

# Dr. Tom Parker-Shemilt

---

**Email**      palfrey@tevp.net

## **03/2023 - 05/2023: Doxy.me - Staff Engineer**

- Worked on their Rust-based deployment tool, including adding initial tests
- Terraform work for a new infrastructure setup

## **02/2020 - 12/2022: Babylon Health - Staff Software Engineer**

I worked on several projects for Babylon. All of these involved working with AWS, Kubernetes and Terraform as foundational tech to all of Babylon's microservices. Along the way, I mentored three developers at different times, all of whom I was able to help improve and who fed back praise for that to my manager.

- Tech lead for a full replacement of a testing system for AI medical advice models, allowing clinicians to provide ground-truth data to test against automated systems. System used Django and React/Typescript, along with integration testing with Docker and Selenium. Data migration from the old system involved a months long discussion with data scientists and clinicians, where I led on getting fixed 100s of incorrect entries in a related system that was blocking the migration.
- Tech lead for a team improving the automation of Babylon's medical model release process across multiple projects. Led on removing several areas of significant toil, reducing both the error rate and process time. During this time, I was effectively also the project manager for the team, looking for and planning out our next sections for improvement and running the scrum process, including fortnightly demo sessions.
- Designed standards for long-term maintenance of Babylon projects that were going to be in a maintenance mode for an extended period of time, and then applied said standards as part of a wider team investigating various older projects and bringing them up to that standard.

## **12/2016 - 02/2020: Oliver Wyman - Senior Developer**

LShift was bought by Oliver Wyman, and I moved as part of that. I worked on:

- Leading a team building a new internal deployment platform using Chef, Docker, Rancher, Jenkins, AWS (primarily EC2, but also RDS, Lambda and CloudFormation, mostly generated with Python and Troposphere) and Hyper-V. Target users were a variety of short-term apps needing best-practice infrastructure with minimal effort.
- Implementing a GDPR-compliant PII data sharing/audit platform including work with Clojure, Node.js, Vue, Kubernetes, Docker, Jenkins and AWS (ECS), including writing documentation and technical specifications
- Test/devops lead for a financial project including an iOS/Android React Native app, Kotlin/Golang backend, Python scripting and Kubernetes
- Creating a web frontend and scheduling manager for a strategic planning application, using Node.js, Vue.js, Bootstrap, Python, C# and AWS (Elastic Beanstalk, CloudFormation, etc)
- Backend work for another financial client in Java with Spring Boot, AWS and Docker

## **01/2015 - 11/2016: LShift - Senior Developer**

In January 2015, I returned to LShift, restarting the lead/senior role I had been in priorly, working on a variety of projects, primarily as the lead developer in most cases.

- Work with The Global Alliance for Genomics and Health (GA4GH) on their reference server
- Python – Django and Flask/SQLAlchemy/Alembic
- Clojure and Clojurescript for an internal project
- Constraint satisfaction solving with Optaplanner for automated scheduling
- Exploratory work towards an energy monitoring system running on embedded hardware
- Docker – both writing a report on Docker and networking, as well as use of containers for deployment and testing
- Server build Automation – Salt, Puppet, Fabric, AWS APIs, including use of Zabbix and the ELK stack for monitoring and logging, and CI testing using Vagrant and Teamcity of the server build configurations

## **11/2012 - 12/2014: Spotlight - Senior Developer**

As a Senior Developer for Spotlight, I was responsible for multiple areas, covering development work, architectural design and discussion with Product Owners regarding their areas of interest

- Development – MVC 2-4; C#; NHibernate; NUnit; ReadyRoll; Redis; SQL Server; PostgreSQL; SignalR; jQuery

