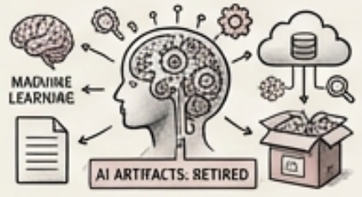
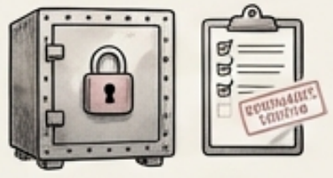
SOFTWARE DECOMMISSIONING: SECURELY RETIRING AI-AUGMENTED APPLICATIONS

A FRAMEWORK FOR OBSOLETE SYSTEMS, DATA SECURITY, AND COMPLIANCE

Software Decommissioning: Securely Retiring AI-Augmented Applications





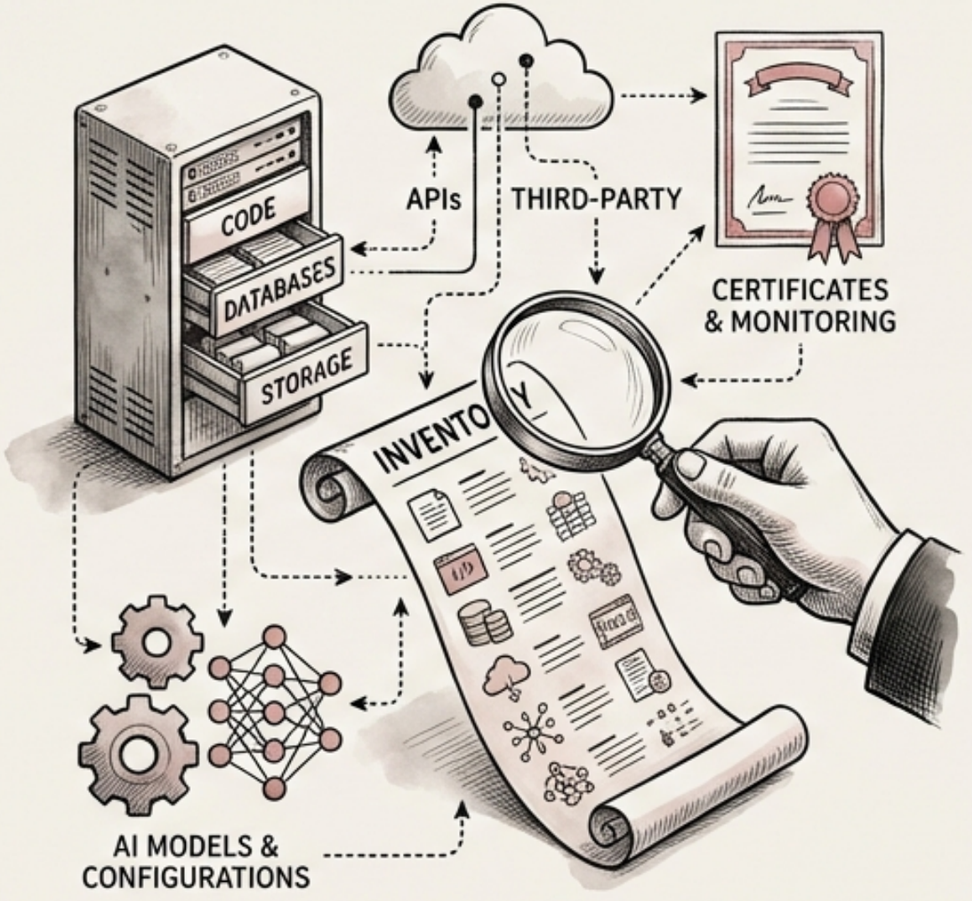
- Software decommissioning is the final phase of the Secure Software Development Lifecycle (SSDLC).
- It involves retiring applications, services, and all associated data securely and completely.
- For AI-augmented applications, decommissioning extends to AI models, training data, AI tool integrations, and AI-generated artifacts.
- Proper decommissioning minimizes security risks and ensures compliance with data privacy regulations.
- Failing to decommission properly leaves systems vulnerable to breaches and data leaks.

Why Decommissioning Matters: Mitigating Critical Security Risks

- Abandoned software poses a significant security risk due to unpatched vulnerabilities and forgotten credentials.
- Zombie applications are unmaintained services running in production and processing **real data**, creating **major security holes**.
- Failure to decommission properly can lead to exposed APIs, **stale data**, and **unauthorized** access.
- Neglecting decommissioning creates **compliance liabilities**, including **data retention** violations and failures to comply with GDPR's right to erasure.
- **Deprecated AI models** may contain **biased outputs**, training data with expired consent, or AI tools with revoked licenses.

