James Pike | Curriculum Vitae

☎ +44 208 638 8805 • ✉ jcv@chilon.net • chilon.net
ohjames

Experienced Full-stack Software Developer with strong skills in many programming languages and Linux.

Education

Imperial College London	**London**
MEng Computing, Second Class Hons	*2000–2004*
Exeter College	**Exeter**
A-Levels, Maths: A, Further Maths: A, Physics: A, Computing: A	*1998–2000*
Sidmouth College	**Sidmouth**
GSCEs, 4 A, 4 A, 1 B*	*1996-1998*

Master's Thesis

title: C♭

description: Transpiler for a teaching language C♭ based on a subset of C# with a plugin system to allow compile-time modifications. The parser was constructed using ANTLR.

Publications

title: The Datacatcher: Batch Deployment and documentation of 130 location-aware, mobile devices that put sociopolitically-relevant big data in people's hands: Polyphonic interpretation at scale.

venue: In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pages 1597–1607, 2016.

authors: W. Gaver, A. Boucher, N. Jarvis, D. W. T. Cameron, M. Hauenstein, S. Pennington, J. Bowers, J. Pike, R. Beitra, and L. Ovalle.

Computer Skills

- Full-stack Web Development
  - JavaScript (16 years), EcmaScript 6/7/beyond (5 years), TypeScript (3 years), flow.
  - Frameworks/Libraries: React (+ Redux, 3 years), redux-observable, redux-saga, Angular 1-7.1 (5 years), ngrx (1 year), Vue, & D3.
  - Node.js (7 years)
  - HTML5: WebSocket (+ Socket.IO), Canvas, WebGL (+ GLSL), Web Workers, IndexedDB, History, Fetch API, Web Components & Shadow DOM.
  - CSS: CSS3, SCSS (sass, 6 years), styled-components, material, bulma, bootstrap, compass & less.
  - Build systems: webpack, npm, gulp, grunt & broccoli.
  - Module systems: webpack, systemjs, jspm, require/amd, bower & browserify.
  - Testing/TDD: jest, enzyme, karma, protractor, jasmine, mocha, chai & sinon.
  - V8 C++ API (including C++ node modules and integration of V8 into C++ projects).
- Databases: RethinkDB, MongoDB, MySQL, PostgreSQL, Redis.
- DevOps / Cloud: AWS (EC2, S3, SES, CloudFormation, ElastiCache), Kubernetes, CoreOS, Docker, docker-compose, rkt.
- C++, C++11, C

  - Boost (mpl, multi index containers, spirit etc.), Meta-programming.
  - Multi-threaded/event-based code - pthread, boost::thread, libuv, boost::asio.
- Embedded Devices: Microsoft Gadgeteer (.NET MF)/Raspberry Pi/Arduino.
- Python (3 years)
- Ruby
  - Event-based/threaded code: EventMachine, Fiber, Thread.
  - Ruby C API including gems written in C++ (using C API and FFI).
  - Servers: Rails 3+4, Sinatra, Goliath.
  - TDD/BDD: Rspec.
- Scala, D (2.0 + vibe.d), C#, zsh, bash, lua, Perl 5.

