Open Source Software Paradigms

Lecture - 13

Selected Topic in
Open Source Software Engineering
[Open source Artificial Intelligence]

Lecturer: Biniam Behailu
Addis Ababa Science and Technology University

Learning Objectives

By the end of this lecture, you will be able to:

- ☞ Articulate the core synergies between Open Source principles and Artificial Intelligence development.
- ☞ Diagram the core layers of the modern Open Source AI stack and identify key technologies in each.
- ☞ Differentiate between AI Software Engineering and traditional software development practices.
- ☞ Evaluate the key challenges including legal, ethical, and economic facing the Open Source AI ecosystem.
- ☞ Identify major trends shaping the future of Open Source AI and strategize a personal learning path.

Contents

Why OSS and AI are Inseparable?

- **Democratization of AI**: Lowering barriers to entry for researchers, startups, and individuals.
- **Collaborative Acceleration**: Global community effort drives innovation faster than any single company.
- **Transparency** & **Scrutiny**: "Open sourcing" models and code allows for auditing of bias, fairness, and security.
- **Avoiding Vendor Lock-in**: Freedom to build, deploy, and migrate without being tied to a proprietary platform.
- **Standardization**: Open source libraries (e.g., TensorFlow, PyTorch) have become the de facto standards.

Defining "Open Source AI"

- It's a spectrum, not a single thing. It involves openness in:

1. **Open Source Code**: The software libraries and frameworks for building/training AI (e.g., PyTorch).
2. **Open Datasets**: The data used for training (e.g., Common Crawl, LAION).
3. **Open Models**: The trained model weights and architecture are publicly available (e.g., Llama 2, Mistral).
4. **Open Tooling**: The entire MLOps pipeline is open source (e.g., MLflow, Kubeflow).

The Evolution of the Ecosystem

Early 2000s: Foundations. Sci-kit Learn, Theano, Caffe. Academic roots.

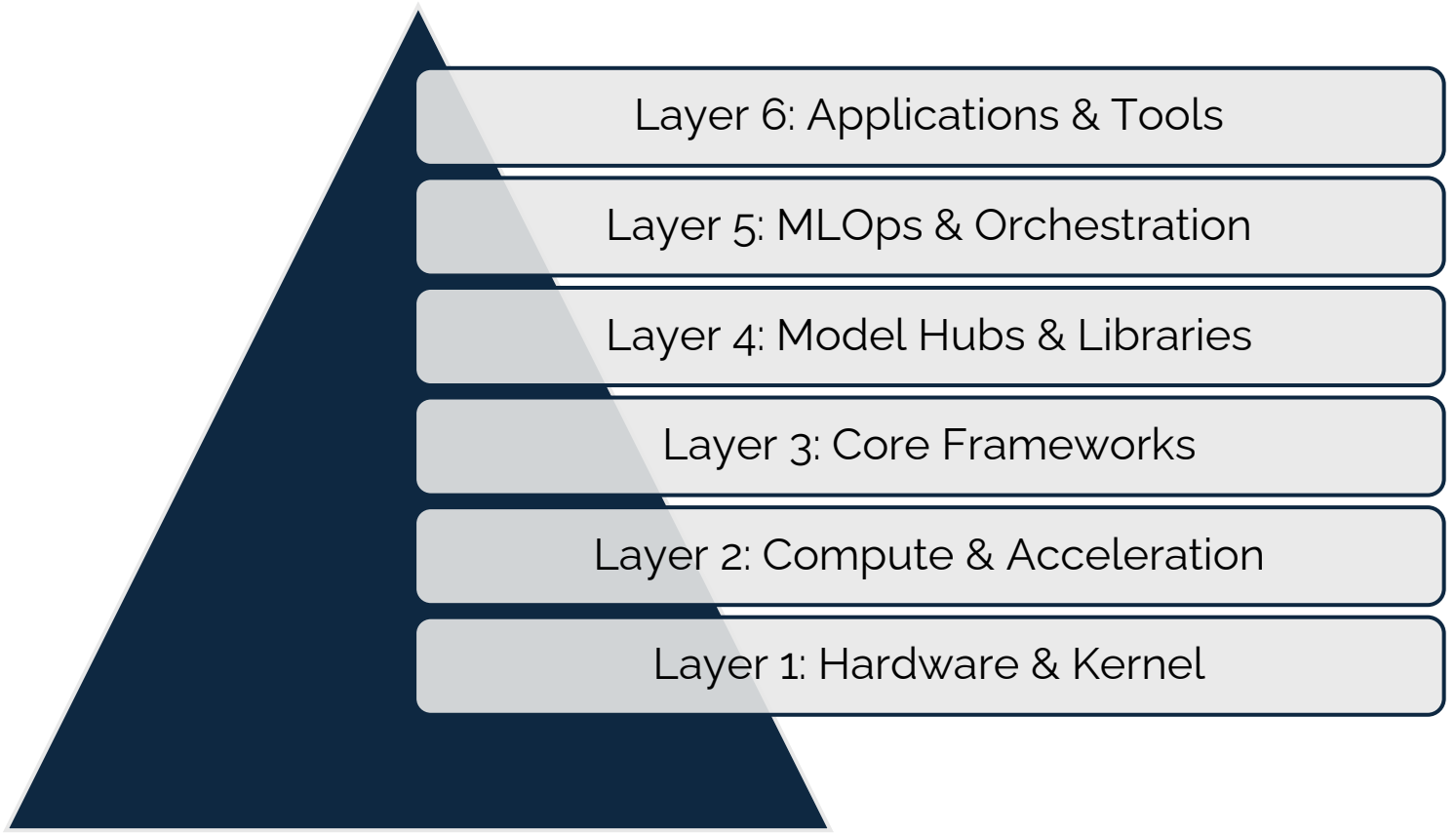
2015: The Big Bang. TensorFlow (Google) is released, mainstreaming neural network development.

