Adrien **Faure**

Computer Engineer, PhD

+33 6 73 18 80 27 adrien.faure@protonmail.com adfaure Grenoble, France

SUMMARY OF WORK EXPERIENCES

Lead Developer	Université Grenoble Alpes	2021 – 2023
Architect and fullstack developer	Aktid	2022 – 2023 (3 months)
Technical Writing	Datadog – ResearchToBusiness	2024 (5 months)
PhD Candidate – Engineer	Atos – Université Grenoble Alpes	2017 – 2020 (3 years)
Teacher	Université Grenoble Alpes	2017 – 2023 (~120h)

TECHNICAL SKILLS

Languages	**Python**, **Rust**, C/C++, SQL, Bash, R, Nix, Java, Go, Javascript, Typescript, Dart, PHP
Framework	**FastApi**, **React**, **Angular**, Spark, Spring, Flutter
Libraries	**pandas**, **numpy**, **tidyverse**, zmq, asyncio, protobuff, thrift, MPI
Environments	**Linux**, AWS, Kubernetes, HPC, Cloud, Big Data, NixOS, debian

FUNCTIONAL SKILLS

Tools	**Git**, Docker, Latex, Nginx, Traefik
RGBD	**Postgres**, Redis, sqlserver, sqlite, mongodb
Methodologies	**Scrum**, Pomodoro, Agile
Documentation	**Documentation**, **Tutorials**, **Talks**, Courses, Scientific Publications, Explanatory Diagrams
Concepts	**Computer architecture**, Data modeling (UML), Distributed systems, Scalability, Parallelism, Asynchronism, Data pipeline, DevOps, SRE (Site Reliability Engineering), Monitoring, Continuous Integration (CI), Continuous Deployment, Testing (unit, integration), Design patterns, Scientific method
Soft Skills	**Autonomous**, Sociable, Curious, Critical thinker, Open to criticism
Langues	**French**, English

EDUCATION

PhD in Computer Science	**2020**	**Grenoble**, France
Master's in Computer Science	**2016**	**Lyon and Grenoble**, France
Bachelor's in Computer Science	**2014**	**Lyon**, France

PROFESSIONAL EXPERIENCES

Université Grenoble Alpes — Lead Developer

Engineer – PostDoc — 2021 – 2023

> **Context**
>
> Two fixed-term contracts within the computer science laboratory at Université Grenoble Alpes (UGA). I worked on a European project involving both private and public actors across Europe. My primary role was mainly focused on the development of an open-source software called OAR3.
>
> OAR3 is an HPC platform management software. It's a complex and distributed software, involving:
> - Large-scale resource management (scheduling thousands of programs),
> - Fine system manipulations for parallel program execution (via Linux cgroups and namespaces),
> - And a user interface in CLI and REST API.

Objective

- **The main objective is to make OAR3 production-ready**
- Address the needs of the European project
  - Control and improvement of scheduling policies for energy management
  - Prepare and present working POCs at scale (deliverables for the European project)
- Participate in laboratory and research team activities
  - Supervising interns
  - Assisting with research experiments
  - Writing scientific papers

Work Done

- **Lead Developer on OAR3 in python3**
  - Development of new features
  - Code quality improvement
    - Refactoring of significant modules (e.g., reimplementing the REST API with FastAPI)
    - Comments on obscure parts
    - Writing tests
  - Ensuring continuous integration with GitHub Actions and maintaining Nix and Debian packages
  - User support:
    - For administrators when setting up a new cluster
    - For users through the addition of new features or debugging
  - International collaboration (Barcelona, Munich, Bologna) for tool integration
  - Development of satellite tools for OAR3: (monitoring, benchmarking, deployment, etc.)
- **Mediation, Documentation:**
  - Documentation: user guides, getting started, diagrams and schematics, reference API
  - Writing tutorials for engineers and researchers:
    - On OAR

  - On Nix: https://nix-tutorial.gitlabpages.inria.fr/nix-tutorial

Results:

- The OAR3 project has been put into production (beta testing at the time of my departure) on a new cluster
- Demonstration of functionalities via a video recording the management of a platform with OAR3 (delivered for a European project)
  - Integration of several tools participating in the REGALE project: monitoring tool, deep learning application
- Scientific contributions

Languages	**Python**, Perl, Nix, Rust, R, C/C++
Tools	**FastApi**, **Linux**, github actions, pandas, numpy, scikit-learn, docker, docker-compose, PostgreSQL, asyncio, MPI
Conceptes	Distributed Systems, Infrastructure Management, Scheduling, Research, Monitoring

Aktid — Architect and fullstack developer

Freelance 2022 – 2023 (3 months)

> **Context**
>
> Aktid is a company that designs and builds sorting centers for waste recovery. As part of R&D work on predictive maintenance artificial intelligence, Aktid needed an architect and developer to create a predictive maintenance dashboard from scratch.

Work Done:

Complete architecture of a new predictive maintenance platform
- Resumption of research work to design an on-site implementation

Design of the *backend*:
- Asynchronous online prediction service (asyncio)
- Design of the data schema for storing predictions
- Python RestApi for the *frontend* part

Design of the *frontend*:
- Dashboard in ReactJs
- Display of online predictions with ReCharts

Deployment in production:
- Production deployment in a factory for a beta version
- Using docker-compose technology on a Debian server

Documentation and training:
- Writing necessary documentation for takeover by other teams
- On-site tool presentation and training

Results: Deployment of version 1 of the predictive maintenance dashboard for Aktid's sorting center factories. The goal was to complete this project within one month to fit my availability. The development was continued by another developer under my guidance.

Languages	Python, Typescript, SQL
Tools	FastApi, React, Linux, pandas, numpy, scikit-learn, docker, docker-compose, Postgres, asyncio
Conceptes	machine learning, predictive maintenance

Datadog – ResearchToBusiness Technical Writing

Freelance 2024 (5 months)

Context

As a freelancer, I collaborated with ResearchToBusiness to develop CIR (Crédit d'Impôt Recherche) documents for the company Datadog. The CIR is a French R&D tax credit program that encourages companies to invest in research and development. I worked on these files for Datadog, a leading company in trace collection and cloud infrastructure monitoring.

Achievements:

- **Technical Study and Analysis:**
  - Reading and understanding internal technical documentation of Datadog.
  - In-depth analysis of event architectures, notably the Husky event store and their distributed trace API.
  - Cross-referencing technical information with the state of the art in distributed and scalable infrastructures.
- **Interactions with Technical Teams:**
  - Interviews in English with Datadog teams to deepen the understanding of research concepts.
  - Clarification of technical details and the R&D approach used by Datadog.
- **CIR File Writing:**
  - Synthesizing the collected information to write CIR files.
  - Highlighting technical achievements and innovations in Datadog's R&D projects.
  - Emphasizing the technical and conceptual aspects of the projects to meet CIR criteria.

Conclusion:

This experience demonstrates my ability to navigate complex technical environments, synthesize detailed technical information, and write structured and relevant technical documents for research and development initiatives.

Tools	Google docs, Kubernetes, Kafka
Concepts	Scientific Writing, Scalable Architecture, Distributed Computing, Kubernetes, AWS, Kafka

Atos – Université Grenoble Alpes PhD Candidate – Engineer

CIFRE Thesis 2017 – 2020 (3 years)

> **Context**
>
> Industrial PhD, between the Grenoble Informatics Laboratory (LIG) and Atos. My research work is in the HPC (High Performance Computing) branch of Atos (formerly BULL), specifically on task schedulers for supercomputer-type clusters.

Thesis Topic(s): The subject of my doctoral studies revolves around HPC platform schedulers. These are complex and distributed software at the core of platform usage. Their central role makes them both an important study subject (inefficient schedulers impact the entire platform, which is costly) and difficult because it's nearly impossible to replace these software for studies without affecting their functioning.

