Synthetic Users SDLC AI/GenAI Addendum

Document ID: CRA-13.1.1-AISDLC-001

Version: 1.0

Effective Date: March 25, 2026

Last Updated: March 25, 2026

Owner: CTO — Artur Ventura

Approved By: CTO — Artur Ventura

Classification: Internal – Confidential

CRA Control: CRA 13.1.1

Parent Policy: Secure Software Development Lifecycle (SDLC) Policy v1.6

---

1. Purpose and Context

This Addendum extends the Synthetic Users Secure Software Development Lifecycle (SDLC) Policy (v1.6, November 2025) to address AI/GenAI-specific development lifecycle requirements mandated by JPMC Control CRA 13.1.1. It supplements — but does not replace — the parent SDLC policy, adding binding requirements wherever AI or Generative AI systems are involved.

JPMC CRA 13.1.1 requires confirmation that a documented SDLC process covers AI/GenAI models, encompassing model validation, risk assessments, policies for retraining models, and testing strategies. This Addendum also formally documents the company's alignment with the **CWE/SANS Top 25** vulnerability standard across all developed applications.

1.1 Why This Addendum Is Needed

Synthetic Users' platform incorporates AI/GenAI capabilities including a multi-provider LLM orchestration layer (LLM Shuffle), a Persona Engine based on the OCEAN personality model, and a Retrieval-Augmented Generation (RAG) pipeline. These capabilities introduce lifecycle considerations not fully addressed by a traditional Secure SDLC:

- Non-deterministic model outputs requiring specialised testing approaches
- Third-party model vendor risks (prompt injection, data exfiltration via outputs)
- Data flow concerns unique to prompts, context windows, and inference logs
- Model selection, validation, and deprecation cycles distinct from traditional software releases
- Regulatory and contractual AI disclosure obligations, including JPMC CRA 13.1.1

1.2 Scope

This Addendum applies to all AI/GenAI features in the Synthetic Users production platform, all LLM providers accessed via API, all engineers and contractors involved in AI/GenAI development, and all RAG components, prompt templates, and agent workflows.

1.3 Exclusions

This Addendum does not apply to internal productivity AI tools (e.g., GitHub Copilot) unless those tools process customer data or JPMC data.

---

2. CWE/SANS Top 25 Alignment

Synthetic Users' SDLC explicitly incorporates the **CWE/SANS Top 25 Most Dangerous Software Weaknesses** as a vulnerability taxonomy for all developed applications, in addition to the OWASP Top 10. This section formally documents that alignment.

2.1 Integration in SAST and DAST Workflows

- SAST tooling (GitHub Advanced Security / CodeQL) is configured to report CWE IDs alongside findings
- DAST results include CWE references in remediation notes provided to engineering
- Penetration test reports are required to map findings to CWE identifiers where applicable
- Vulnerability management SLAs apply to all CWE/SANS Top 25-mapped findings

2.2 Key CWE/SANS Top 25 Weaknesses — Platform Mapping

CWE ID	Weakness	Relevance to Synthetic Users	Mitigation Control
CWE-79	Cross-site Scripting (XSS)	Web app renders dynamic AI output content	Output encoding, CSP headers, LLM response sanitisation
CWE-89	SQL Injection	Database queries for research sessions	Parameterised queries (Prisma ORM); SAST injection checks
CWE-20	Improper Input Validation	Prompt inputs; JPMC API inputs	Input length limits, schema validation, prompt boundary controls
CWE-352	CSRF	State-changing API calls	CSRF tokens; SameSite cookies; Google Firebase session controls
CWE-22	Path Traversal	RAG document ingestion pipeline	Allowlist-based path validation; sandboxed ingestion workers
CWE-732	Incorrect Permission Assignment	Multi-tenant data isolation	RBAC; row-level security; tenant ID propagation checks
CWE-502	Deserialization of Untrusted Data	LLM output parsing; API response handling	Type-safe parsers; schema validation

CWE-862	Missing Authorisation	API endpoints for session management	Middleware auth checks; automated endpoint coverage testing
CWE-798	Hard-coded Credentials	LLM provider API keys	Render environment variables; 1Password; SAST secret detection
CWE-287	Improper Authentication	User login; admin access; JPMC API auth	Google Firebase (application auth); Google Workspace (employee auth); MFA enforcement; API key rotation policy

