



Service Variability in Multi-Tenant Engineering: A Systematic Literature Review on the State of Practice, Limitations and Prospects

**Ouh Eng Lieh
Senior Consultant
Institute of Systems Science, NUS
englieh@nus.edu.sg**



Credits

- Lee Chuk Munn (ISS Staff)
- YungHans Irawan (ISS Staff)
- Anshuman Mathur (Postgraduate Student)
- Jithesh Prabhakaran (Postgraduate Student)
- Samir Kumar (Postgraduate Student)
- Sathish JR (Postgraduate Student)
- Saurabh Saste (Postgraduate Student)
- Venkatesh Dasari (Postgraduate Student)

The value you can get today

Multi-Tenancy and
Service Variability

Service
Architectural
Choices and
Models

Experience
Reports





Stakeholders Concerns

Service Provider

- Market share
- Cost and Revenue
- Maintainability
- Scalability
- Share Resources

Tenant

- Startup cost and time
- Maintenance cost
- Isolation Level

Variability in Multi-Tenant Engineering

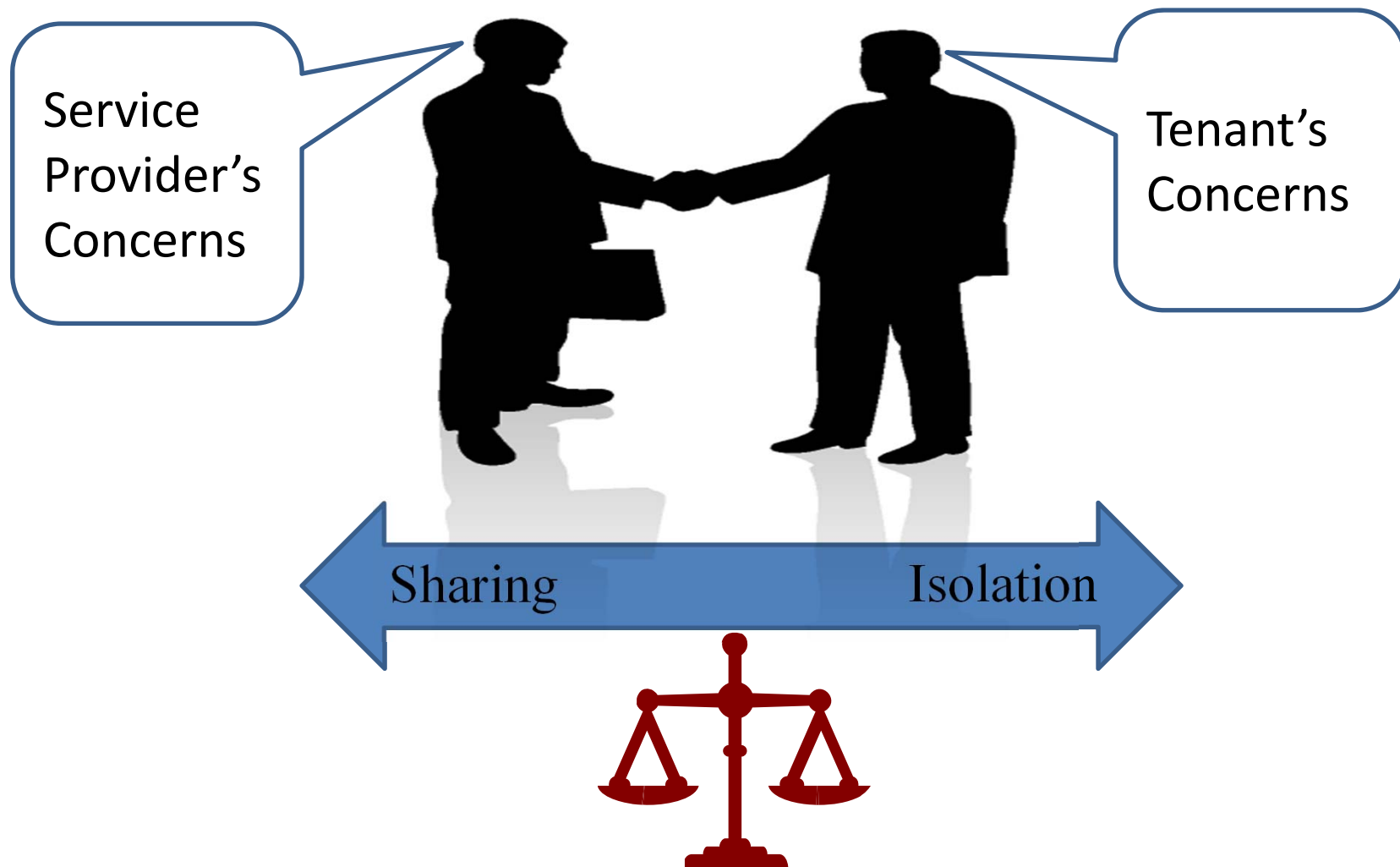
Functional

- Look & Feel
- Business Logic
- Business Workflow
- Data Persistence and Storage

