Welche KIs gibt es? Expertenüberblick für Marketer & Techies

Category: KI & Automatisierung
geschrieben von Tobias Hager | 21. November 2025

Welche KIs gibt es? Der gnadenlos klare Expertenüberblick für Marketer und Techies

Alle reden über KI, die meisten meinen Buzzword-Bingo, und nur wenige wissen, was hinter den Modellen wirklich steckt. Wenn du dich fragst: Welche KIs gibt es, was taugen sie, wer baut sie, wie nutzt man sie sinnvoll im Marketing und im Tech-Stack – hier ist dein kompromissloser Deep Dive. Keine Hypeslides, keine Vendor-Märchen, sondern Architektur, Anwendungsfälle, TCO, Compliance und die Realität von GPUs, Kontextfenstern und Latenzen. Kurz: die Karte für

das Minenfeld zwischen OpenAI, Open Source und On-Prem.

- Welche KIs gibt es – eine vollständige Taxonomie von LLMs, Vision-, Audio-, Video-, Tabular- und Recommender-Modellen
- Anbieterlandschaft 2025: OpenAI, Anthropic, Google, Meta, Mistral, Cohere, Open-Source-Stacks und Enterprise-Optionen
- Technik unter der Haube: Transformer, Diffusion, Embeddings, Kontextfenster, KV-Cache, MoE, Quantisierung und LoRA
- RAG, Agenten und Tool-Use: Wie KI mit deinen Daten, Tools und Workflows spricht – robust, auditierbar, skalierbar
- Bild-, Audio- und Video-KI im Marketing: von Creatives über Voice bis Produktfotos – und was rechtlich heikel ist
- Deployment und MLOps: vLLM, Triton, H100 vs. CPU, Batching, Observability, Prompt-Sicherheit und Kostenfallen
- Evaluierung statt Bauchgefühl: Benchmarks, MT-Bench, TruthfulQA, RAGAs, WER, FID, Human-in-the-Loop
- Compliance & Governance: DSGVO, Datensatz-Herkunft, Prompt-Leaks, Copyright, Policy-Enforcement
- Praxis-Blueprints für Marketer: Content-Pipelines, Social-Automation, Personalisierung, Lead-Qualifizierung
- Checkliste: So triffst du die Modellwahl – Use Case, Daten, Latenz, Budget, Risiko, Roadmap

Die Frage “Welche KIs gibt es” klingt banal, ist aber die wichtigste Weichenstellung, bevor du eine einzige Zeile Prompt in einen Chat klebst. “Welche KIs gibt es” entscheidet über Architektur, Kosten und rechtliche Risiken, noch bevor du Layout, Kampagne oder Use Case definierst. “Welche KIs gibt es” ist auch eine Frage der Reifegrade: reine Spielerei, produktionsreif oder auditierbar im Enterprise-Kontext. “Welche KIs gibt es” bedeutet außerdem: Welche Inferenzpfade, welche Kontextfenster, welche Tokenraten und welche Guardrails sind realistisch. Und ja, “Welche KIs gibt es” ist auch kaufmännisch: OPEX vs. CAPEX, GPU-Slots, Rate Limits und SLAs.

Für Marketer und Techies liegt der Sweet Spot zwischen Modellfähigkeit, Datenzugriff und Time-to-Value. Große Sprachmodelle liefern Text, aber ohne RAG bleiben sie ahnungslos zu deinen Produkten, Richtlinien und Preisen. Vision-Modelle generieren Bilder, doch ohne Style- und IP-Guidelines ertränkst du dich in belanglosem Stock-Look. Speech-Modelle machen Audio, aber ohne saubere Mikrofon- und Prompt-Pipelines bekommst du Artefakte und Hall statt Brand-Voice. Die Landschaft ist riesig, aber systematisch beherrschbar – wenn du die Klassen, Anbieter und Metriken sauber sortierst.

