

Simon Katz's CV

(Latest PDF version is at [Simon Katz's CV as a PDF file.](#))

Personal Details

Simon Katz, London, UK

Email: cv-enquiry@nomistech.com

Blog: blogish.nomistech.com

GitHub: github.com/simon-katz

Objective

I'm interested in:

- development work using Clojure and/or ClojureScript
- passing on knowledge of Clojure and development in general to others
- improving my knowledge of DevOps and cloud computing.

Overview

I'm an intelligent, personable, self-motivated software professional with excellent written and spoken communication skills. I've used incremental and iterative approaches supported by automatic tests for over 20 years, and since 2011 I've worked in environments that make use of modern agile and DevOps ideas.

I like Clojure; I've used it since 2011 and before that I used Common Lisp for many years. (It may be worth saying that Clojure is a modern Lisp.) I've also done many things that aren't Lisp.

I have extensive experience of the whole software development lifecycle from business analysis through to delivery to users and subsequent support and maintenance.

I've worked on web, desktop and command-line-driven applications, in application areas including telecommunications, publishing, software tools that support design processes and software for software people (eg application generators and compilers).

I'm a natural teacher and coach, and I'm calm, well-organised, even-tempered and fair-minded.

Summary of Technical Skills

Process, Design Approaches, etc

- Agile development (eg pair programming, incremental and iterative delivery, close collaboration with the customer, continuous integration, sometimes continuous delivery).
- DevOps and cloud computing: I'm not an expert, but I've been using DevOps approaches since 2011 and AWS since early 2018.
- REPL-based development backed by automatic tests for fast feedback at a micro level.
- Various design approaches including microservices, domain-driven design (DDD), test-driven development (TDD), behaviour-driven development (BDD) and object-oriented design (OOD).
- Programming experience ranging from assembler through to very high-level languages, including functional programming.
- Knowledge of theoretical computer science (analysis of algorithms and complexity theory, theory of computation) and discrete mathematics.

Programming Languages etc

- Clojure and ClojureScript: I've been using Clojure professionally since 2013, and in my spare time since 2011. I'm familiar with Emacs, CIDER and Leiningen, and Clojure libraries including tools.namespace, Component, Integrant, Midje, Schema, Compojure, Ring, clj-http, HTTP Kit, Hiccup, Cheshire, core.async, ClojureScript, Om and Reagent.
- Lisp: Common Lisp and CLOS (13 years); Interlisp (2 years); Emacs Lisp (off and on since 2011).
- Bash (off and on for as long as I can remember).
- Other programming languages: C (3 years); C++ (4 years); Java, Scala, Prolog (6 months each).
- Web technology: HTML, CSS and JavaScript.

Miscellaneous

I've used many tools over the years. Some of the more important and recent ones are:

- Git, Docker, Jenkins, CircleCI, Jira.
- AWS (eg EC2, ECS, RDS, S3, Athena, DynamoDB, Kinesis).
- SQL.
- Telegraf, Influx and Grafana (metrics collection, storage and visualisation tools).
- Various Unix & Linux systems (20+ years) and Windows (~10 years).

Work History

Sept 2014 – Current Contractor

See separate entries below for details.

Feb 2018 – March 2019 Contractor at YouView

Cloud Software Engineer

YouView provides smart set-top TV boxes. I worked on cloud services that support the display of various types of message on set-top boxes.

- Refactored existing services. Improved the design at multiple levels, and made it more visible. Made the services more maintainable and more testable. Increased test coverage from nowhere-near-enough to a good level. When I started, it was difficult, slow, and scary to make changes; when I finished, the code was pleasant to work on and changes could be made with confidence.
- Set up various monitoring dashboards and alerts in SignalFx.
- Improved performance of message querying by a factor of ~100.
- Added new features.
- Dealt with bugs and system issues.
- Began implementing a new system to replace the existing message system. The new system allows messages to be defined in a more convenient manner and has an asynchronous architecture that better suits variable workloads. Worked on architecture, stories, and the beginnings of implementation.
- Tech stack included Clojure, CircleCI, Docker, AWS (eg EC2, ECS, RDS, S3, Athena, DynamoDB, Kinesis), and SignalFx.

Apr 2015 – Aug 2017 Contractor at HSBC

Clojure Specialist, Coach and Developer

Part of a small team that supported a large application team in moving towards DevOps-based development.

