AWS guide for Financial Services risk management of the use of Generative AI

March 2026

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided “as is” without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Additionally, this document does not constitute legal advice and should not be relied on as legal advice. AWS encourages its customers to obtain appropriate advice on their implementation of privacy and data protection environments, and more generally, applicable laws relevant to their business.

Table of contents

Executive Summary

This guide helps financial services customers assess and manage risks when implementing generative AI solutions on AWS. It provides a structured framework of 16 risk categories, including regulatory compliance, model accuracy, data security, and API security. The framework incorporates established security standards like OWASP GenAI Top 10 and MITRE ATLAS, making it accessible for security professionals while addressing AI-specific considerations.

For each risk category, you can find specific technical guidance for implementing mitigations using AWS services. For example, you can use:

- Amazon Bedrock guardrails for content filtering and regulatory compliance
- Amazon Bedrock knowledge bases for retrieval augmented generation (RAG)
- AWS WAF and AWS Shield for API protection
- Amazon SageMaker Clarify for model transparency

The guide includes configuration recommendations and integration patterns to help you implement appropriate mitigations for your use of AI solutions. By providing both risk assessment frameworks and technical implementation guidance, you can move quickly from planning to deployment of secure generative AI solutions.

How to use this guide

This guide provides practical risk identification and mitigation strategies for Financial Services Institution (FSI) customers implementing generative AI solutions. The content is based on AWS's experience working with FSI customers across diverse use cases and regulatory environments.

Important: This guide is not intended as a comprehensive implementation manual for risk management. FSI customers already have established risk management procedures, governance frameworks, regulatory requirements, and compliance protocols. Instead, this guide serves as a focused resource to help your technical teams and risk experts identify generative AI-specific risks that may not be covered by traditional IT risk frameworks.

Target audience: This guide is designed for collaboration between technical architects, security professionals, and risk management teams who need to understand the unique risk profile of generative AI systems in financial services contexts.

Risk prioritization: The risks presented in this guide are ordered according to their relevance and priority, based on patterns and insights AWS has gathered through working with multiple FSI customers implementing generative AI solutions. The most critical and commonly encountered risks appear first, followed by more specialized or context-specific considerations.

Mitigation approach: While this guide recommends specific mitigation measures for each identified risk, it does not aim to provide full implementation details. Instead, it points you toward appropriate AWS services, industry best practices, and additional resources where detailed implementation guidance is available. Use these recommendations as starting points for your own detailed implementation planning.

How to navigate: Each risk category includes specific mitigation strategies using AWS services, references to industry frameworks (OWASP GenAI Top 10, MITRE ATLAS), and practical implementation guidance. Use this structure to map generative AI risks to your existing risk taxonomy and identify where additional controls may be needed.

AI risks and mitigations

This guide describes the most common risks and mitigations specific to generative AI applications. It is not a full and comprehensive risk and mitigation documentation. You should perform threat modelling as described in section “Use threat ” to do your own threat model to identify all applicable risks. This guide does not include security risks that are not specific to AI, such as end user authentication, data protection, networks security, etc.

To help you identify and categorize generative AI risks, this guide draws from risks identified in multiple industry risk frameworks:

- AWS Responsible AI dimensions
- OWASP Top 10 for Large Language Model Applications
- MITRE ATLAS (Adversarial Threat Landscape for Artificial-Intelligence Systems)

Additionally, financial services implementation patterns provide industry-specific context.

Each risk is analyzed for:

- Potential impact on financial services operations
- Available technical mitigations
- Implementation considerations

For a complete mapping of risks to AWS Responsible AI dimensions, see Appendix B: Responsible AI Dimension and risk mapping.

When you conduct risk assessments for AI systems, consider both technical and non-technical factors to structure your risk assessments to specifically target critical areas of concern.

Technical factors:

- System performance and reliability
- Security controls and safeguards
- Data protection measures

Non-technical factors:

- Ethical implications
- Impacts on different user groups
- Societal considerations

This comprehensive approach helps you identify critical areas of concern, evaluate broader impacts, and plan appropriate mitigations.

Note: This guide covers common generative AI risks but is not an exhaustive list of all potential AI-related risks. Similarly, mitigation measures proposed may not be exhaustive. We recommend working with your security and compliance teams to identify additional risks specific to your use case.

1.1 General guidance

1.1.1 Use your existing SDLC process

Apply your established security controls and best practices to AI systems. Include, for example:

- Identity and access management (IAM)
- Secure software development lifecycle practices
- Testing protocols
- Vulnerability management

Integrate AI systems into your existing security and compliance processes instead of creating separate processes. This helps ensure consistent governance across all your systems, including AI systems. Additionally, align your AI implementations with your established control frameworks, such as NIST CSF, ISO 27001, or SOC 2.

