

Introduction to Machine Learning Governance



- Policies, processes, controls, and tools managing models across the full lifecycle



- From data and training through deployment, monitoring, and retirement



- Reduces risk, ensures regulatory readiness, and builds trust



- MLOps ships models; governance keeps them defensible and auditable



Core Concepts, Stakeholders, and the ML Lifecycle



- Governance as the control layer keeping models compliant, fair, and auditable



- Key terms: accountability, transparency, fairness, auditability, reproducibility, traceability



- Touchpoints span problem framing, data, training, deployment, monitoring, and retirement



- Role owners: product owners, ML engineers, data stewards, risk and compliance



- Governance is a continuous loop, not a one-time release gate



Regulatory Landscape: EU AI Act and Current Deadlines

- Prohibited practices, high-risk systems, transparency duties, GPAI models
- Transparency rules enforce from August 2026; watermarking by December 2026
- Annex III high-risk obligations apply from December 2027
- Annex I high-risk obligations apply from August 2028



Regulatory Landscape: US Rules and Cross-Border Implications



- No comprehensive federal AI statute; state laws lead



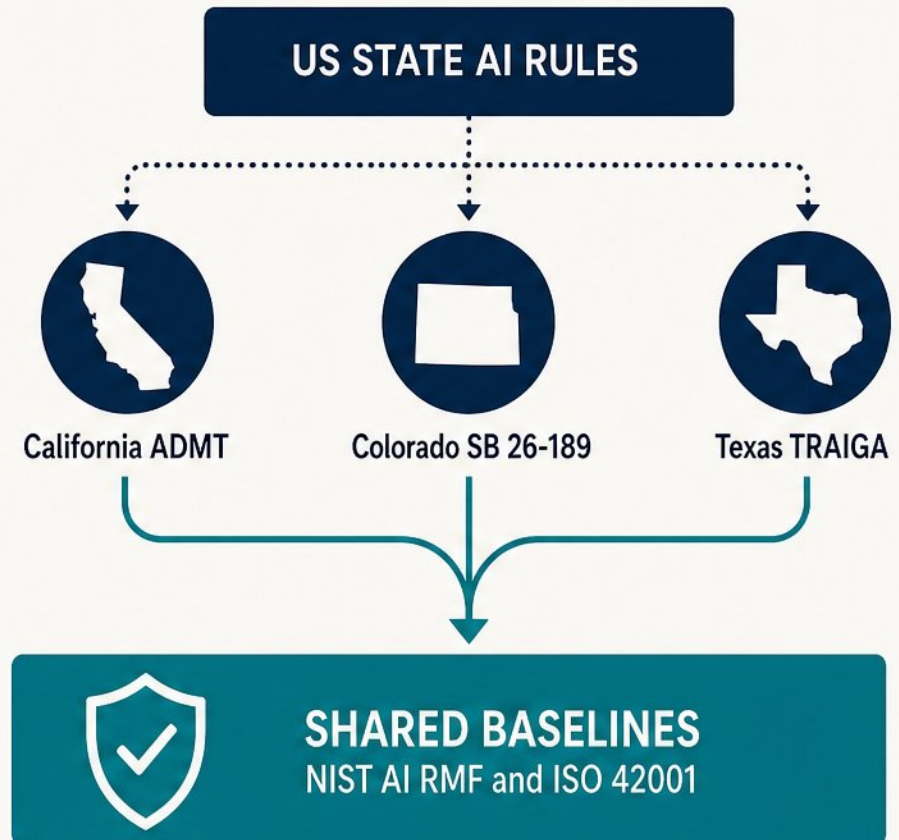
- California ADMT, Colorado SB 26-189, Texas TRAIGA active



