Ordivis

TEIL I · GRUNDLAGEN

Der Ordivis Stack

Wie Ordivis Updater, Ordivis Service und die Fachanwendungen zusammengehören

Produktversion 2026.09.26 · Handbuch-Revision 1.0
Stand 02.10.2026 · Plattform .NET 10 / Windows Server

OU-01 Stack

© 2026 Grams IT · Alle Rechte vorbehalten

Der Ordivis Stack

Dieses Kapitel erklärt, wie eine Ordivis-Installation aufgebaut ist und welchen Platz **Ordivis Updater** darin einnimmt. Es richtet sich an **Administratoren**, die die Anlage betreiben.

Die Installationsschritte stehen in *Einrichten* (OU-10), der laufende Betrieb in *Aktualisieren* (OU-30), der Ernstfall in *Wenn ein Update schiefgeht* (OU-50).

INHALT

1 Was „Ordivis Stack" bedeutet
2 Was Ordivis Updater tut
3 Wie eine Anwendung anschlussfähig wird
4 Die Verbindungen
5 Was Ordivis Updater nie anfasst
6 Signaturen

1 Was „Ordivis Stack" bedeutet

Eine Ordivis-Installation besteht nicht aus einem Programm, sondern aus **Schichten**. Jede setzt auf der darunter auf – wie beim OSI- oder TCP/IP-Modell. Für diesen Aufbau verwenden wir den Begriff **Ordivis Stack**.

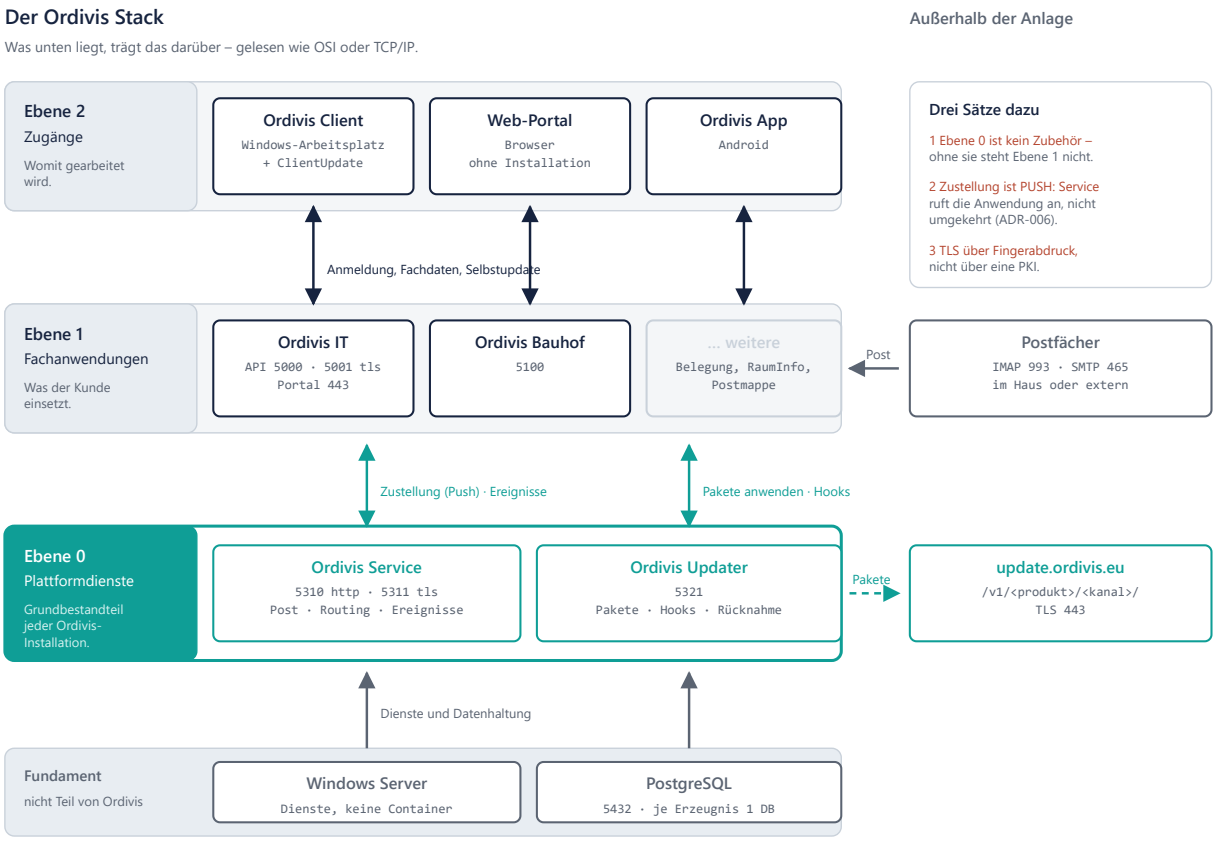

Abb. 1 – Der Ordivis Stack. Was unten liegt, trägt das darüber. Ordivis Updater bildet zusammen mit Ordivis Service die Ebene 0.