Dieser Überblick ordnet die Modelltypen, nennt die relevanten Player, erklärt die Technik und zeigt Produktionspfade, die in echten Unternehmen funktionieren. Du bekommst klare Antworten zu “Welche KIs gibt es” für Text, Bild, Audio, Video, strukturierte Daten und Empfehlungen. Du lernst, wie du RAG, Agenten und Tool-Use robust baust, ohne Security und Compliance zu schrotten. Und du siehst, wie du MLOps und Kosten so planst, dass dein CFO nicht kollabiert, wenn die Prompts viral gehen.

Welche KIs gibt es? KI-Arten, Taxonomie und die richtige Einordnung für Marketing und IT

Wer “Welche KIs gibt es” ernsthaft beantworten will, startet nicht bei Markenlogos, sondern bei Klassen und Lernparadigmen. Es gibt generative Modelle für Text (LLMs), Bild und Video (Diffusion und Transformer), sowie Audio für ASR und TTS. Dazu kommen diskriminative Modelle für Vision-Aufgaben wie Erkennung, Segmentierung und Klassifikation, die im kreativen Prozess oft unterschätzt werden. Hinzu treten Recommender-Systeme, die personalisierte Ausspielungen steuern und im Marketing die unsichtbaren Umsatzmotoren sind. Ein eigener Block sind tabellarische und Zeitreihen-Modelle für Forecasting, Propensity und Pricing, die anders ticken als LLMs. Schließlich existieren Agenten, die als Orchestrierungsschicht Modelle, Tools und Datenbanken verbinden, also die Automatisierungsebene über allem bilden.

LLMs basieren auf Transformer-Architekturen, Tokenisierung und kontextbasiertem Autoregressions-Lernen. Sie liefern Text, Code, Analysen und zunehmend Struktur-Ausgaben via JSON-Mode oder constrained decoding. Bild- und Video-KI arbeiten häufig mit Diffusion, die aus Rauschen iterativ ein Bild oder Clip rekonstruiert, gesteuert durch Text- oder Bild-Prompts. Audio-Modelle teilen sich in Automatic Speech Recognition (ASR) für Transkription und Text-to-Speech (TTS) für synthetische Stimmen mit Emotion und Prosodie. Recommender nutzen Embeddings, Faktorisation und sequentielle Modelle, um Relevanz und Timing zu optimieren, was maßgeblich Conversion und LTV treibt. Tabular-ML setzt auf Gradient Boosting, lineare Modelle oder spezielle Transformer, und gewinnt oft mit purer Feature-Engineering-Disziplin gegen fancy GenAI.

Für die Praxis zählt die Fähigkeit, diese Klassen zu kombinieren, statt sie gegeneinander auszuspielen. Ein Content-Workflow kann ASR für Interviews, LLM für Zusammenfassung, Diffusion für Visuals und TTS für Voiceover verbinden. Ein Commerce-Case kann mit RAG Produktwissen an ein LLM hängen, mit Vision Modelle Produktvarianten erkennen, und mit Recommender die Reihenfolge personalisieren. Wichtig ist, dass jede Klasse eigene Metriken, Latenzprofile und Kostenkurven hat, die du in deinem Architekturdiagramm sichtbar machst. Erst dann siehst du, wo du GPU-Zeit verbrennst und wo du mit Quantisierung und Batching dreistellige Prozentzahlen sparst. Die Frage “Welche KIs gibt es” wird so zum Architekturplan statt zur Hype-Übersicht. Und genau so gehört es sich im Jahr der knappen Budgets und hohen Erwartungen.

Große Sprachmodelle (LLMs) und Multimodale KI: Anbieter, Open Source und technische Stellschrauben

Bei LLMs dominieren API-Anbieter wie OpenAI, Anthropic, Google und Cohere sowie Open-Source-Modelle wie Llama 3, Mistral/Mixtral, Phi, Qwen, Gemma und Yi. Multimodale Varianten wie GPT-4o, Claude 3, Gemini oder Llama 3.2 Vision verarbeiten Text, Bild, teils Audio und Video in einem Modell. Der technische Unterschied spielt für Deployment und Kosten eine enorme Rolle, denn Kontextfenster, Tokenrate, Speicherkonsum und MoE-Routing beeinflussen Throughput und Latenz. Ein 128k-Kontextfenster klingt sexy, frisst aber RAM und kann ohne Retrieval zu teuren Halluzinationen führen. Speculative Decoding, KV-Cache und Batching mit Servern wie vLLM oder TGI heben Tokens-per-Second signifikant, wenn du den Traffic bündeln kannst. Für Enterprise zählen ferner Funktionen wie JSON-guaranteed Output, Tool Calling, System-Prompts, Moderation und Reproducibility über Seed-Kontrolle.

