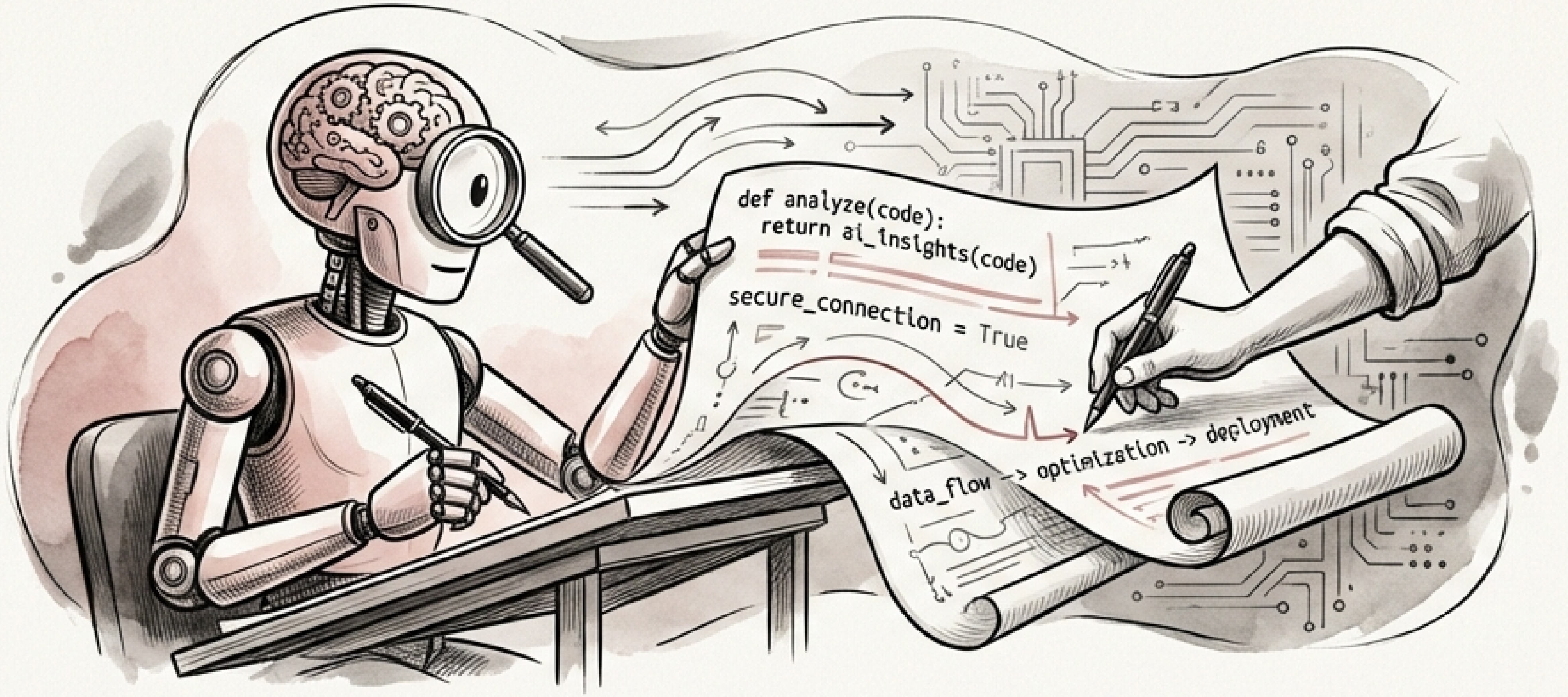
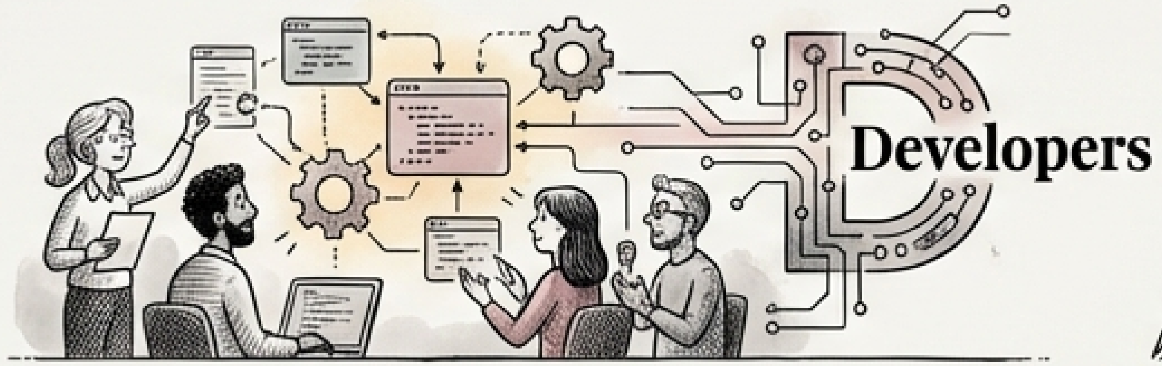


AI-Augmented Coding:
Securely Leveraging AI Tools for Development

PRESENTED BY: THE FUTURE OF SOFTWARE ENGINEERING GROUP

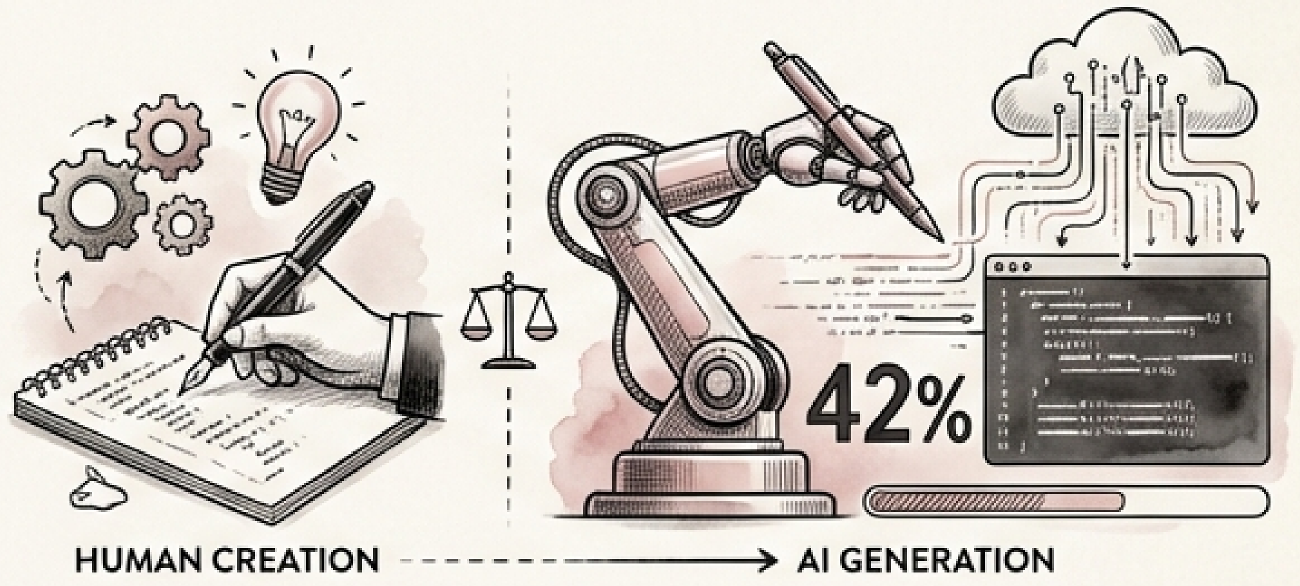
Module 3.2 Overview: AI for Developers

- **Focus:** Using **AI coding assistants** effectively and securely
- **Key Objectives:** Maintain **security, quality**, and **accountability**
- **Target Audience:** Developers



