

ERP-koppelvlak Grossier.nu — specificatie

Versie	0.5, concept
Datum	22 oktober 2026
Status	concept 0.5, openbaar; gepubliceerd op grossier.nu/erp-koppelvlak (wijzigingen in bijlage C)
Technisch deel	Hoofdstuk 10 en verder, plus de OpenAPI-beschrijving (apart document)
Contact	Grossier.nu, ontwikkeling, via support@grossier.nu

Dit document beschrijft wat het Grossier.nu-platform van het ERP van een grossier vraagt om rechtstreeks te koppelen. Het eerste deel (hoofdstuk 1 tot en met 9) is geschreven voor wie bij de ERP-leverancier beslist of en hoe dit gebouwd wordt; het bevat ook de bindende eisen. Het tweede deel (hoofdstuk 10 en verder) is voor de bouwer en beschrijft samen met de OpenAPI de aanbevolen vorm.

De namen in dit document zijn de namen van het koppelvlak, niet van een bestaand ERP. Het ERP mag intern anders heten en anders werken; het koppelvlak is wat het ERP naar buiten aanbiedt.

Deel 1 — voor de beslisser

1. Waarom dit koppelvlak

Grossier.nu is een bestelplatform voor automotive grossiers. Garages bestellen via de webshop van hun grossier of vanuit hun garagesoftware; het platform zorgt voor catalogus, zoeken op kenteken, prijsvorming, levertijden en de bestelroute naar leveranciers. Het ERP van de grossier is de administratie: daar horen klanten, orders, leveringen, facturen en de eigen voorraad.

Elke grossier heeft een ander ERP. Tot nu toe bouwde het platform per ERP een eigen koppeling, op de API die dat ERP toevallig bood. Dat werkt, maar elke koppeling kan net iets anders en elke uitbreiding moet per ERP opnieuw. Dit document keert dat om: het beschrijft één koppelvlak dat een ERP-leverancier implementeert, waarna het platform met één generieke koppeling kan aansluiten.

Het model is ontleend aan de koppeling die het platform sinds medio 2026 met Odoo in productie heeft. Die koppeling is de referentie: alles in het volledige koppelvlak doet het platform daar al; de uitbreidingen zijn daar óf ook gebouwd óf bewust bij het platform gelaten.

Een ERP-leverancier hoeft dit koppelvlak niet integraal over te nemen. Wie een eigen API heeft, kan hieruit kiezen wat nodig is om een bepaalde functie voor zijn grossiers mogelijk te maken, zolang de eisen uit hoofdstuk 5 gelden. Wie de vorm wél overneemt, koppelt zonder maatwerk.

2. Filosofie: wat het platform doet en wat het ERP doet

Het wezenlijke verschil met oudere koppelingen is de werkverdeling. In het klassieke model is het ERP de bron van bijna alles wat de garage ziet: inlog, prijs, voorraad, levertijd. In dit model is het ERP de plek waar orders landen en waar de eigen voorraad en de administratie staan; de rest regelt het platform.

Onderwerp	Platform	ERP
Inloggen van de garage	Beheert de inlog van garages en hun gebruikers.	Kent één integratiegebruiker met een sleutel. Optioneel: valideert zelf de garage-inlog (zie module Login).
Klantgegevens	Bewaart per garage het klant-id van het ERP.	Is de bron van de debiteur: naam, adres, blokkade, kredietruimte.
Catalogus en zoeken	Levert TecDoc-, Aldoc-, banden- en OE-zoeken en kenteken-lookup.	Hoeft geen catalogus of zoekfunctie te bieden.
Artikelstam van de grossier	Leest de artikelstam van het ERP in bulk in om te kunnen zoeken en het assortiment te kennen.	Levert zijn eigen artikelstam: artikelnummer, merk, EAN, omschrijving.
Eigen voorraad	Toont de eigen voorraad van de grossier, live per pagina en als lokale kopie.	Levert vrije voorraad per vestiging, live en in bulk.
Voorraad en prijs van leveranciers	Haalt die uit leveranciersfeeds, buiten het ERP om.	Hoeft die niet te leveren.
Verkoopprijs	Berekent de netto prijs voor de garage en stuurt die mee op de order.	Boekt de meegegeven prijs. Optioneel: is zelf de prijsbron (zie module Prijs).
Routes en levertijd	Berekent de bezorgroutes en belooft een levermoment, per regel als dat per herkomst verschilt.	Neemt het beloofde levermoment over op de order en per regel. Optioneel: levert het routeschema per klant (zie module Routes).
Verkooporder	Plaatst de order zodra de garage bestelt, met alle context (voertuig, referentie, levermoment, betaalwijze).	Maakt de order aan, bevestigt hem en geeft het ordernummer terug.
Inkoop	Weet per orderregel uit welk depot of van welke leverancier het artikel komt.	Verwerkt de inkoop in het eigen proces. Optioneel: ontvangt per leverancier een inkooporder van het platform (zie module Inkoop).
Orderstatus en historie	Toont de garage zijn orders met leverstatus, ook orders die buiten het platform om zijn geplaatst.	Levert orders en wijzigingen per klant.
Facturen	Toont facturen en creditnota's in de shop.	Levert facturen en creditnota's per klant, met betaalstatus.
Retour	Laat de garage een retour aanmelden op een eerdere order.	Registreert de retour, verwerkt ontvangst en creditering.

Twee gevolgen van deze verdeling zijn het benoemen waard:

- **Een storing in het ERP legt de shop niet stil.** Zoeken, prijzen en leveranciersvoorraad werken door; alleen de eigen voorraad en het plaatsen van de order wachten. Orders worden in een wachtrij gezet en opnieuw aangeboden.
- **Het ERP hoeft niets te weten van garages als gebruikers.** Alleen van debiteuren.

3. Drie lagen: minimum, volledig koppelvlak, uitbreidingen

Het platform eist niet dat een ERP alles levert. Het koppelt al op een klein minimum en werkt dan met minder; alles daarboven meldt het ERP in `GET /v1/capabilities`, en het platform past zich daarop aan. Een onderdeel dat ontbreekt geeft geen storing, maar een functie die de garage in de shop mist. Per onderdeel staat daarom wat er dan wegvalt. Hoe de uitbreidingen in de praktijk samengaan, staat in 3.4.

3.1 Minimum om te koppelen

Zonder deze drie onderdelen koppelt het platform niet. Ze dekken de drie identiteiten waar het platform niet zonder kan: artikel, klant en order.

Onderdeel	Wat het ERP levert	Waar het platform het voor gebruikt
Artikelinfo, live	Voor een lijst sleutels (artikelnummer + merk, EAN of ERP-artikel-id, tot 500 per aanroep): kent het ERP het artikel, en wat is de vrije voorraad per vestiging. Exact matches, nooit op een deel van het nummer.	Elke artikelpagina in de shop: "bij de grossier op voorraad", besteleenheid, de controle vlak vóór verzenden of eigen voorraad er nog is.
Klant lezen	Klantgegevens op ERP-klant-id: naam, adres, e-mail, vestiging, blokkade, kredietruimte, betaalvoorwaarden. Een id dat het ERP niet kent is 404.	De garage aan de juiste debiteur koppelen en dat meteen controleren, blokkades en kredietruimte respecteren vóór het bestellen, afleveradres tonen. Elke order verwijst naar dit klant-id.
Order plaatsen, idempotent	Een verkooporder aanmaken en bevestigen op basis van een unieke platformreferentie; dezelfde referentie nogmaals geeft dezelfde order terug, nooit een tweede, en de order is op die referentie terug te vinden. Terug komen het ERP-ordernummer en per regel de status.	Elke bestelling uit webshop en garagesoftware. Veilig opnieuw aanbieden na een time-out.

Met alleen het minimum ziet de garage in de shop de eigen voorraad van de grossier en kan hij bestellen; de grossier koppelt elke garage aan een debiteur die het ERP bevestigt. Orderhistorie, facturen en retouren komen dan alleen uit wat het platform zelf weet: bestellingen die via het platform zijn geplaatst, zonder leverstatus uit het ERP.

3.2 Volledig koppelvlak (aanbevolen)

Onderdeel	Wat het ERP levert	Waar het platform het voor gebruikt	Wat de garage mist als het ontbreekt
Vestigingen	Lijst van vestigingen met code en naam.	Voorraad per vestiging tonen; de klant aan een vestiging koppelen.	Voorraad alleen als één getal, geen "op voorraad in uw vestiging".

Onderdeel	Wat het ERP levert	Waar het platform het voor gebruikt	Wat de garage mist als het ontbreekt
Artikelstam en voorraad, bulk	Volledige export en wijzigingen sinds een tijdstip van de artikelstam en van de voorraad per vestiging, gepagineerd. Optioneel ook als bestand op afroep.	Zoeken op artikelnummer, assortiment kennen, een lokale voorraadkopie waarmee de shop filtert op "leverbaar" zonder per artikel te vragen. Vervangt FTP-bestanden.	Geen filter "alleen leverbaar", eigen artikelnummers van de grossier niet vindbaar, bij een ERP-storing geen eigen voorraad.
Orderstatus en historie	Orders per klant, met wijzigingen sinds een tijdstip, inclusief leverstatus en werkelijk geleverd aantal per regel; één order op id; zoeken op ordernummer, artikelnummer, kenteken en tekst.	Orderhistorie in de shop (ook balie- en telefoonorders), backorderoverzicht, vertrekpunt voor een retour.	Geen leverstatus, geen backorders, geen orders die buiten het platform om zijn geplaatst.
Facturen en creditnota's	Facturen én creditnota's per klant, met datum, vervaldatum, bedragen, betaalstatus en, als beschikbaar, het document.	Factuuroverzicht in de shop, "nog te betalen".	Geen facturen in de shop; de garage belt of mailt ervoor.
Retour	Een retouraanmelding registreren op regels van een eerdere order, met reden en aantal, idempotent op platformreferentie; de status terug kunnen lezen.	Retour aanmelden vanuit de orderhistorie.	Geen retour via de shop; de garage meldt een retour telefonisch of per mail.
Levermoment op de order	Het beloofde levermoment, de leverwijze en een eventuele routecode overnemen op de order en teruggeven bij het lezen.	De garage ziet in shop en ERP dezelfde belofte; het ERP kan erop plannen.	Het ERP kent de belofte aan de garage niet; de orderverwerker leest hem hooguit uit de opmerking.

