WORKBOOK 1
OPENCLAW
FOR DUMMIES

DEIN LEITFADEN FÜR DEN ERSTEN AGENTEN

1. PLANEN
Ziel definieren
Use Case &
Strategie

2. BAUEN
Agenten erstellen
Tools integrieren
Prompting & Logik

3. INTEGRIEREN
Daten anbinden
Wissen verknüpfen
Schnittstellen nutzen

4. TESTEN & VERBESSERN
Evaluieren
Iterieren
Optimieren

5. BEREITSTELLEN
Deployen
Überwachen
Skalieren

WAS DICH ERWARTET:

- ✔ Schritt-für-Schritt-Anleitungen
- ✔ Praxisnahe Beispiele
- ✔ Best Practices
- ✔ Tipps & Tricks
- ✔ Von Null auf Openclaw

Basis

0 VON NULL AUF OPENCLAW
STARTE DEINEN ERSTEN AGENTEN – HEUTE.

KI-AGENTEN · TOOLS · INTEGRATION · SICHERHEIT · WACHSTUM

OpenClaw Setup-Anleitung

— Basis —

Workbook 1 – OpenClaw for Dummies

Kompakt-Setup für VPS und Mac mini

Prof. Dr. Ulf Pillkahn

Operation-zukunft

Stand: April 2026 · Basisversion 1.0

Verifiziert für: OpenClaw (npm), Node.js 24 LTS,
Ubuntu 24.04 LTS, macOS Sequoia 15.x

Hinweise zur Nutzung dieser Anleitung

Die vorliegende Kurzanleitung – die Basisversion des Workbooks 1 – dokumentiert in stark verdichteter Form einen Einrichtungsweg, den der Autor in seinem eigenen KI-Labor erprobt hat. Sie führt von einem unkonfigurierten System zu einer minimal lauffähigen OpenClaw-Installation. Wer Hintergründe, Sicherheitshärtung im Detail, produktiven Dauerbetrieb mit Reverse Proxy und TLS, Telegram-Anbindung sowie weitergehende Anwendungsfälle benötigt, findet diese Themen in der vollständigen Ausgabe.

Diese Anleitung ist ausdrücklich keine Beratung im rechtlichen oder professionellen Sinne. Sie beschreibt einen Ansatz, der für den Autor und einen begrenzten Kreis von Erprobungspartnern verlässlich funktioniert – sie erhebt aber nicht den Anspruch, jede Hardware-Kombination, jede Netzwerktopologie oder jeden organisatorischen Kontext abzudecken. Wer die hier beschriebenen Schritte produktiv einsetzt, trägt die Verantwortung für die Anwendbarkeit auf das eigene System, für die Sicherheit der eigenen Daten und für die rechtskonforme Nutzung der eingesetzten Dienste selbst.

Hinzu kommt die Eigenheit dieses Feldes: Die Werkzeuglandschaft rund um KI-Assistenten, Modellanbieter und Sprachmodelle verändert sich derzeit in einem Tempo, das gedruckte Anleitungen strukturell überfordert. Befehle, die heute funktionieren, können morgen veraltet sein; Anbieter ändern URLs, Preisstrukturen oder API-Schnittstellen oft mit kurzer Vorlaufzeit. Die hier dokumentierten Stände sind daher Momentaufnahmen, keine Dauerwahrheiten.

Der Autor übernimmt keine Haftung für Schäden, die aus der Anwendung dieser Anleitung entstehen, gleich ob technischer, finanzieller oder anderer Natur. Hinweise auf Fehler, Aktualisierungen und Verbesserungsvorschläge sind willkommen.

Geltungsbereich

Diese Basisversion wurde verifiziert für OpenClaw in der zum Stand April 2026 aktuellen npm-Version, Node.js 24 LTS, Ubuntu 24.04 LTS sowie macOS Sequoia 15.x auf Apple Silicon (M1, M2, M4).

Vorwort zur Basisversion

Diese Kurzanleitung führt durch die wesentlichen Schritte zur Inbetriebnahme von OpenClaw – einem persönlichen, lokal ausführbaren KI-Assistenten – auf zwei typischen Zielsystemen: einem virtuellen Server (VPS) bei einem Hosting-Anbieter oder einem Mac mini im eigenen Büro oder Labor. Beide Pfade führen zu einem produktiv nutzbaren System; die Wahl ist konzeptioneller Natur und wird im ersten Abschnitt kurz erläutert.