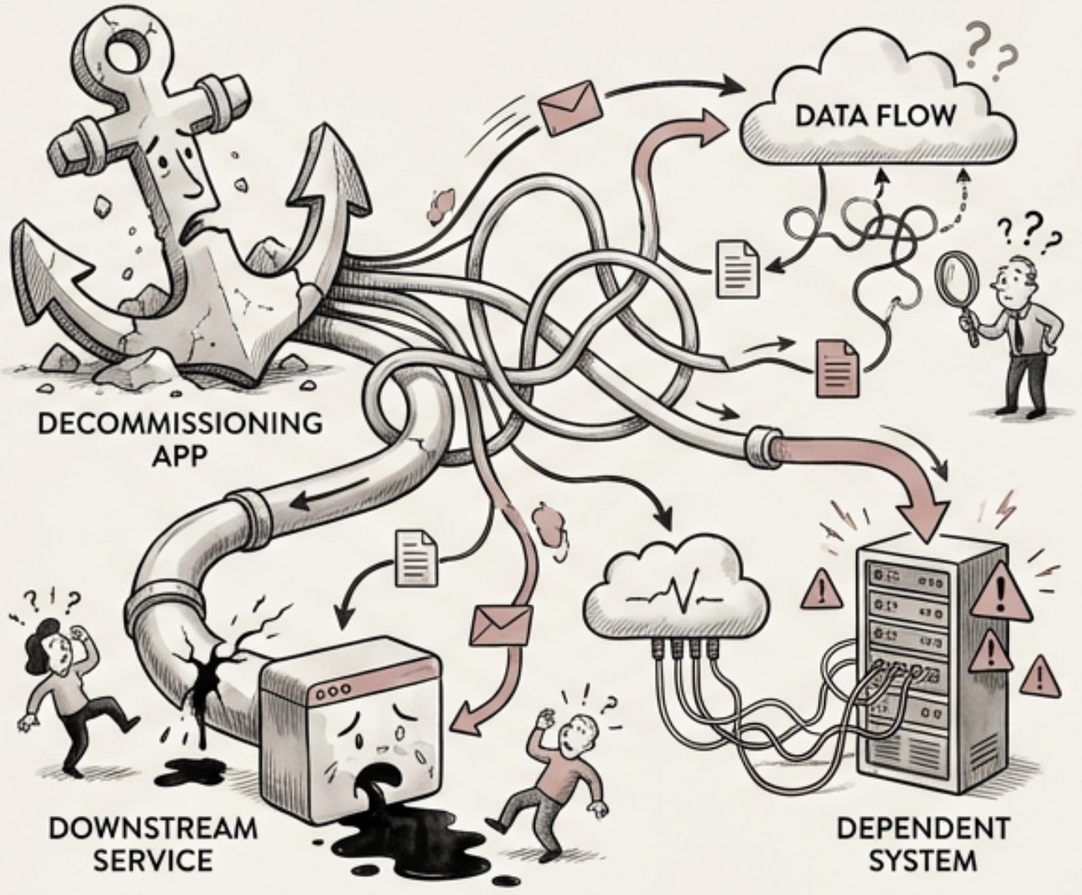
Step 1: Comprehensive Inventory – Identifying All Components

- Thoroughly identify all components associated with the application to be decommissioned.
- This includes application code, databases, APIs, message queues, storage systems, and DNS entries.
- Also, identify certificates, monitoring systems, AI models, and AI tool configurations.
- Consider ALL dependencies and integrations of the application including third-party APIs.
- Accurate inventory is vital to avoid overlooking critical elements during the decommissioning process.



