Lecture 1: Getting started…

Olivier Liechti
AMT

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

Today's agenda

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

13:15 - 13:30	**15'**	**Intro: multi-tiered apps and Java EE**
13:30 - 14:15	45'	AMT-Discovery
14:15 - 14:30	15'	Guidelines for phase 1
14:30 - 15:30	60'	Individual work
15:30 - 15:40	10'	Wrap-up

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Introduction

Multi-Tiered Applications

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

Multi-Tiered Applications

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

The application is structured in “tiers” (tranches)

Multi-Tiered Applications

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

Frameworks

spring

EHCACHE

HIBERNATE

APIs

servlet

EJB

JDBC

JSP

JAX-RS

JAX-WS

Java EE
Application Server

JPA

JMS

Tools

Apache maven

Jenkins

Apache JMeter™

jbehave

docker

Selenium Webdriver

MySQL

mongoDB

Gamification

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

	19.09.2016	Bootcamp	**Les applications multi-tiers et à Java EE** docker compose, serveurs d'application, IDE	As a guest, I can visit the platform home page.
	26.09.2016		**Les tiers client, présentation et métier** Servlets, JSP, EJB, JMeter	As a guest, I can register and create an account. As a registered user, I can provide my credentials and log into the platform.
	03.10.2016		**Les APIs REST** JAX-RS, SonarQube, supertest	As an admin, I have a REST API to do CRUD operations on user accounts
	10.10.2016		**Le tiers d'accès aux données** JDBC	As a registered user, I can manage the list of applications I want to gamify with the platform.
Vendredi 21.10.2016 (23h59) Remise projet 1	17.10.2016		**Le tiers d'accès aux données** JPA	
	31.10.2016	Itération 1	**La documentation et les tests d'API** RAML, Jersey client, supertest	As an app owner, I can define a list of badges via the /badges endpoint As an app owner, I can define a list of point scales via the /pointScales endpoint
Lundi 7.11.2016 Travail écrit 1	07.11.2016			
	14.11.2016	Itération 2	**Les micro-services** Spring Boot et Spring Data	migration à Spring Boot
	21.11.2016			
	28.11.2016	Itération 3	**Les architectures orientées événements**	As an app, I can send a stream of user events via the /events endpoint As an app owner, I can define a rule to award a badge when some conditions are met (via the /rules endpoint)
	05.12.2016			
	12.12.2016	Itération 4		As an app owner, I can define a rule to award points (via the /rules endpoint)
Vendredi 23.12.2016 (23h59) Remise projet 2	19.12.2016			
	09.01.2016	Itération 5		application de démo
Lundi 16.01.2017 Travail écrit 2 **Vendredi 20.01 (23h59)** Remise finale du projet	16.01.2016			
Lundi 23.01 et mercredi 25.01 Présentations & démos	23.01.2016	Wrap-up		

Du 19.09.2016 au 07.11.2016	*n* devoirs (question/QCM soumise via Moodle avec date limite stricte); chaque devoir est accepté ou refusé (pas de note individuelle)
07.11.2016	test en classe (note de 1 à 6)
	note travail écrit 1 = note test en classe - (nombre de devoirs refusés * 0.5)

Du 08.11.2016 au 30.01.2016	*m* devoirs (question/QCM soumise via Moodle avec date limite stricte); chaque devoir est accepté ou refusé (pas de note individuelle)
16.01.2017	test en classe (note de 1 à 6)
	note travail écrit 2 = note test en classe - (nombre de devoirs refusés * 0.5)

2 travaux écrits + questionnaires réguliers (60%)

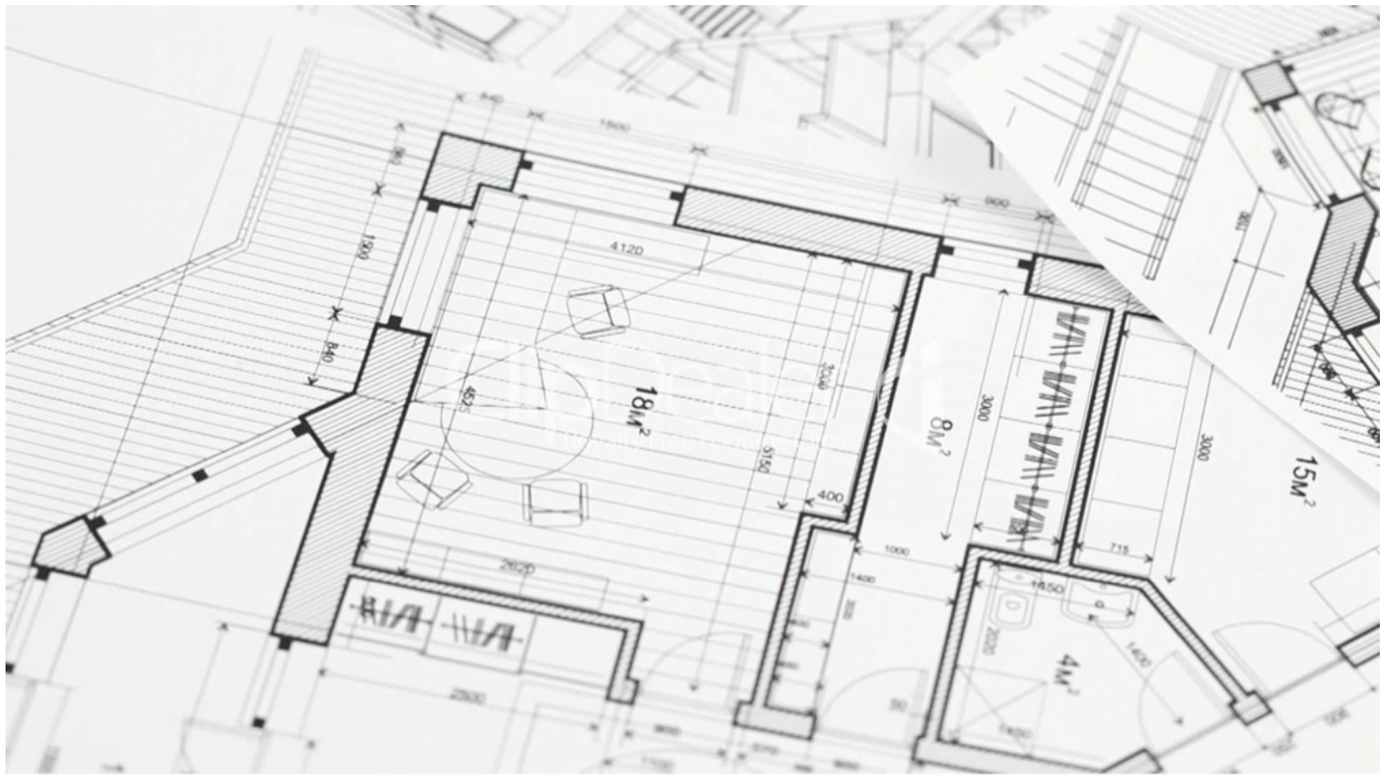
3 notes de projets (40%)

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



Architecture

What do we mean by “**architecture**” “**architectural style**” in the context of software systems?

- There is **no single definition** for software architecture!