Open Source ist kein Billigersatz, sondern ein Architekturwerkzeug mit Kontrolle über Daten, Kosten und Latenz. Mit quantisierten Varianten (int8, int4, GGUF) laufen starke Modelle auf Edge-GPUs oder sogar CPU-optimiert, wenn Latenz nicht kritisch ist. Fine-Tuning per LoRA/QLoRA erlaubt schnelle Domänenanpassung, während Instruct- oder DPO-Verfahren die Steuerbarkeit erhöhen. Distillation komprimiert große Lehrermodelle in leichtere Schülermodelle, die günstiger inferieren. Für Compliance ist On-Prem mit abgeschottetem Netz und Audit-Log oft der Gamechanger, besonders wenn Kundendaten, geistiges Eigentum oder regulatorische Vorgaben im Spiel sind. Trotzdem brauchst du Observability: Tokenkosten, Fehlerraten, Latenz-P50/P95, Tool-Failures und Sicherheitsevents gehören ins Monitoring, nicht ins Bauchgefühl.

Entscheidend ist die Passung: Für Ad-Copy reichen kleinere 7B–13B-Modelle, für Analyse und komplexe Tool-Chains nimmst du stärkere Modelle mit robustem Function Calling. Multimodalität lohnt, wenn du Screenshots, Produktbilder, Layouts oder Tabellen verstehen musst. Für hohe Volumina mit planbarem Traffic baust du eigene Inferenz mit GPU-Batching, für volatile Lasten nimmst du API-Provider mit Rate-Limits und Rückfallpfad. Und du testest nicht an drei Prompts, sondern mit Evals: MT-Bench für Dialog, TruthfulQA für Halluzinationen, G-EVAL oder BERTScore für Summarization und Human Review für brandkritische Texte. Nur dann weißt du, welches Modell im Alltag wirklich performt, statt nur im Launch-Blogpost zu glänzen.

Bild-KI, Video-KI und Audio-KI: Diffusion, Generative Pipelines und die realen Marketing-Use-Cases

Bild-KI wird von Diffusionsmodellen wie Stable Diffusion, SDXL oder proprietären Engines wie Midjourney und DALL·E geprägt. Technisch generierst du aus Rauschen ein Bild, gelenkt durch Text-Conditioning, Bild-Conditioning oder ControlNets, die Komposition und Stil bändigen. Für Markenarbeit brauchst du Style-LoRAs, Referenzbilder und Prompts, die nicht zufällig wirken, sondern konsistente CI liefern. Compositing mit Segmentierungsmodellen (SAM, Segment Anything) und Inpainting/Outpainting macht Produktbilder skalierbar. Für Qualitätsbewertung helfen CLIP-Score, Perceptual Metrics und – unersetzlich – menschliche Abnahme mit klaren Kriterienkatalogen. Kosten senkst du mit Batch-Generierung, Caching und der Trennung von Ideation und Finalisierung, statt jeden Entwurf in 4k zu rechnen.

Video-KI ist kommend, aber reif genug für Teaser, Loops und Produktclips aus Text oder Storyboards. Modelle wie Runway, Pika oder Sora zeigen, wohin die Reise geht, aber Produktionsreife erfordert Post-Processing, Frame-Interpolation und Sounddesign. Für Social reichen oft 5–10 Sekunden mit hohem Hook-Faktor, was die Renderkosten innerhalb eines Budgets hält. Ein praktikabler Weg ist die Pipeline: Storyboard per LLM, Frames via Diffusion, Upscaling mit ESRGAN, Schnitt in DaVinci, Voiceover via TTS und Soundeffekte aus Libraries. Rechteklärung bleibt Pflicht, denn stilistische Nähe zu geschützten Werken kann juristisch knifflig werden. Ausspielung optimierst du per A/B-Tests und Platform-Hooks, nicht per Bauchgefühl in der Kreation.

