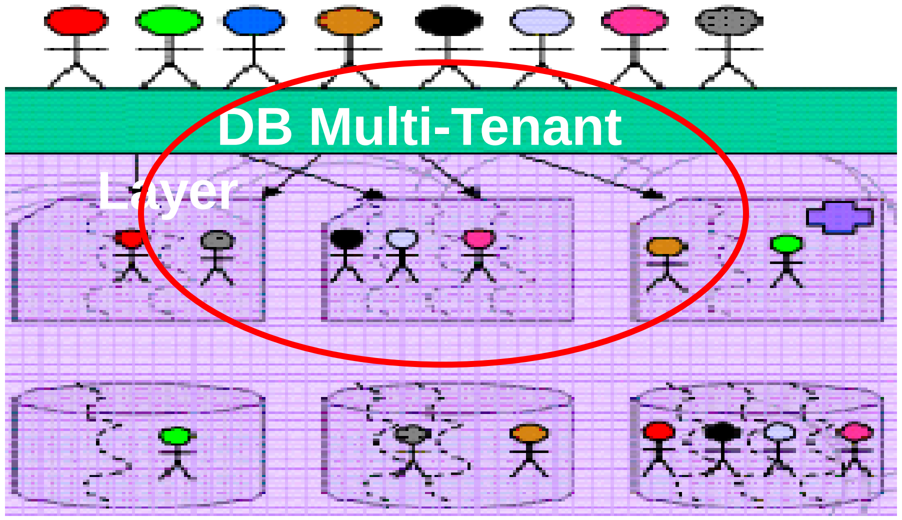
Economy of Multi-Tenancy
- Cost Savings Analysis -

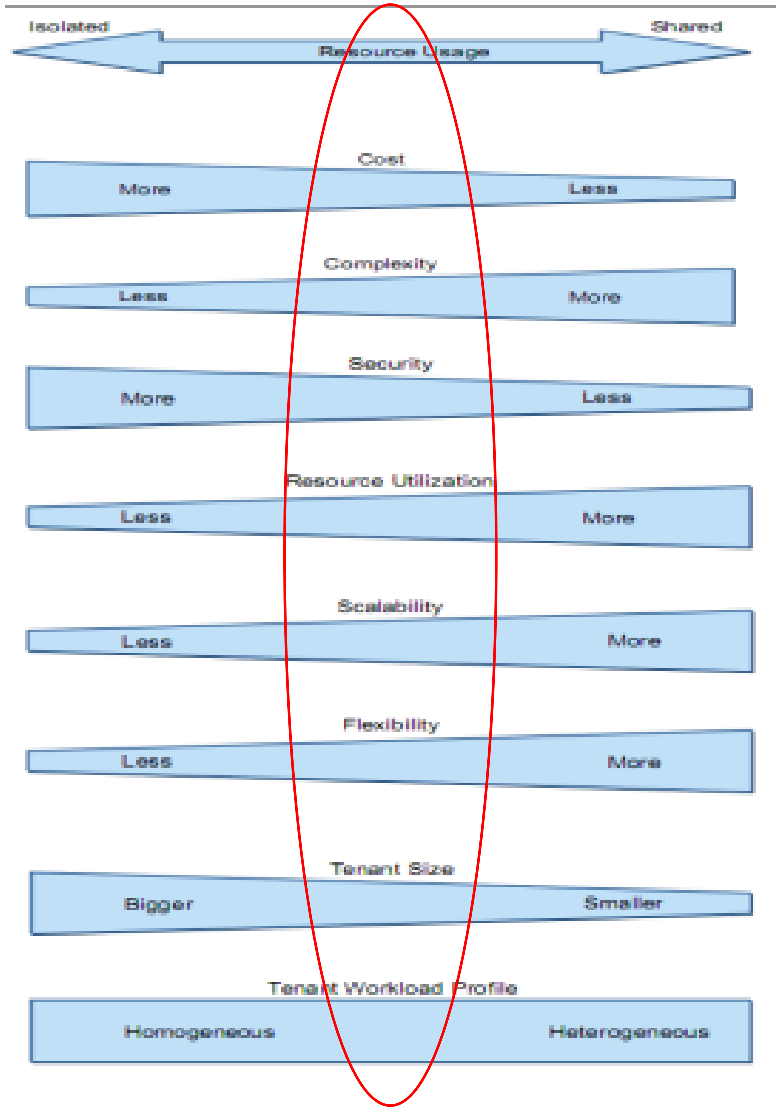


Multi-Tenancy Comparison

		Single-tenant Apps	Multi-tenant Apps
Time to Market	**App development**	▪ deployment	▪ transformation or new developmt ▪ Limited customization
	Tenant on boarding		▪ through tenant subscription
Isolation	**Security** **Performance** **Availability** **Maintenance**	▪ Isolation per Instance	▪ Isolation in application and DB ▪ Row & schema & system level Isolation ▪ Governance ▪ Load balancing & Sharding ▪ High availability & Fault tolerance
Scaling	**Registered Tenants**	▪ 1~3x	▪ 300x
Cost	**HW, SW, daily op's & maint'ce**	▪ 200~400x	▪ 1x

Multi-tenancy Challenges





Isolation, Scalability, Performance, Customization, Resource Utilization, Metering …

Research Challenges

- High Availability and Failover and Load Balancing
    - Large number of instances/databases
    - At the database level, or below the database
- Distributed Fabric
    - Many different levels of failure detection
    - Scale out
- Isolation: data, performance, system, maintenance
- Customization
- SLAs for availability and performance
- Benchmarks
    - Footprint/tenant, cost/tenant