



## Beraterprofil

### Dr. Marc Kronberg

|                                |   |
|--------------------------------|---|
| <b>Ausbildung:</b>             | Dr. rer. nat., Diplom Chemiker  |
| <b>Geburtsjahr:</b>            | 1969  |
| <b>EDV – Erfahrung seit:</b>   | 1989  |
| <b>Nationalität:</b>           | deutsch   |
| <b>Fremdsprachen:</b>          | englisch  |
| <b>Fachliche Schwerpunkte:</b> | Berater und Anwendungsentwickler (Architekt & Senior-IT-Consultant)<br>Spezialist für <ul style="list-style-type: none"><li>▪ Frontend (Angular 2-6, IONIC, Google Polymer, Java Swing)</li><li>▪ Performancetuning, Refactoring</li><li>▪ Java EE, Node.js</li></ul> |
| <b>Branchen:</b>               | Banken, Telekommunikation,<br>Transport- und Verkehrswesen (Bahn)   |
| <b>Kontakt</b>                 | <b>Marc.Kronberg@krocon.de</b><br><b>0151 2012 9999</b>   |
| <b>Verfügbarkeit ab:</b>       | Nach Absprache, Vollzeit, langfristig   |

# Flyer

Ich biete...

- IT-Beratung (Architekt und Senior-IT-Consultant), Mitarbeit in Web- und JEE-Projekten
- Durchführung innovativer und anspruchsvoller Beratungsprojekte für die Themenfelder Banking und Telekommunikation
- Analyse, Design und Programmierung neuer Anwendungen
- Weiterentwicklung & Refactoring einer bestehenden Anwendung (Performance Tuning)
- Gutes Auge für Layout
- Einen sehr guten technologischen Überblick in Bereichen der Web- und Richclient-Entwicklung
- Erfahrungen in der (multikultureller) Teamarbeit (Teamgrößen von 2 - 1000)

Über mich

- IT-Architekt & Macher (selbständige, strukturierte und engagierte Arbeitsweise)
- Frontendspezialist: Webapplications (Progressive Web Apps, Responsive Websites, sowohl Java- als auch JavaScript/TypeScript-basierte Frameworks, Angular, Google Polymer, HTML5, JQuery, ES06) & Java/Swing
- Der Schwerpunkt meiner Tätigkeit liegt in der Softwareentwicklung mit Javascript/TypeScript (Angular 2-6, Google Polymer, Node.js) und auf Basis der Java Plattform (Java EE)
- Ich habe Spaß an Usability und kann mich für 'State of the Art'-Technologien begeistern

Ich bin ein Freund...

- der Bücher "Der Pragmatische Programmierer" und "Clean Code"
- von schlanken Software-Architekturen, wie z. B. Domain-Driven Design (Hibernate Annotations)
- Microservices

Ich mache...

- Intuitives Design ("Dont make me think!")
- in kurzer Zeit funktionale Prototypen
- robusten Code, der auch nach Jahren von Fremden übernommen werden kann

Ich beachte...

- DRY (JSP-Tags, Templates, Generatoren)
- die üblichen Software design patterns (Robuster Code: Wiederverwendbarkeit & Performance)
- die (genauso wichtigen) Anti-Patterns

Ich mag...

- schlanke Vorgehensmodelle (wenig Formalismus), z. B. Scrum, xP, Test-Driven-Development
- agile Projekte mit kurzen Iterationsschleifen
- flache Hierarchien und Feedback

## Projekthistorie

|                             |  |
|-----------------------------|--|
| <b>07/2018 - heute</b>      | <b>selbst</b>  |
| <b>1 Person</b>             | <b>Cryptocurrency Exchange</b>   |
| <b>Projektbeschreibung:</b> | Crypto X besteht aus Bausteinen für den Bau von Cryptocurrency-Börsen für P2P (analog bitcoin.de) und B2P (analog bitpanda.com). Das Backend basiert auf einer Microservice-Architektur (Dockercontainer), es gibt sowohl einen Tomcat mit Spring Boot & Hibernate (JPA), sowie einen nodejs-Server. Datenbank: PostgreSQL. Das Frontend wird mit IONIC entwickelt und ist sowohl als App für iOS und Android verfügbar. |
| <b>Tätigkeit:</b>           | Architektur und Entwicklung  |
| <b>Projektergebnis:</b>     | a) Laufendes Projekt<br>b) 12.04.2018: 60-minütiger Vortrag bei der Deutschen Bank (DB Campus, Frankfurt) 'Kryptowährungen: Einführung und Ausblick'.<br>c) 22.08.2018: 90-minütiger Vortrag 'Kryptowährungen: Einführung und Ausblick' bei Helaba Invest Frankfurt  |
| <b>Realisiert unter:</b>    | Ionic 4, Angular 6 and Cordova; Rust 1.27.0, Java 10, Spring Boot 2.1.0, PostgreSQL 10.4, Hibernate 5.3, JPA; git 2.16   |

