tiagokatcipis

software engineer

contact

Berlin,Germany
+39 349 859 0291
tiagokatcipis@gmail.com
GitHub
LinkedIn
Blog

languages

English, Portuguese

programming languages

Go, C, C++, Python, Lua, Bash, Javascript, Nash

protocols

HTTP, gRPC, SIP, RTP, AMQP

cloud

Kubernetes, AWS, Azure, Google Cloud, Docker, Terraform

automation

Ansible, Make

dev environment

Vagrant, Docker Compose

monitoring

Prometheus, Grafana, StatsD, Sysdig

Introduction

I'm a curious programmer that likes to explore different ways to design, build and test software always trying to understand as much as I can from the entire environment I'm working in.

That led me to do a lot of different things in my 10 years career, from embedded software in C to distributed systems in a variety of languages and protocols.

I'm passionate about automation and minimalism when building scalable, efficient and flexible software.

experience

2019–present **FromAtoB** Berlin, Germany
Software Engineer - Search2Book Team

- Implemented new location service from scratch.
- Added PSD2 compliance on payment method storage service.
- Helped migrate core services from legacy environment to new GKE cluster.
- Built backup routines for critical service (Google Memorystore).
- Improved development environments making them more consistent.
- Integral part of the hiring process doing interviews and pair programming sessions.

2017–2019 **Neoway** Florianópolis, Brazil
Software Engineer - Data Platform Team

- Led the migration of the entire data platform from AWS to Azure.
- Developed tools to automate building infrastructure, like klb.
- Created new service to solve audio captchas (Go, Python, SVM).
- Prototyped image captcha solver using TensorFlow.
- Did 3 different presentations in 2 different conferences.

2015–2017 **Neoway** Florianópolis, Brazil
Lead Software Engineer - Data Capture Team

- Led development of a new data capture architecture.
- Implementation of multiple services for the new architecture (Python, Go).
- Added improved and fully automated monitoring system (Sysdig, StatsD).
- Coached the team on better testing practices and TDD.
- Fully automated dev environments and deployment (Docker, Docker Compose).
- First team on the company to deploy and use Kubernetes to manage more than 100 deployments.

2012–2015 **Dígitro** Florianópolis, Brazil
Lead Software Engineer

- Developed VoIP phone with color touchscreen from scratch (C on a Blackfin DSP).
- Automated development environment for cross compilation (Ansible, Vagrant).
- Replaced legacy audio service that used Flash (RTMP) with an HTTP/HTML5 solution (NodeJS,C).
- Coached team on automated testing and TDD.

2010–2012 **Dígitro** Florianópolis, Brazil
Software Engineer

I started working on a solution to web audio playback with very specific audio effects (like silence removal, change in pitch) that had to be developed using Flash (RTMP). To solve that problem I worked with two different open source C++ projects that did reverse engineering of the RTMP protocol to develop our own Flash Media Server. I worked directly with the integration of the server playback logic with Gstreamer and the plugins that enabled the desired effects on playback.

The next project was a solution to biometric identification using a third party C library that built and scored voice models. I developed a REST service in Lua that integrated with C code that built the voice models and used MongoDB to store the voice models and perform searches on the database.

2008–2010 **Dígitro** Florianópolis, Brazil
Trainee

Helped in the development of an cross platform (Windows and Linux) audio streaming library for a VoIP softphone, aiming at porting the current application that was Windows only to Linux. I also got involved in the development of a prototype for a voice biometrics system.

2007-2008 **Cyclops / LAPIX** Florianópolis, Brazil
Trainee

Worked on adding new features on the system responsible to integrate medical equipment to the DICOM system, developing a cross platform domain specific graphical XML editor. This involved learning C++ and XML parsing, together with developing cross platform GUI applications, on this case using WxWidgets. The code has been tested using CppUnit.

open source projects

2017-now **mdtoc** https://github.com/madlambda/mdtoc
A very simple table of contents generator for markdown.

2016-2018 **nash** https://github.com/NeowayLabs/nash
Nash is a shell language focused on simplicity and having a nicer syntax than traditional shells and support to containers. It also strives to be safer than traditional shells.

2016-2018 **klb** https://github.com/NeowayLabs/klb
klb is used to automate infrastructure creation on AWS and Azure. I got involved on designing the support for Azure since this was the tool used to migrate Neoway production infrastructure from AWS to Azure.

2013 **CppUTest** http://cpputest.github.io
CppUTest is a C /C++ based unit xUnit test framework for unit testing and for test-driving code.
In this project I worked both on improving the documentation and at adding new native types to the mock framework (which involved some refactoring).

2012 **GStreamer** http://www.gstreamer.net
GStreamer is a library for constructing graphs of media-handling components. I contributed with a plugin named *removesilence* and some documentation for the GstCheck documentation.

2010-2011 **Pattern detection on H.264** https://github.com/katcipis/h264.pattern.detection
This is my Bachelor's Thesis and it consists of a prototype of a H.264 CODEC that uses OpenCV and H.264 internal algorithms to do pattern detection and object tracking integrated on the encoding process.
Metadata generated on the encoding process is integrated on the video bitstream on conformance with the standard.

2010-2011 **LuaSofia** https://github.com/ppizarro/luasofia
Lua binding for the Sofia-SIP library. Contributed to the project from the start, helping to make decisions about the design of the software and documenting it.

2010 **GPS tracking system** https://github.com/katcipis/gps.tracking
System designed to provide the location of a device at the receive of a position request using SMS.

2010 **LuaNotify** https://github.com/katcipis/luanotify
Lua library that implements a simple Pub/Sub system inspired on glib GSignal API.

presentations

2018 **Object Orientation in Go** The Developers Conference

For people that come from a background on Java or other classic object oriented languages (like C++) there is also some discussion on if Go is actually object oriented.
In this presentation I try to present Go as a language that is more object oriented than these classic languages, at least according to the original foundations of object orientation.
Presentation source can be found here.

2016 **Building Resilient Services in Go** GopherCon Brazil

Resilience is not about never failing, but how do you recover from it. How can you prevent your services from locking down or exhausting all its resources ? How to perform graceful service degradation ? Can this kind of behaviour be tested properly ?

On Go we have some new features, like Contexts, that helps us to model timeouts and cancellation properly.

They can be combined with other useful features as select and channels to model timeouts and resource pools, which can be essential to provide proper service degradation instead of total failure of the system.

On this talk I try to answer this questions using new features available on Go 1.7, direct from production ready software.

Presentation source can be found here.

2016 **Real Life Kubernetes** The Developers Conference

On this presentation we will give a short introduction on Kubernetes and show the experience of learning and using Kubernetes on production for two very distinct systems.

The first one is a data acquisition system, involving running multiple instances of different crawlers, storage services, captcha breaking services, message brokers (like RabbitMQ) and database integration outside the cluster.

The second one is a web application, involving network analysis using graphs with the ultimate goal of fraud prevention. The application is strongly bounded with the microservices architecture and the twelve factor app.

Graph and document databases, cache layers, a message broker and a distributed filesystem are some of the technologies surrounding the application ecosystem.

Presentation source can be found here.