Jan Schill

Email : janschill@proton.me
Mobile : +45-27620100

Education

- **IT University of Copenhagen** — Copenhagen, Denmark
  Master of Science in Computer Science — *Sep. 2019–Jun. 2021*
- **Hochschule Flensburg University of Applied Science** — Flensburg, Germany
  Bachelor of Science in Media Informatics — *Sep. 2015–Aug. 2019*

Experience

- **Zendesk** — Copenhagen, Denmark
  Software Engineer — *Aug. 2021–Present*
  - **BYOK Service**: Java Service to enable *bring-your-own-key* for Zendesk services.
  - **Ruby Library Distributed Authentication**: Develop a Ruby library for company-wide authentication/authorization usage.
  - **Ruby Starter Kit**: Establish Ruby development guidelines for an organization of +1000 engineers.

  Student Developer — *Oct. 2019–Aug. 2021*
  - **Ruby on Rails**: Incrementally upgrade Rails; develop in large Rails application.
  - **Dual-Boot Framework**: Implement a dual-boot tool to run two versions of Rails parallel in testing pipeline.
  - **AWS RDS**: Find a strategy for resilience in unhealthy database clusters.
- **CERN** — Geneva, Switzerland (remote)
  Student Researcher — *Aug. 2020–Jun. 2021*
  - **CERN-Solid Code Investigation**: Exposure to the code of very popular, complex, well-designed and documented applications of today and tomorrow. Work with interesting technical challenges and expert software developers. Gain in organizational expertise.
- **visuellverstehen GmbH** — Flensburg, Germany
  Full Stack Web Developer — *Jan. 2018–Sep. 2019*
  - **PHP with Laravel**: Custom web application solutions using PHP and the Laravel framework.
  - **Vue.js**: Single Page Application development with frontend reactive framework Vue.js.

Projects

- **Java BYOK Service**: Bring Your Own Key is an encryption technique that allows the encrypting and decrypting of data with externally held encryption keys. This guarantees that plaintext data cannot get into the wrong hands. This project is still ongoing and I have so far set up the Kubernetes deployment of a few coupled services in a Service Mesh environment; composed a solution design for a key rotation and a backfill process; all observability requirements. The day to day work involves programming new features in the Java service.
- **CERN-Solid Code Investigation**: As part of my Master's thesis I worked on a code investigation with CERN and Solid. This research aims at observing benefits and drawbacks on the integration of the Solid specifications into a large and sophisticated software system. Solid is a web decentralization project led by Sir Tim Berners-Lee and was initially developed at the MIT. The idea is to change how web applications work with data, resulting in data ownership fully controlled by the user/content creator. The investigation is carried out in collaboration with CERN. CERN and the Solid project share a lot of common values and is therefore interested in the success of the decentralization project.
- **Compiler Optimization for DOM Diffing**: As a university project a fellow student and I created a functional language with a compiler written in FSharp. The compiler had optimizations like partial evaluation and symbolic execution to improve the DOM diffing algorithm. The idea was in MVC web applications a lot of resources are wasted on comparing two versions of the vDOM when an update occurred. This comparison could be done during compilation of the web application written with our own language and framework. It would compile to JavaScript and instead of doing any diffing in the frontend it would have the change attached to the event producing the change.

Programming Skills

- **Languages**: Ruby, TypeScript, Java
- **Technologies**: Ruby on Rails, Linux, Docker, Kubernetes, DigitalOcean, AWS, Git, Datadog