|                             |  |
|-----------------------------|--|
| <b>08/2016 - heute</b>      | <b>Helaba Invest Kapitalanlagegesellschaft mbH</b>   |
| <b>3 Personen</b>           | <b>Phoenix</b>   |
| <b>Projektbeschreibung:</b> | Implementierung eines Web-Portals für ein elektronisches Reporting (mandantenfähig, Internationalisierung): in einem schnellen Entwicklungsprozess wurde mit Hilfe von Prototypen und kurzen Iterationsphasen die GUI gegen eine Mock-Serviceschicht entwickelt. Dabei wurde mittels SASS (Bootstrap) ein styleguide-konformes CSS-Gerüst erstellt. Die Usability wurde durch schnelles und häufiges Kundenfeedback optimiert. |
| <b>Tätigkeit:</b>           | Architekt/Entwickler:<br>Design der Oberfläche, Auswahl der eingesetzten technischen Komponenten und gestalten des Anwender-Workflows (Usability), Implementierung der Frontend-Schicht.   |
| <b>Projektergebnis:</b>     | Frontend wurde im geplanten Zeitplan (produktionsreif) fertiggestellt. Derzeit (März, April 2017) ersetzen kundeninterne Entwickler die einzelnen Mock-Zugriffe durch 'echte' Datenbankzugriffe. Das Ausrollen in die Produktion steht kurz bevor.   |
| <b>Realisiert unter:</b>    | Angular 2.rc2 - 4.1.3 (TypeScript, WebComponents, Routing, RxJS / Observables, i18n, Drag&Drop), Bootstrap 4.0.0.alpha (SASS), Angular CLI   |

**01/2017 - 06/2018**

**Deutsche Bank (Frankfurt)**

**5 Personen**

**TradeFinder Portals**

**Projektbeschreibung:**

Fortsetzung des Projektes "autobahn TradeFinder" vom 08.2013 - 12.2016. Idee/Design und Implementierung eines Portals auf Basis der neuen "Brand Identity Guidelines for Responsive Webdesign" der Deutschen Bank. Diverse fachliche Erweiterungen bestehender Dialoge im Bereich „Debt Strats“. Bau neuer Funktionalitäten für MiFID II.

**Tätigkeit:**

Entwickler:  
Design der Oberfläche, Auswahl der eingesetzten technischen Komponenten und gestalten des Anwender-Workflows (Usability), Implementierung der Frontend-Schicht.

**Projektergebnis:**

Laufendes Projekt. Diverse Änderungen in Produktion gebracht.

**Realisiert unter:**

Java 1.7, (Plain) Javascript, nvd3, d3.js, Bootstrap 3, CSS (less), Ajax, JSON, node.js, gulp, Atlassian: Jira (Projektmanagementsystem, Issue-Tracker) + FishEye (Repository Browser) + Crucible (CodeReview), Confluence (Wiki), Perforce, Jenkins

|                             |   |
|-----------------------------|---|
| <b>08/2013 - heute</b>      | <b>Deutsche Bank (Frankfurt)</b>  |
| <b>40 Personen</b>          | <b>Autobahn TradeFinder 3</b>   |
| <b>Projektbeschreibung:</b> | <p>autobahn TradeFinder ist ein Teil der autobahn-Produktpalette der Deutschen Bank. Tradefinder erlaubt sowohl bankinternen Tradern als auch externen Kunden schnelle Berechnungen auf der Basis von historischen und real-time Daten durchzuführen.</p> <p>Fortsetzung des Projektes vom 05.2009 - 12.2012 nah am Kunden (Trading Floor).</p> <p>Implementierung neuer Funktionalitäten mit angular.js und bootstrap.</p> <p>Änderung des Designs (autobahn-Style light und dark) auf Basis von Bootstrap 3.</p> <p>Neues Portal für Pricer-Frontends (db-X Pricer) im fullscreen-Design.</p> |
| <b>Tätigkeit:</b>           | <p>Entwicklung:</p> <p>Für das Portal autobahn TradeFinder wurden diverse Änderungen in Produktion gebracht. Eine neue Frontend-Architektur, basierend auf angular.js mit neuen Direktiven (Datepicker mit Börsenkalender, einheitliche Fehlermeldungen usw.) vereinfacht die stete Weiterentwicklung des Projektes. Neues Portal "db-X Pricer" wurde geschaffen.</p>   |
| <b>Projektergebnis:</b>     | <p>Laufendes Projekt. Viele neue Funktionalitäten wurden erfolgreich in Produktion gebracht (Bulk-Eingabe, Eingabe durch intelligentes Parsen von Text, Eingabe über Wizards (Strategien), generischer Ergebnis-Renderer mit diversen Grafiken, usw.). Das Design der Webanwendung wurde erneuert (volle Bildschirmbreite, responsiv Design). Diverse neue Schnittstellen (Booking, Execution) wurden implementiert.</p>  |
| <b>Realisiert unter:</b>    | <p>Java 1.7, Javascript, angular.js (eigene Direktiven), nvd3, d3.js, Bootstrap 3, CSS (less), Ajax, JSON, node.js, gulp, Atlassian: Jira (Projektmanagementsystem, Issue-Tracker) + FishEye (Repository Browser) + Crucible (CodeReview), Confluence (Wiki), Perforce, Jenkins</p>   |

