fuzzy labs

Practical AI, properly engineered.

Service Definition Document

At Fuzzy Labs, we help teams turn ambitious AI ideas into systems that run in the real world.

We focus on three core areas of delivery: Research Engineering, Prototyping, and Production.

Each one serves a different purpose. Each one comes with different challenges.

All are underpinned by MLOps.

MLOps service definition

Fuzzy Labs provides MLOps engineering for machine learning and LLM systems using open-source technologies, covering the full lifecycle from prototyping through platform engineering, deployment, monitoring, and ongoing production operation.

What the service is

Our MLOps service helps you productionise, scale, and secure AI using open source MLOps. We put the tooling and operating patterns in place that bridge the gap between “it works in a notebook” and “it runs reliably in production”.

We design and implement ML and LLM platforms, pipelines, and infrastructure with an emphasis on reproducibility, security, observability, and long-term maintainability. We have experience delivering AI in edge and other constrained environments, with a strong understanding of the associated operational and MLOps requirements.

Our work also includes applied AI research and AI security, encompassing model and tooling evaluation, and analysis of failure modes, misuse risks, and vulnerabilities in ML and LLM systems.

Our engineers come from a mix of industry, academia, and national security backgrounds, with practical experience across data science, machine learning engineering, DevOps, cloud and infrastructure engineering, data engineering, and software engineering.

Practically, that means we design and implement an MLOps platform in your environment, plus the ways of working around it: versioning, traceability, automation, and observability. The goal is a system your team can run, evolve, and audit without heroics.

What we deliver

We tailor the exact stack to your needs, but we typically cover the full lifecycle:

- **Experiment tracking and model provenance** so work is reproducible and reviewable, not tribal knowledge.
- **Data and model versioning** to reduce breakages between teams and improve traceability.
- **Automated training, testing, and deployment pipelines** to make releases boring and consistent.
- **Model deployment and serving** patterns appropriate to your risk profile and latency needs.
- **Monitoring and maintenance hooks** so degradation is visible early, not discovered months later.

- **Shared language between data science, engineering, and ops**, using practical artefacts like model cards, feature stores, and clear model provenance.

Where it helps, we use and contribute to open source accelerators and reference architectures that provision a production-ready MLOps toolchain into your cloud environment. This can include MLflow, Seldon, ZenML and Kubernetes, wired together with sensible defaults for security, deployment and observability. We are cloud agnostic, so we apply the same open source patterns wherever you run, adapting the implementation to fit Azure, AWS, GCP, or a mixed estate so you get a consistent, portable foundation for taking models from experiment to production.

How we work with you (onboarding to live)

We run delivery in three phases: **Discover, Develop, Support**.

Discover (onboarding)
We immerse ourselves in how you currently build and run models: team structure, environments, release process, pain points, and constraints. From that, we agree clear objectives, a delivery plan, and what "good" looks like for your team and your users.

Develop
We build side by side with your data scientists and engineers, iterating and testing as we go. This is where the platform is implemented, integrated into your product landscape, and shaped around your reality rather than a textbook ideal. Training and knowledge transfer happen throughout, not as a last week scramble.

Support (after go live)
Once deployed, we can stay involved for maintenance, ongoing development, or ad hoc troubleshooting. The level and duration of support is agreed per call off, based on what you actually need.

Commercially, we typically agree scope and deliverables up front, with a timeline and cost tied to those deliverables.

Offboarding and exit support

We design for a clean handover because lock in is a risk, even when everyone has good intentions. Our default is open source tooling, and we aim for you to own the solution and be able to operate it independently.

On exit, we provide:

- documentation, runbooks, and architecture notes

- access to code, configuration, and infrastructure as code repositories (where we created them)
- guidance on exporting artefacts from the tools in use (for example model registry entries and experiment logs), aligned to your internal policies

Service constraints and customisation

MLOps is not a single product you switch on. It is a system made of components, and the right choices depend on your security posture, cloud constraints, and operating model.

Constraints are typically driven by:

- your hosting environment and change control
- your preferred tooling standards
- regulatory requirements for traceability and access controls
- integration points (data platforms, CI/CD, identity, logging)

We avoid bespoke for the sake of it, but we will customise where it removes risk or friction for your team.

Service levels (performance, availability, support hours)

Because we usually deploy into your environment, the baseline availability and performance characteristics are primarily determined by the infrastructure and operational model you choose.

What we do provide is the engineering needed to define and measure meaningful service levels:

- instrumentation and monitoring for pipelines and deployed models
- alerting routes and runbooks
- deployment patterns that reduce release risk

Support hours and response targets are agreed per call off. We can design for business hours support with strong observability, or scope wider coverage if your service requires it.

Backup, restore, and disaster recovery

We do not publish a single “one size fits all” backup and DR guarantee, because it depends on where the system runs and what data it processes.

What we *do* build in as standard MLOps hygiene:

- reproducibility through versioning and provenance, so you can trace and recreate what ran in production

- infrastructure as code where appropriate, so environments can be rebuilt safely and repeatably
- clear ownership boundaries for where datasets, models, and logs live, so your existing backup and retention controls can apply cleanly

If you need explicit Recovery Point Objective/Recovery Time Objective targets and a documented BCDR plan, we capture that in Discover and design the platform accordingly.

Outage and maintenance management

If you require explicit outage and maintenance management post support, we will capture this in the Discover phase and factor it into our overall proposal.

Hosting options and locations

Our MLOps work is commonly deployed into buyer controlled cloud environments. We do have the option to host in our environments, which are cloud agnostic. We therefore can deploy to Azure, AWS or GCP, covering a range of locations geographically, including the UK for data sovereignty.

Technical requirements

Exact requirements depend on your stack, but we typically need:

- access to the target cloud environment and networking context
- a source control system for code and infrastructure as code
- CI/CD capability (or agreement on how releases happen)
- a place to run orchestration and serving workloads (often Kubernetes, but not always)

Ultimately we will capture in the Discovery phase what the exact technical requirements and considerations will need to be as part of the proposal and tailor them accordingly.

Security

Fuzzy Labs holds a Cyber Essentials Plus certificate. Cyber Essentials is a UK government-backed, industry-supported scheme designed to help organisations protect against common online threats. Furthermore we are registered on JOSCAR and maintain an up-to-date profile to support standardised supplier pre-qualification and reduce duplicated assurance requests. Finally, we also maintain a Risk Ledger profile, which we can share with clients to streamline third-party security assurance and provide evidence across policies, data protection, and technical controls where applicable.

Access to data upon exit

Because we default to open source tooling and focus on making your team self sufficient, you keep access to your data, models, and the platform components you run. We support orderly transition by providing documentation, export guidance, and handover sessions as needed.

https://www.fuzzylabs.ai/