AI Infrastructure Security Playbook

Practical Controls for Securing AI Across the Risk Gradient

Aligned to ETSI EN 304 223 / UK Code of Practice for AI Cyber Security

---

V1.0 — February 2026

Risk-tiered · Capability-based · Exposure-driven

https://deepcyber.ai

Contents

1. Introduction

This playbook provides practical, checklist-driven infrastructure security controls for organisations deploying AI systems across a range of use cases, from embedded Copilot features through to distributed multi-agent architectures.

It is designed to be used by security architects, cloud platform teams, CISOs, and IT leadership to assess their AI deployments against a structured risk gradient and implement proportionate controls.

1.1 Alignment to ETSI EN 304 223 and the UK Code of Practice

ETSI EN 304 223 (Securing Artificial Intelligence: Baseline Cyber Security Requirements for AI Models and Systems) establishes 13 principles across 72 provisions, organised into five lifecycle phases: Secure Design, Secure Development, Secure Deployment, Secure Maintenance, and Secure End of Life. The companion standard ETSI EN 304 223 reinforces these as a European Norm.

The UK Code of Practice for AI Cyber Security, developed by DSIT and the NCSC, provided the foundational framework for ETSI EN 304 223. The 13 principles and five lifecycle phases are aligned between the two. This playbook translates those principles into infrastructure-specific controls, making them actionable for platform and security engineering teams.

1.2 Why an Infrastructure-Focused Approach?

ETSI EN 304 223 defines what should be true. This playbook defines what the platform team must build and enforce. The distinction matters because AI security controls are only effective when implemented at the infrastructure layer, not merely documented as policy. Prompt-level guardrails can be bypassed; network segmentation cannot.

This playbook covers the full spectrum of AI infrastructure, from SaaS configuration governance to multi-agent mesh security, using a consistent set of seven control dimensions applied at every risk level.

1.3 The Risk Gradient Model

The Risk Gradient (L1–L8) classifies AI deployments by their composite risk profile: capability, exposure, and governance maturity. It progresses from consumer-of-AI (L1) through orchestrator-of-AI (L5) to builder-of-AI (L6–L7) and system-of-systems (L8).

The gradient is descriptive, not prescriptive. Organisations assess their deployments against the baseline tier and then apply cross-cutting risk modifiers to determine the effective risk level and corresponding control set.

Level	Pattern	Examples	Primary Risk Driver
L1	Embedded AI	Copilot, Atlassian Intelligence	Data exposure, prompt injection, shadow AI
L2	AI-Assisted Development	Vibe coding, AI code gen	Unreviewed AI code in production
L3	Citizen Developer Agents	Power Automate, Agentforce	Over-permissioned connectors, workflow abuse
L4	Data Analytics & API Orchestration	LLM API pipelines, RAG	API misuse, credential leakage
L5	Custom Autonomous Agents	LangChain, CrewAI agents	Tool abuse, autonomy escalation
L6	Data Science & ML Pipelines	Fine-tuning, model training	Data poisoning, model supply chain
L7	Model Hosting & Serving	Inference endpoints	Model extraction, adversarial inputs
L8	Distributed / Multi-Agent	Agent meshes, swarms	Lateral compromise, systemic failure

Risk Gradient Overview

The eight levels progress from consumer-of-AI to system-of-systems, with infrastructure complexity and control requirements increasing at each step.

Level	Deployment Pattern	Assurance Zone
L8	Distributed / Multi-Agent Systems	HIGH ASSURANCE
L7	Model Hosting & Serving	↑
L6	Data Science & ML Pipelines	ADVANCED
L5	Custom Autonomous Agents	↑
L4	Data Analytics & API Orchestration	STANDARD
L3	Citizen Developer Agents	↑
L2	AI-Assisted Development	FOUNDATIONAL
L1	Embedded AI (Copilot, SaaS AI)	↑

Read top-to-bottom: L8 requires the most rigorous infrastructure controls; L1 the least. Assurance zones group tiers into Foundational, Standard, Advanced, and High Assurance bands.

1.4 How to Use This Playbook

This playbook is designed for multiple audiences. Use the guide below to find the fastest path to the content most relevant to your role.

Your Role	What to Do	Start Here
CISO / Security Leader	Understand the risk model, assess your AI estate against the gradient, and report posture to leadership.	Section 1.3 (Risk Gradient), then Section 2 (Modifiers), then Minimum Baseline table in Section 4A
Platform / Cloud Engineer	Use the tier checklists as implementation runlists. Start at your assessed risk level and work through each dimension.	Section 3 (Getting Started), then Section 5 (Checklists). Use the cloud-specific companion guide for your platform.
Security Architect	Review the full control model, map to your existing frameworks, and identify gaps across tiers.	Section 4 (Control Dimensions), then Section 8 (ETSI Mapping)
AI Product Owner	Classify your AI deployment against the risk gradient, apply modifiers, and communicate the required control set to your platform team.	Section 1.3 (Risk Gradient), then Section 2 (Modifiers)
GRC / Compliance	Map your obligations to ETSI EN 304 223 provisions and track coverage using CSPM scores and the playbook checklists.	Section 8 (ETSI Principle Reference), then Section 3 (Getting Started, Step 3: DeepCyber ETSI Agent)

Tip: All readers should begin with the Getting Started workflow (five steps) to build an AI inventory and establish a baseline before diving into detailed controls.

2. Cross-Cutting Risk Modifiers

The Risk Gradient provides the baseline. Cross-cutting factors act as modifiers that escalate (or occasionally de-escalate) the effective risk level, determining which control set applies. Each factor at High pushes the effective level up one gradient level. Two or more at High triggers a minimum of +2 levels and a formal risk assessment. If all factors are Low, an organisation may apply controls one level below baseline, but never below L1.

Factor	Low (no change)	Medium (no change)	High (+1 level)
Exposure	Internal, authenticated users	Partner-facing or broad internal	Public-facing or untrusted input
Data Sensitivity	Non-sensitive, public, synthetic	Business confidential, commercial	Personal, special category, regulated
Autonomy	Human approves every action	Human reviews, can intervene	Autonomous execution, chained actions
Integration Privilege	Read-only, sandboxed	Read-write to non-critical systems	Write to production, identity, financial
Physical-World Effect	Informational only	Influences decisions with financial/legal weight	Direct control, irreversible actions

2.1 Worked Example

A Power Automate agent (baseline L3) that is internal-only (Exposure: Low), processes employee absence data (Data Sensitivity: Medium), auto-executes without approval (Autonomy: High, +1), has write access to Dynamics 365 HR (Integration Privilege: High, +1), and triggers payroll adjustments (Physical-World Effect: High, +1). Effective risk level: L6. This citizen-built agent needs controls approaching those of an ML pipeline, despite containing no machine learning.

Modifier Escalation Logic

Apply modifiers to the baseline tier to determine the effective risk level and corresponding control set.

Step		Outcome	Note
1. Classify the AI deployment	→	Assign baseline tier (L1–L8)	
2. Assess each modifier (5 factors)	→	Rate as Low / Medium / High	
3. Count High-rated modifiers	→	Each High = +1 tier	Escalation rule
4. Two or more High modifiers	→	Minimum +2 tiers + formal risk assessment	Mandatory review
5. All modifiers Low	→	May de-escalate 1 tier (never below L1)	Optional

3. Getting Started

Before working through the detailed control checklists, follow these five steps to establish your baseline and create a prioritised action plan. This sequence moves from discovery through assessment to actionable remediation, ensuring you focus effort where risk is highest.

Step 1: Build Your AI Inventory

You cannot secure what you do not know exists. Begin by discovering and documenting every AI workload, tool, and service in use across your organisation. Shadow AI is often the largest unmanaged risk.

- ☐ Conduct a discovery exercise across all business units to identify every AI tool, service, and automation in use — including embedded features (Copilot, Gemini, Atlassian Intelligence), citizen developer automations, API integrations, and ML workloads
- ☐ Classify each AI workload against the Risk Gradient (L1–L8) using the tier definitions in the main playbook
- ☐ For each workload, assess the five cross-cutting risk modifiers (Exposure, Data Sensitivity, Autonomy, Integration Privilege, Physical-World Effect) to determine the effective risk level
- ☐ Document the data sensitivity classification for each workload: what data does it access, process, store, or generate? Include both designed access and potential access given current permissions
- ☐ Assign an owner to each AI workload who is accountable for its security posture
- ☐ Record all entries in a centralised AI Inventory (spreadsheet, CMDB, or dedicated AI governance tool) that is maintained as a living document

Step 2: Assess Your Current Security Baseline with CSPM

For existing AI workloads, your Cloud Security Posture Management (CSPM) tooling provides an immediate, evidence-based view of your current gaps. Enable Microsoft Defender for Cloud (Azure), AWS Security Hub with GuardDuty and Inspector (AWS), or Google Security Command Center Premium (GCP) and review findings across all environments hosting AI workloads.

