harness

Building Enterprise-Ready CI/CD Pipelines using Agentic AI

Hello!

Chinmay Gaikwad

Technical Product Marketing of AI @ Harness

- DevOps
- K8s
- AppSec
- Observability

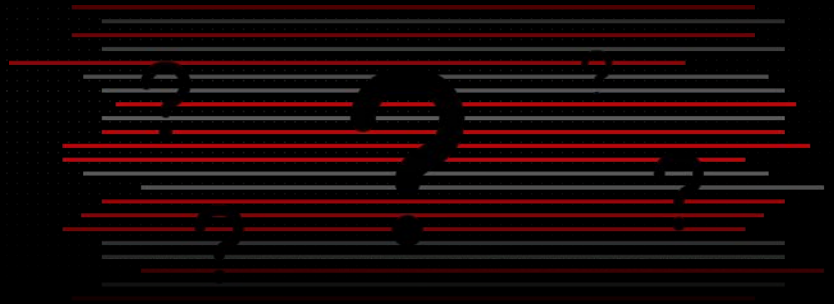
Agenda

- Whats and Whys of Enterprise-Readiness
- Role of Agentic AI and Knowledge Graph
- Governance

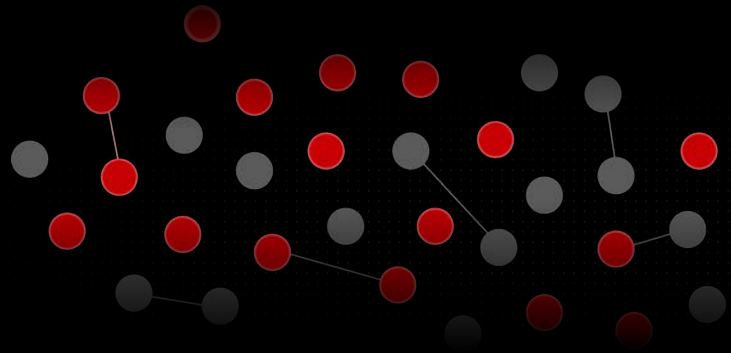
Enterprise-Readiness

Software delivery workflows today are manual & full of toil

Millions of lines of unmanageable home grown scripts & tools



Multiple disjointed point solutions



Velocity
60% of organization still release monthly or quarterly [1]

Efficiency
21% YoY increase in cloud-spend [2]

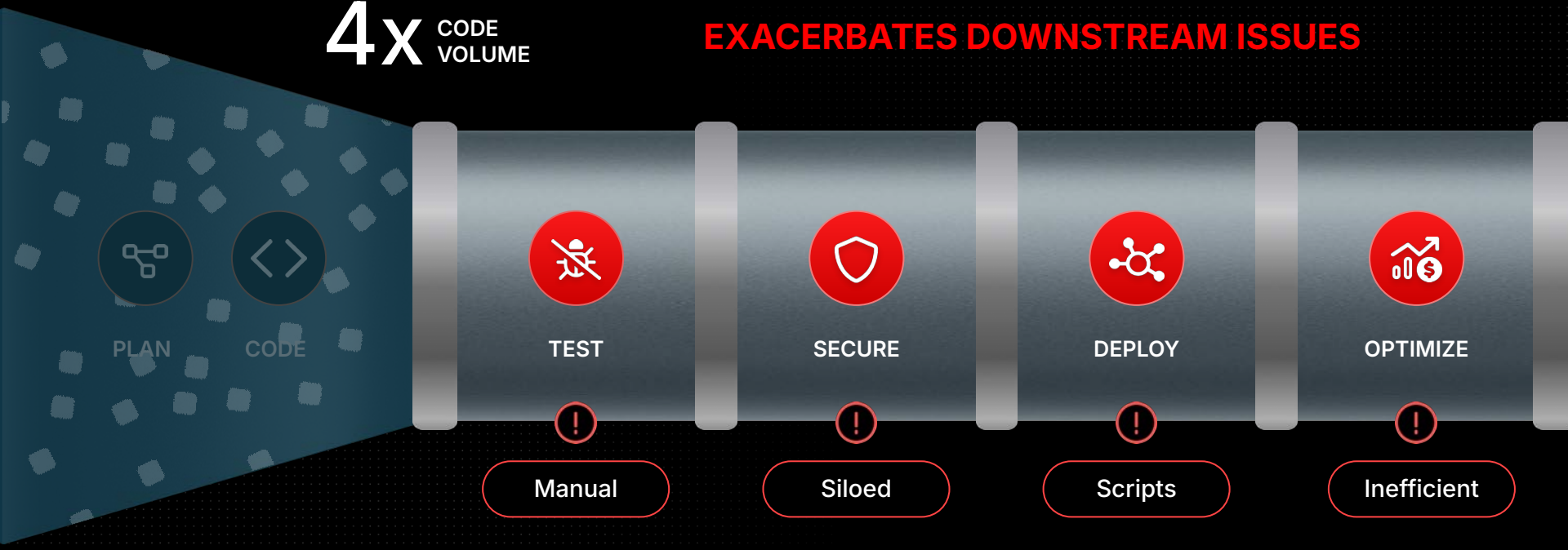
Security
41% of devs have no automated security and governance policies [1]