- Title VII, ECOA, FCRA still govern algorithmic decisions

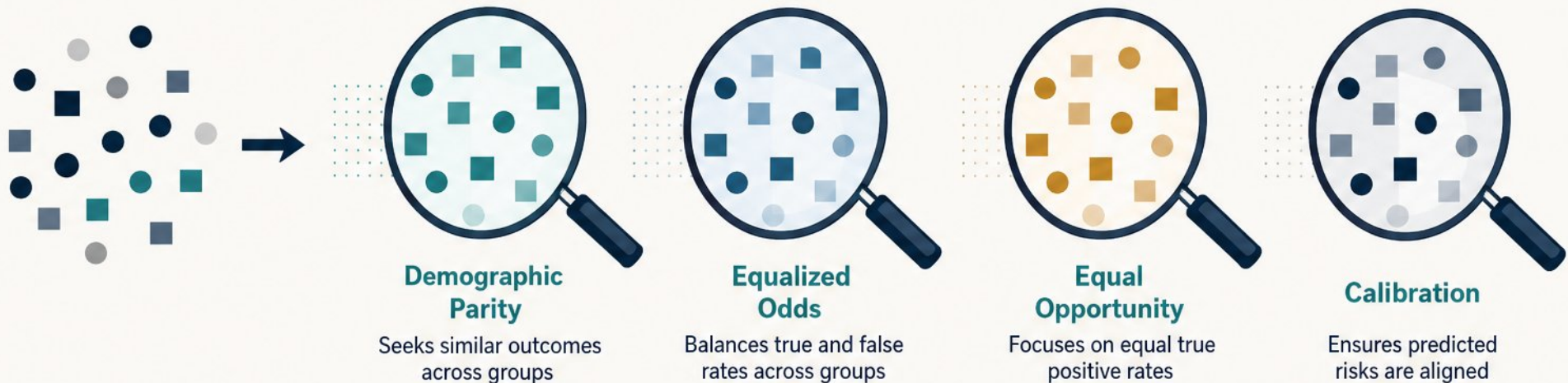


- NIST AI RMF and ISO 42001 serve as portable baselines



Fairness, Bias, and Ethical Review Gates

One dataset. Different fairness lenses.



Bias sources: historical data, labeling, proxy variables, sampling gaps, misuse



Fairness metrics: demographic parity, equalized odds, equal opportunity, calibration



Mitigation: preprocessing, in-processing, postprocessing, plus production monitoring



Ethical gates: system cards, datasheets, decision logs for repeatable fairness



Model Risk Management



- Model risk tiers scale governance based on impact and context



- Key risks: drift, performance decay, data quality, feedback loops



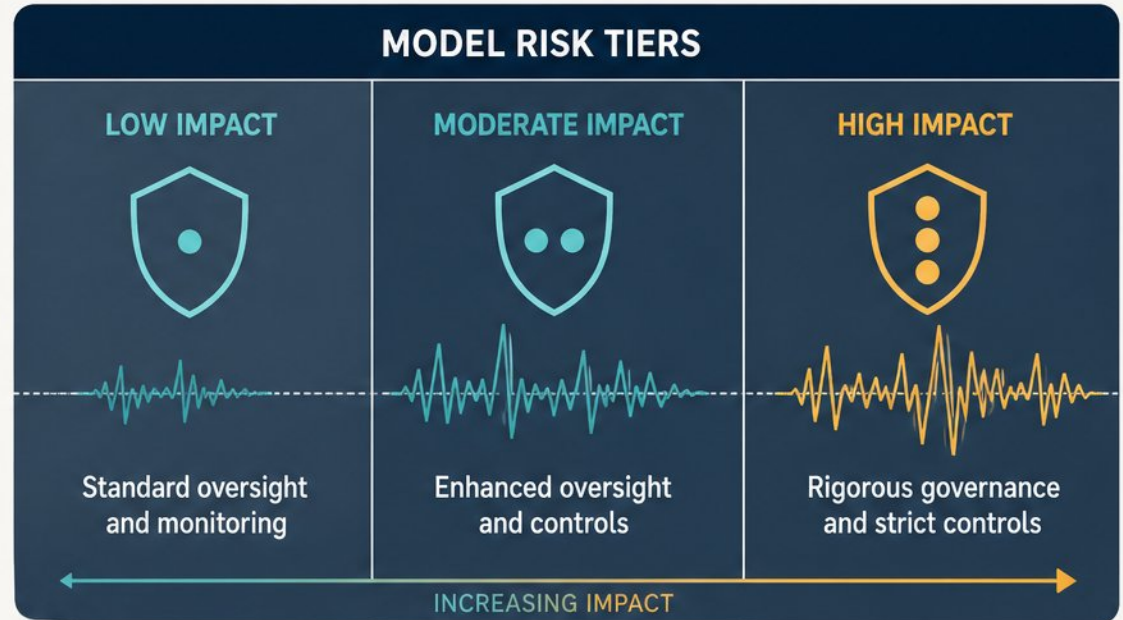
- Monitor performance, data drift, operational health, business outcomes



