



# Beraterprofil Peter Eierdanz



- Diplom Ingenieur technische Informatik
- Senior IT-Berater (seit 2001 freiberuflich)

## 1. Persönliche Daten

|               |  |
|---------------|--|
| Ausbildung    | Diplom Ingenieur technische Informatik (Abschluss 1998)  |
| Geburtsjahr   | 1971   |
| Nationalität  | Deutsch  |
| Fremdsprachen | Englisch   |
| Position      | <b>Senior JAVA Backend Entwickler</b>  |
| Schwerpunkte  | <b>Java JDK 11, Camunda BPMN Prozessmodellierung, Serviceorientierte Architektur Rest+Soap Service, Spring Boot, Authentifizierung über JWT, Performance Optimierung, Schnittstellen Design, Agile Software Entwicklung nach Scrum</b> |
| Branchen      | Öffentlicher Dienst, Bank und Finanz, Versicherung, Automotive, Logistik und Transport, Telekommunikation  |
| Verfügbarkeit | Sofort   |

(Stand: Januar 2021)





## 2. Überblick Projekterfahrungen

|   |  |
|---|--|
| Pifikterfahrungen<br>Branchen<br>Kunden | Versicherung – Urlaubskasse, Antragsverwaltung (Sozialkasse Bau)<br>Finanzbranche – Kernsystem einer Auskunftei (Schufa Holding)<br>Bankenumfeld – Mobile Geldanlage Strategie (Deutsche Bank AG)<br>Bankenumfeld – Mobile Banking App (Finanzinformatik)<br>Bankenumfeld – Ausfall und Risikogeschäft (Commerzbank AG)<br>Logistik – (DB Schenker Rail bzw. DB-Systel)<br>Bankenumfeld – Ratinginformationen (VR-Leasing)<br>Telekommunikation (Deutsche Telekom)<br>Bankenumfeld - Kredit Geschäft (KfW Bank)<br>Investment Banking (Dresdner Kleinwort Wasserstein)<br>Automobil - Industrie (BMW)<br>Bankenumfeld - Umsatzverarbeitung (Dresdner-Bank)<br>Versicherung – (Allianz, AMB Generali Gruppe)<br>Öffentliches Institut (Städel Museum Frankfurt am Main) |
| Entwicklungserfahrungen                 | Client und Server-Seitige Java Programmierung<br>Java JDK 11<br>Camunda BPMN Modellierung<br>JUnit / Mockito / EasyMock / PowerMock<br>JPA / Hibernate / PL-SQL<br>Webservices Rest und SOAP<br>Spring Boot<br>Schnittstellen Design und Implementierung<br>Agile Software Entwicklung nach SCRUM  |
| Wichtigste Tools / Produkte             | IntelliJ und Eclipse<br>Camunda<br>Enterprise Architect<br>Maven<br>Git<br>Atlassian Suite (Jenkins, Bitbucket, Confluence)<br>JBoss<br>Oracle Weblogik<br>Java Flight Recorder<br>Java Web Token<br>Continuus Integration<br>Artifactory zusammen mit TeamCity  |





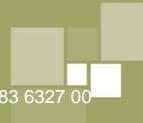
|                             |   |
|-----------------------------|---|
| Zusätzliche Qualifikationen | <p>Entwicklungsleitung – Teamstärke von 15 Mitarbeitern<br/>Sehr gute Kommunikation mit nicht Technikern<br/>IT-Architekturen beraten, überarbeiten, verfeinern<br/>Speziell serviceorientierte Architekturen, SOA und Rest-Services<br/>Erstellen von Projektplänen und Ressourcenplanung<br/>Performanceoptimierungen<br/>Inbetriebnahmen<br/>Security<br/>Umsetzungen und Schätzungen von CR's, LD's und oder FD's<br/>Semantic Web: Erstellen von Ontologien mit RDF, RDFS, OWL, SPARQL, SKOS<br/>Ticket bzw. Bugfixing<br/>Taskforceleitung<br/>Erfahrungen mit verschiedenen firmeneigenen J2EE Frameworks, schnelle selbständige Einarbeitung bestehender Bibliotheken<br/>Fundierte Projekterfahrungen mit komplexen<br/>Versicherungsabläufen<br/>Ausgeprägte Test Erfahrungen mit JUnit und http-Unit, automatisierte WEB UI Tests mit Selenium<br/>Mitarbeiter Schulungen mit Eclipse (Juno)<br/>Konfiguration / Administration von Applikation bzw. Webservern<br/>Erfahrung in agilen Projekte nach Scrum<br/>Anbindung der Figo API für Multibanking<br/>BPMN 2.0 mit Camunda</p> |
|-----------------------------|---|