Resiliency
42% of devs cannot release to prod without risking failures [1]

[1] Source: Harness, 2024 State of Developer Experience Report,
[2] Source: https://www.canalys.com/newsroom/global-cloud-services-q3-2024

harness

Explosion of AI-Generated Code



AI Prototypes versus Enterprise Reality

For Software Delivery

AI Prototypes

- Takes less time to build
- Used to prove initial capacity
- Stops at prompt engineering

Enterprise AI

- Requires robust architecture
- Demands consistency and reliability
- Need to master context engineering

Majority of prototypes fail to scale in enterprise scenarios

Need for Enterprise-readiness

- 80% of SDLC failures occur AFTER code is written
- Tool sprawl without shared context = fragile pipelines
- Manual policy enforcement = governance bottlenecks
- Teams stuck balancing speed vs. safety

Enterprise pipelines must solve for velocity AND control simultaneously → this is the fundamental tension agentic AI can bridge.

What is an Enterprise-Ready pipeline?

Pipeline

Enterprise Ready

Scalable
Secure
Compliant

Enterprise-Readiness

- **Scalable**
  - Multi-environment pipeline generation from natural language
  - Automated resource optimization based on workload patterns
  - Dynamic infrastructure provisioning tied to deployment demands
  - Intelligent caching and artifact management across distributed teams
- **Secure**
  - Real-time vulnerability scanning and remediation before deployment
  - Shift-left security: SAST/DAST integrated into generation, not after
  - Supply chain integrity: track provenance of every artifact and dependency
  - Secret management automation: rotation, least-privilege injection
- **Compliant**
  - Compliance-by-default through policy-as-code automation
  - Audit trail for every artifact (who, what, when, how)
  - Regulatory requirement mapping (SOC 2, HIPAA, PCI-DSS, etc.)
  - Automatic documentation generation for compliance evidence

Agentic AI and Knowledge Graph

The Role of Agentic AI

- Autonomous Problem-Solving: Agents perceive failures, reason about root causes, and recommend real-time fixes.
- Multi-Agent Collaboration: Specialized agents (DevOps, SRE, etc.) work in concert using domain expertise without human orchestration.
- Continuous Learning: Agents improve over time by analyzing execution patterns and deployment outcomes, creating a smart feedback loop.
- Policy-as-Intent: Developers describe what they want ("secure deployment to prod"); AI understands how through enterprise policies and best practices.

Andrej Karpathy
@karpathy

+1 for "context engineering" over "prompt engineering".

