



Bauen Sie eine Webapp - Update

Mögliche Ingredienzien





Das Team

⇒ Mag. Ulrike Walch:

- 1998 Studium der Wirtschaftsinformatik in Wien
- 16 Jahre Berufserfahrung
- https://www.xing.com/profile/Ulrike_Walch

⇒ DI Josef Ornetzmüller:

- 1989 Studium der Informatik in Linz
- 25 Jahre Berufserfahrung
- https://www.xing.com/profile/Josef_Ornetsmueller



Inhalt

⇒ Einleitung

- Vorstellung der Personen
- Das Projekt yalablue

⇒ Web App

- Die Web App yalablue
- Mögliche Ingredienzien: Systemarchitektur & Softwareentwicklung

⇒ Mobile App

- Das Mobile Web
- jQuery Mobile
- Native Apps



Die Web App `yalablue.office`

- ➔ Sie möchten Ihren Klein- oder Mittelbetrieb, einen Verein oder vielleicht auch eine Genossenschaft mit allen Kunden, Mitarbeitern, Betriebsstätten, Rechnungen und Terminen einfach zentral verwalten?
- ➔ Es ist Ihnen wichtig, zu jeder Zeit und an jedem beliebigen Ort und mit einem Internet-Browser auf Ihre Daten zuzugreifen?
- ➔ Sie möchten sich nicht um Installation, Updates oder Sicherungen von Software auf eigenen Rechnern kümmern?
- ➔ Ihr Mandant wächst, Sie möchten eine zusätzliche Betriebsstätte integrieren, dafür aber keinen größeren neuen Rechner kaufen?
Sie wollen mehrere Mandanten hierarchisch abbilden?



Das yalablue Framework

- ➔ Sie suchen als Softwarefirma einen Framework, der alle oben beschriebenen Funktionalitäten abdeckt und dabei mit State-of-the-Art-Technologien umgesetzt wurde?
- ➔ Dann haben wir das Richtige für Sie! Unser Framework beruht auf einem Java Development Kit (JDK), das User Interface arbeitet mit Wicket, die Dreischichtarchitektur wird von Spring getrennt und Hibernate bindet beinahe jedes relationale Datenbanksystem an; ausgeführt wird die Webapplikation mit einem Apache Tomcat.



Anforderungen an eine Web App

⇒ Anforderungen an die Architektur

- Hochverfügbar, sicher, multithreaded
- Performant, einfach zu verwenden, einfach zu warten, skalierbar

⇒ Anforderungen an die Datensicherheit

- Rollenbasierte Authentifizierung und Autorisierung
- Software-immanente Sicherheitsmechanismen (JAAS)
- Sichtbarkeit von UI-Komponenten
- Wahl der Service Providers
- Physischer Standort



Anforderungen an eine Web App

- ⇒ Service Level Agreement („Dienstgütevereinbarung“)
 - Welche Dienste bieten Sie dem Benutzer Ihrer Web App?
 - Welche Dienste bietet Ihr Service Provider Ihnen?



Demo & Links

- <http://www.yalablue.com/>
- <https://office.yalablue.com>
- http://office_test.yalablue.com
- Firefox > Element mit Firebug untersuchen > HTML, CSS, JavaScript Debugger
- Firefox > Webentwickler > Bildschirmgrößen testen
- Google Chrome > Inspektor
- Internet Explorer > DOM Explorer
- Help
- Admin
- Optimistic lock exception
- Code coverage
- Javadoc



Technische Skills

- Pattern
- Tiers
- Software design
- Testing
- HTML
- CSS
- Java
- JavaScript
- SQL
- jQuery
- Hibernate
- Spring
- DB installieren
- DB warten
- Win und UNIX administrieren
- PROD und DEV einrichten
- Ihre UNIX-Box im Internet sichern



