

GREEN AGENTIC AI

Mit Modellwahl, Prompting und Architektur
Rechenleistung dort einsetzen, wo sie Mehrwert schafft

Oktober 2026

MIT MODELLWAHL, PROMPTING UND ARCHITEKTUR RECHENLEISTUNG DORT EINSETZEN, WO SIE MEHRWERT SCHAFFT

Nachdem unser erstes Knowledge Nugget den Weg vom Chatbot über den KI-Agenten bis zum Multi-Agenten-System eingeordnet hat [0] und unser zweites Knowledge Nugget die Wahl einer passenden Agentenarchitektur vertieft hat [1], stellt sich für die nachhaltige Gestaltung agentischer Systeme eine weitere Frage: **Wie lässt sich generative und agentische KI so gestalten, dass die benötigte Qualität ohne unnötigen Ressourcenaufwand entsteht?** Die Grundbegriffe wiederholen wir nur knapp: Ein Agent verfolgt ein Ziel, plant, nutzt Tools und handelt; in einem Multi-Agenten-System arbeiten mehrere spezialisierte Agenten koordiniert zusammen.

Generative KI und Multi-Agenten-Systeme erhöhen den Bedarf an Rechenleistung und treiben damit auch den Ausbau energieintensiver Rechenzentren. Deren Stromverbrauch wird heute nicht ausschließlich aus erneuerbaren Quellen gedeckt. Da erneuerbare Erzeugung nicht jederzeit in gleichem Umfang verfügbar ist, bleibt der tatsächliche Strommix zeit- und ortsabhängig; Erdgas und Kohle sind insbesondere kurzfristig weiterhin wichtige Quellen [15]. Für eine klimaneutrale Zukunft müssen Treibhausgasemissionen weitgehend vermieden und verbleibende Restemissionen ausgeglichen werden. Deshalb ist es entscheidend, den Ressourcenbedarf bereits beim Design von KI-Anwendungen und -Architekturen mitzudenken, statt erst im Betrieb nach Einsparmöglichkeiten zu suchen.

Gerade bei cloudbasierten KI-Diensten sind der tatsächliche Energieverbrauch und die daraus entstehenden Kohlendioxid-Äquivalente (CO₂e) – eine gemeinsame Maßeinheit für die Klimawirkung verschiedener Treibhausgase – in der Entwicklung häufig nicht direkt bekannt. Als praktikabler Proxy kann der **Tokenverbrauch** dienen. Tokens sind vereinfacht gesagt Textbausteine, in die Ein- und Ausgaben für die Verarbeitung durch ein Sprachmodell zerlegt werden. Der Tokenverbrauch bildet die Menge der verarbeiteten Eingabe und erzeugten Ausgabe ab, lässt sich pro Workflow erfassen und ist zugleich unmittelbar kostenrelevant. Tokenzahlen sind kein modellübergreifend vergleichbares Energiemaß, machen aber überflüssigen Verarbeitungsaufwand sichtbar und helfen, ökologische und wirtschaftliche Effizienz gemeinsam im Blick zu behalten [4, 11, 12].

Als Arbeitsbegriff verwenden wir **Green Agentic AI** für die Übertragung des im folgenden Abschnitt erläuterten Green-AI-Prinzips auf das gesamte agentische System. Nicht nur das Basismodell zählt, sondern die vollständige Verarbeitungskette aus Kontext, Modell- und Tool-Aufrufen, Reasoning-Schleifen, Übergaben, Wiederholungen und Prüfungen. Jeder zusätzliche Schritt sollte einen nachvollziehbaren Beitrag zur Zielerreichung leisten. Das Ziel ist nicht „möglichst wenig KI“, sondern **angemessene Rechenleistung für die tatsächlich benötigte Qualität und Zuverlässigkeit**.

Mehr Agenten bedeuten meist zusätzliche Modellaufrufe und Koordination, führen aber nicht zwangsläufig zu einem höheren Gesamtaufwand: Spezialisierung, kompakte Kontexte und bedarfsgerecht gewählte Modelle können den Mehraufwand ausgleichen. Ob das im Einzelfall gelingt, hängt von der konkreten Architektur ab und muss mit realen Testfällen und Kennzahlen über den gesamten Workflow geprüft werden [1, 19].

GREEN AI: QUALITÄT UND AUFWAND GEMEINSAM DENKEN

Green AI ergänzt die Frage „Wie gut ist ein KI-System?“ um die Frage „Mit welchem Aufwand wird dieses Ergebnis erreicht?“ Entsprechend sollte Effizienz neben Genauigkeit und anderen Qualitätsmaßen **als eigenes Bewertungskriterium** behandelt und der rechnerische beziehungsweise finanzielle „Preis“ sichtbar gemacht werden [2]. Ergänzend sollten auch der Entwicklungsaufwand sowie der Einsatz effizienter Hardware und Algorithmen systematisch berücksichtigt werden [14].

Ein System ist nicht allein deshalb nachhaltig, weil es wenige Tokens verbraucht. Es muss fachliche Mindestqualität erreichen, zuverlässig funktionieren und Prozessnutzen stiften. Umgekehrt ist die höchstmögliche Antwortqualität nicht automatisch die beste Lösung, wenn ihr zusätzlicher Nutzen gering, der Aufwand aber sehr hoch ist. Deshalb sollten Qualität, Workflow-Erfolg, Laufzeit und Ressourcenaufwand gemeinsam betrachtet werden – idealerweise pro erfolgreich erledigtem Geschäftsvorgang. Fehlgeschlagene Läufe und erneute Versuche gehören zur Bilanz.