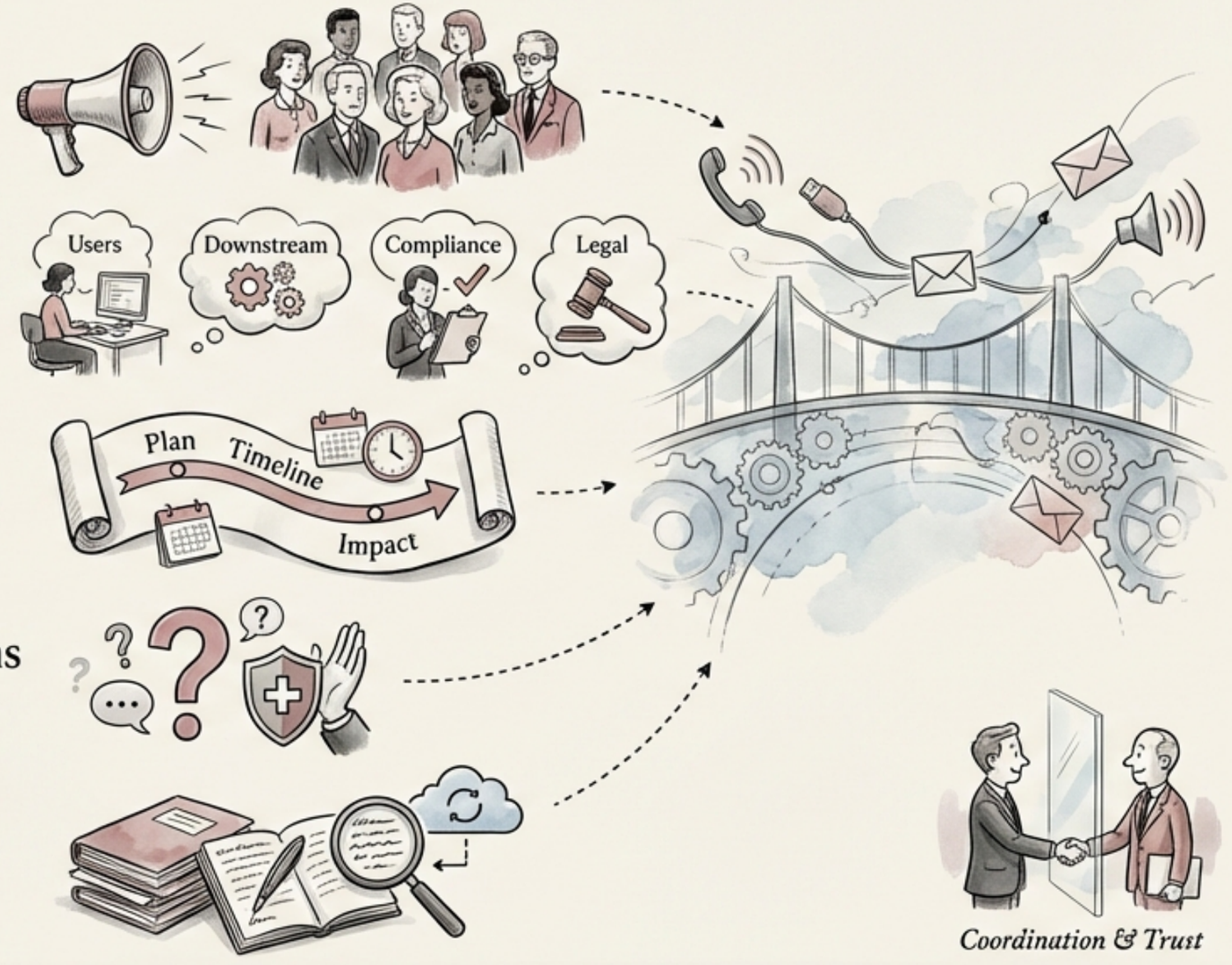
Step 2: Mapping Dependencies – Identifying Downstream Impacts

- Map all dependencies to understand which systems rely on the application being decommissioned.
- Determine what other applications or services depend on this application's functionality.
- Analyze what data flows will be disrupted or broken by the decommissioning.
- Document all upstream and downstream dependencies and data flows.
- Understanding dependencies helps plan a smooth transition and avoid unexpected disruptions.



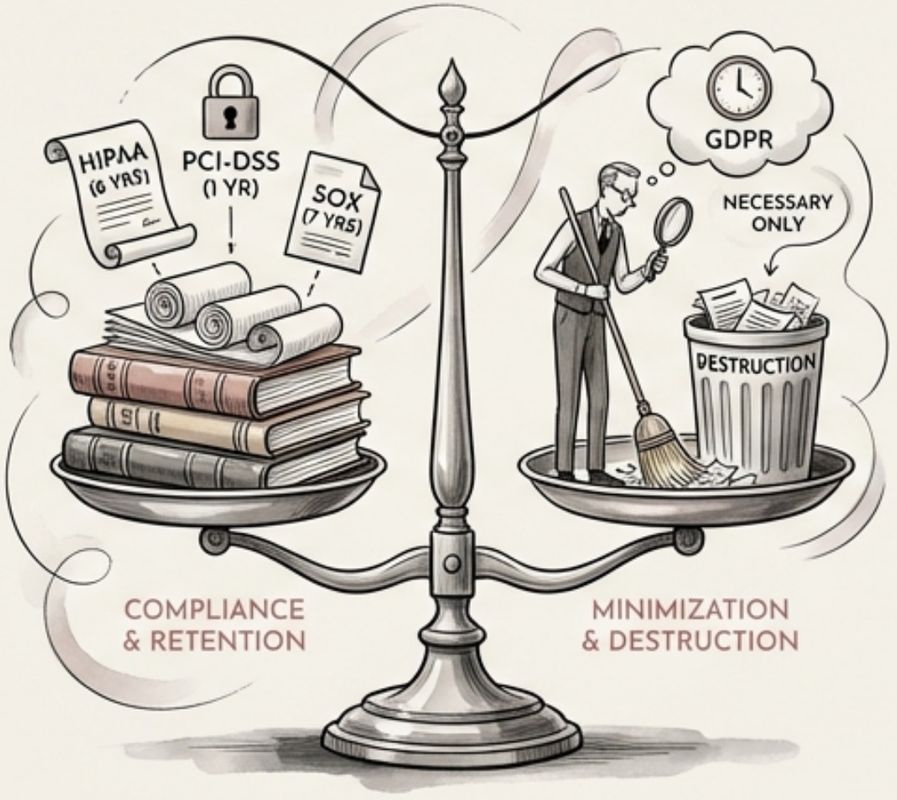
Step 3: Stakeholder Notification – Ensuring Transparency and Coordination

