OpenSSF Tech Talk

Securing Agentic AI in Practice: From OpenSSF Guidance to Real-World Implementation

March 17, 1PM ET

OpenSSF

OPEN SOURCE SECURITY FOUNDATION

Welcome!

- Thank you for joining us today! We will begin at 1:02PM ET
- While we wait for everyone to join, please take a moment to do one (or more) of the following:
  - Please add questions using the Zoom Q&A feature
  - Follow us on X: @openssf, Mastodon: social.lfx.dev/@openssf, LinkedIn: OpenSSF, Bluesky: @openssf.org
  - Visit our website: https://openssf.org
  - Sign up for training: https://openssf.org/training/courses/
- **This Tech Talk is being recorded and slides will be available!**

OpenSSF

OPEN SOURCE SECURITY FOUNDATION

Agenda

- Housekeeping
- Speaker Introductions
- Understanding Agentic AI Security
- SAFE MCP
- A POV. from AI Infrastructure
- Panel Discussion & Audience Q&A
- Important announcements

Help us improve! Tech Talk Survey

OpenSSF

OPEN SOURCE SECURITY FOUNDATION

Antitrust Policy Notice

Linux Foundation meetings **involve participation by industry competitors**, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at **http://www.linuxfoundation.org/antitrust-policy**. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

Code of Conduct

- The Linux Foundation and its project communities are **dedicated to providing a harassment-free experience** for participants at all of our events, whether they are held in person or virtually.
- All event participants, whether they are attending an in-person event or a virtual event, **are expected to behave in accordance with professional standards**, with both this Code of Conduct as well as their respective employer's policies governing appropriate workplace behavior and applicable laws.
- https://openssf.org/community/code-of-conduct/

Q&A

Please submit your questions during the meeting by using the Q&A feature on Zoom.

Thank you!

Introductions

Yesenia Yser

Yesenia Yser, Senior Security Program Manager, Microsoft

Yesenia architects secure AI and software systems by combining 12+ years of application security, supply chain security, digital forensics, and open-source security leadership with deep hands-on research in AI sandboxing and agent integrity.

Yesenia is currently empowering the world with changes for AI Security & Safety and Open Source Security at Microsoft.

She is the founder of The Lioness Instincts, a nonprofit dedicated to educating women across the U.S. in digital privacy and physical self-defense.

You can catch Yesenia as a co-host of the OpenSSF “What’s in the SOSS” podcast, co-leading the BEAR WG, or within the OSSAfrica WG.

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Angela McNeal, CEO & CO-Founder of ThreadAI

Angela McNeal is the CEO and co-founder of Thread AI, an AI company that provides AI orchestration infrastructure for enterprises deploying agents in critical processes and operations.

Prior to founding Thread, she served as Palantir Foundry's Head of AI Product. With a passion for cutting-edge technologies and a penchant for problem-solving, Angela played a pivotal role in driving innovation and shaping the company's AI product strategy.

Angela is known for her advocacy of data privacy, responsible AI, and ethical data handling, shaping policies and best practices.

Frederick Kautz, OpenSSF AI/ML Security Working Group, SAFE-MCP SIG Maintainer

Frederick Kautz is a security architect specializing in AI/ML security and software supply chain integrity. He is a co-author of NIST SP 800-204D and the CNCF Cloud Native Security Whitepaper, advancing secure practices across cloud-native ecosystems.

He previously co-chaired KubeCon + CloudNativeCon and maintains the SAFE-MCP project, contributing to OpenSSF initiatives defining best practices for securing AI systems, agent-based architectures, and verifiable pipelines.

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Hugo Huang, Public Cloud Alliance Director, Canonical

Hugo has 20 years of experience in digital transformation across open source, cloud computing, AI, and cybersecurity. He works closely with hyperscalers and open-source communities to advance secure and scalable AI infrastructure.

Hugo is also a Harvard Business Review contributor, writing on AI security, infrastructure resilience, and the economics of generative AI systems. His work bridges technology, security, and business strategy to help organizations deploy AI responsibly at scale.

He holds an MBA from MIT Sloan School of Management.

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Abdelrahman Hosny

Sr. Silicon Alliances Manager, Canonical

Abdelrahman works at the intersection of AI research and product engineering. With a PhD from Brown University and experience building and deploying AI models at Apple, he helps organizations navigate the gap between cutting-edge research and production-ready AI.

Having built and scaled a YC startup, he has been developing AI strategies, building ML infrastructure, and helping engineering teams deploy and scale models that create real business value.

With 12+ years of tech industry experience, he has contributed to open-source tools, published research in top venues, and built ML pipelines that serve thousands of users.

Understanding Agentic AI Security

Angela McNeal
CEO & Co-Founder, Thread AI

Key problem statements

- Agent Autonomy
- Tool / Model Interaction Trust
- Context Integrity and Misuse

A good question to keep in mind: “ When AI Agents Execute, Who's Responsible?”

Agentic AI ≠ Traditional Software: The Threat Model Changed

Traditional Software

- Deterministic execution paths
- Static access, enumerable at deploy time
- Failures are loud, exceptions, crash dumps
- Authorization is request-scoped and explicit
- Audit = log the API call

AI Agent

- Non-deterministic, same input ≠ same path
- Dynamic tool discovery, access grows at runtime
- Failures are often silent, 95% accuracy hides the 5%
- Auth must cover delegated and on-behalf-of and tool chains
- Audit = you need the reasoning chain, not just the call

Each problem area represents a distinct failure mode. Each requires different architectural responses. All must be addressed together.

Agent Autonomy: Unbounded Action

Agents follow the path of least resistance. Without explicit least-privilege boundaries, autonomy becomes a liability not a feature.

Access surface grows silently with every new workflow.

What can this agent access and do?

Tool/Model Trust: Unverified Interactions

Every MCP server, API, and tool is an untrusted boundary.

Prompt injection through tool responses, poisoned model outputs, and unverified server identity can cascade into real-world system writes.

Can I trust what this tool returned?

Context Integrity: Invisible Reasoning

When regulators ask "how was this decision made?" the answer must be complete and defensible.

Logging inputs and outputs isn't enough. The reasoning chain must be capturable, reproducible, and tamper-proof.

What did the agent do, and why?

Autonomy: Agents Follow the Path of Least Resistance

Failure Pattern:

A claims agent processing a family plan reimbursement without explicit access boundaries pulls PHI for every family member to verify related claims. It did nothing malicious. It was being thorough. That's exactly what makes unbounded agents dangerous. Each new workflow silently expands the attack surface.

Multi-Agent Delegation Abuse:

User: scope → Agent A delegates → Agent B delegates → Agent C: scope=???

Without explicit delegation chain enforcement at each hop = the confused deputy problem at agent scale.

