AppSec Architecture Documentation Package

Views-and-Beyond Style Diagram Backlog

Scope: Intake, Threat Modeling, Vulnerability Management, Exceptions, CI/CD Gates, and Reporting

Document Type:	Architecture Documentation Backlog
Methodology:	Views and Beyond (V&B)
Format:	Epics and User Stories
Status:	Ready for Execution

Application Security Program

Version 1.0

Contents

1 Introduction and Conventions

This document presents a comprehensive diagram backlog for Application Security (AppSec) architecture documentation, organized using the Views-and-Beyond approach. The backlog is structured as architecture documentation packages (epics) containing diagram user stories (deliverables).

The execution order is designed to deliver incremental business value:

Foundation → Current-State → Target-State → Automation/Integration → Metrics & Governance

1.1 View Types (Views & Beyond)

Architectural View Types

Context/Scope (C4 L1)
Boundaries, actors, external dependencies

Process View (BPMN)
End-to-end workflows, swimlanes, decision points

Information/Evidence View (DFD)
Inputs/outputs, evidence artifacts, record systems

Component-and-Connector (C4 L2–L3)
Tools/services and signal movement between them

Allocation View (Deployment/Responsibility)
Where things run, who owns what (RACI)

Beyond Views
Glossary, assumptions, policies, SLAs, decision rules, traceability, roadmap

1.2 Standard Diagram Story Card Fields

Each diagram deliverable follows a consistent story card format with the following fields:

- **Deliverable** — The artifact to be produced
- **Primary Stakeholders & Concerns** — Who needs this and why
- **Inputs** — Required source materials
- **Notation** — Diagram type and modeling language
- **Acceptance Criteria** — Definition of done
- **Dependencies** — Prerequisite deliverables

2 EPIC 0 — Foundation

AppSec Architecture Documentation Package (Foundation)

This foundational epic establishes the baseline artifacts required for all subsequent documentation. It defines stakeholders, boundaries, services, and shared terminology.
Dependencies: None (this is the base for everything else)

2.1 0.1 Stakeholders & Concerns Map

Deliverable: Stakeholder–Concern matrix covering Exec/BOD, Engineering leaders, Development teams, SRE, GRC/Audit, and AppSec
Notation: Table + short narrative
Acceptance Criteria: Every later diagram links back to at least one concern

2.2 0.2 AppSec System Context and Boundaries

Deliverable: System Context diagram showing AppSec as a service and its interfaces with Engineering, CI/CD, Ticketing, CMDB, IAM, and GRC
Notation: C4 Level 1
Acceptance Criteria: Named systems of record for: findings, exceptions, risk acceptance, reporting

2.3 0.3 AppSec Service Catalog

Deliverable: Service catalog with entry criteria, outputs, SLAs, and escalation paths
Notation: Structured catalog page + lightweight service blueprint
Acceptance Criteria: Intake routes map 1:1 to services (no orphan request types)

2.4 0.4 Shared Glossary and Taxonomy

Deliverable: Standard definitions for: “finding,” “vulnerability,” “risk,” “exception/waiver,” “false positive,” “SLA,” “severity,” “gate”
Acceptance Criteria: Used consistently across all BPMN labels and decision tables

3 EPIC 1 — AppSec Intake

Request → Triage → Routing

This epic documents the intake process from initial request through triage, categorization, and routing to appropriate queues.
Dependencies: EPIC 0.3 Service Catalog, EPIC 0.4 Glossary

3.1 1.1 Current-State Intake Workflow

Deliverable: "As-Is" intake BPMN: request submission → triage → categorization → routing → queue/assignment → closure
Primary Stakeholders: Developers (friction), AppSec (load), Engineering managers (predictability)
Inputs: Existing intake channels (email/forms/tickets), categories, current SLAs
Notation: BPMN swimlanes (Dev / AppSec / Eng Manager / GRC as needed)
Acceptance Criteria:

- Single start event and explicit end states (Completed, Rejected, Needs Info, Routed)
- Triage decision points use named criteria (severity, due date, compliance driver)

3.2 1.2 Target-State Intake Workflow + SLA Model

Deliverable: "To-Be" BPMN with standardized intake form fields, auto-routing rules, SLAs by request type
Acceptance Criteria: Every routing decision has an explicit rule and owner

3.3 1.3 Intake Decision Table (Routing Rules)

Deliverable: Decision table that maps request type + risk + due date → queue/owner/SLA
Notation: Decision table (DMN-lite is acceptable)
Acceptance Criteria: No "tribal knowledge" steps remain; all routing logic is documented

3.4 1.4 Intake RACI + Escalation Path

Deliverable: RACI chart + escalation swimlane overlay
Acceptance Criteria: For each step: exactly one **Accountable** role

4 EPIC 2 — Threat Modeling

Design Intake → Model → Findings → Tracking

This epic covers the threat modeling lifecycle from initial design engagement through model creation, finding identification, and remediation tracking.
Dependencies: EPIC 1 Intake (threat modeling typically begins as an intake request)