Prompt vs. Context Engineering

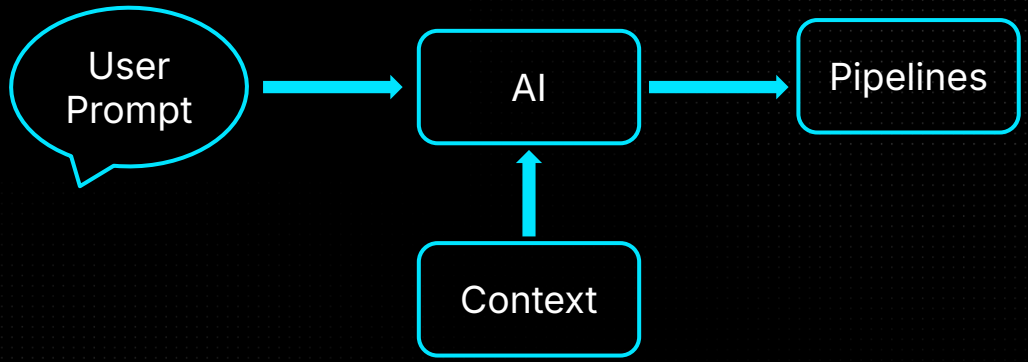
- Prompt engineering (fragile): Quality depends on user skill and specificity
- Context engineering (robust): Quality depends on system knowledge (your policies, templates, infrastructure)

In enterprise CI/CD:

- User says: "Build me a pipeline"
- AI needs rich context: Your governance, your templates, your infrastructure, your team's standards
- Result: Pipelines that are secure, compliant, and aligned - by design, not by luck

This reframes why Knowledge Graph + strong governance matters.
It's your "context" that makes AI reliable at enterprise scale.

Enterprise-Ready CI/CD Pipelines



Scalable
Secure
Compliant

Context: RAG vs Knowledge Graph

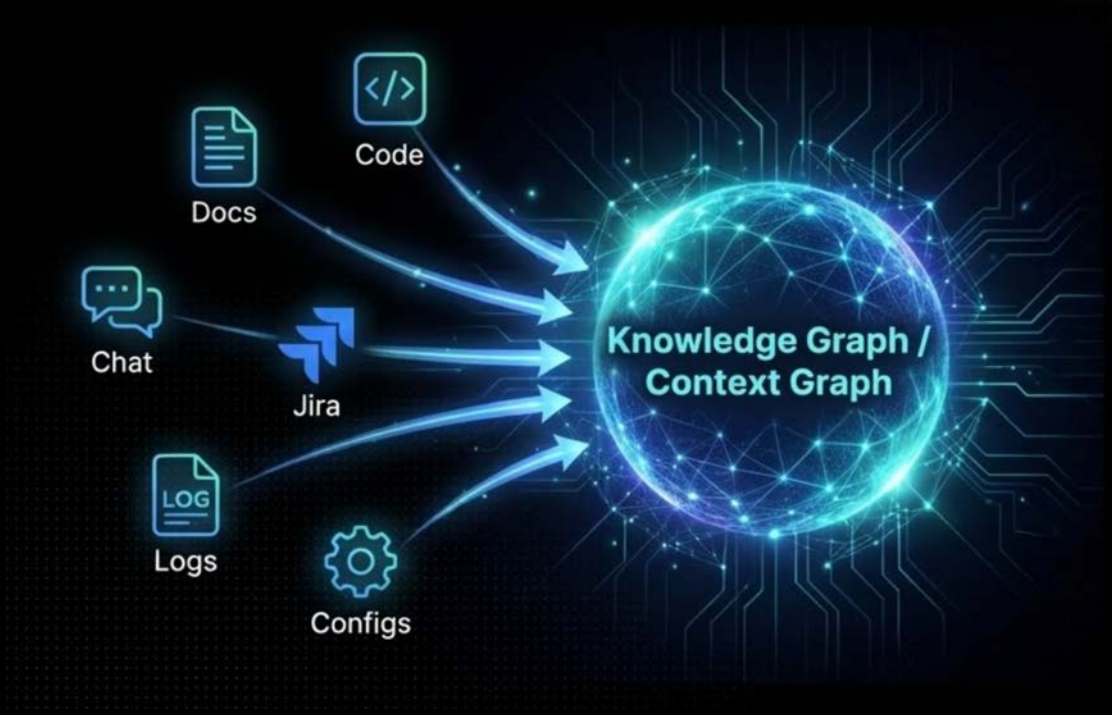
- Knowledge Graph:
  - Use structured data organized as nodes (entities) and edges (relationships) to explicitly represent connections between data points
- RAG:
  - Retrieves unstructured content from external sources feeds this context to a language model to generate answers
- Hybrid:
  - Rich Knowledge Graph using RAG

