HUGO KLEPSCH

905 · 749 · 4627 ◇ hugo.klepsch@gmail.com
5663 Appleby line ◇ Burlington, ON L7M0P3

EDUCATION

University of Guelph September 2015 - April 2020
B.Comp. Software Engineering (Co-op), Honours *Guelph, ON*

- Minor in Marketing
- Dr. Mary McLeish Scholarship recipient (Highest GPA in Software engineering major)
- 2018 Braithwaite Business Scholarship
- Deans list

EXPERIENCE

Arctic Wolf Networks May 2018 - August 2018, January 2019 - August 2019
Member of technical staff *Waterloo, ON*

- Used Esper, Apache Flink and Hadoop as part of a complex event processing pipeline to find patterns in unbounded series of events.
- Part of team that pioneered construction of time-series anomaly detection system.
- Wrote independent auto-scaling services as part of a data processing pipeline that processed 8 billion messages per day.
- Added metrics and stability alerts to services.

Carnegie Technologies May 2017 - December 2017
Native back-end developer *Waterloo, ON*

- Developed native C++ GPS and ephemeris libraries for use in embedded devices.
- Participated in the design of, and implemented REST style microservices using Node.js and RabbitMQ.
- Designed and implemented C++ and Node.js RabbitMQ messaging library with support for a variety of usage patterns (Consumer, Requester).
- Designed and implemented C++ JSON manipulation and validation library with support for proprietary extensions to JSON schema specification.

OPEN SOURCE CONTRIBUTIONS

avast-tl/retdec
December 2017
- Proposed and added Docker support

netmail-open/wjelement
November 2017
- Found and fixed bug causing library not to compile with glibc
- Added date-time support to JSON schema verification system

DefinitelyTyped/DefinitelyTyped
July 2017
- Added typescript type definitions for various Node.js libraries

VOLUNTEER HISTORY

Alumni and programming mentor, former student member September 2013 – Present
M. M. Robinson high school's robotics club, "Rambotics" *Burlington, ON*

- Created various sub-systems for functional mechanisms
- Used PIDF closed-loop control, computer vision, motion profiling, path following, etc. for autonomous control of robot
- Used encoders, potentiometers, limit switches, line followers & cameras as input data for control loops
- Taught high-school students about control flow, program structure, git & the above

VOLUNTEER HISTORY

Computer Languages	Prolog, Haskell, AWK, Erlang, Scheme, ML
Protocols & APIs	XML, JSON, SOAP, REST
Databases	MySQL, PostgreSQL, Microsoft SQL
Tools	SVN, Vim, Emacs