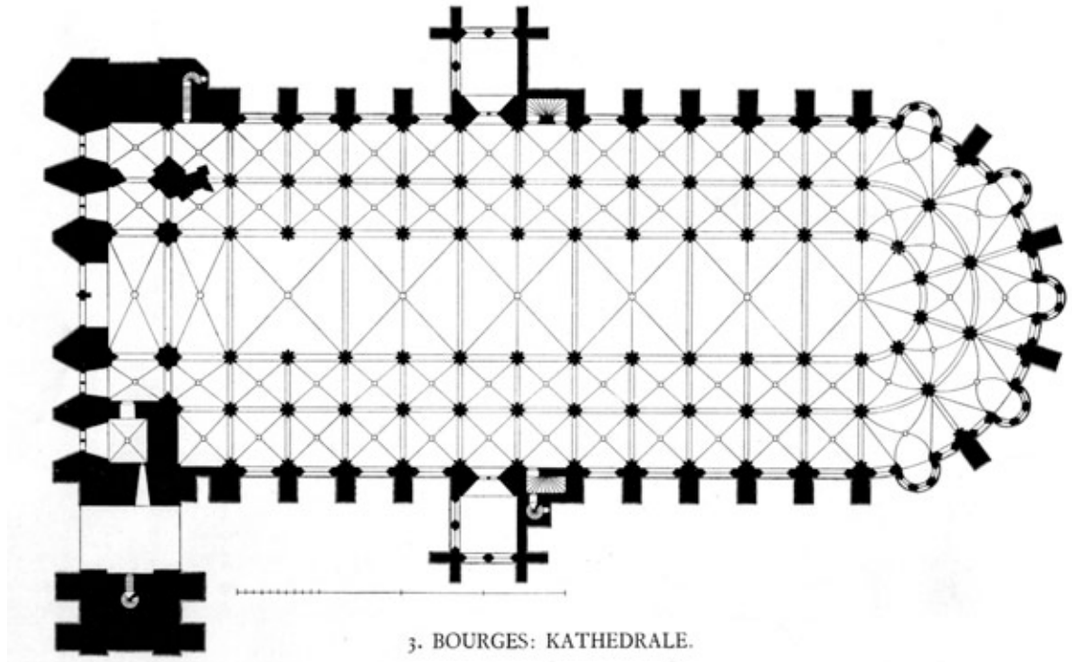
- The **architecture** of **a particular application** is a **description** of its **high-level structure** and **behavior**. It is a description of its **components**, how they **interact** with each other and with **external systems**.

- The **architecture** of **one particular** application is **documented** with a **collection of models** (**UML** diagrams are examples of such models).

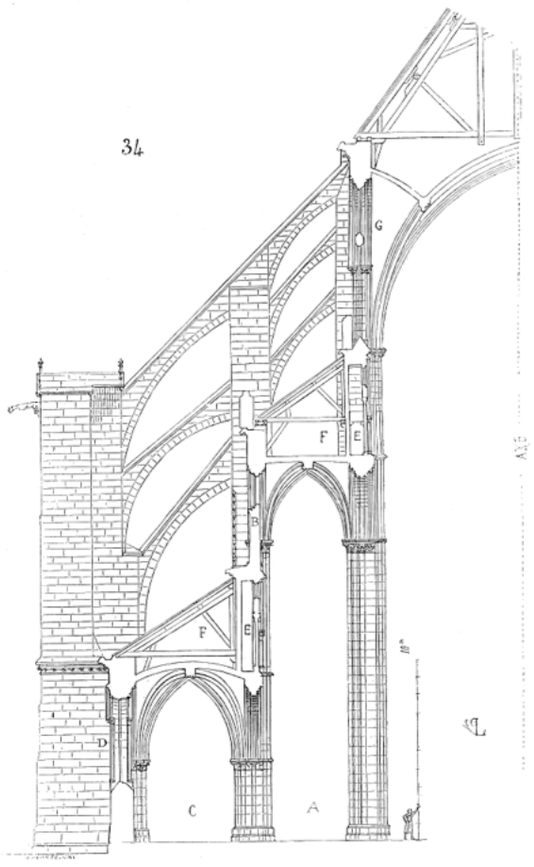
- An **architectural style** is a **set of core principles and rules** (**architectural patterns**), which are **common** to a large number of different applications.

heig-vd

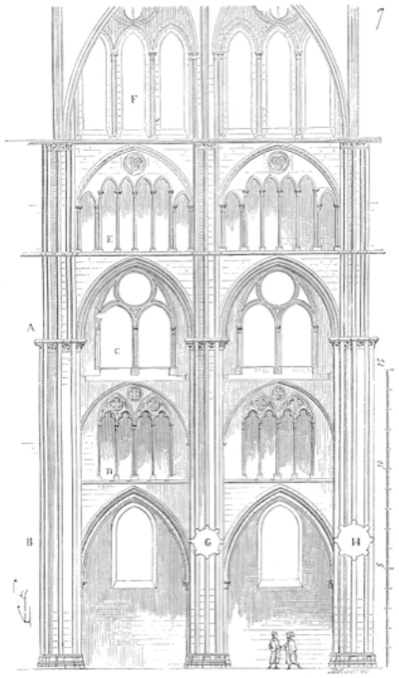
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

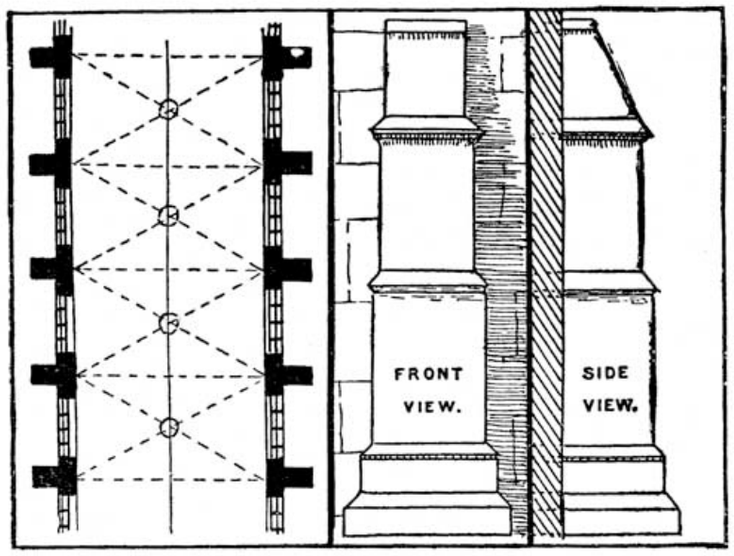


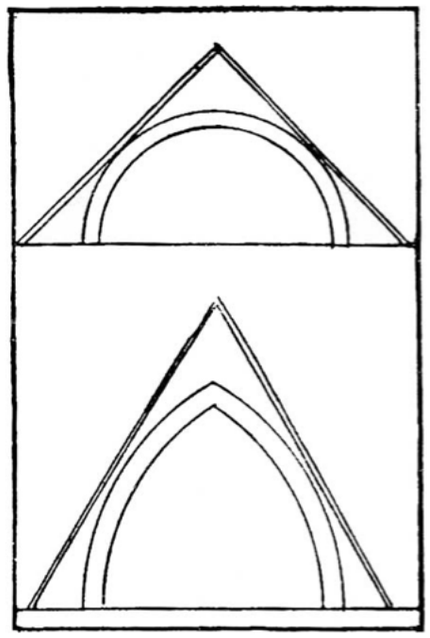
3. BOURGES: KATHEDRALE.

