An Overview of Java EE

Mobile Web Services

Olivier Liechti & Yannick Iseli

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

What is Java Enterprise Edition?

- It is a development platform: it provides high-level APIs to develop software components.

- It is an execution platform: it provides an environment to deploy and bring these components “to live”.

- It is an entreprise platform: it provides support for distributed transactions, security, integration, etc.

- Separation of concerns: "The developer takes care of the business logic. Java EE takes care of the systemic qualities".

http://flickr.com/photos/decade_null/427124229/sizes/m/#cc_license

Java EE and standards

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- Java EE is a specification
  - Defined through the JCP, it is a specification that software editors can decide to implement. Java EE 5 is defined in JSR 244.
- Java EE is an “umbrella” specification
  - Java EE builds upon other specifications (servlets, EJBs, JDBC, etc.) and specifies which specifications (and which versions) need to be implemented by a Java EE certified application server.
  - Java EE also defines a programming model and defines several roles (developer, assembler, deployer, etc.).

Java Community Process

Specification Lead
Bill Shannon Sun Microsystems, Inc.

Expert Group
Barreto, Charlton
Capgemini
Dudney, Bill
Hewlett-Packard
Kohen, Elika S.
Oracle
Pratap, Rama Murthy Amar
Reinshagen, Dirk
Shah, Suneet
Tiwari, Ashish
Umapathy, Sivasundaram

BEA Systems
Chandrasekaran, Muralidharan
E.piphany, Inc.
IBM
Leme, Felipe
OW2
Raible, Matt
SAP AG
Sun Microsystems, Inc.
Tmax Soft, Inc.

Borland Software Corporation
Crawford, Scott
Genender, Jeff
Ironflare AB
Novell, Inc.
Pramati Technologies
Red Hat Middleware LLC
SeeBeyond Technology Corp.
Sybase
Trifork

J2EE or Java EE?

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- Java Enterprise Edition is **not** a recent specification.
- The SDK 1.2 was published in 1999.
- The specification is managed through the JCP since version 1.3
- We used to talk about "Java 2 Enterprise Edition 1.3", or J2EE 1.3
- We then moved from J2EE 1.3 to J2EE 1.4 to... **Java EE 5**
- Today, we should speak of Java EE, but J2EE is still sometimes used...
- Today, most application servers implement Java EE 6
- The early draft 2 for Java EE 7 has been submitted in November 2012; final spec expected for Q2 2013

Java Community Process

Specification Lead
Bill Shannon Sun Microsystems, Inc.

Expert Group

Barreto, Charlton
Capgemini
Dudney, Bill
Hewlett-Packard
Kohen, Elika S.
Oracle
Pratap, Rama Murthy Amar
Reinshagen, Dirk
Shah, Suneet
Tiwari, Ashish
Umapathy, Sivasundaram