2016: The Challenger. PyTorch (Facebook) emerges with a more Pythonic, dynamic approach.

2018-Present: The MLOps Era. Proliferation of tools for data, training, and deployment (Kubeflow, MLflow).

2020-Present: The LLM Revolution. Hugging Face becomes the "GitHub for AI." Transformers library democratizes NLP. Release of models like GPT-Neo, BLOOM, Llama.

The Open Source AI Stack



Layer 1 & 2: Hardware & Compute Acceleration

- **Hardware Abstraction**: CUDA, ROCm, oneAPI
- **Containerization**: Docker is universal for packaging environments.
- **Orchestration**: Kubernetes (K8s) for managing scalable, distributed training and inference workloads.
- **Cloud & Cluster Management**: Slurm, AWS Batch.

Layer 3: Core Frameworks - The Engines

Feature	TensorFlow	PyTorch
Origin	Google	Meta (Facebook)
Graph	Static by default (Graph mode)	Dynamic by default (Eager execution)
Strength	Production deployment (TFX, TFLite)	Research flexibility, debugging
Ecosystem	Mature, vast production tools	Hugging Face, strong research community
Trend	Adopting more dynamic features	Improving production capabilities (TorchScript)

Layer 4: Model Hubs & Specialized Libraries

- **Hugging Face Hub**: A platform with over 300,000 models, 50,000 datasets, and 50,000 demos.
- **transformers Library**: The standard API for using thousands of pre-trained models (NLP, Vision, Audio).
- **datasets Library**: Efficient access and preprocessing for thousands of datasets.
- **accelerate Library**: Simplifies running models on any hardware configuration.

Layer 5: MLOps & Orchestration

- **Experiment Tracking**: MLflow, Weights & Biases (partly open). Track parameters, metrics, and artifacts.
- **Workflow Orchestration**: Kubeflow, Airflow, Prefect. Automate the full ML pipeline.
- **Model Serving**: KServe, Seldon Core, Triton Inference Server. High-performance, scalable serving.
- **Data Versioning**: DVC (Data Version Control). Git for data.
- **Feature Stores**: Feast. Manage and serve pre-computed features.

