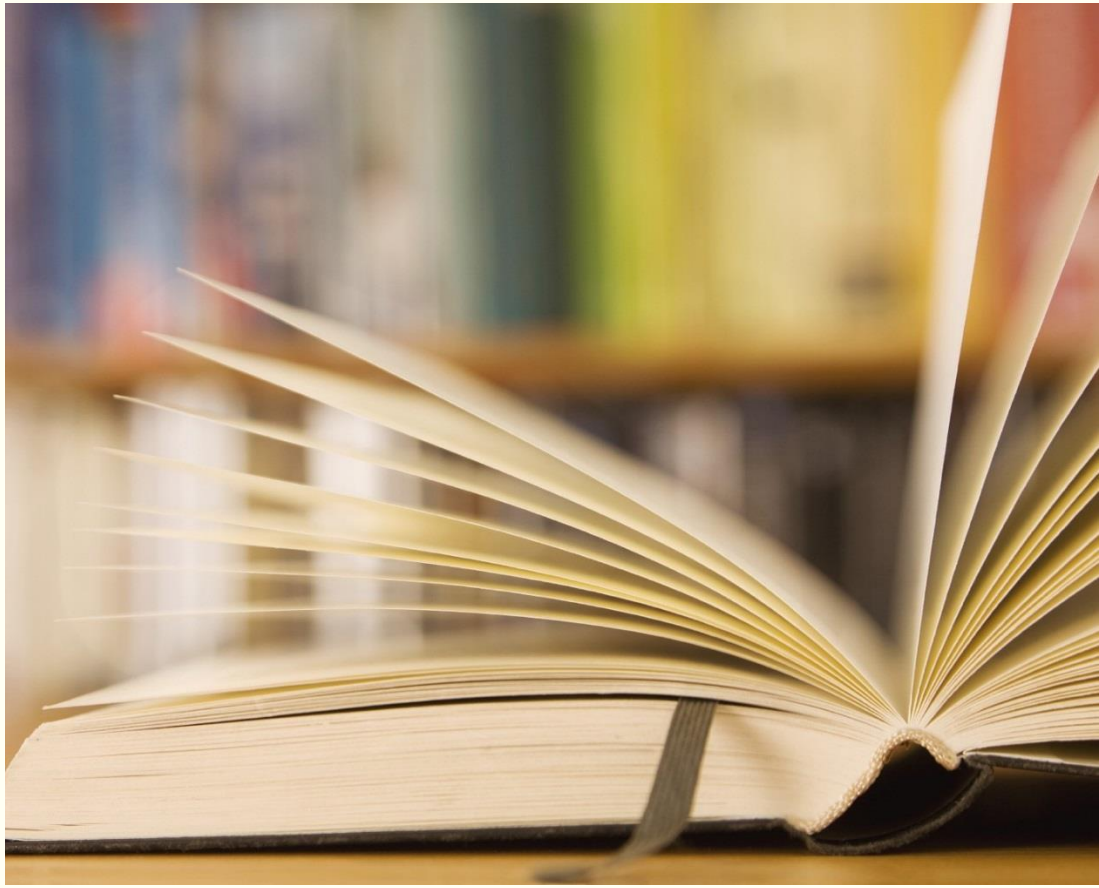
ENTERPRISE ARCHITECTURE

Najeeb Najeeb, PhD



INTRODUCTION TO ENTERPRISE ARCHITECTURE

Computer Software

- Proprietary Software (non-free software, closed-source software)
  - Source code is someone's intellectual property (copyrighted, sometimes patented).
  - -Vendor lock-in
  - -Limited to certain hardware configurations
  - -Cost$$
  - +Support
- Open-Source software
  - Source code released under a license.
  - +Less bugs
  - +More secure
  - +Cheaper
  - -Too slow support
  - -Updates not coming out as fast
  - -May die
  - -Switching from one open source to another could be $$$

Specification

- A useful document that provides guarantees to a user while providing some level of freedom to the implementor.
- Good specifications are precise in the points that matter to the user, but vague in the points that matter to the implementor.
- Java is a Specification
  - HashCode method
    - "the hash value of the Class object is dependent on its identity."
- There could exists several implementations for the same specification (why?)
  - JVM
    - OpenJDK
    - OpenJ9
    - Zing (https://www.azulsystems.com/products/zing/whatisit/)
    - Apache Harmony (2010)
  - JDBC
- Specifications are written using a specialized specification-writing language.

