

# Paul Rule

## Senior Software Developer

Sydney, Australia (open to remote or hybrid) · [linkedin.com/in/paulrule](https://www.linkedin.com/in/paulrule) · [prule.github.io/pauls-blog](https://prule.github.io/pauls-blog)  
References · [linkedin.com/in/paulrule/details/recommendations](https://www.linkedin.com/in/paulrule/details/recommendations)

## Summary

Senior Software Developer with 20+ years of experience building robust, maintainable systems on the JVM – primarily Java with Spring Boot, and increasingly Kotlin. Brings a strong emphasis on clean architecture, recognisable patterns, and thoughtful code design: writing software that communicates its intent clearly, encourages reuse, and stays easy to maintain as teams and requirements evolve.

Collaborative by nature, with a track record of working alongside engineers and stakeholders to shape solutions together – not just delivering code, but bringing people on the journey, building shared understanding, and raising the capability of the team in the process.

Looking for a senior or lead engineering role in a friendly, energetic team that takes pride in what it ships. Open to fully remote or hybrid positions based in Sydney, Australia.

## Projects & Writing

**Blog** – [prule.github.io/pauls-blog](https://prule.github.io/pauls-blog) Technical writing covering Kotlin/Java/Programming, software design, and development practice.

**GitHub** – [github.com/prule](https://github.com/prule)

- **Laptime Insights (WIP)** – A personal project building a dashboard for Assetto Corsa Competizione lap time data. Practising clean architecture with Kotlin, Ktor, Exposed, and WebSockets, integrated with sim racing broadcast telemetry.
- **Data Processing Experiment** – A documented coding exercise following the incremental design and implementation of a configurable data processing pipeline. Equal parts design challenge and architecture writeup.
- **REST/HATEOAS Example** – A reference implementation demonstrating a REST API with HATEOAS-driven navigation, paired with a Spring Boot + React template.
- **JSONata-Java Fork** – Forked the JSONata Java library to replace floating-point arithmetic with **BigDecimal**, solving binary precision issues in expression evaluation.

## Skills

**Languages & Frameworks** Kotlin, Java, Spring Boot, Ktor, Exposed, React, TypeScript, Node.js

**Data** PostgreSQL, Apache Spark, JPA, Hibernate, SpiceDB

**Architecture & Design** Clean architecture, domain modelling, REST/HATEOAS, ReBAC, CRDT

**Testing** TDD, BDD, JBehave, JUnit

**Tooling & Platforms** Git, Gradle, Maven, IntelliJ IDEA, Renovate

**Practices** Agile/Scrum, technical mentoring, code review, pair programming

# Education & Certifications

**Bachelor of Technology, Information Engineering** – Massey University (1990–1993)

**Sun Certified Java Programmer** – Sun Microsystems (2001)

**Certified ScrumMaster** – Scrum With Style (2011)

# Experience

## Senior Software Developer – OneTeam Services

2024–2026

OneTeam Services was building a SaaS platform for business process automation, enabling organisations to define and run complex workflows through a no-code workspace model.

Joined early in the platform's development to help design and build the backend from the ground up. Core contributions included:

- **Flow engine** – Designed and built the Kotlin-based engine responsible for parsing, validating, and executing multi-step automation sequences, with JSONata powering dynamic expression evaluation within flow steps.
- **Core data model** – Worked on the PostgreSQL schema underpinning dynamic workspaces, form definitions, and configurable execution pipelines – a non-trivial modelling challenge given the flexibility the product required.
- **REST API** – Designed and implemented a Kotlin/Ktor/Exposed API consumed by the React frontend, supporting real-time workflow configuration and form submission.
- **Collaborative editing** – Integrated Automerge CRDT to support real-time, conflict-free collaborative editing of workspace documents.
- **Permissions** – Implemented a SpiceDB relationship-based access control (ReBAC) system with a local index, integrated with PostgreSQL row-level security.
- **Architecture uplift** – Introduced clean architecture, object-oriented patterns, and HATEOAS conventions, improving testability, maintainability, and consistency across the codebase.
- **Developer tooling** – Built and iteratively improved a debug/test harness to support faster development and validation of flow execution behaviour.

