

# Webapp vs. Mobile App - gesammelte Berufserfahrungen

LVA Anwendungen in  
Wirtschaft und Technik





# Inhalt

- Einleitung
  - Vorstellung der Personen
- Web App
  - Akteure
  - Three-Tier-Achitecture
- Mobile App
  - Spezielle Anforderungen
  - Show-Case: Hybride App mit Angular JS
- Gesammelte Berufserfahrungen



## Das Team

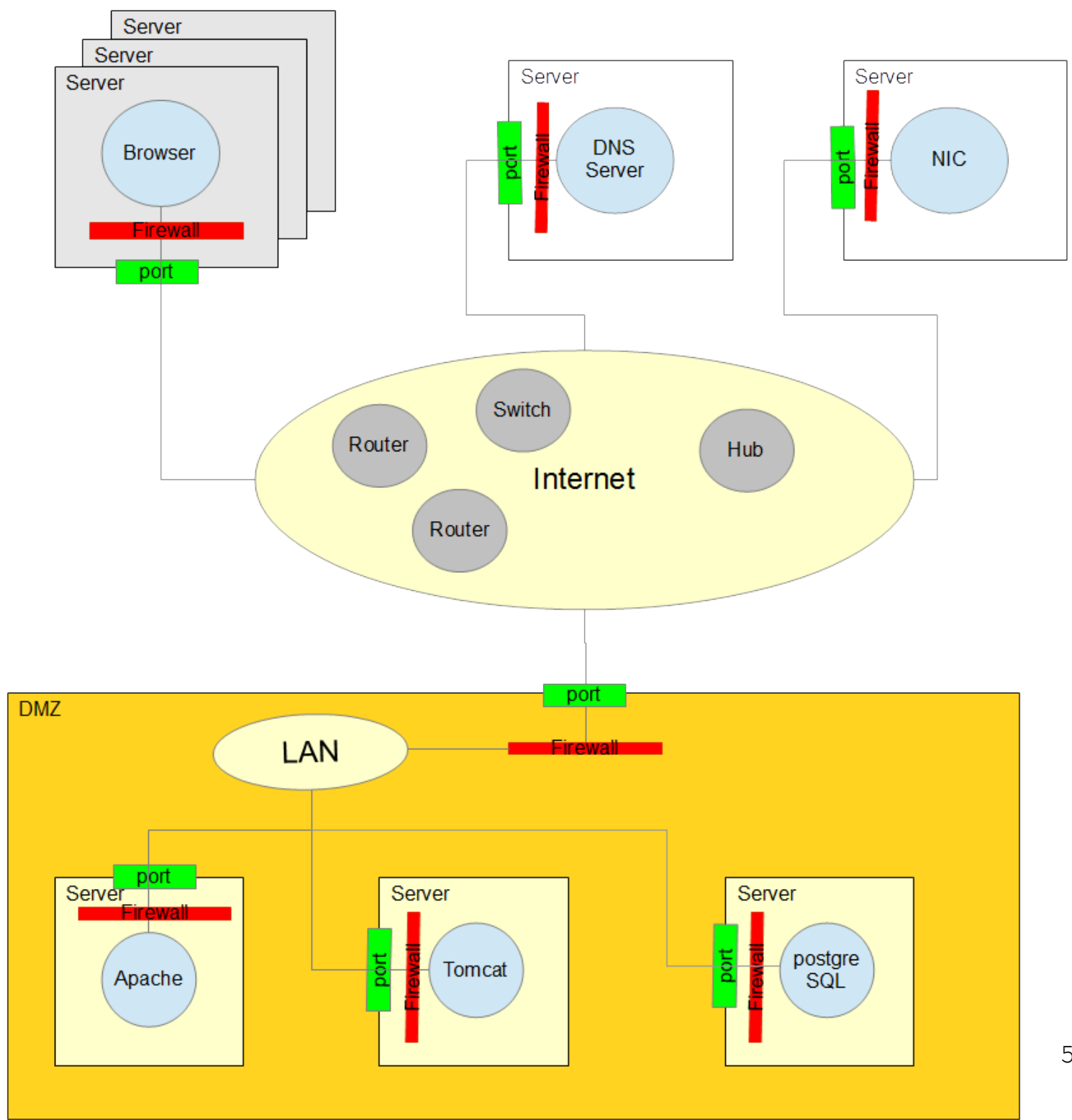
- Mag. Ulrike Walch:
  - 1998 Studium der Wirtschaftsinformatik in Wien
  - 17 Jahre Berufserfahrung
  
- DI Josef Ornetzmüller:
  - 1989 Studium der Informatik in Linz
  - 26 Jahre Berufserfahrung
  - [https://www.xing.com/profile/Josef\\_Ornetsmueller](https://www.xing.com/profile/Josef_Ornetsmueller)



# Thomas Dewar

Minds are like parachutes -  
they only function when open.

# Web-App Akteure



# Akteure 1

- **Router** - zwischen zwei Netzwerken übertragen
- **Switch, Hub** - in einem Netzwerk miteinander kommunizieren
- Ein **Network Information Center** (NIC) oder eine Domain Name Registry verwaltet eine oder mehrere Top-Level-Domains im Domain Name System.
- **Apache HTTP Server** is a powerful, flexible, HTTP/1.1 compliant web server.  
Man bezeichnet einen solchen Server als Webserver, da er ausschließlich Informationen für das Web bereitstellt. Dies können HTML Seiten, Textdokumente, aber auch dynamische datenbankbasierte Seiten sein.

