Das Lagebild befragen (D3) — Schritt für Schritt

Zielgruppe: alle, die mit einem Solution- oder Portfolio-Lagebild arbeiten und Fragen an den aktuellen Stand stellen wollen — ohne Vorkenntnisse in Python oder KI. Dauer: rund 45 Minuten, davon fünf für das Demo-Portfolio. Voraussetzungen: SituationReport ab 0.46.0; für echte Modellantworten ein lokal laufendes Ollama (siehe Ollama installieren) — zum Kennenlernen genügt der eingebaute Provider `mock`, der kein Modell braucht. Am Ende: du stellst Fragen an den Stand, liest Antworten mit Zitaten, erkennst verworfene Antworten und weißt, warum sie verworfen wurden.

Dieses Tutorial gibt es auch als PDF: DE / EN.

1. Worum es geht — und worum nicht

Ein Lagebild hat viele Zahlen und Register: Kennzahlen je ART, Risiken, Abhängigkeiten, Flussprobleme, Solution Areas. Wer eine konkrete Frage hat („Welche Abhängigkeit ist überfällig?"), sucht bisher im Report. „Lagebild befragen" lässt diese Frage ein Sprachmodell beantworten — unter einer Bedingung, die das Werkzeug erzwingt:

> *Jede Aussage der Antwort zitiert den Datenpfad, aus dem sie stammt. Eine Aussage ohne Zitat, ein Zitat auf einen Pfad, den es nicht gibt, oder eine Zahl, die nicht in den zitierten Fakten steht, führt dazu, dass die ganze Antwort verworfen wird.*

Das Modell darf also nur sagen, was es belegen kann. Weiß es nichts, lautet die erlaubte Antwort wörtlich: „Dazu liegt im Lagebild keine Angabe vor."

Was „Lagebild befragen" nicht ist:

- Kein Chat. Jede Frage geht neu gegen den Stand; es gibt kein Gesprächsgedächtnis.

Folgefragen nennen ihren Kontext selbst („… und wer ist bei AD-1 der Nehmer?" wird zu „Wer ist bei der Abhängigkeit AD-1 der Nehmer?").

- Keine Empfehlung. Die Antwort erklärt, was im Stand steht. Sie sagt nicht, was zu

tun ist. Das Urteil bleibt beim Menschen.

- Keine Websuche, keine Fremdquellen. Das Modell bekommt ausschließlich den Stand.
- Keine Personen. Owner- und Melder-Felder gelangen nie zum Modell; Beteiligte sind

Teams, ARTs und Value Streams.

2. In fünf Minuten startklar

2.1 Das Demo-Portfolio erzeugen

Du brauchst Daten. Der Testdaten-Generator baut ein komplettes Demo-Portfolio mit zwei Solutions, sechs ARTs, allen Registern und zwei Snapshots:

```
python -m testdata_generator --scenario portfolio --output demo --seed 42
```

In der Oberfläche: Testdaten-Generator öffnen, unten im Bereich „Demo-Portfolio" auf „Demo-Portfolio erzeugen …" klicken und einen Zielordner wählen.

Danach liegt dort ein vollständiger Datenraum:

```
demo/
■■■ portfolio.json              ← das Portfolio (zwei Solutions)
■■■ solutions/alpha/solution.json   ← Solution Alpha (drei ARTs, alle Register)
■■■ solutions/alpha/registers/      ← risks.json, dependencies.json, flow_problems.json, …
■■■ solutions/beta/solution.json
■■■ snapshots/snapshot_prev.json    ← Stand von vor zwei Wochen
■■■ snapshots/snapshot_now.json     ← aktueller Stand
■■■ README.md                       ← die erzählte Geschichte der Daten
```

In diesem Tutorial fragen wir Solution Alpha (`demo/solutions/alpha/solution.json`). Ihre Geschichte in zwei Sätzen: Die Abhängigkeit AD-1 zwischen ART Alpha-1 und ART Alpha-3 ist blockiert und seit dem 29.09. überfällig; dieselbe Naht hat zum dritten Mal eine Solution Area zum Thema Testumgebung eröffnet.

2.2 Einen Provider wählen

Das Sprachmodell ist austauschbar („Provider"). Für dieses Tutorial:

Provider	Was er tut	Wann sinnvoll
`mock`	Kein Modell. Antwortet auf jede Frage mit dem Unbekannt-Satz.	Ablauf, Dateien und Nachweis kennenlernen — sofort, ohne Installation
`ollama`	Lokales Modell auf deinem Rechner; nichts verlässt ihn.	Echte Antworten. Vorher `ollama pull mistral-nemo`
`claude`	Anthropic-API; Schlüssel nur aus der Umgebungsvariablen `ANTHROPIC_API_KEY`.	Wenn ein externer Dienst erlaubt ist
`export`	Kein Modell. Schreibt den kompletten Prompt in eine Datei (für einen anderen Rechner).	Kapitel 10