3.3 Uitbreidingen (modules)

Module	Wat het ERP extra levert	Wanneer zinvol
Prijs	Per klant en per artikel de bruto en netto prijs, borg en overige toeslagen en de besteleenheid, in dezelfde aanroep als de artikelinfo.	Als de grossier zijn klantcondities in het ERP beheert en die daar wil houden. Zonder deze module berekent het platform de prijs.
Login	De inlog van een garage (naam + wachtwoord) valideren en de bijbehorende debiteur teruggeven.	Als de grossier wil dat garages met hun ERP-inlog blijven werken. Zonder deze module beheert het platform de inlog.
Klant op debiteurnummer	Een klant opzoeken op het debiteurnummer dat de grossier hanteert.	Voor het aansluiten van bestaande klanten zonder dat de garage zelf hoeft te registreren.

Module	Wat het ERP extra levert	Wanneer zinvol
Klant aanmaken	Een nieuwe debiteur aanmaken vanuit het platform, en een bestaande debiteur achteraf van een platformreferentie voorzien.	Als de grossier zelfregistratie van garages via het platform toestaat.
Artikelen aanleggen	Een artikel aanleggen of bijwerken vanuit het platform: nummer, merk, EAN, omschrijving, leverancier en inkoopprijs; leverancier en inkoopprijs zijn daarna op het artikel terug te lezen.	Als het platform de artikelstam van het ERP aanvult met wat garages bestellen en de grossier niet zelf alles vooraf invoert.
Inkoop	Per leverancier een inkooporder ontvangen, gekoppeld aan de verkooporder, bevestigd of als concept.	Als de grossier wil dat het platform de inkoopkant aanmaakt, omdat het platform de leverancier per regel al kent. Zonder deze module regelt het ERP de inkoop zelf.
Borg	Borg (statiegeld op ruildelen) als eigen regel op order en retour, en het oude deel van een eerder geleverde regel apart kunnen crediteren.	Als de grossier met ruildelen werkt.
Routes	Het routeschema per klant: rittijden, besteldeadline, weekdagen.	Als het ERP de bezorgroutes beheert en het platform die moet volgen in plaats van zelf te berekenen.
Export als bestand	Grote exports van artikelstam en voorraad als job met download-URL.	Voor artikelstammen vanaf enkele honderdduizenden rijen.
Testomgeving	Een reset van de testdata en de handelingen die de acceptatiescenario's in het ERP vragen (leveren, annuleren, retourstatus, voorraad), via het koppelvlak (hoofdstuk 8.4).	Zodat het platform de acceptatiescenario's (hoofdstuk 8.2) altijd vanuit dezelfde beginstand en zonder handwerk kan draaien.

Meldingen van het ERP aan het platform (webhooks) zijn voorgenomen, maar geen onderdeel van deze versie; zie hoofdstuk 17.

3.4 Twee werkwijzen die voorkomen

De uitbreidingen zijn losse schakelaars, maar in de praktijk komen ze in twee bundels voor. Het ERP meldt in `GET /v1/capabilities` niet alleen welke modules het heeft, maar ook per onderwerp wie de bron is (`masterData`: prijs, klanten, artikelen, inlog, elk `erp` of `platform`).

	ERP-beheerd	Platform-beheerd
Wanneer	De grossier werkt al jaren in het ERP: artikelstam, debiteuren en klantcondities staan daar.	Het ERP is nieuw of kaal ingericht; het platform vult aan wat ontbreekt.
Prijs	Uit het ERP (module Prijs).	Door het platform berekend en meegestuurd op de order.

	ERP-beheerd	Platform-beheerd
Klanten	Bestaan in het ERP; het platform koppelt op debiteurnummer (module Klant op debiteurnummer) en laat garages eventueel met hun ERP-inlog werken (module Login).	Het platform legt een ontbrekende debiteur aan met <code>POST /v1/customers</code> (module Klant aanmaken) en plaatst daarna de order op het teruggegeven <code>erpCustomerId</code> . De order zelf maakt nooit een klant aan.
Artikelen	Bestaan in het ERP; een onbekend artikel wordt een regel voor handwerk.	Het platform legt een ontbrekend artikel aan, met leverancier en inkoop prijs (module Artikelen aanleggen), en plaatst daarna de order.
Inkoop	Het ERP koopt zelf in op basis van de order.	Het platform plaatst per leverancier een inkooporder (module Inkoop).
Eigen voorraad	Uit het ERP, live en in bulk.	Uit het ERP, live en in bulk. Ook hier heeft het ERP een artikelstam en voorraad; "kaal" betekent niet leeg.
Routes	Het ERP levert het routeschema per klant als het de routes beheert (module Routes).	Het platform berekent de routes; het beloofde levermoment gaat mee op de order.
Borg	Module Borg als de grossier met ruildelen werkt.	Idem.
Export als bestand	Module Export als bestand bij een grote artikelstam.	Idem.
Testomgeving	Module Testomgeving op de testomgeving.	Idem.

Andere combinaties zijn toegestaan; het platform leest per onderwerp af wat het ERP meldt. De acceptatiescenario's draaien in beide werkwijzen.

Wat het platform aanroept, volgt uit `features` en `modules`; die zijn leidend. `masterData` is beschrijvend: het zegt wie de bron is, niet wat het platform mag aanroepen. Spreken ze elkaar tegen (bijvoorbeeld `masterData.prices: erp` zonder module `pricing`), dan legt het platform dat vast in zijn log en volgt het de modules. Elke sleutel van `features` en `modules` staat in het antwoord, expliciet op `true` of `false`.

4. Wat een specificatie vooraf moet benoemen

Een ERP-API die alleen "artikel opvragen" en "order plaatsen" kent, lijkt compleet maar laat de grossier later met gaten zitten: de garage ziet geen orderhistorie, kan geen retour aanmelden, ziet geen facturen, en de overstap stukt op de inlog. Daarom staan die onderdelen in dit document in het volledige koppelvlak, met per onderdeel wat de garage mist als het ontbreekt. Wie dit koppelvlak bouwt of zijn eigen API erop aanpast, hoort vóór de bouw per onderdeel te weten:

1. welke laag: minimum, volledig koppelvlak of uitbreidingen, en zo ja welke;
2. hoe de garage straks inlogt (platform of ERP);
3. welke identiteiten stabiel zijn (klant-id, artikel-id, ordernummer);
4. hoe dubbele orders worden voorkomen (platformreferentie);

5. hoe wijzigingen worden opgehaald (delta sinds tijdstip);
6. welke testomgeving er is en wanneer.

5. Eisen en vorm

Dit koppelvlak bestaat uit twee dingen die een ERP-leverancier los van elkaar kan beoordelen.

De eisen gelden altijd, ook voor wie de functies in zijn eigen, bestaande API inbouwt. Zonder deze eisen kan het platform niet betrouwbaar koppelen, in welke vorm dan ook:

1. **Idempotent bestellen op de platformreferentie.** Bestaat er al een order met die referentie, dan komt die order terug, ongeacht de inhoud van de nieuwe aanvraag; er ontstaat nooit een tweede. De order is op die referentie terug te vinden, en het ERP legt hem vast vóóordat het antwoordt, zodat terugvinden na een verbroken verbinding altijd lukt. De referentie is uniek per grossier, niet per klant, en het ERP controleert haar vóór elke andere validatie.
2. **Exact matchen van artikelen.** Op ERP-artikel-id, EAN of artikelnummer plus merk; gelijkheid na normalisatie, nooit op een deel van het nummer en nooit op een ander merk dan het meegegeven (hoofdstuk 6).
3. **Stabiele identiteiten** voor klant, artikel, order, factuur, retour en inkooporder (hoofdstuk 6).
4. **Wijzigingen sinds een tijdstip**, met een door het ERP teruggegeven `asOf` als volgend vertrekpunt en een stabiele paginering, voor alles wat het platform spiegelt: artikelstam, voorraad, orders, facturen, retouren.
5. **Vrije voorraad** is aanwezig min gereserveerd, per vestiging (of als één getal bij een ERP zonder vestigingen).
6. **Een order faalt niet op een onbekend artikel.** De regel landt zichtbaar, gemarkeerd voor handwerk.
7. **Prijs en levermoment worden geboekt zoals meegegeven**, tenzij de module Prijs actief is.
8. **Echte foutstatussen.** Een fout komt nooit als geslaagd antwoord terug.

De vorm is de aanbevolen uitwerking: de endpoints, velden en foutcodes uit deel 2 en de OpenAPI-beschrijving. Het verschil tussen de twee wegen:

	Vorm overgenomen	Alleen de eisen, in een eigen API
Koppeling	De generieke koppeling van het platform: basis-URL en sleutel instellen, klaar.	Een koppeling op maat, door het platform te bouwen en te onderhouden.
Doorlooptijd	Acceptatiescenario's (hoofdstuk 8.2) draaien direct tegen de testomgeving.	Eerst een analyse van de API, dan bouwen, dan dezelfde scenario's.
Uitbreidingen later	Een nieuwe module is aan beide kanten een instelling.	Elke uitbreiding is opnieuw maatwerk.
Kosten en planning	Geen bouwkosten voor de koppeling aan platformzijde.	In overleg, afhankelijk van de afwijking.

Kiezen per onderdeel kan: wie bijvoorbeeld alleen orders en facturen in de vorm van dit koppelvlak aanbiedt en de rest in een eigen API, bespaart op precies die onderdelen het maatwerk.

6. Identiteiten en stabiliteit

Het platform bewaart de identiteiten van het ERP en bouwt daarop voort. Ze moeten stabiel zijn: een id mag nooit worden hergebruikt voor iets anders.

