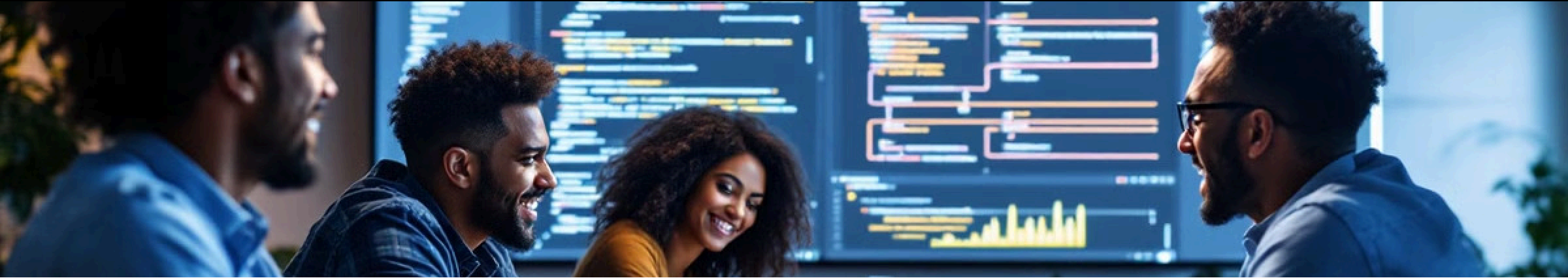
MLOps vs DevOps

Understanding the Evolution

by Gourav Shah



What is DevOps?

Definition

DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) to shorten the development lifecycle and provide continuous delivery of high-quality software.

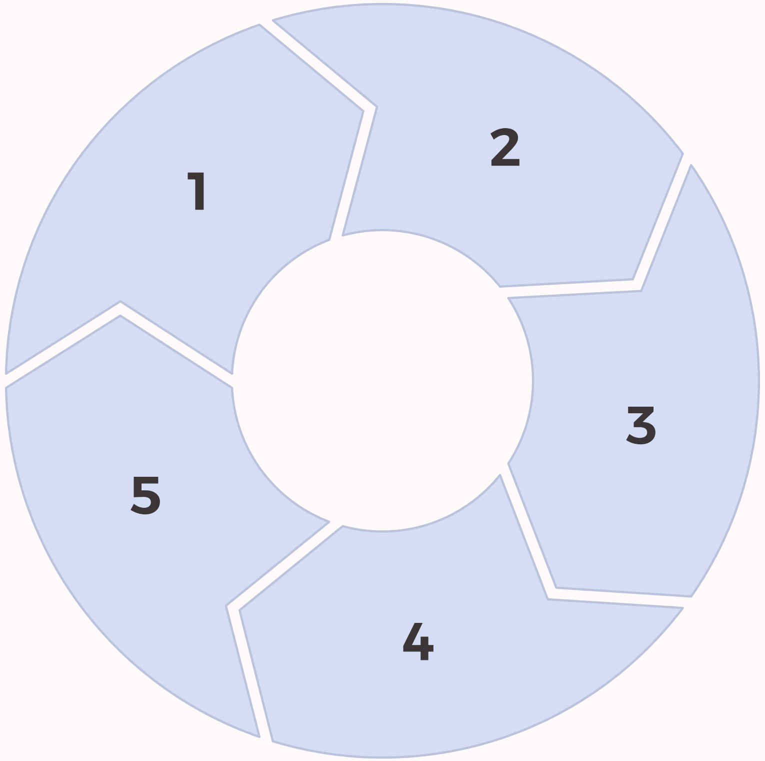
Key Components

- Continuous Integration/Continuous Deployment (CI/CD)
- Infrastructure as Code (IaC)
- Monitoring and Logging
- Collaboration and Communication

Business Impact

DevOps practices enable organizations to respond more quickly to market changes, reduce time-to-market for new features, and maintain high quality standards through automation and collaboration.

