

# Raymond Kroeker

<https://raykroeker.net/>  
[raymond@raykroeker.net](mailto:raymond@raykroeker.net)  
[+1-604-308-0700](tel:+1-604-308-0700)

## About

I'm a software engineer in Vancouver, BC. I have deep experience building distributed systems, data stores, and closed-loop infrastructure control platforms. I'm familiar with multiple languages, protocols and skilled in agile development methodologies and project ownership. My focus is building reliable, scalable platforms and services.

## Experience

**Salesforce:** December 2018-Present

### **Software Engineer Architect Infrastructure**

- Led the definition and refinement of an architecture and roadmap for the control plane platform managing the bare-metal lifecycle for all of Salesforce.
- Led the implementation of OS patching for hundreds of thousands of physical servers intersecting hundreds of roles across multiple data centers.
  - Led dozens of developer (service owner) teams to provide health signaling for their services.
  - Define standards for OS integration and service maturity within Salesforce.
  - Improve security compliance from <60% to >90%.
  - Improve time-to-delivery from *3 months* to < *7 days*.
- Design, implement and operate distributed key/value store.
  - Increase the breadth of consensus for writes across the cluster.
- Led multiple teams to design and implement autonomous software release.
- Led the design and implementation of a framework and platform for scalable, reliable lambda execution.
- Mentor senior engineers and lead cross-team architecture and design discussions.

**eBay:** September 2017-November 2018

### **Software Engineer Architect, Structured Data**

- Design and implement a distributed, scalable knowledge graph data store.
- Implement a query-language and parser for SPAQL-like queries.
- <https://github.com/eBay/beam.git>
- <https://innovation.ebayinc.com/tech/engineering/akutan-a-distributed-knowledge-graph-store/>

**Salesforce:** December 2012-September 2017

### **Software Engineer Principal Member of Technical Staff**

- Design, implement and operate provisioning API for bare metal servers and switches operating hundreds of thousands of devices.
  - Reduce time-to-delivery for data center OS build from 2 months to < 12 hours.
  - Improve reliability of delivery from ~70% to 90%.
- Design, implement and operate a closed-loop control system for bare metal infrastructure.
- Design and implement a RESTful API for signal ingress and consuming/completing work for infrastructure change.
- Design and implement a work scheduler for an infrastructure control plane.

**Ping Identity:** March 2009-October 2012

**Lead Software Engineer**

- Led the design and implementation of various microservices within PingOne's initial Software-as-a-Service offering.
- Design and implement the data model and access layer for PingOne.
- Led the design and implementation of an internal testing platform and framework for the PingFederate product, serving the development team.

**thinkParity Solutions:** September 2005-December 2008

**Software Engineer Founder / Architect**

- Design, implement and operate a SaaS document management system.
- Implement a service to synchronize and manage documents w/in S3.
- Implement a desktop client to browse, share and synchronize documents.

**Gemcom Software:** January 2004-September 2005

**Senior Software Engineer**

- Build a data warehouse enabling customers to optimize their extraction logistics.
- Develop CI pipeline automation for ProdTrak, GEMS and Whittle in Java/C/C++/Visual Basic/Fortran.

**Correlation Technologies:** April 2000-December 2003

**Software Developer**

- Implement project management software as a service for the insurance claims vertical.
- Implement a pub/sub message and notification service.
- Implement project management software as a service for a restoration vertical.

## Education

**CDI College of Business and Technology**

Technology Diploma: Programmer Analyst

## Skills

Languages (Go, Ruby, Java, C/C++), databases (Oracle, PostgreSQL, MySQL), key / value stores and database internals (Bolt, RocksDB), distributed systems, consensus algorithms (Raft, Paxos), authentication (SAML, OAuth), API and protocol design and implementation (REST, gRPC, HTTP), bare-metal and public-cloud infrastructure (AWS), software development methodologies (agile), micro-services, software-as-a-service, service and project ownership.