JPassion.com

JASMINE CONSEIL

JAVA EE Application Development in Practice

Chapter 1: A Big Overview

Sang Shin & Karim Djaafar

AGENDA

- What is Java EE
- JEE Architecture
- Enterprise Application Development
- Java EE, a little story
- The Java EE APIs
- What's new in Java EE 8
- Quick recap

What is Java EE ?

Fundamentals Concepts and Architecture Overview

Introduction to Java EE

- The Java Platform, Enterprise Edition (Java EE) is a collection of API specifications designed to work together when developing server-side, enterprise Java applications
- Extension of Java SE
- Simplify enterprise application development
- Java EE is a standard : there are multiple implementations of the Java EE specifications

Java EE is a Standard

- Java EE go to a standardization process of the JCP, Java Community Process, an organization responsible for the development of Java technology
- JCP members include Oracle (the current steward of the Java platform), and the Java community at large
- The Java Community Process (JCP) allows interested parties to assist in developing standard technical specification for Java technology
- Each Java EE API specification is developed as part of a Java Specification Request (JSR)
- Each JSR is assigned a unique number. JavaServer Faces (JSF) 2.3 is developed as JSR 372, for instance

Java EE Implementation Server Examples

Payara

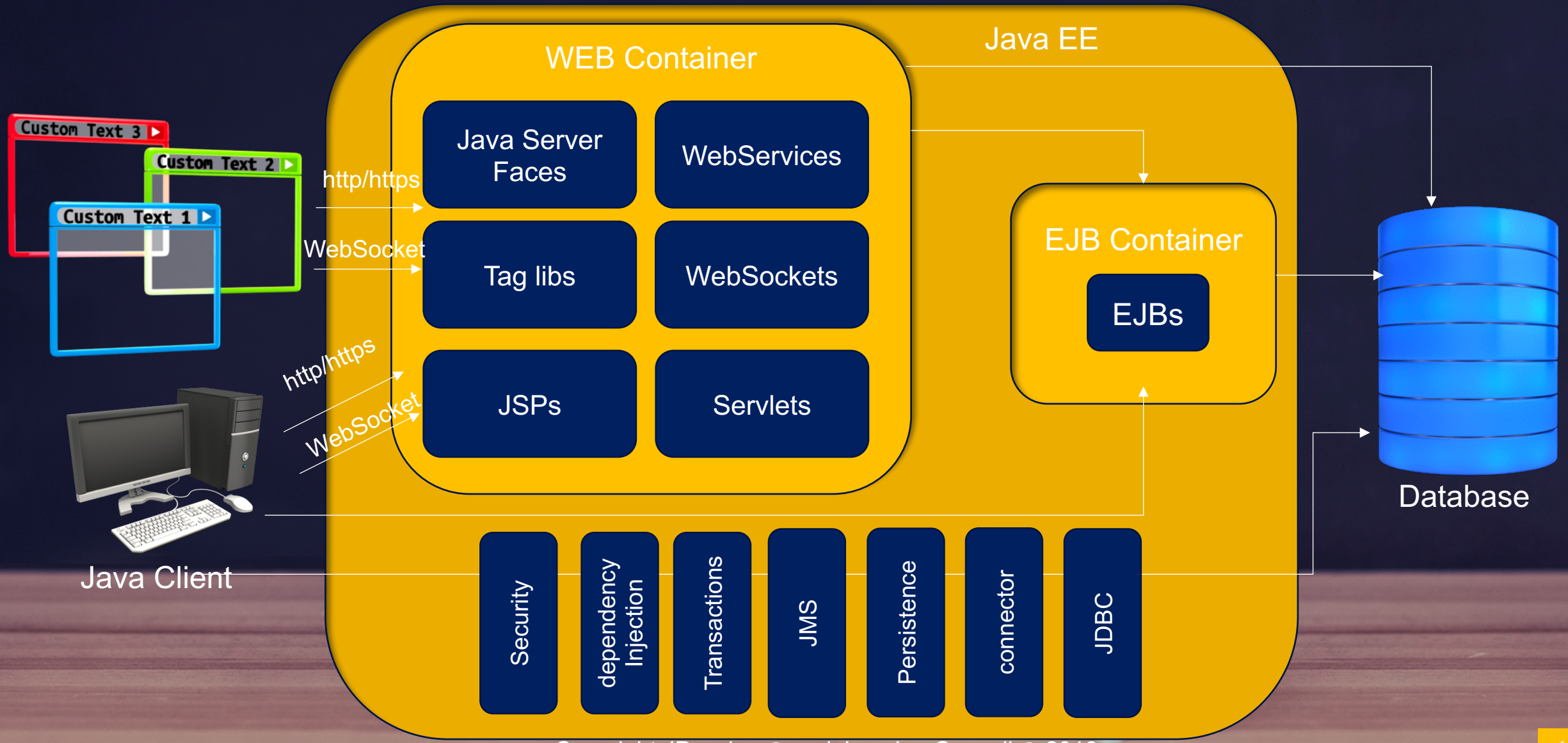
- In this course, we will choose Payara, which support full Java EE 8 specification and JDK 9
- Payara server support ***Microservices***, which leverage standard Java EE API
- Easy to use and deploy : For example It only takes a few quick steps to deploy a Microservice using Java EE APIs !