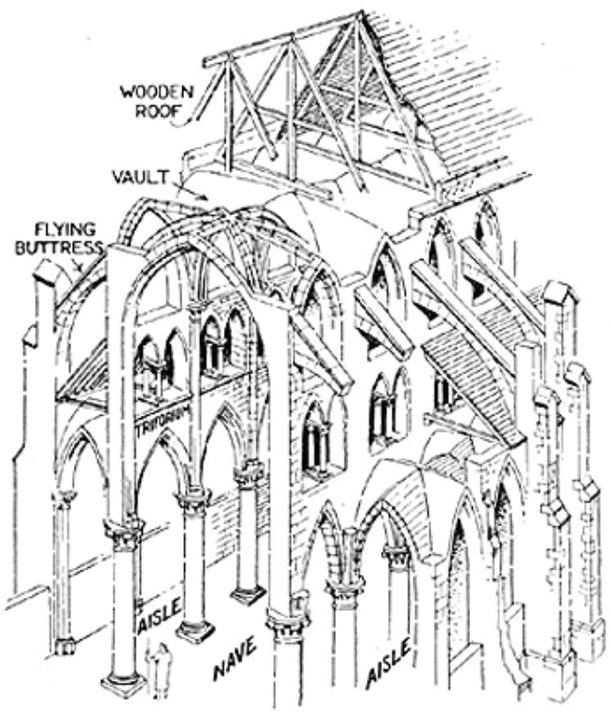


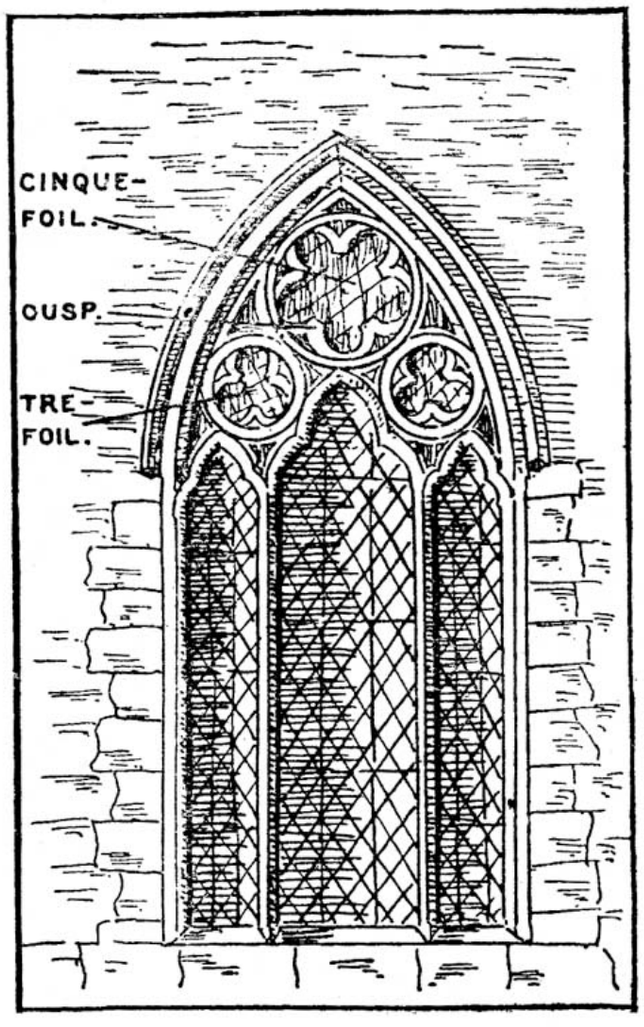
The Architecture of Bourges Cathedral



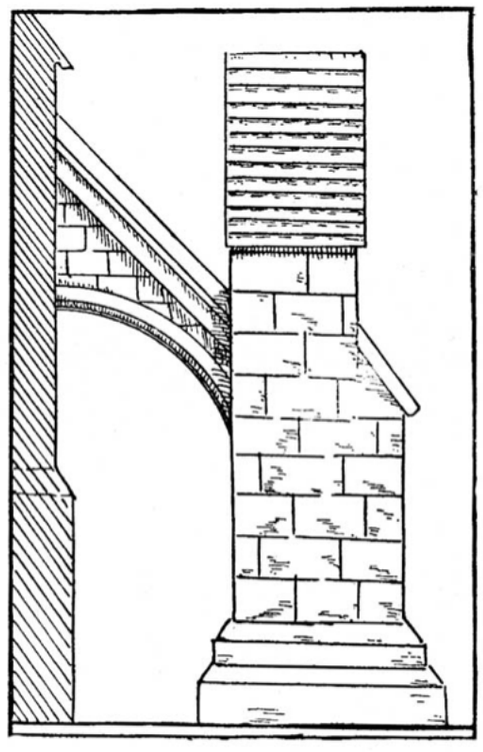








The Gothic Architectural Style



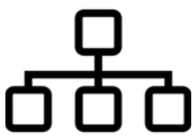
What are some of the well-known **architectural styles?**

- The client-server architectural style
- The 3-tiered architectural style
- **The multi-tiered architectural style**
- The RESTful architectural style
- The service oriented architectural style
- The event-driven architectural style
- The layered architectural style

Architectural styles are **not mutually exclusive**. Many applications combine patterns from several styles!

3-tiers, multi-tiers… **how many tiers** does a multi-tiered application have, and which ones?

human-readable
markup generation

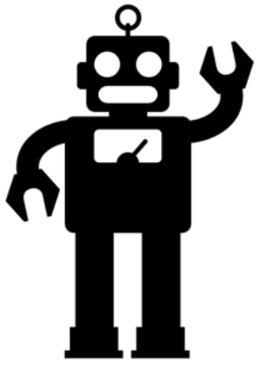


Page navigation

Data access

Shared business
logic and rules

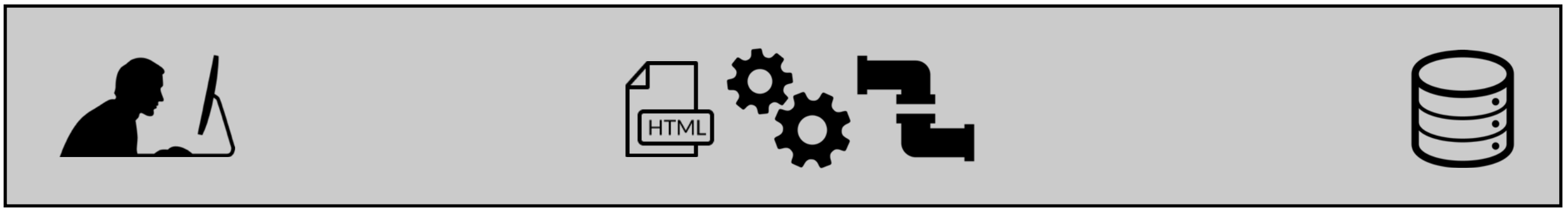
Message/
event bus



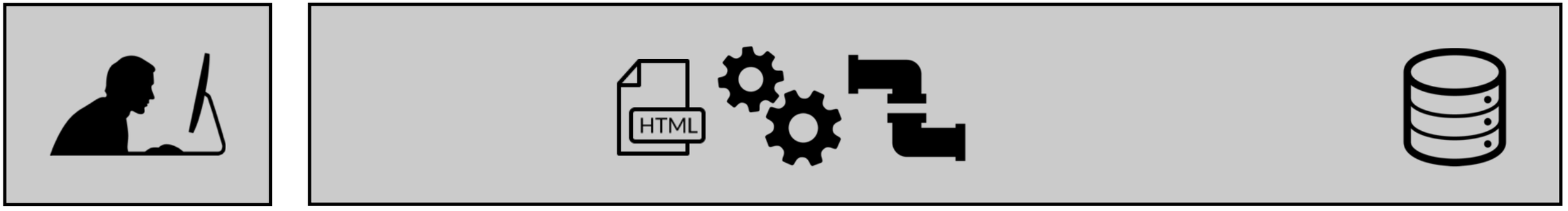
machine-readable
markup generation

User	Client	Presentation	Business	Integration	Resources

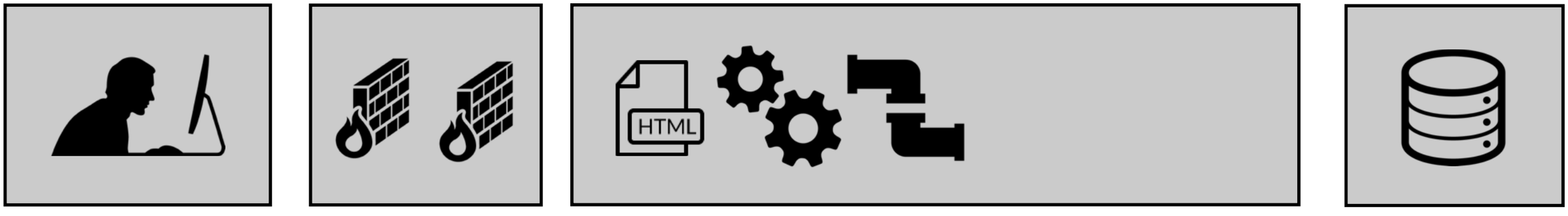
What is the mapping between tiers, **logical nodes** and **physical nodes**?