Identiteit	Eis
<code>erpCustomerId</code>	Stabiel per debiteur. Mag het debiteurnummer zijn, mits dat nooit wordt hergebruikt.
<code>erpArticleId</code>	Stabiel per artikel. Wordt een artikel vervangen, dan krijgt het nieuwe een nieuw id; het oude blijft opvraagbaar en wijst eventueel naar het nieuwe (<code>replacedBy</code>).
<code>erpOrderId</code> , <code>erpOrderNumber</code>	Het id is de stabiele sleutel; het nummer is wat de klant ziet. Beide zijn uniek.
<code>erpInvoiceId</code> , <code>invoiceNumber</code>	Idem voor facturen en creditnota's.
<code>erpReturnId</code> , <code>erpReturnNumber</code>	Idem voor retouren.
<code>erpPurchaseOrderId</code> , <code>erpPurchaseOrderNumber</code>	Idem voor inkooporders (module Inkoop).

Artikelen worden exact gematcht. "Exact" betekent gelijkheid na normalisatie: alleen letters en cijfers tellen, hoofdletters en kleine letters zijn gelijk. Dat geldt voor artikelnummer, merk en debiteurnummer. Bij een EAN tellen voorloophullenv niet (een EAN-13 en dezelfde code als GTIN-14 met een voorloophul zijn gelijk); het controlecijfer toetst het ERP niet. Een match op een deel van het nummer is nooit toegestaan.

Het ERP matcht in deze volgorde en stopt bij de eerste stap die een uitkomst geeft:

1. **`erpArticleId`**. Is het id bekend, dan wint het, zonder controle van de overige sleutelvelden. Is het onbekend, dan volgt stap 2.
2. **EAN**. Past de EAN op één artikel, dan is dat de match. Past hij op meer artikelen, dan stopt de reeks met `known: false`, `reason: ambiguous`. Past hij op geen artikel, dan volgt stap 3.
3. **Artikelnummer plus merk** (`brand`, of één van de schrijfwijzen in `brands`).
4. **Artikelnummer alleen**, als het nummer binnen het ERP uniek is, en alleen als het platform géén merk meestuurt. Stuurt het platform een merk mee en klopt geen merk, dan valt het ERP niet terug op het nummer alleen: het antwoord is `reason: not_found`, met `candidates` als het nummer bij een ander merk wel bestaat.

Inactieve artikelen tellen mee voor de uniciteit: ze blijven opvraagbaar en zijn dus kandidaat. Een EAN met een fout formaat negeert het ERP als er een ander bruikbaar sleutelveld is; `invalid_key` geldt alleen als geen enkel sleutelveld bruikbaar is. Levert een sleutel meerdere artikelen op, dan is het artikel niet bekend (`known: false`, `reason: ambiguous`), eventueel met de kandidaten erbij. Het antwoord zegt met `matchedOn` waarop gematcht is, zodat het platform een verouderd id herkent.

7. Tijd, taal en bedragen

- Tijdstippen in ISO 8601, op milliseconden. Het platform stuurt met tijdzone-offset of in UTC; het ERP antwoordt in UTC (`2026-10-21T11:15:00.000Z`). Een tijdstip zonder offset, in de URL of in de

body, is 400 `validation_error`.

- In een URL codeert het platform `+` als `%2B`. Komt er toch een spatie op de plek van de offset aan (een niet-gecodeerde `+`), dan mag het ERP die als `+` lezen.
- Datumvelden (`invoiceDate`, `dueDate`) zijn kalenderdata zoals ze op het document staan, zonder omrekening naar een tijdzone.
- Kloktijden in het routeschema (module Routes) zijn lokale tijd in Europe/Amsterdam.
- Bedragen als decimaal getal in euro, exclusief btw, tenzij het veld anders zegt. Btw-percentages apart.
- Teksten in UTF-8. Omschrijvingen mogen Nederlands zijn; codes en enumeraties zijn Engels en staan vast in de OpenAPI.
- Een veld met `null` en een weggelaten veld zijn gelijkwaardig: het platform behandelt beide als afwezig, ook waar afwezigheid betekenis heeft (`stock` ontbreekt = voorraad onbekend, `nextCursor` ontbreekt = laatste pagina).

8. Testomgeving en acceptatie

8.1 Wat het platform van een testomgeving verwacht

- Dezelfde API als productie, op een eigen basis-URL en met een eigen sleutel, ten minste een maand vóór de livegang beschikbaar en daarna blijvend.
- Testdata: ten minste twee vestigingen; één testklant met bekend `erpCustomerId`; vijftig of meer artikelen waarvan sommige met en sommige zonder voorraad; voor de testklant enkele orders in verschillende leverstatussen, ten minste één factuur en één creditnota.
- Basis-URL, sleutel en het `erpCustomerId` van de testklant geeft de ERP-leverancier per e-mail door of zet ze in de README van de testomgeving.
- Geen uitwerking naar buiten: orders in test leiden niet tot echte inkoop, levering of mail aan de garage.
- Een manier om testorders op te ruimen of een testklant te verversen, en om de handelingen te doen die de scenario's in het ERP vragen (een voorraadmutatie, een order deels leveren of annuleren, een retour van status laten wisselen). Bij voorkeur de module Testomgeving (8.4), zodat het platform de acceptatiescenario's zelf, steeds vanuit dezelfde beginstand, kan draaien. Module `testing` meldt het ERP alleen met `environment: test`.
- Een wijzigingslog en een versienummer in `GET /v1/capabilities`, zodat het platform ziet met welke versie het praat.

8.2 Acceptatiescenario's

Het platform draait deze scenario's tegen de testomgeving voordat een grossier live gaat. Per scenario staat in de OpenAPI welk endpoint het raakt.

Nr	Scenario	Verwacht resultaat
A1	Artikelinfo voor 500 sleutels, gemengd bekend en onbekend.	Antwoord binnen het tijdsbudget uit 8.3: onder 3 seconden voor 500 sleutels (onder 1 seconde voor 100); onbekende sleutels komen terug met <code>known: false</code> , bekende met voorraad per vestiging.
A2	Bulk export artikelstam en voorraad, daarna delta na een voorraadmutatie (PUT <code>/v1/test/stock/{erpArticleId}</code>).	Export compleet en consistent; delta bevat precies de gewijzigde rijen; verwijderde artikelen komen als inactief terug.
A3	Order plaatsen, daarna nogmaals met dezelfde platformreferentie, ook met een gewijzigd levermoment.	Eerste keer 201 met ordernummer; tweede keer 200 met dezelfde, ongewijzigde order; er bestaat één order.
A4	Order plaatsen waarvan de verbinding wegvalt na het versturen (het platform verbreekt hem zelf); daarna order opzoeken op platformreferentie.	De order is direct te vinden en bevat alle regels; het ERP had hem vastgelegd vóór het antwoord.
A5	Order met een regel voor een artikel dat het ERP niet kent.	De order wordt aangemaakt; de regel komt terug met status <code>manual</code> en staat zichtbaar op de order.
A6	Order met voertuig per regel, klantreferentie, opmerking, beloofd levermoment en betaalwijze, en een verzendtoeslag op orderniveau.	Alle gegevens zijn terug te lezen op de order; de toeslag staat in <code>surcharges</code> op de order, niet als artikelregel, en telt mee in <code>totalNet</code> en <code>totalGross</code> .
A7	Orders ophalen met <code>updatedSince</code> , vervolgens een order in het ERP deels leveren (POST <code>/v1/test/orders/{erpOrderId}/delivery</code>) en opnieuw ophalen.	De order komt in de tweede delta terug met <code>deliveryStatus: partial</code> , <code>statusCode: processing</code> en het geleverde aantal per regel.
A8	Zoeken in orders op ordernummer, artikelnummer, kenteken en tekst.	Alleen orders van de opgegeven klant; de filters werken op regelniveau en geven de hele order terug.
A9	Facturen ophalen, met en zonder <code>unpaidOnly</code> .	Facturen én creditnota's; creditnota herkenbaar aan <code>type</code> ; per rij <code>amountOpen</code> , bij een betaalde factuur 0.

Nr	Scenario	Verwacht resultaat
A10	Retour aanmelden op een regel van een geleverde order; nogmaals met dezelfde referentie; daarna de status wijzigen (<code>POST /v1/test/returns/{erpReturnId}/status</code>).	Eén retour; tweede aanroep geeft dezelfde retour terug; de nieuwe status is opvraagbaar en komt in de delta.
A11	Aanroep met ongeldige sleutel; aanroep met onjuiste body.	401 respectievelijk 400/422 met een <code>problem+json</code> -body; nooit een 200 met een foutmelding erin.
A12	Pagineren doorlopen tot het einde op elke lijst.	Geen dubbele en geen ontbrekende rijen; laatste pagina zonder <code>nextCursor</code> .
A13 (module Prijs)	Artikelinfo met klant: prijs per artikel.	Bruto, netto en toeslagen per artikel; dezelfde prijs wordt op de order geaccepteerd.
A14 (module Inkoop)	Inkooporder plaatsen gekoppeld aan een verkooporder; nogmaals met dezelfde referentie.	Eén inkooporder per platformreferentie; concept of bevestigd zoals gevraagd; per regel het regelnummer van de verkooporder terug.
A15 (module Artikelen aanleggen)	Artikel aanleggen met leverancier en inkoopprijs; daarna opvragen en bestellen met herkomst leverancier.	Het artikel is bekend, leverancier en inkoopprijs staan erbij (<code>suppliers</code>), de orderregel is <code>accepted</code> .
A16	Order met een cartregel die over twee herkomsten is gesplitst (eigen voorraad en leverancier), met dezelfde <code>platformLineRef</code> .	Beide regels zijn terug te lezen met die referentie.
A17 (keten)	Eén klant en één order door de hele keten: klant lezen, artikelinfo voor twee artikelen, order plaatsen met voertuig en klantreferentie, de order terugvinden op platformreferentie en in de historie van die klant, de levering (<code>POST /v1/test/orders/{erpOrderId}/delivery</code>), delta met <code>updatedSince</code> , factuur voor die order, retour op een geleverde regel, retourstatus naar <code>received</code> , retour en order opnieuw lezen.	Elke stap gebruikt de identiteiten uit de vorige stap zonder vertaling: hetzelfde <code>erpCustomerId</code> , <code>erpOrderId</code> , <code>erpOrderNumber</code> , <code>orderLineNo</code> , <code>platformLineRef</code> , <code>erpInvoiceId</code> en <code>erpReturnId</code> komen in elk antwoord gelijk terug; de factuur verwijst naar de order, de retour naar de orderregel. Loopt per werkwijze (3.4) en met de modules die het ERP meldt.

