

Migrationsleitfaden: Vom Langdock-Workspace zur nuwacom-Plattform

Autor: Manus AI · Stand: 03. August 2026 · Version 1.0

1. Die Grundüberlegung vor dem ersten Schritt

Ein Plattformwechsel scheitert selten an fehlenden Daten, sondern fast immer an zu vielen. Wer versucht, jeden Chat, jede Datei und jede Konfiguration mitzunehmen, erzeugt im neuen System eine Halde, in der das Wesentliche nicht mehr auffindbar ist. Der wirtschaftliche Wert eines Workspace liegt nicht in seiner Historie, sondern in dem, was daraus destilliert wurde: den Instruktionen, die gut funktionieren, den Skills, die Arbeit abnehmen, den Referenzdokumenten, auf die wiederholt zugegriffen wird, und dem Profilwissen über Person und Unternehmen, das jede Antwort präziser macht.

Deshalb folgt dieser Leitfaden einem einfachen Prinzip: **Es wird nicht alles migriert, sondern alles bewertet und dann das Wertvolle vollständig migriert.** Alles, was technisch nicht übertragbar ist, wird ausdrücklich benannt, damit keine falschen Erwartungen entstehen, und für jeden dieser Fälle steht ein konkreter manueller Ersatzweg bereit.

Die zweite Grundüberlegung betrifft die Richtung des Datenflusses. Beide Plattformen besitzen APIs, aber sie sind nicht füreinander gebaut. Langdock liefert Daten nur auf **Pull** heraus, und zwar ausschließlich an einen Client, der über einen API-Key verfügt und nicht aus einem Browser heraus agiert. nuwacom nimmt Daten per **Push** an, entweder über die Benutzeroberfläche oder über einen Asset-Endpunkt. Zwischen beiden steht zwangsläufig eine lokale Zwischenstation: ein Ordner auf Deinem Rechner. Dieser Ordner ist nicht ein Umweg, sondern das eigentliche Steuerungsinstrument der Migration, weil sich dort die Struktur festlegen lässt, die später in der nuwacom-Wissensdatenbank erscheinen soll.

2. Kann nuwacom die Daten direkt bei Langdock abholen?

Diese Frage ist berechtigt und die Antwort lautet klar: **nein, nicht auf sinnvolle Weise.** Vier Gründe sprechen dagegen, und sie sind alle dokumentiert.

Erstens blockiert Langdock Anfragen mit Browser-Herkunft bewusst, um API-Keys zu schützen. In der API-Dokumentation heißt es dazu unmissverständlich:

Langdock intentionally blocks browser-origin requests to protect your API key and ensure your applications remain secure. [1]

Ein Systemprompt, der in einer nuwacom-Sitzung läuft, agiert im Browser-Kontext und könnte die Langdock-API damit nicht erreichen. Zweitens ist ein Systemprompt ohnehin nur Text und kein HTTP-Client; ein Sprachmodell kann ohne ein konfiguriertes Werkzeug keine authentifizierte Anfrage stellen. Drittens fehlen in Langdock genau jene Endpunkte, die für einen Direktzugriff interessant wären: Es existiert keine API für Chatverläufe, keine für Memory-Einträge, keine für Workflows und keine für die Promptbibliothek. Selbst mit einem gültigen Key wäre das Wichtigste nicht abrufbar.

Viertens gibt es zwar eine MCP-Schnittstelle, aber sie leistet etwas anderes als erhofft. Langdock stellt einen eigenen MCP-Server bereit, über den externe KI-Clients auf Langdock-Agenten zugreifen können [2]. Dieser Server exponiert jedoch **ausschließlich Agenten als aufrufbare Werkzeuge**, nicht die dahinterliegenden Datenbestände. MCP ist damit ein Nutzungskanal, kein Exportkanal.

