

### Skills

- Rust** Expertise in high-performance, low latency, low-level systems development with Rust, including everything from fundamental data structure implementation to asynchronous distributed systems development
- Clojure** Extensive experience architecting, deploying, and building complex web applications in Clojure and Clojurescript, with a focus on Re-Frame and Reagent. Experience testing distributed systems in Clojure using Jepsen.
- Haskell** Passionate love for pure functional programming as a hobbyist pursuit, but also practical experience building production systems in Haskell at scale, and using Haskell's advanced type system extensions where appropriate to deliver increased ergonomics and safety.
- Nix** Experience with adopting and teaching nix at scale in a production stack both for local development dependencies and for configuring and building production software. Core contributor to a fork of the nix implementation itself (tvix) aimed at providing increased safety, performance, and flexibility.
- Unix/Linux** Experience with administrating highly available distributed systems. Passion for the Unix philosophy of discrete, composable units of functionality.
- Ruby** Experience building both full-stack applications with Ruby on Rails in addition to smaller microservices and custom frameworks. Deep understanding of the internals of the Ruby interpreter and object system.
- Javascript** Experience developing real-time responsive single-page web applications using React, in addition to significant contributions to the React open-source community.
- SQL** Deep understanding of relational databases as an implementer, in the context of an innovative new database implementing a query planner and incremental materialization for the PostgreSQL and MySQL dialects of SQL from the ground up – and of course also a user

### Additional Tools

- Vim ◦ Emacs (yes, also) ◦ Kubernetes ◦ Git ◦ Terraform ◦ AWS ◦ GCP ◦ Datomic ◦ Elasticsearch ◦ Redis ◦ Docker
- Java ◦ Scala ◦ QuickCheck (and similar tools) ◦ Jepsen ◦ Python ◦ Elixir
- Novice Level:** ◦ C++ ◦ Erlang ◦ Prolog ◦ Idris ◦ Agda ◦ Tensorflow

---

## Experience

- 2020–2023 **Staff Software Engineer**, *ReadySet*, Remote.  
Founding engineer at a startup bringing a high performance partially-stateful, incrementally-maintained SQL database based on the Noria thesis to market
- Served as the main technical leadership for the project throughout its maturation from a research codebase to a production-grade system
  - Extended the Noria PhD thesis by implementing methods from multiple research papers, masters theses, and other papers from database research, in addition to original database research and development.
  - Invented or helped develop multiple novel database techniques in partially materialized dataflow, including index planning and selection, pagination, post-lookup aggregate processing, partial “straddled” joins, weak indexes for correct execution of partial joins, and more.
  - Invented novel ways to test SQL databases, including a new deterministic generator for SQL queries.
  - Developed the clustered high availability distributed runtime mode from a buggy research feature into a production ready distributed system that passed a suite of Jepsen tests.
  - Implemented a significant fraction of the SQL query planner, which required both implementing algorithms specified in database research papers and inventing new techniques to work around the limitations of partially materialized dataflow
  - Optimized critical components of the code base, including algorithmic optimizations, CPU cache analysis, low-level data structures, and broad system runtime analysis
  - Implemented a type inference engine and expression evaluator that supported multiple dialects of SQL configured at compile-time, with maximum code reuse while preserving maintainability
  - Mentored multiple junior and senior engineers
  - Open-Source contributions visible at <https://github.com/readyssettech/readysset/commits?author=glittershark>
- 2019–2020 **Engineering Manager**, *Urbint*, New York, NY.
- Lead of the platform team with two direct reports - a senior SRE and a senior software engineer.
  - Performed user research on developers, project managers, product managers, and other internal stakeholders to build the roadmap for the platform team.
  - Built and maintained a system to deploy one-off full stack application instances from pull requests to enable easier testing.
  - Led a large, multi-project migration between CI systems that resulted in a decrease of average build times from 2 hours to less than 10 minutes.
  - Maintained and extended Nix-based build and development infrastructure for both software engineers and machine learning engineers.
- 2018–2019 **Senior Software Engineer**, *Urbint*, New York, NY.
- Built, trained, and maintained a large, deep-learning-based image-detection model for semi-automated (human-in-the-loop) video classification.
  - Designed, built, and maintained a novel in-house tool for collection of training data.
  - Maintained and guaranteed reliability of a large data pipeline for video processing and classification.
- 2017–2018 **Senior Software Engineer**, *Urbint*, New York, NY.
- Integral in the architecture of a novel, serializable ACID transactional graph database built on RocksDB, first in Elixir then in Haskell.
  - Helped ship customer deliverables involving multi-day data processing jobs for disparate data sources.
  - Instructed other developers in the use of and theory behind Haskell
  - Brought computational graph theory to bear on the problem of unifying disparate, highly heterogeneous data sources across the world of open data.
- 2016–2017 **Senior Software Engineer**, *SecurityScorecard, Inc.*, New York, NY.  
Lead frontend developer for a rapidly-moving and growing security software startup.
- Took part in collaborative product design meetings to make UX tradeoffs with product designers and managers.
  - Drove application architecture for a large, complex, data-driven frontend application.
  - Championed increased use of production monitoring and alerting.
  - Worked with business stakeholders to set long- and short-term priorities for application development.
  - Mentored junior team members.

