Amazon Polly

AWS AI Service Cards

AWS AI Service Cards: Amazon Polly

Table of Contents

Amazon Polly

An AWS AI Service Card explains the use cases for which the service is intended, how machine learning (ML) is used by the service, and key considerations in the responsible design and use of the service. A Service Card will evolve as AWS receives customer feedback, and as the service progresses through its lifecycle. AWS recommends that customers assess the performance of any AI service on their own content for each use case they need to solve. For more information, please see AWS Responsible Use of AI Guide and the references at the end. Please also be sure to review the AWS Responsible AI Policy, AWS Acceptable Use Policy, and AWS Service Terms for the services you plan to use.

This Service Card applies to the releases of Amazon Polly that are current as of Feb 11, 2026.

Overview of Amazon Polly

Amazon Polly is a cloud service that generates voice on demand, converting text in a supported language to an audio stream. Polly can also accept other written form texts such as symbols or phonetic language that are prevalent in non-Latin languages such as Mandarin, Korean and Japanese. Using deep learning and generative AI technologies, Polly converts text into natural-sounding speech. Polly provides dozens of lifelike voices across 60 languages for customers to build speech-enabled applications. This AI Service Card applies to the use of Amazon Polly via the Polly console and Polly APIs.

Amazon Polly generates speech output based on plain text or SSML enriched TTS prompts like this:

```
<speak>
    <prosody volume="x-loud">
        Sally sells seashells by the seashore.
    </prosody>
</speak>
```

Amazon Polly offers APIs to convert text strings to speech and audio streams that AWS customers can embed in their audio and speech workflows. For advanced developers, Amazon Polly offers several tools and customization mechanisms (e.g. SSML tags) through which they can further enhance the speech and audio output.

The generated speech is considered "effective" when a skilled human evaluator confirms that it accurately conveys the content of the input text prompt and is free from defects or audio errors

(for example, no glitches, unnatural gaps, cut-offs, or mispronunciations). We recommend that customers incorporate human judgment into their workflows to evaluate speech effectiveness. This can be done either through direct case-by-case assessment (when using the AWS Console UI as a standalone tool) or by setting effective guardrails on number of errors or defects while using automated testing on customers' workloads.

The "overall effectiveness" of any TTS model for a specific use case is based on the percentage of use-case specific inputs for which the model returns an effective result. Customers should define and measure effectiveness for their specific use cases for the following reasons. First, the customer is best positioned to know which voice and inputs will best represent their use case and should therefore be included in an evaluation dataset. Second, each TTS engine (Neural, Long-Form, or Generative) and voice may respond differently to the same inputs. Customers can fine-tune the prompt using Speech Synthesis Markup Language (SSML) Tags to convey different tone and intonation or to adjust the pronunciation, e.g. a string of numbers to be pronounced as address or as a phone number. As an example, to control and generate different speech output, customers can use SSML to adjust the emphasis of a sentence.

```
<speak> I <emphasis level= "strong"> understand your concern </emphasis> and I'll do my
 best to help. </speak>
```

and

```
<speak> I understand your concern and I'll <emphasis level= "strong"> do my best </
emphasis>  to help. </speak>
```

Since different TTS SSML tags affect pronunciation, pacing, and emphasis in various ways, customers should experiment, as needed, to determine how to best adjust input text and SSML to achieve the desired spoken output. Customers can also select different voices and engines to find the best match of voices that are most suitable for their AI applications.

As with all AI/ML solutions, Amazon Polly must overcome issues of intrinsic and confounding variation. Intrinsic variations would include (but not limited to) text prompts, variable dynamic content (names, numbers, dates), SSML tags whereas the confounding variations will include formatting (string of numbers as phone numbers or digits), word ordering and punctuation differences.

Intended Use Cases and Limitations

Amazon Polly is intended for converting text into natural-sounding speech. It is designed for generating human-like speech from text input, enabling both real-time audio streaming and offline audio file generations. AWS customers can access Polly text-to-speech services through Polly console or APIs. The service supports plain text and SSML (Speech Synthesis Markup Language) inputs, allowing fine-tune control of the speech output such as pronunciations, pausing and pacing, emphasizing, pitch, loudness, speaking rate, and so on (see Supported SSML tags for full list). The service also provides Speechmarks functionality for precise text-to-audio alignment, enabling applications to synchronize visual and/or audio elements with speech. Polly supports simultaneous text-to-speech synthesizing with predefined quota limits. To accommodate large volume speech synthesizing requests, customers can raise a ticket to Polly team to request the increase of their quota limits.