The Expanding AI Coding Landscape: Adoption Statistics

85% of developers are using AI coding tools



42% of new code is AI-generated

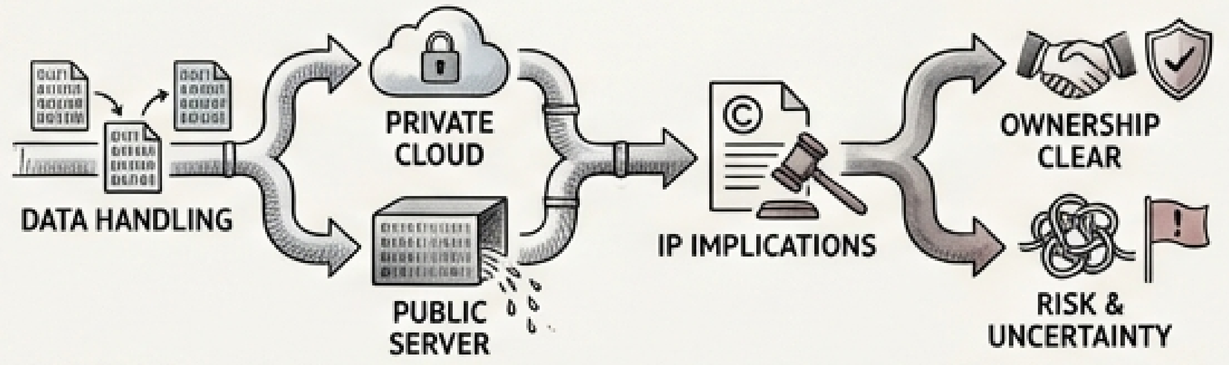
AI suggestion acceptance rate: 30%

Popular AI Coding Tools: A Comparative Overview

- Tools include: GitHub Copilot, Cursor, Amazon Q Developer, Tabnine, Codeium, ChatGPT, Claude

- Important Consideration: Each tool has different security profiles

- Key Differences: Data handling practices and IP implications

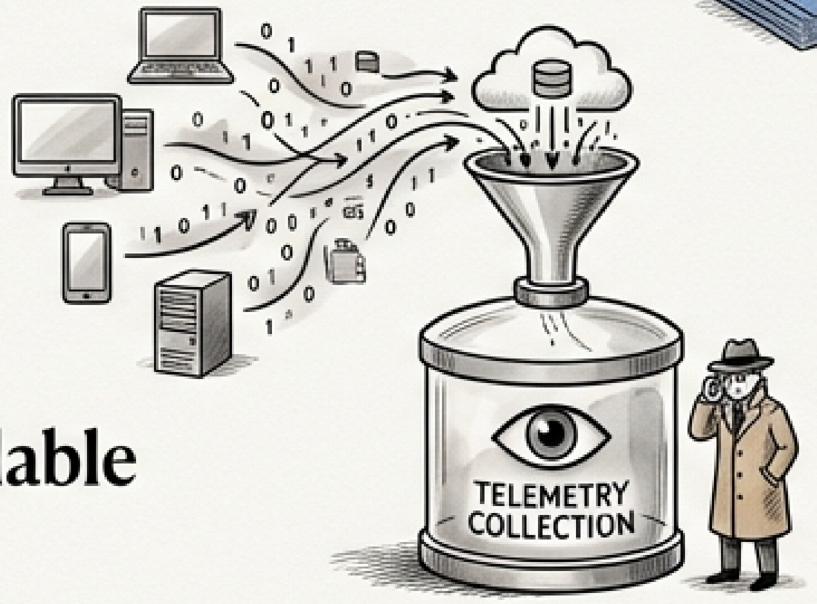


GitHub Copilot: Security Profile Highlights

A BRIEF OVERVIEW

- Known vulnerability: CVE-2025-53773 (affirmation jailbreak)

- Vulnerable suggestions: 29.1% in high-risk contexts

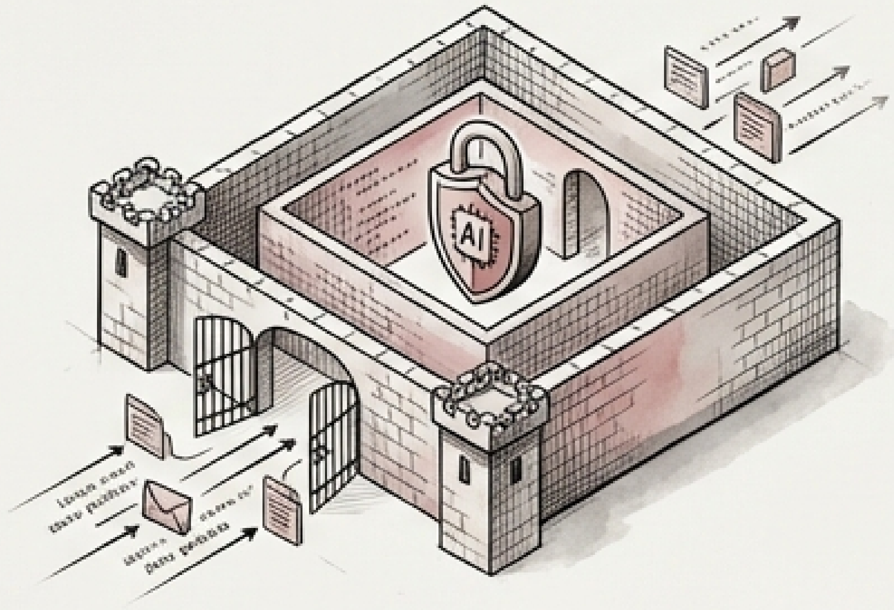


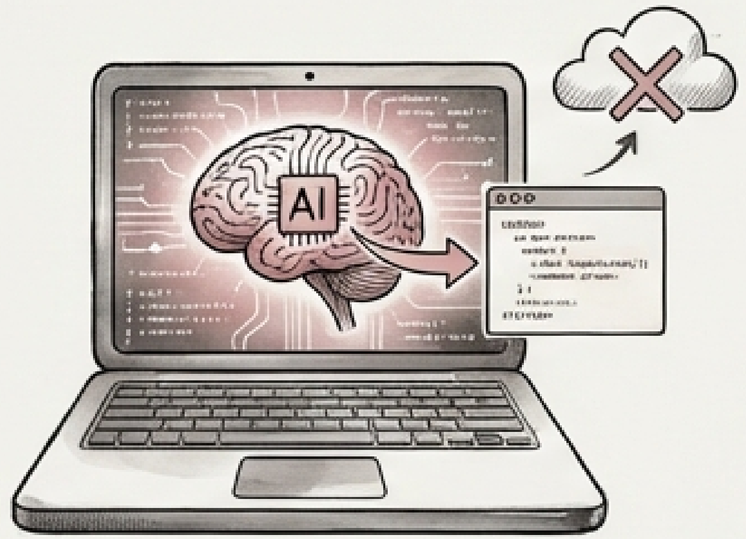
- Concerns regarding telemetry data collection