Quality

- Security
- Privacy
- Performance
- Usability
- Maintainability

Service Variability





Service Architectural Choices

Service Binding

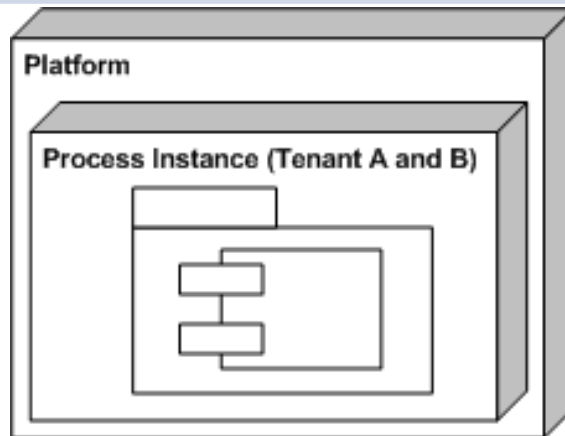
Service Packaging

Service Hosting

Service Hosting

Shared Instance

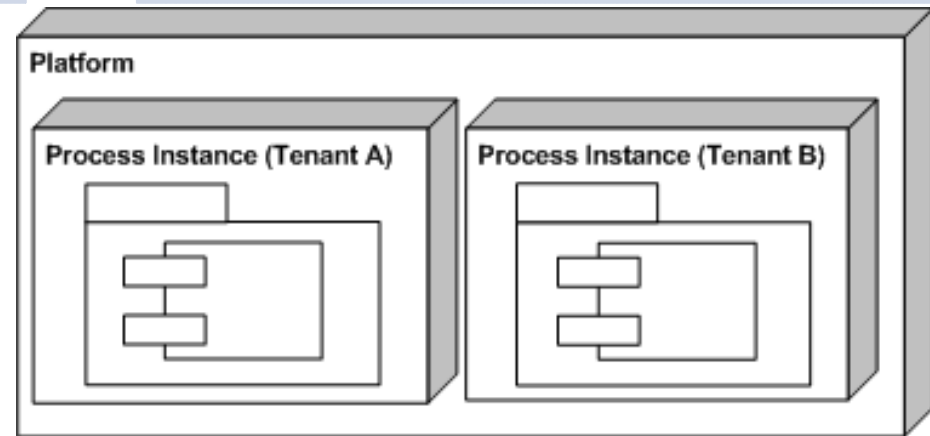
- 1 process instance for **many** tenants