- ☐ Review findings and ensure the following baseline controls are in place for every AI workload:
  - (a) Encryption at rest and in transit is enforced on all AI data stores, model artifacts, and communication channels
  - (b) Identity and access policies follow least privilege — no overly permissive roles on AI resources
  - (c) Network exposure is minimised — no unintended public endpoints on AI services
  - (d) Logging and monitoring are enabled for all AI workloads
  - (e) Vulnerability scanning is active on all compute, containers, and dependencies
- ☐ Record your CSPM secure score or compliance percentage as your starting baseline for AI workloads
- ☐ Export the CSPM findings for your AI workloads for use in Step 3

Step 3: Map Findings to ETSI EN 304 223 with the DeepCyber ETSI Agent

Raw CSPM findings tell you what is misconfigured, but not how it maps to AI-specific security requirements. The DeepCyber ETSI Agent bridges this gap by ingesting your CSPM findings, mapping them against ETSI EN 304 223 provisions, and generating a tailored, prioritised remediation checklist aligned to your assessed risk gradient level.

- ☐ Feed your exported CSPM findings into the DeepCyber ETSI Agent along with your AI Inventory and risk gradient assessments from Step 1
- ☐ The agent maps each finding to the relevant ETSI EN 304 223 principle and provision, identifies which risk gradient levels are affected, and prioritises remediation by risk severity and blast radius
- ☐ Review the generated checklist: it will highlight quick wins (controls that close multiple ETSI provisions simultaneously) and critical gaps (controls missing at your highest risk gradient levels)
- ☐ Use the prioritised output to create your remediation backlog, assigning owners and target dates to each action item

Step 4: Remediate by Priority

Work through the prioritised remediation checklist, starting with critical findings at your highest effective risk gradient levels. Focus on controls that deliver the greatest risk reduction per unit of effort.

- ☐ Address critical and high-severity findings first, prioritising workloads at the highest effective risk gradient level
- ☐ Target quick wins that close multiple ETSI provisions simultaneously: enforcing encryption (P6), enabling MFA (P6), activating logging (P12), and applying least-privilege access (P5, P6)
- ☐ Use the detailed control checklists in this playbook (and the relevant cloud implementation guide) as the reference for implementing each control
- ☐ Track remediation progress against your CSPM secure score and ETSI provision coverage percentage

Step 5: Establish Continuous Governance

AI security is not a one-time exercise. New AI workloads are deployed continuously, cloud configurations drift, and new threats emerge. Establish recurring processes to maintain your security posture over time.

- ☐ Schedule recurring CSPM reviews (minimum monthly) across all AI workload environments
- ☐ Re-run the DeepCyber ETSI Agent periodically (quarterly recommended) to reassess posture against ETSI provisions as your environment evolves
- ☐ Update the AI Inventory whenever new AI workloads are deployed, existing workloads change scope, or workloads are decommissioned
- ☐ Integrate AI workload onboarding into existing change management processes: require risk gradient assessment and security review before any new AI workload enters production
- ☐ Report AI security posture to leadership quarterly using CSPM scores, ETSI provision coverage, and remediation velocity as key metrics

4. The Seven Control Dimensions

Controls at each risk gradient level are organised into seven dimensions. These dimensions are consistent across all tiers, with the depth and rigour scaling to the assessed risk level. Each dimension maps to specific ETSI EN 304 223 principles as indicated.

Dimension	Scope	ETSI Principles
Governance	AI policies, risk assessments, training, human oversight, DPIA, decommissioning	P1, P3, P4, P10
Identity	Access control, authentication, credential management, key management, RBAC, JIT access	P4, P5, P6
Data	Classification, DLP, encryption, data provenance, data lineage, retention, disposal	P5, P8, P13
Processing	Compute isolation, pipeline security, CI/CD gates, guardrails, testing, environment separation	P2, P6, P9
Network	Segmentation, endpoint hardening, firewall rules, private endpoints, egress filtering	P2, P6
Supply Chain	Vendor assessment, dependency scanning, SBOM, artifact integrity, provenance verification	P7
Monitoring & Ops	SIEM, CSPM, alerting, vulnerability management, patching, incident response, posture management	P11, P12

Control Dimensions at a Glance

Each dimension applies at every tier. The table shows what changes as you move up the risk gradient.

Dimension	At L1 (Foundational)	At L8 (High Assurance)	ETSI
Governance	AI acceptable use policy, shadow AI audits	AI safety board, autonomous system risk review, kill-switch governance	P1, P3, P4
Identity	MFA, Conditional Access on licences	mTLS, capability-scoped agent identity, JIT with session tokens	P4, P5, P6
Data	DLP on prompts, data classification	Provenance tracking, lineage audit, cross-agent data flow controls	P5, P8, P13
Processing	Tenant config review, feature toggles	Agent sandboxing, capability tokens, tool-call enforcement	P2, P6, P9
Network	Existing segmentation sufficient	Service mesh, private endpoints, per-agent egress rules, circuit breakers	P2, P6
Supply Chain	Vendor DPA review	Model SBOM, artifact signing, training data provenance, dependency pinning	P7
Monitoring	CSPM, basic alerting	Real-time agent behaviour monitoring, cascade detection, anomaly ML	P11, P12

Same seven dimensions at every tier – radically different infrastructure controls. The detailed checklists specify every control.

4A. Minimum Baseline Controls by Tier

The following table summarises the minimum required infrastructure controls at each risk gradient level. Use this as a quick-reference before working through the detailed checklists. Controls are cumulative: each tier inherits all controls from lower tiers.

Tier	Pattern	Minimum Required Controls (cumulative)
L1	**Embedded AI**	MFA on all AI-enabled accounts, DLP integration with AI endpoints, AI acceptable use policy, CSPM enabled, prompt/response logging, shadow AI discovery, quarterly access reviews
L2	**AI-Assisted Dev**	+ Mandatory code review for AI-generated code, SAST/DAST in CI/CD, AI code provenance tracking, developer training on AI code risks, dependency scanning on AI-suggested packages
L3	**Citizen Agents**	+ Environment separation (dev/prod), connector permission reviews, DLP on low-code connectors, workflow approval gates, JIT access for agent service accounts, agent inventory register
L4	**API Orchestration**	+ API gateway with rate limiting, credential vault (no hardcoded keys), network segmentation for AI services, input validation on all API endpoints, egress filtering, API key rotation policy
L5	**Autonomous Agents**	+ Tool call allowlisting, agent sandboxing, memory TTL enforcement, human-in-the-loop for high-risk actions, permission boundaries preventing self-escalation, kill switches
L6	**ML Pipelines**	+ Data provenance and lineage tracking, artifact signing and integrity verification, training environment isolation, model registry access controls, pipeline audit logging, SBOM for model dependencies
L7	**Model Hosting**	+ Inference endpoint rate limiting, model extraction detection, adversarial input filtering, private endpoints (no public exposure), model versioning with rollback, A/B deployment gates
L8	**Multi-Agent**	+ mTLS between agents, capability-scoped identity per agent, trust boundary enforcement, circuit breakers, cascade failure detection, real-time agent behaviour monitoring, autonomous privilege escalation prevention

Note: "+" indicates controls added at this tier, in addition to all controls inherited from lower tiers. The detailed checklists in the following section provide the full implementation specification.

4B. Common AI Infrastructure Failure Patterns

AI infrastructure failures are rarely model failures – they are identity, network, and supply chain failures amplified by AI capability. The following patterns represent the most common infrastructure-level failures observed in AI deployments. Each maps directly to controls in the tier checklists that follow.

Failure Pattern	What Happens	Controls That Prevent It
Over-permissioned agent with production write access	A citizen-built Power Automate agent granted broad connector permissions writes directly to production HR or finance systems. No approval gate, no audit trail.	L3+: Governance, Identity
LLM API key committed to a public repository	An API key for a paid LLM service is hardcoded in source and pushed to GitHub. Automated scanners find it within minutes. The key has no spending cap or IP restriction.	L4+: Identity, Supply Chain
Fine-tuned model promoted without integrity verification	A model trained on sensitive data is promoted from staging to production with no cryptographic hash check. A tampered artifact enters the serving pipeline undetected.	L6+: Processing, Supply Chain
Inference endpoint exposed without rate limiting	A self-hosted LLM endpoint is deployed on a public subnet with no WAF, no rate limit, and no query pattern monitoring. Systematic extraction queries exfiltrate the model weights.	L7+: Network, Processing, Monitoring
Static service account shared across agent mesh	Multiple agents in a multi-agent system share a single service account with broad IAM permissions. One compromised agent inherits the access of all others.	L8: Identity, Governance
Cross-agent implicit trust enabling lateral compromise	Agents in a multi-agent system accept task delegations from any peer without verifying identity or capability. An injected prompt in one agent propagates through the mesh.	L8: Processing, Network, Identity
Shadow AI tool adopted without security review	A team enables an AI meeting transcription service and grants it access to calendar and email. No DPIA, no DLP, no vendor security review. Sensitive data flows to an unvetted third party.	L1+: Governance, Data
AI-generated code deployed without review	A developer uses AI code completion to generate database queries. The AI hallucinates a dependency and introduces an SQL injection vulnerability. No human review, no SAST scan.	L2+: Processing, Supply Chain

Every pattern above was preventable with controls already in this playbook. The tier checklists that follow provide the specific implementation steps.