Dev environment



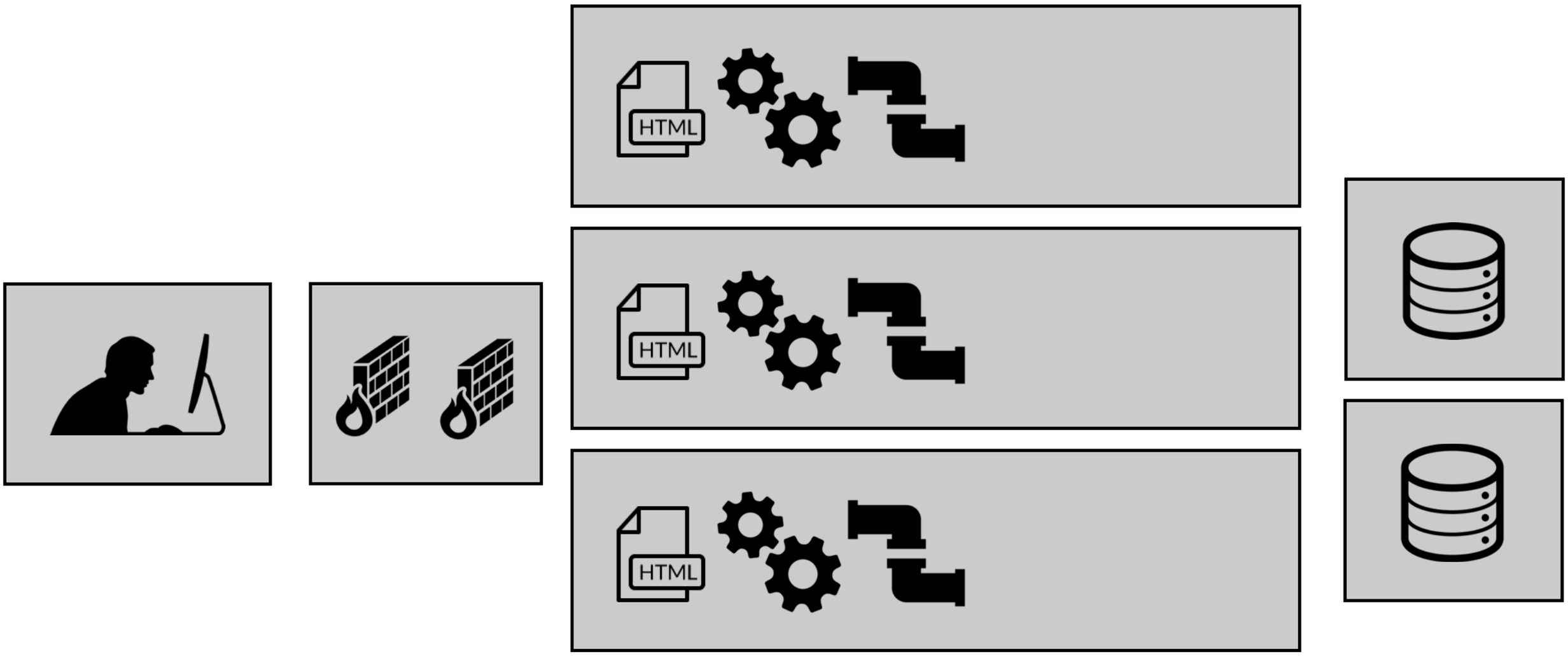
Minimal environment



Simple environment

User | **Client** | **Presentation** | **Business** | **Integration** | **Resources**

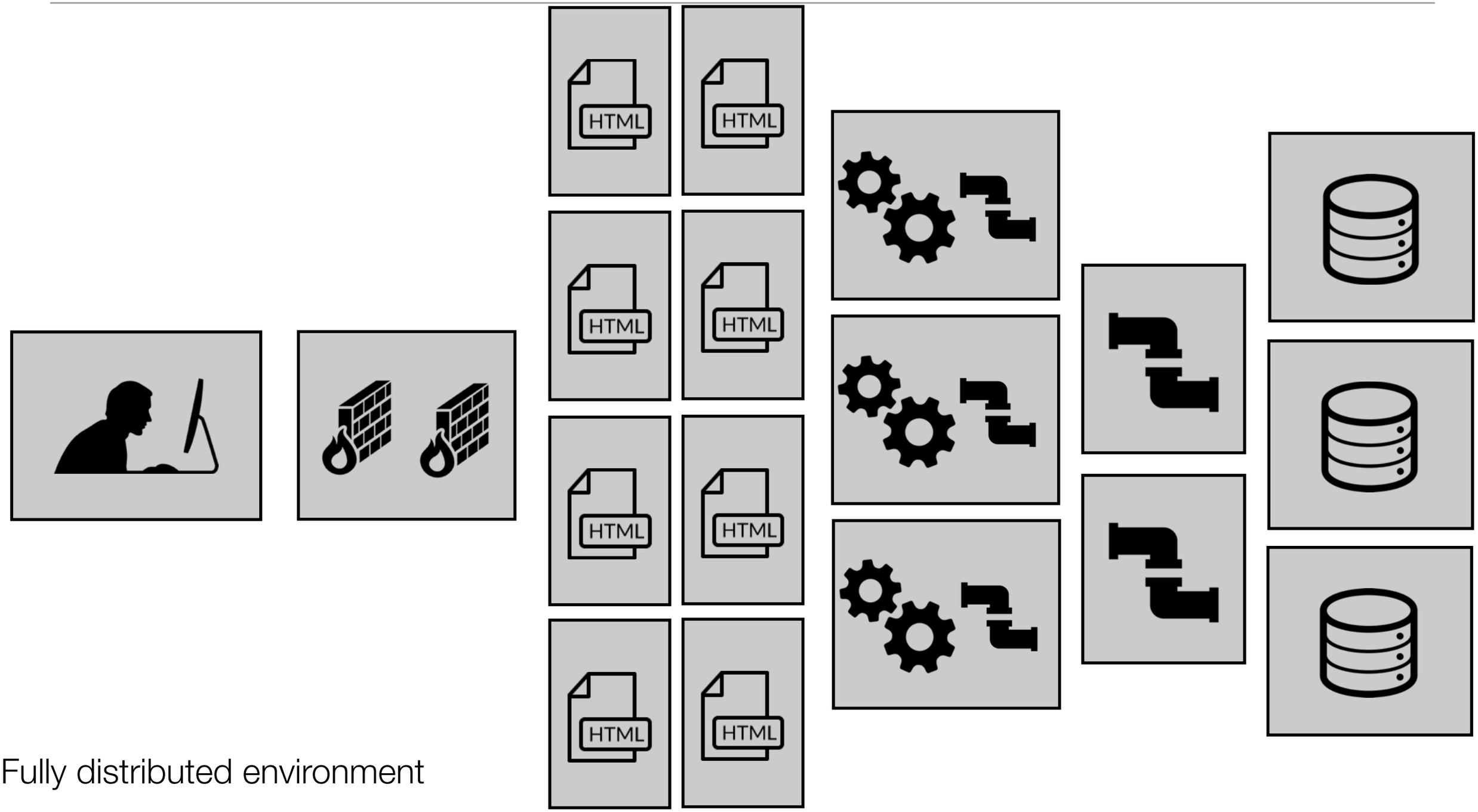
What is the mapping between tiers, **logical nodes** and **physical nodes**?



