

OTEP — Open Tracking Event Protocol

Preklad pre pohodlie — anglická verzia je záväzná.

Stav: Návrh v0.1 · Otvorená, na výrobcu neutrálna špecifikácia · Licencia: otvorená (pozri §10)

OTEP je otvorený, na výrobcu neutrálny protokol pre životný cyklus akéhokoľvek sledovateľného subjektu. Definuje jeden model udalostí a jeden slovník stavov, aby sa sledovacie udalosti z akéhokoľvek zdroja — doručenie vlastnou flotilou, kuriéri tretích strán, prepravné štítky a ďalšie — mohli vymieňať, chápať a premietiť do medzinárodných štandardov bez opätovnej integrácie pre každú stranu.

Tento dokument je špecifikáciou: štruktúra, polia, tabuľky stavov, stavový automat, mapovania na externé štandardy a pravidlá zhody pre vytvorenie kompatibilnej implementácie. Je nezávislý od implementácie — opisuje protokol, nie vnútornosti konkrétneho výrobcu. Kľúčové slová RFC-2119 (MUST / SHOULD / MAY) sú normatívne.

1. Čo OTEP rieši

Sledovanie je fragmentované: každý prepravca pomenúva polia a stavové kódy odlišne, každý doručovací kanál podáva správy vo vlastnom tvare a pripojenie ku globálnym štandardom znamená opätovnú integráciu znova a znova. OTEP poskytuje producentom a konzumentom jeden spoločný jazyk: producent vysiela udalosti OTEP raz a každý konzument OTEP im rozumie a môže ich premietnuť do štandardu, ktorý potrebuje.

2. Princípy návrhu

1. Event-sourced. Časová os udalostí je zdrojom pravdy; „aktuálny stav“ je vždy projekcia — stav najnovšej udalosti.
2. Čo / Kedy / Kde / Prečo. Každá udalosť je formovaná okolo týchto štyroch dimenzií — tých istých dimenzií, ktoré zdieľajú GS1 EPCIS, IATA ONE Record a UN/CEFACT — takže tieto štandardy sú výstupnými projekciami udalosti OTEP, a nie paralelnými modelmi.
3. Zdroje bez zoznamu nie sú prekážkou. Zdroj, ktorý vystavuje iba aktuálny stav (bez histórie), sa spracuje syntetizovaním jednej udalosti na každú pozorovanú zmenu (§5).
4. Aditívny. OTEP je vystavený popri akomkoľvek existujúcom sledovacom API; jeho prijatie nikdy nevyžaduje prelomovú zmenu toho, čo konzumenti už používajú.

3. Časová os OTEP

Časová os je obálka nesúca sledovaný subjekt a usporiadaný zoznam udalostí.

```
{
  "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 */ ]
}
```

Udalosť 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 okrem `subject`, `occurred_at`, `status_code` a `source` je voliteľné — čiastočné zdroje vyplnia to, čo majú. Skenovania uzlov/hubov nastaví `location` na skenujúce zariadenie. Vysokofrekvenčná GPS telemetria NIE JE udalosťou OTEP; udalosť sa vysielá pri zmene stavu alebo pri skenovaní uzla.

4. Slovník

4.1 Stavové kódy

20 kanonických kódov životného cyklu. `phase` je vždy odvoditeľná z kódu.

code	phase	terminálny	POD	význam
<code>information_submitted</code>	<code>pre_shipment</code>			prijaté informácie o objednávke
<code>booking_confirmed</code>	<code>pre_shipment</code>			potvrdený prepravca/rezervácia
<code>awaiting_pickup</code>	<code>pre_shipment</code>			prípravené na vyzdvihnutie
<code>out_for_pickup</code>	<code>pickup</code>			na ceste vyzdvihnúť
<code>picked_up</code>	<code>pickup</code>		✓	vyzdvihnuté od odosielateľa
<code>pickup_failed</code>	<code>exception</code>			pokus o vyzdvihnutie zlyhal
<code>pickup_rescheduled</code>	<code>exception</code>			vyzdvihnutie sa zopakuje
<code>received</code>	<code>inbound</code>			prijaté v zariadení
<code>arrival_scan</code>	<code>inbound</code>			skenovanie príchodu v uzle
<code>in_transit</code>	<code>transit</code>			v pohybe
<code>package_outbound</code>	<code>transit</code>			opustilo zariadenie
<code>removed_from_route</code>	<code>exception</code>			odobraté z trasy
<code>route_cancelled</code>	<code>exception</code>			trasa zrušená

