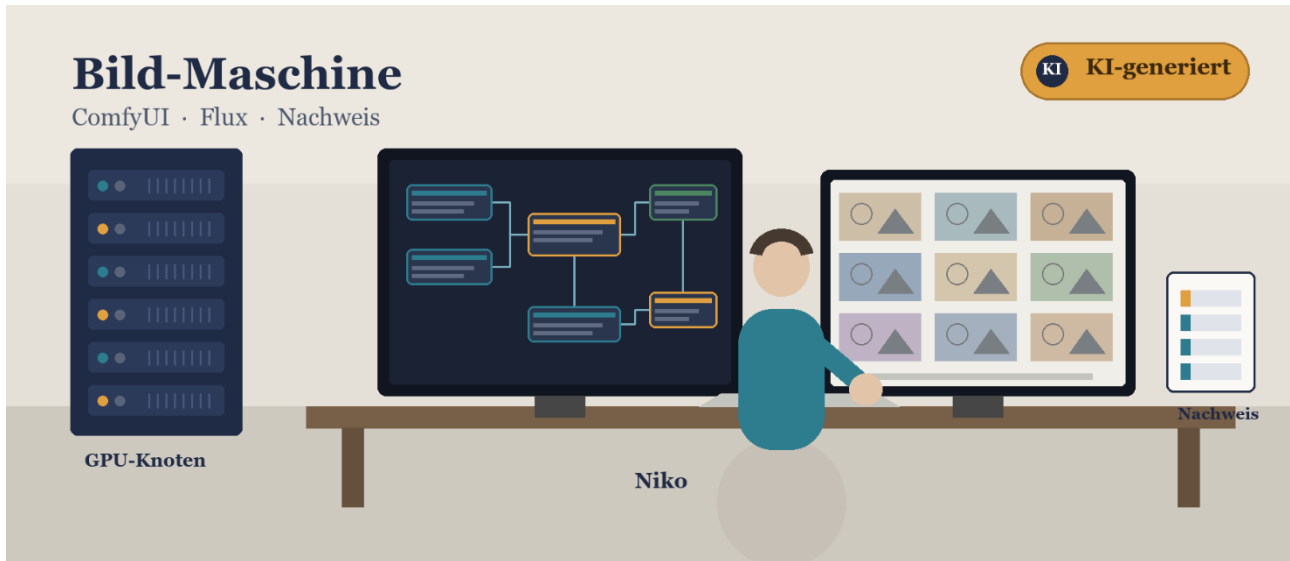


Aufbau einer skalierbaren Bild-Maschine mit ComfyUI und Flux



Niko zwischen Node-Graph, Kontaktbogen und GPU-Knoten: So sieht eine Bild-Maschine im Alltag aus.

WISSEN	BERATUNG	SCHULUNG
Alles zur KI-Bildgenerierung im Überblick KI-Bildgenerierung	Wir bauen die Maschine mit dir zusammen KI-Bildgenerierung – Beratung	Dein Team lernt Workflow, Prompt und Nachweis KI-Bildgenerierung – Schulung

Es ist Donnerstagnachmittag, und irgendjemand aus dem Marketing braucht bis Montag vierzig Bilder. Gleicher Stil, gleiche Lichtstimmung, drei Formate, und bitte so, dass die Rechtsabteilung nicht gleich die Augenbraue hochzieht. Wer an dieser Stelle ein Browserfenster öffnet, einen Prompt eintippt und hofft, hat kein Werkzeugproblem. Der hat ein Fabrikproblem.

Eine Bild-Maschine ist genau das: keine Zauberkiste, sondern eine Fertigung. Sie hat einen Eingang, definierte Stationen, eine Qualitätsstufe, eine Ablage und ein Protokoll. Das Modell, das die Pixel erzeugt, ist darin nur eine Station von fünf — die lauteste zwar, aber nicht die wichtigste. Die wichtigste ist die, an der nachher steht, wer was womit erzeugt hat. Seit dem 2. August 2026 ist das in der EU keine Fleißaufgabe mehr, sondern gilt verbindlich.

In diesem Beitrag baust du diese Maschine mit [ComfyUI](#) als Workflow-Schicht und Flux als Modell. Wir gehen durch die Architektur, durch die Hardware, durch die Lizenzlage — die bei Flux ausgesprochen eigenwillig ist — und durch den Betrieb: Versionierung, Batch-Läufe, Upscaling-Stufe, Ablage mit Metadaten. Am Ende steht eine Pipeline, die du am Freitag um sechs noch anwerfen kannst, ohne dass dir jemand hinterherrschaft, woher das Bild eigentlich kommt.

Wenn du zuerst wissen willst, wie Diffusion überhaupt funktioniert, ist [Basiswissen KI-Bildgenerierung: Wie Diffusion funktioniert](#) der bessere Einstieg. Hier setzen wir voraus, dass du Prompt, Seed und Sampler schon einmal von Hand angefasst hast, und kümmern uns um das, was danach kommt: den Betrieb.

Was eine Bild-Maschine von einem Bildgenerator unterscheidet

Ein Bildgenerator ist ein Automat: Text rein, Bild raus, Ergebnis überraschend. Eine Bild-Maschine ist eine Fertigungsstraße: Auftrag rein, Asset raus, Ergebnis erwartbar. Der Unterschied liegt nicht in der Bildqualität — moderne Modelle liefern auch im Browserfenster hervorragende Bilder. Der Unterschied liegt in vier Eigenschaften, die im Einzelbild niemanden interessieren und ab dem zwanzigsten Bild über alles entscheiden.

Wiederholbarkeit. Du kannst das Bild von vorletzter Woche noch einmal erzeugen, in derselben Anmutung, mit einer winzigen Änderung. Das setzt voraus, dass Seed, Prompt, Modellstand und Workflow-Version irgendwo stehen — und zwar an einem Ort, an dem sie auch noch stehen, wenn die Kollegin, die sie kannte, gerade im Urlaub ist.

Durchsatz. Vierzig Bilder sind kein vierzigfacher Einzelvorgang, sondern ein Batch-Lauf mit einer Sichtung. Der Engpass ist dabei fast nie die GPU. Der Engpass ist die Entscheidung, welche Bilder wegkommen. Wer das Sichten nicht organisiert, hat am Ende eine sehr schnelle Maschine und einen sehr vollen Ordner.

Nachweis. Jedes Bild trägt seine Herkunft mit sich: Modell, Modellversion, Lizenzstufe, Prompt, Seed, Freigeber, Verwendungszweck. Das ist der Teil, den alle für Bürokratie halten, bis zum ersten Mal jemand fragt, ob dieses eine Motiv auf der Messewand eigentlich kommerziell zulässig erzeugt wurde.

Trennung von Spielwiese und Produktion. Explorative Arbeit und produktive Arbeit sind zwei verschiedene Betriebsarten mit zwei verschiedenen Regelwerken. In der Exploration darf alles ausprobiert werden. In der Produktion läuft nur, was freigegeben, versioniert und lizenzrechtlich geklärt ist. Wer beides in derselben Instanz macht, hat irgendwann ein Kampagnenbild aus einem Modell, das niemand geprüft hat.

FAKTEN — Was die Maschine ausmacht

Eine Bild-Maschine besteht aus fünf Schichten: Auftrag, Workflow, Render, Veredelung, Ablage mit Nachweis. Nur die Render-Schicht ist austauschbar — ob Flux lokal, Flux über die API oder ein anderes Modell, ändert an den übrigen vier Schichten nichts. Genau deshalb lohnt es sich, die vier anderen zuerst zu bauen.