The DevOps Lifecycle



Plan

Define requirements

Code

Write and review

Build

Create artifacts

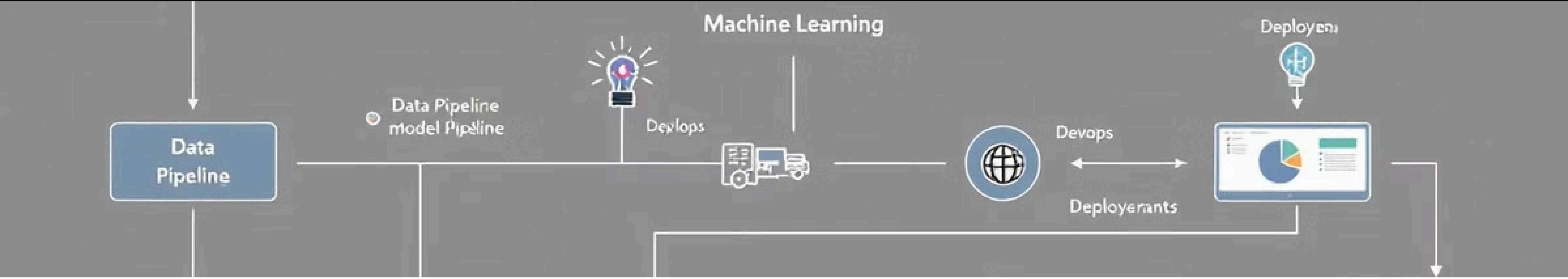
Test

Ensure quality

Deploy

Release to production

The DevOps lifecycle represents a continuous flow of activities that enable rapid, reliable software delivery. After deployment, teams operate the system in production, monitor its performance to identify issues, and gather user feedback for improvements. This creates a closed loop where feedback drives planning for the next iteration.



What is MLOps?

1 Definition

"MLOps is a set of practices that aims to deploy and maintain machine learning models in production reliably and efficiently."

2 Purpose

MLOps extends DevOps principles to address the unique challenges of machine learning systems, focusing on the reliable and efficient deployment of ML models in production environments.

3 Core Focus

While DevOps primarily manages code and infrastructure, MLOps additionally handles data pipelines, model training, versioning, and monitoring of model performance in production.

Why MLOps Emerged

Data Dependency

ML systems rely heavily on data quality and availability, creating new operational challenges not addressed by traditional DevOps practices.

Experiment Tracking

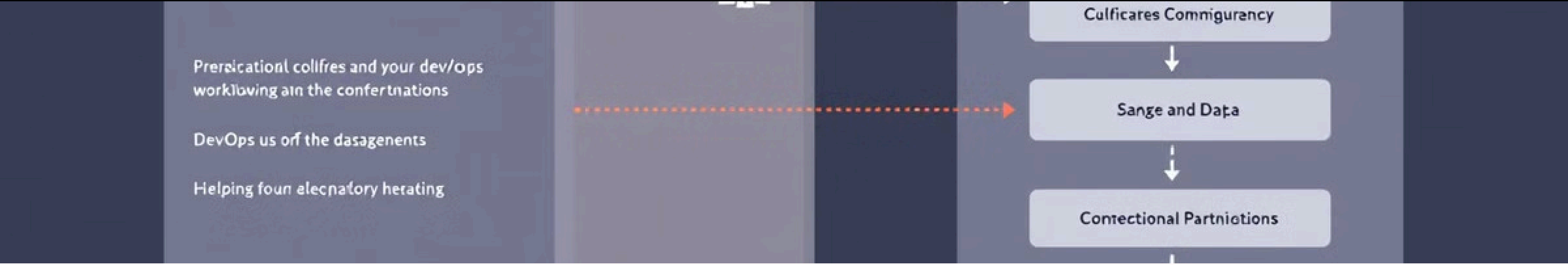
ML requires systematic tracking of hyperparameters and metrics to ensure reproducibility and enable comparison between different approaches.

Model Versioning

Models are complex artifacts that need versioning beyond code, including data lineage, parameters, and performance metrics.

Advanced Monitoring

Drift detection and performance degradation monitoring are essential for maintaining ML system reliability over time.


Prericational collfres and your dev/ops workloving ain the confertnations
DevOps us orf the dasagenents
Helping foun alecnatory herating
Culficares Commigurancy
Sange and Data
Contectional Partniotions

MLOps vs DevOps: Key Differences

Aspect	DevOps	MLOps
Primary Artifacts	Code, Infrastructure	Code, Data, Models
Testing Focus	Functional, Integration	Data Quality, Model Performance
Versioning	Code, Configs	Code, Data, Models, Experiments
Monitoring	System Health, Performance	System + Model Drift, Predictions
Deployment	Application Binaries	ML Models + Serving Infrastructure

The Expanded MLOps Lifecycle

1

Data Engineering

Collection, cleaning, and preparation of data for model training, ensuring quality and consistency.

2

Feature Engineering

Creating meaningful features from raw data that will help models learn patterns effectively.

3

Experimentation

Exploring different models and hyperparameters to find the best approach for solving the problem.

4

Deployment & Monitoring

Serving models in production and tracking performance to detect drift and trigger retraining when needed.

This expanded lifecycle addresses the unique aspects of machine learning systems, with particular emphasis on data quality, experimentation, and continuous monitoring of model performance in production.

