

OTEP — Open Tracking Event Protocol

Překlad pro pohodlí — anglická verze je závazná.

Stav: Draft v0.1 · Otevřená, na dodavatelích nezávislá specifikace · Licence: otevřená (viz §10)

OTEP je otevřený, na dodavatelích nezávislý protokol pro životní cyklus libovolného sledovatelného předmětu. Definuje jeden model událostí a jeden slovník stavů, takže sledovací události z libovolného zdroje — doručení vlastní flotilou, kurýři třetích stran, přepravní štítky a další — lze vyměňovat, chápat a promítat do mezinárodních standardů bez nutnosti znovu integrovat každou stranu zvlášť.

Tento dokument je specifikace: struktura, pole, tabulky stavů, stavový automat, mapování na externí standardy a pravidla shody pro vytvoření kompatibilní implementace. Je nezávislá na implementaci — popisuje protokol, nikoli vnitřní mechanismy konkrétního dodavatele. Klíčová slova RFC-2119 (MUST / SHOULD / MAY) jsou normativní.

1. Co OTEP řeší

Sledování je roztříštěné: každý dopravce pojmenovává pole a stavové kódy odlišně, každý doručovací kanál hlásí ve svém vlastním tvaru a propojení s globálními standardy znamená opakovanou integraci. OTEP dává producentům a konzumentům jeden společný jazyk: producent vydá události OTEP jednou a každý konzument OTEP jim porozumí a dokáže je promítnout do standardu, který potřebuje.

2. Návrhové principy

1. Event-sourced. Časová osa událostí je zdrojem pravdy; „aktuální stav“ je vždy projekce — stav nejnovější události.
2. Co / Kdy / Kde / Proč. Každá událost je formována kolem těchto čtyř dimenzí — tytéž dimenze sdílejí GS1 EPCIS, IATA ONE Record a UN/CEFACT — takže tyto standardy jsou výstupní projekce události OTEP, nikoli paralelní modely.
3. Zdroje bez seznamu nejsou překážkou. Zdroj, který vystavuje pouze aktuální stav (bez historie), je zpracován syntetizováním jedné události pro každou pozorovanou změnu (§5).
4. Aditivní. OTEP je vystaven vedle libovolného existujícího sledovacího API; jeho přijetí nikdy nevyžaduje zpětně nekompatibilní změnu toho, co konzumenti již používají.

3. Časová osa OTEP

Časová osa je obálka nesoucí sledovaný předmět a uspořádaný seznam událostí.

```
{
  "otep_version": "0.1",
  "profile": "parcel", // domain profile (§8)
  "subject": {
    "tracking_number": "SR123...", // ?1 identifier required
    "order_id": 12345, // optional
    "package_id": 67890, // optional
    "external_tracking_number": "1Z...", // optional
    "gs1_sccc": "00...", // optional — enables EPCIS epcList
    "piece_id": "...", // optional — enables ONE Record linkage
  },
  "current_status": "delivered", // projection of the latest event
  "delivered": true,
  "events": [ /* OTEP events, §4 */ ]
}
```

Událost OTEP

```
{
  // WHAT — carried once at the timeline level (subject above)
  // WHEN
```

```

"occurred_at": "2026-06-10T09:30:00-04:00", // event instant, ISO-8601 with offset
"recorded_at": "2026-06-10T09:45:23-04:00", // ingestion instant (optional)
"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", // an OTEP status code (§4)
"phase": "out_for_delivery", // derived from status_code
"incident_reason": null, // an OTEP incident reason (§4) when exception
// 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, // your raw status code (preserve it)
  "raw": { /* original payload */ }
},
// PROOF
"pod": {
  "photos": [ { "url": "...", "content_base64": null } ],
  "signature": [ { "url": "...", "content_base64": null } ],
  "recipient": "John Smith"
}
}

```

Každé pole kromě `subject`, `occurred_at`, `status_code` a `source` je volitelné — částečné zdroje vyplní, co mají. Skeny v uzlu/hubu nastaví `location` na skenující zařízení. Vysokofrekvenční GPS telemetrie NENÍ událostí OTEP; událost je vydána při změně stavu nebo při skenu v uzlu.

