WELCOME TO THE 11TH ANNUAL HACKING CONFERENCE

The Institute of Internal Auditors Chicago

Check-In Code: larusso

ISACA Chicago Chapter

Data Wax On/Off : Auditing AI in the Lakehouse

Velu Natarajan | Krishnakumar Kunka Mohanram

IIA/ISACA Chicago - 11th Annual Hacking Conference

About us

Velu Natarajan

- Data & FinOps Practitioner
- Data Clouds Architect
- Co-Author and Technical Reviewer of Technical Books
- 20+ Years of Experience in Data and Cloud Technologies
- Focused on AI-ready data platforms, multi-cloud interoperability, cost, governance, and security

Krishnakumar

- Cloud and data architecture specialist
- Public speaker
- Oracle ACE Pro
- Co-Author and Technical reviewer of books
- 20+ years in enterprise data and cloud technologies
- Specialized in catalog-driven governance, AI-ready lakehouse design, and secure multi-engine data access across cloud platforms.

Disclaimer

This presentation is based on our own perspectives and professional interests. It does not reflect the official views, positions, or interests of our company.

What We'll Cover Today

01 What is a Lakehouse & Its Evolution

02 Threat Landscape

03 Core Security Domains

04 Lakehouse Attack Surfaces

05 Design Principles to Reduce Exposure

06 Agentic AI: The New Frontier

What is a Lakehouse & Its Evolution

From siloed warehouses to unified, AI-ready data platforms

1990s – 2000s

Data Warehouse

Structured, governed,
but siloed and costly

2010s

Data Lake

Cheap storage, schema-
on-read, but ungoverned

2020s – Today

Lakehouse

Open formats + ACID +
governance + AI-ready

Open Table Formats

Apache Iceberg, Delta Lake, Apache Hudi.
ACID transactions, time travel, schema evolution on object storage.

