Benjamin Diedrichsen

Senior Software Engineer
Dipl. Wirtschaftsinformatik(FH)[2008]

Nationality: German
Year of Birth: 1984

mail: b.diedrichsen@gmail.com
github: https://github.com/bennidi

stackoverflow: bennidi

Abstract

For about 10 years I have been working as a professional engineer in different industries. Working mostly as a freelancer, I had plenty of opportunities to gain experience with a variety of technologies, software landscapes and teams.

Fluent in multiple programming languages, I am most comfortable with JavaScript and **Node.js** as well as **Java** and **J2EE** related technologies.

I definitely enjoy upfront **domain modeling and API design** as building block for clean and focused coding.

I also tend to enjoy writing of documentation (Open API, **UML**, JsDoc etc.) and unit/integration tests aligning with use cases.

MDSD

My career started 2007 with a Thesis on Model Transformations using an MDSD approach.

It was a contribution to large scope persistence model migration and allowed me to learn within a team of very experienced developers right from the beginning. It also deepened my interest in the field of **Domain Specific Languages**, API design and domain modeling.

J2EE

Growing up in the Java Enterprise world I worked on various typical enterprise applications: three-tier designs (**EJB/Spring**) message queues for job processing and service decoupling, object-relational mappers (Hibernate, Data Nucleus) on top of relational data stores (Oracle, Postgres, Mysql) and some more or less sensible view technology (JSF, Wicket, Struts).

Code was written in Java, sometimes Groovy, following mostly **object-oriented design** principles.

GoF / DDD / TDD

The use of design patterns (GoF, Fowler, Uncle Bob etc.) to keep code DRY and **SOLID** has become daily practice by then. I draw ideas and inspirations from **Domain Driven Design**, design by contract (OCP), test-driven development.

Using Groovy and Spock I learned to enjoy **well written test specifications** not only to ensure correct functioning of code but also as documentation and a great way to support good API design.

Proper **test coverage** has become an essential ingredient to my well-being as a user-friendly engineer.

My work included implementations of **persistence layers**, caching solutions, design of database models, **internal DSLs** as well as refactorings/rewrites, performance tuning (JProfiler, JMeter) and a high-throughput event bus (github:mbassador).

In a Scala project, I became proficient with the **fundamentals of FP**:

higher-order functions, currying, monadic structures (futures, streams etc.) and the usual data processing tools (map-reduce/fold).

Fullstack

JavaScript lead me to my first frontend projects based on knockout or react and backends based on Node.js.

Within the JS ecosystem I learned to appreciate fast turn-around cycles [integration tests within seconds, hot-code reload...] in local envs and CI-pipelines as an enabler for delivery in short iterations.

Cloud and SaaS

With AWS setting the trend, cloud software has become a common place in software development.

Deployment into clusters, transparent service discovery and monitoring, distributed message queues and data stores are all daily business.

I understand many of the challenges of cloud software environments (logging, request tracing, synchronization, data consistency) because they arise in most distributed software systems.

Even though my services deployed to AWS and used DynamoDB, S3, SQS – I do not consider myself an experienced cloud user. I learn what I need to know to get the job done.

DevOps

Regarding DevOps (CI-Pipelines, Deployment with k8s, swarm, ansible...), I consider myself a user. I integrated several project templates to work with requirements of CI environments but never needed to build my own pipelines or design/implement cluster deployments.

Degree program

01.10.2004 –30.09.2008	**Co-operative degree program 'Economics and computer science (Dipl. FH)' in co-operation with ottogroup**
Focus areas	Model driven software development ■ Modern software technologies ■ Datastream Managementsystems
Diploma thesis	Implementation of a generic transformation component for XML based models and its application to a model migration problem
Final grades	1.0 (thesis), 1.5 (diploma)

Software Projects

10/2019 – 06/2020 **Fullstack Software Engineering (node.js, react.js)**

Kientalerhof AG (Suisse)
Hotel Management

Kientalerhof offers room rental for tourism and seminar guests. In-house IT uses old-fashioned client-server software for room and guest managemend. The system was to be replaced with modern web application offering components for workshift planning across multiple departments, resource planning and integration with 3rd-party services for accounting, email templating and document generation.