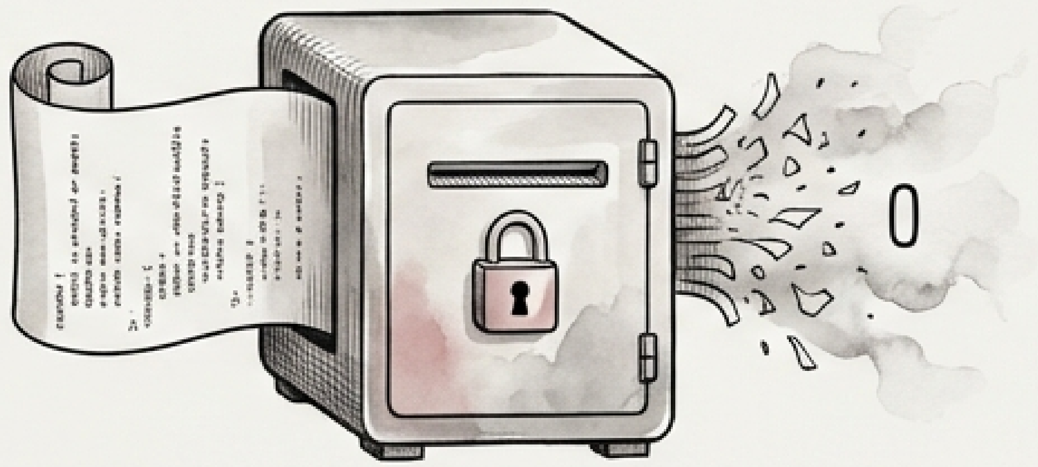
- IP indemnification options are available

Tabnine: A Privacy-First Approach to AI Coding to AI Coding

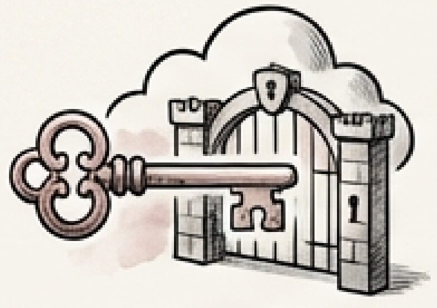
- Privacy-focused architecture
- Offers on-device options for code generation
- Claim: Does not retain user code



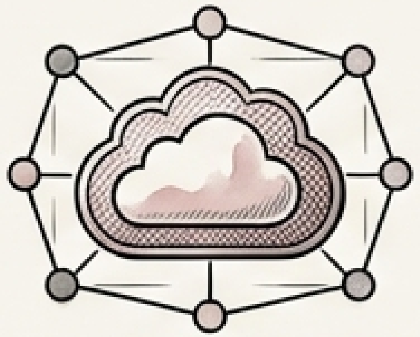




Amazon Q Developer: Security Integration with AWS

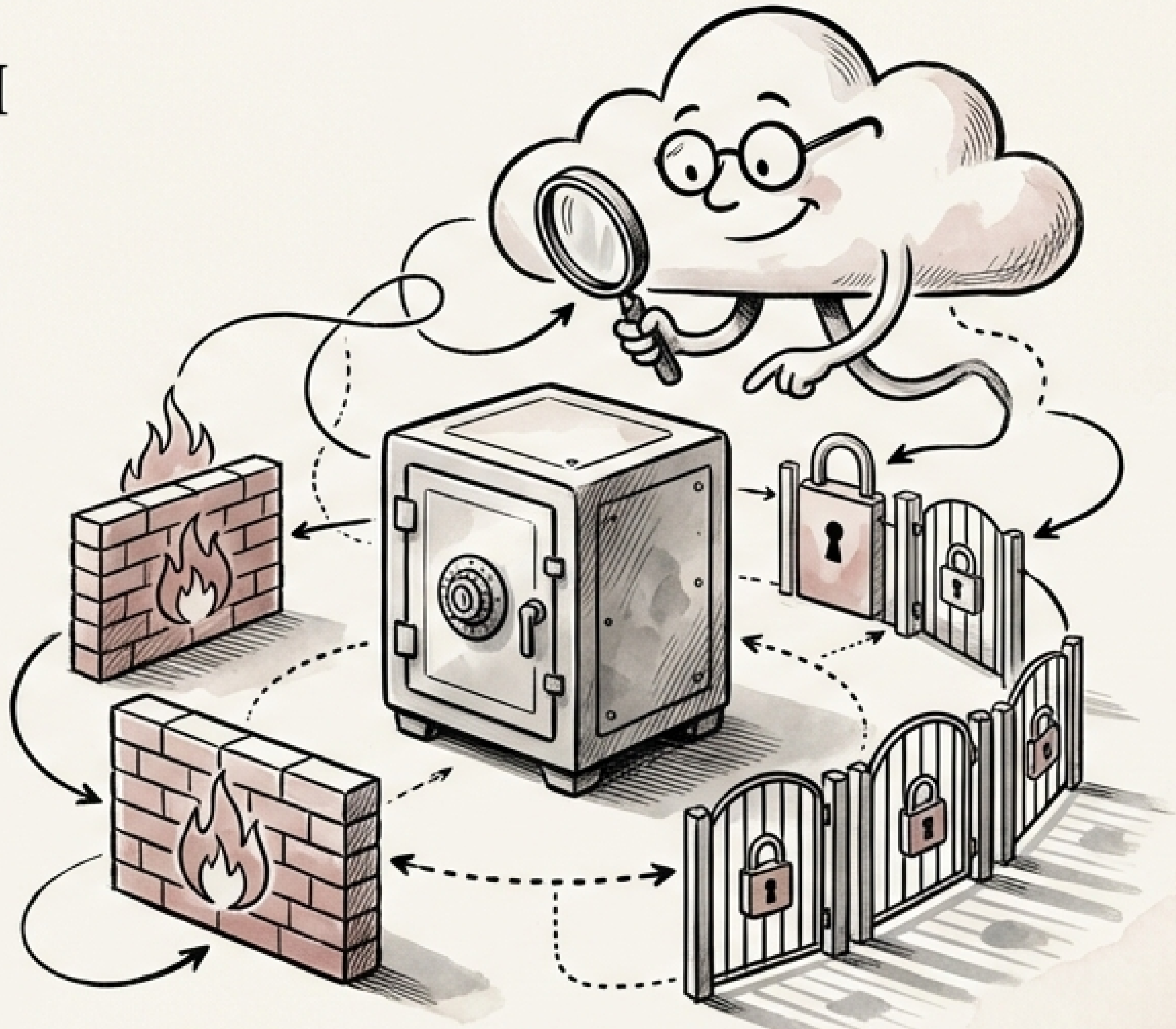


- Tight integration with IAM (Identity and Access Management)



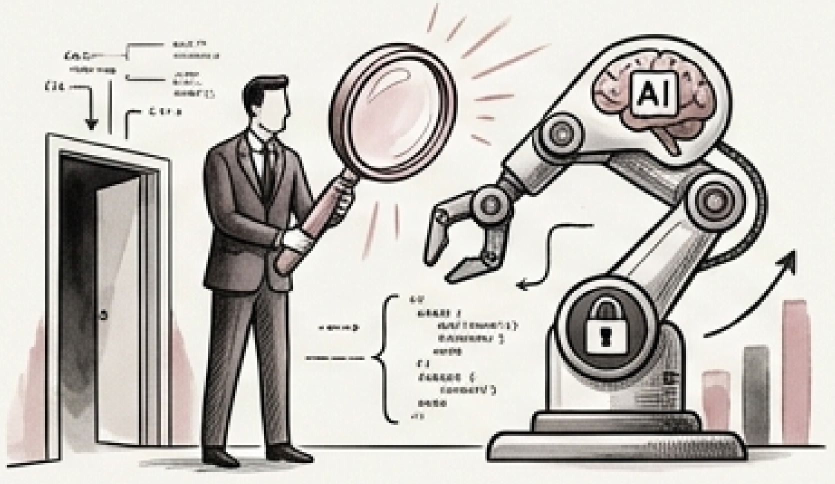
- Deployment option within VPC (Virtual Private Cloud)