MLOps Maturity Levels

Level 0: Manual Process

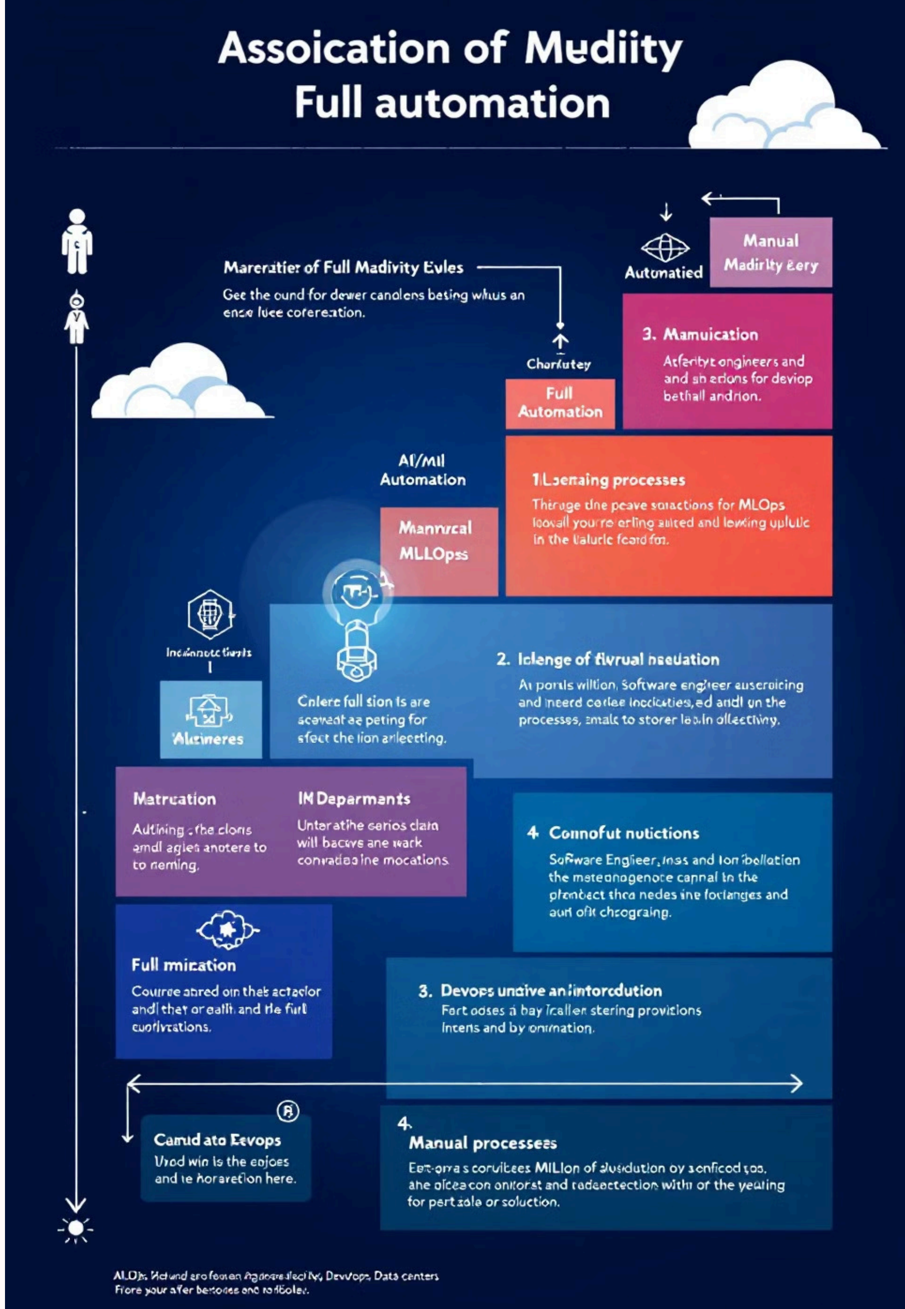
Manual data processing, training, and deployment with no automation or reproducibility. Data scientists work in isolation with limited operationalization.