Wer mehr Tiefe sucht – ausführliche Begründungen, Sicherheitshärtung Schritt für Schritt, Reverse Proxy mit TLS, Telegram-Bot, E-Mail-Integration, Backup-Strategien, Wartung und Fehlerdiagnose – findet diese Themen in der vollständigen Ausgabe des Workbooks 1. Diese Basisversion beschränkt sich auf das, was zur ersten lauffähigen Installation tatsächlich nötig ist.

Jeder Abschnitt schließt mit einer kurzen Verifikation – einer Prüfung, die sichtbar macht, ob der vorangegangene Schritt erfolgreich war. Diese Disziplin verhindert das klassische Problem, dass Fehler erst mehrere Schritte später kaskadenartig zutage treten.

1. Voraussetzungen

1.1 VPS oder Mac mini?

Ein VPS – ein virtueller Server bei einem Anbieter wie Hetzner oder Contabo – ist die richtige Wahl, wenn der Assistent permanent erreichbar sein soll, etwa als Anlaufstelle für Nachrichten zu jeder Tageszeit. Der VPS läuft, während der eigene Rechner schläft. Der Preis dafür ist das monatliche Abonnement (etwa fünf bis fünfzehn Euro) und die Verantwortung für eine sicher konfigurierte Linux-Instanz.

Ein Mac mini im eigenen Netz ist die richtige Wahl, wenn Datenschutz, lokale Verfügbarkeit oder die Integration in bestehende Apple-Geräte im Vordergrund stehen. Sensible Daten verlassen das eigene Netz nicht; die Performance eines Apple-Silicon-Geräts übertrifft die meisten Mittelklasse-VPS deutlich.

Pragmatische Empfehlung: Wer OpenClaw zunächst evaluieren möchte, beginnt auf dem Mac. Wer ein produktives, dauerhaft erreichbares System will, wählt den VPS.

1.2 Hardware – Minimum

Komponente	Empfehlung
VPS	4 GB RAM, 2 vCPU, 40 GB SSD (z. B. Hetzner CX22)
Mac mini	M1 mit 8 GB genügt; M2/M4 mit 16 GB für längeren Komfort
Betriebssystem VPS	Ubuntu 24.04 LTS
Betriebssystem Mac	macOS Sequoia 15.x oder Sonoma 14.x

1.3 API-Schlüssel besorgen

OpenClaw selbst denkt nicht – das eigentliche Sprachmodell läuft beim Anbieter. Vor der Installation muss ein API-Schlüssel bereitliegen. Die naheliegende Wahl im wissenschaftlichen Kontext ist Anthropic (Claude); OpenAI und Google funktionieren ebenfalls.

Bei Anthropic: *platform.anthropic.com* aufrufen, Konto anlegen, Telefonnummer verifizieren, unter Settings → Plans & Billing eine Zahlungsmethode hinterlegen und – wichtig – ein monatliches Ausgabenlimit setzen. Anschließend in der linken Seitenleiste API Keys wählen, einen neuen Schlüssel mit aussagekräftigem Namen (etwa „openclaw-prod“) erzeugen und sofort in einen Passwortmanager kopieren. Der Schlüssel beginnt mit `sk-ant-api03-` und wird **nur einmalig** angezeigt.

Sicherheitshinweis

API-Schlüssel niemals in ein Git-Repository einchecken – auch nicht in ein privates. GitHub durchsucht öffentliche Repositories systematisch nach API-Schlüsseln; Schlüssel sind dort typischerweise binnen

Minuten kompromittiert. Versehentlich veröffentlichte Schlüssel sofort widerrufen und neue erzeugen.

1.4 Kostenrahmen

Bei moderater persönlicher Nutzung sind etwa 5–30 € pro Monat für die API realistisch, hinzu kommen 5–15 € beim VPS oder marginale Stromkosten beim Mac mini. Wer die Limits beim Anbieter explizit setzt, schützt sich vor unerwarteten Rechnungen, etwa durch eine versehentlich in einer Schleife laufende Konversation.

2. System vorbereiten

2.1 VPS – minimaler Pfad (Ubuntu 24.04)

Beim Hosting-Anbieter (empfohlen: Hetzner Cloud, CX22 in Falkenstein oder Nürnberg) einen Server mit Ubuntu 24.04 LTS bestellen. Beim Anlegen den eigenen SSH-Schlüssel hinterlegen, statt ein Passwort vergeben zu lassen – das ist der entscheidende Schritt für die spätere Sicherheit.