Architectural Response

- Dynamic, task-scoped access grants that expire on task completion - not session end
- RBAC baseline and Fine-Grained Authorization for per-resource policies
- RFC 8693 token exchange: on-behalf-of steps enforced to initiating user's exact scope
- Explicit delegation chain binding
- Zero hardcoded credentials in workflow logic, secrets managed at infra layer

Trust: Every Tool Response Is an Untrusted Boundary

Why This Is Different for Agents:

The model has no native mechanism to distinguish content being analyzed from instructions to follow. Tool-use agents have real write access - the blast radius is not a confusing output, it's a committed transaction in a system of record.

Also in this threat category:

MCP server impersonation , dynamic tool registration abuse, data poisoning, backdoored model weights, specification gaming, token scope creep across servers

Defense-in-Depth: Four Layers

- **Model - Instruction hierarchy:** system prompt weighted over user turn. RLHF/CAI for injection robustness.
- **Agent - Tool output sanitization:** responses parsed as structured untrusted data - never re-injected as instructions. Output schemas validated before writes.
- **Infra- Explicit allow-lists:** agent cannot discover or invoke tools outside the workflow's predefined set. Dynamic registration requires explicit re-auth.
- **Human- Confidence-gated HITL:** irreversible writes require human confirmation below confidence threshold or on first-time action type.

Context: Logging Output Isn't Enough, You Need the Reasoning Chain

Real World Failure:

A bank deploys an AI lending workflow. It performs well for months. A regulatory examiner asks: "Walk me through exactly how this decision was made." The team can show input and output but the reasoning chain: which data sources were consulted, which rules applied, where the model's judgment influenced the outcome is not captured in a defensible or reproducible way. The infrastructure was never designed to record it.

Context Misuse Attack Vectors

- Context window poisoning: attacker controls earlier turns in a long conversation to shift agent behavior on later turns
- Memory / RAG injection: poisoned documents in the retrieval corpus influence grounded responses
- Session bleed: state from one user's execution leaking into another's context window

What Full Context Integrity Requires

- Distributed tracing that separates *decision* (what model planned) from *action* (API call executed) as distinct audit events
- Chain-of-thought captured per step not just terminal output and tamper-proof storage
- Full data input provenance: which sources, which versions, at what timestamp
- PII auto-redacted from logs before storage; retention policies at platform level
- State isolation between concurrent executions with zero context leakage across runs
- When HITL fires: human identity, rationale, and approved action all logged as part of the audit trail

Controlled Autonomy: Addressing All Problem Areas Together

Agent Autonomy: CONTROL

Dynamic task-scoped access grants

RBAC and FGA (ReBAC/ABAC)

RFC 8693 on-behalf-of scoping

Delegation chain enforcement per hop

Secrets at infra layer, zero hardcoded

Tool/Model Trust: RELIABILITY

Exactly-once via idempotency keys

Step-level checkpointing - resume, not restart

Output schema validation before writes

State isolation across concurrent runs

Exp. backoff + jitter for transient faults

Context Integrity: GOVERNANCE

Decision traced separately from action

Chain-of-thought per step, tamper-proof

HITL as native primitive

PII auto-redacted before log storage

Human identity and rationale in audit trail

Landscape of Agent Identity Standards

OAuth 2.1
- Delegated access, scoped tokens
- MCP's primary auth method. Needs Client Credentials + PKCE for agent flows.

OIDC
- Authentication + identity tokens
- Supports "on-behalf-of" for delegation chains.

SPIFFE / SPIRE
- Workload attestation, SVIDs
- Cyptographic identity per agent workload, short-lived certs.

SCIM
- Identity lifecycle mgmt
- Designed for humans - Agent identity metadata (version, model, capabilities) not standardized.

NGAC
- Fine-grained access control
- Graph-based policies, event-driven updates, native delegation.

Some Fun Open Questions from Broader Community:

- Ephemeral vs. fixed identity: should agent identity be task-scoped (tied to a specific execution) or persistent (tied to the agent deployment)?
- Delegation binding: when Agent A spawns Agent B, how do you cryptographically bind the delegation chain back to the originating human authorization?
- Context sensitivity in authz: if an agent aggregates data from 3 sources, can it access the aggregated result even if the user can't access all 3 individually?
- Capability attestation: what metadata in an agent's identity should declare what it CAN do, vs. what it's authorized to do?

Build the Security Primitives!

Remember! Every AI-driven decision touching a customer, financial record, or compliance obligation must be reproducible and defensible 30 days later. Regulators don't distinguish between a human decision and an agent decision - the organization is equally accountable for both.

The threat model is fundamentally different for Agentic AI (e.g. contextual non-determinism, dynamic access, silent failures, etc.)

Apply existing security frameworks, but know where they need to be extended!

Secure AI/ML-Driven Software Development (LFEL1012)

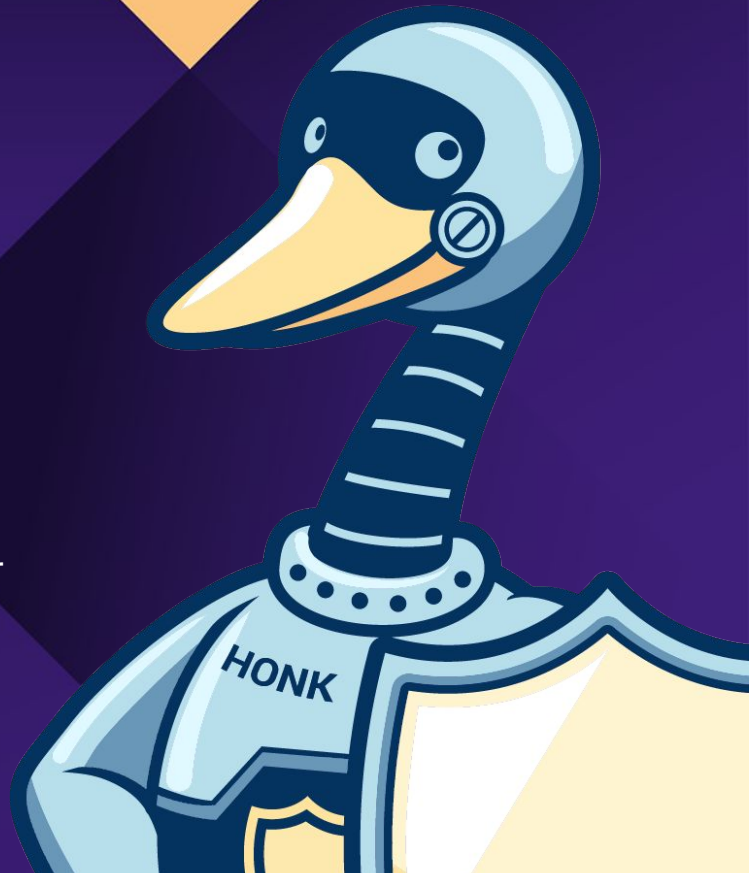
This course is designed for anyone developing software, either closed source or open source software. Our goal is to help you use AI for software development while maintaining security.