JEE Architecture

JEE Concepts

- Java Enterprise Edition is an umbrella specification
- Container, services and Components
  - Web Container
  - EJB Container
- Application Client Container (ACC)
- Packaging and Deployment
- Client interaction

JEE Big picture



Java SE and Java EE: a complementary couple

- **Java SE**
  - APIs for example to handle collections
  - The JVM is a container
  - Low level services
- **Java EE**
  - APIs to handle transaction, messaging, security…
  - Code runs in a container
  - High level services using metadata (see further)

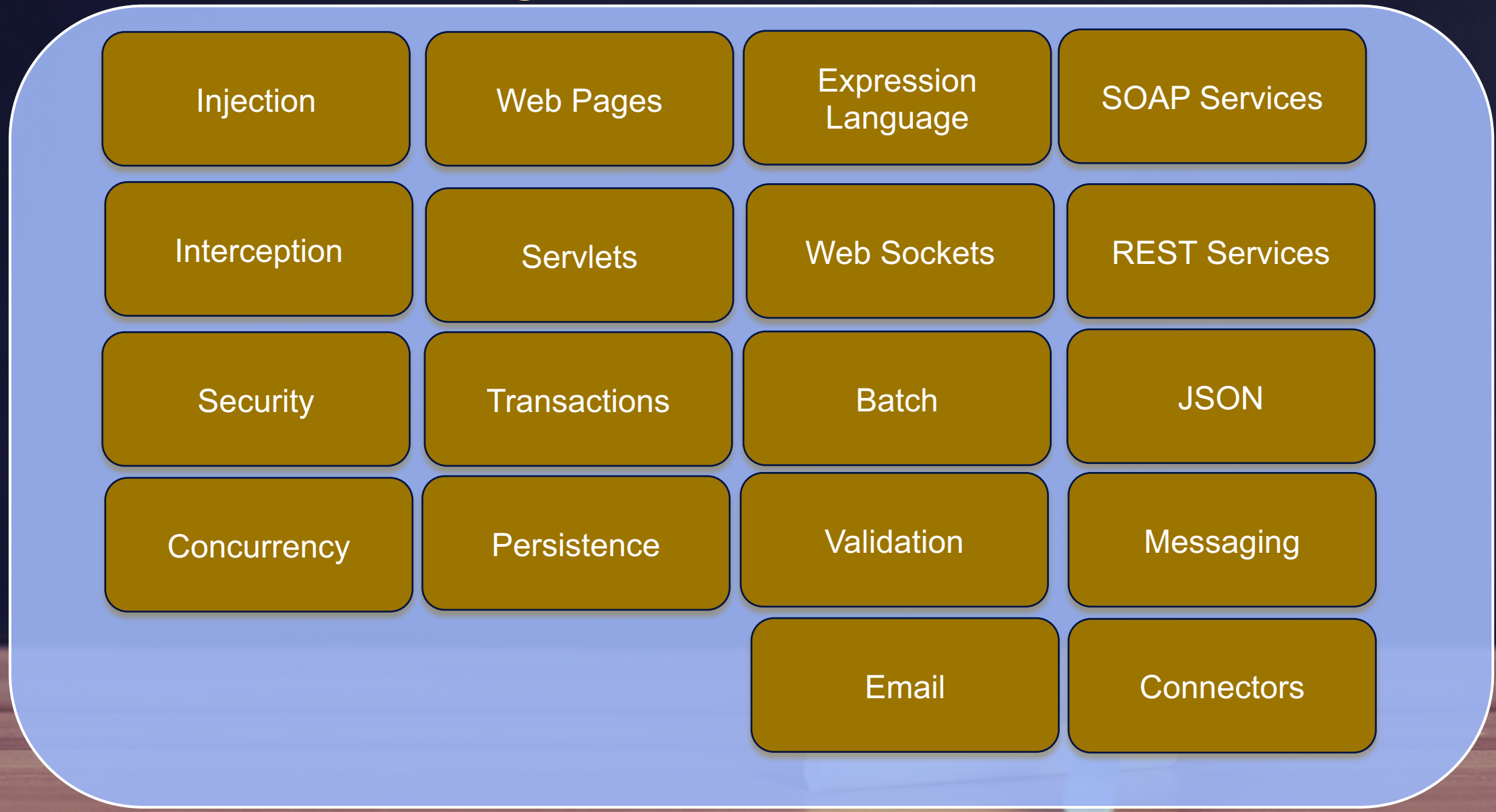
JEE Container

- Runtime environment
- Hide technical complexity
- Enhance portability
- Host applications
- Administer and monitor hosted Java Applications
- 4 types of containers:
  - Web Container
  - EJB Container
  - Application Client Container
  - Applet (obsoleted)

Services and Components

- A ***component*** is a static or dynamic web pages
  - Server side classes
  - Handle business code
- A component can be a business or a technical component (example database access component)
- Container provides services to component using some metadata description