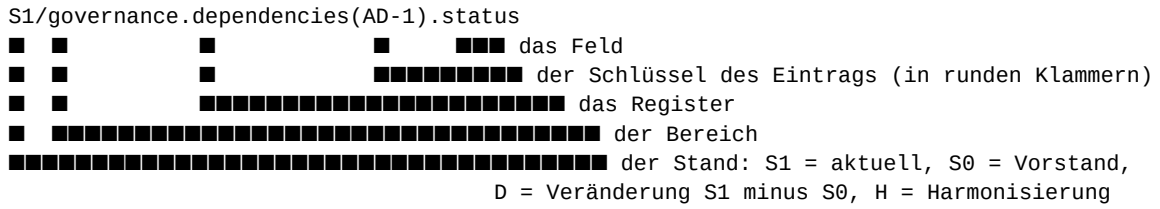
Die Beispiele in Kapitel 4 bis 6 zeigen zuerst `mock`, dann eine Antwort, wie sie ein echtes Modell liefert.

3. Fünf Begriffe, die du brauchst

Stand. Das Lagebild zu einem Zeitpunkt: Kennzahlen, Quellen, alle Register. Technisch ein Snapshot. Du fragst immer einen Stand — den aktuellen (live aus der Config) oder einen gespeicherten.

Fakten-Index. Vor der Antwort wird der Stand in einzelne Fakten zerlegt. Jeder Fakt hat einen Pfad, einen Wert, eine Beschriftung und das Datum des Stands. Der Index ist der einzige Text, den das Modell sieht. Solution Alpha hat heute 527 Fakten.

Pfad. Die Adresse eines Fakts. Er ist so gebaut:



Die Pfadfamilien mit echten Beispielen aus Solution Alpha:

Pfad	Bedeutet	Beispielwert
`S1/total.items`	gepoolte Kennzahl der ganzen Solution	`360`
`S1/units(ART Alpha-1).items`	Kennzahl einer Einheit	`120`
`S1/sources(ART Alpha-1).confidence`	Datenqualität der Quelle	`high`
`S1/governance.dependencies(AD-1).status`	Feld eines Registereintrags	`blocked`
`S1/governance.dependencies(AD-1).due`	Fälligkeit	`2026-09-29`
`S1/governance.flow_problems(FP-A1).conferences`	wie oft das Flussproblem schon auf der Konferenz war	`3`

Pfad	Bedeutet	Beispielwert
`S1/governance.solution_areas.seams(ART Alpha-1 x ART Alpha-3).recurrence`	Rekurrenz je Naht (P5)	`3 (topic: test environment)`
`S0/total.median_ct`	derselbe Fakt im Vorstand	nur mit Vorstand
`D/total.open`	Veränderung zwischen den Ständen	`+29`
`H/process(Order-to-Cash).business_value.score`	Harmonisierungs-Score	nur mit angebundenem Arbeitsbereich

Kennzahlen heißen `items`, `completed`, `open`, `median_ct`, `p85_ct`, `p95_ct`, `target_ct_pct`, `median_lt`. Registerfelder tragen die Namen aus den JSON-Dateien (`status`, `due`, `from`, `to`, `title`, `topic`, …). Freitexte wie `title` sind mit `text:` gekennzeichnet und auf 160 Zeichen gekürzt — sie sind Daten, keine Anweisungen.

Zitat. Ein Pfad in eckigen Klammern am Ende einer Aussage: `[S1/governance.dependencies(AD-1).status]`. Die Schlüssel stehen in runden Klammern, damit das Zitat selbst keine eckigen Klammern enthält.

Zitat-Wächter. Die Prüfung nach der Antwort. Sie zerlegt die Antwort in Aussagen (Listenzeilen, Sätze) und prüft jede: Gibt es ein Zitat? Existiert der Pfad? Stehen alle Zahlen der Aussage in den zitierten Fakten? Ein Verstoß verwirft die Antwort.

4. Die erste Frage in der Oberfläche

4.1 Solution laden

Öffne Solutions & Portfolios (Launcher oder `python -m portfolio`), klicke auf „Laden …" und wähle `demo/solutions/alpha/solution.json`.