Course Outline

- Chapter 1. Course Introduction
- Chapter 2. Key AI Concepts for Secure Development
- Chapter 3. Security Risks of Using AI Assistants
- Chapter 4. Best Practices for Secure Assistant Use
- Chapter 5. Writing More Secure Code with AI
- Chapter 6. Reviewing Changes in a World with AI
- Chapter 7. Wrap-Up

SAFE MCP

Frederick Kautz
AI/ML Security Working Group, SAFE-MCP SIG Maintainer



What is SAFE-MCP

SAFE-MCP is a threat catalog for agentic AI systems

- Structured Catalog of attacks against Tool-Based LLM Workflows
- Inspired by MITRE ATT&CK
- Crosswalks with major standards such as NIST SP 800-53r5

Scope

- Prompts and context injection
- Tool invocation and capability abuse
- Data exfiltration and lateral movement
- Multi-agent coordination risks

Goal: Standardize how we name, detect, and mitigate attacks in AI systems

Why We Need It

Agentic AI is the new attack surface

- Inputs are executable in practice
- Tools extend the blast radius of a prompt
- Context is dynamic and often untrusted
- Agents implicitly trust other agents

What breaks

- Traditional input validation models
- Static supply chain assumptions
- Clear Trust Boundaries

Today's Gaps are no shared language for:

- Prompt injection variants
- Tool misuse patterns
- Agent to agent abuse

SAFE-MCP Examples

- Initial Access | SAFE-T1001 | Tool Poisoning Attack (TPA) | Attackers embed malicious instructions within MCP tool descriptions that are invisible to users but processed by LLMs
- Persistence | SAFE-T1201 | MCP Rug Pull Attack | Time-delayed malicious tool definition changes after initial approval
- Privilege Escalation | SAFE-T1301 | Cross-Server Tool Shadowing | Malicious MCP servers override legitimate tool calls to gain elevated privileges

OpenSSF Relevance

Safe-MCP maps naturally to:

- SLSA concepts of provenance and integrity
- In-toto attestations for execution tracing
- Secure build to secure inference continuum

It enables

- Threat modeling for AI systems
- Standardized control across ecosystems
- Shared direction and response patterns

Call to Action

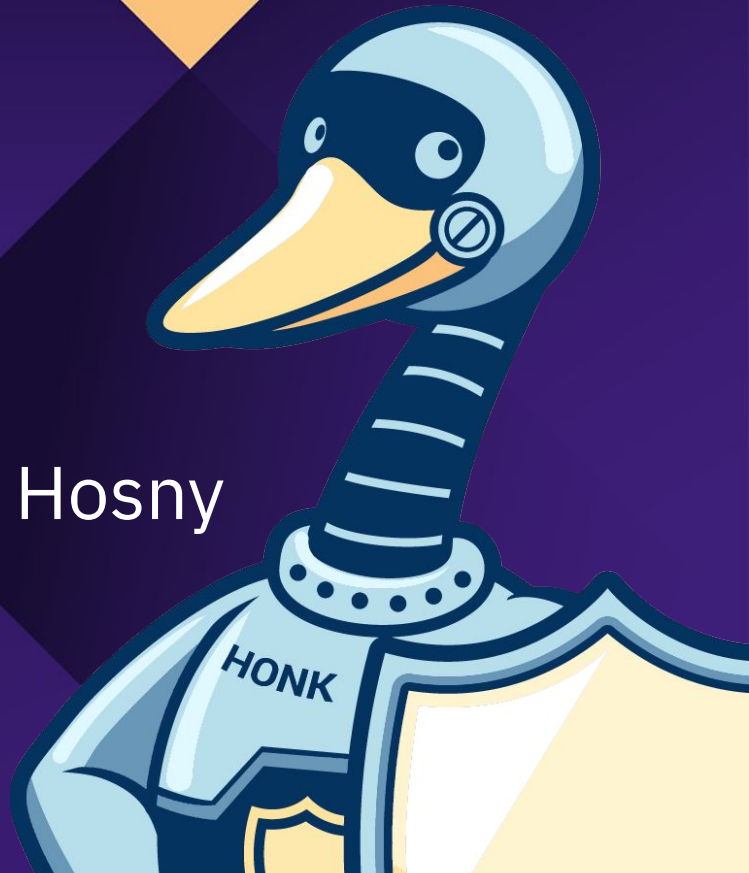
- Contribute techniques and real world use cases
- Contribute to the standards crosswalks
- Give us feedback on what works or what doesn't make sense
- Use it in your environment
- Recommend it to others

