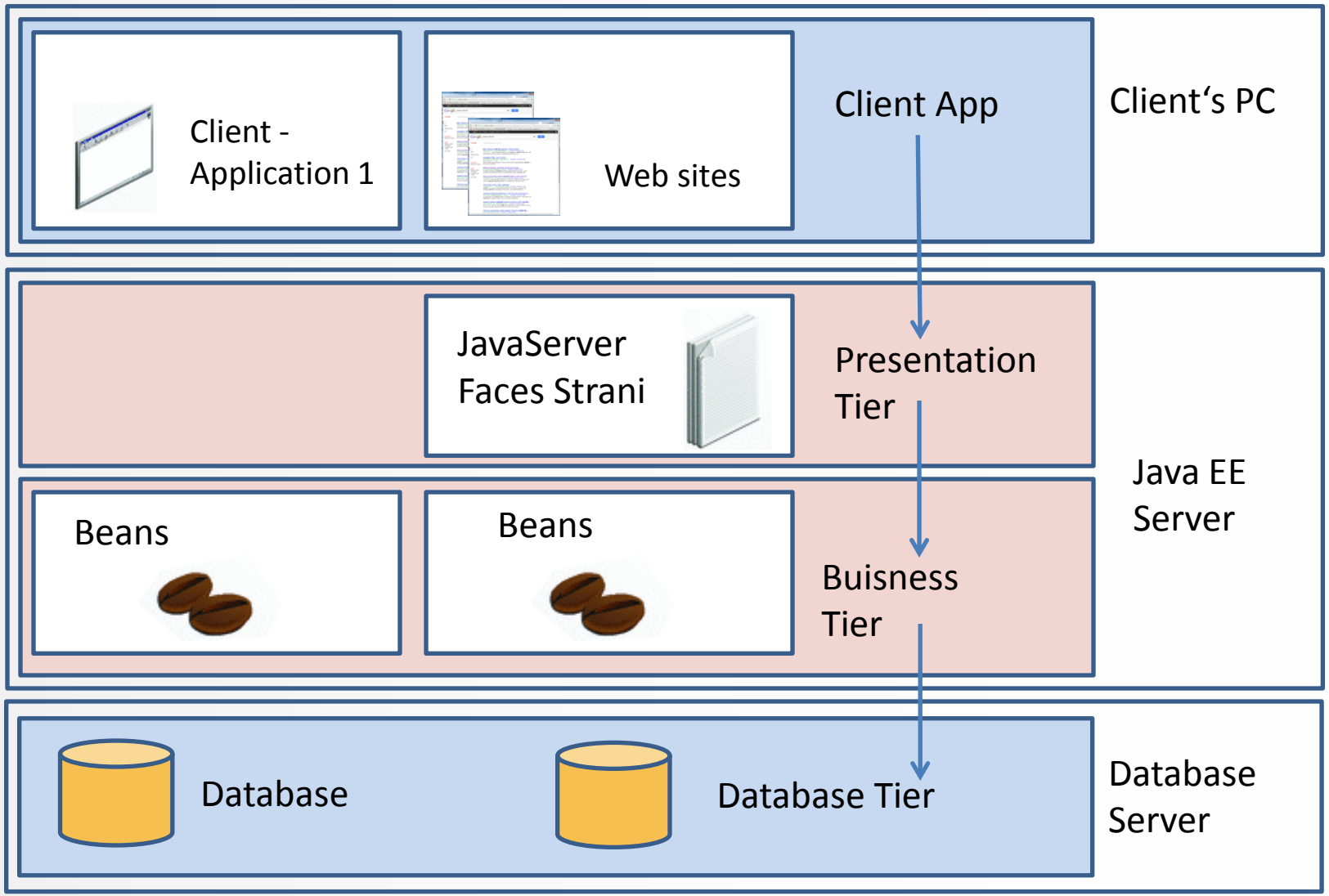
Towards a cloud-enabled Java EE platform

Rok Povše, prof. dr. Matjaž Branko Jurič

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko
Laboratorij za integracijo informacijskih sistemov
Kompetenčni center za SOA www.soa.si
Center za računalništvo v oblaku www.cloud.si

