OTEP — Open Tracking Event Protocol

Traduzione di cortesia — la versione inglese è quella autorevole.

> Stato: Bozza v0.1 · Una specifica aperta e neutrale rispetto ai fornitori · Licenza: aperta (vedi §10)

OTEP è un protocollo aperto e neutrale rispetto ai fornitori per il ciclo di vita di qualsiasi soggetto tracciabile. Definisce un unico modello di evento e un unico vocabolario di stato affinché gli eventi di tracciamento provenienti da qualsiasi origine — consegna con flotta propria, corrieri di terze parti, etichette dei vettori e oltre — possano essere scambiati, compresi e proiettati verso gli standard internazionali senza dover reintegrare per ogni parte.

Questo documento è la specifica: la struttura, i campi, le tabelle di stato, la macchina a stati, le mappature verso gli standard esterni e le regole di conformità per costruire un'implementazione conforme. È indipendente dall'implementazione: descrive il protocollo, non gli interni di un particolare fornitore. Le parole chiave RFC-2119 (MUST / SHOULD / MAY) sono normative.

1. Cosa risolve OTEP

Il tracciamento è frammentato: ogni vettore nomina campi e codici di stato in modo diverso, ogni canale di consegna riporta nella propria forma e connettersi agli standard globali significa reintegrare ancora e ancora. OTEP fornisce a produttori e consumatori un unico linguaggio comune: un produttore emette eventi OTEP una sola volta e ogni consumatore OTEP li comprende e può proiettarli verso lo standard di cui ha bisogno.

2. Principi di progettazione

1. Event-sourced. La cronologia degli eventi è la fonte di verità; lo "stato corrente" è sempre una proiezione — lo stato dell'evento più recente.
2. Cosa / Quando / Dove / Perché. Ogni evento è modellato attorno a queste quattro dimensioni — le stesse dimensioni condivise da GS1 EPCIS, IATA ONE Record e UN/CEFACT — così che quegli standard siano proiezioni di output di un evento OTEP anziché modelli paralleli.
3. Le origini senza elenco non sono un ostacolo. Un'origine che espone solo uno stato corrente (senza cronologia) viene gestita sintetizzando un evento per ogni cambiamento osservato (§5).
4. Additivo. OTEP viene esposto accanto a qualsiasi API di tracciamento esistente; adottarlo non richiede mai una modifica incompatibile a ciò che i consumatori già utilizzano.

3. La cronologia OTEP

Una cronologia è un involucro che trasporta il soggetto tracciato e un elenco ordinato di eventi.