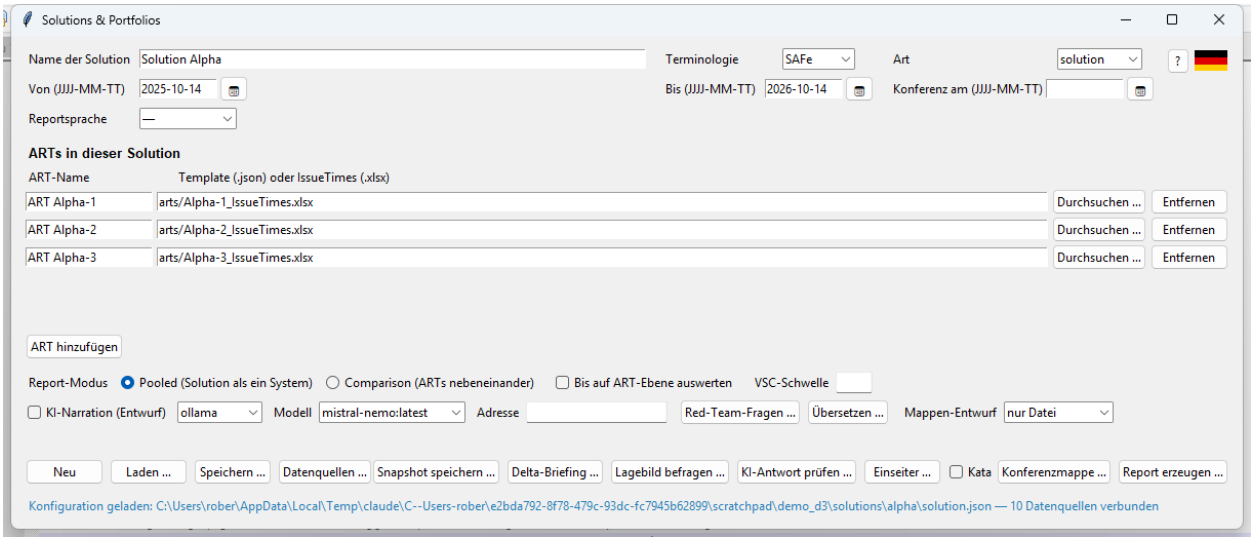


Abb. 1: Das Hauptfenster nach dem Laden. Unten rechts die Knöpfe; „Lagebild befragen …" öffnet den Frage-Dialog. In der KI-Zeile darüber wählst du den Provider.

In der Zeile KI-Narration wählst du den Provider (`mock`, `ollama`, `claude`, `export`) und bei Ollama das Modell. Der Haken „KI-Narration (Entwurf)" spielt für Fragen keine Rolle — der Dialog nutzt den gewählten Provider immer.

4.2 Den Dialog öffnen

Klicke auf „Lagebild befragen …".

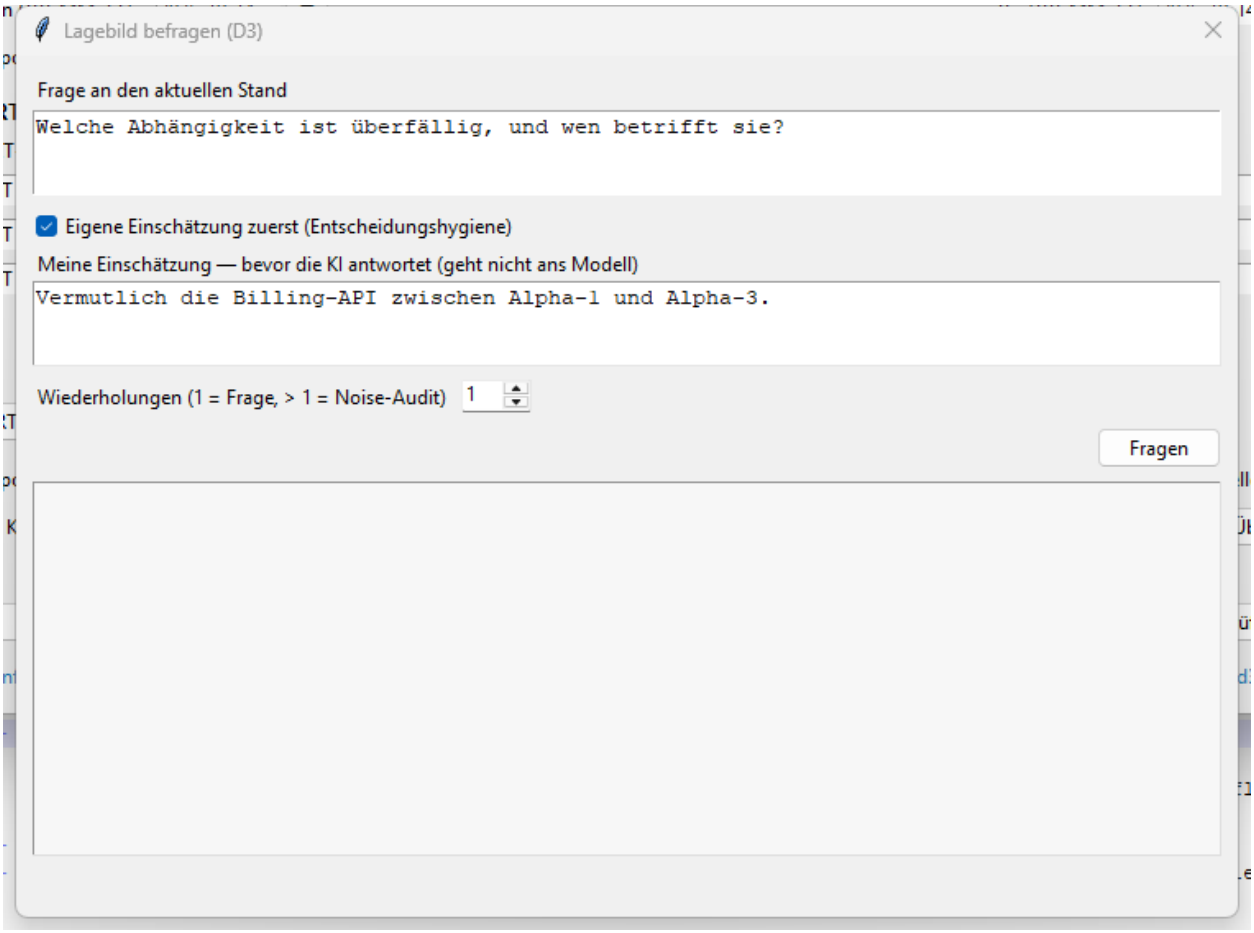


Abb. 2: Der Frage-Dialog. Oben die Frage, darunter die eigene Einschätzung (Kapitel 8), dann Wiederholungen (Kapitel 11) und unten das Antwortfeld.