Experience

chilon.net **London**

Independent IT Professional *2014–Current*

- **kchomp (March 2018 - Current)**: react 16.4, redux-observable, TypeScript 3.1, typesafe-actions, node, kubernetes, digital ocean: Migrated project from ES2017 to TypeScript 3.1. The redux-saga side-effects were rewritten using redux-observable. Rewrote back-end using koa2/typescript with a postgresql database (used via knex/knex-migrations). Made several improvements to the UI and UX, including the implementation of a custom virtual list that supports the React 16.4 lifecycle events, scroll restoration and transparent infinite virtual scrolling. All new features were implemented using TDD and tested via enzyme and jest. I also migrated the cloud deployment from AWS/CloudFormation to Kubernetes/DigitalOcean.
- **Press Association (April - December 2018)**: Angular 4.3-7.1, TypeScript 2.5-3.1, ngrx, rxjs, redux, Angular Material 6-7.1, TDD, jasmine, karma, agile, node. Worked on a mission critical rebuild of a legacy AngularJS project. Took 20,000 lines of poorly designed "copy and paste" style code and replaced it with 4,000 lines of well tested DRY code using the ngrx framework (an Angular rxjs based implementation of redux and redux-observable). Incrementally upgraded the project from Angular 4.3 to 7.1 and from TypeScript 2.5 to 3.1. Used new features provided in the newer release of TypeScript to improve type-safety throughout the project. Implemented several significant new features from scratch during and after the refactor. Wrote a docker-compose based containerised development environment to allow developing the project on a variety of systems and to simplify the CI processes. Subsequently lead the redevelopment of a second legacy AngularJS app, rewriting the project from scratch using Angular 7.1 and a TDD approach.
- **Dectech (November 2017 - March 2018)**: Angular 5, TypeScript, rxjs, python, highcharts, docker-compose, agile: Lead developer on a data analytics and visualisation portal for data relating to UK commercial law. Wrote the front-end using immutable data and rxjs, with OnPush change detection enabled for all components. Wrote python tools to transform, manipulate and validate the data. Implemented a container based deployment strategy using docker/docker-compose and set up CI using CircleCI 2. Helped arrange sprints and estimate work using Jira agile tools.
- **Session Master (October 2017 - January 2018)**: Android, Kotlin, Android Architecture Components, ConstraintLayout, Dagger2: Coded/designed an android app to help flexible shift workers track their time.
- **Bulb Energy (March - August 2017)**: node, react/redux, TypeScript, ES6/7, flow, apollo-client, GraphQL, styled-components, webpack, jest, CircleCI 1/2: Took over a key react/redux/flow application (customer dashboard) from a junior developer and updated it to follow modern React best practices. Wrote a nodejs GraphQL server in TypeScript to sit in front of all existing APIs. Refactored the existing react/redux/flow dashboard website to use the graphQL endpoints via apollo-client. Simultaneously developed new features and fixed bugs in the existing code. Tested all functionality with e2e tests and jest/mocha/chai unit tests.
- **Collaborative Playlist (Feb, April, September-October 2017)**: Angular 2/4, TypeScript, rxjs, nodejs, koa 2, websockets: Full-stack development on a web app that lets users collaboratively edit a music playlist together in realtime and listen to it in sync. All components using observables and the OnPush change detection strategy.
- **Dangler (May 2017)**: Angular 2/4, TypeScript, rxjs, immutablejs, websockets: A stock ticker for crypto-currencies. All components use observables and the OnPush change detection strategy.
- **Clickon Media (Jan - March 2017)**: nodejs, ES6, mariadb, docker: Wrote an analytics API to calculate historical "trending" data from view counts for all articles.
- **kchomp (July 2016 - Jan 2017)**: nodejs, react/redux, redux-saga, enzyme, webpack, IndexedDB, web workers, web animations: Reimplemented an Aurelia based front-end. Many improvements were added including a local IndexedDB cache. A react library was written for producing hardware hardware optimised

animations using the FLIP technique and the Web Animation API. Implemented scripts for deployment, mainly using AWS/CloudFormation, automatically deployed via a CircleCI based CI/CD process.

- **Ticket Text (Feb 2015 - Dec 2016)**: nodejs, mongo, express, ES6/7, docker, CoreOS, angular 1.x, webpack, sass, bootstrap, bluebird, protractor, karma, mocha, redis, AWS, cloudformation, ElastiCache, CircleCI: Full-stack development as a technical lead on a ticket website. At times working alone and at other times leading a small technical team. The back-end was written as an SOA using a TDD approach and with end to end testing. Services communicated via Redis pubsub provided by ElastiCache. The front-end code was structured using a component based architecture. I was responsible for designing and implementing all DevOps processes, these were entirely automated using AWS, docker, docker hub, CoreOS, etcd2 and CloudFormation. CICD was used to automate testing and building of the code. The site has been used for several ticket sales where thousands of tickets were sold in a few minutes.
- **kchomp (Jan 2015 - Apr 2016)**: ES6/7, nodejs, rethinkdb, express, aurelia, scss, gulp, jspm, SystemJS, CircleCI, CoreOS, docker: Full-stack development on an all in one content aggregator web application. The back-end was written in ES6/7 and the front-end was written in TypeScript using the Aurelia framework. Performed all DevOps, deploying to a CoreOS cluster running on Digital Ocean and load balanced via haproxy (configured via confd + etcd2). CICD was used to automate testing and deployment.
- **We are Everywhere (Dec 2014 - Feb 2015)**: ember-cli, ember-data, nodejs, sails, mongodb: Full-stack development on a fitness/nutrition SPA. Built and designed the code for the nutrition, dashboard and payment front-end components working with a designer. Worked on a fitness data visualisation using D3. Built the node.js back-end component to interface with a payment processor (UPG/SHP) using TDD and helped improve and bugfix existing back-end and front-end components.
- **Goldsmiths, University of London (Aug - Dec 2014)**: nodejs, mongodb, mocha, chai, redis, angular, sass, bootstrap, google maps, C#, embedded development, MS Gadgeteer, protractor, aws, ec2, s3, SoA: Full-stack development and design for a "big-data" interactive research project including a server component built with a service oriented architecture and a front-end component. I worked in a team containing one other senior developer. The backend component was developed using a TDD approach. End-to-end tests were written using Protractor. Moved the system from a local server to an AWS cloud using CloudFormation. I also designed the code architecture for an embedded hardware client written in C# using the Microsoft Gadgeteer (.NET MF) platform and developed the code. This included writing a driver for the SIM900 GSM/GPRS modem.

