



JavaFX

im - sinnvollen! - Anwendungseinsatz

JAVAFORUM
stuttgart

Björn Müller

- 1991-2001 SAP
 - Anwendungsentwicklung
 - Architekturentwicklung (Java)
- 2001-2007 Casabac Technologies GmbH
 - AJAX
- Seit 2007 CaptainCasa Community
 - Swing für Unternehmensanwendungen
 - Seit 2012: JavaFX für Unternehmensanwendungen



CaptainCasa Community

- Verbindung von >50 Unternehmen, die eine gemeinsame Frontendarchitektur definieren...

...und nutzen!

- Mittelständische Softwarehäuser
- Beratungs-/Serviceunternehmen
- Industrieunternehmen mit eigener Softwareentwicklung

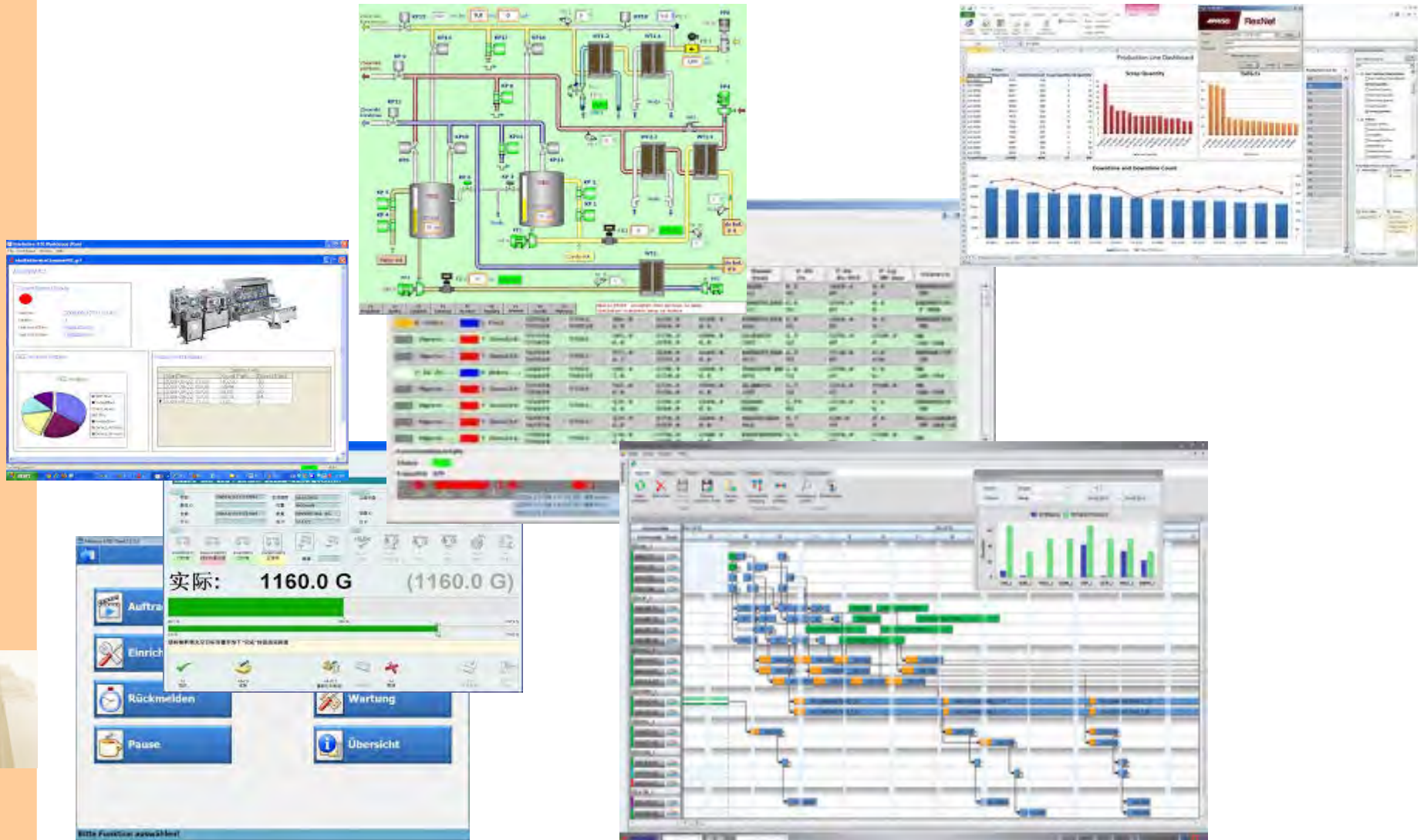


Wir stellen uns mal vor...

- ...wir würden heute mit der Entwicklung einer Anwendung starten:
 - Fertigungsauftragsverwaltung
 - Eigene Nutzer
 - „Werker“ (BDE, MDE)
 - „Sachbearbeiter“ (Disponenten, Schichtleiter)
 - „Planer“ (Logistik-Management)
 - Fremdnutzer
 - Auftragsstatusverfolgung

- Welche UI Technologie würden wir nehmen?

