

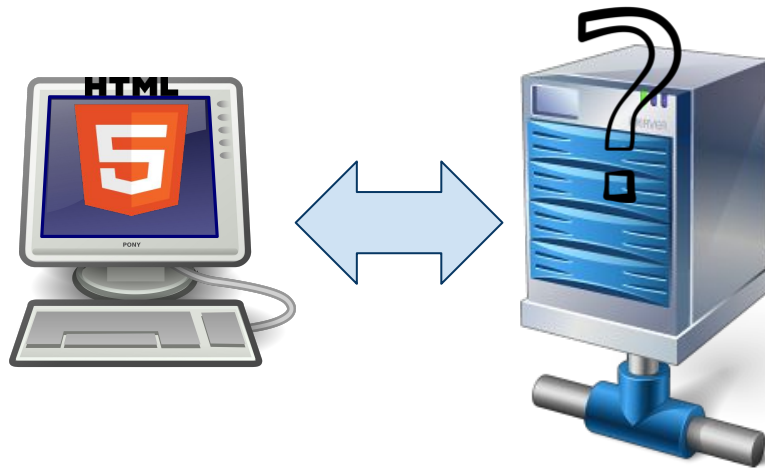
Server Side Intro

BWA Slides #2

Content

- Client and Server Side
- Platforms
- Frameworks
- Middleware
- Java Enterprise Edition 7 (JEE7)
- Web application deployment
- Web application Structure

Client And Server Side



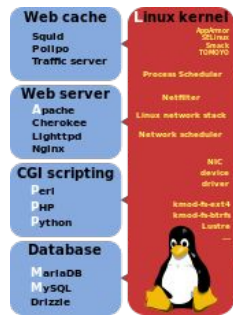
Have had a quick look on basic client side and client side programming.

- Many things to learn, ... but we know what to learn!
- HTML5 rules on client side!

But a web application has (at least) two parts: Client side and Server side

- Server side at least a [Web Server](#) but ...
- ... server side much more troublesome...
- ... many, many, options

Platforms



Not feasible to write a full (client and server side) web application from scratch!

- I.e. writing the server
 - Networking, concurrency, security, transactions, etc, ...
 - Platform (hardware, OS, runtime libraries)
 - Framework(s) more to come ...
 - Middleware, external APIs, more to come ...
 - Tools (development, testing, ...)

Need to decide on server side platform!

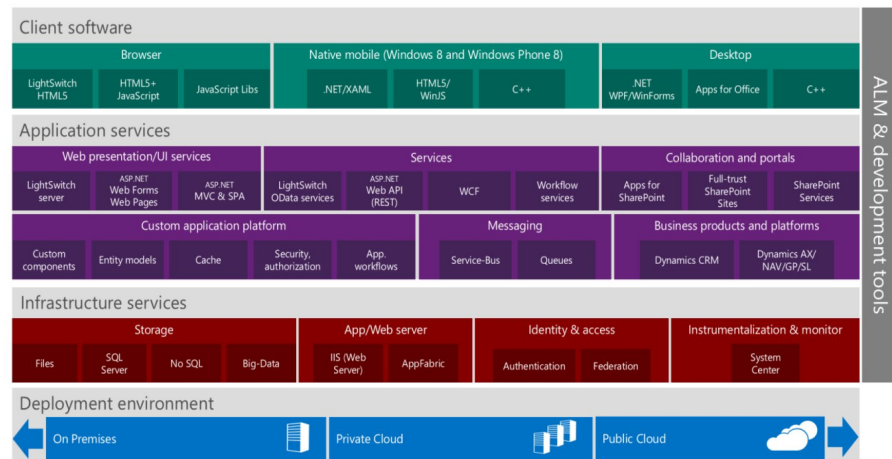
Platform not a well defined term. I'll would say

- A platform is a huge "ecosystem" for building/running software.
- It should be possible to develop and run the entire application inside the platform (possibly connecting to other systems ([legacy systems](#)))
- Aka "software stack"

Well known platforms

- .NET, Windows/Windows Phone, SQLServer, IIS, VisualS Studio, ASP.NET 4-5, VB.NET/C#...
- Java EE, Linux/Android/other..., [Application Servers](#), "any" database, Java, Eclipse/NetBeans/IntelliJ...
- LAMP: Linux/Apache/MySQL/PHP, PHPStorm/Sublime
- MEAN: MongoDB/Express/Node/AngularJS, JavaScript, ... Komodo/WebStorm?