TS-Associates — London

Senior Software Engineer — *2008–2014*

Worked on latency sensitive server side projects and web front-ends in various languages and at all stages in the software development lifecycle.

- Node.js/MongoDB/Bootstrap (sass)/Angular/Grunt/Bower/RequireJS/EcmaScript 6 (traceur): Lead a two man team for the redesign and reimplementation of a legacy PHP UI. It was rewritten as a web application using modern HTML5 components structured with AngularJS in object-oriented JavaScript style using traceur and tested using Protractor.
- JavaScript/C++/V8: Designed and developed a system to allow customers to write message decoder plugins in JavaScript for the TipOff latency monitoring appliance. This included writting an open source library to simplify binding between C++ and V8. Extra functionality was added in JavaScript to make the system take advantage of a dynamically typed environment.
- C++11/RFA/Lua/JavaScript: Designed, implemented and tested a high performance (800,000 messages per second) interactive real-time multi-threaded C++ server (TREP Feedhandler) for relaying large volumes of stock market and news data using RWF.
  - Boost: Use of asio, mpl, multi_index_container, regexp and others.
  - C++: Hand-optimised latency critical code using template meta-programming techniques and boost::mpl.
  - valgrind/kcachegrind: Profiling the software to find and fix bottlenecks in performance.
  - C++11: Upgraded project from C++03 to C++11 in order to simplify code and improve performance.
  - Lua: Designed and developed a programmable configuration system based on lua with bindings to C++ to allow a great deal of flexibility in the configuration of the product.
  - JavaScript/Node.js/TDD: Designed an extensive event-based regression system using Node.js involving a mock server that can be customised with JavaScript code.
- C++/boost::asio/boost::python/Python: Participated in the standardisation process of an open source inter-party latency monitoring transport system and wrote a reference implemention in C++ (boost::asio) with python bindings (boost::python).
- C++/PHP/MySQL/Infobright: Developed high-performance MySQL middleware and server components including a low-memory bulk insertion system and an RRD system similar to RRD tool both for MySQL and Infobright including PHP and JavaScript (V8) bindings.

- Scala: A daemon for polling data via XML-RPC and storing it in RRDs.
- C++/Ruby/TDD: Developed message decoders for the TipOff latency monitoring system and developed regression tests for all code in ruby.

Maven Labs **Singapore**

Software Engineer and Systems Administrator *2007–2008*

I worked on "web 2.0" websites, administered a small network of servers and ran dev ops.

- Ruby on Rails/Jboss Seam/PHP: Developing high profile interactive websites from scratch alone and in small teams.
- JavaScript/JQuery/Underscore/Backbone/HTML/Haml/CSS/Sass - Developed front-end code for high profile websites alone and in small teams.
- SysAdmin/Shell/Zsh: Administered a small network of 15 machines running RedHat, CentOS and Ubuntu. Installed hardware in a data center in addition to maintaining the network and firewall. Maintained scripts for automating common sysadmin/devops tasks.

Yahoo! **London**

Software Engineer *2007*

I was part of a team that provides location services.

- C++/STL/Boost: Designed/implemention of high performance multi-threaded webserver including support for Comet (precursor to WebSocket).
- JavaScript/HTML/CSS/Comet/AJAX: Website for submitting corrections to a Yahoo's semantic analyser.

Amadeus **London**

Software Engineer *2004–2007*

C++ development on a large multi-tier, multi-threaded real time check in system containing over one million lines of code that is used by millions of people daily and running in thousands of airports worldwide.

- C++/STL/boost/meta-programming: Design/implementation of middleware and customer check-in server.
- Python/DevOps: Designed and implemented tools for source code and infrastructure management.

Others **London**

MIK Consultancy (C++/C/Qt) 2003, Astraguard (C++/Qt) 2001 *2001-2003*