



Daniel Janus

[dj@danieljanus.pl](mailto:dj@danieljanus.pl)

[www.danieljanus.pl](http://www.danieljanus.pl)

(+48) 501-180-396

#### Address

Béli Bartóka 8/127

02-787 Warsaw

Poland

# Daniel Janus

## Software Engineer

### Technical Expertise

#### Programming languages

**Expert**<sup>1</sup>: C, Clojure, Common Lisp; **Experienced**<sup>2</sup>: Ruby, Java, Bourne shell, JavaScript, ClojureScript; **Intermediate**<sup>3</sup>: Python, Haskell, OCaml, PHP, R, Standard ML, x86 assembly, C++, Perl, Smalltalk

#### Other languages

HTML/XHTML + CSS, SQL, T<sub>E</sub>X, L<sub>A</sub>T<sub>E</sub>X, Markdown

#### Concepts

Cloud computing, software portability, unit/component/regression/integration testing, *i18n* and *l10n*, MVC, functional programming, scalability, machine learning & data mining (kernel methods, genetic algorithms/evolution programming, neural networks), agile development

#### APIs, Libraries & Frameworks

POSIX, Ring, React, Ember, Backbone, jQuery, Rails, Django, EJB, JPA, Tapestry, Django, YUI

#### Operating systems

Linux (Debian, Ubuntu, RHEL, CentOS), Unix, Windows, VMS

#### Version control systems

Git, Mercurial, Darcs, Subversion, CVS

#### Database management systems

PostgreSQL, MySQL / MariaDB, SQLite, Sybase, MS SQL, AllegroCache, Redis

## Professional Experience

### Jan 2015 – present, *Software Engineer, Rebased*

Full-stack development of bespoke Web-based software in Ruby on Rails, liaising directly with customers.

Customers:

- *Erudion* – a Facebook-like social networking solution for companies, with frontend written in Backbone and Marionette.js; I helped reduce the project's technical debt after joining it in its late stage
- *Apps for Good* – maintained and developed the code powering up this UK charity's website, including the e-learning application for schoolchildren teams and their teachers
- *Crowdfind* – co-architected the lost-and-found application for event organizers and large venues and helped integrate it with external services

**Environment:** Ruby, Rails, Ember, React, Backbone, Solr, jQuery, PostgreSQL, MySQL, Git, Linux, Heroku

<sup>1</sup>I've used the language professionally extensively and mastered its intricacies to a very detailed level.

<sup>2</sup>I've been using the language for commercial or otherwise notable projects on a daily basis.

<sup>3</sup>I have written at least one non-trivial project in the language, but there are areas that I either haven't learned or need brushing up my knowledge of.



Daniel Janus

dj@danieljanus.pl  
www.danieljanus.pl  
(+48) 501-180-396

**Address**

Béli Bartóka 8/127  
02-787 Warsaw  
Poland

### Sep 2011 – Aug 2014, Senior Programmer/Analyst, Citigroup

I was a member of the Correlation team within the Global Credit Technology department, taking part in creating code that tracked, processed and priced credit default swaps (CDSs), collateralised debt obligations (CDOs) and other exotic credit derivatives.

I worked with an internal Clojure-based system that allows for easy creation of tabular and graphical reports with arbitrary logic and aggregates data from heterogeneous sources. I maintained a number of reports, ranging from profit-and-loss (PnL) reporting for the front-office desk through trade data aggregators forming an input to the pricing system to scripts that are part of business processes fulfilling Basel III and other regulatory requirements. I also took part in enhancing the reporting system itself, having written an Emacs mode and a number of libraries. I liaised with other teams and the desk on a daily basis.

**Environment:** Clojure, Java SE/EE, Maven, JIRA, Sybase, MS SQL Server, Gemfire, Tibco EMS, Windows 7, Linux (RHEL)

### Aug 2009 – Aug 2011, Co-Founder / Lead Programmer, Fablo

Fablo (now Retailic) is a startup company developing a state-of-the-art instant search engine and a set of related tools for Polish e-commerce. At the start of the project, we wanted to build a semantic search engine – one that would guess the product regardless of the exact wording. I experimented with various methods and created a working prototype using Generalized Latent Semantic Analysis trained on a big sample of Polish texts. The engine also could learn proper ranking of the products depending on user preference, using Supervised Semantic Indexing (SSI) as described by Bai et al. Due to the unreliability of these methods on certain groups of queries, we switched to more predictable retrieval based on cosine similarity enhanced with various heuristics.

I implemented the search engine in Clojure (experimenting with Mathematica along the way), co-assisted design and implementation of the Amazon AWS-based cloud infrastructure, and wrote the initial implementation of a JavaScript/AJAX client interface, using jQuery, Underscore.js, and Backbone. The Clojure code interoperated with Java libraries (Jetty, Apache Commons, Lucene, IM4Java, TagSoup); I fixed bugs in these libraries as needed.

**Environment:** Clojure, Java SE 6, Maven, Redis, Jetty, PostgreSQL, Amazon AWS (EC2, S3 using Jets3t), JavaScript, jQuery, C, JNA, Git, FogBugz, Linux, Mac OS X

### Sep 2008 – Aug 2009, Software Engineer, ALX

Designed and implemented AJAX intranet applications in Python/Django, PHP, JavaScript, SQL, liaised with customers and provided technical support. The design of most of these applications was database-driven – data logic was written in PL/pgSQL, and the UI used an in-house open-sourced AJAX library (Kato) that almost directly manipulated the SQL tables and views.