„Qualität“ der Dialoge



Technologie Optionen

- HTML/JavaScript
 - GWT, JQuery, Sencha, ...
 - Vaadin, *Faces, ...

- Java
 - JavaFX
 - (Swing, Eclipse, ...)

- (.Net, Qt)



JavaFX == Java Programmierung

- Sprache mit klarer, bekannter, einsichtiger Struktur
- Produktive, umfangreiche Entwicklungs- und Toolumgebung
- Stabile, performante Laufzeitumgebung mit bekannten Verhaltensmustern
- Java eignet sich sowohl zur UI Entwicklung als auch zur Entwicklung der Geschäftslogik.



JavaFX == Solide UI „Library“

- JavaFX Library umfasst...
 - Core Komponenten
 - Animationen, Transformationen
 - Styling
 - Property-Binding Konzepte
 - Programmierung von UIs und deskriptive Beschreibung von UIs (FXML)
- ...ist erweiterbar um eigene / fremde Komponenten
- ...ist in großen Teilen „gut strukturiert“ und einfach nutzbar

JavaFX == „Stand alone App“

- WebStart im Rahmen einer zentralen Java Installation
- (Applet)
- Self-containing Java Anwendung
 - Installation per „setup.exe“
 - Java als interner Bestandteil



JavaFX != Web Application UI

- „Applet“ ist keine strategische Option!
 - Läuft nur in (Desktop-) Browsern
 - Wenige mögen es..

- Keine Abhängigkeit vom Browser
 - Nur Abhängigkeit vom Betriebssystem
 - Kein „Zero-Install“



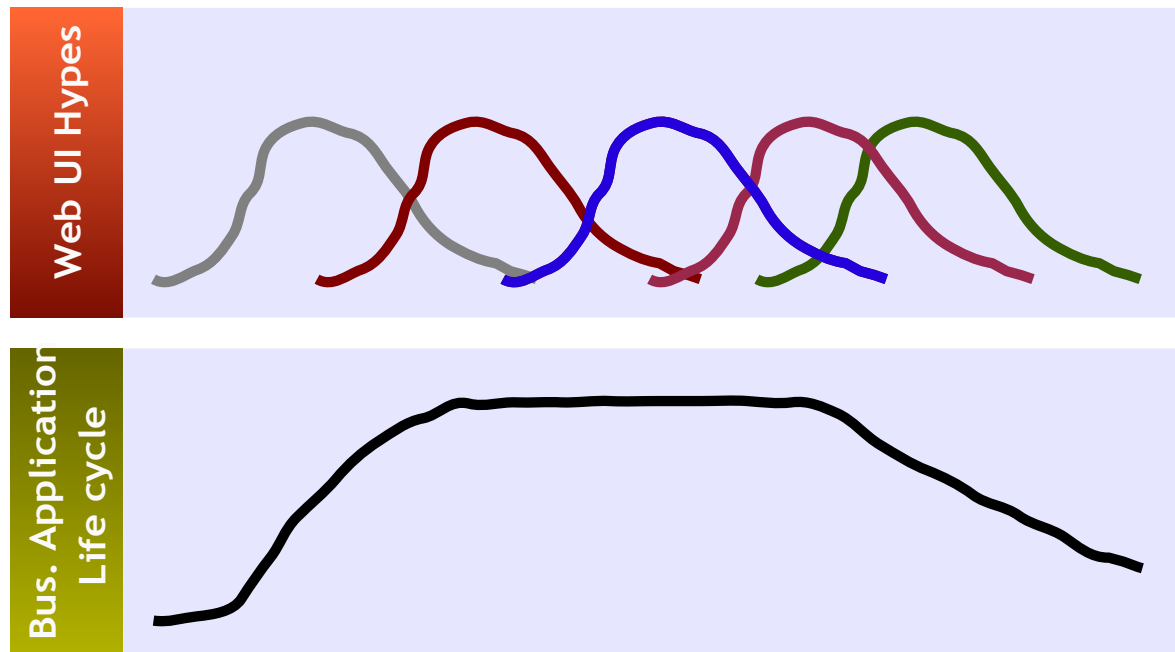
JavaFX != Web Application UI

- Android
 - Browser-basierte Frontends
 - Native Apps
- Desktop
 - Browser-basierte Frontends
 - JavaFX-Apps