Exploration und Produktion sind zwei Betriebsarten

In der Exploration baust du Graphen um, testest Sampler, wirfst Seeds, probierst LoRAs. Das ist Handwerk und soll auch so bleiben; niemand findet einen guten Stil, indem er ein Formular ausfüllt. Was du in dieser Phase erzeugst, ist aber kein Asset, sondern ein Befund: „Dieser Prompt-Aufbau mit diesem Modell bei diesen Schritten trifft unseren Look.“

Der Befund wird dann zur Produktionsvorlage eingedampft. Das heißt: Parameter einfrieren, Zwischenschritte entfernen, Eingaben auf das reduzieren, was der Auftrag wirklich variiert — meist Motivtext, Format und Seed. Alles andere gehört fest in den Graphen. Ein Produktions-Workflow, in dem man an neun Reglern drehen kann, ist kein Produktions-Workflow, sondern eine Spielwiese mit Aktenzeichen.

Merkmal	Explorations-Workflow	Produktions-Workflow
Ziel	Look finden, Grenzen ausloten	gleiches Ergebnis, jederzeit, durch

		jeden
Parameter	alles offen	eingefroren bis auf Motiv, Format, Seed
Modellwahl	frei, auch experimentell	nur freigegebene, lizenzgeprüfte Modelle
Versionierung	Snapshots reichen	Git, Versionsnummer, Änderungsprotokoll
Ergebnis	Befund und Notiz	Asset mit vollständigem Datensatz
Wer darf	geschultes Kernteam	auch Fachabteilung über einfache Oberfläche

Die Trennung hat noch einen angenehmen Nebeneffekt: Sie beendet die Debatte, ob die KI jetzt Kreativität ersetzt. Die Maschine übernimmt die Wiederholung. Der Kopf übernimmt den Look. Wer die Wiederholung von Hand macht, ist nicht kreativ, sondern nur beschäftigt.

Warum ausgerechnet ComfyUI

ComfyUI zerlegt die Bilderzeugung in einzelne Knoten: Modell laden, Prompt kodieren, Sampler laufen lassen, Latent dekodieren, hochskalieren, speichern. Das wirkt beim ersten Blick auf den Graphen wie ein Schaltplan aus einem Elektronikbuch — ist aber genau die Eigenschaft, die du für eine Fertigung brauchst: Jeder Schritt ist sichtbar, einzeln austauschbar und maschinell ansteuerbar.

Drei Eigenschaften machen ComfyUI zur Werkbank der Wahl. Erstens die **Subgraphs**: Du kannst eine Gruppe von Knoten zu einem einzigen Knoten zusammenfalten, benennen und mit definierten Ein- und Ausgängen versehen. Subgraphs lassen sich verschachteln, wieder auspacken und über ein Parameter-Panel von außen bedienen, ohne dass man sie öffnen muss. Damit wird aus dem Kabelsalat eine Stückliste: „Basis-Render“, „Upscaling-Stufe“, „Ablage-Stufe“ — drei Kacheln statt sechzig Knoten.

Zweitens die **API**. ComfyUI ist nicht nur eine Oberfläche, sondern ein Server, der standardmäßig auf Port 8188 lauscht. Über HTTP schickst du einen Auftrag an /prompt, bekommst eine prompt_id zurück, fragst den Stand über /history ab und holst das Ergebnis über /view. Dazu kommen Endpunkte für die Warteschlange, den Upload von Eingangsbildern, die Systemauslastung und ein WebSocket für Fortschrittsmeldungen. Das ist alles, was du brauchst, um die Maschine von außen zu füttern.

Drittens die **Metadaten**. ComfyUI schreibt den kompletten Graphen in die PNG-Datei. Zieh ein erzeugtes Bild zurück in die Oberfläche, und du hast den Workflow wieder vor dir, mit allen Parametern. Für die Rekonstruktion eines Motivs ist das Gold wert. Als Nachweissystem taugt es trotzdem nicht — dazu später mehr, denn genau hier tappen die meisten in die erste Falle.

TIPP — Zwei JSON-Formate, ein Stolperstein

Der Workflow, den du in der Oberfläche speicherst, ist **nicht** das Format, das die API akzeptiert. Für /prompt brauchst du den Export im API-Format. Wer das verwechselt, bekommt eine Validierungsmeldung und sucht eine Stunde am falschen Ende. Nimm dir an, beide Fassungen jedes Workflows gemeinsam abzulegen — Oberflächenformat zum Weiterbauen, API-Format zum Ausführen.

Flux: Varianten, Lizenzen und was das für dein Budget heißt

Flux kommt von Black Forest Labs, einem Team, in dem Leute sitzen, die vorher an Stable Diffusion gebaut haben. Fachlich ist das eine der stärksten Modellfamilien für Bilderzeugung und Bildbearbeitung. Organisatorisch ist es eine Familie, bei der du die Lizenzbedingungen tatsächlich lesen musst, bevor du irgendetwas veröffentlichst. Die Varianten unterscheiden sich nämlich nicht nur in der Qualität, sondern in dem, was du mit dem Ergebnis machen darfst.

Die Varianten im Überblick

FLUX.2 [dev] ist das offene Schwergewicht: ein Flow-Matching-Transformer mit 32 Milliarden Parametern, der Text-zu-Bild und Bildbearbeitung in einem einzigen Checkpoint vereint, einschließlich Bearbeitung mit mehreren Referenzbildern und Ausgabe bis vier Megapixel. Die Gewichte sind frei herunterladbar — und stehen unter der FLUX Non-Commercial License. Für kommerzielle Nutzung brauchst du eine separate Lizenz von Black Forest Labs.

FLUX.2 [klein] ist die destillierte Familie für schmale Hardware. Die 4B-Variante steht seit dem 15. Januar 2026 unter Apache 2.0 — also uneingeschränkt kommerziell nutzbar, weiterverteilbar, abwandelbar — und erzeugt Bilder in vier Schritten, teilweise unter einer Sekunde. Die 9B-Variante ist qualitativ deutlich stärker, steht aber wieder unter der nicht-kommerziellen Lizenz. Wer das übersieht, hat sich genau den Fallstrick eingebaut, den die Apache-Variante eigentlich beseitigen sollte.

FLUX.2 [pro] und [flex] sind proprietär und laufen nur über die API. [pro] ist auf Qualität und Geschwindigkeit getrimmt, [flex] gibt dir Schritte und Guidance in die Hand und ist stark beim Rendern von Text im Bild. Abgerechnet wird nach Megapixeln: Für [pro] liegt der Einstiegspreis bei rund drei US-Cent pro Megapixel, Bildbearbeitung etwas höher, ein Vier-Megapixel-Render entsprechend bei etwa zwölf Cent. Abonnement gibt es keins, du zahlst, was du erzeugst.