4.1 2.1 Threat Modeling Service Blueprint

Deliverable: Service blueprint showing frontstage developer experience + backstage AppSec work + support systems
Primary Stakeholders: Development leads, AppSec, Architects
Acceptance Criteria: Includes entry criteria, artifacts required, and defined outputs (model, mitigations, backlog items)

4.2 2.2 Current-State Threat Modeling BPMN

Deliverable: As-Is BPMN: kickoff → context gathering → trust boundaries/data flows → threat enumeration → mitigations → sign-off → tracking
Acceptance Criteria: Artifacts are explicit outputs (DFD, threat list, mitigations)

4.3 2.3 Target-State Threat Modeling BPMN (Shift-Left)

Deliverable: To-Be BPMN integrating threat modeling into SDLC stages (PRD/design review, architecture review, pre-implementation)
Acceptance Criteria: Shows when threat modeling is mandatory vs. optional

4.4 2.4 Threat Model Artifacts View

Deliverable: Standard artifact set diagram: system context, DFD, trust boundaries, abuse cases, mitigations, residual risk
Notation: DFD + labeled trust boundaries + checklist
Acceptance Criteria: Each artifact mapped to where it's stored and how it's versioned

4.5 2.5 Findings Traceability (Threats → Requirements → Tickets)

Deliverable: Traceability diagram tying threat scenarios to security requirements and tracked work items
Acceptance Criteria: One canonical system of record for mitigations and closure evidence

5 EPIC 3 — Vulnerability Management

Discover → Triage → Remediate → Verify → Close

This epic documents the complete vulnerability management lifecycle from discovery through verified closure.
Dependencies: EPIC 0 Foundation

5.1 3.1 Vulnerability Management Value Stream Map

Deliverable: Value stream map with lead time and wait states (discovery → SLA start → fix → verify → close)
Primary Stakeholders: Engineering leadership, AppSec, GRC
Acceptance Criteria: Identifies top 3 bottlenecks with data sources for measurement

5.2 3.2 Current-State Vulnerability Management BPMN

Deliverable: As-Is BPMN including: deduplication, false positive handling, severity assignment, ticket creation, ownership assignment, remediation, verification, closure
Acceptance Criteria: Explicitly models re-open conditions and "won't fix" outcomes

5.3 3.3 Target-State Vulnerability Management BPMN (Automation-First)

Deliverable: To-Be BPMN with automation steps (auto-ticketing, auto-dedupe, SLAs, exception triggers)
Acceptance Criteria: Automated vs. manual steps clearly annotated

5.4 3.4 Severity and Prioritization Decision Model

Deliverable: Decision table: severity inputs (CVSS, exploitability, asset criticality, exposure) → priority and SLA
Acceptance Criteria: Approved by Engineering + GRC (or documented dissent + rationale)

5.5 3.5 Evidence & Audit Trail View (Vulnerability Closure)

Deliverable: Evidence flow diagram for closure: scan result → ticket → fix PR → deployment → rescan → closure record
Acceptance Criteria: For each closure state, required evidence is listed and retrievable

6 EPIC 4 — Exceptions and Risk Acceptance

Waivers

This epic covers the exception lifecycle including risk acceptance governance, compensating controls, and evidence retention.
Dependencies: EPIC 3 Prioritization Model (exceptions depend on severity/impact framing)

6.1 4.1 Exception Lifecycle BPMN

Deliverable: BPMN: request → justification → compensating controls → approval → expiry/review → revoke/renew
Primary Stakeholders: GRC, Product owners, AppSec, Engineering leadership
Acceptance Criteria: Every exception has: owner, scope, expiry date, review cadence, rollback plan

6.2 4.2 Risk Acceptance Governance View

Deliverable: Governance diagram: who can accept what risk, thresholds, escalation rules, required approvers
Notation: RACI + decision table
Acceptance Criteria: Clear separation between "AppSec recommends" vs. "business accepts"

6.3 4.3 Compensating Controls Catalog (Linked to Exceptions)

Deliverable: Catalog mapping common exceptions to compensating controls and monitoring requirements
Acceptance Criteria: Each compensating control maps to measurable signals or checks

6.4 4.4 Exception Evidence & Reporting View

Deliverable: Data-flow diagram showing exception records, approvals, and evidence retention
Acceptance Criteria: Audit can answer: "What exceptions exist right now and why?"

7 EPIC 5 — CI/CD Security Gates

Checks

This epic documents the CI/CD security gate architecture including policy enforcement, toolchain integration, and emergency override procedures.
Dependencies: EPIC 4 Exceptions (override policy and exception policy must align)

7.1 5.1 Secure CI/CD Gate Model (Policy-to-Pipeline)

Deliverable: Gate architecture diagram mapping required checks to pipeline stages (SAST/SCA/Secrets/IaC/Container as applicable)
Notation: Pipeline flow diagram + control mapping
Acceptance Criteria: Each gate has: purpose, pass/fail criteria, owner, override policy

7.2 5.2 Current-State CI/CD Gate BPMN

Deliverable: BPMN for "code change → build/test → security checks → decision → deploy"
Acceptance Criteria: Shows all outcomes: block, warn, create ticket, require approval