Daraus ergibt sich eine praktisch nützliche Nebenerkenntnis: Für eine **Übergangsphase**, in der beide Systeme parallel laufen, lässt sich ein Langdock-Agent mit angehängter Wissensdatenbank per MCP aus nuwacom heraus befragen. Das ist eine elegante Brücke, um während der Umstellung nicht den Zugriff auf noch nicht migriertes Wissen zu verlieren. Es ersetzt aber keine Migration, weil es Retrieval bleibt und die Daten weiterhin bei Langdock liegen.

3. Die Migrationsmatrix: Was geht wie

Die folgende Übersicht ist das Herzstück des Leitfadens. Sie ordnet jeden Objekttyp seinem besten Übertragungsweg zu und benennt den realistischen Aufwand.

Objekt	Weg aus Langdock heraus	Weg in nuwacom hinein	Aufwand	Charakter
Skills	Skills-API (<code>List</code> und <code>Get</code>) oder manuelles Kopieren von <code>SKILL.md</code> in der Skill-Detailansicht	Skills → Add Skill → Upload a Skill mit <code>SKILL.md</code> , <code>.zip</code> oder <code>.skill</code>	niedrig	verlustfreie Übertragung
Dateien und Dokumente	Library → Recent Files → Listenansicht → Alle auswählen → Sammel-Download	Knowledge Base → Ordner-Upload per Drag-and-drop oder Asset-API	niedrig	Struktur bleibt erhalten
Wissensdatenbank-Inventar	Knowledge Folder API <code>GET /knowledge/{folderId}/list</code>	begleitendes Manifest als CSV oder Markdown	niedrig	Kontrollinstrument
Agenten	Agent-API <code>GET /agent/v1/get</code> liefert die vollständige Konfiguration als JSON	Agent manuell anlegen, Instruktionen einsetzen	mittel	Konfiguration übertragbar, Anbindungen neu
Profilwissen aus Memory und Chats	MEGA Prompt und die Zusatz-Prompts A bis D	Markdown-Datei in die Wissensdatenbank	mittel	separat gelöstes Teilprojekt
Promptbibliothek	kein Export, manuelles Kopieren des Prompt-Textes	Prompts → Create prompt, Variablensyntax anpassen	mittel	Auswahl statt Vollübertragung
Workflows	kein Export, Struktur dokumentieren	Beschreibung in den nuwacom-Chat geben, Workflow wird generiert	mittel	Neubau mit Rückenwind
Integrationen und OAuth	nicht übertragbar	manuell neu verbinden	mittel	zwingend manuell
Wörtliche Chatverläufe	kein Export	–	hoch	bewusst nicht migrieren
Custom Instructions des Workspace	Einstellungen einsehen und kopieren	Workspace Settings → AI Guardrails	niedrig	wird häufig übersehen
Projekte	Projektinstruktionen kopieren	nuwacom Projects mit projektbezogenen Guardrails	niedrig	manuell, schnell

4. Der Ablauf in acht Etappen

Etappe 1: Inventur und Priorisierung

Bevor irgendetwas exportiert wird, entsteht eine Liste dessen, was überhaupt existiert. Öffne Langdock und notiere für jeden Bereich, was vorhanden ist: Skills, Agenten, Workflows, Prompts, Knowledge bases, Folders und Projekte. Diese Liste beantwortet danach eine einzige, entscheidende Frage pro Eintrag: Wurde das in den letzten Monaten tatsächlich genutzt?

Für Agenten und Projekte lässt sich diese Frage objektiv beantworten, denn die **Usage Export API** liefert Nutzungsdaten zu Nutzern, Agenten, Modellen, Projekten und API-Keys als CSV oder JSON [3]. Damit erhältst Du eine belastbare Rangliste statt eines Bauchgefühls. Alles, was seit Monaten unberührt liegt, wird nicht migriert, sondern nur in der Inventurliste vermerkt. Diese Liste ist selbst ein wertvolles Dokument und gehört später mit in die neue Wissensdatenbank, weil sie dokumentiert, was bewusst zurückgelassen wurde.