Die Felder von oben nach unten:

1. Frage an den aktuellen Stand — deine Frage in ganzen Sätzen. Nenne Dinge beim Namen, wie sie im Lagebild heißen (ART-Namen, Register-IDs wie `AD-1`, Begriffe wie „Abhängigkeit", „Flussproblem", „Solution Area").
2. Eigene Einschätzung zuerst — Haken und Feld. Was du selbst glaubst, bevor die KI antwortet. Geht nie ans Modell (Kapitel 8).
3. Wiederholungen — `1` ist eine normale Frage. Größer als 1 startet ein Noise-Audit (Kapitel 11).
4. Fragen — startet den Lauf. Zuerst wirst du nach dem Speicherort der Antwortdatei gefragt (Vorschlag `Solution Alpha_Frage_<Datum>.ask.md`).
5. Antwortfeld — zeigt Banner und Antwort, oder die Verstöße, wenn die Antwort verworfen wurde.

4.3 Fragen und lesen

Tippe ein: Welche Abhängigkeit ist überfällig, und wen betrifft sie? — und klicke auf Fragen. Die Statuszeile zeigt, dass der Lauf läuft. Lokale Modelle brauchen 30 bis 120 Sekunden; `mock` antwortet sofort.

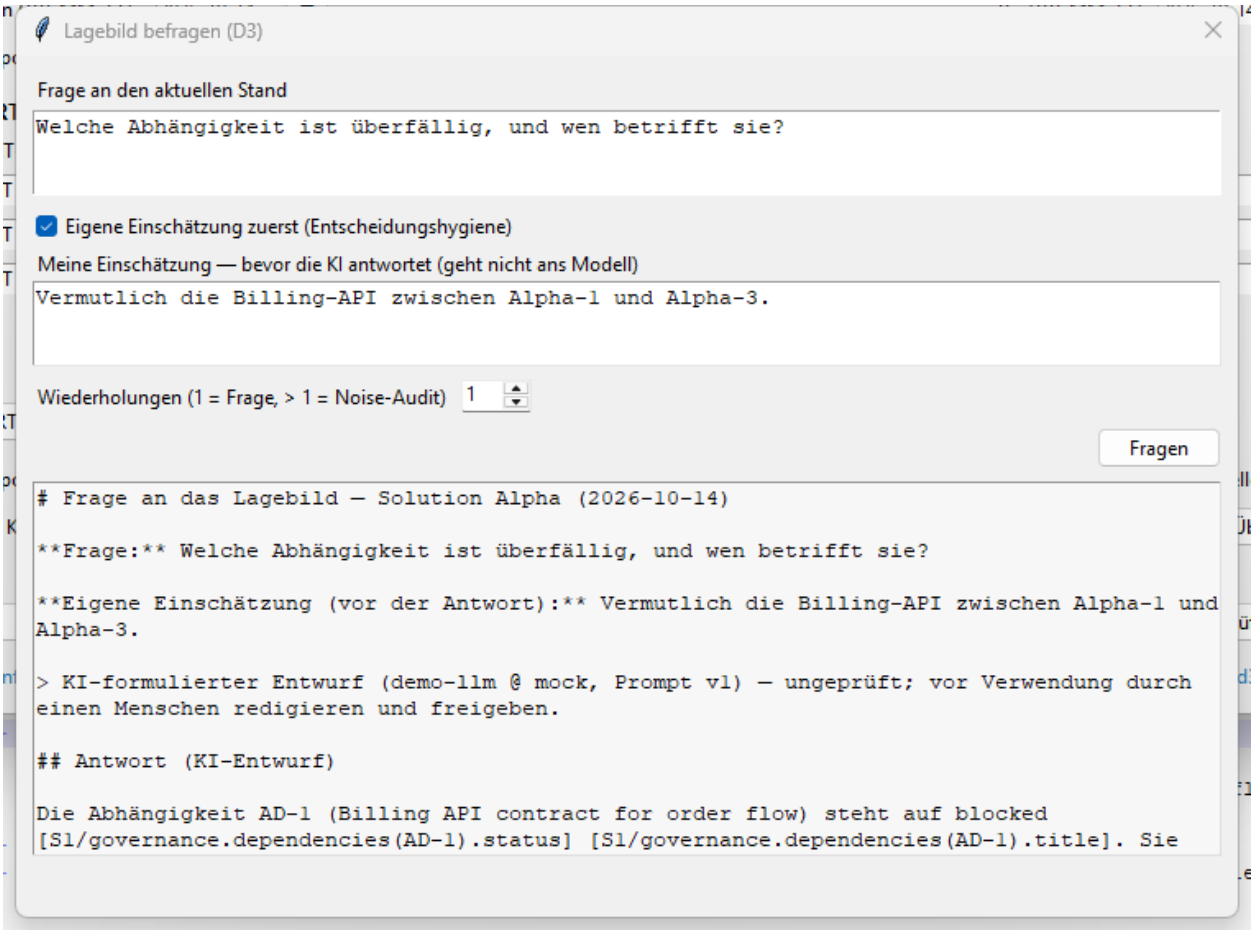


Abb. 3: Die Antwort im Dialog. Ganz oben das Banner, darunter der Antworttext mit Zitaten in eckigen Klammern.