Wer noch kein SSH-Schlüsselpaar hat, erzeugt es lokal mit `ssh-keygen -t ed25519 -C "name@laptop"`. Der öffentliche Schlüssel liegt anschließend in `~/.ssh/id_ed25519.pub` und kann ins Eingabefeld des Anbieters kopiert werden. Der private Schlüssel verlässt den eigenen Rechner niemals.

Erste Anmeldung am Server (IP-Adresse anpassen):

```
ssh root@203.0.113.42

# Sofort System aktualisieren und Basis-Werkzeuge installieren:
apt update && apt upgrade -y
apt install -y curl wget git ufw ca-certificates gnupg

# Unprivilegierten Benutzer anlegen (statt direkt als root zu arbeiten):
adduser openclaw
usermod -aG sudo openclaw

# SSH-Schlüssel des root-Kontos kopieren, damit der neue Benutzer
# sich ohne Passwort anmelden kann:
mkdir -p /home/openclaw/.ssh
cp /root/.ssh/authorized_keys /home/openclaw/.ssh/authorized_keys
chown -R openclaw:openclaw /home/openclaw/.ssh
chmod 700 /home/openclaw/.ssh
chmod 600 /home/openclaw/.ssh/authorized_keys

# Minimale Firewall einrichten:
ufw default deny incoming
```

```
ufw default allow outgoing
ufw allow ssh
ufw enable
```

Ab jetzt sollte ausschließlich der Benutzer openclaw verwendet werden. Eine umfassendere Härtung – Passwortanmeldung deaktivieren, root-Login sperren, fail2ban einrichten – ist in der Vollversion beschrieben und für den produktiven Dauerbetrieb dringend empfohlen.

Verifikation
In einem zweiten Terminal ssh openclaw@<IP-Adresse> ausführen. Die Anmeldung muss ohne Passwortabfrage gelingen, die Eingabeaufforderung sollte openclaw@hostname:~$ lauten. Erst dann die alte root-Sitzung schließen.

2.2 Mac mini – minimaler Pfad

Voraussetzung ist ein aktuelles macOS, idealerweise Sequoia 15.x. Im Apple-Menü unter „Über diesen Mac“ prüfen; gegebenenfalls vorher ein Systemupdate einspielen.

Im Terminal (Spotlight: Cmd+Leertaste, „Terminal“) zunächst die Apple Command Line Tools installieren: `xcode-select --install`. Es erscheint ein Dialog – mit „Installieren“ bestätigen, Lizenz akzeptieren, abwarten.

Anschließend Homebrew installieren – den De-facto-Standardpaketmanager für macOS:

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# Auf Apple Silicon: Homebrew zum PATH hinzufügen
echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> ~/.zprofile
eval "$(/opt/homebrew/bin/brew shellenv)"
```

Für Dauerbetrieb in den Systemeinstellungen unter „Energie“ drei Punkte setzen: „Computer im Ruhezustand bei Inaktivität“ auf „Nie“, „Bei Netzwerkzugriff aufwecken“ aktivieren, „Nach einem Stromausfall automatisch starten“ aktivieren.

Verifikation
Im Terminal brew --version eingeben. Die Ausgabe muss eine Versionsnummer beginnend mit „Homebrew“ enthalten. Wird der Befehl nicht gefunden, ein neues Terminalfenster öffnen oder die eval-Zeile erneut ausführen.

3. Node.js installieren

OpenClaw ist eine Node.js-Anwendung. Benötigt wird Version 22.14 oder höher, idealerweise Version 24.

3.1 VPS – Node.js 24 über NodeSource

Ubuntu liefert in seinen Paketquellen nur eine ältere Node-Version. Daher wird die offizielle NodeSource-Paketquelle eingebunden:

```
# GPG-Schlüssel der Paketquelle hinzufügen
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://deb.nodesource.com/gpgkey/nodesource-repo.gpg.key | \
  sudo gpg --dearmor -o /etc/apt/keyrings/nodesource.gpg

# Paketquelle für Node.js 24 eintragen
NODE_MAJOR=24
echo "deb [signed-by=/etc/apt/keyrings/nodesource.gpg] \
https://deb.nodesource.com/node_$NODE_MAJOR.x nodistro main" | \
  sudo tee /etc/apt/sources.list.d/nodesource.list