Als Ergebnis dieser Etappe entsteht der lokale Migrationsordner. Seine Struktur bestimmt, wie das Wissen später in nuwacom aussieht, deshalb lohnt hier ein sorgfältiger Moment:

```
migration-langdock-nuwacom/
├── 00_inventar/
├── 01_profilwissen/
├── 02_skills/
├── 03_agenten/
├── 04_prompts/
├── 05_workflows/
├── 06_dateien/
│   ├── referenzdokumente/
│   ├── vorlagen/
│   └── ergebnisse/
└── 07_integrationen/
```

Etappe 2: Zugang schaffen

Ein API-Key entsteht in Langdock unter Workspace Settings im Bereich Products bei der API-Konfiguration. Beim Anlegen werden die Berechtigungsbereiche gewählt; für dieses Vorhaben sind `SKILL_API`, `AGENT_API` und `KNOWLEDGE_FOLDER_API` relevant, ergänzt um den Bereich der Usage-Export-Funktion. Die Basis-URL lautet `https://api.langdock.com`, bei einem dedizierten Deployment stattdessen `https://<deine-domain>/api/public` [1].

Zwei Fallstricke sind hier wichtig. Erstens muss jeder Agent, dessen Konfiguration ausgelesen werden soll, **ausdrücklich mit dem API-Key geteilt werden**, indem man den Agenten öffnet, die Freigabe aufruft und den Namen des Keys hinzufügt; bei Subagenten gilt das für jeden einzelnen. Zweitens gilt dasselbe für Knowledge bases, wobei der Key für lesende Zugriffe die Viewer-Rolle und für schreibende die Editor-Rolle benötigt. Beide Freigaben können nur Administratoren erteilen.

Auf der Zielseite erstellst Du parallel einen nuwacom-API-Key unter Workspace Settings im Bereich API Keys. Auch das ist Administratoren vorbehalten. Die instanzspezifische Schnittstellenbeschreibung findest Du unter `https://{dein-tenant}.nuwacom.ai/api/openapi.json`, komfortabel erkundbar über `.../api/scalar` oder `.../api/swagger` [4].

Etappe 3: Skills übertragen — der beste Hebel

Skills sind der erfreulichste Teil der Migration, weil beide Plattformen dasselbe Dateiformat verwenden. Langdock beschreibt einen Skill als Markdown-Datei mit einem YAML-Kopf, der `name` verpflichtend sowie `slug`, `description` und `license` optional enthält, gefolgt vom eigentlichen Instruktionstext [5]. nuwacom akzeptiert beim Anlegen eines Skills exakt diese Formate:

Upload a Skill — Import an existing Skill file in one of the supported formats (SKILL.md, .zip, or .skill) [6]

Damit sind Skills **ohne Nachbau eins zu eins übertragbar**. Sogar die Grenzwerte stimmen überein: In beiden Systemen sind zwanzig Skills pro Nutzer gleichzeitig aktiv, und der Instruktionstext darf fünfzigtausend Zeichen umfassen. nuwacom erlaubt sogar größere Begleitdateien, sodass beim Transfer in diese Richtung keine Grenze verletzt wird.

Praktisch gehst Du so vor: Über die Skills-API listest Du alle Skills und rufst jeden einzelnen ab. Aus Name, Slug, Beschreibung und Instruktionen entsteht je Skill eine `SKILL.md`, die Du unter `02_skills/` ablegst. Wo Begleitdateien existieren, packst Du sie mit der `SKILL.md` in ein ZIP. Ohne API-Zugang funktioniert auch der Handweg: In der Skill-Detailansicht öffnet man `SKILL.md`, klickt auf Bearbeiten und kopiert Beschreibung und Instruktionen heraus [5]. Bei einer überschaubaren Anzahl von Skills ist das oft schneller als der Skriptweg.

In nuwacom lädst Du die Dateien anschließend über Skills, Add Skill und Upload a Skill hoch. Ein Detail lohnt Aufmerksamkeit: Die Beschreibung ist in beiden Systemen der **Auslöser**, an dem das Modell erkennt, wann ein Skill greifen soll. Wenn ein Skill in Langdock schon unzuverlässig ansprang, ist der Umzug der richtige Moment, die Beschreibung mit konkreten Auslösephrasen zu schärfen, statt den Fehler mitzunehmen.