Mit `mock` steht im Antwortfeld:

```
KI-formulierter Entwurf (mock-1 @ mock, Prompt v1) — ungeprüft; vor Verwendung durch
einen Menschen redigieren und freigeben.

Dazu liegt im Lagebild keine Angabe vor.
```

Das ist richtig so: `mock` ist eine Attrappe ohne Modell und darf nichts behaupten. Der Unbekannt-Satz ist die einzige Antwort, die ohne Zitat gültig ist. Mit einem echten Modell steht hier die Antwort aus Kapitel 5.

5. Eine Antwort lesen — Zeile für Zeile

So sieht die gespeicherte Datei `Solution Alpha_Frage_2026-10-14.ask.md` aus. Der Antworttext ist in diesem Tutorial von Hand geschrieben, damit das Beispiel gut lesbar ist; er wurde aber wie eine Modellantwort durch den Zitat-Wächter geprüft und bestanden. Ein lokales Modell antwortet ähnlich, nur mit anderen Worten.

```
# Frage an das Lagebild — Solution Alpha (2026-10-14)

**Frage:** Welche Abhängigkeit ist überfällig, und wen betrifft sie?

**Eigene Einschätzung (vor der Antwort):** Vermutlich die Billing-API zwischen Alpha-1 und Alpha-3.

> KI-formulierter Entwurf (demo-llm @ mock, Prompt v1) — ungeprüft; vor Verwendung durch
> einen Menschen redigieren und freigeben.

## Antwort (KI-Entwurf)

Die Abhängigkeit AD-1 (Billing API contract for order flow) steht auf blocked
[S1/governance.dependencies(AD-1).status] [S1/governance.dependencies(AD-1).title].
Sie war am 2026-09-29 fällig [S1/governance.dependencies(AD-1).due], der Stand ist vom
2026-10-14 [S1/total.items]. Betroffen sind ART Alpha-1 als Geber und ART Alpha-3 als
Nehmer [S1/governance.dependencies(AD-1).from] [S1/governance.dependencies(AD-1).to].
```

```
Dieselbe Naht hat zum dritten Mal eine Solution Area zum Thema test environment
eröffnet [S1/governance.solution_areas.seams(ART Alpha-1 x ART Alpha-3).recurrence].

---
Stand: S1 = 2026-10-14 · Fakten im Index: 527 · Index-Prüfsumme (SHA-256): 43aa08d6… ·
Modell: demo-llm @ mock · own-view hash: 56dc697e80339a25
```

Was die Teile bedeuten:

Zeile	Bedeutung
Überschrift	Solution und Stand, den du gefragt hast
Frage	deine Frage, bereinigt (Steuerzeichen entfernt, maximal 500 Zeichen)
Eigene Einschätzung	dein Text von vor der Antwort — oder „— (übersprungen)"
Banner (>)	Pflichtkennzeichnung: Modell, Betriebsklasse (`local`, `remote`, `mock`), Prompt-Version. Bleibt stehen, bis ein Mensch den Text redigiert und freigibt
Antwort	jede Aussage endet mit einem oder mehreren Zitaten
Fußzeile	Stand (S1, ggf. S0), Zahl der Fakten, Prüfsumme des Index (damit ist nachvollziehbar, welchen Stand das Modell sah), Modell, Hash deiner Einschätzung

Ein Zitat nachschlagen

Nimm `[S1/governance.dependencies(AD-1).due]`. Das ist: aktueller Stand, Register `dependencies`, Eintrag `AD-1`, Feld `due`. Im Datenraum findest du den Eintrag in `demo/solutions/alpha/registers/dependencies.json`:

```
{"id": "AD-1", "title": "Billing API contract for order flow",
 "from": "ART Alpha-1", "to": "ART Alpha-3", "status": "blocked", "due": "2026-09-29", ...}
```

Die Zahl `2026-09-29` in der Antwort steht genau dort. Das ist der ganze Trick: Jede Aussage hat eine Adresse, und die Adresse führt zu einer Datei, die du öffnen kannst.

Den kompletten Fakten-Index mit allen 527 Pfaden bekommst du, wenn du den Provider `export` wählst (Kapitel 10): Das Prompt-Bündel enthält ihn vollständig.

Warum zwei Zitate hinter einer Aussage?