1.1.2 Perform initial assessment

Consider evaluating your generative AI workload before implementation to help improve preparedness.

1.1.3 Evaluate models

Select foundation models that best fit your requirements. You can use Amazon Bedrock evaluations to compare and select models for your use case.

For more information, see Evaluate large language models for quality and responsibility on the AWS Security Blog.

1.1.4 Use threat modelling

Apply threat modelling to your generative AI workloads to identify potential risks. Consider AI-specific concerns such as:

- Model poisoning
- Prompt injection
- Data extraction attacks

While this guide describes common security risks, your implementation might have unique requirements. We recommend conducting detailed threat modelling for your specific workload.

For more information, see Threat modeling your generative AI workload on the AWS Security Blog.

1.1.5 Test mitigations

Verify that your mitigations work as intended:

- Test both positive and negative cases
- Check for false positives that might block valid inputs
- Evaluate your retrieval augmented generation (RAG) workflow using Amazon Bedrock knowledge bases

1.1.6 Monitor and improve

Implement monitoring to help ensure your generative AI systems maintain:

- Compliance
- Accuracy
- Appropriate behaviour

Update your foundation models when new versions become available.

1.1.7 Manage AI vulnerabilities

Monitor and update defences against evolving AI vulnerabilities like prompt injection.

AI vulnerabilities being very similar to traditional security vulnerabilities, consider including AI vulnerability management in your existing vulnerability monitoring and remediation processes.

1.2 AI risk categories and their mitigations

1.2.1 Non-compliant output

Your generative AI system might produce output that doesn't meet regulatory requirements. In financial services, this is particularly important when the AI system output for example includes:

- Financial guidance
- Product recommendations
- Sales information

Regulatory restriction may apply, including:

- Restrictions on providing financial advice (vs. “guidance”)
- Mis-selling rules
- Consumer duty requirements

Causes

- Insufficient content filtering
- Lack of appropriate context for the use case
- Missing regulatory compliance checks

Relationships

Relates to	Topic
AWS responsible AI dimensions	Controllability, Safety, Transparency
MITRE ATLAS	External Harms (AML.T0048)

Mitigations

Mitigations or controls	Reference
Use prompt engineering techniques to guide the model toward appropriate topics and prevent unwanted responses.	Amazon Bedrock User Guide – Prompt engineering concepts
Configure content filters and guardrails to restrict model responses to approved topics.	Amazon Bedrock User Guide – Guardrails – Denied topics

Mitigations or controls	Reference
Use Retrieval-Augmented Generation (RAG) to enhance your model responses with information from trusted knowledge bases.	Amazon Bedrock User Guide – Knowledge Bases
Automated Reasoning checks in Amazon Bedrock Guardrails uses automated reasoning to verify that natural language content complies with your defined policies. This mathematical verification helps ensure that your content strictly follows your guardrails.	Automated Reasoning checks in Amazon Bedrock Guardrails
For internal AI systems, validate outputs with human review before business use (human-in-the-loop).	Get user confirmation before invoking action group function AWS re:Invent 2025 - Implementing Human-in-the-Loop Controls for Multi-Agent AI Systems Implement human-in-the-loop confirmation with Amazon Bedrock Agents
Maintain audit logs of AI-generated outputs and the guardrails applied to support regulatory reporting and post-incident analysis	Monitoring the performance of Amazon Bedrock

Practical guidance

Use your existing compliance materials to create AI guardrails:

- Employee compliance policies
- Training materials
- Procedure documents
- Incident reports

1.2.2 Off-topic and inappropriate output

Your generative AI system might produce responses that:

- Don't align with your business purpose
- Include inappropriate content
- Cover unrelated topics

In financial services applications, off-topic responses can:

- Undermine professional credibility
- Create reputational risks
- Reduce operational efficiency
- Breach applicable regulatory requirements

Causes

- Insufficient topic boundaries
- Missing content filters
- Unclear business context in system configuration

Relationships

Relates to	Item
AWS responsible AI dimensions	Controllability, Safety, Fairness
MITRE ATLAS	External Harms (AML.T0048)

Mitigations

Mitigations or controls	References
Use prompt engineering techniques to guide the model toward appropriate topics and prevent unwanted responses. Include an allowlist of approved topics aligned with the business purpose.	Amazon Bedrock User Guide – Prompt engineering concepts
Configure content filters and guardrails to restrict model responses to approved topics.	Amazon Bedrock User Guide – Guardrails – Denied topics
Use Amazon Bedrock Guardrails to detect and filter hallucinations in model responses by performing contextual grounding checks when you provide a reference source and query.	Contextual grounding check