Etappe 4: Dateien übertragen

Für Dateien existiert ein überraschend bequemer Weg, der ohne jede Programmierung funktioniert. Der Bereich **Recent Files** in der Library zeigt alle Dateien des Workspace, unterteilt nach Herkunft in selbst erzeugte, hochgeladene und aus Integrationen importierte Dateien. Er bietet Filter nach Dateityp, Zeitraum und Herkunft sowie Stapelaktionen: In der Listenansicht wählt ein Klick auf das Kopfkästchen alle sichtbaren Dateien aus, danach erweitert eine Schaltfläche die Auswahl auf sämtliche Treffer, und ein Sammel-Download liefert alles auf einmal. Erzeugte Dokumente lassen sich dabei als Word, PDF oder **Markdown** herunterladen [7].

Markdown ist für den Import in eine Wissensdatenbank die beste Wahl, weil die Struktur erhalten bleibt und die Datei ohne Konvertierungsverlust zerlegbar ist. Wo die Wahl besteht, sollte sie also auf Markdown fallen.

Zwei Einschränkungen sind zu beachten. Dateien, die in Folders liegen, und solche, die nur innerhalb von Workflows existieren, erscheinen nicht in Recent Files [7]. Und Dateien aus einer Knowledge base lassen sich **nicht per API herunterladen**, weil die Schnittstelle nur Metadaten und automatisch erzeugte Zusammenfassungen liefert, aber keinen Inhaltsabruf [8]. Für Knowledge bases gibt es dafür einen eleganten Umweg über die Benutzeroberfläche: Die Funktion **Convert to folder** verwandelt eine Wissensdatenbank in einen gewöhnlichen Ordner, wodurch die Dateien direkt zugänglich und herunterladbar werden. Dabei gelten allerdings die niedrigeren Ordnergrenzen von hundert Dateien und zweihundertfünfzig Megabyte, und die Umwandlung ist blockiert, solange die Wissensdatenbank noch mit API-Zugriffen, Assistenten, Chats oder Automatisierungen verbunden ist [9].

Auf der Zielseite spielt nuwacom seine Stärke aus. Die Wissensdatenbank nimmt nicht nur einzelne Dateien, sondern **komplette Ordnerstrukturen inklusive Unterordner** per Drag-and-drop an, und das Formatspektrum ist deutlich breiter als bei Langdock: Neben Text, Markdown, PDF und Office-Dokumenten werden auch Tabellen, Bilder und Audiodateien verarbeitet [10]. Genau deshalb ist der lokale Migrationsordner so wertvoll: Was Du dort strukturierst, erscheint in nuwacom in derselben Ordnung.

Wer den Vorgang wiederholbar machen will, nutzt den Asset-Endpunkt. Ein `POST` auf `/api/kb/assets/upload` erwartet als Formulardaten die Space-Kennung und die Datei, akzeptiert optional aber auch ein `path`-Feld als Array von Ordnernamen und ein `metadata`-Feld für eigene Kennzeichnungen [11]. Damit lässt sich die Hierarchie beim

Hochladen mitgeben. Besonders angenehm ist das Verhalten bei Wiederholungen: Der Endpunkt antwortet mit dem Status 201, wenn eine Datei neu angelegt wurde, und mit 200, wenn sie bereits existierte. Der Vorgang ist also gefahrlos wiederholbar und erzeugt keine Duplikate.

Etappe 5: Agenten übertragen

Für Agenten liefert die Agent-API mit `GET /agent/v1/get?agentId=<UUID>` die vollständige Konfiguration der veröffentlichten Version. Die Antwort enthält Name, Beschreibung, das Feld `instruction` mit dem eigentlichen Systemprompt, das Modell, die Temperatur, die Gesprächseinstiege, den Eingabetyp sowie die Schalter für Websuche, Bildgenerierung und erweitertes Denken. Die Agenten-Kennung findest Du in der Adresszeile, wenn Du einen Agenten zum Bearbeiten öffnest.