Layer 6: Applications & End-User Tools

- Open Source Alternatives to ChatGPT:
  - ☞ **GPT4All**: Run LLMs locally on your laptop.
  - ☞ **Ollama**: User-friendly tool for running and managing local LLMs.
  - ☞ **LocalAI**: Drop-in replacement for OpenAI API, but self-hosted.
- **Image Generation**: Stable Diffusion WebUI (AUTOMATIC1111) - a powerful desktop GUI.
- **Code Assistants**: Tabby, CodeGeeX. Self-hosted alternatives to GitHub Copilot.

Case Study: The Llama 2 Ecosystem

- **Meta releases Llama 2**: A powerful LLM with a custom "community license."
- Immediate Community Response:
  - ☞ **Quantization**: llama.cpp allows running on consumer hardware.
  - ☞ **Fine-tuning**: Libraries like Axolotl, Unsloth, and TRL make it easy to adapt Llama 2.
  - ☞ **Interfaces**: Tools like Ollama and GPT4All integrate it seamlessly.
  - ☞ **Specialized Variants**: Medical Llama, Code Llama, Vicuna (from fine-tuning).
- **Result**: A vibrant, innovative ecosystem that Meta alone could not have built.

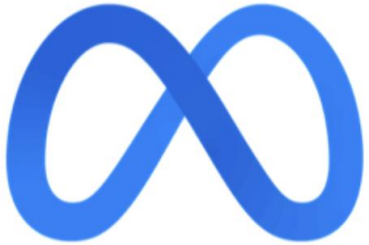


***Image source**: Dida, "Screenshot of Meta and Dida icons," Dida Blog, 9-Dec-2024. [Online]. Available: https://cdn.dida.do/blog/screenshot-2024-12-09-at-12.57.07.png. [Accessed: 01-Nov-2025].*

SE in the Open - Development Lifecycle & Tools

Software Engineering Practices for Open Source AI

How does building AI software differ from traditional software?

- **The "Data" Component:** Data is a first-class citizen, requiring its own versioning and lineage.
- **The "Model" Component**: The model is a new, non-deterministic artifact to be managed, versioned, and deployed
- **Experimentation is a Core Phase**: The path from idea to code is not linear but highly iterative.
- **Reproducibility is Paramount**: An experiment's result must be recreatable exactly.

The AI Software Development Lifecycle (SDLC)

1. Problem Formulation & Data Collection (OSS Datasets, Web Scraping)
2. Data Preprocessing & Exploration (Pandas, DVC)
3. Model Experimentation & Training (PyTorch/TF, MLflow tracking)
4. Model Evaluation & Validation (Robust metrics, bias detection)
5. Model Packaging & Deployment (Docker, Seldon Core, KServe)
6. Monitoring & Continuous Learning (Evidently AI, Prometheus)

Documentation as a Product

- ☞ **Getting Started Guide**: A simple "5-minute" tutorial.
- ☞ **API Reference**: Auto-generated from docstrings (Sphinx, MkDocs).
- ☞ **Conceptual Guides**: Explaining why, not just how.
- ☞ **Tutorials & Notebooks**: Jupyter notebooks are the standard for interactive examples.
- ☞ **Model Cards & Datasheets**: Standardized documentation for models and datasets, detailing intended use, limitations, and biases.

Testing & CI/CD for AI

- **Code Testing**: Standard unit/integration tests (pytest).
- **Data Validation**: Check for schema drift, data skew, anomalies (Great Expectations).
- **Model Validation**: Testing for fairness, bias, and minimum performance thresholds.
- **CI/CD for ML**: Automate testing, training, and deployment.
- **Example GitHub Actions workflow**: On a PR, it runs tests, trains a small model, and validates its accuracy before merging.

The Inevitable Challenges of Openness

The Future & Strategic Implications

Where is This All Heading?