8.3 Prestaties en beschikbaarheid

Onderwerp	Verwachting
Time-out	Het platform breekt elke aanroep af na 10 seconden.
Artikelinfo live	Het drukste pad: elke artikelpagina doet één aanroep. Streefwaarde onder 1 seconde voor 100 sleutels, onder 3 seconden voor 500.
Order plaatsen	Onder 5 seconden. Het ERP mag de inkoopkant asynchroon afhandelen.
Bulk	Pagina's tot 5.000 rijen voor voorraad en 1.000 voor artikelstam, elk onder 10 seconden.
Beschikbaarheid	Bij fouten schakelt het platform het leespad tijdelijk uit (na drie fouten op rij 45 seconden geen leescalls) en zet orders in een wachtrij. Een onderhoudsvenster graag vooraf gemeld.
Limieten	Als het ERP aanroepen begrenst, dan met 429 en <code>Retry-After</code> ; het platform respecteert dat.

8.4 Beheerhandelingen in de testomgeving (module Testomgeving)

Scenario's A2, A7 en A10 vragen een handeling in het ERP. Met module `testing` biedt het ERP die handelingen in het koppelvlak aan, onder `/v1/test/`, zodat het platform alle scenario's zonder handwerk tegen de testomgeving van elk ERP kan draaien. Op productie antwoorden deze endpoints 501 `module_disabled`.

Endpoint	Wat het doet	Scenario
<code>POST /v1/test/reset</code>	Zet alle testdata terug naar de beginstand. Cursors van vóór de reset mogen 400 <code>invalid_cursor</code> geven.	Alle
<code>PUT /v1/test/stock/{erpArticleId}</code>	Zet per vestiging de aanwezige en de gereserveerde voorraad (en eventueel <code>expectedAt</code>) van een artikel.	A2
<code>POST /v1/test/orders/{erpOrderId}/delivery</code>	Levert een order (deels): zet per regel het geleverde aantal; het ERP leidt daaruit de leverstatus en <code>statusCode</code> af, of neemt een meegegeven <code>deliveryStatus</code> over.	A7
<code>POST /v1/test/orders/{erpOrderId}/cancel</code>	Annuleert een order; daarna <code>deliveryStatus</code> en <code>statusCode</code> <code>cancelled</code> .	Annulering lezen
<code>POST /v1/test/returns/{erpReturnId}/status</code>	Zet de status van een retour volgens de toegestane overgangen (16), bij <code>credited</code> met het gecrediteerde bedrag.	A10

Elke handeling werkt `updatedAt` bij zoals een gewone mutatie in het ERP, zodat het resultaat in de eerstvolgende delta staat.

9. Publicatie en versies

Dit document en de OpenAPI-beschrijving zijn openbaar en staan op grossier.nu/erp-koppelvlak, met versienummer en een wijzigingslog per versie (bijlage C). De status is concept zolang het

versienummer onder 1.0 ligt: de inhoud kan nog wijzigen op basis van wat ERP-leveranciers terugmelden, en een conceptversie kan ook een eis aanscherpen; bijlage C noemt zulke wijzigingen. Vanaf versie 1.0 zijn wijzigingen binnen /v1 alleen toevoegend (nieuwe optionele velden, nieuwe modules); een niet-compatibele wijziging krijgt dan een nieuw basispad.

Vragen en terugmeldingen over het koppelvlak gaan naar support@grossier.nu.

Deel 2 — voor de bouwer

De normatieve beschrijving van endpoints, velden en foutcodes staat in `openapi.yaml`. Dit deel licht de keuzes toe die de OpenAPI niet kan uitdrukken.

10. Transport, toegang en algemene regels

Onderwerp	Regel
Protocol	HTTPS (TLS 1.2 of hoger), JSON (<code>application/json; charset=utf-8</code>). Basispad <code>/v1</code> .
Toegang	<code>Authorization: Bearer <sleutel></code> . Eén sleutel per grossier per omgeving. Twee sleutels mogen tegelijk geldig zijn voor rotatie. Geen sessies, geen garage-inlog in de aanroep. Een 401 draagt de header <code>WWW-Authenticate: Bearer</code> . 403 geeft het ERP alleen als het rechten per sleutel kent. De sleutel geldt voor de hele grossier: elke klant en elke order van die grossier is ermee te lezen, en het platform controleert zelf of een order bij de klant hoort.
Correlatie	Het platform stuurt <code>X-Request-Id</code> mee; het ERP neemt die over in het antwoord en in zijn logging. Ontbreekt de header, dan maakt het ERP zelf een id; is hij langer dan 64 tekens, dan neemt het ERP hem toch over.
Fouten	Altijd een echte HTTP-statuscode met een body volgens RFC 9457 (<code>application/problem+json</code>) en een vaste <code>code</code> uit de OpenAPI. Een fout komt nooit als 200 terug. <code>Problem.type</code> is vrij; het platform stuurt op <code>code</code> . Statussen buiten tabel 18: 405, 413 en 415 met <code>validation_error</code> ; een onbekend pad is 404 <code>not_found</code> . 503 geldt ook voor <code>GET /v1/capabilities</code> .
Paginerig	Cursor-gebaseerd: <code>limit</code> en <code>cursor</code> in de aanvraag, <code>nextCursor</code> in het antwoord; ontbreekt <code>nextCursor</code> , dan is dit de laatste pagina. Sortering is stabiel (zie onder). De cursor draagt de positie en de <code>asOf</code> van de eerste pagina en is ten minste 24 uur geldig; de filters stuurt het platform bij elke pagina opnieuw mee. Wijken die af van de eerste pagina, dan mag het ERP ze negeren of 400 <code>invalid_cursor</code> geven. Een <code>limit</code> boven het maximum wordt afgekapt; een <code>limit</code> onder 1 of die geen getal is, is 400 <code>validation_error</code> ; een ongeldige of verlopen cursor is 400 <code>invalid_cursor</code> . <code>totalCount</code> komt op elke pagina mee.
Sortering	Bij een delta altijd oplopend op <code>updatedAt</code> , daarna id. Zonder delta en zonder <code>sort</code> : orders en retouren aflopend op aanmaakmoment, facturen aflopend op factuurdatum, daarna id.
Delta	<code>updatedSince</code> filtert inclusief op de wijzigingstijd van het ERP. Het antwoord geeft <code>asOf</code> terug: de servertijd waarop de reeks is bepaald, op elke pagina gelijk, die het platform als volgende <code>updatedSince</code> gebruikt. Zo speelt klokverschil geen rol. Rijen die exact op <code>updatedSince</code> liggen komen opnieuw; het platform verwerkt idempotent. Rijen die tijdens het doorlopen wijzigen, komen in de volgende delta. Delta en zoekfilters mogen samen; alle filters gelden dan tegelijk.

Onderwerp	Regel
Filterwaarden	Een onbekende <code>branchCode</code> als filter is 400 <code>validation_error</code> ; een onbekende <code>erpCustomerId</code> als filter is 404 <code>not_found</code> .
Verloren antwoord	Krijgt het platform op een <code>POST</code> die een object aanmaakt (order, retour, klant, inkooporder) een time-out, netwerkfout of 5xx, dan zoekt het eerst terug op <code>platformReference</code> en biedt het pas daarna opnieuw aan. Het ERP legt vast vóór het antwoordt, dus ook na een 5xx kan het object bestaan.
Onbekende velden	Het ERP negeert velden die het niet kent en voegt zelf alleen velden toe die in de OpenAPI staan of met <code>x-</code> beginnen.
Capabilities	<code>GET /v1/capabilities</code> geeft de versie van het koppelvlak en van de implementatie, de aanwezige onderdelen van het volledige koppelvlak (<code>features</code>), de uitbreidingen (<code>modules</code>) en per onderwerp wie de bron is (<code>masterData</code>). <code>features</code> en <code>modules</code> zijn leidend (3.4). Het platform roept een onderdeel niet aan als het ERP het niet meldt; roept het er toch een aan, dan antwoordt het ERP 501 <code>module_disabled</code> .

Wat onder welke capability valt, waar dat niet uit de naam volgt:

- `GET /v1/orders` met alleen `platformReference` hoort bij het minimum; elke andere combinatie van filters, en `GET /v1/orders/{erpOrderId}`, vraagt feature `orders.history`.
- `GET /v1/customers/{erpCustomerId}` hoort bij het minimum. Feature `customers.list` dekt de filters `platformReference`, `email` en `updatedSince` op `GET /v1/customers`; `debtorNumber` is module Klant op debiteurnummer.
- `GET /v1/articles/{erpArticleId}` valt onder feature `articlesBulk`.
- Het blok `delivery` op de order is altijd verplicht (minstens `method`), ook voor een ERP met alleen het minimum. Feature `delivery` betekent dat het ERP het beloofde levermoment (`promisedAt` op de order en, als meegegeven, per regel) honoreert en bij het lezen teruggeeft.
- Zonder feature `branches` kent het ERP geen vestigingen: voorraad komt als één getal zonder `branchCode` (11).
- Feature `invoiceDocuments` dekt de documentlinks en de document-endpoints van facturen en orders (15).
- Module `testing` meldt het ERP alleen met `environment: test`.

11. Artikelinfo live (`POST /v1/articles/lookup`)

Het platform stuurt tot 500 sleutels. Elke sleutel heeft een `requestId` die het ERP ongewijzigd terugstuurt, zodat de volgorde of het aantal antwoorden niet hoeft te kloppen met de aanvraag. Per sleutel komt één antwoord: `known: false`, of `known: true` met het artikel en de vrije voorraad per vestiging.

Matchen gaat volgens hoofdstuk 6; het antwoord zegt met `matchedOn` waarop gematcht is. Het platform mag voor het merk meerdere schrijfwijzen sturen (`brands`); één hoeft te matchen. Een fout in één sleutel raakt alleen die sleutel (`known: false` met een `reason`), nooit de hele aanroep: dit is het drukste pad van de shop. Uitzondering is wat het antwoord per sleutel onmogelijk maakt: een sleutel zonder `requestId`, een `requestId` die meer dan eens voorkomt, of meer dan 500 sleutels. Dan is de hele aanroep 400 `validation_error`. Een inactief artikel komt terug als bekend, met `active: false` en zijn opvolger.

