OTEP — Open Tracking Event Protocol

Traducción por conveniencia — la versión en inglés es la autoritativa.

> Estado: Borrador v0.1 · Una especificación abierta y neutral respecto al proveedor · Licencia: abierta (ver §10)

OTEP es un protocolo abierto y neutral respecto al proveedor para el ciclo de vida de cualquier sujeto rastreable. Define un único modelo de eventos y un único vocabulario de estados para que los eventos de seguimiento de cualquier fuente — entrega con flota propia, mensajeros de terceros, etiquetas de transportistas y más allá — puedan ser intercambiados, comprendidos y proyectados a estándares internacionales sin necesidad de reintegrar para cada parte.

Este documento es la especificación: la estructura, los campos, las tablas de estados, la máquina de estados, las correspondencias con estándares externos y las reglas de conformidad para construir una implementación conforme. Es independiente de la implementación: describe el protocolo, no los detalles internos de ningún proveedor en particular. Las palabras clave de RFC-2119 (MUST / SHOULD / MAY) son normativas.

1. Qué resuelve OTEP

El seguimiento está fragmentado: cada transportista nombra los campos y códigos de estado de forma diferente, cada canal de entrega reporta con su propia forma, y conectar con estándares globales significa reintegrar una y otra vez. OTEP brinda a productores y consumidores un único lenguaje común: un productor emite eventos OTEP una sola vez, y cada consumidor OTEP los comprende y puede proyectarlos al estándar que necesite.

2. Principios de diseño

1. Basado en eventos. La línea temporal de eventos es la fuente de la verdad; el "estado actual" es siempre una proyección — el estado del último evento.
2. Qué / Cuándo / Dónde / Por qué. Cada evento se estructura en torno a estas cuatro dimensiones — las mismas dimensiones compartidas por GS1 EPCIS, IATA ONE Record y UN/CEFACT — de modo que esos estándares son proyecciones de salida de un evento OTEP en lugar de modelos paralelos.
3. Las fuentes sin lista no son un obstáculo. Una fuente que expone solo un estado actual (sin historial) se gestiona sintetizando un evento por cada cambio observado (§5).
4. Aditivo. OTEP se expone junto a cualquier API de seguimiento existente; adoptarlo nunca requiere un cambio disruptivo en lo que los consumidores ya usan.

3. La línea temporal OTEP

Una línea temporal es un sobre que transporta el sujeto rastreado y una lista ordenada de eventos.

