

# Table of Contents

Während der Design Phase von **JVx** wurde höchster Wert auf **Offenheit** gelegt. Wir haben ein **Komplettpaket** (Full Stack Framework) entwickelt das Out-of-the-Box alle Anforderungen von **Enterprise Applikationen** abdeckt und dennoch die Erweiterbarkeit nicht vernachlässigt.

Es wurden **keine neuen Standards** geschaffen, da die Auswahl an Standards bereits groß genug ist. Mit JVx beschränkt wir einen Weg mit **bekanntem Techniken** und **neuen Ideen**.

Aufbauend auf einer Multi Tier Architektur besteht JVx aus folgenden Teilen:



Die Tiers übernehmen folgende Aufgaben:

- **Client Tier**

ie Client Tier erlaubt sämtliche (Web, Mobile & Desktop) UI Technologien. Dazu zählen für uns unter anderem Java Swing, JavaFX, SWT und natürlich auch moderne Javascript Bibliotheken und Techniken wie Vaadin, jQuery, extJS, qooxdoo uvm.

Die Client Tier hat keinen direkten Zugriff auf Daten bzw. die Data Tier und kann ausschließlich mit der Enterprise Tier kommunizieren.

- **Enterprise Tier**

In der Enterprise Tier ist das Herzstück von JVx verankert. Ein Kommunikationsserver übernimmt die Anfragen von der Client Tier, verarbeitet diese und übermittelt das aufbereitete Ergebnis. Der Server selbst ist eine austausch- und erweiterbare Komponente und bedient sich weiterer Komponenten für die Session Verwaltung, den direkten Objektzugriff und die Zugriffssteuerung.

Für den Datenzugriff wird ein Persistence API bereitgestellt um einen Technologie unabhängigen Zugriff auf die Data Tier zu ermöglichen.

- **Data Tier**

Die Data Tier ist ein beliebiger Datenhalter. Dazu zählen relationale Datenbanksysteme (z.B. Oracle, DB2, MySQL) oder auch spezielle Dateiformate (z.B. XML, XLS).

Weitere Informationen finden Sie in der [JVx - Paketübersicht](#).

From:

<http://doc.sibvisions.com/> - **Documentation**

Permanent link:

[http://doc.sibvisions.com/de/jvx/system\\_architecture](http://doc.sibvisions.com/de/jvx/system_architecture)



Last update: **2018/01/29 13:59**