# Przemysław Spodymek

*Curriculum Vitae*

## Summary

I am a **Python Developer** with eight years of work experience, which covers a variety of Python-based projects. Since 2014 all projects I have been involved in were remote.

## Experience

2019 – ongoing **Senior Python Developer**, *Intive*, Wrocław, Poland.

- My first project at Intive was a mobile payment platform for tolls is the US. As a backend developer, I worked on a series of features such as **ACL** for administration, **push notifications**, **reverse geolocation**. I've worked on implementing new **Dataflow** pipelines, GAE services, 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 also migrated password hashing to passlib and tests to pytest. Moreover, I gained experience with **Google Datastore**.

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

- For over 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.

*+48 605 136 778* • *przemyslaw@spodymek.com*

- **KITT.AI NLU** is a Natural Language Understanding engine able to hold a multi-turn conversation. One of my tasks was to implement payments and limit tracking. Because NLU was one part of a bigger system, it was challenging to synchronize user quota. As a final solution, I created a separate service aggregating used resources and handling a Stripe connection. This task taught me how to handle complex payments with **Stripe**. During this project, I mainly used my knowledge of **Django** and **Django Rest Framework**. Other technologies I used include **JWT**, **Nginx**, **Vagrant**, **Ansible**, **Celery**, and **Amazon S3**.
- **SnowBoy** is an offline hot word detection engine that can use different hotword models ("OK, Google" "Alexa" or any other sentence). I was given the software for training such models, and my task was to create an online platform for hotword creation. The most difficult part of this was the reliable and fast streaming of audio blobs from a browser to the server. I have solved this problem by implementing two types of endpoints: **WebSocket** (using **Tornado**) and standard **REST API** (**Django Rest Framework**). Both were prepared as **AMI** for **Amazon Elastic Load Balancer**. The result was highly **scalable** with instant feedback on the detected hotword. Moreover, the solution used such technologies as **Amazon S3**, **Amazon RDS**, Nginx, and Supervisor.
- **Tobacco Watcher** is a real-time surveillance system monitoring tobacco-related events on the web. My responsibilities in this project were broad, as the project consisted of many distinct parts. 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**.

2010 **Software Engineering Intern**, *Samsung R&D Poland*, Warsaw, Poland.

As a mobile phone developer, I have contributed to the creation of two games for Bada phones. During summer internship I was responsible for the development of mobile IM application (system design, protocol implementation, and front-end).

## 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.

## Open Source contributions

- **vscode-coverage-gutters** This is a useful tool for displaying test coverage inside VS Code text editor. I was an author of two small pull requests: major bugfix (wrong coverage was displayed) and performance improvement.
- **cobertura-parse** I added a branch parsing for Cobertura style coverage (with proper tests).
- **CPython** My contribution was fixing MacOS specific bug in **pathlib**.

## Important courses

- Machine Learning by Stanford University

*+48 605 136 778* • *przemyslaw@spodymek.com*

- Learning How to Learn by McMaster University & University of California San Diego

## Skills

| | |
|---|---|
| My usual stack | Python, pytest, Django/Flask, Nginx, Amazon Web Services, PostgreSQL |
| I am good at | Python, PEP8, Django, Flask, Celery, Git |
| I work on | macOS & Linux |
| Experience with | Django Rest Framework, Tornado, Flask, Pyramid, JavaScript, Angular, Shell, LaTeX, Hadoop, RabbitMQ, MongoDB, Datastore, SQLAlchemy, Stripe, Crowdsourcing, Web Crawling, Web Service Design, Map Reduce, Continous Integration, TDD |

## Interests

Software Engineering books & articles, Game design

Swimming, Gym, Photography, Coffee, Travels, Science

## Languages

| | |
|---|---|
| English | Professional working proficiency |
| Polish | Native |

## Websites

github.com/spodym

spodymek.com

*+48 605 136 778* • *przemyslaw@spodymek.com*