Shared Instance

Dedicated Instance

- 1 process instance **per** tenant

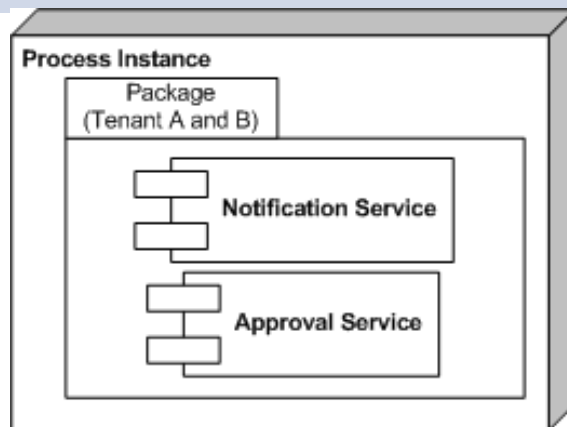


Dedicated Instance

Service Packaging

Service Level Packaging

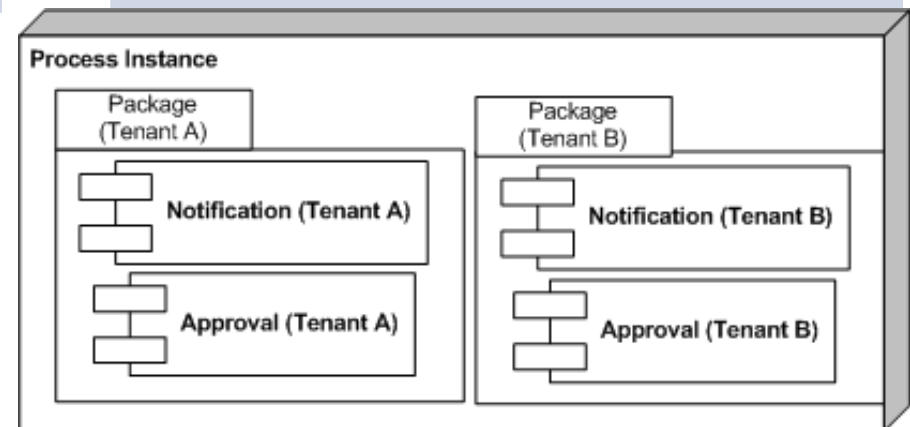
- Components are packaged as services to serve **any** tenants



Service Level Packaging

Tenant Level Packaging

- Components are packaged to serve **specific** tenants



Tenant Level Packaging

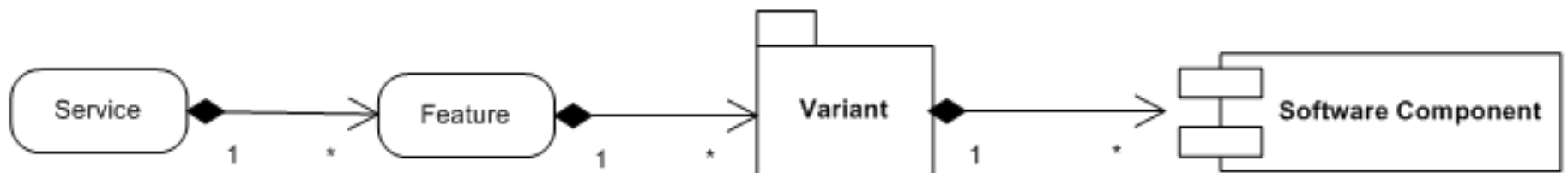
Service Binding

Static Binding

- During design, pre-compilation E.g. UML Design, MDA
- During compilation, linking and assembly time E.g. `#ifdef`