5. Control Checklists by Risk Gradient Level

The following sections provide detailed, actionable checklists for each risk gradient level, organised by the seven control dimensions. Use these as assessment and implementation guides. Items marked with a checkbox (□) are actionable controls to implement and verify.

5.1 L1: Embedded AI

Example: Microsoft 365 Copilot, GitHub Copilot Chat, Atlassian Intelligence, Salesforce Einstein, Google Duet AI

Primary Risk: Data exposure, prompt injection, shadow AI proliferation

ETSI Lifecycle Focus: Primarily Deployment and Operation (P10, P11, P12)

Assurance Level: Foundational

Governance [P1, P3, P4, P10]

- □ Publish and enforce an AI Acceptable Use Policy covering all embedded AI features
- □ Maintain a register of all embedded AI services enabled across the organisation
- □ Conduct a Data Protection Impact Assessment (DPIA) where personal data may be processed by embedded AI
- □ Define prohibited use cases (e.g. pasting classified data into public Copilot) and communicate to all staff
- □ Assign an AI security owner responsible for embedded AI configuration governance
- □ Review ETSI P1 awareness requirements: deliver tailored prompt-hygiene and data-leakage training to all AI users
- □ Conduct periodic shadow AI audits to discover unapproved AI tool usage across the estate

Identity [P4, P5, P6]

- □ Enforce Conditional Access policies to control who can access AI features (e.g. Copilot licence assignment by group)
- □ Require MFA for all accounts with AI feature access
- □ Apply role-based access control (RBAC) to AI administrative settings
- □ Review and restrict AI feature availability per user role and data classification entitlement
- □ Audit AI feature activation logs quarterly to detect unauthorised enablement

Data [P5, P8, P13]

- □ Integrate Data Loss Prevention (DLP) policies with AI endpoints to block sensitive data in prompts
- □ Classify data accessible to embedded AI using your existing data classification scheme
- □ Verify data residency and processing location for each AI service (check vendor documentation)
- □ Enable prompt and response logging where the platform supports it for audit purposes
- □ Review Microsoft Purview / Google DLP / equivalent policies to ensure AI interactions are covered
- □ Confirm vendor data processing agreements (DPAs) are in place and reviewed
- □ Ensure AI features cannot access data above the user's own classification clearance

Processing (Apps, APIs, Integration) [P2, P6, P9]

- □ Review tenant-level AI feature configuration (e.g. M365 Copilot admin settings, Atlassian AI toggles)
- □ Disable AI features in tenants or workspaces where they are not approved
- □ Verify that AI features respect existing information barriers and ethical walls
- □ Ensure AI plugins and extensions are governed through an approval process before enabling
- □ Test that AI-generated outputs respect sensitivity labels applied to source documents

Network [P2, P6]

- □ Identify and document all network endpoints used by embedded AI services
- □ Apply firewall rules or web proxy policies to control access to AI service endpoints

- ☐ Block access to unapproved third-party AI services (shadow AI) at the network perimeter
- ☐ Verify TLS 1.2+ is enforced on all AI service connections
- ☐ Consider DNS-level filtering to prevent data exfiltration via unapproved AI endpoints

Supply Chain [P7]

- ☐ Conduct vendor security assessment of each embedded AI provider (SOC 2, ISO 27001, penetration test reports)
- ☐ Review vendor AI-specific security documentation (e.g. Microsoft AI security whitepaper, Atlassian Trust Centre)
- ☐ Verify sub-processor lists for AI data processing
- ☐ Monitor vendor security advisories and AI feature changelogs
- ☐ Ensure contractual right to audit and data deletion on termination
- ☐ Map ETSI P7 supply chain requirements to vendor assurance documentation

Monitoring and Operations [P11, P12]

- ☐ Aggregate AI interaction logs into SIEM (e.g. Microsoft Sentinel, Splunk, Google Chronicle)
- ☐ Enable Cloud Security Posture Management (CSPM): use Microsoft Defender for Cloud / AWS Security Hub / Google SCC to detect AI misconfigurations
- ☐ Set alerts for anomalous data volumes flowing to AI endpoints
- ☐ Monitor for bulk copy/paste or export actions involving AI-generated content
- ☐ Review AI usage reports monthly (e.g. Copilot usage analytics in M365 admin)
- ☐ Establish incident response procedures specific to AI data leakage events
- ☐ Track AI feature configuration drift using posture management tooling

5.2 L2: AI-Assisted Development (Vibe Coding)

Example: GitHub Copilot code completion, Cursor, Amazon CodeWhisperer, AI-generated code in IDEs, ChatGPT for code generation

Primary Risk: Unreviewed AI-generated code entering production, hallucinated dependencies, licence contamination

ETSI Lifecycle Focus: Primarily Design and Development (P2, P5, P7, P9)

Assurance Level: Foundational

Governance [P1, P3, P4, P10]

- ☐ Publish an AI Code Generation Policy defining when AI-generated code is permitted and the review requirements
- ☐ Mandate that all AI-generated code must pass human security review before merge to main branch
- ☐ Require developers to tag or flag AI-generated code in commit messages or PR descriptions
- ☐ Include AI code generation risks in the organisation's secure development lifecycle (SDLC) documentation
- ☐ Conduct developer training on AI-generated code risks: hallucinated packages, embedded secrets, vulnerable patterns
- ☐ Define acceptable AI coding tools and prohibit unapproved alternatives

Identity [P4, P5, P6]

- ☐ Control AI coding tool licences via identity group membership
- ☐ Enforce MFA on developer accounts with AI tool access
- ☐ Ensure AI coding tools authenticate via SSO (no standalone credentials)
- ☐ Restrict AI tool administrative settings to security/platform engineering teams

Data [P5, P8, P13]

- ☐ Configure AI coding tools to exclude sensitive repositories from context (e.g. secrets repos, compliance code)
- ☐ Audit whether AI tools are transmitting proprietary code to external services
- ☐ Review AI tool data retention policies: ensure code snippets are not retained for model training without consent
- ☐ Scan AI-generated code for hardcoded secrets, API keys, and credentials before commit
- ☐ Verify licence compliance: scan AI-generated code for copyleft or restricted licence patterns

Processing (Apps, APIs, Integration) [P2, P6, P9]

- ☐ Integrate SAST (Static Application Security Testing) gates in CI/CD that run on all code including AI-generated
- ☐ Integrate DAST (Dynamic Application Security Testing) in the pipeline
- ☐ Enable dependency scanning (e.g. Dependabot, Snyk, OWASP Dependency-Check) to catch hallucinated or vulnerable packages
- ☐ Enforce branch protection rules: no direct push to main, mandatory PR reviews
- ☐ Require minimum two reviewers for PRs containing AI-generated code (or flag AI-generated code for enhanced review)
- ☐ Run SCA (Software Composition Analysis) to validate all AI-suggested dependencies are legitimate and version-pinned

Network [P2, P6]

- ☐ Permit AI coding tool traffic only from approved developer networks or VPN-connected endpoints
- ☐ Block AI coding tool endpoints from production and CI/CD runners that do not require them
- ☐ Verify TLS encryption on all connections to AI coding services

Supply Chain [P7]

- ☐ Verify every AI-suggested dependency exists in the official package registry before installation

- ☐ Pin all dependency versions: do not accept unpinned or latest references from AI suggestions
- ☐ Maintain an SBOM (Software Bill of Materials) that includes AI-generated components
- ☐ Audit third-party AI coding tools for their own dependency and supply chain security (ETSI P7)
- ☐ Monitor for dependency confusion / typosquatting attacks targeting AI-suggested package names

Monitoring and Operations [P11, P12]

- ☐ Track the ratio of AI-generated vs human-written code per repository
- ☐ Monitor SAST findings specifically from AI-generated code to identify recurring vulnerability patterns
- ☐ Alert on introduction of dependencies not present in the organisation's approved dependency catalogue
- ☐ Review AI coding tool usage logs to detect misuse (e.g. exfiltrating code context)
- ☐ Include AI-generated code metrics in vulnerability management reporting and KPIs
- ☐ Run periodic retrospective security reviews of AI-generated code already in production

5.3 L3: Citizen Developer Agents

Example: Power Automate flows, Power Apps with AI Builder, Salesforce Agentforce, ServiceNow AI agents, Zapier AI automations

Primary Risk: Over-permissioned connectors, workflow abuse, unreviewed automation accessing production data, credential delegation

ETSI Lifecycle Focus: Development and Deployment (P4, P5, P6, P10)

Assurance Level: Standard

Governance [P1, P3, P4, P10]

- ☐ Publish a Citizen Developer Governance Policy covering agent creation, approval, and decommissioning
- ☐ Require all citizen-built agents to pass a security review before accessing production connectors
- ☐ Mandate environment separation: citizen developers build in sandbox, promote to production via approval gate
- ☐ Conduct citizen developer security training focused on connector permissions, least privilege, and data classification
- ☐ Maintain a central register of all citizen-built agents with owner, purpose, connectors, and data access scope
- ☐ Apply ETSI P4 human responsibility: mandate human approval steps in workflows that write, delete, or modify production data
- ☐ Define and enforce maximum permission scopes for citizen developer connectors (no global admin connectors)
- ☐ Conduct quarterly access reviews of all active citizen-built agents and their connector permissions