## Akteure 2

- Das **Hypertext Transfer Protocol** zustandsloses Protokoll zur Übertragung von Daten auf der Anwendungsschicht über ein Rechnernetz; hauptsächlich eingesetzt, um Webseiten (Hypertext-Dokumente) aus dem World Wide Web (WWW) in einen Webbrowser zu laden
- **Apache Tomcat** is a web server that is an open source software implementation of the Java Servlet and JavaServer Pages technologies. Tomcat uses a thread pool, and each request will be served by any idle thread in the thread pool. maxThreads decides the maximum number of threads that Tomcat can create to service requests.

# Akteure 3

- Ein Wide Area Network (**WAN**, dt. Weitverkehrsnetz) ist ein Rechnernetz, das sich im Unterschied zu einem LAN oder MAN über einen sehr großen geografischen Bereich erstreckt. Andere WANs werden durch **Internetdiensteanbieter** errichtet oder erweitert, um einen Zugang zum Internet anbieten zu können.
- Die **Firewall-Software** dient dazu, den Netzwerkzugriff zu beschränken, basierend auf Absender- oder Zieladresse und genutzten Diensten. Sie überwacht den durch die Firewall laufenden Datenverkehr und entscheidet anhand festgelegter Regeln, ob bestimmte Netzwerkpakete durchgelassen werden oder nicht.



# Akteure 4

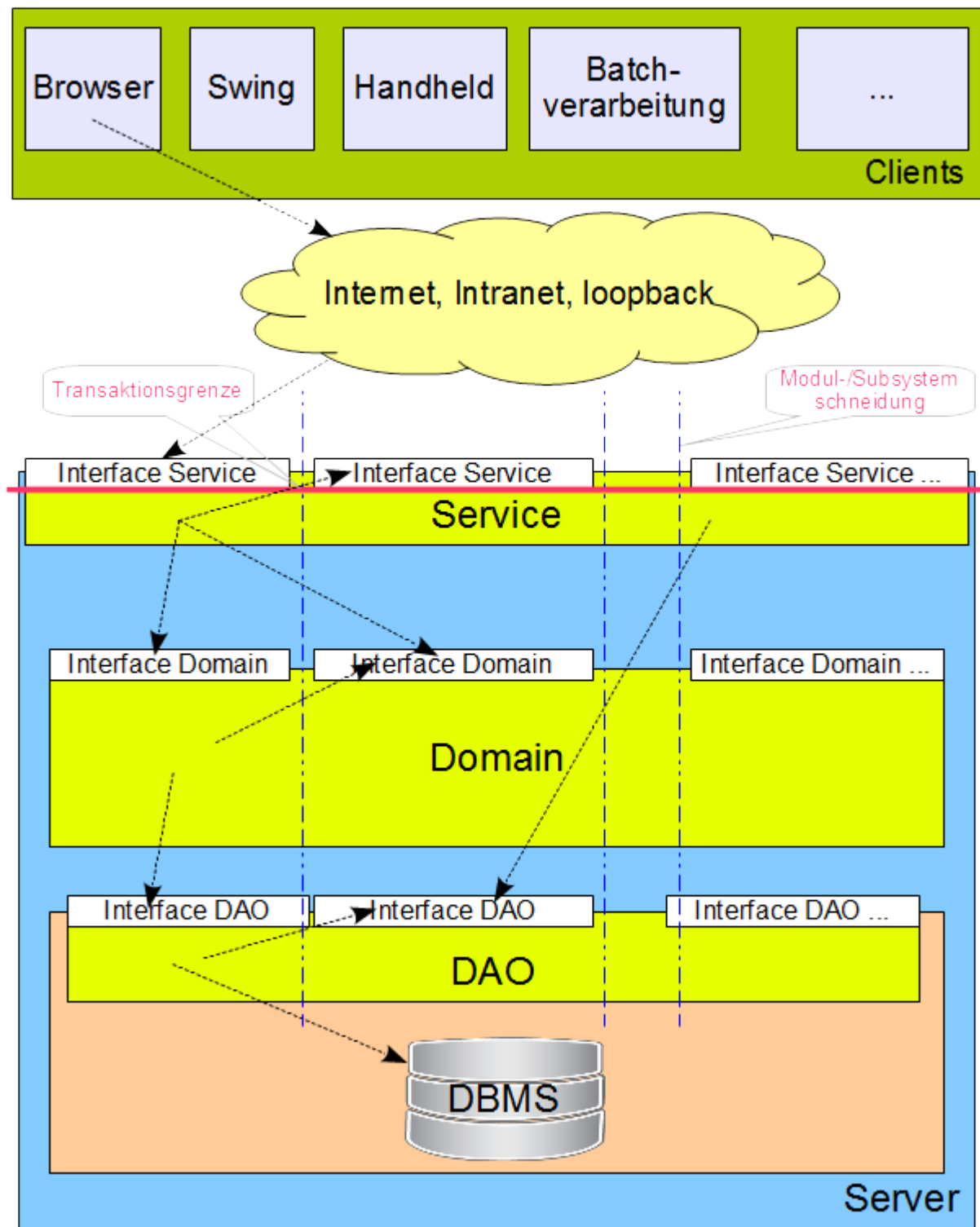
- Eine **Demilitarized Zone** (DMZ, auch ent- oder demilitarisierte Zone) bezeichnet ein Computernetz mit sicherheitstechnisch kontrollierten Zugriffsmöglichkeiten auf die daran angeschlossenen Server.