Variante	Zugang	Lizenz	Sinnvoll für
FLUX.2 [klein] 4B	Gewichte, lokal	Apache 2.0	Serienbetrieb ohne Lizenzfragen, Vorschauen, hoher Durchsatz
FLUX.2 [klein] 9B	Gewichte, lokal	nicht-kommerziell	interne Tests, Qualitätsvergleich
FLUX.2 [dev]	Gewichte, lokal	nicht-kommerziell, kommerziell nur mit Zusatzlizenz	höchste Qualität im eigenen Haus, wenn die Lizenz steht
FLUX.2 [pro]	nur API	proprietär, kommerziell	Spitzenqualität ohne eigene GPU, Abrechnung je Megapixel
FLUX.2 [flex]	nur API	proprietär, kommerziell	Text im Bild, feine Steuerung von Schritten und Guidance

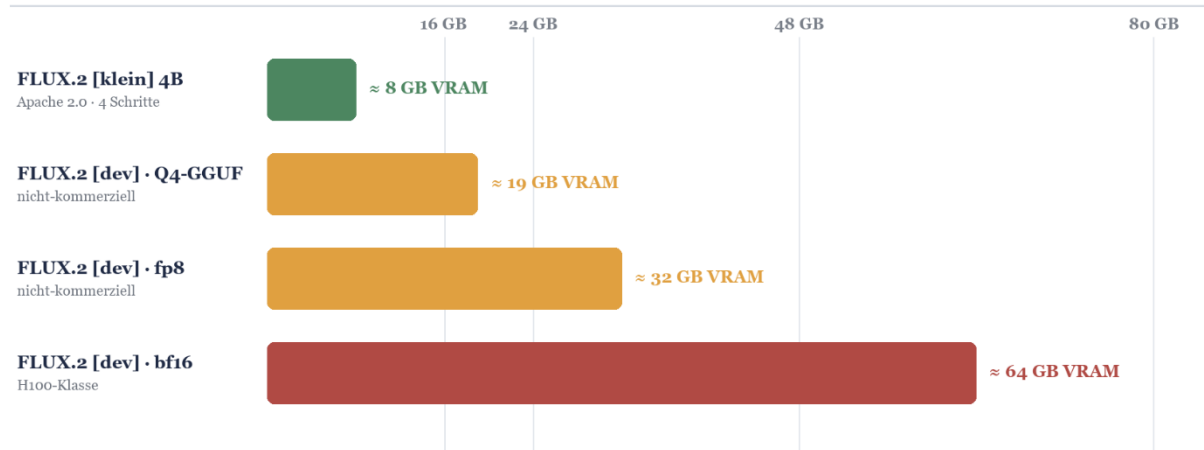
Die Selbsthosting-Lizenzen sind gestaffelt: eine Einstiegerstufe für Entwicklerteams mit Feintuning- und LoRA-Rechten und rund zehntausend Bildern im Monat auf einer Domain, eine Plattformstufe mit rund hunderttausend Bildern im Monat für Produktteams, eine Agenturstufe für eine begrenzte Zahl von Kunden mit Zusatzgebühr je weiterem Kunden, und eine Unternehmensstufe mit ausgehandelten Bedingungen und Zugriff auf alle Modelle. Dazu kommt eine gesonderte Lizenz für den Fall, dass du mit Flux-Ausgaben eigene Modelle trainieren willst.

WARNUNG — Die nicht-kommerzielle Lizenz ist kein Formfehler

Ein Bild aus FLUX.2 [dev] oder [klein] 9B ohne kommerzielle Zusatzlizenz auf der Unternehmenswebsite, im Verkaufsprospekt oder in einer Kundenpräsentation ist ein Lizenzverstoß — auch dann, wenn es „nur intern angefangen“ hat und später doch nach außen ging. Das gilt unabhängig davon, ob das Bild je jemandem auffällt. Wenn dein Betriebsmodell Produktivbilder vorsieht, klärst du die Lizenzstufe, bevor die erste GPU-Stunde läuft, nicht danach.

Was die Varianten an Speicher fressen

Speicherbedarf: welche Flux-Variante auf welche Karte passt



Dazu kommt der Text-Encoder.

FLUX.2 nutzt ein Mistral-basiertes Sprachmodell als Encoder — als fp8-Variante oder ausgelagert auf einen zweiten Dienst.

Speicherbedarf der Flux-Varianten gegen übliche Kartengrößen. Der Text-Encoder kommt jeweils obendrauf.

Die Zahlen entscheiden über deine Hardwarefrage. FLUX.2 [dev] in voller Präzision braucht eine Karte der H100-Klasse. In fp8 landet man bei etwa 32 Gigabyte für das Diffusionsmodell, ein Q4-Quantisat drückt das auf rund 19 Gigabyte und passt damit auf eine 24-GB-Karte — mit Abstrichen bei feinen Details, die je nach Motiv kaum auffallen oder sofort. FLUX.2 [klein] 4B kommt mit rund acht Gigabyte aus und läuft damit auf Hardware, die in vielen Häusern ohnehin herumsteht.

Dazu kommt der Text-Encoder, und der ist bei FLUX.2 kein kleiner CLIP mehr, sondern ein Mistral-basiertes Sprachmodell mit 24 Milliarden Parametern. Es gibt ihn als fp8-Variante, und man kann ihn auf einen separaten Dienst auslagern, damit er nicht dauerhaft die Renderkarte blockiert. Praktisch brauchst du drei Dateien in ComfyUI: das Diffusionsmodell unter `diffusion_models`, den Text-Encoder unter `text_encoders` und die zugehörige VAE unter `vae`. Was eine VAE eigentlich tut, steht in [Checkpoints, Modelle und VAE](#).

Der Encoderwechsel hat eine Konsequenz, die viele Prompt-Gewohnheiten über den Haufen wirft: Ein Sprachmodell als Encoder reagiert auf beschreibende Sätze, nicht auf Stichwortsalat mit Kommata. Ein kurzer Absatz zu Bildaufbau, Licht, Material und — falls nötig — wörtlich zitiertem Text im Bild schlägt dreißig aneinandergereihte Schlagworte deutlich. Wer in [Prompt und Negativ-Prompt in der KI-Bildgenerierung](#) die alten Regeln gelernt hat, muss sie für diese Modellgeneration umlernen.

Als Schrittzahl liegt FLUX.2 [dev] typischerweise zwischen 20 und 50, mit 28 als brauchbarem Ausgangswert; die destillierten klein-Varianten kommen mit vier Schritten aus. Warum mehr Schritte ab einem bestimmten Punkt nichts mehr bringen außer Stromkosten, erklärt [Sampling-Steps erklärt](#); welcher Sampler dazu passt, steht in [Sampler & Scheduler erklärt](#).