Dynamic Binding

- During deployment E.g. configuration files
- At runtime E.g. dynamic libraries



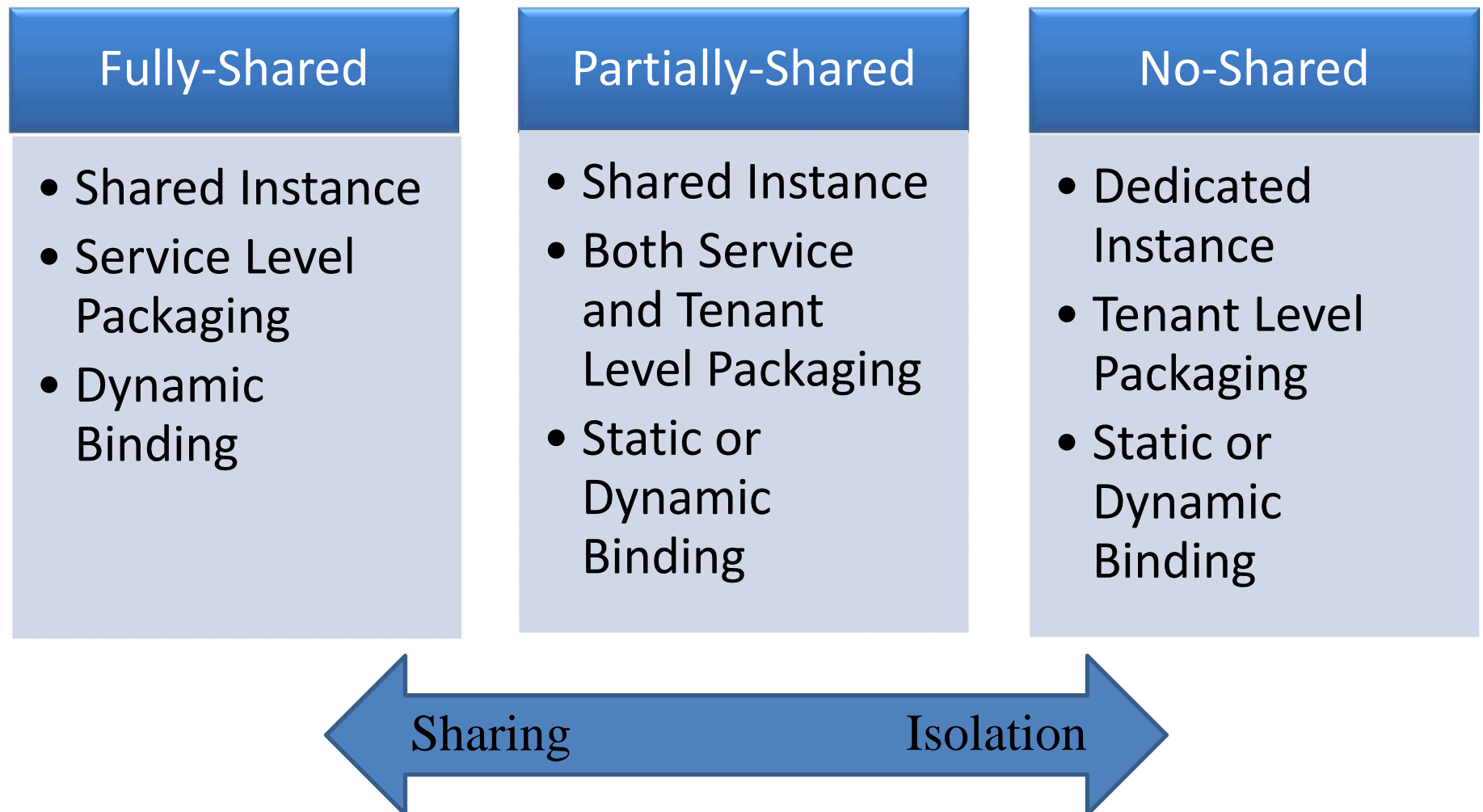


Dynamic Binding

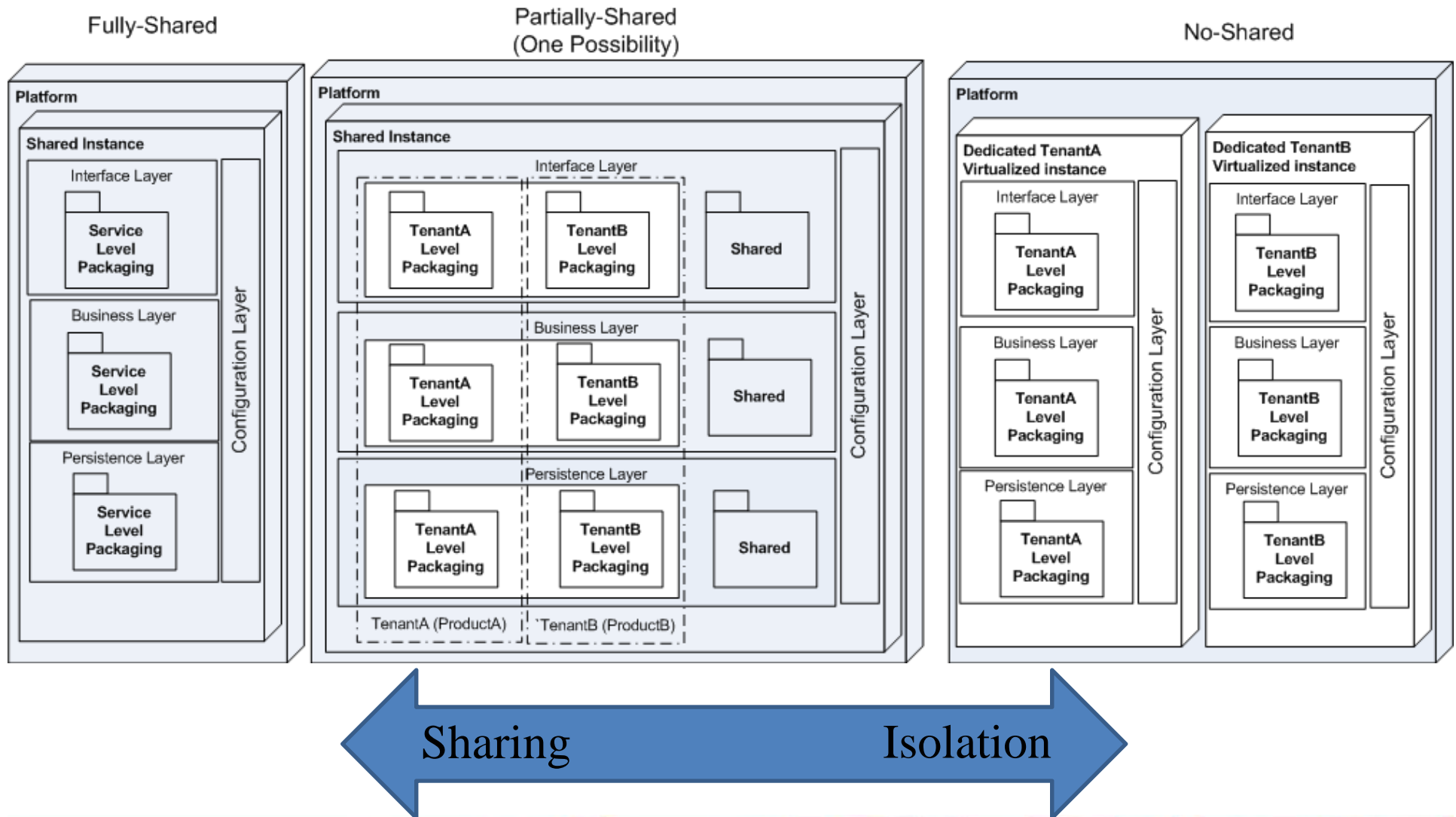
Abstraction, Configuration & Parameterization

