Platforms and Frameworks

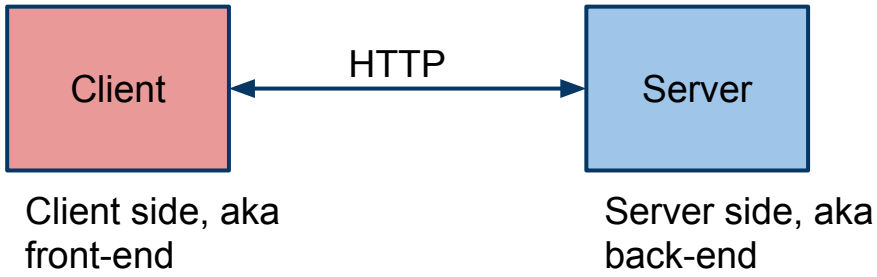
BWA Slides #1

Web Application Characteristics

It's a client/server stateless pull architecture*)

Client side (browser) sends HTTP-requests (the pull) and receives HTTP-responses

Server side receives HTTP-requests and sends HTTP-responses



*) Will present another approach later

Prerequisites

Not feasible to write a web application from scratch

- Networking, parsing, concurrency, etc, …

Need to decide on

- Platform (hardware, OS, runtime libraries)
- Framework(s). Will give the application structure and supply services
- Middleware, external APIs
- Tools (testing and more)

Programming model

Many MBs before a single row of application written (aka programming by bulldozer)

Platforms and Frameworks

Platform = Huge "ecosystem" for building/ running software

- Win/.NET/libraries
- Android/Linux/Java/libraries
- LAMP: Linux/Apache/MySql/PHP/libraries
- iOS/iPhone/libraries

Framework = generic structure and service provider for a specific type of application. Frameworks are built on top of platforms

- Some call an API for a framework, I try to distinguish (API no structure)
- Game frameworks
- Web app frameworks
- ...

Framework Philosophy

Have a look at a few web frameworks for client and/or server side!

How are we supposed to survive as developers?

- Very hard to just get an overview of area!
- Very hard to make own evaluations, too many choices…
- How much time should we invest to learn, possible gone after a few years?
- Area suffers from “hypes”

Course Platform and Framework

Requirements

- Preferably Java based (student educational background)
- Should be platform independent (students use different platforms, should be possible to move application)
- Should be open source (student money gone to … ?)
- Should adhere to some standard …(i.e. many implementations)

Pedagogical issues

- Learning curve vs other … (time, quality, …)

Server Side Platform and Framework

Platform: **Java Enterprise Edition 7 (JEE 7)**

" … provides an API and runtime environment for developing and running enterprise software, including network and web services, and other large-scale, multi-tiered, scalable, reliable, and secure network applications.
// Wikipedia

- Owned by Oracle (open source)
- Many developer communities
- Open source, platform independent
- Huge…!

Framework; None

- Have tried Spring, no real success
- Frameworks too time consuming to learn
- We do some in-house design

