nicolashillegeer

engineer, computer scientist

about

kattestraat 51
9150 kruibeke
belgium

Born on
February 11, 1988
Place of birth
Beveren-Waas, Belgium

nicolas@hillegeer.com
http://www.aktau.be

languages

native: **dutch**
fluent: **english, spanish**
proficient: **portuguese**
notions: **french, german**

programming

Java, Scala
C, C++, Qt
Node.js, Go
PHP, Python & Lua
CSS3, HTML5 &
JavaScript

about

I like creating elegant systems using the best tools for the job. Making them as fast and fault-tolerant as possible, whether they be server software, desktop apps, web apps, …

To dive into a project and learn new skills while applying the expertise I received by doing other – seemingly unrelated – projects is a joy for me. I always strive for maintainability, reliability, performance and security.

education

2010-2011 **Master of Science** — Katholieke Universiteit Leuven
Majoring in Computer Science
Specialization in Artificial Intelligence

2009–2010 **Master of Science** — Universidad Autónoma de Barcelona
Majoring in Computer Science
Specialization in Artificial Intelligence

2006–2009 **Bachelor of Science** — Kahtolieke Universiteit Leuven
Engineering Sciences, specialization in Computer Science

experience

10/12-05/14 **Freelance, Antwerp-Heidelberg** — Freelance consultant

Development of large and small software systems, usually featuring a client-server architecture. Focus on web technologies.

My latest assignment was a solo project for an Antwerp-based media company aiming to develop a digital signage system (advertisement and media) that allowed them to have high-definition media streamed to remote players everywhere at the touch of a button – in stores, at events, in parking places, et cetera. In the project description section below you can find a more detailed explanation of the technical details and the technologies that were used.

As this was a rather large solo project, I performed all the tasks normally associated with a large-scale software project, from customer relations to architect, developer and even sysadmin. This gave me a lot of experience with topics that I wouldn't have come into contact with if I had only ever worked as part of a larger team.

09/11-09/12 **Capgemini, Diegem** Consultant
Development and maintenance of a large data warehouse for a chemical industry giant using Oracle technology.

09/10–07/11 **Department of Computer Science, Katholieke Universiteit Leuven, Leuven** Student
The objective of my thesis project was to construct various tools with the goal of aiding and optimizing the workflow of archaeologists trying to restore frescos all over the world. Primarily this means digitally reconstructing the fresco with the aid of specific algorithms, and helping the operator judge the results whilst allowing him/her to suggest new possibilities. A GUI was needed that combined the best of both human and computer to be able to solve the problem. I used C++/Qt to get the job done.

08/10-09/10 **Department of Physics, Katholieke Universiteit Leuven, Leuven** Researcher
Development of a hybrid neural network for recognition of material hardness through soundwave analysis.

10/09-06/10 **Department of Computer Science, Katholieke Universiteit Leuven, Leuven** Student
Creating a pathfinding robot in team.

08/09-09/09 **Department of Biology, Katholieke Universiteit Leuven, Leuven** Developer
Pilot project: construction of an educational game for children teaching them about evolution.

capabilities

On the **frontend** – for direct interaction with the user – I'm fond of using modern **HTML5**, **CSS3** and **JavaScript** when I'm working on a web app or the **Qt** toolkit when on the desktop. I gladly make use of libraries when they drastically shorten the time needed to completion, but I'm not afraid to develop custom tools from scratch when required.

There are not many web-facing projects I can freely advertise, luckily one of them is currently online at **www.smuspimpen.de**. The site was mostly frontend work, using **vector graphics** and **html5** audio so it could be displayed (and listened to) on mobile browsers as well. It was co-developed by myself and a design artist for a music producer who wanted something a bit out of the ordinary to promote his work.

I've also gained a lot of experience working on **backend** technologies, developing systems from scratch. Setting up and provisioning virtualized servers with **UNIX** (mostly *Linux*). Creating distro packages and writing shell scripts for smooth daily operations. System-level programming in **C** for heavy-duty tasks that require great efficiency.

As for **data**, an increasingly important factor in our rapidly globalizing world, I've gotten personal with my fair share of database and data storage technologies. Arguably the biggest assignment was a year of data warehouse management for a global player in the chemical industry, where I learned the ins and outs of *Oracle* database technologies, designing data representations and queries that could run over terabytes of customer and sales data in seconds.

That said, I've also succesfully used many alternatives (*PostgreSQL, SQLite, MySQL, MongoDB and redis*) to complete projects. In the end, the most important thing is what the client requires – taking into account the budget available – and picking the right tool for the job.

