Michiel Joris van Oosten

 $1/2 \text{ Rosslare Ct} - \text{Kuluin QLD 4558} - \text{Australia} \\ \text{\mathbb{P}} +61 0480 200 584 \quad \bullet \quad \boxtimes \text{mail@axion5.net} \quad \bullet \quad \text{$^{\bullet}$} \text{ www.axion5.net} \\ & \text{\mathbb{P}} \text{ michieljoris} \quad \bullet \quad \text{\mathbb{D}} \text{ michieljoris} \\ \end{aligned}$

Company Recruitment team

April 11, 2020

Dear Sir or Madam,

I am a Dutchman who has been living in Australia since 1994.

I am self-taught in both programming and piano. I began programming in my early teens and started playing piano at nineteen. I finished a conservatorium education in Australia in my mid twenties. Programming has been an undercurrent throughout my whole life, and after having explored music for some time I focused on programming since about 2012-13.

I hope to bring thoroughness, ideas and architectural (and artistic) insight to a team and projects. I enjoy designing and thinking about long term design and implementation of these designs.

My coding environment: laptop running Linux (Ubuntu), i3wm, Emacs and Chrome/Firefox, my own tools. All my code is on github. A guide is on my site at axion5.net/my-github.html.

Some more background info:

- I went to VWO (Science preparatory school), courses in maths, history and philosophy at Universities of Leiden and Amsterdam. Degree in piano at Brisbane Griffith Conservatorium.
- Favorite language is lisp after working through TEOCS in Common Lisp.
- Wrote my own server, build and module system. It does for me what most people use Express, Grunt/Gulp/Brunch and Browserify for.
- Java and C++ put me thoroughly off OOP. Functional approach is better.
- Studying CS curriculum when time allows (SICP will be finished one day)
- Working knowledge of html and css (and sass and less etc), but not overly fond of it.
- Would love to write in and learn more about Haskell and Erlang, but also D, Rust and Julia.
- Dream project: write Lisp operating system...
- Major long term interest: computer systems' software reliability, robustness. Ultimately software should be somehow self repairing, self-reliant and with built-in redundancy, and as a whole extremely fault tolerant.

Some more background info is on careers.stackoverflow and linkedin. Yours faithfully,

Michiel Joris van Oosten

Attached: curriculum vitæ