.NET Platform

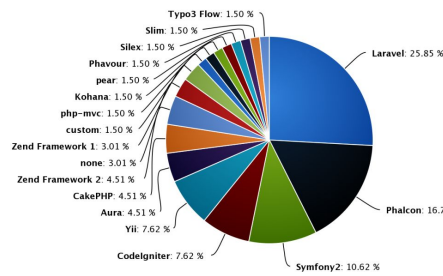


An example

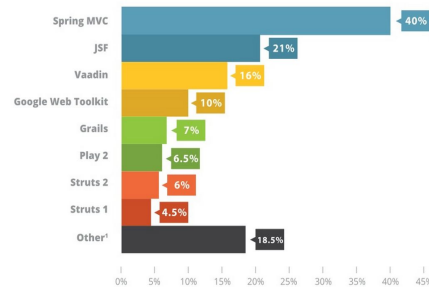
- MS normally has one (or a few, possible third party) products for each purpose
 - JEE and others normally have (very) many implementations for a purpose
- MS are products
 - Good: Selection done
 - Bad: Possibly vendor lock in
- JEE (and possibly others) are specifications or open source
 - Good: No vendor lock in
 - Bad: Which to choose?

Frameworks

Framework popularity, end of 2013; SitePoint



Web frameworks in use *



REBEL LABS

* Multiple selections were possible and the results were normalized to exclude non-users
† including Wicket, Seam, Tapestry, Play 1, ZK framework, Vaaptor and about 40 others



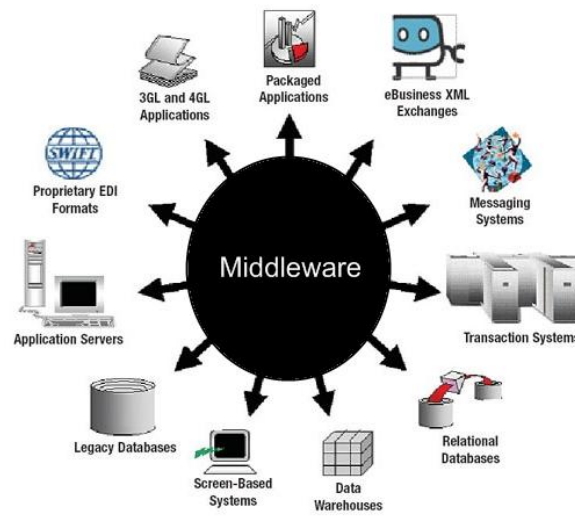
After deciding platform we need to decide on [framework](#) (if any)

- Framework will give application structure (MVC or others), inter-application services
- Framework will react, will be active in some way
- Most platforms have many choices of framework(s)
- This is also hard, can't learn (or even test) them all

In course we normally can't use frameworks because of time constraints

- Time to learn
- Will try AngularJS on client side, more to come ...
- Will use components (I don't count components as frameworks)
 - Components/APIs are passive