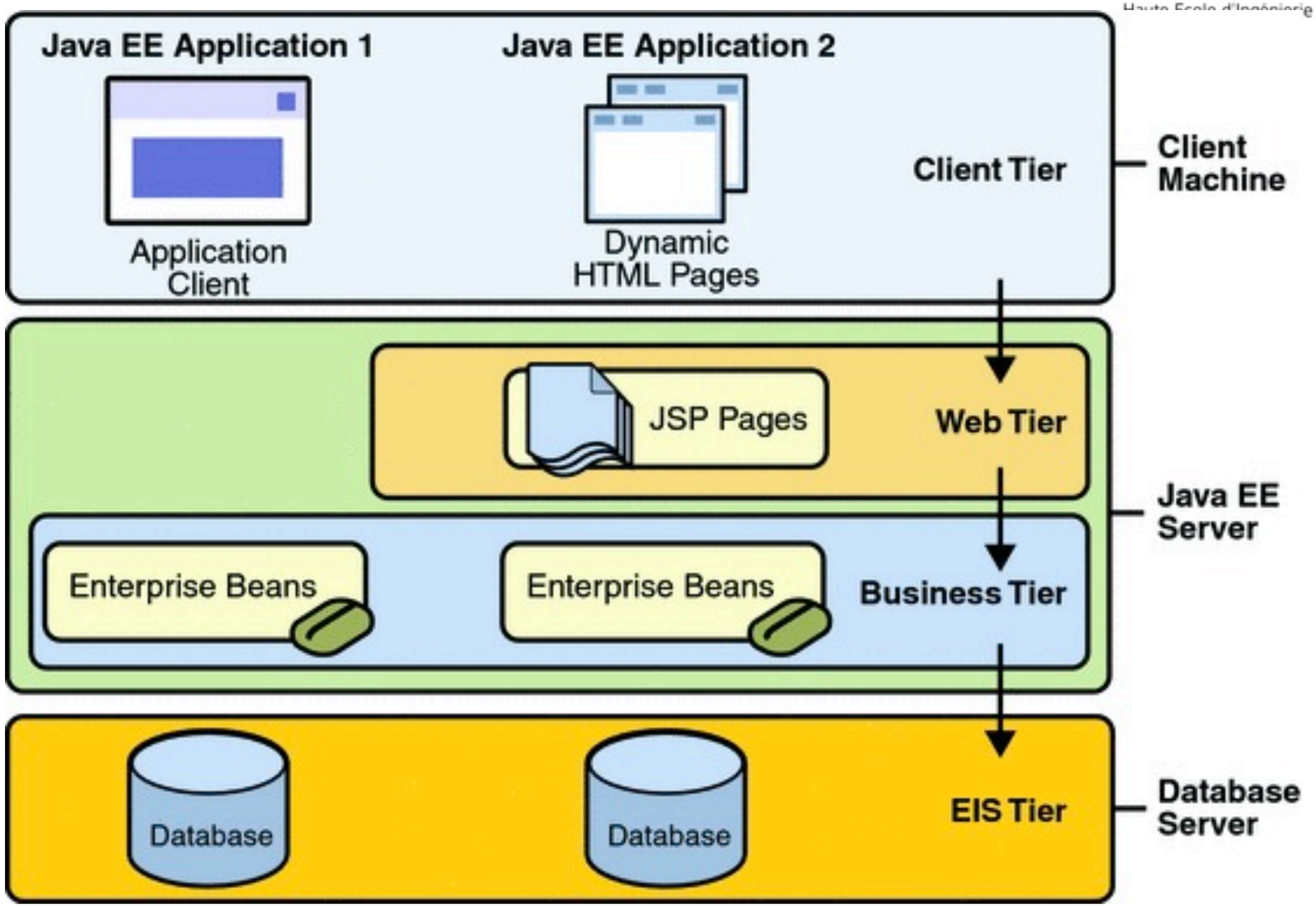
BEA Systems
Chandrasekaran, Muralidharan
E.piphany, Inc.
IBM
Leme, Felipe
OW2
Raible, Matt
SAP AG
Sun Microsystems, Inc.
Tmax Soft, Inc.

Borland Software Corporation
Crawford, Scott
Genender, Jeff
Ironflare AB
Novell, Inc.
Pramati Technologies
Red Hat Middleware LLC
SeeBeyond Technology Corp.
Sybase
Trifork