JavaFX == Java Standard

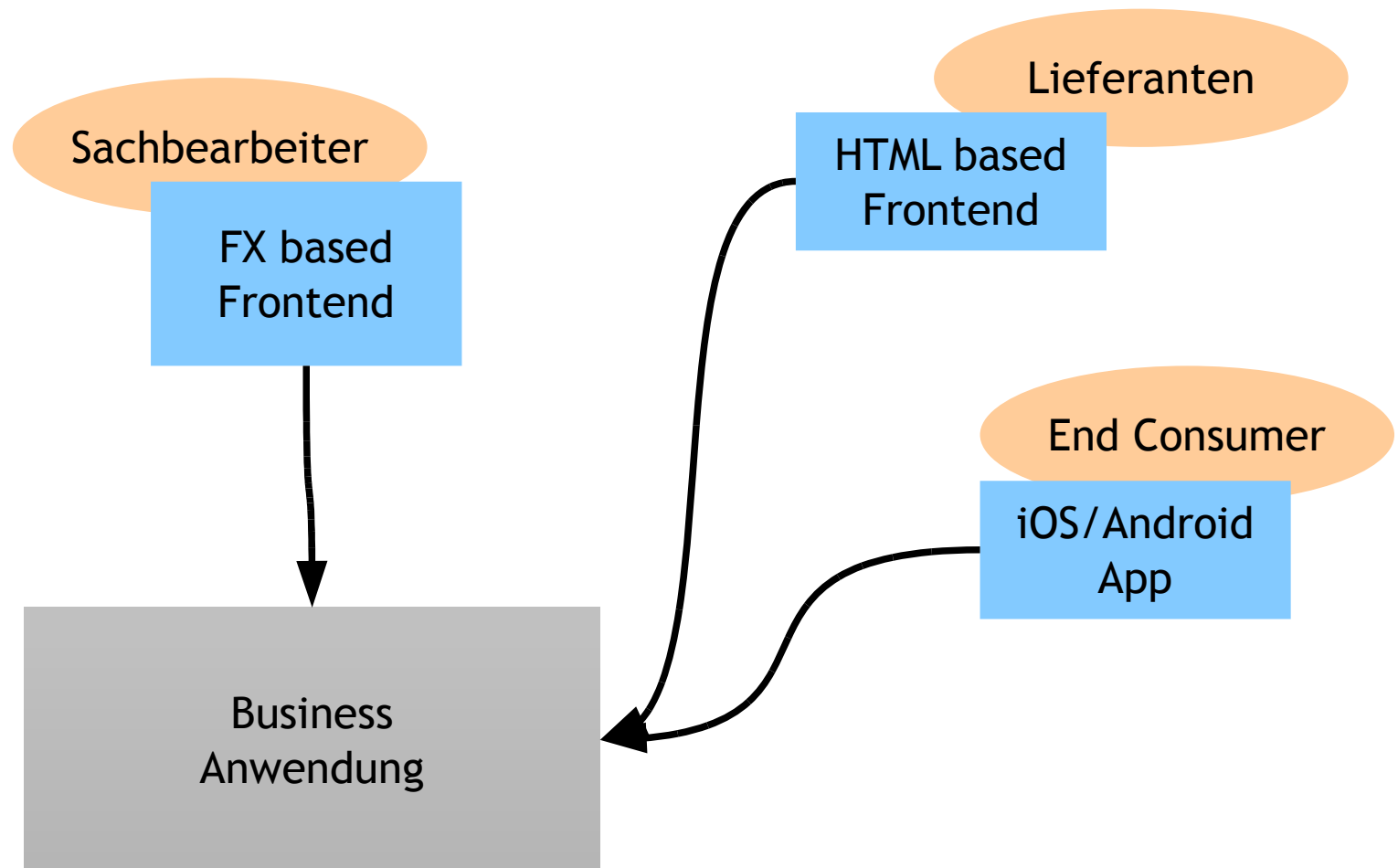
- Bereits lange (und tlw. leidvolle) Geschichte
 - Klares Commitment seitens Oracle / Java
 - Klares Commitment zu Open Source
 - In Tradition von Swing (Eclipse...)