Vrije voorraad is wat nu nog verkoopbaar is: fysiek aanwezig min al gereserveerd voor andere orders. Het ERP noemt altijd alle vestigingen, ook met 0. Het platform gebruikt dit twee keer: bij het tonen, en vlak vóór het verzenden van de order om te controleren of eigen voorraad er nog is. In dat laatste geval geldt: ontbreekt het blok `stock`, dan is de voorraad onbekend, niet nul.

Welke vestigingen tellen:

- Alleen vestigingen uit `GET /v1/branches` tellen. Voorraad op een vestigingscode die daar niet in staat, negeert het platform.
- De vestiging van de klant (`Customer.branchCode`) toont de shop als "uw vestiging"; de andere vestigingen samen als "elders bij de grossier".
- Of en wanneer voorraad van een andere vestiging leverbaar is, bepaalt het platform met zijn routes, niet het ERP.
- Een ERP zonder vestigingen (feature `branches` `false`) levert per artikel één voorraadgetal zonder `branchCode`. De shop toont dat als "bij de grossier".

Met module Prijs stuurt het platform `erpCustomerId` mee en geeft het ERP per artikel het blok `price` terug: netto exclusief toeslagen, toeslagen apart met hun btw-tarief, optioneel staffels, en `validUntil` als de termijn waarbinnen het ERP deze prijs op een order accepteert. `net` is de prijs bij het gevraagde aantal (`quantity`); de staffels beginnen bij aantal 1. Een geblokkeerde klant krijgt gewoon een prijs; voor een inactief artikel komt geen `price`-blok. Is de klant onbekend, dan komt de artikelinfo zonder `price` terug, met een waarschuwing. Zonder de module laat het ERP `price` weg en negeert het een meegestuurde `erpCustomerId`, zonder waarschuwing; het platform berekent dan zelf.

12. Bulk: artikelstam en voorraad

Twee gepagineerde lijsten vormen de basis:

- `GET /v1/articles` levert de artikelstam: `erpArticleId`, artikelnummer, merk, EAN, omschrijving, besteleenheid, actief/inactief, `replacedBy`, `updatedAt`, en als het ERP ze kent het btw-tarief, de bruto adviesprijs en de leveranciers met inkoopprijs (`suppliers`). Zonder `updatedSince` de volledige set.
- `GET /v1/stock` levert de voorraad per vestiging: `erpArticleId`, artikelnummer, merk, EAN, vestiging, vrije voorraad, `updatedAt`. Zonder `updatedSince` de volledige set; een artikel zonder voorraad komt met `quantityFree: 0` terug zodat de kopie kan worden bijgewerkt. Een voorraadrij krijgt een nieuwe `updatedAt` bij elke mutatie van de aanwezige of de gereserveerde voorraad of van `expectedAt`. Rijen van een inactief artikel blijven staan. Wijzigingen in nummer, merk of EAN komen uit `/v1/articles`. Welke vestigingen tellen, staat in 11; zonder vestigingen is er één rij per artikel, zonder `branchCode`.

Waarom gepagineerd en niet alleen een bestand: dezelfde toegang, dezelfde velden en dezelfde foutafhandeling als de rest van het koppelvlak, zonder FTP-accounts, bestandsformaten en bestandsnamen die per grossier anders zijn. Een delta van een kwartier is dan klein en snel; het platform haalt de voorraad zo om de tien minuten op.

Voor artikelstammen vanaf enkele honderdduizenden rijen is de module Export als bestand bedoeld: `POST /v1/exports` start een job, `GET /v1/exports/{id}` geeft de status en een download-URL naar een gzip-gecomprimeerd NDJSON-bestand met exact dezelfde rijen als de gepagineerde lijsten. De download-URL is zonder sleutel op te halen (hij kan op een opslagdienst staan) en ten minste een uur

geldig; de job is ten minste 24 uur op te vragen. Het platform vraagt de status op met een oplopende tussenpoos tot 5 seconden. Dat is een toevoeging; de gepagineerde delta blijft ook dan beschikbaar.

Met module Artikelen aanleggen kan het platform de stam ook vullen (`POST /v1/articles`): een artikel met nummer, merk, EAN, omschrijving en per leverancier de inkoopprijs, aanleggen of bijwerken. Wat het platform aanlevert, is daarna terug te lezen op het artikel. Voor een bestaand artikel geldt:

- Het ERP zoekt het artikel volgens hoofdstuk 6. Een meegegeven `erpArticleId` dat het ERP niet kent is 404 `not_found`; een sleutel die op meer artikelen past is 422 `business_rule`.
- `articleNumber`, `brand` en `ean` blijven ongemoeid: de schrijfwijze van de grossier wint. De overige meegegeven velden worden bijgewerkt; een veld met `null` of zonder waarde wijzigt niets.
- `suppliers` wordt per `supplierCode` bijgewerkt of aangevuld; een leverancier die niet in de aanvraag staat, blijft staan.
- Een inactief artikel wordt weer actief.
- Een inkoopprijs van 0 of lager is 400 `validation_error`. Een onbekende `supplierCode` bewaart het ERP zoals meegegeven: dat is de naam zoals het platform de leverancier kent (17, Inkoop).

De artikelstam uit het ERP is níet de catalogus. Het platform gebruikt hem om te weten wat de grossier zelf voert (assortiment, eigen artikelnummers, EAN-koppeling) en om op artikelnummer te kunnen zoeken. Voertuigtoepassingen, OE-nummers en productgegevens komen uit de catalogi van het platform.

13. Order plaatsen (`POST /v1/orders`)

Een order draagt alles wat het ERP nodig heeft om hem te verwerken zonder navraag:

- `platformReference`: uniek per bestelling, max 64 tekens. Dit is de idempotentiesleutel, uniek per grossier (niet per klant). Het ERP controleert de referentie vóór elke andere validatie: bestaat er al een order met deze referentie, dan antwoordt het ERP 200 met die bestaande order in zijn huidige stand, ongeacht de inhoud van de nieuwe aanvraag, ook als die nu ongeldig zou zijn; het ERP vergelijkt niet. Een order op platformreferentie moet ook via `GET /v1/orders?platformReference=` te vinden zijn, voor het geval het antwoord op de `POST` verloren ging; daarom legt het ERP de order vast vóór het antwoordt. Bij een time-out of 5xx zoekt het platform eerst terug en biedt het pas daarna opnieuw aan (10).
- `erpCustomerId`, `customerReference` (referentie van de garage, bijvoorbeeld een werkordernummer), `remark`. De klant bestaat al: in de platform-beheerde werkwijze legt het platform hem eerst aan met `POST /v1/customers` (3.4).
- `orderedAt`: het tijdstip waarop de garage bestelde, zoals het platform het meegeeft; het ERP boekt dat tijdstip. `updatedAt` is de eigen tijd van het ERP.
- `source`: `webshop` (de garage in de shop), `integrator` (vanuit garagesoftware) of `platform` (een order die de grossier in het platform namens de garage invoert).
- `delivery`: leverwijze (`route`, `pickup`, `shipping`) met daarnaast de eigen code van de grossier (`methodCode`), beloofd levermoment en einde van het bezorgvenster, `routeCode`, en de vestiging die levert. Het blok is altijd verplicht, met minstens `method`. Het levermoment op de kop geldt voor elke regel zonder eigen levermoment.
- `payment`: betaalwijze (`account`, `online`, `cash`) en bij online betaling de betaalreferentie van de betaalprovider.

- `lines`: per regel een `lineNo`, een `platformLineRef` (de regel in het platform; meerdere orderregels mogen dezelfde dragen als het platform een regel over herkomsten splitst), de artikelsleutel, omschrijving, aantal (altijd positief; een retour gaat via `/v1/returns`), de netto stukprijs die de garage zag, btw-percentage (ontbreekt het, dan geldt het tarief van het artikel), voertuig (kenteken of VIN, catalogus, typecode, omschrijving), herkomst (`ownStock` met vestiging, of `supplier` met leverancierscode en verwachte aankomst), toeslagen per stuk (borg en overige), `recognized`, en een eigen beloofd levermoment (`promisedAt`) als dat afwijkt van de kop.
- `surcharges`: toeslagen op orderniveau, optioneel: verzendkosten (`shipping`), kleine-orderkosten (`smallOrder`), bezorgtoeslag (`delivery`) of overig (`other`), elk met een bedrag voor de hele order exclusief btw (`amountNet`) en eventueel code, omschrijving en btw-tarief.

De ERP-regels:

- De order wordt direct bevestigd aangemaakt, niet als offerte. Terug komen `erpOrderId`, `erpOrderNumber`, de orderstatus `statusCode` (14; na het plaatsen ten minste `confirmed`), een leesbare `status` (vrije tekst in de taal van het ERP) en per regel een regelstatus (zie onder).
- Een onbekende klant in de body is 404 `not_found`; een onbekende vestiging in `delivery.branchCode` of `source.branchCode` is 400 `validation_error` (tabel 18 is leidend). Een geblokkeerde klant is 422 `customer_blocked`.
- Kredietlimiet: het ERP neemt een order boven de limiet aan en mag de regels op `manual` zetten, met `reason`. Weigert het de order om krediet, dan is dat 422 `business_rule` met de reden in `detail`.
- Eigen voorraad wordt gereserveerd op `source.branchCode`, anders `delivery.branchCode`, anders de vestiging van de klant.
- Het ERP boekt de meegegeven prijs. Met module Prijs vergelijkt het ERP de meegegeven stukprijs exact, op twee decimalen, met zijn actuele berekening bij het bestelde aantal; wijkt die af, dan mag het ERP de prijs corrigeren naar zijn eigen berekening en meldt het dat op de regel met `priceCorrected: true` en `requestedUnitPriceNet`. Bij `payment.method: online` is er al afgerekend: dan boekt het ERP de meegegeven prijs en corrigeert het niet. Toeslagen, per regel en op orderniveau, boekt het ERP altijd zoals meegegeven; die corrigeert het nooit.
- Toeslagen op orderniveau legt het ERP vast zoals het wil (als eigen kostenregel of als veld op de kop) en geeft ze bij het lezen terug in `surcharges` op de order; die lijst is er altijd, ook leeg. Het platform stuurt verzendkosten, kleine-orderkosten en bezorgtoeslag nooit als artikelregel; een ERP maakt er dus nooit een regel met status `manual` van.
- Levermoment per herkomst. Het routeschema (module Routes, of de routes die het platform berekent) geldt voor regels uit eigen voorraad (`source.kind: ownStock`). Voor een regel die de grossier eerst bij een leverancier haalt, bepaalt het platform het levermoment zelf: uit `expectedAtWholesaler` als het ERP die geeft, anders uit de levertijd van de leverancier. Wijkt het levermoment van een regel af van de kop, dan stuurt het platform het per regel mee (`promisedAt` op de regel); dat gaat voor die regel boven `delivery.promisedAt`. Het ERP neemt het levermoment per regel over zoals meegegeven en geeft het bij het lezen terug.
- Een onbekend artikel is geen reden om de order af te wijzen. De regel landt met de meegegeven omschrijving en prijs en status `manual`, zodat de orderverwerker hem ziet. Met module Artikelen aanleggen legt het platform het artikel vóór de order aan; de regel volgt daarna de gewone regelstatussen. Omdat een nieuw artikel geen eigen voorraad heeft, is dat `accepted` bij herkomst `supplier` en anders `backorder`.