- Notify all relevant stakeholders well in advance of the decommissioning.
- This includes users, downstream service owners, compliance teams, and legal departments.
- Provide a clear communication plan outlining the decommissioning timeline and potential impact.
- Address potential concerns and questions from stakeholders proactively.
- Document all stakeholder communications and feedback.



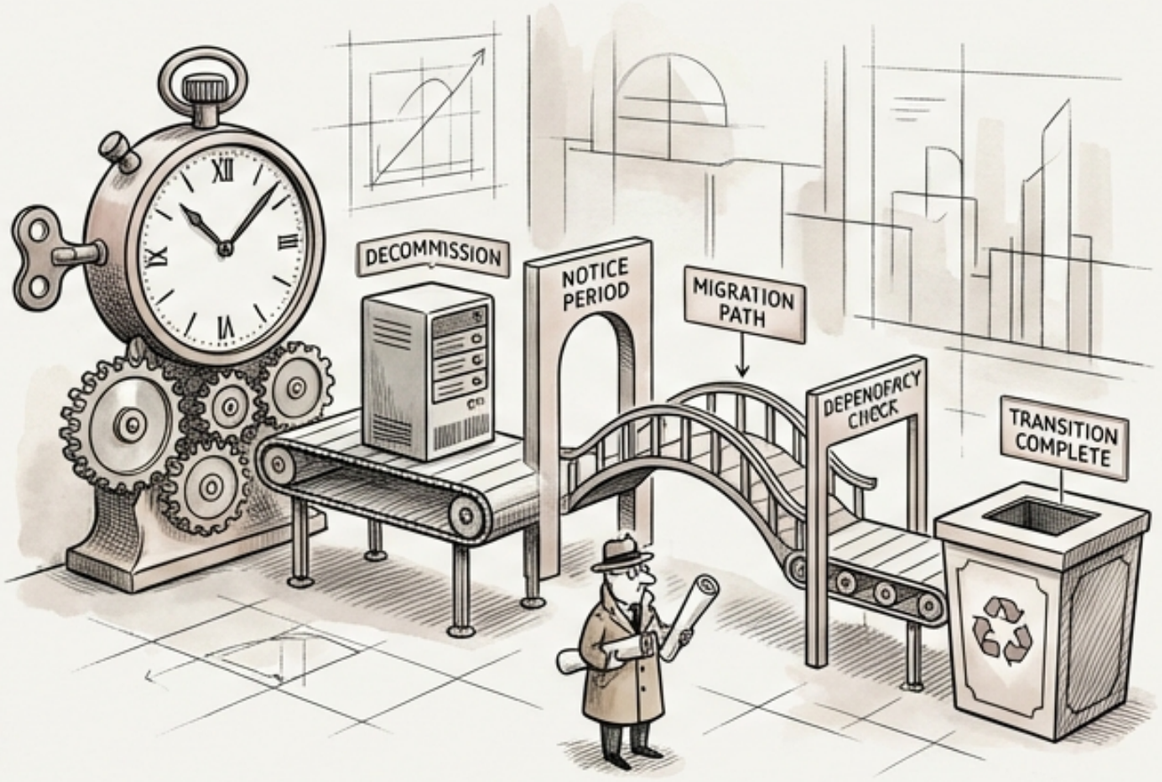
Step 4: Data Retention Requirements – Balancing Compliance and Minimization

- Define data retention requirements to determine what data must be kept for compliance and what can be destroyed.
- Retention requirements vary by regulation, such as PCI-DSS (1 year of audit logs), HIPAA (6 years), and SOX (7 years).
- GDPR requires data to be retained only as long as necessary for the stated purpose.
- Map each data store to its classification and applicable retention requirement.
- Develop a data retention schedule based on legal and regulatory obligations.