  - Transaction Management
  - Security
  - Naming
  - ...
- Services are configurable

JEE Architecture



Java EE Application

- Aggregation of components
- Examples:
  - Web pages
  - Web resources
  - Database access components
- Packaged and Deployed in an archive according a standard format (EAR, WAR, JAR)

Packaging JEE application

- Aggregation of components
- Examples:
  - Web pages
  - Web resources
  - Database access components
- A Java EE application is delivered in a Java Archive (JAR) file, a Web Archive (WAR) file, or an Enterprise Archive (EAR) file

Client Interaction

- Web Clients using a browser (IE, google chrome, …)
- Standalone Java applications (Java SE, Java FX)
- Web Applications
- REST/SOAP services
- Mobile Apps

Enterprise Application Development

Applications

Java SE
Standard
Applications

Java Mobiles
Applications

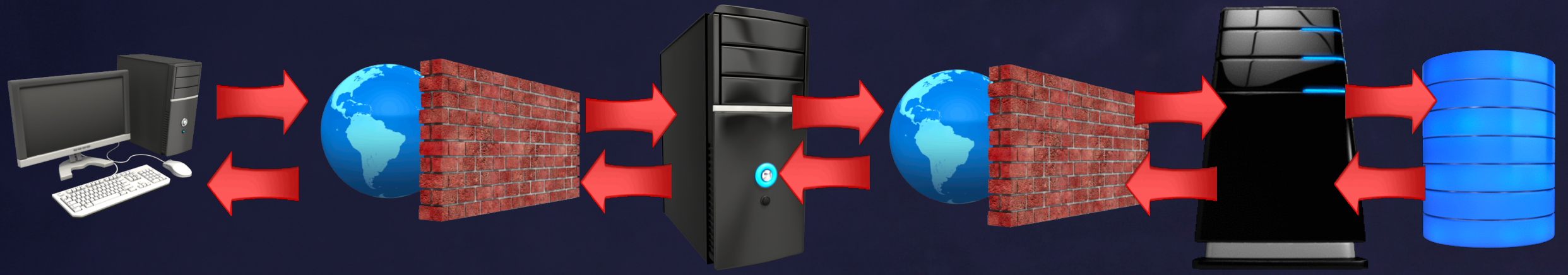
Java FX
Rich User interfaces

Java EE
Enterprise Applications

Enterprise Applications

- Multi-tiered
- Scalable
- Modulable
- Secure
- Respond to the needs of large and complexes business needs
- For individual developers and small organizations

