

Dr. rer. nat. Sandeep Sadanandan [↗](#) [↗](#)



Contact

Saalburstr. 49
60385 – Frankfurt,
Germany
+49-176-666-45928

sands@whybenormal.org

[Homepage](#) [↗](#)

[Code Artist](#) [↗](#)

Languages

English, German,
Malayalam, Tamil,
Hindi

Programming

C, Go, C++,
Python, Ruby, C#,
Java, JavaScript, Objective
C, Swift

github.com/sillyfellow



General

Vim, Rails, Git,
CLI, AWS Services,
PostgreSQL, PostGIS,
L^AT_EX, Wireshark, GDB,
GNU Make, Emacs

Strengths

Algorithms,
Analytical, Quick
learner, Mentoring,
Energetic, Industrious,
Resourceful

Reference

Mr. Andreas Leicher –
CTO at a former
employer

[Email](#) [↗](#)

Academic Credentials

- 2006–2010 **Doctorate** in Computer Science (**Algorithms**) Magna Cum Laude
Technical University of Munich **TUM**, Germany
Thesis: Counting in the Jacobian of Hyperelliptic curves
- 2003–2005 **Master** of Technology, Computer Science 8.9 / 10
IIT, Indian Institute of Technology, Madras, India
Advanced algorithms, algebra, ...
- 1999–2003 **Bachelor** of Technology, Computer Science First Class
University of Calicut, India
Introduction to CS, Algorithms, Compiler Design, ...

Professional Experience

- March. 2016–**The Code Artist - Consulting and Development** Mainz, Germany
Chief Consultant, Code artist, Dev Ops

GO BACKEND (RESTFUL API) SERVER: Currently designing and building a backend server in Golang, for processing large files. (eg. large image processing, project report generations involving multiple roles, stages, statuses, etc.) (Continuous delivery)

Go (Golang), CircleCI, AWS, S3

CLI TOOL FOR IOT DEVELOPERS: A Golang + Rails project where the client (go) communicates with server (rails) for code generation for embedded iot devices. The server handles request and processes them in queues (sqs/job-handling/etc.)

The tool (Go) interacts with the server purely via RESTful API.

Closely worked with client through every step of the project with short sprints.

Ruby on Rails, Go (Golang), CircleCI, Heroku, S3, SQS

C++ REVENUE MANAGEMENT Concluding a six month project on a revenue management simulation for global logistics firms based in Europe. The challenges include processing huge amounts of data combined with revenue management optimisation and inventory control. Various **optimisation algorithms** are employed to maximise the revenue.

The core, process heavy part of the project is implemented in C++, which interacts with R for user interactions.

C++, R

PROCESS OPTIMIZATION TOOL: A tool to solve an NP-Hard problem on process optimisation, using **genetic algorithms**. Can be run as daemon or directly from CLI, to accept the dependency graph of the problem to produce the optimal solution to the optimisation.

Go (Golang), Python

The key-points here are the sophistication of the base algorithm and the optimisation growth using clever mutations through generations in **genetic algorithm**.

KENKEN IN PYTHON Working on a pet project to implement the game/puzzle **Kenken™** for arbitrary levels of complexity, in a flask-server to be served for a frontend-mobile-app via RESTful api.

Python, Flask, RESTful

GAMIFIED E-LEARNING: A white labelled, gamified Learning app was built for companies which want to train employees & partners.

The app contains lessons being served from the server, and gaining points & premiums by correctly answering quizzes.

Lead a team of four developers, Created the whole server infrastructure with dev/staging/production environments – with possibility to change the domains without a new release.

Ruby on Rails, React Native, JavaScript, AWS, Heroku, CircleCI

<https://itunes.apple.com/us/app/mavenport/id1174847132>

<https://play.google.com/store/apps/details?id=de.mavenport.app>

MARKET-PLACE SERVERS: Working on building a Marketplace platform for covered by NDA in German market, for a Berlin based startup. The platform is a pure (responsive) web app, which combines elements of gamification.

Ruby on Rails, JavaScript

OAUTH/SSO BACKEND SERVERS: Completed a mobile app project with premium shop

1. Built and recruited the development team with on-site and remote members
2. Designed and created the technical architecture for OAuth/SSO/App servers
3. Created the backend (RESTful API) systems (RoR and Go) with provisions for SSO/Auth/App related separate services.

3 Servers: Ruby on Rails, 1 server: Golang, App Development in Java & Swift, Dev Ops

DESIGN, BUILD, AND LAUNCH: Analyse/Evaluate the clients' needs and desires, generate the best fitting design and model. Assemble technologies for the optimal MVP, mentor the team-members to build it, and transform the concept into pure binary. Utilise the expertise in Dev Ops to get the product placed in the stores and the cloud.

2014–2016

Lead Developer / Senior System Architect - Gamified apps
Mainz/Frankfurt, Germany

STARTUP: Play pivotal role in making decisions in a fast paced atmosphere of a startup.

- Determine the architecture and approach for the projects.
- Make release plans/schedules.
- Decide the methodology and confirm that the standards are met.

MENTOR/TECH-LEAD: Guide and assist the members of the team to achieve their potential

- Interview candidates; On-board the new hires.
- Assure the optimal performance of local and remote teams, by delegating relevant tasks.