WO DER RESSOURCENAUFWAND ENTSTEHT

Das Training großer Basismodelle – einschließlich der Entwicklungs- und Experimentierläufe – verursacht einen hohen, überwiegend in der Entwicklungs- und Bereitstellungsphase anfallenden Ressourcenaufwand. Die Inferenz – also die Nutzung eines bereits trainierten Modells für konkrete Anfragen – fällt dagegen **bei jeder Nutzung erneut an**. Bei stark verbreiteten Modellen kann der kumulierte Energiebedarf der Inferenz über die Lebensdauer den Trainingsaufwand erreichen oder übersteigen. Wann dieser Punkt eintritt, lässt sich nicht allgemeingültig bestimmen: Das Verhältnis hängt von Modellarchitektur, Hardware und Rechenzentrum ebenso ab wie von Auslastung sowie Anzahl, Länge und Komplexität der Anfragen. Für aktuelle proprietäre Modelle fehlen zudem meist vergleichbare Angaben zu Training und Inferenz, die mit derselben Systemgrenze und Messmethode erhoben wurden [12, 13, 14].

Ein vergleichsweise umfassendes aktuelles Beispiel für Umwelttransparenz liefert Mistral AI. Das Unternehmen veröffentlichte für Mistral Large 2 eine extern geprüfte Lebenszyklusanalyse, die sowohl absolute Umweltwirkungen des Trainings als auch die zusätzliche Umweltwirkung einer beispielhaften einzelnen Anfrage ausweist. Als dritte zentrale Kennzahl nennt Mistral den Anteil der gesamten Inferenz an den gesamten Lebenszykluswirkungen, veröffentlicht dafür jedoch noch keinen allgemeinen Verhältniswert [18]. Das unterstreicht: Training gehört zur Gesamtbilanz und sollte bei einer Lebenszyklusbetrachtung berücksichtigt werden. Für Unternehmen, die eine agentische Anwendung auf einem bestehenden Basismodell entwerfen, liegt der unmittelbar beeinflussbare Hebel jedoch **vor allem in der Inferenz**. Hier entscheiden Modellwahl, Kontextmenge, Antwortlänge, Zahl der Modellaufrufe, Tools, Schleifen sowie Single- oder Multi-Agenten-Architektur über den laufenden Ressourcenbedarf [4, 10].

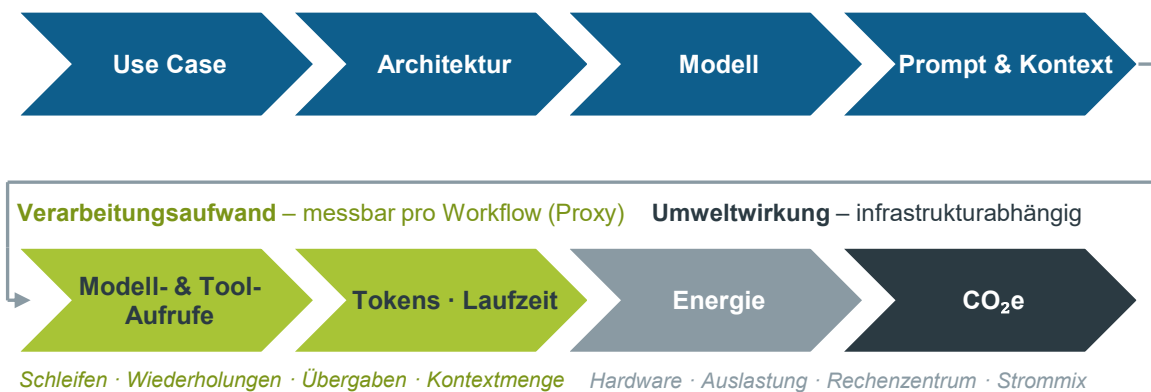
Generative Modelle erzeugen ihre Ausgabe schrittweise: Längere Antworten verlängern die Verarbeitung und erhöhen in der Regel den Energiebedarf. Poddar et al. zeigen über verschiedene Modelle und Aufgaben hinweg eine starke Korrelation der Inferenzenergie mit Antwortzeit und Outputlänge; die Outputlänge war in ihren Messungen ein stärkerer Einflussfaktor als die Inputlänge [4].

WO DER RESSOURCENAUFWAND ENTSTEHT

Der Aufwand hängt jedoch nicht nur von Tokens ab. Relevant sind Modellfamilie und Modellgröße, Hardwaregeneration, Auslastung und technische Betriebsoptimierungen wie Batch-Verarbeitung oder Quantisierung sowie die konkrete Aufgabe [4, 10]. In agentischen Systemen kommen Modell- und Tool-Aufrufe, mitgeführter Kontext, Übergaben, Prüf- und Korrekturschleifen sowie Wiederholungen nach Fehlern hinzu. In einer Studie [3] wurden generative Textaufgaben – etwa das Erzeugen oder Zusammenfassen frei formulierter Texte – mit Textklassifikation verglichen. Bei der Textklassifikation wird ein Text lediglich einer vorgegebenen Kategorie zugeordnet, beispielsweise „positiv“ oder „negativ“, während ein generatives Modell eine neue Ausgabe Token für Token erzeugt. Im betrachteten Versuchsaufbau benötigten die generativen Textaufgaben im Mittel mehr als zehnmal so viel Energie pro 1.000 Inferenzvorgänge wie die Textklassifikation [3]. Der Faktor ist nicht universell, zeigt aber: **Für stabile, eng umrissene Aufgaben sollte nicht automatisch ein generalistisches generatives Modell eingesetzt werden.**