Identity [P4, P5, P6]

- ☐ Enforce service accounts or managed identities for agent execution rather than user delegated credentials where possible
- ☐ Apply Conditional Access policies to restrict agent execution to approved environments
- ☐ Require MFA for accounts used to create or administer agents
- ☐ Implement just-in-time (JIT) access for elevated connector permissions
- ☐ Ensure agent credentials are stored in a managed credential vault (e.g. Azure Key Vault, AWS Secrets Manager) with automatic rotation
- ☐ Audit connector credential usage: alert on credential use outside normal business hours or from unusual locations

Data [P5, P8, P13]

- ☐ Map all data flows for each agent: source systems, transformations, destination systems, data classifications
- ☐ Apply DLP policies to agent data flows (e.g. Power Platform DLP policies to block connectors combining business and personal data)
- ☐ Ensure agents cannot access data above the owner's classification clearance
- ☐ Encrypt data at rest and in transit within all automation flows
- ☐ Verify that agent execution logs capture data access events for audit trail
- ☐ Implement data masking or tokenisation for agents processing sensitive data in non-production environments

Processing (Apps, APIs, Integration) [P2, P6, P9]

- ☐ Enforce environment separation: sandbox, test, production with distinct connector configurations
- ☐ Implement API gateway controls between citizen agents and backend systems (no direct database access)
- ☐ Apply rate limiting on agent-to-backend API calls to prevent runaway automations
- ☐ Deploy runtime monitoring of flow execution (e.g. Power Platform Analytics, Flow run history retention)

- ☐ Enforce approval workflows as mandatory pipeline steps for flows that trigger financial, HR, or customer-facing actions
- ☐ Test flows against over-privilege scenarios: what happens if the connector has more access than intended?
- ☐ Block or restrict premium connectors (e.g. HTTP, custom connectors) to approved users only

Network [P2, P6]

- ☐ Segment citizen developer execution environments from production networks
- ☐ Route agent-to-backend traffic through API gateways or service meshes, not direct connections
- ☐ Apply network-level controls to restrict which backend systems agents can reach
- ☐ Enforce private endpoints for connectors accessing sensitive systems (e.g. Azure Private Link)
- ☐ Block outbound internet access from agent runtimes unless explicitly required and approved

Supply Chain [P7]

- ☐ Vet and approve all third-party connectors and templates before making them available to citizen developers
- ☐ Maintain a catalogue of approved connectors with security ratings
- ☐ Monitor connector marketplace for malicious or vulnerable connectors (ETSI P7)
- ☐ Require connector publisher verification before enabling custom connectors
- ☐ Document all third-party dependencies in citizen-built agents

Monitoring and Operations [P11, P12]

- ☐ Deploy CSPM (e.g. Microsoft Defender for Cloud Apps) to monitor Power Platform / Salesforce agent activity
- ☐ Alert on new connector additions, especially premium or HTTP connectors
- ☐ Monitor agent execution frequency and data volumes: alert on spikes indicating runaway or abused flows
- ☐ Track failed execution attempts which may indicate permission issues or adversarial probing
- ☐ Log all agent CRUD operations (create, update, delete) with user identity and timestamp
- ☐ Integrate agent monitoring into existing SIEM and SOC workflows
- ☐ Conduct periodic dormant agent audits: identify and decommission unused flows with active credentials

5.4 L4: Data Analytics and API Orchestration

Example: Python scripts calling LLM APIs for report summarisation, data transformation pipelines using OpenAI/Claude APIs, retrieval-augmented generation (RAG) pipelines, business intelligence AI integrations

Primary Risk: API key leakage, data transformation errors, credential exposure, excessive API permissions, data exfiltration via API

ETSI Lifecycle Focus: Development and Operation (P5, P6, P8, P9)

Assurance Level: Standard

Governance [P1, P3, P4, P10]

- ☐ Publish an API Usage Policy for AI services covering approved providers, data classification restrictions, and cost controls
- ☐ Require threat modelling for any pipeline that processes business confidential or personal data via external AI APIs
- ☐ Mandate code review for all API orchestration scripts and pipelines before production deployment
- ☐ Document all AI API dependencies, data flows, and transformation logic under version control
- ☐ Conduct ETSI P3 risk assessment covering API abuse, data exfiltration, and transformation integrity scenarios
- ☐ Define cost alerting thresholds to detect API abuse or runaway consumption

Identity [P4, P5, P6]

- ☐ Store all API keys and tokens in a cloud key management service (Azure Key Vault, AWS Secrets Manager, GCP Secret Manager)
- ☐ Enforce automatic API key rotation on a defined schedule (minimum quarterly, ideally monthly)
- ☐ Apply least-privilege scoping to API keys: restrict to specific models, endpoints, and rate limits
- ☐ Never embed API keys in source code, environment variables on shared systems, or CI/CD logs
- ☐ Use managed identities or workload identity federation where supported instead of static API keys
- ☐ Audit API key usage: alert on keys used from unexpected IP ranges or at unusual times

Data [P5, P8, P13]

- ☐ Classify all data entering and exiting API orchestration pipelines
- ☐ Apply data minimisation: send only the minimum necessary data to external AI APIs
- ☐ Verify AI API provider data retention policies: ensure prompts and responses are not retained for training
- ☐ Encrypt all data in transit (TLS 1.2+) and at rest within pipeline storage
- ☐ Implement input validation and output sanitisation for all AI API interactions
- ☐ Log all API requests and responses (or representative samples) for audit trail
- ☐ Apply data transformation validation: verify outputs against expected schemas and business rules

Processing (Apps, APIs, Integration) [P2, P6, P9]

- ☐ Deploy API gateway (e.g. Azure API Management, AWS API Gateway, Kong) as the single entry point for all AI API calls
- ☐ Enforce rate limiting, throttling, and quota management at the API gateway
- ☐ Apply schema validation on API requests and responses at the gateway layer
- ☐ Implement retry logic with exponential backoff and circuit breakers to prevent cascading failures
- ☐ Separate development, staging, and production API environments with distinct credentials
- ☐ Run API fuzzing and negative testing as part of CI/CD pipeline
- ☐ Version all pipeline code and transformation logic in Git with branch protection

Network [P2, P6]

- ☐ Route all AI API traffic through the API gateway: no direct-to-provider calls from application code
- ☐ Apply network segmentation: isolate API orchestration runtimes from other workloads

- ☐ Use private endpoints where AI providers support them (e.g. Azure OpenAI private endpoint)
- ☐ Implement egress filtering: restrict outbound connections to approved AI API endpoints only
- ☐ Log and monitor all network traffic to/from AI API endpoints

Supply Chain [P7]

- ☐ Conduct security assessment of each AI API provider (SOC 2, ISO 27001, data processing terms)
- ☐ Scan all pipeline dependencies (Python packages, npm modules) for known vulnerabilities
- ☐ Pin all dependency versions and use lock files (requirements.txt, package-lock.json)
- ☐ Monitor for AI provider API changes, deprecations, and security advisories
- ☐ Maintain an SBOM for all pipeline components including AI SDKs and client libraries (ETSI P7)

Monitoring and Operations [P11, P12]

- ☐ Monitor API call patterns: alert on unusual volumes, error rate spikes, or latency changes
- ☐ Track API costs in real time: set budget alerts at 50%, 80%, 100% thresholds
- ☐ Log credential usage events from key vault and alert on anomalies
- ☐ Integrate pipeline health monitoring into SIEM/SOC dashboards
- ☐ Run vulnerability scanning on pipeline infrastructure and dependencies on a regular schedule
- ☐ Enable CSPM for the cloud accounts hosting API orchestration workloads
- ☐ Implement alerting for data transformation failures or unexpected output patterns

5.5 L5: Custom Autonomous Agents

Example: LangChain/LangGraph agents, AutoGPT-style systems, custom tool-calling agents built on Claude/GPT-4 APIs, agent frameworks (CrewAI, Semantic Kernel agents) with memory and tool access

Primary Risk: Tool abuse, autonomy escalation, prompt injection chains, memory poisoning, uncontrolled tool-calling

ETSI Lifecycle Focus: Full lifecycle (all principles apply with significant depth)

Assurance Level: Advanced

> *Controls marked ★ are advanced maturity targets. Prioritise unmarked controls first for initial compliance.*

Governance [P1, P3, P4, P10]

- □ Require formal threat modelling (STRIDE adapted for AI) before any autonomous agent enters production
- □ Mandate graduated autonomy tiers: define which actions agents can take autonomously vs requiring human approval
- □ Publish an Agent Security Standard defining tool-call permission boundaries, memory policies, and escalation paths
- □ Conduct ETSI P3 risk assessment covering prompt injection chains, tool escalation, and memory manipulation
- □ Require security sign-off for any agent granted write access to production systems
- □ Define and enforce agent scope boundaries: agents must not be able to expand their own permissions
- □ Establish an agent decommissioning procedure including credential revocation, memory purge, and access removal
- □ ★ Conduct red-team exercises against agents before production deployment (ETSI P9)

Identity [P4, P5, P6]