Soft Skills

- ⇒ KISS
- ⇒ Focus on point
- ⇒ Fragen stellen
- ⇒ Teamfähigkeit
- ⇒ Veranschaulichung von Problemen und Lösungsansätzen
- ⇒ Kreativität
- ⇒ Kundenkontakt
- ⇒ Einarbeitung in fachliche Fragestellungen
- ⇒ Probleme in Arbeitspakete zerlegen
- ⇒ Das gemeinsame Ziel verfolgen



Open Source 3rd Party Software

- Ant
- Apache
- BoneCP
- Code coverage
- Cruise Control
- DokuWiki
- Eclipse J2EE
- Google Analytics
- Hibernate
- Hibernate Lucene
- JQuery Calendar
- Jasper Reports
- Java SDK
- Joda Time
- LIQUIBASE
- Log4J
- Mail
- PuTTY
- Spring
- TestNG
- Tomcat
- Wicket
- guava
- postgresSQL
- Show 3rd party
- Eclipse PlugIns: Subclipse, ER-Master, SortIt, CommonClipse, UML, PMD

The Server

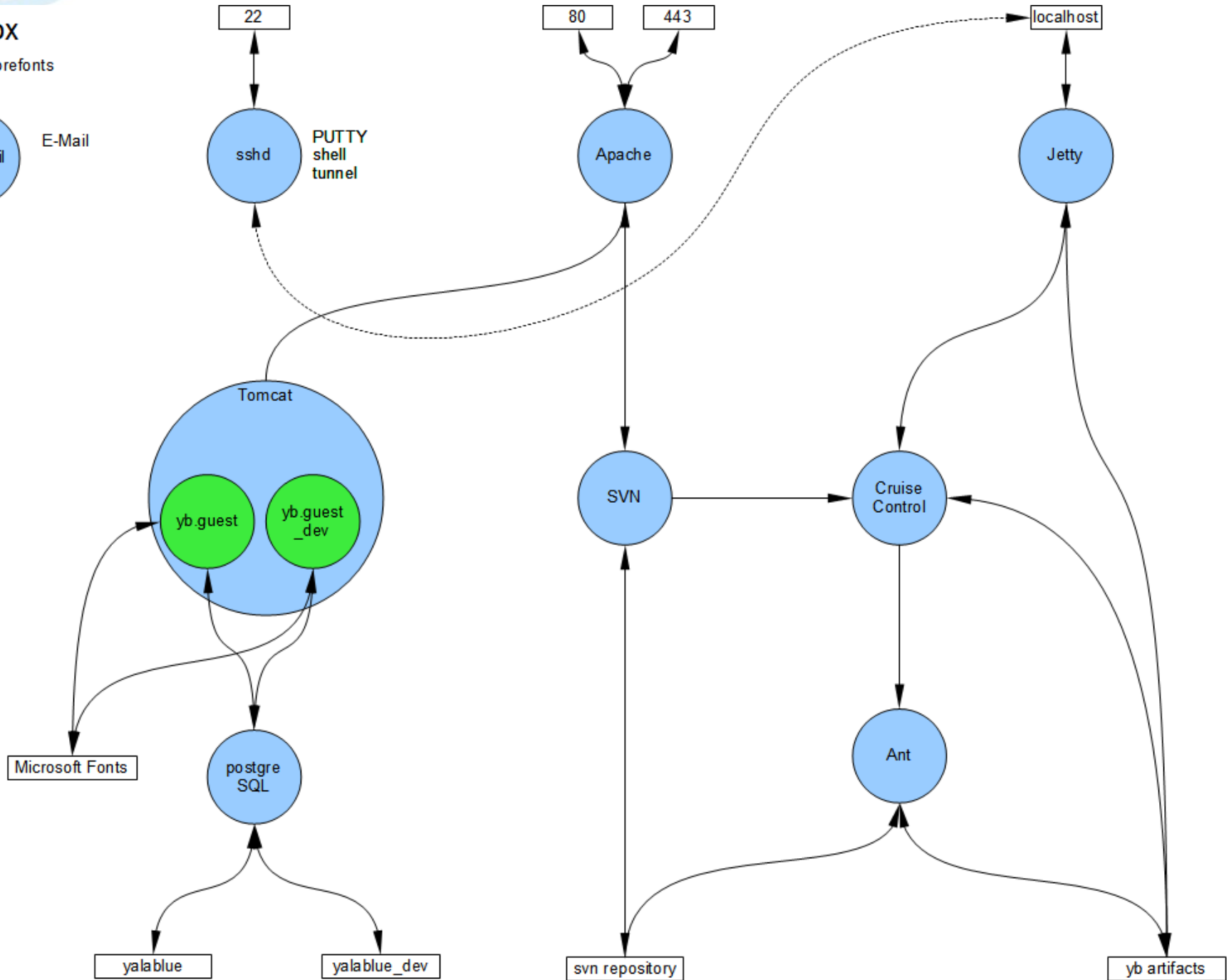


UNIX Box

Microsoft Corefonts
jdk x.y

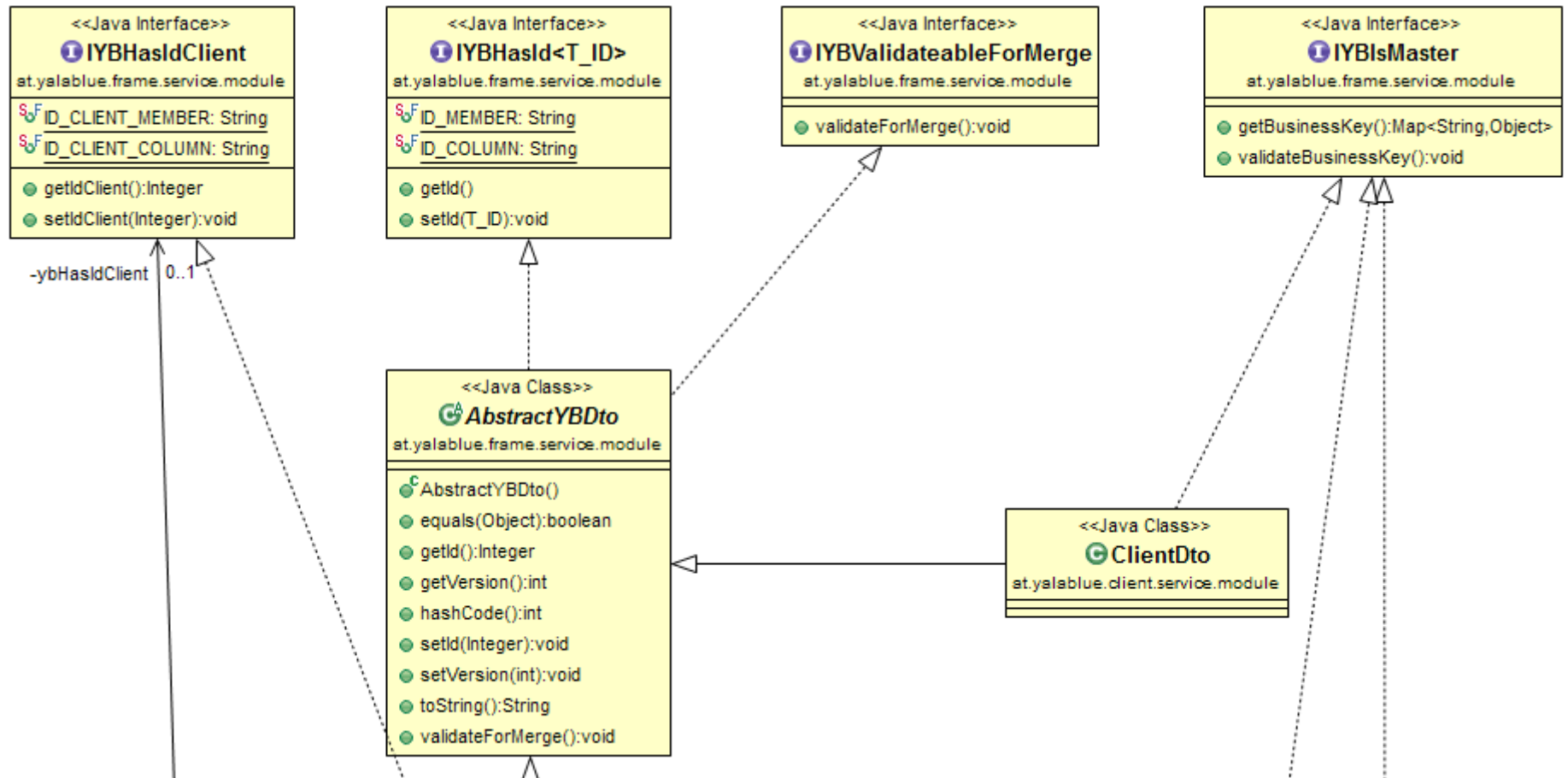


E-Mail



