MLOps Architecture Tools/Software

1. Model training and registry

Tool	Description	Scalability	Cost effectiveness	Flexibility	Accessibility	Integrated features
MLflow	MLflow is a versatile, expandable, open-source platform for managing workflows and artifacts across the machine learning lifecycle.	Supports distributed training and can scale with underlying infrastructure eg Kubernetes	Open-source allows for more control over expenses; depends on underlying infrastructure.	Modular design with components for tracking experiments, packaging artifacts into reproducible runs and deploying models; scalable depending on underlying infrastructure.	Platform-agnostic; supports multiple languages and frameworks.	Stage transition tags; model lineage; model file versioning; model packaging
AWS Sagemaker	Amazon SageMaker is a fully managed machine learning (ML) service. With SageMaker, data scientists and developers can quickly and confidently build, train, and deploy ML models into a production-ready	Fully managed and automatically scales to handle large datasets and complex models.	PAYG model; comes with automatic model tuning to optimise costs.	Supports multiple frameworks and offers built-in algorithms and jupyter notebooks for development.	Fully integrated with AWS, allowing for ease of utilisation of other AWS services.	No stage transition tags; limited model lineage; model file versioning; limited model packaging

	hosted environment. It provides a UI experience for running ML workflows that makes SageMaker ML tools available across multiple integrated development environments (IDEs).					

Design Decisions:

The team chose *MLflow* for model registry and training to account for the budgets of users of the pre-built pipelines, its versatility in terms of integrated features, and the fact that it does not have much use for other AWS resources beyond the S3.

2. Model deployment and serving

Tool	Description	Model dependency management	Compatibility with SKLearn and PyTorch	Flexibility
MLflow	MLflow is a versatile, expandable, open-source platform for managing workflows and artifacts across the machine learning lifecycle.	Seamless	Fully compatible with both	Flawless integration via KServe for Kubernetes
BentoML	BentoML is a model serving framework for building AI applications with Python. It can be installed as a library with pip, or through Yatai for Kubernetes. Yatai is the Kubernetes deployment operator for BentoML.	Yes, through MLflow integration	Fully compatible with both	Same as MLflow, via BentoCloud
FastAPI	FastAPI is a modern, fast (high-performance), web framework for building APIs with Python based on standard Python type hints.	Manual	Fully compatible with both	Seamless once container set up through any Kubernetes deployment platform

Design Decisions:

The team chose *MLflow* for model deployment and serving given its integrated dependency management, compatibility with Scikit-Learn and PyTorch, and flexibility in terms of integration with Kubernetes.

3. Model monitoring

Tool	Description	Hardware metrics	Model performance in production
Evidently.AI	Evidently.ai, a powerful open-source tool, simplifies ML Monitoring by providing pre-built reports and test suites to track data quality, data drift, and model performance	Yes	Yes
Grafana + Prometheus	Prometheus is an open-source systems monitoring and alerting toolkit originally built at SoundCloud. Prometheus collects and stores its metrics as time series data, i.e. metrics information is stored with the timestamp at which it was recorded, alongside optional key-value pairs called labels. Grafana is a multi-platform open source analytics and interactive visualization web application. It can produce charts, graphs, and alerts for the web when connected to supported data sources.	Limited	No

Design Decisions:

The team chose EvidentlyAI for model monitoring given its coverage of both machine-learning specific and general metric capabilities for hardware.

4. Data store and retrieval

Tool	Description	Scalability	Cost effective	Security	Accessibility	Flexibility
AWS S3	https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html	Highly scalable without human intervention	PAYG model	Built in features including IAM policies and bucket policies, either can involve RBAC, keys, encryption.	Data can be accessed from anywhere via HTTP/HTTPS, allowing for smooth data retrieval via APIs for features like real-time data retrieval. S3 can also also centralised	Agnostic to file types and other storage formats; high fault tolerance (99.9%); allows different software systems to communicate via APIs
APIs (for third-party data retrieval)	An application programming interface is a way for two or more computer programs or components to communicate with each other. It is a type of software interface, offering a service to other pieces of software.	Ability to enable scalable communication between systems	Option to reduce costs via modular system design for reuse of existing components/ code	Option to implement robust authentication (eg OAuth, API keys) and authorization mechanisms to control access to data	Supports real-time data retrieval and updates	Allows different software systems to communicate regardless of underlying technology stack (via REST, SOAP)
Data lakes	A data lake is a	Optimal for large	Often more	Option to	Centralised	Store data in raw format

	system or repository of data stored in its natural/raw format, usually object blobs or files.	amount of structured and unstructured data without predefined schemas	cost effective than DBs for storing large amounts of data	implement security measures like access controls and encryption	repository making data accessible to different teams within an organisation	across wide range of file types
Databases	A database is an organised collection of data or a type of data store based on the use of a database management system, the software that interacts with end users, applications, and the database itself to capture and analyse the data.		Cloud based - PAYG	User auth, RBAC, encryption	Optimized query engines and indexes for efficient data access; easy integration with most applications and BI tools	Supports various data models eg relational, document, key-value.

Design Decisions:

The team chose Amazon S3 as the data storage system. Digital Catapult is already using AWS for a few other projects and our technologists are comfortable with this technology.

An evaluation of feature store software the team considered using, and decided on, can be found in the Resources section of this page.

5. Data and feature engineering automation

Tool	Description	Cost effectiveness	Flexibility	Scalability
Apache Airflow	Airflow is an open source workflow orchestration tool used for orchestrating distributed applications. It works by scheduling jobs across different servers or nodes using DAGs (Directed Acyclic Graphs). A DAG is the core concept of Airflow, collecting Tasks together, organised with dependencies and relationships to say how they should run	Free	supports the creation of dynamic workflows through Directed Acyclic Graphs (DAGs), enabling users to define complex dependencies and task relationships. Also viable for retraining/A-B testing/CICD.	Ease for Helm chart setup.
Prefect	Prefect decreases negative engineering by building a DAG structure with an emphasis on enabling positive with an orchestration layer for the current data stack.	Paid	Python package that makes it easier to design, test, operate, and construct complicated data applications. It has a user-friendly API that doesn't require any configuration files or boilerplate. It allows for process orchestration and	To run Prefect, the official Helm chart requires additional configurations to be set up.

			monitoring using best industry practices.	
Dagster	Dagster is a Machine Learning, Analytics, and ETL Data Orchestrator. It handles the basic function of scheduling, effectively ordering, and monitoring computations.	Paid		

Design Decisions:

The team chose *Apache Airflow* for the automation of data and feature engineering. While alternatives like Dagster and Prefect have certain features designed to address the limitations of Airflow, the team opted to account for variation in the budgets of users of the MLOps pre-built pipelines, and Airflow is both open-source and free to use.