Audio teilt sich in ASR, TTS und Voice-Cloning. Whisper, NeMo oder Vosk transkribieren robust, während ElevenLabs oder Azure TTS erstaunlich natürliche Stimmen liefern. Für Brand-Voice brauchst du saubere Referenzaufnahmen, Noise-Reduction und klare Sprechanweisungen in Prompts. Latenz entscheidet, ob du Voice-Chat in Echtzeit schaffst oder besser asynchron bleibst. Metriken wie WER (Word Error Rate) für ASR und subjektive MOS-Scores für TTS sind Standard, ergänzt durch inhaltliche QA. Datenschutz ist heikel: Sprachdaten gelten oft als personenbezogen, also verschlüsseln, lokal verarbeiten oder vertraglich absichern – sonst brennt dir Legal die Pipeline ab.

RAG, Agenten und Tool-Use: Wie

KI wirklich mit deinen Daten und Systemen arbeitet

Ohne Retrieval-Augmented Generation bleibt jedes LLM blind für deine Produkte, Prozesse und Policies. RAG koppelt Embeddings, Vektordatenbanken und Prompting zu einer Kette, die Kontext präzise in die Inferenz einspeist. Du chunkst Dokumente, berechnest Embeddings, speicherst sie in Weaviate, Qdrant, Milvus, Pinecone oder Postgres mit pgvector und ziehst beim Prompt die Top-k-Snippets. Qualität hängt an Chunking-Strategien, Rekordaktualität und Retrieval-Metriken wie Recall@k oder nDCG. RAGAs und LLM-as-a-Judge helfen bei der Bewertung, aber Ground-Truth-basierte Tests bleiben Königsklasse. Guardrails wie Zitatpflicht, Quellenverweise und verbotene Antworten reduzieren Risiken und erhöhen Vertrauen. Caching auf Query- und Tokenebene spart massiv Kosten, wenn du viel wiederkehrende Fragen hast.

Agenten sind die nächste Ebene: Das Modell plant, ruft Tools auf, schreibt in Systeme und iteriert, bis ein Ziel erreicht ist. Tool-Use via Function Calling bindet Datenbanken, CRM, E-Mail, Analytics, Payment oder Ad-APIs ein. Frameworks wie LangChain, LlamaIndex oder Guidance orchestrieren Schritte, doch Stabilität kommt erst mit expliziten Zustandsmaschinen. Multi-Agent-Ansätze verteilen Rollen wie Recherche, Entwurf, QA und Freigabe auf spezialisierte Policies, was Skalierbarkeit und Nachvollziehbarkeit verbessert. Sicherheit heißt hier: strikte Tool-Schemas, Sandboxen, Rechtemanagement und Output-Validierung per JSON-Schema. Ohne diese Sicherungen verwandelt sich ein Agent schnell in einen übermotivierten Praktikanten mit Adminrechten.

- Use Case definieren: Frage, Zielvariable, Qualitätskriterien, Latenzbudget, SLA.
- Daten anbinden: Quellen inventarisieren, ETL/ELT mit Airflow oder dbt, Berechtigungen klären.
- Retrieval designen: Chunking, Embedding-Modell wählen, Index bauen, Quality-Evals etablieren.
- Prompting strukturieren: Systemprompt, Kontext, Zitationspflicht, JSON-Output, Fehlermeldungen.
- Tool-Use härten: Funktionsschemas, Timeouts, Rate Limits, Idempotenz, Retry-Logik.
- Observability: Logs, Traces, Kosten pro Anfrage, P95-Latenz, Alarmierung, Feedback-Loops.
- Governance: Policy-Filter, PII-Redaction, Audit-Logs, Red-Teaming und Freigabeprozesse.