2.3 Remediation SLAs

Severity	Definition	SLA	Escalation
Critical	CWE/SANS Top 25 + active exploitation or CVSS ≥ 9.0	24 hours	CTO + CEO immediate notification
High	CWE/SANS Top 25 + CVSS 7.0–8.9	7 calendar days	CTO notified; tracked in sprint
Medium	CWE/SANS Top 25 + CVSS 4.0–6.9	30 calendar days	Engineering lead tracks
Low	CVSS < 4.0 or informational	90 calendar days	Tracked in backlog

---

3. AI/GenAI Model Lifecycle

AI/GenAI systems at Synthetic Users follow a structured lifecycle that integrates with the parent SDLC. Unlike traditional software, AI/GenAI features involve the selection, validation, integration, monitoring, and retirement of third-party language models — not the training of models from scratch.

Phase	Activities	Responsible	Gate Criteria
1. Model Selection & Feasibility	Identify candidate LLM providers via LLM Shuffle evaluation; assess capabilities, data handling terms, security posture	CTO / Engineering Lead	Provider must pass DPA review; no customer PII sent without DPA
2. Security & Privacy Review	Review provider's data retention, logging, fine-tuning opt-outs, incident response SLAs; prompt injection risk assessment; DFD update	CTO + Legal	Signed DPA; DFD updated; no customer data logged without opt-in
3. Integration & Development	Implement API via LLM Shuffle; develop prompt templates; build RAG connectors; apply output sanitisation	Engineering	Code review complete; SAST passing; no hardcoded keys
4. AI-Specific Testing	Execute AI/GenAI test suite; validate output quality, safety, security; conduct adversarial prompt testing	Engineering + QA	Test suite passing; prompt injection scenarios tested and mitigated
5. Staged Deployment	Deploy to staging; canary release to production subset; monitor output quality	Engineering	Canary metrics within bounds; rollback procedure tested
6. Production Monitoring	Output quality monitoring; anomaly detection; latency tracking; periodic adversarial re-testing	Engineering + CTO	Automated alerts active; AI/GenAI incident playbook in place
7. Model Retirement	Plan migration; update prompt templates; re-run test suite; confirm data deletion from retired provider	Engineering + Legal	Migration tests passing; provider data deletion confirmed in writing

4. Model Validation Framework

Before any AI/GenAI model or provider is enabled in production, it must pass a structured validation process.

4.1 Validation Checklist — New Model or Provider

Category	Requirement	Evidence
Data Handling	Provider does not train on or retain API request data by default	Written confirmation via provider DPA
Data Handling	No customer PII or JPMC data sent to provider without DPA coverage	DFD review; legal sign-off
Security	Provider enforces TLS 1.2+ in transit; at-rest encryption confirmed	Provider security docs or SOC 2 report
Security	Prompt injection risk assessment completed	Internal assessment document; adversarial test cases
Security	API key lifecycle defined: rotation frequency, storage in Render env vars	SAST scan; 1Password entry confirmed
Quality	Model output evaluated against representative test set	Evaluation report; pass/fail criteria documented
Quality	Output hallucination and safety behaviour assessed	Test report; content policy acceptance recorded
Contractual	Provider terms permit B2B commercial use	Legal review sign-off
Incident Response	Provider's incident response SLA and notification obligations confirmed	DPA or security exhibit reference
Resilience	Fallback provider identified in LLM Shuffle configuration	LLM Shuffle fallback config documented

4.2 Validation for Model Capability Changes

When a provider releases a significant model update, a lightweight re-validation is conducted:

- Review of provider change notes for data handling or safety behaviour changes
- Re-run of core AI test suite against the updated model endpoint
- Comparison of output quality metrics against baseline
- CTO sign-off before switching default model in LLM Shuffle

---

5. AI/GenAI Risk Assessment

AI/GenAI features are subject to a dedicated risk assessment in addition to standard threat modelling.