- Every alert needs a runbook and named owner



- Incident response includes rollback path and auto-recovery triggers



INCIDENT RESPONSE



Runbook Activated



Rollback Path



Auto-Recovery Triggers

Reproducibility, Lineage, and Audit Evidence



- Code alone is insufficient: track data version, parameters, dependencies, environment, and hardware.



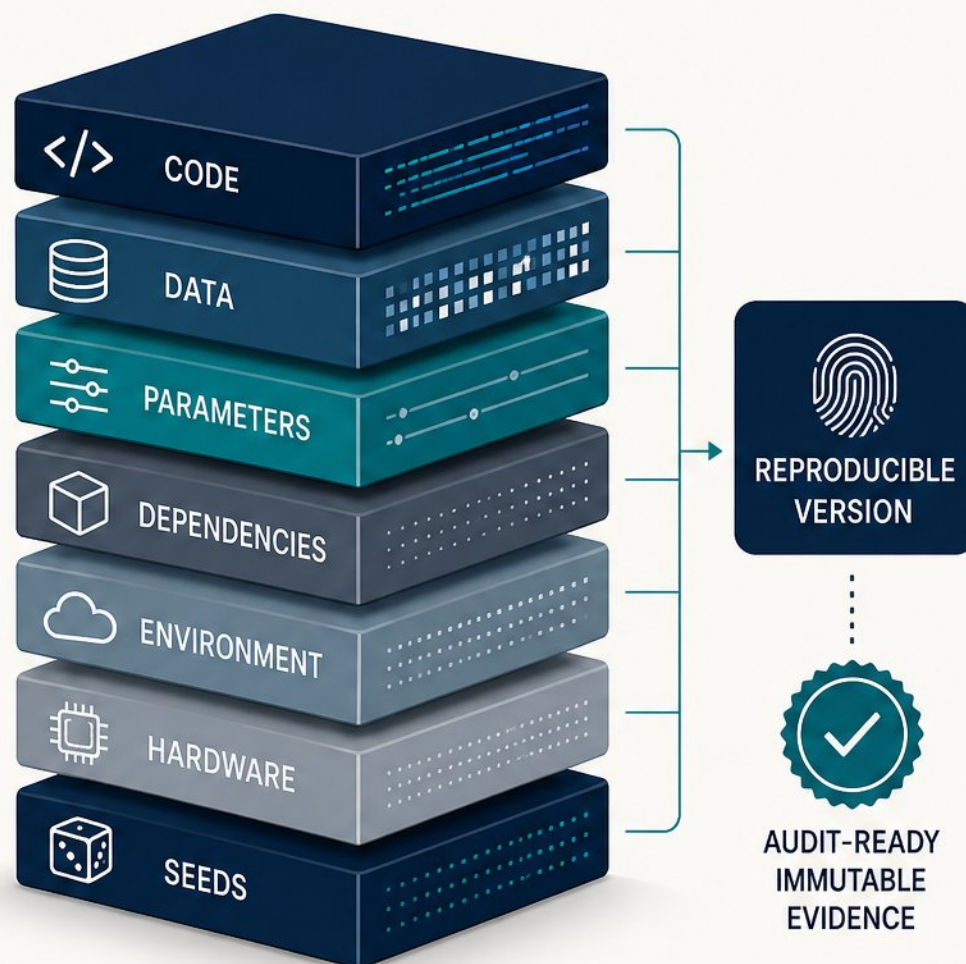
- Build audit-ready artifacts: model cards, datasheets, signed evidence, and immutable audit trails.