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

L'evento 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"
  }
}
```

Ogni campo eccetto `subject`, `occurred_at`, `status_code` e `source` è opzionale — le origini parziali riempiono ciò che hanno. Le scansioni di nodo/hub impostano `location` alla struttura di scansione. La telemetria GPS ad alta frequenza NON è un evento OTEP; un evento viene emesso a un cambiamento di stato o a una scansione di nodo.

4. Vocabolario

4.1 Codici di stato

20 codici canonici del ciclo di vita. `phase` è sempre derivabile dal codice.

code	phase	terminale	POD	significato
`information_submitted`	pre_shipment			informazioni dell'ordine ricevute
`booking_confirmed`	pre_shipment			vettore/prenotazione confermata
`awaiting_pickup`	pre_shipment			pronto per il ritiro
`out_for_pickup`	pickup			in viaggio per il ritiro
`picked_up`	pickup		✓	ritirato dal mittente
`pickup_failed`	exception			tentativo di ritiro fallito
`pickup_rescheduled`	exception			il ritiro verrà ritentato
`received`	inbound			ricevuto presso la struttura
`arrival_scan`	inbound			scansione di arrivo al nodo
`in_transit`	transit			in movimento

code	phase	terminale	POD	significato
`package_outbound`	transit			partito dalla struttura
`removed_from_route`	exception			tolto dal percorso
`route_cancelled`	exception			percorso annullato
`out_for_delivery`	out_for_delivery			sul veicolo
`delivered`	delivered	✓	✓	consegnato al destinatario
`delivery_failed`	exception			tentativo di consegna fallito
`delivery_rescheduled`	exception			verrà ritentato / riconsegnato
`return_to_sender`	return	✓		in restituzione all'origine
`rejected_by_recipient`	return	✓		il destinatario ha rifiutato
`cancelled`	return	✓		ordine annullato

4.2 Fasi

`pre_shipment` · `preparing` · `pickup` · `inbound` · `in_custody` · `transit` · `out_for_delivery` · `delivered` · `exception` · `return` . (`preparing` e `in_custody` sono riservati ai profili non-parcel — §8.)

4.3 Tipi di origine e tipi di tempo

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

4.4 Motivi di incidente

Quando un evento è nella fase `exception` SHOULD riportare un `incident_reason` da questo vocabolario normalizzato:

- Vettore: `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`
- Rivenditore/mittente: `retailer_cancelled` , `retailer_incorrect_data` , `retailer_not_ready` , `retailer_incorrect_parcel` , `retailer_incorrect_dimensions` , `retailer_packaging_issue`
- Destinatario: `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`
- Dogana: `customs_delay` , `customs_documentation` , `customs_duties_unpaid` , `customs_prohibited` , `customs_inspection`
- Forza maggiore: `weather_delay` , `natural_disaster` , `force_majeure`
- Altro: `parcel_being_researched` , `security_issue` , `regulatory_hold` , `unknown`

5. Macchina a stati e origini solo-stato

Le fasi avanzano in avanti; le eccezioni interrompono e si risolvono tornando indietro. Gli stati terminali (`delivered` , `return_to_sender` , `rejected_by_recipient` , `cancelled`) chiudono il soggetto.

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

Regole:

- I consumatori MUST ordinare gli eventi per `occurred_at` e MUST tollerare arrivi fuori ordine.
- Le transizioni verso uno stato terminale sono idempotenti; le ripetizioni vengono deduplicate.

- Dopo uno stato terminale, un produttore MUST NOT emettere ulteriori eventi tranne un flusso documentato di RMA / riapertura.
- Un'origine che espone solo uno stato corrente MUST sintetizzare un evento per ogni cambiamento osservato (con una chiave di deduplicazione stabile) anziché omettere la cronologia. Col tempo gli snapshot si accumulano in una cronologia.

6. Mappature verso standard esterni

Le quattro dimensioni di OTEP si allineano campo per campo con i principali standard, così che ciascuno sia una proiezione di output di un evento 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

I valori per codice (EPCIS CBV `bizStep` / `disposition`, ONE Record `eventCode`, codice UN/CEFACT) sono pubblicati nel codebook leggibile dalla macchina. Oltre a questi standard internazionali, una cronologia OTEP può anche essere proiettata in tracce OpenTelemetry, osservazioni OGC SensorThings e comuni piattaforme di commercio (AfterShip, Shopify, Amazon, Walmart, BigCommerce, Magento, WooCommerce, Etsy).