Lokal, Cloud-GPU oder API — die ehrliche Rechnung

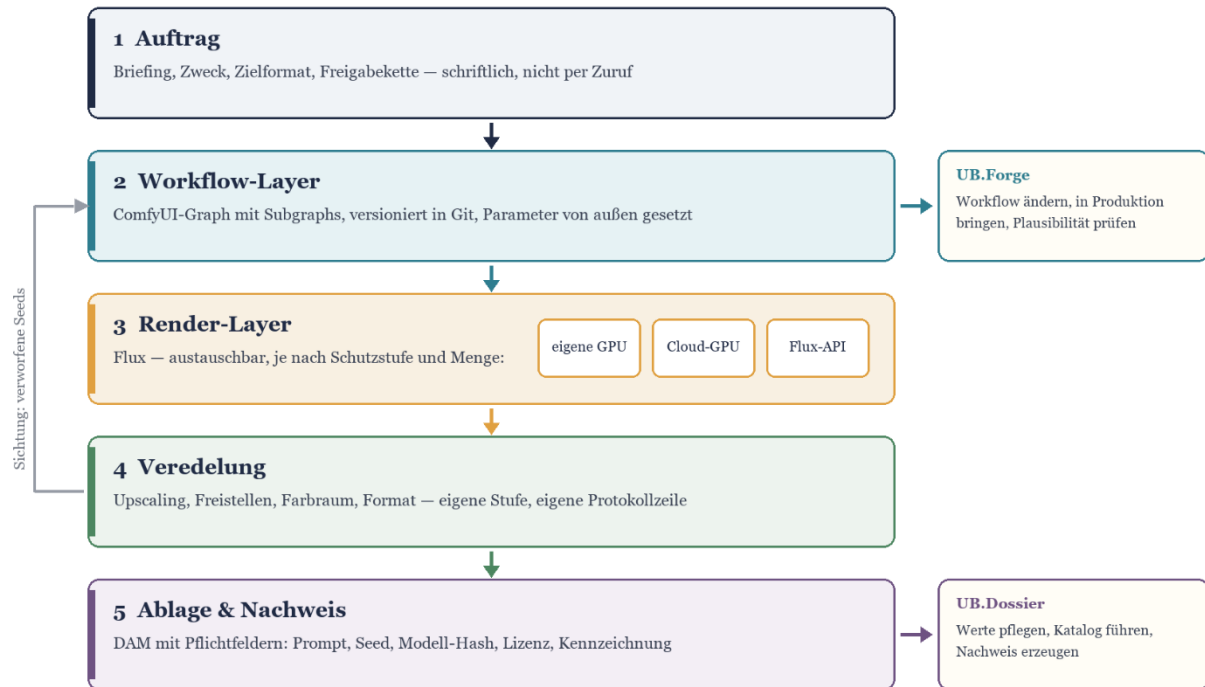
Es gibt keine allgemeingültige Antwort, aber es gibt eine ehrliche Rechnung. Eine eigene Karte kostet einmal Geld und danach Strom; eine Cloud-GPU kostet pro Stunde; die API kostet pro Megapixel. Stundensätze für gemietete GPUs bewegen sich bei spezialisierten Anbietern derzeit grob zwischen einem halben und einem Dollar für eine RTX-5090-Klasse, um anderthalb Dollar für eine L40S und zwischen zweieinhalb und gut drei Dollar für eine H100 — bei den großen Hyperscalern deutlich darüber. Die API liegt, wie erwähnt, bei wenigen Cent je Bild.

Kriterium	Eigene GPU im Haus	Cloud-GPU auf Zuruf	Flux-API
Kostenmodell	Anschaffung plus Strom	pro Stunde, elastisch	pro Megapixel
Lohnt ab	durchgehender Betrieb	Stoßzeiten, Projektspitzen	geringe bis mittlere Mengen
Daten verlassen das Haus	nein	ja, an den Anbieter	ja, an den Modellanbieter
Lizenzfrage	musst du selbst klären	musst du selbst klären	im Preis enthalten
Modellauswahl	alle offenen Gewichte	alle offenen Gewichte	nur pro und flex
Betriebsaufwand	hoch: Treiber, Updates, Monitoring	mittel: Image und Ablauf pflegen	gering
Vorhersagbarkeit	hoch, aber Kapazität gedeckelt	mittel, Preise schwanken	hoch, skaliert mit der Rechnung

Die Praxis läuft meist auf eine Mischung hinaus: destillierte klein-Modelle lokal für Vorschauen und Massenvarianten, die API für die letzten fünf Prozent Qualität am Hero-Motiv. Das setzt voraus, dass dein Workflow die Render-Stufe austauschen kann, ohne dass der Rest der Maschine wackelt — was genau der Grund ist, warum wir sie als eigene Schicht bauen. Die grundsätzliche Abwägung behandelt [Cloud vs. lokale KI-Bildgenerierung](#) ausführlicher, und wer noch andere Modelle in die engere Wahl nimmt, findet in [Die 15 wichtigsten Modelle zur lokalen Bildgenerierung](#) das Vergleichsfeld.

Die Architektur: fünf Schichten, die du getrennt halten musst

Die fünf Schichten einer produktiven Bild-Maschine



Die fünf Schichten der Bild-Maschine. Nur die Render-Schicht ist austauschbar — der Rest bleibt.

Die Architektur ist bewusst langweilig aufgebaut, weil langweilige Architekturen den Betrieb überleben. Jede Schicht hat eine Aufgabe, eine Schnittstelle und ein Protokoll. Wenn du eine Schicht austauschst, bleiben die anderen vier stehen.

Schicht 1 und 2: Auftrag und Workflow

Der Auftrag ist der Punkt, an dem die meisten Maschinen schon verlieren. Ein Zuruf im Flur ist kein Auftrag. Ein Auftrag hat einen Zweck, ein Zielformat, eine Freigabekette und einen Verwendungsort — und der Verwendungsort ist keine Formalie, sondern entscheidet über die Lizenzstufe und über die Kennzeichnungspflicht. Ob das ein Ticket, ein Listeneintrag oder ein Formular ist, ist gleichgültig, solange es schriftlich und auffindbar ist.

Die Workflow-Schicht ist der ComfyUI-Graph. Hier gilt ein einziges hartes Prinzip: **Der Workflow ist Quellcode**. Er liegt in einem Repository, hat eine Versionsnummer, eine Änderungshistorie und einen Verantwortlichen. Er wird nicht auf dem Produktionsserver „mal eben“ umgeklickt. Wenn er geändert wird, geht das über denselben Weg wie jede andere Änderung: ändern, prüfen, freigeben, ausrollen.

Genau an dieser Stelle setzt UB.Forge an: Änderungen am Workflow werden dort erfasst, auf Plausibilität geprüft und kontrolliert in die Produktion gebracht, statt direkt am laufenden Graphen zu passieren. Der Unterschied klingt nach Formalismus und ist in Wahrheit der Unterschied zwischen „wir wissen, warum die Bilder seit Dienstag anders aussehen“ und „wir wissen es nicht“.

Für die Struktur des Graphen selbst haben sich drei Regeln bewährt. Erstens: Jede Stufe ist ein Subgraph mit sprechendem Namen. Zweitens: Was der Auftrag variiert, wird über Eingangsknoten hereingereicht, alles andere ist fest verdrahtet. Drittens: Keine Pfade, Modellnamen oder Seeds irgendwo tief im Graphen verstecken — sie gehören an die Oberfläche, wo man sie sieht und protokollieren kann.

Subgraph	Aufgabe	Von außen einstellbar
01_Eingang	Prompt, Negativ-Prompt, Format, Seed-Liste entgegennehmen	alles
02_Basis-Render	Modell laden, kodieren, sampeln, dekodieren	nur Seed und Schrittzahl
03_Stilanker	LoRA oder Referenzbilder für den Hauslook	an/aus, Gewichtung
04_Veredelung	Upscaling, Schärfung, Farbraum, Beschnitt	Zielauflösung
05_Ablage	Dateiname, Ordner, Metadaten schreiben	Auftragsnummer

Schicht 3: Render — und warum Batch nicht gleich Batch ist

In der Render-Schicht steckt die einzige wirklich rechenintensive Arbeit. Hier entscheidet sich auch, ob dein Durchsatz stimmt. Ein Batch-Lauf erzeugt mehrere Bilder in einem Anlauf, und dabei gibt es zwei völlig verschiedene Sorten von Batch, die im Alltag ständig verwechselt werden.