Architecture

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

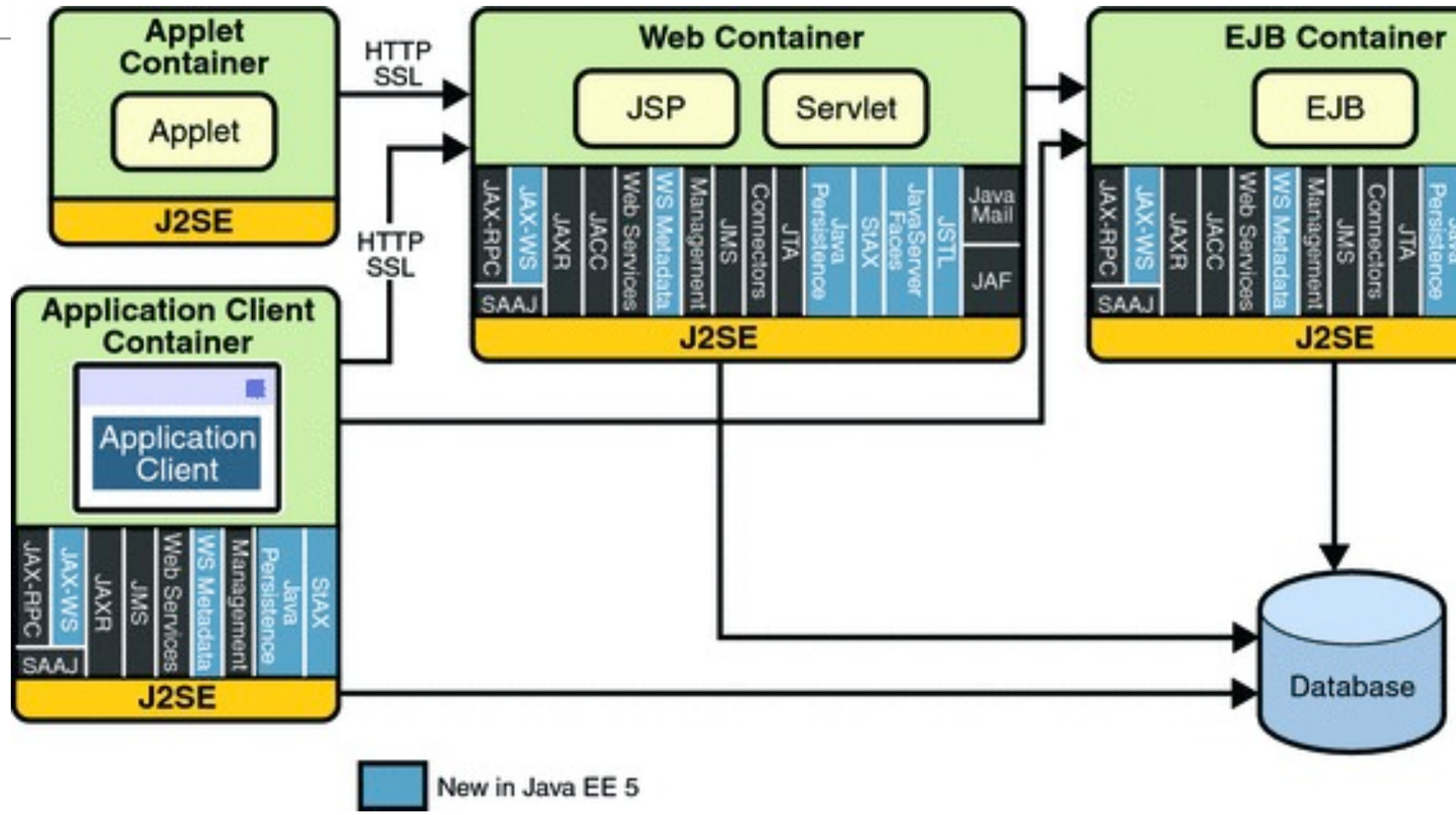
- The software that implements the Java EE specification is called an "**application server**"
  - There are **open source** and **proprietary** application servers.
  - Glassfish, JBoss, WebSphere, BEA WebLogic are examples of application servers.
    - Editors compete on aspects that are not defined the specification (clustering, administration, etc.).
- Key notion in the Java EE architecture: the **containers**
  - a container is an **environment** in which we deploy components;
  - a container **provides services** (transactions, security, etc.) through APIs;
  - there are different containers in Java EE: the "**web**" container, the "**ejb**" container and even a "**client**" container that can be used for rich clients.



http://java.sun.com/javaee/5/docs/tutorial/doc/bnaay.html

Java EE 5 APIs

- Enterprise JavaBeans Technology
- Java Servlet Technology
- JavaServer Pages Technology
- JavaServer Pages Standard Tag Library
- JavaServer Faces
- Java Message Service API
- Java Transaction API
- JavaMail API
- JavaBeans Activation Framework
- Java API for XML Processing
- Java API for XML Web Services (JAX-WS)
- Java Architecture for XML Binding (JAXB)
- SOAP with Attachments API for Java
- Java API for XML Registries
- J2EE Connector Architecture
- Java Database Connectivity API
- Java Persistence API
- Java Naming and Directory Interface
- Java Authentication and Authorization Service
- Simplified Systems Integration



http://java.sun.com/javaee/5/docs/tutorial/doc/bnacj.html

Components, packaging & deployment

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- The first version of J2EE put a lot of emphasis on the separation of roles:
  - **Component developer** builds self-contained, reusable **components** and focuses on business logic (“Here is a customer management brick”, “Here is a pricing brick”).
  - **Application assemblers** create **applications** by gluing together different components (“I connect the customer management and the pricing bricks in order to have an insurance application”).
  - **Deployers** have to take applications and to put them into production. Every application needs resources (DBs, user repositories, etc.). The deployer has to connect the application to the resources (declaratively!!).
- We also want to be able to manage the life-cycle of applications and to have control on the release of new versions. It really helps to create **packages**!