- □ Assign each agent a unique managed identity with least-privilege permissions scoped to approved tools only
- □ Enforce tool-call permissions at the infrastructure layer, not just in the agent prompt or system message
- □ ★ Implement capability-based access control: agents receive explicit capability tokens for each tool
- □ Require human approval (break-glass) for any agent tool call classified as high-privilege (e.g. delete, send email, modify permissions)
- □ Store agent credentials in cloud KMS with automatic rotation
- □ Audit all agent identity and credential usage events
- □ ★ Implement session-scoped tokens that expire and cannot be reused across agent invocations

Data [P5, P8, P13]

- □ Encrypt agent memory stores at rest and in transit (AES-256 / TLS 1.2+)
- □ Apply access controls on agent memory: agents should only read their own memory, not other agents' memory
- □ Implement memory hygiene: set TTL (time-to-live) on memory entries, purge stale data
- □ Classify data in agent context windows and enforce DLP rules on agent outputs
- □ Log all data accessed, generated, and stored by agents for audit trail
- □ Implement input sanitisation for all data entering agent context to mitigate prompt injection
- □ Apply output filtering to prevent agents from leaking sensitive data in responses

Processing (Apps, APIs, Integration) [P2, P6, P9]

- □ Execute agents in sandboxed environments (containers, VMs) with restricted system access
- □ Enforce tool-call allowlists at the infrastructure layer: agents cannot call tools not on the approved list
- □ Implement rate limiting on agent tool calls to prevent runaway execution

- ☐ ★ Deploy circuit breakers that halt agent execution if error rates or tool-call volumes exceed thresholds
- ☐ Version all agent configurations, system prompts, and tool definitions in Git
- ☐ Implement guardrails (input/output filters) as infrastructure components, not just prompt instructions
- ☐ Test agents against adversarial prompt injection, jailbreak attempts, and tool abuse scenarios (ETSI P9)
- ☐ Deploy agent execution in isolated compute with no implicit network access to unapproved systems

Network [P2, P6]

- ☐ Apply network policies (e.g. Kubernetes NetworkPolicy, security groups) restricting agent runtime to approved endpoints only
- ☐ Block all outbound internet access from agent runtimes unless explicitly allowlisted
- ☐ Route agent-to-tool communication through authenticated API gateways
- ☐ ★ Implement network-level kill switches that can immediately isolate a compromised agent
- ☐ Log and monitor all network traffic from agent execution environments

Supply Chain [P7]

- ☐ Verify provenance of agent frameworks (LangChain, CrewAI, etc.) and pin versions
- ☐ Scan agent framework dependencies for known vulnerabilities before deployment
- ☐ Audit prompt templates and system prompts sourced from external repositories
- ☐ Verify integrity of external tools and APIs consumed by agents
- ☐ Maintain SBOM covering all agent components including frameworks, tools, and model API clients (ETSI P7)

Monitoring and Operations [P11, P12]

- ☐ Log all agent actions with full context: tool calls, parameters, responses, timestamps, and user/session identity
- ☐ Establish behavioural baselines for each agent and alert on deviations (unusual tool calls, frequency spikes)
- ☐ Monitor agent memory store growth and alert on unexpected expansion
- ☐ Integrate agent action logs into SIEM with correlation rules for prompt injection and tool abuse patterns
- ☐ Enable CSPM for all cloud accounts hosting agent infrastructure
- ☐ ★ Deploy canary inputs periodically to verify agents are behaving as expected
- ☐ Implement real-time alerting on high-privilege tool calls (even if approved) for SOC visibility
- ☐ Conduct post-incident forensic capability: ensure agent execution logs are retained and tamper-evident

5.6 L6: Data Science, Fine-Tuning and ML Pipelines

Example: Fine-tuning LLMs on proprietary data, training custom ML models, feature engineering pipelines, MLflow/Weights & Biases experiment tracking, Jupyter notebook environments, Databricks/SageMaker ML workloads

Primary Risk: Data poisoning, model supply chain compromise, training data leakage, experiment environment abuse, bias injection, model theft

ETSI Lifecycle Focus: Primarily Design and Development (P2, P3, P5, P6, P7, P8, P9) with P4 for model promotion

Assurance Level: Advanced

> *Controls marked ★ are advanced maturity targets. Prioritise unmarked controls first for initial compliance.*

Governance [P1, P3, P4, P10]

- ☐ Require formal threat modelling covering training-time attacks (data poisoning, backdoor injection) for all ML projects
- ☐ Mandate model risk assessment before any model enters production (ETSI P3)
- ☐ Publish a Model Lifecycle Governance Policy covering training, validation, promotion, and retirement
- ☐ Require human review and sign-off for model promotion from staging to production (ETSI P4)
- ☐ Maintain model cards for all production models documenting purpose, training data, limitations, and known biases
- ☐ Conduct DPIA where models are trained on personal data
- ☐ Define model retraining governance: triggers, approval process, and validation requirements
- ☐ ★ Establish responsible AI review covering fairness, bias, and safety for all production models
- ☐ Map ETSI P8 documentation requirements to experiment tracking and model registry metadata

Identity [P4, P5, P6]

- ☐ Enforce RBAC on ML platforms: separate roles for data scientists, ML engineers, and platform admins
- ☐ Apply least-privilege access to training data stores, feature stores, and model registries
- ☐ Use managed identities for ML pipeline execution (not shared service accounts)
- ☐ Implement JIT access for GPU/TPU cluster provisioning
- ☐ Restrict model registry write access to approved CI/CD pipelines only (no manual uploads)
- ☐ Audit all access to training data, model artifacts, and experiment tracking systems
- ☐ Enforce MFA on all ML platform accounts
- ☐ Implement cryptographic signing of model artifacts for integrity verification (ETSI Provision 5.2.4-1.2)

Data [P5, P8, P13]

- ☐ Implement training data provenance tracking: document source, collection method, consent basis, and transformations for every dataset
- ☐ Apply data classification to all training datasets and enforce access controls accordingly
- ☐ Isolate training data from production inference environments (ETSI Provision 5.2.2-3)
- ☐ Encrypt training data at rest (AES-256) and in transit (TLS 1.2+)
- ☐ Use cloud KMS (Azure Key Vault, AWS KMS, GCP Cloud KMS) for encryption key management with automatic rotation
- ☐ Implement data versioning (e.g. DVC, LakeFS) with immutable audit trail
- ☐ ★ Scan training data for poisoning indicators: statistical anomalies, outliers, adversarial examples
- ☐ Apply data sanitisation to remove PII from training datasets where not explicitly required
- ☐ Verify feature store data lineage end-to-end from source to feature to model
- ☐ Implement data retention policies: purge training data when no longer needed

Processing (Apps, APIs, Integration) [P2, P6, P9]

- ☐ Provision dedicated, isolated training compute environments separate from production (ETSI Provision 5.2.2-3)

- ☐ Enforce environment separation: development notebooks, training pipelines, staging, and production inference are distinct environments
- ☐ Implement immutable pipeline definitions (pipeline-as-code in Git) with mandatory PR review
- ☐ Enforce compute quotas and cost controls on training workloads to detect misuse (e.g. cryptomining)
- ☐ Run security scanning on ML library dependencies (TensorFlow, PyTorch, HuggingFace, etc.)
- ☐ Apply container image scanning for all training and inference containers
- ☐ Implement experiment reproducibility: log all hyperparameters, data versions, code versions, and random seeds
- ☐ ★ Deploy model validation gates in CI/CD: accuracy, bias, adversarial robustness, and security tests must pass before promotion
- ☐ ★ Test models against adversarial inputs, data poisoning detection, and model extraction resistance (ETSI P9)

Network [P2, P6]

- ☐ Isolate training environments on dedicated VPCs/VNets with restricted internet access
- ☐ Apply network segmentation between training, staging, and production inference environments
- ☐ Restrict egress from training environments to approved package registries and data sources only
- ☐ Use private endpoints for model registries, feature stores, and data lakes
- ☐ Block GPU/TPU cluster access from the public internet
- ☐ Implement network-level logging for all training environment traffic

Supply Chain [P7]

- ☐ Verify provenance of pre-trained base models: document origin, licence, and known vulnerabilities
- ☐ Implement cryptographic hash verification of model artifacts at every transfer point (ETSI Provision 5.2.4-1.2)
- ☐ Scan all ML library dependencies for known vulnerabilities before use
- ☐ Maintain SBOM covering ML frameworks, data processing libraries, and model serving components
- ☐ Monitor ML framework security advisories (PyTorch, TensorFlow, HuggingFace, etc.)
- ☐ Vet and verify all third-party datasets used for training or fine-tuning
- ☐ Implement secure artifact storage for model weights, checkpoints, and training snapshots
- ☐ Apply integrity verification when loading models into serving infrastructure

Monitoring and Operations [P11, P12]

- ☐ Monitor training pipelines for data drift, loss anomalies, and resource utilisation spikes
- ☐ Alert on unexpected compute resource patterns (may indicate cryptomining or unauthorised workloads)
- ☐ Track model performance metrics in staging and alert on degradation before production promotion
- ☐ Enable CSPM (Defender for Cloud, Security Hub) for all cloud accounts hosting ML workloads
- ☐ Implement vulnerability scanning for all ML platform infrastructure and containers
- ☐ Log all model registry operations (upload, download, delete, promote) with user identity
- ☐ ★ Monitor feature store access patterns for anomalous queries
- ☐ Run periodic model integrity checks: verify deployed model matches approved registry artifact (hash comparison)
- ☐ Integrate ML platform monitoring into SIEM and SOC workflows