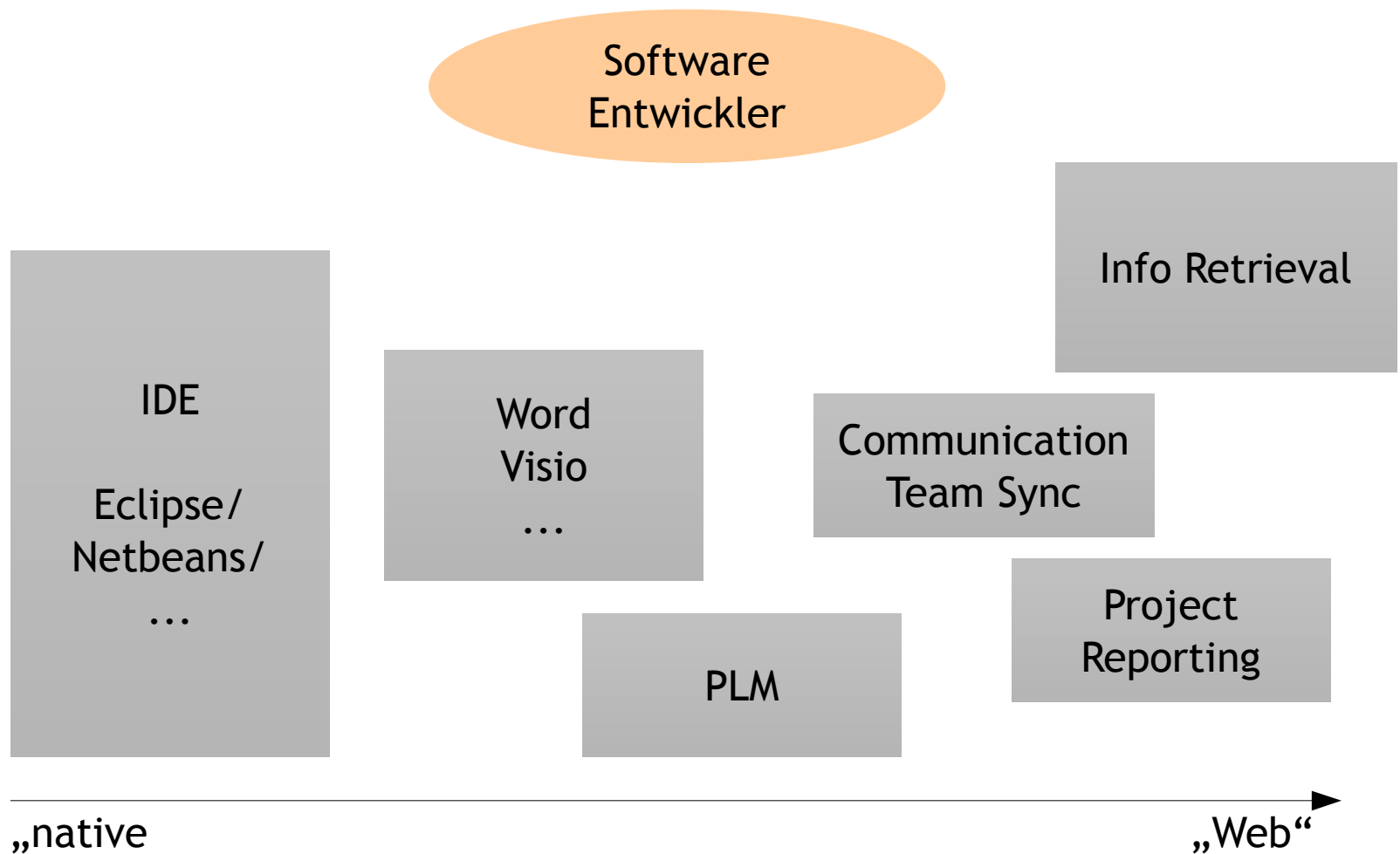
Indikatoren für „JavaFX macht Sinn!“

- „Bekannte“ Endnutzer (Intranet, ...)
- Gewisser Umfang (z.B. Anzahl Dialoge)
- Gewisser Grad an UI Komplexität
- Umfangreiche(re) Logik, die direkt unterhalb des UIs läuft
- Einbindung von Subdevices
- Gewisser Anspruch an Langfristigkeit
- Entwickler, die Java beherrschen oder gar aus dem Swing-Lager kommen

Bitte nicht: „JavaFX everywhere“



Beste Vergleich - Eigene Nase



Auf geht's mit JavaFX!



Auf geht's mit JavaFX!

- ...und dann stellt sich heraus:
 - JavaFX ist aus Entwicklungssicht ein UI Framework mit Kern UI Komponenten.
 - „Mehr nicht!“

- JavaFX beinhaltet kein Kochrezept „So baue ich meine komplette Anwendung“
 - ...und das ist auch gut so!

