

OTEP — Open Tracking Event Protocol

Vertaling voor het gemak — de Engelse versie is gezaghebbend.

Status: Draft v0.1 · Een open, leveranciersneutrale specificatie · Licentie: open (zie §10)

OTEP is een open, leveranciersneutraal protocol voor de levenscyclus van elk traceerbaar subject. Het definieert één gebeurtenismodel en één statusvocabulaire, zodat trackinggebeurtenissen uit elke bron — bezorging met eigen vloot, externe koeriers, vervoerderslabels en daarbuiten — kunnen worden uitgewisseld, begrepen en geprojecteerd naar internationale standaarden zonder voor elke partij opnieuw te hoeven integreren.

Dit document is de specificatie: de structuur, de velden, de statustabellen, de state machine, de mappings naar externe standaarden en de conformiteitsregels voor het bouwen van een conforme implementatie. Het is implementatie-onafhankelijk — het beschrijft het protocol, niet de interne werking van een bepaalde leverancier. RFC-2119-sleutelwoorden (MUST / SHOULD / MAY) zijn normatief.

1. Wat OTEP oplost

Tracking is gefragmenteerd: elke vervoerder benoemt velden en statuscodes anders, elk bezorgkanaal rapporteert in zijn eigen vorm, en aansluiten op wereldwijde standaarden betekent keer op keer opnieuw integreren. OTEP geeft producenten en consumenten één gemeenschappelijke taal: een producent zendt OTEP-gebeurtenissen één keer uit, en elke OTEP-consument begrijpt ze en kan ze projecteren naar de standaard die hij nodig heeft.

2. Ontwerpprincipes

1. Event-sourced. De gebeurtenistijdslijn is de bron van waarheid; "huidige status" is altijd een projectie — de status van de laatste gebeurtenis.
2. Wat / Wanneer / Waar / Waarom. Elke gebeurtenis is opgebouwd rond deze vier dimensies — dezelfde dimensies die GS1 EPCIS, IATA ONE Record en UN/CEFACT delen — zodat die standaarden uitvoerprojecties van een OTEP-gebeurtenis zijn in plaats van parallelle modellen.
3. Bronnen zonder lijst zijn geen blokkade. Een bron die alleen een huidige status blootlegt (geen geschiedenis) wordt verwerkt door één gebeurtenis per waargenomen wijziging te synthetiseren (§5).
4. Additief. OTEP wordt blootgesteld naast elke bestaande tracking-API; de adoptie ervan vereist nooit een breaking change voor wat consumenten al gebruiken.

3. De OTEP-tijdslijn

Een tijdslijn is een envelop die het getraceerde subject en een geordende lijst van gebeurtenissen draagt.

```
{
  "otep_version": "0.1",
  "profile": "parcel", // domeinprofiel (§8)
  "subject": {
    "tracking_number": "SR123...", // ?1 identifier vereist
    "order_id": 12345, // optioneel
    "package_id": 67890, // optioneel
    "external_tracking_number": "1Z...", // optioneel
    "gs1_sccc": "00...", // optioneel — schakelt EPCIS epclList in
    "piece_id": "...", // optioneel — schakelt ONE Record-koppeling in
  },
  "current_status": "delivered", // projectie van de laatste gebeurtenis
  "delivered": true,
  "events": [ /* OTEP-gebeurtenissen, §4 */ ]
}
```

De OTEP-gebeurtenis

```
{
  // WHAT – eenmalig gedragen op tijdslijnniveau (subject hierboven)
  // WHEN
  "occurred_at": "2026-06-10T09:30:00-04:00", // gebeurtenismoment, ISO-8601 met offset
  "recorded_at": "2026-06-10T09:45:23-04:00", // moment van opname (optioneel)
  "time_type": "actual", // actual | estimated | scheduled
  // WHERE
  "location": {
    "name": "Toronto Hub",
    "code": "YYZ-2",
    "gln": "0614141000005", // GS1 GLN ? EPCIS SGLN
    "lat": 43.6777, "lng": -79.6248,
    "country": "CA" // ISO 3166-1 alpha-2
  },
  // WHY / WHAT HAPPENED
  "status_code": "out_for_delivery", // een OTEP-statuscode (§4)
  "phase": "out_for_delivery", // afgeleid van status_code
  "incident_reason": null, // een OTEP incident reason (§4) bij uitzondering
  // WHO
  "actor": { "type": "driver", "name": "Jane D.", "phone": "+1..." },
  // PROVENANCE
  "source": {
    "type": "self_delivery", // self_delivery | third_party_delivery | carrier_label
    "provider_id": null,
    "carrier_code": null,
    "external_event_code": null, // jouw ruwe statuscode (bewaar deze)
    "raw": { /* originele payload */ }
  },
  // PROOF
  "pod": {
    "photos": [ { "url": "...", "content_base64": null } ],
    "signature": [ { "url": "...", "content_base64": null } ],
    "recipient": "John Smith"
  }
}
```

