Giovanni Capuano

Senior Software Engineer

Based in: **London**

Email: **hey@giovannicapuano.net**

About me

Trust, vision, meaning. With these values as my guiding principles, I'm motivated by a desire to **collaborate** closely with innovators to bring their vision to life. At Honeypot we helped families living in difficult parts of the world make their dreams come true, and at Simply Business, we took care of all the legwork so that our customers could focus on what really matters, safely.

I aim to bring more than just technical expertise to the table. I adopt an **holistic** approach to application development focusing on the **user experience, product workflows, and the overall business impact**. Additionally, I want to help teams to find the practices that work best for them allowing to **perfection development experience for delivering quality work** that is easy to **maintain and evolve**.

I am passionate in designing and developing **small, fast, language-agnostic, disposable, well tested and carefully documented** pieces of softwares and architectures, which behaviors and interactions are easily **traced, monitored and observable**. Ecosystems engineers are glad and incentivised to work with and extend, feel **ownership** of and are keen to **experiment and innovate**.

Ease of change, scalability, disaster recovery and keeping of decision records are topics I value to stress early on.

Professionally, in the latest couple of years I have worked extensively with tools and technologies such as

- **Docker Terraform** and **GitHub Actions**
- **Ruby on Rails Grape Kotlin** and **NodeJS**
- **PostgreSQL** and **MongoDB**
- **Kafka Kafka/Mongo Connector** and **Redis**
- **AWS S3** and **SQS**
- **Pacts**
- **CDC/Event schemas**
- **OpenTelemetry NewRelic Backstage**
- **Solidity** and **Smart contracts**

I started my path as an engineer back in 2012 when I wanted to build applications for supporting my friends and me building tactics and strategies for competitive Pokémon battles. That also led me to approach web development first with the a classic LAMP stack and then then with Ruby (Sinatra, Hanami and Ruby on Rails), Kotlin and the Rust programming language, which I started studying since its beta stage. I also enjoy building web-components with both modern Vanilla JS and React.

I also grew curiosity towards Arduino, Raspberry PI, Android and so on, interests that gave me reasons to look into

languages such as `Swift` and practice a bit more `Java` and `C`.

In 2015 I accepted to join a digital studio company to meet a bunch of highly skilled professionals which gave me the chance to further **grow** as human being and engineer, and at the same time having the opportunity to **shape products** that had a real **impact** on people's life

From there I joined Honeypot, a startup that was in the process of going live, in which I contributed as second in-house engineer to the foundation of the culture and tech infrastructure. During that period I also contributed to the organization of the first edition of RustFest (https://rustfest.global) which had place in Berlin on 2016.

Currently, at Simply Business my team mates (a brilliant, cohesive and passionate bunch!) and I are building and delivering **critical components** of the tech infrastructure while constantly **expanding and sharing** our knowledge, with frequent and efficient **feedback and retrospectives** - and still I enjoy writing code as hobbist, in the attempt to create something **meaningful** for me and others, or just having fun by doing it.

Work experience

Simply Business

Senior Software Engineer

October, 2019 - Present

simplybusiness.com

Acquired by Travelers [NYSE: TRV] in 2017 and based in London and Boston, SB is the **leading online business insurance brokerage** that specializes in protecting businesses providing the protection suited to their risks and priorities and have helped over 1M businesses across thousands of different trades among **UK and USA**.

- Designed and delivered a Change-Data-Capture solution for **gguaranteeing atomic, reliable, and consistent processing of real-time events between complex systems**, which ensured that only the most recent data that successfully committed to global and multi-region databases is delivered to consumer services.

  Some of the issues we intended to mitigate with our solution were dual writes; Kafka failing to deliver under load or network errors; different failure rates due to database and kafka being on different networks; difficulty to observe or guarantee SLIs (freshness, delay...) when hopping through the pipeline. Our solution included attaching a MongoDB-Kafka Source Connector to a replica instance from which reading the change stream and, after hopping through a transformer service to normalise the data, publishing to read-view topics for consumption by other systems.

  We also built a solution for AWS S3 publishing relying on its notification systems and the SQS stack.
- Revamped the way we used to **organise information with the goal of making it easily accessible, structured, collaborative and pleasant to work with**. In this framework, I've got appreciation for setting up an archive for things describing past states and digital ephemera: documents, emails, walkthroughs, diagrams, notes, briefs, incidents... that proved useful for answering questions and promote what I would call **"librarian" culture.**
- Conducted research and development to **build a platform that automated workflows** previously done by call center operators. The workflows involved large, complex, and slow spreadsheets. During the development process, we held fortnight demos to get feedback from a small sample of users and stakeholders. Few specifications remained set in stone and responding to their feedback thanks to quick iterations led us to bring promptly more happiness and efficiency among the teams.
- The organization's data amount got close to the hard limit of what the current infrastructure could support. After analyzing the needs of different teams and stakeholders, we transitioned a large part of the data to a **completely different kind of storage solution, saving the company a dramatic amount of cash, making read and writes faster** and allowing us to keep everything running smoothly without any disruptions. We worked close to the ORM to ensure no code changes were needed even though part of the data was on different storage solution.
- As part of my responsibilities, I have **interviewed engineers** applying to join the organization.