Java EE


Client -
Application 1
Web sites
Client App
Client‘s PC
JavaServer
Faces Strani
Presentation
Tier
Beans
Beans
Buisness
Tier
Java EE
Server
Database
Database Tier
Database
Server

Cloud Computing

- Paradigm for provisioning computing resources
- The most attention was recently given to:
  - Infrastructure as a Service (IaaS),
  - Platform as a Service (PaaS),
  - Software as a Service (SaaS)
- For this work, platform is the most relevant resource as a service
- Existing well-known PaaS providers:
  - Windows Azure
  - Google App Engine
  - Elastic Beanstalk

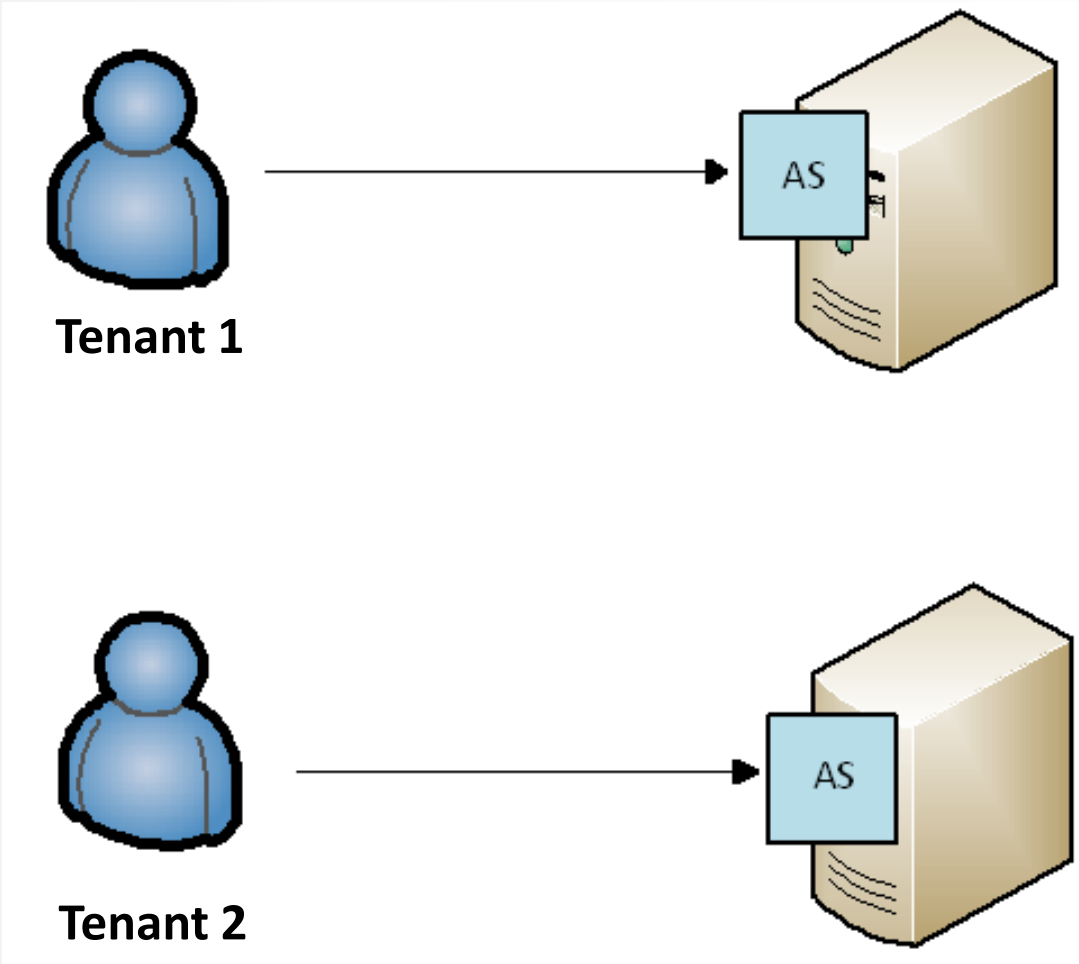
Java EE in the cloud

- Current Java EE containers are not fully architected to execute applications in the cloud.
  - No support for: **multitenancy**, **elasticity** and **monitoring**
  - Multitenancy: Principle of sharing resource among several tenants
  - Elasticity, scaling: Meeting the demand by scaling-up resources
  - Monitoring: Monitoring usage, scaling, SLA/SLO…