Ebene	Was dort liegt	Beispiel
2 – Zugänge	Womit gearbeitet wird	Ordivis Client, Web-Portal, App
1 – Fachanwendungen	Was der Kunde einsetzt	Ordivis IT, Ordivis Bauhof
0 – Plattformdienste	Der gemeinsame Unterbau	Ordivis Service, **Ordivis Updater**
Fundament	Nicht Teil von Ordivis	Windows Server, PostgreSQL

> **Ebene 0 ist kein Zubehör.** Ordivis Updater ist ein **Grundbestandteil jeder Ordivis-Installation**. Ohne ihn aktualisiert sich jede Fachanwendung selbst – mit eigenem Sicherungsverfahren, eigener Rücknahme und eigenem Zeitpunkt. Bei zwei Anwendungen ist das unübersichtlich, bei fünf nicht mehr zu verantworten.

2 Was Ordivis Updater tut

Er ist der **Wartungsdienst** der Anlage:

Schritt	Was geschieht
Bestand aufnehmen	Welche Ordivis-Anwendungen liegen auf diesem Rechner, in welcher Fassung
Katalog abgleichen	Was der Hersteller anbietet
Planen	Was in welcher Reihenfolge zu tun ist – mit Abhängigkeiten
Freigeben	Ein Mensch bestätigt mit Namen; die Freigabe ist ein Nachweis
Anwenden	Sichern, Dienst anhalten, tauschen, migrieren, prüfen
Zurücknehmen	Wenn die Prüfung fehlschlägt – selbsttätig

⚠️ **Inventar und Katalog sind zwei verschiedene Dinge, und die Unterscheidung ist wichtig.**

	sagt
Inventar	„Ich *sehe* diese Installation" – die Bestandsaufnahme findet jede auf dem Rechner
Katalog	„Ich *kann* sie aktualisieren" – dafür braucht es ein signiertes Paket im Feed

Eine Anwendung, die im Inventar steht, aber nicht im Katalog, kann der Updater **nicht** versorgen.

3 Wie eine Anwendung anschlussfähig wird

Ordivis Updater braucht keinen Code in der Fachanwendung. Sie liefert eine **Beschreibung** mit — `ordivis-hooks.json`, im selben Verzeichnis wie ihre Programmdatei:

```
{
  "productId": "bauhof",
  "phasen": {
    "pre":         [ { "art": "Datenbanksicherung", "verbindung": "haupt" } ],
    "migrate":     [ { "art": "Prozess", "programm": "Ordivis.Bauhof.Api.exe", "argumente":
    "healthcheck": [ { "art": "Http", "adresse": "http://localhost:5100/gesundheit" } ],
    "rollback":    [ { "art": "Datenbankwiederherstellung", "verbindung": "haupt" } ]
  }
}
```

Vier Regeln, an denen es sonst scheitert:

1. ⚠️ `healthcheck` **ist Pflicht und muss fachlich prüfen** – Datenbank erreichbar, Schema auf Stand. „Prozess läuft" genügt nicht: Er ist die *einzige* Aussage, auf die sich der Updater bei „hat es geklappt?" stützt.
2. ⚠️ **Wer** `migrate` **zusagt, muss** `rollback` **zusagen.** Eine Anwendung, die vorwärts migriert und nicht zurückkann, macht den Rückweg des **ganzen Laufs** unmöglich.
3. ⚠️ `migrate` **muss wirklich migrieren** – mit einem Schritt `Prozess`, etwa `<App>.exe schema-abgleich`. Eine Dateiprüfung allein sagt eine Migration zu, führt aber keine aus: Das Schema bleibt alt, und die neue Fassung scheitert beim Start (Ordivis Bauhof, 18.09.2026). `ordivis-hooks pruefen` warnt davor.

4. ⚠️ **Die Sicherung gehört in** `pre`. Dort läuft die Anwendung noch, und es ist nichts angefasst — schlägt sie fehl, endet der Lauf, bevor er begonnen hat.

4 Die Verbindungen

Von	Nach	Port / Adresse
Ordivis Updater	`update.ordivis.eu`	TLS 443, **ausgehend**
Administrator	Verwaltung Updater	5321
Arbeitsplatz-Client	Updater (Ringe, optional)	5321
Ordivis Updater	Dienste der Anwendungen	örtlich, kein Netz

⭐ **Nur eine Verbindung verlässt die Anlage**, und sie geht **hinaus**. Von außen muss nichts hereinkommen.

5 Was Ordivis Updater nie anfasst

Muster	Warum
`appsettings*.json`	Token, Anbindung, Verbindungszeichenfolgen
`*.key`	Schlüssel der Anlage
`data/**`	Nutzdaten
`server-url.txt`	Serveradresse, je Maschine
`*.db`	örtliche Datenbanken

⚠️ **Diese Regel gilt für das Anwenden UND die Nachprüfung – als eine Regel, nicht als zwei Listen.** Als getrennte Listen führte das dazu, dass die Prüfung eine geschützte Datei für „fehlt" hielt und eine Rücknahme auslöste. Bei jedem Lauf, endlos.

6 Signaturen

Pakete sind **ES256-signiert**, und die Prüfung ist nicht abschaltbar. Jedes Produkt hat ein **eigenes** Schlüsselpaar; der Katalog nennt bei jedem Erzeugnis seine `keyId`.

⭐ **Ein Schlüssel für alles wäre ein Generalschlüssel.** Wer den Schlüssel eines Produkts erbeutet, kann damit gefälschte Pakete *dieses* Produkts unterschreiben – aber keine anderen.

⚠️ **Der private Schlüssel erreicht keine Anlage.** Er bleibt beim Hersteller. Auf der Anlage liegt ausschließlich der öffentliche Teil, unter `C:\ProgramData\Ordivis\Updater\schluessel\`.