4. Slovník

4.1 Stavové kódy

20 kanonických kódů životního cyklu. `phase` je vždy odvoditelná z kódu.

code	phase	terminální	POD	význam
<code>information_submitted</code>	<code>pre_shipment</code>			informace o objednavce přijaty
<code>booking_confirmed</code>	<code>pre_shipment</code>			dopravce/rezervace potvrzena
<code>awaiting_pickup</code>	<code>pre_shipment</code>			připraveno k vyzvednutí
<code>out_for_pickup</code>	<code>pickup</code>			na cestě k vyzvednutí
<code>picked_up</code>	<code>pickup</code>		✓	vyzvednuto od odesílatele
<code>pickup_failed</code>	<code>exception</code>			pokus o vyzvednutí selhal
<code>pickup_rescheduled</code>	<code>exception</code>			vyzvednutí bude opakováno
<code>received</code>	<code>inbound</code>			přijato v zařízení
<code>arrival_scan</code>	<code>inbound</code>			sken při příjezdu do uzlu
<code>in_transit</code>	<code>transit</code>			v pohybu
<code>package_outbound</code>	<code>transit</code>			opustilo zařízení
<code>removed_from_route</code>	<code>exception</code>			staženo z trasy
<code>route_cancelled</code>	<code>exception</code>			trasa zrušena
<code>out_for_delivery</code>	<code>out_for_delivery</code>			ve vozidle

code	phase	terminální	POD	význam
delivered	delivered	✓	✓	doručeno příjemci
delivery_failed	exception			pokus o doručení selhal
delivery_rescheduled	exception			bude opakováno / znovu doručeno
return_to_sender	return	✓		vrací se k odesílateli
rejected_by_recipient	return	✓		příjemce odmítl
cancelled	return	✓		objednávka zrušena

4.2 Fáze

pre_shipment · preparing · pickup · inbound · in_custody · transit · out_for_delivery · delivered · exception · return . (preparing a in_custody jsou vyhrazeny pro neparcelní profily — §8.)

4.3 Typy zdrojů a typy času

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

4.4 Důvody incidentů

Když je událost ve fázi exception , SHOULD nést incident_reason z tohoto normalizovaného slovníku:

- 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/shipper: 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
- Customs: customs_delay , customs_documentation , customs_duties_unpaid , customs_prohibited , customs_inspection
- Force majeure: weather_delay , natural_disaster , force_majeure
- Other: parcel_being_researched , security_issue , regulatory_hold , unknown

5. Stavový automat a zdroje pouze se stavem

Fáze postupují vpřed; výjimky je přerušují a vracejí se zpět. Terminální stavy (delivered , return_to_sender , rejected_by_recipient , cancelled) uzavírají předmět.

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

Pravidla:

- Konzumenti MUST uspořádat události podle occurred_at a MUST tolerovat příchod mimo pořadí.
- Přechody do terminálního stavu jsou idempotentní; opakování jsou deduplikována.
- Po terminálním stavu producent MUST NOT vydat další události s výjimkou zdokumentovaného toku RMA / znovuootevření.
- Zdroj vystavující pouze aktuální stav MUST syntetizovat jednu událost pro každou pozorovanou změnu (se stabilním deduplikačním klíčem) místo vynechání historie. Postupem času se snímky kumulují do časové osy.

6. Mapování na externí standardy

Čtyři dimenze OTEP odpovídají pole po poli hlavním standardům, takže každý je výstupní projekcí události OTEP.

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

Hodnoty pro jednotlivé kódy (EPCIS CBV `bizStep / disposition`, ONE Record `eventCode`, kód UN/CEFACT) jsou publikovány ve strojově čitelném číselníku. Kromě těchto mezinárodních standardů lze časovou osu OTEP promítnout také do OpenTelemetry trasování, OGC SensorThings pozorování a běžných obchodních platforem (AfterShip, Shopify, Amazon, Walmart, BigCommerce, Magento, WooCommerce, Etsy).

Spolehlivost: Hodnoty EPCIS CBV jsou stabilní standardní URN. Hodnoty kódů ONE Record a UN/CEFACT jsou nejlepší možnou volbou pro poslední míli a před externím použitím by měly být ověřeny vůči oficiálním seznamům kódů. „Doručeno příjemci“ nemá přesný CBV `bizStep` — používá se nejbližší odpovídající (`receiving + received`) nebo rozšiřující URN z uživatelského slovníku.

7. API OTEP

OTEP je konzumováno přes malé read-only HTTP rozhraní; všechny koncové body jsou veřejné.