- Methodology:
  - Identifying models of multitenant environment
  - Identifying metrics to support scaling
  - Identifying monitoring metrics
  - Defining parameter system for identified metrics
  - Defining Java meta-data based on parameter system
  - Validate meta-data system

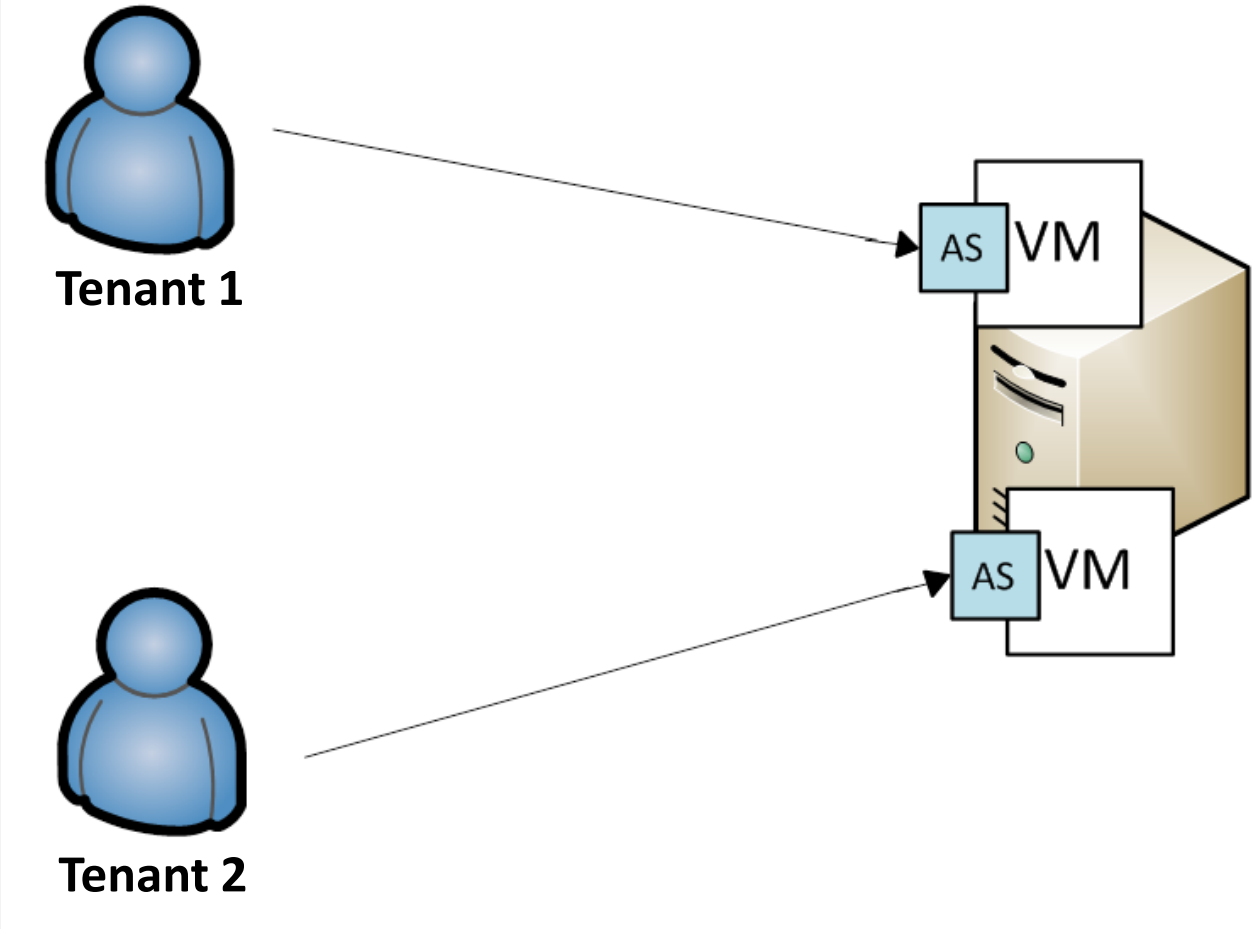
Multitenancy models

- Model type 1



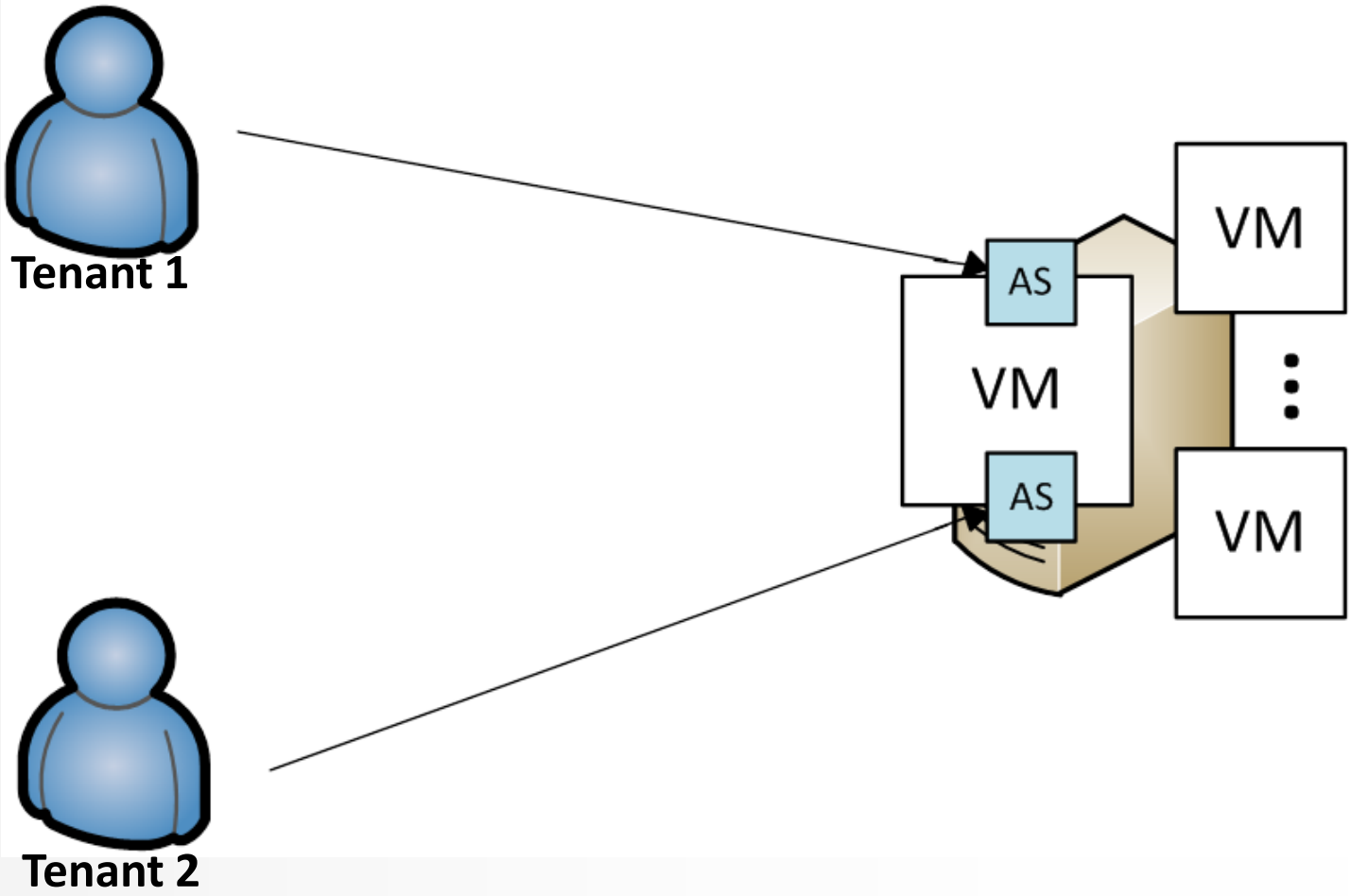