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

El 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"
  }
}
```

Todos los campos excepto `subject`, `occurred_at`, `status_code` y `source` son opcionales — las fuentes parciales rellenan lo que tienen. Los escaneos de nodos/centros establecen `location` en la instalación de escaneo. La telemetría GPS de alta frecuencia NO es un evento OTEP; un evento se emite ante un cambio de estado o un escaneo de nodo.

4. Vocabulario

4.1 Códigos de estado

20 códigos canónicos de ciclo de vida. `phase` siempre se puede derivar del código.

code	phase	terminal	POD	significado
`information_submitted`	pre_shipment			información del pedido recibida
`booking_confirmed`	pre_shipment			reserva/transportista confirmado
`awaiting_pickup`	pre_shipment			listo para recolección
`out_for_pickup`	pickup			en ruta para recoger
`picked_up`	pickup		✓	recogido del remitente
`pickup_failed`	exception			intento de recolección fallido
`pickup_rescheduled`	exception			la recolección se reintentará
`received`	inbound			recibido en instalación
`arrival_scan`	inbound			escaneo de llegada en nodo
`in_transit`	transit			en movimiento

code	phase	terminal	POD	significado
`package_outbound`	transit			salió de la instalación
`removed_from_route`	exception			retirado de la ruta
`route_cancelled`	exception			ruta cancelada
`out_for_delivery`	out_for_delivery			en el vehículo
`delivered`	delivered	✓	✓	entregado al destinatario
`delivery_failed`	exception			intento de entrega fallido
`delivery_rescheduled`	exception			se reintentará / reentregará
`return_to_sender`	return	✓		regresando al origen
`rejected_by_recipient`	return	✓		el destinatario rechazó
`cancelled`	return	✓		pedido cancelado

4.2 Fases

`pre_shipment` · `preparing` · `pickup` · `inbound` · `in_custody` · `transit` · `out_for_delivery` · `delivered` · `exception` · `return`. (`preparing` y `in_custody` están reservados para perfiles que no son de paquetería — §8.)

4.3 Tipos de fuente y tipos de tiempo

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

4.4 Razones de incidente

Cuando un evento está en la fase `exception` SHOULD llevar un `incident_reason` de este vocabulario normalizado:

- Transportista: `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`
- Minorista/remitente: `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`
- Aduanas: `customs_delay`, `customs_documentation`, `customs_duties_unpaid`, `customs_prohibited`, `customs_inspection`
- Fuerza mayor: `weather_delay`, `natural_disaster`, `force_majeure`
- Otros: `parcel_being_researched`, `security_issue`, `regulatory_hold`, `unknown`

5. Máquina de estados y fuentes de solo estado

Las fases avanzan hacia adelante; las excepciones interrumpen y se resuelven hacia atrás. Los estados terminales (`delivered`, `return_to_sender`, `rejected_by_recipient`, `cancelled`) cierran el sujeto.

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

Reglas:

- Los consumidores MUST ordenar los eventos por `occurred_at` y MUST tolerar la llegada en desorden.

- Las transiciones a un estado terminal son idempotentes; las repeticiones se deduplican.
- Después de un estado terminal, un productor MUST NOT emitir más eventos salvo un flujo documentado de RMA / reapertura.
- Una fuente que expone solo un estado actual MUST sintetizar un evento por cada cambio observado (con una clave de deduplicación estable) en lugar de omitir el historial. Con el tiempo, las instantáneas se acumulan en una línea temporal.

6. Correspondencias con estándares externos

Las cuatro dimensiones de OTEP se alinean campo por campo con los principales estándares, de modo que cada uno es una proyección de salida de un evento OTEP.

OTEP	GS1 EPCIS 2.0	IATA ONE Record	UN/CEFACT
`occurred_at` (+offset)	`eventTime` + `eventTimeZoneOffset`	`eventDate` + `eventTimeType=Actual`	Fecha/hora del evento
`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`	Código de estado de transporte
`actor`	sourceList / extension	Actor / Party	Party
`incident_reason`	disposition / ErrorDeclaration	event remark	Código de razón de estado

Los valores por código (CBV de EPCIS `bizStep` / `disposition`, `eventCode` de ONE Record, código de UN/CEFACT) se publican en el libro de códigos legible por máquina. Más allá de estos estándares internacionales, una línea temporal OTEP también puede proyectarse a trazas de OpenTelemetry, observaciones de OGC SensorThings y plataformas de comercio comunes (AfterShip, Shopify, Amazon, Walmart, BigCommerce, Magento, WooCommerce, Etsy).

