J2EE

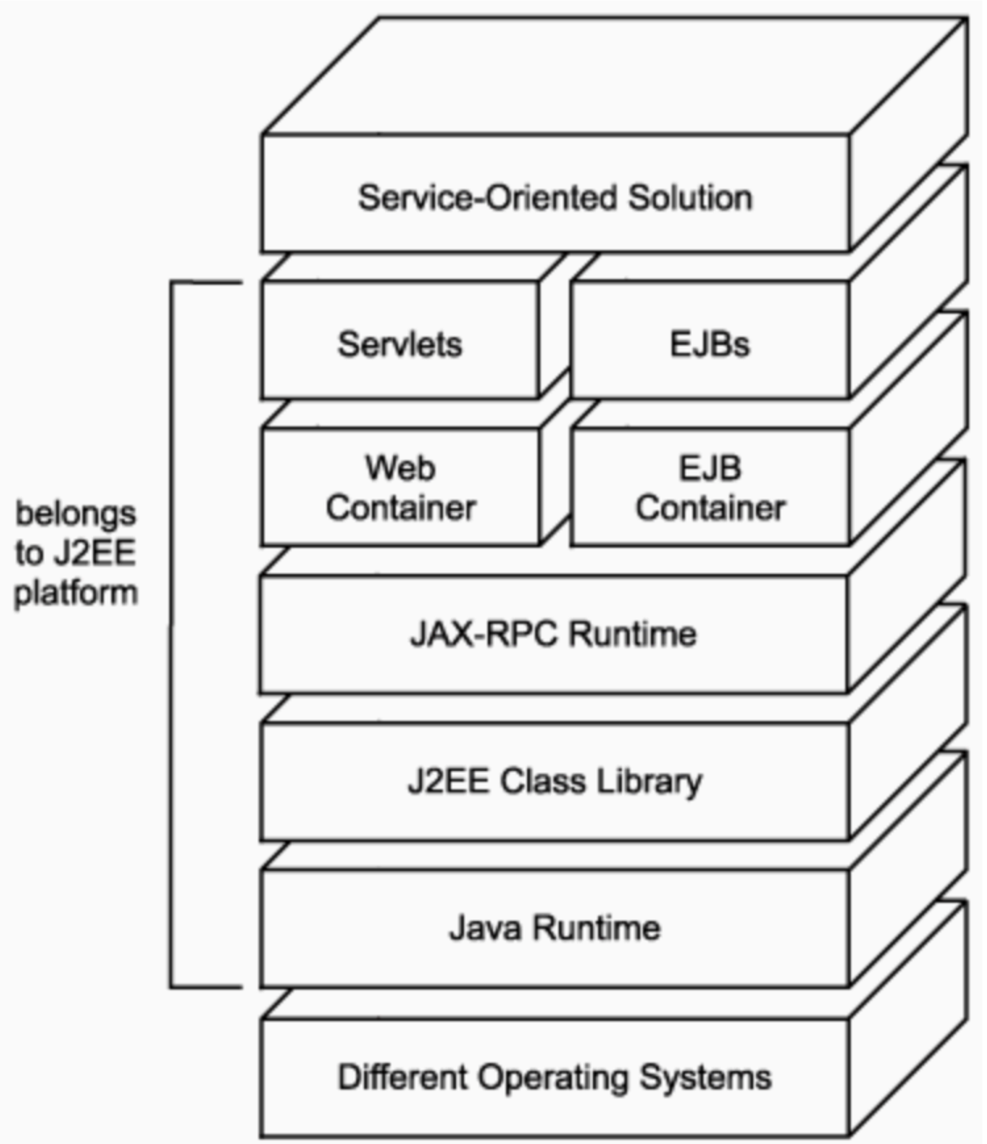
Chelsi Kothari
2021UCS1635

- Java 2 Platform, Enterprise Edition (J2EE) is widely used to develop enterprise-level applications with support for Service-Oriented Architecture (SOA). It provides a set of standardized components and APIs that enable the creation of distributed, interoperable, and reusable services.