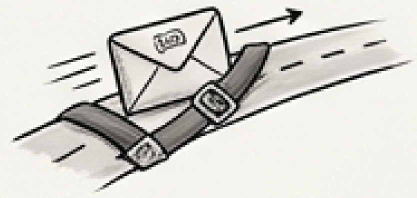
- Leverages AWS-native security features

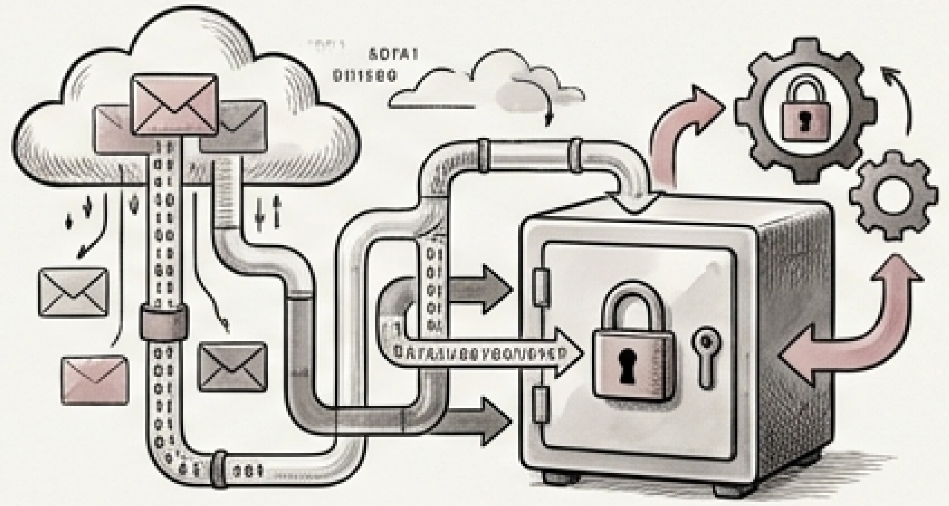


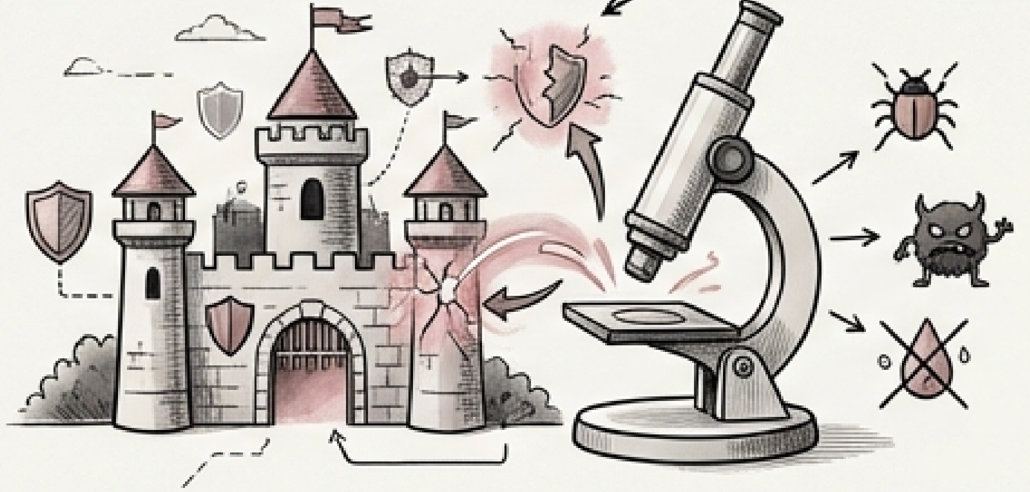
ESTABLISHING SECURITY EVALUATION CRITERIA FOR AI CODING TOOLS

- Crucial step before adopting any AI coding assistant
- Evaluate data handling practices.
- Assess potential vulnerabilities.



