# Bram Geron
## Backend / systems engineer

bram@bram.xyz | Berlin | github.com/bgeron | linkedin.com/in/bgeron

2020–present — Mobile app for small group productivity with AI (app name under NDA)

- Led strategy and implementation of the data and telemetry layers (**Rust**/**PostgreSQL**/**gRPC**; backend, client, and ORM). Big part in implementing FFI for our first React client (and later our Swift client). **Typescript**ified the web client, and extended it to mobile with Cordova. Led implementation of our pre-LLM rules-based AI.
- Various end-to-end features, e.g. extended AI suggestions to ingested images, and enabled our AI to access Perplexity for up to date answers. Prompt engineering.
- Envisioned and built deser-incomplete, an open source Rust library to parse incomplete JSON as it is streaming.
- Significant work on CI/CD and developer experience (**Kubernetes**, docker-compose)

2019–2020 — Backend and SQL engine at Contiamo

- Development of the **REST** API (**Scala**, **Java**) in conversation with the front-end stakeholders. It became a conversation because of me; before, it was a one-way street.
- Led development of our **JSON Schema**, which allowed the frontend team to autogenerate bindings and increase productivity.
- Extended and improved on the Calcite-based SQL query rewriter to add support for more SQL databases and data types.
- Integrated our product with many third-party clients like Tableau, by implementing **PostgreSQL's wire protocol**.

2014–2019 — **PhD in theoretical computer science** at University of Birmingham → *www.bram.xyz/thesis*

- Proposed and studied a novel form of program modularity based on coroutines/effects.
- Compared to existing approaches, my variant uses lexical scoping to categorically rule out certain categories of bugs, as shown with formal proofs on denotational semantics.
- Helped teach algorithms, OCaml, and Haskell modules.

2016 — Internship at Google on V8, the Javascript engine in **Chrome** → *www.bram.xyz/cv/code*

- Built a store-store elimination reduces output code size, and improves CPU i-cache hit rate.
- This helped Turbofan get good enough that the old compiler could finally be retired, removing a large amount of technical debt.
- Countless usability improvements in the Javascript visualiser for the intermediate graph structures, so increase everyone's understanding of what happens internally.

2007–2013 — Masters in Computer Science (Eindhoven University of Technology, cum laude), specializing in algorithms, security, programming languages, formal modeling and verification.

- 2009-2012: first line worker at university tech support
- Lab help for intro to embedded systems
- Outreach for high school students
- Treasurer at the student computer club
- President of the student dance club

2000 Primary school: wrote polling station software in Basic for school elections.
2001: Rewrite in Pascal
2004: Toy PHP web application framework
2006: "LispCMS" Common Lisp web application framework
2007: Toy Lisp dialect with an interpreter. Finished part of a compiler into the Perl 6 VM. Fixed bugs in the Perl 6 VM.

## Skills

| | |
|---|---|
| **Languages** | Rust, Swift, Scala/Java, Python, SQL, TypeScript/JS, Go, C/C++, bash/fish/nushell, Haskell, OCaml, Common Lisp. |
| | English (near-native), German (advanced), Dutch (native). |
| **Tools and techniques** | PostgreSQL, gRPC/OpenAPI, Bazel, GitLab CI/CD, OpenTelemetry/Jaeger, gRPC, k8s/Docker, nginx/Apache, Git/Jujutsu, PEG, WASM, Linux |

For coding samples and more details:

**https://github.com/bgeron**
**https://bram.xyz/cv/code**