

# Java EE 7 – Auf in die Wolke

Standards-based Cloud vs. Vendor Proprietary

- Oracle: Standards-based Public Cloud Framework  
– Java, BPEL, SQL, XML, SOA, Groovy, Web Services

- Salesforce.com: Vendor Proprietary Cloud  
– APEX, Heroku, Force.com, Appforce, Siteforce, v...

- Oracle Fusion Applications run Everywhere  
– Oracle Cloud, etc. ... and On-Premise



1. Java EE – Past, Present and Future
2. Java EE 7 – Platform as a Service
3. PaaS Roadmap
4. Focus Areas
5. All the Specs

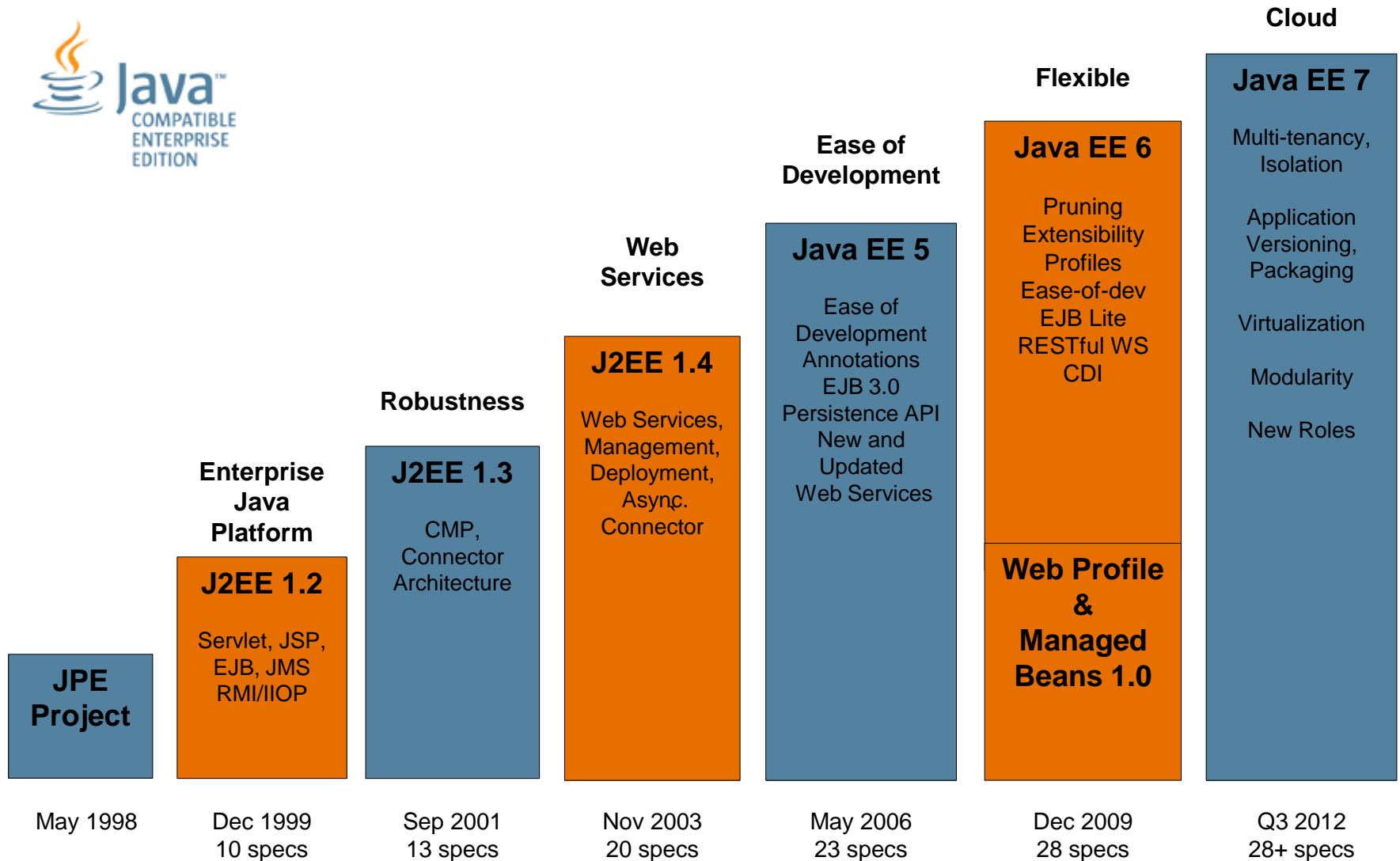
Exadata  
The First OLTP Database  
With Sun FlashFire Technology