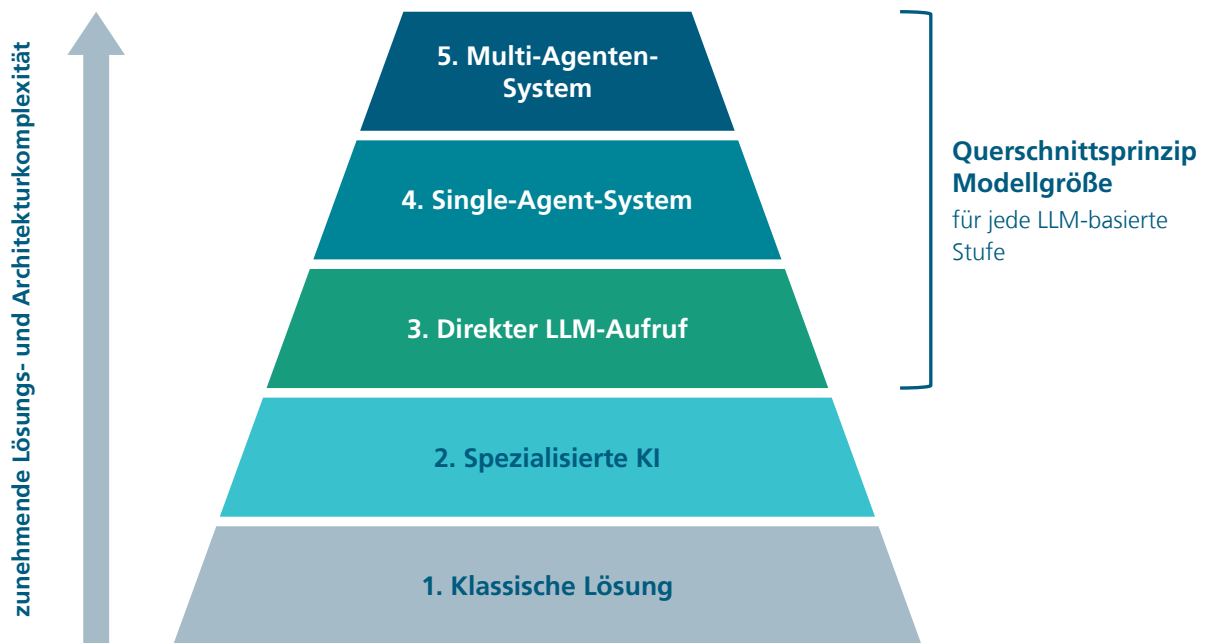
Tokens, Outputlänge, Laufzeit, Modellaufrufe, Tool-Schritte, Wiederholungsrate und Kosten sind nützliche Proxy-Metriken, **ersetzen aber keine CO₂e-Messung**. Hardware, Energieeffizienz des Rechenzentrums und Strommix verändern die Umweltwirkung erheblich [11, 12]. Auch die Systemgrenze zählt: Für das finale Training von BLOOM – einem offen entwickelten großen Sprachmodell mit 176 Milliarden Parametern – wurden rund 24,7 Tonnen CO₂e aus dynamischer Stromnutzung ausgewiesen. Unter Einbeziehung zugerechneter Herstellungs- und Leerlaufemissionen stieg die Trainingsbilanz auf rund 50,5 Tonnen CO₂e; die spätere Inferenz ist darin nicht enthalten [13]. Deshalb müssen Annahmen und Grenzen einer Bilanz transparent sein.

Gestaltungshebel – beim Design beeinflussbar



SO VIEL KI WIE NÖTIG: EINE ESKALATIONSLOGIK

Eine wichtige Green-AI-Entscheidung fällt vor dem ersten Prompt: Reicht für den Use Case eine klassische oder spezialisierte Lösung? Falls nicht, prüfen wir einen direkten LLM-Aufruf. Ein einzelner Agent kommt für Toolnutzung, Planung oder Verifikation infrage; mehrere Agenten erst dann, wenn Spezialisierung, getrennte Kontexte oder Arbeitsteilung den zusätzlichen Koordinationsaufwand rechtfertigen. **Jede zusätzliche Stufe muss ihren Nutzen für Qualität, Zuverlässigkeit oder Effizienz des gesamten Workflows zeigen.**



Die Stufen beschreiben zunehmende Lösungs- und Architekturkomplexität, nicht automatisch einen zunehmenden Ressourcenverbrauch. Durch Spezialisierung, kleinere Rollenmodelle und fokussierte Kontexte kann eine spätere Stufe im konkreten Fall sogar effizienter sein als eine frühere, monolithische Lösung.

Querschnittsprinzip Modellgröße (Stufen 3-5). Für jede LLM-basierte Stufe gilt: Eingesetzt werden sollte das kleinste Modell, das die definierte Qualitäts- und Erfolgsgrenze zuverlässig erreicht. An ein leistungsfähigeres Modell wird nur eskaliert, wenn die zusätzliche Modellkapazität im konkreten Fall benötigt wird.

Die Reihenfolge ist keine starre Vorschrift. Eine einfache Ausgangslösung dient als prüfbare Baseline und ist häufig leichter zu testen, zu überwachen und zu warten.

Die folgenden Abschnitte vertiefen die drei zentralen Hebel dieser Logik: Prompt und Kontext, Modellwahl und Routing sowie die Architektur des gesamten agentischen Systems.

GREEN PROMPTING: PRÄZISION STATT KÜRZE UM JEDEN PREIS

Der erste Hebel liegt in der Gestaltung von Prompt und Kontext. **Green Prompting bedeutet dabei nicht, jeden Prompt möglichst kurz zu machen.** Ein knapper oder mehrdeutiger Prompt kann lange Antworten, Rückfragen und weitere Iterationen auslösen. Entscheidend ist der Gesamtaufwand bis zum brauchbaren Ergebnis. Gute Prompts formulieren das Ziel, liefern relevanten Kontext, erklären fach- oder organisationsspezifische Begriffe, legen Zielgruppe, Ausgabeformat und Umfang fest und benennen bei Bedarf Akzeptanzkriterien. Rollen und Beispiele sind hilfreich, wenn sie das Verhalten tatsächlich präzisieren; andernfalls erhöhen sie nur den Input.