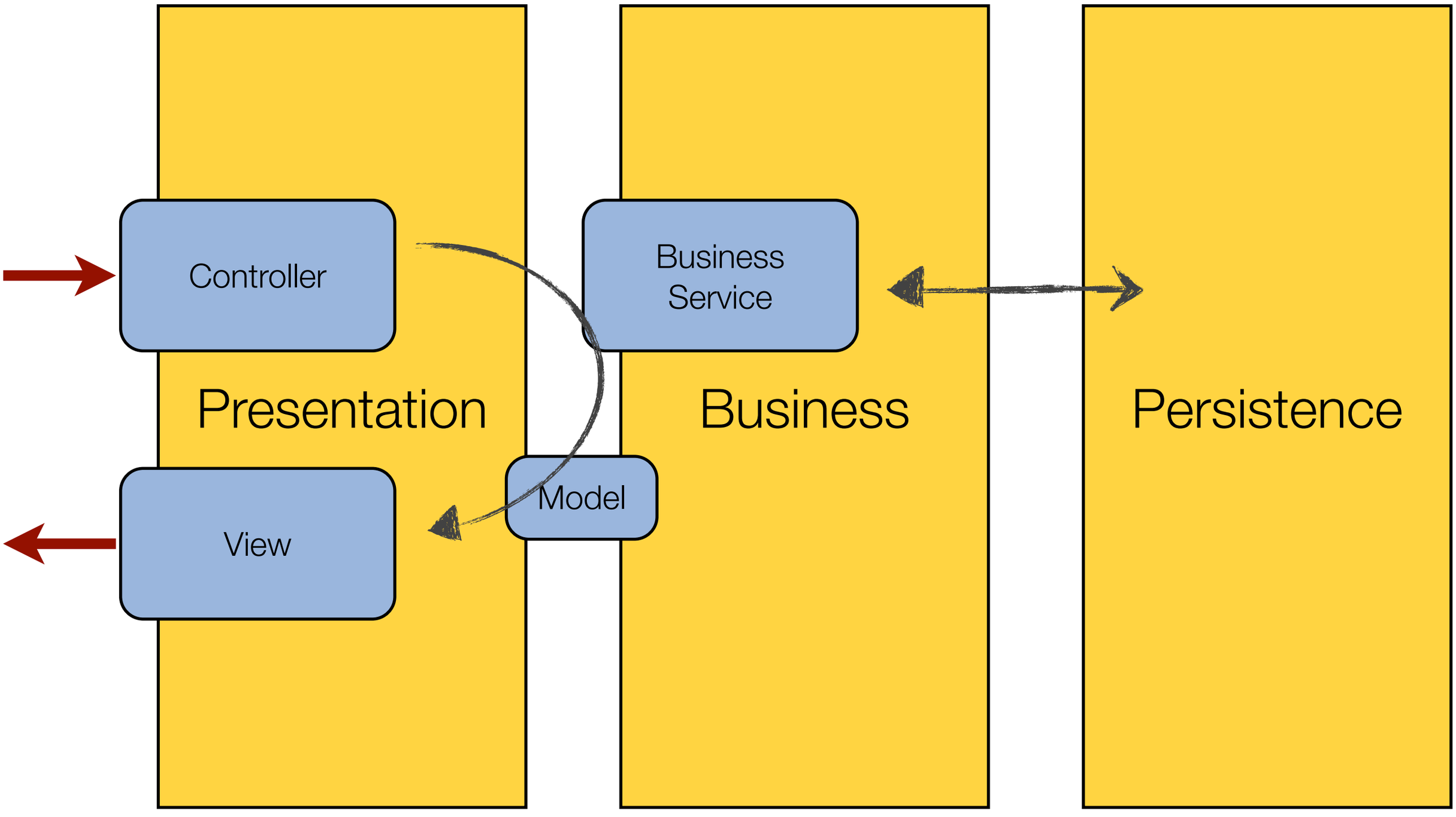
Components, packaging & deployment

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- Different types of components, different types of packages/archives:
  - Java libraries are useful in enterprise applications; we can bundle classes into **traditional .jar files (Java ARchives)** and integrate them in our applications.
  - Some applications do not require all the features provided by the Java EE platform; they are “lightweight” can live entirely in the web container. We package these applications in **.war files (Web ARchives)**.
  - Business services can be packaged and deployed independently from a presentation front-end. We can package **EJBs in .jar files**.
  - Full-blown Java EE applications bundle together web-tier components, service components, libraries. A Java EE application is packaged in a **.ear file (Enterprise ARchive)**. The .ear file contains a .war files and one or more .jar files.

The Business Tier

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



Containers and components

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- There are **different ways** to design, build and deploy the “business service” components.
- In the **simplest scenario**, think of a plain vanilla web application, where you create a “controllers”, a “services” and a “model” packages. Here, you implement services as POJOs and **do everything yourself**:
  - Instantiation of services, concurrency control, transactions, security, etc.
- When the application/system has some complexity (because of the functionality, because of the volume of transactions, because of the integration, etc.), it becomes interesting to use **“managed” components**.
- The idea is to let “something” in the **infrastructure** manage the life cycle of your service components. That “something” can take different shapes.
- In Java EE, that “something” is the Enterprise Java Bean **container**.