I often use JVM-based languages, such as **Java** and **Scala**, for both personal and professional projects. The latest of which was creating a front-end for said big data warehouse that allowed developers to drill down in the dataset and keep it in sync with their development environments – which previously had significant administrative overhead – saving time, money and frustration.

Of course, this is not an exhaustive listing, so in case of doubt please contact me and we can have a talk about what it is that needs to be done.

project descriptions

This section provides a more technical view of the tasks performed on certain projects, for those interested.

10/12-05/14 **Digital signage player** Antwerp-Heidelberg

A solo project aiming to develop a digital signage system that lets many media player units connect to a central server. The server is responsible for the worldwide distribution of content in a secure and efficient fashion. The client is responsible for playing this content and reporting on its activities. Advertisers can pay to advertise on this network, for example by putting content and presentations on player units in their vicinity to attract customers. As the advertiser pays for time on the player units, it is necessary that the unit be as stable as possible and recover from any error it may encounter. Supported media types are RSS feeds, weather data, custom websites, HD video and flash presentations. This is easily extendable.

On the technical side, I researched and used a lot of (sometimes unfamiliar) technologies that were a good match for the project. A sampling:

puppet *for deploying both the server and clients, making everything reproducible and maintainable. New player boxes are deployed by popping in a USB stick containg a customized debian, further installation is hands-off. This makes it fast and easy to setup new units and deliver them to costumers on time.*

linux (debian) *both the server and client run debian linux (wheezy), and extensively use both standard and custom packages for provisioning, making upgrades and bug-fixes on both all and specific units a trivial matter. A custom debian repository runs on the server.*

C *To display the various media types a web browser (chromium) is used as the main viewport. Yet, no web browser for linux supported hardware playback of high-definition content. So a browser plugin was written in C that interfaces the browser with a video player that can take advantage of the low-cost hardware and play HD content without breaking a sweat.*

nginx *both the server and the clients run instances of nginx to coordinate the various running services (media download, video playing, et cetera.*

node.js, websockets, server-sent events *for managing such a large network some remote control is necessary, however many customer locations have very locked-down networks. One of the few ways to evade this in a general way is to use HTTP over SSL (HTTPS) as a transport protocol. This is often one of the few unblocked protocol/port combinations (employees use gmail...). This technique has the added advantage of securing all communications from unwanted eyes. When possible, this remote control system uses WebSockets, with a fallback to server-sent events, which is indistinguishable from plain HTTP to any router/proxy.*

redis, mongodb *players are sending information and reconnecting all the time for various reasons, which is why the datastore for managing this information needs to be very efficient for both writing and reading. Redis – a popular in-memory database – was chosen for this task and has proven to perform admirably. The players themselves keep all information about the playlists (duration, type, file location, ...) locally in a mongodb instance.*

PHP *large parts of the digital signage application were written in PHP, which has proven itself a stable and performant web technology.*

Javascript, CSS *except for the videos, most media is pure HTML. To provide for smooth transitions and a pleasing appearance the player uses a combination of javascript and CSS.*

Antwerp-Heidelberg

golang *Developed some long-running daemons in Golang: 1. A database synchronization daemon that transfers data from a legacy MySQL database to a PostgreSQL database, helping to make the transition to the new generation of the platform seamless. 2. A remote control daemon that can handle many more connections than the old Node.js implementation. It features a cleaner architecture, decreased memory and cpu usage on both the server and clients. This enables lower-cost hardware and makes deploying a breeze. These daemons are managed by the* ***runit*** *low overhead process supervisor.*

09/11-09/12 **Data warehouse management** Brussels

Working as part of a team of consultants, I was responsible for the management of a very large data warehouse for a big international player in the chemical industry. The warehouse itself was hosted in an ***Oracle database****, and most of the processing logic happened in* ***PL/SQL*** *for performance reasons (the system was used by thousands of users concurrently). I added new data analysis tools which were later used by management to gauge the effectiveness of management decisions.*

Further I transferred extra data about the worldwide materials stock from ***SAP*** *to the data warehouse so it could be aggregated, processed and used for decision making. Moreover I provided tech support and helped optimize the data layout and queries so users did not have to wait for minutes for their reports to update, thereby saving time and resources. As a last subproject within this assigment, I quickly developed a web interface to the data warehouse in* ***Java*** *and* ***Scala****.*

activities

Leisure Antwerp, Heidelberg, Barcelona

traveling, reading, skiing, playing soccer, going to the gym, visiting festivals, learning languages