**Stack: Kotlin, Ktor, Exposed, PostgreSQL, SpiceDB, Automerge, TypeScript, React**

# Senior Software Developer – Propection

2019–2024

Propection is a health data analytics platform providing insights to pharmaceutical and healthcare organisations. Over five years, contributed across data infrastructure, pipeline architecture, and product capability – consistently pushing toward more configurable, maintainable, and scalable solutions. Led a small team of three developers for a significant portion of this time.

- **Generative AI assistant** – Part of the team that built a natural language interface to the Propection platform, delivering a production Gen AI capability powered by ChatGPT.
- **Common data format & task pipeline** – Introduced a common data format and designed the task pipeline to process it, replacing bespoke per-client development with a generic, configurable approach. Added dependency-aware concurrent task execution to maximise throughput and minimise processing time.
- **Configurable data transforms** – Prototyped a transformation framework allowing data to be mapped to a standard model using a combination of generic and custom logic – reducing reliance on code deployments for routine changes and improving standardisation across data sources. Separately implemented data transformations using Apache Spark supporting multiple input schemas and formats.
- **Zero-downtime data refresh** – Implemented background data loading using A/B schema switching, ensuring seamless data updates without system downtime.
- **Simplified data loading prototype** – Explored using natural business keys to reduce processing complexity, resulting in faster loads and significantly easier debugging.
- **Pipeline visibility** – Built a React frontend for controlling, monitoring, and observing the task pipeline, giving the team a clean operational interface.

**Stack: Java, Kotlin, Spring Boot, Apache Spark, React, PostgreSQL, OpenAI API**

---

# Senior Software Developer – Mobiusly

2014–2018

Mobiusly is a boutique software consultancy delivering web, iOS, and Android products for clients across financial services, education, health, retail, and telecommunications. Worked as a senior full-stack developer across multiple client engagements, taking responsibility for solution architecture, design, development, and testing within Agile/Scrum teams.

Beyond client delivery, contributed to the consultancy's own capability by building internal tools, reusable services, and project templates – reducing ramp-up time on new engagements and improving consistency of delivery across the team.

**Stack: Java 8, Spring Boot, React, JavaScript, HTML5, CSS, PostgreSQL, Heroku, AWS, REST, HATEOAS, Microservices, Swift, iOS, Android, Node.js, Ember, Gradle, Git**

# Software Developer – Reserve Bank of Australia

2014

Short-term engagement as part of a Scrum team developing file processing services for batch data loading. Contributed to solution architecture, design, development, and testing, and participated actively in sprint ceremonies – including demos and stakeholder feedback sessions.

**Stack:** Java, Spring, JPA, JBehave, JUnit, TDD, Maven, Tomcat, Git

---

# Software Developer (Contract) – Department of Premier and Cabinet, NSW Government

Mar 2013–Feb 2014

Contracted to design, build, and test an extranet application for collecting progress reports from agencies reporting to cabinet. Beyond the application itself, established the team's entire software development capability from the ground up – setting up version control, build tooling, repository management, a development server with LDAP, email and database, and a full automated testing stack including BDD and Selenium-based acceptance tests.

**Stack:** Java 7, Spring, Spring Security, Spring Data, JSF2, PrimeFaces, JPA2, MSSQL Server, JBehave, Selenium WebDriver, Thucydides, JUnit, Maven, Nexus, Tomcat, Git, JIRA

---

# Software Engineer – VMware / SpringSource

Aug 2012–Feb 2013

Part of the SpringSource R&D team in Sydney, working on Elastic Runtime – a next-generation, fault-tolerant, automatically elastic Platform as a Service built on Linux. A complex distributed system involving a scalable column store, request routing, horizontal scaling, and sophisticated security mechanisms.

Contributed to the design and implementation of key platform functionality, including a Netty-based networking layer for inter-component request routing. Also introduced and implemented JBehave acceptance testing for the network component, using REST mocks to verify behaviour in isolation. Stood in as ScrumMaster during the team lead's absence, facilitating release planning, sprint planning, reviews, and retrospectives.

**Role ended when the Australian office closed following acquisition.**

**Stack:** Java, Spring, Netty, REST, LXC, Linux, JBehave, JUnit, TDD, Git

---

For earlier career history, see [LinkedIn](#)