Polly offers multiple voice options across a set of languages and regional accents (see list of Supported Languages). Polly also offers Polyglot voices that are capable of speaking multiple languages with the same voice (see Bilingual Voices for details).

In addition, Polly offers managed Brand Voice service. Polly can help customers create custom branded TTS voices based on their chosen professional voice talent recordings and offer access to this custom TTS voice only to the respective customer. Customers cannot create brand voice on their own through Polly.

Polly's TTS capabilities enable a range of applications including but not limited to followings:

- **Customer Experience:** Build natural-sounding IVR (Interactive Voice Response) systems that reduce caller frustration and create low-latency chatbots for real-time customer interactions. Customers can store and replay Polly speech output to prompt callers through interactive or automated voice response systems
- **Learning and Development:** Create engaging e-learning modules with accurate multilingual voice delivery and develop interactive language learning apps with voices like native speakers
- **Game, Media & Entertainment:** Produce audio content such as podcasts and audiobooks with consistent voice quality, create human-like voices for characters in animated content and game, as well as generate multilingual voice-overs of the content for global audiences
- **Publishing:** Transform text content to engaging audio for hands-free listening experiences and create narration with emotional ranges and expressiveness for news, article, stories, and books

- **Voice-enabled Applications:** Create speech content for smart devices, public announcements, and other voice-enabled applications
- **Accessibility:** Convert text articles to audio for visually impaired users and generate clear audio versions of written content and digital documents for alternative modality

When assessing an Amazon Polly voice for a particular use case, we encourage customers to specifically define the use case by considering at least the following factors: the **business problem** being solved; the **stakeholders** in the business problem and deployment process; the **workflow** that solves the business problem, with the model and oversight as components; key system **inputs and outputs**; the types of **errors** possible; and the relative impact of each.

Consider the following use case of utilizing Amazon Polly voice as a creative tool to help a customer support designer generate natural-sounding IVR responses. The **business goal** is to **deliver clear, natural, and expressive spoken responses to callers through the IVR system**. The stakeholders include the IVR system designer, who wants to configure and optimize TTS responses to handle customer queries effectively; the customer service manager, who wants to use the system's spoken responses to improve caller satisfaction; and the callers or listeners using the IVR system, who can hear the intended information announced clearly through natural-sounding voices in respective languages and accents. The workflow is:

1. The designer creates text prompts for IVR responses
2. Prompts are sent to Amazon Polly for speech synthesis
3. The designer listens to and reviews the generated speech for the IVR system
4. The designer makes changes to the text prompts, including adding the required SSML tags, and regenerates the speech for the IVR system
5. The designer finalizes the speech prompts for the IVR system after completing 1 more round of listening assessment
6. The designer deploys the finalized IVR text prompts to the IVR system that can call Polly APIs for the responses in realtime during end-consumer calls
7. The IVR system receives the appropriate speech from Polly in near-realtime (per the latency of the TTS engine).

Input prompts contain information regarding caller intents, menu options, dynamic content such as account or transaction data, and system messages. Output contains information regarding spoken

rendition of the prompts, including pronunciation, prosody, emphasis, pacing, and any voice-specific expressive features added by Polly.

Different error types, along with use cases and an example of the customer experience while using the text-to-speech generation service is listed below:

Error type	Use Case	Example customer experienc e
Mispronunciation	Pronouncing the words that is consistent with the country-specific terminology (e.g. US versus UK)	Mispronunciations of words can lead to customer confusion. For example, in the US, the word 'Advertisement' is pronounced as ad-VER-ti s-muhnt whereas the same word is pronounced as uhd-VER-tis-muhnt. Similarly, US speakers often stress the first syllable in the word 'Garage', as GAE-rahj compared to the British pronunciation as guh-RAHZH).
Omission or truncation	Navigation system cutting off street names	Customers following turn-by-turn navigation may make a wrong turn or miss an exit if a street or avenue name is omitted by the text-to-speech system.
Unnatural prosody or pacing	Customer service IVR system	A stressed customer calling up their banking IVR system may be greeted with a cheerful text-to-speech voice, leading to trust bursting customer experience.

Error type	Use Case	Example customer experienc e
Incorrect handling of dynamic content or SSML tags	Using <prosody> SSML tag to slow the speaking rate	Due to incorrect handling of the <prosody> SSML tag, the text-to-speech output may speak at regular rate instead of speaking slowly and it may be difficult for a hard-of-h earing customer to follow the speech output.