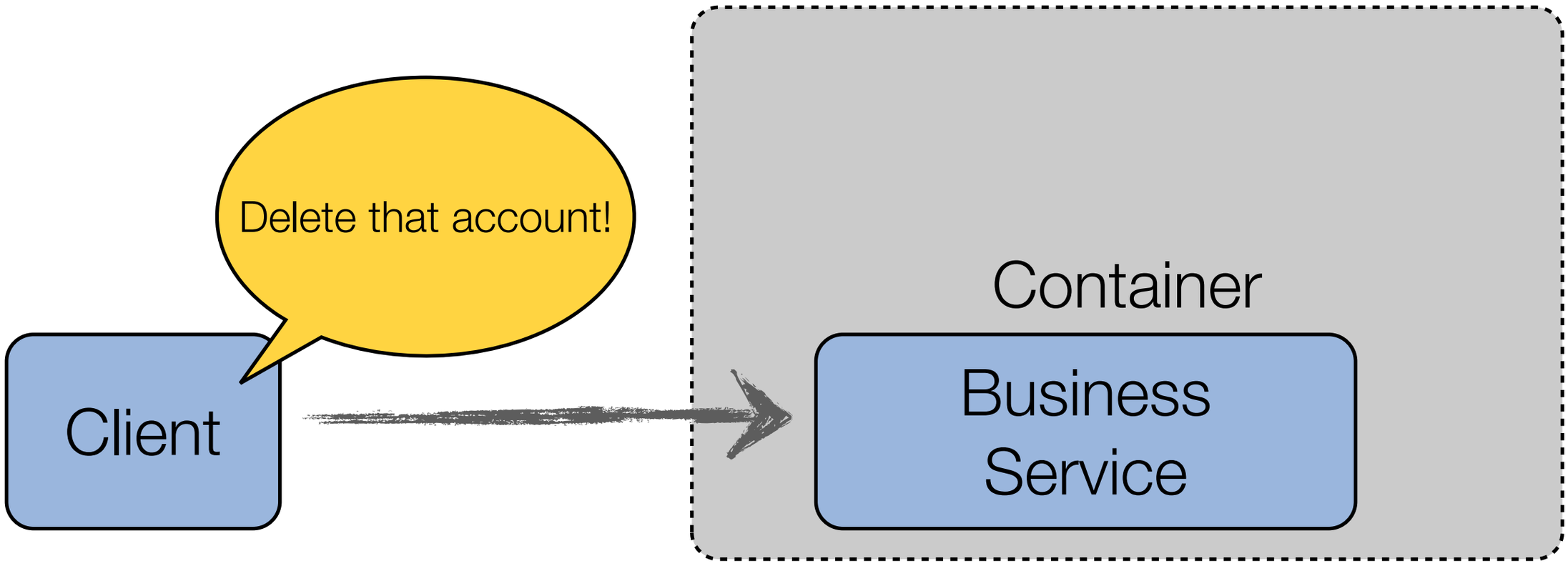
Containers and components

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

- The **EJB container** is one sub-system of the Java EE container.
- It provides a **runtime environment** for the business services, implemented as Enterprise Java Beans.
- The **developer** knows that services will be “available” to its components, once they are deployed in the container. He can thus use these services through standard **APIs**.
- The **production manager** configures the EJB container and tunes it in different ways (this is implementation specific).
- With **some application servers**, the EJB container can be **distributed** over several physical nodes (for scalability and availability purposes). This distribution is **transparent to the clients** - they sill only see “one runtime environment”.