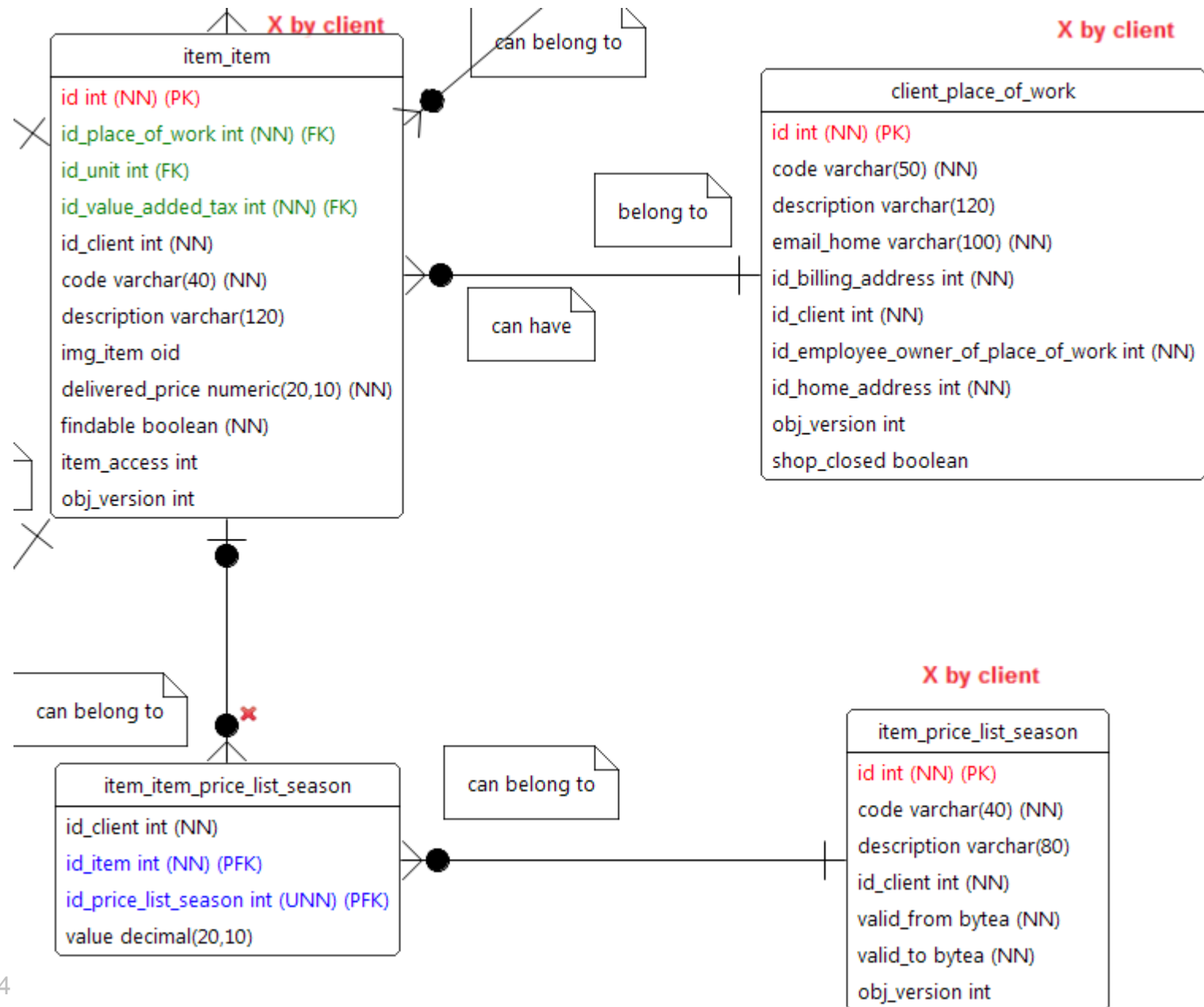


UML Class diagram



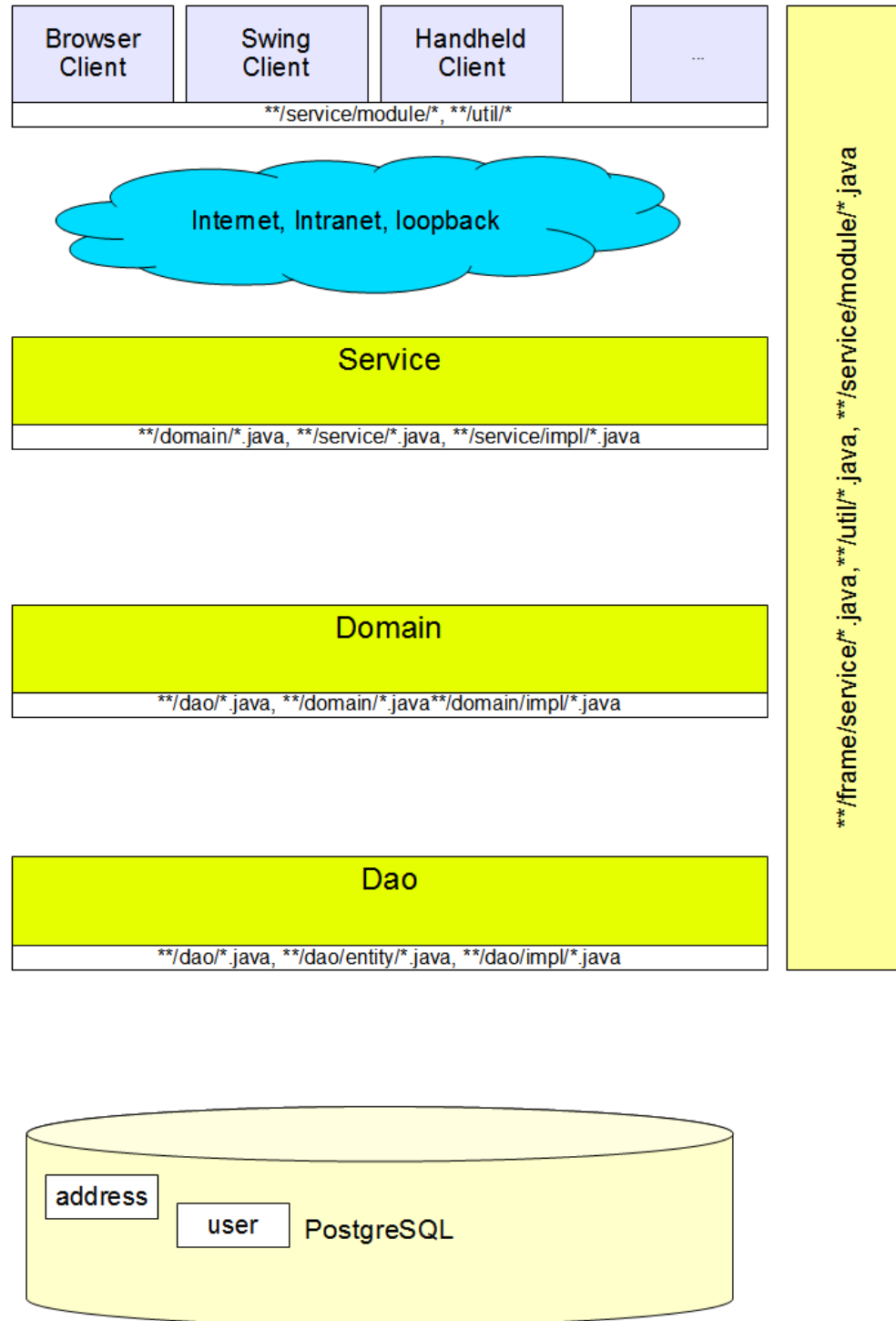


ER diagram



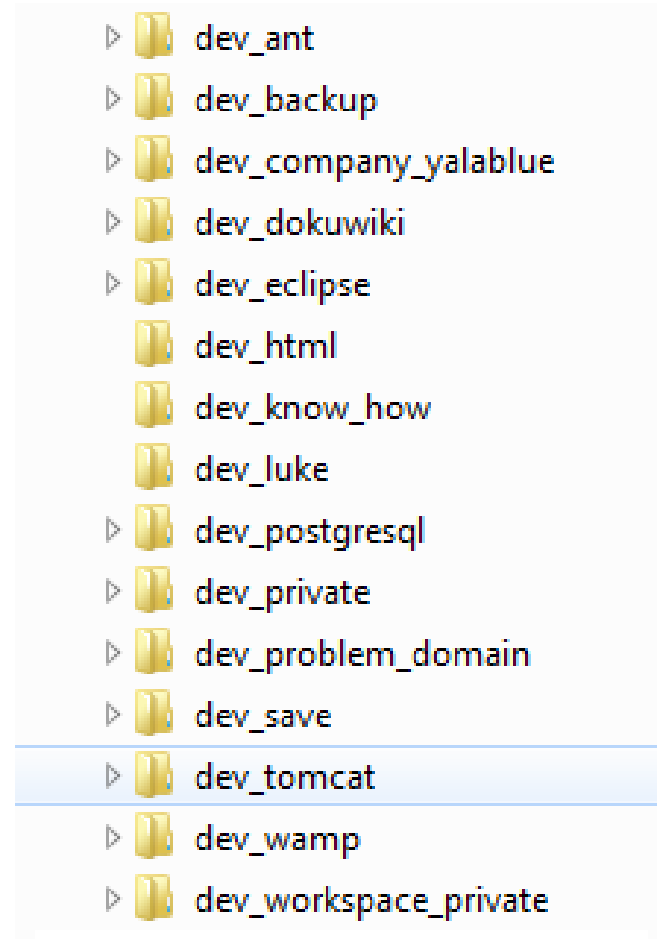
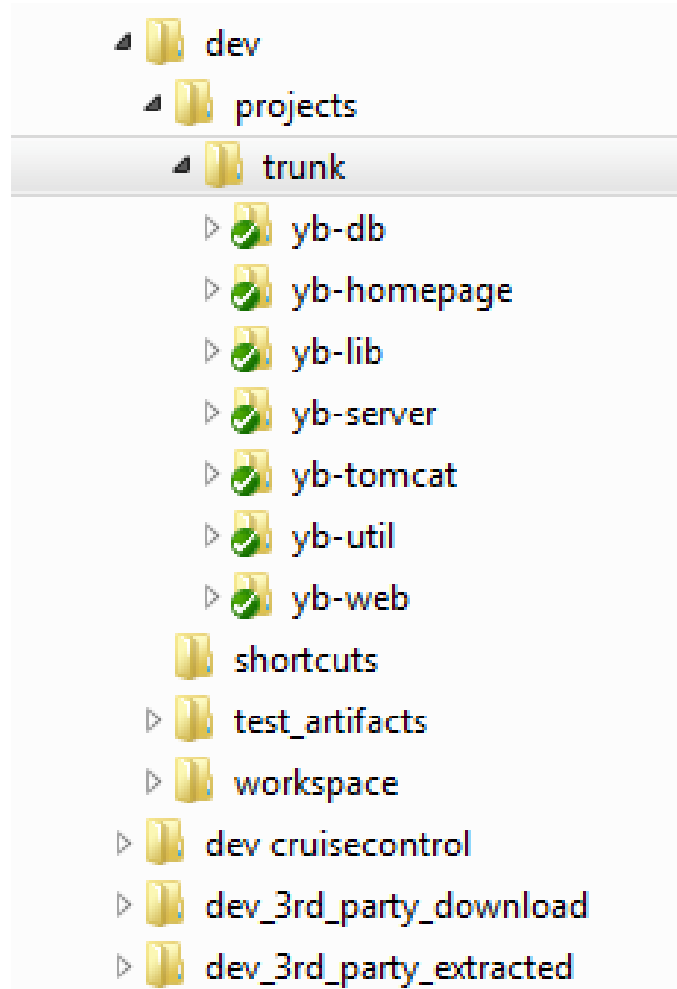