With this in mind, we would expect the IVR system designer to test an example prompt in the Amazon Polly console or via API and review the completion.

Example Prompt:

```
<speak>
Hello, thank you for calling Polly Bank. Please listen carefully to the following
 options.
<emphasis level="strong">Press 1</emphasis> for account information,
<emphasis level="strong">Press 2</emphasis> for card services, or
<emphasis level="strong">Press 3</emphasis> to speak with a representative.
</speak>
```

Example output: {Audio generated by Amazon Polly for the IVR prompts}.

Assessing the TTS output generated in the first step, we observe: no unsafe, unfair, or otherwise inappropriate content; all required elements of the prompt were included (voice style, clarity, pacing); and variations in intonation that could be adjusted using SSML tags or alternative voice parameters.

When using SSML tags or modifying input prompts to adjust voice characteristics, Amazon Polly will not exactly reproduce all other elements of the output (for example, in the previous scenario, changing the emphasis level will slightly alter perceived emphasis on some words).

After continued experimentation in the Amazon Polly console or via API, the customer should finalize their own measure of effectiveness based on the impact of errors, run a scaled-up test via

the API, and use the results of human judgments (with multiple judgments per test prompt) to establish a benchmark effectiveness score.

Amazon Polly can be integrated into an array of systems such as a contact center, conversational assistants, creative AI systems, and human-AI collaboration systems. For more technical information about how Amazon Polly may be integrated into AI systems, see the Amazon Polly Developer Guide and Amazon Polly Features List.

Amazon Polly has several limitations that require careful consideration while being deployed in customer workloads:

- **Appropriateness for Use:** Because of non-deterministic nature of speech synthesis, Amazon Polly may produce inaccurate speech output. Customers should evaluate outputs for accuracy and appropriateness for their use case, especially if they will be directly surfaced to end users. Additionally, if Amazon Polly is used in customer workflows that produce consequential decisions, customers must evaluate the potential risks of their use case and implement appropriate human oversight, testing, and other use case-specific safeguards to mitigate such risks. For more information, see the AWS Responsible AI Policy.
- **Language Processing:** The system has certain limitations in handling mixed-language content and specialized terminology. For mixed-language scenarios, such as text containing quotes or phrases from multiple languages within the same input, the system may produce inconsistent pronunciations or unexpected language switches. Technical vocabulary and professional terms-of-art across diverse fields (e.g., medical, legal, scientific domains) may be mispronounced due to incomplete specialized pronunciation dictionaries.
- **Voice Diversity:** Our current voice offerings represent carefully selected examples within the full spectrum of human vocal diversity. Each localized voice (such as en-US Matthew) should be understood as a single sample point rather than being fully representative of that language variant's speaker population. Speaker populations vary widely in accents, speech patterns, and vocal characteristics within any given language or region, reflecting the natural diversity of human communication.
- **Inputs and Outputs:** Amazon Polly supports up to 3,000 characters for real-time synthesis and up to 100,000 characters for asynchronous synthesis. For full details and setup instructions, see the Amazon Polly Developer Guide.
- **Safety Filters:** While Amazon Polly does not generate unsafe content by design, it does not filter inappropriate or sensitive input text. Customers should implement additional content moderation or filtering mechanisms in their applications to help make generated speech suitable

for their audience. For example, customer can use Amazon Bedrock Guardrails to implement customizable safeguards to help filter out harmful content.

- **Supported Languages:** Amazon Polly supports an expanding list of languages and voices. For the most up-to-date information on available languages and features, see Amazon Polly Features.

Design of Amazon Polly

Machine learning

Amazon Polly performs text-to-speech using machine learning, specifically, generative AI and TTS technologies. Its neural TTS models contain hundreds of millions of parameters, while its generative models leverage billion-plus-parameter transformers that convert raw text into speech codes. These codes are then processed by a convolution-based decoder, which generates waveforms in an incremental, streamable manner. Amazon Polly is available pursuant to the AWS Customer Agreement or other relevant agreements with AWS.

We say that an Amazon Polly model exhibits a particular "behavior" when it generates the same kind of output for the same kinds of inputs and configuration (for example, pause, emphasis, volume, pronunciation, etc.). For a given model architecture, the control levers that we have over the behaviors are primarily:

1. The unlabeled pre-training voice corpus (which we filter to exclude duplicates, low-quality voice, and content that violates internal design policies).
2. The labeled fine-tuning corpus.
3. Voice configuration options, which allow us to guide pitch, emphasis, and intonation.