Die eine Sorte ist die Variantenerzeugung: gleicher Prompt, verschiedene Seeds. Das ist die Sichtung, der Kontaktbogen, die Auswahl. Die andere Sorte ist die Serienerzeugung: verschiedene Prompts, jeweils fester Seed, weil vierzig Produktbilder in gleicher Anmutung gebraucht werden. Die erste Sorte braucht Streuung, die zweite braucht Stabilität. Wer beides in denselben Lauf packt, bekommt vierzig Bilder, von denen dreißig nicht zusammenpassen.

WICHTIG — Batch-Größe verändert das Ergebnis nicht — die Seed-Vergabe schon

Ein Bild ist reproduzierbar, wenn Seed, Prompt, Modellstand, Sampler, Schrittzahl und Auflösung gleich sind. Wenn ein Lauf mit batch_size 8 andere Bilder liefert als acht Einzelläufe, liegt das in aller Regel daran, wie die Seeds über den Batch vergeben werden. Das ist wichtig genug, um es einmal sauber nachzulesen: [batch_size und Reproduzierbarkeit](#) und [Der Seed bei der KI-Bildgenerierung](#).

Für den Hauslook ist die Render-Schicht auch der Ort, an dem Stilanker sitzen — ein trainiertes LoRA auf euren Bildstil oder Referenzbilder für die Multi-Referenz-Bearbeitung, die FLUX.2 ohne zusätzliches Training beherrscht. Was ein LoRA ist und wann sich eines lohnt, steht in [Was ist ein LoRA?](#); wie man daraus einen über Monate stabilen Markenlook macht, zeigt [Markenkonsistente KI-generierte Bilder](#).

Schicht 4: Die Veredelungsstufe verdient eine eigene Stufe

Direkt in Zielauflösung zu rendern ist verlockend und meistens falsch. Der Standardweg ist: in moderater Auflösung erzeugen, sichten, und nur die ausgewählten Motive in einer separaten Stufe hochskalieren. Das spart den Löwenanteil der Rechenzeit, weil du neununddreißig verworfene Bilder nicht in Druckqualität berechnest.

Die Veredelungsstufe macht mehr als Vergrößern. Sie setzt das Seitenverhältnis auf das Zielformat, beschneidet definiert statt zufällig, legt den Farbraum fest, schärft maßvoll und schreibt das Ausgabeformat. Dass Auflösung und Seitenverhältnis keine Nachgedanken sind, sondern das Ergebnis der Bilderzeugung selbst beeinflussen, behandelt [Auflösung & Seitenverhältnis bei KI-Bildern](#).

Wichtig ist, dass die Stufe versioniert ist wie alles andere. Ein getauschtes Upscaling-Modell verändert die Anmutung der gesamten Serie. Wenn im Protokoll steht, welche Upscaler-Version gelaufen ist, findest du die Ursache in fünf Minuten. Wenn nicht, diskutierst du drei Tage über Monitore.

Schicht 5: Ablage mit Metadaten — der Teil, der später zählt

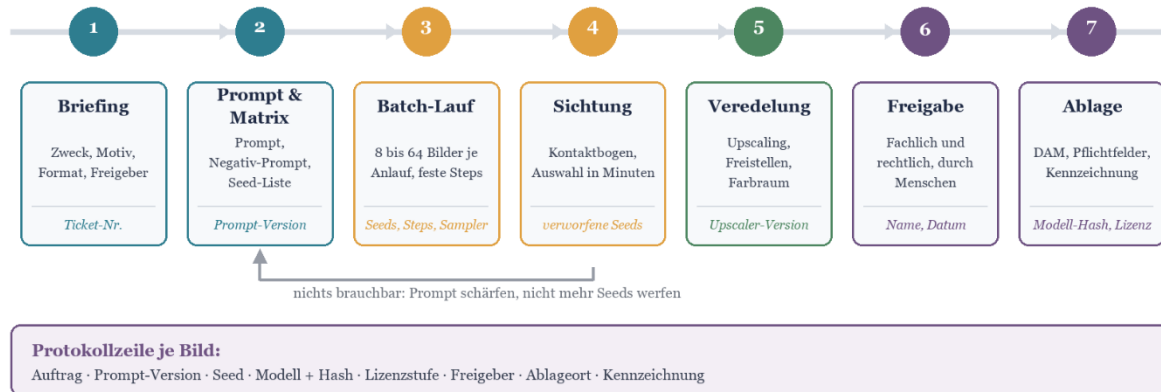
Ein Bild ohne Datensatz ist eine Datei, kein Asset. Die Ablage-Schicht schreibt deshalb nicht nur die Datei, sondern den kompletten Herkunftsdatensatz. ComfyUI legt den Graphen zwar in die PNG-Metadaten, aber darauf allein darfst du dich nicht verlassen: Jede Nachbearbeitung, jede Formatkonvertierung, jeder gut gemeinte Komprimierungsdienst kann diese Metadaten entfernen. Der Nachweis muss außerhalb der Bilddatei stehen — in einem Katalog, der auch dann noch da ist, wenn jemand das JPEG durch ein Web-Optimierungstool gejagt hat.

Feld	Beispielinhalt	Wofür du es brauchst
Auftrag	KAMP-2026-114	Zweck und Freigabekette wiederfinden
Workflow	produktfoto_v4.2	Wiederholbarkeit, Ursachensuche bei Abweichungen
Modell + Hash	FLUX.2 [klein] 4B, sha256 3f9c...	Lizenznachweis, Versionsnachweis
Lizenzstufe	Apache 2.0 / Selbsthosting Plattform	Beleg, dass die Nutzung zulässig ist
Prompt / Negativ-Prompt	voller Wortlaut	Nachvollziehbarkeit, spätere Varianten
Seed, Steps, Sampler, CFG	774312 / 28 / euler / 3.5	exakte Reproduktion
Veredelung	Upscaler v2, 4x, sRGB	Serienkonsistenz
Freigabe	Name, Datum, Rolle	wer die Verantwortung getragen hat
Kennzeichnung	KI-generiert, sichtbar + maschinenlesbar	Transparenzpflicht seit August 2026

Das ist die Schicht, in der UB.Dossier arbeitet: Werte pflegen, Katalog führen, Nachweis auf Knopfdruck erzeugen. Der Reiz liegt nicht darin, eine Tabelle zu haben — den Reiz merkst du an dem Tag, an dem jemand für ein einzelnes Bild wissen will, welches Modell unter welcher Lizenz es erzeugt hat, und du nicht in vierzehn Ordnern suchst. Wie man so einen Katalog mit Bordmitteln aufsetzt, beschreibt [Digital Asset Management für KI-Bilder mit SharePoint Online und Power Platform](#).

Betrieb: Versionierung, Warteschlange, Skalierung, Kennzahlen

Ein Batch-Auftrag von der Anfrage bis zum Nachweis



Die Maschine wird nicht dadurch schneller, dass du mehr Bilder erzeugst, sondern dadurch, dass du schneller entscheidest, welche du wegwirfst.

Ein Auftrag durchläuft sieben Stationen. An jeder entsteht eine Protokollzeile.

Workflow-Versionierung, die tatsächlich gelebt wird

Workflows verhalten sich wie Skripte, also behandelst du sie wie Skripte. Beide JSON-Fassungen — Oberflächenformat und API-Format — liegen im Repository, jede Änderung bekommt eine kurze Begründung, und die Versionsnummer steht im Dateinamen und wird in jedes erzeugte Bild geschrieben. Wenn produktfoto_v4.1 und produktfoto_v4.2 nebeneinander existieren, muss aus dem Protokoll hervorgehen, welche Serie mit welcher gelaufen ist.