|                             |   |
|-----------------------------|---|
| <b>2016</b>                 | <b>Open source</b>  |
| <b>1 Person</b>             | <b><a href="https://www.npmjs.com/~krocon">https://www.npmjs.com/~krocon</a></b>                                |
| <b>Projektbeschreibung:</b> | Diverse kleinere Lernprojekte, siehe: <a href="https://www.npmjs.com/~krocon">https://www.npmjs.com/~krocon</a> |
| <b>Tätigkeit:</b>           | Entwicklung   |
| <b>Projektergebnis:</b>     | Siehe: <a href="https://www.npmjs.com/~krocon">https://www.npmjs.com/~krocon</a>                                |
| <b>Realisiert unter:</b>    | AngularJS, Google Polymer, ES06, (Bootstrap 4 via SASS), node.js (socket.io), gulp (usemin, webpack, babel)     |

|                             |  |
|-----------------------------|--|
| <b>02/2013 - 07/2013</b>    | <b>Acarda</b>  |
| <b>5 Personen</b>           | <b>PMO Dashboard</b>   |
| <b>Projektbeschreibung:</b> | Weiterentwicklung und Refactoring eines Reporting-Portals. Aus diversen Portalimplementierungen wurde ein ‚Produkt‘ erstellt, welches durch Konfiguration für Endkunden anpassbar ist. |
| <b>Tätigkeit:</b>           | Entwicklung  |
| <b>Projektergebnis:</b>     | Neues Produkt mit neuen Funktionalitäten (beim Endkunden im Einsatz).  |
| <b>Realisiert unter:</b>    | DHTMLX (dhtmlx.com), JDBC, Hibernat, Spring, Java 1.7, JAXB, Javascript, ANT   |

|                             |   |
|-----------------------------|---|
| <b>01/2013 - 01/2013</b>    | <b>Selbst</b>   |
| <b>1 Person</b>             | <b>Editlicious</b>  |
| <b>Projektbeschreibung:</b> | <p>Editlicious ist eine Machbarkeitsstudie für einen auf Java-Swing basierenden Texteditor.</p> <p>Es wurde ein lauffähiger (ausbaufähiger) Prototyp auf solider Codebasis entwickelt.</p> <p>Bei diesem sowohl performanten als auch anwenderfreundlichem Texteditor sind diverse interessante Techniken (Multithreading usw.) und Tricks eingesetzt bzw. beachtet worden.</p> |
| <b>Tätigkeit:</b>           | Entwicklung   |
| <b>Projektergebnis:</b>     | Lauffähiger (ausbaufähiger) Prototyp, siehe editlicio.us.   |
| <b>Realisiert unter:</b>    | IntelliJ, Java 1.7, JAXB, Javascript, ANT   |

**05/2009 - 12/2012**

**Deutsche Bank (Frankfurt)**

**20 Personen**

**TradeFinder 2**

**Projektbeschreibung:**

autobahn TradeFinder ist ein Teil der autobahn-Produktpalette der Deutschen Bank. Tradefinder erlaubt sowohl bankinternen Tradern als auch externen Kunden schnelle Berechnungen auf der Basis von historischen und real-time Daten durchzuführen.

Fortsetzung des Projektes vom 06.2007 - 09.2008 nah am Kunden (Trading Floor).

Design und Implementierung neuer Funktionalitäten (iGoogle-artige Frontpage, regelbasierte Pricer, Reporting für Endkunden).

Änderung des kompletten Designs (nach neuem DB-Style-Guide).

Migration von prototype auf jQuery.

Performancesteigerung durch (kontinuierliches) Refactoring (vereinfachte einheitliche CSS, simpler und robuster HTML-Code, Optimierung von Javascript, usw).

Erstellung von Vorgaben für Entwickler.

**Tätigkeit:**

Entwicklung

**Projektergebnis:**

Laufendes Projekt. Viele neue Funktionalitäten wurden erfolgreich in Produktion gebracht.

**Realisiert unter:**

IntelliJ, Eclipse, MatLab (Distributed Computing Server), Java 1.6, JSP (JavaServer-Pages), Custom-Tags und Tag-Files, Javascript, jQuery, Prototype, script.aculo.us, HTML, CSS, Ajax, JSON, JDBC, myBatis und Hibernate, Apache FOP (PDF-Generierung), Xalan, Xerces

**11/2008 - 04/2009**

**DZ Bank (Frankfurt)**

**2 Personen**

**AlgoTrading ApamaGUI**

**Projektbeschreibung:**

Implementierung eines Trading-Frontends für Progress Apama (Algorithmic Trading).

Deutliche Steigerung der Effizienz beim Handel durch die Bündelung der Funktionalitäten dreier Trading-Frontends (Xetra, Eurex und Progress Apama) in einem (vereinheitlichten) GUI.