- Trend 1: The Rise of Smaller, Specialized Models
- Trend 2: Multi-Modality as Standard
- Trend 3: AI for AI: Automated Development
- Trend 4: The Edge Computing Frontier
- Trend 5: Tighter Regulation and "Verified" Openness

Trend 1: Smaller, Smarter Models

Why? Cost, speed, privacy, and customizability.

- **Techniques**:
  - ☞ **Quantization**: Reducing numerical precision of weights (e.g., from 16-bit to 4-bit).
  - ☞ **Distillation**: Training a small "student" model to mimic a large "teacher" model.
  - ☞ **Pruning**: Removing redundant neurons or connections.
  - ☞ **Mixture-of-Experts** (MoE): Using only parts of a model for a given task.
- **Examples**: Microsoft's Phi-2, Google's Gemma.

Trend 2: Multi-Modality by Default

- **What?** Models that can seamlessly understand and generate across different types of data.
- Open Source Examples:
  - ☞ LLaVA: Large Language and Vision Assistant.
  - ☞ ImageBind (Meta): A model that links six modalities.
  - ☞ Stable Diffusion: Text-to-Image is just the beginning.
- **Implication**: Richer, more intuitive human-computer interaction. An OS model that can be queried with a picture, a sound, and text simultaneously.

Trend 3: AI for AI (Automated Development)

Using AI to Build AI

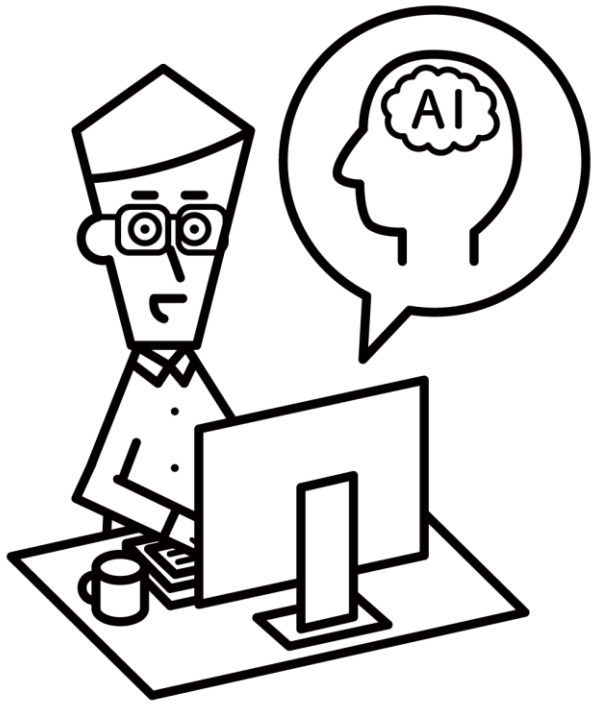
- **Automated Data Labeling**: Using a foundation model to label data for a smaller, specialized model.
- **Hyperparameter Tuning** & **Architecture Search**: Tools like Weights & Biases Sweeps and Optuna.
- **AI-Powered Code Generation**: Using LLMs (like Code Llama) to generate boilerplate ML code, documentation, and tests.
- **Self-Improving Systems**: Models that can generate their own training data or fine-tune themselves based on feedback.

Trend 4: The Push to the Edge

Why? Latency, bandwidth cost, privacy (data never leaves the device), offline operation.

- Open Source Tech:
  - ☞ **TensorFlow Lite, PyTorch Mobile**: Frameworks for mobile and embedded devices.
  - ☞ **ONNX Runtime**: High-performance inference engine across diverse hardware.
  - ☞ **llama.cpp, MLC LLM**: Techniques to run LLMs on MacBooks, Raspberry Pis, and phones.
- **Use Cases**: Real-time translation, personalized assistants, industrial IoT.

