

James Darby

james-darby.com | +44 7503 469 322 | james.michael.darby@gmail.com
linkedin.com/in/j-darby | github.com/Darbjm | [Parallel Assembly Ltd](#)

Experience

Tech Lead & Architect

August 2026 – Present

Gecko Risk

London, UK

- **Cloud Cost Optimisation** – Cut cloud spend by **50% on my trial day**, with a roadmap to a **90% total reduction**.
- **End-to-End Data Platform** – Own the data platform from architecture to delivery: cost-efficient ETLs feeding client-facing dashboards and data-sharing APIs.
- **AI-Driven Insights** – Building AI features that turn our data into actionable insights delivered directly to customers.
- **Infrastructure & Security** – Architect cloud infrastructure and security across the platform.

Full-Stack Engineer

September 2022 – August 2026

ComplyAdvantage

London, UK

- **AI Alert-Review Agent** – On the agentic team, built an agent adjudicating screening alerts: **2.5 years of human analyst work completed in 12.5 hours**. On a labelled dataset it wrongly dismissed only 5 of 1,000 true positives — **99.5% against a 99% safety bar, beating human error rates** — with a second verification agent as backstop.
- **Multi-Tenant Data Lakehouse** – Core engineer on a multi-cloud (AWS + GCP) Databricks Delta Lakehouse powering live client dashboards and exports; built Kotlin Spark Structured Streaming ingestion of Kafka Protobuf events with per-tenant KMS encryption, run in cost-saving batch mode, **reducing ETL time from 12 hours to 20 minutes**.
- **Config-Driven ETL & Data Quality** – Co-built a PySpark ETL framework with declarative column maps, data-quality checks, dedup, and latest-state/history merge strategies — new pipelines shipped as YAML config, not code. Added CI compatibility checks diffing view changes against production data to catch breaking changes and perf regressions before release.
- **Multi-Tenant BI** – Embedded Superset with dashboards-as-code and row-level-security dynamic views: **one dashboard and one table serve every tenant**, replacing per-client duplication and simplifying migrations.
- **Unified Front-End Interface** – Built a React-based UI with Cypress-Gherkin E2E tests, localisation, component libraries, and Vite/NX package splitting; **cut CI/CD runtime from 60 minutes to 15**.
- **Microservices & Operations** – Developed Kotlin-Spring microservices integrating Yugabyte, Databricks, and Superset with **load times under 15 seconds**; canary and blue-green deployments via ArgoCD, observability with Datadog, Sentry, and Grafana in a SOC 2-compliant environment.

Full-Stack Engineer

September 2021 – September 2022

E.ON Next

London, UK

- **Full-Stack Architecture** – Designed a complete overhaul from **DevOps to Front-End**, optimising costs and improving efficiency using Terraform, React TypeScript, Django, Python, and AWS.
- **Large Migration** – Managed the migration of **6 million customers** to a new system.
- **Team Leadership** – Planned for prospective hires, trained new team members, and led teams while conducting technical design analysis and architectural planning.

Education

Central Saint Martins

Bachelor of Arts in Product Design

London, UK

2014 – 2017

General Assembly

Software Engineering Immersive Course

London, UK

2019 – 2020

Projects

Mesh UI | *TypeScript, React, Vite, NX, Cypress, Docker*

February 2023 – Present

- Part of the main Front-End team managing architecture, structure, environment, and patterns for other Front-End feature teams.
- Introduced AI tools such as Langfuse, evals, and sub-agents.
- Collaborated with Product and Design teams to deliver Front-End features.

Datalake | *Kotlin, Python, Spark, Spring, Databricks, Superset*

February 2023 – January 2025

- Built and maintained Spark ETLs, from Kafka ingestion through to presentation-layer views.
- Created and managed versioned SQL migrations with apply-time integrity checks.
- Automated cost monitoring: SQL alerts on slow queries and weekly spend-increase alerts.
- Integrated Databricks with Kotlin back-end services and Superset for BI.

Technical Skills

Languages: TypeScript, Python, Kotlin, SQL (Postgres), Terraform, HTML/CSS

Frameworks: React, Django, Spring, Spark, Pytest

Developer Tools: Git, Docker, GCP, AWS, VS Code, DataBricks, Superset

Libraries: Zustand, React Query, RTL, Styled-Components, MSW, Pandas, NumPy