Honeypot

Software Engineer

November, 2019 - May, 2018

honeypot.io

Honeypot is Europe's developer-focused job platform that let companies do the first move, using a powerful set of **search engine filters and keywords** and also **receving tailored emails** showing talents that are most relevant for them. Recruiters are also able to connect to their account **ATS** such as *Greenhouse* or *Personio* to avoid any additional manual work.

I joined the company as **second in-house full-time engineer** and during the 3 years and half (then it got acquired by the German LinkedIn competitor: XING/NewWork) I have contributed to enstablish a **developer-focused culture, shaped a solid and mantainable infrastructure, sane and safe workflows** and at every opportunity shared knowledge for providing a **clear understanding** of the software development processes among all the non-tech colleagues.

- I have done **daily round of interviews** to developers applying to join the organization and worked with my colleagues on establishing a sound internal hiring process.
- Supported the **Business Intelligence** team with the creation of a foundation of databases and workflows to allow them applying their algorithms and analysis on sandboxed, reliable data.
- Built an open source search engine allowing the search for talents becoming dozen of times faster, with the service being resilient and rock solid. I started working on it when Rust was not yet v1.0 and many features and libraries were not stable or fully-functioning yet - thus a lot of resources went to contributing to other projects (including the Rust buildpack for Heroku), to Rollbar and CircleCI and to the whole workflow to deploy and orchestrate all the different stages. Initial challenges aside, Rust proved to be a solid choice for such a kind of a project.
- Technologies I have been working with include: Ruby on Rails, Rust, Elixir, Ember.JS, Redis, Heroku, CircleCI and TravisCI, AWS Cloudfront and S3, Grape (RESTful APIs based on JWT and TOTP authentication), Postmark and Mailchimp, Greenhouse and Personio, MindMatch AI, TalkJS...

- https://github.com/honeypotio/reina – a CLI tool and GitHub bot for solving our need of orchestrating the deployments of the feature stagings of our applications on Heroku
- https://github.com/honeypotio/searchspot – ElasticSearch-backed client/server written in Rust used by recruiters to search the talents they are looking for
- https://github.com/benashford/rs-es – Rust client for the ElasticSearch REST API I contributed to in order to make Searchspot
- https://github.com/RoxasShadow/rollbar-rs/ – library to track and report errors, exceptions and messages from Searchspot application to Rollbar
- Blog posts such as https://blog.honeypot.io/errors-and-exceptions-in-rust

Moze

Software Engineer

April, 2014 - October, 2015

honeypot.io

Moze is an Italian studio of designers and developers which designs and builds digital products for entrepreneurs and forward-looking companies to shape or enhance digital products and services through a pragmatic design process and agile, incremental development.

I have been the first in-house full-time engineer after years the company relied mostly on contractors. I was close to every side of the projects, from the **front-end to the back-end to the dev-ops**.

- Among the other projects, we built Bugree: similarly to Proton Mail and MEGA, it was the first portal to make communications happens under NDA, and the first portal with **client-side encryption bounded to a dynamic key**, therefore messages never left any electronic footprint. There were more features like scrolling text, torch mode, limited openings and watermarks. Both messages and attachments were secured by end-to-end encryption and notifications were **real-time** thanks to a websockets-based NodeJS application we made. Its frontend was built with ReactJS and AngularJS and we implemented multiple payment processors including PayPal and Stripe.
- We also built GODO Shopping (for which we also made and open sourced zanox-rb, as we needed a good library to interact with the APIs of the popular affiliate marketing network service), which was a marketplace that with every purchase was granted a percentage in cashback to the buyer.
- Tourizan was another good project: like AirBnB but for enjoying unique experiences and places in Italy.

Bio

Born near **Naples, Italy**. Graduated in scientific high school and attended **Computer Engineering** at the University of Naples "Federico II" before leaving my hometown to accept a role at Moze, in **Milan**.

Italian, Neapolitan, English and some Spanish aside, I have been studying **Japanese** and joined an intensive course in a language school in Kyoto, in the meantime I was living with a local family thanks to the homestay program.

I love travelling and discovering new cuisines of all world but also all the regional cuisines of Italy. I love to prepare food and experiment new dishes everyday in my kitchen (and share the results on my Instagram!).

Being on holiday means for me means hiking mountains with friends and go discovering new pizza places or enjoying the seasonal menus of my favorites!

I'm a also passioned videogamer since I have memory of: always thirst of superb game and level designs, great soundtracks and meaningful stories.

Giovanni Capuano - hey@giovannicapuano.net - References on request