Die in der DMZ aufgestellten Systeme werden durch eine oder mehrere Firewalls gegen andere Netze (z. B. Internet, LAN) abgeschirmt. Durch diese Trennung kann der Zugriff auf öffentlich erreichbare Dienste (Bastion Hosts mit z. B. E-Mail, WWW o. ä.) gestattet und gleichzeitig das interne Netz (LAN) vor unberechtigten Zugriffen von außen geschützt werden.

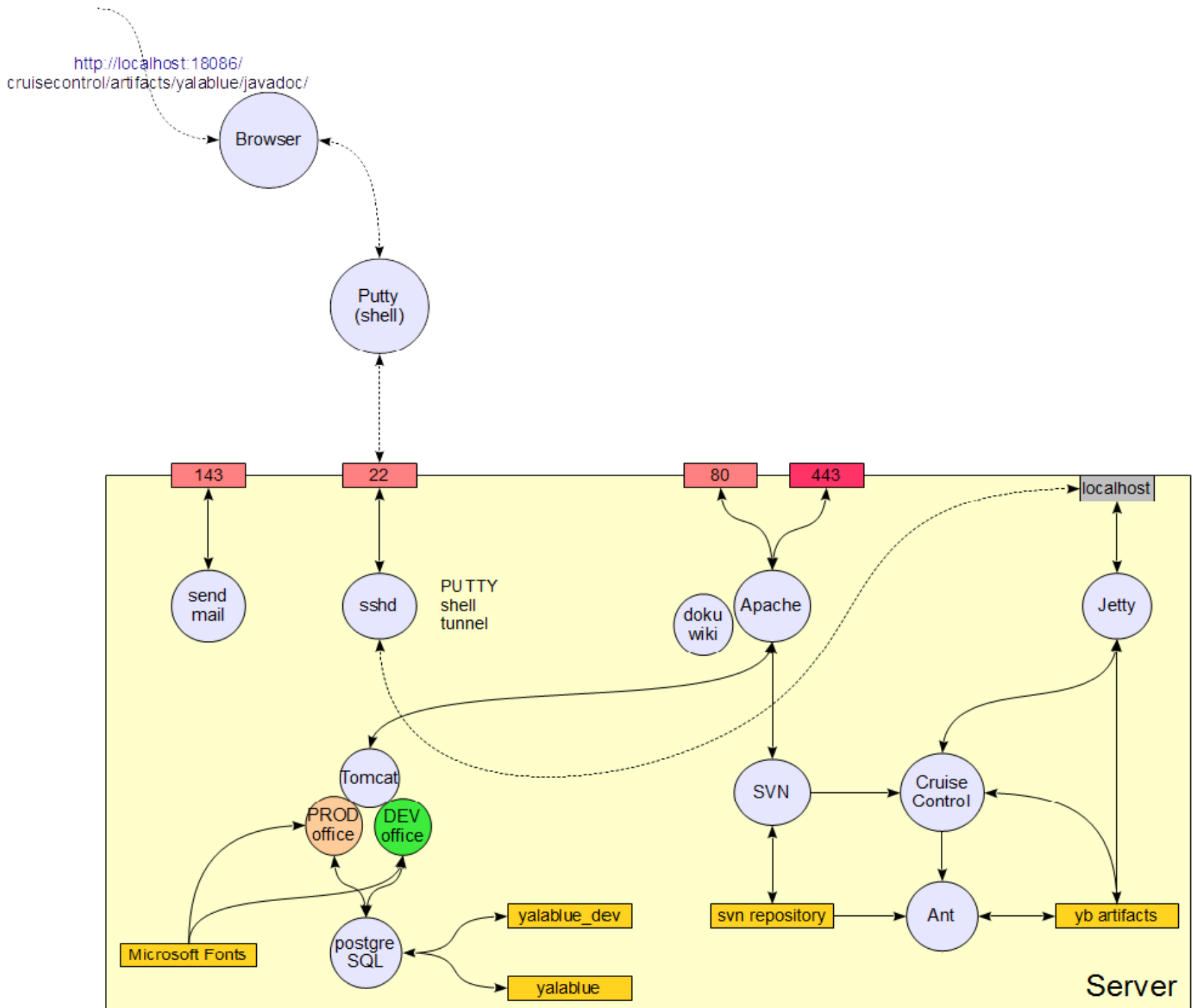
# Akteure 5

- Das **Internet** (von englisch internetwork, zusammengesetzt aus dem Präfix inter und network ‚Netzwerk‘ oder kurz net ‚Netz‘), ist ein weltweiter Verbund von Rechnernetzwerken, den Autonomen Systemen. Es ermöglicht die Nutzung von Internetdiensten wie **WWW**, E-Mail, Telnet, Usenet und FTP.
- Ein **Port** ist der Teil einer Netzwerk-Adresse, der die Zuordnung von TCP- und UDP-Verbindungen und -Datenpaketen zu Server- und Client-Programmen durch Betriebssysteme bewirkt. Zu jeder Verbindung dieser beiden Protokolle gehören zwei Ports, je einer auf Seiten des Clients und des Servers.