Langdock benennt in seinem eigenen Migrationsleitfaden die Grenzen offen, und sie gelten für den Wechsel zu nuwacom umso mehr. Anhänge werden nicht übertragen, sondern nur als Kennungen genannt, weshalb die Dateien separat aus Recent Files kommen müssen. Aktionen und Integrationen tragen arbeitsbereichsspezifische Kennungen und müssen neu konfiguriert werden. OAuth-Verbindungen sind grundsätzlich nicht per Schnittstelle übertragbar. Formularfelder sind in der Antwort unvollständig beschrieben, sodass Feldtypen, Auswahloptionen und Dateitypbeschränkungen von Hand nachzubauen sind. Und Gesprächsverläufe wandern ohnehin nicht mit.

In nuwacom legst Du den Agenten neu an, wahlweise aus einer Vorlage, mit KI-Unterstützung oder von Null. Die Konfigurationsfelder entsprechen den Langdock-Feldern weitgehend: Avatar, Name, Beschreibung, Kategorie, Instruktionen, angehängtes Wissen und Starteinstellungen mit entweder einem Startformular oder bis zu vier Gesprächseinstiegen [12]. Der Langdock-Systemprompt aus dem Feld `instruction` lässt sich unverändert einsetzen. nuwacom empfiehlt für Instruktionen eine Gliederung nach Aufgabe, Handlung, Antwortregeln und Ziel; wenn ein übernommener Prompt sich mühelos in dieses Raster bringen lässt, ist das eine gute Gelegenheit zur Aufräumarbeit.

Ein Blick in die Vorlagenbibliothek von nuwacom lohnt vor jedem Nachbau. Für verbreitete Zwecke wie Recherche, Korrektur, Prompt-Optimierung oder Prüfroutinen existieren bereits fertige Agenten. Wo eine Vorlage Deinen Zweck erfüllt, ist sie dem Nachbau vorzuziehen, weil sie auf die Eigenheiten der Plattform abgestimmt ist.

Etappe 6: Prompts übertragen

Die Langdock-Promptbibliothek besitzt keinen Export. Prompts müssen also einzeln kopiert werden, was den Blick zwangsläufig auf die Frage lenkt, welche Prompts diesen Aufwand wert sind. Erfahrungsgemäß ist das eine kleine Minderheit, und genau darin liegt die Chance: Der Umzug ist die beste Gelegenheit, eine über Jahre gewachsene Sammlung auf die tatsächlich genutzten Stücke zu reduzieren.

Ein technisches Detail entscheidet über Funktionsfähigkeit oder Fehlverhalten. Langdock kennzeichnet Variablen mit **doppelten** geschweiften Klammern in der Form `{{ tonalitaet }}` [13], nuwacom hingegen mit **einfachen** Klammern in der Form `{tonalitaet}` [14]. Wer Prompts unverändert überträgt, erhält Platzhalter, die nicht als Eingabefelder erkannt werden. Die Umschreibung ist trivial, aber unverzichtbar.

In nuwacom öffnest Du einen neuen Chat, tippst `@`, fährst über Prompts und wählst `Manage prompts`, um über `Create prompt` einen Prompt mit Titel, Kategorie, Beschreibung und Instruktion anzulegen. Zusätzlich lässt sich festlegen, ob der Prompt im Editor, im Chat oder im Erstellungsbereich erscheinen soll. Wer in Langdock mit Formularen gearbeitet hat, findet die Entsprechung in der Option, Variablen als Formularfelder anzuzeigen [14].