5.7 L7: Model Hosting and Serving

Example: Self-hosted LLM inference (vLLM, TGI, Triton), Azure ML managed endpoints, SageMaker endpoints, custom inference APIs, on-premises model serving infrastructure

Primary Risk: Model extraction via systematic querying, inference abuse, adversarial inputs, denial of service, model theft, output manipulation

ETSI Lifecycle Focus: Primarily Deployment and Maintenance (P6, P10, P11, P12) with P7 for serving supply chain

Assurance Level: High Assurance

> *Controls marked ★ are high-assurance targets suited to defence, critical infrastructure, and regulated sectors. Implement unmarked controls first.*

Governance [P1, P3, P4, P10]

- ☐ Publish a Model Serving Security Standard covering endpoint hardening, access control, and monitoring requirements
- ☐ Require security review and sign-off before any model endpoint is exposed (even internally)
- ☐ Implement a model promotion pipeline with mandatory security gates before production serving
- ☐ Define and enforce SLAs for model endpoint availability, latency, and security incident response
- ☐ Establish rate-limiting policies designed to impede model extraction attacks (ETSI Provision 5.2.2-2)
- ☐ Maintain inference endpoint inventory with access policies, model versions, and consumer documentation
- ☐ Define rollback procedures: ability to revert to previous model version within defined RTO

Identity [P4, P5, P6]

- ☐ Authenticate all inference endpoint consumers via API key, OAuth token, or mTLS certificate
- ☐ Apply per-consumer rate limiting and quota management
- ☐ Enforce least-privilege access: consumers only access specific model endpoints they are authorised for
- ☐ Store inference API credentials in cloud KMS with automatic rotation
- ☐ Audit all inference endpoint access events including consumer identity, query volume, and error rates
- ☐ Implement anomaly detection on consumer query patterns to identify extraction attempts

Data [P5, P8, P13]

- ☐ Apply input validation on all inference requests: reject malformed, oversized, or adversarial inputs
- ☐ Implement output filtering to prevent sensitive data leakage in model responses
- ☐ Encrypt inference traffic end-to-end (TLS 1.2+ minimum, mTLS where feasible)
- ☐ Log inference requests and responses (or representative samples) for audit and forensics
- ☐ Implement PII detection on model outputs where models may generate personal data
- ☐ Apply content filtering on model outputs to detect harmful, biased, or policy-violating content

Processing (Apps, APIs, Integration) [P2, P6, P9]

- ☐ Harden inference endpoints: rate limiting, request size limits, timeout enforcement
- ☐ Verify model artifact integrity before loading into serving infrastructure (cryptographic hash check)
- ☐ ★ Implement canary deployments for model updates: route small percentage of traffic to new model before full rollout
- ☐ ★ Deploy blue-green infrastructure enabling zero-downtime model updates and instant rollback
- ☐ Isolate model serving instances: one model per container/pod or robust multi-tenant isolation
- ☐ Apply container security: read-only file systems, non-root execution, minimal base images
- ☐ Implement inference-time guardrails as separate infrastructure components (not embedded in the model)
- ☐ Test endpoints against adversarial inputs, model extraction probes, and denial-of-service scenarios (ETSI P9)

Network [P2, P6]

- □ Place inference endpoints behind load balancers with DDoS protection
- □ Apply WAF (Web Application Firewall) rules on inference endpoints to filter malicious requests
- □ Implement network segmentation: inference endpoints on dedicated subnets/security groups
- □ Use private endpoints for internal model consumers; API gateway for external consumers
- □ Monitor and log all network traffic to/from inference infrastructure
- □ Implement egress controls: inference containers should have no outbound internet access unless required

Supply Chain [P7]

- □ Verify model artifact provenance: confirm the model being served is the model that was approved in the registry
- □ Scan inference container images for vulnerabilities before deployment
- □ Pin and verify all serving framework versions (vLLM, TGI, Triton, etc.)
- □ Monitor serving framework security advisories and apply patches promptly
- □ Implement container image signing and verification in the deployment pipeline

Monitoring and Operations [P11, P12]

- □ Monitor inference latency, error rates, and throughput in real time with dashboards (Grafana, CloudWatch, Azure Monitor)
- □ ★ Implement output distribution monitoring: alert on shifts indicating model drift or manipulation
- □ ★ Deploy query pattern analysis to detect model extraction attempts (high-volume systematic queries)
- □ ★ Send canary queries periodically and alert if outputs deviate from expected baselines
- □ Enable CSPM for all cloud accounts hosting inference infrastructure
- □ Integrate inference monitoring into SIEM with correlation rules for abuse patterns
- □ Implement alerting for model serving failures, crash loops, and resource exhaustion
- □ Conduct vulnerability scanning on all serving infrastructure on a regular schedule
- □ Track model version deployed vs model version approved to detect supply chain tampering

5.8 L8: Distributed and Multi-Agent Systems

Example: Multi-agent orchestration (AutoGen, CrewAI multi-agent), agent-to-agent task delegation, agent meshes and swarms, MCP server architectures, A2A protocol implementations

Primary Risk: Lateral agent compromise, emergent behaviour, systemic cascading failure, agent impersonation, trust delegation abuse

ETSI Lifecycle Focus: Full lifecycle (all principles apply at maximum depth)

Assurance Level: High Assurance

> *Controls marked ★ are high-assurance targets suited to defence, critical infrastructure, and regulated sectors. Implement unmarked controls first.*

Governance [P1, P3, P4, P10]

- □ Require formal system-level threat modelling covering inter-agent trust, lateral movement, and cascading failure before deployment
- □ Publish a Multi-Agent Security Architecture Standard defining trust boundaries, communication protocols, and failure domains
- □ Mandate zero-trust principles: no implicit trust between agents regardless of network location
- □ Define blast radius boundaries: a compromised agent must not be able to affect agents in other trust domains
- □ Conduct multi-agent red team exercises simulating agent compromise, impersonation, and cascade scenarios (ETSI P9)
- □ Establish system-wide kill switches and circuit breakers at the infrastructure layer
- □ ★ Define governance for emergent behaviour: who is responsible when agent interactions produce unexpected outcomes?
- □ ★ Implement a formal agent registration and deregistration authority
- □ Map all 13 ETSI principles at maximum depth for multi-agent deployments

Identity [P4, P5, P6]

- □ Assign each agent a cryptographically verifiable unique identity (certificate, signed token)
- □ Implement mTLS for all inter-agent communication
- □ Apply capability-based access control: agents present capability tokens to peer agents for each interaction
- □ Prevent trust delegation abuse: Agent A cannot pass its credentials to Agent B to act on its behalf without explicit authorisation
- □ Implement agent identity rotation and revocation mechanisms
- □ Audit all inter-agent authentication events and trust delegation chains
- □ ★ Deploy an agent registration authority that validates agent identity before mesh participation

Data [P5, P8, P13]

- □ Encrypt all inter-agent data exchanges (mTLS + payload encryption where warranted)
- □ Apply data classification to inter-agent messages and enforce access controls
- □ Prevent data leakage between trust domains: agents should not share data across trust boundaries without policy enforcement
- □ Implement signed messages to ensure data integrity and non-repudiation between agents
- □ Log all inter-agent data exchanges for audit trail and forensic capability
- □ Apply output validation on agent-generated tasks before delegation to peer agents

Processing (Apps, APIs, Integration) [P2, P6, P9]

- □ Deploy centralised orchestration plane with security monitoring and human override capability
- □ Implement circuit breakers at the infrastructure layer that halt agent interactions when error rates or cascading indicators are detected
- □ Enforce agent isolation: each agent runs in its own sandboxed environment with no shared state except through defined APIs
- □ Deploy health checking across the agent mesh: detect and isolate unhealthy agents automatically

- ☐ Implement rate limiting on inter-agent communication to prevent message flooding
- ☐ Version all agent configurations and inter-agent protocol definitions in Git
- ☐ ★ Test for emergent behaviour: run simulation environments with adversarial agents to identify unexpected interaction patterns
- ☐ Implement graceful degradation: system must remain safe even if individual agents fail or are compromised

Network [P2, P6]

- ☐ Apply network policies preventing lateral movement between agent instances in different trust domains
- ☐ Enforce mTLS on all inter-agent communication channels
- ☐ Segment agent communication networks from broader infrastructure
- ☐ Implement network-level circuit breakers and kill switches that can isolate agent groups
- ☐ Deploy distributed denial-of-service protection on inter-agent communication infrastructure
- ☐ Log and monitor all inter-agent network traffic with anomaly detection

Supply Chain [P7]

- ☐ Verify identity and integrity of all peer agents before accepting communication (agent attestation)
- ☐ ★ Implement an agent registration authority that validates agent provenance before mesh admission
- ☐ Scan and verify all agent framework dependencies across the entire mesh
- ☐ Monitor for compromised or rogue agents joining the mesh
- ☐ Maintain SBOM covering all agents, frameworks, tools, and communication infrastructure (ETSI P7)

Monitoring and Operations [P11, P12]

- ☐ Deploy system-level monitoring across the agent network: inter-agent communication patterns, message volumes, error rates
- ☐ Implement cascade detection: alert when failures in one agent propagate to downstream agents
- ☐ Monitor inter-agent trust delegation chains for anomalous patterns
- ☐ ★ Deploy mesh health dashboards showing agent status, communication topology, and trust relationships in real time
- ☐ Integrate multi-agent monitoring into SIEM with correlation rules for lateral movement and compromise patterns
- ☐ Enable CSPM for all cloud accounts hosting multi-agent infrastructure
- ☐ Implement forensic logging: retain full interaction history for post-incident analysis
- ☐ ★ Conduct periodic chaos engineering exercises: inject failures and compromised agents to validate resilience
- ☐ Track agent population: alert on unexpected agent registration or deregistration events

6. Recommended Posture Management and Tooling

The following tools and capabilities support implementation across the seven control dimensions. These are representative, not exhaustive; equivalent tooling from other vendors serves the same purpose.