JEE Tiered Applications



Client tier

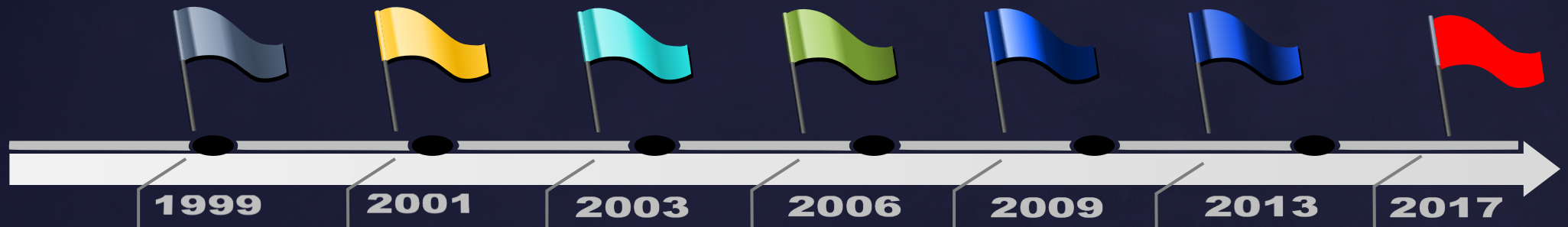
Web Server

Application Server

Database

Java EE, a little story...

A Brief History of Java EE



J2EE 1.2
Servlets, JSP, EJB, JMS nt.

J2EE 1.3
CMP,JCA .

J2EE 1.4
JAX/RPC Management API, Deployment.

Java EE 5
Annotations, EJB 3.0, JPA, JSF, JAXB, JAX-WS

Java EE 6
JAX-RS, CDI, Servlet 3.0, Web Profile, Pruning

Java EE 7
JAX-RS 2.0, JSON-P, WebSocket, JMS 2.0, Batch

Java EE 8
Servlet 4.0, JAX-RS 2.1, JSON-B, JSON-P 1.1, Security

Java EE 8 main evolutions

- **Java SE 8 alignement**: DateTime API, CompetableFuture, repeatable annotations
- **CDI 2.0**: asynchronous events, events ordering, better integration in other specs. With this release, CDI confirms its role of foundation of the Java EE platform
- **Servlet 4.0:** HTTP/2 support (Server Push)
- **JAX-RS 2.1:** Server-Sent Event, reactive extensions
- **JSON Processing 1.1** and **JSON Binding 1.0**
- **Security:** simplification, secret management, modernization, OAuth2 and OpenID support

Where is Java EE going ?

- Oracle announced its intention to open source Java EE (Enterprise Edition) in order **to make Java EE more agile and responsive** to changing industry and technology demands (According David Delabassee, Oracle Java EE evangelist)
- It seems that Oracle prefers to concentrate its effort on a new open source project named Fn, a serverless framework similar to Amazon Lambda and IBM OpenWhisk (now an Apache project)

The Java EE APIs

Java EE is a collection of APIs

- Application Programming Interface specifications
- Used to develop server-side, enterprise Java application
- Some APIs supported by Java EE:
  - JavaServer Faces (JSF) 2.3
  - Servlet 4.0
  - Java Persistence API (JPA) 2.2
  - Enterprise JavaBeans (EJB) 3.2
  - Contexts and Dependency Injection for the Java EE Platform (CDI) 2.0
  - Java API for JSON Processing (JSON-P) 1.1
  - Java API for JSON Binding (JSON-B) 1.0
  - Java API for WebSocket 1.0
  - …