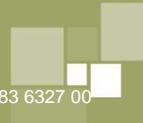
### 3. Ausgewählte Projekte

|                                      |   |
|--------------------------------------|---|
| <b>Zeitraum</b>                      | 01.04.2020 – heute  |
| <b>Projektname</b>                   | Kairos  |
| <b>Kunde</b>                         | SOKA-Bau / Wiesbaden und Rüsselsheim  |
| <b>Position</b>                      | Senior Java Backend Developer mit Schwerpunkt Camunda Prozess Modellierung  |
| <b>Entwicklung</b>                   | Java 11<br>Backendentwicklung<br>BPMN 2.0 mit Camunda<br>REST und Soap Service Implementierung (Jax-RS)<br>JUnit Test mit Mockito und Powermock   |
| <b>Fachliche Projektbeschreibung</b> | Die SOKA-Bau (Sozialkasse der Bauwirtschaft) ist die gemeinsame Dachmarke für die Urlaubs- und Lohnausgleichskasse der Bauwirtschaft und die Zusatzversorgungskasse des Baugewerbes. Mit dem Projekt Kairos werden die bestehende in Produktion befindlichen Portale eines Melde- bzw. Antragssystem für Urlaubsansprüche äquivalent abgelöst. Es gibt sowohl eine Benutzeroberfläche für die Kunden der SOKA-Bau als auch ein Portal für die internen Sachbearbeiter des Supports. Die Kunden können Urlaubsansprüche beantragen und verschiedene Arten von Meldungen, z. B. Fremdzeiten, Ausbildungszeiten, Krankheitstage oder ähnliches registrieren. Die Software kalkuliert aus den zur Verfügung stehenden Informationen auf Basis der in Tarifverträgen vereinbarten Regeln die erreichten Ansprüche und verifiziert diese mit den eingereichten Anträgen |





|                                      |  |
|--------------------------------------|--|
| <b>Zeitraum</b>                      | 24.07.2017 – 31.03 2020  |
| <b>Projektname</b>                   | SPIRIT   |
| <b>Kunde</b>                         | Schufa Holding AG / Wiesbaden  |
| <b>Position</b>                      | Senior Java Backend Developer  |
| <b>Entwicklung</b>                   | Java 8<br>Backendentwicklung<br>BPMN 2.0 mit Camunda<br>Datenmodellierung mit Enterprise Architekt<br>REST Service Implementierung (Jax-RS)<br>JUnit Test mit Mockito und Powermock  |
| <b>Projekt Architektur</b>           | Anbindung bestehender Backend Systeme über serviceorientierter Architektur<br>Entwicklung eigener Nachrichtenannahme<br>Transformationen verschiedener fachlichen Nachrichtenformate<br>Entwicklung über Message Driven Beans<br>Mockito und EasyMock  |
| <b>Darüber hinaus</b>                | Im Projekt wurde agile nach SCRUM gearbeitet.<br>14 tägige Entwicklungszyklen (Sprints)<br>Stetige Optimierung der Prozesslogiken mit Schwerpunkt Perfomanceoptimierung<br>Beratung in Design und Konzeption   |
| <b>Fachliche Projektbeschreibung</b> | Im Projekt SPIRIT bei der Schufa Holding AG in Wiesbaden wird das bestehende Kern-System der Auskunftei neu entwickelt. Hierbei werden in einem hochverfügbaren und sehr performancekritischen Umfeld höchste Ansprüche an die Produktivsysteme gestellt.<br><br>Durch das Projekt SPIRIT können authentifizierte Partnerunternehmen aus den verschiedensten Branchen (Finanzsektor, Telekomunikation, Handel usw.) Informationen, wie beispielsweise ein Bonitäts-Scoring, über registrierte Unternehmen und oder Privatpersonen erhalten.<br><br>Die Schufa ist einer der führenden Auskunfteien in Deutschland. |