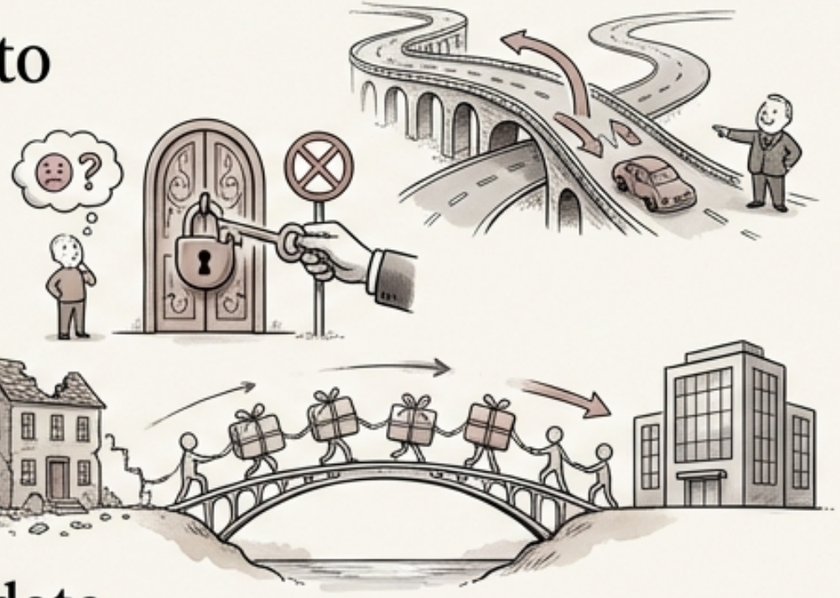
Step 5: Decommissioning Schedule – Planning for a Smooth Transition

- Develop a detailed decommissioning schedule with a realistic timeline.
- Allow adequate notice period for users and dependent systems to adjust.
- Plan for migration of dependent systems to alternative services or solutions.
- Consider dependencies and potential disruptions when creating the schedule.
- Communicate the schedule clearly to all stakeholders.



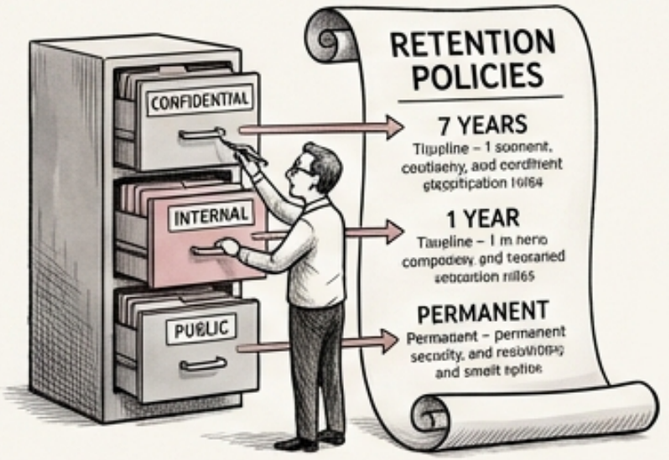
Step 6: Phased Execution – Minimizing Risks and Disruptions

- Execute the decommissioning process in phases to minimize risks and disruptions.
- **Phase 1:** Disable new access to the application.
- **Phase 2:** Migrate dependencies to alternative services.

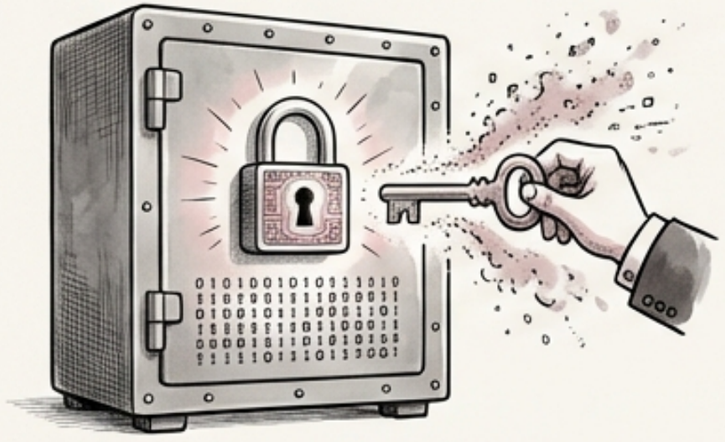


- Phase 3: Archive required data according to retention policies.
- Phase 4: Destroy remaining data securely.

