

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++(11)
PHP, Python & Lua
CSS3, HTML5 &
JavaScript

interests

Creating elegant systems using the latest technologies and methodologies. Making those systems as fast and fault-tolerant as possible. I have a special interest in web technology, both on the server and client-side. The web is a rapidly changing environment and it is imperative to find the right technology that allows one to get an edge over the competition, both in speed of development, reliability and performance.

education

- | | | |
|-----------|---|-----------------------------------|
| 2010-2011 | Master of Science Majoring in Computer Science Specialization in Artificial Intelligence | Katholieke Universiteit Leuven |
| 2009-2010 | Master of Science Majoring in Computer Science Specialization in Artificial Intelligence | Universidad Autónoma de Barcelona |
| 2006-2009 | Bachelor of Science Engineering Sciences, specialization in Computer Science | Katholieke Universiteit Leuven |

experience (part 1)

- | | | |
|-------------|--|----------------------|
| 10/12-07/13 | Freelance, Antwerp-Heidelberg <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 their high-definition media (and that of their clients) streamed to remote players everywhere – in stores, at events, in parking places, ... In the project description section there is a more detailed explanation of the technical details of the project.

As this was a 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 been working as part of a larger team such as my earlier projects. It allowed me to see many things with a fresh perspective and come up with good ways to integrate the different roles and responsibilities and streamline the process in future projects, whether they be small and nimble or vast and long-lasting.

experience (part 2)

- 09/11-09/12 **Cappgemini, Diegem** Consultant
Development and maintenance of a large data warehouse for a chemical industry company with Oracle technology.
- 09/10-07/11 **Department of Computer Science, Katholieke Universiteit Leuven, Leuven** Student
In this year I worked on my thesis project. The objective 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.
- 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.

activities

Leisure

Antwerp, Heidelberg, Barcelona

Travel, reading, skiing, soccer, fitness, festivals, language learning

project descriptions

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

A solo project aiming to develop a digital signage system (i.e.: advertisement and media on customer's premises) that lets many media player units connect to a central server. The server is responsible for the worldwide distribution of content (videos, presentations, weather, news, ...) in a secure and efficient fashion, and 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.

On the technical side, I researched and used a lot of (previously 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 one USB stick and afterwards running one command, 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 presentations (flash or HTML), RSS feeds, weather and support easily editable HTML templates for multiple playlists, it was advantageous to use a web browser (chromium) as the main viewport. However, 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 are running instances of nginx to coordinate the various running services (media download, video playing, et cetera).*

node.js, websockets, server-sent events *many customer locations have very locked-down networks, one of the few ways to evade this in a general way and provide the player units with the ability to report on the items that have been played, is to use HTTP over SSL (HTTPS) as a transport protocol. The reason for this is, is that it is common for employees to visit HTTPS websites (such as gmail) from inside the network. This technique has the added advantage of being secure against eavesdroppers and man-in-the-middle (MITM) attacks. When possible, this remote control system will use WebSockets, as it is more efficient for full-duplex communication than vanilla HTTP. However if that fails the system will fallback to server-sent events, which is indistinguishable from plain HTTP to any router/proxy.*

redis *many player units 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.*

mongodb *the player units themselves keep all information about the playlists (duration, type, file location, ...) in a mongodb instance.*

PHP *large parts of the digital signage application were written in PHP, which has proven itself as a web technology that allows for very rapid development of an application that can run for a very long time with decent performance.*

Javascript, CSS *except for the videos, some items that the customers want shown are for example RSS feeds and weather reports. To provide for smooth transitions and a pleasing appearance the player used a combination of javascript and CSS.*