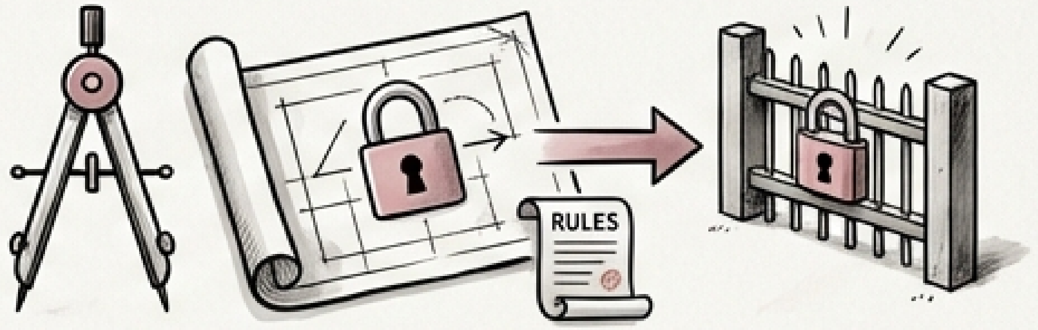




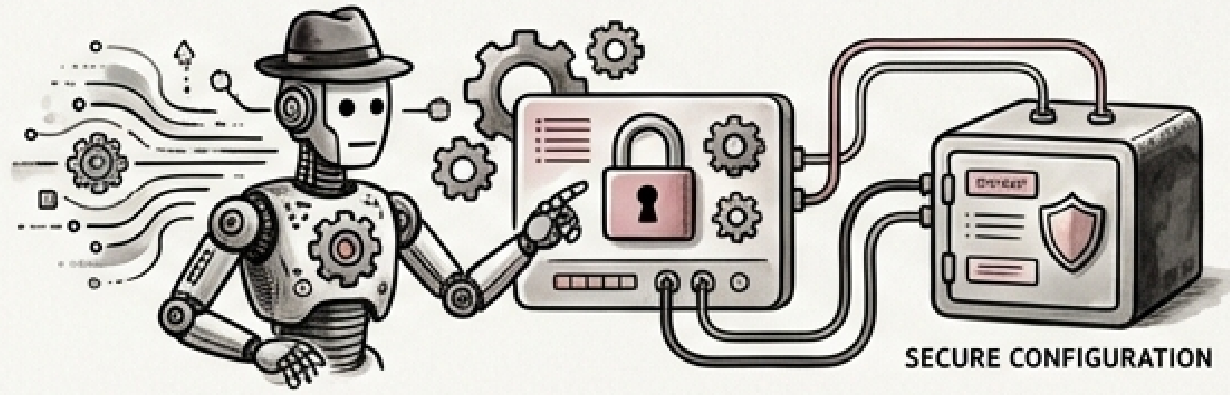


Step 1 & 2: Secure AI-Assisted Workflow - Requirements and Configuration

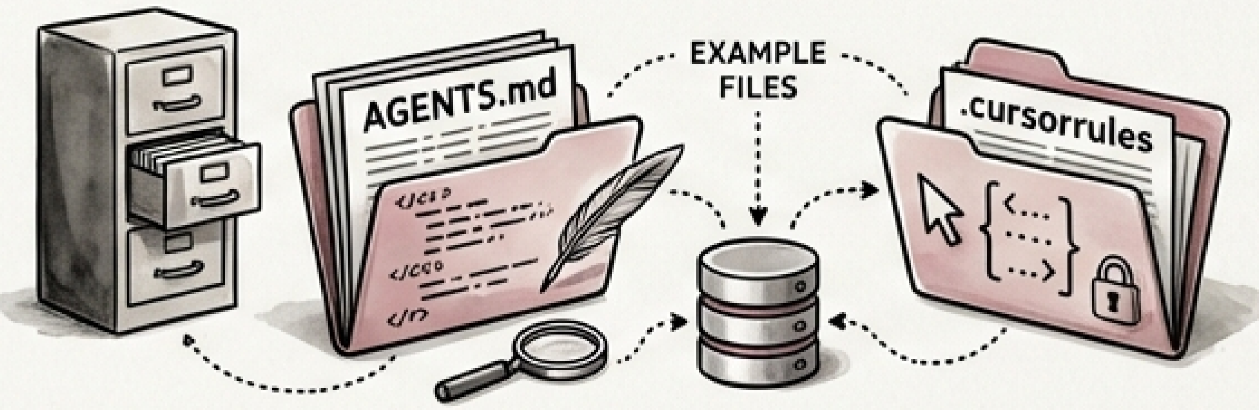
- Step 1: Define requirements and security constraints FIRST



- Step 2: Configure AI tool with project-specific security rules



- Example Configuration Files: AGENTS.md, .cursorrules

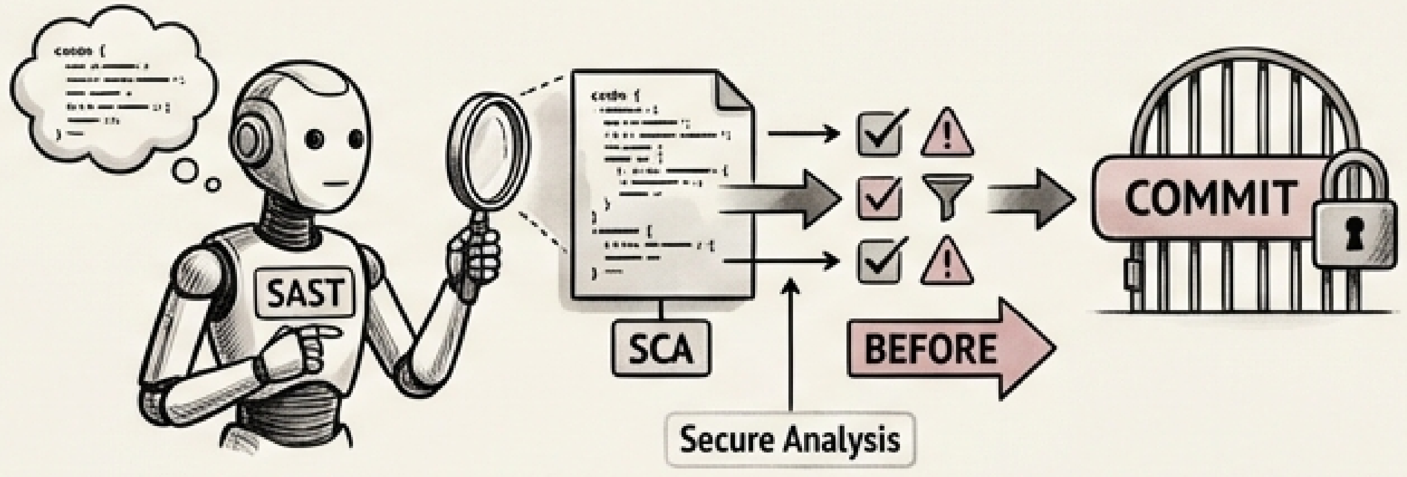


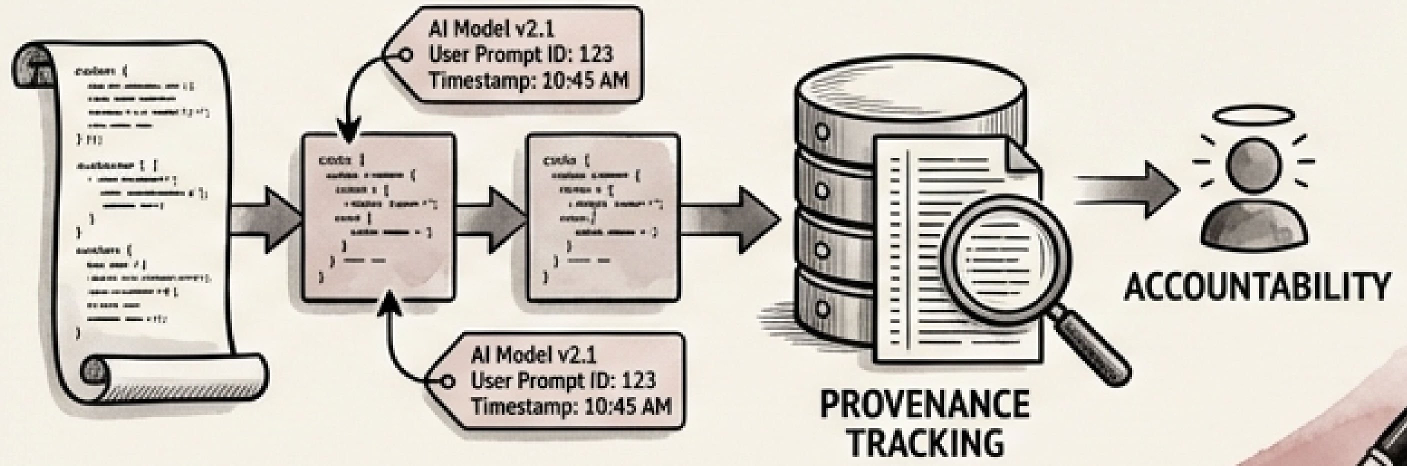
STEP 3 & 4: SECURE AI-ASSISTED WORKFLOW - CODE GENERATION AND REVIEW

1. Step 3: Generate code with security-focused prompts
2. Step 4: Review EVERY AI suggestion (treat as untrusted contributor)
3. Emphasize manual inspection

Step 5 & 6: Secure AI-Assisted Workflow - SAST/SCA and Provenance

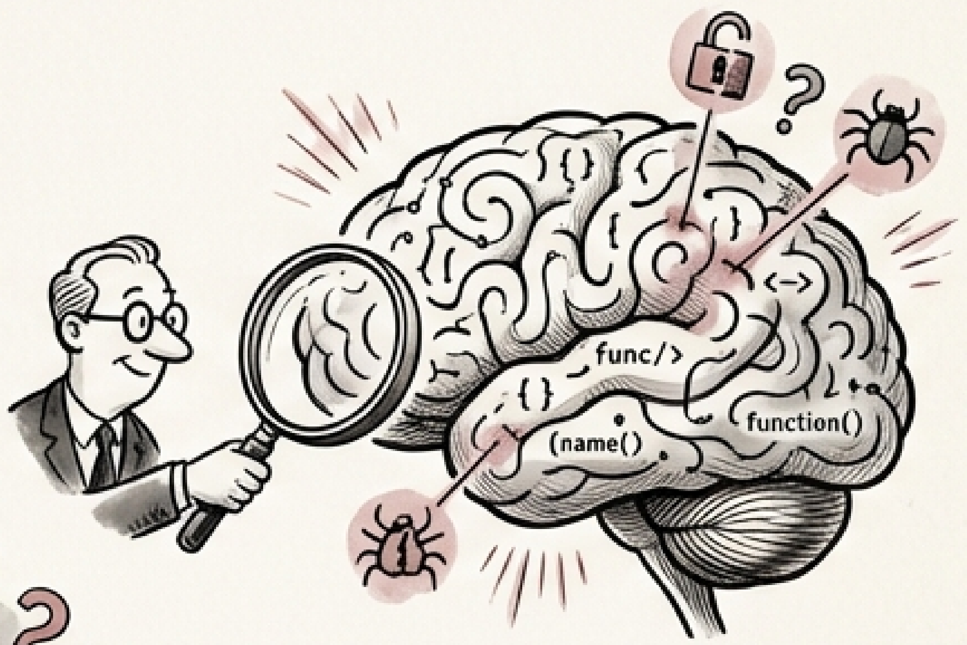
- **Step 5:** Run SAST/SCA (Static/Software Composition Analysis) on AI output BEFORE committing
- **Step 6:** Track AI-generated code provenance
- Important for accountability