Mediated access

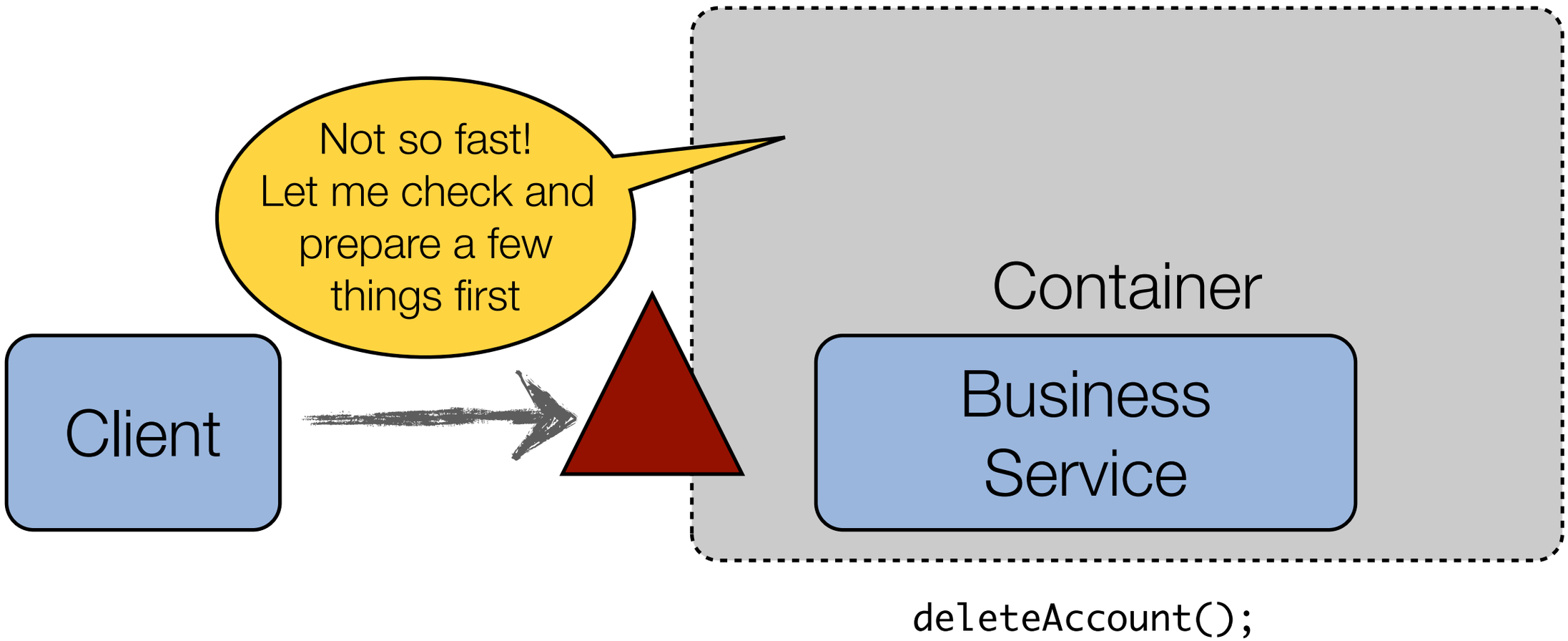
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud




```
deleteAccount();
```

Mediated access

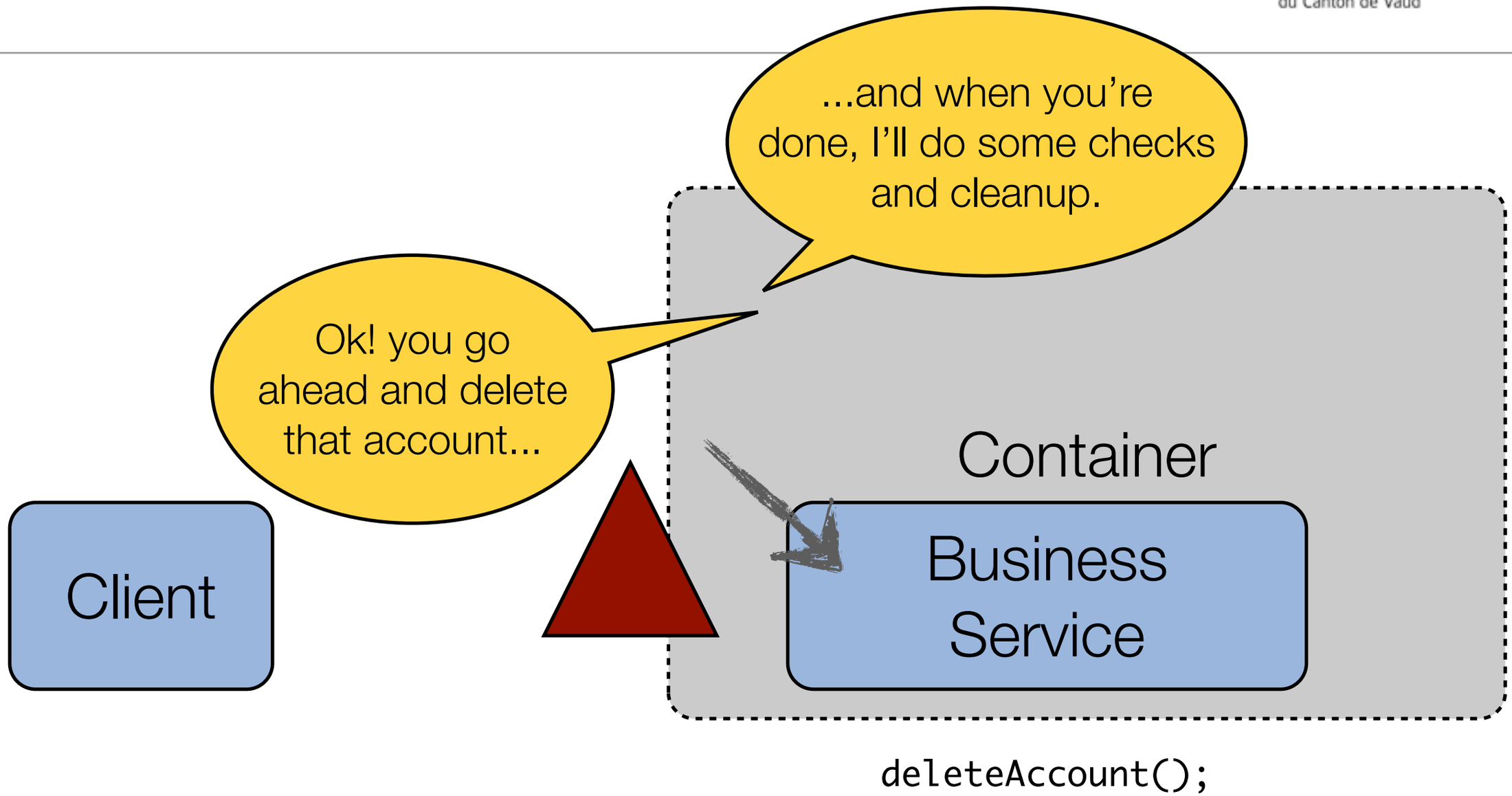
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud




```
deleteAccount();
```