- For DevOps
  - Relationship-heavy environments
  - For example, an e-commerce company running 100s of microservices in K8s
    - *If I update service A, which downstream services will be affected?*

Preferred approach:
Hybrid (Rich Knowledge Graph + ingest RAG)

Building the Knowledge Graph

harness



Extract entities
(services, users, incidents)

↓

Enrich with metadata
(owners, versions, policies)

↓

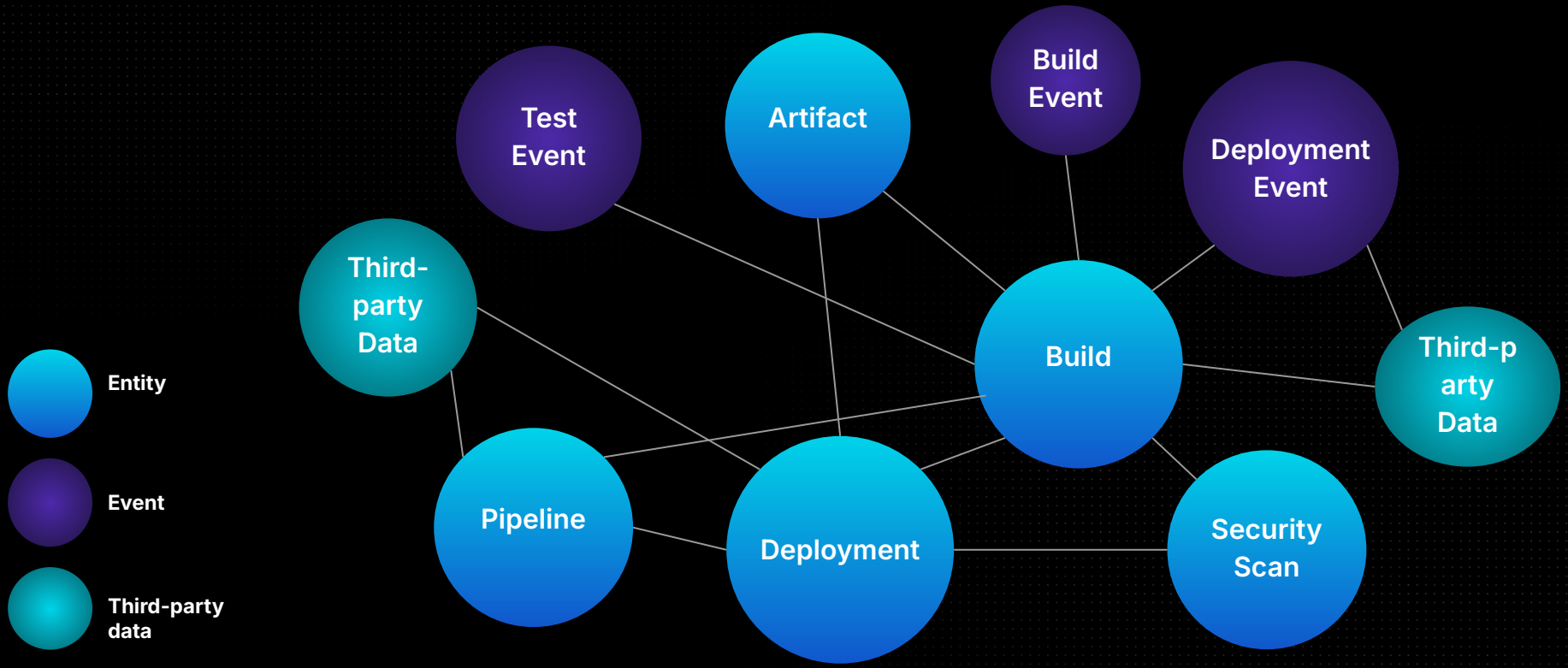
Link relationships
(who owns what, what depends on what)