Dazu gehört ein zweiter, unbequemerer Punkt: die Umgebung. Ein Workflow ist nur reproduzierbar, wenn auch die installierten Erweiterungen reproduzierbar sind. ComfyUI lebt von Custom Nodes, und Custom Nodes aktualisieren sich gern. Friere die Versionsstände ein, dokumentiere sie, und aktualisiere sie als bewussten Vorgang mit anschließendem Testlauf. Ein fester Testauftrag mit fixem Seed, der nach jedem Update durchläuft und dessen Ergebnis du vergleichst, kostet drei Minuten und erspart dir die Erkenntnis vier Wochen später, dass sich seit einem Update alle Hauttöne verschoben haben.

TIPP — Der Kanarienvogel-Auftrag

Leg einen festen Referenzauftrag an: ein Prompt, ein Seed, feste Parameter, bekanntes Ergebnis. Nach jedem Update von ComfyUI, Custom Nodes, Treibern oder Modellen läuft er einmal durch. Weicht das Bild ab, weißt du sofort, dass sich etwas verändert hat — und zwar bevor es in einer Kundenserie auffällt.

Warteschlange und Anbindung

ComfyUI arbeitet Aufträge über eine Warteschlange ab. Für den Anfang reicht das: eine Instanz, eine Karte, Aufträge laufen der Reihe nach. Der Ausbau folgt einem bewährten Pfad, und man sollte ihn nicht überspringen: erst einen Workflow sauber über /prompt und /history von Anfang

bis Ende zum Laufen bringen; dann den WebSocket dazunehmen, wenn jemand Fortschritt sehen soll; dann den Bild-Upload, wenn Workflows Eingangsbilder brauchen.

Erst wenn Prioritäten, Wiederholungen oder Mengenbegrenzungen nötig werden, kommt eine eigene Auftragsverwaltung davor — eine Warteschlange, die entscheidet, wer wann darf, und ComfyUI nur noch als Ausführungsmaschine benutzt. Das ist der Punkt, an dem aus dem Werkzeug ein Dienst wird. Es ist auch der Punkt, an dem man aufhören sollte, alles selbst zu bauen, und ehrlich prüft, ob der Eigenbau den Aufwand noch wert ist.

Ausbaustufe	Aufbau	Trägt bis etwa	Preis dafür
Werkbank	eine Instanz, eine GPU, Bedienung über die Oberfläche	ein bis drei Personen	kein Betriebsaufwand, aber alles manuell
Dienst	eine Instanz, Aufträge über die API, feste Workflows	eine Abteilung	Skripte und eine einfache Auftragsliste
Fertigung	mehrere Worker, vorgeschaltete Warteschlange, Prioritäten	mehrere Abteilungen	echter Betrieb: Monitoring, Bereitschaft, Updates
Hybrid	lokale Worker plus API für Spitzen und Spitzenqualität	schwankende Last	zwei Kostenmodelle, zwei Lizenzlagen

Kennzahlen, die etwas aussagen

Kennzahlen sind in diesem Umfeld schnell albern. „Erzeugte Bilder pro Woche“ misst, wie gern das Team auf einen Knopf drückt, sonst nichts. Nützlich sind Kennzahlen, die etwas über die Qualität der Maschine sagen, nicht über ihre Lautstärke.

Kennzahl	Was sie misst	Worauf sie hinweist, wenn sie schlecht ist
Trefferquote erster Lauf	Anteil Aufträge, die ohne neuen Anlauf durchgehen	unklare Briefings oder ein Prompt-Bauplan, der nicht sitzt
Durchlaufzeit	Auftragseingang bis freigegebenes Asset	Freigabe ist der Engpass, nicht die Rechenzeit
Sichtungsdauer je Serie	Zeit von Batch-Ende bis Auswahl	zu viele Varianten, zu wenig Kriterien
Kosten je Asset	Rechenzeit plus Arbeitszeit geteilt durch Assets	Rechenzeit wird für Bilder verbrannt, die niemand ansieht
Nachweisquote	Anteil Assets mit vollständigem Datensatz	die Ablage-Schicht wird umgangen — das teuerste aller Probleme
Wiederverwendungsquote	Anteil Aufträge, die vorhandene Assets lösen	der Katalog ist nicht durchsuchbar

Die Nachweisquote ist die einzige, bei der hundert Prozent das Ziel ist. Alle anderen dürfen schwanken. Wenn die Wiederverwendungsquote steigt, hast du übrigens nichts falsch gemacht: Der billigste Rendervorgang ist der, den du nicht startest, weil das Bild schon existiert und jemand es gefunden hat.

Ein Praxisbeispiel: die Bild-Maschine hinter boddenberg.de

Weil Ratschläge leichter zu geben als zu befolgen sind: Die Bilder auf boddenberg.de kommen aus genau so einer Maschine. Sie läuft auf einer lokalen Karte, arbeitet mit versionierten ComfyUI-Workflows, erzeugt Varianten in Serien, schiebt nur ausgewählte Motive durch die Veredelungsstufe und schreibt zu jedem veröffentlichten Bild einen Datensatz mit Modell, Lizenzstufe, Prompt, Seed und Workflow-Version. Die Kennzeichnung als KI-generiert hängt an

derselben Stelle im Ablauf wie der Dateiname — nicht, weil das hübsch ist, sondern weil sie sonst irgendwann vergessen wird.

Der Ertrag zeigt sich nicht am einzelnen Bild, sondern an der Wiedervorlage. Ein Motiv, das vor Monaten entstanden ist, lässt sich in einer neuen Variante erzeugen, ohne dass jemand rekonstruieren muss, wie es damals aussah. Und wenn in einem Beitrag eine Person mit einem Namen auftaucht, ist nachvollziehbar, welcher Stilanker dahintersteckt. Warum ich Werkzeuge erst selbst benutze, bevor ich sie empfehle, steht in [Dogfooding: Ich setze ein, was ich berate](#).

Ehrlich bei den Grenzen: Diese Maschine ist auf eine Person plus Gastnutzung ausgelegt. Sie hat keine vorgeschaltete Warteschlange mit Prioritäten, kein Mehrmandantenkonzept und keine Bereitschaft um drei Uhr nachts. Für einen Konzern mit acht Marken und vierzig Leuten in der Produktion sieht die Sache anders aus — dann brauchst du die Fertigungsstufe aus der Tabelle oben, mit allem, was an Betrieb daran hängt. Wer an diesem Punkt Architektur und Betriebsmodell zusammen durchrechnen will, ist bei der [KI-Bildgenerierung – Beratung](#) richtig; für das Team, das die Maschine danach bedient, gibt es die [KI-Bildgenerierung – Schulung](#).

Nachweis, Lizenz und Recht: der unspektakuläre Teil, der dich rettet

Dieser Abschnitt ist der Grund, warum die Maschine fünf Schichten hat und nicht drei. Technisch funktioniert eine Pipeline auch ohne Protokoll wunderbar. Rechtlich funktioniert sie dann nicht mehr.