- https://github.com/safe-agentic-framework/safe-mcp

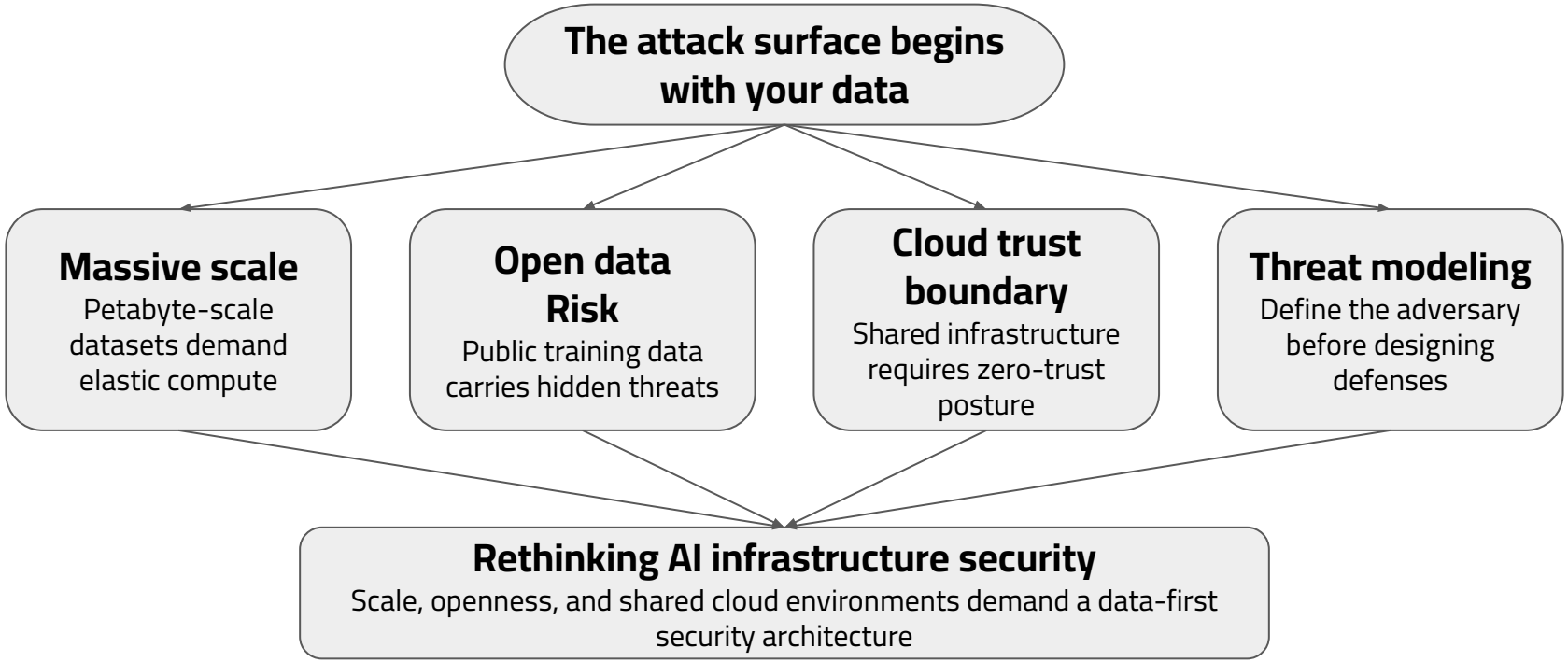
A POV. from AI Infrastructure

Hugo Huang and Abdelrahman Hosny

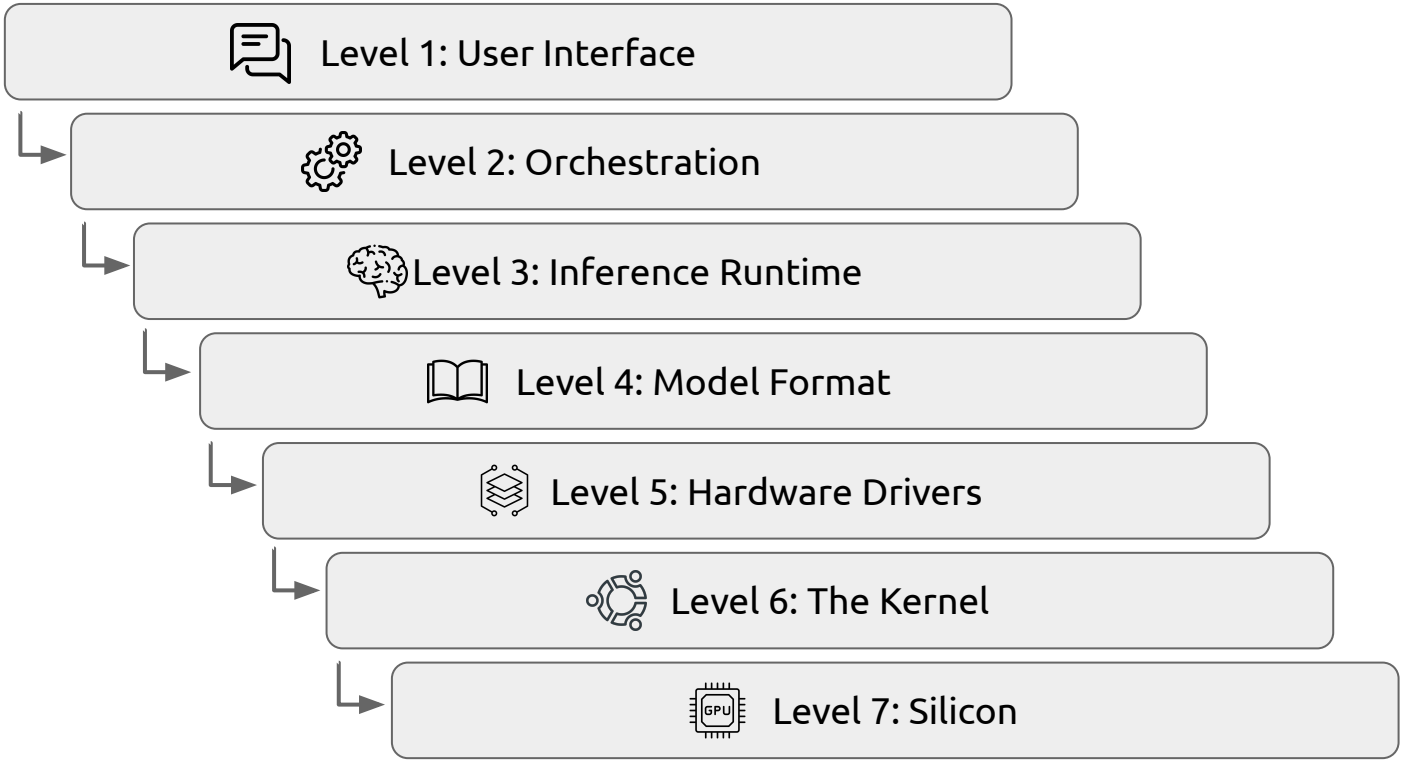
Public Cloud Alliance Director;
Silicon Alliance Manager, Canonical



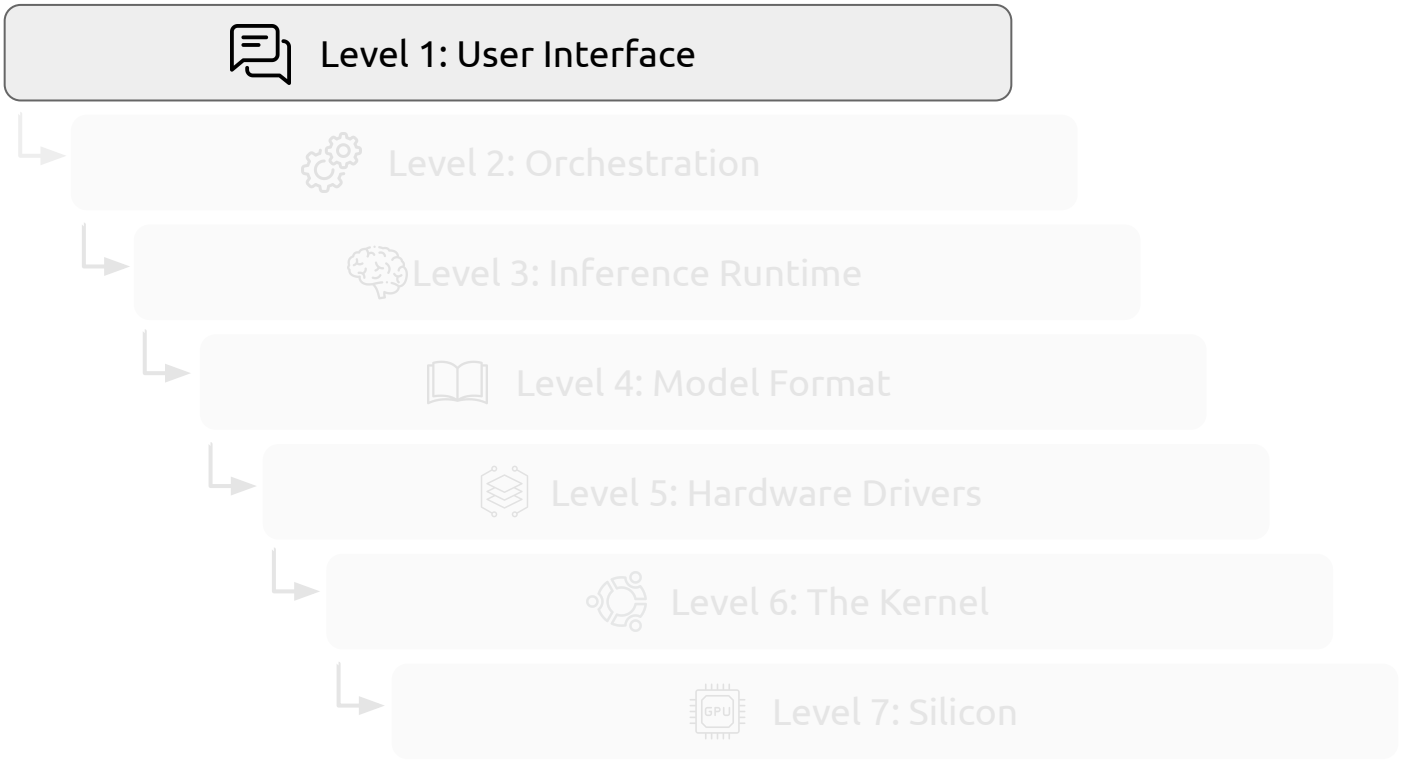
Rewriting the requirements for AI Infrastructure