ALL ROUNDER: Have been managing and performing in both cross functional, and remote teams; Get involved in developing new features in all spectrum of technology in the firm, as required.

Delta Sync: Played leading role in the team to develop means to speed up (10x) the sync of game elements between the apps and the server.

Season Migration: Designed and deployed seamless transition of the apps via gamified seasons.

LEAD DEVELOPER: Owner of the Rails/Go platforms, gamification engine, API end points, and the web portal. (PostGIS, Ruby on Rails, Golang, {Java/Coffee} script)

MAPTO{BIKE, HIKE, SNOW}: Maintain the existing, and develop new features in the scalable, extensible platform with which multiple apps communicate. The server handles the calculations for the gamification elements - geo-tracking, contests, quests, etc.

GAMIFIED BATTLES: Designed, developed and deployed private battles from inception. In battles, users challenge each other and compete in their own sports. Implemented battle invitations to attract new users.

ADVENTURES: Designed the action based adventures for the apps, and led the app teams as well as developing the server.

DEV OPS: Manage AWS based server/deployment.

2010–2014

Security Developer - *Communication archival*

Karben, Germany

EMAIL IMPORT/ARCHIVE SOFTWARE: Spearheaded the development of multi-tiered software to collect/convert various forms of communication to MIME. With the huge quantity of the data, the project was enjoyably challenging from technology as well as algorithms perspective. (C#, Reporting Tools)

FOUR STAGE LINUX SECURE BOOT: With trusted GrUB as framework, created a secure, four stage, Linux boot system which uses a Trusted Platform Module, and RSA signature verification. (C, Assembly)

ENCRYPTION LIBRARY WITH OPENSSL BIO: Designed and developed an encryption library with configurable layers of encryption and compression - generating digests on the fly, for future integrity checks. (C, C++)

POSTGRESQL MIGRATION BYPASS: Scalable tool for migration of old psql databases to new PostgreSQL, with drastic reorganisation of the data - with provision for plugging in various data conversion methods, for individual fields. (C)

SIDE PROJECTS: Glibc iconv + pInvoke for global encoding converter for Net; Standalone DRM decryption server; XSLT bypass for Amazon S3; Lotus Notes to MIME converter tool; etc. (C#, Java, C, Lotus-script)

2006–2010

Doctoral Candidate

Munich, Germany

Technical University Munich *Chair for Efficient Algorithms*

RESEARCH & DISSERTATION: COUNTING IN THE JACOBIAN: Researched on methods to speed up the computation of cardinality of Jacobian of hyperelliptic curves. The counting was aimed at finding the feasibility of building a secure cryptosystem using the curve.

The research concentrated on primordial group order counting and baby-step, giant-step methods. Also taken into consideration were the chances of different attacks based on the size of jacobian. New, improved algorithms were designed and a system was implemented in order to do faster computations. (Python, SAGE, Magma, Maple in Linux/Solaris).

http://www.mayr.in.tum.de/personen/sadanand/local_files/phdthesis.pdf 

TEACHING: ALGORITHMS, DATA STRUCTURES, PYTHON: Taught and assisted in courses on Algorithms, Data Structures, and Introduction to Computer Science in various semesters. Completely designed and conducted the course “Python for fine programmers” in two semesters.

<http://www.mayr.in.tum.de/personen/sadanand/#Teaching> 

MS-CRM, MS-OFFICE INTEGRATION (FIRST VERSION OF OFFICE/CRM PLUG-INS): Was the sole designer and developer in bringing MSCRM features to MS Word and MS Excel, using visual studio tools for office. The work included building GUI for MS-Office, which used Platform-APIs to communicate with CRM. (C#)

DUPLICATE DETECTION: Initiated the duplicate detection module design for Microsoft Dynamics CRM 4.0. The task consisted of investigating SQL stored procedures for digest generation for records. (C#)

Honours and Selected Publications/Talks

- “Gamification im Outdoorsport” – talk at [WebMontag Frankfurt, June 2014](#) ↗.
- “Lightweight Cryptography: Hyperelliptic Curve Cryptography” – Talk at “**Bundes Sicherheits Institut**” (National Security Institute), Bonn, Germany. (March 2009)
- German Research Foundation (DFG, Germany) scholarship for doctoral studies.
- German Academic Exchange Service (DAAD, Germany) scholarship for master thesis.
- 99.68 percentile score in GATE (Graduate Aptitude Test); All India Rank of 118 in GATE (IIT).
- Best project in Bachelors (2003) – “Porting Valgrind from Linux to FreeBSD” – Computer Science & Engineering, Government College Thrissur, India.
- “Process Tracing using `ptrace` - Parts I, II and III”, Linux Gazette: Issues 81, 83 & 85. (<http://www.tldp.org/LDP/LG/issue85/sandeep.html> ↗)
- “GCC-Inline-Assembly-Howto” (2002) (<http://ibiblio.org/gferg/ldp/GCC-Inline-Assembly-HOWTO> ↗)
- “Valgrind Howto”, (2003) (<https://www.linux.com/learn/docs/ldp/807-Valgrind-HOWTO> ↗)