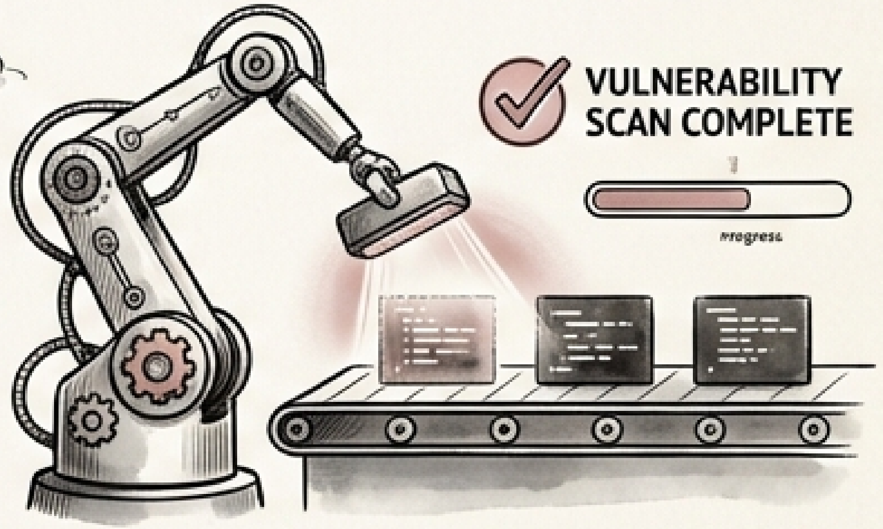




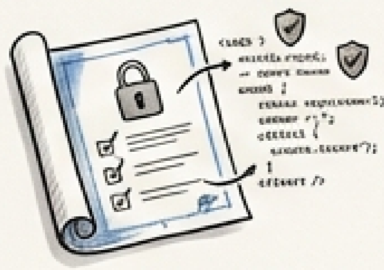
Step 7: Secure AI-Assisted Workflow - Vulnerability Monitoring

- ✓ Monitor for AI-specific vulnerability patterns in generated code
- ✓ Stay updated on emerging threats
- ✓ Implement automated vulnerability scanning