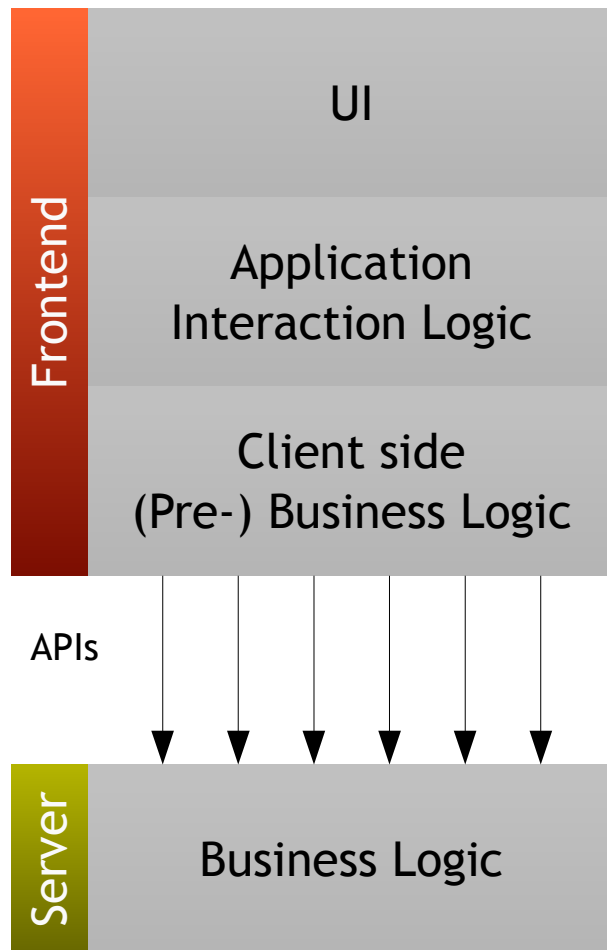


Grundlegende Architekturfrage

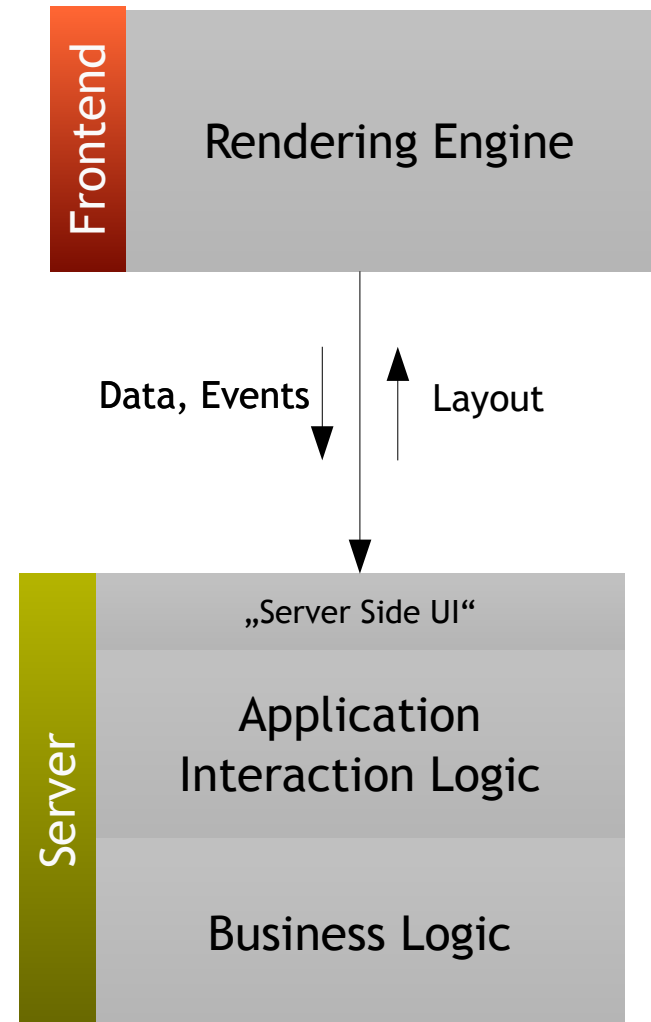
- Client-centric?
- oder
- Server-centric?