Our development process exercises these control levers as follows:

1. We pre-train the TTS LLM using curated publicly available and proprietary data spanning diverse voices, languages, and speaking styles
2. We adjust model weights via supervised fine-tuning (SFT) to increase alignment between Polly voices and our design goals

Performance expectations

Performance of Polly will differ between applications, even if they support the same use case. Consider two applications A and B: Application A delivers real-time transport announcements

in public spaces, requiring clear enunciation and audibility in noisy environments. Application B uses the same voice to read bedtime stories, where softness, pacing, and emotional tone are more critical. Even though the voice and system are identical, the differing demands of each use case led to different perceived performance. Developers using the Amazon Polly service should consider using different TTS engines as well as SSML tags to make sure that the resulting output is effective for the use case they are empowering.

Now consider two versions of Application B, each using a different voice. One voice is neutral and flat; the other is expressive and warm. Although the task—reading bedtime stories—remains the same, the listener experience and engagement can vary significantly depending on the voice's characteristics. This illustrates how both use case context and voice selection influence the effectiveness and quality of TTS output, even within the same deployment.

As a result, the overall utility of Amazon Polly will depend both on the selected voice and on the workflows it enables. Since performance results depend on a variety of factors including Amazon Polly's engine itself, the customer workflow, and the evaluation dataset, we recommend that customers test Amazon Polly using their own content.

With AI/ML services, performance results vary with the evaluation dataset used to generate the results. We recommend that customers test all AI/ML services from any vendor (not just AWS) on their own content. We measure the overall performance of the service along four effectiveness criteria: intelligibility, naturalness, prosody, and contextual appropriateness. The following sections contain information related to the methodology, metrics, datasets, and results used to evaluate Amazon Polly.

Test-driven methodology

Amazon Polly's performance is evaluated using multiple complementary approaches to help provide consistent, high-quality synthetic speech across a wide range of use cases. Each voice is assessed individually against quality criteria including **intelligibility, naturalness, prosody, and contextual appropriateness**.

1. **Prompt diversity:** Each voice is synthesized across a wide range of text prompts varying in domain, length, and complexity. This helps test the model for different use cases, from real-time announcements to narrative storytelling.
2. **Human evaluation:** Ratings are collected from trained evaluators to assess the clarity, naturalness, expressiveness, and contextual appropriateness of the synthesized speech.
3. **Automated metrics:** Complementary metrics, such as error rate analysis, help identify specific issues in pronunciation, prosody, or misalignment with input text.

4. **Cross-use-case benchmarking:** Voices are tested in scenarios representative of intended applications to help make performance consistent regardless of the context.

This test-driven methodology helps make the voices clear, natural, expressive, and robust and automated evaluation helps ensure all deployed voices meet stringent criteria for clarity, naturalness, expressiveness, and robustness.

Voice Quality Assurance (VoiceQA): For each voice/locale we release, we synthesize 1,000+ audio samples from a range of text types (e.g., customer service, fiction, non-fiction) which were evaluated by 50+ native listeners to check for various kinds of errors, such as mispronunciations, missing text (cutoffs), and extra text (hallucinations).

Customer Satisfaction (CSAT): A sample of 50 or more native listeners evaluate 15 audio samples split into three genres and score how much they like the voice across a range of characteristics in each genre. They also provide an overall score based on the complete set of audio samples.

Amazon Polly frequently launches new TTS voices in existing languages and new languages for AWS customers to use in their speech workflows. Amazon Polly maintains a high-quality bar for voices with Standard TTS and Neural TTS engines, with most of the TTS voices scoring a voice CSAT of 5.5 (on a scale of 1-7) in human listening evaluations and a critical pronunciation error rate of less than 5%. The Generative TTS/GTTS engine, launched in May 2024, has a voice CSAT of 5.8 (on a scale of 1-7) and critical pronunciation errors less than 3%. We continue to maintain and enhance the quality bar for all voices available on Amazon Polly with every model update.

For each voice, the **CSAT** is calculated by having at least 50 human evaluators listen to 15 audio samples and rate their satisfaction on a 1 to 7 scale, where 1 means very dissatisfied and 7 means very satisfied. The CSAT score is the average of the ratings from all evaluators.

