

# Evgenii Petrov

## Summary

I am a software engineer with a lot of production experience.

- I love reading and writing, be that code or designs or proposals.
  - Ask me about books!
  - Tell me about books!
- I love digging deep to figure out why things are the way they are.
  - As in [digging into CPython implementation](#) to find out why self-referential tuples are printed a specific way.
  - As in digging into Linux kernel internals to figure out how CPU accounting works, along the way reading Linux Kernel Development by Robert Love, System Performance by Brendan Gregg and The Linux Programming Interface by Michael Kerrisk.
- I am familiar with Python, Java and Go.
- I am curious about Clojure and OCaml, and FP languages in general.
- I am curious about theorem proving and formal specification.

## Professional experience

### Senior Site Reliability Engineer at Bloomberg LP, London, 2021 - current

Worked as an SRE in a team transitioning from operational work to engineering work.

- Researched reliability/velocity trade-offs, wrote proposals, worked with engineering and product stakeholders to get approval and worked on implementing those proposals that got approval. As part of the proposals I created (naive) Monte Carlo simulations in Python based on subject matter expert's quantified uncertainty.
- Investigated technology choices appropriate for the skill set of people on the team to increase feature velocity, proposed and led relevant migrations. One such choice was to migrate a UI written in TypeScript/React on the frontend with Python on the backend to Python with HTMX on the backend.
- Provided thorough code and proposal reviews with detailed reasoning for suggestions, which helped both junior and senior members of the team to learn and improve.
- Noticed incompatibility in how different teams conceptualise a common system and came up with a shared lens that allowed the teams to agree on a path towards improvement.
- I was on-call for critical production services (running on Linux), wrote postmortems for outages and participated in production readiness reviews.

### Senior Site Reliability Engineer at Arrival Ltd, London, 2019 - 2021

Worked as a first SRE on a rapidly growing engineering team.

- Established production and staging environments (Linux on Kubernetes on AWS) and patterns of use for both.
- Pushed for and set up on-demand deployments, which enabled rapid iteration on a multitude of code bases.
- Established SLO, monitoring and alerting practices, evangelised the use of metrics for production observation.
- Established oncall, incident response and postmortem practices.

- Ran "disaster roleplay" sessions, to get people familiar with documentation, tools and practices relevant to oncall.
- Wrote extensive documentation on various systems relevant to production: "life of a request", "life of a container image", "life of an alert".
- Mentored junior members of the team.
- Pushed for convergence of execution environments (Linux on AWS Batch vs Linux on Kubernetes), which enabled sharing of experience between R&D parts of the team and software engineering parts of the team, and enabled easier integration of R&D developments into core systems and products.
- Contributed pull requests to [kubernetes/autoscaler](#) and [google/cloudprober](#).

## Site Reliability Engineer at Google LLC, Dublin/London, 2014 - 2019

I was SREing for an Ads Quality team in Dublin, and a Cloud team in London.

- I noticed discontent with configuration used by ~100 people, designed a refactoring, implemented refactoring tools and executed the refactoring affecting ~2000 configuration files spanning ~100000 lines of config.
- I evaluated alternative paths for migration from a deprecated data store and designed the overall structure and critical components of the migration plan. I also executed parts of that migration plan.
- I wrote libraries, tools and automation that made SREing easier both for teams I was part of as well as other SRE teams at Google.
- I was on-call for critical production services (running on Linux), wrote postmortems for outages and participated in production readiness reviews.

## Open source projects

### Proof Checking Euclid

[Proof Checking Euclid](#) is a study in making formal proofs written by expert mathematicians easier to understand for people without the relevant expertise.

I've studied [GeoCog](#), untangled the implicit patterns in it and wrote tools to make implicit parts of the proofs explicit.

I learned a lot about Coq, Parsing Expression Grammars and Prolog, as well as the underlying mathematical concepts and approaches to proofs.

### Zabby

[Zabby](#) is a monitoring agent intended to be a drop in replacement for Zabbix agent.

Writing Zabby have immensely increased my knowledge of Python, unit testing, UNIX daemons, Linux proc filesystem, large scale production deployments (deploying to several hundred servers) and backwards compatibility.

I gave a [talk about zabby](#) at Zabbix Conference 2013.

## Contact information

- Email: [blinunleius@gmail.com](mailto:blinunleius@gmail.com)
- Place of residence: London, United Kingdom