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 II — *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.
  - **Security Champion**: Threat modeling of our service; foster a security mindset within the team; manage a cryptography meet-up.
  - **Reliability Champion**: Advocate good practices for logging, monitoring, and alerting in the team; define SLAs, SLOs, and SLIs together with stakeholders

  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 encrypting and decrypting of data with an externally held encryption keys. This gives the one holding the key the control over confidentiality on their data. First the project had to be designed as a Java service providing BYOK to Zendesk's vast product ecosystem without substantially comprising request latency of the around 250,000 requests per second to Zendesk's Edge layer. A few implementation details done by me are: *Dockerizing*, configuring deployments for our Istio service mesh and setting up a continuous integration pipeline; setting up and defining logs, metrics and tracing with DataDog monitors attached to commit to service level agreements; compiling decision records on customer key rotation and backfill of existing data; development of several modules in the newly built Java encryption 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.

Programming Skills

- **Languages**: Ruby, JavaScript, Java, Python
- **Technologies**: Ruby on Rails, Vert.x, Linux, Docker, Kubernetes, Service Mesh, GitHub Actions, DigitalOcean, AWS, Git, DataDog, Sentry