My main contributions are:
- Methods and tools for evaluating the performance of HPC schedulers.
- Scheduling algorithms for HPC schedulers with task redirection.
- Use of Nix for reproducible computing experiments.

Work Done:
- Development of tools for emulating distributed applications
  - Distributed and parallel execution engine: experiment orchestration, parallelization with Spark
  - Design and development of scheduling algorithms in Rust
  - Programming parallel applications in MPI
- Data analysis in Python and R, using pandas, Jupyter, tidyverse, and rmarkdown.
- Writing scientific papers in English
- Supervision of interns
- Teaching courses as a lecturer

Results:
- Awarded the title of Doctor from Université Grenoble Alpes
- The thesis is available online – https://theses.fr/2020GRALM056.

Languages	**Rust**, **Python**, **R**, C/C++, Scala, Go, Bash, Nix
Tools	**Linux**, **Spark**, Nixos, Tidyverse
Conceptes	**Parallel Programming**, Distributed Systems, Scheduling, Critical Thinking, Writing, Synthesis, Scrum Methodology

Université Grenoble Alpes — Teacher

Adjunct Professor 2017 – 2023 (~120h)

Context

Working in the laboratory is an excellent opportunity to find teaching assignments at the university. From 2016 to 2023, during my time at UGA, I completed several adjunct positions.

List of Positions Held:

- Module Coordinator IUT2 – Software Vulnerabilities (1 semester)
- Module Coordinator IUT2 – Software Quality (1 semester)
- Master's 1 TP Instructor – DevOps (1 semester)
- L3 MIAGE TD Instructor – Statistics and Visualization (1 semester)
- L3 Info TD Instructor – Advanced Algorithms (2 years)

Languages and Tools	Java, Python, R, Docker, Docker-compose, Git
Concepts	Pedagogy, Autonomy, DevOps

Atos R&D Intern

Master's Internship 2016 (6 months)

i Context

End-of-studies internship at Atos. Internship within the R&D department, at the heart of Atos Echrirolles' High Performance Computing activity.

Objective: Build a distributed system emulator by transparently replacing the libc with a simulation library. The emulator was named simunix.

Work Done:

Development of a C library simulator: The simulator executes a distributed application (resource manager type) in an isolated environment. The distributed software runs, without realizing it, on a single machine instead of being deployed at scale. The project works through a re-implementation of the libc, and mechanisms for rewriting binary files. Here is a list of functions that I had to rewrite:

- Blocking system functions: Mutex, CondVar.
- Time management functions: sleep, gettimeofday.
- Process management functions: fork, execv, etc.
- Network management functions: socket API (send/recv, connect/accept).

Result:

- Emulation of a distributed system on multiple nodes (30 by the end of the internship) on a single machine.
- Presentation of simunix at the Slurm user days (https://slurm.schedmd.com/SLUG16/slug16_simunix.pdf).

Languages	**C++**, **C**, Python
Tools	**Linux**, **libc**, Thrift, Protobuf, ZMQ, Simgrid, Slurm
Conceptes	Linux system programming (sockets, threads, mutex, etc.), Emulation, Scheduling, Scrum Methodology

Wizacha — Web Development Intern

L3 Internship 2014 (3 months)

> **Context**
>
> End-of-degree internship at the startup Wizacha. In 2014, Wizacha was a startup in the field of marketplaces. The concept was to allow sellers to use the platform to sell their products.

Objective: Participate in development activities within a team using the Scrum methodology.

Work Done:

- JavaScript development of the user cart
  - Backend part in PHP and FrontEnd in JavaScript
  - Writing associated unit tests
- Participation in sprints using Agile methodology
- Use of Mercurial for version control

Result:

- User cart in production
- Learning the fundamentals of the Scrum methodology

Languages	**Javascript**, **PHP**
Tools	**Linux**, **Mercurial**, jQuery
Conceptes	**Web Programming**, Scrum Methodology