Ungünstig:

„Erkläre mir ausführlich alles über Cloud Computing.“

- Ziel und Zielgruppe bleiben offen
- Weder Format noch Umfang vorgegeben
- Lange Antworten, Rückfragen und weitere Iterationen



Zielgerichteter:

„Erkläre Cloud Computing für Nicht-Techniker in maximal fünf Stichpunkten. Behandle Definition, Funktionsweise, zwei Vorteile und ein Unternehmensbeispiel.“

- Zielgruppe und Inhalt klar benannt
- Format und Umfang begrenzt
- Weniger Interpretationsspielraum und weniger Iterationen

Der zweite Prompt ist nicht zwingend kürzer, begrenzt aber Interpretationsspielraum und Output. Podder et al. berichten in ihrer Studie für die untersuchten Modelle und Szenarien über Reduktionen von **32 bis 48 Prozent** beim geschätzten Energieverbrauch und bei den daraus abgeleiteten betrieblichen Emissionen durch sechs Green-Prompting-Praktiken [5]. Die Werte wurden mit EcoLogits, einem modellbasierten Verfahren zur Abschätzung von Umweltwirkungen, ermittelt und beruhen nicht auf direkten Strommessungen der verwendeten Cloudmodelle. Das ist keine allgemeine Garantie: Die Wirkung war szenarioabhängig, und ungünstige Einschränkungen können Qualität verschlechtern oder neue Iterationen auslösen. Eine kleinere, auf Code-Aufgaben beschränkte Studie liefert ergänzende Hinweise auf Effizienzgewinne durch strukturierte Prompt-Bestandteile [6].

Bei agentischer KI sind insbesondere die **System-Prompts der Agenten** ein großer Hebel. Ein System-Prompt ist die dauerhaft mitgeführte Arbeitsanweisung, die Rolle, Ziel, Verhalten und Grenzen eines Agenten festlegt. Solche Anweisungen werden innerhalb eines Workflows häufig bei vielen Modellaufrufen erneut mitgeführt und prägen damit sowohl die Ergebnisqualität als auch den kumulierten Input- und Outputumfang. Deshalb sollten sie so präzise wie nötig, aber so kompakt wie möglich formuliert, mit repräsentativen Testfällen geprüft und auf das gewünschte Ergebnisformat sowie angemessene Antwortlängen ausgerichtet werden. Jede vermiedene Fehlleitung, unnötig lange Ausgabe oder zusätzliche Iteration spart Tokens und Kosten und kann – abhängig von Modell und Infrastruktur – operative Emissionen reduzieren [4, 5, 6].

Ebenso wichtig ist bewusstes Kontextmanagement. Mitgeführt werden sollten nur Informationen, die für die aktuelle Aufgabe tatsächlich benötigt werden. Irrelevanter Kontext erhöht die Zahl der Input-Tokens und damit meist die Kosten; zudem kann er Modelle ablenken und die Ergebnisqualität verschlechtern [4, 16]. Für Agentensysteme bedeutet das: Speicher, Chatverläufe, aus Wissensquellen abgerufene Inhalte und Übergaben sollten gefiltert, verdichtet oder gezielt neu aufgebaut werden. Auch im einfachen Chat gilt: Wer nach einem längeren Gespräch zu einem anderen Thema wechselt, startet besser einen neuen Chat, statt den gesamten alten Kontext weiterzutragen.

DAS RICHTIGE MODELL – UND ROUTING STATT PAUSCHALWAHL

Nach Prompt und Kontext ist die Modellwahl der zweite zentrale Hebel. Selbst ein gut formulierter Prompt bleibt unnötig ressourcenintensiv, wenn jede Anfrage pauschal an das größte verfügbare Modell geht. Die Botschaft lautet deshalb nicht „kleine Modelle sind immer nachhaltiger“, sondern: **Nutze das kleinste Modell, das die erforderliche Qualität zuverlässig erreicht**. Ein kleineres Modell kann mehr Tokens erzeugen, zusätzliche Schleifen benötigen oder öfter scheitern; dann verschwindet sein nomineller Effizienzvorteil auf Systemebene. Umgekehrt ist ein großes Modell für Routineanfragen oft überdimensioniert.

Aufgabentyp	Geeigneter Ansatz
Stabile, eng umrissene Aufgaben	Klassische oder spezialisierte Lösung wie bspw. Machine Learning
Klar begrenzte Aufgaben wie Extraktion, Klassifikation oder Formatprüfung	Kleineres Modell
Anspruchsvolle Aufgaben, etwa zentrale Orchestrierung	Leistungsfähigeres Modell
Aufgaben mit unterschiedlich schwierigen Anfragen	Dynamisches Routing: Standardpfad für einfache Fälle, Eskalation nur bei Bedarf

Eine mögliche Lösungsstrategie ist ein **Green-AI-Router**. Darunter verstehen wir eine vorgeschaltete Entscheidungsinstanz, die jede Anfrage anhand ihrer Schwierigkeit, Qualitätsanforderung und – soweit belastbar abschätzbar – ihres erwarteten Ressourcenaufwands an den einfachsten voraussichtlich ausreichenden Verarbeitungspfad leitet. Das kann ein direkter Modellaufruf, ein Single-Agent-System oder ein Multi-Agenten-System sein. Bei Bedarf eskaliert der Router kontrolliert an ein leistungsfähigeres Modell oder eine komplexere Architektur. Die Idee baut auf kosten- und qualitätsbewusstem Model Routing auf [7, 8, 9].

DAS RICHTIGE MODELL – UND ROUTING STATT PAUSCHALWAHL

FrugalGPT zeigte in seinen Benchmarks Kosteneinsparungen von bis zu 98 Prozent bei vergleichbarer Leistung [7]. **Hybrid LLM** erreichte in einer untersuchten Modellkonfiguration bis zu 40 Prozent weniger Aufrufe des großen Modells ohne Rückgang der dort gemessenen Antwortqualität [8]. **RouteLLM** zeigte deutliche Kostenvorteile bei weitgehend erhaltener Leistung in den verwendeten Testaufgaben; deren Umfang hängt vom Benchmark, der gewählten Qualitätsschwelle und der Vergleichsbasis ab [9]. Diese Ergebnisse zeigen einen Kosten- und Qualitätshebel, belegen aber keinen automatischen CO₂e-Effekt: Preise bilden den Energieverbrauch nicht proportional ab, und Informationen über Hardwareeffizienz, Rechenzentrumsauslastung und Strommix der Anbieter sind häufig nicht vollständig verfügbar. Ein Green-AI-Router sollte deshalb zusätzlich Tokens, Laufzeit, Fehlleitungen und Eskalationen beobachten.