From data to intelligence

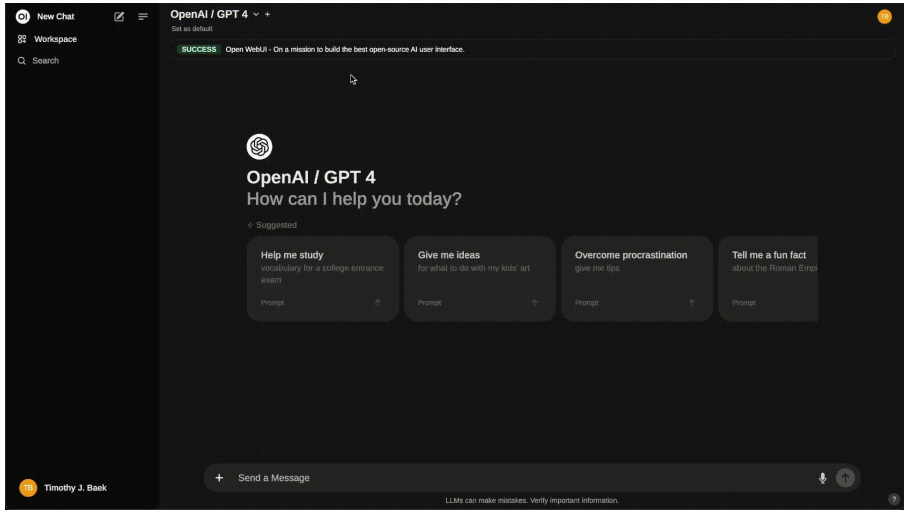


From data to intelligence



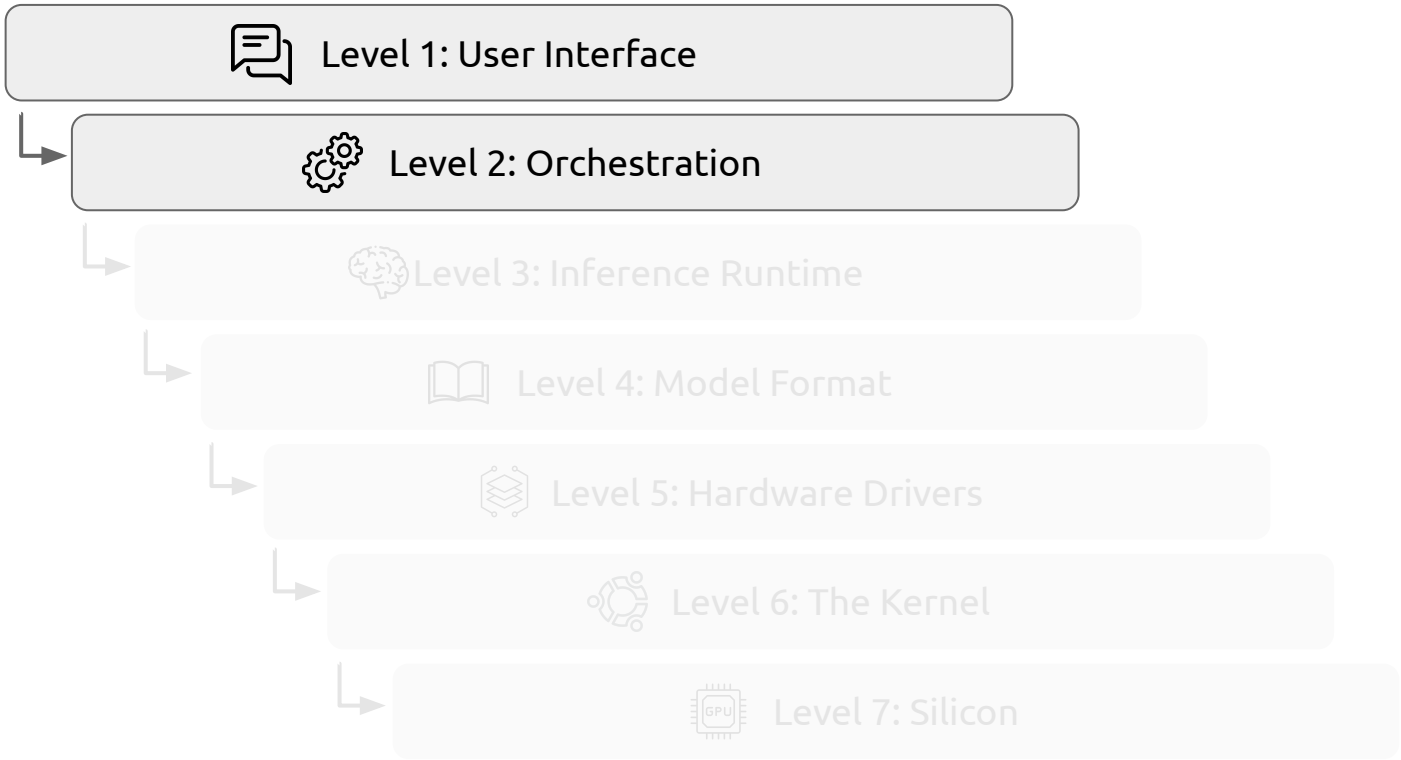
Level 1: User Interface

You type your prompt into a chat window.



Open Source		
Open WebUI	SillyTavern	Streamlit
Dependencies (SBOM)		
1,433	1,268	3,136

From data to intelligence



Level 2: Orchestration

The interface hands the message to a "manager" that keeps the model loaded and ready.