Specification VS Method Contract

- int add(float x, float y);
  - Good method contract.
  - Not a great specification.
- Interfaces define method contracts.
- Specifications define method behavior(s).
- What Interfaces are to Classes is what Specifications are to Frameworks.
- Specifications are documentations of guarantees that libraries and frameworks provide for solving certain problems in Java.
- Types of Java problems
  - Standard
  - Enterprise

Benefits of Specifications

- Prescribe solutions to common faced challenges.
- Abstract
  - Interfaces, contracts, public to all developers.
- Standardized
  - Well-defined criteria set by experts
  - Gone through Java Community Process (JCP) Java Specification Request (JSR)
  - Industry-tried and industry-tested
- Switch implementations with no code change.
  - Performance.
    - https://terrazadearavaca.blogspot.com/2008/12/jpa-implementations-comparison.html
  - Features.
    - https://en.wikipedia.org/wiki/Java_Persistence_API

Java Specifications

- JSE
  - Java Platform, Standard Edition (J2SE)
    - Portable code for desktop and server environments.
    - OpenJDK is the reference implementation (RI).
- JME
  - Java Platform, Micro Edition (J2ME)
- JEE
  - Jakarta Enterprise Edition (Java EE, J2EE)
    - Specifications extending JSE for enterprise features (distributed computing and web services).
    - Java EE 8, RI is GlassFish.
    - Jakarta EE 9, RI is Open Liberty (https://openliberty.io/)

What is JEE

- Java EE is a JSR.
  - A collection of Specifications to address enterprise application needs.
  - Umbrella JSR (Java EE 7 – JSR-342, Java EE 8 – JSR 366).
  - RI: GlassFish Application Server.
  - Implementations of the specification is called Application Server
- Application Server
  - A concrete implementation of the JEE (or umbrella JSR).
  - Abstract the developer from mundane issues like
    - Data-source pooling, caching, clustering … etc.
  - Passes a TCK (for umbrella JSR TCK)
- Programming to JEE
  - Swap application server vendors (satisfying same JSR).

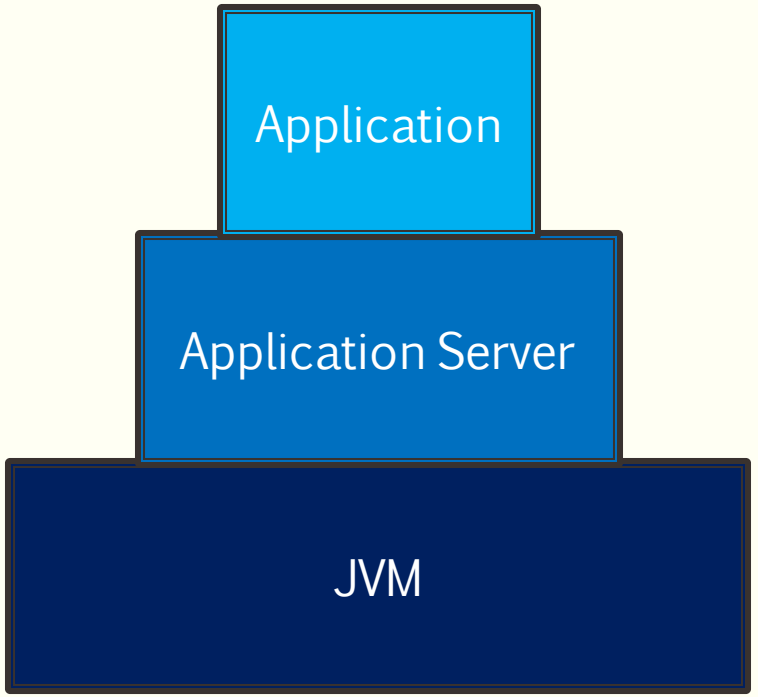
Main Point

- Open-Source projects are a good solution to some problems, but they come with some limitations. Specifications provide a standard, well accepted, and a proven solution with guarantees to the user and flexibility to the developers.
- Trying to solve all problems in life is hard. The highest first principle is to focus on what is important and everything else will fall in place.

SIMPLE ENTITIES

How to Run a Specification?

- Java EE is a Specification, it is abstract.
- Applications Server: Implementation of Java EE Specifications
- Application Servers
  - JBoss WildFly
  - Apache TomEE
  - Payara Server
  - GlassFish (RI)
  - OpenLiberty (RI)



Problems with Specification

- Trying to do too much
  - Too slow.
  - RI becomes a project by itself instead of a proof of concept.
- Having to convince the specification owner of the industry needs (Oracle)
- What if the specification is lagging in the market
- Who is the Specification owner
  - Big companies (the Apple, Samsung, Google analogy does not apply to SW).
  - The owners should be the developers.
- Maybe Jakarta EE can come to the rescue? Or is it too late? Is it better to play side by side?

Why learn Spring

- Spring
  - Develop a product (uniform solution).
  - Solve only what you need to solve, use other tools for the rest.
  - Once the Container, JVM, or tools catchup then switch.
- What is going on between Spring and Jakarta EE
  - Pivotal Software merged with VmWare joined Eclipse Foundation.
  - Microsoft joined Eclipse Foundation.
- Spring is the industry standard
  - Java EE implementing ideas done in Spring.
- IBM & Oracle sometimes overdesign (backwards compatibility)
  - Many needs in industry are less demanding (but still complex).
  - Jakarta EE is trying to convert JEE to a more developer focused and less backwards compatible
- Software is better done by a software business
  - Did not go well for Sun, IBM, and Oracle.
  - https://www.eclipse.org/membership/exploreMembership.php#allmembers
- Spring 5.3.x (Oct 2020) JDK 8,11,17
  - https://github.com/spring-projects/spring-framework/wiki/Spring-Framework-Versions