Mitigations or controls	References
For internal AI systems, validate outputs with human review before business use (human-in-the-loop).	Get user confirmation before invoking action group function AWS re:Invent 2025 - Implementing Human-in-the-Loop Controls for Multi-Agent AI Systems Implement human-in-the-loop confirmation with Amazon Bedrock Agents

1.2.3 Misinformation through inadvertent or malicious action

Your generative AI system might provide incorrect responses due to:

- Compromised input data
- Outdated knowledge bases
- Intentional system manipulation
- Unverified information sources

In financial services applications, inaccurate information can affect:

- Customer decisions
- Regulatory compliance
- Service reliability
- Business reputation

Causes

- Insufficient data validation
- Missing knowledge base updates
- Limited domain expertise

Relationships

Relates to	Item
AWS responsible AI dimensions	Veracity and robustness, Controllability, Safety, Fairness
OWASP GenAI Top 10	Misinformation (LLM09:2025)
MITRE ATLAS	External Harms (AML.T0048)

Mitigations

Mitigations or controls	References
Use prompt engineering techniques to guide the model toward appropriate topics and prevent unwanted responses.	Amazon Bedrock User Guide – Prompt engineering concepts
Verify that your knowledge base data sources are up-to-date, accurate, reliable, and complete.	Sync your data with your Amazon Bedrock knowledge base

Mitigations or controls	References
For internal AI systems, validate outputs with human review before business use (human-in-the-loop).	Get user confirmation before invoking action group function AWS re:Invent 2025 - Implementing Human-in-the-Loop Controls for Multi-Agent AI Systems Implement human-in-the-loop confirmation with Amazon Bedrock Agents
Use source attribution in RAG-based response for end users to verify provenance of information.	RetrieveAndGenerate API Reference
Use integrity monitoring on knowledge base data sources to detect unauthorized modifications. Track changes to documents used in knowledge bases.	For example on S3 data sources use Amazon S3 event notification to track changes to documents.

See also *1.2.8 – Prompt injection*.

1.2.4 Model output is abusive or harmful

Your generative AI system might produce toxic responses such as:

- Offensive language
- Inappropriate content
- Culturally insensitive material

Toxic responses can affect:

- Customer experience
- Brand reputation
- Professional environment
- Regulatory compliance

Causes

- Training data quality issues
- Missing content filters
- Insufficient cultural context
- Intentional system misuse

Relationships

Relates to	Item
AWS responsible AI dimensions	Fairness, Safety
OWASP GenAI Top 10	Prompt Injection (LLM01:2025)
MITRE ATLAS	Manipulate AI Model (AML.T0018) LLM Prompt Injection (AML.T0051)

Mitigations

Mitigations or controls	References
Amazon provides AI Service Cards for models that are pre-trained for AWS services like Amazon Bedrock and Amazon Q. These cards help you understand how Amazon addresses toxicity in each model.	AWS AI Service Cards
Use Amazon Bedrock's guardrails to detect and filter harmful content.	Amazon Bedrock Guardrails

Mitigations or controls	References
Foundation Model Evaluations (FMEval) evaluates your model to detect inappropriate content, including sexual references, profanity, hate speech, aggression, insults, flirtation, identity-based attacks, and threats.	Amazon SageMaker - Toxicity
Implement a user reporting mechanism that allows end users to flag abusive or harmful outputs. Reported incidents a reviewed within a defined process to refine content filters.	Monitor model invocation using CloudWatch Logs and Amazon S3

Practical guidance

Content filters might occasionally flag legitimate business terms, restrict valid communications, or generate unnecessary alerts. To help maintain system effectiveness, create allowlists for business terms that include approved terminology for: brand names, product names, industry terms, and technical vocabulary. Also test filter settings to verify that your content filters allow necessary business communications and generate accurate alerts.

Monitor and adjust regularly your filtering system to reduce false positives. Keep the allow lists for approved terms updated to maintain service quality.

1.2.5 Model output is biased

Your generative AI system might produce responses that show unintended preferences or exclusions based on:

- Demographic factors
- Geographic location
- Other sensitive attributes

Unbalanced responses can affect:

- Fair treatment of customers
- Regulatory compliance
- Service accessibility
- Business reputation

Causes

Sources of potential bias and model imbalance can be found in:

- Training data selection
- Model predictions based on misrepresented features (for example, age, location or any other feature that might be sensitive in nature)

Relationships

Relates to	Item
AWS responsible AI dimensions	Fairness, Transparency
OWASP GenAI Top 10	Data and Model Poisoning (LLM04:2025)
MITRE ATLAS	Poison Training Data (AML.T0020)