Mediated access

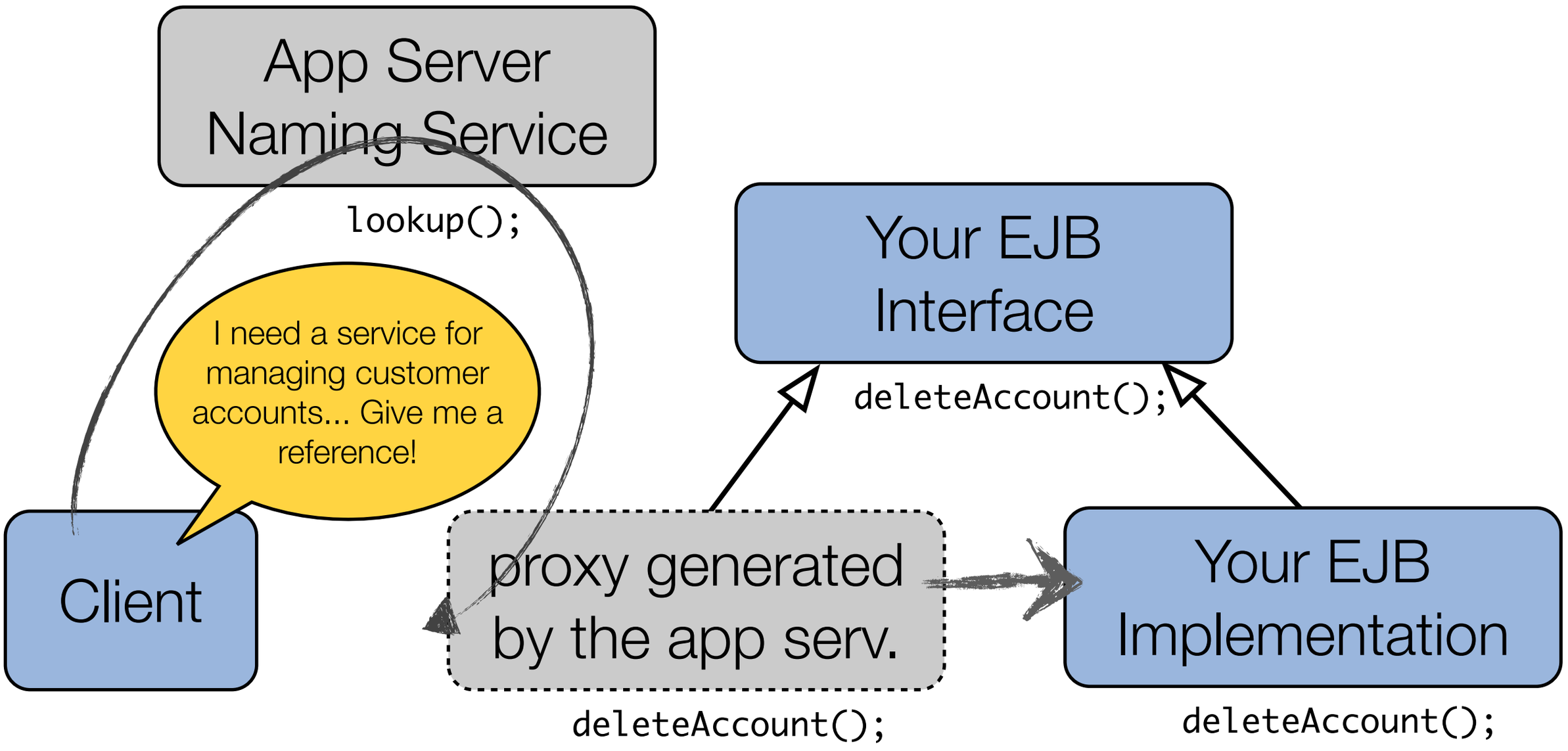
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud




```
deleteAccount();
```