Die Transparenzpflicht ist da, nicht angekündigt

FAKTEN — Seit dem 2. August 2026 gilt Artikel 50

Die Transparenzpflichten des EU AI Act sind seit dem 2. August 2026 verbindlich anwendbar. KI-erzeugte oder KI-veränderte Bild-, Video- und Audioinhalte müssen maschinenlesbar gekennzeichnet sein; bei Inhalten, die täuschend echt wirken — insbesondere Darstellungen von Personen, Ereignissen und Orten — kommt eine erkennbare Offenlegung gegenüber dem Publikum hinzu. Für erkennbar künstlerische oder satirische Werke genügt eine zurückhaltende Form, etwa ein Hinweis in der Bildunterschrift. Bußgelder können bis zu 15 Millionen Euro oder drei Prozent des weltweiten Jahresumsatzes betragen.

Praktisch heißt das zweierlei. Die maschinenlesbare Kennzeichnung gehört in die Ablage-Schicht, automatisch, ohne dass jemand daran denken muss. Herkunftsmetadaten nach dem C2PA-Standard sind dafür eine der zulässigen Umsetzungsformen — signierte Angaben zu Urheberschaft, Erzeugungszeitpunkt, verwendeten Werkzeugen und jedem Bearbeitungsschritt. Die sichtbare Offenlegung dagegen ist eine redaktionelle Entscheidung pro Verwendungsort und gehört deshalb in den Auftrag: Wer nicht weiß, wo ein Bild erscheint, kann nicht entscheiden, wie es gekennzeichnet werden muss.

Und weil Metadaten unterwegs verloren gehen, ist die Kennzeichnung im Katalog der belastbare Teil und die in der Datei der komfortable. Was die Erkennung von KI-Bildern technisch hergibt und wo sie an Grenzen stößt, behandelt [KI-generierte Bilder erkennen](#). Den größeren regulatorischen Rahmen und was seither sonst noch gilt, fasst [EU AI Act: Was seit dem 2. August 2026 gilt](#) zusammen.

Lizenznachweis je Bild, nicht je Installation

Die Lizenzfrage stellt sich nicht auf der Ebene „wir setzen Flux ein“, sondern auf der Ebene „dieses Bild stammt aus diesem Modell“. Eine Maschine, in der ein Apache-lizenziertes klein-Modell und ein nicht-kommerziell lizenziertes dev-Modell nebeneinander installiert sind, ist völlig in Ordnung — solange für jedes erzeugte Bild feststeht, welches von beiden gelaufen ist. Deshalb steht der Modellname mitsamt Prüfsumme im Datensatz und nicht nur in einer Betriebsdokumentation.

WARNUNG — Modelle nicht frei herunterladbar lassen

Wenn jeder beliebige Gewichte aus dem Netz in den Modellordner legen kann, ist dein Lizenznachweis wertlos, und du hast zusätzlich ein Sicherheitsproblem: Modelldateien und Custom Nodes sind ausführbarer Fremddcode auf einem Rechner mit Netzzugang. Kuratierte Modellliste, Schreibrechte nur für die Administration, Erweiterungen nur nach Prüfung.

Datenschutz und die Frage, was das Haus verlässt

Bei rein synthetischen Motiven ist die Datenschutzfrage überschaubar. Heikel wird es an zwei Stellen. Erstens bei Eingangsdaten: Sobald echte Gesichter, Innenaufnahmen, Werksfotos oder Kundenunterlagen als Referenzbilder in den Workflow wandern, verarbeitest du personenbezogene oder vertrauliche Daten und brauchst eine Rechtsgrundlage. Zweitens beim Abfluss: Bei API-Nutzung verlassen Prompt und Referenzbilder das Haus. Wenn im Prompt ein unveröffentlichtes Produkt beschrieben wird, ist das kein Datenschutzthema mehr, sondern ein Geheimnisschutzthema — und oft das schwerwiegendere.

Die saubere Regel lautet: Was nicht hinausgehen darf, wird lokal gerendert. Genau deswegen ist die austauschbare Render-Schicht keine Architekturspielerei, sondern eine Compliance-Funktion. Der Auftrag entscheidet über die Schutzstufe, die Schutzstufe entscheidet über den Renderweg. Warum das für viele Häuser der eigentliche Grund ist, lokal zu bauen, steht in [KI-Bildgenerierung im Unternehmen – Die strategische Alternative zur Cloud](#).

Rollen: vier reichen

Rolle	Verantwortet	Wird oft verwechselt mit
Workflow-Verantwortung	Graphen, Versionen, Freigabe von Änderungen	„der, der sich mit KI auskennt“
Produktion	Aufträge annehmen, Läufe fahren, sichten	der Kreativabteilung als Ganzes
Betrieb	Hardware, Updates, Modellordner, Monitoring	der Person, die ComfyUI installiert hat
Nachweis	Katalog, Metadatenfelder, Kennzeichnung, Lizenzstand	niemandem — und genau da liegt das Problem

Der häufigste Fehler ist, alle vier Rollen auf eine Person zu legen, die das gerne macht und gut kann. Das funktioniert bemerkenswert lange und fällt dann auf einen Schlag aus. Mindestens die Nachweisrolle sollte jemand tragen, der nicht selbst produziert — schon deshalb, weil Menschen ihre eigene Arbeit ungern kontrollieren.

Stolperfallen, die man in freier Wildbahn regelmäßig sieht

- **Siebenunddreißig Workflows, keiner dokumentiert.** Jeder Sonderfall bekommt einen eigenen Graphen, nichts wird zusammengeführt. Gegenmittel: eine Bibliothek, eine

verantwortliche Person, und die Regel, dass ein neuer Workflow eine Begründung braucht, warum er kein Parameter eines bestehenden ist.

- **Der Prompt-Schamane.** Eine Person kennt die Formeln, teilt sie nicht, und der Betrieb hängt an ihrem Kalender. Gegenmittel: Prompts gehören in den Workflow und in den Katalog, nicht in einen Kopf.
- **Die Endlosschleife.** Noch ein Seed, noch ein Synonym, noch ein Anlauf. Rechenzeit ist billig, Arbeitszeit nicht. Gegenmittel: Vorab festlegen, wie viele Anläufe ein Auftrag bekommt, und was „gut genug“ konkret bedeutet.
- **Bilder auf Einzelrechnern.** Das freigegebene Motiv liegt in einem Downloadordner, der Katalog kennt es nicht, und in drei Monaten wird es neu erzeugt. Gegenmittel: Die Ablage-Stufe ist Teil des Workflows, nicht eine Sache, die man hinterher noch macht.
- **Das Modell wird ohne Test getauscht.** Neue Version, bessere Werte, alle Serienbilder sehen plötzlich anders aus. Gegenmittel: der Referenzauftrag aus dem Tipp-Kasten oben.
- **Speicher überschätzt.** Jemand fährt eine 4-Megapixel-Bearbeitung mit voller Präzision auf einer 24-GB-Karte und wundert sich über den Abbruch. Gegenmittel: Grenzen im Produktions-Workflow fest verdrahten, statt auf Selbstbeschränkung zu hoffen.

Und eine, die weniger komisch ist: Wenn dir die KI hartnäckig etwas anderes liefert als beschrieben, liegt das selten an mangelnder Rechenleistung. Die üblichen Ursachen sammelt [Warum die KI den Prompt ignoriert](#); und wenn es speziell um Finger geht, gibt es dafür ebenfalls eine vernünftige Erklärung in [Warum KI keine Hände kann](#).

Häufige Fragen

Brauche ich zwingend eine eigene GPU?