- Regels staan per voertuig gegroepeerd; het ERP mag de voertuiggegevens als kopregel of als regelvelden tonen, maar bewaart ze per regel en geeft ze bij het lezen per regel terug. Ook `platformLineRef`, `herkomst`, `recognized` en opmerking zijn per regel terug te lezen.
- `totalNet` en `totalGross` gaan over het bestelde aantal, inclusief toeslagen per regel en op orderniveau, zonder regels met status `rejected`.
- Het ERP maakt géén eigen inkooporder aan als module Inkoop actief is; dan doet het platform dat. Zonder die module is inkoop de zaak van het ERP.
- Annuleren of wijzigen van een order via het koppelvlak is voorzien maar in deze versie niet gespecificeerd; het platform leest een annulering als `deliveryStatus: cancelled`. In de testomgeving is een annulering na te bootsen met module Testomgeving (8.4).

De regelstatussen:

Status	Wanneer	Velden
<code>accepted</code>	Aangenomen en leverbaar: uit eigen voorraad gereserveerd, of herkomst <code>supplier</code> (dan zonder reservering op eigen voorraad).	<code>expectedAt</code> bij herkomst leverancier als het ERP een datum kent.
<code>backorder</code>	Aangenomen, maar (deels) niet uit eigen voorraad leverbaar. Het beschikbare deel wordt gereserveerd.	<code>quantityBackorder</code> is het niet gereserveerde deel; <code>expectedAt</code> als het ERP een datum kent.
<code>manual</code>	Een mens moet kijken: artikel onbekend, dubbelzinnig of inactief, aantal geen veelvoud van de besteleenheid, <code>recognized: false</code> (ook als het ERP het artikel wel kent), of de order gaat boven de kredietlimiet. Verzendkosten, kleine-orderkosten en bezorgtoeslag zijn nooit een regel en dus nooit <code>manual</code> ; die staan in <code>surcharges</code> op de order.	<code>reason</code> .
<code>rejected</code>	Het ERP weigert de regel definitief: het artikel is geblokkeerd voor verkoop of niet leverbaar aan deze klant.	<code>reason</code> , altijd.

14. Orderstatus en historie (GET /v1/orders)

Het platform haalt periodiek de wijzigingen op (`updatedSince`), over alle klanten tegelijk of per klant, en leest daarnaast live als de garage zoekt. Bij een delta is de sortering oplopend op wijzigingstijd en id, zodat een afgebroken run veilig kan hervatten; bij zoeken kiest het platform de sortering (`sort`, standaard aflopend op `orderedAt`). Per order komen de orderstatus (`statusCode`), de genormaliseerde leverstatus (`pending`, `partial`, `full`, `cancelled`), bedragen, de toeslagen op orderniveau, de platformreferentie als die er is, documentlinks (pakbon, factuur; zie 15) en per regel het bestelde én het werkelijk geleverde aantal. Ook orders die niet via het platform zijn geplaatst horen erin: zo ziet de garage zijn volledige historie.

De orderstatus `statusCode` is een vaste lijst:

<code>statusCode</code>	Betekenis	<code>deliveryStatus</code>
<code>open</code>	Ontvangen, nog niet bevestigd.	<code>pending</code>
<code>confirmed</code>	Bevestigd, nog niets geleverd.	<code>pending</code>

statusCode	Betekenis	deliveryStatus
processing	In behandeling of deels geleverd.	pending of partial
delivered	Volledig geleverd.	full
invoiced	Gefactureerd.	full
cancelled	Geannuleerd.	cancelled

- Een order via het platform is direct bevestigd (13); `open` komt alleen voor bij orders die buiten het platform om binnenkomen en nog niet bevestigd zijn.
- Een order die deels gefactureerd is en nog niet volledig geleverd, is `processing`.
- Regels met status `rejected` tellen niet mee, niet voor `statusCode` en niet voor `deliveryStatus`: een order waarvan alle overige regels geleverd zijn, is `delivered` en `full`. Zijn alle regels `rejected`, dan is de order `cancelled`.
- `status` blijft vrije tekst in de taal van het ERP, alleen als toelichting. De shop toont `statusCode` (vertaald) en `deliveryStatus`; `status` hoogstens als toelichting bij de muisaanwijzer.

De zoekfilters:

Filter	Werking
<code>orderNumber</code>	Exact op het ordernummer, na normalisatie (6).
<code>articleNumber</code>	Exact op het artikelnummer van een regel, na normalisatie.
<code>vehicleRegistration</code>	Kenteken of VIN van een regel, na normalisatie.
<code>text</code>	Deel van de tekst, hoofdletterongevoelig, in ordernummer, referenties (<code>customerReference</code> , <code>platformReference</code>), opmerkingen op order en regel, en omschrijvingen van regel en voertuig.
<code>deliveryStatus</code>	Eén of meer leverstatussen.

Een filter op regelniveau geeft de hele order terug, met alle regels. Filters en `updatedSince` mogen samen. Een onbekende `erpCustomerId` als filter is 404 `not_found`.

`GET /v1/orders/{erpOrderId}` geeft elke order van de grossier, ongeacht de klant: de sleutel is per grossier, en het platform controleert zelf of de order bij de klant hoort die erom vraagt.

Retouren die via het platform zijn aangemeld staan altijd onder `/v1/returns`, nooit als order. `isCredit: true` is voor creditorders die het ERP zelf aanmaakt; die mogen in de orderlijst.

15. Facturen en documenten (`GET /v1/invoices`)

Per klant, met `updatedSince` en `unpaidOnly`. Elke rij heeft een `type` (`invoice` of `creditNote`), nummer, datum, vervaldatum, bedragen, `paymentStatus` (`open`, `partial`, `paid`), de ordernummers waarop hij betrekking heeft en, als beschikbaar, een `documentUrl` naar de PDF. `invoiceDate` en `dueDate` zijn kalenderdata zoals ze op het document staan (7); `dueDate` is de vervaldatum, niet de datum waarop betaald is.

Bedrag	Wat het is
amountNet	Totaal exclusief btw.
amountGross	Totaal inclusief btw.
amountPaid	Wat betaald is, inclusief btw: dezelfde grondslag als amountGross.
amountOpen	Wat nog openstaat, inclusief btw, zoals het ERP het boekt. Bij een factuur is dat amountGross - amountPaid.

Bij een creditnota zijn amountNet en amountGross negatief en is amountPaid 0; amountOpen is negatief zolang de creditnota niet verrekend is, en daarna 0. Een open creditnota telt mee in unpaidOnly. De shop toont amountGross als bedrag, amountOpen als openstaand, paymentStatus als status en dueDate als vervaldatum.

Voor alle documentlinks (Invoice.documentUrl, Order.documents.packingSlipUrl en Order.documents.invoiceUrl) geldt één regel:

- De link is voor het platform, niet voor de garage. Het platform haalt het document op en geeft het zelf aan de garage door. De link hoeft alleen vanaf het platform bereikbaar te zijn, niet vanaf internet; het ERP mag er niet van uitgaan dat de eindgebruiker hem opent.
- De link is zonder sleutel op te halen; hij kan bijvoorbeeld op een opslagdienst staan.
- De link is tijdgebonden en ten minste een uur geldig.
- Het ERP geeft bij elke aanroep een nieuwe link uit. Het platform bewaart links niet, maar vraagt ze op als het een document toont.

Daarnaast levert het ERP elk document met de sleutel: GET /v1/invoices/{erpInvoiceId}/document voor een factuur of creditnota, en GET /v1/orders/{erpOrderId}/documents/{kind} voor de documenten van een order (kind is packingSlip of invoice). Ook deze zijn alleen voor het platform. Bestaat het document (nog) niet, dan is dat 404 not_found. Zo kan het platform een document ophalen zonder van de levensduur van een link af te hangen. Beide vallen onder feature invoiceDocuments.

16. Retour (POST /v1/returns, GET /v1/returns/{id})

Een retour verwijst per regel (lineNo is het nummer van de retourregel) altijd naar een regel van een eerdere order: met erpOrderId of erpOrderNumber, plus orderLineNo of de artikelsleutel (article). Staat het artikel op meer regels van die order, dan is dat 422 business_rule met het verzoek orderLineNo mee te geven. Per regel gaan mee: aantal (positief; het ERP zet het retour-teken), reden (code plus tekst), voertuig, opmerking, en kind: article (het onderdeel gaat terug) of coreOnly (alleen het oude deel van een eerder geleverde regel wordt ingeleverd; vereist module Borg). Borg wordt alleen gecrediteerd als hij als toeslag op de retourregel staat.

Per orderregel mag hoogstens het geleverde aantal min eerder aangemelde retouren terug, per soort apart: article en coreOnly tellen los van elkaar, en afgewezen retouren tellen niet mee.