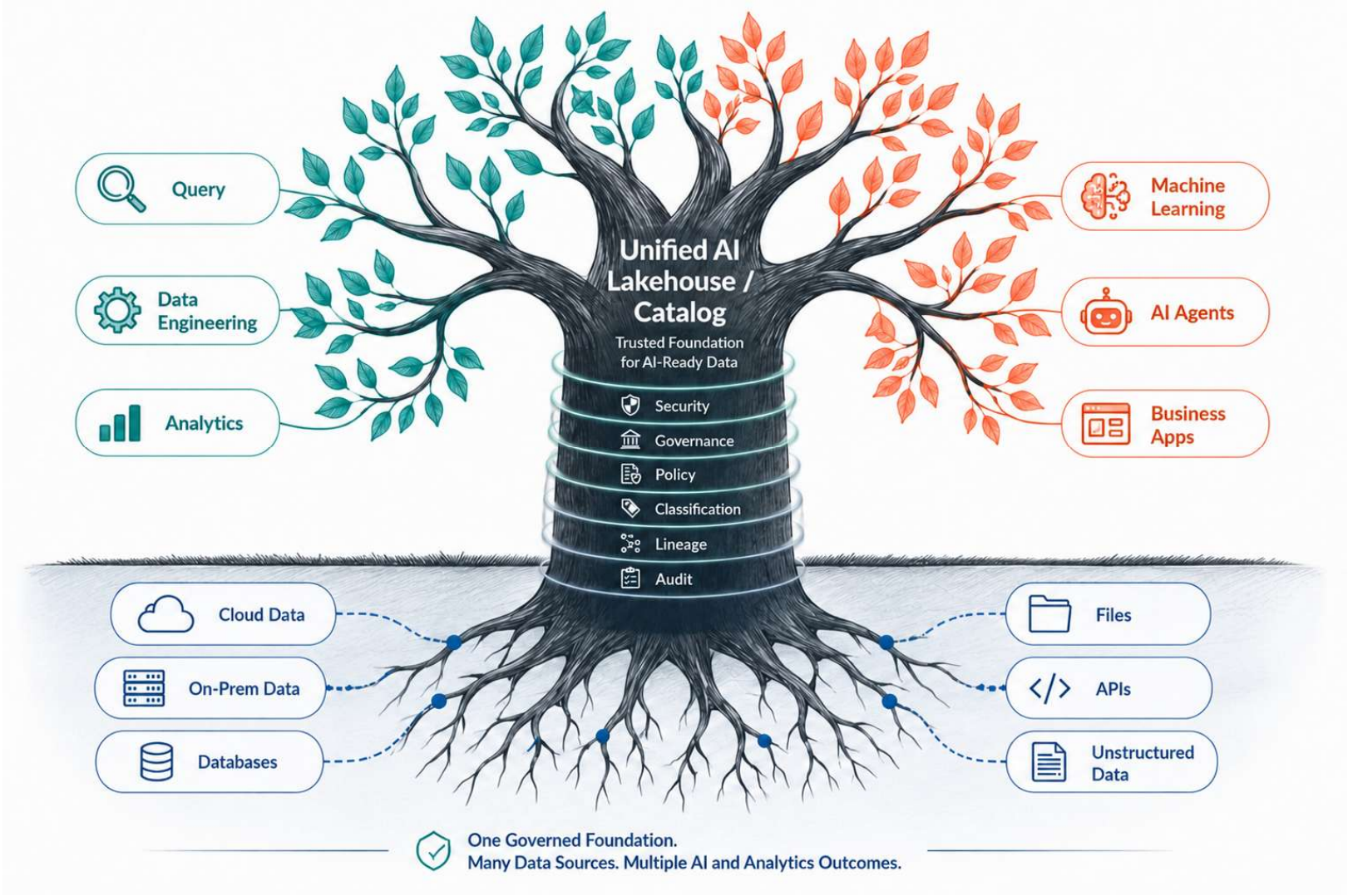
Unified Governance

Single catalog and policy plane across BI, ML, and AI agents.
Lineage, masking, and access control travel with the data.

AI & Agent Ready

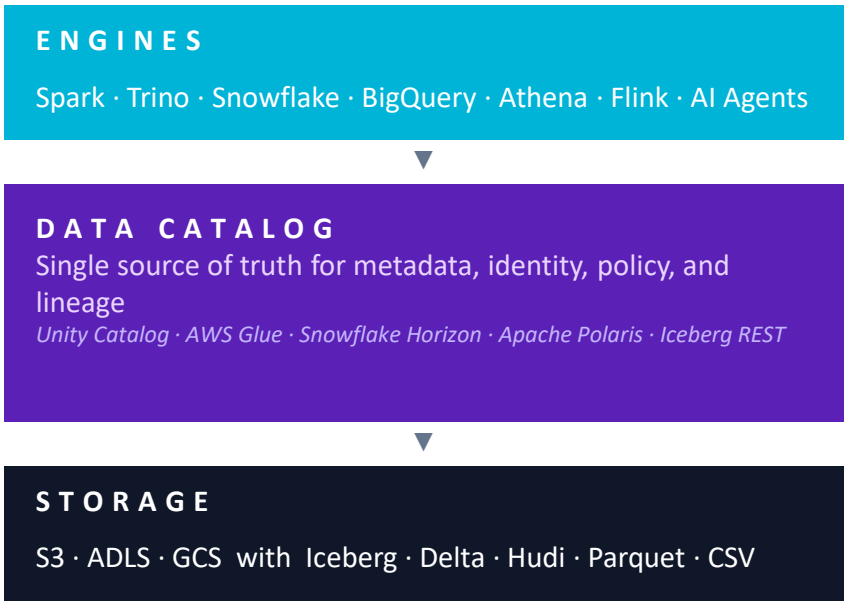
Vector search, semantic models, and tool-callable APIs.
One platform serves analysts, ML pipelines, and agentic workloads.

What is a Lakehouse & Its Evolution



Data Catalog

The metadata-and-policy plane that sits between storage and every engine that queries it