For Voice QA, human evaluators listen to between 1,000 and 2,000 prompts and flag any critical errors. These are categorized as inconsistent speaker identity, cutoff, audio glitch, hallucination, inappropriate persona, intonation, pausing, pronunciation, and text normalization. The **Voice QA Error%** is the proportion of prompts with critical errors.

Voice Quality results by Gender

Gender	Voice QA (%)	CSAT (out of 7)
Female	1.20	5.91

Gender	Voice QA (%)	CSAT (out of 7)
Male	1.47	5.98

Voice Quality results by Engine

Engine	Voice QA (%)	CSAT (out of 7)
Generative	1.79	6.15
Long-form	1.53	6.09
Neural	0.77	5.75

We use multiple datasets and human teams to evaluate the performance of Amazon Polly models. No single evaluation dataset suffices to completely capture performance. This is because evaluation datasets vary based on use case, intrinsic and confounding variation, the quality of ground truth available, and other factors. Our development testing involves automated testing against proprietary speech datasets; human evaluation of generated speech for intelligibility, naturalness, and expressiveness; manual red teaming; and more. Our development process examines Amazon Polly's performance using all these tests and takes steps to improve the model and the suite of evaluation datasets.

Automated testing provides apples-to-apples comparisons between candidate Polly models by substituting an automated "assessor" mechanism for human judgment, which can vary. Automated assessments can take several forms. One form is to evaluate generated speech using multiple speech datasets to assess intelligibility, naturalness, prosody, and contextual appropriateness. One industry standard measurement of this form is Mean Opinion Score (MOS).

Veracity

Amazon Polly synthesizes speech from text using underlying linguistic representations such as phonemes and prosody markers. The system is designed to render input text accurately according to these representations. Customers can use SSML to control pronunciation, speaking rate, pitch, emphasis, and language for specific words to improve alignment with intended content. Evaluations of generated speech consider intelligibility, naturalness, prosody, and

contextual appropriateness across voices and languages using multiple licensed datasets and listener studies. Performance is measured using metrics such as Mean Opinion Score (MOS), naturalness ratings, error rates, and listener preference studies, along with additional internal benchmarks, to provide high-quality synthetic speech.

Fairness

Our goal is for Amazon Polly to accurately generate audio from text for diverse range of human voices for different speaker groups. We define speaker groups by demographic attributes such as age, gender and locales. We test on dataset with those demographic labels. As an example, on a 1,000 to 2,000 prompts per local dataset across 29 locales, male voices show slightly higher Voice QA percentage (0.27 percentage points higher) and marginally better customer satisfaction (0.07 points higher on the 7-point scale). Both genders have similar performance with high CSAT scores (>5.9) and low Voice QA percentages (<1.5%).

Robustness

Robustness refers to a model's ability to perform reliably across a wide range of inputs, variations of spellings, or context-dependent pronunciation (e.g. *'live'* a life vs *'live'* telecast of a game), conditions, and speaker variations. Amazon Polly maximizes robustness by training on large and diverse datasets, including recordings from speakers of different ages, genders, accents, and speaking styles. Training on both formal and conversational speech helps the voices perform well across a variety of content types. Rigorous evaluation across voices, languages, and scenarios helps provide consistent, high-quality performance in real-world applications. We recommend that customers use SSML controls, voice and language selection, and speech parameter customization, along with maintaining workflow consistency and conducting periodic testing, to help support robustness in their own applications. In our test on a dataset of 1,000 to 2,000 prompts per local dataset across 29 locales. The Generative engine achieved 1.79% Voice QA Error rate and 6.15 CSAT score. The Long-form engine achieved 1.53% error rate and 6.09 CSAT score. The Neural engine achieved 0.77% error rate and 5.75 CSAT score.

Privacy

Amazon Polly is a managed service and prompts and generations are never shared across customers. AWS does not store nor use inputs or outputs generated through the Amazon Polly service to train Amazon Polly service. For more information, see Section 50.3 of the AWS Service Terms and the AWS Data Privacy FAQs. For service-specific privacy information, see Data Privacy in the Amazon Polly FAQs.

Security

Customers can use AWS PrivateLink to establish private connectivity between customized Amazon Polly models and on-premises networks without exposing customer traffic to the internet. For example, Amazon CloudWatch can help track usage metrics that are required for audit purposes, and AWS CloudTrail can help monitor API activity and troubleshoot issues as Amazon Polly is integrated with other AWS systems. Customer data is always encrypted in transit and at rest. Amazon Polly is not storing inputs nor outputs for text-to-speech synthesis functionality. The only customer provided content that is stored in the service is "pronunciation lexicons" encrypted with service owned encryption keys. Additionally, customers can also choose to store output generated through asynchronous synthesis functionality in their own encrypted Amazon Simple Storage Service (Amazon S3) bucket.