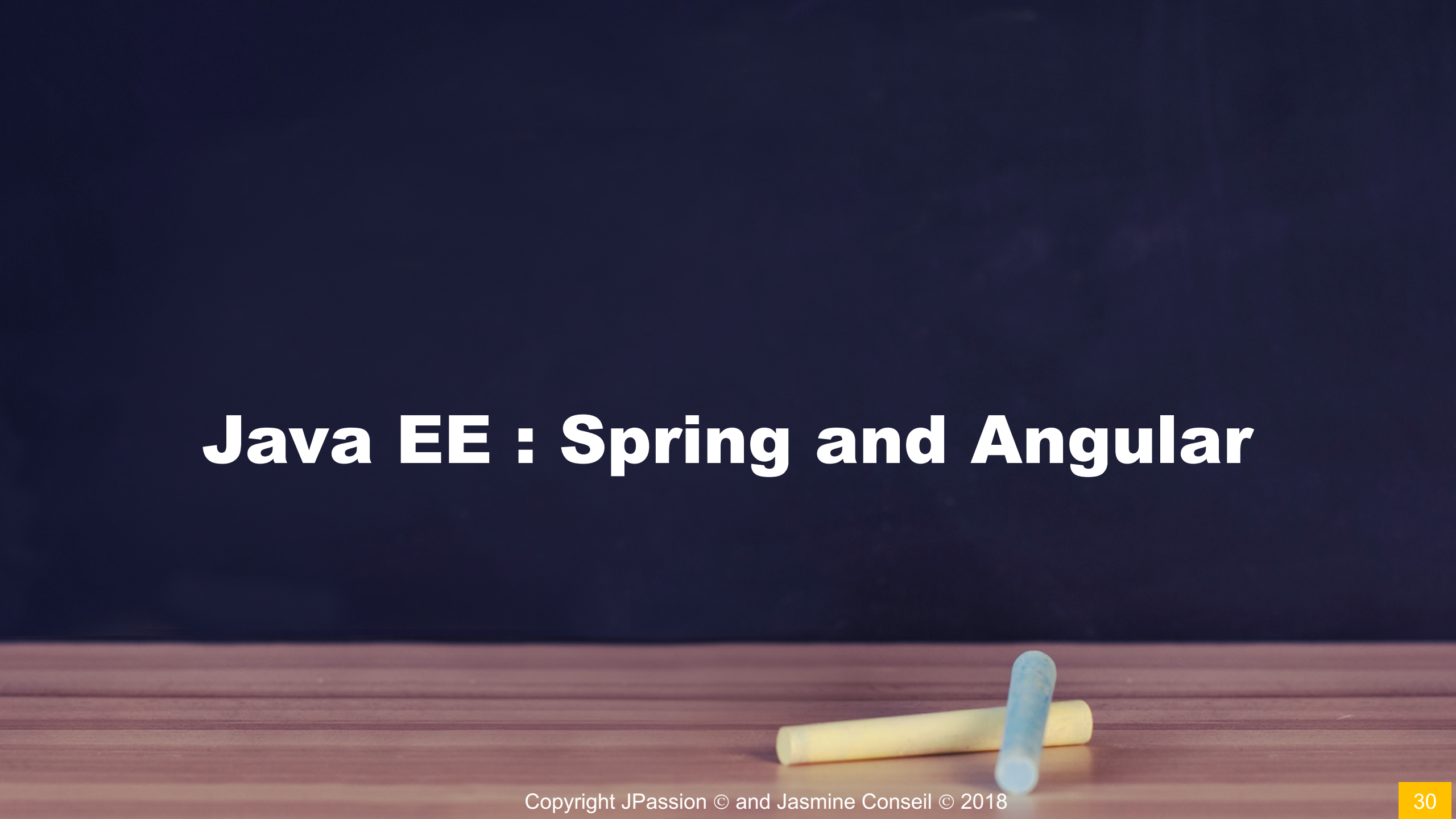
Java EE API

Java EE API	Description
JavaServer Faces (JSF) 2.3	JSF is a component library that greatly simplifies the development of web applications
Java Persistence API (JPA) 2.2	JPA is the Java EE standard Object-Relational Mapping (ORM) API. It makes it easy to interact with relational databases
Enterprise JavaBeans (EJB) 3.2	EJB's allow us to easily add enterprise features such as transactions and scalability to our Java EE applications
Contexts and Dependency Injection (CDI) 2.0	CDI allows us to easily define the life cycle of Java objects and provides the ability to easily inject dependencies into Java objects; it also provides a powerful event mechanism
Java API for JSON Processing (JSON-P) 1.1	JSON-P is an API that allows working with JSON strings in Java
Java API for JSON Binding (JSON-B) 1.0	JSON-B provides the ability to easily populate Java objects from JSON streams and back

Java EE API

Java EE API	Description
Java API for WebSocket 1.0	WebSocket is a standard Java EE implementation of the Internet Engineering Task Force (IETF) WebSocket protocol, which allows full duplex communication over a single TCP connection