# Three Tier Architecture



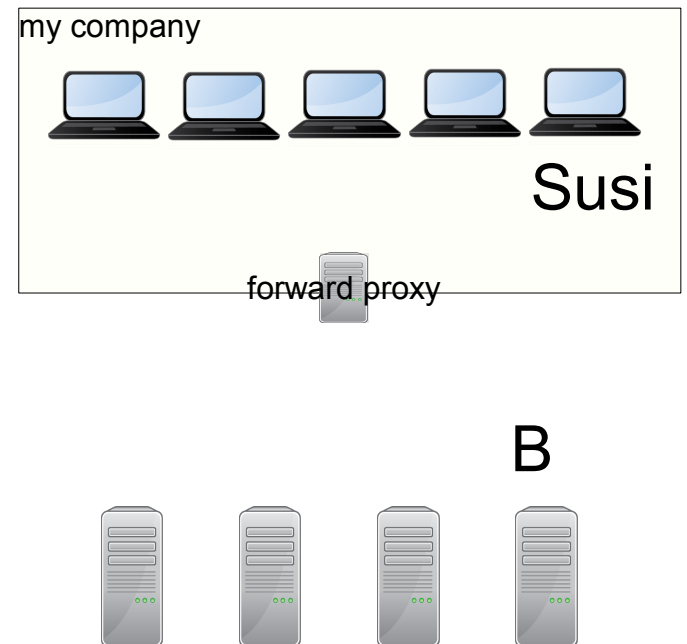
# Buildserver Akteure



# Forward Proxy

- Susi holt sich Websitedaten über einen "forward proxy"
- Eigentlich möchte Susi direkt nach B; Susi kann nicht direkt nach B weil...
  - Administrator blockiert Susis Internetzugriff auf B
  - Z.B. XY.com wird für Angestellte blockiert

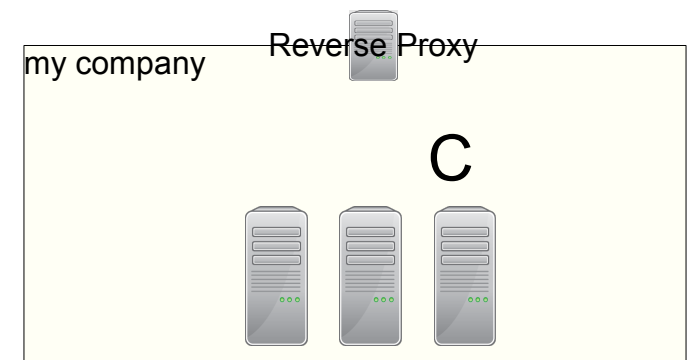
- Administrator von B blockiert Susis IP Adresse, warum...
- Susi möchte B hacken
- Susi möchte B spammen



# Reverse Proxy

- Susi möchte direkt zu C
- C möchte den 'herreinkommenden' traffic über den Reverse Proxy leiten; kein direkter Zugriff!
- Unterschied zu froward Proxy, Susi weiß nicht, dass Sie über einen Reverse Proxy zugreift
- Reverse Proxy ist weniger sichtbar; Susi muss nichts einstellen oder wissen; Susi glaubt sie greift direkt zu
- C ist 'Internetgigant'; ein einziger Webserver schafft traffic nicht

- C versteckt hinter dem Reverse Proxy viele Webserver; C möchte nicht seinen Hauptserver direkt angreifbar machen
- Spam-Angriff legt nicht Hauptserver lahm



# Webapp - Anforderungen

- Hochverfügbar (7 x 24; 99%)
- Servlet-Container: Tomcat, Jetty, ...; Application-Server: JBoss, WebSphere, ...
- Sicher (Angriffe, ...)
- Robust (NullPointerException, ...)
- Threadsafe (Multithreading)
- Performant (Cluster); Skalierbar; eine CPU, Memory mehr
- User Interface ist intuitiv
- Einfach zu warten
- Rollenbasierte Authentifizierung (Wer) und Autorisierung (Was)
- Sichtbarkeit von UI-Komponenten
- Wahl der Service Providers; Physischer Standort
- ...