- Template and Configuration
- Aspect Oriented - Variants as aspects and aspect weaver to weave aspects into the service kernel
- Service Oriented - Formally specify business process and interaction protocols and integrate to variants via registry-based

Service Architectural Models

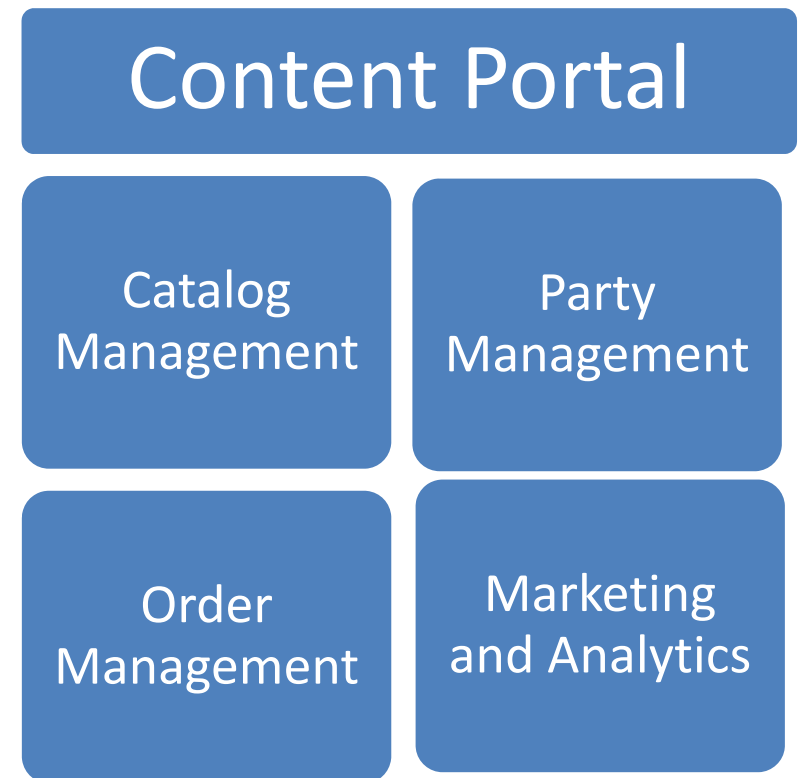


Service Architectural Models



Experience Report (Multi-Tenant Migration)

- SCRUM
- Sprint 0 – Derive the feature model and baseline architecture for multi tenant content portal application





What happen...