- 2015–2016 **Lead Developer**, *Nomi, Inc.*, New York, NY.  
 Lead web services developer transitioning to a full-stack role implementing shared software components and architecting a large, complex microservices application ingesting hundreds of gigabytes of IoT data per week.
- Lead application architecture of the majority of the backend services to encourage consistent REST API design and code sharing.
  - Championed the use of Haskell for rapid, safe development of the API Gateway service.
  - Took ownership of operations and server maintenance of a >100-instance AWS account using Puppet.
- 2014–2015 **Lead Developer**, *LandlordsNY, LLC*, New York, NY.  
 Sole engineer for a small startup connecting landlords and property managers and facilitating the online sharing of information in a historically technology-averse industry.
- Drove product design, visual design, and UX architecture for a major revamping of the core product.
  - Interfaced with customers to set priorities for new feature development.
  - Conducted hiring and recruiting to build out an engineering team.
- 2012–2014 **Associate Developer**, *Visionlink Inc.*, Boulder, CO.  
 Integral member of an agile development team building the nation's most-used Information and Referral platform for organizations such as United Way Worldwide and the American Red Cross.
- Refactored and revamped legacy code to increase performance and long-term maintainability.
  - Worked on several triage-teams to rapidly fix production bugs with strict deadlines.
  - Built a complex, yet highly-performant tool for searching human services by category.
  - Acted as a core designer and developer of a major product revamp.
    - Drove a complete rethinking of the data model in the product, leading to greater unification, simplicity, and consistency;
    - Championed the adoption of a test-driven-development model;
    - Drove product documentation and code standardization.

## Project Highlights

- **How much does Rust's bounds checking actually cost?** – <https://blog.readysset.io/bounds-checks/>  
 Blog post providing a deep evaluation of the runtime cost of bounds checking in safe languages like Rust. Front page of Hacker News, doubled month-over-month ReadySet waitlist signups
- **Tvix** – <https://tvix.dev>  
 Modular Reimplementation of the Nix build tool in Rust
- **Panettone** – <https://code.tvl.fyi/tree/web/panettone>  
 Aggressively simple bug-tracker developed in Common Lisp for the community involved in the development of Tvix. Hosted at <https://b.tvl.fyi>
- **Org-Clubhouse** – <https://github.com/glittershark/org-clubhouse>  
 Emacs library for integration between org-mode and the Clubhouse issue tracker
- **Github Bug Bounty** – <https://bounty.github.com/researchers/glittershark.html>  
 Discovered and responsibly disclosed a persistent XSS on Github's main website
- **core-async-storage** – <https://github.com/glittershark/core-async-storage>  
 Simple Clojurescript wrapper around React Native's AsyncStorage using core.async