MLOps, Deployment und Kosten:

Von GPU-Realität bis vLLM – was in der Praxis zählt

Deployment entscheidet, ob dein KI-Projekt eine Demo bleibt oder Umsatz treibt. Für LLM-Inferenz dominieren vLLM, TGI und Triton mit effizientem KV-Cache, Continuous Batching und tensoroptimierten Kernels. H100 und A100 liefern die beste Performance, aber mit cleverer Quantisierung und MoE kannst du kleinere Instanzen wirtschaftlich fahren. Latenz ist ein Produkt aus Tokenrate, Kontextgröße und Netzwerkwegen; Edge-Serving oder regionale Replikation senkt P95 spürbar. Autoscaling braucht Warm-Pools, sonst zahlst du Kaltstart-Latenzen mit Abbrüchen. Für Bild/Video sind GPU-Zeitfenster planbar, also batchen, Render-Farm planen und Priorisierung einführen. Kostenmonitoring auf Token-, Job- und Team-Ebene verhindert Überraschungen, wenn Kampagnen anziehen.

MLOps ist mehr als ein Modell speichern: Versionierung von Daten, Code, Prompts, LoRA-Weights und Evaluierungen ist Pflicht. MLflow, Weights & Biases oder SageMaker Experiments helfen bei Reproduzierbarkeit und Vergleichbarkeit. CI/CD-Pipelines für Modelle testen nicht nur Unit- und Integration, sondern Guardrails, Output-Schemata und Sicherheitsprüfungen. Canary-Releases und A/B-Serving ermöglichen risikominimierte Updates mit realer Nutzerlast. Incident-Response gehört in den Runbook-Ordner: Was tun bei Halluzinationen, Kostenexplosion, Model-Downgrade oder Data-Drift. Ohne diese Basics wird jede neue Modellversion zum Glücksspiel, und Marketing-KPIs fahren Achterbahn.

Cloud-Optionen vereinfachen Start und Skalierung: Vertex AI, Bedrock und Azure OpenAI liefern Managed-Modelle, private Endpunkte und Abrechnung pro Nutzung. On-Prem oder Private Cloud lohnt, wenn Datenhoheit, Latenz oder Kostenstruktur es erzwingen. Hybride Setups erlauben sensible Pfade intern und elastische Last extern. Ein sauberer Architekturentscheid beginnt bei Use Case und Budget, nicht bei der Lieblings-Cloud. Und bevor du investierst, rechnest du TCO: Modellkosten, Rechenzeit, Storage, Egress, Observability, Wartung, Security, Schulung und Prozesskosten. Nur so erkennst du, ob “günstig pro Token” am Ende teuer pro Conversion ist.

Recht, Sicherheit und Governance: DSGVO, Urheberrecht, Prompt-Leaks und verlässliche Evaluierung

Rechtliche Fragen sind kein Bremser, sondern der Unterschied zwischen Experiment und Produkt. DSGVO verlangt Zweckbindung, Datenminimierung und

Transparenz, besonders bei PII in Prompts, Kontext oder Logs. Data Residency und Verschlüsselung im Transit und at Rest sind Standard, nicht Kür. Bei Trainingsdaten und Stilreferenzen lauert Urheberrecht, also prüfe Lizenzlage und vermeide unverwechselbare Stilkopien ohne Freigabe. Markenrichtlinien gehören in Systemprompts, vertragliche SLAs in Anbieterverträge und Audit-Logs in dein Archiv. Moderationslayer sind Pflicht, wenn Nutzer frei prompten, inklusive Blocklisten, Safe Completion und Escalation-Pfaden. Ohne diese Schutzgeländer bricht dir das Modell bei der ersten heiklen Anfrage die Marke auseinander.

Sicherheit im Betrieb beginnt bei Secret-Management, Least Privilege und isolierten Umgebungen. Prompt-Leaks verhinderst du mit Redaction, Stripping sensibler Felder und internen Context-Filtern. Output-Validierung via JSON-Schema, Regex-Guards und Policy-Engines verhindert API-Missbrauch durch “überredete” Modelle. RAG-Quellen benötigen Metadaten für Herkunft, Aktualität und Rechte, damit Antworten nachvollziehbar bleiben. Red-Teaming deckt Jailbreaks, Injection und Tool-Missbrauch auf, bevor es Kunden treffen kann. Und wenn doch etwas passiert, brauchst du Forensik: vollständige Logs, Prompt- und Kontextversionen sowie reproduzierbare Seeds, um Fehler schnell zu verstehen.