# Installation
sudo apt update
sudo apt install -y nodejs

# Verifikation
node --version   # muss v24.x.y ausgeben
npm --version    # 11.x oder neuer
```

3.2 Mac mini – Node.js über Homebrew

```
brew install node

# Verifikation
node --version
npm --version
```

Sollte `node --version` eine Version unter 22.14 zurückgeben, hilft `brew update && brew upgrade node` oder im Zweifel die explizite Version: `brew install node@24`.

Verifikation
Auf beiden Pfaden: node --version muss eine Version 22.14 oder höher (idealerweise 24.x) ausgeben, npm --version eine Version 10 oder höher. Nur dann ist die Voraussetzung für die OpenClaw-Installation erfüllt.

4. OpenClaw installieren

4.1 Globale Installation

Die eigentliche Installation ist – mit den vorbereiteten Voraussetzungen – ein einziger Befehl. Auf beiden Pfaden identisch:

```
npm install -g openclaw@latest
```

Auf dem VPS kann ein „EACCES“-Fehler auftreten, wenn npm keine Schreibrechte auf das globale Verzeichnis hat. In diesem Fall einmalig ein benutzereigenes Globalverzeichnis einrichten:

```
mkdir -p ~/.npm-global
npm config set prefix '~/.npm-global'
echo 'export PATH=~/.npm-global/bin:$PATH' >> ~/.bashrc
source ~/.bashrc

# Anschließend ohne sudo:
npm install -g openclaw@latest
```

4.2 Onboarding-Assistent

OpenClaw bringt einen interaktiven Konfigurationsassistenten mit. Er fragt nach dem Modellanbieter, dem API-Schlüssel und einigen Gateway-Optionen, und richtet auf Wunsch automatisch den Hintergrunddienst ein:

```
openclaw onboard --install-daemon
```

Drei Fragen werden nacheinander gestellt. Erstens der bevorzugte Modellanbieter (für die meisten wissenschaftlichen Anwendungen ist Anthropic eine gute Wahl). Zweitens der API-Schlüssel – aus dem Passwortmanager kopieren und einfügen. Drittens einige Gateway-Vorgaben (Standardport 18789, Standardagent); für den Anfang sind die Voreinstellungen passend.

Nach Abschluss läuft das Gateway als Hintergrunddienst, lauscht auf Port 18789, und die Konfiguration liegt unter ~/.openclaw/.

4.3 Konfigurationsdateien verstehen

Zwei Dateien sind zentral. Die Hauptkonfiguration `~/.openclaw/openclaw.json` steuert Modell, Port und Verhalten:

```
{
  "agent": {
    "model": "anthropic/claude-opus-4-7"
  },
```

```
  "gateway": {
    "port": 18789
  }
}
```

Die zweite Datei ~/.openclaw/.env enthält den API-Schlüssel und alle weiteren Geheimnisse:

```
# OpenClaw Umgebungsvariablen – sensible Daten!
ANTHROPIC_API_KEY=sk-ant-api03-...
```

Sicherheit der .env-Datei

Die .env-Datei enthält Klartext-API-Schlüssel und ist hochsensibel. Die Berechtigungen mit chmod 600 ~/.openclaw/.env so einschränken, dass nur der Eigentümer sie lesen kann. Niemals in ein Repository einchecken. Wer den Schlüssel versehentlich veröffentlicht hat, sollte ihn beim Anbieter sofort widerrufen und einen neuen erzeugen.

5. Erststart und Funktionsprüfung

5.1 Status prüfen

Da der Onboarding-Schritt mit --install-daemon ausgeführt wurde, läuft das Gateway bereits als Hintergrunddienst. Status prüfen:

```
openclaw gateway status

# Falls nicht aktiv: starten
openclaw gateway start

# Im Vordergrund mit ausführlicher Ausgabe (gut beim ersten Start)
openclaw gateway --port 18789 --verbose
```

5.2 Erreichbarkeit prüfen

Ein einfacher Test direkt aus dem Terminal:

```
curl http://localhost:18789/health
```

Eine erfolgreiche Antwort enthält einen JSON-Block mit dem Status „ok“. Damit ist die niedrigste Stufe der Funktionsprüfung erreicht.

5.3 Erste Nachricht an den Assistenten

Der eigentliche Test:

```
openclaw agent --message "Hallo, kannst du dich kurz vorstellen?"
```

Nach wenigen Sekunden erscheint die Antwort des Modells im Terminal. Erfolgt sie zügig und inhaltlich sinnvoll, ist das System grundsätzlich funktionsfähig – die API ist erreichbar, das Modell antwortet, das Gateway leitet korrekt durch.

Auf dem Mac mini lässt sich zusätzlich das grafische Dashboard im Browser öffnen: `openclaw dashboard` startet den Standardbrowser auf `http://localhost:18789` mit einer schlichten Oberfläche aus Chat, Sitzungsübersicht und Konfiguration. Auf dem VPS funktioniert das nicht direkt – ein Server hat keinen Browser; ein SSH-Tunnel hilft:

```
# Auf dem lokalen Rechner ausführen:
ssh -L 18789:localhost:18789 openclaw@203.0.113.42

# Solange diese SSH-Sitzung offen ist, im lokalen Browser
# http://localhost:18789 öffnen.
```

5.4 Wenn nichts funktioniert

Bleibt die Antwort aus oder erscheint eine Fehlermeldung, sind die häufigsten Ursachen in absteigender Wahrscheinlichkeit: ein nicht oder falsch gesetzter API-Schlüssel, ein nicht laufender Daemon, ein nicht erreichbarer Port. Der Reihe nach prüfen:

```
# 1. Läuft das Gateway?
openclaw gateway status

# 2. Ist der API-Schlüssel gesetzt?
cat ~/.openclaw/.env | grep API_KEY

# 3. Was sagen die Logs?
openclaw logs --tail 50
```

Eine 401-Meldung deutet auf einen ungültigen Schlüssel, eine 429-Meldung auf ein erschöpftes Kontingent, eine Verbindungs-Zeitüberschreitung auf Netzwerkprobleme.

Verifikation
Wenn an dieser Stelle eine Antwort vom Assistenten zurückkommt, ist die Grundinstallation abgeschlossen. Alle weiteren Schritte – dauerhafter Betrieb mit Reverse Proxy und TLS, Telegram-Anbindung, E-Mail-Versand, Wartung und Backups – sind Veredelungen, die in der Vollversion ausführlich behandelt werden.

6. Minimale Wartung

Auch eine Basis-Installation muss gelegentlich gepflegt werden. Drei Routinen genügen für den Anfang.

Updates einspielen

OpenClaw veröffentlicht regelmäßig neue Versionen. Mindestens monatlich:

```
npm install -g openclaw@latest
openclaw gateway restart
```

Selbstdiagnose

Das eingebaute Diagnose-Werkzeug prüft Konfiguration, Versionen und Berechtigungen:

```
openclaw doctor
```

Logs prüfen

Ein gelegentlicher Blick in die Logs fängt Probleme ab, bevor sie zu Ausfällen werden:

```
openclaw logs --tail 100
openclaw logs --follow   # Live, mit Strg+C beenden
```

Häufige Fehlerbilder

Bei „EADDRINUSE: address already in use“ ist der Port 18789 bereits belegt – meist von einer alten OpenClaw-Instanz. Mit lsof -i :18789 (Linux) bzw. lsof -nP -iTCP:18789 | grep LISTEN (macOS) den Prozess identifizieren und beenden.

Bei „429 Too Many Requests“ ist das Anbieterkontingent erreicht – entweder das Budget aufstocken oder kurz warten. Vorbeugen lässt sich durch harte Budgetgrenzen im Anbieter-Dashboard.

Bei „connection refused“ oder „getaddrinfo ENOTFOUND“ liegt ein Netzwerkproblem vor. Statusseiten der Anbieter (status.anthropic.com, status.openai.com) verraten, ob ein bekannter Ausfall vorliegt.

Wer mit einem konkreten Fehler nicht weiterkommt, findet auf github.com/openclaw/openclaw und auf docs.openclaw.ai die offiziellen Anlaufstellen.

Glossar / Abkürzungen

Begriffe, die in dieser Anleitung verwendet werden, in alphabetischer Reihenfolge:

Begriff	Erläuterung
API-Schlüssel	Zeichenkette, die Sie beim Aufruf eines externen Webdienstes (Anthropic, OpenAI, Google) authentifiziert. Funktional ein Passwort, an Ihr Konto und Ihre Abrechnung gebunden.
Daemon	Programm, das im Hintergrund läuft, ohne direkten Bezug zu einem Terminal. Auf Linux verwaltet von systemd, auf macOS von launchd.
Firewall	Komponente, die Netzwerkverkehr nach Regeln filtert. Auf Ubuntu wird hier ufw (Uncomplicated Firewall) eingesetzt.
Gateway	Im OpenClaw-Kontext der zentrale Hintergrundprozess, der Anfragen entgegennimmt, an Modellanbieter weiterleitet, Antworten formatiert und Kommunikationskanäle verwaltet.
Homebrew	De-facto-Standardpaketmanager für macOS. Übernimmt die Rolle, die unter Ubuntu apt spielt.
LTS	Long Term Support – langzeitunterstützte Version eines Produkts. Ubuntu 24.04 LTS und Node.js 24 LTS werden über mehrere Jahre mit Sicherheitsupdates versorgt.
npm	Node Package Manager. Werkzeug zur Installation und Verwaltung von Node.js-Paketen, wird zur Installation von OpenClaw eingesetzt.
Node.js	Laufzeitumgebung für JavaScript jenseits des Browsers. OpenClaw ist in TypeScript geschrieben und läuft auf Node.js.
Onboarding	Der interaktive Konfigurationsassistent von OpenClaw, aufgerufen mit openclaw onboard. Führt durch alle wesentlichen initialen Einstellungen.
SSH	Secure Shell. Protokoll für die verschlüsselte Anmeldung an entfernten Servern. Authentifizierung per Passwort (unsicher) oder Schlüsselpaar (sicher).
systemd	Dienstverwalter moderner Linux-Distributionen. Sorgt für Start, Stopp und Überwachung von Hintergrunddiensten.
VPS	Virtual Private Server. Virtualisierter Linux-Server bei einem Hosting-Anbieter, üblicherweise monatlich abgerechnet.
Workspace	Verzeichnis ~/.openclaw/workspace, in dem Skills, Projektdaten und Sitzungs-Historie abgelegt werden.

Ausblick: Workbook 2

Diese Basisversion hat den Boden bereitet: ein lauffähiger, persönlich konfigurierter KI-Assistent steht bereit. Damit ist die infrastrukturelle Mindestbasis gelegt – die Vollversion des Workbooks 1 baut darauf auf und ergänzt produktiven Betrieb mit Reverse Proxy und TLS, Telegram-Anbindung als Kommunikationskanal, einen ersten Anwendungsfall mit E-Mail-Versand sowie Wartung, Updates und Fehlerdiagnose im Detail.

Workbook 2 unter dem Arbeitstitel „The Automated Future: Strategieentwicklung mit KI“ verschiebt die Perspektive vom „Wie installiere ich es?“ auf die strategisch interessantere Frage: „Wie nutze ich es methodisch fundiert für Foresight, Strategiearbeit und Wettbewerbsanalyse?“. Im Mittelpunkt steht das Konzept einer Agenten-Architektur: spezialisierte KI-Agenten – etwa für Markt- und Wettbewerbsanalyse, für Disruptions-Scouting, für Strategieszenarien und für Umsetzungsplanung – arbeiten auf einer gemeinsamen Daten-Infrastruktur und liefern greifbare Outputs wie automatisierte Marktberichte, Zukunftsszenarien und strategische Empfehlungen.

Die geplanten Hauptkapitel: Konzeption einer Agenten-Architektur für strategische Aufgaben; Anbindung interner und externer Datenquellen, einschließlich Datenschutz- und Lizenzfragen; Aufbau einer Pilot-Umgebung mit ersten Szenarien; organisationale Verankerung und Rollenverteilung zwischen menschlichen Expertinnen und Experten als Entscheider und KI-Agenten als Analysten; sowie Skalierung vom Initialpilot zur vollständigen Integration.

Workbook 2 setzt die in diesem Buch errichtete Infrastruktur als Grundlage voraus, ist aber so geschrieben, dass es auch mit anderen Agenten-Plattformen umsetzbar bleibt. Wer Workbook 1 vollständig durchgearbeitet hat, ist für den methodischen Teil bestens gerüstet – die technischen Hürden sind genommen, der Blick kann sich auf die inhaltliche Anwendung richten.

Workbook 2 — The Automated Future: Strategieentwicklung mit KI

Erscheint voraussichtlich 2026 — Operation-zukunft / operation-zukunft.de

— Ende der Basisversion —