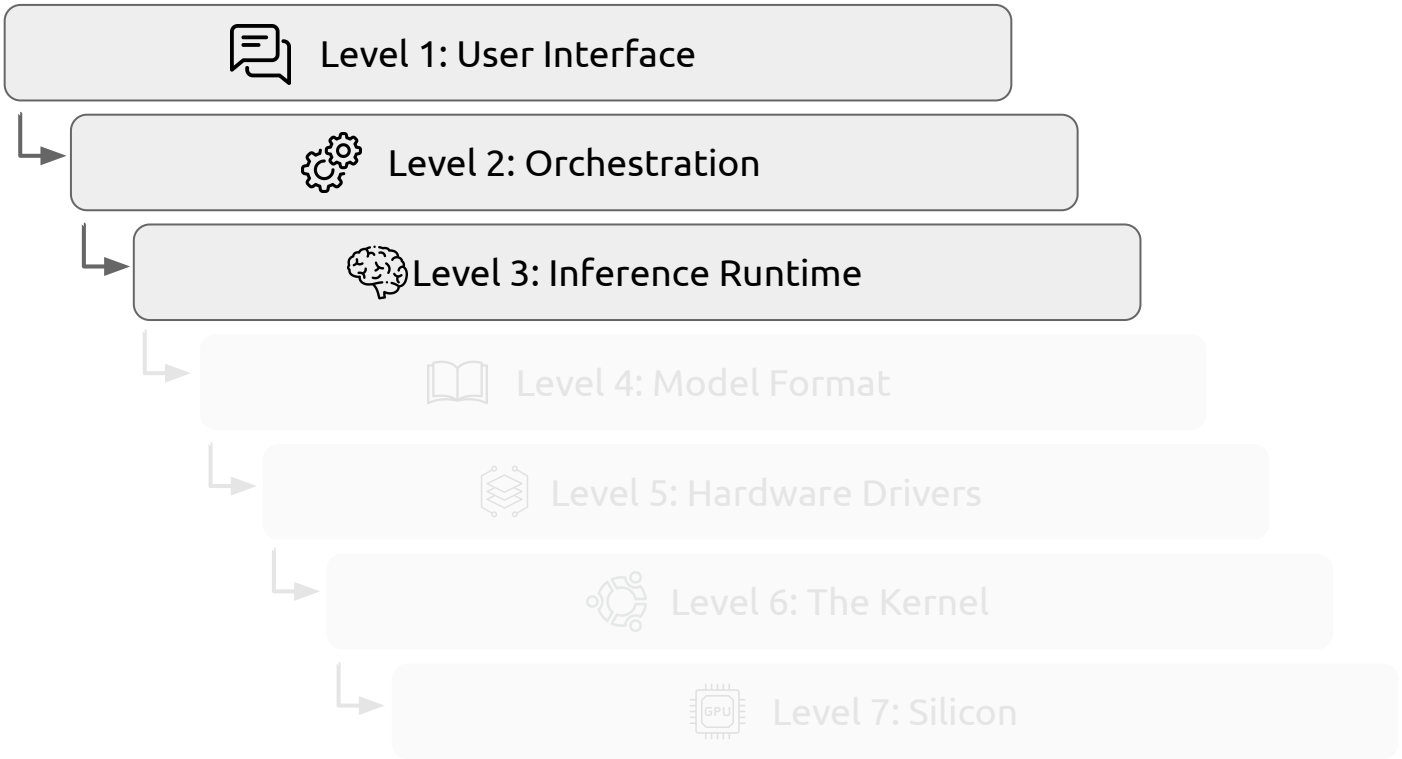
Bridge: request from the UI → API calls

Model resources: model loading,
memory management, concurrency, caching.

Standardizes access: universal launchpad,
tens of different models; no unique "launch codes"

Open Source		
Ollama	LocalAI	vLLM
		
Dependencies (SBOM)		
965	912	580

From data to intelligence



Level 3: Inference Runtime

The text is turned into math, and the "brain" begins to think.

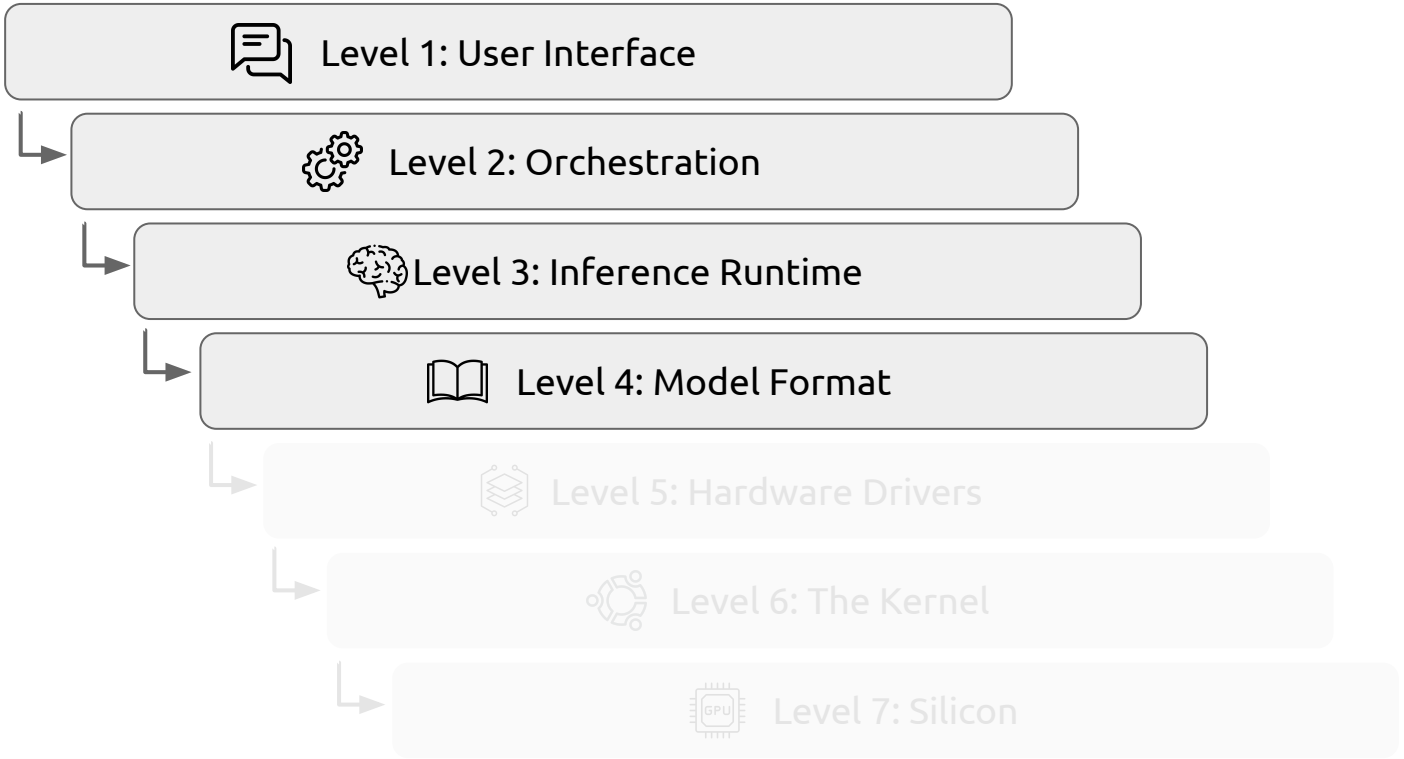
Execution: heavy mathematical lifting, text input into vector math, generating the response.

