Jason Sun's Resume

sunapi386@gmail.com sunapi386.ca linkedin.com/in/sunapi386 +1 (408) 599-0428 28/11/2021

Profile

A generalist with expertise in a wide range of software stacks, leading cross team efforts to design and integrate multiple tech stacks to form new features in an architect role.

- Experienced in leading projects, project management, leading cross functional team efforts, design docs, building org-wide processes.
- Experience in real time systems, full stack web, full stack robotics, cryptocurrency blockchain.
- Mentoring and manage engineers/interns, experienced interviewer.
- Wide range of software expertise, as an architect role, and leading efforts to design and integrate multiple tech stacks to form new features.
- Create designs for large features, break down and delegate parts to cross team and team engineers.
- Experienced with complete product development lifecycle, from conception, to ramp up for shipping, support and maintenance for end of life.
- Product development from concept to production. Idea to concept at AutoX, new product development at Amazon, mature iPhones at Apple.
- Experienced working in fast paced startup environments, as well as large corporates.
- Have been coding since 2004 in the early days on Slackware Linux and Arch Linux.
- Cardano cryptocurrency stake pool operator and developer.
- Driven and self-motivated personality with many hobbies (see blog at *blog.sunapi386.ca*).

Education

- *University of Waterloo* Bachelor of Computer Science, 2015. Computer Science Exchange Student at *EPFL (Swiss Federal Institute of Technology at Lausanne).*

Work Experience

- Software Engineer *Amazon Lab 126* June 2019 – present Sunnyvale, California

Worked on developing Amazon Astro. It's a home robot which it enables Alexa to take a physical form and have mobility. Develop platform infrastructure on real time mapping pipeline. Worked on this product for over 2 years, from prototype to production. Use ROS, C++, Python Java, full stack web. Mentored junior engineers and interns. Create new architecture and drive cross functional team features.

- Autonomous Systems Engineer *AutoX Technologies Inc.* in August 2017 – May 2019 San Jose, California

HD mapping infrastructure working with point clouds for large scale map building used in localization, with Hadoop.

Lead efforts to build API backend for location-based delivery service AutoX apps. GraphQL, Node.js, React. General architect role, lead team efforts to design and integrate tech stacks for product feature development using C++, Python, Javascript. Full-stack autonomous system integration, low/high-speed delivery vehicles.

Experience understanding of SLAM, LOAM, lidars, cameras, radars, ultrasonics.

Delivered features: remote control (networked video streaming and throttle/steering control), hybrid human-and-AI decision making for unhandled driving scenarios (in patent process), design and implement vehicle-to-cloud APIs (LTE networked), embedded systems with touchscreen UI.

Cluster computing automation using Docker, Kubernetes, ROS (Protobuf). Point cloud HD mapping, localization, vehicle remote vehicle control, hybrid human-vehicle interactions, cloud to vehicle API, big data.

- Software Engineer *Apple Inc.* in November 2015 – January 2017 Cupertino, California

Improved existing product reliability tools, conceived and built full-stack website (Rails API & Ember.js & nginx & mySQL) for managing iOS devices. Created client-sided task runners for delegating test-work, where work jobs were queued from the website. Also worked on internal iOS and macOS apps (Swift & Objective-C).

- Undergrad Research Assistant at University of Waterloo in May 2015 – September 2015 Waterloo, Ontario

Worked with my AI professor and a 6 person team to develop a data tool for analysis of chat logs, and built a search engine prototype (using TF-IDF in Apache Lucene). Prototyped a search engine for customer support chat dialogues with Apache Lucene using TF-IDF indexing for feature recognition and searching.

- Software Engineer Intern *Shutterfly Inc.* in July 2014 – August 2014 Redwood City, California

Design and implemented a distributed REST API service, in Java/Scala based upon the Apache Cassandra database. The API was designed by myself, with guidance from full-time employee members of the web infrastructure platform team, to be used by other Shutterfly services. Create additional network load tests for our distributed image hosting service.

- Software Developer at *Encircle Inc.* in May 2014 – June 2014 Kitchener, Ontario

Worked on the website (CoffeeScript & Python Tornado server) and Android app, at a 3 engineer startup in the *Velocity Garage*, which is a University of Waterloo startup incubator.

- Software Tools Developer Intern at RIM (BlackBerry) in September 2013 – December 2013 Ottawa, Ontario

Built internally used features to GitLab (Ruby on Rails) and helped with database migration from Github to Gitlab. Developed a testing framework for integration and regression testing website user interfaces using the Selenium Webdriver.

- Physics Teaching Assistant at *Wilfrid Laurier University* in September 2011 – April 2012 Waterloo, Ontario

Developed a spectrometer reading program in Python, using the pySerial library, to automate the reading of lab samples, and generate a spreadsheet file. Supports multiple spectrometer readings in parallel.

Related Classes

- Real-time Operating Systems (CS 452): Built real-time operating system from scratch to control model trains on a track.
- Artificial Intelligence (CS 486): Machine learning, probabilistic models, intelligent agents.

- Advanced Algorithms (EPFL CS 450): A graduate course in algorithms, learned theoritical techniques and their applications to solve problems. Interesting techniques such as network flow, randomization, dynamic programming.
- Distributed Systems (CS 454): Studies on concurrency, problems and solutions with distributed systems in a network.
- Computer Architecture (CS 450): Designed CPU pipeline in Verilog, supporting 8 instructions for computer architecture class. This is sufficient to run machine code produced by the MIPS compiler, from the CS 251 compilers class.