Etappe 7: Workflows neu aufsetzen — schneller als erwartet

Workflows sind in Langdock nicht exportierbar, und selbst wenn sie es wären, ließe sich das Ergebnis nicht importieren, weil nuwacom ein eigenes Format verwendet. Innerhalb von nuwacom existiert zwar ein Austauschweg, bei dem ein Workflow über das Dreipunktmenü als JSON heruntergeladen und von anderen Nutzern wieder hochgeladen wird [15], doch dieser Weg beginnt erst in nuwacom.

Die gute Nachricht ist, dass der Neubau kaum Aufwand kostet, wenn man ihn richtig angeht. nuwacom kann einen mehrstufigen Workflow **aus einer Beschreibung im Chat automatisch erzeugen** und stellt ihn danach im visuellen

Builder zur Feinjustierung bereit [15]. Der effizienteste Weg besteht daher darin, jeden erhaltenswerten Langdock-Workflow einmal sauber in Prosa zu beschreiben und diese Beschreibung in den nuwacom-Chat zu geben.

Für die Beschreibung genügt ein knappes Raster: Was löst den Ablauf aus, welche Schritte folgen in welcher Reihenfolge, welche Prompts oder Agenten kommen je Schritt zum Einsatz, welche Verzweigungen und Bedingungen existieren, welche externen Systeme werden berührt, und was soll am Ende entstehen. Diese Beschreibungen legst Du unter `05_workflows/` ab. Sie haben doppelten Wert, denn sie sind gleichzeitig Bauanleitung und Dokumentation.

Etappe 8: Instruktionen, Integrationen und Projekte

Zum Abschluss folgen die Elemente, die häufig vergessen werden, obwohl sie die Antwortqualität am unmittelbarsten prägen. Was in Langdock als Custom Instructions oder als Workspace-Beschreibung hinterlegt ist, gehört in nuwacom in die **AI Guardrails** der Workspace-Einstellungen. Dort werden Tonalität, verbindliche Regeln, zu vermeidende Formulierungen sowie Beispiele für erwünschten und unerwünschten Stil festgelegt. Wird dieser Schritt übersprungen, verhält sich die neue Plattform generisch, obwohl alles andere korrekt migriert wurde.

Integrationen und OAuth-Verbindungen müssen ausnahmslos manuell neu eingerichtet werden. Das ist kein Mangel, sondern eine Sicherheitseigenschaft, denn Zugriffstoken sind absichtlich nicht portierbar. Notiere in `07_integrations/`, welche Verbindungen bestanden, wer sie freigegeben hatte und welche Agenten oder Workflows darauf angewiesen waren. Diese Liste wird zur Abarbeitungsliste im neuen System.

Projekte schließlich übertragen sich über ihre Instruktionen. nuwacom-Projekte nehmen Dateien als Projektkontext auf und erlauben projektbezogene Guardrails für Stil, Markenregeln und verbindliche Fakten [16]. Der Inhalt der Langdock-Projektinstruktionen findet dort seinen Platz.

5. Reihenfolge und Zeitrahmen

Die Etappen sind nicht beliebig vertauschbar. Zugänge müssen vor allen Abrufen bestehen, Dateien vor den Agenten wandern, die auf sie verweisen, und die Guardrails sollten früh gesetzt sein, weil alles Folgende in ihrem Rahmen arbeitet.

Reihenfolge	Etappe	Aufwand	Abhängigkeit
1	Inventur und Priorisierung	ein bis zwei Stunden	keine
2	Zugänge und Freigaben	dreißig Minuten	Administratorrechte in beiden Systemen
3	Profilwissen exportieren	ein bis zwei Stunden	MEGA Prompt und Zusatz-Prompts
4	Skills übertragen	fünfzehn Minuten je Skill	Etappe 2
5	Dateien übertragen	ein bis drei Stunden	Etappe 1 für die Struktur
6	Guardrails setzen	dreißig Minuten	keine
7	Agenten nachbauen	zwanzig bis vierzig Minuten je Agent	Etappen 4 und 5
8	Prompts übertragen	zwei bis drei Minuten je Prompt	keine
9	Workflows generieren	dreißig bis sechzig Minuten je Workflow	Etappe 7
10	Integrationen verbinden	je nach System	Administratorrechte

Für einen typischen Arbeitsbereich mit einer Handvoll Skills, wenigen aktiven Agenten und einem überschaubaren Dateibestand ist die Migration an zwei konzentrierten Arbeitstagen zu schaffen. Es empfiehlt sich, beide Plattformen anschließend zwei bis vier Wochen parallel zu betreiben und Nachzügler bei Bedarf zu holen, statt vorsorglich alles mitzunehmen.