<http://blog.eisele.net>

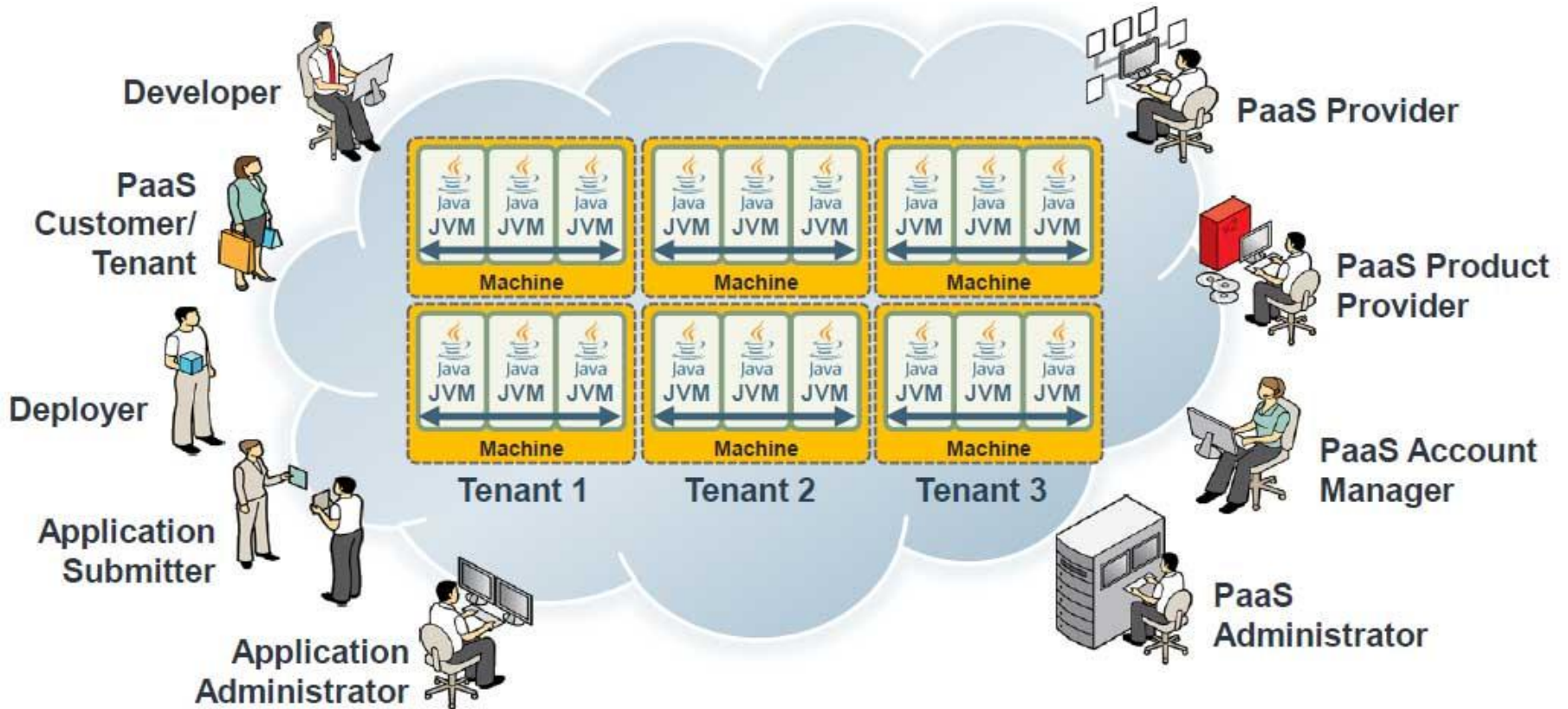
<http://twitter.com/myfear>

[markus.eisele@msg-systems.com](mailto:markus.eisele@msg-systems.com)