Typical environment for scalability and availability

User | **Client** | **Presentation** | **Business** | **Integration** | **Resources**

What is the mapping between tiers, **logical nodes** and **physical nodes**?



Fully distributed environment

User | **Client** | **Presentation** | **Business** | **Integration** | **Resources**

What is the mapping between tiers, **logical nodes** and **physical nodes**?

- Just because it is **possible to physically separate the presentation and the business tier** does not mean that it is always a good idea!
- One reason to keep them in the same node is that **invoking a remote service is a lot slower than invoking a local service**.
- Introduce a **reverse proxy** as soon as possible. In simple environments (e.g. dev.), it does not require a physical node.
- **Database nodes** are often shared by several applications. Optimization techniques are fairly different for database and application nodes. They are often physically separated.

(Rapidly evolving) virtualization technologies gives us more options. Think about **Docker**, which allows you to “run a data center” on your laptop.

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

Java Enterprise Edition

What is Java Enterprise Edition?

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

- 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.

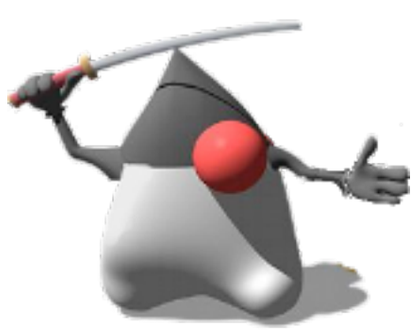
Expert Group

Barreto, Charlton	BEA Systems	Borland Software Corporation
Capgemini	Chandrasekaran, Muralidharan	Crawford, Scott
Dudney, Bill	E.piphany, Inc.	Genender, Jeff
Hewlett-Packard	IBM	Ironflare AB
Kohen, Elika S.	Leme, Felipe	Novell, Inc.
Oracle	OW2	Pramati Technologies
Pratap, Rama Murthy Amar	Raible, Matt	Red Hat Middleware LLC
Reinshagen, Dirk	SAP AG	SeeBeyond Technology Corp.
Shah, Suneet	Sun Microsystems, Inc.	Sybase
Tiwari, Ashish	Tmax Soft, Inc.	Trifork
Umapathy, Sivasundaram		

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...
- Java EE 7 has been adopted on May 28th, 2013
- Java EE 8 is underway…

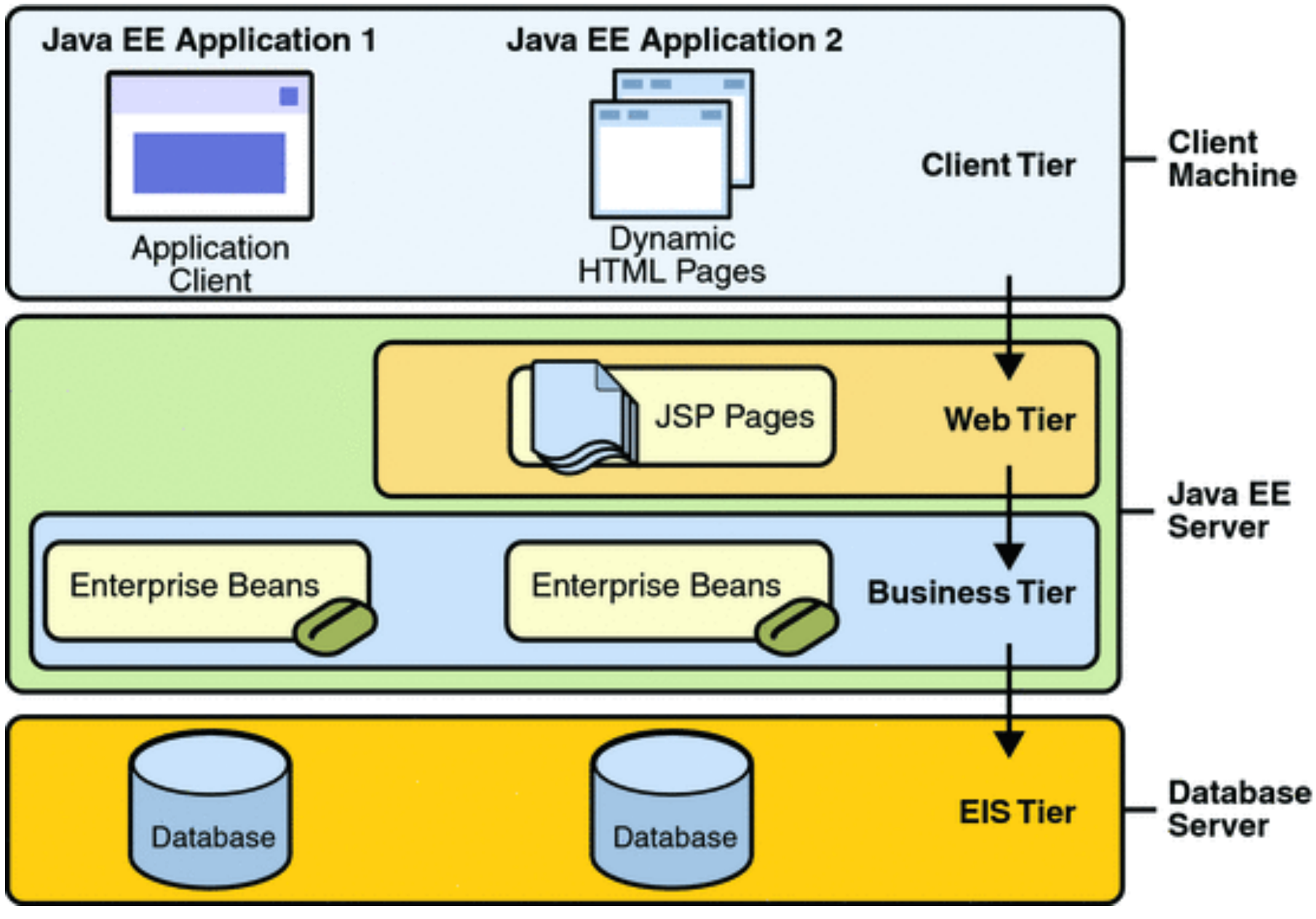


Watch out! Java EE (formerly known as J2EE) **has evolved a lot** over the years!