7.3 5.3 Target-State CI/CD Gate BPMN (Risk-Based Enforcement)

Deliverable: To-Be BPMN implementing: severity thresholds, repo criticality, branch protections, staged enforcement rollout
Acceptance Criteria: Includes a rollout plan path (monitor-only → warn → enforce)

7.4 5.4 Toolchain Component-and-Connector View

Deliverable: C4 L2/L3 showing connectors among: SCM, CI, scanners, artifact repo, ticketing, reporting, IAM
Acceptance Criteria: Every finding source has a defined ingestion path and deduplication strategy

7.5 5.5 Break-Glass / Override Workflow

Deliverable: BPMN for emergency override including approvals, logging, expiry, and post-incident review
Acceptance Criteria: Override events automatically generate a review item and are reportable

8 EPIC 6 — Reporting, Metrics, and Executive Visibility

Outcomes

This epic establishes the metrics framework, reporting cadence, and executive dashboards for AppSec program visibility.
Dependencies: EPIC 3 Evidence Model, EPIC 4 Exception Records, EPIC 5 Toolchain Flows

8.1 6.1 AppSec Metrics Tree (KPI/OKR Alignment)

Deliverable: KPI tree connecting operational metrics → risk outcomes → business outcomes
Primary Stakeholders: Executives/BOD, Engineering leadership, GRC
Acceptance Criteria: Every metric has an owner, source, and decision it supports

8.2 6.2 Scorecard + Cadence Map

Deliverable: Scorecard (weekly operational / monthly leadership / quarterly exec) + meeting/decision cadence diagram
Acceptance Criteria: Cadence includes: vulnerability backlog review, exception review, gate policy changes, major escalations

8.3 6.3 Risk Posture Dashboard Model

Deliverable: Dashboard wireframe + data lineage diagram (what data feeds what chart)
Acceptance Criteria: Defines "single source of truth" per metric and refresh frequency

8.4 6.4 Evidence Traceability End-to-End

Deliverable: End-to-end traceability map from controls → checks → artifacts → evidence store → reports
Acceptance Criteria: Supports audit questions without manual reconstruction

9 Cross-Epic "Beyond Views" Items

Reusable Artifacts Across All Epics

These artifacts are created once and referenced throughout all documentation packages. They ensure consistency, navigability, and maintainability of the architecture documentation.

9.1 BV-1: Diagram Index and Navigation

Deliverable: One index page linking each diagram to its stakeholder concerns and related policies/SOPs
Acceptance Criteria: A new team member can find "how intake works" in under 2 minutes

9.2 BV-2: Standards, Policies, and Control Traceability

Deliverable: Mapping: policy statements → process steps → automated checks → evidence artifacts
Acceptance Criteria: Each "must" statement has a verification method

9.3 BV-3: Change Log and Ownership

Deliverable: Diagram ownership + update cadence + change log
Acceptance Criteria: Every diagram has an accountable owner and a review date

10 Suggested Execution Order

The following execution order is recommended to deliver the fastest business value:

Phase	Epic	Rationale
1	**EPIC 0** — Foundation	Establishes shared vocabulary, boundaries, and stakeholder alignment
2	**EPIC 1** — Intake	Standardizes request handling and reduces friction
3	**EPIC 3** — Vulnerability Management	Core operational process with highest volume
4	**EPIC 4** — Exceptions	Governance for risk acceptance decisions
5	**EPIC 5** — CI/CD Gates	Automation and enforcement capabilities
6	**EPIC 2** — Threat Modeling	Best executed once intake is stable
7	**EPIC 6** — Reporting	Meaningful once data/evidence flows exist

Implementation Note

If this backlog needs to be converted to ticketable work items, each deliverable can be formatted as a Jira-ready user story with:

- Story points
- Assigned owners (by role)
- Explicit Definition of Ready (DoR) and Definition of Done (DoD) checklists

The Views-and-Beyond packaging structure should be preserved as epic-level organization.

Appendix A: Epic Dependency Map

Epic	Depends On	Enables
EPIC 0 (Foundation)	—	All subsequent epics
EPIC 1 (Intake)	EPIC 0.3, 0.4	EPIC 2
EPIC 2 (Threat Modeling)	EPIC 1	EPIC 6
EPIC 3 (Vulnerability Mgmt)	EPIC 0	EPIC 4, EPIC 6
EPIC 4 (Exceptions)	EPIC 3.4	EPIC 5
EPIC 5 (CI/CD Gates)	EPIC 4	EPIC 6
EPIC 6 (Reporting)	EPIC 3, 4, 5	—

Appendix B: Notation Quick Reference

Notation	Usage
C4 Level 1	System context diagrams showing boundaries and external actors
C4 Level 2/L3	Container and component diagrams for toolchain architecture
BPMN	Process workflows with swimlanes and decision gateways
DFD	Data flow diagrams for information and evidence views
DMN (lite)	Decision tables for routing and prioritization rules
RACI	Responsibility assignment matrices
Service Blueprint	Customer journey + frontstage/backstage mapping
Value Stream Map	Lead time analysis with wait states and bottlenecks