- Build/Deploy automation – Teamcity; Octopus Deploy (responsible for pushing this to be used in Spotlight – now critical to the build pipeline); MSBuild; Rake/Albacore; FAKE (F# make)
- Server build automation – spearheading Chef development work within Spotlight, including work with Vagrant for development (both for Windows and Linux machines). Various patches from which have been pushed back upstream. Included both local VMs and work with EC2/CloudFormation.
- RabbitMQ – design and implementation of a system-wide change notification system using RabbitMQ

### **03/2010 - 10/2012: LShift - Senior Developer**

My work for LShift covered a wide variety of areas, both as part of a team and as the primary client contact/lead developer for various projects with up to 3-4 developers in a team, using Agile development throughout

- ASP.NET projects - MVC 2 and 3; C#; Entity Framework; NHibernate; NAnt (including work with custom tasks); automated testing and deployment with Selenium, NUnit, Teamcity, Jenkins and Migrator.Net
- Python/Django on Linux (Debian and RHEL) with jQuery integration
- Java/Swing web services
- Working on the RabbitMQ C client
- Puppet configuration, including writing of custom modules in Ruby

### **04/2008 - 02/2010: Imperial College London - Research Associate in Body and Visual Sensor Networks**

My primary focus was in embedded systems development for networked, battery powered devices

- Small devices ( 3cm<sup>3</sup> - “body sensors”) attached to and monitoring people. 2KB of RAM, using TinyOS
  - Working on various sport-related applications; built virtual “orientation sensor” out of multiple raw sensors
- Large devices ( 50cm<sup>3</sup> - “vision sensors”) with a full embedded Linux system and camera module
  - Built a generalised software development platform and applications for the devices
  - Created userspace middleware for easy development (C API and Python bindings) ; Kernel development (primarily driver debugging and extending); Documentation and tutorials for end users

### **09/2003 - 01/2008: Delft University of Technology - PhD in Wireless Sensor Networks**

Thesis focused on the nature of abstraction in sensor networks - examining how this causes problems through the various layers of the software stack, and showing how rethinking the way in which we approach the problems (using techniques derived from the relationship between linguistics and how it is related to how we think about ideas) can provide better solutions to the difficulties faced in sensor networks, with evidence from my implementations of novel protocols both in simulation and for node hardware.

- Published novel, peer-reviewed work on a variety of areas in Wireless Sensor Networks, including Localisation, Routing, Time Synchronisation and Aggregation techniques
- Extensive work with implementations of protocols for TinyOS, including building low-level systems for two different hardware platforms (ATMega128- and MSP430-based), and helping to create node hardware testbeds

### **Selected Publications**

- The  $\lambda$ MAC framework: redefining MAC protocols for Wireless Sensor Networks – published in Springer Wireless Networks, Volume 16, Issue 7 (February 2010)
- Guesswork: Robust Routing in an Uncertain World - presented at the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2005)

### **Other Software Engineering Skills**

Expert Python (20+ years) programmer. Other languages that I have commercial experience in include C, C#, Clojure, Java, Javascript/Typescript, Kotlin, Ruby and Rust. Expert level experience with Linux systems, particularly with Debian-derived and embedded systems. Worked with Kubernetes, Rancher and Dokku, as well as various custom Docker management systems. Have three apps on the Android store (one written in Kotlin, one in Java and a 3rd in Flutter). I’ve worked with most of the major CI and devops deployments systems (as well as having a go at building my own). I can talk knowledgeably about reproducible build systems, AMQP, and Terraform internals among a long and expanding list of topics.

I’ve done some commercial work in Rust, but I’ve done a lot more with it in my personal time. I’m the main maintainer for the awesome-rust list, I’ve got a small patch in the language itself and I’ve written a lot of open source projects in it. One of those projects has now exceeded 6 million downloads.

I was involved with the Gnome project as part of the BugSquad, aiding in bug triage work, and was responsible for closing over 11,000 bugs (mainly due to duplicate reports). I’ve also contributed back accepted patches for a wide variety of open-source projects from dbus to freetype, network-manager to packagekit, and some of my work used to be in the Linux kernel (until the architecture got removed).