↓

Expose via Graph API for AI agents to query

If context is the new code, knowledge graph is the compiler

The Knowledge Graph



Governance

Four Pillars of Enterprise AI Governance

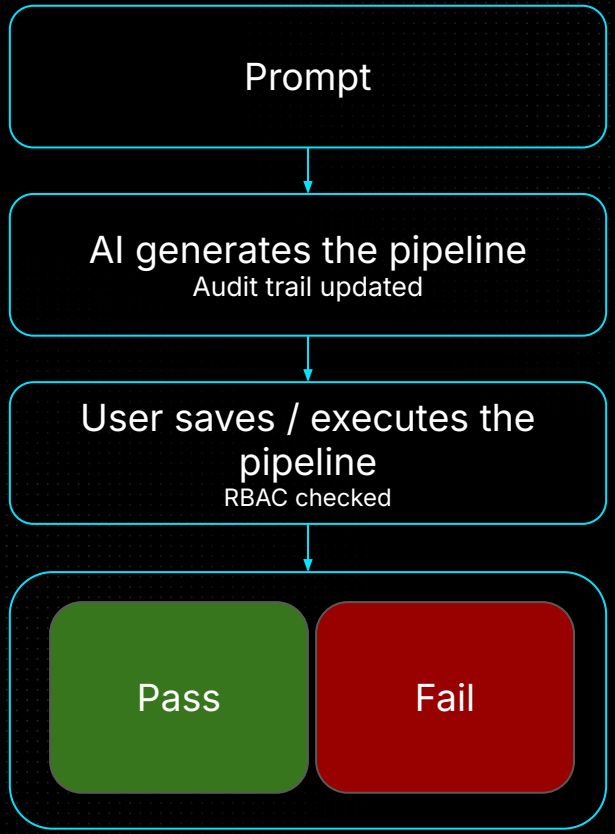
Who	**Access & Identity Governance**	Defines who can do what	Role-Based Access Control (RBAC), model/tool permissions, tenant isolation, contextual authorization, API-scope controls
What	**Data & Privacy Guardrails**	Protects what data is used and where it flows	PII detection/redaction, data residency enforcement (GDPR, DPDP), encryption, prompt masking, context-scope control
How	**Policy Enforcement & Guardrails**	Enforces how AI behaves in real time	Pre/post-model validators, unsafe prompt blocking, content safety filters, grounding verification, output moderation
Why	**Auditability & Compliance**	Provides traceability and accountability	Full audit trails of prompts, responses, model versions, and tool calls; compliance alignment with ISO 42001, NIST RMF, SOC 2; drift & policy reports

Context is the new code. Governance is the new runtime

RBAC-Bounded AI

Scenario	User Can?	AI Can?
Deploy to production	No	No
Create pipelines	Yes	Yes
View secrets	No	No
Modify policies	Only admins	Only admin AI

Enterprise Adoption Framework



The Harness Platform

harness

DevOps
Velocity

- Continuous Delivery
- Continuous Integration
- Infrastructure as Code
- Internal Developer Portal
- Database DevOps NEW
- Artifact Registry NEW
- Cloud Development Environments NEW

Testing
Reliability

- Feature Management & Experimentation
- Chaos Engineering
- AI SRE NEW
- AI Test Automation NEW

