Professional Experience

VT Enterprise, startup, USA and Poland — Remote **2018 — 2023**

Fullstack developer

Developed features for the python app: SlashDB automatically constructs REST API from the database using ORM classes for reading and writing, so their content becomes accessible to authorized users and applications in JSON, CSV, XML and HTML.

- Increased performance of JSON data stream by 4 times
- Added support for new database backends: snowflake, databricks, sqlcipher
- Added JSON schema for API — straightforward validation, strict and type-safe API
- Support for tables without primary keys and views in sqlalchemy
- Dockerized testing databases which reduced the amount of manual work and increased test coverage to 90%+
- Added a SAML SSO support with autoconfiguration from metadata URL for secure sign in
- Security fixes, SQL injection, rate limiting, account enumeration prevention, password strength policy, and XML XXE fix which made the product pass the security audit
- Better build process by using Packer. Easier to maintain than bash scripts
- Decreased docker image size by 40% from 500 to 300 MB
- Migrated code base from python2 to python3, startup to systemd to keep project support and development easier
- Improved distributed system mode support by fixing synchronization of state between different nodes/processes

Chat bot for handling customer support in Redmine:

- Slack bot for handling support tickets with OAuth login

Environment: Python, Pyramid, sqlalchemy, SQL, Docker, Linux, JSON schema, Git, mercurial, CoffeeScript, javascript, pytest, AngularJS, HTML, CSS, JSON, XML/XSD, REST, Mysql, MariaDB, Postgres, db2, oracle DB, mssql, SQLite, snowflake, databricks, bash, VirtualBox, packer, rpm, deb, Bitbucket, nginx, uwsgi, asyncio, redmine, ruby, slack

Supl.biz, startup, Tomsk, Russia — Partially remote **2017 — 2018**

Python backend developer

Performed development for the backend of the Django service. This service is for connecting suppliers and customers.

- Integrated with external REST API for checking the reliability of a company
- Implemented search for users’ profiles with Elasticsearch. This decreased response time by ~90%
- Developed search filters for proposals catalog and refactored proposals categorization which increased showed to user proposals number by ~30%

- Added logging of users and staff activity to aid issues investigation
- Fixed exception handling to prevent Sentry bug report flooding
- Covered code with unit and API tests using Django TestCase.
- Created a web parser with the requests library to get orders volume from public data

Environment: Python 3, Django REST framework, Docker, Linux, Elasticsearch, Kibana, Git, JSON, HTML, CSS, JavaScript, React, Angular, RabbitMQ, Celery, Ansible, PostgreSQL, Redis, Sentry, Jira, redux, Bitbucket, SSH, SCRUM, code review

MainConcept — DivX, Tomsk, Russia **2015 — 2017**

White-box QA

Automated testing of multimedia components: HEVC/h265 decoder and encoder, muxers, demuxers, and their integration. Distributed team (Russia, US), remote management (Germany).

- Increased speed of automated tests 4-5 times by using RAM instead of HDD
- Automated functional testing and performance profiling of High-Efficiency Video Coding decoder with use of python, which reduced need for manual work
- Automated integration testing: with DirectShow and other multimedia products — demuxers, renders, encoders, and DivX player using internal tools in C++. This decreased manual testing from ~5 days to ~2 hours
- Created Ansible playbook for the deployment of the test stand for Linux, MacOS, and Windows
- Profiled android binaries over ssh and windows binaries in Intel VTune
- OpenSource: Fixed handling of long version numbers in ansible's module apt-cyg

Environment: Python, Linux, Mac, Windows, Android, iOS, VirtualBox, C++, GitLab, Jira, Ansible, git, FFmpeg

LCC «Ilma», Tomsk, Russia **2013 — 2015**

Embedded firmware developer

Developed firmware for environmental sensors and data loggers. Performed RND for new sensor types.

- Increased measurement precision by ~50% by using digital processing and automated calibration. Reduced the manual work 3 times.
- Increased battery lifetime x2 by optimizing work/sleep cycle and analyzing leak currents
- Developed desktop tools for initializing and debugging sensors using Python and Qt
- Performed RND for new sensor types. Analyzed data using NumPy
- Covered code with unit tests using CppUTest,

Environment: Embedded C and C++, Python, Qt, STM32, IAR, Eclipse, TDD, UART

Skills

Specialist: Python, JavaScript.
Familiar with: Bash, C, C++, CoffeeScript, HTML, SQL, CSS, NASM x86 assembly
Django, React, Angular, Pyramid, SQLAlchemy, FastAPI, Flask
Threading, multiprocessing, async, Linux, REST API, Scrum, TDD, Git

Tools: asyncio, pytest, scrappy, requests, jsonschema, lxml, uwsgi
CI/CD**,** Docker, Docker Swarm, Docker Compose, Kubernetes

Education

TUSUR, Tomsk — Russia **2008 — 2013**
Master's degree in the Tomsk State University of Control Systems and Radioelectronics