What a catalog does

1 Registers
Every table, view, and file — managed, external, and federated — in a single namespace.

2 Authorizes
Identity-aware access at table, row, and column level — enforced before the query runs.

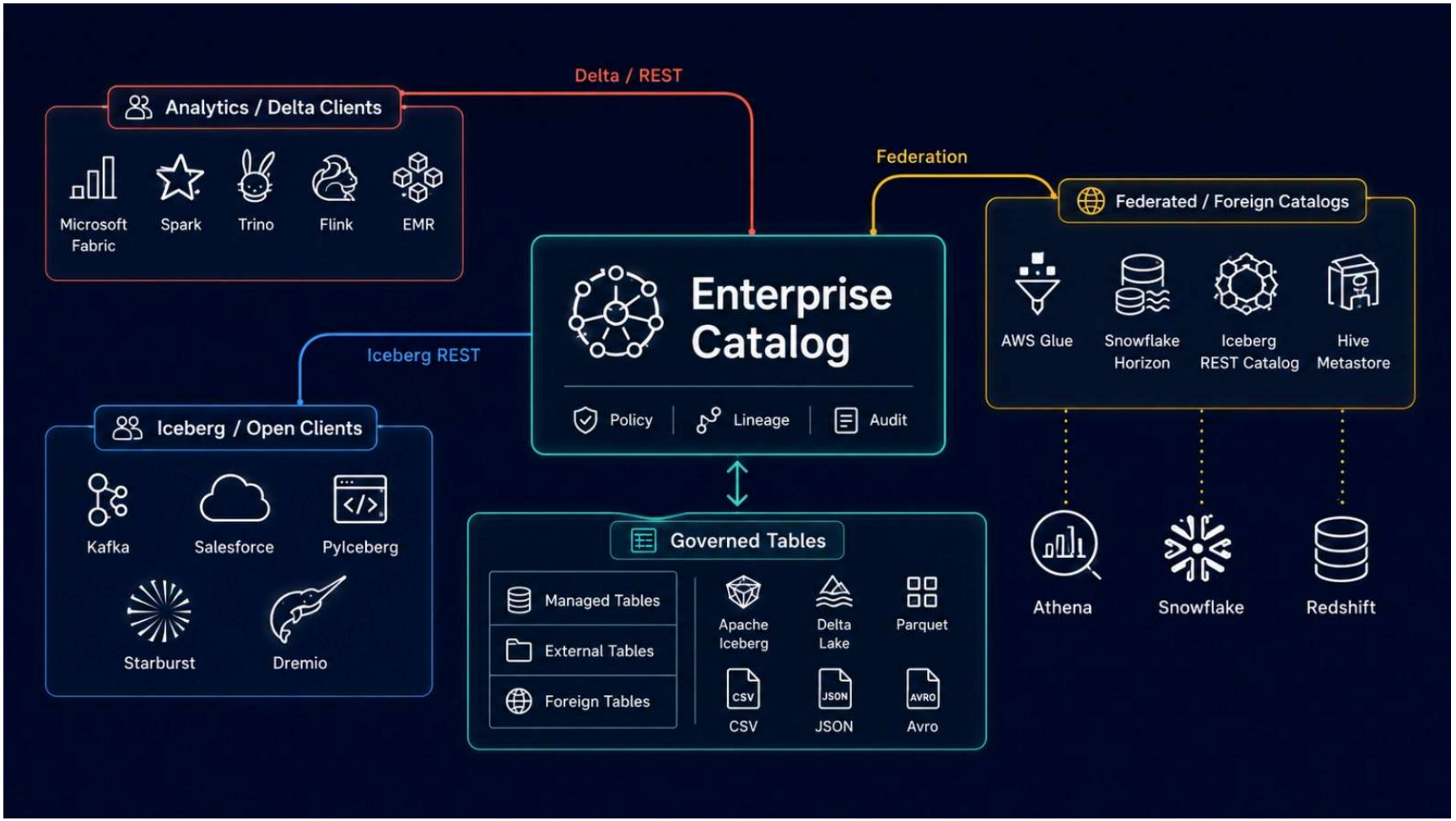
3 Audits
Every access decision, schema change, and lineage hop captured as immutable evidence.

4 Federates
Exposes foreign catalogs (Snowflake, Glue, HMS) under one governance plane — no copy.

The catalog is the only place where policy can be enforced consistently across every engine — and audited as a single source of truth.

How the Catalog Enables Unbundled Multi-Engine Compute

One catalog. Many engines. Same governance, same audit trail.



Audience Poll

1. What is the biggest shift the lakehouse introduces compared to traditional data platforms?

A. Cheaper storage at scale

B. Combining BI, ML, and AI into one governed platform

C. Eliminating the need for data governance

D. Replacing all existing data warehouses

Audience Poll

1. What is the biggest shift the lakehouse introduces compared to traditional data platforms?

A. Cheaper storage at scale

B. Combining BI, ML, and AI into one governed platform

C. Eliminating the need for data governance

D. Replacing all existing data warehouses

Analytical Data Stores

Open, unbundled, agent-aware — the analytics stack is being rewritten

Open Table Format Interoperability

Apache Iceberg drives vendor-neutral storage. The format war is over; interoperability is pervasive.