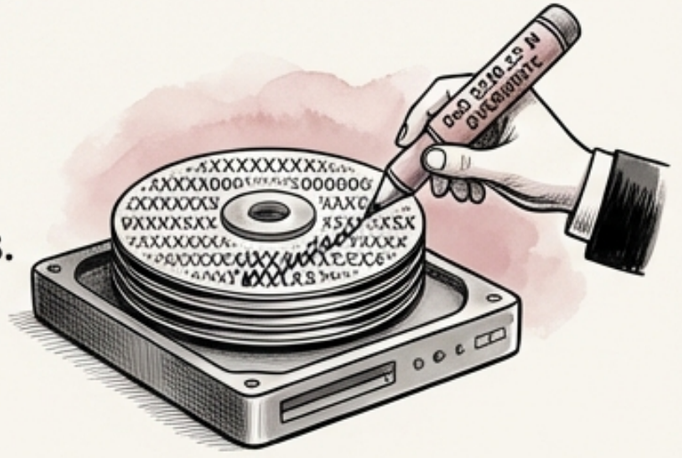
Data Retention and Destruction: Ensuring Compliance and Security



- **Data classification drives retention policies;** map each data store to its classification and requirements.



- **Cryptographic erasure** involves destroying encryption keys to render data unreadable.



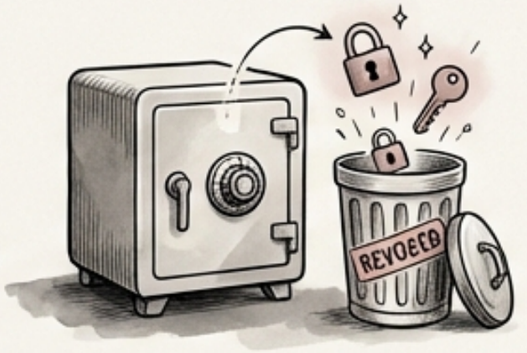
- **Secure overwrite** uses methods like DoD 5220.22-M to overwrite data multiple times.

- **Physical destruction** is used for hardware, ensuring complete data elimination.

- **Verification** through certificates of destruction, audit trails, and compliance sign-off is vital.

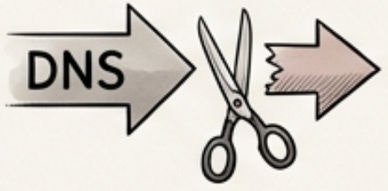
Credential and Secret Revocation: Preventing Unauthorized Access

- Revoke all credentials associated with the decommissioned application: API keys, service accounts, database credentials, OAuth clients, certificates, and SSH keys.

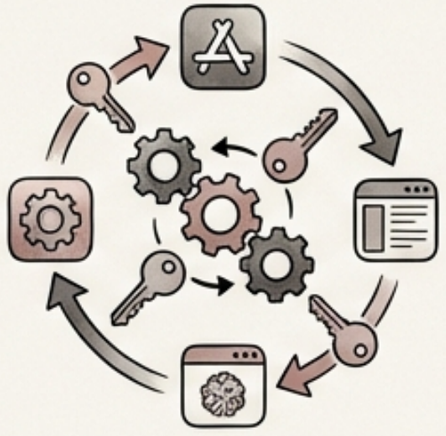


- Remove secrets from secret managers in addition to revoking them to prevent confusion and accidental reuse.

- Revoke TLS certificates and remove them from certificate transparency logs if possible.



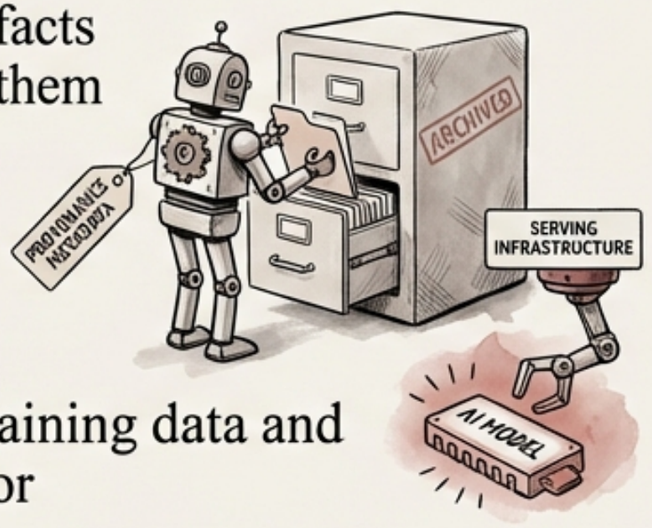
- Remove DNS entries pointing to decommissioned services to prevent dangling DNS attacks.

