

ALESYA HUZIK

+61 427 990-909 ◊ mail@alesya.me

Sydney, Australia

EDUCATION

Belarusian State University of Informatics and Radioelectronics, Minsk Faculty of Computer Systems and Networks M.S. in Computer Science & Software Engineering	<i>July 2013</i>
Belarusian State University of Informatics and Radioelectronics, Minsk Faculty of Computer Systems and Networks B.S. in Systems Engineering	<i>July 2012</i>

CERTIFICATION

CO0401EN Beyond the Basics: Istio and IBM Cloud Kubernetes Service	<i>December 2019</i>
---	----------------------

HAVE EXPERIENCE WITH

Operating Systems	Linux (NixOS, Archlinux, Debian, CentOS, Ubuntu), Mac OS X, Windows (3.11 – 11), DOS
Programming Languages	Clojure, Ruby, C, Nushell, POSIX shell (bash, zsh), JavaScript, Common Lisp, Vala, Erlang, Rakudo, Perl, Maude, Factor, Lua, Terra, C++, Python, Golang and some others
Libraries and Frameworks	React, Reagent, core.async, malli, clojure.spec (+v2), prismatic/schema, jdbc.next, component, mount-lite, mount, Compojure, Kioo, Enlive
Markup and Typesetting	L ^A T _E X, HTML, Haml, Slim, CSS, SASS/SCSS, Less, Bootstrap
Cloud	AWS (EC2, S3, IAM, RDS, DynamoDB, Route 53, EKS, ECS), DigitalOcean, Cloudflare, Vultr, Linode, Heroku
Linux, DevOps, system administration, networking	Docker/podman, docker-compose, Kubernetes, Terraform, Consul, Chef, libvirt (kvm, libvirt-lxc), dokku, Nginx, Apache httpd, statsd, Postfix, Prosody, vsftpd, Squid, Corosync, Pacemaker, DRBD
Nix-based technologies	NixOS, nix flakes, Colmena, NixOps
SQL Databases	PostgreSQL, MySQL/MariaDB, SQLite
NoSQL Databases	Redis, Datomic, DynamoDB, Neo4j, OrientDB, MongoDB, Cassandra
Programming paradigms	Imperative, Object-oriented (class-based, prototype-based), Functional, Concatenative (stack-based), Logical (predicate logic, rewriting logic).

WORK POSITIONS

CIM <i>Senior Software Engineer</i>	25 September 2023 - 15 January 2024 <i>Sydney, Australia</i>
---	---

- Lead the effort to migrate our internal SDK to use AWS SDK for Java v2, so we can migrate from JDK 11 to JDK 21. Updated the SDK, tests, and a few core services to use it.
- Wrote a parser for a new meter data format

- Reworked parser resolution from "magic" 'requiring-resolve'-based to explicit one, so various clojure tools would see a proper namespace dependency tree and not a bunch of dead code namespaces.

Audience Republic

Software Architect

February 2023 - 31 July 2023

Sydney, Australia

- Redesigned url shortening for SMS by defining a linear space for them and using format-preserving encryption. This addressed the bottleneck in unique short URL generation for escalating SMS volumes and enabled concurrent URL allocation without collision issues by being able to maintain a pool of 1 million preallocated URLs for efficient, fail-safe usage.
- switched all production servers from manual long-term ssl certificates to Let's Encrypt wildcard certificates with automatic refresh and sync across servers
- Designed and built a campaign bot that allows to simulate user activity, with time travel, for an arbitrary number of users on campaigns made for demo and sales purposes. This allows the sales team to create campaigns that mimic what that company would have and populate it with reasonable-looking user activity, so potential customers can see stats, graphs with activity distributions (sharing via social media etc.), referral registration percentages spanning days and months etc.
- Embedded a code coverage library (cloverage) into the backend server request-handling code (non-production environment only), enabling direct code coverage measurement from UI-driven integration tests (which was the majority in our test suite). This approach revealed precise backend code paths activated during live interactions, significantly enhancing the precision and relevance of our test coverage analysis.
- While working on the automation project I created a dsl wrapper around our resource dsl that was significantly more readable. Once other engineers discovered it they asked for it to be moved into the core of the project as a primary interface for working with db entities, which we gladly did.
- Automation: new frontiers - search-based triggers. Automation became a popular part of the platform for customers but they wanted to have an ability to trigger certain actions once a user starts or stops to match certain criteria (the platform already had very involved functionality for building multistage filters and saving them as segments). I designed and implemented search-based triggers in a way that was performant and didn't affect the rest of the platform. This was crucial since tables used for filters are the largest (few terabytes) and, despite a lot of denormalisation and other optimisation work, some segments are so complex they still require several minutes to run (since customers are allowed to make them arbitrarily complex)