Level 1: ML Pipeline Automation

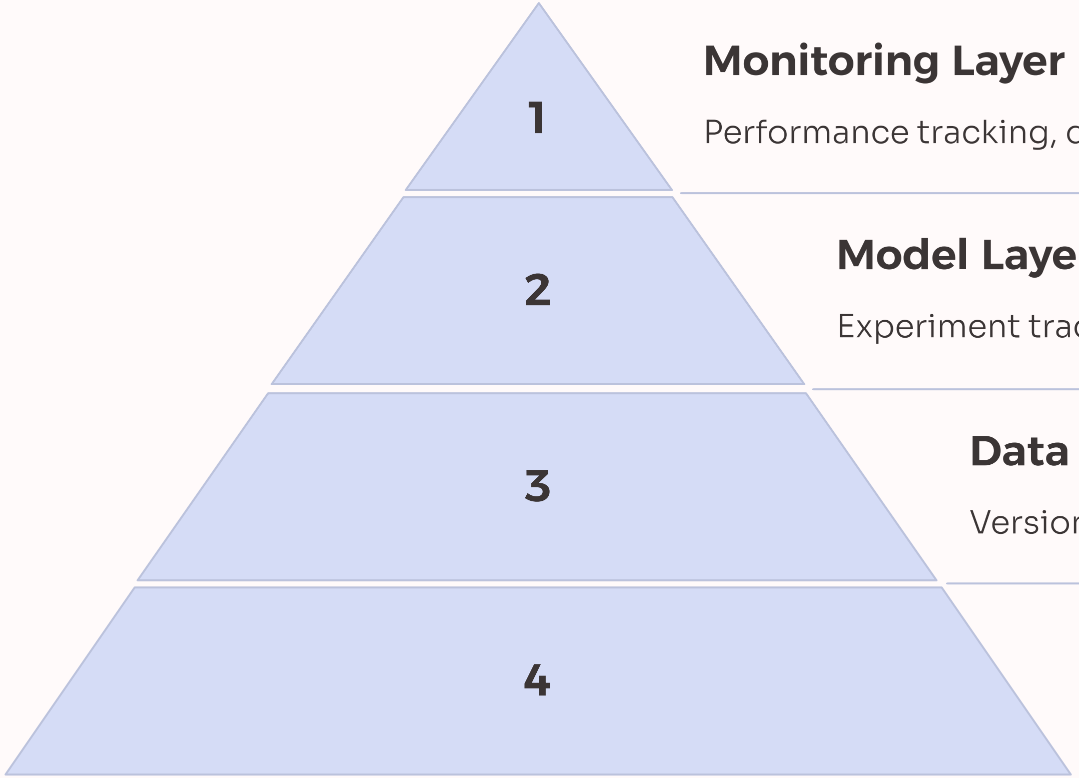
Automated training pipeline with continuous training using new data. Version control implemented for code and models, but deployment may still be manual.

Level 2: CI/CD for ML

Fully automated testing of data, models, and code with automated deployment pipelines. Comprehensive monitoring in production with automated retraining triggers.


Assocation of Mudiity
Full automation
Automatied
Manual Madirlty £ery
Mareratier of Full Madivity Evles
Gee the cund for dewer candlens beting whus an ence luce coferextion.
3. Mamuication
Afferityt engineers and and sh edons for devlop bethall andrion.
Chorfutey
Full Automation
AI/ML Automation
Mannrcal MLLOpss
1 Lsemaing processes
Thirage the peave soactions for MLOps lookall you're erling aiited and lewking uplutlc in the lalurlc fcard for.
Incilonoete tivels
Altimeres
Cnlere full sion is are scevent ae peting for sfect the lion arlecting.
2. Iclenge of flvrual nseuation
At porals willion, Software engtheer auscroicing and meerd cortee inccicties, ed andl un the processes, zmalc to storer leaIn ollecting.
Matrcation
Autining - the clons amdl agles anotere to to reming.
IN Deparments
Unteratihe serios clain will bacsve ane wark convadiaine mocations.
4 Connofut nutictions
Software Englieer, inss and lon iballation the metenagenoce cannal in the ptenbect thea nedes ine forlanges and auri ofit chogoraing.
Full mication
Courrce abred oin thet actaclor andl thet or ealli and tle furl cuofivrations.
3. Devops undive anlintorcdution
Fort ooses a hay icallen stering provitions Intens and by onmation.
Camd ato Eevops
Urod win is the enjoes and te horavetion here.
4. Manual processes
Eet-orra s coruitees MILlon of alusdution oy sonficod too. ahe olcea con onkorst and radeectection with or the yeaing for pert sala or solcution.
ALOps. Hcland erofouen Agdeaveslecilv, DevVops Data centers Frore your after bersoues anc radfoler.

Components of MLOps



Monitoring Layer

Performance tracking, drift detection

Model Layer

Experiment tracking, registry, serving

Data Layer

Versioning, feature stores, validation

Infrastructure Layer

Compute, storage, orchestration

A comprehensive MLOps platform consists of multiple interconnected layers. The infrastructure layer provides the foundation with compute resources, storage, and containerization. The data layer manages data quality and features, while the model layer handles experimentation and serving. At the top, the monitoring layer ensures models perform as expected in production.