Mitigations

Mitigations or controls	References
AI model providers typically share information about their responsible AI practices, including how they address potential bias. For example, when you use pre-trained models through AWS services like Amazon Bedrock or Amazon Q, Amazon provides AI Service Cards. These cards explain how Amazon addresses fairness and bias for each specific model. The model provider is responsible for training models to produce unbiased outputs.	AWS AI Service Cards

Mitigations or controls	References
Use prompt engineering techniques to guide the model toward appropriate topics and prevent unwanted responses.	Amazon Bedrock User Guide – Prompt engineering concepts
Use Amazon Bedrock's guardrails to detect and filter harmful content.	Amazon Bedrock Guardrails
Use Bedrock Evaluations to measure bias.	Amazon Bedrock Evaluations
Use Amazon SageMaker Clarify to detect bias, increase transparency, and explain predictions for your fine-tuned and self-trained AI models. This helps you better understand your data and models.	Amazon SageMaker Clarify
Develop and maintain a bias testing dataset that includes representative test cases across demographic groups, geographic regions, and other sensitive attributes relevant to your use case. Run these test cases periodically and after model updates.	Use prompt datasets for model evaluation in Amazon Bedrock

1.2.6 Sensitive information disclosure

Your generative AI system might inadvertently include sensitive information in its responses, such as:

- Private business data
- Confidential information
- Internal system details

Unintended information disclosure can affect:

- Data privacy
- Regulatory compliance
- Business confidentiality
- Customer trust

Causes

Sensitive information disclosure can be caused by:

- Inadvertent inclusion of sensitive data in the training set or RAG database
- Exposure of personal information or trade secret
- Sensitive content reproduced from AI system memory

Relationships

Relates to	Item
AWS responsible AI dimensions	Privacy and security, Transparency
OWASP GenAI Top 10	Sensitive Information Disclosure (LLM02:2025)
MITRE ATLAS	LLM Data Leakage (AML.T0057)

Mitigations

Mitigations or controls	References
Use Amazon Bedrock Guardrails to detect and filter structured sensitive information in model inputs and outputs, such as personally identifiable information (PII), protected health information (PHI).	Amazon Bedrock Guardrails Remove PII from conversations by using sensitive information filters
Implement data classification scanning and access controls on the data sources connected to your AI system to prevent disclosure of company-confidential or proprietary information.	Implement effective data authorization mechanisms to secure your data used in generative AI applications

Mitigations or controls	References
Create and manage strict access controls for Amazon Bedrock API access.	Identity and access management for Amazon Bedrock
If you implement model invocation logging for the LLM or custom logging logic in your application, make sure to mask sensitive information in your log data.	Amazon CloudWatch - Help protect sensitive log data with masking
Protect your training and fine-tuning data by following security best practices.	AWS Well-Architected – Data Protection
Monitor personally identifiable information (PII) in your data when you train models, fine-tune them, or use retrieval-augmented generation (RAG).	Amazon Macie – Discover and protect your sensitive data at scale
Remove, mask, or tokenize personally identifiable information (PII) or sensitive data before you use it for training, fine-tuning, or retrieval-augmented generation (RAG).	Amazon Macie – Discover and protect your sensitive data at scale

Practical guidance

1. Implement least privilege for identities associated with agents and tool services.
2. Where supported by the tool service ensure that communications to tool services or agents are authorized by the end user.
3. Customers building their own tool services should consider propagating end-user identities separately; ensuring these identities can be validated and are not revealed to unauthorized third parties.

1.2.7 Hallucination

Large language models (LLMs) might generate responses that sound plausible but contain incorrect or fabricated information, also known as hallucinations. This can happen when the model:

- Lacks accurate information about a topic
- Makes assumptions beyond its training data
- Creates connections that aren't valid

In financial services operations: inaccurate responses can affect:

- Customer trust
- Business decisions
- Service quality
- Regulatory compliance

Causes

- Limited training data
- Missing domain expertise
- Insufficient fact validation
- Complex topic interactions

Relationships

Relates to	Item
AWS responsible AI dimensions	Veracity and robustness, Explainability, Transparency
OWASP GenAI Top 10	Misinformation (LLM09:2025)

Mitigations

Mitigations or controls	References
Use prompt engineering techniques to guide the model toward appropriate topics and prevent unwanted responses.	Amazon Bedrock User Guide – Prompt engineering concepts
Use Retrieval-Augmented Generation (RAG) to enhance your model responses with information from trusted knowledge bases.	Amazon Bedrock User Guide – Knowledge Bases