Audience Republic

Team Lead

July 2021 - February 2023

Sydney, Australia

- Leading the team working on the Automation project from inception to production release and beyond
- Sourcing, interviewing, hiring candidates for the automation team
- Training and mentoring new hires
- Building and iterating on processes for managing the team (I was the first team lead at the company, previously there were just developers under the CTO)
- Introduced JIRA in the company (previously a mix of Trello and Notion pages were used, JIRA started as a tool for my team and then became the main and only project management tool for the company)
- Work estimation, planning, tracking team progress
- Monitoring team health with 1-on-1 meetings, negotiating team needs with CTO and CEO (introducing paid leave and public holidays for engineers working remotely as contractors)
- CNAME project (allowing customers to use customer-titled subdomains for their campaigns)
- Wave (company's long shot experiment) – a mobile app for one-to-many communications of artists/promoters with their fans. I designed the backend with quick prototyping and future scaling in mind (messaging backend was based on the Matrix protocol and Synapse was used as a messaging backend) and setup

all necessary infrastructure. My team (plus a contractor mobile developer) built it. The project was released on Google Play and Apple AppStore, but was quickly deprioritised due lack of adoption.

- Migrated the project from an abandoned postgres.async sql driver to standard jdbc (and jdbc.next clojure library). Wrote tools to find all sql queries that rely on positional parameter numbers (postgres.async uses \$1, \$2 to refer to parameters, so unlike with jdbc they can be reused in a query), rewrote queries, added a shim layer that allowed both sql backends to coexist by transparently translating old-style queries into the new style as long as all parameters are used in the query and are in a sequential order.

Audience Republic

Senior Backend Developer

January 2021 - July 2021

Sydney, Australia

- Designed Automation subsystem of the product. It's a custom visual programming language for promoters to describe workflows that trigger in response to certain fan actions. E.g. "when fan purchases a ticket to this event we want to add a tag to that fan, send them an email, wait until event date, send another email with an event reminder" etc.
- Participated in sourcing, interviewing, hiring candidates
- Found a solution to a scaling issue related to generation of tracking links when sending emails
- Designed a solution to scale integrations synchronisation without collisions
- Introduced database migrations via migratus library. Previously db changes were just written as sql text in release notes and were only made by the CTO
- When I joined the project had 2 "app" repositories and one "shared library" repository. This was leading to very complicated and slow process when "library" code was touched both during development and release processes (especially given the team was really small). I merged all 3 repositories in a way that preserved all the history of all three of them, so we ended up with a single clojure project having multiple main namespaces and ability to still have 'git blame' for every piece of the source code. Iteration became much easier and we never looked back.
- Formed a team around me that ended up working on the automation project

Atlassian, Confluence Server

Senior Developer (Synchrony, Clojure)