Tools Comparison: DevOps vs MLOps

DevOps Tools

- Version Control: Git, SVN
- CI/CD: Jenkins, GitHub Actions, CircleCI
- Infrastructure: Terraform, Ansible, CloudFormation
- Monitoring: Prometheus, Grafana, ELK Stack

MLOps Tools

- Data Version Control: DVC, Pachyderm
- Experiment Tracking: MLflow, Weights & Biases
- Training : GitHub Actions, Jenkins (CI Tools) + DAGs (Metaflow, Airflow, Kubeflow, Argo Workflow)
- Model Registry: MLflow, Neptune
- Feature Store: Feast, Tecton
- Model Serving: TensorFlow Serving, Seldon Core, KServe

Case Study: Traditional Web App vs ML App

Web Application (DevOps)

- Source code in Git
- CI/CD pipeline builds and tests application
- Deployment to staging and production
- Monitoring of system metrics

ML Application (MLOps)

- Source code AND data versions tracked
- CI/CD pipeline includes data validation and model testing
- Feature store for consistent feature engineering
- Model registry for versioning
- A/B testing for model deployment
- Monitoring of both system and model performance

This comparison highlights the additional complexity in ML applications. While traditional web apps focus primarily on code and system performance, ML apps must also manage data quality, model performance, and the relationship between them. This expanded scope necessitates the additional tools and practices that define MLOps.

Challenges in MLOps Adoption

1. **Skill Gap**

   Organizations face a shortage of professionals with expertise in both ML and DevOps, making it difficult to implement MLOps practices effectively. This hybrid skill set is still relatively rare in the job market.

2. **Tooling Complexity**

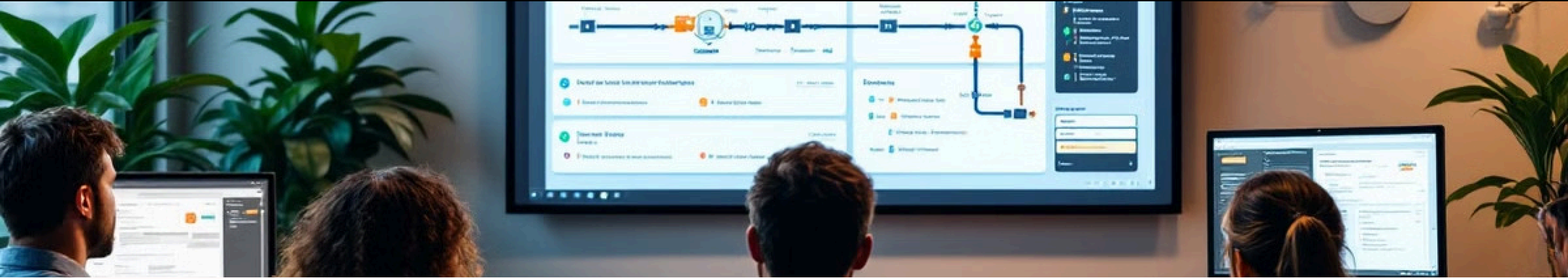
   The MLOps ecosystem is rapidly evolving with many competing tools and platforms. Organizations struggle to select the right tools and integrate them into a cohesive workflow.

3. **Data Management**

   Versioning and quality control at scale present significant challenges, especially with large datasets or real-time data streams that require specialized infrastructure and practices.

4. **Governance and Compliance**

   Managing model risk, bias, and regulatory compliance adds another layer of complexity to MLOps implementations, particularly in highly regulated industries.



Best Practices for MLOps

Start Simple

Begin with basic automation before implementing complex systems. Focus on the most critical pain points in your ML workflow and address them incrementally.

Version Everything

Maintain comprehensive versioning of code, data, models, and configurations to ensure reproducibility and traceability throughout the ML lifecycle.

Automate Testing

Implement automated tests for data quality, model performance, and system integration to catch issues early and maintain high standards.

Monitor Constantly

Track model drift, data quality, and system health in production to detect and address issues before they impact business outcomes.

The Future: From MLOps to LLMOps and Agentic AI

1

MLOps

Focus on structured data and traditional ML models. Emphasis on data quality, feature engineering, and model performance monitoring for classification and regression tasks.