Unbundled Multi-Engine Compute

Workload-specific engines optimize independently for cost and performance. Storage/compute separation is the default, not the exception.

System of Context > System of Record

Hybrid multi-cloud serving model training, inference, and stateful agents. The question isn't where data lives — it's what context it provides.

Federation & Marketplace Sharing

Faster network interconnects, no egress fees, and open standards driving sharing of data products with security and governance.

Context — not record-keeping — is the new center of gravity for analytical platforms.

A SHIFT IN WORKLOADS

Agentic Workloads — the New Tenant in Your Lakehouse

Same data, same platform — but a tenant that doesn't sleep, doesn't tire, and doesn't ask permission the way a human would.

BI & Dashboards | ETL Pipelines | ML Training | Ad-Hoc Analytics + **Agentic AI**

What are AI Systems?

The workloads your lakehouse must now serve — and audit

AI Systems

Software or services relying on AI techniques — generative AI, agentic AI, machine learning, large language models — to generate outputs, make decisions, or perform tasks. Includes third-party software with embedded AI, even when AI is not the primary purpose.

Audit lens: any system that produces a decision a human would otherwise have made.

Generative AI

Models that produce new content — text, code, summaries, images — from a prompt. Examples: ChatGPT, Claude, Gemini, Copilot.

Audit lens: outputs are non-deterministic. Same prompt may produce different answers — audit must capture both.

Machine Learning

Systems that learn from data to make predictions or classifications without being explicitly programmed. Examples: fraud scoring, churn models, recommendation engines.

Audit lens: training data lineage and model drift are first-class controls.

Agentic AI

AI that acts autonomously to achieve multi-step goals — planning, calling tools, querying data, taking actions. Moves beyond chat into operations.

Audit lens: every tool call, query, and decision must be attributable, scoped, and reversible.

01

Threat Landscape

The evidence is no longer theoretical.

The Cost of Getting It Wrong

$4.88M

Average cost of a data breach (IBM, 2024)

83%

Of breaches involve cloud or data platform assets (Verizon DBIR, 2024)

68%

Of data lake incidents caused by misconfiguration (Gartner, 2024)

194 days

Average time to identify a breach (IBM, 2024)

Sources: IBM Cost of a Data Breach 2024 | Verizon DBIR 2024 | Gartner Cloud Security Report 2024

02

Core Security Domains

Six layers. All must hold.

The Six Pillars of Lakehouse Security

1 Identity & Access

Fine-grained RBAC, column/row-level security, least privilege, service account lifecycle

2 Data Encryption

Encryption at rest & in transit, KMS, tokenization, PII/PHI masking

3 Audit & Observability

Query audit logs, anomaly detection, SIEM integration, access pattern baselining

4 Data Governance

Classification, metadata tagging, lineage tracking, policy enforcement across zones

5 Network Security

Private endpoints, VNet integration, IP allowlisting, ZTNA

6 Regulatory Compliance

GDPR, HIPAA, SOC 2, PCI-DSS — mapped across ingestion, storage, and query layers

Audience Poll

2. ***What is the most likely cause of a data breach in modern lakehouse environments?***

A. Nation-state cyberattacks

B. Insider fraud

C. Misconfiguration of cloud/data platforms

D. Hardware failures

Audience Poll

2. *What is the most likely cause of a data breach in modern lakehouse environments?*

A. Nation-state cyberattacks

B. Insider fraud

C. Misconfiguration of cloud/data platforms

D. Hardware failures

03

Lakehouse Attack Surfaces

Every layer is a door. Which ones are unlocked?

What Risk Does Agentic Analytics Add?

Agents aren't just users with API keys — they change the risk profile of every layer

Acting on behalf of users

Agents inherit user permissions but may access more than the user expected — especially via implicit context.

Cross-source aggregation

Agents query, combine, and summarize sensitive data across systems — surfacing what was previously siloed.

Prompt injection via data

Malicious content in retrieved documents or rows can manipulate agent behavior at runtime.

Over-permissioned scope

Service accounts granted broad SELECT can create large-scale exposure across thousands of queries.

Runaway query loops

Recursive behavior inflates cost and obscures audit signals — making suspicious access patterns harder to spot.

Full-stack auditability required

Every prompt, query, tool call, policy decision, and output must be captured — not just SQL.