Da ein Router selbst Aufwand verursacht und falsch entscheiden kann, braucht er Qualitätsgrenzen, einen sicheren Rückfallpfad und Tests mit der realen Aufgabenverteilung. **Ein Router wählt den geeigneten Verarbeitungspfad; ein Orchestrator steuert dagegen innerhalb eines komplexeren Pfads die Zusammenarbeit und Reihenfolge der beteiligten Agenten.** Rollenbezogenes Routing ist besonders interessant: Klar begrenzte Aufgaben wie Extraktion, Klassifikation oder Formatprüfung können gegebenenfalls mit kleineren Modellen erledigt werden. Für die zentrale Orchestrierung gilt das nicht pauschal. Muss ein Orchestrator Aufgaben zerlegen, Agenten auswählen, Zwischenergebnisse bewerten und den weiteren Ablauf steuern, kann gerade dort ein leistungsfähigeres Modell Vorteile bieten; eine aktuelle Vorabstudie mit kontrollierten Vergleichen deutet darauf hin, dass die Systemleistung stärker von der Kapazität des Orchestrators als von der Größe der Subagenten abhängen kann [17]. Wo Routing dagegen deterministisch über Regeln oder eine einfache Klassifikation erfolgt, ist kein großes LLM zwingend erforderlich.

GREEN AGENTIC AI BEGINNT BEI DER ARCHITEKTUR

Prompting und Modellwahl allein genügen jedoch nicht: Der dritte Hebel liegt in der Architektur des gesamten Workflows. Tools, Gedächtnis, Planung, Reviewer und zusätzliche Agenten erweitern Fähigkeiten, erhöhen aber meist Kontextmenge, Inferenzschritte, Laufzeit und Koordinationsaufwand. Deshalb sollte ein Use Case nicht reflexartig als „LLM + Websuche + Planner + Reviewer + mehrere Agenten“ umgesetzt werden, wenn ein einzelner Modellaufruf oder einfacher Workflow genügt. Die ausführlichen Kriterien haben wir in unserem Knowledge Nugget [1] beschrieben; hier geht es um ihre Ressourcenseite.

Aus den bisherigen Befunden und Architekturüberlegungen leiten wir vier Gestaltungsprinzipien ab:

1

Mit einer einfachen Ausgangslösung beginnen.

Jede zusätzliche Komponente sollte gegen diese Ausgangslösung getestet und nur bei einem messbaren Mehrwert beibehalten werden.

2

Kontext rollenbezogen begrenzen.

Jeder Agent sollte die für seine Aufgabe nötigen Informationen erhalten; irrelevanter Kontext sollte möglichst entfallen.

3

Budgets und Abbruchregeln definieren.

Modellaufrufe, Wiederholungen, Tool-Fehler und Schleifen sollten begrenzt werden, damit fehlerhafte oder nicht terminierende Abläufe keinen unkontrollierten Aufwand erzeugen.

4

Komplexität nur bei Bedarf eskalieren.

Ein einfacher oder Single-Agent-Pfad kann den Regelfall bearbeiten. Nur bei hoher Komplexität, Unsicherheit oder fehlgeschlagener Verifikation wird an spezialisierte Agenten oder ein Multi-Agenten-System übergeben.

Ein Multi-Agenten-System kann Teil einer Green-AI-Strategie sein, wenn klare Rollen mit kompakten Kontexten und passend dimensionierten Modellen arbeiten. Spezialisierte Agenten können einzelne Teilaufgaben mit kleineren Modellen bearbeiten, während nur anspruchsvolle Schritte ein leistungsfähigeres Modell nutzen. Dadurch kann ein gut gestaltetes Multi-Agenten-System im Einzelfall ähnlich effizient oder effizienter sein als ein monolithisches Single-Agent-System, das jede Teilaufgabe mit demselben großen Modell bearbeitet [1, 19]. Der mögliche Effizienzvorteil ist jedoch **keine Eigenschaft des Labels „Multi-Agent“, sondern eine Hypothese über eine konkrete Architektur**. Sie muss mit realen Testfällen, Qualitätsgrenzen und Kennzahlen über den gesamten Workflow belegt werden [1, 4, 7, 8, 9, 10, 19].

Wie sich diese Abwägung in konkreten Konfigurationen auswirkt, zeigen die folgenden Praxisbeobachtungen.

PRAXISBEOBACHTUNGEN AUS UNSERER EVALUATION

In einem kontrollierten Vergleich verschiedener Konfigurationen eines agentischen Ticket-Workflows zeigte sich exemplarisch, wie eng Architektur, Modellwahl und tatsächlicher Inferenzaufwand zusammenhängen. Verglichen wurden ein Single-Agent-System mit GPT-5.1, ein Multi-Agenten-System mit GPT-5.1 sowie ein Multi-Agenten-System, das überwiegend GPT-5 mini nutzte. Das untersuchte Single-Agent-System bearbeitete die Testfälle am zuverlässigsten und benötigte deutlich weniger Tokens, Laufzeit und Verarbeitungsschritte. Das Multi-Agenten-System mit GPT-5.1 erzielte dagegen die höchste Antwortqualität. Der Einsatz eines kleineren Modells bei mehreren Agenten führte innerhalb der betrachteten Multi-Agenten-Architektur zu qualitativ nur leicht schlechteren Ergebnissen, die immer noch deutlich besser waren als die des Single-Agent-Systems. Der Einsatz des kleineren Modells führte dabei jedoch nicht automatisch zu einem geringeren Gesamtaufwand. Im konkreten Setup erzeugte die kleinere Modellvariante längere Ausgaben, durchlief weiterhin einen mehrstufigen Workflow und erreichte das Ziel weniger zuverlässig; ein möglicher grundsätzlicher Effizienzvorteil des kleineren Modells pro Aufruf spiegelte sich daher nicht in den beobachteten Gesamtkennzahlen wider. Da wir Energieverbrauch und CO₂e bei dem cloudbasierten Dienst nicht direkt messen konnten, nutzten wir den Tokenverbrauch als Proxy: Die Zahl der verarbeiteten Input- und Output-Tokens bildet den Umfang der Modellverarbeitung und damit den Inferenzaufwand zumindest näherungsweise ab; zugleich ist sie unmittelbar kostenrelevant. Die Untersuchung ist auf den betrachteten Use Case begrenzt.