5.1 When Required

- Introduction of a new LLM provider or model to production
- New AI/GenAI feature processing user-provided data or JPMC data
- Changes to prompt templates affecting data flow
- Changes to the RAG pipeline (ingestion sources, retrieval logic)
- Changes to output post-processing or content filtering

5.2 Required Topics

Risk Area	Key Questions	Typical Controls
Prompt Injection	Can a malicious user override system instructions or exfiltrate context window data?	Input validation; prompt boundary enforcement; adversarial test cases
Data Leakage via Output	Can the model output other tenants' data or RAG index content it shouldn't expose?	Tenant ID scoping in RAG; output PII scanning; session isolation

Hallucination Risk	Could incorrect outputs cause harm in synthetic user research (false attribution, fabricated personas)?	AI-generated labelling; downstream validation; user disclaimers
Third-Party Model Risk	What are the provider's retention, logging, and training data practices?	DPA review; training opt-out; API log review
Availability & Resilience	What is the impact of LLM provider downtime? Is there a fallback?	LLM Shuffle fallback; circuit breaker patterns; degraded-mode UX
Regulatory Compliance	Does the AI feature trigger GDPR, CCPA, or JPMC contractual obligations?	DPIA if required; data minimisation; consent; JPMC notification
Model Misuse	Could the AI feature be used to generate harmful or deceptive content?	Content filtering; system prompt scoping; rate limiting; abuse detection

---

6. Retraining and Model Update Policy

6.1 Current Architecture — No Fine-Tuning

Synthetic Users does not fine-tune or retrain base language models. All AI/GenAI capabilities are delivered through third-party LLMs via API, with context provided at inference time via RAG. This architecture materially reduces model retraining risk.

Because no proprietary model weights are maintained, "retraining" in the traditional sense does not apply. Instead, the company manages **prompt template versioning**, **RAG index updates**, and **LLM provider version management** as the functional equivalents.

6.2 Prompt Template Versioning

- All prompt templates are stored in the version-controlled repository under `/prompts/` with semantic versioning

- Changes to prompt templates affecting data flow or output behaviour require engineering review and testing
- Prompt template releases follow the same CI/CD pipeline as code changes
- Breaking changes to system prompts require CTO approval

6.3 RAG Index Update Policy

- New document sources added to the RAG index require security review
- RAG index contents are scoped by tenant — no cross-tenant retrieval is permitted
- JPMC data is not ingested into the shared RAG index; JPMC context is provided in the prompt context window only
- Index integrity checks are run after bulk updates

6.4 Model Provider Version Management

Trigger	Action	Approval
Provider deprecates model version in production	Migration plan created ≥30 days before deprecation; test suite re-run	Engineering Lead + CTO
Provider releases major new version	Lightweight validation; AI test suite re-run; output comparison	Engineering Lead; CTO for default switch
Provider changes data handling policy	Legal and CTO review; re-validate DPA; update DFD	CTO + Legal
Security vulnerability in model or provider API	Immediate risk assessment; consider fallback via LLM Shuffle	CTO — immediate escalation

7. AI/GenAI Testing Strategies

7.1 Required Test Types

Test Type	What It Validates	Frequency	Tooling
Prompt Regression Testing	Prompt templates produce expected output structure and content for fixed representative inputs	Every deployment modifying a prompt template	Custom test harness; CI/CD assertions
Adversarial Prompt Testing	System resists prompt injection, jailbreak, and instruction overrides; no context window leakage	Before each new model integration; quarterly	Manual red-team + automated injection pattern library
Output Safety Testing	Outputs are free of harmful content; AI-generated content is correctly labelled	Before each model or system-prompt change	Content policy checklist; manual output review
Data Isolation Testing	Tenant A's RAG context is not retrievable by Tenant B; no cross-tenant leakage	Before each RAG pipeline change; quarterly	Multi-tenant test accounts; retrieval audits
Hallucination & Accuracy	Synthetic persona outputs accurately reflect input parameters	Per release cycle; sampled in production	Golden-set comparison; human review
Performance & Latency	LLM response times meet SLA; fallback provider activation tested	Load tested quarterly; canary phase	k6; LLM Shuffle failover drill
Integration Security	LLM API auth, key rotation, error handling correct; no keys in logs	Every deployment; annual pentest scope	SAST (CodeQL); code review; pentest