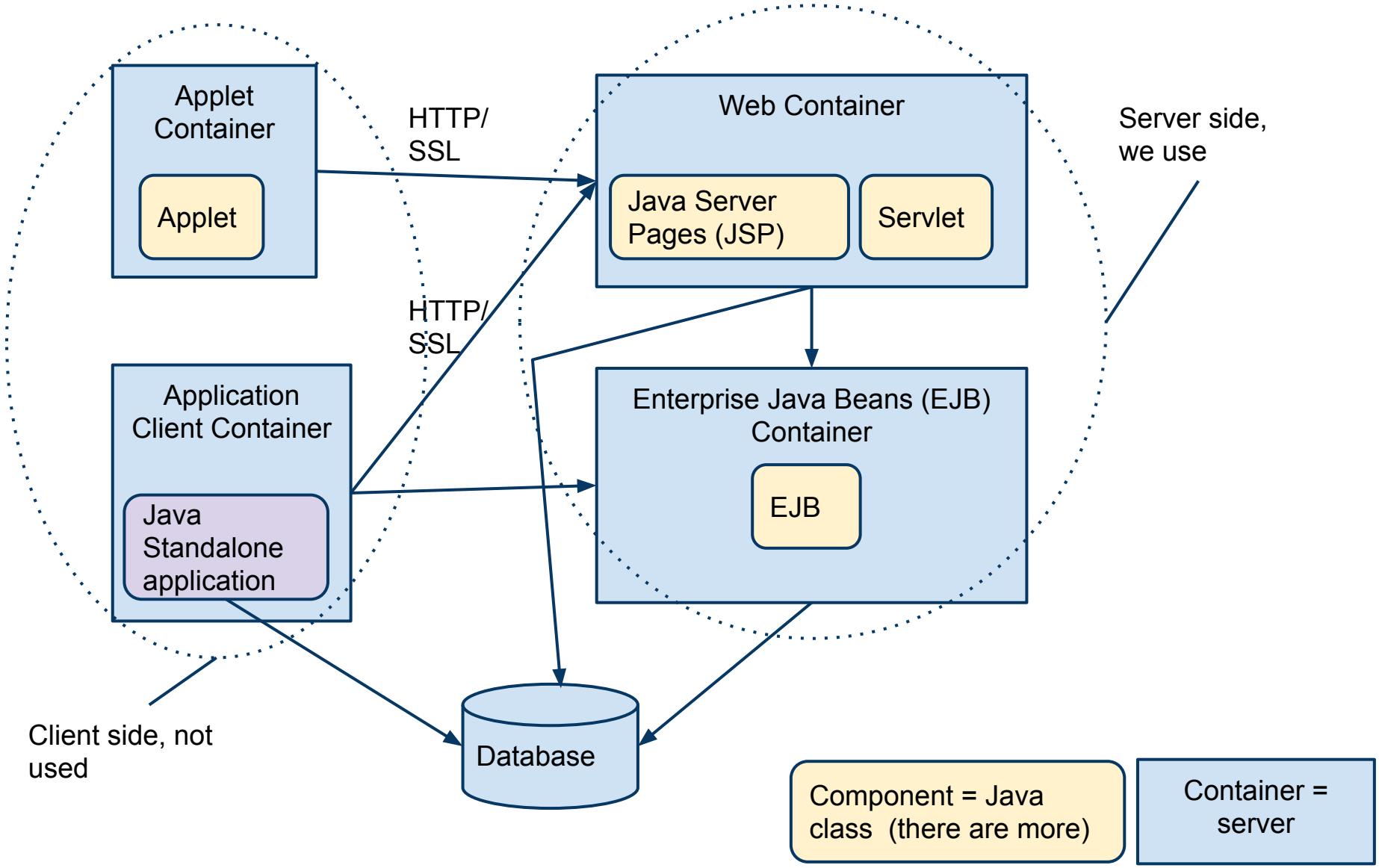
Java Enterprise Edition 7

JEE 7 is an umbrella specification (i.e. not a product) for web/enterprise multitier applications