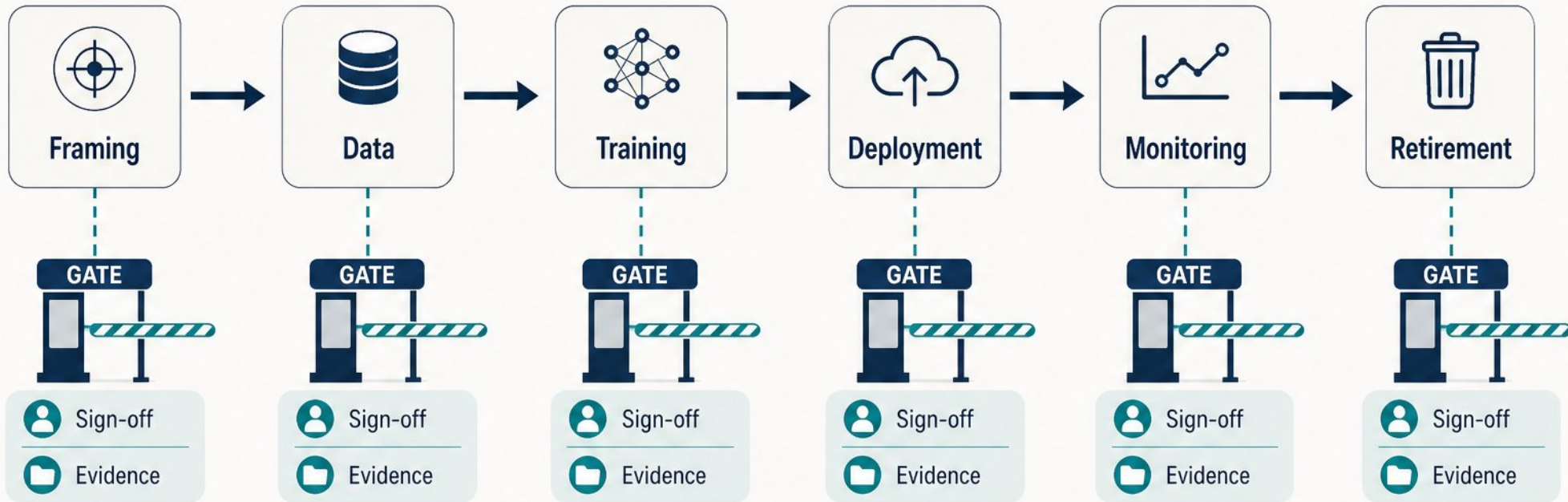
- Use lineage tools like MLflow, DVC, and Weights & Biases for queryable experiment history.



- Good lineage turns a production incident into a database query, not a forensic investigation.



Governance Gates Across the ML Lifecycle



- Gates at framing, data, training, deployment, monitoring, retirement
- Define sign-off roles and evidence for each stage
- Pre-determined updates avoid breaking auditability
- Substantial modifications trigger re-assessment
- Minimum viable governance: inventory, tiering, hard gates

