AI Technical Requirements

The following AI Technical Requirements are used in HUD's guidance on artificial intelligence (AI).

AI Requirement Category	AI Requirement
Data Usage and Training Integrity	1. Provide a detailed data flow diagram and pseudocode illustrating how the AI System enforces zero-retention policies during inference or fine-tuning. The diagram must identify and describe auditing mechanisms that enable the assessment of whether anonymized aggregates from federated datasets could inadvertently contribute to global model updates. 2. Provide a detailed description of how the AI System technically isolates client-specific prompts or embeddings from shared training pipelines. 3. Provide the means and methods to enforce zero-retention of client data and keep it segregated from shared models, with auditable proof that client data does not influence any global model or system improvement. 4. Provide a sandbox demonstration of the opt-out feature showing no leakage to Contractor enhancements, and furnish clear documentation of training data provenance, high-level model architecture (as feasible), and the basis for outputs or decisions. 5. Provide documentation on the AI System's training data, model architecture (to the extent feasible), and the logic of outputs or decisions where appropriate.
Third-Party Integrations and Backend Dependencies	1. Identify any/all reliance of the proposed AI on backend or external models (e.g. OpenAI via Azure, various AI models via Amazon Bedrock, etc.) requiring transparencies on dependencies that could expose data. 2. Provide the full dependency tree for the proposed/used AI stack (tools, platforms, and technologies), including versions of third-party LLMs or Application Programming Interfaces (APIs). 3. Provide endpoint traces showing how data is masked or tokenized before transmission to prevent external training exposure. 4. Provide a detailed description of how the architecture handles failover or updates in the backend providers without risking data spillage. 5. Provide penetration test results specifically simulating attacks on these integrations in a FedRAMP authorized setup. 6. Explain the technical audit trails for any cross-vendor data flows, including Service Level Agreements (SLAs) for data sovereignty. 7. Demonstrate how the AI System would detect and block unauthorized backend queries that could lead to non-public data being repurposed.
Suitability for Sparse or Limited Datasets	1. For supervised learning features like fraud detection, provide custom bench markers using synthetic datasets with only 100-500 labeled instances (simulating HUD's rare events). Include precision-recall curves

AI Technical Requirements

	and error analysis to show efficacy of the supervised learning methods without overfitting or high false positives. 2. Describe alternative techniques (e.g. semi-supervised, transfer learning etc.) integrated into the AI System for handling imbalanced classes in sparse environments and furnish code snippets or demonstration script illustrating AI model training on fewer than 1000 examples while maintaining a high F1-score. 3. Provide a detailed description of how the AI System incorporates active learning loops to augment limited labeled data without requiring external sourcing. Provide validation metrics from analogous low-data federal use cases to substantiate efficacy claims.
DevSecOps Practices	1. Provide a detailed description of the DevSecOps pipeline architecture, including automated security scans (e.g. SAST/DAST tools), for AI System components using Continuous Integration/Continuous Delivery (CI/CD). Provide a clear, end-to-end description of the DevSecOps pipeline for the AI System showing how code and models move from commit to production with automated security checks and release gates. Include a simple Gitlab flow or Jenkins or equivalent example showing a required threat-model review and approval before deployment. 2. Provide audit-backed evidence of DevSecOps maturity (e.g., deployment frequency and mean time to resolve security incidents) and explain how infrastructure templates enforce AI-specific protections such as tamper-evident model versioning, locked model repositories, policy guardrails, and automatic rollback if tampering or drift is detected.
Security	1. Provide a detailed description of the adversarial robustness testing protocol, beyond FedRAMP authorization, for AI models. Provide specific tools used (e.g. ART library), results from red-team exercises targeting prompt injection, and/or model inversion attacks. 2. Provide a zero-trust architecture blueprint for the AI System, detailing micro-segmentation of data pipelines. Demonstrate via API calls how attribute-based access controls prevent unauthorized feature extraction. 3. Implement and maintain security controls consistent with NIST SP 800-53 Rev. 5 as applicable to system impact level. At minimum, AI systems must demonstrate compliance with the following control families: a. Access Control (AC): AC-2 Account Management; AC-3 Access Enforcement; AC-6 Least Privilege; AC-17 Remote Access; AC-20 Use of External Systems. • All AI system access must enforce role-based access and multi-factor authentication. b. Identification actor Authentication (IA) and Multi-Factor Authentication (MFA): IA-2 MFA Enforcement; IA-5 Authenticator Management.

AI Technical Requirements

	• AI System administrative access must require phishing-resistant MFA. c. Audit and Accountability (AU): AU-2 Event Logging; AU-6 Audit Review; AU-12 Audit Generation. • The Contractor shall log: Prompt inputs (where applicable); model outputs affecting mission decisions; Administrative model changes; and Data ingestion events. • The Contractor shall ensure logs are retained consistent with the HUD retention policy. d. System and Information Integrity (SI): SI-3 Malicious Code Protection; SI-4 System Monitoring; SI-7 Software Integrity; SI-10 Input Validation • The Contractor shall implement safeguards against: Prompt injection; Data poisoning; Model tampering; and Unauthorized parameter modification e. Risk Assessment (RA): RA-3 Risk Assessment; RA-5 Vulnerability Monitoring • The Contractor shall conduct: AI-specific risk assessments; Bias and fairness evaluation; and Adversarial testing. f. System and Services Acquisition (SA): SA-8 Security Engineering Principles; SA-11 Developer Testing and Evaluation; SA-15 Development Process Standards; SA-22 Unsupported Components g. Privacy Controls (AP / AR) • Where AI System processes PII: Data minimization is required; Training on HUD PII is prohibited without written approval; and Privacy risk assessment is required h. Incident Response (IR): IR-4 Incident Handling; IR-6 Incident Reporting • AI System-related incidents must be reported within 1 hour if involving: Data leakage; Model compromise; Unauthorized data exposure; or Operational impact due to AI System malfunction
Transparency, Explainability, and Bias Mitigation	1. When the AI model backend is opaque, provide examples of how the AI model parameters change based on a simulated query from a HUD system, explain how AI model's behavior stays consistent as the AI System scales without affecting performance. 2. Provide a detailed description of the bias auditing codebase (e.g. AI Fairness 360 (AIF360) library) utilized, including scripts for slicing metrics across protected attributes. Provide pre- and post- mitigation reports from deployments with spare, imbalanced data.