Defines

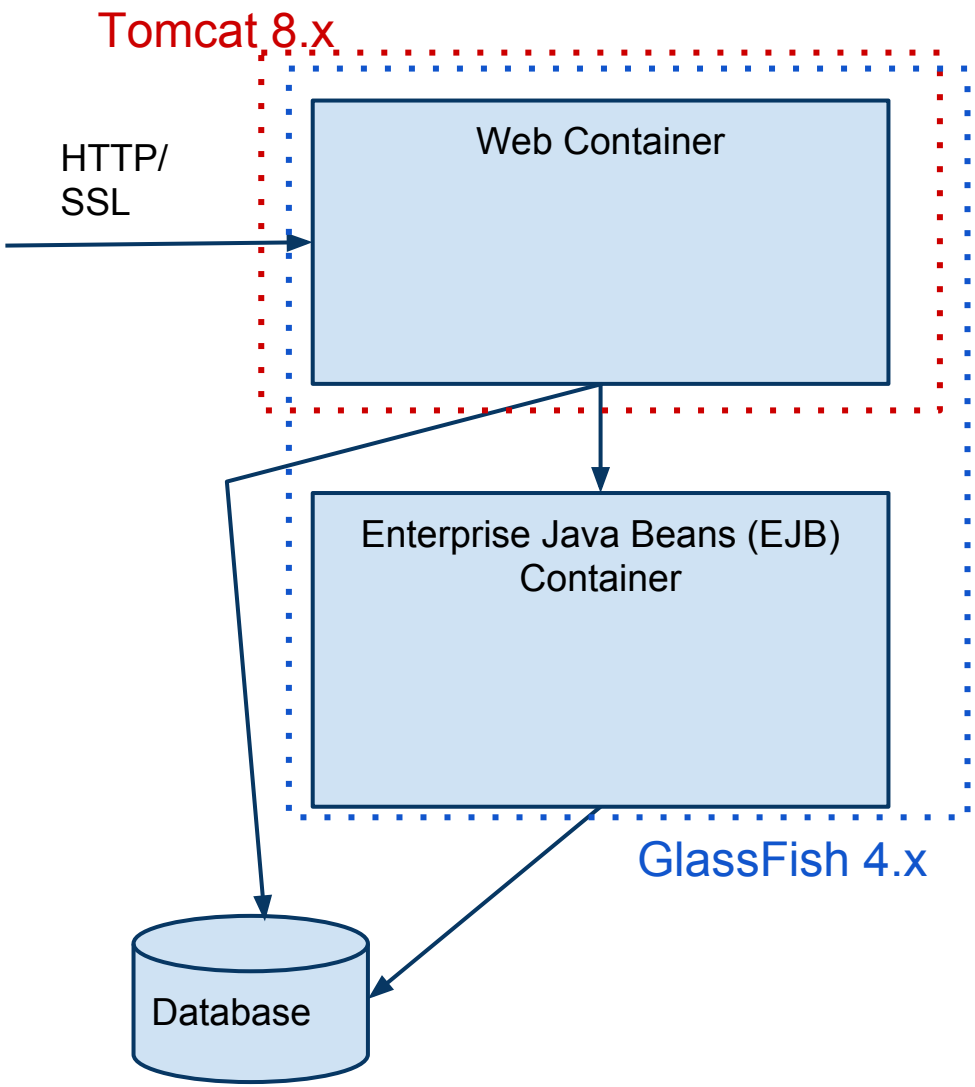
- An architecture with containers (aka servers, aka runtime environments) and components
- Containers offers services to our applications
- APIs to use the services
- Many implementations, low risk for vendor lock-in!
- But sometimes overwhelming … (name/version confusion)

JEE 7 Containers and Components



Container Implementations in Course

We'll use **Tomcat,** a Web container (aka Servlet container), and **GlassFish** a full application server (Web and EJB container)



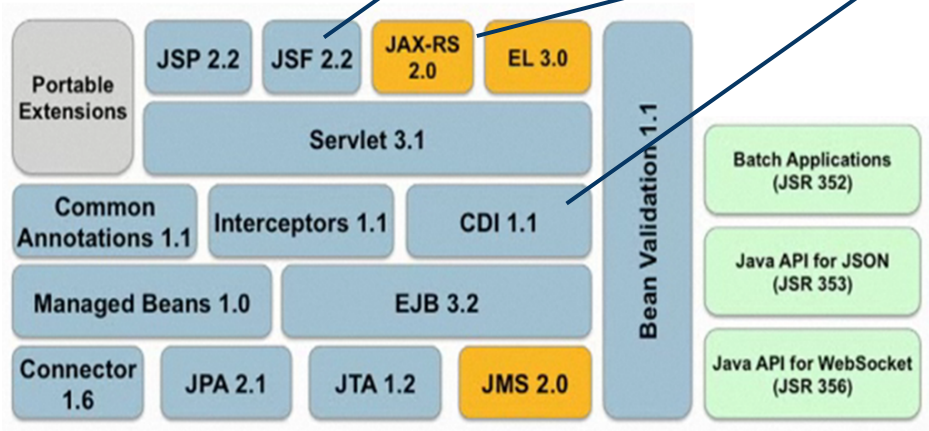
Java EE 7 Services and APIs

Services realized as a number of API's (to be used by our application).