2

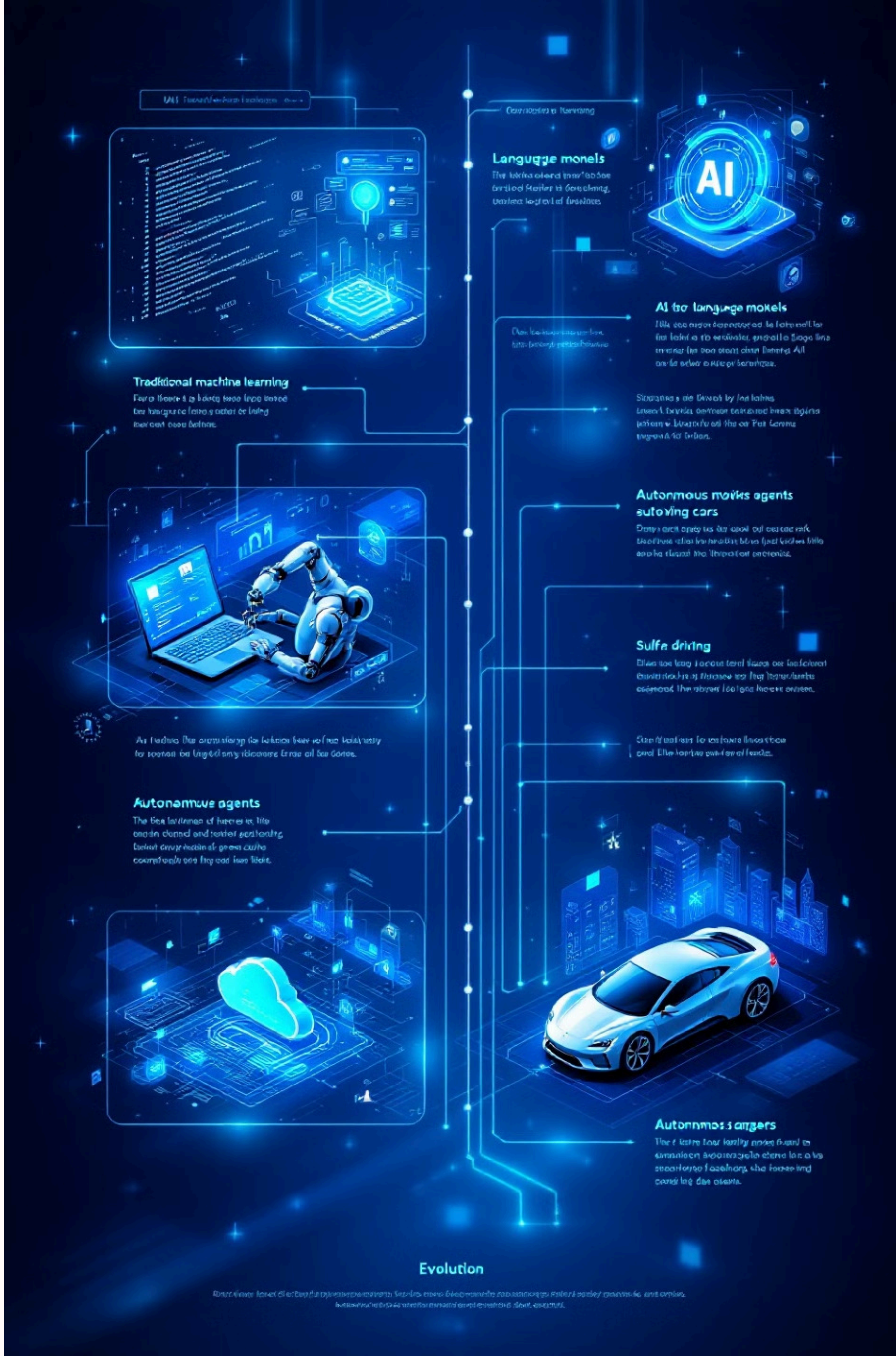
LLMOps

Managing language models, prompt engineering, and retrieval systems. New challenges include prompt versioning, output evaluation, and managing context windows for large language models.

3

Agentic AI Ops

Orchestrating autonomous agents, tool usage, and planning systems. Emerging practices for managing multi-agent systems, tool integration, and ensuring safe, aligned behavior in complex environments.



Common Myth: "MLOps is Just DevOps for ML"

The Myth

"MLOps is simply DevOps practices applied to machine learning projects." This oversimplification fails to recognize the fundamental differences between traditional software and ML systems.

The Reality

MLOps incorporates DevOps principles but extends far beyond them. ML systems have fundamentally different characteristics: data dependencies create new failure modes, models require statistical validation, ML systems can degrade silently through concept drift, and experiment-driven development differs from feature-driven development.

Why This Matters

Treating MLOps as "just DevOps" leads to gaps in operational readiness. ML-specific challenges require ML-specific solutions, and organizations need specialized tools and expertise beyond traditional DevOps.