„Die Abhängigkeit AD-1 (Billing API contract for order flow) steht auf blocked" nennt zwei Fakten — den Titel und den Status. Beide werden zitiert. Der Wächter verlangt mindestens ein Zitat je Aussage; mehr sind erlaubt und oft nötig, damit jede Zahl und jeder Wert belegt ist.

6. Dieselbe Frage auf der Kommandozeile

```
python -m portfolio demo/solutions/alpha/solution.json \
    --ask "Welche Abhängigkeit ist überfällig, und wen betrifft sie?" \
    --narrate ollama
```

Ausgabe:

```
Question to stand 2026-10-14 via 'ollama' (527 facts) ...
Answer written (draft for human review): demo/solutions/alpha/solution_2026-10-14.ask.md
```

Die Datei liegt neben der Config; daneben entsteht `llm_audit.jsonl` (Kapitel 12).

Wert), `claude`, `mock`, `export`		
`tral-nemo`		

a (dann Klasse `remote`)		
	Antwortsprache (Standard `de`)	
twort (Kapitel 8)		
en statt den aktuellen (Kapitel 9)		
`S0/…` und `D/…` (Kapitel 9)		
	registers`	welche Fak
n wird		
gen (Kapitel 11)		

Rückgabewerte: `0` Antwort geschrieben, `3` Antwort verworfen (nichts geschrieben).

7. Wenn eine Antwort verworfen wird

Das wird dir passieren, vor allem mit kleinen lokalen Modellen. Es ist kein Fehler des Werkzeugs — es ist das Werkzeug, das arbeitet. Zwei echte Meldungen:

Zahlen-Wächter (eine Zahl, die nirgends im Stand vorkommt):

```
REJECTED: Narration discarded: the model invented number(s) 45 that do not occur in the
briefing. The LLM writes, it does not calculate — re-run, switch the model, or use the
deterministic briefing as is.
```

Zitat-Wächter (eine Aussage ohne Zitat):

```
REJECTED: Answer discarded: every statement must cite a fact path of the stand and every
number must come from the cited facts. 1 violation(s): uncited: Die Abhängigkeit AD-1
ist blockiert und sollte eskaliert werden.
```

Beachte das zweite Beispiel: Die Aussage war inhaltlich sogar richtig (AD-1 ist blockiert) — aber sie trug kein Zitat und enthielt eine Empfehlung („sollte eskaliert werden"). Beides darf nicht durch.

Die drei Verstoßarten:

Meldung	Bedeutet
`uncited: …`	eine Aussage ohne `[Pfad]`
`unknown path: …`	ein Zitat auf einen Pfad, den der Index nicht hat (meist ein Tippfehler des Modells)
`number not in cited facts: …`	eine Zahl, die zwar irgendwo im Stand vorkommt, aber nicht in den Fakten, die die Aussage zitiert

Was du tun kannst — in dieser Reihenfolge:

1. Noch einmal fragen. Modelle streuen; der zweite Lauf ist oft sauber.
2. Die Frage schärfen. „Was ist mit den Abhängigkeiten?" lädt zum Erzählen ein;

„Welche Abhängigkeiten haben den Status blocked?" führt zu Zitaten.

3. Größeres Modell. Kleine Modelle vergessen die Zitierregel eher.
4. Beim deterministischen Report bleiben. Die Zahl steht im Report, auch ohne KI.

Im Dialog erscheinen die Verstöße im Antwortfeld, die Statuszeile sagt „Antwort verworfen". Auf der Kommandozeile ist der Rückgabewert `3`. In beiden Fällen wird keine Antwortdatei geschrieben, aber der Nachweis (Kapitel 12) hält den Lauf mit `guard_passed: false` fest.

8. Eigene Einschätzung zuerst

Wer die Antwort einer KI liest, bevor er selbst nachgedacht hat, übernimmt ihre Sicht, ohne es zu merken. Die Entscheidungsforschung nennt das Verankerung; die KI-Denkschrift macht daraus Prinzip 7: erst unabhängig urteilen, dann die KI fragen.

Darum steht im Dialog das Feld „Meine Einschätzung — bevor die KI antwortet":

- Der Haken ist standardmäßig gesetzt. Ihn zu entfernen ist ein bewusster Klick; die

Einstellung merkt sich dieser Rechner (nicht die Solution-Config).

- Was du dort schreibst, geht nie ans Modell. Es steht in der Antwortdatei über der

KI-Antwort — so siehst du beide nebeneinander.

- Der Nachweis hält nur einen Hash deiner Einschätzung fest, nicht den Text.

Auf der Kommandozeile: `--own-view "Vermutlich die Billing-API zwischen Alpha-1 und Alpha-3."`

Ein guter Gebrauch: Vor der Value-Stream-Konferenz notiert jedes Mitglied des Stabs seine Einschätzung zu derselben Frage. Dann fragt einer die KI. Wo die Einschätzungen auseinanderliegen, ist das Gespräch — nicht die KI-Antwort.