Each API specified by a **Java Specification Requests, JSR's**

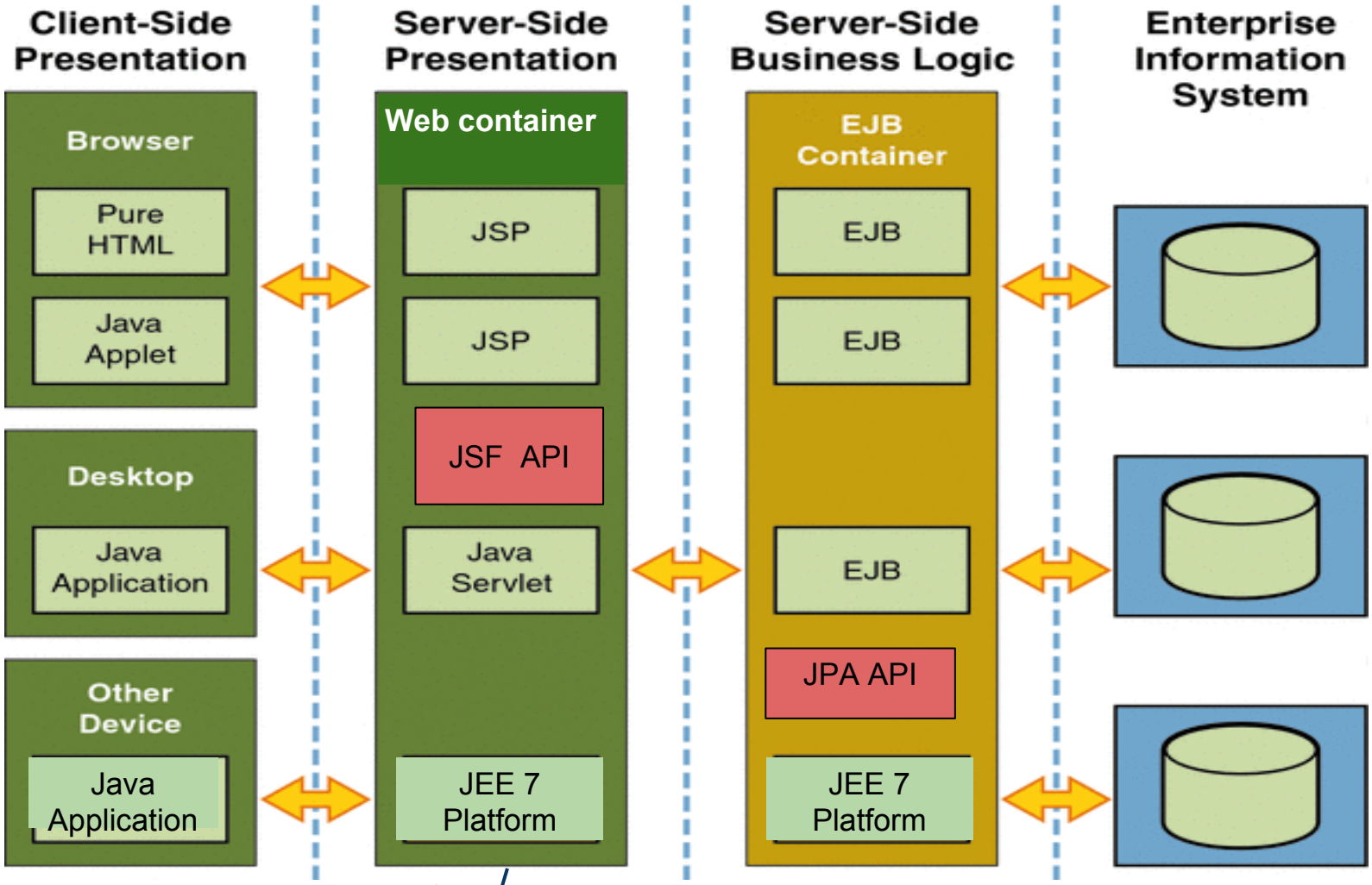
- JSR's are numbered and many have named reference implementations: **Metro, Mojarra, Jersey, Weld**, ...