code	phase	terminálny	POD	význam
out_for_delivery	out_for_delivery			vo vozidle
delivered	delivered	✓	✓	doručené príjemcovi
delivery_failed	exception			pokus o doručenie zlyhal
delivery_rescheduled	exception			zopakuje sa / opätovné doručenie
return_to_sender	return	✓		vracia sa na pôvod
rejected_by_recipient	return	✓		príjemca odmietol
cancelled	return	✓		objednávka zrušená

4.2 Fázy

pre_shipment · preparing · pickup · inbound · in_custody · transit · out_for_delivery · delivered · exception · return . (preparing a in_custody sú vyhradené pre profily mimo zásielok — §8.)

4.3 Typy zdrojov a typy času

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

4.4 Dôvody incidentov

Keď je udalosť vo fáze exception , MAL by niesť incident_reason z tohto normalizovaného slovníka:

- 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 len so stavom

Fázy postupujú dopredu; výnimky prerušia a vyriešia sa späť. Terminálne stavy (delivered , return_to_sender , rejected_by_recipient , cancelled) uzatvárajú subjekt.

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

Pravidlá:

- Konzumenti MUSIA usporiadať udalosti podľa occurred_at a MUSIA tolerovať príchod v nesprávnom poradí.
- Prechody do terminálneho stavu sú idempotentné; opakovania sa deduplikujú.
- Po terminálnom stave producent NESMIE vyslať ďalšie udalosti okrem zdokumentovaného toku RMA / opätovného otvorenia.
- Zdroj vystavujúci iba aktuálny stav MUSÍ syntetizovať jednu udalosť na každú pozorovanú zmenu (so stabilným deduplikačným kľúčom), namiesto vynechania histórie. Časom sa snímky nahromadia do časovej osi.

6. Mapovania na externé štandardy

Štyri dimenzie OTEP sa pole po poli zhodujú s hlavnými štandardmi, takže každý z nich je výstupnou projekciou udalosti OTEP.

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

Hodnoty na jednotlivé kódy (EPCIS CBV bizStep / disposition , ONE Record eventCode , kód UN/CEFACT) sú publikované v strojovo čitateľnom číselníku. Okrem týchto medzinárodných štandardov možno časovú os OTEP premietnuť aj do OpenTelemetry trace, OGC SensorThings observations a bežných obchodných platforiem (AfterShip, Shopify, Amazon, Walmart, BigCommerce, Magento, WooCommerce, Etsy).

Spôľahlivosť: Hodnoty EPCIS CBV sú stabilné štandardné URN. Hodnoty kódov ONE Record a UN/CEFACT sú najlepším priblížením pre poslednú mŕľu a pred externým použitím by sa mali overiť oproti oficiálnym zoznamom kódov. „Delivered to consignee “ nemá presný CBV bizStep — používa sa najbližšia zhoda (receiving + received) alebo rozširujúce URN používateľského slovníka.

7. API OTEP

OTEP sa konzumuje cez malé read-only HTTP rozhranie; všetky koncové body sú verejné.

Sloveso	Cesta	Vracia
GET	/api/v1/otep/trackings/{tracking_number}	časovú os pre sledovacie číslo
GET	/api/v1/otep/trackings/{tracking_number}/events	iba udalosti
POST	/api/v1/otep/trackings/batch	mnoho sledovacích čísel v jednom volaní
POST	/api/v1/otep/validate	kontrolu zhody pre odoslanú časovú os (§9)

K dispozícii je aj GraphQL dotaz vystavujúci tú istú časovú os.