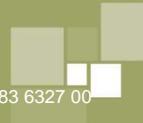


|                                      |   |
|--------------------------------------|---|
| <b>Zeitraum</b>                      | 09.01.2017 – 30.06.2017   |
| <b>Projektname</b>                   | PWS / dDPM  |
| <b>Kunde</b>                         | Deutsche Bank AG  |
| <b>Position</b>                      | Senior Java Developer   |
| <b>Entwicklung</b>                   | Java 8<br>Backendentwicklung Rest-Service über Spring Boot 4<br>Schnittstellenabstimmung zu Angular JS Service Komponenten<br>Spring Security – CSRF-Token und OAuth2<br>JUnit Test mit Mockito und Easymock<br>Common Library Implementierung<br>Dynamische Spring Properties Verwaltung   |
| <b>Projekt Architektur</b>           | AngularJS 4 Frontend<br>Spring<br>Anbindung bestehender Backend Systeme über serviceorientierter Architektur<br>Mockito und EasyMock  |
| <b>Darüber hinaus</b>                | Im Projekt wurde agile nach SCRUM gearbeitet.<br>Die Entwicklungszyklen (Sprints) waren direkt mit dem Kunden bzw. PO abgestimmt.<br>Stetige Optimierung des Ablaufes der Prozesslogiken.<br>Beratung in Design und Konzeption  |
| <b>Fachliche Projektbeschreibung</b> | Über dDPM wird eine neue moderne Investment Plattform für verschiedenen Endkunden zur Verfügung gestellt. Hierbei werden sowohl existierenden Deutsche Bank Kunden als auch Neukunden über das bekannte PWS Angebot (Public Websites) der Deutschen Bank eine individuelle Anlagestrategie on demand angeboten. Der Verarbeitungsprozess für dDPM wird hierbei in Profiling, Strategieberechnung und Depoteröffnung untergliedert. Im Profiling werden alle relevanten Kundendaten, wie beispielsweise Angaben zur finanziellen Situation oder Risikobereitschaft des Kundens erfragt. Anschließend werden diese Werte aufbereitet und an die entsprechenden Backendsysteme der Deutschen Bank übermittelt, die dann eine individuelle Anlagestrategie errechnen. Am Ende wird online ein Depoteröffnung durchgeführt und der Kunde somit als Bestandskunde registriert |





|                                      |  |
|--------------------------------------|--|
| <b>Zeitraum</b>                      | 10.10.2016 – 31.12.2016  |
| <b>Projektname</b>                   | M-Konto  |
| <b>Kunde</b>                         | 1822direkt / Finanz Informatik Solutions Plus GmbH   |
| <b>Position</b>                      | Senior Java Developer  |
| <b>Entwicklung</b>                   | Anbindung der Figo-API für Multibanking<br>Backendentwicklung Rest-Service mit JAX-RS<br>Schnittstellenabstimmung zur MKonto App (JSON)<br>Weitere Implementierung von Rest Services für neue Geschäftsfälle<br>JUnit Test mit Mockito und Easymock  |
| <b>Projekt Architektur</b>           | Oracle 11g Datenbank<br>Hibernate 4.2.7<br>Spring, EJB<br>Mockito und EasyMock   |
| <b>Darüber hinaus</b>                | Im Projekt wurde nach Scrum gearbeitet.<br>Die ganzen Entwicklungszyklen (Sprints) waren direkt mit dem Kunden abgestimmt<br>Stetige Optimierung des Ablaufes der Prozesslogiken<br>Beratung in Design und Konzeption<br>Die M-Konto App für Endanwender wurde termingerecht fertig gestellt.  |
| <b>Fachliche Projektbeschreibung</b> | Die 1822direkt ist eine Tochter der Frankfurter Sparkasse.<br>Mit der neuen Mobile-Konto App sollen vor allem jüngere Kunden gewonnen werden. Grundsätzlich unterstützt die App alle Standard Kontofunktionalitäten des Online-Bankings.<br>Darüber hinaus werden weitere Funktionen wie beispielsweise das Multibanking über die Figo-API angeboten.<br>Die App ist mit AngularJS implementiert.<br>Mein Schwerpunkt lag in der Entwicklung von Rest-Servicen mit JAX-RS, die für die App zur Verfügung gestellt wurden |