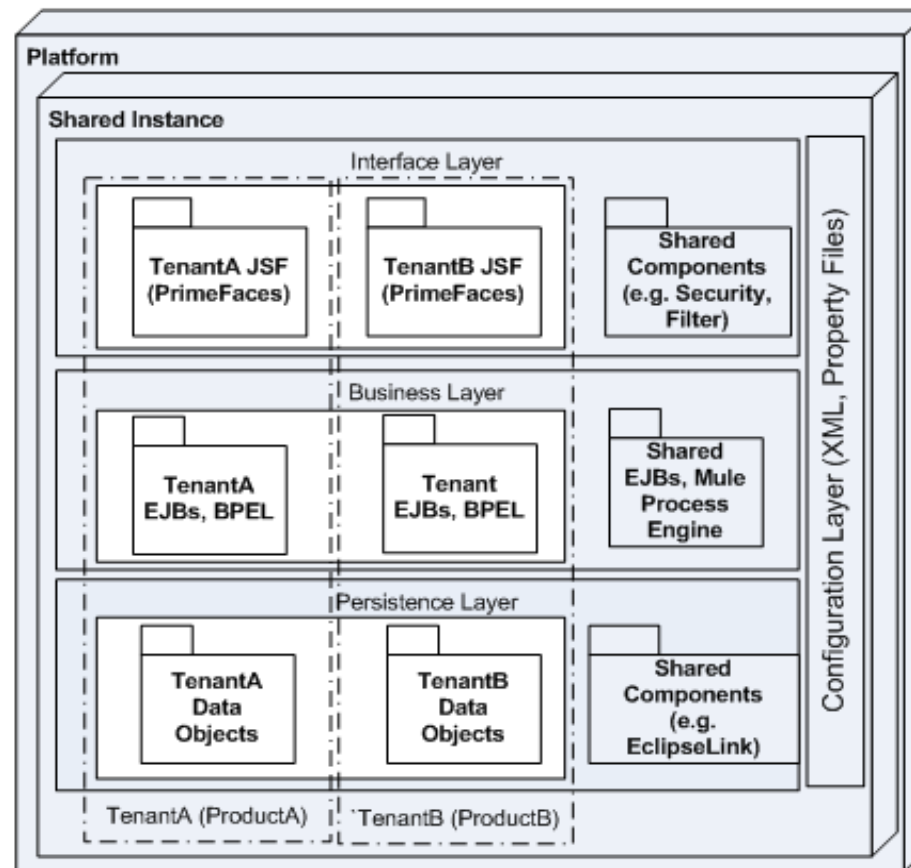
- Challenging to determine the scope of what is configurable/sharable in the initial sprint.
- Started with fully-shared model, ended with partially-shared model.
- Interface and persistence layers should be tenant level packaged for better isolation, customization and maintenance.
- Business layer contains the highest sharing of resources.

Experience Report (Multi-Tenant Migration)

Feature Design
& Implementation



Deployment



Experience Report (Tenant On-boarding)

- Key Questions focus on
 - Architectural model adopted to support multi-tenancy
 - Key issues in tenant onboarding and service evolution



