How to Build an Approved Component Registry

Evaluating, approving, and managing third-party components

Step 1: Define Evaluation Criteria

Establish a scoring rubric for third-party components before adding them to your approved list:

- Maintenance activity: commits in last 90 days, issue response time
- Security posture: CVE history, security policy, signed releases
- License compatibility: approved license list (MIT, Apache 2.0, BSD)
- Community health: contributor count, bus factor, OpenSSF Scorecard
- AI provenance: check for slopsquatting risk (AI-hallucinated package names)

Step 2: Implement Approval Workflow

Create a lightweight process for requesting and approving new dependencies:

- Developer submits request with justification and evaluation scores
- Security team reviews within 5 business days
- Approved components added to allow-list in package manager config
- Denied components documented with rationale for future reference

Step 3: Lifecycle Management

Approved components need ongoing monitoring:

- Automated vulnerability scanning (Dependabot, Snyk, Renovate)
- Quarterly review of component health (is it still maintained?)
- Deprecation process: 90-day migration window when component is removed from approved list
- Emergency process: immediate ban + replacement plan for critical CVEs