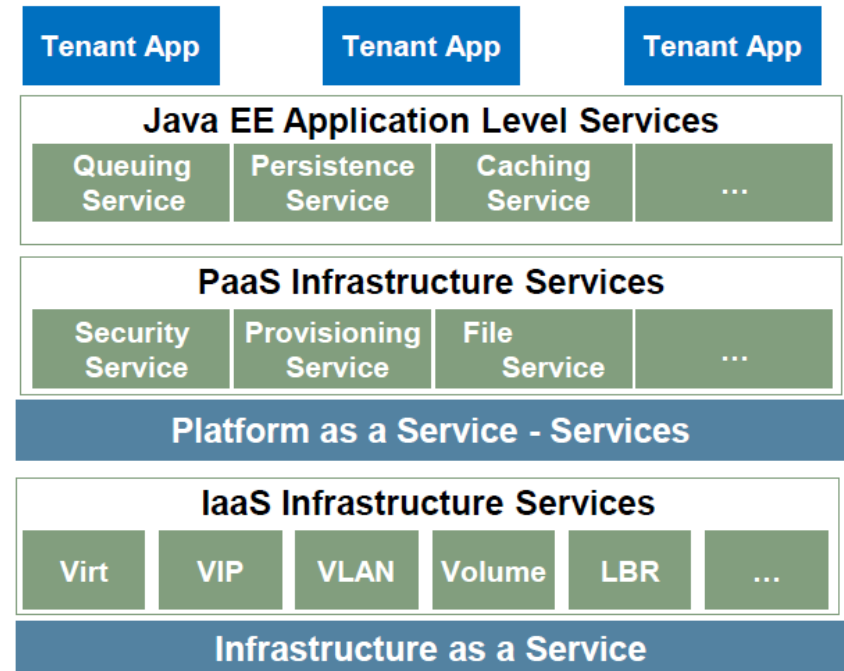


- Provide way for customers and users to leverage public, private, and hybrid clouds
- PaaS support entails evolutionary change
- Next logical step for Java EE
  - J2EE -> Java EE 6 : The Java EE Platform provides services
  - Java EE 7 : The Java EE Platform IS a service

- Define new platform roles to accommodate PaaS model
- Add metadata
  - For service provisioning and configuration
  - For QoS, elasticity
  - For sharing of applications and resources
  - For (re)configurability and customization
- Add useful APIs for cloud environment
  - JAX-RS client API, Caching API, State Management, JSON,...
- Extend existing APIs with support for multitenancy