## Demo & Links

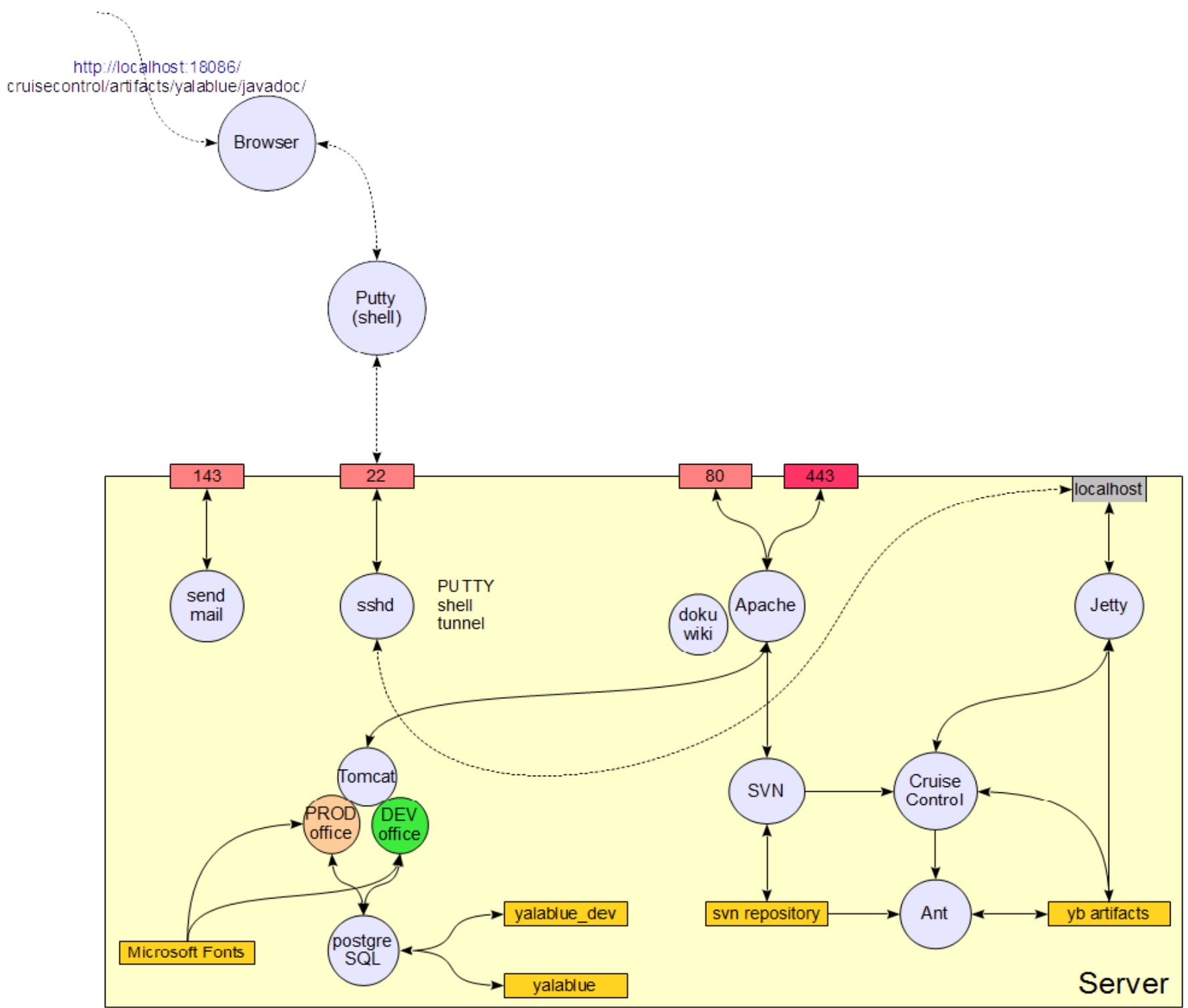
- 7 €/Monat für Ihre Zukunft
- <http://www.yalablue.com/>
- <https://office.yalablue.com>
- [http://office\\_test.yalablue.com](http://office_test.yalablue.com)
- Help
- Wiki
- Firefox; Firebug; HTML, CSS, JavaScript Debugger; Bildschirmgrößen
- Google Chrome; Inspektor
- Internet Explorer; DOM Explorer
- Optimistic lock exception
- Code coverage
- JavaDoc



