Résumé for **Erik Wilson**

Erik.E.Wilson@gmail.com

CAREER SUMMARY

Software Engineer with many years of experience working with teams of varying size, knowledge, and locality. Exposure to a wide variety of languages, architectures, operating systems, libraries, and supporting software, from national supercomputer systems to microprocessors. Specialize in researching and developing solutions for difficult problems in business and scientific software design. Always expanding breadth and depth of knowledge while seeking excellence in engineering.

SKILLSET SUMMARY

Languages: Rust, Go, Java, C, C++, Lua, Javascript, Python, Perl, Shell Scripting
Software: Kubernetes, Docker and ContainerD, Linux, MacOS, Windows, CI/CD

WORK EXPERIENCE

Software Engineer   Tempe, AZ

Red Canary   2022 - Present
- Multi-arch migration of Kubernetes services to reduce operational costs.
- Rust Linux EDR sensor development utilizing eBPF and Golang services.

SUSE - Rancher   2018 - 2021
- Feature development and bug fix support for Rancher server, a platform for launching and management of custom Kubernetes distributions on cloud providers.
- Developer and maintainer for K3S, an open source and lightweight Kubernetes distribution with a focus on ease of use and edge capabilities.

Nextiva   2018
- Frontend development with React and Javascript, implementing design requirements for user identity and access management administration.
- Backend support with Python and Java, code reviews, debugging, and bug fixes.

Citrix - Octoblu Inc   2014 - 2017
- Assisted in architecting and developing an Internet of Things platform.
- Frontend development with Angular and React, backend development using Docker with Javascript (Node.js, CoffeeScript, ES6). Devicec integration with a variety of languages, including: Node.js, Lua, Java, Groovy, and C++.

Iced Development   2012 - 2014
- Software engineering and server administration services for a global Advance Deposit Wagering platform. Use of RedHat Enterprise Linux, Apache, Tomcat, MySQL, MongoDB, Java, JavaScript, and Node.
- Assist in essential implementations for product launch. Integration with partner totalizer and deposit systems. Bottleneck and optimizations analysis and resolution post-launch.

Software Engineer, **Boston University**   Boston, MA

Center for Integrated Space Weather Modeling (CISM), Astronomy   2005 - 2013
- Port various models and scientific packages to other platforms (e.g. LLNL's A++P++, Overture, and PnMPI to AIX and Cray; IBM's OpenDX to OSX).
- Maintain core infrastructure of hardware and software (use of RAID; Linux, Apache, MySQL, PostgreSQL; Perl, Python, BASH scripts; Subversion, Mercurial, Git; OpenLDAP; Make/GNU toolchain; Intel, PGI, IBM, & GNU compilers; OpenMPI, MPICH2, NetCDF, HDF).

John Lyon (LFM), Astronomy   2008 - 2013
- Performance analysis and optimizations of the Lyon-Fedder-Mobarry (LFM) magnetohydrodynamics model. Analysis requires performance reviews of the LFM model and associated libraries on various national supercomputer systems, from CPU cache management to MPI communication inefficiencies.
- Developed a C++ I/O library to utilize parallel file systems on supercomputer platforms, allowing unified access to HDF4 or HDF5 through a common API. Optional A++/P++ support allows for data super-domains and automatic array meta-data extraction.

Harlan Spence (NASA/LRO/CRaTER), Astronomy   2007 - 2010
- Creation of TCP/UDP socketed Perl server used to decompose, calibrate, and redistribute real-time network data from the Cosmic Ray Telescope for the Effects of Radiation (CRaTER) instrument on the Lunar Reconnaissance Orbiter (LRO).
- Refactor of C++ data pipeline used on raw data received from the Mission Operations Center to create calibrated multi-level scientific data sets for scientific analysis and inclusion in NASA's Planetary Data System archive.

Nathan Schwadron (EMMREM), Astronomy   2007 - 2009
- Developed a C++ I/O library for the Earth-Moon-Mars Radiation Environment Module (EMMREM), which ingests a text-based start-up configuration and periodically dumps multi-processor snapshots of the model's simulation state, uses MPI and the NetCDF3 API.

David Coker Group, Chemistry   Summer 2004
- Reimplementation of a FORTRAN 77 quantum monte carlo particle simulation to more compact and extensible modular Fortran 90 framework.

EDUCATION **Northern Arizona University**   Flagstaff, AZ
Baccalaureate of Science in Computer Science and Engineering   2000 - 2004
(BSCSE, ABET accredited) with minor in Linguistics

OTHER *Artificial Intelligence*
EXPERIENCE
- Participant in 2011 Google Ants AI challenge. Using C++, Python, and JavaScript to test and visualize novel hill climbing algorithms.
- Cognitive and Neural Systems (CNS510) at Boston University, development of a general purpose ordinary differential equations solver using the Runge-Kutta method and creation of a C++ interface to the GNU Scientific Library. Used to solve leaky integrator type neural models.

Electronics
- Assisted in design and development of CactusCon electronic badges, including a USB3 breakout and ethernet tap designed in Eagle, and custom ULP for importing vector graphics from InkScape. Programming for ESP8266 and ESP32 devices using NodeMCU and Lua.
- Arduino and similar microprocessor C/C++ development; analysis of sensor input and stepper motor control.

Volunteer Experience
- Member of HeatSync Labs, a volunteer based maker community in Mesa, AZ.
- Event volunteerism, including HackPHX, NodeBots, DefCon, and CactusCon.

Github repository available at https://github.com/erikwilson.