Multitenancy models

- Model type 2



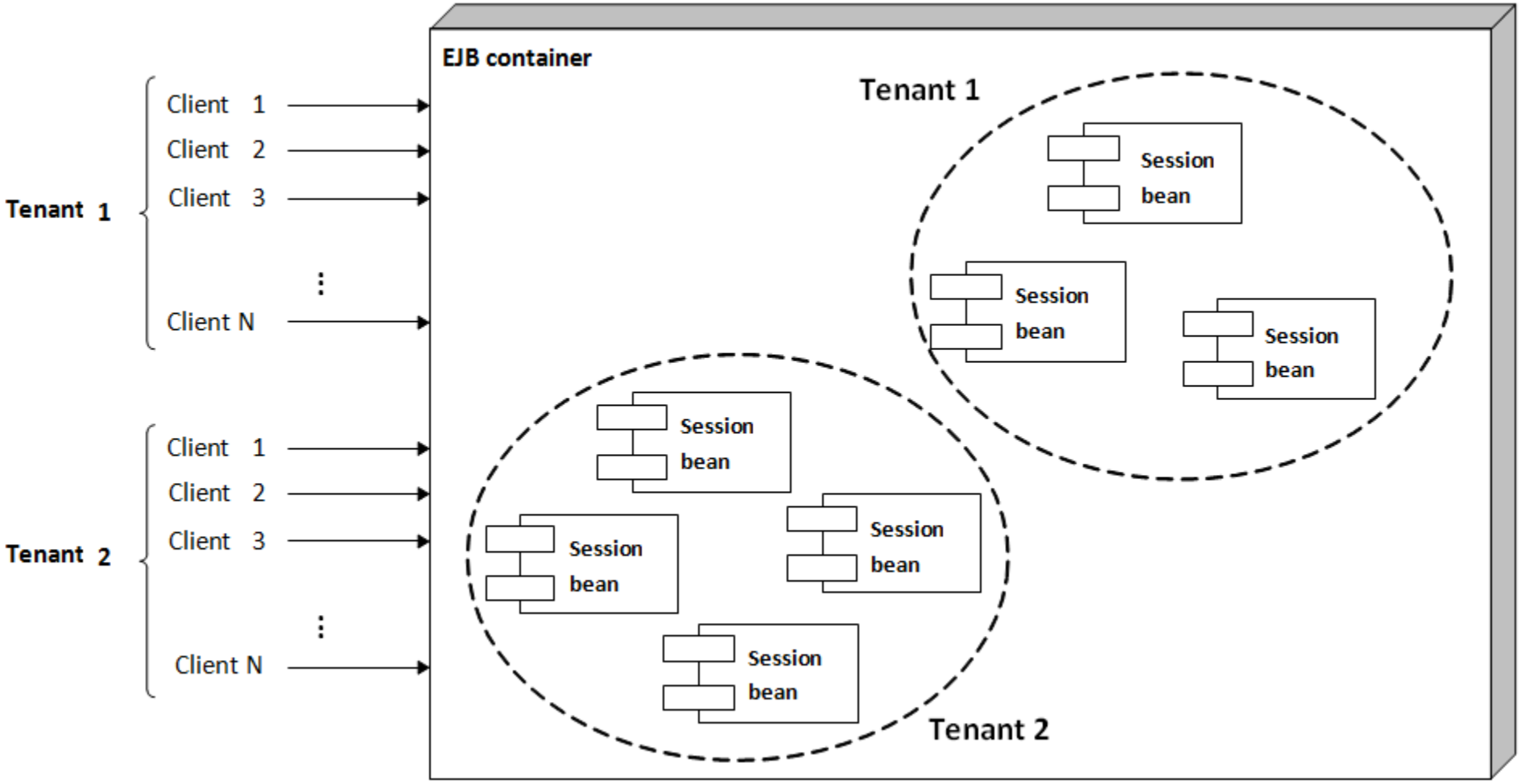
Multitenancy models

- Model type 3



Multitenancy models

- Model type 4 (Shared/Non-shared container)