Agentic AI in the Lakehouse — A New Threat Model

THREATS

Prompt injection via data
Malicious content in a database record instructs the agent to exfiltrate data

Hallucination & ungrounded output
Agent invents facts when context is missing — inconsistent metadata and poor data quality amplify the risk

Credential escalation
Agent reuses service account credentials beyond intended scope

Runaway query loops
Recursive agent behavior generates millions of queries — billing shock and obscured audit

Shadow data access
Agent discovers undeclared tables through catalog browsing

MITIGATIONS

- ✓ Agent identity isolated from human identities
- ✓ Tool access allow-listed per agent persona
- ✓ Query rate limits and circuit breakers on agents
- ✓ Agent output reviewed before leaving the platform
- ✓ Audit tags every query with agent_id, session_id
- ✓ GraphRAG grounds answers in approved documents + graph context
- ✓ Semantic layer enforces business-meaning scope, not just table access

04

Design Principles to Limit Exposure

Wax on: build the right controls. Wax off: strip what doesn't belong.

1. Unified Governance

Governance travels with the data — not behind it

One control plane for all personas — including AI agents

- Centralized data retention policies
- Unified policy engine across compute clusters
- Policies enforced at catalog, not just at query time
- Single audit trail for every policy decision
- Same controls apply to humans and agents
- Enterprise test: governance must span >1 application, >1 source system, and business stakeholder vocabulary

Data Masking

PII auto-masked at column level. Snowflake Dynamic Data Masking — SSN → XXX-XX-1234 for non-privileged roles.

Access Control

RBAC + ABAC enforced at catalog layer. Apache Ranger — analyst_tier1 blocked from finance.transactions.

Row-Level Filtering

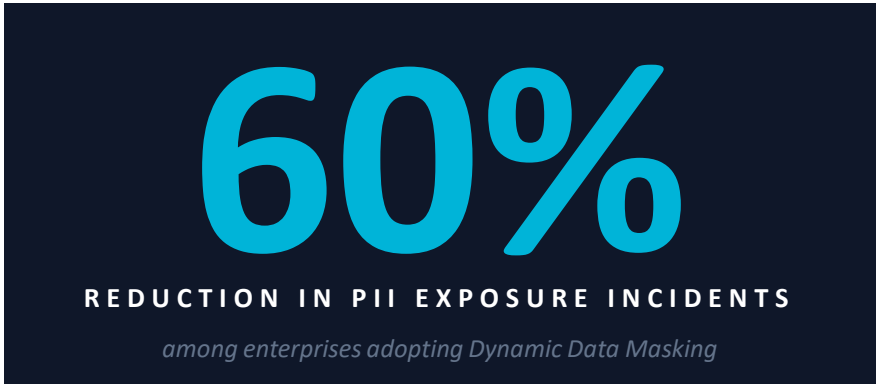
Identity-aware row scope. Delta Lake — WHERE region = current_user_region() injected transparently.

Model Registry

AI agents may only invoke models registered in the catalog with steward sign-off — Unity Catalog MLflow, SageMaker Model Registry.

Real-World Impact: Dynamic Data Masking

Evidence that unified governance pays back in measurable risk reduction



What changed

- Snowflake Dynamic Data Masking went GA in 2022 — column-level masking enforced at query time.
- Policies attach to columns, not to roles or queries — every query path is automatically protected.
- Analysts and AI agents see tokenized values; only privileged roles see raw PII.
- Audit log captures every masking decision — supports COBIT APO13.01 evidence requirements.

Auditor's takeaway

Governance controls applied at the catalog layer — not at the query layer — produce measurable, auditable risk reduction. This is the proof point for unified governance over reactive policy enforcement.