Mitigations or controls	References
Detect hallucinations in your retrieval augmented generation (RAG) and agent-based systems to improve response accuracy and reliability.	Detect hallucinations for RAG-based systems Reducing hallucinations in LLM agents
For internal AI systems, validate outputs with human review before business use (human-in-the-loop).	Get user confirmation before invoking action group function AWS re:Invent 2025 - Implementing Human-in-the-Loop Controls for Multi-Agent AI Systems Implement human-in-the-loop confirmation with Amazon Bedrock Agents
Automated Reasoning checks in Amazon Bedrock Guardrails uses automated reasoning to verify that natural language content complies with your defined policies. This mathematical verification helps ensure that your content strictly follows your guardrails.	Automated Reasoning checks in Amazon Bedrock Guardrails
You can use Amazon Bedrock Guardrails to detect and filter hallucinations in model responses by performing contextual grounding checks when you provide a reference source and query.	Contextual grounding check
Implement response disclaimers in customer-facing applications, to inform end users that AI-generated responses should be verified for critical decisions.	AWS Well-Architected Framework Generative AI Lens - Implement guardrails to mitigate harmful or incorrect model responses

1.2.8 Prompt injection

Users might attempt to manipulate prompts to influence system output. Manipulated prompts may:

- Bypass system controls
- Influence model responses
- Alter intended behaviours

Prompt injection can affect:

- Regulatory compliance
- Service accessibility
- Business reputation

Causes

AI system may be vulnerable to prompt injection due to:

- Insufficient input validation
- Overly permissive system prompts
- Direct prompt exposure
- Inadequate output filtering

Relationships

Relates to	Item
AWS responsible AI dimensions	Privacy and security, Controllability, Safety
OWASP GenAI Top 10	Prompt Injection (LLM01:2025)
MITRE ATLAS	LLM Prompt Injection (AML.T0051)

Mitigations

Mitigations or controls	References
Implement prompt engineering aligned with best practices to avoid prompt injection attacks.	Best practices to avoid prompt injection attacks
Input Validation – Before you send user input to Amazon Bedrock or the tokenizer, validate and sanitize it by removing special characters or using escape sequences. Make sure the input matches your expected format.	Prompt injection security

Mitigations or controls	References
Secure Coding Practices – To protect your resources, use secure coding practices such as use parameterized queries, avoid string concatenation for input, grant minimal access privileges to resources.	Prompt injection security
Security Testing – Test your applications regularly for prompt injection and other security vulnerabilities. Use penetration testing, static code analysis, and dynamic application security testing (DAST).	Prompt injection security
Stay Updated – Keep your Amazon Bedrock SDK, libraries, and dependencies current to receive the latest security patches and updates.	Prompt injection security
Use Amazon Bedrock Guardrails to detect and block user inputs that attempt to override system instructions through prompt attacks.	Amazon Bedrock - Prompt injection security

1.2.9 Excessive agency

Excessive agency refers to a generative AI system taking actions outside its intended scope, such as:

- Unauthorized operations
- Unplanned system changes
- Unexpected communications
- Unintended transactions

This risk is particularly relevant when the AI system is integrated with external systems, APIs, or can execute commands.

Causes

Excessive agency may occur when too much autonomy or control over systems is given to the AI system:

- Lack of AI system constrains
- Missing human oversight

Relationships

Relates to	Item
AWS responsible AI dimensions	Controllability, Safety, Privacy and security, Governance
OWASP GenAI Top 10	Excessive Agency (LLM06:2025)
MITRE ATLAS	Manipulate AI Model (AML.T0018) Impersonation (AML.T0073)

Mitigations

Mitigations or controls	References
Use Amazon Bedrock AgentCore to manage complex tasks and connect securely with AWS and third-party services.	Amazon Bedrock AgentCore – Overview
When you integrate plugins with external systems, grant only the minimum permissions required for them to function.	AWS Well-Architected Framework Generative AI Lens - Implement least privilege access and permissions boundaries for agentic workflows

Mitigations or controls	References
For internal AI systems, validate outputs with human review before business use (human-in-the-loop).	Get user confirmation before invoking action group function AWS re:Invent 2025 - Implementing Human-in-the-Loop Controls for Multi-Agent AI Systems Implement human-in-the-loop confirmation with Amazon Bedrock Agents
Define and enforce explicit action boundaries in the agent configuration, specifying which operations the agent is permitted to perform, and which are explicitly prohibited.	Amazon Bedrock AgentCore Policy: Control Agent-to-Tool Interactions
Implement audit logging of all actions taken by AI agents, including the reasoning chain that led to each action.	Observe your agent applications on Amazon Bedrock AgentCore Observability
Enforce transaction value thresholds and action boundaries on agent tool calls (for example to cap financial transaction amounts).	Amazon Bedrock AgentCore Policy: Control Agent-to-Tool Interactions
Monitor agent call rates and alarm upon exceeding defined thresholds.	Evaluate agent performance with Amazon Bedrock AgentCore Evaluations