Java Message Service (JMS) 2.0	JMS is a standard API that allows Java EE developers to interact with Message Oriented Middleware (MOM
Java EE Security API 1.0	The Java EE Security API aims to standardize and simplify the task of securing Java EE applications
Java API for RESTful Web Services (JAX-RS) 2.1	JAX-RS is an API for creating RESTful web services endpoints and clients
Java API for XML Web Services (JAX-WS) 2.2	“JAX-WS is an API that allows the creation of Simple Object Access Protocol (SOAP) web service
Servlet 4.0	The servlet API is a low-level API used to implement server-side logic in web applications

Java EE : Spring and Angular



Java EE and Spring

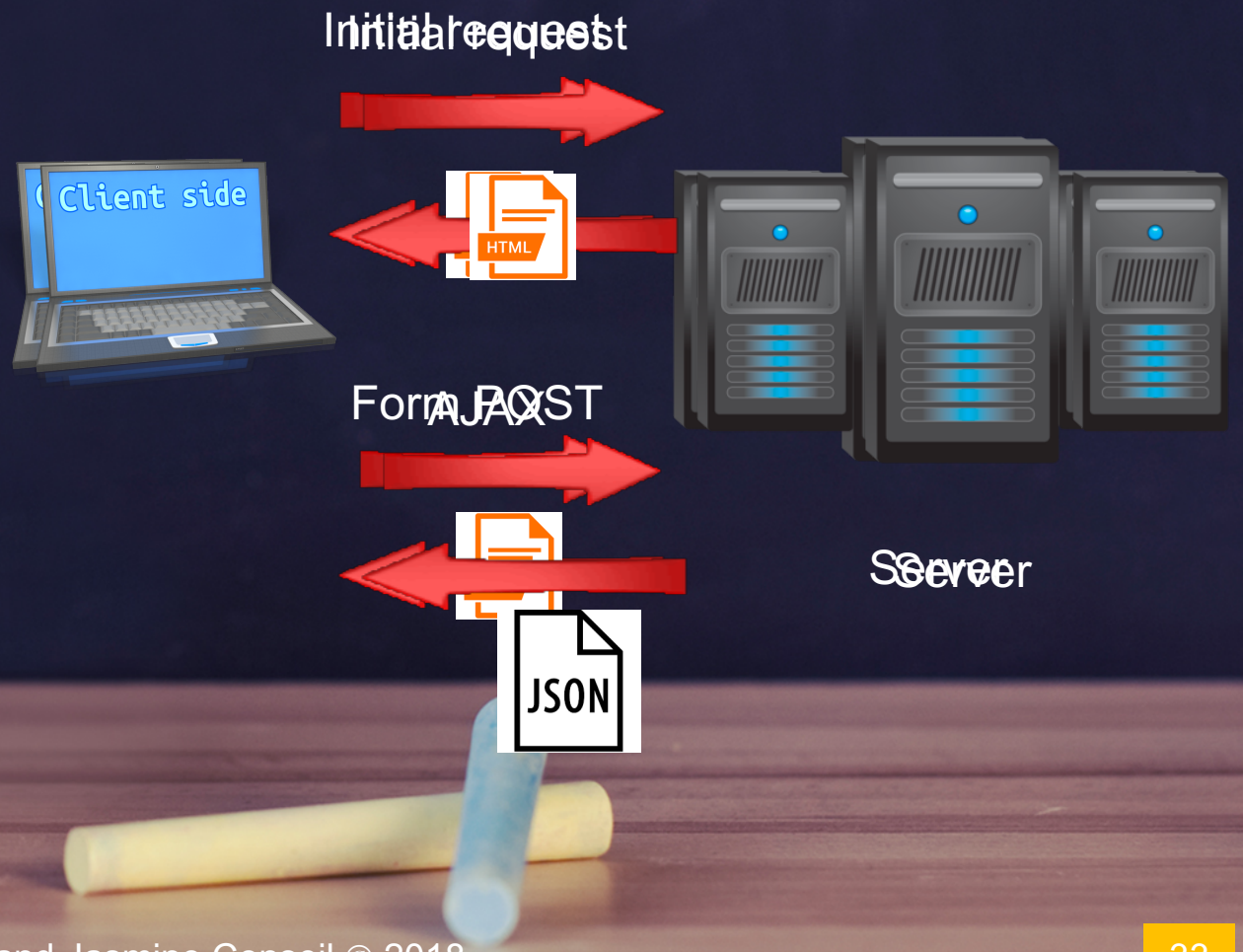
- Spring and Java EE are pretty popular choices, and the general consensus is that both are quite excellent
- Java EE influenced by Spring Framework and Spring Boot by Java E
- Think about it:
  - if someone uses servlets, do they use Java EE? It's a part of the specification, sure, but really? Would you call it a Java EE application? What about if you sprinkle your model with some JPA annotations? Where's the line drawn?