Durch Händler leicht anpassbare Strategien (Strategien: Trailing Stop, Hidden Limit, Flexible Iceberg, Pegging, Xetra / Eurex Standard-Ordertypes).

**Tätigkeit:**

Entwicklung

Komplette (Neu-)Entwicklung einer Swing-GUI (inkl. Business-Schicht) gegen einen Apama-Server für Multimarket-Algo-Trading (algorithmischer Handel).

**Projektergebnis:**

Produktiv seit 05.2009

**Realisiert unter:**

IntelliJ, Apama-Server, Java 1.6, Swing, JAXP



06/2007 - 09/2008

Deutsche Bank (Frankfurt)

15 Personen

TradeFinder

**Projektbeschreibung:**

autobahn TradeFinder ist ein Teil der autobahn-Produktpalette der Deutschen Bank. Tradefinder erlaubt sowohl bankinternen Tradern als auch externen Kunden schnelle Berechnungen auf der Basis von historischen und real-time Daten durchzuführen.

Cross Market Analyse, Trade Scanning und Portfolioanalyse sind in Tradefinder die Schlüsselfunktionalitäten, die dem Kunden ein schnelles Erkennen von Anomalien im Markt oder im Portfolio erlauben (Market Overview, Trade Scanning, Portfolio Management and Optimization).

(Siehe: <http://globalmarkets.db.com/resources/tradefinder.pdf> oder auch <http://tradefinder.db.com>)

**Tätigkeit:**

Entwicklung

**Projektergebnis:**

Für die sich seit 2005 im produktivem Einsatz befindliche Software wurden diverse neue Funktionen ausgerollt.

Neben dem Bau neuer Funktionalitäten wurden große Teile der Webserver-Browser-Kommunikation auf Ajax (JSON) umgestellt.

Diese führte zu einem schnelleren Laden der Seite im Browser und zu einem besseren Feedback für den Anwender bei langlaufenden Server-Operationen.

Die Anwendung wurde (sanft) refactored und die Frontendschicht von Businesslogik und Persistenzcode befreit. Es wurde eine Businessschicht und eine Persistenzschicht eingeführt.

Die Frontendschicht wurde durch einige Maßnahmen (Einsatz von CSS, Eliminieren von Codedoubletten, Minimierung von Javascript-Files usw.) wartbarer und performanter.

Für das Einlesen von Daten eines Fremdsystems wurde ein hochperformanter XML-Parser entwickelt.

**Realisiert unter:**

IntelliJ, Eclipse, MatLab, Java 1.5, JSP, Javascript, Prototype, script.aculo.us, Ajax, JSON, JDBC und Hibernate, Apache FOP (PDF-Generierung), Xalan, Xerces

08/2006 - 04/2007

T-Online International AG Darmstadt

5 Personen

### ActivePartner (APP)

#### Projektbeschreibung:

Das Programm ActivePartner (APP) ermöglicht den Vertrieb von T-Online und T-Com Produkten (Tarife und Hardware-Bundles) durch Privatpersonen, kleine Gewerbetreibende und Website-Betreiber (Vertriebspartner) und erschließt dadurch eine neue, externe Salesforce.

Es handelt sich bei diesem Affiliateprogramm (Partnerprogramm) um eine internetbasierte Vertriebslösung, bei der der Erfolg der Vertriebspartner durch eine Provision vergütet wird. Über das Programm werden Werbemittel (Banner und Printmedien) zur Verfügung gestellt, welche der Affiliate auf seinen Seiten zur Bewerbung der Produkte verwenden kann. Die Status seiner Aufträge sind dem Vertriebspartner über das Haupt-Portal ([www.activepartner.de](http://www.activepartner.de)) ersichtlich.

Der Verkauf durch die Vertriebspartner wird über einen mandantenfähigen Internet-Shop abgewickelt. Für die Pflege von Stammdaten, Werbemittel, Produkten, usw dient den Mitarbeitern von Customer Care das Portal CSR.

(Stichworte: Webanwendung, Onlineshop, Administrationstool, Auftragserfassung, Prämiensystem, Refactoring der GUI, starker Einsatz von CSS).

#### Tätigkeit:

Analyse (Lastenhefte), Konzeption (Pflichtenhefte), Entwicklung

Für das Hauptportal (IFrame-Architektur) wurden diverse neue Frames entwickelt. Für das nach einem Rollenkonzept gebaute CSR-Portal wurde die Frontend-Schicht (Struts, tiles) refactored. Die bestehende Architektur wurde weitestgehend beibehalten, jedoch wurde durch den massiven Einsatz von CSS das Layout von den JSPs getrennt.

Für die (häufig vorkommende) Darstellung von Tabellen wurde eine neue Tabellenkomponente (nach dem MVC-Pattern) entwickelt, die ein Paging-Mechanismus bietet und das Sortieren der Tabellen (über das Model) unterstützt.

Die Fachabteilung wurde beim Erstellen der Lastenhefte unterstützt und es wurden neue Pflichtenhefte erstellt.