Für diesen Anwendungsfall illustriert sie jedoch eine zentrale Aussage unseres Knowledge Nuggets [1]: Zusätzliche Agenten sind nur dann sinnvoll, wenn ihr Mehrwert den zusätzlichen Koordinations- und Ressourcenaufwand rechtfertigt. Liefert ein Single-Agent-System bereits gute Ergebnisse, kann ein Multi-Agenten-System die Antwortqualität zwar noch etwas verbessern, allerdings zu deutlich höherem Token-, Laufzeit- und Kostenaufwand. Eine bewusste Abwägung zwischen zusätzlichem Qualitätsgewinn und Tokenverbrauch ist daher sowohl aus wirtschaftlichen Gründen als auch aus Green-AI-Sicht geboten. **Nicht das kleinste Modell ist automatisch die nachhaltigste Wahl, sondern das kleinste Modell, das die benötigte Qualität im konkreten System zuverlässig und mit geringem Gesamtaufwand erreicht.**

PRAXISBEOBACHTUNGEN AUS UNSERER EVALUATION

Auch die Granularität der Agentenrollen erwies sich als relevanter Effizienzhebel. AgentMaster [20] zeigt beispielhaft eine modulare Architektur mit einem Orchestrator und spezialisierten Retrieval-Agenten für Aufgaben wie Datenbankabfragen, Informationsrecherche oder Bildanalyse; das Model Context Protocol (MCP) standardisiert dabei den Zugriff auf Tools und Datenquellen. In ergänzenden Versuchen führte bei uns eine noch feingranularere Aufteilung, bei der einzelne Werkzeugfähigkeiten jeweils über eigene Agenten angesprochen wurden, zu vielen Übergaben zwischen Orchestrator und Agenten. Fachlich zugeschnittene Agenten, die innerhalb ihres Verantwortungsbereichs mehrere Tools selbst auswählen und kombinieren konnten, benötigten weniger Orchestrationsiterationen und reduzierten dadurch den Tokenaufwand. **Rollen sollten daher fachlich klar, aber nicht so fein zugeschnitten sein, dass jeder Werkzeugaufwurf zusätzliche Agentenkommunikation auslöst.**

Als weiterer Effizienzhebel zeigte sich die rollenbezogene Modellwahl. In weiteren Versuchen mit den zum Zeitpunkt unserer Evaluation aktuellen OpenAI-Modellen Luna und Sol lieferte das größere Sol-Modell bei fachlichen Textgenerierungsaufgaben etwas bessere Ergebnisse, verursachte jedoch mehr als zehnmal so hohe tokenbasierte Modellkosten. Für die meisten spezialisierten Agenten reichte in unserem Setup das kostengünstigere Luna-Modell aus. In ergänzenden Tests ließ sich Luna auch als Orchestrator einsetzen. Gegenüber Sol war der Qualitätsverlust dabei gering, während Laufzeit und Kosten deutlich sanken. Das leistungsfähigere Sol-Modell war somit vor allem dann sinnvoll, wenn seine zusätzliche Qualität für die Orchestrierung erforderlich war. Die Kostenmessung ist kein direkter Nachweis geringerer Energie- oder CO₂e-Wirkungen, macht aber den wirtschaftlichen Nutzen einer gezielten Modellzuordnung sichtbar. **Da sich Modellfähigkeiten, Versionen und Preise verändern, sollte diese Zuordnung regelmäßig mit repräsentativen Testfällen neu evaluiert werden.**

i
1

Diese Beobachtungen beziehen sich auf die untersuchten Konfigurationen und sind keine allgemeine Aussage gegen Multi-Agenten-Systeme. Andere Rollenverteilungen, kompaktere Kontexte, hybride Verarbeitungspfade, fachlich anders zugeschnittene Agenten oder eine gezielte Zuordnung unterschiedlich leistungsfähiger Modelle können zu einem anderen Verhältnis von Qualität und Aufwand führen [1].

Aus diesen Beobachtungen ergeben sich nicht nur technische, sondern auch organisatorische Anforderungen.

GREEN AGENTIC AI IM UNTERNEHMEN

VERANKERN

Green Agentic AI muss als Design- und Betriebsprinzip in Produktentwicklung, Architektur und Governance verankert werden. **Am Anfang stehen Zielwerte:** Welche Mindestqualität und Erfolgsrate sind erforderlich? Welche Latenz, Kosten und Ressourcenbudgets sind akzeptabel? Erst dann lassen sich Modelle, Routing und Architektur sinnvoll vergleichen.



Model Routing

Einfache Fälle direkt bearbeiten und nur schwierige Fälle per Routing an ein leistungsfähigeres Modell oder eine komplexere Architektur eskalieren.



Messung

Qualität, erfolgreichen Abschluss und Aufwand über den gesamten Workflow messen – Tokens, Laufzeit, Aufrufe, Kosten sowie Energie und CO₂e, soweit verfügbar.



Infrastruktur

Hardware, Rechenzentren und Strommixe verändern die CO₂e-Wirkung desselben Workloads.



Use-Case- und Architekturprüfung

Einfache Ausgangslösung aufbauen, Varianten vergleichen und die Architektur nur bei messbarem Mehrwert erweitern.