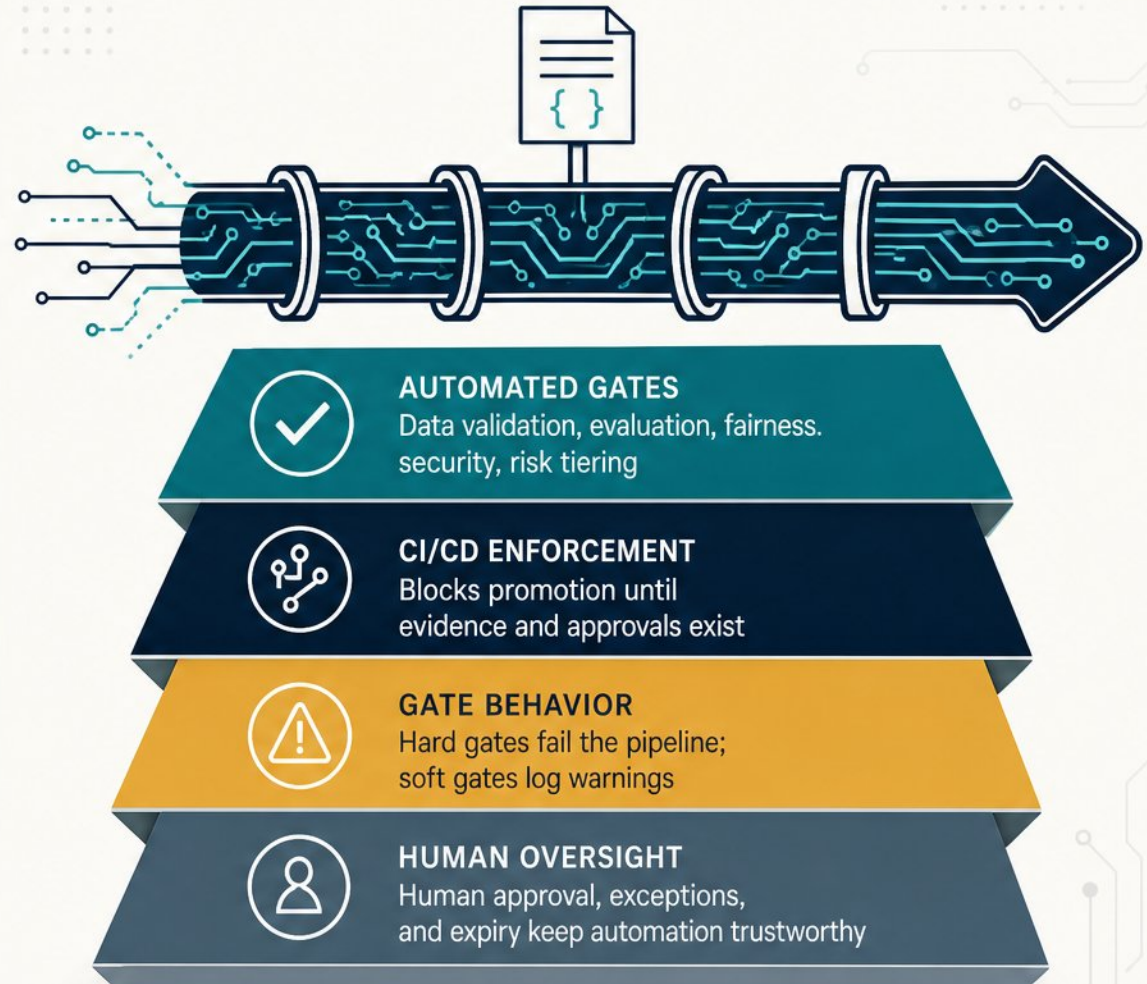


Governance Automation and Policy-as-Code

- Make compliance executable inside pipelines, not paper checklists
- Automated gates: data validation, evaluation, fairness, security, risk tiering
- CI/CD blocks promotion until evidence and approvals exist
- Hard gates fail the pipeline; soft gates log warnings
- Human approval, exceptions, and expiry keep automation trustworthy

