Adam Byrne

166 Drominbeg, Rhebogue, Co. Limerick V92 W0X4
bxrne.com | github.com/bxrne | adamrbyrne@gmail.com | +353 (0)87 384 3771

Profile

A software engineer with a passion for learning new things. I have worked on Manufacturing Systems applications and services, Radar ML for vehicles, 3D teleoperation for UR robots, AI analysis for investment funds, Agentic Tooling Development for regulated environments and Simulation, Manufacturing and Design of High Powered Rockets and accompanying Edge AI Payloads.

Currently working on a time-centric database for high throughput loads (Written in Zig).

Experience

Residency 5 - Dissertation (*Johnson & Johnson, Limerick*) *12/01/2026 - 19/05/2026*

- Developed an Agentic Orchestrator as a scalable alternative to Low-Code/No-Code development tools
- Implemented ISA95 ISO standard enforcement and Product Quality ISO 25010 compliance via plugin architecture
- Orchestrated headless agents via @anomalyco/opencode SDK, forming autonomous agent teams for end-to-end development automation
- Transformed requirements into automated development workflows, eliminating manual low-code platform dependencies
- Validated framework effectiveness through comprehensive testing against industrial MES requirements

Graduate Development Engineer (*Johnson & Johnson, Limerick*) *07/2024 - 12/2024, 05/2025 - 07/2025*

- Following on from my residency I collaborated on a Local Machine Integration to automate the operation of Zeiss Calypso CMMs
- Matured my MES integration adding observability (logs, metrics and alerts), improved generalisation via hot-swappable configmaps
- Created a RESTCanary to performance test OpenAPIv3 APIs by hitting them with spec-based payloads to imitate real usage
- Integrated cross cluster logging via oauth2 proxies to integrate with existing Prometheus stack
- Developed metrics scraping for a Low-Code tool's Postgres Database as its monitoring was limited
- Setup GxP backup pipeline for Storage Account containers
- An integration between two seperate teams to integrate IBM Maximo Gauge data into EVA project.

Residency 4 - AI Platform (*Shannonside Capital, Limerick*) *01/2025 - 05/2025*

- Developed software infrastructure stack from a blank slate via Terraform to Azure
- Stood up the observability stack and security measures such as Code Scanning, SBOM scanning and VPN.
- Developed lightweight Python web applications to allow AI inference of financial statistics on various instruments.

Residency 3 - Pharma 4.0 (*Johnson & Johnson, Limerick*) *06/2024 - 08/2024*

- Participated in RBAC automation and monitoring for Korber PAS-X Global Deployments to sites.
- Designed Proof of Concept interface for EVA Web interface
- Developed a MES interface for a greenfield EVA project to allow Inspection data integration with factories.
- Added simulation mode to my MES API to allow developers to test it before the environment was provisioned.
- Developed Installation Qualification automated processes for MES Interface

Residency 2 - 5D Perception® (*Provizio, Shannon*) *09/2023 - 12/2023*
- Integrated SAHI inference technique with ONNX runtime for Radar data
- Built NVIDIA DGX-compatible k-fold cross-validation with parallelisation
- Developed profiling tool for model inference
- Benchmarked and evaluated multiple model architectures

Residency 1 - "Human in the loop AI" (*Analog Devices – Catalyst, Limerick*) *06/2023 - 08/2023*
- Worked on industrial collaborative robots as part of a 6-engineer team
- Benchmarked edge-compatible computer vision detection & segmentation models
- Developed teleoperation tool and simulation environment for UR6e robotic arm using Analog Devices 3D ToF LiDAR sensor and XBOX Kinect sensor.

