OTEP — Open Tracking Event Protocol

Tradução por conveniência — a versão em inglês é a autoritativa.

> Estado: Rascunho v0.1 · Uma especificação aberta e neutra em relação a fornecedores · Licença: aberta (ver §10)

OTEP é um protocolo aberto e neutro em relação a fornecedores para o ciclo de vida de qualquer sujeito rastreável. Ele define um modelo de evento e um vocabulário de estado para que os eventos de rastreamento de qualquer origem — entrega por frota própria, transportadoras terceirizadas, etiquetas de transportadoras e além — possam ser trocados, compreendidos e projetados para padrões internacionais sem reintegrar para cada parte.

Este documento é a especificação: a estrutura, os campos, as tabelas de estado, a máquina de estados, os mapeamentos para padrões externos e as regras de conformidade para construir uma implementação compatível. É independente da implementação — descreve o protocolo, não os componentes internos de qualquer fornecedor em particular. As palavras-chave RFC-2119 (MUST / SHOULD / MAY) são normativas.

1. O que o OTEP resolve

O rastreamento é fragmentado: cada transportadora nomeia campos e códigos de estado de forma diferente, cada canal de entrega reporta no seu próprio formato, e conectar-se a padrões globais significa reintegrar repetidamente. O OTEP dá aos produtores e consumidores uma única linguagem comum: um produtor emite eventos OTEP uma vez, e todo consumidor OTEP os compreende e pode projetá-los para o padrão de que precisa.

2. Princípios de design

1. Baseado em eventos. A linha do tempo de eventos é a fonte da verdade; o "estado atual" é sempre uma projeção — o estado do evento mais recente.
2. O quê / Quando / Onde / Por quê. Cada evento é moldado em torno destas quatro dimensões — as mesmas dimensões partilhadas por GS1 EPCIS, IATA ONE Record e UN/CEFACT — de modo que esses padrões são projeções de saída de um evento OTEP em vez de modelos paralelos.
3. Origens sem lista não são um obstáculo. Uma origem que expõe apenas um estado atual (sem histórico) é tratada sintetizando um evento por cada alteração observada (§5).
4. Aditivo. O OTEP é exposto ao lado de qualquer API de rastreamento existente; adotá-lo nunca exige uma alteração disruptiva ao que os consumidores já utilizam.

3. A linha do tempo OTEP

Uma linha do tempo é um envelope que carrega o sujeito rastreado e uma 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 */ ]
}
```

O 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"
  }
}
```

Cada campo, exceto `subject`, `occurred_at`, `status_code` e `source`, é opcional — as origens parciais preenchem o que têm. As leituras de nó/hub definem `location` para a instalação de leitura. A telemetria GPS de alta frequência NÃO é um evento OTEP; um evento é emitido numa alteração de estado ou numa leitura de nó.

4. Vocabulário

4.1 Códigos de estado

20 códigos canónicos de ciclo de vida. `phase` é sempre derivável do código.

código	phase	terminal	POD	significado
`information_submitted`	pre_shipment			informação do pedido recebida
`booking_confirmed`	pre_shipment			transportadora/reserva confirmada
`awaiting_pickup`	pre_shipment			pronto para recolha
`out_for_pickup`	pickup			a caminho da recolha
`picked_up`	pickup		✓	recolhido do expedidor
`pickup_failed`	exception			tentativa de recolha falhou
`pickup_rescheduled`	exception			a recolha será repetida
`received`	inbound			recebido na instalação
`arrival_scan`	inbound			leitura de chegada no nó
`in_transit`	transit			em movimento

código	phase	terminal	POD	significado
`package_outbound`	transit			saiu da instalação
`removed_from_route`	exception			retirado da rota
`route_cancelled`	exception			rota cancelada
`out_for_delivery`	out_for_delivery			no veículo
`delivered`	delivered	✓	✓	entregue ao destinatário
`delivery_failed`	exception			tentativa de entrega falhou
`delivery_rescheduled`	exception			será repetida / reentregue
`return_to_sender`	return	✓		a retornar à origem
`rejected_by_recipient`	return	✓		destinatário recusou
`cancelled`	return	✓		pedido cancelado

4.2 Fases

`pre_shipment` · `preparing` · `pickup` · `inbound` · `in_custody` · `transit` · `out_for_delivery` · `delivered` · `exception` · `return` . (`preparing` e `in_custody` estão reservados para perfis que não sejam de encomendas — §8.)

4.3 Tipos de origem e tipos de tempo

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

4.4 Motivos de incidente

Quando um evento está na fase `exception` , ele SHOULD carregar um `incident_reason` deste vocabulário normalizado:

- 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. Máquina de estados e origens apenas de estado

As fases avançam para a frente; as exceções interrompem e resolvem-se de volta. Os estados terminais (`delivered` , `return_to_sender` , `rejected_by_recipient` , `cancelled`) encerram o sujeito.

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

Regras:

- Os consumidores MUST ordenar os eventos por `occurred_at` e MUST tolerar a chegada fora de ordem.
- As transições para um estado terminal são idempotentes; as repetições são desduplicadas.

