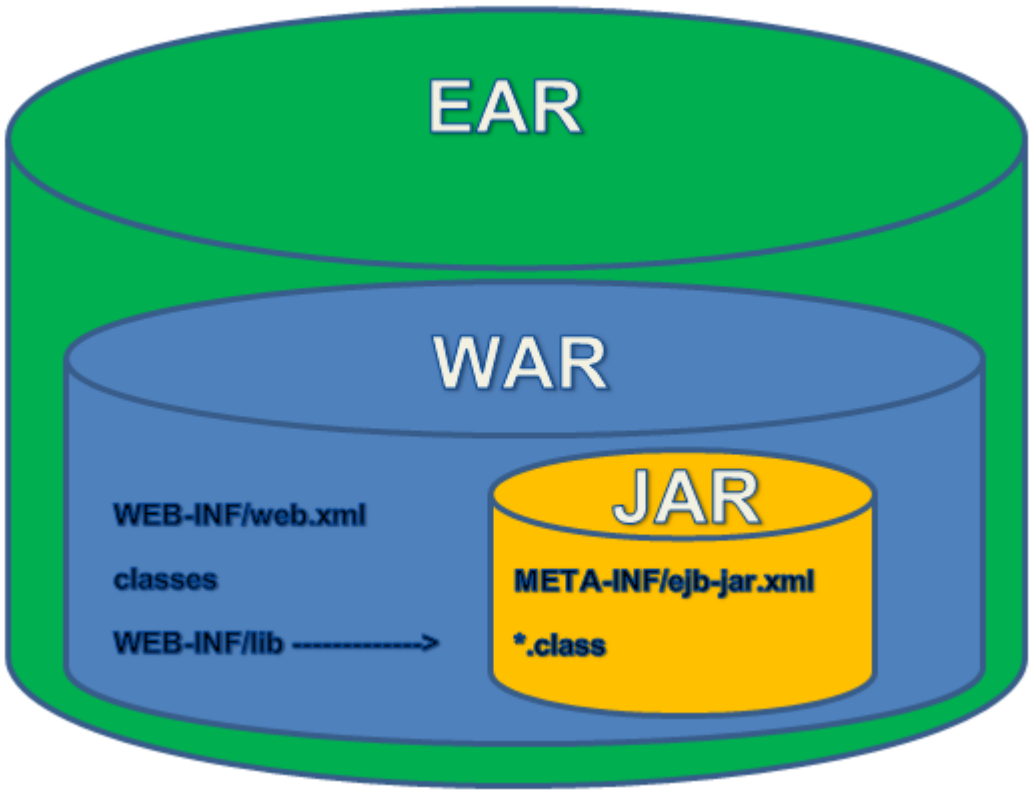
EAR
WAR
WEB-INF/web.xml
classes
WEB-INF/lib ---------->
JAR
META-INF/ejb-jar.xml
*.class

Jar, Fat Jar, War, Ear

	Jar	**fat jar**	**war**	**ear**
Use case	For Standalone apps with a simple structure.	For standalone apps with bundled external dependencies.	For Java web apps to be deployed on servlet containers like Tomcat.	For enterprise apps with multiple modules.
Size	Compact as it only includes application-related files.	Larger as it includes all the dependencies.	depends on the web app content and dependencies.	depends on the complexity of the enterprise app.
Deployment	Deployed as a standalone app.	deployed as a standalone app.	Deployed on servlet containers like Tomcat.	Deployed on Java EE app servers as WildFly /JBoss.
Dependencies:	Requires external dependencies to be in the classpath.	Self-contained with all dependencies included.	Depend on servlet containers to manage dependencies.	Require Java EE app servers to Manages dependencies internally.

Isolation and Modularity:	Limited isolation; dependencies are resolved at runtime.	Encapsulates dependencies, but limited modularity.	Encapsulates web application components for better isolation.	Supports modularity with distinct modules for different functionalities.
Versioning	Versioning is handled at the file level.	Versioning can be more complex due to bundled dependencies.	Versioning is essential for web apps, specified in the deployment.	Versioning is crucial for managing the overall enterprise application.
Scalability	convenient for small to medium-sized apps.	convenient for large projects; need to consider the dependencies.	Scalable for web apps, especially when deployed on cloud platforms.	Provides scalability for EE apps with modular components.
Development Environment	Simple and easy for local development and testing.	Ensures consistent dependencies across all environments.	Mimics the production environment closely for accurate testing.	Requires a Java EE application server for comprehensive testing.