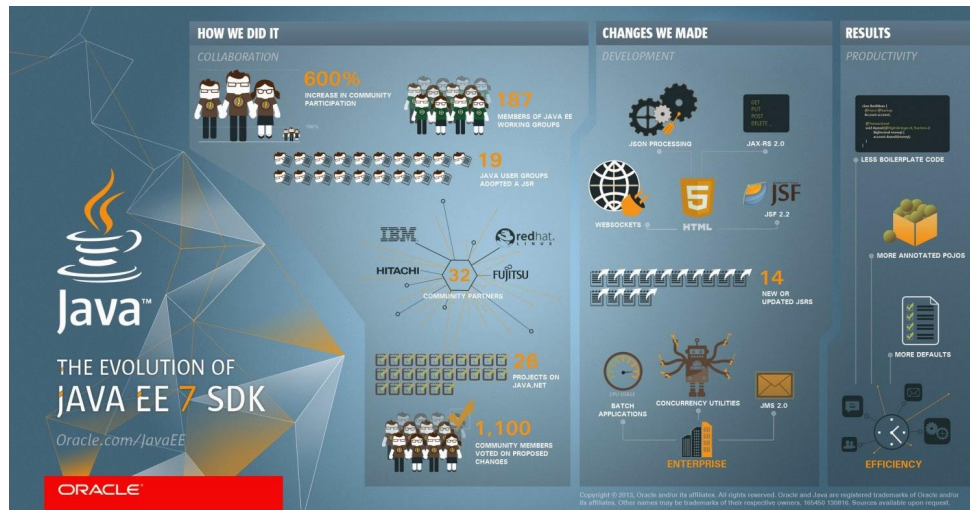
Middleware



Middleware

- "Software glue"
- Examples
 - Software to connect application and database (database drivers)
 - Software for reliable messaging between physically distributed parts of application (messaging)
 - Software to connect to external services, banking, login, ...
 - ...

Course Platform?



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 (i.e. no cost)
- Should be open ended (not quick ad hoc solution for specific problem)
- Should support sound programming concepts

Pedagogical issues

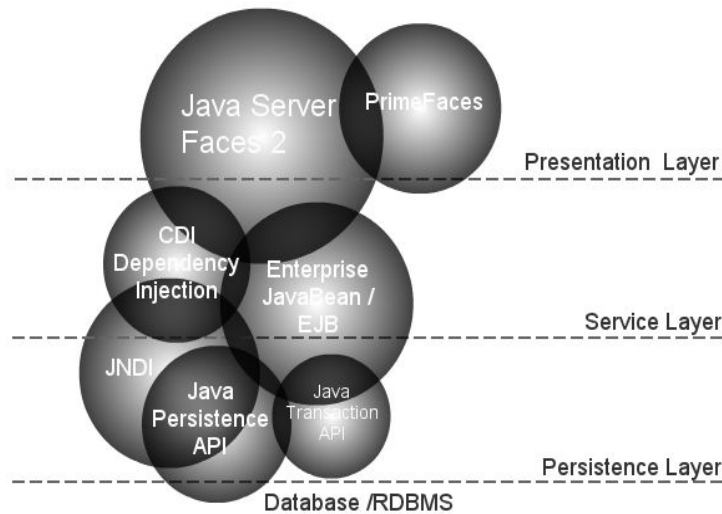
- Learning curve vs other ... (time, quality, ...)

No perfect solution but it ends up to : Java Enterprise Edition

Also: By looking closer at a specific platform you will get experience to judge others

- As said: Will get a reference point
- The concepts and problems are the same

Java Enterprise Edition 7



Java Enterprise Edition 7 (JEE 7)

Platform for developing and running web/enterprise software, including network and web services, and other large-scale, multi-tiered, scalable, network applications.

- Owned by Oracle but open source
- Platform independent
- HUGE! A few pieces shown in picture
 - If needed it's normally there, possibly hard to find

Defined by numbered Java Specification Requests, [JSRs](#), [JEE7 is an umbrella specification](#) including a specific subset of JSRs.