AppSec
Security

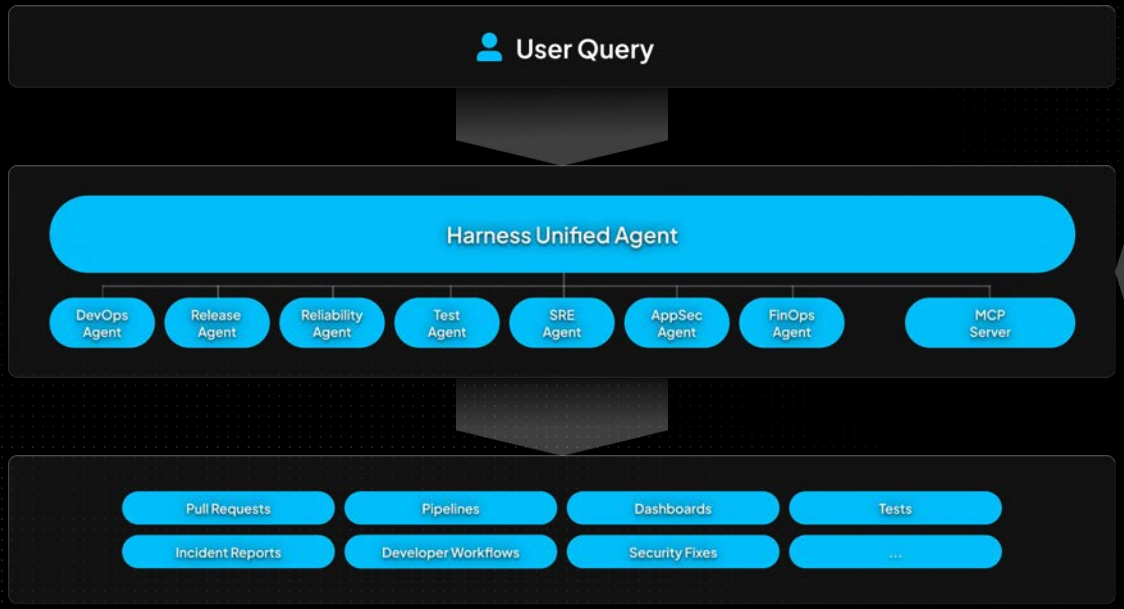
- Security Testing Orchestration
- Supply Chain Security
- App & API Posture Management
- App & API Security Testing
- Application & API Protection

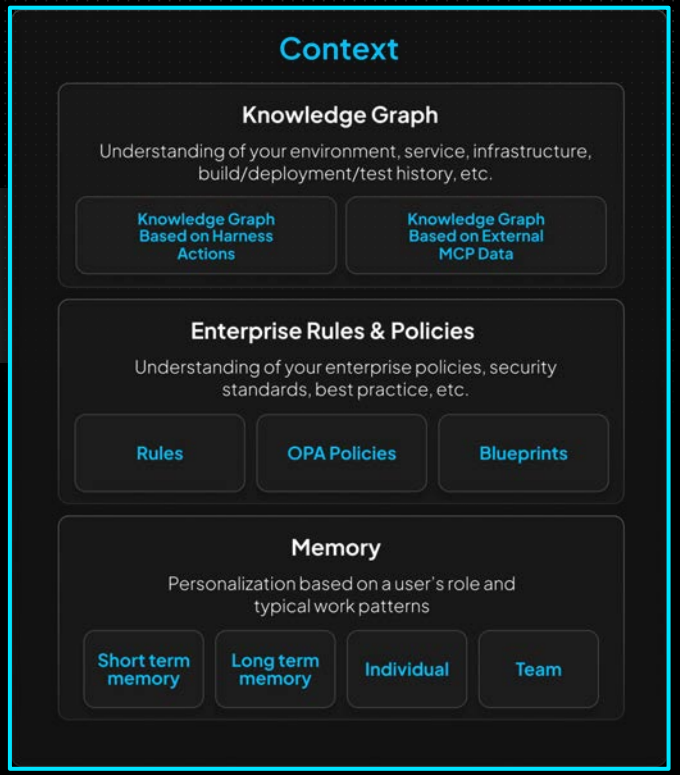
FinOps
Efficiency

- Cloud Cost Management
- Software Engineering Insights

Harness AI

How Harness AI Works?





harness

Thanks!

https://www.linkedin.com/in/chinmay-gaikwad/