|                                      |  |
|--------------------------------------|--|
| <b>Zeitraum</b>                      | 01.10.2015 – 07.10.2016  |
| <b>Projektnname</b>                  | CIMT   |
| <b>Kunde</b>                         | Commerzbank AG   |
| <b>Position</b>                      | Senior Java Developer  |
| <b>Entwicklung</b>                   | <p>Die Entwicklung basierte auf FRAME - dem neuen Commerzbank Projekt Standard.</p> <p>Frontentwicklung mit Techniken wie JSF 2, Primefaces, AJAX, Java-Script, CSS.</p> <p>Backendentwicklung mit Hibernate, Oracle, SOA Web-Services</p> <p>JUnit Test zusammen mit PowerMock und EasyMock</p> <p>Frontend Testing u. a. mit Selenium</p> <p>Ticketanalysen und Ticketfixing</p>   |
| <b>Projekt Architektur</b>           | <p>Oracle 11g Datenbank</p> <p>Hibernate 4.2.7</p> <p>JSF 2 inklusive (Primefaces 5.2, AJAX, Javascript, JQuery, CSS...)</p> <p>Spring, Spring MVC, Spring Web Flow</p> <p>EasyMock und PowerMock</p>  |
| <b>Darüber hinaus</b>                | <p>In einem agilen Ansatz (Scrum) wird sehr eng mit der Fachseite zusammengearbeitet.</p> <p>Durch kurze Entwicklungszyklen werden Zwischenergebnisse gemeinsam erarbeitet.</p> <p>Stetige Optimierung des Ablaufes der Prozesslogiken</p> <p>Beratung in Design und Konzeption</p>  |
| <b>Fachliche Projektbeschreibung</b> | <p>Mit dem Projekt CIMT kann der Fachbereich im Bereich Risikovorsorge nach dem IFRS9 Standard Ausfallscenarien erfassen und pflegen. Außerdem können darüber alle notwendigen Vorsorge Maßnahmen verwaltet werden. Cashinflows sowie Cashoutflows können für die relevanten Geschäftsfälle erfasst und dokumentiert werden. Ausfallwahrscheinlichkeiten können erfasst bzw. berechnet werden.</p> <p>Über SOA-Webservices werden die Daten zu weiteren Systemen der Bank verteilt um notwendigen Maßnahmen zu automatisieren.</p> |





|                                      |  |
|--------------------------------------|--|
| <b>Zeitraum</b>                      | 10.05.2014 – 30.09.2015  |
| <b>Projektnname</b>                  | EWDS (Empty Waggon Disposition System)   |
| <b>Kunde</b>                         | DB-Systel / DB Schenker Rail   |
| <b>Position</b>                      | Senior Java Developer  |
| <b>Entwicklung</b>                   | Technisches Design, Klassendiagramm, Sequenzdiagramme mit Enterprise Architect<br>Schnittstellen Spezifikationen für Rest und SOA Services<br>Frontendentwicklung mit Techniken wie JSF 2, Primefaces...<br>Entwicklung spezieller JSF Komponenten<br>Frontend Testing u. a. mit Selenium  |
| <b>Projekt Architektur</b>           | Oracle 11g Datenbank<br>Hibernate 4.2.7<br>SOA basierende Architektur >> JAX-WS<br>JSF 2 inklusive (Richfaces 4.3.5 , Primefaces 4.0, AJAX, Javascript, JQuery...)   |
| <b>Darüber hinaus</b>                | Enge Zusammenarbeit und Abstimmung mit Fachseite<br>Stetige Optimierung des Ablaufes der Prozesslogiken<br>Beratung in Design und Konzeption<br>Performanceoptimierungen   |
| <b>Fachliche Projektbeschreibung</b> | In dem Projekt EWDS für DB Schenker Rail, wurde ein Dispositionssystem entwickelt, über den die Bestellungen von Güterwagen zu den passenden Beständen zugeordnet werden können. Das System bietet eine automatische Zuordnung oder ermöglicht es den Disponenten Bestellungen, Bestände und Dispositionen manuell über eine Benutzeroberfläche zu ergänzen. Zur Ermittlung von Fahrzeiten, Fahrplanauskünften oder Fahrtkosten werden weitere Bahninterne System mit EWDS gekoppelt.<br>Die Benutzeroberfläche unterstützt modernste Frontendkomponenten wie beispielsweise eine LazyLoading (automatisches dynamisches Nachladen) einer TreeTable mit mehreren Hierarchieebenen.<br>EWDS ist ein international einsetzbares System was als Ziel die optimale Unterstützung der Endanwender hat. Darüber hinaus sollen durch EWDS mehrere ältere System der Bahn abgelöst werden. |