Für den Betrieb empfehlen wir, für jeden Workflow das verwendete Modell, die Promptversion, Input- und Output-Tokens, Modell- und Tool-Aufrufe, Schritte, Wiederholungen, Fehler und den erfolgreichen Abschluss zu protokollieren. Mittelwerte reichen oft nicht aus; Ausreißer und lange Laufzeiten können Kosten und Nutzererlebnis prägen. Verfügbare Energie- oder CO₂e-Daten sollten ergänzt, Proxy-Metriken ausdrücklich als solche gekennzeichnet und nur bei konstanten Systemgrenzen verglichen werden [2, 4, 11].

Wie sich die dafür notwendige kontinuierliche Evaluation, geeignete Akzeptanzkriterien und der Umgang mit Veränderungen von Prompts, Daten und Modellversionen in agile Entwicklungsprozesse integrieren lassen, vertieft unser Knowledge Nugget zur agilen Entwicklung generativer KI-Agenten [21].

Daraus entsteht ein Verbesserungsprozess: einfache Ausgangslösung aufbauen, Varianten mit repräsentativen Testfällen vergleichen, Qualität und Aufwand gemeinsam messen und die Architektur nur bei messbarem Mehrwert erweitern. Auch Beschaffung und Infrastruktur gehören dazu, denn Hardware, Rechenzentren und Strommixe verändern die CO₂e-Wirkung desselben Workloads [11, 12, 13]. **Green Agentic AI ist damit eine gemeinsame Aufgabe von Fachbereich, Produktmanagement, Architektur, Betrieb und Nachhaltigkeitsmanagement.**

FÜNF FRAGEN FÜR DESIGN UND BETRIEB

1

Brauchen wir für diese Aufgabe überhaupt generative KI? Oder erfüllt eine klassische, regelbasierte oder spezialisierte Lösung den Zweck zuverlässig?

2

Welche Mindestqualität und Erfolgsrate müssen wir erreichen – und was ist dafür die kleinste zuverlässig ausreichende Lösung? Können wir einfache Fälle direkt bearbeiten und nur schwierige Fälle per Routing an ein leistungsfähigeres Modell oder eine komplexere Architektur eskalieren?

3

Sind System-Prompts, Kontext und Output auf das für die Aufgabe notwendige Maß begrenzt? Ziel, Begriffe, relevante Informationen, Format und Umfang sollten klar definiert und mit repräsentativen Fällen getestet werden.

4

Stiftet jeder zusätzliche Modell- oder Tool-Aufruf, Reasoning-Schritt und Agent einen messbaren Nutzen? Ermöglicht eine Aufteilung auf spezialisierte Rollen zugleich kompaktere Kontexte oder den gezielten Einsatz kleinerer Modelle – und überwiegt dieser Vorteil den zusätzlichen Koordinationsaufwand? Sind Budgets, Abbruchregeln und ein sicherer Rückfallpfad für Fehler und Schleifen definiert?

5

Messen wir über den gesamten Workflow sowohl Qualität und erfolgreichen Abschluss als auch den Aufwand? Dazu gehören mindestens Tokens, Outputlänge, Laufzeit, Modell- und Tool-Aufrufe, Wiederholungen und Kosten sowie Energie, CO₂e und Infrastrukturinformationen, soweit diese belastbar verfügbar sind. Nutzen wir diese Daten anschließend zur kontinuierlichen Verbesserung?

i

Green Agentic AI ist kein neues Architekturmuster, sondern ein Gestaltungsprinzip. Leistungsfähige Modelle und Multi-Agenten-Systeme bleiben wichtig, wenn sie für die Aufgabe einen nachweisbaren Mehrwert schaffen. „So einfach wie möglich“ bedeutet dabei nicht automatisch „so wenige Agenten wie möglich“, sondern die einfachste Gesamtarchitektur, die die erforderliche Qualität und Zuverlässigkeit mit dem geringsten belastbar gemessenen Gesamtaufwand erreicht. Nachhaltig wird der Einsatz, wenn Rechenleistung nicht pauschal maximiert, sondern bewusst skaliert wird: so einfach wie möglich, so leistungsfähig wie nötig und über den gesamten Workflow messbar.