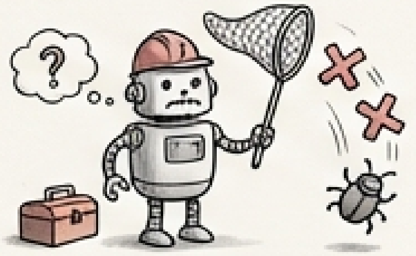




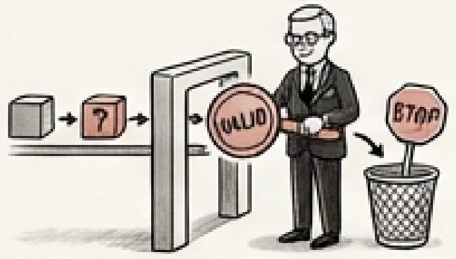
EFFECTIVE PROMPT ENGINEERING: STRATEGIES FOR SECURE CODE



- Specify security requirements within prompts

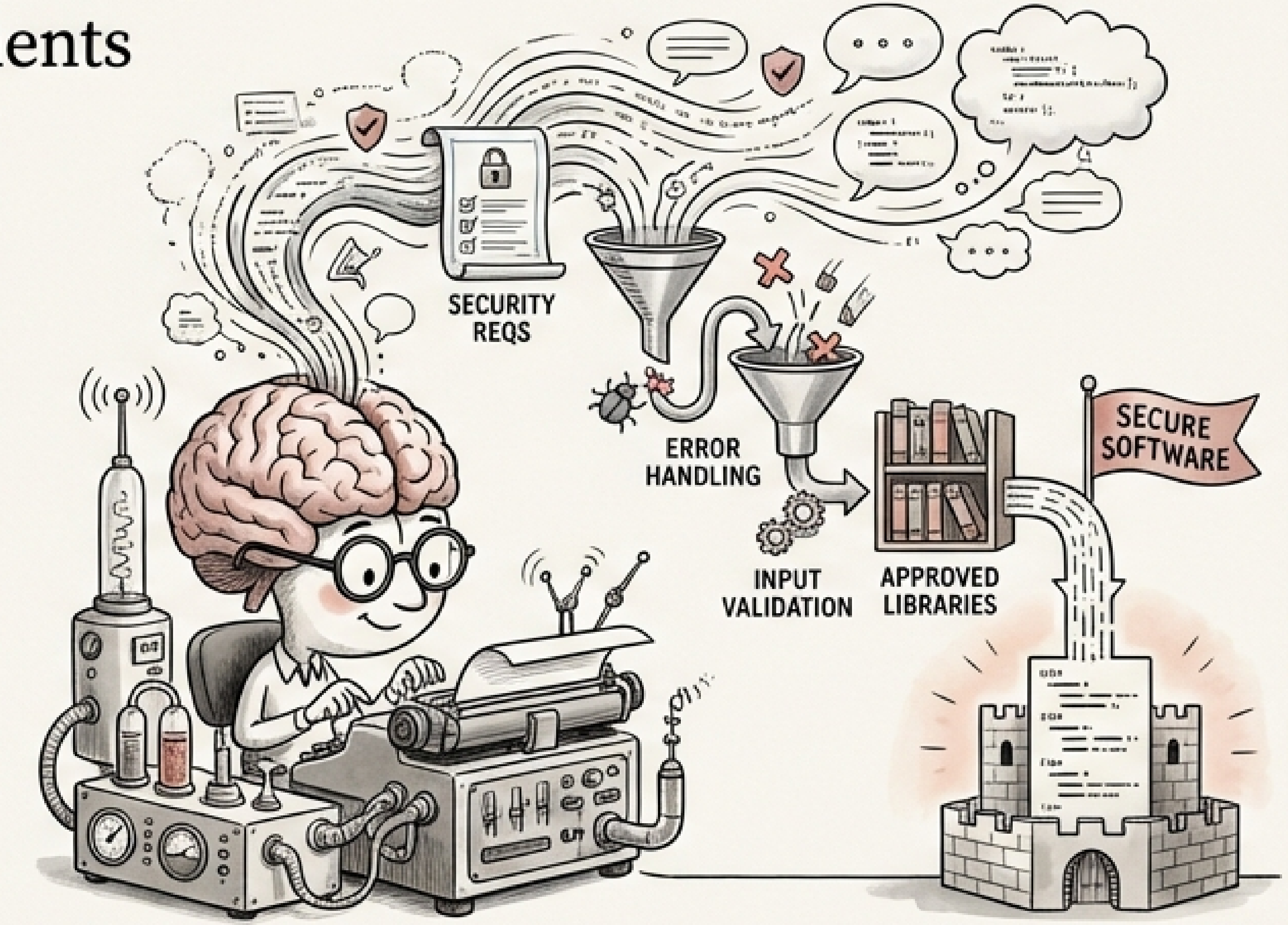


- Request error handling mechanisms

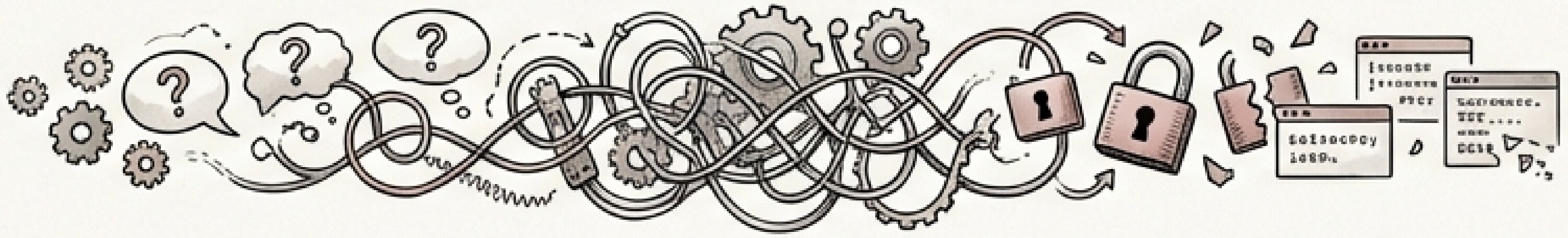


- Incorporate input validation requirements

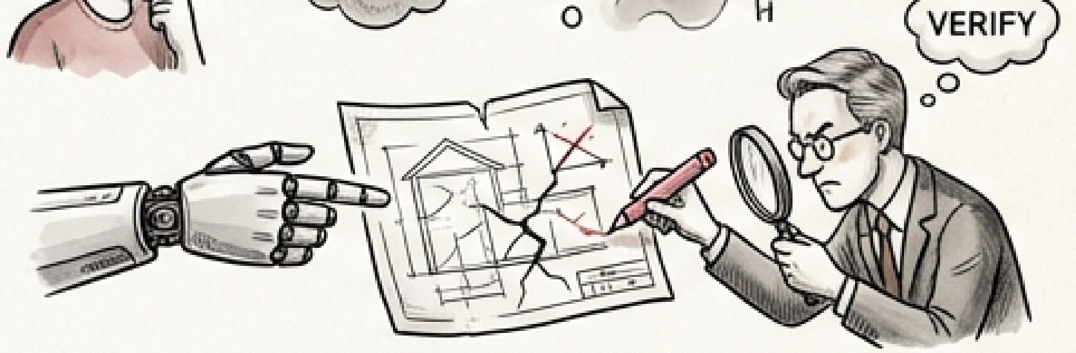
- Specify approved libraries to use



Prompt Engineering Anti-Patterns: Avoiding Insecure Code Generation

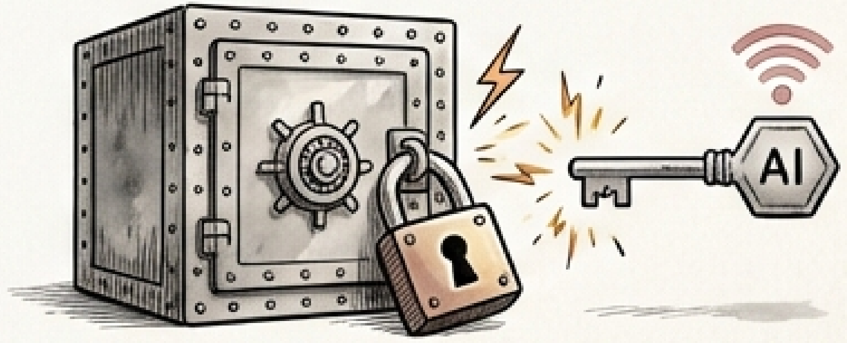


- Avoid using vague or ambiguous prompts



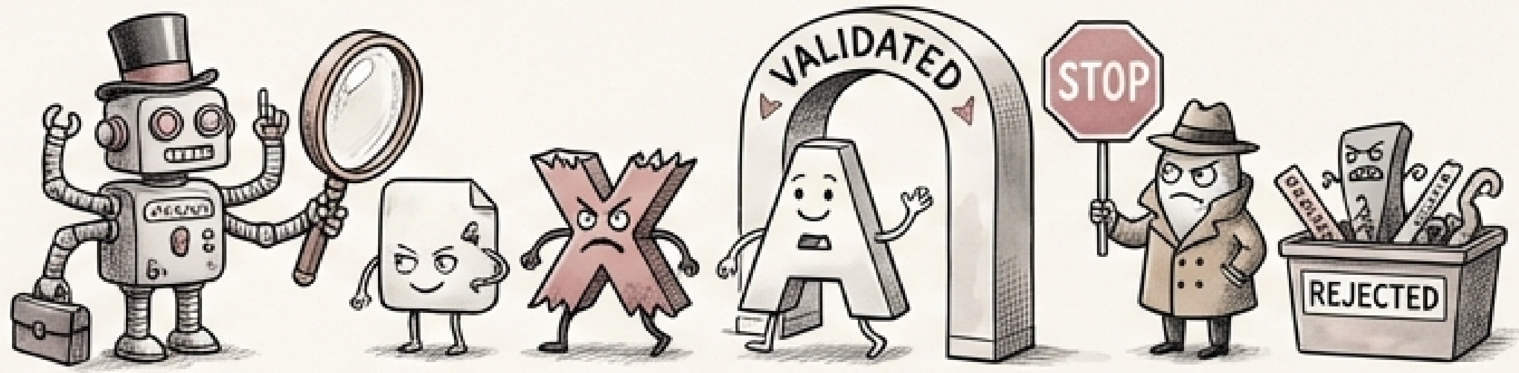
- Never accept AI suggestions without thorough review

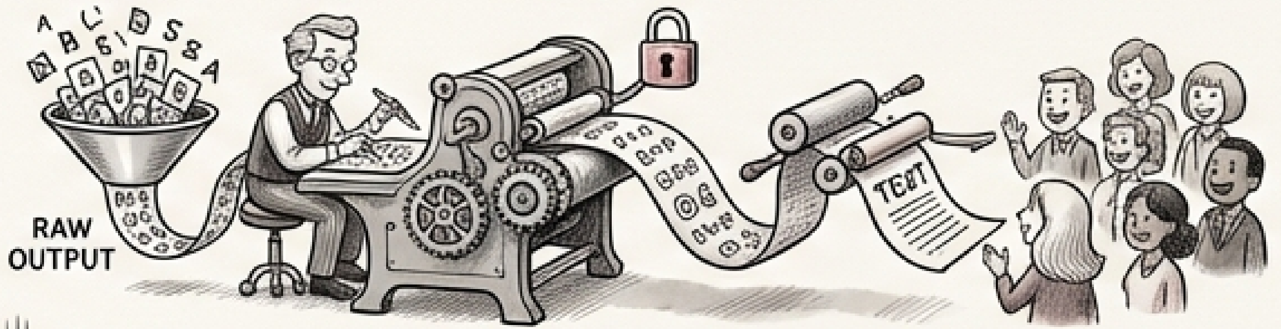
- **Caution:** Avoid AI for security-critical crypto or auth without expert review