POLICY-AS-CODE

Governance made executable



Model Registries, Deployment Strategies, and Rollback



- Registry is the canonical record of versions, owners, and lineage



- Shadow, canary, blue-green: validate risk before full traffic



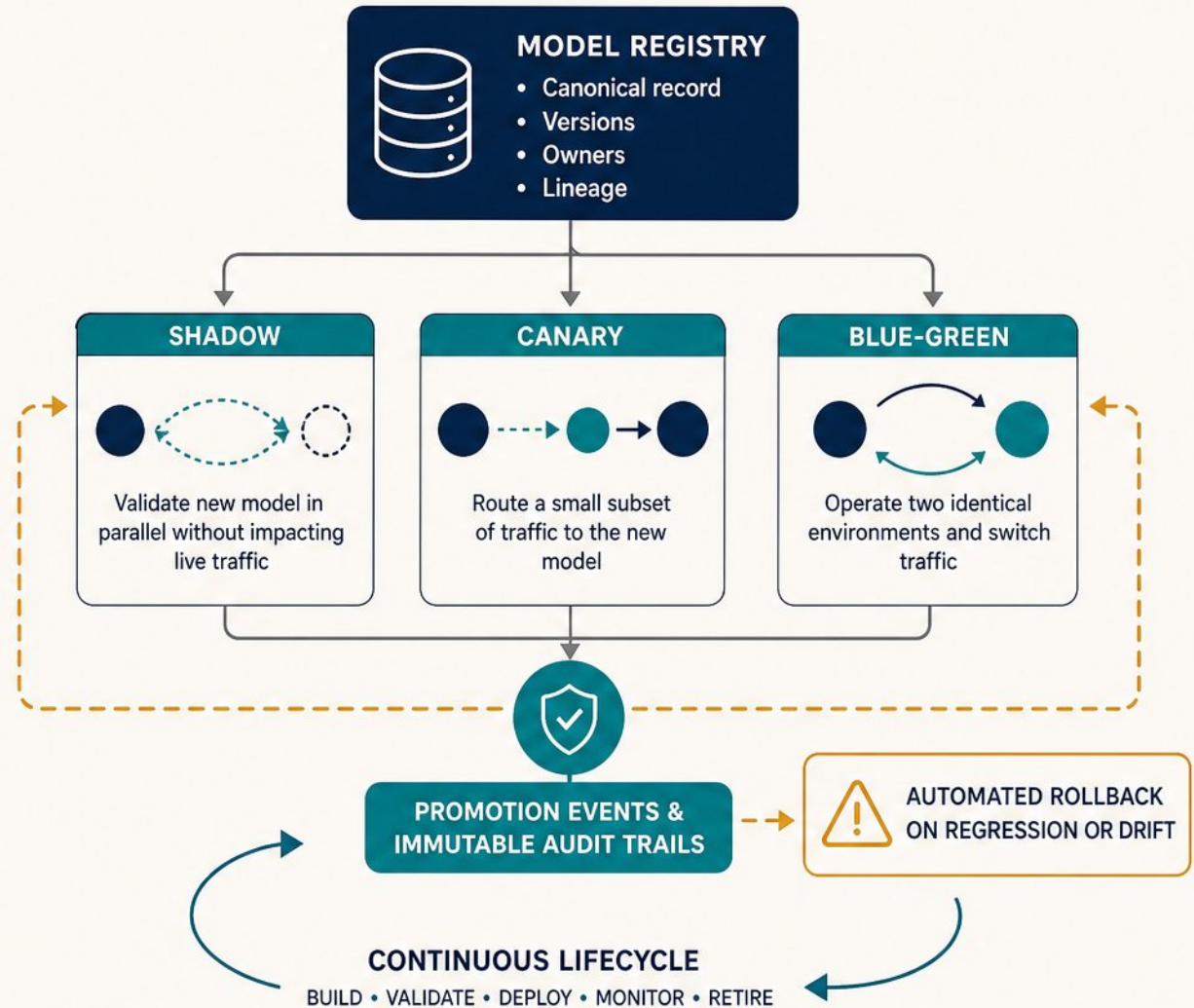
- Promotion events and immutable audit trails prove every deployment