Proposed parameter system

Parameter group	Parameter subgroup	Description
Monitoring	**Global application parameters**	**Annotations for server related metadata and descriptor for controlling the location of application execution.**
	Local parameters	**Annotations for CDN—caching, annotations for triggering methods upon creation of new Web/EJB container and annotations for SLO violation notification.**
Elasticity	Horizontal elasticity parameters	Descriptors for defining: number of EJB/Web containers, rules for defining auto-scaling and descriptor (or annotation) for defining session replication type.
Multitenancy	EJB parameters	Annotations for declaring EJB container either shared or unshared.
	JPA parameters	Annotations for defining type of tenant isolation in database.
	Servlet parameters	Annotations for declaring Web container shared or unshared.
	JMS parameters	Annotations for consuming tenant specific JMS Topics and Queues.
	Bean scope parameters	Annotations for new tenant scoped bean.
	Mail session and DataSoruces	Annotations for defining multitenant DataSources and tenant-specific mail session.

Proposed parameter system

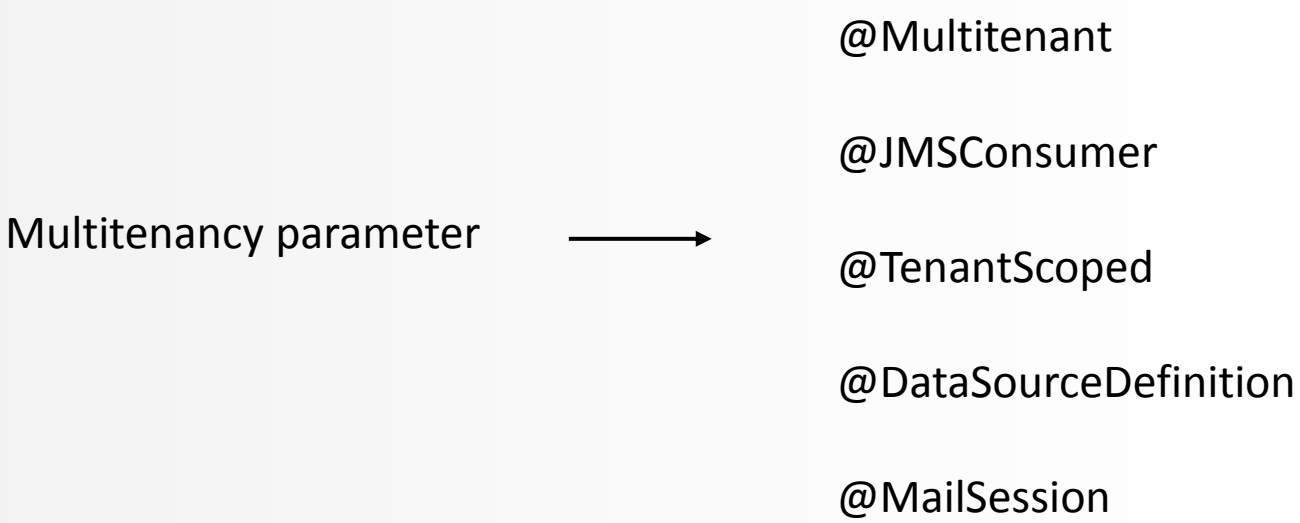
Parameter group	Parameter subgroup	Description
Monitoring	Global application parameters	Annotations for server related metadata and descriptor for controlling the location of application execution.
	Local parameters	Annotations for CDN—caching, annotations for triggering methods upon creation of new Web/EJB container and annotations for SLO violation notification.
Elasticity	**Horizontal elasticity parameters**	**Descriptors for defining: number of EJB/Web containers, rules for defining auto-scaling and descriptor (or annotation) for defining session replication type.**
Multitenancy	EJB parameters	Annotations for declaring EJB container either shared or unshared.
	JPA parameters	Annotations for defining type of tenant isolation in database.
	Servlet parameters	Annotations for declaring Web container shared or unshared.
	JMS parameters	Annotations for consuming tenant specific JMS Topics and Queues.
	Bean scope parameters	Annotations for new tenant scoped bean.
	Mail session and DataSoruces	Annotations for defining multitenant DataSources and tenant-specific mail session.