Tiers





Directories





Build

- ➔ Build und Test laufen schichtweise ab; bottom up
- ➔ Cruisecontrol steuert; Ant 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



Big ant tasks

⇒ Server ant tasks

- new db
- delivery db
- base
- checkin
- mass db
- all methods

⇒ Client ant tasks

- checkin
- screenplay
- massdb



Dokumentation

- ⇒ JavaDoc nutzen
- ⇒ Nahe am Code
- ⇒ Automatisch generieren
- ⇒ Sie lesen viel Code
- ⇒ Sie finden einen alten Dokumentationsfriedhof
- ⇒ Unwartbar
- ⇒ Zuviel Aufwand
- ⇒ ...



Das Mobile Web

➔ Unterschiede:

- Design für kleine Auflösungen, langsame Downloadraten, geringe Akkulaufzeit, ...
- Verwendung in einem anderen Kontext
- Der integrierte Browser des Smartphones ist nicht ausreichend

➔ Beispiel: <https://office.yalablue.com>

- Eintippen der Adresse → kurze URL
- Probleme mit dem Zertifikat -> große Vorsicht geboten
- Man kann auf Anhieb gar nichts lesen
- Man muss vergrößern und hinscrollen

➔ Redesign unter Berücksichtigung der Zielgruppe



Die Mobile App

- ➔ Native Betriebssystem für Smartphones und Tablets
 - Apple iPhone, iPad: iOS
 - Google Android: Android
 - Windows Phone, Blackberry, Palm, ...

- ➔ Sie brauchen eine mobile Variante Ihrer Website
 - HTML5 / CSS3
 - Native App
 - Hybride App

- ➔ „The mobile web is a moving target.“



jQuery Mobile

- ➔ „Touch Optimized Web-Framework“ für Smartphones & Tablets
- ➔ <http://jquerymobile.com>, Release 1.4.2
- ➔ Basierend auf jQuery bietet jQuery Mobile ein HTML5-basierte UI-System für alle großen mobilen Betriebssysteme
- ➔ Verwendet HTML5, CSS3, Ajax
- ➔ Unterstützte Plattformen: <http://jquerymobile.com/gbs/>
- ➔ Simuliert natives Geräteverhalten