Elk veld behalve `subject`, `occurred_at`, `status_code` en `source` is optioneel — gedeeltelijke bronnen vullen aan wat ze hebben. Node-/hub-scans stellen `location` in op de scannende faciliteit. Hoogfrequente GPS-telemetrie is GEEN OTEP-gebeurtenis; een gebeurtenis wordt uitgezonden bij een statuswijziging of een node-scan.

4. Vocabulaire

4.1 Statuscodes

20 canonieke levenscycluscodes. `phase` is altijd afleidbaar uit de code.

code	phase	terminaal	POD	betekenis
information_submitted	pre_shipment			orderinformatie ontvangen
booking_confirmed	pre_shipment			vervoerder/boeking bevestigd
awaiting_pickup	pre_shipment			klaar voor ophalen
out_for_pickup	pickup			onderweg om op te halen
picked_up	pickup		✓	opgehaald bij verzender
pickup_failed	exception			ophaalpoging mislukt
pickup_rescheduled	exception			ophalen wordt opnieuw geprobeerd
received	inbound			ontvangen bij faciliteit
arrival_scan	inbound			aankomstscan bij node
in_transit	transit			onderweg

code	phase	terminaal	POD	betekenis
package_outbound	transit			faciliteit verlaten
removed_from_route	exception			van route gehaald
route_cancelled	exception			route geannuleerd
out_for_delivery	out_for_delivery			op het voertuig
delivered	delivered	✓	✓	bezorgd bij geadresseerde
delivery_failed	exception			bezorgpoging mislukt
delivery_rescheduled	exception			wordt opnieuw geprobeerd / opnieuw bezorgd
return_to_sender	return	✓		terug naar afzender
rejected_by_recipient	return	✓		ontvanger heeft geweigerd
cancelled	return	✓		order geannuleerd

4.2 Fasen

pre_shipment · preparing · pickup · inbound · in_custody · transit · out_for_delivery · delivered · exception · return. (preparing en in_custody zijn gereserveerd voor niet-pakketprofielen — §8.)

4.3 Brontypes & tijdtypes

source.type: self_delivery · third_party_delivery · carrier_label. time_type: actual · estimated · scheduled (standaard actual).

4.4 Incident reasons

Wanneer een gebeurtenis zich in de exception -fase bevindt, SHOULD deze een incident_reason dragen uit dit genormaliseerde vocabulaire:

- Carrier: carrier_damaged_parcel, carrier_sorting_error, carrier_address_not_found, carrier_parcel_lost, carrier_not_enough_time, carrier_vehicle_issue, carrier_capacity_exceeded, carrier_mechanical_delay
- Retailer/verzender: retailer_cancelled, retailer_incorrect_data, retailer_not_ready, retailer_incorrect_parcel, retailer_incorrect_dimensions, retailer_packaging_issue
- Consignee: consignee_refused, consignee_business_closed, consignee_not_available, consignee_not_home, consignee_cancelled, consignee_verification_failed, consignee_incorrect_address, consignee_access_restricted, consignee_safe_place_unavailable
- Douane: customs_delay, customs_documentation, customs_duties_unpaid, customs_prohibited, customs_inspection
- Overmacht: weather_delay, natural_disaster, force_majeure
- Overig: parcel_being_researched, security_issue, regulatory_hold, unknown

5. State machine & bronnen met alleen status

Fasen schuiven voorwaarts op; uitzonderingen onderbreken en lossen weer op. Terminale toestanden (delivered, return_to_sender, rejected_by_recipient, cancelled) sluiten het subject af.

```
pre_shipment ? preparing ? pickup ? inbound ? in_custody ? transit ? out_for_delivery ? delivered ?
                ???????????? exception ????????????
                ?????????????? return / cancelled ?
```

Regels:

- Consumenten MUST gebeurtenissen ordenen op occurred_at en MUST aankomst in willekeurige volgorde tolereren.

- Overgangen naar een terminale toestand zijn idempotent; herhalingen worden ontdubbeld.
- Na een terminale status MUST een producent geen verdere gebeurtenissen uitzenden, behalve een gedocumenteerde RMA-/heropeningsstroom.
- Een bron die alleen een huidige status blootlegt MUST één gebeurtenis per waargenomen wijziging synthetiseren (met een stabiele dedupe-sleutel) in plaats van de geschiedenis weg te laten. Na verloop van tijd hopen de snapshots zich op tot een tijdlijn.