Java EE meet Angular

- The frontend of web applications were, in most of the cases, *server-side* rendered HTML pages, powered by JSP or JSF
- The HTML was crafted on the server on demand, that is, on request, and returned to the client
- In order to realize that, the JSP or JSF pages, respectively, have to reside on the backend
- Therefore, the whole enterprise application would be shipped and deployed as single artifact
- With the new frontend technologies, basically sophisticated JavaScript frameworks and especially **single page applications (SPA**), those technologies as been deprecated

What is SPA (in short) ?

- **Single-Page Applications** (SPAs) are Web **apps** that load a **single** HTML **page** and dynamically update that **page** as the user interacts with the **app**
- SPAs use AJAX and HTML5 to create fluid and responsive Web apps, without constant page reloads



Java EE and Angular: the JavaScript era

- The more client-centric and powerful the JavaScript frameworks became, the more the communication between frontend and backend went from tightly coupled requests and responses to a more *API-like* usage, typically JSON via HTTP
- This also meant that the server-side became more client-agnostic
- For example, communicating solely via *RESTful-like*, JSON-format APIs enables native or mobile clients such as smartphones to use the same API like the frontend does
- Angular support this new shift

Angular in short

- Extend HTML to add dynamic nature so that we can build modern web applications with ease
- Brings you back to HTML
- Declarative approach
- Eliminate the need of DOM manipulation by two way bindng mechanism
- Ideal for building powerful SPA applications
- Enhance JEE by combining the power of JEE for the backing and the flexibility of the AngularJS framework
- We will use Angular as a front stack during our course combined to JEE APIs

The Future of Java EE...

The Jakarta EE Project

- Last August, Oracle announced its decision to move the Java EE platform to an open source foundation, and after discussing with several candidates, the company chose the Eclipse Foundation
- With Java EE 8 now available, the Eclipse Foundation has begun to take charge by launching the **Eclipse Enterprise for Java (EE4J) project, code name Jakarta EE**
- It is a high-level project of the foundation that aims to define and develop API specifications, reference implementations, and technology compatibility kits (TCKs) for Java application server implementations and cloud natives implementation

Summary

In Summary

A powerful platform

Java EE Is a powerful platform for developing powerful Java Enterprise application

Standard

Java EE is based on specifications which each vendor can adapt

Adapted to the Market constraints

Java EE evolve and can be combined with other powerful frameworks thanks to is modular architecture