Proposed parameter system

Parameter group	Parameter subgroup	Description
Monitoring	Global application parameters	Annotations for server related metadata and descriptor for controlling the location of application execution.
	Local parameters	Annotations for CDN—caching, annotations for triggering methods upon creation of new Web/EJB container and annotations for SLO violation notification.
Elasticity	Horizontal elasticity parameters	Descriptors for defining: number of EJB/Web containers, rules for defining auto-scaling and descriptor (or annotation) for defining session replication type.
Multitenancy	**EJB parameters**	**Annotations for declaring EJB container either shared or unshared.**
	JPA parameters	**Annotations for defining type of tenant isolation in database.**
	Servlet parameters	**Annotations for declaring Web container shared or unshared.**
	JMS parameters	**Annotations for consuming tenant specific JMS Topics and Queues.**
	Bean scope parameters	**Annotations for new tenant scoped bean.**
	Mail session and DataSoruces	**Annotations for defining multitenant DataSources and tenant-specific mail session.**

Metadata system

- Metada-data system is developed from parameter system:
  - 9 monitoring, 4 elascitcy , 9 mutitenancy metadata



Metadata system

- Example 1 (Anotation):

Declaration

```
@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface Multitenant {
  MultitenantType type() default
            MultitenantType.SHARED;
};
```

Use case

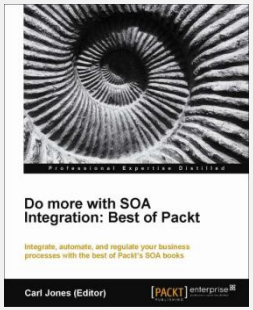
```
@Multitenant(type=SHARED)
@WebServlet(urlPatterns = {"/SimpleServlet"})
public class SimpleServlet extends HttpServlet {
  ...

}
```

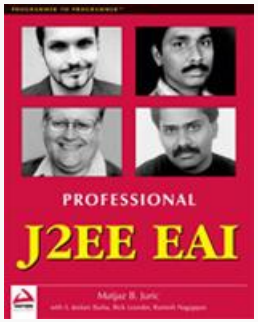
- Example 2 (global application descriptor):

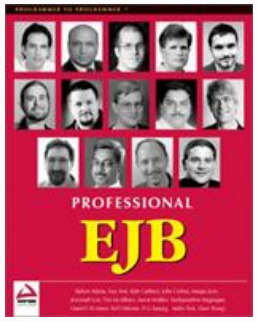
```
<application id="ID_app" >
  <display-name>Application.ear</display-name>
  <description> Description. </description>
  ...
  <provisioning-policy>
    <container name="EJB">
      <instances count="5" />
    </container>
   ...
  </provisioning-policy>
</application>
```

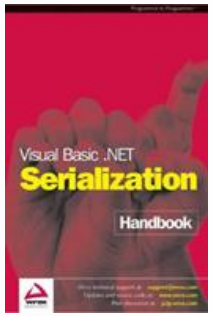
Evaluation and conclusion

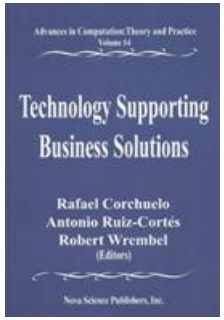
- Metadata for specifying number of EJB and number of Web containers was implemented on Java EE 6 compliant Oracle Glassfish Server
- Exchange rates application was implemented:
  - Once on server without cloud specific extensions
  - Once on server with cloud specific extensions
- Configuration time:
  - In first case it took the user on average 54 seconds of manual intervention to configure cluster (trigger VM deployment, configure server)
  - In second case, configuration tasks were executed in a just over 2 seconds.
- Even more important then time to configure:
  - No need for personnel on stand-by waiting to take scaling action.

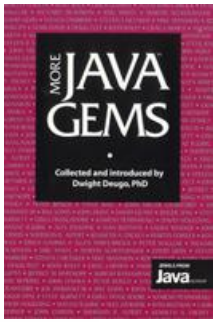












Java Champions · ORACLE ACE Director · IBM CHAMPION · ORACLE Certified Expert · Certified for IBM SOA technology · Certified for IBM CLOUD COMPUTING technology

e-naslov: **http://www.cloud.si**

e-naslov: **http://www.soa.si**

e-pošta: **info@cloud.si**