6. Mappings naar externe standaarden

De vier dimensies van OTEP komen veld voor veld overeen met de belangrijkste standaarden, zodat elk een uitvoerprojectie van een OTEP-gebeurtenis is.

OTEP	GS1 EPCIS 2.0	IATA ONE Record	UN/CEFACT
<code>occurred_at (+offset)</code>	<code>eventTime + eventTimeZoneOffset</code>	<code>eventDate + eventTimeType=Actual</code>	Event Date/Time
<code>recorded_at</code>	<code>recordTime</code>	<code>recordedAt</code>	—
<code>subject (sscc/piece)</code>	<code>epcList (SSCC URN)</code>	<code>linkedObject</code> → Piece/Shipment	Consignment
<code>location (gln/lat/lng)</code>	<code>readPoint / bizLocation (SGLN)</code>	<code>recordedAtLocation</code> → Location	Location
<code>status_code</code>	<code>bizStep + disposition</code>	<code>eventCode</code>	Transport status code
<code>actor</code>	<code>sourceList / extension</code>	Actor / Party	Party
<code>incident_reason</code>	<code>disposition / ErrorDeclaration</code>	<code>event remark</code>	Status reason code

Per-code-waarden (EPCIS CBV `bizStep / disposition`, ONE Record `eventCode`, UN/CEFACT-code) worden gepubliceerd in het machineleesbare codeboek. Naast deze internationale standaarden kan een OTEP- tijdlijn ook worden geprojecteerd naar OpenTelemetry-traces, OGC SensorThings-observaties en gangbare commerceplatforms (AfterShip, Shopify, Amazon, Walmart, BigCommerce, Magento, WooCommerce, Etsy).

Vertrouwen: EPCIS CBV-waarden zijn stabiele standaard-URN's. ONE Record- en UN/CEFACT-codewaarden zijn best-fit voor de last mile en moeten worden gevalideerd tegen de officiële codelijsten vóór extern gebruik. "Delivered to consignee" heeft geen exacte CBV `bizStep` — de dichtstbijzijnde passende (`receiving + received`) wordt gebruikt, of een uitbreidings-URN uit een gebruikersvocabulaire.

7. De OTEP-API

OTEP wordt geconsumeerd via een klein, alleen-lezen HTTP-oppervlak; alle endpoints zijn openbaar.

Werkwoord	Pad	Retourneert
GET	<code>/api/v1/otep/trackings/{tracking_number}</code>	de tijdlijn voor een trackingnummer
GET	<code>/api/v1/otep/trackings/{tracking_number}/events</code>	alleen gebeurtenissen
POST	<code>/api/v1/otep/trackings/batch</code>	meerdere trackingnummers in één aanroep
GET	<code>/api/v1/otep/trackings/{tracking_number}?view=public</code>	openbare veilige weergave: zonder chauffeursnamen of interne id' s, beperkt tot het bedrijf van het domein, ook verhuis- en opslagordernummers
GET	<code>/api/v1/client/otep/trackings/{identifiser}</code>	(API-token) tijdlijn en details van uw eigen zending: pakket, verhuizing, opslag
GET	<code>/api/v1/client/otep/locker-storage-rentals/{id}</code>	(API-token) tijdlijn van een lockeropslag, op record-id
GET	<code>/oteptrack</code>	openbare trackingpagina: pakket, verhuizing, opslag; details na controle
POST	<code>/api/v1/otep/validate</code>	conformiteitscontrole voor een geposte tijdlijn (§9)

Een GraphQL-query die dezelfde tijdlijn blootlegt is eveneens beschikbaar.

Content negotiation

Dezelfde tijdlijn wordt geserialiseerd naar de representatie die je aanvraagt, via een `?format=` - queryparameter of een `Accept` -profiel:

Verzoek	Representatie
<code>?format=otep</code> (standaard)	native OTEP-tijdlijn
<code>?format=epcis</code>	GS1 EPCIS 2.0 (JSON-LD <code>ObjectEvent s</code>)
<code>?format=onerecord</code>	IATA ONE Record (<code>LogisticsEvent s</code>)
<code>?format=uncefact</code>	UN/CEFACT transport status
<code>?format=otlp</code>	OpenTelemetry-traces
<code>?format=sensorthings</code>	OGC SensorThings-observaties
<code>?format=aftership</code> <code>shopify</code> <code>amazon</code> <code>walmart</code> <code>bigcommerce</code> <code>magento</code> <code>woocommerce</code> <code>etsy</code>	de tracking-/fulfillmentvorm van het platform

Gebeurtenissen waaraan in een projectie geen code kan worden toegewezen, worden overgeslagen en geteld (nooit stilzwijgend weggegooid).

