

# 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

- |             |   |                                   |
|-------------|---|-----------------------------------|
| 09/10-09/11 | <b>Master of Science</b><br>Majoring in Computer Science<br>Specialization in Artificial Intelligence | Katholieke Universiteit Leuven    |
| 09/09-06/10 | <b>Master of Science</b><br>Majoring in Computer Science<br>Specialization in Artificial Intelligence | Universidad Autónoma de Barcelona |
| 09/06-06/09 | <b>Bachelor of Science</b><br>Engineering Sciences, specialization in Computer Science                | Katholieke Universiteit Leuven    |

## experience

- |             |  |                      |
|-------------|--|----------------------|
| 10/12-08/14 | <b>Freelance, Antwerp-Heidelberg</b><br><i>Development of large and small software systems, usually featuring a client-server architecture. Focus on web technologies.</i> | Freelance consultant |
|-------------|--|----------------------|

*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 architect, developer to sysadmin and even customer relations. 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.*

*For more information, please see the project description part of the CV.*

- 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 backend I've performed a lot of roles, both creating systems from scratch and maintaining already written software. System-level programming in **C** and **Go** for heavy-duty tasks that require great efficiency and being frugal with system resources. Provisioning (virtualized) servers running **UNIX** (mostly *Linux*). Creating distro packages and writing shell scripts for smooth daily operations and error logging/prevention. Using *perf*, *strace* and friends to debug performance issues, et cetera.

As for **data**, an increasingly important factor in our rapidly globalizing world, I've gotten personal with my fair share of database and 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 tech, designing representations and queries that could run over terabytes of customer and sales data in seconds.

That said, I've also successfully used many alternatives (*PostgreSQL*, *SQLite*, *MySQL*, *MongoDB* and *redis*) to complete projects. In the end, the most important thing is picking the right tool for the job.

I have worked with 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 devs to drill down in the dataset and keep it in sync with their dev environments – which previously had significant wait times – saving time, money and frustration.

I've also gained a lot of experience working on **frontend** technologies – 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.

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 small 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 containing 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 (legacy), WebSockets, Server-Sent Events** *for managing such a cluster, robust remote control is necessary, however many customer locations have locked-down networks. One way to evade this is to use HTTPS as a transport protocol. This is often one of the few unblocked protocol/port combinations (employees use gmail...). It has the added advantage of securing communications from prying 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 below layer 7.*

**Redis, PostgreSQL** *players are sending information and reconnecting all the time, which is why the datastore for managing this volatile information needs to be quite 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 PostgreSQL instance.*

**PHP** *Some parts of the digital signage application were written in PHP, which has proven itself a stable web technology with rapid turnaround.*

**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.*

**Go** *Developed some long-running daemons in Go: 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 sizeable data warehouse for a big international player in the chemical industry. It contained the majority of the data that the conglomerate and its ~450 subcompanies collected (finance, product stock, personnel, ...). The warehouse itself was hosted in an **Oracle database**, and most of the processing logic happened in **PL/SQL** for performance and homogeneity reasons. The system was used by thousands of users concurrently. I added new data analysis tools which were later used by management. Both to gauge the effectiveness of their decisions, as well as to base new policies on.*

*Further I transferred extra data about the worldwide materials stock from **SAP** (deciphering its data structures) to the data warehouse so it could be aggregated, processed and presented. 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 assignment, I quickly developed a web interface to the data warehouse in **Scala** that was used by developers to save lots of time while triaging bugs and other data issues uncovered by user reports.*

## activities

### Leisure

Antwerp, Heidelberg, Barcelona

traveling, reading, skiing, open-source, playing soccer, going to the gym, going to festivals, learning languages