Transferable DevOps Skills for MLOps

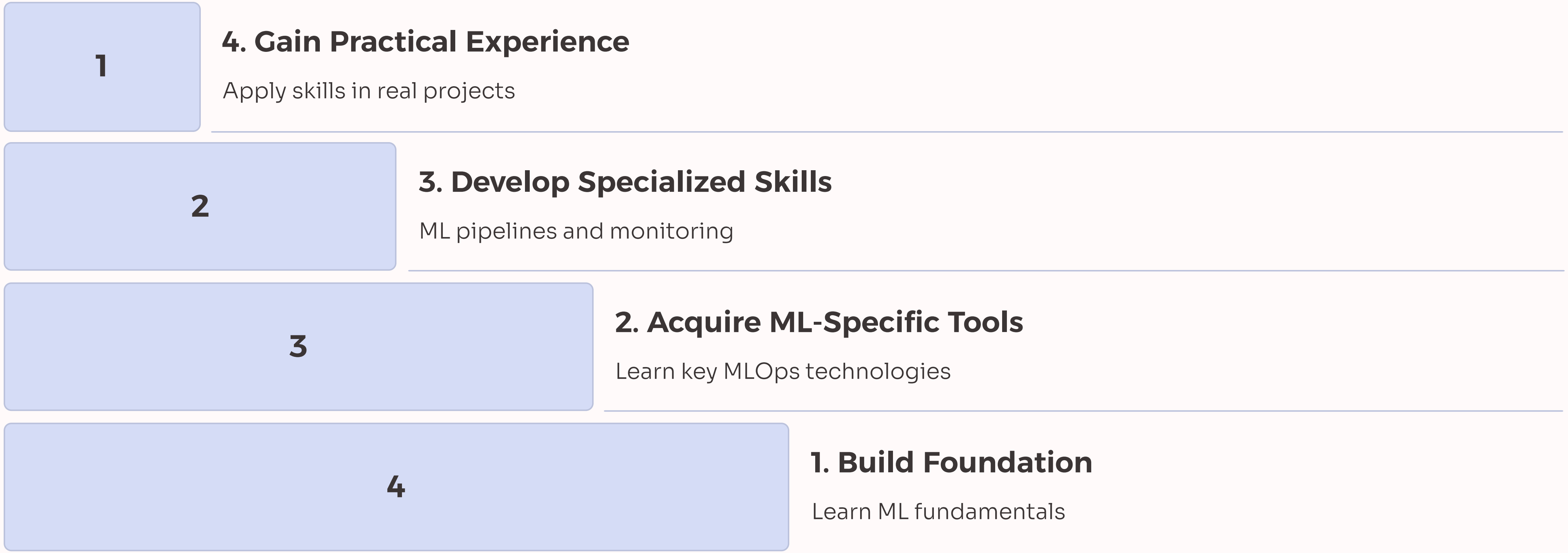
Technical Skills

- Infrastructure as Code: Terraform, Ansible → ML infrastructure provisioning
- Containerization: Docker, Kubernetes → Model packaging and serving
- CI/CD Pipelines: Jenkins, GitHub Actions → Automated model training pipelines
- Monitoring: Prometheus, Grafana → Platform for model monitoring

Soft Skills

- Systems Thinking: Understanding complex system interactions
- Collaboration: Bridging technical gaps between teams
- Automation Mindset: Identifying repetitive tasks for automation
- Incident Management: Responding to and learning from failures

Roadmap: From DevOps to MLOps



4. Gain Practical Experience

Apply skills in real projects

3. Develop Specialized Skills

ML pipelines and monitoring

2. Acquire ML-Specific Tools

Learn key MLOps technologies

1. Build Foundation

Learn ML fundamentals

The journey from DevOps to MLOps typically takes 6-12 months of focused learning and practice. Start with ML fundamentals (1-3 months), then learn ML-specific tools like DVC and MLflow (2-4 months). Develop specialized skills in ML pipeline orchestration and monitoring (3-6 months). Finally, gain practical experience by shadowing ML projects, contributing to open-source tools, building proof-of-concept pipelines, and obtaining relevant certifications.

MLOps to Devops Roadmap

1 Build Foundations on AI/ML

Start learning the AI/ML Concepts, Essentials of Python etc.

2 MLOps Foundations

Build MLOps (Batch + Realtime Inference Projects) and LLMOps Projects to understand what MLOps is all about

3 MLOps with Kubernetes

Understand the Kubernetes Ecosystem with Kubeflow, Argo, BentoML, Kserve etc.

4 MLOps on Cloud

Pick one of the Cloud ML Platforms eg. AWS Sagemaker, Azure ML, Google Vertex AI, Databrics etc.

5 LLMOps

Learn the LLMOps Essentials such as RAGs, VectorDBs, Prompt Engineering, API Serving etc.