8. Domeinprofielen

OTEP is een algemeen protocol, geen pakketprotocol. De protocollaag (gebeurtenisenvelop, faseruggengraat, state machine) is universeel; concrete statuscodes behoren tot een profiel dat op de tijdlijn wordt gedeclareerd via `profile`. De codes in §4 vormen het **parcel** -profiel. Andere domeinen — verhuizen, voedselbezorging, opslag en daarbuiten — voegen hun eigen codeverzamelingen toe onder hun profiel, genaamruimted als `otep:<profile>:<code>`, elk gemapt op dezelfde faseruggengraat. Een profiel toevoegen is een uitbreiding, geen protocolwijziging.

9. Conformiteitsspecificatie (normatief)

Een producent of consument is OTEP-conform wanneer elke gebeurtenis die hij uitzendt of accepteert voldoet aan deze tabellen en regels.

9.1 Tijdlijnenvelop

Veld	Type	Vereist	Beperkingen
<code>otep_version</code>	string	MUST	semver, bijv. <code>0.1</code>
<code>profile</code>	string	MUST	een geregistreerd profiel
<code>subject</code>	object	MUST	§9.2
<code>current_status</code>	string null	SHOULD	een statuscode (§4)
<code>current_phase</code>	string null	SHOULD	MUST gelijk zijn aan de fase van <code>current_status</code> als beide aanwezig
<code>delivered</code>	boolean	SHOULD	<code>true</code> alleen als <code>current_status = delivered</code>
<code>events</code>	array	MUST	gebeurtenisobjecten (§9.3), ordenbaar op <code>occurred_at</code>

9.2 subject

Ten minste ÉÉN van `tracking_number` / `order_id` / `package_id` MUST aanwezig zijn.

Veld	Type	Beperkingen
<code>tracking_number</code>	string	niet-leeg
<code>order_id</code> / <code>package_id</code>	integer null	
<code>external_tracking_number</code>	string null	
<code>gs1_sssc</code>	string null	18 cijfers
<code>piece_id</code>	string null	

9.3 Gebeurtenisobject

#	Veld	Type	Vereist	Beperkingen
1	<code>occurred_at</code>	string	MUST	ISO-8601 met offset
2	<code>recorded_at</code>	string null	SHOULD	ISO-8601
3	<code>time_type</code>	string	MAY (standaard <code>actual</code>)	<code>actual</code> <code>estimated</code> <code>scheduled</code>
4	<code>status_code</code>	string null	MUST ¹	een code in §4.1
5	<code>phase</code>	string null	SHOULD	MUST gelijk zijn aan de fase van <code>status_code</code>
6	<code>incident_reason</code>	string null	SHOULD ²	een reden in §4.4
7	<code>description</code>	string null	MAY	leesbaar voor mensen
8	<code>location</code>	object null	MAY	§9.4
9	<code>actor</code>	object null	MAY	<code>{ type, name?, phone? }</code> ; <code>type</code> ∈ {driver, operator, carrier, system}
10	<code>source</code>	object	MUST	§9.5
11	<code>pod</code>	object null	MAY ³	<code>{ photos[], signature[], recipient? }</code>

¹ Een gecodeerde gebeurtenis MUST een statuscode uit §4.1 dragen. Een ruwe scan die je nog niet kunt classificeren MAY `status_code = null` instellen, maar MUST de native code bewaren in `source.external_event_code` en MUST worden geteld, nooit weggegooid. ² Een gebeurtenis in de `exception`-fase SHOULD een `incident_reason` dragen. ³ Gebeurtenissen met `status_code` ∈ {`delivered`, `picked_up`} SHOULD een `pod` dragen.

9.4 location

`name` (string) · `code` (string) · `gln` (GS1 GLN, 13 cijfers) · `lat` / `lng` (WGS-84) · `country` (ISO 3166-1 alpha-2). Alle optioneel.

9.5 source

Veld	Type	Vereist	Beperkingen
<code>type</code>	string	MUST	<code>self_delivery</code> <code>third_party_delivery</code> <code>carrier_label</code>
<code>provider_id</code>	integer null	MAY	
<code>carrier_code</code>	string null	MAY	
<code>external_event_code</code>	string null	SHOULD ⁴	jouw ruwe statuscode
<code>raw</code>	object null	MAY	originele payload

⁴ MUST aanwezig zijn wanneer `status_code` null is, zodat de native code nooit verloren gaat.