> Affidabilità: i valori EPCIS CBV sono URN standard stabili. I valori dei codici ONE Record e UN/CEFACT sono il miglior adattamento per l'ultimo miglio e dovrebbero essere validati rispetto agli elenchi di codici ufficiali prima dell'uso esterno. "Delivered to consignee" non ha un bizStep CBV esatto — viene usato l'adattamento più vicino (`receiving` + `received`), oppure un URN di estensione del vocabolario utente.

7. L'API OTEP

OTEP viene consumato su una piccola superficie HTTP di sola lettura; tutti gli endpoint sono pubblici.

Verbo	Percorso	Restituisce
GET	`/api/v1/otep/trackings/{tracking_number}`	la cronologia per un numero di tracciamento
GET	`/api/v1/otep/trackings/{tracking_number}/events`	solo gli eventi
POST	`/api/v1/otep/trackings/batch`	molti numeri di tracciamento in una sola chiamata
POST	`/api/v1/otep/validate`	controllo di conformità per una cronologia inviata (§9)

È disponibile anche una query GraphQL che espone la stessa cronologia.

Negoziazione del contenuto

La stessa cronologia viene serializzata nella rappresentazione richiesta, tramite un parametro di query `?format=` o un profilo `Accept`:

Richiesta	Rappresentazione
`?format=otep` (predefinito)	cronologia OTEP nativa
`?format=epcis`	GS1 EPCIS 2.0 (JSON-LD `ObjectEvent` s)
`?format=onerecord`	IATA ONE Record (`LogisticsEvent` s)

Richiesta	Rappresentazione
`?format=uncefact`	stato di trasporto UN/CEFACT
`?format=otlp`	tracce OpenTelemetry
`?format=sensorthings`	osservazioni OGC SensorThings
`?format=aftership` | `shopify` | `amazon` | `walmart` | `bigcommerce` | `magento` | `woocommerce` | `etsy`	la forma di tracciamento/evasione della piattaforma

Gli eventi a cui non può essere assegnato un codice in una proiezione vengono saltati e conteggiati (mai scartati silenziosamente).

8. Profili di dominio

OTEP è un protocollo generale, non uno per i pacchi. Il livello di protocollo (involucro dell'evento, dorsale delle fasi, macchina a stati) è universale; i codici di stato concreti appartengono a un profilo dichiarato sulla cronologia tramite `profile`. I codici in §4 sono il profilo **`parcel`**. Altri domini — trasloco, consegna di cibo, deposito e oltre — aggiungono i propri set di codici sotto il proprio profilo, con namespace `otep:<profile>:<code>`, ciascuno mappato sulla stessa dorsale delle fasi. Aggiungere un profilo è un'estensione, non una modifica al protocollo.

9. Specifica di conformità (normativa)

Un produttore o consumatore è conforme a OTEP quando ogni evento che emette o accetta soddisfa queste tabelle e regole.

9.1 Involucro della cronologia

Campo	Tipo	Req.	Vincoli
`otep_version`	string	MUST	semver, es. `0.1`
`profile`	string	MUST	un profilo registrato
`subject`	object	MUST	§9.2
`current_status`	string|null	SHOULD	un codice di stato (§4)
`current_phase`	string|null	SHOULD	MUST essere uguale alla fase di `current_status` se entrambi presenti
`delivered`	boolean	SHOULD	`true` se e solo se `current_status` = `delivered`
`events`	array	MUST	oggetti evento (§9.3), ordinabili per `occurred_at`

9.2 `subject`

Almeno UNO tra `tracking_number` / `order_id` / `package_id` MUST essere presente.

Campo	Tipo	Vincoli
`tracking_number`	string	non vuoto
`order_id` / `package_id`	integer|null	
`external_tracking_number`	string|null	
`gs1_sscc`	string|null	18 cifre
`piece_id`	string|null	

9.3 Oggetto evento

#	Campo	Tipo	Req.	Vincoli
1	`occurred_at`	string	MUST	ISO-8601 con offset
2	`recorded_at`	string|null	SHOULD	ISO-8601

#	Campo	Tipo	Req.	Vincoli
3	`time_type`	string	MAY (predefinito `actual`)	`actual` | `estimated` | `scheduled`
4	`status_code`	string|null	MUST[1]	un codice in §4.1
5	`phase`	string|null	SHOULD	MUST essere uguale alla fase di `status_code`
6	`incident_reason`	string|null	SHOULD[2]	un motivo in §4.4
7	`description`	string|null	MAY	leggibile dall'uomo
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] Un evento codificato MUST riportare un codice di stato da §4.1. Una scansione grezza che non puoi ancora classificare MAY impostare `status_code = null`, ma MUST preservare il codice nativo in `source.external_event_code` e MUST essere conteggiata, mai scartata. [2] Un evento in fase `exception` SHOULD riportare un `incident_reason`. [3] Gli eventi con `status_code` ∈ {`delivered`, `picked_up`} SHOULD riportare un `pod`.