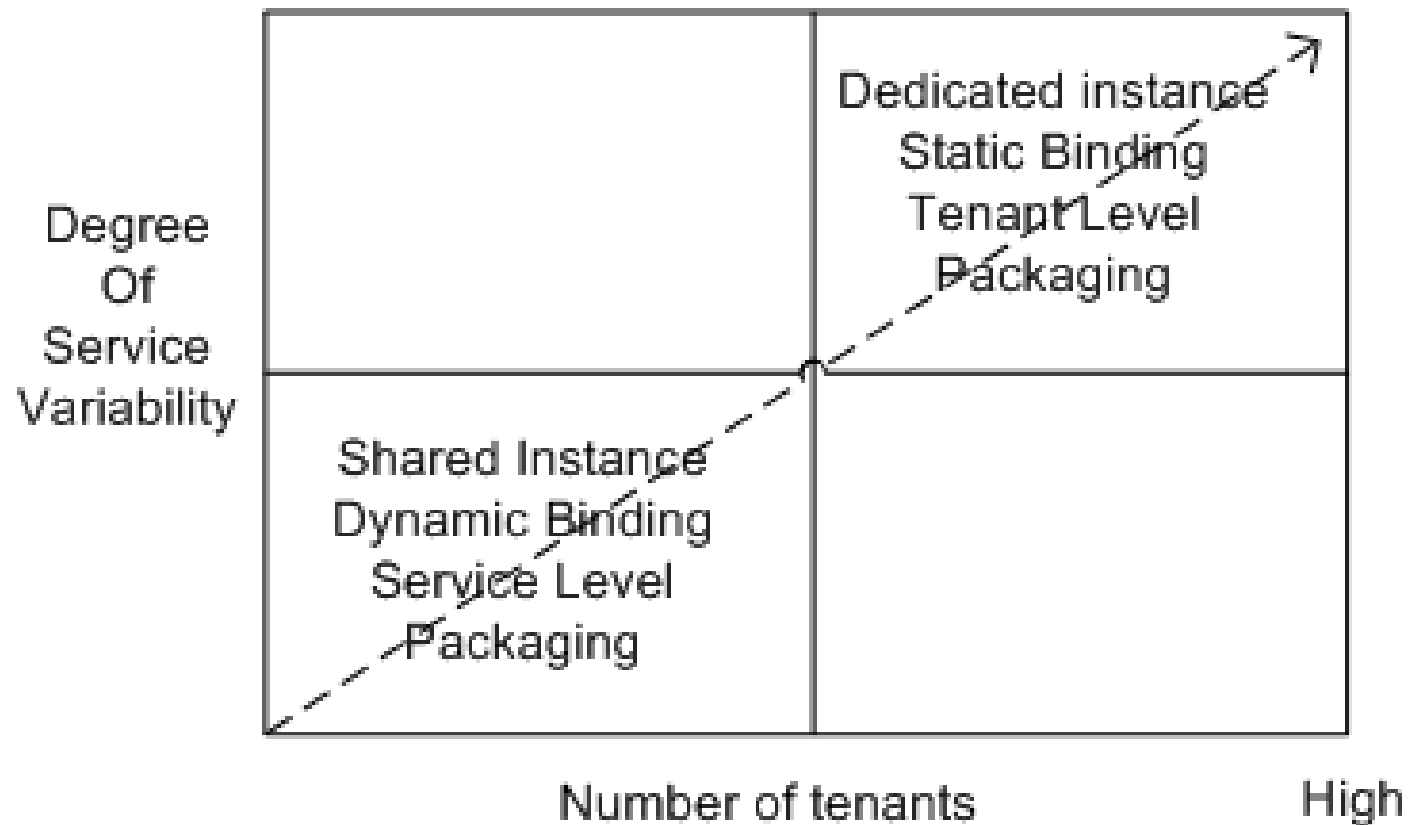


Key Summary Points

- 73% of the projects start with the fully-shared model with dynamic binding based on the service oriented architecture.
- 55% of the projects encountered “unfit” during new tenant on-board in a fully-shared model especially for domains where the requirements of new tenant varies widely or the requirements of existing tenants are changing often.
- Adoption of hybrids among these models are also very common. The tenant placement algorithm is to be extended to choose the appropriate model within the hybrid during service provisioning .