Sloveso	Cesta	Vrací
GET	<code>/api/v1/otep/trackings/{tracking_number}</code>	časovou osu pro sledovací číslo
GET	<code>/api/v1/otep/trackings/{tracking_number}/events</code>	pouze události
POST	<code>/api/v1/otep/trackings/batch</code>	mnoho sledovacích čísel v jednom volání
GET	<code>/api/v1/otep/trackings/{tracking_number}?view=public</code>	veřejné bezpečné zobrazení: bez jmen řidičů a interních ID, omezené na firmu domény, i čísla stěhování a skladování
GET	<code>/api/v1/client/otep/trackings/{identifier}</code>	(API token) časová osa a podrobnosti vlastní zásilky: balík, stěhování, skladování
GET	<code>/api/v1/client/otep/locker-storage-rentals/{id}</code>	(API token) časová osa úschovy v boxu, podle ID záznamu
GET	<code>/oteptrack</code>	veřejná stránka sledování: balík, stěhování, skladování; podrobnosti po ověření
POST	<code>/api/v1/otep/validate</code>	kontrolu shody pro odeslanou časovou osu (\$9)

K dispozici je také GraphQL dotaz vystavující tutéž časovou osu.

Vyjednávání obsahu

Tatáž časová osa je serializována do té reprezentace, kterou si vyžádáte, prostřednictvím dotazového parametru `?format=` nebo profilu

Accept :

Požadavek	Reprezentace
<code>?format=otep</code> (výchozí)	nativní časová osa OTEP
<code>?format=epcis</code>	GS1 EPCIS 2.0 (JSON-LD <code>ObjectEvent</code> s)

Požadavek	Reprezentace
?format=onerecord	IATA ONE Record (LogisticsEvent s)
?format=uncefact	UN/CEFACT transport status
?format=otlp	OpenTelemetry trasování
?format=sensorthings	OGC SensorThings pozorování
?format=aftership shopify amazon walmart bigcommerce magento woocommerce etsy	tvár sledování/plnění dané platformy

Události, kterým nelze v projekci přiřadit kód, jsou přeskočeny a započítány (nikdy nejsou tiše zahozeny).

8. Doménové profily

OTEP je obecný protokol, nikoli parcelní. Protokolová vrstva (obálka událostí, páteř fází, stavový automat) je univerzální; konkrétní stavové kódy patří do profilu deklarovaného na časové ose pomocí `profile`. Kódy v §4 jsou profil **parcel**. Další domény — stěhování, doručování jídla, skladování a další — přidávají vlastní sady kódů pod svým profilem, s jmenným prostorem `otep:<profile>:<code>`, přičemž každý se mapuje na tutéž páteř fází. Přidání profilu je rozšíření, nikoli změna protokolu.

9. Specifikace shody (normativní)

Producent nebo konzument je OTEP-konformní, když každá událost, kterou vydá nebo přijme, splňuje tyto tabulky a pravidla.

9.1 Obálka časové osy

Pole	Typ	Pož.	Omezení
<code>otep_version</code>	string	MUST	semver, např. 0.1
<code>profile</code>	string	MUST	registrovaný profil
<code>subject</code>	object	MUST	§9.2
<code>current_status</code>	string null	SHOULD	stavový kód (§4)
<code>current_phase</code>	string null	SHOULD	MUST se rovnat fázi <code>current_status</code> , pokud jsou přítomny obě
<code>delivered</code>	boolean	SHOULD	<code>true</code> právě když <code>current_status = delivered</code>
<code>events</code>	array	MUST	objekty událostí (§9.3), seřaditelné podle <code>occurred_at</code>

9.2 subject

Alespoň JEDEN z `tracking_number` / `order_id` / `package_id` MUST být přítomen.

Pole	Typ	Omezení
<code>tracking_number</code>	string	neprázdné
<code>order_id</code> / <code>package_id</code>	integer null	
<code>external_tracking_number</code>	string null	
<code>gs1_sccc</code>	string null	18 číslic
<code>piece_id</code>	string null	

9.3 Objekt události

#	Pole	Typ	Pož.	Omezení
1	<code>occurred_at</code>	string	MUST	ISO-8601 s offsetem
2	<code>recorded_at</code>	string null	SHOULD	ISO-8601

#	Pole	Typ	Pož.	Omezení
3	<code>time_type</code>	string	MAY (výchozí <code>actual</code>)	<code>actual</code> <code>estimated</code> <code>scheduled</code>
4	<code>status_code</code>	string null	MUST ¹	kód v §4.1
5	<code>phase</code>	string null	SHOULD	MUST se rovnat fázi <code>status_code</code>
6	<code>incident_reason</code>	string null	SHOULD ²	důvod v §4.4
7	<code>description</code>	string null	MAY	čitelné člověkem
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 Kódovaná událost MUST nést stavový kód z §4.1. Surový sken, který zatím nemůžete klasifikovat, MAY nastavit `status_code = null`, ale MUST zachovat nativní kód v `source.external_event_code` a MUST být započítán, nikdy nezahozen. ² Událost ve fázi `exception` SHOULD nést `incident_reason`. ³ Události se `status_code` ∈ {`delivered`, `picked_up`} SHOULD nést `pod`.