9.4 `location`

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

9.5 `source`

Campo	Tipo	Req.	Vincoli
`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]	il tuo codice di stato grezzo
`raw`	object|null	MAY	payload originale

[4] MUST essere presente quando `status_code` è null, così che il codice nativo non venga mai perso.

9.6 Regole

1. Tempo. `occurred_at` MUST essere analizzabile come ISO-8601. Normalizza epoch numeriche e `.NET /Date(ms)/` all'ingestione; non emettere quelle forme.
2. Ordinamento / deduplicazione. I consumatori MUST ordinare per `occurred_at`, tollerare arrivi fuori ordine, e deduplicare su (`subject`, `status_code`, `occurred_at`).
3. Macchina a stati. Dopo uno stato terminale, non emettere ulteriori eventi tranne un RMA / riapertura documentato.
4. Nessuna perdita silenziosa. Gli eventi che non possono essere codificati MUST essere conteggiati, mai scartati.
5. Derivazione. `phase`, `current_status`, `current_phase`, `delivered` sono proiezioni — se presenti MUST essere coerenti con la cronologia degli eventi.

9.7 Livelli di conformità e validazione

- Livello 1 — emette l'involucro (§9.1), eventi con i campi richiesti (§9.3), codici di stato validi (§4.1), fasi valide (§4.2) e rispetta la macchina a stati (§5).
- Livello 2 — emette inoltre almeno una proiezione verso standard esterni (§6) e, ove applicabile, codici specifici del profilo (§8).

Verifica il tuo output inviando una cronologia tramite POST a `POST /api/v1/otep/validate`. Tratta qualsiasi `errors` come bloccante; affronta i `warnings`. Uno schema JSON leggibile dalla macchina e il codebook completo (ogni codice di stato, fase e mappatura esterna) sono pubblicati per la validazione offline.

10. Apertura e governance

OTEP è una specifica aperta, libera per qualsiasi parte da implementare.

- Normativo vs informativo. Normativi: l'involucro dell'evento, la dorsale delle fasi, il vocabolario di stato, la macchina a stati e le mappature dei campi verso gli standard esterni. Il modo in cui un implementatore lega OTEP ai propri sistemi interni è una sua questione e fuori dall'ambito di questo documento.
- Identificatori stabili. I codici sono indirizzati come `otep:<profile>:<code>`, le fasi come `otep:phase:<name>`. Una volta pubblicato in una versione rilasciata, il significato di un identificatore è immutabile.
- Versionamento. Versionamento semantico. Aggiungere codici/profili è una modifica MINOR (compatibile all'indietro); cambiare il significato di un codice esistente è una modifica MAJOR e SHOULD essere evitato. La versione del protocollo viaggia con ogni cronologia (`otep_version`).
- Estensione. I nuovi profili e codici vengono proposti rispetto a questa specifica anziché creando un fork, così che gli implementatori indipendenti convergano. I codici sperimentali MAY usare un prefisso `x-` (`otep:parcel:x-my_code`) finché non sono registrati.
- Licenza. La specifica è destinata a essere rilasciata sotto una licenza aperta — da definire, in attesa di approvazione.

11. Interoperabilità tra fornitori — porta il tuo standard

OTEP accoglie altri fornitori che portano il proprio standard di eventi di tracciamento affinché OTEP possa interoperare con esso, in entrambe le direzioni:

- Proietta OTEP → il tuo standard. Definisci una mappatura da una cronologia OTEP al tuo formato, riutilizzando il vocabolario di stato OTEP. È una pura trasformazione — cronologia in ingresso, la tua struttura in uscita — così che la mappatura sia scritta una sola volta e ogni produttore OTEP possa emettere il tuo formato.
- Mappa il tuo standard → OTEP. Fornisci un crosswalk dal tuo vocabolario di stato verso i codici OTEP (§4) più, se necessario, un profilo (§8). I tuoi codici grezzi sono preservati in `source.external_event_code`; i codici non mappati vengono conteggiati, mai scartati.

Anche se non puoi adottare OTEP direttamente, sei il benvenuto ad aggiungere un singolo `otep_status` normalizzato alle risposte della tua API e a condividere i tuoi codici di eventi di tracciamento per la mappatura crosswalk. Proponi le mappature rispetto a questa specifica (anziché creare un fork) così che gli implementatori convergano; i nuovi formati si collegano alla stessa negoziazione del contenuto `?format=` e non rompono mai un consumatore esistente.