Technologies

ES7, node.js, koa.js, react.js, mobx, antd, mui, enzyme, webpack, jquery calendar, ag-grid

Tasks

- Implementation of backend and frontend using node.js for RESTful data access and react.js with mobx for client
- Design of meta-models for data exchange
- Integration with jquery calendar for workshift planning
- Integration with ag-grid for queryable, sortable master detail views

03/2018 – 11/2018 **Senior Software Engineer (node.js)**

free2move
Car Sharing Service

Design and feature complete implementation of backend for free float car sharing service. Backend written in JavaScript (ES7) with koa middlewares connecting to MongoDB and Redis. Highly scalable RESTful API with many integration to third party services. Working around inconsistencies of all third party services to provide consistent programming mechanics and API to mobile clients.

Technologies

ES7, node.js, koa.js, bull.js, OpenAPI, MongoDB, Redis, RESTful

Tasks

- Design and coding of RESTful API for Android and iOS clients
- Implementation of data model for fleet management and all intermediary data needed for integrations with 3rd party
- Load testing to ensure scalabilty with peak-load (1000+ requests/sec)
- Job scheduling with bull.js, lots of failover handling with persistent messages using Redis

12/2016 – 07/2017 **Senior Software Engineer (Java, Spring Boot, node.js)**

Affinitas GmbH
eDarling, ElitePartner

Development of backend services in a [multi-monolith] software landscape powering multiple online dating platforms. A transition to a microservice architecture with automated AB-deployment on AWS was ongoing. Work was organized in agile teams following SCRUM methodology.

Technologies

Java 8, Spring Boot, node.js, CoffeeScript, koa, gulp, SQS, DynamoDB, Kinesis, RAML, Eureka, Spinnaker

Tasks

- Rewrite of push notification service with Spring Boot and Java. Instance local message queue with automatic overflow into SQS
- Prototyping of new world node.js microservice template
- Design of RESTful API for in-app micropayment. API first approach with RAML. Implemention on node.js with koa.js and

coffeesript. Integration with existing billing system

05/2016 – 10/2016 **Fullstack Software Engineer (CoffeeScript, React)**

www.mein-schulplan.de

Rewrite of existing web application to new technology stack and mobile first approach

Technologies

React, CoffeeScript, ES6, Gulp, PouchDB, jQuery, Bootstrap, LESS

Tasks

- Technology evaluation and implementation of first prototype with React and CoffeeScript
- Prototyping of local storage with auto-syncing capabilities to backend storage using PouchDB
- Integration with jQuery calendar plugin

07/2015 – 12/2015 **Fullstack Developer (Scala, Java, Javascript)**

Okotta GmbH

Insurance Business

Development of backend services and back office tools for a mobile app (iOS, Android) in insurance market. The team's mission was to build a relatively complex MVP. RESTful API with flexible query capabilities built with JVM technologies like Scala, Java 8, Spring Boot. Standalone API integration tests written in Groovy.

Technologies

Scala 2.11, Scalatra, MongoDB, Swagger, WireMock, React, ES6

Tasks

- Requirements engineering, domain modeling and
- Prototyping of RESTful API with Scala, Scalatra and MongoDB
- Implementation of various RESTful data repositories
- Integrate WireMock for integration testing using JSON fixtures
- Prototyping of back-office tool based on React.js

06/2014-12/2014 **Backend Developer (Java 7, GAE)**

Small Improvements
small-improvements.com

Java Web application running on Google App Engine. BigTable was one and only storage model and bad fit for existing domain model. Extensive use of Memcache. A migration towards a more RESTful backend and decoupled frontend was ongoing.