Some APIs (JSRs)



Java SE APIs subset of JEE (but can't use everything, IO, Socket, Thread, ...)

JEE Multitier Architecture (where do API belong?)



Client machine

Each tier possible on separate machine(s) or in cloud

JEE Container Services

Container "runs" the application
- No `public static void main(...)`

Many objects handled by container
- Objects created, initiated, handled,..., destroyed by container
- Container handles input/output to application

Container uses dependency injection
- To wire together the application (connecting objects)

Concurrency, security, transactions, ... handled, a bit confusing
- Container managed or...
- ...application managed (application run in container but handle the issue, often used for customization, we avoid)

JEE Web application

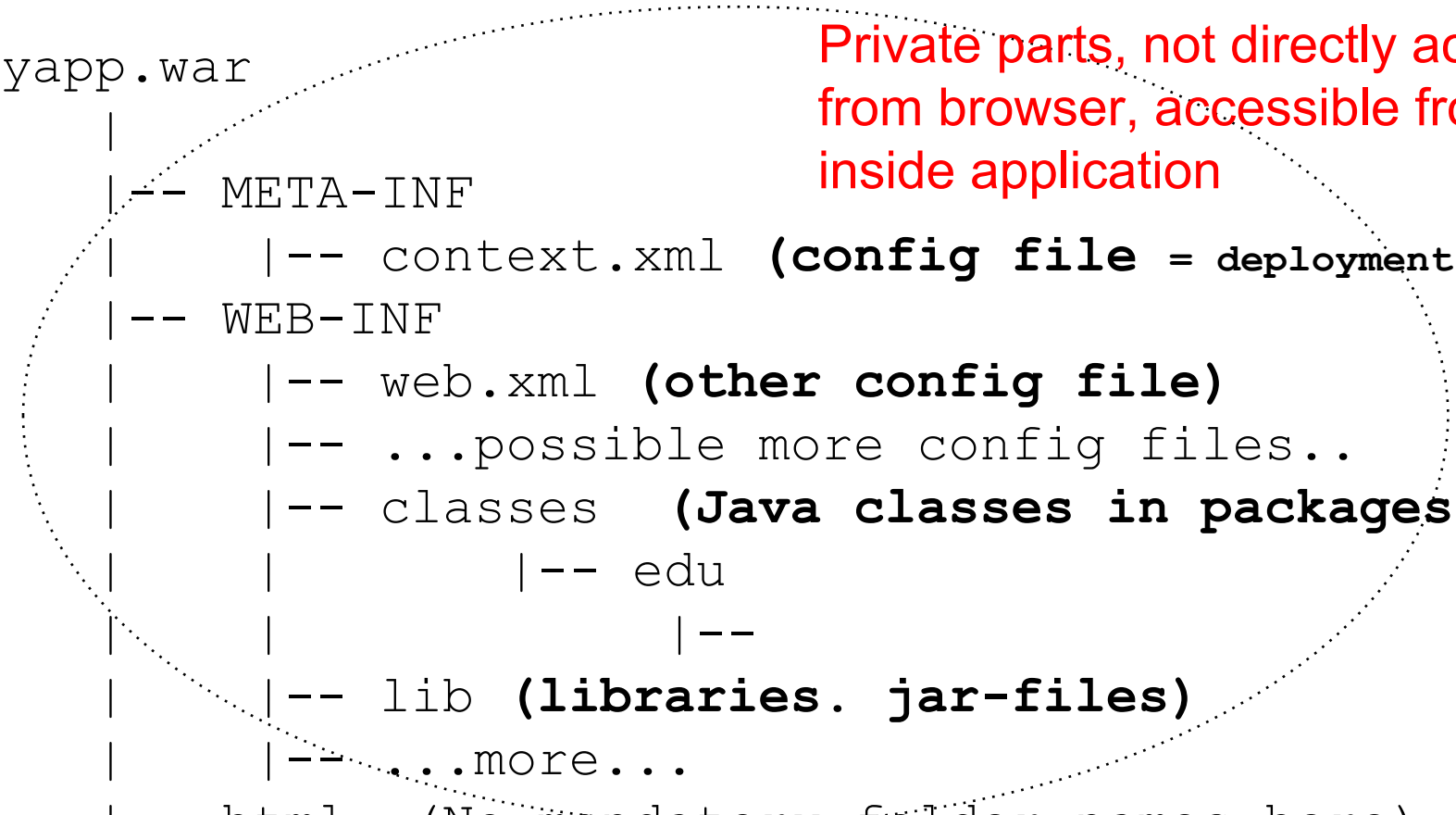
To make it possible for a container to run the application it must be packaged as a "Web-application"