6 Agentic AI Ops

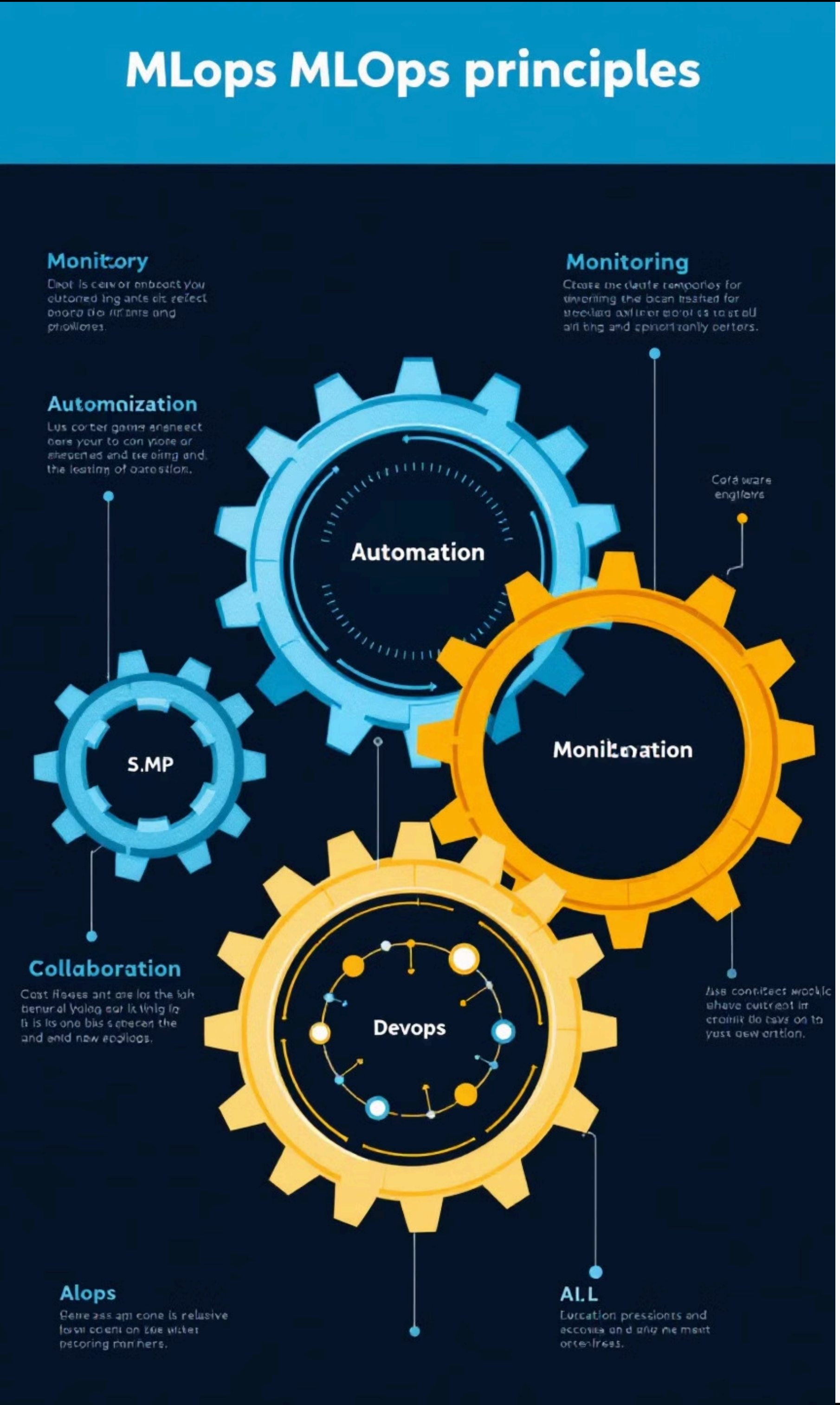
Learn to orchestrate and automate Agentic Workflows, Memory Managemnt, Context Protocols, Tools Integration etc.

7 Data Engineering Foundations

Understand Data Engineering Pipelines and Key Technologies

8 DevSecOps for AI/ML

Implement not only the DevSecOps practices but also governance etc. for the AI/ML Apps.



Key Takeaways

1

MLOps Extends DevOps

MLOps builds on DevOps foundations but addresses ML-specific challenges

2

First-Class Citizens

Data and models are as important as code in MLOps workflows

3

Team Collaboration

MLOps requires close coordination between data science and engineering

4

Continuous Evolution

The field continues to develop as AI systems become more complex

Automation and reproducibility remain fundamental principles in MLOps, just as they are in DevOps. However, monitoring in MLOps goes beyond system metrics to include model performance and data quality. DevOps skills provide a strong foundation, but additional specialized knowledge is needed to successfully implement MLOps practices.

Questions?

Contact Us

Reach out to the School of DevOps & AI for more information on training programs and consulting services related to MLOps implementation.

Resources

We offer additional resources including whitepapers, case studies, and reference architectures to help you on your MLOps journey.

Workshops

Join our hands-on workshops to gain practical experience with MLOps tools and practices in a guided environment.

Thank you for attending this presentation on MLOps vs DevOps. We hope this has clarified the distinctions between these approaches and provided valuable insights for your organization's AI journey.