Przemysław Spodymek

Curriculum Vitae

Summary

I am a **Senior Software Engineer** with nine years of work experience. I worked on a wide range of Python-based projects and web services, mostly in small teams. My main focus is usually on backend development.

Experience

2019 – ongoing **Senior Software Engineer**, *Intive*, Wrocław, Poland.

- **PayTollo** is a mobile payment platform for tolls in the US. As a backend developer, I worked on features such as **ACL**, **push notifications**, **reverse geolocation**. I've worked on implementing new **Dataflow** pipelines, GAE services, local GAE services emulators, and flows using **Google Cloud taskqueue**. One of the hardest parts of the project I handled was migration from webapp2 to **Flask** and later from **Python 2 to 3**. I used **Strangler pattern** which made it possible to gradually rewrite code while app remained operational for end users.

2017 – 2019 **Python Developer**, *Freelancer*, Europe.

- For two years, I was working on the maintenance and development of an in-house ERP software for **Spreenauten GmbH**. The main app was created with **Pyramid** & **SQLAlchemy**, and most tasks I was assigned to were related to these technologies. Some were also connected to **Angular.js** front-end. Besides feature and bugfix tasks, I have managed to improve codebase on many levels. That includes considerably improving test suit and coverage, extending the app with async tasks using **Celery**, reorganizing migrations using **Alembic**, migrating frontend build to **Webpack** and many more. Other tools that I extended knowledge about while working on this app include **pytest**, **Docker**, docker-compose, and GitLab CI.

2014 – 2017 **Python Developer**, *10Clouds*, Wrocław, Poland.

- **Truststamp** As an R&D team member, I was delegated to find a solution for a couple of strategic issues. One of them involved filling a simple user form based only on a photo of one's ID. My solution was composed of multiple steps: OCR, state detection, extrapolation of ID edges, creating and applying data layout templates. The results were highly accurate, which was the primary requirement. This and other subprojects at Truststamp required **fluency** not only in **Python** but also in **Django**, **Nginx**, **Supervisor**, **Celery**, **Ansible**, and **JavaScript**. Some other tools and parts of the project taught me technologies like Tesseract, Google Cloud Vision, OpenCV, Flask, HTML5 Canvas, scrapy and many more.

- **KITT.AI NLU** is a Natural Language Understanding engine able to hold a multi-turn conversation. During this project, I mainly used my knowledge of **Django** and **Django Rest Framework**. Other technologies I used include **JWT**, **Nginx**, **Vagrant**, **Ansible**, **Celery**, **Stripe**, and **Amazon S3**.
- **SnowBoy** is an offline hotword creation & detection engine (think "OK, Google" or any other sentence). My task was to create an online platform with reliable and fast streaming of audio blobs from a browser to the server. I solved this by implementing two types of endpoints: **WebSocket** (**Tornado**) and **REST API** (**Django Rest Framework**). Both were prepared as **AMI** for **Amazon ELB**. The result was highly **scalable** with instant feedback on the detected hotword. Moreover, the solution used such technologies as **Amazon S3**, and **Amazon RDS**.
- **Tobacco Watcher** is a real-time surveillance system monitoring tobacco-related events on the web. I was working on tuning custom **web crawlers** (Google, Bing, Twitter, RSS), **Celery**, **Solr** search server, **MongoDB** database and other parts of this complex system. Some tasks required running **MapReduce** scripts and analyzing gathered data. During this project, I gained a better understanding of **MongoDB** and **Celery**.

2011 – 2013 **Python Developer**, *10Clouds*, Warsaw, Poland.

As a back-end developer, I worked on several **crowdsourcing** and e-commerce projects. Most of the time I was working in **Python** (Django), but I was also responsible for **JavaScript** programming, system design, and **deployment** with **continuous integration**.

Education

2016 **MSc in Computer Science**, *University of Warsaw*, Warsaw.

Wikipulse: supporting Wikipedians through article suggestion. In my thesis, I implemented a tool facilitating Wikipedians work and encouraging new users to contribute to Wikipedia by suggesting articles to edit. WikiPulse combines **TF-IDF text analysis** and **web crawlers**. I used **Hadoop MapReduce** to process all Wikipedia articles and crawled data. WikiPulse was deployed for a few weeks. During this time, its precision of suggestions was good (testers scored more than 60% of relations as accurate). I learned how to work with **Hadoop MapReduce** on a cluster with **Slurm**. Other tools I used in the project included: Python, Go, Fabric, Django, Celery.

Skills

Fluency with Python, pytest, Flask, JavaScript, Git

Experience with Tornado, Pyramid, Django, Celery, TypeScript, Angular, React.js, Shell, LaTeX, Hadoop, RabbitMQ, MongoDB, PostgreSQL, Datastore, SQLAlchemy, Stripe, WebSocket, Web Crawling, Web Service Design, Map Reduce, Continous Integration, TDD, REST, Crowd-sourcing, Nginx, Amazon Web Services, Google App Engine

Languages

English - Professional working proficiency

Polish - Native