- Working closely with an ops person who was learning software development, developed a set of services and tools to support the collection and use of metrics produced by application services.
- Working closely with a support person who was learning software development, produced a tool that allows information about support incidents to be entered in a single place, and syncs information to and from other systems. Previously support staff spent silly amounts of time doing things manually.
- Wrote a web client and server app that displays a summary of the department's deployed services. The app highlights differences across hosts and across environments, and allows the user to filter by criteria such as service name, host name, environment, and consistency of versions across hosts and environments.
- Produced a template that allows new Clojure applications to be created easily. The template provides a set of standard functionality including setting up logging, dealing with exceptions, providing a health endpoint and making REPL-based development work nicely. Previously it took a lot of work to produce a new app that just did these basics, and most of our apps didn't do everything properly.
- Produced universally highly-regarded Wiki pages that explained the design and use of the tools.
- Coached developers and Ops people in the effective use of Clojure.
- Technologies included microservices, Clojure, ClojureScript, Leiningen, tools.namespace, Component, Plumatic Schema, Midje, Compojure, clj-http, core.async, Reagent, Telegraf, Influx and Grafana.

Sept 2014 – Dec 2014 Contractor at uSwitch.com

Developer on the back-end system of a comparison website for mobile phone deals, filling in while a team member was on paternity leave. The system processes feeds that define suppliers' deals, transforming the data in each feed so that all deals are described using the same format. The transformed data is used by the front-end system.

Contributed to re-writing an old Ruby system in Clojure and maintaining the old Ruby system. Technologies included microservices, Clojure, Leiningen, Component, Plumatic Schema, Bidi, Liberator, core.async, ClojureScript and Om.

Feb 2014 – Aug 2014 Bwin Party

Developed a microservice that obtains tweets in real time from Twitter's streaming API and passes them on to a client. Used a Clojure stack including core.async, HTTP Kit, WebSockets, Compojure, Ring, Cheshire, Leiningen, tools.namespace and Midje.

Coached a development team new to functional programming, getting them up to speed with Scala and core functional programming concepts; also coached on general design and style issues that are not specific to functional programming.

My department ceased to exist as part of a broad range of cost-cutting measures and I was made redundant.

Nov 2011 – Jan 2014 ThoughtWorks

Learnt modern development approaches such as test-driven development, pair programming, web applications, microservices, continuous integration and continuous delivery.

- Developer on a system that assesses applicants' eligibility for loans, taking into account legislation and personal circumstances. Technologies included microservices, Clojure, Leiningen, tools.namespace, Midje, Compojure, Ring, Hiccup, Cheshire, Java, Jersey and Jackson.
- Developer on an academic publisher's content delivery system. Technologies included microservices, Scala, Scalatra, Scalate, XML and XQuery. Responsible for implementing stories and liaising with business analysts, quality analysts and with other developers. Some element of mentoring client staff to improve their use of TDD and bring in a more domain-driven approach.
- Developer on a system that administers pension funds. Technologies include Java, JavaScript, Hibernate and Oracle. Responsible for implementing stories and liaising with business analysts, quality analysts and with other developers.

2000 – 2011 Contractor at Mobinets

An application generator:

- Building on ideas developed during my time at Mobile Systems International, conceived and developed a more powerful application generator that takes UML-like formal specifications as input and generates software conforming to the

specification as output, building upon a core run-time framework that implements the architecture. The tool is implemented in Common Lisp and makes use of Lisp's meta-object protocol (MOP).

- Using the application generator as part of my consultancy work.

For Mobinets, a supplier of software tools and consultancy to mobile telephone network operators, developed a network planning application that:

- supports the designing of the fixed part of GSM, GPRS, EDGE and UMTS mobile telephone networks
- has a rich data model and produces many outputs relating to the dimensioning and costing of networks
- has a graphical user interface that displays nodes and links both geographically using a geodata backdrop and topologically emphasising logical relationships
- interfaces with other software via sockets for functionality such as simulation of network traffic, optimisation of transmission and storage to database.

Responsible for all aspects of development from requirements capture through to testing, and liaising with the customer at all levels.

Specific responsibilities:

- Requirements capture and analysis
- Functional specification (object models, events and "domain algorithms")
- User interface design
- Design and programming
- Testing
- Integration with software modules produced by the customer
- Release management
- Support, bug fixing and small enhancements

Also implemented supporting tools for automatic building and automatic regression testing.

1999 – 2000 Career Break

Fulfilled a long-term ambition to travel around the world.

1993 – 1999 Mobile Systems International

Team leader, technical architect and senior developer on projects that produced tools to support the designing of the fixed part of GSM mobile telephone networks.

Initially a lead developer on a project that developed a network planning tool in C++.

Later, conceived and developed an application generator and used it to generate a network planning tool. This tool was implemented in Common Lisp.

The company's network consultants used the tools to design the Base Station Subsystems of GSM networks, with much greater ease than their previous spreadsheet-based approaches.

1983 – 1993 Logica

Designed and implemented software on a large number of projects, initially with an emphasis on systems software such as compilers and garbage collectors, and later with an emphasis on object-oriented analysis, design and programming. Also spent time on expert systems and their underlying technology.

Education

University College London: BSc (Hons) in Computer Science, 2(i). 1980–1983.

A Levels: Maths (A), Further Maths (A), Physics (A).
