Timothy Hobbs - Python developer and graph algorithm specialist

Updated `09-2022`

linked-in: timothy-vladimir-hobbs-5a6a80229

github: timthelion

email: tim@gradesta.com

2017 → present - gradesta s.r.o.

technologies: `rust`, `python`, `cap'n proto`, `0MQ`, `graph-algorithms`, `github actions / ci`, `golang`, `protobuf2`, `OBS`, `figma`, `shotcut`, `gohugo`

project-description: Graph protocol for synchronizing subgraphs of non-finite dynamic graphs between systems

- Designed a novel algorithm for selecting subgraphs based on their topology
- Created an interactive graph editor
- Developed a graph based meta language that could be written in the graph editor
- Wrote a self-hosting compiler for the graph-based meta language in the graph editor
- Built a POC protocol server in Golang
- Worked with a graphic designer to develop wireframes for an improved graph editor

2018 → 2022 - dopracenakole.cz

technologies: `kubernets (k8s)`, `django`, `python`, `redis`, `postgres`, `postgis`, `AWS`, `DigitalOcean`, `celery`, `typescript`, `Vue.js`, `REST`, `circle-ci`, `travis-ci`, `sentry`, `SES`, `S3`, `IAM (AWS)`, `leaflet`, `pytest`

project-description: National bike to work event

- Maintained the legacy codebase for over 4 years across 5 campaigns
- Aided junior developers with their contributions and reviewed their code
- Adapted the application quickly to changing market conditions during COVID
- Migrated all systems from EC2 auto-scaling groups to kubernetes (Digital Ocean)
- Added features such as the interactive calendar view
- Worked with an external team to build a mobile app (Wrote the REST endpoints and provided technical support to the app developers)
- Initiated transition to REST + Vue.js architecture
- Administered the application and monitored it using sentry and hotjar
- Provided technical support for other team members
- Integrated django-helpdesk
- Analyzed UX using hotjar and user interviews

2019 → 2021 - klub-přátel

technologies: `django`, `python`, `redis`, `postgres`, `celery`, `SES`, `AWS`, `S3`, `IAM`, `kubernets`, `heroku`, `linux`, `sentry`, `pytest`

project-description: CRM system used by auto-mat z.s.

- Aided junior developers with their contributions
- Implemented REST endpoints
- Migrated project from heroku to kubernetes
- Managed database backups and performed the recovery procedure including merging of old & new data

2019 → 2021 - Mapa Městem na kole

technologies: `django`, `javascript`, `openlayers`, `geoserver`, `openstreatmaps`, `kubernetes`

project-description: Online interactive bike map

- Added new map tile layers
- Resolved bugs and administered system
- Migrated project from a VPS to kubernetes
- Configured the geoserver heatmap rendering process

2018 → 2021 - django-helpdesk

technologies: `IMAP`, `POP`, `django`, `python`, `OOP`, `security-review`

project-description: Ticketing system for django

- Performed security review on all endpoints
- Refactored security checks to use class-based views
- Added support for teams and filtering/displaying tickets based on team membership
- Refactored email parsing code and added support for Celery
- Aided in code review

2019 - petgraph

technologies: `rust`, `jupyter`, `graph-algorithms`, `graphviz`

project-description: Rust library for working with graph data structures.

- Wrote a series of articles on using the library and its internals
- Created a mechanism for displaying petgraph graphs within a Jupyter Notebook
- Developed a toy library for working with state machines using petgraph

2019 → present - django-import-export-celery

technologies: `django`, `celery`, `xls`, `csv`, `json`, `redis`, `python`

project-description: Small library which adds celery support to the larger django-import-export library

- Created the library for using celery to import/export tables using django-import-export
- Worked with outside contributors to review and merge pull requests
- Wrote documentation

2014 → 2017 - subuser.org

technologies: `python`, `git`, `Docker`, `X11`, `d-bus`, `linux`, `go-lang`, `PulseAudio`, `XPRA`

project-description: Run desktop applications securely in Docker on Linux

- Designed the architecture for the subuser.org application
- Designed the device subsystem for Docker (before there were no unified means for managing devices) (golang)
- Added the `--device` flag to Docker (golang)
- Designed daemon-free system for sharing X11 sessions between application instances (python)
- Created XPRA bridge to allow a secure connection to GUI applications running in Docker (python)
- Created chained build system for re-using previous docker images (python)
- Thoroughly tested the application using automated integration tests (python)

2014 - Trezor one

- Wrote the manual for the first hardware wallet for bitcoin

Education:

- 2013: BA Pedagogy - Charles University, Prague, Czechia
  Thesis - “Developing alternative refreshable Braille display and analyzing the process of reading text output supported by the new tool”
- 2008: AA Math education - Bellevue College, Bellevue, Washington, USA