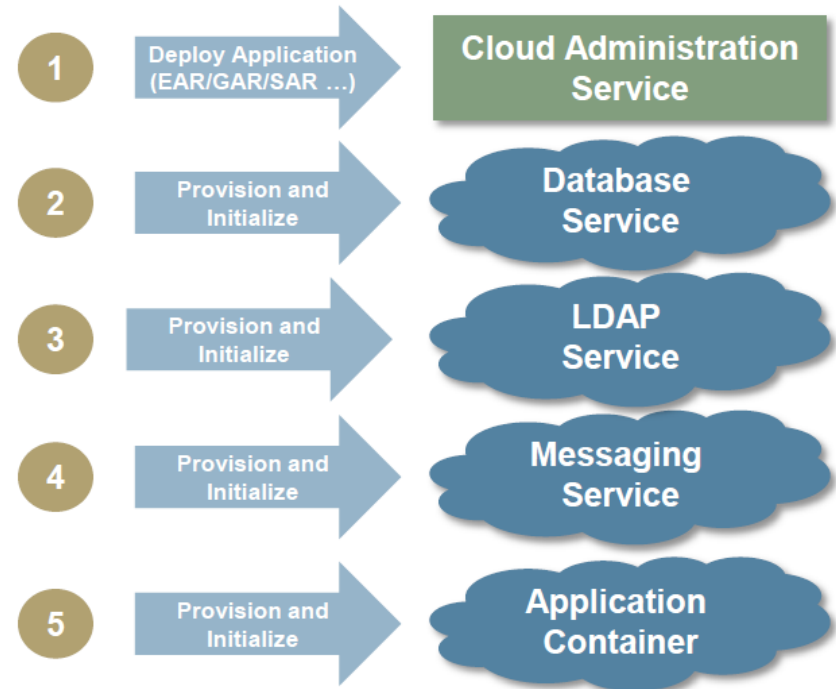


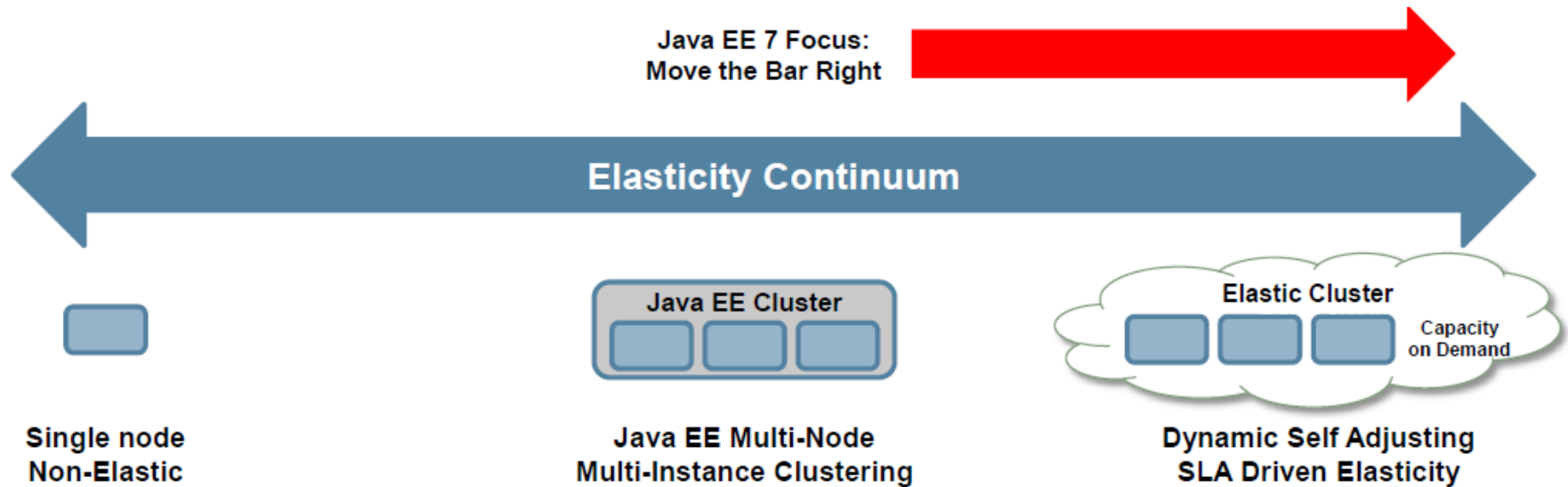
- In the cloud tenant applications consume services
- PaaS administrators host, configure, and manage application and infrastructure services
- Existing APIs in Java EE need to be updated to be service-enabled and tenant-aware
  - Example: pluggable services, late binding and tenant id injection





- Old Java EE Model
  - Configure Java EE resources – JDBC, JMS etc
  - Deploy Application EAR
- Java EE 7 Model
  - Provision and deploy application resources (e.g. LDAP stripe, data source instantiation and connection ...)
- Extensible Deployment Models Supporting Multiple Frameworks
  - Spring, Seam, Play ...





- Service Levels
- Minimum and Maximum Instances
- Futures – Self Adjustment, Capacity On Demand

- Alignment of ManagedBeans across CDI, EJB, JSF, ...
  - POJO -> ManagedBean -> Enterprise JavaBean
  - Extension of container-managed transactions beyond EJB
- Further simplifications for ease-of-development
  - JMS 2.0 focus on ease-of-development
  - Expanded use of dependency injection
  - Expanded service metadata; improved configuration
- Pruning
  - EJB CMP and BMP, JAX-RPC, Deployment API
- Update to Web Profile

- **JPA 2.1 ([JSR 338](#))**

The first spec to include new features is the JPA 2.1. The new features can be described with the following short list:

- Multi-Tenancy (Table discriminator)
- Stored Procedures
- Custom types and transformation methods - Query by Example
- Dynamic PU Definition
- Schema Generation (Additional mapping metadata to provide better standardization)

- **JMS 2.0 ([JSR 343](#))**

This could be considered as the most mature spec in general. It had a long 9 years to go since it's last maintenance release (April 2002).