Source: Snowflake State of Data Product Security, 2024

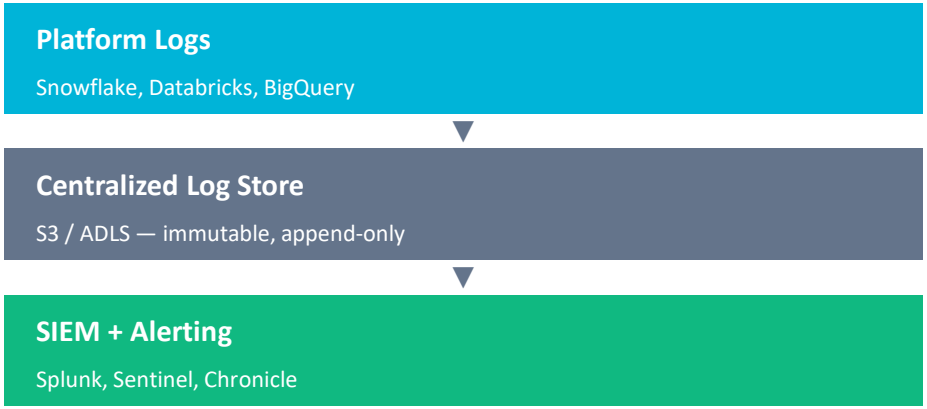
2. Audit & Observability

Every user action is evidence — collect it all, act on what matters

What to Capture

- Who accessed what data, when, from where
- What queries were run (full SQL capture)
- What data was returned (row/column counts)
- What transformations were applied
- Agent identity vs. human identity — separated
- **Semantic context: business term, glossary definition, ontology relationship, policy that guided the answer**

Unified Audit Trail Architecture

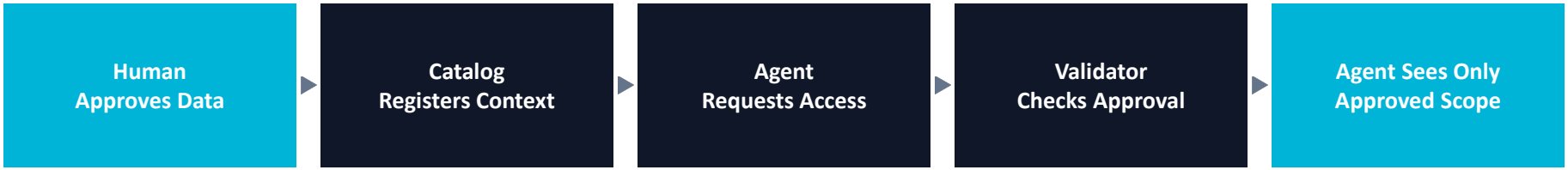


⚠ **ALERT**

Agent 'etl_agent_prod' queried customer PII table 3,200 times in 4 minutes — baseline is 40/hour

3. Context Management: Agents Need Data and Context

Context management is the guardrail between AI agents and your data



Only human-approved datasets and models reach agentic workloads. Context enforces this at runtime — not at design time.

Context and meaning embedded with source data make common understanding machine-readable — the foundation of an enterprise semantic layer.

Approved Model Registry

AI agents may only call models registered in the enterprise catalog (e.g., Unity Catalog MLflow models with data steward sign-off).

Data Contract Enforcement

Agentic workload declares its data contract at instantiation. Context manager validates against approved contracts before granting table access.