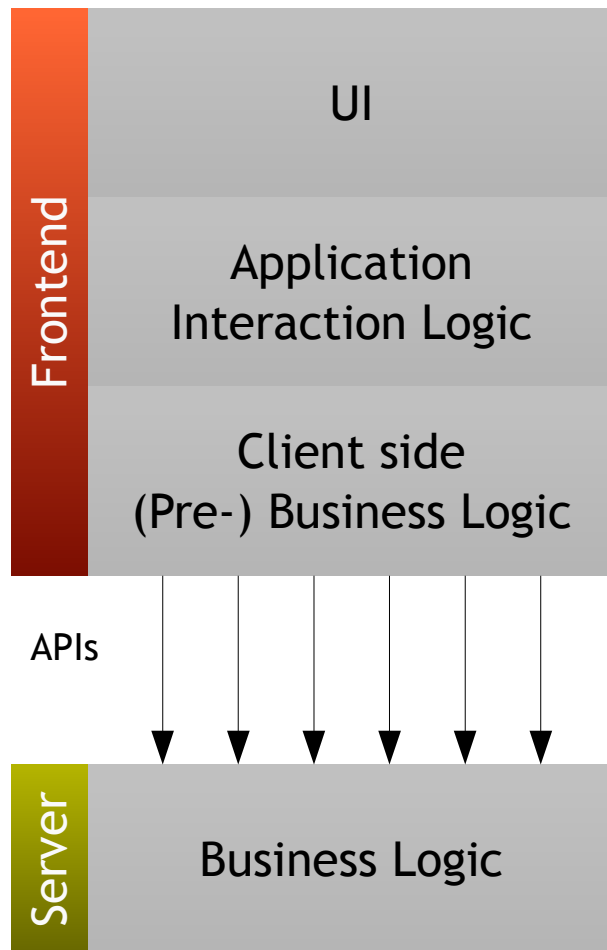
Client Centric



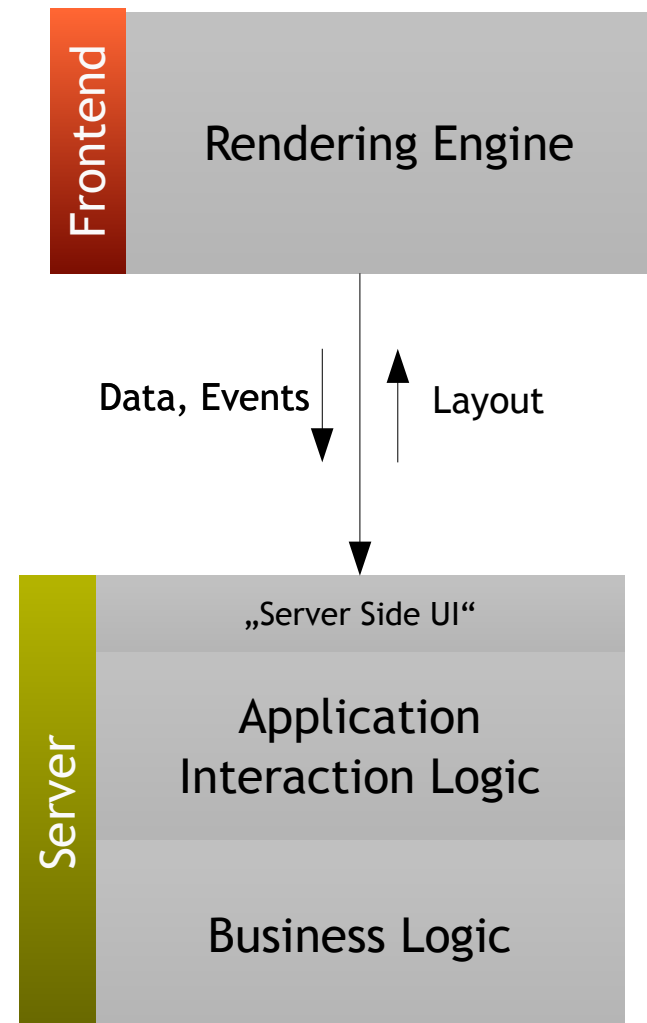
Server Centric



Client Centric

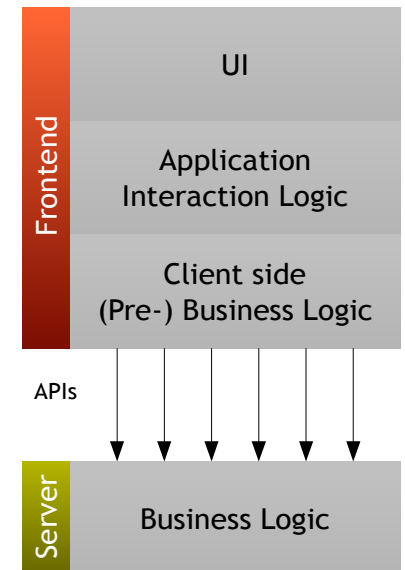


Server Centric



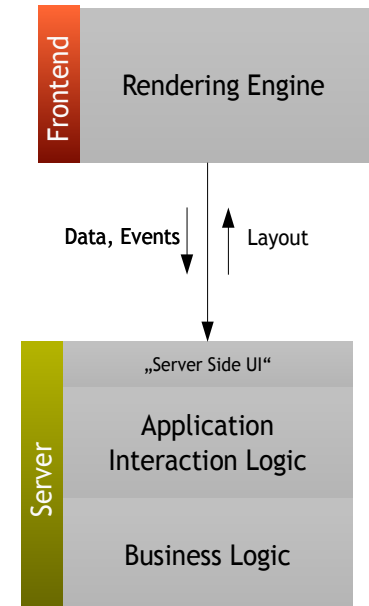
Client Centric

- Pro
 - Direktes Loslegen
 - Voller Zugriff auf UI Möglichkeiten
 - Lokale Datenhaltung möglich
 - State auf Client Seite
- Contra
 - Viele Schnittstellen zum Server
 - Immer größer werdender Client
 - Frontend und Backendentwicklung
- Sinnvoller Einsatz
 - Kleine(re) Szenarien
 - Hohe Disziplin in der Entwicklung



Server Centric

- Pro
 - Nur ein Ort der Entwicklung
 - Nur eine Schnittstelle
 - Einfaches Deployment
 - Abstraktion vom UI
- Contra
 - Kein direkter Zugriff auf Client
 - Synchronisation mit Logik nur über Roundtrip
 - Server State (!)
- Sinnvoller Einsatz
 - Umfangreiche Anwendungen



Enterprise Client - Verwendung von JSF



Client
Form
Processing

Additional
Controls

JavaFX

http(s)