9. Die Veränderung befragen

Bisher hast du einen Stand gefragt. Mit einem Vorstand kommen zwei Pfadfamilien dazu: `S0/…` (derselbe Fakt im Vorstand) und `D/…` (die Veränderung, S1 minus S0).

```
python -m portfolio demo/portfolio.json \
    --ask "Was hat sich seit dem letzten Stand verändert?" \
    --ask-snapshot demo/snapshots/snapshot_now.json \
    --since-snapshot demo/snapshots/snapshot_prev.json \
    --narrate ollama
```

Hier wird das ganze Portfolio gefragt, mit gespeichertem Stand (`--ask-snapshot`) und Vorstand (`--since-snapshot`). Der Index wächst auf 1882 Fakten; darunter stehen jetzt Zeilen wie diese:

```
- [D/total.items] change of items = +83 (as of 2026-09-30 -> 2026-10-14; Demo Portfolio)
- [D/total.completed] change of completed = +54 (as of 2026-09-30 -> 2026-10-14; Demo Portfolio)
- [D/total.open] change of open = +29 (as of 2026-09-30 -> 2026-10-14; Demo Portfolio)
- [D/total.median_ct] change of median_ct = -0.1 (as of 2026-09-30 -> 2026-10-14; Demo Portfolio)
- [D/units(Solution Alpha).items] change of items = +45 (as of 2026-09-30 -> 2026-10-14; Demo Portfolio)
```

Eine Antwort darauf könnte lauten: „Im Portfolio sind 83 Items dazugekommen [D/total.items], davon 45 bei Solution Alpha [D/units(Solution Alpha).items]." Die Fußzeile der Datei nennt dann beide Stände: `S1 = 2026-10-14, S0 = 2026-09-30`.

Zwei Hinweise:

- Der Dialog in der Oberfläche fragt immer den aktuellen Stand ohne Vorstand. Für

Fragen zur Veränderung nimm die Kommandozeile.

- Gespeicherte Stände erzeugst du mit „Snapshot speichern …" oder `--snapshot DATEI`.

Wer den Stand speichert, bevor er fragt, kann später nachvollziehen, was genau das Modell gesehen hat — die Index-Prüfsumme in der Fußzeile ist dann reproduzierbar.

10. Ohne Modell auf diesem Rechner: Prompt exportieren und Antwort zurückprüfen

Darf auf deinem Rechner kein Modell laufen, oder willst du ein anderes Modell anderswo nutzen? Dann wähle den Provider `export`:

```
python -m portfolio demo/solutions/alpha/solution.json \
    --ask "Welche Abhängigkeit ist überfällig?" --narrate export
```

Statt einer Antwort entsteht `prompt_d3_question_<Zeitstempel>.md` neben der Ausgabe (und der Text liegt in der Zwischenablage). Das Bündel enthält die Anleitung, den Systemprompt und als „Eingabe" den vollständigen Fakten-Index — genau das, was ein Modell hier gesehen hätte. Führe es auf dem anderen Rechner aus, speichere die Antwort als Datei und bring sie zurück:

```
python -m llm check antwort.md --model "gpt-irgendwas"
```

`llm check` findet das Bündel neben der Antwort, lässt dieselben Wächter laufen — Zahlen und, weil es eine D3-Frage ist, auch die Zitate — und schreibt bestanden `antwort.md.checked.md` mit dem Banner „Extern erzeugter KI-Entwurf". Abgelehnt wird nichts geschrieben. Der Nachweis verknüpft Hin- und Rückweg über dieselbe Eingabe-Prüfsumme. In der Oberfläche: Knopf „KI-Antwort prüfen …".

11. Wie laut ist das Modell? Das Noise-Audit

Ein Sprachmodell antwortet auf dieselbe Frage nicht zweimal gleich. Bevor du einer Antwort traust, lohnt die Gegenprobe: dieselbe Frage mehrfach, leicht umformuliert, gegen denselben Stand.

Im Dialog: Wiederholungen auf `5` stellen und Fragen klicken. Auf der Kommandozeile: `--noise 5`.

Lauf 1 ist deine Frage unverändert, dann folgen feste Umformulierungen („Kurz gefasst: …", „Für die Value-Stream-Konferenz: …", …). Das Protokoll `….noise.md` beginnt so (hier mit `mock`, darum überall „keine Angabe"):

```
# Noise-Audit — Solution Alpha (2026-10-14)

**Frage:** Wie steht es um die Abhängigkeiten?

Die Streuung ist ein Befund über das Modell und die Frage, kein Urteil über die Lage.
Was daraus folgt, entscheidet der Mensch.

## Läufe (3)

| Lauf | Variante | Modell | Zitate | Zahlen | Ergebnis |
|---|---|---|---|---|---|
| 1 | Wie steht es um die Abhängigkeiten? | mock-1 | 0 | - | keine Angabe |
| 2 | Kurz gefasst: Wie steht es um die Abhängigkeiten? | mock-1 | 0 | - | keine Angabe |
| 3 | Für die Value-Stream-Konferenz: Wie steht es um die Abhängigkeiten? | mock-1 | 0 | - | keine Angabe |

## Zitier-Konsistenz (mittlere paarweise Jaccard-Ähnlichkeit)

**100.0 / 100**
```

Die Maße lesen:

Maß	Bedeutet	Lesehilfe
Zitier-Konsistenz	wie ähnlich die zitierten Pfadmengen der Antworten sind, 0–100	100 = alle zitieren dasselbe; unter 50 zitiert jede Antwort etwas anderes
Kern	Pfade, die die Mehrheit der gültigen Antworten zitiert	das, worauf sich das Modell verlässlich stützt
Streuung	Pfade, die nur eine Antwort zitiert — mit der Aussage dazu	was eine einzelne Umformulierung hervorgeholt hat
Zahlen-Abweichungen	derselbe Pfad, verschiedene Zahlen in verschiedenen Antworten	sollte leer sein; sonst liest das Modell den Index unsauber
keine Angabe / verworfen	Anteile der Läufe	hohe Quoten heißen: Frage oder Modell passen nicht zum Stand

Das Protokoll bewertet nicht. Ob 70 „gut genug" ist, hängt davon ab, was du mit der Antwort vorhast. Eine verworfene Antwort wird nicht wiederholt — dass das Modell bei einer Umformulierung aus dem Contract fällt, gehört zum Befund.

Mit `--views stab.txt` (eine Einschätzung je Absatz) stehen die Einschätzungen des Stabs nummeriert unter dem KI-Kern — ohne Bewertung durch das Werkzeug.

12. Der Nachweis

Jeder Lauf, bestanden oder verworfen, hängt eine Zeile an `llm_audit.jsonl` neben der Ausgabe:

```
{"ts": "2026-10-14T14:45:48", "purpose": "d3_question", "provider": "demo",
 "model": "demo-llm", "deployment_class": "mock", "prompt_version": "v1",
 "input_sha256": "6c565e86…", "output_sha256": "b49cb706…", "duration_s": 0.0,
 "guard_passed": true}
```

Es stehen nur Hashes darin — weder die Frage noch die Antwort noch deine Einschätzung im Klartext. Mit `input_sha256` lässt sich belegen, welcher Contract (Frage plus Index) zum Modell ging; `guard_passed` sagt, ob die Antwort durchkam. Das ist der Betreiber-Nachweis, den die KI-Verordnung von dir verlangt — ohne dass du etwas dafür tun musst.

13. Fehlerbilder

Beobachtung	Ursache und Abhilfe
„Dazu liegt im Lagebild keine Angabe vor." bei jeder Frage	Provider `mock` gewählt — er kennt kein Modell. Auf `ollama` wechseln. Oder: Die Frage betrifft etwas, das kein Register enthält (z. B. Personen, Budgets)
`REJECTED: … uncited: …`	Kapitel 7. Noch einmal fragen, Frage schärfen, größeres Modell
`REJECTED: … invented number(s) …`	Das Modell hat gerechnet oder geschätzt. Dasselbe Vorgehen
`Could not reach Ollama`	Ollama läuft nicht. Starten und auf das Symbol im Infobereich warten
Der Lauf dauert „ewig"	Lokale Modelle rechnen 30–120 Sekunden; der Index ist mit 500+ Fakten groß. `--ask-scope metrics` oder `registers` verkleinert ihn
Die Antwort nennt eine Person	Kann nicht aus dem Index kommen — er enthält keine Personenfelder. Das Modell hat sie erfunden; die Antwort ist zu verwerfen, auch wenn der Wächter sie durchließ, weil keine Zahl verletzt wurde
Antwort auf Englisch	`--llm-lang de` setzen; im Dialog folgt die Sprache der Oberfläche
Der Dialog fehlt in der Oberfläche	Version prüfen: „Lagebild befragen …" gibt es seit 0.46.0, „Wiederholungen" seit 0.47.0

14. Was du mitnehmen solltest

- Du fragst einen Stand, nicht ein Modell. Das Modell formuliert; die Fakten kommen aus dem Lagebild, und jede Aussage trägt ihre Adresse.
- Eine verworfene Antwort ist ein Erfolg des Werkzeugs, kein Defekt. Lieber keine Antwort als eine mit erfundener Zahl.
- Erst selbst urteilen, dann fragen. Das Einschätzungsfeld ist kein Formular, es ist deine Verankerung gegen die der KI.
- Streuung messen, bevor du vertraust. Fünf Wiederholungen kosten zwei Minuten und zeigen, wie fest die Antwort steht.
- Nichts verlässt dein Haus ohne Freigabe. Das Banner bleibt, bis ein Mensch es bewusst entfernt.

Anhang: Fragen, die gut funktionieren

Fragen, die auf Register und Felder zielen, bekommen Zitate; Fragen nach Gründen und Ratschlägen bekommen den Unbekannt-Satz oder werden verworfen.

Gut	Weniger gut
Welche Abhängigkeiten haben den Status blocked?	Warum sind die Abhängigkeiten blockiert?
Wie viele Items sind bei ART Alpha-3 offen?	Ist Alpha-3 zu langsam?
Welches Flussproblem war am häufigsten auf der Konferenz?	Was sollen wir mit dem Flussproblem tun?
An welcher Naht gibt es wiederkehrende Solution Areas?	Sollten wir Alpha-1 und Alpha-3 zusammenlegen?
Wie hat sich der Median der Cycle Time seit dem Vorstand verändert?	Wird es besser?