Die Entwicklung der Mobile App

- ➔ Responsive Design
- ➔ HTML5/CSS3, native Mobile App oder hybrid App?
- ➔ Welche Betriebssysteme werden unterstützt?
- ➔ Wie wird getestet?
 - Endgeräte
 - Emulatoren
 - Spezielle Testplattformen, z.B. Device Anywhere
- ➔ Publishing: Signieren und Einstellen der Mobile App



Native App Beispiel: Windows Phone Toolkit

```
<?xml version="1.0" encoding="utf-8"?>
<Deployment xmlns="http://schemas.microsoft.com/windowsphone/2009/deployment"
  <App xmlns="" ProductID="{d31dcd2b-b1f2-4cf6-9354-579eb5b1d481}" Title="\
  <IconPath IsRelative="true" IsResource="false">ic_launcher-web.png</IconPath>
  <Capabilities>
    <Capability Name="ID_CAP_IDENTITY_DEVICE" />
    <Capability Name="ID_CAP_IDENTITY_USER" />
    <Capability Name="ID_CAP_LOCATION" />
    <Capability Name="ID_CAP_NETWORKING" />
    <Capability Name="ID_CAP_WEBBROWSERCOMPONENT" />
    <Capability Name="ID_CAP_APPOINTMENTS" />
    <Capability Name="ID_CAP_CONTACTS" />
    <Capability Name="ID_CAP_ISV_CAMERA" />
    <Capability Name="ID_CAP_MEDIALIB" />
```

- ➔ Windows 8, WindowsPhone registrieren, Microsoft Account.
- ➔ Build tool: MS-Build.
- ➔ Externes XAP Sign Tool.



Fragen

- ➔ 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?...



Noch mehr Fragen

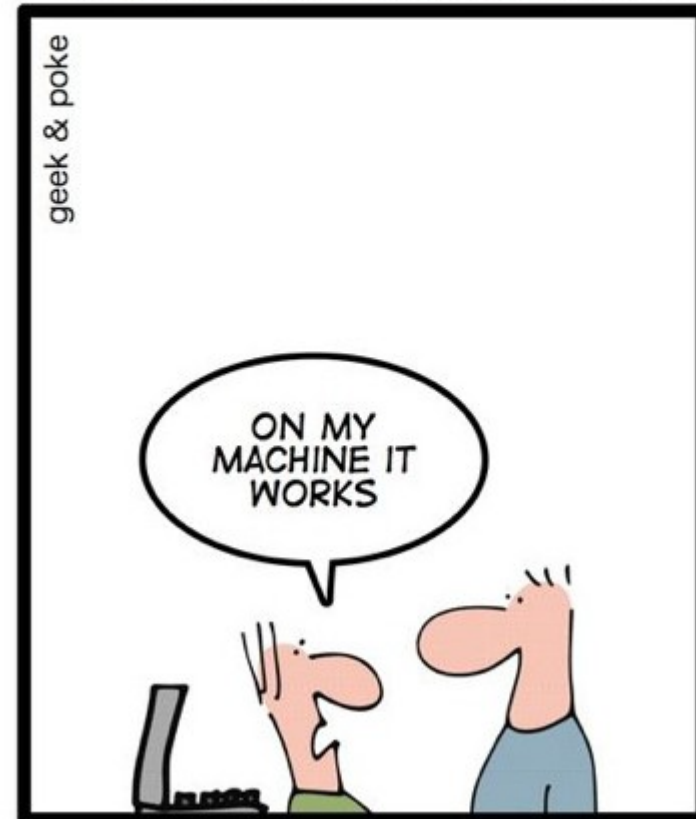
- ➔ Ist der Code noch refactorfähig?
- ➔ Kann ich sorgenfrei Bugfixen, einen Feature Request abarbeiten?
- ➔ Habe ich überwiegend Spass an der Arbeit?
- ➔ Gibt es einen letzten Breakpoint?
- ➔ Wie finde ich mich in der Packagestruktur zurecht?



Ende

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

WAIT UNTIL YOU HEAR THIS:



➔ Fragen?