> Confianza: los valores CBV de EPCIS son URNs estándar estables. Los valores de código de ONE Record y UN/CEFACT son el mejor ajuste para la última milla y deben validarse contra las listas de códigos oficiales antes de su uso externo. "Entregado al destinatario" no tiene un bizStep CBV exacto — se usa el ajuste más cercano (`receiving` + `received`), o una URN de extensión de vocabulario de usuario.

7. La API de OTEP

OTEP se consume a través de una pequeña superficie HTTP de solo lectura; todos los endpoints son públicos.

Verbo	Ruta	Devuelve
GET	`/api/v1/otep/trackings/{tracking_number}`	la línea temporal de un número de seguimiento
GET	`/api/v1/otep/trackings/{tracking_number}/events`	solo eventos
POST	`/api/v1/otep/trackings/batch`	muchos números de seguimiento en una sola llamada
GET	`/api/v1/otep/trackings/{tracking_number}?view=public`	vista pública segura: sin nombres de conductores ni identificadores internos, limitada a la empresa del dominio, también números de mudanza y almacenamiento
GET	`/api/v1/client/otep/trackings/{identifier}`	(token de API) línea de tiempo y detalles de su propio envío: paquete, mudanza, almacenamiento
GET	`/api/v1/client/otep/locker-storage-rentals/{id}`	(token de API) línea de tiempo de un depósito en taquilla, por id del registro
GET	`/oteptrack`	página pública de seguimiento: paquete, mudanza, almacenamiento; detalles tras verificar
POST	`/api/v1/otep/validate`	verificación de conformidad para una línea temporal enviada (§9)

También está disponible una consulta GraphQL que expone la misma línea temporal.

Negociación de contenido

La misma línea temporal se serializa en cualquier representación que solicites, mediante un parámetro de consulta `?format=` o un perfil `Accept`:

Solicitud	Representación
`?format=otep` (por defecto)	línea temporal OTEP nativa
`?format=epcis`	GS1 EPCIS 2.0 (JSON-LD `ObjectEvent`s)
`?format=onerecord`	IATA ONE Record (`LogisticsEvent`s)
`?format=uncefact`	estado de transporte UN/CEFACT
`?format=otlp`	trazas de OpenTelemetry
`?format=sensorthings`	observaciones de OGC SensorThings
`?format=aftership` | `shopify` | `amazon` | `walmart` | `bigcommerce` | `magento` | `woocommerce` | `etsy`	la forma de seguimiento/cumplimiento de la plataforma

Los eventos que no pueden asignarse a un código en una proyección se omiten y se contabilizan (nunca se descartan silenciosamente).

8. Perfiles de dominio

OTEP es un protocolo general, no uno de paquetería. La capa de protocolo (sobre del evento, columna de fases, máquina de estados) es universal; los códigos de estado concretos pertenecen a un perfil declarado en la línea temporal mediante `profile`. Los códigos de §4 son el perfil **`parcel`**. Otros dominios — mudanzas, entrega de alimentos, almacenamiento y más allá — agregan sus propios conjuntos de códigos bajo su perfil, con espacio de nombres `otep:<profile>:<code>`, cada uno mapeándose a la misma columna de fases. Agregar un perfil es una extensión, no un cambio de protocolo.

9. Especificación de conformidad (normativa)

Un productor o consumidor es conforme a OTEP cuando cada evento que emite o acepta satisface estas tablas y reglas.

9.1 Sobre de la línea temporal

Campo	Tipo	Req.	Restricciones
`otep_version`	string	MUST	semver, p. ej. `0.1`
`profile`	string	MUST	un perfil registrado
`subject`	object	MUST	§9.2
`current_status`	string|null	SHOULD	un código de estado (§4)
`current_phase`	string|null	SHOULD	MUST ser igual a la fase de `current_status` si ambos están presentes
`delivered`	boolean	SHOULD	`true` si y solo si `current_status` = `delivered`
`events`	array	MUST	objetos de evento (§9.3), ordenables por `occurred_at`

9.2 `subject`

Al menos UNO de `tracking_number` / `order_id` / `package_id` MUST estar presente.

Campo	Tipo	Restricciones
`tracking_number`	string	no vacío
`order_id` / `package_id`	integer|null	
`external_tracking_number`	string|null	

Campo	Tipo	Restricciones
`gs1_sscc`	string|null	18 dígitos
`piece_id`	string|null	

9.3 Objeto de evento

#	Campo	Tipo	Req.	Restricciones
1	`occurred_at`	string	MUST	ISO-8601 con offset
2	`recorded_at`	string|null	SHOULD	ISO-8601
3	`time_type`	string	MAY (por defecto `actual`)	`actual` | `estimated` | `scheduled`
4	`status_code`	string|null	MUST[1]	un código en §4.1
5	`phase`	string|null	SHOULD	MUST ser igual a la fase de `status_code`
6	`incident_reason`	string|null	SHOULD[2]	una razón en §4.4
7	`description`	string|null	MAY	legible por humanos