9.4 location

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

9.5 source

Pole	Typ	Pož.	Omezení
<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	váš surový stavový kód
<code>raw</code>	object null	MAY	původní payload

⁴ MUST být přítomen, když je `status_code` `null`, aby se nativní kód nikdy neztratil.

9.6 Pravidla

- Čas. `occurred_at` MUST být zpracovatelný jako ISO-8601. Při příjmu normalizujte číselné epochy a `.NET /Date(ms)/`; tyto formy nevydávejte.
- Uspořádání / deduplikace. Konzumenti MUST řadit podle `occurred_at`, tolerovat příchod mimo pořadí a deduplikovat podle (`subject`, `status_code`, `occurred_at`).
- Stavový automat. Po terminálním stavu nevydávejte další události s výjimkou zdokumentovaného RMA / znovuotevření.
- Žádná tichá ztráta. Události, které nelze kódovat, MUST být započítány, nikdy zahozeny.
- Odvození. `phase`, `current_status`, `current_phase`, `delivered` jsou projekce — pokud jsou přítomny, MUST být konzistentní s časovou osou událostí.

9.7 Úrovně shody a validace

- Level 1 — vydává obálku (§9.1), události s požadovanými poli (§9.3), platné stavové kódy (§4.1), platné fáze (§4.2) a respektuje stavový automat (§5).
- Level 2 — navíc vydává alespoň jednu projekci do externího standardu (§6) a, kde je to relevantní, kódy specifické pro profil (§8).

Ověřte svůj výstup odesláním časové osy na `POST /api/v1/otep/validate`. Jakékoli `errors` považujte za blokující; řešte `warnings`. Pro offline validaci jsou publikovány strojově čitelné JSON Schema a kompletní číselník (každý stavový kód, fáze a externí mapování).

10. Otevřenost a správa

OTEP je otevřená specifikace, kterou může implementovat kdokoli zdarma.

- Normativní vs informativní. Normativní: obálka událostí, páteř fází, slovník stavů, stavový automat a mapování polí na externí standardy. To, jak implementátor naváže OTEP na své vlastní interní systémy, je jeho věc a je mimo rozsah tohoto dokumentu.
- Stabilní identifikátory. Kódy jsou adresovány jako `otep:<profile>:<code>`, fáze jako `otep:phase:<name>`. Jakmile je identifikátor publikován ve vydané verzi, jeho význam je neměnný.
- Verzování. Sémantické verzování. Přidání kódů/profilů je MINOR (zpětně kompatibilní) změna; změna významu existujícího kódu je MAJOR změna a SHOULD jí být vyhnuto. Verze protokolu putuje s každou časovou osou (`otep_version`).
- Rozšíření. Nové profily a kódy jsou navrhovány vůči této specifikaci, nikoli forkovány, takže nezávislí implementátoři konvergují. Experimentální kódy MAY používat prefix `x-` (`otep:parcel:x-my_code`), dokud nejsou registrovány.
- Licence. Specifikace má být vydána pod otevřenou licencí — bude doplněno, čeká na schválení.

11. Interoperabilita dodavatelů — přineste si vlastní standard

OTEP vítá další dodavatele, aby přinesli svůj vlastní standard sledovacích událostí, aby s ním OTEP mohl interoperovat, v obou směrech:

- Projekce OTEP → váš standard. Definujte mapování z časové osy OTEP do vašeho formátu, s opětovným využitím slovníku stavů OTEP. Je to čistá transformace — vstup časová osa, výstup vaše struktura — takže mapování se napíše jednou a každý producent OTEP může vydávat váš formát.
- Mapování váš standard → OTEP. Poskytněte převodní tabulku z vašeho slovníku stavů na kódy OTEP (§4) plus, pokud je třeba, profil (§8). Vaše surové kódy jsou zachovány v `source.external_event_code`; nenamapované kódy jsou započítány, nikdy zahozeny.

I když nemůžete přijmout OTEP přímo, jste vítáni přidat jediný normalizovaný `otep_status` do svých vlastních API odpovědí a sdílet své kódy sledovacích událostí pro převodní mapování. Navrhujte mapování vůči této specifikaci (místo forkování), aby implementátoři konvergovali; nové formáty se zapojí do stejného vyjednávání obsahu `?format=` a nikdy nenaruší existujícího konzumenta.