- Após um estado terminal, um produtor MUST NOT emitir mais eventos, exceto um fluxo documentado de RMA / reabertura.
- Uma origem que expõe apenas um estado atual MUST sintetizar um evento por cada alteração observada (com uma chave de desduplicação estável) em vez de omitir o histórico. Ao longo do tempo, os instantâneos acumulam-se numa linha do tempo.

6. Mapeamentos para padrões externos

As quatro dimensões do OTEP alinham-se campo a campo com os principais padrões, de modo que cada um é uma projeção de saída de um 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

Os valores por código (EPCIS CBV `bizStep` / `disposition`, ONE Record `eventCode`, código UN/CEFACT) são publicados no codebook legível por máquina. Para além destes padrões internacionais, uma linha do tempo OTEP também pode ser projetada para traces de OpenTelemetry, observações de OGC SensorThings e plataformas de comércio comuns (AfterShip, Shopify, Amazon, Walmart, BigCommerce, Magento, WooCommerce, Etsy).

> Confiança: os valores EPCIS CBV são URNs de padrão estáveis. Os valores de código de ONE Record e UN/CEFACT são a melhor aproximação para a última milha e devem ser validados contra as listas de códigos oficiais antes do uso externo. "Delivered to consignee" não tem um bizStep CBV exato — a aproximação mais próxima (`receiving` + `received`) é utilizada, ou uma URN de extensão de vocabulário do utilizador.

7. A API OTEP

O OTEP é consumido através de uma pequena superfície HTTP apenas de leitura; todos os endpoints são públicos.

Verbo	Caminho	Retorna
GET	`/api/v1/otep/trackings/{tracking_number}`	a linha do tempo para um número de rastreamento
GET	`/api/v1/otep/trackings/{tracking_number}/events`	apenas eventos
POST	`/api/v1/otep/trackings/batch`	vários números de rastreamento numa só chamada
POST	`/api/v1/otep/validate`	verificação de conformidade para uma linha do tempo enviada (§9)

Também está disponível uma consulta GraphQL que expõe a mesma linha do tempo.

Negociação de conteúdo

A mesma linha do tempo é serializada na representação que solicitar, através de um parâmetro de consulta `?format=` ou de um perfil `Accept`:

Solicitação	Representação
`?format=otep` (padrão)	linha do tempo OTEP nativa
`?format=epcis`	GS1 EPCIS 2.0 (JSON-LD `ObjectEvent`s)

Solicitação	Representação
`?format=onerecord`	IATA ONE Record (`LogisticsEvent` s)
`?format=uncefact`	estado de transporte UN/CEFACT
`?format=otlp`	traces de OpenTelemetry
`?format=sensorthings`	observações de OGC SensorThings
`?format=aftership` | `shopify` | `amazon` | `walmart` | `bigcommerce` | `magento` | `woocommerce` | `etsy`	o formato de rastreamento/fulfillment da plataforma

Os eventos que não podem receber um código numa projeção são ignorados e contabilizados (nunca silenciosamente descartados).

8. Perfis de domínio

O OTEP é um protocolo geral, não um protocolo de encomendas. A camada de protocolo (envelope de evento, espinha de fases, máquina de estados) é universal; os códigos de estado concretos pertencem a um perfil declarado na linha do tempo através de `profile`. Os códigos em §4 são o perfil **`parcel`**. Outros domínios — mudanças, entrega de comida, armazenamento e além — adicionam os seus próprios conjuntos de códigos sob o seu perfil, com namespace `otep:<profile>:<code>`, cada um mapeando para a mesma espinha de fases. Adicionar um perfil é uma extensão, não uma alteração de protocolo.

9. Especificação de conformidade (normativa)

Um produtor ou consumidor é conforme com OTEP quando cada evento que emite ou aceita satisfaz estas tabelas e regras.

9.1 Envelope da linha do tempo

Campo	Tipo	Req.	Restrições
`otep_version`	string	MUST	semver, e.g. `0.1`
`profile`	string	MUST	um perfil registado
`subject`	object	MUST	§9.2
`current_status`	string|null	SHOULD	um código de estado (§4)
`current_phase`	string|null	SHOULD	MUST ser igual à fase de `current_status` se ambos estiverem presentes
`delivered`	boolean	SHOULD	`true` se e só se `current_status = delivered`
`events`	array	MUST	objetos de evento (§9.3), ordenáveis por `occurred_at`

9.2 **`subject`**

Pelo menos UM de `tracking_number` / `order_id` / `package_id` MUST estar presente.