- **Web Services (SOAP and REST)**
- JAX-RPC (Java API for XML-Based RPC): JAX-RPC was initially used in J2EE for building SOAP-based web services, which allowed Java applications to invoke remote services over HTTP using XML.

- **Java Message Service (JMS)**
- JMS provides a messaging standard that supports asynchronous communication between services, which is essential in an SOA for decoupling components. JMS can be used to send and receive messages between distributed components, enhancing the loose coupling characteristic of SOA.

- The Servlets + EJBs and Web + EJB Container layers (as well as the JAX-RPC Runtime) relate to the Web and Component Technology layers established earlier in the SOA platform basics section. They do not map cleanly to these layers because to what extent component and Web technology is incorporated is largely dependent on how a vendor chooses to implement this part of a J2EE architecture.
- The components shown in adjcanet figure inter-relate with other parts of the overall J2EE environment to provide a platform capable of realizing SOA.



J2EE Standards Relevant to SOA

- Java 2 Platform Enterprise Edition Specification: Sets foundational standards for distributed J2EE architecture, requiring compliance for vendors to claim J2EE compatibility.
- Java API for XML-based RPC (JAX-RPC): Defines the JAX-RPC environment, core APIs, and Service Endpoint Model, essential for J2EE Web services.
- Web Services for J2EE: Specifies the J2EE service architecture, detailing what can be developer-built, vendor-specific, or must follow J2EE standards. It introduces the Port Component Model for service providers.

Runtime Environments

- Java Runtime: Foundation for core Java processing in J2EE.
- JAX-RPC Runtime: Supports SOAP and WSDL processing, enabling Web services.
- Component Containers:
- EJB Container: Hosts EJBs, providing services like transaction management, security, and concurrency.

J2EE Architecture Components

- Java Server Pages (JSPs): Text-based files with HTML and Java code for dynamic web pages.
- Struts: An extension for developing complex UIs and navigation in J2EE Web applications.
- Java Servlets: Compiled programs that handle HTTP requests/responses, hosted on the web server.
- Enterprise JavaBeans (EJBs): Core business components deployed on application servers, supporting features like transaction management and typically used for web services.

APIs for SOA in J2EE

- Java API for XML Processing (JAXP): Processes XML documents, supports DOM and SAX models, and enables XML transformation with XSLT. Key packages include:
- javax.xml.parsers: Classes for different XML parsers.
- org.w3c.dom & org.xml.sax: DOM and SAX document model standards.
- javax.xml.transform: Classes for XSLT transformation.

In J2EE, web services can be implemented using either Servlet-based or EJB-based configurations, each suitable for different needs:

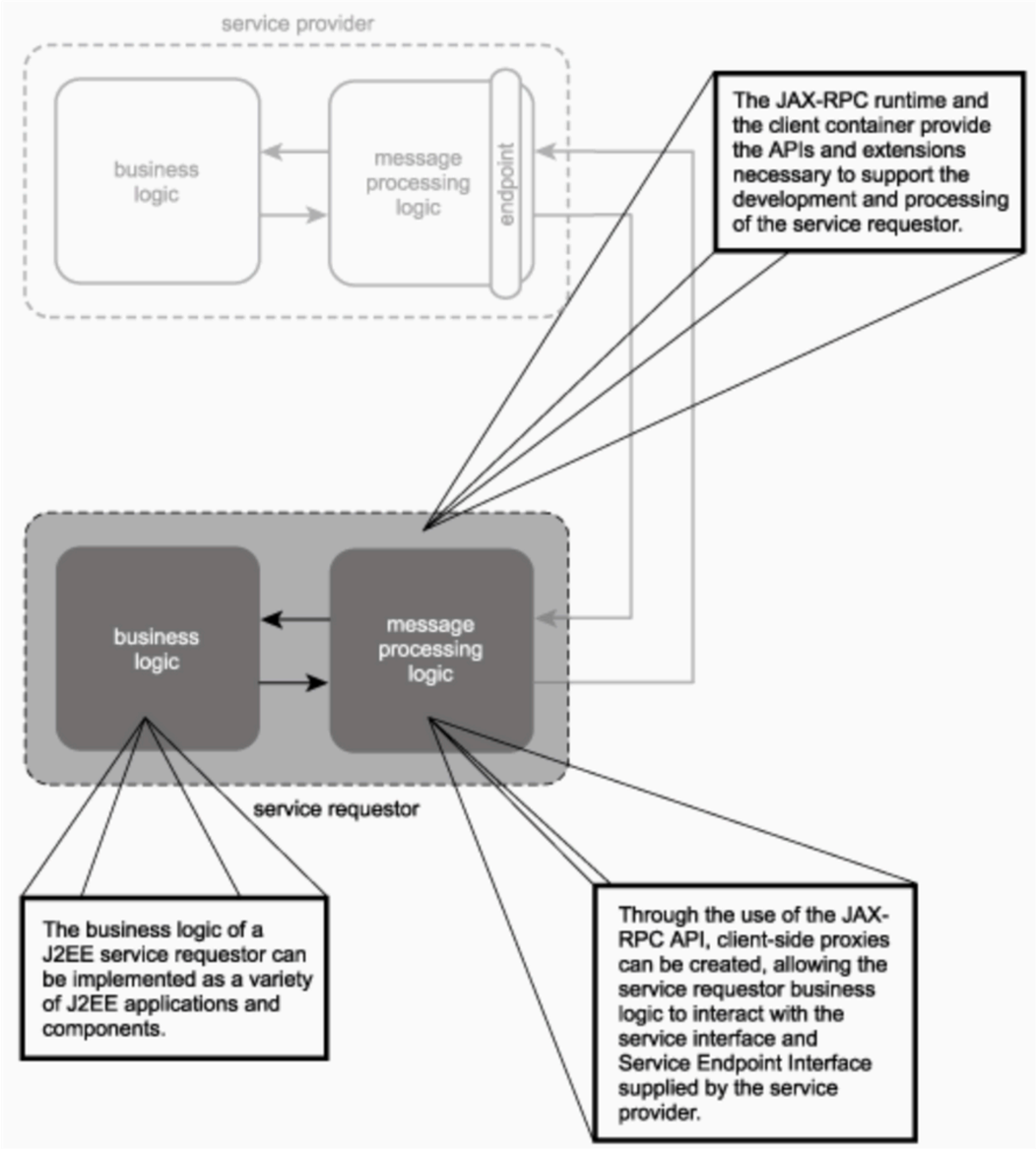
JAX-RPC Service Endpoint (Servlet-based):

- Typically implemented as a servlet and hosted within a Web container.
- Commonly used for lightweight services that do not require advanced EJB container features.
- Ideal for straightforward web services, as the servlet-based model provides a simple, efficient deployment for services primarily focused on handling HTTP requests.

EJB Service Endpoint (EJB-based):

- Exposes an Enterprise JavaBean (EJB) as a web service, requiring a specific type of EJB: the Stateless Session Bean.
- This approach is well-suited for services that need to encapsulate complex business logic, such as legacy systems, or when enterprise features (like transaction management, security, and resource pooling) are needed.
- Hosted within an EJB container, which provides robust runtime capabilities, making it ideal for services that rely on advanced infrastructure.

I



- This diagram illustrates the architecture of a Java API for XML-based Remote Procedure Call (JAX-RPC) setup, showing how a service provider and service requestor communicate in a J2EE (Java 2 Platform, Enterprise Edition) environment.
- Service Provider: Contains business logic (core functionality) and message processing logic. It communicates with the service requestor through an endpoint.
- Service Requestor: Also has business logic and message processing logic. The requestor's business logic can be implemented with various J2EE applications or components.
- JAX-RPC Runtime: Provides APIs and client-side proxies that facilitate the interaction between the service requestor's business logic and the service provider's endpoint. This allows the requestor to interact with the service provider as if it were a local object, abstracting the complexities of remote communication.