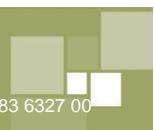


|                                      |  |
|--------------------------------------|--|
| <b>Zeitraum</b>                      | 01.11.2013 – 30.04.2014  |
| <b>Projektnname</b>                  | Rating Abbildung   |
| <b>Kunde</b>                         | VR-Leasing AG in Eschborn  |
| <b>Position</b>                      | Senior Java Developer / IT-Architektur   |
| <b>Entwicklung</b>                   | <p>Frontendentwicklung mit Techniken wie JSF, HTML, CSS, JavaScript, Richfaces, AJAX</p> <p>Anbindung an Webservices anderer Systeme (meist SAP-Systemen)</p> <p>Entwicklung von Java Webservices</p> <p>Entwicklung spezieller JSF Komponenten</p> <p>Ticketanalysen und Ticketfixing</p> <p>Applikationserver Oracle Weblogic</p>  |
| <b>Besondere Motivation</b>          | <p>In dem Projekt (Rating Abbildung) bei der VR-Leasing in Eschborn war ich als „normaler“ Java Senior Developer tätig. Nachdem bei meinem vorherigen Projekt bei der Deutschen Telekom (THOP) die eigentliche Softwareentwicklung mehr und mehr in den Hintergrund geraten ist, war es mir sehr wichtig den Schritt zurück in die Entwicklung wieder zu finden.</p> <p>Meiner Meinung nach kann man weitere Aufgaben wie Design, Architekturberatung oder Teamleitung nur dann qualifiziert leisten, wenn man einen möglichst guten Kenntnisstand über die aktuell gängigen Entwicklungswerkzeuge besitzt. Und um diesen Kenntnisstand sicher zustellen ist ein regelmäßiges Auffrischen von Nöten. Dieses Auffrischen versuche ich speziell durch neue Entwicklungseinsätze zu bekommen.</p> |
| <b>Fachliche Projektbeschreibung</b> | <p>Mit dem Projekt "Rating Abbildung" bei der VR-Leasing AG, sollen Mitarbeiter des Unternehmens in dessen Kerngeschäft besser unterstützt werden.</p> <p>Um alle verfügbaren Informationen über die Bonität eines Kunden zu bekommen waren bisher mehrere unterschiedliche Tools notwendig. Durch die "Rating Abbildung" wird ein führendes Rating basierend auf verschiedene Quellen errechnet und somit dem Mitarbeiter als zentrale Information über den Kunden zur Verfügung gestellt.</p> <p>Darüber hinaus bietet die "Rating Abbildung" einige weitere Funktionalitäten wie beispielsweise ein zentrales Reporting oder ein übersichtliches Mapping zwischen den Ratings verschiedener Ratingagenturen.</p>  |