Diverse neue Lieferstufen wurden entwickelt (Schwerpunkt Frontend) und in Produktion gebracht.

#### Projektergebnis:

Für die sich seit 2005 im produktivem Einsatz befindliche Software wurden diverse neue Funktionen ausgerollt.

#### Realisiert unter:

JSP & Struts 1.2 (Tiles), Java 1.4 , Bea WebLogic 8.1, Datenbank: Oracle

09/2005 - 07/2006

## DB-Systems

10 Personen

### FRED (Fahrzeug-Ressourcen-Einsatzplanung Disposition)

**Projektbeschreibung:**

FRED (Fahrzeug-Ressourcen-Einsatzplanung und -Disposition) ist ein umfassendes Informationssystem zur Unterstützung der mittelfristigen Planung und der zeitnahen Disposition des Einsatzes von Fahrzeugressourcen. Über Änderungen in der Einsatzplanung von Triebfahrzeugen und Reisezugwagen informiert FRED alle Beteiligten durch automatische Benachrichtigungen. Plan- und Ist-Zustände werden von FRED dokumentiert.

Die individuelle Bereitstellung der Informationen für jeden Anwender erfolgt über eine grafische Benutzeroberfläche. Die Einsatzplanung von Fahrzeugen wird über einen wählbaren Zeitraum hinweg als Gantt-Diagramm dargestellt, das es dem Anwender zudem gestattet, Planungsänderungen per Mausklick (Drag and Drop) durchzuführen. Die grafische Anzeige von Leistungen ausgewählter Laufpläne unterstützt zusätzlich die Arbeit des Planers.

Technischer Aufbau: mehrschichtiges Anwendungssystem, bestehend aus Rich-Client (Swing), einem J2EE-Applikationsserver und einem Host-basierten Backendsystem.

Als Backendsystem wird das bestehende Verfahren DIF verwendet, dessen Module über spezielle Host-Schnittstellen angesteuert werden. Der wesentliche Bestandteil der Geschäftslogik von FRED ist in diesen Modulen abgebildet. Ein weiterer Teil der Geschäftslogik ist in Komponenten innerhalb des J2EE-Applikationsservers implementiert, die gleichzeitig die Schnittstelle zum Client bilden. Der Rich-Client ist eine Java-Swing-Applikation, die dem Benutzer eine höchstmögliche Interaktion mit dem Anwendungssystem ermöglicht.

Seit Januar 2004 befindet sich das System im produktiven Einsatz.

**Tätigkeit:**

Programmierung Swing: Änderungen am bestehenden Code (Bugfixing, kleines Refactoring von Dialogen) und Bau neuer Funktionalitäten im Bereich GUI

**Projektergebnis:**

Für die sich seit Januar 2004 im produktivem Einsatz befindliche Software wurde eine neue Version ausgerollt.

**Realisiert unter:**

JSP & Struts 1.1 (Tiles), Java 1.4 , Bea WebLogic 7.0 SP2 / 8.1, Datenbank: Oracle 9.2.0.5

**04/2005 - 08/2005**

**T-Online International AG Darmstadt**

**30 Personen**

**Ordermanagementsystem (OMS)**

**Projektbeschreibung:**

Das Projekt OMS ist eine 3-Tier Webarchitektur (J2EE) zur Erfassung und Bearbeitung von Kundenaufträgen unter Verwendung von Oracle und BEA-Technologie.

Implementierung von Kundenanforderungen im Frontend.

Umfangreiches Refactoring der Frontend-Schicht (JSP, Struts).

Definieren von Programmierstandards.

**Projektergebnis:**

In 6 Monaten wurden zwei neue Versionen in Produktion gebracht.

**Realisiert unter:**

JSP & Struts 1.1 (Tiles), Java 1.4 , Bea WebLogic 7.0 SP2 / 8.1, Datenbank: Oracle 9.2.0.5

**04/2004 - 12/2004**

**T-Online International AG Darmstadt**

**15 Personen**

**Produktreferenzsystem**

**Projektbeschreibung:**

Mehrschichtiges Intranet-Portal auf der Basis von Java Server Pages (JSP) und Struts (&Tiles) für die Pflege von Stammdaten eines Ordersystems.

Design von diversen Funktionalitäten nach Absprache mit dem Endkunden.

Implementierung großer Teile des Frontends.

Reengineering der bestehenden Software.

Design und Implementierung von Architekturkomponenten (Strutskonformes Menü, Tabellen-Tag mit Table-Modell und konfigurierbaren Renderern, sowie diverse Tags).

Unterstützung des Testteams.