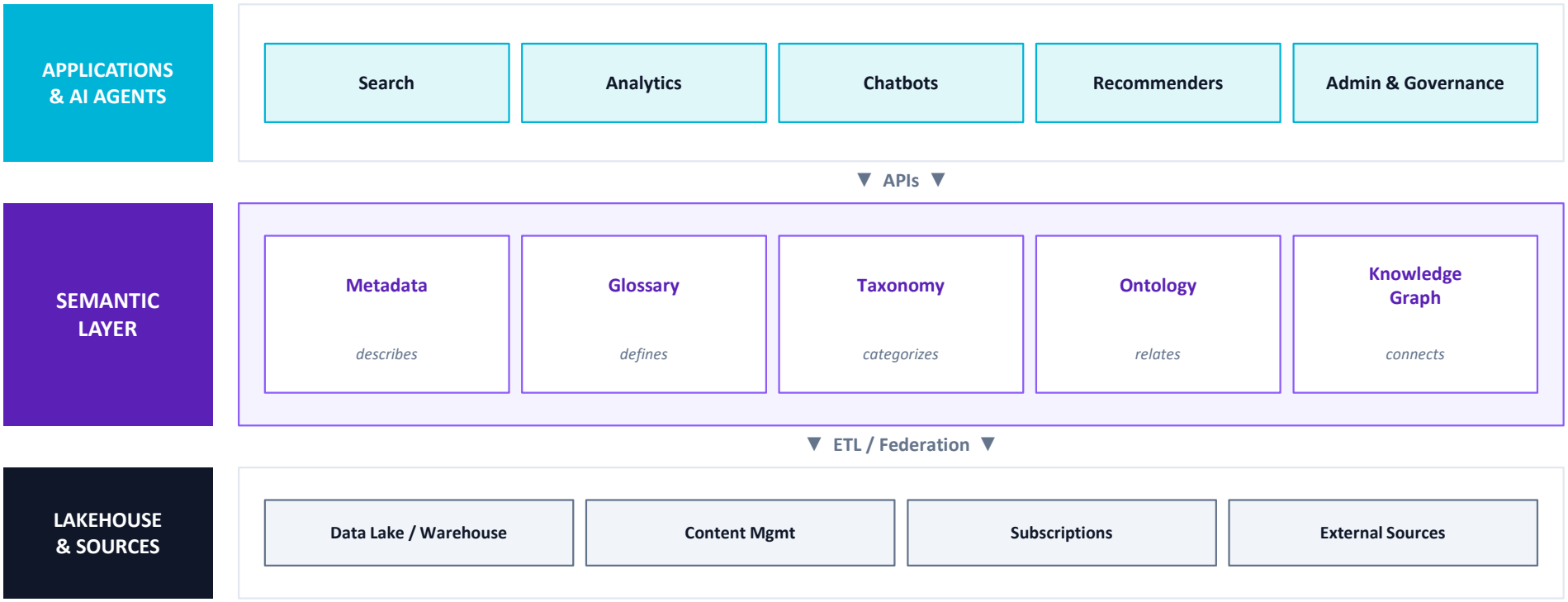
Same Data, Many Consumers

All three query customer #12345 → identical SSN → XXX-XX-1234, same row filters, same approved fields.

Context — not record-keeping — is the new center of gravity for analytical platforms.

Semantic Layer as an AI Control Plane

The bridge between governed lakehouse data and trustworthy agentic AI



4. Synthetic Data

Accelerate POCs without exposing production PII

Why use synthetic data

- Statistically representative for AI model training
- Eliminates PII/PHI exposure during proof-of-concept
- Enables realistic load and pen-testing without regulatory risk
- Self-serve provisioning. No Data Loss Prevention (DLP) approval bottleneck
- Trap records embedded for breach detection

	Production Data	Synthetic Data
Regulatory risk	**HIGH**	**NONE**
AI training fidelity	100%	85–95%
POC speed	Slow (approvals)	**Fast (self-serve)**

Trap Record Technique

Embed a known synthetic customer record. If it appears in a report or model output outside controlled tests, you have confirmed data leakage.

Audience Poll

3. ***What is the biggest new risk introduced by agentic AI in data platforms?***

A. Faster query performance

B. Higher infrastructure costs

C. Autonomous, large-scale data access beyond human patterns

D. Reduced data availability

Audience Poll

3. *What is the biggest new risk introduced by agentic AI in data platforms?*

A. Faster query performance

B. Higher infrastructure costs

C. Autonomous, large-scale data access beyond human patterns

D. Reduced data availability

Key Takeaways

Where Attackers Get In

HIGH

Object Storage Layer (S3 / ADLS / GCS)

Bucket misconfig, public exposure, cross-account access, insecure presigned URLs

HIGH

Table & Catalog Layer (Iceberg / Delta / Hudi)

Metadata tampering, time-travel abuse, snapshot poisoning, catalog spoofing

MED

Query & Compute Layer

SQL injection via dynamic queries, compute hijacking, cross-workspace exfiltration

HIGH

Ingestion Pipelines (CDC / Kafka / Debezium)

Stream tampering, malicious payloads, insecure API connectors, schema drift attacks

HIGH

Governance Gaps → Maps to Unified Governance

Unclassified PII, missing data contracts, shadow data, role explosion, orphaned service accounts

HIGH

Audit Blind Spots → Maps to Audit & Observability

Log tampering, missing agent baselines, retention gaps, unsigned events, siloed SIEM coverage

Closing Thoughts

1. **Governance must travel with data**

   Not stop at the warehouse door

2. **Audit everything — baseline AI agents separately**

   Humans and agents have different normal

3. **Context boundaries are your first AI control**

   Human approval gates prevent scope creep

4. **Synthetic data is your safe sandbox**

   Stop using production PII for testing

5. **Semantic context is now part of the audit boundary**

   Audit definitions and approved meaning — not just identities and logs

Questions?

Let's build lakehouses that can take a hit.