Optimization: code adaptation from training clusters to efficient deployments

Data movement: how data is stored in RAM and VRAM during the inference process

Open Source			
llama.cpp	TensorRT	Max	ONNX
			
Dependencies (SBOM)			
804	3,475	1,095	3,307

From data to intelligence



Level 4: Model Format

The engine reads the model's "brain structure" from a file on your hard drive.

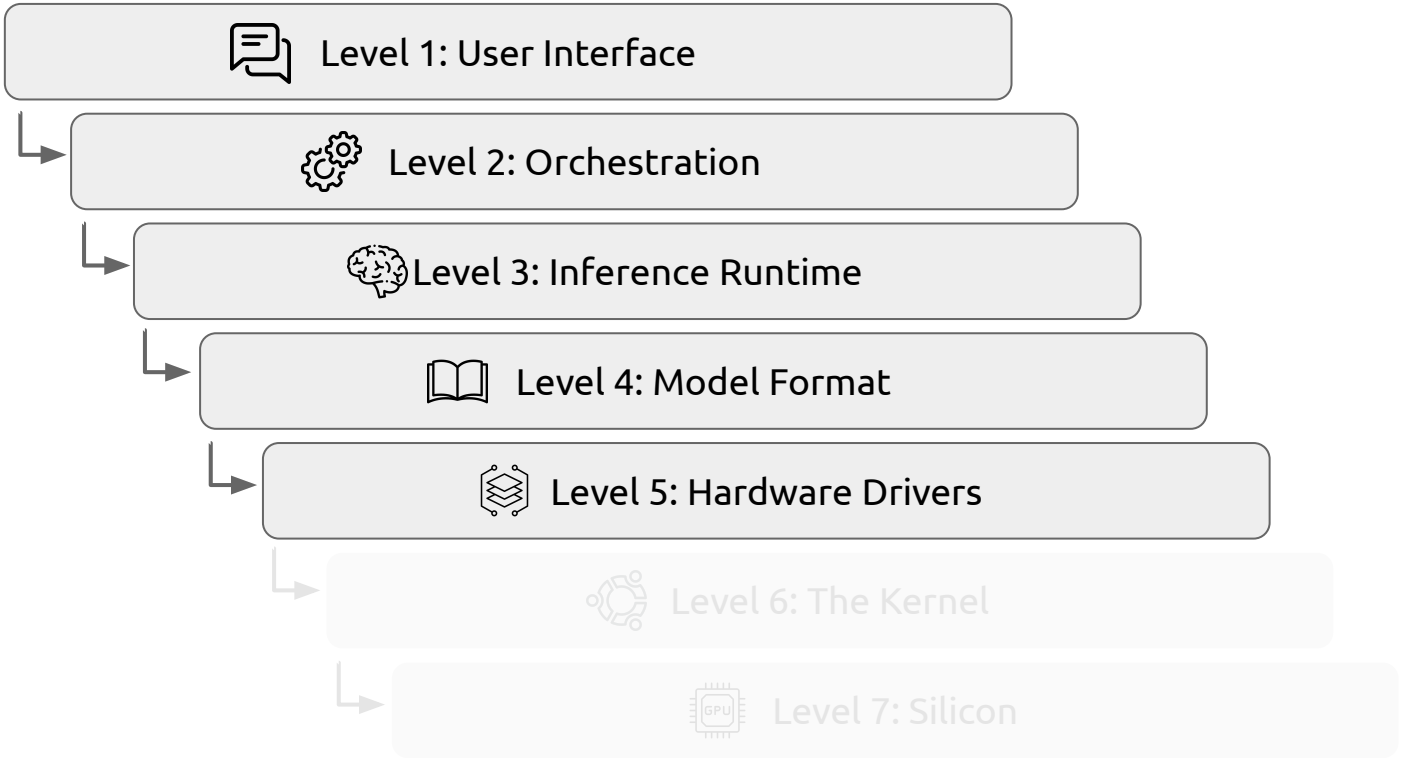
Stores the intelligence: the tangible file on your hard drive containing the model's neural network weights and parameters.

Reduces hardware strain: community-driven compression (quantization) to allow high-performance models to load faster and use less RAM.

Guarantees ownership: CapEx preservation

Open Source		
GGUF	Safetensors	PyTorch
		
Dependencies (SBOM)		
14	53	503

From data to intelligence



Level 5: Hardware Drivers

The software needs to talk to the hardware (GPU).

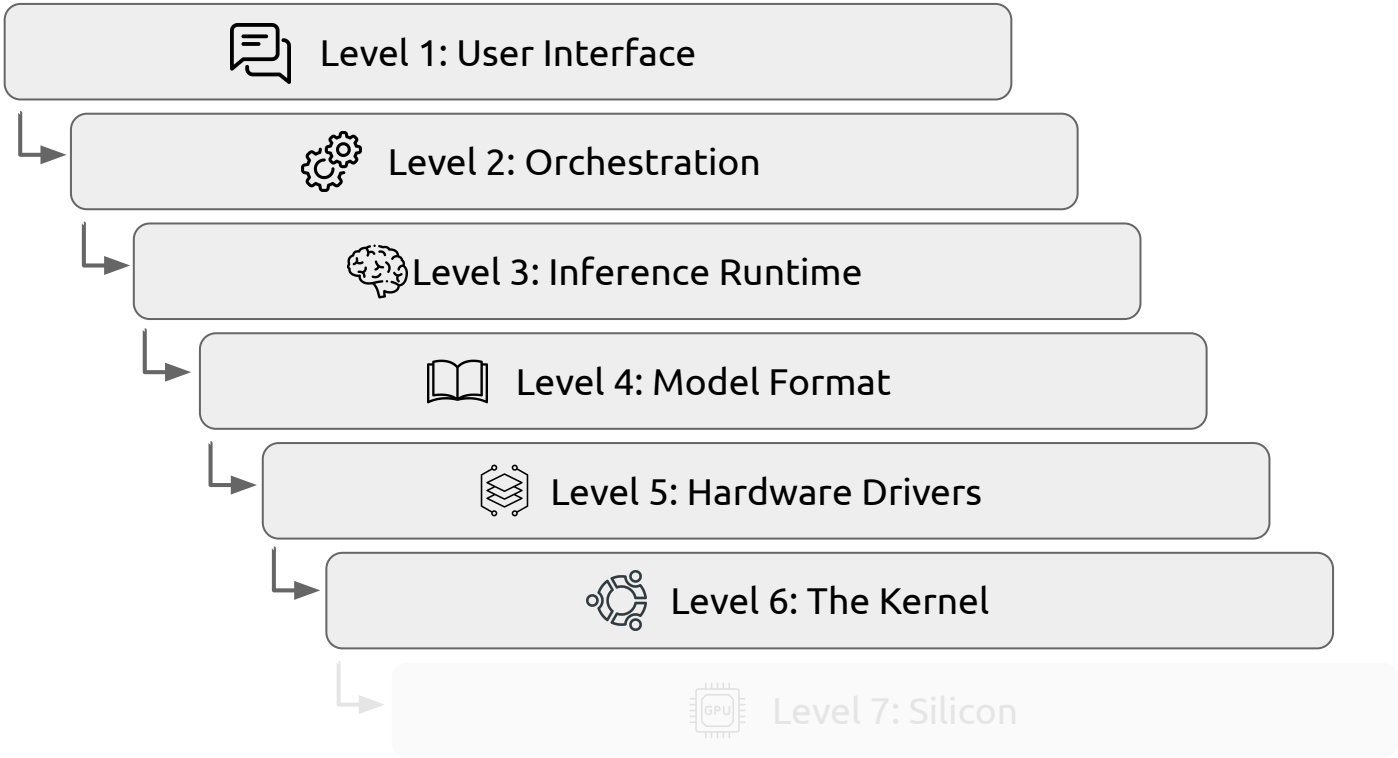
Translates for hardware: communicate instructions specifically to your physical accelerator/GPU.

