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 premietať 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_sscc": "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 nastavia `location` na skenujúce zariadenie. Vysokofrekvenčná GPS telemetria NIE JE udalosťou OTEP; udalosť sa vysiela 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
`information_submitted`	pre_shipment			prijaté informácie o objednávke
`booking_confirmed`	pre_shipment			potvrdený prepravca/rezervácia
`awaiting_pickup`	pre_shipment			pripravené na vyzdvihnutie
`out_for_pickup`	pickup			na ceste vyzdvihnúť
`picked_up`	pickup		✓	vyzdvihnuté od odosielateľa
`pickup_failed`	exception			pokus o vyzdvihnutie zlyhal
`pickup_rescheduled`	exception			vyzdvihnutie sa zopakuje
`received`	inbound			prijaté v zariadení
`arrival_scan`	inbound			skenovanie príchodu v uzle
`in_transit`	transit			v pohybe
`package_outbound`	transit			opustilo zariadenie
`removed_from_route`	exception			odobraté z trasy
`route_cancelled`	exception			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 vysielať 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).

> Spoľahlivosť: Hodnoty EPCIS CBV sú stabilné štandardné URN. Hodnoty kódov ONE Record a UN/CEFACT sú najlepším priblížením pre poslednú míľu 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
GET	`/api/v1/otep/trackings/{tracking_number}?view=public`	verejné bezpečné zobrazenie: bez mien vodičov a interných ID, obmedzené na firmu domény, aj čísla sťahovania a skladovania
GET	`/api/v1/client/otep/trackings/{identifier}`	(API token) časová os a podrobnosti vlastnej zásielky: balík, sťahovanie, skladovanie
GET	`/api/v1/client/otep/locker-storage-rentals/{id}`	(API token) časová os úschovy v boxe, podľa ID záznamu
GET	`/oteptrack`	verejná stránka sledovania: balík, sťahovanie, skladovanie; podrobnosti po overení
POST	`/api/v1/otep/validate`	kontrolu zhody pre odoslanú časovú os (§9)

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

Vyjednávanie 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)

Požiadavka	Reprezentácia
`?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
`otep_version`	string	MUST	semver, napr. `0.1`
`profile`	string	MUST	registrovaný profil
`subject`	object	MUST	§9.2
`current_status`	string|null	SHOULD	stavový kód (§4)
`current_phase`	string|null	SHOULD	MUSÍ sa rovnať fáze `current_status`, ak sú prítomné obe
`delivered`	boolean	SHOULD	`true` práve vtedy, keď `current_status` = `delivered`
`events`	array	MUST	objekty udalostí (§9.3), usporiadateľné podľa `occurred_at`

9.2 `subject`

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

Pole	Typ	Obmedzenia
`tracking_number`	string	neprázdne
`order_id` / `package_id`	integer|null	
`external_tracking_number`	string|null	
`gs1_sscc`	string|null	18 číslic
`piece_id`	string|null	

9.3 Objekt udalosti

#	Pole	Typ	Pož.	Obmedzenia
1	`occurred_at`	string	MUST	ISO-8601 s offsetom
2	`recorded_at`	string|null	SHOULD	ISO-8601
3	`time_type`	string	MAY (predvolené `actual`)	`actual` | `estimated` | `scheduled`

#	Pole	Typ	Pož.	Obmedzenia
4	`status_code`	string|null	MUST[1]	kód v §4.1
5	`phase`	string|null	SHOULD	MUSÍ sa rovnať fáze `status_code`
6	`incident_reason`	string|null	SHOULD[2]	dôvod v §4.4
7	`description`	string|null	MAY	čitateľné pre človeka
8	`location`	object|null	MAY	§9.4
9	`actor`	object|null	MAY	`{ type, name?, phone? }`; type ∈ {driver, operator, carrier, system}
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	`self_delivery` | `third_party_delivery` | `carrier_label`
`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

1. Čas. `occurred_at` MUSÍ sa dať analyzovať ako ISO-8601. Pri príjme normalizujte číselné epochy a `.NET /Date(ms)/`; tieto formy nevysielajte.
2. Usporiadanie / deduplikácia. Konzumenti MUSIA triediť podľa `occurred_at`, tolerovať príchod v nesprávnom poradí a deduplikovať na (`subject`, `status_code`, `occurred_at`).
3. Stavový automat. Po terminálnom stave nevysielajte ďalšie udalosti okrem zdokumentovaného RMA / opätovného otvorenia.
4. Žiadna tichá strata. Udalosti, ktoré nemožno zakódovať, MUSIA byť započítané, nikdy nie zahodené.
5. Odvodenie. `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 — naviac 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 (spätne 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 vysielať 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.