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

Today's agenda

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

13:15 - 13:30	15'	Intro: multi-tiered apps and Java EE
13:30 - 14:15	**45'**	**AMT-Discovery**
14:15 - 14:30	15'	Guidelines for phase 1
14:30 - 15:30	60'	Individual work
15:30 - 15:40	10'	Wrap-up

AMT-Discovery

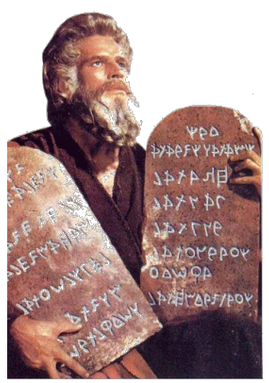
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

NetBeans

IntellijIDEA

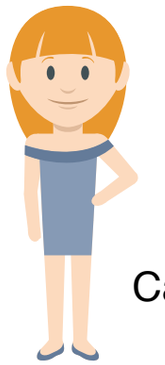
Apache maven

Alice is a developer



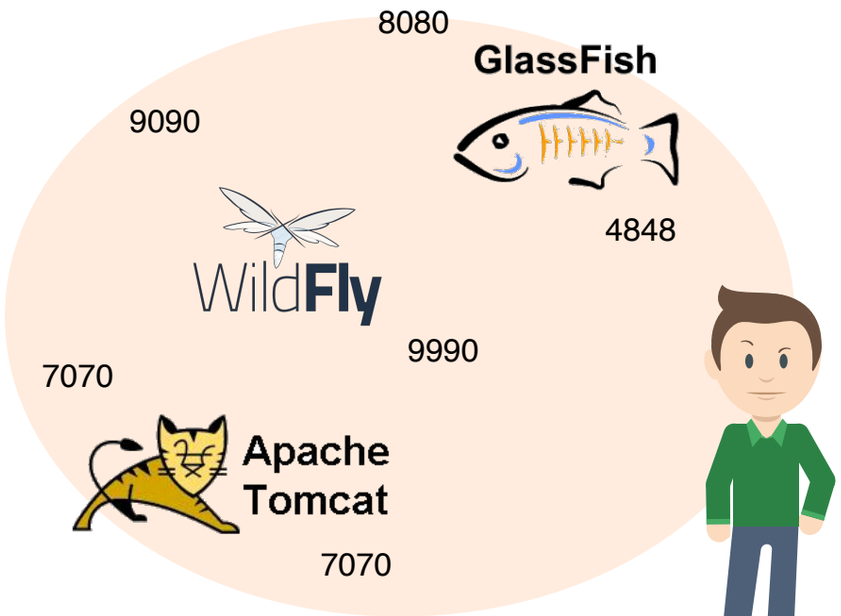
Java EE is a specification

The Java EE development cycle



Carol is a end-user

Application packaged in a .war or .ear file



AMT-Discovery

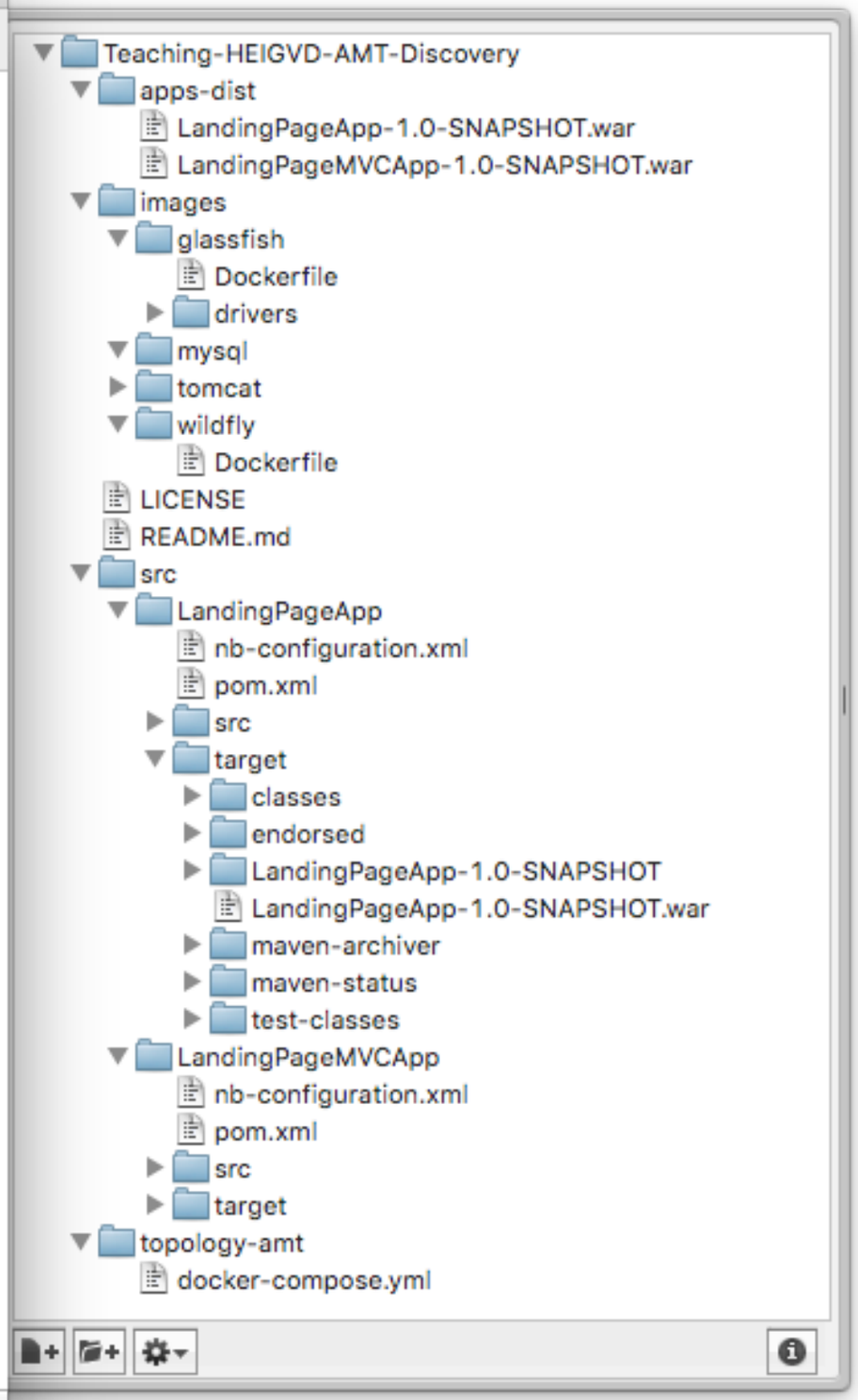
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