**Projektergebnis:**

In 8 Monaten wurden drei Versionen entwickelt, die sich im produktiven Einsatz befinden

**Realisiert unter:**

JSP & Struts 1.1 (Tiles), Java 1.4 , Tomcat 4.1, Apache Avalon (Excalibur, Merlin), Hibernate 2, Datenbank: Oracle 9i

|                             |  |
|-----------------------------|--|
| <b>02/2004 - 03/2004</b>    | <b>In house</b>  |
| <b>1 Person</b>             | <b>Web-Portal SearchMe</b>   |
| <b>Projektbeschreibung:</b> | Design und Implementierung von Portal-Funktionalitäten.  |
| <b>Projektergebnis:</b>     | Web-Portal auf der Basis von JSP, Struts, Tiles.<br><br>Folgende Funktionalitäten wurden erfolgreich umgesetzt: Login-Dialog mit eMail-Verifikation, internes Mailsystem, Metasuche von Kontaktanzeigen und das Anzeigen von News, die via RSS geliefert werden. |
| <b>Realisiert unter:</b>    | Java (Struts & Tiles und JDBC, Log4J), Tomcat 4.1, Datenbank: MySQL, Build-Tool ANT  |

|                             |   |
|-----------------------------|---|
| <b>02/2003 - 01/2004</b>    | <b>DIT (Fondsgesellschaft)</b>  |
| <b>3 Personen</b>           | <b>Produktdatenbank 2</b>   |
| <b>Projektbeschreibung:</b> | Siehe Projekt 11/01 - 12/01.<br><br>Die Produktdatenbank (PDB) ist ein Big-Client/Server-System für Investmentfonds inkl. Workflow, Historisierung und Vieraugenprinzip, Trennung von Business Logik und Datenspeicherung durch Schichtenmodell mit abstrakter Persistenzschicht. |
| <b>Projektergebnis:</b>     | Die neue Version 1.3.0 wird die bereits in Produktion befindliche Vorgängerversion ersetzen. Neue Features sind Reporting via Pdf, sowie XML-Schnittstellen.  |
| <b>Realisiert unter:</b>    | XML, XSD, XSL, Java (Swing und JDBC, Log4J), Datenbank: Oracle 9i, Build-Tool ANT, XML-Spy  |

|                             |  |
|-----------------------------|--|
| <b>12/2002 - 01/2003</b>    | <b>In house</b>  |
| <b>1 Person</b>             | <b>Dito (Database Investigation Tool)</b>  |
| <b>Projektbeschreibung:</b> | Dito ist eine GUI zur Visualisierung und Manipulation von JDBC-fähigen Datenbanken wie z.B. Oracle und DB2. Es besteht aus den drei Hauptmodulen Scheme-Browser, SQL-Window und Reengineering.<br><br>Technisch basiert das Tool auf einer dreischichtigen Architektur (GUI-, Business- und Datenbankschicht), welche sich bereits in anderen Projekten als vorteilhaft erwies. Der Datenaustausch zwischen den Schichten wurde über Businessobjekte realisiert.<br><br>Technische Features sind: Multithreading, konfigurierbarer Window-Manager. |
| <b>Projektergebnis:</b>     | Veröffentlichung als Betaversion   |
| <b>Realisiert unter:</b>    | Java (Swing und JDBC), Datenbank: Oracle 9i, DB2   |

11/2002 - 11/2002

DIT (Fondsgesellschaft)

1 Person

Global Search (als Teilprojekt vom XyzBlotter)

**Projektbeschreibung:**

Parallel zum Projekt DIT-Blotter beim selben Kunden.

Global Search ist eine visuelle Java-Komponente, mit welcher der Endanwender Datenbankabfragen für ein bestimmtes Datenbankschema zur Laufzeit selber gestalten und absetzen kann (Query-Builder).

Um gegenüber Datenbankänderungen möglichst flexibel zu sein, wurde die Komponente in einer dreischichtigen Architektur (GUI-, Business- und Datenbankschicht) entwickelt. Der Informationsaustausch zu Vater-Applikation wurde über Events und Businessobjekte realisiert.

Obwohl die Komponente lokalisierbar (Umstellung der Sprache zur Laufzeit) und stark konfigurierbar ist, ist ihre Integration in bestehende Produkte sehr einfach. Weitere Features sind: Sortierbare Tabellen, mehrzeilige Tooltips als Vorschau auf Detailinformationen, intelligentes Verhalten bezüglich der Ergebnismenge.

**Projektergebnis:**

Die Komponente wurde gemäß Pflichtenheft und weiteren Kundenvorgaben entwickelt (einschl. Dokumentation) und in zwei Produkte integriert.

**Realisiert unter:**

Java (Swing und JDBC), Datenbank: Oracle 9i

05/2002 - 11/2002

DIT (Fondsgesellschaft)

10 Personen

DIT-Blotter

**Projektbeschreibung:**

Der DitBlotter ist eine zentrale Applikation für den Wertpapierhandel einer Investmentgruppe. Er stellt einen Baustein innerhalb des straight-through-process (STP) dar, der die vollelektronische und somit im Idealfall papierlose Orderweitergabe innerhalb des Investmentprozesses beschreibt.

Zentrale Aufgabe des Dit-Blotters ist die Unterstützung der Fachabteilung Handel bei der Abwicklung von Wertpapierorders. Neben den Kernfunktionalitäten Sammeln, Splitten und Gruppieren von Handelsaufträgen zu Orders ist ein weiterer wesentlicher Bestandteil des xyzBlotters die elektronische Anbindung der Geschäftspartner (Broker). Diese erfolgt unter Verwendung einer FIX-Engine auf der Basis des FIX-Protokolls.