January 2018 - December 2020

Sydney, Australia

- Analyzed L1-L2 support tickets related to the project, identified that majority of them are related to a troublesome Synchrony server setup in the Data Center configuration, came up with the solution to implement automatic management of Synchrony by Confluence in DC to eliminate the need for separate Synchrony server setup, and collaborated with Confluence Server Scale team to implement it
- Fork Synchrony to maintain a version for Confluence server, separate from Confluence Cloud, as teams are very far and don't communicate, and projects have very different infrastructure and needs. Removed cloud-specific build steps, code and configurations (docker cloud tests, building PaaS jar, cloud loadtest, s3 and dynamodb storage backends, redis caching, cloud-specific encryption, ec2 elb node discovery, aws logging, PaaS statsd, cloud metrics, etc.)
- Build and development setup simplifications and improvements (move to just leiningen instead of bash+gulp+leiningen, remove most leiningen profiles, don't compile java, don't preprocess JS)
- Refactored the code from component to mount-lite. This led to significant simplification of the code, with more explicit graph of dependencies, and significantly improved ability to work with code interactively from the REPL
- Collected all configuration decisions in a single namespace. Previously many configuration options existed in a configuration namespace, but a lot of things have been using environ directly or using other means to detect certain runtime configurations (e.g. running in a cloud PaaS environment, on dev machine, in loadtest etc.). This allowed to ensure that differences between code paths in dev, test and prod environments are explicit and kept to a very minimum

- Implemented generic managed cluster-shared state as atoms on top of hazelcast's IMap. This allowed working with this distributed state the same way as if it were a normal local clojure atom, getting notifications when the state changes, using all clojure standard library functions for atoms, etc.
- Found a workaround for a bug in core.async that causes exceptions to be thrown from an incorrect stack frame (<https://clojure.atlassian.net/browse/ASYNC-198>)
- Implemented hub locking for the data eviction project
- Created simple and extendable cli tooling for all development, build, test, release and other tasks
- Worked with support team to investigate customer issues
- Introduced Renovate to automatically manage dependency version upgrades in Confluence Server

Atlassian, Confluence Cloud

Senior Developer (Synchrony, Clojure)

December 2016 - January 2018

Sydney, Australia

- Code health improvements (code reviews, eliminating tech debt, improving development workflow)
- GDPR and data eviction project
- Pushed for changing synchronization data from linear to tree format for the new editor integration, so standard operational transformation logic won't break the document structure. Implemented a tree diffing algorithm to support tree format synchronization
- Proposed and added jvm memory consumption metrics, so we can see when GC happens and can understand it's implications on the dynamic behavior of the system
- Decreased application bundle size twice (from 100mb to 50mb), which made deployments noticeably faster
- Calculated cluster startup dynamodb usage and increased limits accordingly, which allowed deploying during peak hours with no downtime (previously, deploying during peak hours could lead to downtime up to half an hour)
- Designed and implemented an automated versioning project
- Participated in interviewing potential candidates
- Have been helping with onboarding of new team members
- Led synchrony architecture bootcamp
- Participated in the team on-call rotation
- Participated in a company-wide hackathon (Ship-It) and got into finals

Filemporium/Ourmedian

Lead Clojure Developer

July 2015 - December 2016

Remote via Upwork

- Interviewed potential candidates
- Regularly did code reviews
- Documented project structure, project-specific code conventions, technical decisions, troubleshooting, and Amazon S3 project-specific step-by-step configuration guide
- Did pair programming (to assist others with complicated tasks, to share project knowledge, to get back on track when I'm stuck)
- Revised architecture in a way that drastically simplified client-side state management and allowed live page update of all active user sessions
- Reengineered project build system using boot (previously leiningen were used) Fixed project build time (full recompilation now takes just a couple of minutes instead of an hour). Adjusted project code to work with reloaded workflow
- Refactored most of the project, implemented lots of functionality and fixed lots of bugs (e.g added config schema validation, cleaned up garbage logging (like `(println "!!! F00:" x)`) and implemented proper configurable logging throughout the system, implemented chunked file upload with an automatic reconnection, etc.)
- Setup temporary deployment via docker and dokku

- Implemented production-ready multiserver setup with zero-downtime deployment using NixOS/NixOps and Consul
- Let go of a programmer that have been writing terrible code