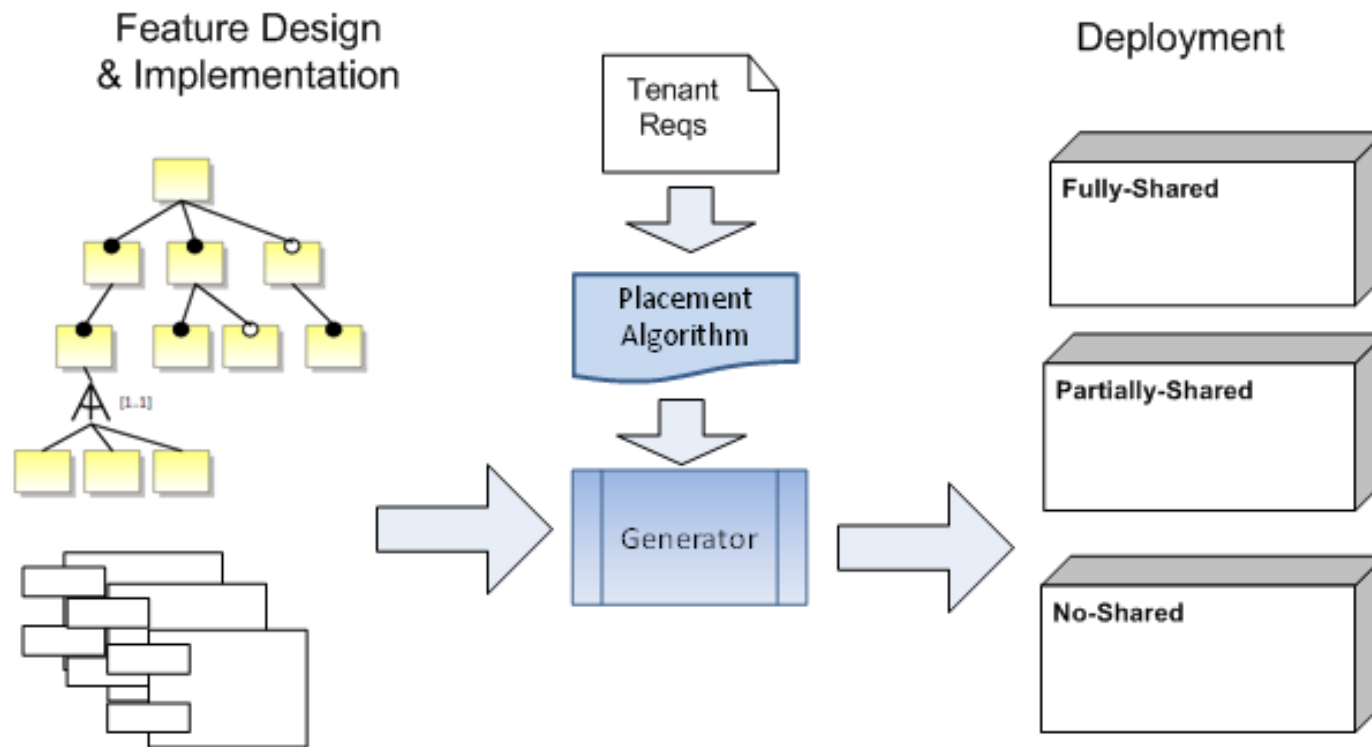
Key Takeaways

- One size does not fit all

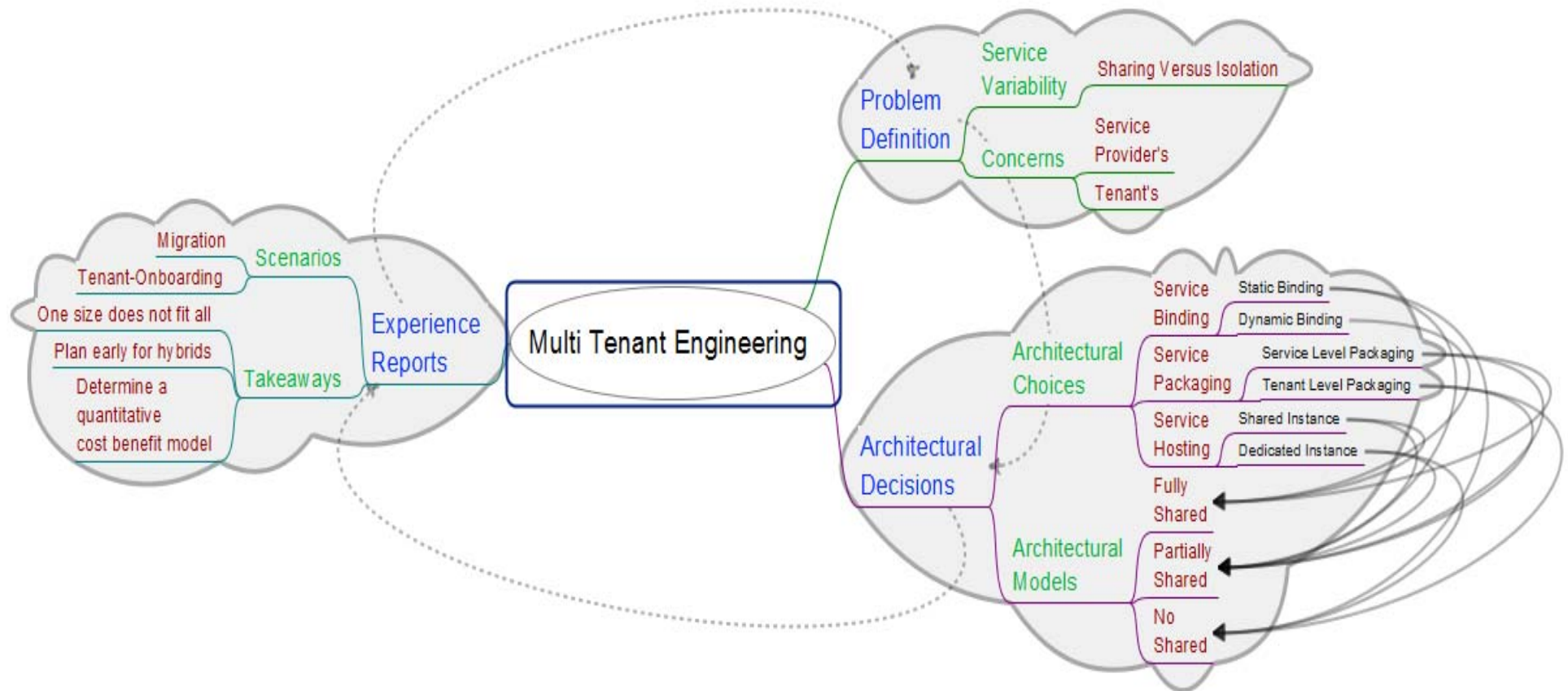


Key Takeaways

- Early provision for hybrids of architectural models



Summary





THANK YOU!

Q & A