- JSRs have reference implementations with names (often starting as open source projects), Tomcat, WELD, Mojarra, ... sometimes confusing
- Many JSRs implemented by many vendors (many products)
- Also defines runtime environments for applications, JEE containers

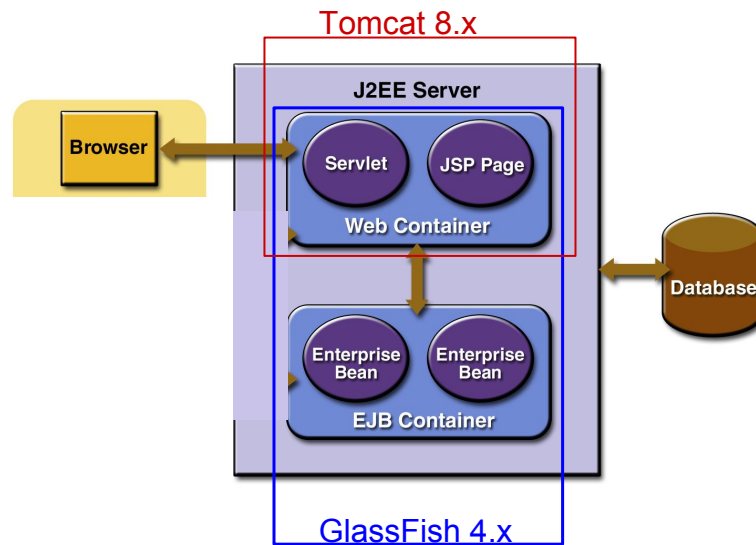
Platform is evolving. Things change ...

- Be aware of old examples on web
- Documentations is union of techniques ... must check for latest

JEE Characteristics

- Convention over configuration
- Application or container? Possible to customize, next slide ...
- Declarative (XML) vs. programmatic (Annotations @..)

JEE 7 Containers



JEE Containers are server software (normally runs on dedicated server machines)

- We run the container locally
- Container "run" our application
 - No public static void main(...)
 - Our application is an extension to the server
 - Any container may run multiple applications.
- Container handles object lifecycle
 - Objects created, initiated, in service, destroyed
- Container handles input/output to application
- Container handles concurrency, security, transactions, ...
- ... and much much more ("any" service needed by application)!
- NO need for us to code the infrastructure!

A JEE Web application (more to come) must run in a Web Container

- We use Tomcat or GlassFish servers bundled with NetBeans

A JEE Enterprise application must run in an EJB Container

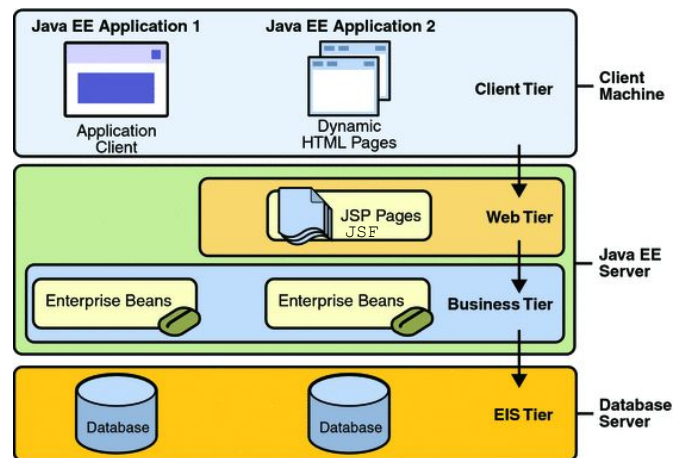
- We use GlassFish (i.e. an application server)

Our application uses standard JEE classes (components) like Servlets, JSP Pages, Enterprise Beans etc. to benefit from the container services

- ... and also ordinary Java classes and own classes, plain old Java classes POJO's
- Java Standard Edition (JSE) is a subset of JEE
- Approaches
 - "Container managed": Works like above, we try to use this all the time