Server
Form
Processing

J2EE- Servlet/JSF

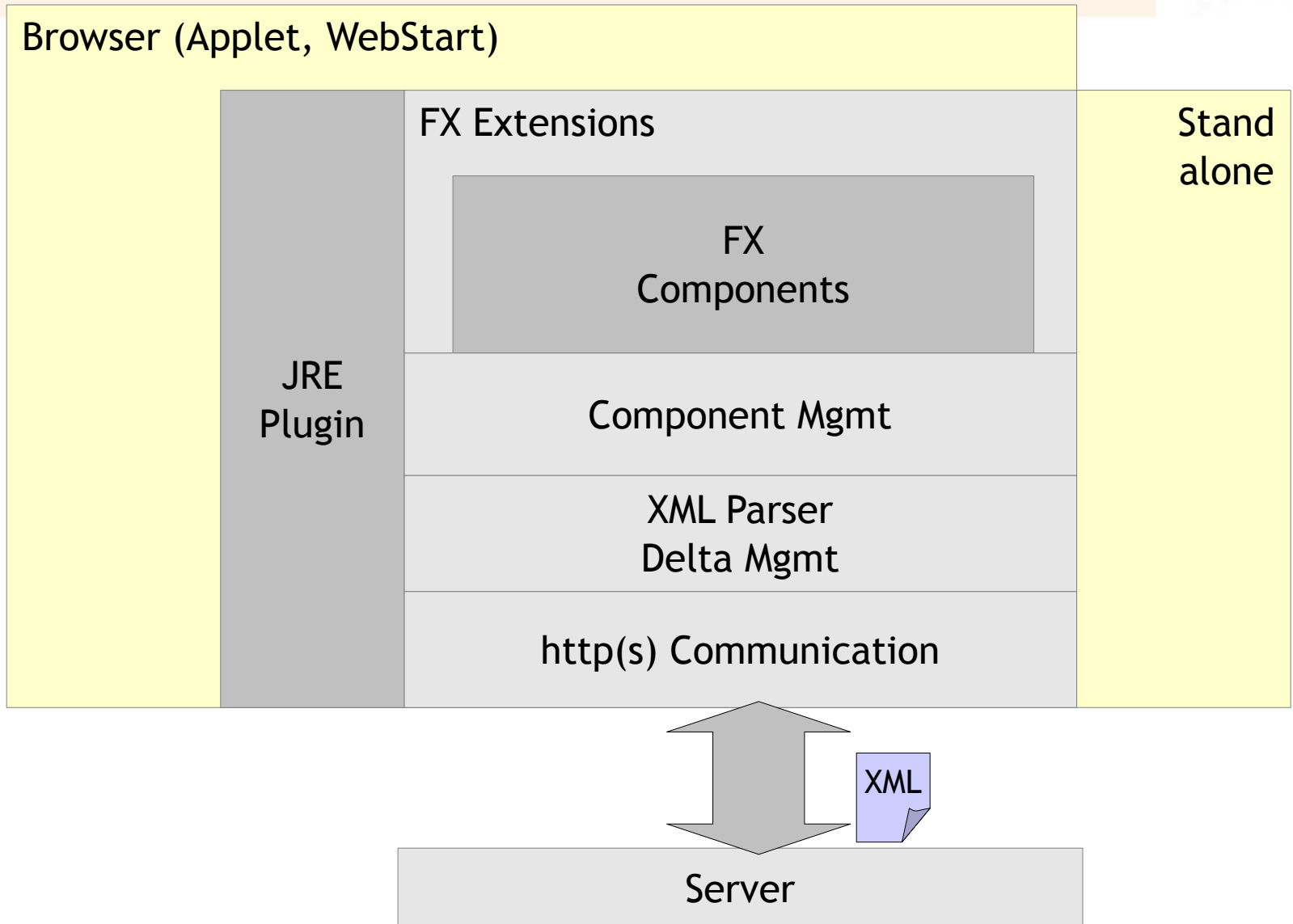
XML
Forms

Form
Java
Logic

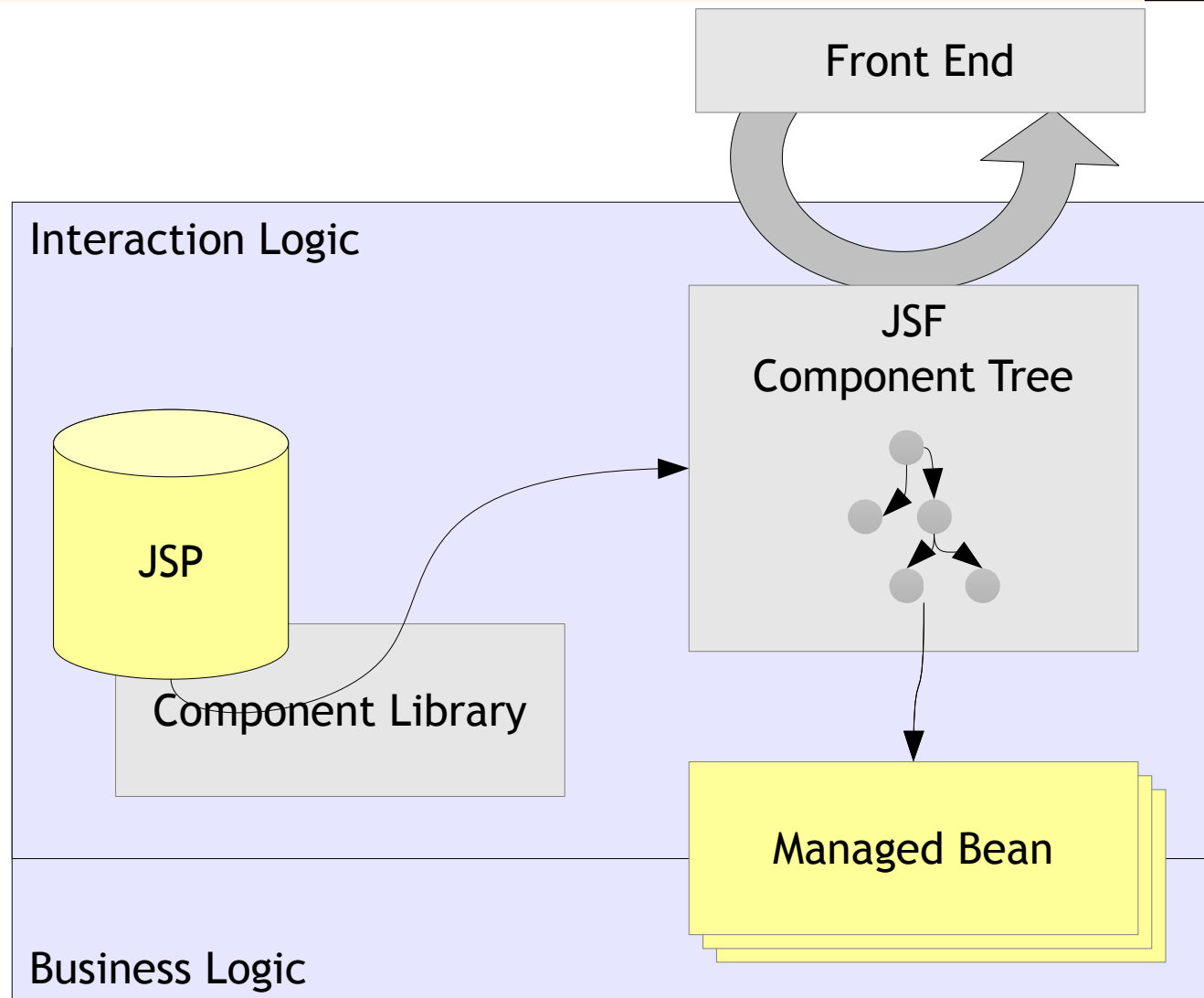
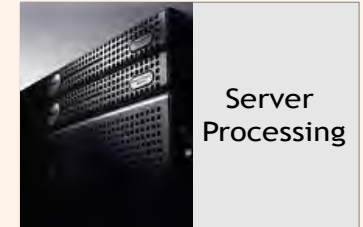
Business Logic