1.2.10 Out-of-date training data

Your generative AI system might provide responses based on outdated training data. In financial services applications, current information is essential for:

- Customer decision-making
- Regulatory compliance
- Accurate service delivery

Causes

- Limited access to current information sources
- Missing data refresh mechanisms
- Insufficient use of retrieval augmented generation (RAG)
- Lack of regular model updates

Relationships

Relates to	Item
AWS responsible AI dimensions	Veracity and robustness, Controllability, Safety, Fairness
OWASP GenAI Top 10	Misinformation (LLM09:2025)
MITRE ATLAS	External Harms (AML.T0048)

Mitigations

Mitigations or controls	References
Use Retrieval-Augmented Generation (RAG) to enhance your model responses with information from trusted knowledge bases.	Amazon Bedrock User Guide – Knowledge Bases
Keep your knowledge bases up to date.	Sync your data with your Amazon Bedrock knowledge base

Mitigations or controls	References
For internal AI systems, validate outputs with human review before business use (human-in-the-loop).	Get user confirmation before invoking action group function AWS re:Invent 2025 - Implementing Human-in-the-Loop Controls for Multi-Agent AI Systems Implement human-in-the-loop confirmation with Amazon Bedrock Agents
Include data currency disclaimers in AI system responses where appropriate. Use source attribution in RAG-based response for end users to verify currency of information.	RetrieveAndGenerate API Reference

1.2.11 Unbounded consumption

Without proper controls, your generative AI system might experience:

- Excessive resource usage
- Unexpected costs
- Performance issues
- Service interruptions

Resource management issues can affect:

- System availability
- Response times
- Operating costs
- Service quality

Causes

AI system allowing users to conduct excessive and uncontrolled inferences.

Relationships

Relates to	Item
AWS responsible AI dimensions	Privacy and security
OWASP GenAI Top 10	Unbounded Consumption (LLM10:2025)
MITRE ATLAS	Denial of AI Service (AML.T0029)

Mitigations

Mitigations or controls	References
You can protect your LLM APIs and Amazon Bedrock-hosted LLMs by using AWS WAF and AWS Shield Advanced.	Securing PartyRock: How we protect Amazon Bedrock endpoints using AWS WAF
To protect your API endpoints, set maximum length limits for input requests when you use large language models (LLMs) directly or through Amazon Bedrock.	Applying rate limiting to requests in AWS WAF Throttle requests to your REST APIs in Amazon API Gateway

Mitigations or controls	References
You can protect your API endpoints by implementing rate limits and quotas for APIs that access large language models (LLMs), including those hosted on Amazon Bedrock.	Throttle requests to your REST APIs Applying rate limiting to requests in AWS WAF
Track, allocate, and manage your costs and usage for generative AI.	Track, allocate, and manage your generative AI cost and usage with Amazon Bedrock

Practical guidance

A quota in AWS Bedrock represents predefined limits on service usage. Bedrock has default quota on model inference based on token usage when invoking different foundation models. By optimizing the *max_tokens* parameter, you can efficiently utilize your allocated quota capacity. To help inform your decision about this parameter, you can use Amazon CloudWatch, which automatically collects metrics from AWS services, including token usage data in Amazon Bedrock (see also Optimizing the max_tokens parameter).

1.2.12 Supply chain vulnerabilities

LLM supply chains are vulnerable to various security risks, including:

- Compromised training data, models, and deployment platform integrity
- Third-party model and data tampering or poisoning
- Third-party package vulnerabilities
- Licensing issues
- Outdated models
- Weak model provenance
- Unclear terms and conditions regarding data privacy

Causes

Supply chain vulnerabilities may occur upon insufficient supply chain risk management:

- Lack of contractual controls
- Inadequate technical and organizational measures

Relationships

Relates to	Item
AWS responsible AI dimensions	Controllability, Safety, Privacy and security, Governance
OWASP GenAI Top 10	Supply Chain (LLM03:2025)
MITRE ATLAS	AI Supply Chain Compromise (AML.T0010)

Mitigations

Mitigations or controls	References
Control access to serverless models and marketplace models in Amazon Bedrock.	Controlling Access to Amazon Bedrock Marketplace Models Control over model access through IAM policies and Service Control Policies (SCPs)

Mitigations or controls	References
To onboard a model, follow these steps to evaluate legal, security, and compliance requirements: • Review the model's end-user license agreement (EULA) • Complete the procurement process • Follow your organization's security and compliance procedures • Assess model risk management (MRM) requirements • Document your evaluation findings • Get necessary approvals from stakeholders	Access Amazon Bedrock foundation models Set up Amazon Bedrock Marketplace
Update existing third-party risk management processes to continuously monitor model providers and third-party dependencies, including tracking vendor security advisories, model deprecation notices, and change to terms and conditions.	Access Amazon Bedrock foundation models
Maintain a model inventory that records the provenance, version, license terms, and risk assessment status of all models in use across the organization.	Access Amazon Bedrock foundation models

Practical guidance

Implement an allow-list of models using a Service Control Policy (SCP) for your AWS organization (see Managing access in AWS Organizations). This allows to centrally govern access to models vetted and approved by your organization.