Overschrijden is 422 business_rule met errors[] per regel. Een terugbetaal-IBAN kan mee.

Idempotent op platformReference, zoals een order (13), en terug te vinden met GET /v1/returns?platformReference=.

Het ERP geeft `erpReturnId`, `erpReturnNumber`, per regel de orderverwijzing en de opmerking terug, en een status: `announced`, `received`, `credited` of `rejected` (overgangen: `announced` → `received` → `credited`; `announced` of `received` → `rejected`). De status staat op de kop; alle regels volgen hem. Aantallen en bedragen in het retourantwoord zijn positief (het is een retourdocument; de creditnota zelf is negatief). `creditAmountNet` is pas gevuld bij `credited`; dan vult het ERP ook `creditNoteNumber`, het nummer van de creditnota.

Het platform leest retouren zoals orders: periodiek de wijzigingen over alle klanten (GET `/v1/returns?updatedSince=`, oplopend op wijzigingstijd en id, met dezelfde cursor- en `asOf`-regels als orders en facturen; zie 10), en live per klant (`erpCustomerId`) als de garage zijn retouren opent. De garage ziet per retour het `erpReturnNumber` en de `status`, per regel het aantal en `creditAmountNet`, en bij `credited` het creditnotanummer. Meldingen van het ERP aan het platform blijven voorgenomen (17).

17. Uitbreidingen (modules)

Module	Endpoints	Capability
Prijs	POST <code>/v1/articles/lookup</code> met <code>erpCustomerId</code> ; <code>price</code> -blok in het antwoord	<code>pricing</code>
Login	POST <code>/v1/customers/authenticate</code>	<code>customers.authenticate</code>
Klant op debiteurnummer	GET <code>/v1/customers?debtorNumber=</code>	<code>customers.lookupByDebtorNumber</code>
Klant aanmaken	POST <code>/v1/customers</code> , PATCH <code>/v1/customers/{erpCustomerId}</code> (alleen <code>platformReference</code>)	<code>customers.create</code>
Artikelen aanleggen	POST <code>/v1/articles</code>	<code>articles.manage</code>
Inkoop	GET <code>/v1/suppliers</code> , POST <code>/v1/purchase-orders</code> , GET <code>/v1/purchase-orders</code>	<code>purchaseOrders</code>
Borg	<code>surcharges</code> met <code>kind: deposit</code> op order- en retourregels; <code>kind: coreOnly</code> op retour	<code>deposit</code>
Routes	GET <code>/v1/customers/{id}/delivery-schedule</code>	<code>deliverySchedule</code>
Export als bestand	POST <code>/v1/exports</code> , GET <code>/v1/exports/{id}</code>	<code>exports</code>
Testomgeving	POST <code>/v1/test/reset</code> , PUT <code>/v1/test/stock/{erpArticleId}</code> , POST <code>/v1/test/orders/{erpOrderId}/delivery</code> , POST <code>/v1/test/orders/{erpOrderId}/cancel</code> , POST <code>/v1/test/returns/{erpReturnId}/status</code> (8.4)	<code>testing</code>

Bij de modules:

- **Klanten** (Login, Klant op debiteurnummer, Klant aanmaken): zie bijlage A.2 en A.3.
- **Inkoop:** op een inkooporder stuurt het platform alleen een leverancierscode uit `GET /v1/suppliers`. De grossier legt in het platform vast welke leverancier bij welke code in het ERP hoort. Heeft het platform voor een leverancier geen code van het ERP, dan maakt het voor die leverancier geen inkooporder in het ERP, maar koopt het zelf in; op het artikel en op de orderregel staat dan de naam zoals het platform de leverancier kent, en het ERP bewaart die als tekst. Een onbekende leverancier op een inkooporder is 422 `business_rule`, omdat een inkooporder op een bestaande crediteur moet landen. Uniek is alleen `platformReference`; een tweede inkooporder voor dezelfde verkooporder en leverancier met een andere referentie is toegestaan. Per regel komen het regelnummer van de verkooporder (`orderLineNo`), de omschrijving en de verwachte aankomst bij de grossier terug, zodat het platform per verkoopregel ziet of hij is ingekocht. Zie ook bijlage A.4.
- **Routes:** het routeschema is in Europe/Amsterdam en kent geen uitzonderingen (feestdagen en dergelijke). Een klant zonder routes krijgt een lege lijst, geen 404. Het schema geldt voor regels uit eigen voorraad; voor regels die de grossier eerst bij een leverancier haalt, bepaalt het platform het levermoment zelf en geeft het dat per regel mee (13).

Voorgenomen, niet in deze versie: Meldingen (webhooks). Het ERP zou wijzigingen (order bevestigd, geleverd, gefactureerd, retourstatus) actief aan het platform melden, zodat de status in de shop sneller bijwerkt zonder periodiek navragen. Daarvoor zijn de registratie van URL en geheim, het herhaalschema en de ondertekening nog niet vastgelegd, en het platform heeft nog geen ontvangend endpoint. Deze module wordt gespecificeerd zodra het platform meldingen kan ontvangen; tot die tijd haalt het platform wijzigingen op met de delta (10).

18. Foutcodes

HTTP	code	Betekenis	Platform
400	<code>validation_error</code>	Body of parameters ongeldig; <code>errors[]</code> noemt de velden. Ook: tijdstip zonder offset, <code>limit</code> onder 1, onbekende <code>branchCode</code> (filter of body), lookup zonder of met dubbele <code>requestId</code> . Bij 405, 413 en 415 met dezelfde code.	Definitief
400	<code>invalid_cursor</code>	Cursor ongeldig of verlopen, of filters die afwijken van de eerste pagina.	Reeks opnieuw beginnen
401	<code>unauthorized</code>	Sleutel ontbreekt of is ongeldig; met <code>WWW-Authenticate: Bearer</code> .	Opnieuw proberen, beheerder waarschuwen
401	<code>invalid_credentials</code>	Garage-inlog onjuist (module Login).	Inlog geweigerd
403	<code>forbidden</code>	Sleutel geldig, maar deze bewerking is voor deze sleutel niet toegestaan; alleen als het ERP rechten per sleutel kent.	Opnieuw proberen, beheerder waarschuwen
404	<code>not_found</code>	Klant, order, artikel, retour of document bestaat niet, ook als verwijzing in de body of als filter; onbekend pad.	Definitief

HTTP	code	Betekenis	Platform
422	customer_blocked	Klant is geblokkeerd; order niet aangemaakt.	Definitief
422	business_rule	Andere inhoudelijke weigering; detail zegt welke. Bijvoorbeeld: order geweigerd op krediet, artikelsleutel bij aanleggen past op meer artikelen, retour boven het toegestane aantal of zonder eenduidige orderregel, onbekende leverancier op een inkooporder, dubbele e-mail of KvK bij klant aanmaken.	Definitief
429	rate_limited	Te veel aanroepen; Retry-After verplicht.	Opnieuw na Retry-After
500	internal_error	Onverwachte fout in het ERP.	Opnieuw proberen; bij een aanmakende POST eerst terugzoeken op referentie (10)
501	module_disabled	Onderdeel of module niet aanwezig volgens capabilities.	Capabilities opnieuw lezen
503	temporarily_unavailable	Onderhoud of storing, ook op GET /v1/capabilities; Retry-After gewenst.	Opnieuw proberen; bij een aanmakende POST eerst terugzoeken op referentie

"Definitief" betekent dat opnieuw aanbieden zonder wijziging geen zin heeft; het platform legt de bestelling dan bij de grossier neer.

Bijlage A — varianten

Het koppelvlak is één geheel; wat per grossier verschilt, zit in de uitbreidingen uit hoofdstuk 3.3. Hoe die in twee werkwijzen samengaan staat in 3.4; hieronder vier losse varianten, uitgeschreven zodat een ERP-leverancier per onderdeel kan kiezen wat hij aanbiedt.

A.1 Prijs uit het ERP

De grossier beheert klantcondities, staffels en acties in het ERP en wil dat de garage in de shop exact die prijs ziet.

- Module Prijs is actief. Het platform stuurt bij elke artikelaanvraag `erpCustomerId` mee en toont de teruggegeven netto prijs, bruto prijs en toeslagen.
- De order draagt de prijs die getoond is; het ERP accepteert die als ze op twee decimalen gelijk is aan zijn eigen berekening op dat moment bij het bestelde aantal, of corrigeert en meldt dat op de regel. Bij een online betaalde order corrigeert het ERP niet; toeslagen corrigeert het nooit.
- Het platform berekent in deze variant niets zelf. De besteleenheid en de milieubijdrage komen uit dezelfde prijsaanvraag.

- Consequentie voor het ERP: de prijsaanvraag zit op het drukste pad en moet de prestatie-eisen van hoofdstuk 8.3 halen.

A.2 Login bij het ERP

De grossier wil dat garages met hun bestaande ERP-inlog in de shop komen.

- Module Login is actief. Bij het inloggen roept het platform `POST /v1/customers/authenticate` aan met naam en wachtwoord en krijgt de debiteur terug.
- De inlognaam is hoofdlettergevoelig, het wachtwoord hoofdlettergevoelig. Een geblokkeerde klant logt gewoon in en komt terug met `isBlocked: true`; het platform respecteert de blokkade.
- Het platform onthoudt de koppeling tussen de inlog en `erpCustomerId`; latere aanroepen gaan op dat id, niet op het wachtwoord. Een storing in het ERP blokkeert daardoor alleen het eerste inloggen, niet een lopende sessie.
- Zonder deze module beheert het platform de inlog en is een eenmalige overgang van bestaande garages nodig: inlognaam behouden, nieuw wachtwoord kiezen.

A.3 Klant opzoeken op debiteurnummer

De grossier wil bestaande klanten aansluiten zonder dat de garage zich registreert.