Nein. Über die API kommst du ohne eigene Hardware aus und zahlst je Megapixel. Eigene Hardware lohnt sich, wenn drei Dinge zusammenkommen: gleichmäßige Auslastung, Motive, die das Haus nicht verlassen dürfen, und der Wunsch, offene Modelle und eigene LoRAs zu fahren. Bei gelegentlicher Nutzung ist eine gekaufte Karte meist das teurere Vergnügen, das dann elf Monate im Jahr Staub sammelt.

Welche Flux-Variante darf ich kommerziell verwenden?

Uneingeschränkt: FLUX.2 [klein] 4B unter Apache 2.0. Über die API: [pro] und [flex], die kommerzielle Nutzung ist im Preis enthalten. Mit Zusatzlizenz: FLUX.2 [dev] und [klein] 9B, die im Auslieferungszustand unter der nicht-kommerziellen Lizenz stehen. Black Forest Labs staffelt diese Selbsthosting-Lizenzen nach Bildmenge und Einsatzform, von einer Einsteigerstufe mit rund zehntausend Bildern im Monat bis zur Unternehmensstufe mit ausgehandelten Bedingungen.

Wie viel VRAM brauche ich wirklich?

Für FLUX.2 [klein] 4B rund acht Gigabyte. Für FLUX.2 [dev] als Q4-Quantisat etwa neunzehn, in fp8 etwa zweiunddreißig Gigabyte für das Diffusionsmodell; volle Präzision setzt eine Karte der H100-Klasse voraus. Der Mistral-basierte Text-Encoder kommt in allen Fällen dazu, lässt sich aber quantisiert laden oder auslagern. Eine 24-GB-Karte ist der sinnvolle Einstieg, wenn du bei dev bleiben willst; für klein reicht deutlich weniger.

Wie stelle ich sicher, dass eine Serie wirklich gleich aussieht?

Indem alles außer dem Motivtext festgeschrieben ist: Modellstand, Sampler, Schrittzahl, Guidance, Auflösung, Stilanker und Veredelungsstufe. Der Seed bleibt entweder fest oder folgt einer dokumentierten Liste. Und der Workflow wird während der Serie nicht angefasst — auch nicht „nur kurz“. Wenn du mitten in einer Serie etwas verbesserst, hast du zwei Serien.

Reichen die Metadaten in der PNG-Datei als Nachweis?

Als Arbeitshilfe ja, als Nachweis nein. ComfyUI schreibt den Graphen in die Datei, und du kannst das Bild zurück in die Oberfläche ziehen, um den Workflow wiederherzustellen — sehr praktisch. Sobald die Datei aber konvertiert, komprimiert oder durch ein Optimierungstool geschickt wird, können diese Metadaten verschwinden. Der belastbare Nachweis liegt im Katalog neben dem Asset.

Wie fange ich an, ohne mich zu verzetteln?

Mit einem einzigen Anwendungsfall, der oft vorkommt und wenig Varianten hat. Einen Workflow bauen, ihn versionieren, zwanzig Bilder damit produzieren, vollständig protokollieren und dann erst den zweiten Anwendungsfall anfassen. Wer mit einer Plattformarchitektur für acht Abteilungen beginnt, hat nach einem Quartal ein Architekturdokument und kein Bild.

Was mache ich mit Workflows, wenn ComfyUI aktualisiert wird?

Vorher sichern, nachher den Referenzauftrag laufen lassen, Ergebnis vergleichen. Aktualisierungen bringen regelmäßig neue Knoten und Leistungsverbesserungen mit — die September-Fassung 0.35 etwa einen Compiler für schnellere und speichersparsamere Läufe. Der Preis dafür ist, dass sich Verhalten ändern kann. Updates gehören in ein Wartungsfenster, nicht in den Freitagnachmittag vor der Kampagnenlieferung.

Kann ich die Maschine an Microsoft 365 anbinden?

Ja, und das ist meist der pragmatischste Weg zur Ablage: Auftragserfassung über eine Liste oder ein Formular, Übergabe an die ComfyUI-API, Ablage der freigegebenen Assets in einer Bibliothek mit Pflichtfeldern für Prompt, Seed, Modell, Lizenz und Kennzeichnung. Die Details dazu stehen in [Digital Asset Management für KI-Bilder mit SharePoint Online und Power Platform](#).

Wie kommen die Bilder in Präsentationen und Dokumente?

Über die Ablage, nicht über den Desktop des Erstellers. Wer Bilder direkt beim Schreiben erzeugen will, findet in [Black Forest FLUX.2 in PowerPoint und Copilot](#) den Weg über die Office-Werkzeuge. Für die Beschreibung von Motiven und die Übersetzung eines Briefings in einen brauchbaren Prompt leisten Sprachmodelle inzwischen gute Dienste; wie man die im Unternehmen sauber einbindet, steht unter [Claude im Unternehmen](#).

Lohnt sich ein eigenes LoRA für den Hauslook?

Sobald derselbe Look über Monate und über verschiedene Motive hinweg gebraucht wird, ja. Für drei Bilder im Quartal nicht — da kommst du mit Referenzbildern und einem sauber gebauten Prompt weiter. Die Entscheidungsgrundlagen stehen in [Was ist ein LoRA?](#) und, mit Blick auf die Markenwirkung, in [Markenkonsistente KI-generierte Bilder](#).

Fazit

Eine skalierbare Bild-Maschine ist zu neunzig Prozent Organisation und zu zehn Prozent Modell. Das Modell ist der Teil, über den geredet wird; die Organisation ist der Teil, der entscheidet, ob du in sechs Monaten noch weißt, wie ein Bild entstanden ist. ComfyUI liefert dir dafür die richtige Werkbank: sichtbare Schritte, faltbare Subgraphs, eine ansteuerbare Schnittstelle. Flux liefert die Bildqualität — mit einer Lizenzlandschaft, die du einmal richtig verstehen und dann pro Bild nachweisen musst.

Der Bauplan ist unspektakulär: fünf Schichten, sauber getrennt. Auftrag mit Zweck und Verwendungsort. Workflow als versionierter Quellcode. Render-Schicht austauschbar zwischen lokaler Karte und API. Veredelung als eigene Stufe, die erst nach der Sichtung läuft. Ablage mit einem Datensatz, der das Bild überlebt. Wer diese fünf hat, kann die Menge beliebig hochdrehen. Wer sie nicht hat, wird beim Hochdrehen einfach nur schneller unordentlich.

Fang klein an, aber fang an der richtigen Stelle an: nicht bei der Frage, welche Grafikkarte du kaufst, sondern bei der Frage, was zu jedem Bild dokumentiert sein muss. Die Karte kannst du austauschen. Die Bilder, die du ohne Nachweis erzeugt hast, nicht.

WEITERLESEN — Wo es weitergeht

- › Der Überblick über das gesamte Thema: [KI-Bildgenerierung](#)
- › Architektur und Betriebsmodell gemeinsam durchrechnen: [KI-Bildgenerierung – Beratung](#)
- › Das Team an Workflow, Prompt und Nachweis heranzuführen: [KI-Bildgenerierung – Schulung](#)
- › Reproduzierbarkeit im Detail: [batch size und Reproduzierbarkeit](#) und [CFG-Scale: Guidance in KI-Bildgeneratoren](#)
- › Der Marktüberblick zum Abgleich: [Übersicht der wichtigsten KI-Bildgeneratoren](#)