- Automate rollback triggers on metric regression or drift

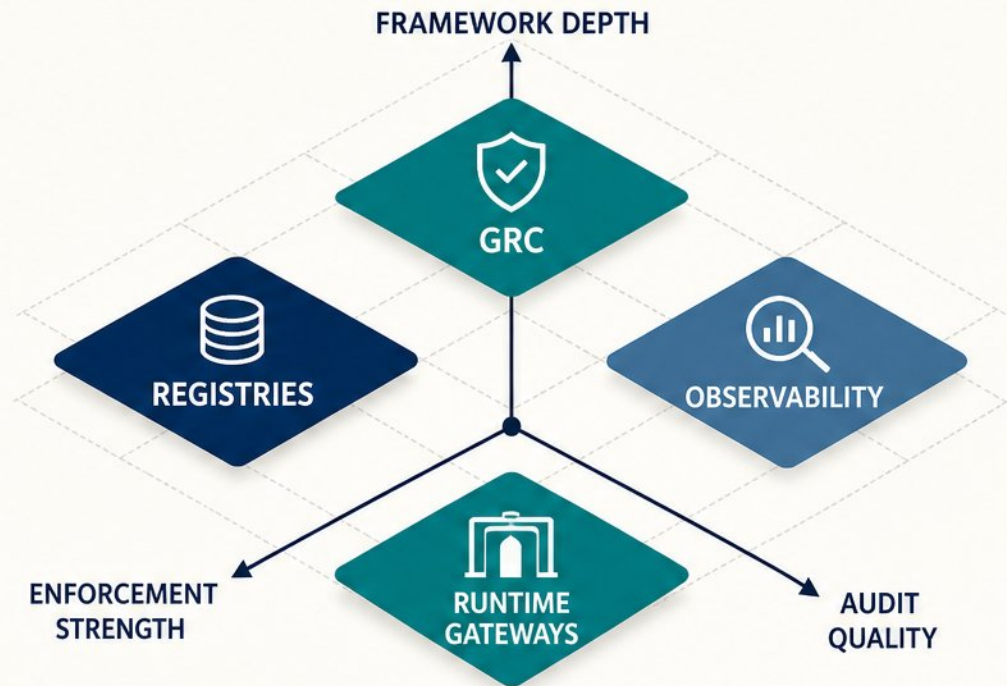


- Retirement requires deprecation targets, credential revocation, and archiving



Tools and Technology Landscape

- Four categories: GRC, registries, runtime gateways, and observability
- GRC leaders: Credo AI, Holistic AI, IBM watsonx.governance, OneTrust
- Lifecycle and enforcement: MLflow, Fiddler AI, Arthur AI, gateways
- No single tool covers all layers: most teams combine categories
- Evaluate framework depth, enforcement, audit trails, and integration cost