docker-compose.yml — Teaching-HEIGVD-AMT-Discovery

docker-compose.yml

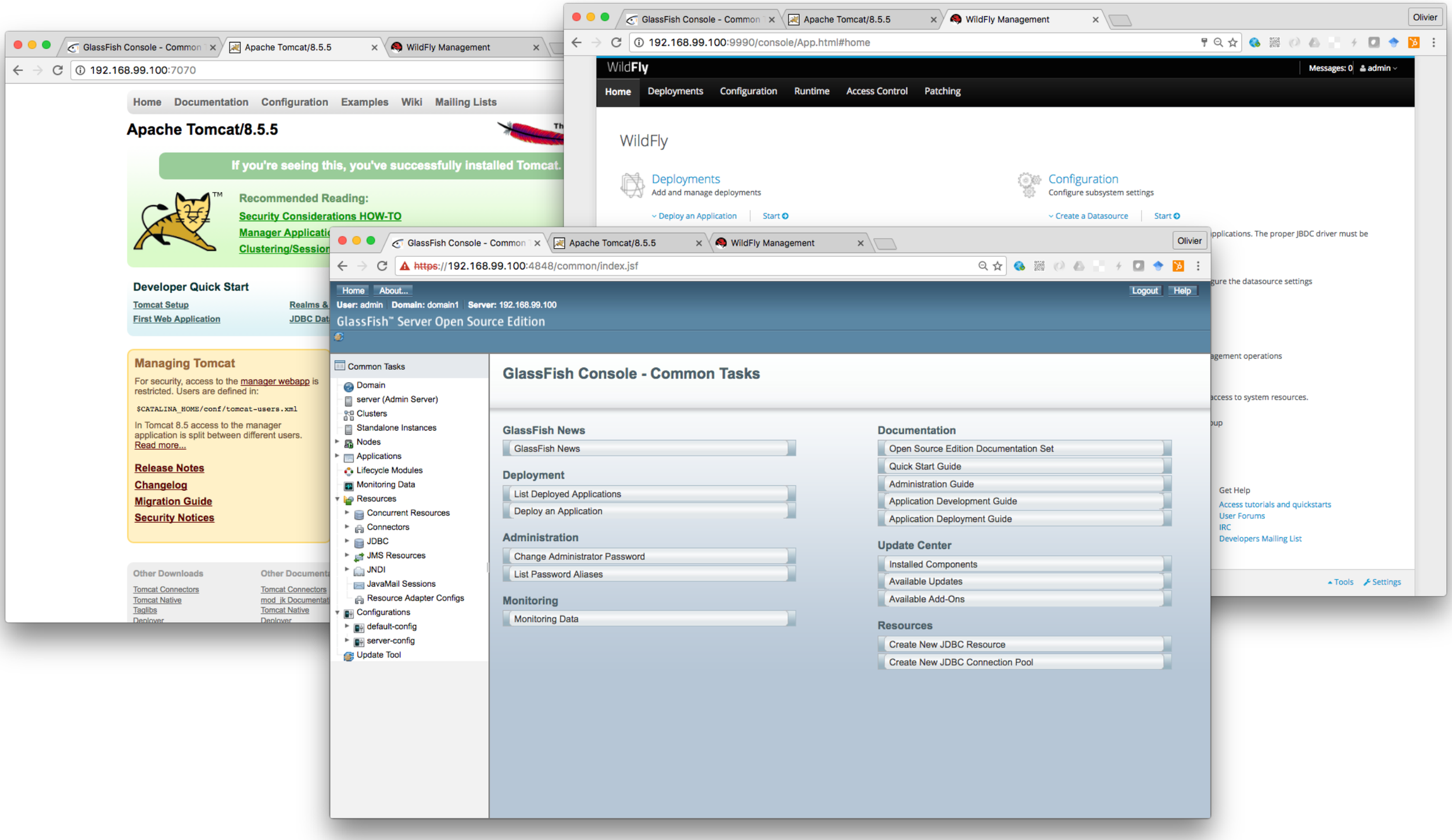
```yaml
version: '2'
services:
  glassfish:
    build: ../images/glassfish
    ports:
      - "8080:8080"
      - "4848:4848"
  wildfly:
    build: ../images/wildfly
    ports:
      - "9090:8080"
      - "9990:9990"
  tomcat:
    build: ../images/tomcat
    ports:
      - "7070:8080"
```

Line: 16 Column: 19 YAML Soft Tabs: 4



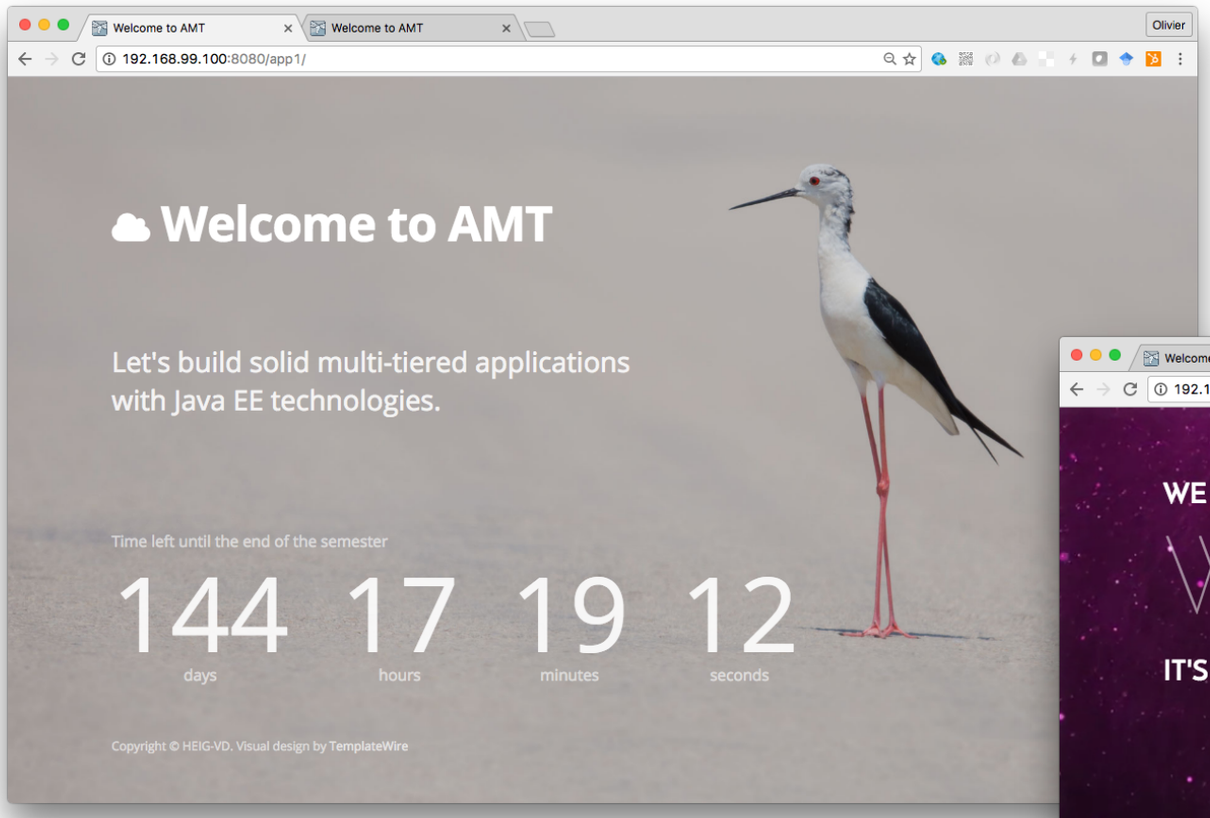
AMT-Discovery

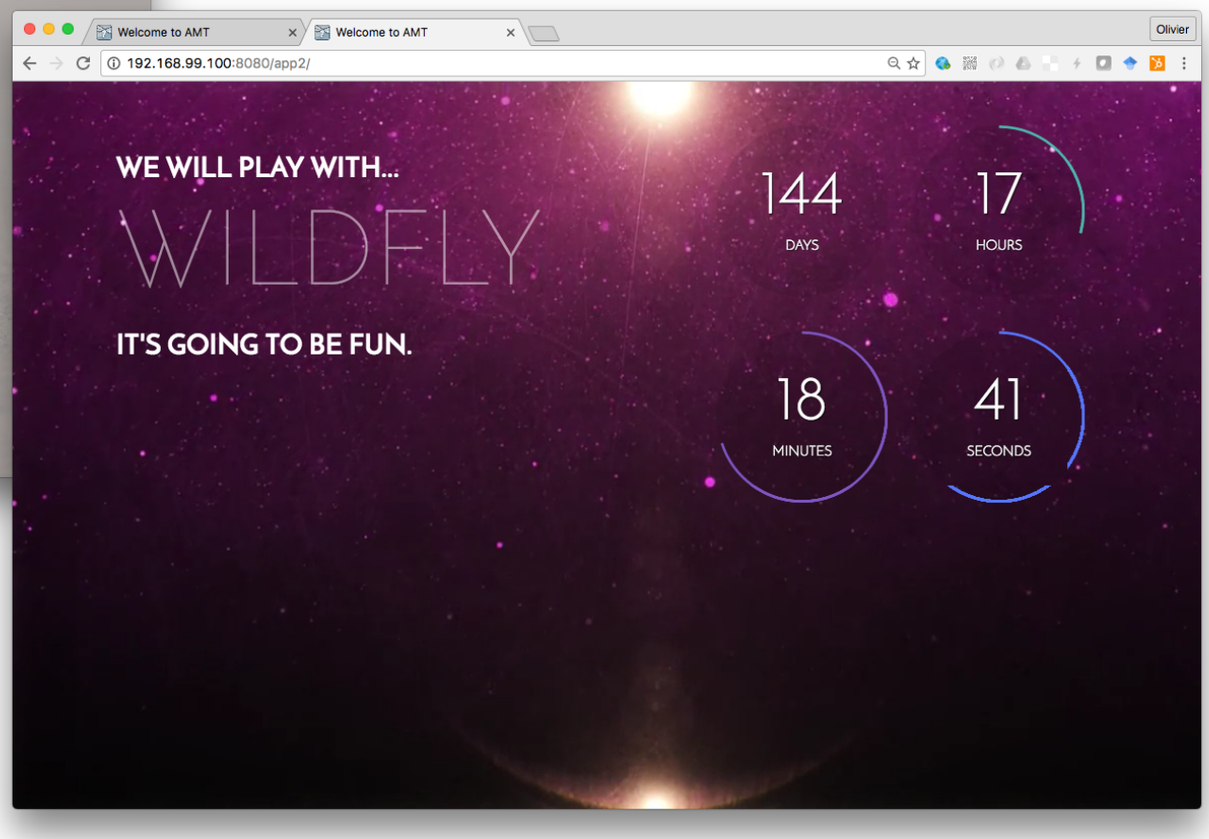
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