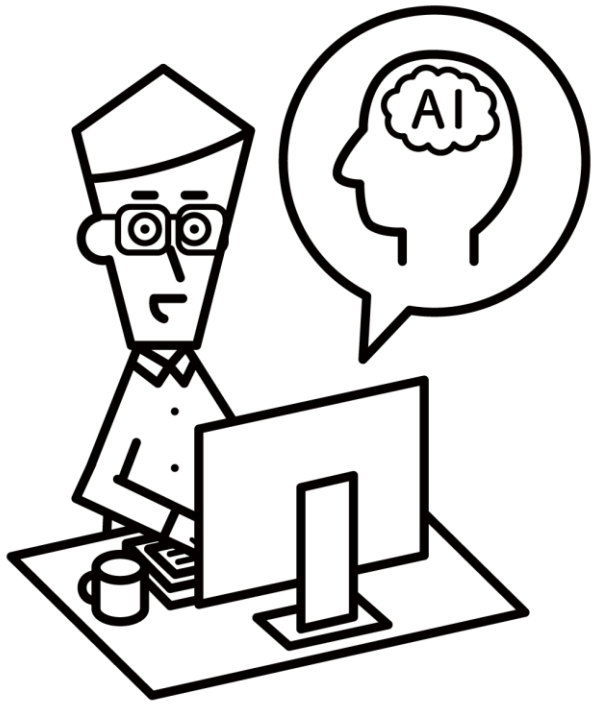
The Role of the Open Source AI Engineer



1. **Deep ML Knowledge**: Understanding of model architectures, training techniques.
2. **Software Engineering Mastery**: Systems design, testing, CI/CD, containers, orchestration.
3. **Data Engineering**: Pipelines, ETL, data governance.
4. **MLOps Expertise**: Proficiency with the toolchain for production AI.
5. **Legal & Ethical Acumen**: Understanding licenses, bias, and societal impact.
6. **Community Skills**: Collaboration, communication, and mentorship.

***Image source**: "Illustration of a person contemplating AI," Kuku Keke, 2020. [Online]. Available: https://kuku-keke.com/wp-content/uploads/2020/03/14337_1.png. [Accessed: 01-Nov-2025].*

How to Dive Into the Open Source AI World?



1. **Foundations**: Python, Git, basic ML theory.
2. **Core Framework**: Pick PyTorch (recommended for newcomers) and do tutorials.
3. **Hugging Face Ecosystem**: Learn the transformers and datasets libraries. Try fine-tuning a model.
4. **MLOps Tooling**: Get hands-on with MLflow and DVC on a personal project.
5. **Contribute**: Find a project on GitHub, start by fixing a typo in the docs, then tackle a "good first issue."

***Image source**: "Illustration of a person contemplating AI," Kuku Keke, 2020. [Online]. Available: https://kuku-keke.com/wp-content/uploads/2020/03/14337_1.png. [Accessed: 01-Nov-2025].*

Key Open Source AI Projects to Watch

- **Frameworks**: PyTorch, TensorFlow, JAX
- **Model Hubs & Libraries**: Hugging Face transformers, diffusers, accelerate
- **MLOps**: MLflow, Kubeflow, Prefect, Seldon Core
- **Inference**: vLLM, ONNX Runtime, Triton
- **Local LLMs**: Ollama, llama.cpp, LM Studio
- **Application Frameworks**: LangChain, LlamaIndex

Summary

- **A Powerful Partnership**: Open Source and AI are inseparable, driving democratization, collaboration, and transparency.
- **A Full-Stack Ecosystem**: A mature stack exists, from hardware (Layer 1) to user applications (Layer 6), all powered by open source.
- **A New Engineering Discipline**: Building AI requires new practices for managing data, models, and the experimental lifecycle (MLOps).
- **Significant Challenges**: Openness brings complex legal, ethical, security, and economic challenges that the community is actively tackling.
- **An Exciting Future**: Trends point towards smaller models, multi-modal AI, automated development, and edge computing.
- **Your Journey Awaits**: Start with Python, PyTorch, and Hugging Face, then contribute to a project to join the global community.

Thank you!

"Open source software is a testament to the power of collaboration; it transforms ideas into innovations, empowering individuals and communities to build a better future together."

Lecturer: Biniam Behailu
Addis Ababa Science and Technology University