Internship (*ToDesktop YC21 – Remote*) *07/2021 - 06/2022*
- Created a cloud storage optimisation tool to cross reference active paying customers with files to reduce unused storage with undo strategy.
- Worked on improving the configuration pages on the Desktop App

Education

University of Limerick
MSc. Immersive Software Engineering 09/2021-08/2026
A new approach to teach Software as an integrated Master's where I spent half of every year on industry placement (residencies 1-5) and other half in the lecture hall over 4 years (42wk/yr) gaining valuable experience from industry focused lecturers and a project form of assessment.
MSc Dissertation: "An Agentic Framework as a Scalable Alternative to Low-Code Development" (supervised by Mr JJ Collins, in collaboration with Johnson & Johnson)
BSc Thesis: "Launchrail: Modelling Atmospheric Jet Streams using Black-Scholes for Improved High-Powered Rocket Simulation" (supervised by Prof. Marie Travers and Mr. Eoin O' Brien)

Glenstal Abbey School (Secondary School) *09/2015-07/2021*
Achieved 625/625 points in the Leaving Certificate
Studied Applied Maths, Physics, Maths, Latin, German, English and Irish.
Member & Speaker of the Debate Club and Public Speaking classes
Prefect

Skills

Programming Languages OCaml, Zig, Go, Typescript, Python, Bash, C, C++, Lua, Javascript

Development Tools	Kubernetes, Helm, Git, GitHub Actions, Linux, Make, CMake, JFrog, Neovim, Jenkins & ArgoCD

Selected Projects

`macha` - *Payload Lead for Mach25 (2nd Place)* *06/2025*

- Goal: Deployable CanSat that runs AI on the edge from our Rocket
- Developed a Python multi threaded app orchestrated by systemd persisting to SQLLite
- Collected GPS, IMU and Camera data
- Ran Segmentation model to detect safe/unsafe/unknown landing zones using tensorflow and TPU
- Communicated telemetry to ground station
- Hardware: Raspberry Pi 4B with Ubuntu
- Reduced CPU Average temp from 80C to 50C since last competition
- Rocket flew to above 2km in a nominal flight, CanSat deployed at 500m and imaged til landing.

`logmgr` - *Minimal Config high performance logger* *06/2025*

- Developed a Golang logger that favours convention over configuration
- Handles multiple sinks seamlessly
- Utilises a non-blocking ring buffer
- Structured field based logging
- Used an object pool to reduce garbage collection overhead
- Automated CI/CD with SemVer

`Competiton Rocket Simulations` - *Simulation Lead for EuRoC24, IRW24 and Mach24* *06/2024*

- Managed competition requirements and simulating Rocket flight using OpenRocket and developing RocketPy simulations.
- Also worked on propulsion for the rocket
- Developing Monte Carlo simulations for variability in conditions within flight envelope
- Used to measure mission efficacy in Post Flight Review
- Designed environment so a team of many could contribute with solid dependency management

`ass` - *Runtime Assertions for Go* *09/2025*

- Go library for runtime invariants and assertions inspired by Tiger Style philosophy
- Provides fluent API for defining named, reusable invariants with automatic and manual checking
- Features auto-wrapping for state changes and batch validation via invariant suites
- Implements fail-fast approach to prevent corrupted state propagation in distributed systems
- Optimized for safety-critical code and stateful systems where debugging is complex

`lumen` - *Ray Tracer in OCaml* *11/2025*

- Implemented a complete ray tracer following “Ray Tracing in One Weekend” tutorial in OCaml
- Features PPM image output with proper color representation and vector mathematics
- Utilizes OCaml’s functional programming paradigm for geometric computations
- Implements sphere-ray intersection with proper surface normal calculations
- Built with Dune build system and comprehensive unit tests using OUnit2

camlq - *jq-inspired JSON Parser in OCaml* *10/2025*

- Pure OCaml JSON parser and query tool built from scratch without external dependencies
- Features jq-like query syntax with property access, array operations, and chaining
- Implements recursive descent parser with tail recursion for stack safety
- Uses OCaml's variant types and pattern matching for type-safe JSON AST handling
- Provides CLI interface for JSON inspection in Unix pipelines

BUCOL - *Parser & Lexical Analyser* *04/2024*

-Group project to write a parser in C++ and accompanying lexical analysis tool for a syntactically limited language (BUCOL) using flex and bison. The tool should be able to parse a file written in BUCOL and output whether the program is correctly formed w.r.t. its structure. I implemented Cuckoo hashing for the symbol table which gives constant time worst case lookups and insertions (amortized) which helps with collision resolution.