**Projektergebnis:**

Migration von Informix nach Oracle + diverse Verbesserungen der GUI. Release 1.5 ist in Produktion. Folgerelease sind geplant.

**Realisiert unter:**

Java-Frontend (fat client) unter Windows NT, Datenbank: Oracle 9i, Informix Server, Java-Middleware

**02/2002 - 04/2002**

**Bank**

**30 Personen**

**Online Trading System**

**Projektbeschreibung:**

Applet-/Servlet-Design und Implementierung innerhalb des Portalprojektes, Teilfunktionalität für automatische Aktualisierung, mittels Multithreading, vom Webserver über HTTP-Requests zum Browser realisiert. Java SWING Programmierung für alle Applets (spezielle Table-Renderer). Verwendung der Schnittstellen (Integration) zu den hausinternen Komponenten. Durchführung von Performance- und Speicheranalysen. Vorgehensmodell zur Realisierung nach eSP, Programmierung nach xP.

**Projektergebnis:**

Alle Tasks bezüglich GUI von mir erfolgreich umgesetzt.

**Realisiert unter:**

Java, Browser (Internet Explorer, Netscape), J2EE, Windows NT/2000, Oracle 8.1.6, BEA WebLogic 6.1, ANT 1.4.1, CVS, XML & XSL-T, HTML, CygWin 1.3.10, Bugzilla, XEmacs, Optimizelt

**11/2001 - 12/2001**

**DIT (Fondsgesellschaft)**

**3 Personen**

**Produktdatenbank**

**Projektbeschreibung:**

Produktdatenbank für Investmentfonds

Die Produktdatenbank (PDB) ist ein Big-Client/Server-System für Investmentfonds inkl. Workflow, Historisierung und Vieraugenprinzip, Trennung von Business Logik und Datenspeicherung durch Schichtenmodell mit abstrakter Persistenzschicht.

Die GUI wurde alleinig von mir in Swing mit einem eigens dafür entwickeltem schlanken Framework realisiert.

**Projektergebnis:**

Funktionsfähiger Prototyp, der mittlerweile im produktivem Einsatz ist.

**Realisiert unter:**

JBUILDER 5, Java2EE/Swing, Oracle 8.1.7 & Tools, sql2java, ErWin, Continuum, JRefactory

|                             |  |
|-----------------------------|--|
| <b>08/2001 - 10/2001</b>    | <b>DIT (Fondsgesellschaft)</b>   |
| <b>12 Personen</b>          | <b>Security Lending System</b>   |
| <b>Projektbeschreibung:</b> | Rolle: Projekt-Manager.<br>Leitung und Coaching eines 12-köpfigen Projektteams sowie Steuerung & Verfolgung des Projektes SLS (Security Lending System). Festpreisprojekt im Auftrag eines Finanzdienstleisters unter Verwendung des Miracle Frameworks.<br>Vorgehensmodell zur Realisierung nach eSP, Programmierung nach xP. |
| <b>Projektergebnis:</b>     | Erfolgreiches Erreichen der ersten drei Meilensteinen, jedoch wurde aufgrund der Fusion Allianz & Dresdner Bank das Projekt eingestellt.   |
| <b>Realisiert unter:</b>    | Java-Portalapplikation unter Windows, Java-Middleware auf Unix-Rechner.<br>Java, J2EE, Windows NT/2000, Oracle 8.1.7, ANT, CVS, JBuilder   |

|                             |  |
|-----------------------------|--|
| <b>05/2001 - 07/2001</b>    | <b>Commerzbank</b>   |
| <b>1 Person</b>             | <b>Volatility Editor</b>   |
| <b>Projektbeschreibung:</b> | Neuentwicklung eines Java-Applets (Volatility-Editor), mit dem Volatility-Daten (Risikofaktoren) editiert werden können. Die Daten können sowohl durch Editieren von Tabellenwerten, als auch durch Verschieben von Wertpunkten in Grafiken (Volatility-Smile) verändert werden. Tabellen- und Grafikwerte blieben dabei immer synchron.<br>Zu diesem Zweck wurde eine eigene visuelle Komponente entwickelt (Liniendiagramm). |
| <b>Projektergebnis:</b>     | Lauffähiges Programm, kam aufgrund bankinterner ‚Gründe‘ nicht zum Einsatz.  |
| <b>Realisiert unter:</b>    | Java 1.3, Applet   |