Filemporium/Ourmedian

Clojure Developer

May 2015 - July 2015

Remote via Upwork

- Automated design updates
- Added compile-time template checks to kioo templating library to ensure component correctness after a design update
- Added support for using arbitrary npm libraries from ClojureScript code (to be able to utilize existing js React components)
- Started writing project documentation. Documented actions needed to setup a project, update the design, add an npm library

Softswiss Casino Software

Senior Software Engineer

October 2014 - May 2015

Minsk, Belarus

- Implemented integrations with external game providers (CasinoTechnology, Fengaming)
- Implemented completely custom design for a new customer (HTML/CSS)
- Worked on an external wallet api implementation

Rubyroid Labs, LLC

Senior Software Engineer/Team Leader

April 2014 - September 2014

Minsk, Belarus

- Designed application architecture
- Managed project development
- Did code reviews
- Solely implemented some internal services

Intetics Co.

Senior Software Engineer

July 2013 - April 2014

Minsk, Belarus

- Made fully-automated production server setup
- Worked on refactoring legacy codebase
- Worked on security-related features (IP whitelisting, XSS testing)
- Implemented backend service for mobile apps.
- Implemented automatic management of VPN servers DNS rotation
- Did code reviews

Belarusian State University of Informatics and Radioelectronics

Teaching assistant at Electronic Computing Machines Department (part-time)

February 2013 - January 2014

Minsk, Belarus

- Taught first-year students programming in C
- Taught fourth-year students IP networking
- Taught students how to use Git and GitHub
- Together with students formalized grading criteria
- Formalized some code quality metrics
- Regularly reviewed students' code
- Taught Linux for interested students in my spare time

PowerMeMobile, Inc.

Problem solver

January 2013 - February 2013

Minsk, Belarus

- Gave an idea of automating deployment process (new tier deployments may take up to a month of SysAdmin team work)
- Implemented initial stages of deployment automation (installing base cluster software, configuring corosync/pacemaker, installing and configuring DRBD and nginx as resource agents) using Chef
- Made entire deployment configurable from a single place (from chef workstation using node attributes)
- Got an agreement on opensourcing this efforts

Altoros Systems, Inc.

October 2011 - September 2012

Software Engineer in Ruby department

Minsk, Belarus

- Proved that custom multisite functionality is a bad idea. Dropped the hacks and refactored application to use rails 3 engines
- Participated in porting internal RightScale services (mostly sinatra+cassandra) to JRuby to utilize native Thrift
- Participated in all stages of design and development on many projects

Itransition, Inc.

February 2011 - October 2011

Junior Developer in Ruby department

Minsk, Belarus

- Solely ported large social gaming engine from Rails 2 to Rails 3
- Initiated using SCSS and Compass, which led to stylesheets development and modification speedup
- Configured production server from scratch and setup automated Capistrano deployment

SOME FACTS TO BETTER UNDERSTAND WHAT KIND OF PERSON I AM

- I decided to tie my work to computers when I was 5
- First program in BASIC at age of 11, first HTML and JavaScript at 12, first program in Pascal at 13
- I started playing with Linux when I was 14 (it was Mandrake 10 in 2005)
- I use Dvorak keyboard layout
- I use NixOS, and maintain a few packages in it. Before NixOS I used to use Archlinux
- For many years I used to use tiling window managers and a very minimalistic desktop setup. Few years ago I learned GNOME internals and switched to it.
- I use Emacs since 2010. Since 2011 I use it with Evil (vim emulation layer)
- I have dozens of personal opensource projects and have contributed to upstream of at least 20 other
- One of my primary interests is expressivity of programming languages. I hope to study it as a PhD student one day

INTERESTS AND GROWTH DIRECTIONS

- Machine Learning and LLMs (I have some personal projects utilising them)
- Semantics of programming languages
- Programming music (Overtone, Extempore) and visuals (Quil, Processing, Fluxus)
- Electronic music production