Amazon Bedrock Evaluations can help to evaluate models against specific types of attacks by automating your test cases, scoring, reporting and to enable comparison of different models.

1.2.13 Improper output handling

Without proper validation, your generative AI system's responses might include:

- Unintended system commands
- Invalid application code
- Inappropriate content
- Unverified information

Insufficient output validation can lead to vulnerabilities such as:

- Remote code execution
- Cross-site scripting
- Server-side request forgery (SSRF)
- Privilege escalation

Causes

Lack of validation of AI system's responses.

Relationships

Relates to	Item
AWS responsible AI dimensions	Privacy and security, Controllability, Safety
OWASP GenAI Top 10	Improper Output Handling (LLM05:2025)
MITRE ATLAS	Manipulate AI Model (AML.T0018)

Mitigations

Mitigations or controls	References
Implement output validation rules specific to the expected response format. For example, if the AI system is expected to return structured data (JSON, SQL), validate the output against the expected schema before processing.	Application security – AWS Well-Architected Security Pillar
Apply context-specific output sanitization based on the downstream consumer. For example, apply HTML encoding for web applications, SQL parameterization for database queries, and command escaping for system integrations.	Application security – AWS Well-Architected Security Pillar

Practical guidance

1. Treat the model output as you would traditionally handle any other user input. Use the same validation and encoding mechanisms on model outputs as you used on any other user inputs.

2. Use Amazon Bedrock Agents to securely integrate with AWS native and third-party services and implement output encoding in the action group Lambda function under an Amazon Bedrock Agent. Encoding all output text presented to end-users makes it automatically non-executable by JavaScript or Markdown.

1.2.14 Training data and model poisoning

Training data and model poisoning refers to a security vulnerability where an attacker deliberately manipulates or corrupts the training data used to train AI models by:

- Introducing malicious samples to training dataset
- Modifying existing data
- Injecting biased information to the training dataset

Poisoned training data influences negatively the resulting model behaviour or performance as it may:

- Learn incorrect patterns
- Make biased decisions
- Create backdoor that can be exploited later

Causes

- Lack of change monitoring on training dataset
- Lack of control over used data sources
- Missing access control to training datasets

Relationships

Relates to	Item
AWS responsible AI dimensions	Veracity and robustness, Safety, Fairness, Privacy and security, Governance
OWASP GenAI Top 10	Data and Model Poisoning (LLM04:2025)
MITRE ATLAS	Poison Training Data (AML.T0020) RAG Poisoning (AML.T0070)

Mitigations

Mitigations or controls	Reference
When you train or fine-tune models, protect your training datasets by following data protection best practices for confidentiality, integrity, and availability.	AWS Well-Architected Machine Learning Lens – Protect against data poisoning threats AWS Well-Architected Machine Learning Lens – Protect sensitive data privacy

Mitigations or controls	Reference
Use trusted data sources for your training data. Implement audit controls that let you track and review changes, including who made them and when they occurred.	AWS Well-Architected Machine Learning Lens – Protect against data poisoning threats
Monitor your training data for pattern and distribution changes to detect data drift and assess its impact on prediction variance. Changes in data distribution can indicate potential data drift and help you identify unauthorized access attempts to your training data early.	Amazon SageMaker Model Monitor – Data quality AWS Well-Architected Machine Learning Lens – Evaluate data drift
Before deploying to production, compare your retrained model's performance against previous iterations using historical test data as a baseline to identify any negative impacts on results.	Amazon SageMaker AI – Model Registration and Deployment with Model Registry
Create a rollback plan by using versioned training data and models. This ensures that you can revert to a stable, working model if failures occur.	Amazon SageMaker AI Feature Store
Use low-entropy classification cases to monitor your model. Set threshold boundaries and identify unexpected classifications. Create alerts when your retrained model exceeds these thresholds or produces unexpected changes in classification patterns.	Amazon SageMaker Model Monitor
When consuming third-party models, evaluate the model performance testing procedures used by the vendors. For Amazon trained models, Amazon enhances transparency about its testing procedures though AWS AI Service Cards.	AWS AI Service Cards