Vyjednávane obsahu

Tá istá časová os sa serializuje do ľubovoľnej reprezentácie, ktorú požadujete, cez ?format= query parameter alebo profil Accept :

Požiadavka	Reprezentácia
?format=otep (predvolené)	natívna časová os OTEP
?format=epcis	GS1 EPCIS 2.0 (JSON-LD ObjectEvent s)
?format=onerecord	IATA ONE Record (LogisticsEvent s)
?format=uncefact	UN/CEFACT transport status
?format=otlp	OpenTelemetry trace
?format=sensorthings	OGC SensorThings observations
?format=aftership shopify amazon walmart bigcommerce magento woocommerce etsy	tvar sledovania/plnenia danej platformy

Udalosti, ktorým nemožno priradiť kód v projekcii, sa preskočia a započítajú (nikdy nie sú ticho zahodené).

8. Doménové profily

OTEP je všeobecný protokol, nie len protokol pre zásielky. Vrstva protokolu (obálka udalosti, chrbtica fáz, stavový automat) je univerzálna; konkrétne stavové kódy patria do profilu deklarovaného na časovej osi cez `profile`. Kódy v §4 sú profilom **parcel**. Ďalšie domény — sťahovanie, doručovanie jedla, skladovanie a ďalšie — pridávajú vlastné množiny kódov pod svojím profilom, s priestorom mien `otep:<profile>:<code>`, pričom každý mapuje na tú istú chrbticu fáz. Pridanie profilu je rozšírenie, nie zmena protokolu.

9. Špecifikácia zhody (normatívna)

Producent alebo konzument je zhodný s OTEP, keď každá udalosť, ktorú vysiela alebo prijíma, spĺňa tieto tabuľky a pravidlá.

9.1 Obálka časovej osi

Pole	Typ	Pož.	Obmedzenia
<code>otep_version</code>	string	MUST	semver, napr. 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	MUSÍ sa rovnať fáze <code>current_status</code> , ak sú prítomné obe
<code>delivered</code>	boolean	SHOULD	<code>true</code> práve vtedy, keď <code>current_status = delivered</code>
<code>events</code>	array	MUST	objekty udalostí (§9.3), usporiadateľné podľa <code>occurred_at</code>

9.2 subject

MUSÍ byť prítomný aspoň JEDEN z `tracking_number` / `order_id` / `package_id`.

Pole	Typ	Obmedzenia
<code>tracking_number</code>	string	neprázdne
<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 udalosti

#	Pole	Typ	Pož.	Obmedzenia
1	<code>occurred_at</code>	string	MUST	ISO-8601 s offsetom
2	<code>recorded_at</code>	string null	SHOULD	ISO-8601
3	<code>time_type</code>	string	MAY (predvolené <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	MUSÍ sa rovnať fáze <code>status_code</code>
6	<code>incident_reason</code>	string null	SHOULD ²	dôvod v §4.4
7	<code>description</code>	string null	MAY	čitateľné pre človeka
8	<code>location</code>	object null	MAY	§9.4
9	<code>actor</code>	object null	MAY	{ <code>type</code> , <code>name?</code> , <code>phone?</code> }; <code>type</code> ∈ { <code>driver</code> , <code>operator</code> , <code>carrier</code> , <code>system</code> }

#	Pole	Typ	Pož.	Obmedzenia
10	source	object	MUST	\$9.5
11	pod	object null	MAY ³	{ photos[], signature[], recipient? }

¹ Kódovaná udalosť MUSÍ niesť stavový kód z §4.1. Surové skenovanie, ktoré ešte nedokážete klasifikovať, MÔŽE nastaviť `status_code = null`, ale MUSÍ zachovať natívny kód v `source.external_event_code` a MUSÍ byť započítané, nikdy nie zahodené. ² Udalosť vo fáze `exception` by MALA niesť `incident_reason`. ³ Udalosti so `status_code ∈ { delivered, picked_up }` by MALI niesť `pod`.

9.4 location

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

9.5 source

Pole	Typ	Pož.	Obmedzenia
type	string	MUST	<code>self_delivery</code> <code>third_party_delivery</code> <code>carrier_label</code>
provider_id	integer null	MAY	
carrier_code	string null	MAY	
external_event_code	string null	SHOULD ⁴	váš surový stavový kód
raw	object null	MAY	pôvodný payload

⁴ MUSÍ byť prítomný, keď je `status_code = null`, aby sa natívny kód nikdy nestratil.