7.2 OWASP Top 10 for LLM Applications — Alignment

OWASP LLM Risk	Synthetic Users Control
LLM01 — Prompt Injection	Input validation; system prompt boundary enforcement; adversarial test cases in CI/CD
LLM02 — Insecure Output Handling	Output sanitisation before rendering; HTML encoding; CSP headers; no raw LLM output to DOM
LLM03 — Training Data Poisoning	Not applicable (no fine-tuning); RAG index integrity checks and source validation
LLM04 — Model Denial of Service	Rate limiting on AI endpoints; timeout and retry logic; LLM Shuffle fallback
LLM05 — Supply Chain Vulnerabilities	LLM provider security reviews; DPA validation; Dependabot on AI SDK dependencies
LLM06 — Sensitive Information Disclosure	RAG tenant scoping; no PII in shared prompts; output review for data leakage patterns
LLM07 — Insecure Plugin Design	All LLM tool/function calls reviewed for injection risk; allowlist of permitted actions
LLM08 — Excessive Agency	Agent permissions constrained; human-in-the-loop for high-risk actions
LLM09 — Overreliance	Outputs labelled as AI-generated; downstream validation steps documented
LLM10 — Model Theft	API keys rotated regularly; access logs monitored; no model weights stored internally

8. SDLC Toolchain Security

Access to the AI/GenAI SDLC toolchain is controlled with the same rigour as general software development:

Toolchain Asset	Access Control	Privileged Access
GitHub repository (prompts/, AI code)	RBAC; PR review required for all merges	Admin rights limited to CTO; engineers via SSO + MFA
LLM provider API keys	Stored in Render env vars and 1Password; never in code	Rotation requires CTO approval; CI/CD uses scoped service keys
RAG index / vector store	Tenant-scoped; no direct production access outside break-glass	Break-glass requires CTO authorisation; fully logged
AI monitoring dashboards	Read-only for engineers; alert config limited to Engineering Lead	SSO enforced; no shared credentials
LLM provider admin portals	CTO-only access to billing, usage analytics, and model management	MFA enforced; access reviewed quarterly

9. Roles and Responsibilities

Role	Responsibilities
CTO — Artur Ventura	Policy owner; approves new model integrations; approves breaking changes to system prompts; leads AI/GenAI risk assessments; escalation for AI security incidents
Engineering Lead	Implements and enforces this Addendum; owns AI test suite; coordinates adversarial testing; reviews model validation checklists
Engineers	Follow AI/GenAI coding standards; include adversarial test cases in PRs; flag anomalous model behaviour; never hardcode API keys
Legal / Compliance	Reviews and approves DPAs for new LLM providers; advises on JPMC data handling; reviews model retirement data deletion confirmations

CEO — Kwame Ferreira	Receives immediate notification of Critical AI/GenAI security incidents; approves strategic AI provider decisions

10. Evidence and Audit Requirements

Evidence Artefact	Created When	Retention	Storage
Model Validation Checklist	Each new model or provider integration	3 years	Internal compliance folder
AI/GenAI Risk Assessment	Each material AI feature change	3 years	Internal compliance folder
Prompt Template Change Log	Each prompt template release	3 years	GitHub repository history
AI Test Suite Results	Each relevant deployment	1 year rolling	CI/CD pipeline artefacts
Adversarial Test Reports	Quarterly + per new model integration	3 years	Internal compliance folder
RAG Index Audit Logs	Continuous	1 year rolling	AWS CloudWatch / logging service
Provider DPA Records	Each provider engagement	Contract + 3 years	Legal folder
Model Retirement Records	Each model deprecation or provider exit	5 years	Legal + compliance folder

11. Change Management

Version	Date	Author	Summary

1.0	March 25, 2026	Artur Ventura, CTO	Initial release. Establishes AI/GenAI lifecycle, model validation, risk assessment, retraining policy, CWE/SANS Top 25 alignment, and AI-specific testing strategies as mandated by JPMC CRA 13.1.1.

This Addendum is reviewed annually and whenever a material change occurs to the AI/GenAI platform, LLM provider roster, or applicable regulatory requirements.

---

12. Related Documents

- Secure Software Development Lifecycle (SDLC) Policy v1.6
- AI/GenAI Algorithm Design Document & Data Flow Diagram
- AI/GenAI Decommissioning Policy
- Responsible AI & Risk Management Overview
- Third-Party Risk Management Policy
- Information Governance & Records Management Standard
- Incident Response Plan

Synthetic Users, Inc. — 4223 Glencoe Ave, Suite C215-523, Marina del Rey CA 90292