Unlocks hardware power: OS can utilize the accelerator processing capabilities for AI math

Ensures long-term support: maintain hardware compatibility and fix bugs even if the original manufacturer stops supporting the device.

Open Source		
Intel IGCL	NV Open GPU Kernel	AMD GPU Open Drivers
		

From data to intelligence



Level 6: The Kernel

The bedrock that holds it all together.

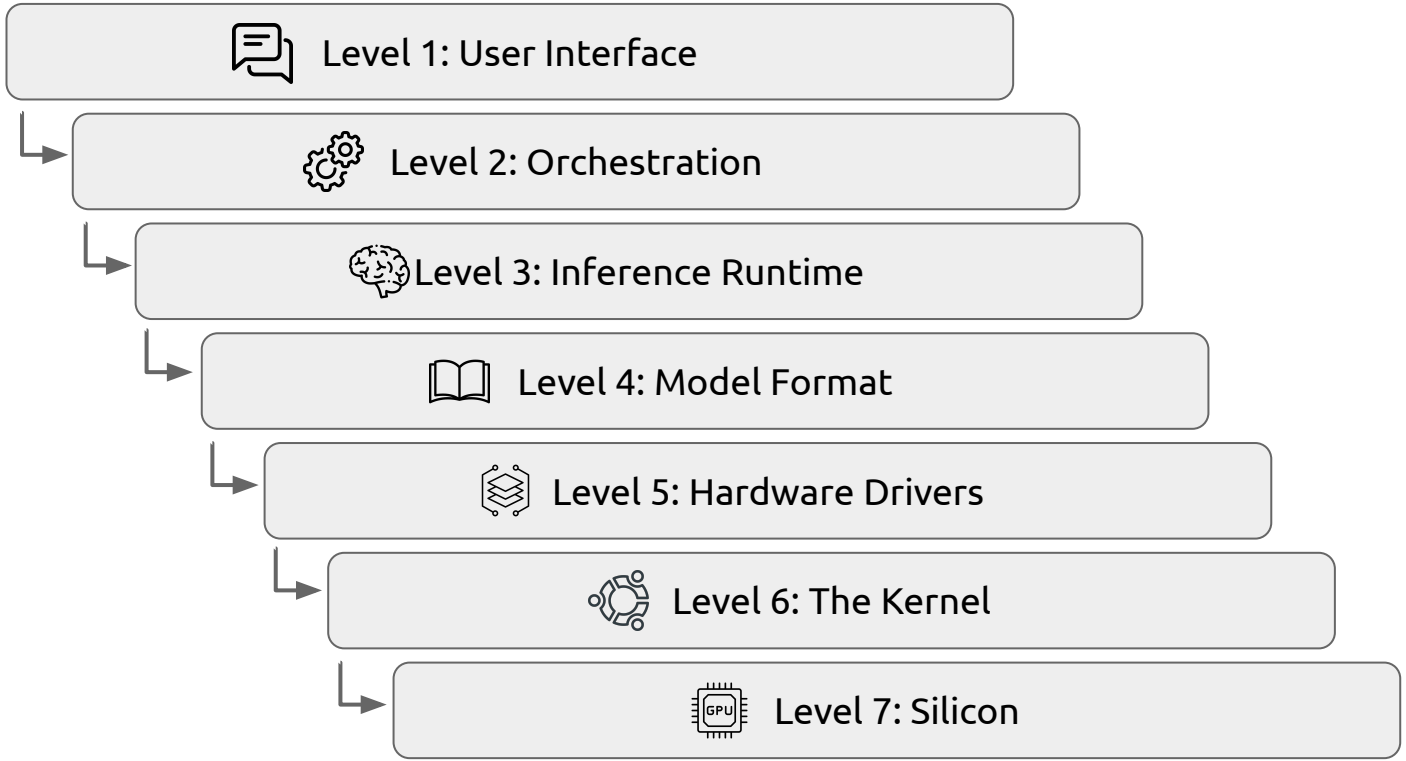
Orchestrates the ecosystem: the allocation of all system resources; memory, storage access, and processor cycles.

Provides the playground: creates the stable environment where drivers, engines, and interfaces can co-exist.

Administrative sovereignty: total control over the system, preventing OS-level surveillance or arbitrary restrictions on what software you can run.

Open Source		
Linux	Debian	Ubuntu
		

From data to intelligence



The AI Stack

Would the AI stack work without open source software?

In the AI Stack, more than **3,000 open-source dependencies** need to be secured against supply chain attacks

Get enterprise open source support

Vulnerability and patch management

Clear visibility into dependencies

Trustworthiness of the software source

Controlled security maintenance periods

Completion of in-house skills and experience

The top challenges for organizations when dealing with open source software supply chain. Source: IDC report, March 2025

Canonical commitment to open source

Timely fixes

Critical vulnerabilities fixed in 24h on average

Regression testing

Rigorously tested before the fix is published

Universal coverage

15 year security for over 40,000+ packages

API stability

No forced upgrades thanks to CVE fixes backporting

Panel Discussion & Audience Q&A

Call to Action!

Join the AI/ML Security Working Group!

Enroll LFEL1012 for free

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Upcoming Events

Learn more & register

Learn more & register

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Ways to Participate

- Join a Working Group/Project
- Come to a Meeting (see Public Calendar)
- Collaborate on Slack
- Contribute on GitHub
- Become an Organizational Member
- Keep up to date by subscribing to the OpenSSF Mailing List

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Engage with us on social media

X
@openssf

LinkedIn
OpenSSF

Mastodon
social.lfx.dev/@openssf

YouTube
OpenSSF

Facebook
OpenSSF

Bluesky
OpenSSF.org

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Is your organization a member?

Questions? Contact membership@openssf.org

openssf.org/join

Take our quick Tech Talk Survey

Help us improve!

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Thank You

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Legal Notice