9.6 Regels

1. Tijd. `occurred_at` MUST parsen als ISO-8601. Normaliseer numerieke epochs en `.NET /Date(ms)/` bij ingest; zend die vormen niet uit.
2. Ordening / dedup. Consumenten MUST sorteren op `occurred_at`, aankomst in willekeurige volgorde tolereren, en ontdubbelen op (`subject`, `status_code`, `occurred_at`).
3. State machine. Zend na een terminale status geen verdere gebeurtenissen uit, behalve een gedocumenteerde RMA / heropening.
4. Geen stilzwijgend verlies. Gebeurtenissen die niet kunnen worden gecodeerd MUST worden geteld, nooit weggegooid.
5. Afleiding. `phase`, `current_status`, `current_phase`, `delivered` zijn projecties — indien aanwezig MUST ze consistent zijn met de gebeurtenistijdlijn.

9.7 Conformiteitsniveaus & validatie

- Level 1 — zendt de envelop uit (§9.1), gebeurtenissen met de vereiste velden (§9.3), geldige status- codes (§4.1), geldige fasen (§4.2), en respecteert de state machine (§5).

- Level 2 — zendt daarnaast minstens één projectie naar een externe standaard uit (§6) en, waar van toepassing, profielspecifieke codes (§8).

Verifieer je uitvoer door een tijdlijn te POSTen naar `POST /api/v1/otep/validate`. Behandel elke `errors` als blokkerend; verhelp `warnings`. Een machineleesbaar JSON Schema en het volledige codeboek (elke statuscode, fase en externe mapping) worden gepubliceerd voor offline validatie.

10. Openheid & governance

OTEP is een open specificatie, vrij voor elke partij om te implementeren.

- Normatief vs informatief. Normatief: de gebeurtenisenvolop, faseruggengraat, statusvocabulaire, state machine en veldmappings naar externe standaarden. Hoe een implementeerder OTEP koppelt aan zijn eigen interne systemen is zijn eigen zaak en valt hier buiten de scope.
- Stabiele identifiers. Codes worden geadresseerd als `otep:<profile>:<code>`, fasen als `otep:phase:<name>`. Eenmaal gepubliceerd in een uitgebrachte versie is de betekenis van een identifier onveranderlijk.
- Versionering. Semantische versionering. Codes/profielen toevoegen is een MINOR (achterwaarts compatibele) wijziging; de betekenis van een bestaande code wijzigen is een MAJOR wijziging en SHOULD worden vermeden. De protocolversie reist mee met elke tijdlijn (`otep_version`).
- Uitbreiding. Nieuwe profielen en codes worden voorgesteld tegen deze spec in plaats van geforkt, zodat onafhankelijke implementeers convergeren. Experimentele codes MAY een `x-`-prefix gebruiken (`otep:parcel:x-my_code`) totdat ze geregistreerd zijn.
- Licentie. De specificatie is bedoeld om te worden uitgebracht onder een open licentie — TBD, in afwachting van goedkeuring.

11. Interoperabiliteit met leveranciers — breng je eigen standaard mee

OTEP nodigt andere leveranciers uit om hun eigen trackinggebeurtenisstandaard mee te brengen, zodat OTEP ermee kan interopereren, in beide richtingen:

- Projecteer OTEP → jouw standaard. Definieer een mapping van een OTEP-tijdlijn naar jouw formaat, met hergebruik van het OTEP-statusvocabulaire. Het is een pure transformatie — tijdlijn erin, jouw structuur eruit — zodat de mapping één keer wordt geschreven en elke OTEP-producent jouw formaat kan uitzenden.
- Map jouw standaard → OTEP. Lever een crosswalk van jouw statusvocabulaire naar de OTEP-codes (§4) plus, indien nodig, een profiel (§8). Jouw ruwe codes worden bewaard in `source.external_event_code`; niet-gemapte codes worden geteld, nooit weggegooid.

Zelfs als je OTEP niet rechtstreeks kunt adopteren, ben je welkom om een enkele genormaliseerde `otep_status` toe te voegen aan je eigen API-responses en om je trackinggebeurteniscodes te delen voor crosswalk-mapping. Stel mappings voor tegen deze spec (in plaats van te forken) zodat implementeers convergeren; nieuwe formaten pluggen in dezelfde `?format=` content negotiation en breken nooit een bestaande consument.