1.2.15 Vector and embedding weaknesses

When using retrieval augmented generation (RAG), weaknesses in how vectors and embeddings are generated, stored, or retrieved can be exploited by malicious actors to:

- Inject harmful content
- Manipulate outputs
- Access sensitive information

Causes

- Too broad permissions on vector and embedding databases
- Lack of data validation
- Unverified sources of data

Relationships

Relates to	Item
AWS responsible AI dimensions	Veracity and robustness, Privacy and security, Safety
OWASP GenAI Top 10	Vector and Embedding Weaknesses (LLM08:2025)

Mitigations

Mitigations or controls	References
Apply the principle of least privilege to control access to your vector and embedding database. Only grant users and services the minimum permissions they need to perform their tasks.	Amazon Bedrock Knowledge Bases - Permissions
Validate your knowledge base data sources.	Sync your data with your Amazon Bedrock knowledge base
Add data only from trusted sources to knowledge bases.	Amazon Bedrock User Guide – Knowledge Bases
Monitor and log all activities in knowledge base control plane.	Monitor Amazon Bedrock API calls using CloudTrail
Enable encryption at rest and in transit for vector and embedding databases.	Data encryption in Amazon Bedrock

Mitigations or controls	References
Implement access controls at the document or record level within knowledge bases where different users or applications should only have access to specific subsets of data. Use Amazon Bedrock Knowledge Bases metadata filtering to enforce data segmentation.	Multi-tenancy in RAG applications in a single Amazon Bedrock knowledge base with metadata filtering

Practical guidance

Use Amazon Bedrock Knowledge Bases to implement Retrieval Augmented Generation (RAG) instead of self-building such a system. Amazon Bedrock Knowledge Bases is a fully managed capability with in-built session context management and source attribution that helps you implement the entire RAG workflow from ingestion to retrieval and prompt augmentation without having to build custom integrations to data sources and manage data flows. Relying on this proven capability allows you to securely connect FMs and agents to additional data sources.

Appendix A: AWS Responsible AI

AWS has eight responsible AI dimensions:

Responsible AI Dimension	Description
Fairness	Considering impacts on different groups of stakeholders.
Explainability	Understanding and evaluating system outputs.
Privacy and security	Appropriately obtaining, using, and protecting data and models.
Safety	Preventing harmful system output and misuse.
Controllability	Having mechanisms to monitor and steer AI system behavior.
Veracity and robustness	Achieving correct system outputs, even with unexpected or adversarial inputs.
Governance	Incorporating best practices into the AI supply chain, including providers and deployers.
Transparency	Enabling stakeholders to make informed choices about their engagement with an AI system.

Responsible AI principles serve as essential guideposts for conducting effective risk assessments. By considering key responsible AI elements such as described above, organizations can structure their risk assessments to specifically target critical areas of concern. This alignment ensures that risk evaluations go beyond mere technical considerations to encompass broader societal implications, ethical concerns, and potential impacts on diverse user groups. As a result, responsible AI considerations help focus risk assessments on the most relevant and meaningful aspects of AI system deployment, making the assessment process more comprehensive and purposeful.

Appendix B: Responsible AI dimensions and risk mapping

Responsible AI dimension	Applicable risks
Fairness	• Off-topic and inappropriate output • Out of date training data • Misinformation through inadvertent or malicious action • Model output is abusive or harmful • Model output is biased • Data and model poisoning
Explainability	• Hallucination
Privacy and security	• Prompt injection • Sensitive information disclosure • Supply chain vulnerabilities • Data and model poisoning • Improper output handling • Excessive agency • Vector and embedding weaknesses • Unbounded consumption
Safety	• Non-compliant output • Off-topic and inappropriate output • Out of date training data • Misinformation through inadvertent or malicious action • Model output is abusive or harmful • Prompt injection • Supply chain vulnerabilities • Data and model poisoning • Improper output handling • Excessive agency • Vector and embedding weaknesses
Controllability	• Non-compliant output • Off-topic and inappropriate output • Out of date training data • Misinformation through inadvertent or malicious action • Prompt injection • Supply chain vulnerabilities • Improper output handling • Excessive agency
Veracity and robustness	• Out of date training data • Misinformation through inadvertent or

Responsible AI dimension	Applicable risks
	malicious action • Hallucination • Data and model poisoning • Vector and embedding weaknesses
Governance	• Supply chain vulnerabilities • Training data and model poisoning • Excessive agency
Transparency	• Hallucination • Non-compliant output • Model output is biased • Sensitive information disclosure

Document revisions

Date	Description
March 2026	First version.