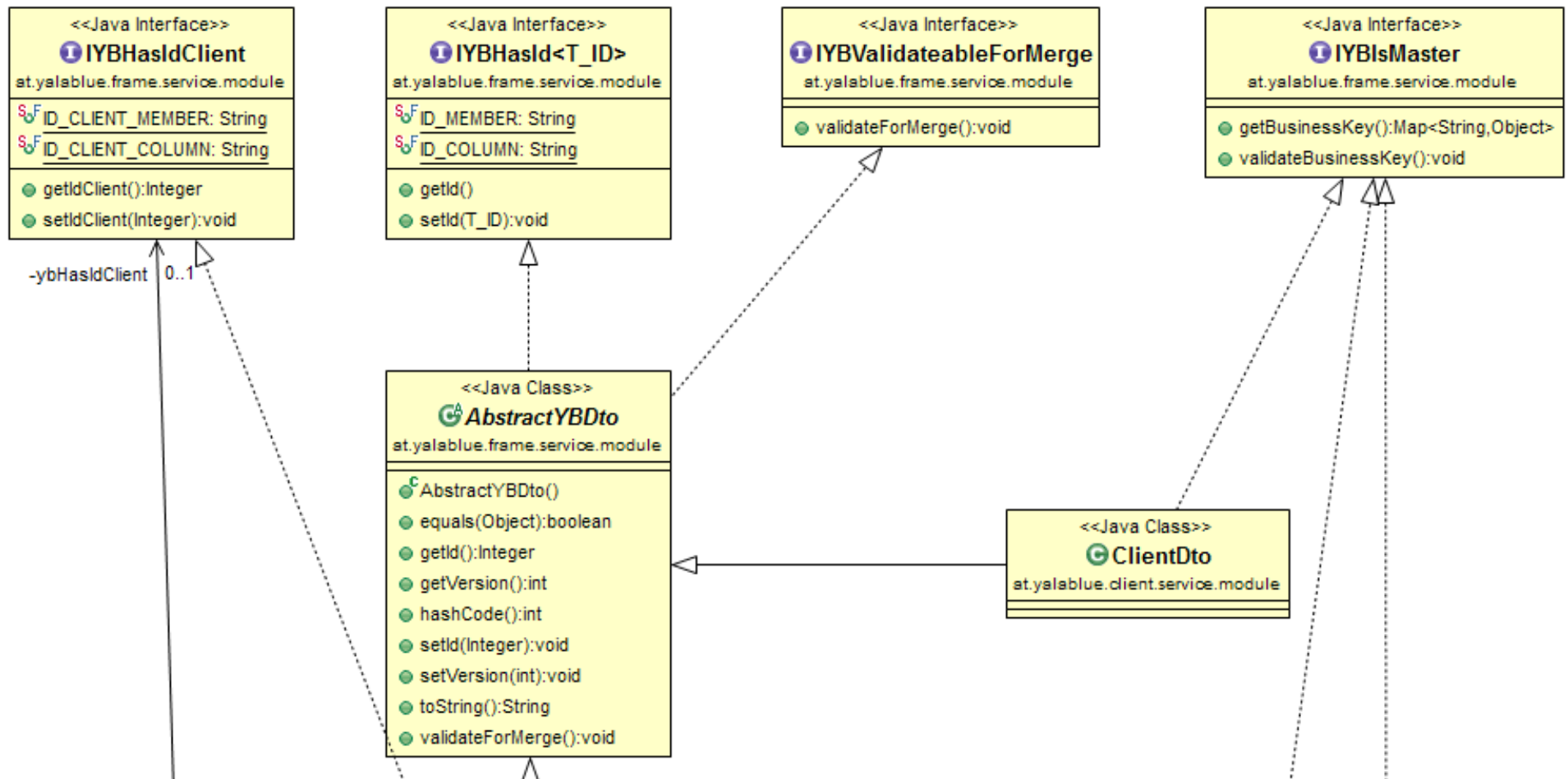
# 3<sup>rd</sup> Party Software

- Java SDK
- Apache, Ant
- Connection Pool
- Code coverage
- Cruise Control
- DokuWiki
- Eclipse J2EE
- Google Analytics
- Hibernate; Hibernate Lucene
- JQuery Calendar
- Jasper Reports
- LIQUIBASE
- Log4J
- Mail
- Putty
- Spring
- TestNG
- Tomcat; Wicket
- guava; postgresSQL
- Eclipse PlugIns: Subclipse, ER-Master, SortIt, CommonClipse, UML, PMD