|                             |   |
|-----------------------------|---|
| <b>01/2001 - 04/2001</b>    | <b>Dresdner Bank</b>  |
| <b>2 Personen</b>           | <b>Weiterentwicklung des Systems „DRIFT“ für den Handelsbereich Investment Fonds</b>  |
| <b>Projektbeschreibung:</b> | <p>Entwicklung eines IT-Systems zur Erfassung von Handelsgeschäften mit Investmentfonds und Weiterleitung dieser Geschäfte zur Abwicklung. Ablösung der Altanwendung und Ersatz der bisher manuellen Prozesse für das standardisierte Geschäft sowie für das Zahlungs-Lieferungs-Geschäft. Anbindung des Systems an Schnittstellen zur elektronischen Weiterleitung der Geschäfte anstelle von gedruckten Tickets. Analyse zur Ausdehnung auf weitere Produktarten und Umsetzung neuer Funktionalitäten zur Automatisierung.</p> <p><b>Aufgabe:</b> Entwicklung, fachliche Konzeption, Spezifikation.</p> |
| <b>Projektergebnis:</b>     | <p>System befindet sich im produktiven Betrieb.</p> <p>Automatisierung und Rationalisierung der Abläufe im Front-Office des Investment Fund Tradings sowie in der Überleitung zum Back-Office.</p> <p>Kundenreferenz (s. Gulp) vorhanden.</p>   |
| <b>Realisiert unter:</b>    | Powerbuilder-Frontend unter Windows NT, Datenbank: Sybase SQL Server  |

|                             |  |
|-----------------------------|--|
| <b>05/1998-12/2000</b>      | <b>Deutsche Bahn (Transport- und Verkehrswesen)</b>  |
| <b>5 Personen</b>           | <b>Teilprojekt Stammdaten</b>  |
| <b>Projektbeschreibung:</b> | <p>Teamleitung des 5-köpfigen Clientteams im Teilprojekt Stammdaten (30 Personen), Implementierung von GUI-Funktionalitäten, sowie Erstellen von Design-Dokumenten.</p> <p>Das Großprojekt (1000 Personen) CXU bildet die komplette Geschäftslogik von DB Cargo ab. Im Teilprojekt SD werden alle Stammdaten den anderen Teilprojekten bereitgestellt. Zur Pflege der Stammdaten wird eine C/S-Anwendung entwickelt.</p> <p>Großprojekt CXU (AVM, SD),<br/> AVM: C/S- Anwendung zum Erstellen und Pflegen von Auftragsdaten, SD:<br/> C/S- Anwendung zum Pflegen von Stammdaten,<br/> Teamleiter des Client- Teams im Teilprojekt SD (Stammdaten),<br/> Objektorientierte Sprachen,<br/> Rationale Datenbanken,<br/> Datenmodellierung,<br/> Software-Engineering,<br/> Client/Server-Software</p> |
| <b>Projektergebnis:</b>     | Erfolgreiche Einführung von drei Releases in den Produktionsbetrieb.   |
| <b>Realisiert unter:</b>    | Objektorientierte Sprachen, Rationale Datenbanken, Datenmodellierung, Software-Engineering, Client/Server-Software Powerbuilder (Frontend), Java, SQL, Cobol (Server), Perl, Windows NT, FCP Production (Middleware), MQ Series, CSR Map   |

|                          |  |
|--------------------------|--|
| <b>1998</b>              | <b>Universität</b>   |
|                          | Nebentätigkeit während der Promotion   |
| <b>Beschreibung:</b>     | Für den Bereich Chemie einer Universität Webserver unter Linux eingerichtet und HTML-Seiten erstellt (inklusive Javascript). Administration des CIP-Pools des Fachbereichs (Novell Netware Server) |
| <b>Ergebnis</b>          | Der Webserver (www.chemie.uni-hannover.de) ist immer noch aktiv.   |
| <b>Realisiert unter:</b> | HTML, Javascript, Novell-Netware   |

|                          |   |
|--------------------------|---|
| <b>1989 - 1994</b>       | <b>Universität</b>  |
|                          | Nebentätigkeit zum Studium  |
| <b>Beschreibung:</b>     | Auftragsarbeiten im Bereich Programmierung für den AK:<br><br>Spektren-Explorer (Delphi),<br>Schwingungsanalyse (Visual Basic),<br>Sinus (Manipulation von IR-Spektren),<br>Formular-Druckprogramm entwickelt: K16-Druckprogramm (Pascal) |
| <b>Ergebnis:</b>         | Alle Programme sind im Einsatz  |
| <b>Realisiert unter:</b> | Objektbasierte Sprachen unter Windows   |

|                          |   |
|--------------------------|---|
| <b>1989</b>              | <b>Gymnasium</b>  |
|                          | Nebentätigkeit zum Gymnasium  |
| <b>Beschreibung:</b>     | Folgende Tools und Spiele in Eigenregie entwickelt:<br><br>DOS-Tool: Xtree<br>DOS-Tool: XDel<br>DOS-Tool: Run<br>DOS-Tool: XCD<br>PC-Spiel: Blockbuster<br>PC-Spiel: Tetris<br>PC-Spiel: Pacman |
| <b>Ergebnis:</b>         | Alle von mir entwickelten Programme wurden in der Zeitschrift , „PC-Amstrad International“ bzw. auf der entsprechenden Monats-Diskette veröffentlicht.  |
| <b>Realisiert unter:</b> | Turbo Pascal unter MS-DOS   |