|                     |   |
|---------------------|---|
| <b>Zeitraum</b>     | 01.01.2008 – 31.10.2013   |
| <b>Projektnname</b> | THOP (T-Home Order Process)   |
| <b>Kunde</b>        | Deutsche Telekom AG – Darmstadt und Bonn  |
| <b>Positionen</b>   | <p>Java Senior Developer und mehr</p> <p>Neben der Haupttätigkeit als Java Senior Developer habe ich weitere Aufgaben wahrgenommen:</p> <ul style="list-style-type: none"><li>• Unterstützung der Projektleitung</li><li>• Analyse, Design und Konzeption</li><li>• Entwicklungsleitung – Teamstärke ca. 15 Personen</li><li>• IT-Architektur</li></ul>   |
| <b>Techniken</b>    | <p>IDE: Eclipse (Juno)<br/>Application Server: JBoss<br/>Datenbank: Oracle</p> <p>Java: JDK 1.6</p> <p>Weiteres: JSF, Webservices, Spring, EJB, Richfaces, Ajax, HTML, CSS, SQL, XML, JMS, JUnit...</p> <p>Versionsverwaltung mittels ClearCase und SVN</p>   |
| <b>Besonderes</b>   | <p>In dem Projekt (THOP) bei der Deutschen Telekom habe ich einige verschiedene Rollen bzw. Tätigkeiten wahrgenommen. Wie bereits oben tabellarisch beschrieben, waren meine Aufgaben neben der Entwicklung auch in der Analyse und Design. In den letzten beiden Jahren des Projektes habe ich zusätzlich noch die Projektleitung bei speziellen Aufgaben, in denen sehr gute Kenntnisse des Gesamtsystems und deren Komponenten Voraussetzung sind unterstützt.</p> <p>Insgesamt konnte ich mir einiges Wissen über die verschiedenen Systeme der Telekom aneignen, weshalb ich auch über die Grenzen des Projektes hinweg als kompetenter Ansprechpartner gelte.</p> <p>Das Projekt THOP habe ich mehrere Jahre begleitet. Dabei konnte ich alle Phasen eines Projektes von der Erstellung eines Prototypens über die eigentliche Entwicklung und dem anschließenden Testphasen (Systemtest, Clustertest, Last und Performancetests) bis zu den erfolgreichen Inbetriebnahmen miterleben. Am Ende lief die Software stabil im Wirkbetrieb und wurde nur noch punktuell weiter verbessert und erweitert. Hierfür wurden beispielsweise weitere fachliche Change Requests umgesetzt oder Performanceoptimierungen analysiert und entwickelt.</p> |





## Schwerpunkten im Projekt THOP

### Unterstützung Projektleitung:

- Erstellung von Projektplänen
- Resourcenplanungen
- Kostenschätzungen

### Design:

- Erstellen, verfeinern und überarbeiten von technischen Designs und Fachkonzepten inklusive UML Diagramme und Schnittstellenvereinbarungen.
- Enge Abstimmungen mit Fachseiten bis hin zur technischen Beratung

### Architektur:

- SOA und Rest-Serviceorientierte Architektur
- Perfomanceanalysen und Optimierungen
- Security

### Entwicklung:

- Entwicklung von Java Webservices
- Anbindung an Webservices anderer Systeme
- Datenbankanbindung über Hibernate
- Frontentwicklung - Techniken: JSF, HTML, CSS, JavaScript, Richfaces, AJAX
- Entwicklung spezieller JSF Komponenten
- Ticketanalysen und Ticketfixing





## Ausführliche Projektbeschreibung von THOP

Im Projekt THOP wurde eine J2EE Web-Applikation entwickelt, mit dem die Partner des Unternehmens, viele verschiedene Geschäftsprozesse abwickeln können. So können zum Beispiel Neukunden Neuanschlüsse, Produktwechsel, Änderungen von Bestandskundendaten oder auch Auftragsrecherchen durchgeführt werden. Hierbei wird auf bestehende Backend Systeme zurückgegriffen.

In der Anwendung, wird der Benutzer, durch verschiedene Masken geführt bis er zum Abschluss eines jeden Geschäftsprozesses einen Auftrag abschließen kann. Hierbei werden per Webservice mehrere bestehende Backend Systeme aufgerufen. Die Seiten der Web-Anwendung enthalten zum einen verschiedene spezifische Telekommunikations-Informationen, zum Beispiel Produktauswahlseiten, Verfügbarkeitsprüfungen, Telefonbucheinträge oder auch Rufnummernauswahl, und darüber hinaus auch „alltägliche“ Seiten wie zum Beispiel Bonitätsprüfungen oder auch allgemeine Geschäftsbedingungen, die man auch in anderen Web-Anwendungen finden kann.

Technologisch war es ein sehr anspruchsvolles Projekt, da viele verschiedene neue Technologien im Zusammenspiel gebracht wurden. So wurde als J2EE Web-Framework JSF (Java Server Faces) verwendet. Darüber hinaus wurden bei der Frontendentwicklung zusätzlich Richfaces, Ajax, und Portlets eingesetzt. Mit den Backends wird in der Regel über Webservices kommuniziert, hier sind die Standards JAX-WS und Axis im Einsatz. Einige Systeme wurden auch direkt über MQ angebunden. Ansonsten sind noch weitere Standards im Einsatz wie z. B. Spring, EJB, usw...

Weitere Projekterfahrungen inklusive ausführlichen Beschreibungen zu meinen Tätigkeiten finden Sie online unter <https://www.eierdanz.com> oder auf Anfrage...