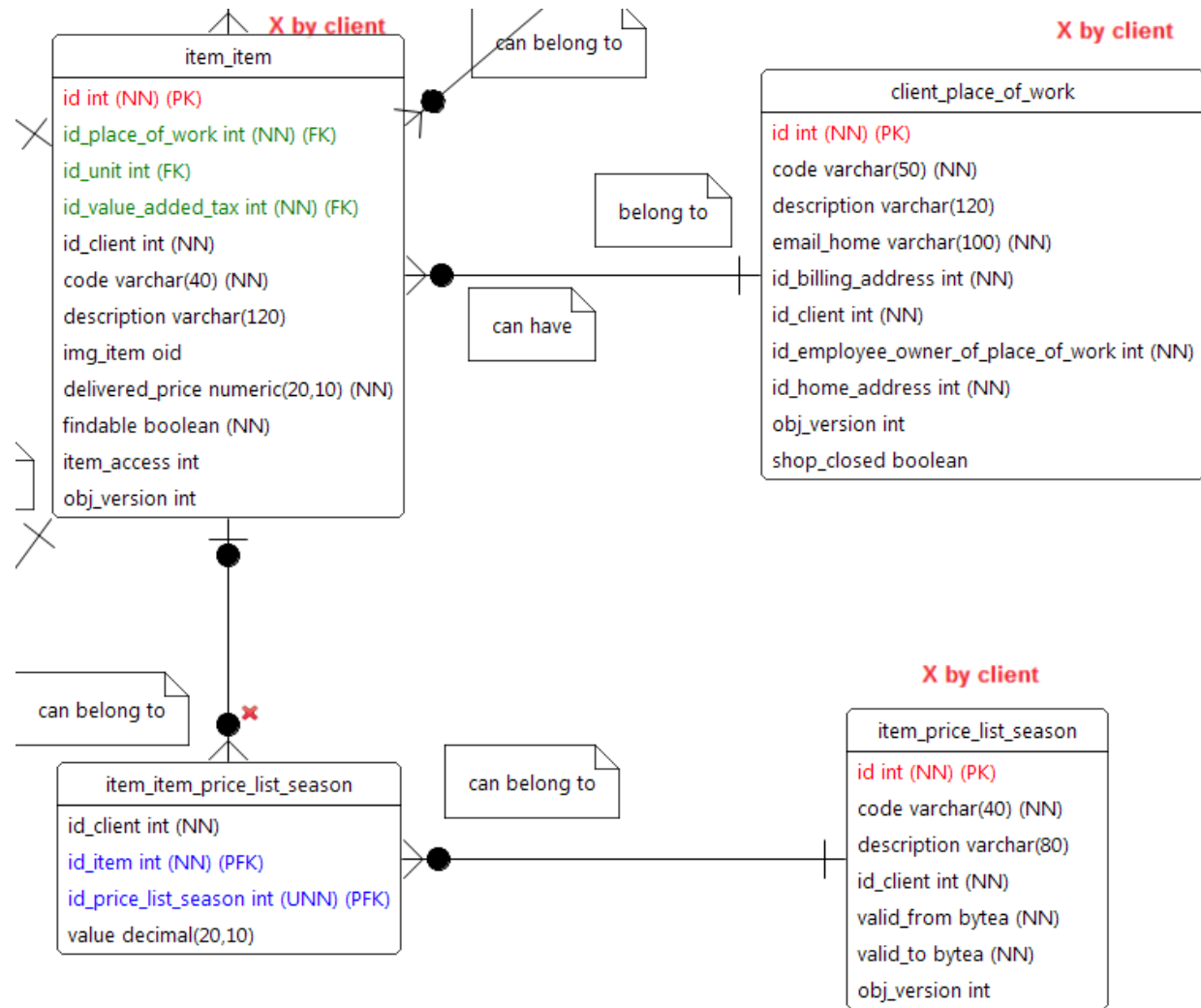
# Build Server Akteure



# UML Class diagram



# ER diagram



# Build

- Build und Test laufen schichtweise ab; bottom up
- Cruisecontrol (Jenkins) steuert; Ant (Maven) führt aus
- Artefakte wie WAR-files, Datenbanken, Javadoc, ... werden erzeugt
- Toolbox unterstützt Entwickler und Cruisecontrol mit Werkzeugen wie clean tc, all; Javadoc, native2ASCII, ...
- Grundstruktur build: clean, build, test
- Ant tasks für db, tomcat, yb-server, yb-web  
SourceSafe, CVS, SVN, Git

# Lösungen für Industrie 4.0



# Der Weg zur mobilen Applikation

- Klassische Browser-Anwendungen für große Businessanwendungen
- Kennzeichen:
  - großes Datenvolumen, das aufbereitet und dargestellt werden muss
  - aufwendiges User Interface
  - komplexe Workflows und aufwendige User-Interaktionen
- Aber: Benutzer verwenden Mobile Devices
- Im einfachsten Fall: Mobile Client für bestehenden Server



# Spezielle Herausforderungen an Mobile Apps

- Viele verschiedene Hersteller, Betriebssysteme, Betriebssystemversionen
- Geringe Bandbreiten
- Netzverfügbarkeit
- Geringe Devicegröße:
  - Wieviele Daten können sinnvoll dargestellt werden?
  - Wie kann der Workflow sinnvoll gestaltet werden (Userinteraktion)?
- ...





# Grundsätzliche Entscheidungen

- Browser-Anwendung portieren ja/nein?
- Wenn Mobile App ja:
  - Zielgruppe, Zielplattform?
  - Native oder hybride App?
- Native: Technologiestack durch die Plattform vorgegeben
- Hybrid: HTML5/CSS3 mit Native Features via Cordova/PhoneGap



# Show-Case: Materialliste mit Suchkomponente

## Anforderungen:

- Zielgruppe: „alle“
  - Stichwortsuche: „mobile app marktanteile“
  - <http://www.app-entwicklung.info/2014/12/mobile-betriebssysteme-verbreitung-und-marktanteile-mit-stand-dezember-2014/>
  - Zielplattform: Android, iOS, (WP)
- Hybride App (HTML5/CSS3/JavaScript) auf Basis von Cordova/PhoneGap

## Materialliste



| Name       | Bestand |
|------------|---------|
| Schraube X | 25      |
| Schraube Y | 21      |
| Dichtung   | 21      |
| Gewinde    | 2       |

Was brauchen Sie, um eine Cordova-Anwendung unter Android zu entwickeln?

- Android Studio 1.x: Java; Device Emulatoren, Deploy auf echtes Device
- \* Device-Abstraktion: Cordova mit diversen Plugins
- \* lokale Datenbank: SQLite
- \* Angular JS mit ngCordova
- \* Ionic: <http://ionicframework.com/>
- Google Chrome: Debuggen der Anwendung auf echtem Device

## Materialliste



| Name       | Bestand |
|------------|---------|
| Schraube X | 25      |
| Schraube Y | 21      |



# Debug Android in Google Chrome

The screenshot shows the Google Chrome browser interface with the DevTools panel open. The 'Devices' panel is active, displaying two virtual devices:

- SM-T520** #3204A1E7875AB18D  
Materialiste file:///android\_asset/www/index.html#/search at (0, 50) size 2560 x 808  
[inspect](#)
- WebView in com.ionicframework.white456034 (Version/4.0 Chrome/30.0.0.0)**  
iMobility file:///android\_asset/www/index.html#/olSideMenu/materialAdmin hidden at (0, 50) size 2560 x 1550  
[inspect](#)

A context menu is open over the 'Geräte untersuchen' (Inspect devices) option in the left sidebar. The menu items are:

- App-Verknüpfungen erstellen...
- Erweiterungen
- Taskmanager Umschalt+Esc
- Browserdaten löschen... Strg+Umschalt+Entf
- Codierung
- Quelltext anzeigen Strg+U
- Entwicklertools Strg+Umschalt+I
- JavaScript-Konsole Strg+Umschalt+J
- Geräte untersuchen** (highlighted)

Simultaneously, a standard Chrome context menu is open over the top right of the browser window. The menu items are:

- Neuer Tab Strg+T
- Neues Fenster Strg+N
- Neues Inkognito-Fenster Strg+Umschalt+N
- Lesezeichen
- Zuletzt geöffnete Tabs
- Bearbeiten | Ausschneiden | Kopieren | Einfügen
- Seite speichern unter... Strg+S
- Suchen... Strg+F
- Drucken... Strg+P
- Zoomen - 100% +
- Verlauf Strg+H
- Downloads Strg+J
- Einstellungen
- Über Google Chrome
- Hilfe
- Weitere Tools** (highlighted)
- Beenden Strg+Umschalt+Q



# Round-Trip vs. Single-Page Application

- Round trip

Klassische Web-Anwendung, die auf dem HTTP-Protokoll basiert (Request / Response)

- Single page

Initiales HTML-Dokument, Views werden dann über AJAX häppchenweise geladen

- Angular JS: MVC und Unit-Testing

# Gesammelte Berufserfahrungen

Zuerst kommt das...

...das  
Wichtige  
dann das  
Richtige

...das  
Richtige  
dann das  
Wichtige



# Technische Skills

- Software Design:
  - Pattern, Tiers
  - Domain Driven Design
  - Test Driven Design
- Programmiersprachen:
  - C++, Java (SDK), Cobol
- Betriebssysteme:
  - Windows, Unix/Linux
  - Ihre UNIX-Box im Internet sichern
- PROD und DEV, TEST einrichten
- Datenbanken (SQL, DBMS)
- CVS, SVN, Git
- SCRUM
- Server-Technologien:
  - J2EE EJB
  - Spring, Hibernate
- Client-Technologien:
  - JSP, JSF, Wicket
  - HTML, CSS, JavaScript
- Mobile App:
  - jQuery Mobile
  - Angular JS
  - Cordova/Phone Gap





# Soft Skills

- KISS
- Focus on point
- Fragen stellen
- **Teamfähigkeit**
- Veranschaulichung von Problemen und Lösungsansätzen
- Durchstich
- Humor
- Kreativität
- Kundenkontakt
- Einarbeitung in Fachlichkeit (Domain)
- Probleme in Arbeitspakete zerlegen
- Am falschen Weg umkehren
- Das gemeinsame Ziel verfolgen

# Dokumentation

- JavaDoc sinnvoll nutzen
- Nahe am Code
- Tools generieren automatisch
- Sie lesen viel Code
- Sie finden einen alten Dokumentationsfriedhof
- Unwartbar; Zuviel Aufwand
- Sowas wie Confluence und JIRA
- Übertreiben Sie nicht
- Token – Querbeet
- Wie 'tief' – ist 6 Monate im Team



# Update-to-Date bleiben

- Konzeptwissen Halbwertszeit von 10 bis 15 Jahren, Produktwissen anderthalb bis zwei Jahren
- Lesen lesen lesen
- Konferenzen besuchen
- Flexibilität bewahren
- Zertifizieren Sie sich
- Nützen Sie die Gelegenheit zu Technologieevaluierungen
- Fragen Sie Kollegen, profitieren Sie von den Fehlern, die andere schon gemacht haben



# Programmier-Alltag I

- Wie schnell kann ich den Fehler reproduzieren?
- Wie schnell bin ich an der „frisch“ programmierten Stelle?
- Wie lange brauche ich um eine „Seite“ Code zu verstehen?
- Wer darf nicht auf Urlaub gehen, damit ich weiter arbeiten kann?
- Bin ich von externen Systemen abhängig (DB, Server) oder kann ich autark arbeiten (lokale IDE)?
- Gibt es fachliche Ansprechpartner, aktuelle fachliche Dokumentation oder ist die Dokumentation der Code mit den „sprechenden“ Variablennamen?..



# Programmier-Alltag II

- Ist der Code noch refactorfähig?
- Kann ich sorgenfrei Bugfixen, einen Feature Request abarbeiten?
- Habe ich überwiegend Spass an der Arbeit?
- Gibt es einen letzten Breakpoint?
- Wie finde ich mich in der Packagestruktur zurecht?
- Typen:
  - IchWeißAlles
  - Mitschwimmer
  - Buddy



# Ende

- Fragen?

*JUST IN CASE YOU'RE STILL NOT  
SURE WHETHER YOU'RE IN A  
SOFTWARE PROJECT*

*WAIT UNTIL YOU HEAR THIS:*