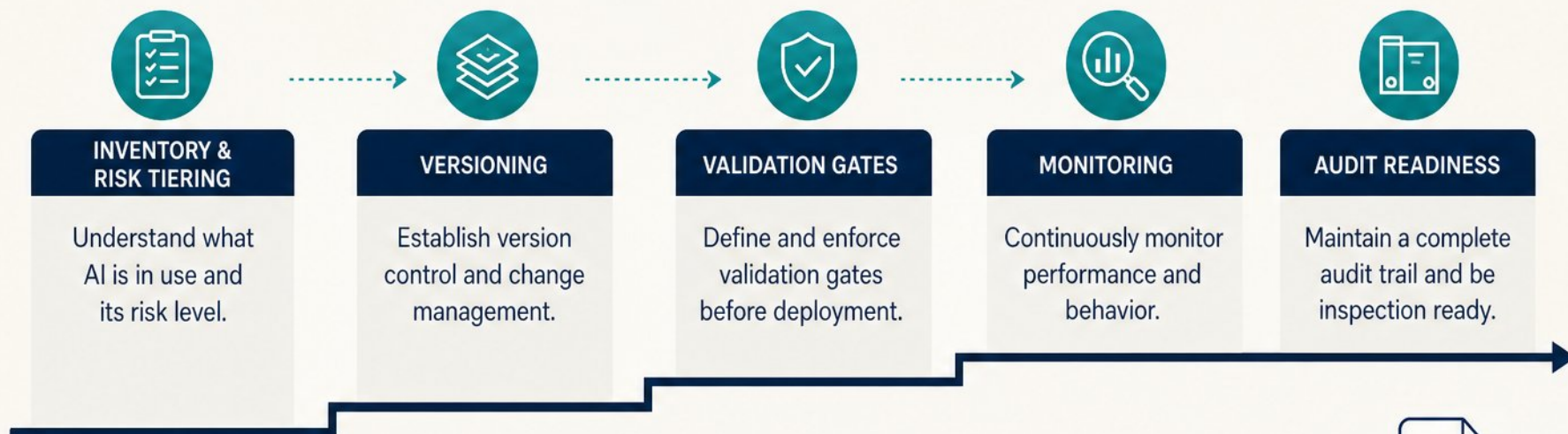


LAYERED CONTROL STACK



END-TO-END GOVERNANCE ACROSS THE AI LIFECYCLE

Practical Adoption: Starting Light and Scaling



- **Start with an AI inventory and risk tiering before buying tools**



- **Maturity path: versioning, validation gates, then monitoring, then audit readiness**



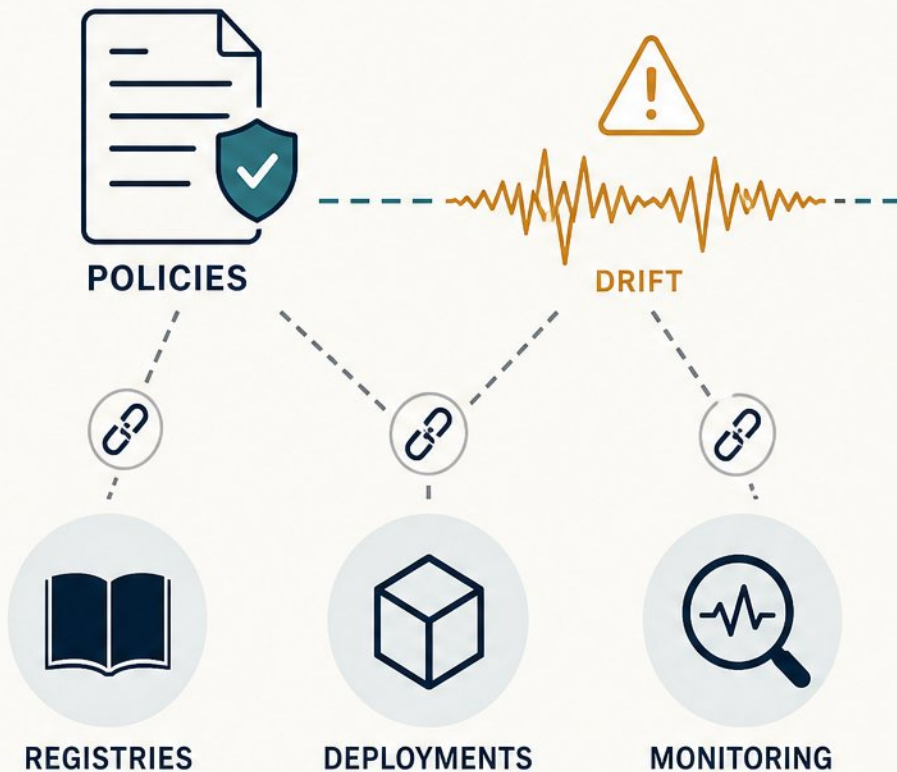
- **Use starter artifacts: system cards, model cards, fairness reports, monitoring plans**



- **Track health metrics: approval time, evidence completeness, exceptions, drift detection time**



Common Failure Modes and How to Avoid Them



- Governance without enforcement: policies never enter registries, deployments, or monitoring
- Recurring mistakes: measuring only global accuracy and unclear intended use
- More traps: overreliance on automation, testing once, ignoring retirement
- Alerting without runbooks, ownership, and escalation creates noise, not control
- Treat user feedback, appeals, and complaints as risk management inputs

Implementation Roadmap and Next Steps

From lightweight controls



To evidence-on-demand operating system



- **Phase 1:** Inventory models, then add lightweight controls and gates



- **Phase 2:** Wire CI/CD integration, monitoring, and audit readiness



- **Product owners:** define intended use and risk tier for each model



- **ML engineers:** version data, code, and environments together



- **Risk teams:** adopt NIST AI RMF and ISO/IEC 42001 baselines



- **Governance** is an operating system, not a handbook



PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/10d248a2/public