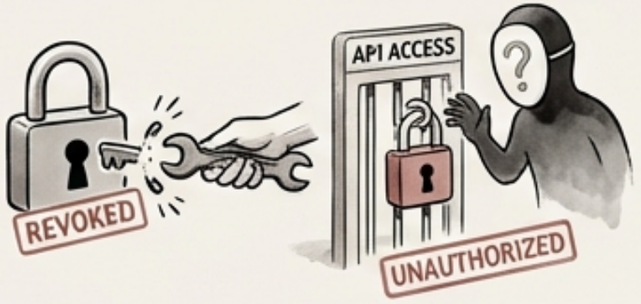


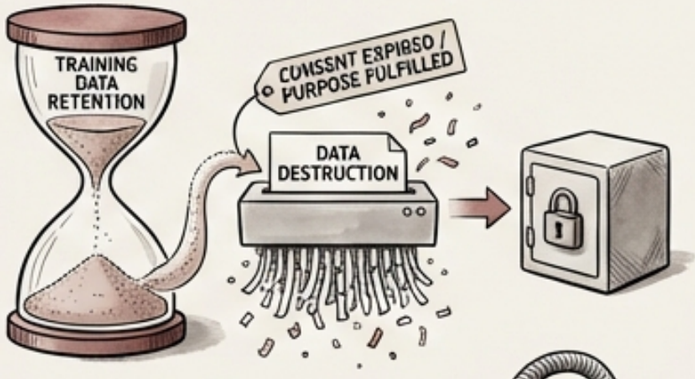
- Rotate any credentials that may have been used by the decommissioned application but are still in use by other applications.

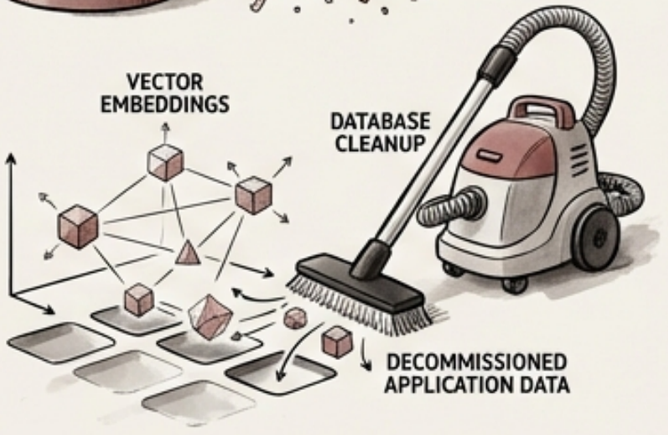
AI Model and Tool Decommissioning: Addressing Unique Challenges

- Retire AI models by archiving model artifacts with provenance metadata and removing them from serving infrastructure.
- Revoke AI model API keys to prevent unauthorized access.
- Determine retention requirements for training data and destroy data where consent has expired or purpose is fulfilled.
- Revoke AI tool API keys, remove AI tool configurations from repositories, and disable AI tool integrations in IDEs.
- Clean up embedding/vector databases by removing vector embeddings derived from decommissioned application data.