Mediated access

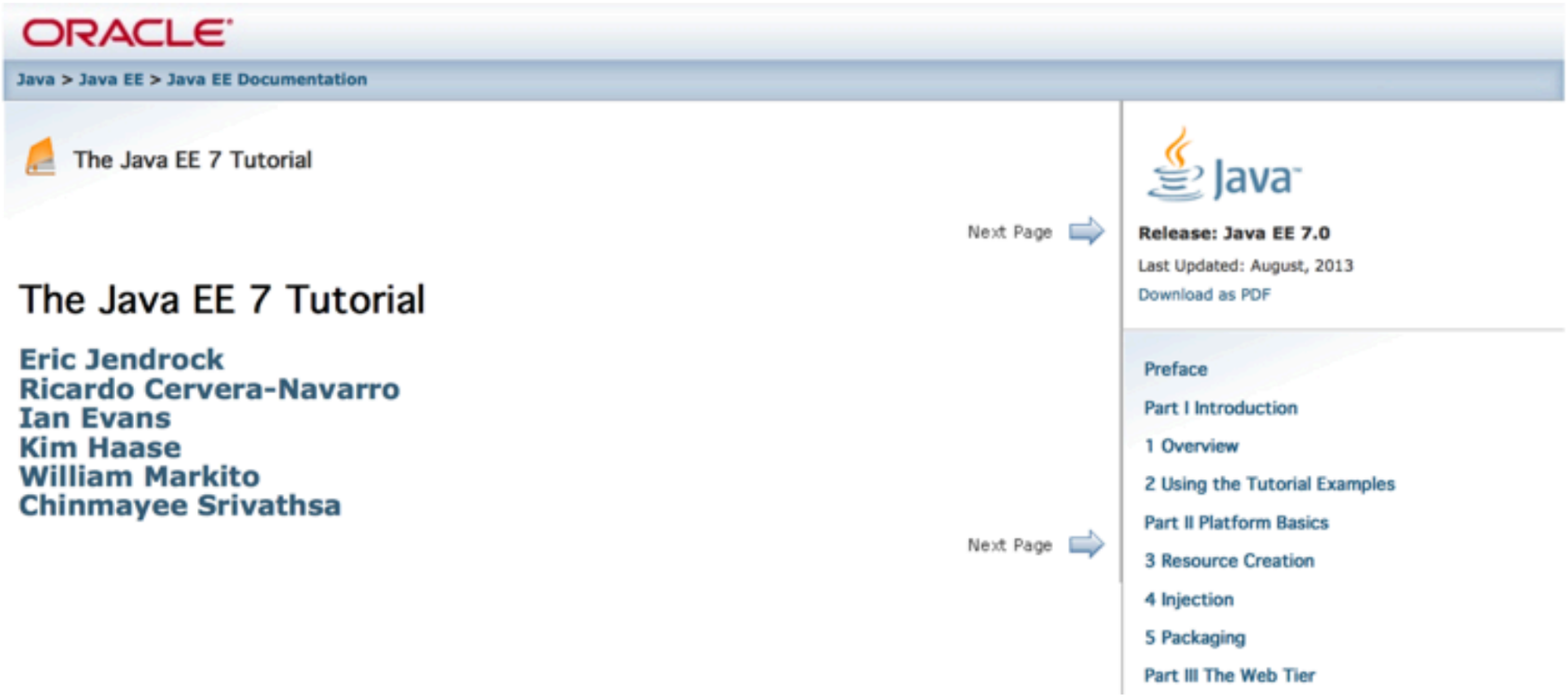
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



Reference: the Java EE 7 Tutorial

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

http://docs.oracle.com/javaee/7/tutorial/doc/


ORACLE

Java > Java EE > Java EE Documentation

The Java EE 7 Tutorial

Next Page

Java

Release: Java EE 7.0

Last Updated: August, 2013

Download as PDF

The Java EE 7 Tutorial

Eric Jendrock
Ricardo Cervera-Navarro
Ian Evans
Kim Haase
William Markito
Chinmayee Srivathsa

Next Page