6.1 Cloud Security Posture Management (CSPM)

- Microsoft Defender for Cloud: Security posture scoring, regulatory compliance dashboards (CIS, NIST, PCI DSS), AI workload recommendations, container scanning
- AWS Security Hub: Aggregated findings from GuardDuty, Inspector, Macie; automated compliance checks; integration with SageMaker security recommendations
- Google Security Command Center (SCC): Asset inventory, vulnerability scanning, compliance monitoring for GCP AI workloads
- Enable continuous compliance monitoring against ETSI EN 304 223 control mappings using custom policy definitions in these tools

6.2 Encryption and Key Management

- Azure Key Vault / AWS KMS / GCP Cloud KMS for all encryption key management
- Enforce encryption at rest (AES-256) for all AI data stores: training data, model artifacts, memory stores, vector databases
- Enforce TLS 1.2+ for all data in transit including inter-service, AI API, and inter-agent communication
- Implement automatic key rotation (minimum quarterly for API keys, annually for encryption keys)
- Use customer-managed keys (CMK) for high-sensitivity workloads (L5+)
- Implement cryptographic signing of model artifacts for integrity verification (ETSI Provision 5.2.4-1.2)

6.3 Vulnerability Management and Patching

- Deploy container image scanning (Trivy, Snyk Container, Defender for Containers) across all AI workloads
- Run SAST/DAST on all AI application code, agent code, and pipeline definitions
- Scan ML library dependencies (PyTorch, TensorFlow, LangChain, etc.) for known CVEs
- Implement automated patching for OS, container base images, and framework dependencies
- Define patch SLAs: Critical within 48 hours, High within 7 days, Medium within 30 days
- Track vulnerability remediation KPIs per risk gradient level

6.4 Identity and Access Management

- Enforce MFA on all accounts with AI system access (non-negotiable at all tiers)
- Use managed identities / workload identity federation instead of static credentials where possible
- Implement Privileged Identity Management (PIM / JIT access) for administrative access to AI platforms
- Deploy secrets scanning in CI/CD pipelines (e.g. GitLeaks, TruffleHog) to detect leaked API keys
- Centralise secret management in cloud KMS: no secrets in code, environment variables, or CI/CD logs

6.5 Monitoring, Alerting and SIEM Integration

- Aggregate all AI system logs into centralised SIEM (Sentinel, Splunk, Chronicle, Elastic)
- Deploy AI-specific detection rules: prompt injection patterns, unusual API call volumes, model extraction indicators
- Implement real-time alerting for: credential misuse, DLP policy violations, agent behavioural anomalies, infrastructure misconfigurations
- Enable audit logging on all AI platforms, model registries, and agent execution environments

- Integrate CSPM findings into SOC workflows with defined triage and response procedures
- Deploy canary mechanisms: canary queries for inference endpoints, canary inputs for agents
- Retain logs for minimum 12 months for forensic and compliance purposes

6.6 Data Protection

- Microsoft Purview / AWS Macie / Google DLP for data classification and DLP policy enforcement
- Deploy sensitivity labels across all data stores accessible to AI systems
- Implement data lineage tracking for ML pipelines (e.g. Apache Atlas, Collibra, OpenLineage)
- Apply data masking / tokenisation in non-production AI environments
- Implement data retention and disposal policies aligned with ETSI P13

7. Cross-Cloud Control Alignment

AI Infrastructure Security Playbooks — With ETSI EN 304 223 Mapping

This table maps cloud-neutral security invariants to specific AWS, Azure, and GCP implementations across all four assurance zones. Controls are aligned to the playbook's eight-tier risk gradient (L1–L8) and mapped to ETSI EN 304 223 provisions.

Zone	Security Invariant	AWS	Azure	GCP	ETSI EN 304 223
FOUNDATIONAL (L1–L2)					
	Identity & Access Control	IAM + MFA + Identity Center + SCPs	Entra ID + Conditional Access + PIM	Cloud Identity + 2SV + Context-Aware Access + Org Policies	P4, P5, P6
	Encryption & Key Management	SSE-KMS (S3/EBS) + CMK	Storage Encryption + Key Vault CMK	CMEK via Cloud KMS + Org Policy enforcement	P6
	Secrets Management	Secrets Manager (auto-rotation)	Azure Key Vault (auto-rotation)	Secret Manager (auto-rotation)	P5, P6
	CSPM Baseline Posture	Security Hub + GuardDuty + Inspector	Defender for Cloud + Sentinel	SCC Premium + Chronicle	P11, P12
	Data Classification & DLP	Macie + Lake Formation	Microsoft Purview + DLP policies	Cloud DLP + Data Catalog	P5, P8, P13
	CASB & Shadow AI Controls	Third-party CASB (Netskope, etc.)	Defender for Cloud Apps	Third-party CASB / Workspace controls	P3, P5, P8
	Credential Hygiene	IAM roles (no long-lived keys)	Managed Identity (no client secrets)	Workload Identity (org policy: disable SA key creation)	P4, P5
STANDARD (L3–L4)					
	Private Service Isolation	VPC Endpoints + PrivateLink + deny-all egress	Private Endpoints + VNet integration	Private Service Connect + VPC Service Controls	P2, P6
	API Gateway & Validation	API Gateway + WAF rate limiting + request validation models	APIM + WAF + request schema validation	Apigee + Cloud Armor + API schema validation	P2, P6, P11
	Circuit Breakers & Throttling	Step Functions Choice states + API Gateway throttling	Logic Apps + APIM rate limiting	Workflows + Apigee spike arrest	P2, P11
	Egress Enforcement	Network Firewall + VPC endpoint-only (aws:sourceVpc)	Azure Firewall + UDR egress lock	Cloud NAT + Firewall Rules + VPC SC	P2, P6
ADVANCED (L5–L6)					
	Per-Agent Identity Isolation	Dedicated IAM role per agent + permission boundaries	Unique Managed Identity per agent + deny assignments	Unique service account per agent + IAM Deny policies	P4, P5, P6
	Identity Self-Mutation Prevention	IAM deny policies (iam:Create/Attach/PutRolePolicy, sts:AssumeRole to self)	Entra deny assignments; agents cannot modify own role assignments	IAM Deny policies (modify bindings, create SA, generate keys)	P4, P5

Zone	Security Invariant	AWS	Azure	GCP	ETSI EN 304 223
	SA Key / Secret Prohibition	IAM roles only; no long-lived access keys for agents	Managed Identity only; client secrets prohibited unless unavoidable	Workload Identity only; iam.disableServiceAccountKeyCreation org policy	P4, P5, P6
	Automated Kill Switch	IAM deny + SG modify + Network Firewall + SNS; note: STS creds invalidated by removing permissions, not direct revocation	Entra WI disable + NSG modify + Firewall; coordinated containment runbook, not a single feature	IAM deny + Firewall automation + WI revoke; coordinated containment runbook	P2, P4, P11
	Model Registry & Integrity	SageMaker Registry + SHA-256 hash; compare approved vs deployed; block on mismatch	Azure ML Registry + SHA-256 hash; compare approved vs deployed; block on mismatch	Vertex AI Registry + SHA-256 hash; compare approved vs deployed; block on mismatch	P7, P9
	Training Environment Isolation	VPC-only SageMaker + EnableNetworkIsolation=true	Azure ML Managed VNet + workspace isolation	Vertex AI VPC Peering + CMEK + egress controls	P2, P6, P13
	Signed Images & SBOM	ECR scanning + Inspector SBOM + AWS Signer	ACR + Content Trust + Defender scanning	Artifact Registry + Binary Authorization + Container Analysis	P7
	Periodic Model Integrity Verification	Recompute deployed hash vs Registry (daily min); automate via EventBridge + Lambda	Recompute deployed hash vs Registry (daily min); automate via Azure Automation / Logic Apps	Recompute deployed hash vs Registry (daily min); automate via Cloud Scheduler + Cloud Functions	P7, P9, P12
	Secure Decommissioning	Delete IAM roles; schedule KMS CMK deletion (7–30 day wait); purge model artifacts via S3 lifecycle; retain audit logs per compliance; document decommission in change management	Remove Managed Identity; purge Key Vault keys (soft-delete + purge protection window); delete model artifacts via Blob lifecycle; retain Sentinel logs per retention policy; close change record	Delete Service Account; destroy Cloud KMS key versions; purge model artifacts via GCS lifecycle + Retention Lock expiry; retain Cloud Audit Logs per org retention; close change record	P13, P6
HIGH ASSURANCE (L7–L8)					
	mTLS Between Agents	App Mesh + ACM Private CA; or Envoy sidecars (mechanism may vary)	AKS Service Mesh (Istio/Linkerd/OSM); or equivalent mesh	Anthos Service Mesh; or Envoy/Istio-compatible sidecars (mechanism may vary)	P2, P6
	Agent Certificate Lifecycle	ACM Private CA; certs bound to per-agent identity; <24h lifetime; automated rotation	AKS mesh certs bound to pod identity; <24h lifetime; shared certs prohibited	ASM certs bound to Workload Identity; <24h lifetime; automated rotation	P6
	Capability-Scoped Identity Tokens	Signed short-lived JWT; audience-bound, action-scoped, minute TTL, cryptographically validated	Entra token with audience/scope restrictions; short-lived	IAM conditions + signed tokens; audience-bound, short TTL	P4, P5, P6
	Model Extraction Detection	API Gateway + WAF logs + SageMaker Data Capture + CloudWatch anomaly detection	APIM + Sentinel analytics + Azure Monitor anomaly	Apigee + Cloud Monitoring + Cloud Logging anomaly	P11, P12
	Workload Policy Enforcement	EKS PSA (restricted) + OPA Gatekeeper / Kyverno admission controllers	AKS Pod Security + Azure Policy / Gatekeeper	GKE PSA (restricted) + OPA Gatekeeper / Kyverno	P2, P4, P7