- Modest scope
- Ease of development
- Pluggable JMS provider
- Extensions to support “Cloud”

- **EJB 3.2 ([JSR 345](#))** The goal of Enterprise JavaBeans 3.2 is to consolidate these advances and to continue to simplify the EJB architecture as well as to provide support for the Java EE platform-wide goal of further enabling cloud computing. The scope of EJB 3.2 is intended to be relatively constrained in focusing on these goals.
  - Incremental factorization (Interceptors)
  - Further use of annotations to simplify the EJB programming model
  - Proposed Optional: BMP/CMP
  - Proposed Optional: Web Services invocation using RPC

- **CDI 1.1 ([JSR 346](#))** Since the final release of the CDI 1.0 specification a number of issues have been identified by the community and a update to the specification will allow these to be addressed. A list of proposed updates is provided here, however the EG will consider other issues raised as the JSR progresses.
  - Embedded mode
  - Lifecycle Events
  - Declarative package scanning
  - Global ordering of interceptors and decorators
  - Injection Static Variables

- **Servlet 3.1 ([JSR 340](#))**

In developing the servlet specification 3.1 the EG will take into consideration any requirements from the platform to optimize the Platform as a Service (PasS) model for web applications. Beside this, the following areas should be addressed.

- Cloud support
- NIO.2 async I/O
- Leverage Java EE concurrency
- Security improvements
- Web Sockets support
- Ease-of-Development



- **JSF 2.2 ([JSR 344](#))** The new JSF JSR will be a significant feature update that builds on the advances from the previous JavaServer Faces versions.
  - Ease-of-Development
  - HTML 5 support (Forms, Headings, Metadata)
  - New components
  - Portlet Integration

- **JAX-RS 2.0 ([JSR 339](#))**

JAX-RS addresses most requested community features. To name a few: - Client API

- Hypermedia

- The primary API utilized for validation will be the Bean Validation API

- Ease-of-Development

- **Expression Language 3.0 ([JSR 341](#))** The Expression Language (EL), has been part of JSP specification since JSP 2.0. In Java EE 7 this will become a separate JSR.
  - Standalone JSR
  - Easier to use outside container
  - Criteria-based Collection selection
  - New operators
  - CDI events for expression eval

- **Bean Validation 1.1 ([JSR 349](#))** Being a version 1.0, Bean Validation stayed on the conservative side feature wise. The community has expressed interest in additional features to enhance the work done in the first version of the specification.
  - Integration with other JSRs (JAXRS, JAXB, JPA, CDI, EJB, JSF)
  - Method-level validation
  - Constraint composition

- Common Annotations 1.2 MR
- JAX-WS 2.3 MR
- JTA 1.2 MR
- JSP 2.3 MR
- JASPIC 1.2 MR

- Jcache 1.0 (JSR 107)
- Concurrency Utilities 1.0 (JSR-236),
- State Management 1.0
- Batch Processing 1.0
- JSON 1.0

- Nearly all JSRs up and running
- Remaining ones to be filed in next few weeks
- Final release target: Q3 2012
- Date-driven release: anything not ready will be deferred to Java EE 8

- Enterprise Software Development mit Java Blog:
  - <http://blog.eisele.net>
- Java EE 7 Expert Group Page
  - <http://javaee-spec.java.net>
  - One project per spec – e.g., jpa-spec, jax-rs-spec, jms-spec...
- Java EE 7 Reference Implementation
  - <http://glassfish.org>
- The Aquarium
  - <http://blogs.oracle.com/theaquarium>
- Arun Gupta's Blog „Miles to go...“
  - <http://blogs.oracle.com/arungupta>



**Vielen Dank für Ihre Aufmerksamkeit**

**Markus Eisele**

[markus.eisele@msg-systems.com](mailto:markus.eisele@msg-systems.com)

[www.msg-systems.com](http://www.msg-systems.com)



**.consulting .solutions .partnership**