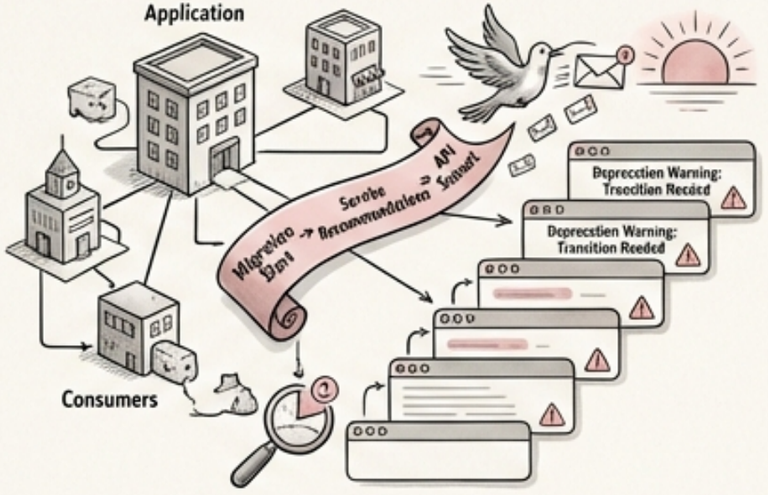


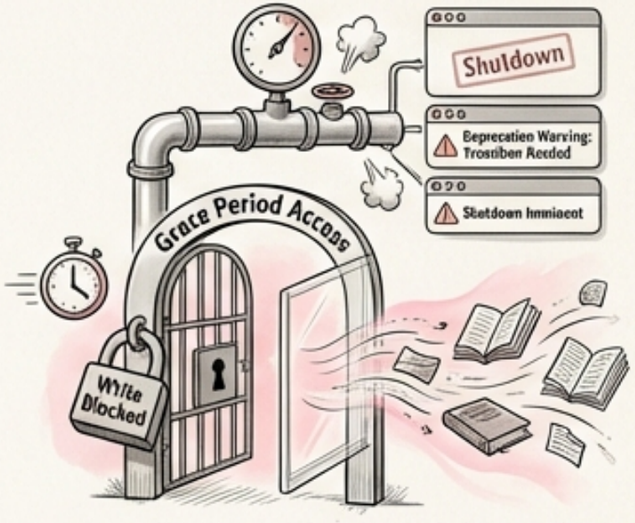


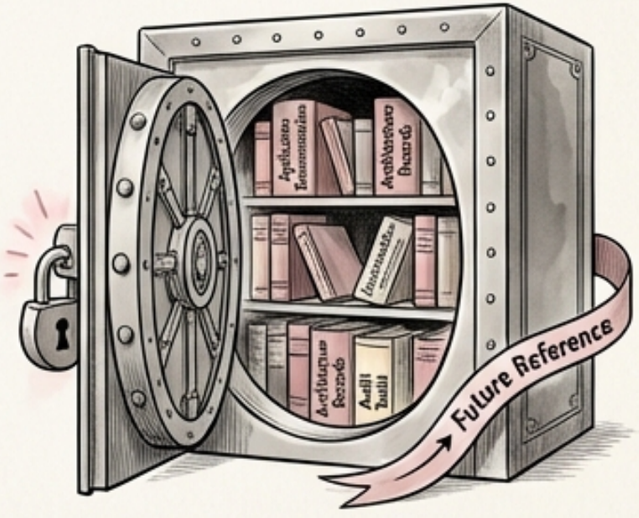


Dependency Notification: A Phased and Communicative Approach

- 1. Notify all downstream consumers of the application with a migration timeline, alternative service recommendations, and an API sunset schedule.
- 2. Implement gradual deprecation by returning deprecation warnings in API responses before shutdown.
- 3. Maintain read-only access during a grace period to ease the transition.
- 4. Document what was decommissioned, why, when, and where functionality moved (if anywhere).
- 5. Archive application documentation and architecture records for future reference and audit purposes.





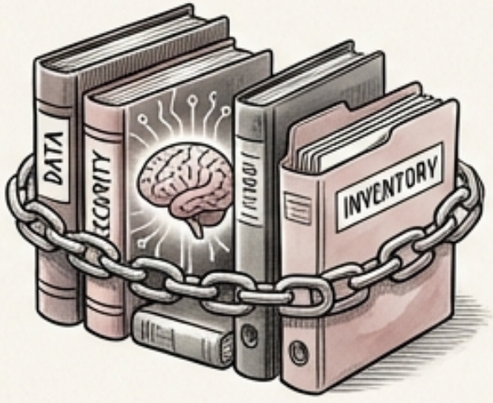


Post-Decommmissioning Verification: Ensuring Complete Removal

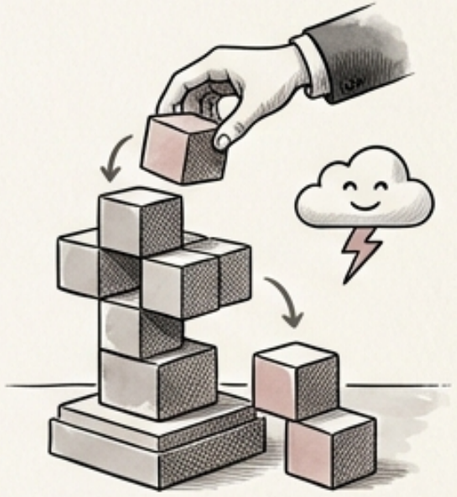
- Verify all infrastructure has been removed, ensuring no orphaned VMs, containers, databases, storage buckets, or load balancers remain.
- Verify all credentials have been revoked by scanning for any remaining active credentials associated with the decommissioned service.
- Verify all DNS records have been cleaned up, ensuring no dangling CNAME or A records exist.
- Verify monitoring has been removed, ensuring no stale alerts are firing for non-existent services.
- Verify compliance by confirming data destruction certificates have been filed, retention records have been updated, and the asset inventory has been updated.

Best Practices for Secure AI-Augmented Application Decommissioning

- 1. Create a detailed decommissioning plan and checklist tailored to AI-augmented applications.



- 3. Implement a phased comsiderations into all stages of the retention, and security, decommissioning to minimize risks and disruptions.



- 2. Incorporate AI-specific considerations into all stages of the decommissioning process, including inventory, data

- 4. Communicate clearly and frequently with all stakeholders throughout the decommissioning process.

- 5. Verify the complete removal of all application components and data after decommissioning.

Q&A: Securely Retiring AI-Augmented Applications

Open the floor for questions and discussion about software decommissioning.

Address any concerns or questions from the audience.

Provide additional resources and guidance as needed.

Thank the audience for their participation.

Provide contact information for follow-up questions.

Thank You

- Questions?