Zone	Security Invariant	AWS	Azure	GCP	ETSI EN 304 223
	Cascade Detection & Correlation	CloudWatch composite alarms + Detective cross-account correlation	Sentinel cross-workspace correlation + Defender alerts	SCC custom findings + Chronicle cross-namespace correlation	P11, P12

CROSS-CLOUD SECURITY INVARIANT: *Agents must never be permitted to modify their own identity bindings, access policies, or trust relationships under any circumstance.*

Clarifications

Data governance and CASB controls are not cloud-exclusive. Enterprise tools such as Microsoft Purview, Defender for Cloud Apps, Netskope, BigID, Collibra, and others can operate across AWS, Azure, and GCP environments.

mTLS implementation may vary. The requirement is cryptographic peer authentication and encrypted east-west traffic. App Mesh, Anthos Service Mesh, and AKS service mesh are the primary implementation options; Envoy sidecars or equivalent are acceptable alternatives.

Kill switches are containment runbooks, not single features. Containment requires coordinated identity revocation, network isolation, and workload quarantine. Network containment alone is insufficient.

ETSI provision references. P2 = Network Security, P3 = Governance, P4 = Access Control, P5 = Identity & Authentication, P6 = Cryptographic Controls, P7 = Supply Chain, P8 = Data Protection, P9 = Integrity, P11 = Monitoring, P12 = Logging & Audit, P13 = Data Governance.

7A. Non-Negotiable AI Infrastructure Invariants

The following invariants apply at every tier (L1–L8) and across all clouds. They are non-negotiable and must be validated via automated policy enforcement where technically feasible. Exceptions require documented risk acceptance signed by the CISO or equivalent.

#	Invariant	Requirement	Enforcement Mechanism	ETSI Provisions
I1	**Identity Immutability**	Agents must never be permitted to modify their own identity bindings, access policies, or trust relationships under any circumstance	IAM deny policies, permission boundaries, Entra deny assignments, org policy constraints	P4, P5, P6
I2	**Unique Principal per Agent**	Every AI agent or autonomous workload must operate under a unique identity. Shared identities between agents are prohibited	Per-agent IAM roles, Managed Identities, Workload Identity bindings	P4, P5
I3	**No Long-Lived Credentials**	AI workloads must use platform identity (IAM roles, Managed Identity, Workload Identity) with short-lived tokens. Long-lived API keys, service account keys, and client secrets are prohibited	Org policies, SCP restrictions, Sentinel/GuardDuty alerting on key creation	P5, P6
I4	**Encryption Everywhere**	All AI data at rest and in transit must be encrypted with customer-managed keys (CMK). TLS 1.2+ minimum on all endpoints. mTLS required at L7–L8	KMS CMK enforcement, org policy, Config/Defender rules, service mesh mTLS	P6
I5	**Deny-All Egress by Default**	AI workloads must have no outbound internet access unless explicitly allowlisted. All cloud service access must route through private endpoints or VPC endpoints	Network Firewall / Azure Firewall / VPC SC, SG/NSG deny-all egress, VPC endpoint policies	P2, P6
I6	**Artifact Integrity Verification**	Model artifacts must be hashed (SHA-256) at registry approval and compared against deployed artifacts before promotion and periodically (minimum daily) in production. Block on mismatch	Model Registry metadata, CI/CD pipeline gates, scheduled Lambda/Functions/Automation verification	P7, P9
I7	**Kill Switch Readiness**	Every AI workload at L5+ must have a tested, automated containment runbook that can revoke identity, isolate network, and quarantine workloads within minutes. Containment must be exercised quarterly	Automated runbooks (EventBridge, Azure Automation, Cloud Functions), SIEM-triggered containment	P2, P4, P11
I8	**Cumulative Controls**	All controls are cumulative: higher tiers inherit and extend lower-tier controls. Controls must be validated via automated policy enforcement where technically feasible. Manual attestation is acceptable only where automation is not yet available	CSPM continuous monitoring, Config/Policy rules, admission controllers, CI/CD gates	P1, P3, P11, P12

7B. AI Incident Containment Flow

When a SIEM alert, anomaly detection, or manual trigger identifies a compromised or misbehaving AI workload, containment must follow four sequential phases. Each phase must complete before the next begins. The entire sequence should execute within minutes via automated runbooks.

#	Phase	Actions	AWS / Azure / GCP Tools	Success Criteria
1	IDENTITY REVOKE	Deny all actions on the agent's identity. Attach explicit deny-all IAM policy, disable Managed Identity / Workload Identity, revoke active sessions where possible. Block future role assumptions	AWS: IAM deny policy + STS policy update. Azure: Entra WI disable + role revoke. GCP: IAM deny policy + WI unbind	Agent cannot authenticate or authorise any action. Verified via test API call returning 403
2	NETWORK ISOLATE	Block all inbound and outbound traffic for the affected workload. Modify Security Groups / NSG / Firewall rules to deny-all. Isolate from service mesh if applicable. Cut inter-agent communication channels	AWS: SG deny-all + Network Firewall. Azure: NSG deny-all + Azure Firewall. GCP: Firewall deny-all + VPC SC perimeter	Zero network reachability confirmed via VPC Flow Logs / NSG Flow Logs showing denied traffic
3	ARTIFACT FREEZE	Lock all associated artifacts: model registry entries, container images, agent configurations, prompt templates, memory stores. Prevent deployment pipeline from promoting any version. Enable storage immutability	AWS: S3 Object Lock + ECR immutability + CodePipeline disable. Azure: Blob immutability + ACR quarantine. GCP: Bucket Lock + AR immutability + Cloud Build disable	No artifact can be modified, deleted, or promoted. Hash of frozen artifacts recorded for chain of custody
4	FORENSIC CAPTURE	Preserve all evidence: CloudTrail / Audit Logs / Cloud Audit Logs, VPC Flow Logs, SIEM correlation data, agent memory state, inference logs, inter-agent communication records. Export to immutable storage. Record timeline	AWS: S3 Object Lock + Detective. Azure: Blob immutable + Sentinel investigation. GCP: Cloud Storage Bucket Lock + Chronicle investigation	Complete forensic package in immutable storage with documented chain of custody. Investigation can proceed without evidence destruction risk

Target: Full containment (Phases 1–4) completed within 15 minutes of trigger. Quarterly tabletop exercises must validate the runbook end-to-end, with results documented and gaps remediated within 30 days.

8. ETSI EN 304 223 Principle Reference

The 13 principles are applied across all risk gradient levels with depth scaling to the assessed risk. The following provides a quick reference mapping.

Lifecycle Phase	ID	Principle
Secure Design	**P1**	Raise awareness of AI security threats and risks
Secure Design	**P2**	Design the AI system for security as well as functionality and performance
Secure Design	**P3**	Evaluate the threats and manage the risks to the AI system
Secure Design	**P4**	Enable human responsibility for AI systems
Secure Development	**P5**	Identify, track and protect the assets
Secure Development	**P6**	Secure the infrastructure
Secure Development	**P7**	Secure the supply chain
Secure Development	**P8**	Document data, models and prompts
Secure Development	**P9**	Conduct appropriate testing and evaluation
Secure Deployment	**P10**	Communication and processes associated with end-users and affected entities
Secure Maintenance	**P11**	Maintain regular security updates, patches and mitigations
Secure Maintenance	**P12**	Monitor the system's behaviour
Secure End of Life	**P13**	Ensure proper data and model disposal