Daniel Janus

[dj@danieljanus.pl](mailto:dj@danieljanus.pl)

[www.danieljanus.pl](http://www.danieljanus.pl)

(+48) 501-180-396

**Address**

Béli Bartóka 8/127

02-787 Warsaw

Poland

**Customers:**

- *Marcel* – a solution for managing the process of mandatory veterinary inspections of cattle in Poland, deployed in several Polish voivodeships (programs of prevention of enzootic bovine leukemia, Bang's disease and BSE). The project used YUI instead of Kato at the customer's request; I implemented a Kato-like framework on top of YUI, which facilitated rapid development of new functionality.
- *Makro* – a system for managing the service requests of a major Polish warehouse network, allocating time to servicants and tracking their work. I co-designed and implemented this system using Kato, Django (making heavy use of Django templates), and PL/pgSQL.

**Environment:** JavaScript, Prototype, YUI, PostgreSQL, Python, Django, SQLAlchemy, PHP, Subversion, Mantis, Linux

**Nov 2005 – Aug 2008, Software Engineer / Quality Assurance Specialist, Sentivision**

As a SE, I created software in C and C++ for various parts of VOD/IPTV systems (client software running on set-top-boxes, RTSP streaming server; sample tasks: implementing ISMA payload type in the streaming server, making it compatible with QuickTime players; implementing a TV widget with channel switching).

I also worked on an IPTV middleware server that managed the TV channels, VOD content and other data sets that were then sent to set-top-boxes using XML-RPC. The server was initially written in Common Lisp with its Web interface written in the UCW continuation-based framework, then rewritten in Java EE to provide zero single points of failure (designed from scratch to work on Glassfish clusters). It also was designed to be modular – it included a flexible core and allowed for optional modules to conform to particular customers' requirements.

As a QAS, I designed, implemented, deployed and maintained an in-house testing infrastructure (end-to-end/regression testing) for complex VOD/IPTV systems. The components of that infrastructure included:

- A system for automatically running test scenarios written in a DSL created on top of Common Lisp – the scenarios could include interaction with Web applications, set-top-boxes (remote control simulation), or mix both
- An isolated testing environment consisting of a number of devices (STBs and PCs) and a mechanism of remotely controlled execution of arbitrary software on any of these devices
- An automatic build system, written in Python, that continually built all the projects from all branches and created build reports

I also reviewed co-workers' code, co-authored and proofread technical documentation (in particular, wrote an extensive manual for the automated tester described above).

**Environment:** C, GDB, GNU make, Valgrind, ElectricFence, C++, STL, Java EE 5, EJB 3, JUnit, Glassfish AS, Tapestry, Ant, Common Lisp, UCW, AllegroCache, CVS, Darcs, Mercurial, PostgreSQL, pgPool, Mantis, FogBugz, Python, Ruby, Linux



Daniel Janus

[dj@danieljanus.pl](mailto:dj@danieljanus.pl)  
[www.danieljanus.pl](http://www.danieljanus.pl)  
(+48) 501-180-396

#### Address

Béli Bartóka 8/127  
02-787 Warsaw  
Poland

## Feb 2004 – Sep 2005, *Programmer, Institute of Computer Science, Polish Academy of Sciences*

Main author of large corpora searching and indexing software, Poliqarp (now open-source), used in the IPI PAN Corpus, National Corpus of Polish and the Portuguese LX-Corpus. I initially wrote a GUI for the application in Java and Swing, a Web version in PHP, then redesigned the entire suite to give its architecture more consistency and implemented an indexing mechanism that yielded a speed-up by up to two orders of magnitude. Some of the work was done within my M.Sc. thesis that was a continuation of this work.

## Open Source

I am the author of several open-source projects, hosted at GitHub (<http://github.com/nathell>), including:

- *clj-tagsoup*, a Clojure interface to the TagSoup HTML parser that can automatically detect character encodings;
- *Skyscraper*, a Clojure library that assists scraping entire websites in a structural way;
- *Smyrna*, a search tool for hand-built corpora of Polish;
- *Lithium*, a Clojure-based x86 assembler and toy Lisp compiler;
- *clj-iter*, a library for iteration over multiple collections at once in Clojure, similar in spirit to Common Lisp ITERATE;
- *clj-bitfields*, a way to access C99-compatible bit-field structures in Clojure; Sunflower, a tool for extracting essence from a set of similarly-structured HTML documents.

I have contributed bug fixes and reports to a number of open-source libraries.

## Education

### 2000–2006, *University of Warsaw, M.S. in Computer Science*

(M.S. thesis „Methods of searching large text corpora” defended as very good)

### 1996–2000, *II Lyceum of Starachowice*

## Languages

Polish	native
English	fluent written and spoken
German	basic
Czech	basic

## Other

### Soft skills

Creative, excellent learner, good team player, non-violent communication practitioner

### Conference talks

Clojure eXchange 2016, EuroClojure 2016, 4Developers 2016, Lambda-Days 2014, EuroClojure 2013

### Interests

Literature, poetry translation, Scrabble, cycling, hiking, cats