Campo	Tipo	Restrições
`tracking_number`	string	não vazio
`order_id` / `package_id`	integer|null	
`external_tracking_number`	string|null	
`gs1_sscc`	string|null	18 dígitos
`piece_id`	string|null	

9.3 Objeto de evento

#	Campo	Tipo	Req.	Restrições
1	`occurred_at`	string	MUST	ISO-8601 com offset

#	Campo	Tipo	Req.	Restrições
2	`recorded_at`	string|null	SHOULD	ISO-8601
3	`time_type`	string	MAY (padrão `actual`)	`actual` | `estimated` | `scheduled`
4	`status_code`	string|null	MUST[1]	um código em §4.1
5	`phase`	string|null	SHOULD	MUST ser igual à fase de `status_code`
6	`incident_reason`	string|null	SHOULD[2]	um motivo em §4.4
7	`description`	string|null	MAY	legível 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] Um evento codificado MUST carregar um código de estado de §4.1. Uma leitura bruta que ainda não consegue classificar MAY definir `status_code = null`, mas MUST preservar o código nativo em `source.external_event_code` e MUST ser contabilizado, nunca descartado. [2] Um evento na fase `exception` SHOULD carregar um `incident_reason`. [3] Os eventos com `status_code` ∈ {`delivered`, `picked_up`} SHOULD carregar um `pod`.

9.4 `location`

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

9.5 `source`

Campo	Tipo	Req.	Restrições
`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]	o seu código de estado bruto
`raw`	object|null	MAY	payload original

[4] MUST estar presente quando `status_code` é null, para que o código nativo nunca se perca.

9.6 Regras

1. Tempo. `occurred_at` MUST ser analisável como ISO-8601. Normalize epochs numéricos e `.NET /Date(ms)/` na ingestão; não emita essas formas.
2. Ordenação / desduplicação. Os consumidores MUST ordenar por `occurred_at`, tolerar a chegada fora de ordem, e desduplicar por (`subject`, `status_code`, `occurred_at`).
3. Máquina de estados. Após um estado terminal, não emita mais eventos, exceto um RMA / reabertura documentado.
4. Sem perda silenciosa. Os eventos que não podem ser codificados MUST ser contabilizados, nunca descartados.
5. Derivação. `phase`, `current_status`, `current_phase`, `delivered` são projeções — se presentes, MUST ser consistentes com a linha do tempo de eventos.

9.7 Níveis de conformidade e validação

- Nível 1 — emite o envelope (§9.1), eventos com os campos obrigatórios (§9.3), códigos de estado válidos (§4.1), fases válidas (§4.2), e respeita a máquina de estados (§5).
- Nível 2 — adicionalmente emite pelo menos uma projeção para padrão externo (§6) e, quando aplicável, códigos específicos do perfil (§8).

Verifique a sua saída enviando uma linha do tempo via POST para `POST /api/v1/otep/validate`. Trate quaisquer `errors` como bloqueadores; resolva os `warnings`. Um JSON Schema legível por máquina e o codebook completo (cada código de estado, fase e mapeamento externo) são publicados para validação offline.

10. Abertura e governação

O OTEP é uma especificação aberta, livre para qualquer parte implementar.

- Normativo vs informativo. Normativo: o envelope de evento, a espinha de fases, o vocabulário de estado, a máquina de estados e os mapeamentos de campo para padrões externos. Como um implementador vincula o OTEP aos seus próprios sistemas internos é da sua própria responsabilidade e está fora do âmbito aqui.
- Identificadores estáveis. Os códigos são endereçados como `otep:<profile>:<code>`, as fases como `otep:phase:<name>`. Uma vez publicado numa versão lançada, o significado de um identificador é imutável.
- Versionamento. Versionamento semântico. Adicionar códigos/perfis é uma alteração MINOR (compatível com versões anteriores); alterar o significado de um código existente é uma alteração MAJOR e SHOULD ser evitada. A versão do protocolo viaja com cada linha do tempo (`otep_version`).
- Extensão. Os novos perfis e códigos são propostos contra esta especificação em vez de serem bifurcados, para que os implementadores independentes convirjam. Os códigos experimentais MAY usar um prefixo `x-` (`otep:parcel:x-my_code`) até serem registados.
- Licença. A especificação destina-se a ser lançada sob uma licença aberta — TBD, pendente de aprovação.

11. Interoperabilidade entre fornecedores — traga o seu próprio padrão

O OTEP convida outros fornecedores a trazerem o seu próprio padrão de evento de rastreamento para que o OTEP possa interoperar com ele, em qualquer direção:

- Projetar OTEP → o seu padrão. Defina um mapeamento de uma linha do tempo OTEP para o seu formato, reutilizando o vocabulário de estado OTEP. É uma transformação pura — linha do tempo na entrada, a sua estrutura na saída — pelo que o mapeamento é escrito uma vez e cada produtor OTEP pode emitir o seu formato.
- Mapear o seu padrão → OTEP. Forneça um crosswalk do seu vocabulário de estado para os códigos OTEP (§4) mais, se necessário, um perfil (§8). Os seus códigos brutos são preservados em `source.external_event_code`; os códigos não mapeados são contabilizados, nunca descartados.

Mesmo que não consiga adotar o OTEP diretamente, é bem-vindo a adicionar um único `otep_status` normalizado às respostas da sua própria API e a partilhar os seus códigos de evento de rastreamento para mapeamento de crosswalk. Proponha os mapeamentos contra esta especificação (em vez de bifurcar) para que os implementadores convirjam; os novos formatos integram-se na mesma negociação de conteúdo `?format=` e nunca quebram um consumidor existente.