QUELLEN

- [0] Fetzer, D., Gimpel, H., Jung, C., & Strasser, S. (2026). Vom Chatbot zum Multi-Agenten-System: Wie aus reaktiver Inhaltsproduktion koordinierte Autonomie entsteht [Knowledge Nugget]. Fraunhofer-Institut für Angewandte Informationstechnik FIT.
https://www.wi.fit.fraunhofer.de/content/dam/fit/wirtschaftsinformatik/dokumente/01_AWA_Chatbot-Agents-MAS.pdf
- [1] Eichinger, F., Wedelich, P., Fetzer, D., Jung, C., Kraus, M., & Schoch, M. (2026). Wann ist ein Multi-Agenten-System eine gute Wahl? Warum nicht jeder Use Case mehrere Agenten braucht – und wann Multi-Agenten-Systeme sinnvoll sind [Knowledge Nugget]. Fraunhofer-Institut für Angewandte Informationstechnik FIT.
<https://www.wi.fit.fraunhofer.de/content/dam/fit/wirtschaftsinformatik/dokumente/Wann-ein-Multi-Agenten-System-sinnvoll-ist.pdf>
- [2] Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. Communications of the ACM, 63(12), 54–63. <https://doi.org/10.1145/3381831>
- [3] Luccioni, A. S., Jernite, Y., & Strubell, E. (2024). Power Hungry Processing: Watts Driving the Cost of AI Deployment? In Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency (FAcCT '24). <https://doi.org/10.1145/3630106.3658542>
- [4] Poddar, S., Koley, P., Misra, J., Ganguly, N., & Ghosh, S. (2025). Towards Sustainable NLP: Insights from Benchmarking Inference Energy in Large Language Models. In Proceedings of NAACL-HLT 2025, 12688–12704. <https://doi.org/10.18653/v1/2025.naacl-long.632>
- [5] Podder, S., Date, H., & Murthy, S. (2026). Green prompt engineering for sustainable generative AI. Environmental Science and Ecotechnology, 30, 100684.
<https://doi.org/10.1016/j.es.2026.100684>
- [6] Rubei, R., Moussaid, A., Di Sipio, C., & Di Ruscio, D. (2025). Prompt engineering and its implications on the energy consumption of Large Language Models. arXiv:2501.05899.
<https://doi.org/10.48550/arXiv.2501.05899>
- [7] Chen, L., Zaharia, M., & Zou, J. (2023). FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. arXiv:2305.05176.
<https://doi.org/10.48550/arXiv.2305.05176>
- [8] Ding, D., Mallick, A., Wang, C., Sim, R., Mukherjee, S., Rühle, V., Lakshmanan, L. V. S., & Awadallah, A. H. (2024). Hybrid LLM: Cost-Efficient and Quality-Aware Query Routing. International Conference on Learning Representations (ICLR 2024).
https://proceedings.iclr.cc/paper_files/paper/2024/hash/b47d93c99fa22ac0b377578af0a1f63a-Abstract-Conference.html
- [9] Ong, I., Almahairi, A., Wu, V., Chiang, W.-L., Wu, T., Gonzalez, J. E., Kadous, M., & Stoica, I. (2025). RouteLLM: Learning to Route LLMs from Preference Data. International Conference on Learning Representations (ICLR 2025), 34433–34448.
https://proceedings.iclr.cc/paper_files/paper/2025/hash/5503a7c69d48a2f86fc00b3dc09de686-Abstract-Conference.html
- [10] Samsi, S., Zhao, D., McDonald, J., Li, B., Michaleas, A., Jones, M., Bergeron, W., Kepner, J., Tiwari, D., & Gadepally, V. (2023). From Words to Watts: Benchmarking the Energy Costs of Large Language Model Inference. IEEE High Performance Extreme Computing Conference (HPEC 2023).
<https://doi.org/10.1109/HPEC58863.2023.10363447>
- [11] Lannelongue, L., Grealey, J., & Inouye, M. (2021). Green Algorithms: Quantifying the Carbon Footprint of Computation. Advanced Science, 8(12), 2100707.
<https://doi.org/10.1002/advs.202100707>

QUELLEN

- [12] Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., & Dean, J. (2021). Carbon Emissions and Large Neural Network Training. arXiv:2104.10350. <https://doi.org/10.48550/arXiv.2104.10350>
- [13] Luccioni, A. S., Viguier, S., & Ligozat, A.-L. (2023). Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model. Journal of Machine Learning Research, 24(253), 1–15. <https://www.jmlr.org/papers/v24/23-0069.html>
- [14] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 3645–3650. <https://doi.org/10.18653/v1/P19-1355>
- [15] International Energy Agency (IEA). (2025). Energy and AI. IEA, Paris. <https://www.iea.org/reports/energy-and-ai>
- [16] Shi, F., Chen, X., Misra, K., Scales, N., Dohan, D., Chi, E. H., Schärli, N., & Zhou, D. (2023). Large Language Models Can Be Easily Distracted by Irrelevant Context. In Proceedings of the 40th International Conference on Machine Learning (PMLR 202), 31210–31227. <https://proceedings.mlr.press/v202/shi23a.html>
- [17] Żywot, A., Chen, X., Yuan, Y., Søgaard, A., & de Rijke, M. (2026). Can Small Agents Collaborate to Beat a Single Large Language Model? arXiv:2601.11327. <https://doi.org/10.48550/arXiv.2601.11327>
- [18] Mistral AI. (2025, 22. Juli). Our contribution to a global environmental standard for AI. <https://mistral.ai/news/our-contribution-to-a-global-environmental-standard-for-ai/>
- [19] Chen, W., Yuan, J., Qian, C., Yang, C., Liu, Z., & Sun, M. (2025). Optima: Optimizing Effectiveness and Efficiency for LLM-Based Multi-Agent System. In Findings of the Association for Computational Linguistics: ACL 2025, 11534–11557. <https://doi.org/10.18653/v1/2025.findings-acl.601>
- [20] Liao, C. C., Liao, D., & Gadiraju, S. S. (2025). AgentMaster: A Multi-Agent Conversational Framework Using A2A and MCP Protocols for Multimodal Information Retrieval and Analysis. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 52–72. <https://doi.org/10.18653/v1/2025.emnlp-demos.5>
- [21] Fetzer, D., Gimpel, H., Jung, C., Schoch, M., Strasser, S., & Szaruga, P. (2026). Agile Entwicklung von KI-Agenten: Warum generative Agenten zentrale Annahmen agiler Entwicklung infrage stellen – und vier Handlungsempfehlungen für die Praxis [Knowledge Nugget]. Fraunhofer-Institut für Angewandte Informationstechnik FIT. <https://www.wi.fit.fraunhofer.de/content/dam/fit/wirtschaftsinformatik/dokumente/Agile%20Entwicklung%20generativer%20KI-Agenten.pdf>

Fraunhofer-Institut für Angewandte Informationstechnik FIT
Generative AI Lab

Alter Postweg 101, 86159 Augsburg

www.fit.fraunhofer.de<https://www.linkedin.com/company/fraunhofer-fit-generative-ai-lab/>**Ansprechpersonen:**

Prof. Dr. Henner Gimpel

henner.gimpel@fit.fraunhofer.de<https://www.linkedin.com/in/henner-gimpel/>

Prof. Dr. Manfred Schoch

manfred.schoch@fit.fraunhofer.de<https://www.linkedin.com/in/manfred-schoch-42257877/>

**Dieses Knowledge Nugget ist im Rahmen des Forschungsprojekts Agentic Work
Automation (AWA) entstanden**

Finanzielle Förderung:Bayerische
Transformations- und
Forschungsstiftung**Projektpartner:**Ancud*IT* **Fraunhofer**
FIT

soffico

Mitwirkende an diesem Knowledge Nugget:Philip Wedelich, Frank Eichinger, Sebastian Höfling, Louis Sturm
Alle von der DATEV eG