Technologies

Google App Engine, Java 6, Wicket, BigTable, Memcache, JMeter, JProfiler, Jackson, Groovy, Spock, Gradle

Tasks

- (Re-)Implementation of REST-API layer with composable serializers and less boilerplate code
- Performance optimization of object serialization based on JMeter and JProfiler insights
- Integration of caching layer using Guava Cache

03/2013 - 09/2013 **Senior Software Engineer (Java 7, J2EE, EJB)**

ProfitBricks GmbH

Cloud and IaaS Provider
www.profitbricks.com

J2EE backend services for provisioning of virtual data center components mapped to high-performance hardware. Technically challenging system requirements due to timing issues and call orchestration when crossing system boundaries between hardware management run in Linux world and application code run in JVM

Technologies

J2EE, Java 7, EJB 3.1, EclipseLink, JPA 2.1, Glassfish, jUnit, cglib, OSGI, Postgress, VisualVM, H2, Git, Maven, Jenkins

Tasks

- Profiling and performance optimizations of production code
- Implementation of start/stop feature

- Rewrite of entire testing layer. Implementation of lazy evaluation DSL with cglib proxies and integration with in-memory DB

02/2012 - 12/2012 **Fullstack Software Engineer (Java, Spring, JavaScript)**

Alecto GmbH

Insurance Industry
www.friendsurance.de

JOnline platform offering a trust network environment for insurance customers. Very traditional three-tier application built with Java 6, Spring Services, Spring Web MVC and persistence layer on top of Hibernate and Postgres

Technologies

Java 7, EJB 3.1, Hibernate (JPA 2.1), Spring 3, Tomcat 7, Glassfish, jUnit, JSF 2, Selenium, jQuery, Postgres, Servlet, Git, Maven

Tasks

- Migration from EJB 3.1 (Glassfish) to Spring (Tomcat)
- Redesign of the persistence layer. Social network graph based on flat immutable data (Postgres) with time-travel capabilities
- Implementation of selenium test infrastructure on grid server

Archived

10/2008 – 03/2009 **Software Developer and Project Manager**

www.gentleware.com

Java Software Development and project management of nearshore developer team in Ukraine

06/2009 – 06/2010 **IT Teacher (Mexico)**

www.weltwaerts.de

Teaching students in secondary schools in rural areas of Mexico as part of a voluntary program for intercultural exchange

12/2010 – 03/2012 **Software Engineer (Java J2EE)**

www.atmosfair.de

Rewrite of emmission calculation engine using service oriented architecture based on J2EE (EJB, JBoss AS).

Co-operative course of studies

15.10.2007 – 30.09.2008 **Softwaredevelopment ottogroup retail systems**

www.ottogroup.com

Preparation and partial migration of the object relational mapping layer (JDO → JPA) in a large J2EE landscape. 100+ domain classes with multiple client systems reading from data services.

Technologies

Java 6, JDO 2.0, JPA 1.0, jUnit, Spring, Xtext, Xtend,Xpand,

Tasks

- Extension of existing persistence infrastructure for operating JDO and JPA in parallel (soft migration strategy)
- Design and implementation of small model-to-model transformation language using Xtext, Xtend, Xpand
- Implementation of mapping rules to transform existing JDO mappings

Open Source Projects

[github:mbassador]	High-Performance Java Event Bus library: Implementation of publish-subscribe pattern. Low memory footprint, nonblocking iterators, high throughput
[github:mailody]	Online trash mail service built with Node.js, MongoDB, restify, knockout.js, HTML 5, Java client for Selenium
[github:kermit]	Kermit is an elegant, stream-based, extensible web scraper system. It is packed with pluggable features and offers a plugin mechanism for easy extension. Written in Node.js and coffeescript with NeDB as datastore.
[github:scalatrest]	Scala and Scalatra based stack for building standard RESTful APIs on top of [reactive] MongoDB. Model and Data driven. Very scalable (1000 write ops per sec)
[gihub:daoism}	Data access layer abstraction using DAO pattern. Back in the days, this was still cool.

Languages

*Foreign**

	oral	writing
English	C1	C2
Spanish	B2	B2
French	A1	A2

** Self evaluation according to rEuropean Language Portfolio (see appendix or www.coe.int/portfolio)*

Native Deutsch