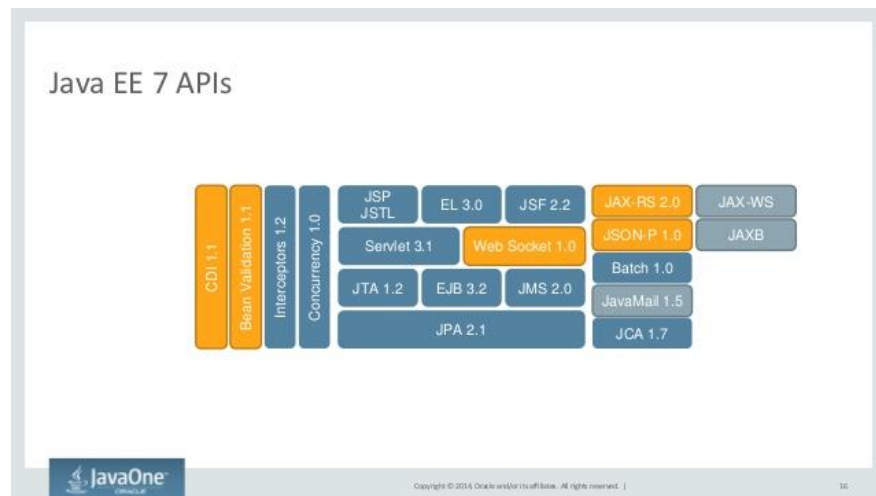
- "Application managed": If custom needs possible to let application (not container) handle the issue (avoid)

Layers, Tiers and Components



The components are designed for different (abstraction) layers and often distributed to different tiers

Some JEE APIs

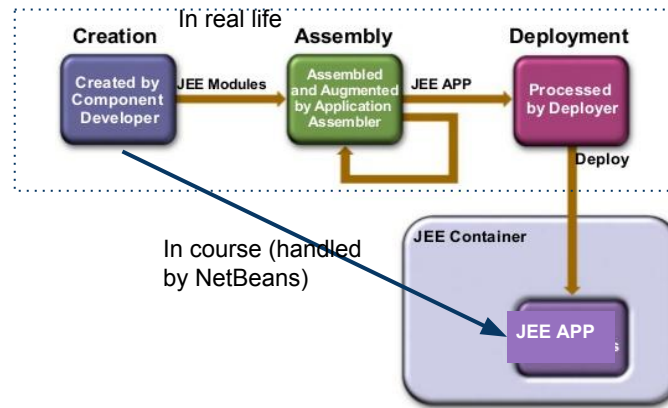


We'll meet a few ... more later

- JSP, EL and JSTL, view parts
- JAX-RS, REST API service backend
- JPA, object relational mapping API, database middleware
- CDI, dependency injection, object handling
- ... (there are many more)

Application Deployment

The JEE Life Cycle



Deployment:

New important phase (besides coding, compiling, testing)

- JEE application verified during deployment (deployment descriptors, war structure, upcoming ... , and much more...)
- Successful compilation doesn't guarantee successful deployment
- If application erroneous, will not be installed (i.e. not run)
- NetBeans will deploy "in place" , as noted we run locally
- Watch output in NetBeans for deployment errors

JEE Web Application Structure

```
myapp.war
|
|-- META-INF
|   |-- context.xml (deployment descriptor=dd)
|-- WEB-INF
|   |-- web.xml (other dd)
|   |-- ...possibly more config files..
|   |-- classes (class files in packages)
|       |-- edu
|       |--
|   |-- lib (libraries)
|   |-- private resources (pages etc.)
|-- public resources (pages etc.)
```

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

- A *.war-file = a packed (zip) directory structure of compiled code (and resources, libs) with some fixed directory names and a couple of XML configuration files
 - WEB-INF directory only accessible from within application (not via browser address field or similar)
 - web.xml and context.xml are deployment descriptors
 - If structure wrong or missing descriptor(s) -> possibly deployment error
- Structure not same as development structure (Maven project)
- Maven will create war-file during build (using archetype for JEE web application)
- Possible to inspect war-file content in NetBeans (Files tab). Useful during troubleshooting