Transparency

Amazon Polly provides information to customers through the following locations: the Amazon Polly AWS page, Polly FAQs, AWS educational channels (for example, blogs, developer classes), the AWS Console, and this Service Card. We accept feedback through customer support mechanisms such as account managers. Where appropriate for their use case, customers who incorporate Amazon Polly in their workflow should consider disclosing their use of ML to end users and other individuals impacted by the application, and customers should give their end users the ability to provide feedback to improve workflows. In their documentation, customers can also reference this Service Card.

Governance

We have rigorous methodologies to build our AWS AI services responsibly, including a working backwards product development process that incorporates Responsible AI at the design phase, design consultations, and implementation assessments by dedicated Responsible AI science and data experts, routine testing, reviews with customers, best practice development, dissemination, and training.

Deployment and performance optimization best practices

We encourage customers to build and operate their applications responsibly, as described in AWS Responsible Use of AI Guide. This includes implementing Responsible AI practices to address key dimensions including controllability, safety, fairness, veracity, robustness, explainability, privacy, security, transparency, and governance.

The performance of any application using Amazon Polly depends on the design of the customer workflow, including the factors discussed below:

1. **Effectiveness Criteria:** Customers should establish clear guidelines for which AI use cases they will implement, acceptable input/output parameters for each use case, and how human judgment should be applied in decision-making. These guidelines should ensure systems are controllable, safe, fair, and aligned with key customer requirements.
2. **Configuration:** Customers can configure Amazon Polly using factors such as:
   a. **Input text quality:** Ensure input text is well-structured, correctly punctuated, and contextually appropriate to avoid unnatural or incorrect speech output.
   b. **SSML usage:** Use SSML consistently to help control pronunciation, pauses, speech rate, pitch, and emphasis, particularly in content with complex or dynamic intonation requirements.
   c. **Voice selection:** Choose voices across genders, accents, and styles that best fit the application, such as a warm, conversational voice for customer service or a clear, formal voice for announcements.
   d. **Pronunciation lexicons:** Define custom lexicons for domain-specific terms, acronyms, or names with uncommon pronunciations.
   e. **Content filtering and sensitive content handling:** Implement controls to help identify and filter out problematic, harmful, or sensitive input text, particularly for regulated or public-facing applications. Customers should consider which configuration choices will provide the most effective results for their specific workflows. More details are available in the Amazon Polly Developer Guide.
3. **Human Oversight:** If a customer's application workflow involves a high risk or sensitive use case, such as a decision that impacts an individual's rights or access to essential services, human review should be incorporated into the application workflow where appropriate and periodically retested.
4. **Performance Drift:** A change in the types of inputs that a customer submits, application domain, or Amazon Polly updates might lead to different outputs. To address these changes, customers should consider periodically retesting the performance of Amazon Polly and adjust their workflow if necessary.
5. **Latency:** Polly SynthesizeSpeech is a streaming API designed for low latency use cases. To minimize delay, audio should be played as soon as the first chunk of data is received. Waiting for the entire response before playback can increase latency.
6. **Workflow Consistency:** Customers should define policies around input preparation, voice usage, SSML application, and content review to ensure consistency across outputs. These policies

should be applied fairly across different user groups and contexts to avoid unintentional bias or degraded experience.

Further information

- For service documentation, see Amazon Polly Developer Guide.
- For details on privacy and other legal considerations, see the following AWS policies: Acceptable Use, Responsible AI, Legal, Compliance, and Privacy.
- For help optimizing workflows, see Generative AI Innovation Center, AWS Customer Support, AWS Professional Services, Ground Truth Plus, and Amazon Augmented AI.
- If you have any questions or feedback about AWS AI service cards, please complete this form.

Glossary

Controllability: Steering and monitoring AI system behavior.

Privacy & Security: Appropriately obtaining, using and protecting data and models.

Safety: Preventing harmful system output and misuse.

Fairness: Considering impacts on different groups of stakeholders.

Explainability: Understanding and evaluating system outputs.

Veracity & Robustness: Achieving correct system outputs, even with unexpected or adversarial inputs.

Transparency: Enabling stakeholders to make informed choices about their engagement with an AI system.

Governance: Incorporating best practices into the AI supply chain, including providers and deployers.