9.6 Pravidlá

- Čas. `occurred_at` MUSÍ sa dať analyzovať ako ISO-8601. Pri prijíme normalizujte číselné epochy a `.NET /Date(ms)/`; tieto formy nevysielajte.
- Usporiadanie / deduplikácia. Konzumenti MUSIA triediť podľa `occurred_at`, tolerovať príchod v nesprávnom poradí a deduplikovať na (`subject`, `status_code`, `occurred_at`).
- Stavový automat. Po terminálnom stave nevysielajte ďalšie udalosti okrem zdokumentovaného RMA / opätovného otvorenia.
- Žiadna tichá strata. Udalosti, ktoré nemožno zakódovať, MUSIA byť započítané, nikdy nie zahodené.
- Ododenie. `phase`, `current_status`, `current_phase`, `delivered` sú projekcie — ak sú prítomné, MUSIA byť konzistentné s časovou osou udalostí.

9.7 Úrovne zhody a validácia

- Level 1 — vysiela obálku (§9.1), udalosti s povinnými poľami (§9.3), platné stavové kódy (§4.1), platné fázy (§4.2) a rešpektuje stavový automat (§5).
- Level 2 — navyše vysiela aspoň jednu projekciu na externý štandard (§6) a tam, kde je to použiteľné, kódy špecifické pre profil (§8).

Overte svoj výstup odoslaním časovej osi cez `POST /api/v1/otep/validate`. Akékoľvek `errors` považujte za blokujúce; riešte `warnings`. Strojovo čitateľná JSON Schema a kompletný číselník (každý stavový kód, fáza a externé mapovanie) sú publikované pre offline validáciu.

10. Otvorenosť a správa

OTEP je otvorená špecifikácia, voľne implementovateľná ktoroukoľvek stranou.

- Normatívne vs. informatívne. Normatívne: obálka udalosti, chrbtica fáz, slovník stavov, stavový automat a mapovania polí na externé štandardy. To, ako implementátor naviaže OTEP na vlastné interné systémy, je jeho vec a je tu mimo rozsahu.

- Stabilné identifikátory. Kódy sa adresujú ako `otep:<profile>:<code>`, fázy ako `otep:phase:<name>`. Po publikovaní vo vydanej verzii je význam identifikátora nemenný.
- Verziovanie. Sémantické verziovanie. Pridanie kódov/profilov je MINOR (späťne kompatibilná) zmena; zmena významu existujúceho kódu je MAJOR zmena a MALA by sa jej vyhnúť. Verzia protokolu cestuje s každou časovou osou (`otep_version`).
- Rozšírenie. Nové profily a kódy sa navrhujú oproti tejto špecifikácii, namiesto forkovania, aby nezávislí implementátori konvergovali. Experimentálne kódy MÔŽU používať predponu `x-` (`otep:parcel:x-my_code`), kým nie sú zaregistrované.
- Licencia. Špecifikácia má byť vydaná pod otvorenou licenciou — TBD, čaká sa na schválenie.

11. Interoperabilita výrobcov — prineste si vlastný štandard

OTEP víta, aby iní výrobcovia priniesli vlastný štandard sledovacích udalostí, aby OTEP mohol interoperovať s ním, v oboch smeroch:

- Projekcia OTEP → váš štandard. Definujte mapovanie z časovej osi OTEP do vášho formátu, pričom znova použijete slovník stavov OTEP. Je to čistá transformácia — časová os dnu, vaša štruktúra von — takže mapovanie sa napíše raz a každý producent OTEP môže vyslať váš formát.
- Mapovanie váš štandard → OTEP. Poskytnite prepojenie z vášho slovníka stavov na kódy OTEP (§4) plus, ak treba, profil (§8). Vaše surové kódy sú zachované v `source.external_event_code`; nezmapované kódy sa započítajú, nikdy nie zahodia.

Aj keď nemôžete prijať OTEP priamo, môžete do svojich API odpovedí pridať jediný normalizovaný `otep_status` a zdieľať svoje kódy sledovacích udalostí na prepojovacie mapovanie. Navrhujte mapovania oproti tejto špecifikácii (namiesto forkovania), aby implementátori konvergovali; nové formáty sa zapoja do toho istého vyjednávania obsahu `?format=` a nikdy nepokazia existujúceho konzumenta.