- Module Klant op debiteurnummer is actief. De beheerder van de grossier voert het debiteurnummer in; het platform haalt de klant op en legt `erpCustomerId` vast. Het ERP vergelijkt genormaliseerd (6) en geeft alle klanten terug die passen; zijn dat er meer dan één, dan is het nummer dubbelzinnig en koppelt het platform niet.
- Met daarnaast module Klant aanmaken kan het platform ook nieuwe debiteuren aanmaken bij zelfregistratie (`POST /v1/customers`); zonder die module maakt de grossier de debiteur in het ERP aan en koppelt hij op debiteurnummer. Bij aanmaken geldt:
 - Het ERP kent het debiteurnummer en de betaalvoorwaarden toe en geeft ze terug.
 - `email` is optioneel: het platform heeft niet van elke garage een e-mailadres.
 - Het platform heeft één adresregel. Ontbreekt `houseNumber`, dan mag `street` straat en huisnummer samen bevatten.
 - Bestaat er al een debiteur met hetzelfde e-mailadres of KvK-nummer, dan mag het ERP weigeren met `422 business_rule`.
 - Daarna plaatst het platform de order op het teruggegeven `erpCustomerId` (3.4).
- Een verloren koppeling herstelt het platform via `GET /v1/customers` op platformreferentie of e-mailadres. Zodat dat ook werkt voor een debiteur die het platform niet zelf heeft aangemaakt, kan het platform met `PATCH /v1/customers/{erpCustomerId}` achteraf een `platformReference` op een bestaande debiteur zetten; dat endpoint wijzigt alleen dat veld.

A.4 Inkoop door het platform

De grossier wil dat de inkooporders per leverancier automatisch ontstaan uit wat de garage bestelt.

- Module Inkoop is actief. Direct na de verkooporder plaatst het platform per leverancier een inkooporder met verwijzing naar het verkoopordernummer, bevestigd of als concept (per leverancier instelbaar door de grossier).

- Regels uit eigen voorraad leveren geen inkoopregel op. Een regel zonder bekende inkoopprijs wordt overgeslagen en gemeld, nooit tegen nul geboekt.
- Uniek is alleen de platformreferentie van de inkooporder; een tweede inkooporder voor dezelfde leverancier mag. Een leverancier die het ERP niet kent, is 422 `business_rule`.
- Het platform gebruikt op een inkooporder alleen codes uit `GET /v1/suppliers`; de grossier legt de koppeling tussen leverancier en code in het platform vast. Voor een leverancier zonder code in het ERP maakt het platform geen inkooporder in het ERP, maar koopt het zelf in (17).
- Per inkoopregel geeft het ERP het regelnummer van de verkooporder, de omschrijving en de verwachte aankomst bij de grossier terug, zodat het platform kan controleren of elke verkoopregel is ingekocht.
- Het ERP maakt in deze variant zelf geen inkooporders aan op basis van de verkooporder.

Bijlage B — begrippen

Begrip	Betekenis
Grossier	De onderdelenhandel die het platform huurt en het ERP gebruikt.
Garage	De klant van de grossier; in het ERP een debiteur.
Platform	Grossier.nu: webshop, garagesoftware-koppeling en de generieke ERP-koppeling.
Vestiging	Een magazijn of filiaal van de grossier met eigen voorraad.
Uw vestiging	De vestiging waaraan de klant in het ERP hangt (<code>Customer.branchCode</code>); de andere vestigingen samen heten in de shop "elders bij de grossier".
Vrije voorraad	Fysiek aanwezig min al gereserveerd voor andere orders.
Platformreferentie	De unieke referentie die het platform op order, retour en inkooporder zet; idempotentiesleutel.
Levermoment	Het tijdstip (of venster) dat het platform de garage heeft beloofd: op basis van de route voor eigen voorraad, op basis van de aankomst bij de grossier voor artikelen van een leverancier. Op de order, en per regel als het afwijkt.
Ordertoeslag	Verzendkosten, kleine-orderkosten, bezorgtoeslag of een andere toeslag voor de hele order (<code>surcharges</code> op de order); nooit een artikelregel.
Orderstatus	De vaste status van een order (<code>statusCode</code> : open, confirmed, processing, delivered, invoiced, cancelled), naast de leverstatus en de vrije tekst van het ERP.
Openstaand bedrag	Wat van een factuur of creditnota nog openstaat, inclusief btw (<code>amountOpen</code>).
Borg	Statiegeld op een ruildeel, terug bij inlevering van het oude deel.

Bijlage C — wijzigingen per versie

Versie	Datum	Wat is er veranderd
0.5	22 oktober 2026	Uitkomst van een doorloop van de webshop op een ERP dat alleen dit koppelvlak implementeert: acht plekken waar een bouwer moest raden. Toeslagen op orderniveau (<code>OrderSurcharge</code>: verzendkosten, kleine-orderkosten, bezorgtoeslag, overig) in <code>OrderCreate.surcharges</code> en het nieuwe verplichte <code>Order.surcharges</code>; ze tellen mee in de totalen en zijn nooit een artikelregel of <code>manual</code> (13). Welke vestigingen tellen: alleen die uit <code>GET /branches</code>, de vestiging van de klant als "uw vestiging", zonder vestigingen één voorraadgetal zonder <code>branchCode</code>, dat daarom niet meer verplicht is (11). Levermoment per herkomst: het routeschema geldt voor eigen voorraad, voor leveranciersartikelen bepaalt het platform het levermoment; nieuw optioneel <code>promisedAt</code> per orderregel dat de kop overschrijft (13, 17). Nieuw verplicht <code>Order.statusCode</code> met een vaste lijst (open, confirmed, processing, delivered, invoiced, cancelled) en de samenhang met <code>deliveryStatus</code>; een order met alleen afgewezen regels is <code>cancelled</code>; <code>status</code> blijft vrije tekst als toelichting (14). Factuurbedragen per veld beschreven, <code>amountPaid</code> inclusief btw, nieuw verplicht <code>amountOpen</code>, <code>dueDate</code> is de vervaldatum (15). Documentlinks zijn voor het platform, niet voor de garage, en hoeven alleen vanaf het platform bereikbaar te zijn (15). Retouren teruglezen als delta en per klant, met wat de garage terugziet en <code>creditNoteNumber</code> bij <code>credited</code> (16). Op een inkooporder alleen leverancierscodes uit <code>GET /suppliers</code>; zonder code koopt het platform zelf in en bewaart het ERP de leveranciersnaam als tekst (17, A.4). Aanscherpingen in de zin van 9: <code>Order.statusCode</code>, <code>Order.surcharges</code> en <code>Invoice.amountOpen</code> zijn verplicht.
0.4	22 oktober 2026	Klant lezen (<code>GET /customers/{erpCustomerId}</code>) van het volledige koppelvlak naar het minimum: elke order verwijst naar een klant-id van het ERP, en de grossier moet die koppeling kunnen controleren voordat de eerste order gaat. Laag 1 is daarmee artikel, klant en order. Feature <code>customers.read</code> vervalt; de filters op <code>GET /customers</code> vallen onder de nieuwe feature <code>customers.list</code>. Ketenscenario A17 toegevoegd aan 8.2: één klant en één order van klant lezen tot retourstatus, met de eis dat alle identiteiten over de stappen heen gelijk blijven.
0.3	21 oktober 2026	Uitkomst van een tweede bouwtest, waarin een nieuwe bouwer zonder voorkennis een ERP uit v0.2 bouwde en de generieke koppeling erop werd bijgetrokken. Artikel geeft leveranciers met inkoopprijs, btw-tarief en adviesprijs terug (<code>suppliers</code>, <code>vatRate</code>, <code>salesPriceGross</code>); vastgelegd wat <code>POST /articles</code> met een bestaand artikel doet; regelstatussen afgebakend, ook na aanleggen; matchen zonder terugval bij een niet-passend merk, EAN zonder voorloophnullen, dubbele <code>requestId</code> is 400; retourregel verwijst altijd naar een order, hoeveelheidsregel per soort, status op de kop; documentlinks zonder sleutel, ten minste een uur geldig en niet te bewaren, plus <code>GET /orders/{id}/documents/{kind}</code>; idempotentie vóór validatie en uniek per grossier, terugzoeken na een 5xx; capabilities: elke module in tabel 3.4, <code>modules</code> leidend, alle sleutels expliciet; tijd op milliseconden, datumvelden als kalenderdatum, <code>null</code> gelijk aan weglaten; cursor 24 uur geldig, standaardsortering, <code>totalCount</code> altijd; ordernummer exact, tekstzoeken en filters beschreven; prijsvergelijking op twee decimalen, toeslagen nooit gecorrigeerd; klant aanmaken met optioneel e-mailadres en één adresregel, <code>PATCH /customers/{id}</code> voor de platformreferentie; inkoopregels met verkoopregelnummer, omschrijving en verwachte aankomst; module Testomgeving met beheerhandelingen onder <code>/test/</code> (8.4); Meldingen (webhooks) uit de modules en als voorgenomen uitbreiding in 17, <code>WebhookEvent</code> uit de OpenAPI; zes identiteiten als rij in 6; foutcodes en transportregels aangevuld.

Versie	Datum	Wat is er veranderd
0.2	21 oktober 2026	Uitkomst van een bouwtest waarin een bouwer zonder voorkennis een ERP uit de spec bouwde en het platform er in één keer op koppelde. Twee werkwijzen benoemd (3.4) met <code>masterData</code> in capabilities; module Artikelen aanleggen in plaats van aanmaken bij een order; idempotentie puur op platformreferentie (geen 409 op inhoud) en vastleggen vóór antwoorden; matchen vastgelegd (normalisatie, volgorde, <code>matchedOn</code>, <code>brands</code>, dubbelzinnig, fouten per sleutel); prijscorrectie zichtbaar op de regel en niet bij online betaling, staffels; regelstatussen gedefinieerd, <code>platformLineRef</code>, herkomst en opmerking terug te lezen, <code>methodCode</code>; paginering en delta aangescherpt, delta over alle klanten, <code>sort</code> en <code>totalCount</code>; document-URL zonder sleutel; retouren altijd onder <code>/returns</code>, retourregel verwijst naar de orderregel, statusovergangen en hoeveelheidsregel; foutcodes aangevuld met wat het platform ermee doet; module Testomgeving; zes stabiele identiteiten; tijd in UTC.
0.1	21 oktober 2026	Eerste openbare concept: drie lagen, bindende eisen los van de aanbevolen vorm, OpenAPI met 24 operaties.