Client Side

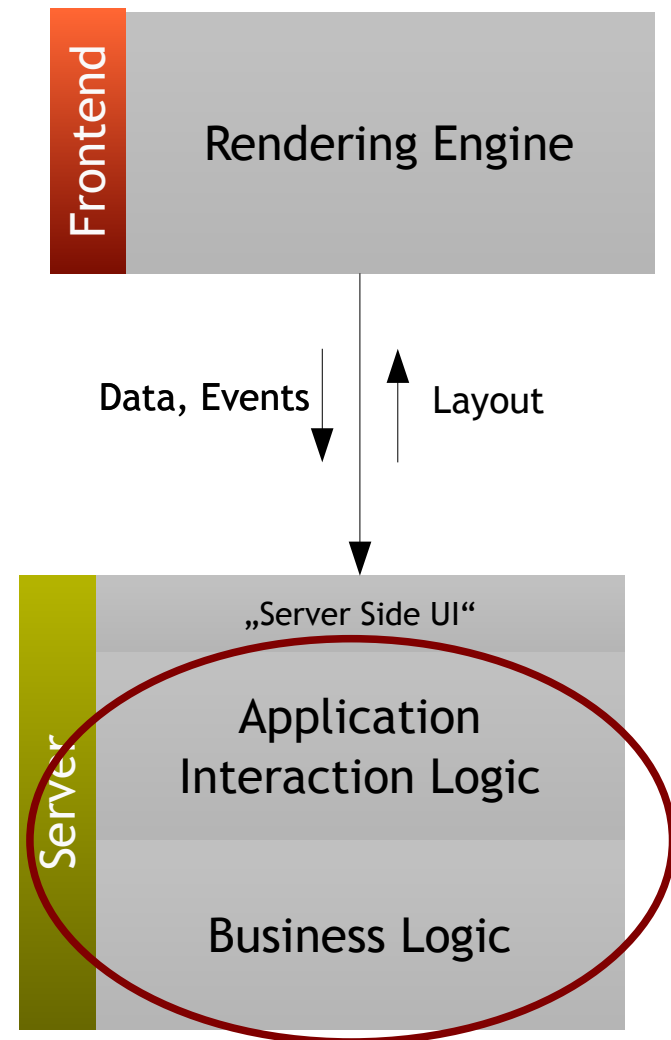


Application: Building Blocks



Server-centric und das Web...

- Server-centric Architekturen sind immer auch „Web-fähig“
- Verwendung eines Standard-Web-Frameworks (z.B. JSF basiert)



Wer JavaFX sagt, muss auch sagen, wie er HTML macht!

- Eine JavaFX-only Architektur ist für umfangreiche
Geschäftsanwendungen nicht möglich
 - Web-Integration in andere Anwendungen
 - Simple-Seiten für Selten-Nutzer
 - Seiten zur Kommunikation mit externen Partnern



Fazit

- JavaFX-basierte Anwendungen werden wie Pilze aus dem Boden sprießen
 - und keiner wird's so richtig mitbekommen...
- Bereiche
 - Komplexe Clients
 - „Apps“
 - Umfangreiche Geschäftsanwendungen
- Wer geschickt ist, positioniert JavaFX nicht gegen HTML5
 - Es gibt genug nicht-Web Bereiche, in denen JavaFX sich etablieren kann

Fazit

- Das Planen von Frontendarchitekturen ist nach wie vor essenziell und wird durch JavaFX nicht abgenommen!

