

# Dimitrije Radojević

[dimitrije.website](#) 🌐 | [me@dimitrije.website](mailto:me@dimitrije.website) ✉️  
[dimitrijer](#) 🐙 | [@dradojevic](#) 🐦 | [dimitrijer](#) in

March 7, 2021



## Curriculum Vitae

(this page is also available for [download in PDF](#))

I am a software engineer with more than eight years of professional experience in backend architecture and development, video game development, infrastructure and DevOps, machine learning engineering, and, most recently, Site Reliability Engineering.

My diverse background provides me with deep knowledge of computer systems, from low-level hardware intricacies to high-level software abstractions, and from user-facing websites and apps to number-crunching processes tirelessly working behind the scenes. This, along with the ability to quickly learn new key concepts and effectively map them to what I know, gives me unique versatility. Over the years I have come to appreciate this as my biggest strength.

I enjoy designing and developing distributed, fault-tolerant and highly concurrent architectures. My passion is to find concise, elegant, maintainable and performant solutions, and build them with the right tools for the job. I have a lot of those under my belt, having worked with multitude of well-known and widely used technologies.

With a knack for explaining and teaching, and a big pool of patience and understanding, I love sharing my knowledge and helping out. At the same time, I communicate clearly and efficiently, and thrive in challenging environments where I, too, can learn and grow. I am sensitive, compassionate and warm to others, with a strong sense of justice and fair play. All of the above makes me a pretty good team player.

My hobbies include playing video and board games, reading science fiction and fantasy, toying with SoC boards, riding my bike - and doing all of this in the wonderful company of my Shih-Tzu Mochi.

## Experiences

- January 2020 - present: *Systems Engineer, SRE*, [Google](#)
- March 2018 - October 2020: *Machine Learning Engineer*, [Nordeus](#)
  - Part of ML/AI team with the mission of using bleeding edge machine learning algorithms for building unique and novel features in our games.
  - I was responsible for integrating proof-of-concept and prototype projects developed within the team with actual production infrastructure, while maintaining required level of performance and quality.

- ▶ Also tasked with establishing and maintaining good engineering practices, without considerably hindering velocity of the research part of the team.
- ▶ Designed, built, configured and maintained a machine-learning pipeline, which included data mining, multi-stage preprocessing, human-in-the-loop review process integration, feeding reviewed data to a machine learning algorithm, and automating testing and algorithm performance evaluation with regards to code changes.
- ▶ Administered GPU cluster used for training and inference.
- ▶ Applied classical AI algorithms, such as behaviour trees and utility functions, and reinforcement learning to develop several iterations of AI bots for [Heroic: Magic Duel](#).
- February 2012 - March 2018: *Software Development Engineer*, [Nordeus](#)
  - ▶ Worked as a video game developer on four different game projects, including Nordeus' most successful game, [Top Eleven](#).
  - ▶ Supported development of various game features on both backend and frontend side.
  - ▶ Designed and developed distributed login service that at its peak had ~7 million unique daily logins.
  - ▶ Was a part of the effort to convert a legacy, monolithic backend service to microservice architecture.
  - ▶ Designed and built a proxy that allowed WebSocket frontends to communicate with backend services in proprietary binary protocol. This service eventually became the edge service for all clients.
  - ▶ Introduced several significant concepts to backend teams over the years, including key-value stores, message queues, observability in terms of ubiquitous metrics and distributed tracing, modern build tools and CI/CD pipeline, continuous code quality and code health metrics and centralized linting.
- October 2009 - May 2014: *Junior Teaching Assistant*, Seminar of Astronomy, [Petnica Science Center](#)
  - ▶ Taught basics of programming to high-school students at the seminar, i.e. introduction to programming with C and Python, version control, working in terminal, writing data mining scrapers etc. Petnica Science Center is an independent and nonprofit organization for extracurricular, formal and informal science education.

## Tech Skills

- Programming languages, in order of proficiency:
  - ▶ **Java** - full professional proficiency, and my language of choice. I did most of my backend development in Java.
  - ▶ **C#** - full professional proficiency. Most of my experience with C# comes from scripting in Unity game engine.
  - ▶ **Python** - full professional proficiency. My favorite quick and dirty scripting language. Used it for developing small backend services, command line tools, machine learning and rapid GUI development.
  - ▶ **Clojure** - high proficiency.
  - ▶ **C/C++** - high proficiency.

- ▶ Haskell, Go - basic proficiency.
- Versatile and under-the-hood knowledge of UNIX systems, especially GNU/Linux: usage, configuration and administration.
- Shell scripting, and broad knowledge of a plethora of command line utilities for manipulating and transforming data.
- Networking: understanding of L3/L4 protocols like TCP/IP and UDP, and higher level protocols like HTTP, HTTPS, HTTP/2, WebSockets, QUIC, DNS etc. Knowledge of low-level networking facilities like sockets, blocking and non-blocking syscalls (`select` and `epoll`), experience with well-known and widely used networking libraries, such as Netty, Jetty, requests, Flask, ring, compojure and zeromq, and debugging tools like `ss`, `netcat`, `iperf`, `tcpdump`, Wireshark etc. Some knowledge of network stack and its optimization on Linux systems.
- Multithreading primitives like mutexes, semaphores, locks, conditions, and high-level abstractions such as futures, promises, coroutines and async programming.
- IPC mechanisms such as pipes, shared memory, UNIX sockets and the MPI library.
- Databases:
  - ▶ **PostgreSQL** - full professional proficiency. Thorough understanding of MVCC, locking, indexing, views, triggers, partitioning, stored procedures, as well as Postgres specific features: `plpgsql`, foreign data wrappers, materialized views, statistics tables (`pg_*`), advisory locks, access control, replication, backup, configuration, document data types, full text search support, functional indexes, GIN indexes etc. My database of choice, and one I used the most.
  - ▶ **SQLite** - high proficiency. My favorite embeddable database. If querying data is any more complex than performing a dictionary lookup, this is what I use.
  - ▶ **Redis** - high proficiency, used for things such as game leaderboards and caching.
  - ▶ Aerospike, CouchDB, Cassandra, MongoDB, memcached - basic proficiency.
- Machine learning frameworks and libraries, such as numpy, scipy, scikit-learn, Keras and Tensorflow.
- Build tools, such as Makefiles, GNU autotools, CMake, Maven, Gradle, Leiningen, and msbuild/xbuild.
- Observability tooling, such as Prometheus, Icinga and Zabbix for monitoring and alerting, Grafana for dashboards and Jaeger and Zipkin for distributed tracing.
- Provisioning with Ansible, vagrant and Docker.
- Editors and IDEs, in order of preference: Vim, Eclipse, IntelliJ IDEA, Visual Studio.
- Version control: Git, Mercurial, SVN.

## Education

- October 2009 - October 2016: *BSc in Computer Science*, University of Electrotechnics, Belgrade, Serbia
- September 2005 - Jun 2009: *High School Degree*, Mathematics Dpt., Gymnasium Užice, Serbia

## Languages

- English - full professional proficiency.
- Serbian - native proficiency.

## Publications

[1] M. Warchalski, D. Radojevic, and M. Milosevic, “Deep RL Agent for a Real-Time Action Strategy Game,” *arXiv:2002.06290 [cs]*, Feb. 2020 [Online]. Available: <http://arxiv.org/abs/2002.06290>