- A *.**war-file**, a packed directory structure with some fixed directory names and a few configuration files (**deployment descriptors**)

Not the same structure as Maven!

- NetBeans will transform the Maven structure to a Web application structure during build
- Possible to inspect war-file content (Files tab). Useful during troubleshooting

War-file Structure



Private parts, not directly accessible from browser, accessible from inside application

```
myapp.war
     |
     |-- META-INF
     |     |-- context.xml (config file = deployment descriptor)
     |-- WEB-INF
     |     |-- web.xml (other config file)
     |     |-- ...possible more config files..
     |     |-- classes  (Java classes in packages)
     |     |        |-- edu
     |     |             |--
     |     |-- lib (libraries. jar-files)
     |     |-- ...more...
     |-- html  (No mandatory folder names here)
     |-- resources
           |-- css
           |-- img
```

Deployment of Web Applications

Deployment: Install application (war-file) on server (no need in course, NetBeans runs "in place")

New important phase (besides coding, compiling)

Application verified during deployment
- Deployment descriptors
- war-structure
- ... much more...

If application erroneous, will not be installed (i.e. not run)
- Successful compilation doesn't guarantee successful deployment
- Watch output in NetBeans for deployment errors

Configuring Web Applications

JEE has adopted "convention over configuration" (aka configuration by exception)
- Famous example: Ruby on Rails framework
- Principle: Configuration should have reasonable default values

Many (most) things will work out of the box
- Sometimes a bit confusing, ... when does it not?

If not satisfied modify
- "Everything" in JEE is configurable/customizable
- Configuration uses XML files or annotation on classes

Client Side Platform and Frameworks

Platform: Some "Browser"

- Java client side deprecated
- Client side dominated by JavaScript

Framework: Angular JS

- Will give structure to the client side (and more)
- New this year: Will it bear it's weight? Thrilling…!

What's defining the platform (what's a browser)?

The Browser Standards

A Browser is a piece of software implementing standards (recommendations) defined by the World Wide Web consortium (w3c) and others

- Main entry: W3C Standards

Some standards a browser should implement

- HTTP 1.1, RFC 2616 (HTTP 2.0 round the corner..?)
- Hypertext Markup Language, HTML 4.01
- HTML 5.x draft
- The Document Object Model, DOM
- Cascading Style Sheets, CSS
- ECMA Script ed. 5.1 (JavaScript is a dialect of)

Got to get some understanding of this …

Programmer view of Browser

An in memory DOM tree
- Tree described by an HTML page

A layout engine
- Render the DOM-tree using CSS rules

A JavaScript engine
- Programmatically manipulating the DOM tree (and more, … in particular handling events)

Browsers and Engines

Browser	**Layout engine (CSS)**	**JavaScript engine**
Chrome v28/Chromium	Blink (WebKit on iOS)	V8
Internet Explorer	Trident	Chakra (> IE 9)
Firefox	Gecko	Spider Monkey/Trace Monkey/Jäger Monkey
Safari	WebKit	Nitro

Browsers Compliance

Browser should follow the standards
- ...but not all do (getting better, ... worst ever IE 6)
- No guarantees that everything works in all browsers (especially smartphone browsers)

Code samples tested on Chrome (and to some extent Firefox)

Tools Overview