6. Wie Datenchaos vermieden wird

Vier Regeln genügen, um im neuen System Ordnung zu halten.

Die erste ist **Nutzung als Filter**: Was in Langdock über Monate nicht angefasst wurde, wird nicht migriert. Die Usage-Export-Daten machen diese Entscheidung überprüfbar statt willkürlich.

Die zweite ist **Verdichtung statt Vollständigkeit**: Chatverläufe wandern nie wörtlich, sondern ausschließlich als thematische Wissens-Extrakte. Ein Jahr Rohprotokolle in einer Wissensdatenbank verschlechtert die Trefferqualität, weil semantische Suche dann auf Gesprächsrauschen statt auf Aussagen stößt.

Die dritte ist **eine Struktur, festgelegt vor dem ersten Upload**: Der lokale Migrationsordner ist die einzige Stelle, an der Ordnung entsteht. Da nuwacom Ordnerstrukturen mitsamt Unterordnern übernimmt, wird diese Ordnung unmittelbar zur Ordnung im Zielsystem.

Die vierte ist **ein Manifest als Kontrollinstrument**: Aus den Listenabrufen der Knowledge-Folder-API entsteht eine Übersicht aller Dateien mit Namen, Typ, Datum und automatisch erzeugter Zusammenfassung [8]. Diese Übersicht gehört als eigene Datei unter `00_inventar/` und wird mit importiert. Sie beantwortet später die Frage, ob etwas fehlt, ohne dass Du beide Systeme vergleichen musst.

7. Was definitiv nicht mitkommt

Ehrlichkeit an dieser Stelle erspart Enttäuschungen. Wörtliche Chatverläufe sind nicht exportierbar, da Langdock weder eine Exportfunktion noch eine Chat-API anbietet; die einzige Freigabemöglichkeit richtet sich an den Langdock-Support. Memory-Einträge sind nur über die Benutzeroberfläche und über Prompts erreichbar, nicht über eine Schnittstelle. Workflow-Definitionen und Prompt-Bibliothek besitzen keinen maschinellen Ausgang. OAuth-Verbindungen und Aktionskennungen sind arbeitsbereichsgebunden. Nutzungsstatistiken und Auditprotokolle lassen sich exportieren, aber nicht in ein anderes System einspeisen. Und Dateien aus Knowledge bases sind nur über den Umweg der Ordnerumwandlung oder über Recent Files zu bekommen, nicht per API-Download.

Für jeden dieser Punkte existiert ein manueller Ersatzweg, und in Summe sind es wenige Stunden Handarbeit. Entscheidend ist, sie eingeplant zu haben, statt mitten in der Migration darauf zu stoßen.

References

[1] [API Introduction – Langdock Docs](#) [2] [Model Context Protocol – Langdock Docs](#) [3] [Usage Export API – Langdock Docs](#) [4] [API Introduction – nuwacom Docs](#) [5] [Creating Skills – Langdock Docs](#) [6] [Skills – nuwacom Docs](#) [7] [Recent Files – Langdock Docs](#) [8] [Retrieve Files from Knowledge Folder – Langdock Docs](#) [9] [Knowledge bases – Langdock Docs](#) [10] [Knowledge – nuwacom Docs](#) [11] [Upload an asset – nuwacom API Reference](#) [12] [Agents – nuwacom Docs](#) [13] [Prompt Library – Langdock Docs](#) [14] [Prompts – nuwacom Docs](#) [15] [Workflows – nuwacom Docs](#) [16] [Getting Started – nuwacom Docs](#)