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 codificado MUST llevar un código de estado de §4.1. Un escaneo en bruto que aún no puedes clasificar MAY establecer `status_code = null`, pero MUST preservar el código nativo en `source.external_event_code` y MUST contabilizarse, nunca descartarse. [2] Un evento de fase `exception` SHOULD llevar un `incident_reason`. [3] Los eventos con `status_code` ∈ {`delivered`, `picked_up`} SHOULD llevar un `pod`.

9.4 `location`

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

9.5 `source`

Campo	Tipo	Req.	Restricciones
`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]	tu código de estado en bruto
`raw`	object|null	MAY	carga útil original

[4] MUST estar presente cuando `status_code` es null, para que el código nativo nunca se pierda.

9.6 Reglas

1. Tiempo. `occurred_at` MUST analizarse como ISO-8601. Normaliza las épocas numéricas y `.NET /Date(ms)/` en la ingesta; no emitas esas formas.
2. Ordenación / deduplicación. Los consumidores MUST ordenar por `occurred_at`, tolerar la llegada en desorden, y deduplicar en (`subject`, `status_code`, `occurred_at`).
3. Máquina de estados. Después de un estado terminal, no emitas más eventos salvo un flujo documentado de RMA / reapertura.
4. Sin pérdidas silenciosas. Los eventos que no se pueden codificar MUST contabilizarse, nunca descartarse.

5. Derivación. `phase`, `current_status`, `current_phase`, `delivered` son proyecciones — si están presentes MUST ser consistentes con la línea temporal de eventos.

9.7 Niveles de conformidad y validación

- Nivel 1 — emite el sobre (§9.1), eventos con los campos requeridos (§9.3), códigos de estado válidos (§4.1), fases válidas (§4.2), y respeta la máquina de estados (§5).
- Nivel 2 — además emite al menos una proyección de estándar externo (§6) y, donde corresponda, códigos específicos de perfil (§8).

Verifica tu salida enviando una línea temporal mediante POST a `POST /api/v1/otep/validate`. Trata cualquier `errors` como bloqueante; atiende los `warnings`. Se publican un JSON Schema legible por máquina y el libro de códigos completo (cada código de estado, fase y correspondencia externa) para validación sin conexión.

10. Apertura y gobernanza

OTEP es una especificación abierta, libre para que cualquier parte la implemente.

- Normativo vs informativo. Normativo: el sobre del evento, la columna de fases, el vocabulario de estados, la máquina de estados y las correspondencias de campos con estándares externos. Cómo un implementador vincula OTEP a sus propios sistemas internos es asunto suyo y queda fuera del alcance aquí.
- Identificadores estables. Los códigos se direccionan como `otep:<profile>:<code>`, las fases como `otep:phase:<name>`. Una vez publicado en una versión liberada, el significado de un identificador es inmutable.
- Versionado. Versionado semántico. Agregar códigos/perfiles es un cambio MINOR (compatible hacia atrás); cambiar el significado de un código existente es un cambio MAJOR y SHOULD evitarse. La versión del protocolo viaja con cada línea temporal (`otep_version`).
- Extensión. Los nuevos perfiles y códigos se proponen contra esta especificación en lugar de bifurcarse, de modo que los implementadores independientes converjan. Los códigos experimentales MAY usar un prefijo `x-` (`otep:parcel:x-my_code`) hasta que se registren.
- Licencia. La especificación está destinada a publicarse bajo una licencia abierta — por determinar, pendiente de aprobación.

11. Interoperabilidad entre proveedores — trae tu propio estándar

OTEP da la bienvenida a otros proveedores para que traigan su propio estándar de eventos de seguimiento, de modo que OTEP pueda interoperar con él, en cualquier dirección:

- Proyectar OTEP → tu estándar. Define una correspondencia de una línea temporal OTEP a tu formato, reutilizando el vocabulario de estados de OTEP. Es una transformación pura — línea temporal de entrada, tu estructura de salida — de modo que la correspondencia se escribe una sola vez y cada productor OTEP puede emitir tu formato.
- Mapear tu estándar → OTEP. Proporciona una equivalencia de tu vocabulario de estados a los códigos de OTEP (§4) más, si es necesario, un perfil (§8). Tus códigos en bruto se preservan en `source.external_event_code`; los códigos no mapeados se contabilizan, nunca se descartan.

Incluso si no puedes adoptar OTEP directamente, eres bienvenido a agregar un único `otep_status` normalizado a las respuestas de tu propia API y a compartir tus códigos de eventos de seguimiento para la correspondencia de equivalencias. Propón correspondencias contra esta especificación (en lugar de bifurcar) para que los implementadores converjan; los nuevos formatos se integran en la misma negociación de contenido `?format=` y nunca rompen un consumidor existente.