AMT-Discovery

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





AMT-Discovery

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

- **Install the last version of docker**
  - We will use docker compose
- **Fork and clone the AMT-Discovery repo**
  - https://github.com/SoftEng-HEIGVD/Teaching-HEIGVD-AMT-Discovery
- **Follow the instructions in the repo**
  - You should have 3 different application servers running in 3 docker containers (Apache Tomcat, Oracle Glassfish and JBoss Wildfly)
  - You should deploy 2 Java EE apps (.war packages) in each app server.
- **Webcasts**
  - https://www.youtube.com/playlist?list=PLfKkysTy70Qa7tSIkbsvOrRc6Ug_c0nZz
  - Bootcamp 1.1, 1.2, 1.3 and 1.4

Today's agenda

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

13:15 - 13:30	15'	Intro: multi-tiered apps and Java EE
13:30 - 14:15	45'	AMT-Discovery
14:15 - 14:30	**15'**	**Guidelines for phase 1**
14:30 - 15:30	60'	Individual work
15:30 - 15:40	10'	Wrap-up

Project - Feature 1 acceptance criteria

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

As a guest, I can visit the platform home page.

- You can work in **pairs of 2 students** (one can focus on creating the webapp, the other on the docker image)
- Every team must create **a new GitHub repo** (not a fork from AMT-Discovery)
- Every team must decide to work either with **Glassfish** or **Wildfly**
- Every team must **create a new simple webapp** that displays a welcome message (Netbeans is recommended). Pick a free HTML template to have something visually attractive.
- For the feature to be "**done**", the customer must be able to:
  - Clone the student repo
  - Have instructions in the **README.md** that tell me where to run **`docker-compose up`**
  - Have instructions in the **README.md** that tell me which URL to access (e.g. http://192.168.99.100:6060/platform)
  - See your welcome message in a nice HTML template.

Feature 1 tasks

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

Search for information about the 'autodeploy' feature of Glassfish or Wildfly.

Prepare a custom Docker image, so that the .war is deployed when the container start. Use the AMT-Discovery .war files while the new app is being developed.

Open the AMT-Discovery apps in Netbeans and inspect structure.

Prepare the Github repo.

Create a new Java EE web app (with Netbeans) and generate a .war package.

Find a nice HTML template.

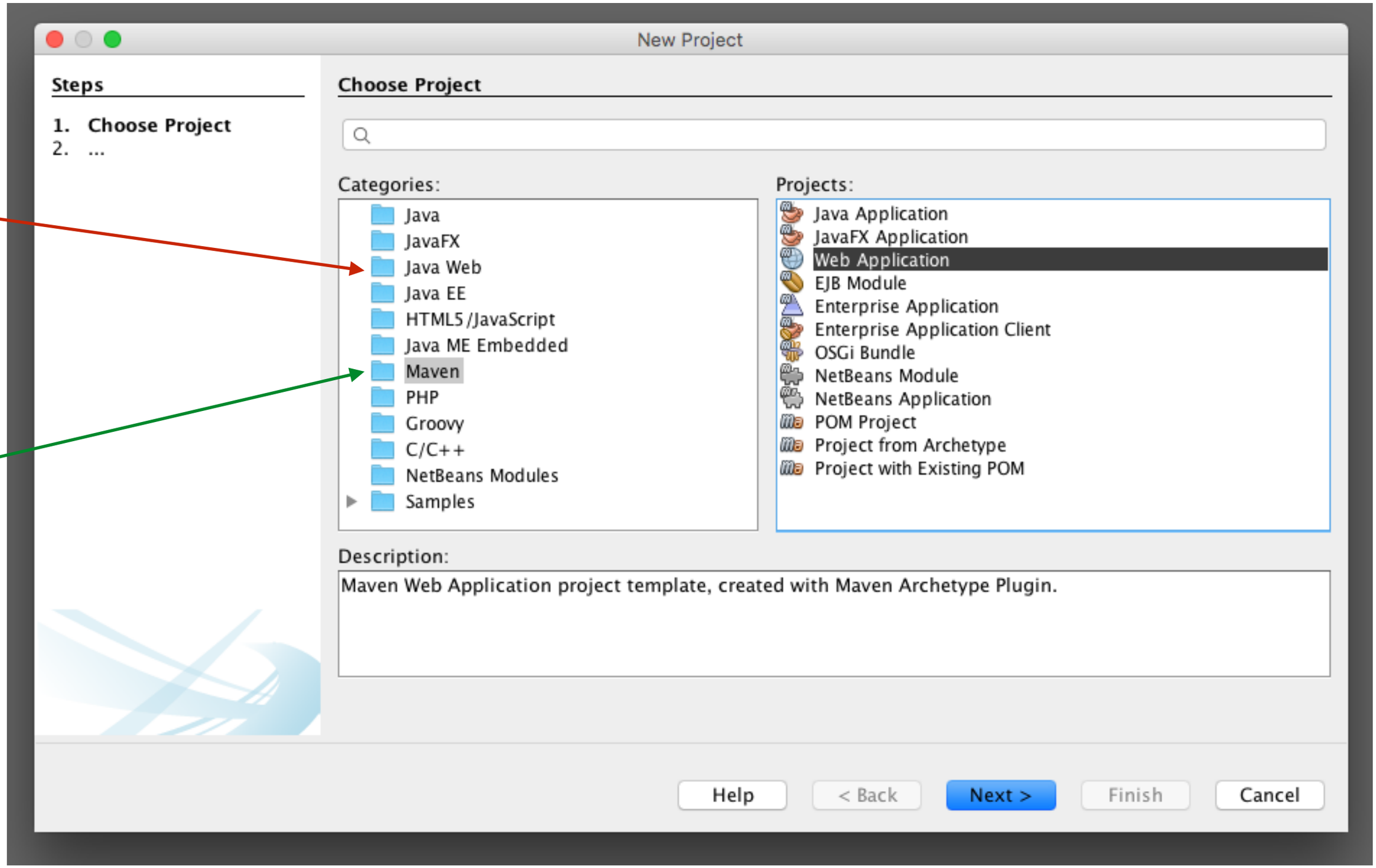
Integrate the final .war package in the Docker image.

Integrate the HTML template in the web app sources

Validate the acceptance criteria.

Create a new Java EE webapp

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



Documentation

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

- https://glassfish.java.net/documentation.html
- https://docs.jboss.org/author/display/WFLY10/Documentation