Evaluierung ist kein Einmal-Event, sondern laufende Qualitätssicherung. Für Text nutzt du MT-Bench, TruthfulQA, MMLU, aber ergänzt immer domänenspezifische Tests. Für RAG misst du Retrieval-Recall@k, Answer-Faithfulness und Korrektheitsraten mit menschlichem Review. Für ASR zählt WER, für TTS MOS, für Bilder Ähnlichkeits- und Perzeptionsmetriken, für Video Konsistenz und Artefaktrate. Human-in-the-Loop bleibt unverzichtbar, sobald rechtliche oder markenrelevante Risiken bestehen. Reporting gehört in Dashboards, die Stakeholder verstehen, mit Trends und Alerting auf negative Drifts. So wird Qualität kein Bauchgefühl, sondern ein Vertrag mit Zahlen.

Für Marketer ergeben sich klare Muster, wie KI im Alltag spürbaren Mehrwert liefert. Content-Pipelines starten mit Recherche-Agenten, gehen über Drafting, Fact-Checking via RAG und enden in Kanal-spezifischen Varianten. Creatives werden mit Bild-KI vorvisualisiert, dann von Designern finalisiert, statt komplett synthetisch “aus der Dose” zu kommen. Social-Automation verknüpft Scheduling, Moderation und Tonfall-Richtlinien per Tool-Use und Guardrails. Personalisierung nutzt Recommender und LLMs für Texte, aber Datenhoheit und Frequency Capping bleiben heilig. Sales Enablement bündelt Produktwissen und Einwände in gutem RAG, statt irrelevante Links zu verschicken. So wird KI zum Verstärker, nicht zum Risiko.

Technisch ist die Richtung eindeutig, wenn du Skalierung und Zuverlässigkeit willst. Einheitliche Prompt- und Kontext-Standards verhindern Fragmentierung, Versionierung hält alles reproduzierbar. Feature-Stores, Vektorindizes und Event-Streams bilden die Datenbasis, ohne die jede Antwort improvisiert wirkt. API-Governance stellt sicher, dass Tools konsistent funktionieren und Änderungen kontrolliert ausgerollt werden. Kostenkontrolle kommt mit Limits, Pre-Flight-Schätzungen und dynamischem Routing auf kleinere Modelle. Am Ende zählt nicht, wer das größte Modell ruft, sondern wer die stabilste, nachvollziehbarste Kette baut.

Die Kernantwort auf “Welche KIs gibt es” ist damit weniger die Liste und mehr die Landkarte, auf der du dich bewegst. LLMs für Sprache und Struktur, Diffusion für Bilder und Clips, ASR/TTS für Stimme, Recommender für Relevanz, Tabular-ML für harte KPIs und Agenten als Orchestrierung. Dazu RAG als Brücke zu deinen Daten, MLOps als Betriebssystem und Governance als Geländer entlang jeder Kante. Setzt du diese Bausteine bewusst zusammen, bekommst du Systeme, die schnell liefern, sauber skalieren und rechtlich standhalten. Ignorierst du sie, landest du im Demo-Limbo, wo Budget verbrennt und Vertrauen verschwindet. Willkommen im echten Spiel – es ist anspruchsvoll, aber fair zu denen, die es ernst nehmen.

Wenn du jetzt die Frage “Welche KIs gibt es” in einem Satz beantworten musst, sag: Es gibt die, die du bedienen kannst, und die, die dich bedienen – du willst die erste Kategorie. Baue klein an, messe hart, automatisiere vorsichtig und standardisiere alles, was wiederkehrt. Dann wird KI vom Buzzword zur Maschine hinter deinen KPIs.

Fassen wir zusammen: Wähle Modellklassen nach Use Case, nicht nach Hype. Kopple LLMs mit RAG, baue Agenten mit strengen Tools, beobachte alles, sichere alles, evaluiere alles. Respektiere rechtliche Leitplanken, dokumentiere Entscheidungen und halte Kosten unter Kuratierung. So liefert KI Wert, statt Kosten und Risiko zu multiplizieren.