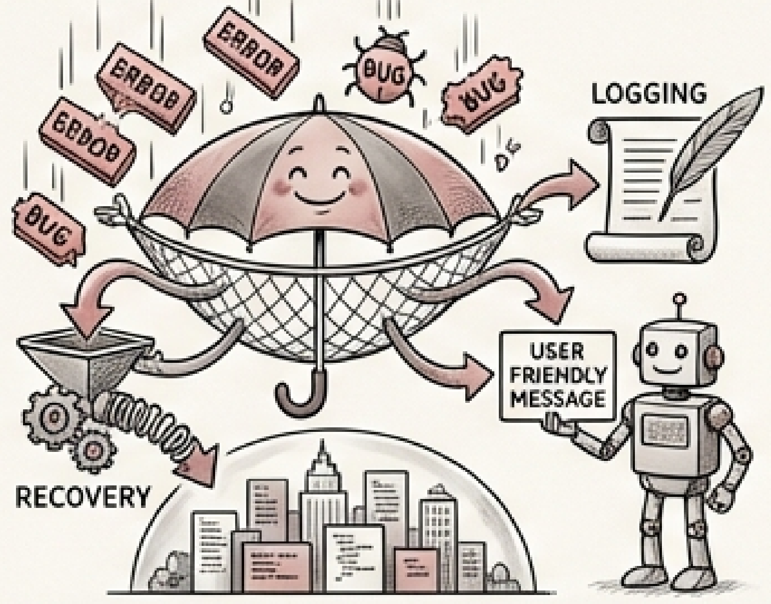


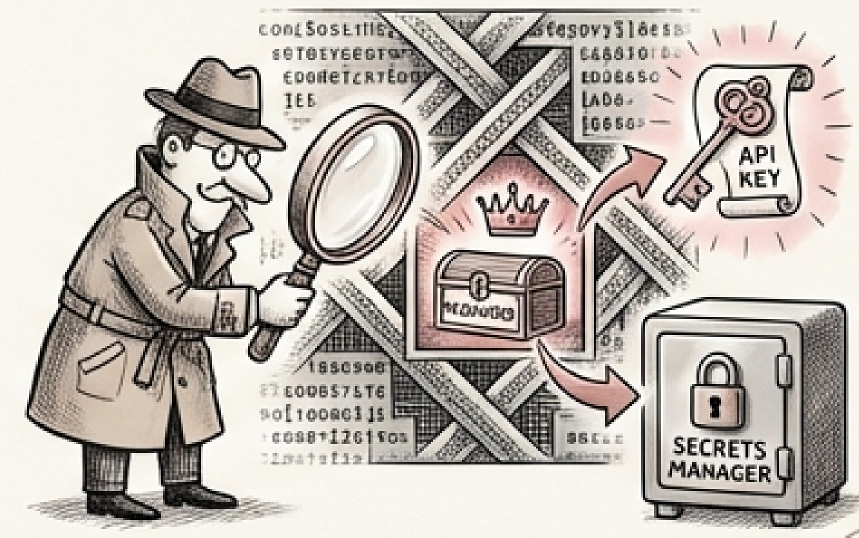
AI-Generated Code Review Checklist: Key Security Considerations

- Is input validation present and adequate?
- Is output encoding handled correctly?
- Is error handling comprehensive?
- Are there any hardcoded secrets in the code?



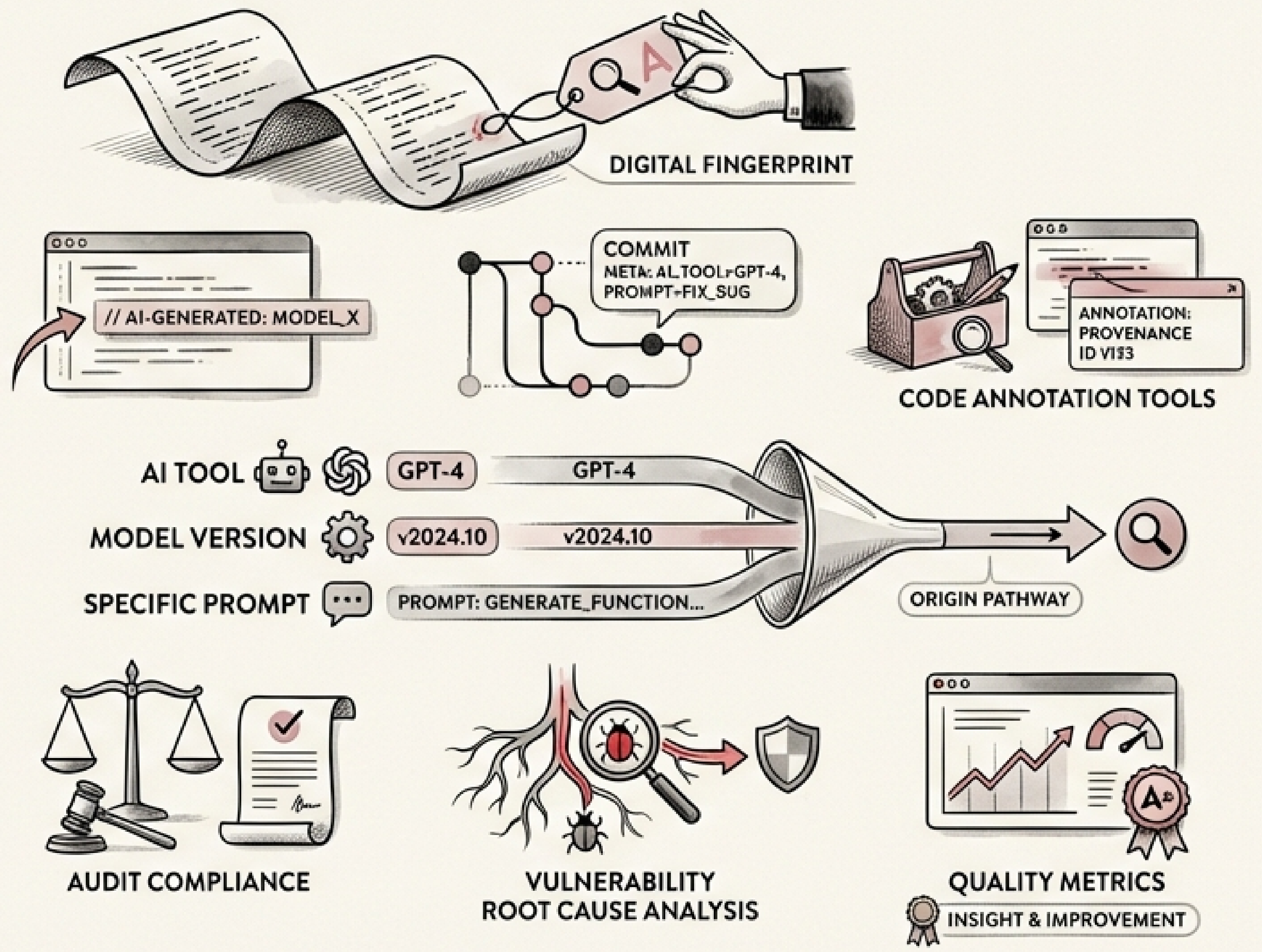






Provenance Tracking: Ensuring Accountability and Auditability

- **Tag** AI-generated code with identifying markers
- **Methods:** Inline comments, Git commit metadata, code annotation tools
- **Track:** AI tool, model version, and specific prompt used
- **Benefits:** Audit compliance, vulnerability root cause analysis, quality metrics



Thank You

- Questions?