OTEP — Open Tracking Event Protocol

Tłumaczenie dla wygody — wersja angielska jest wiążąca.

> Status: Wersja robocza v0.1 · Otwarta, neutralna wobec dostawców specyfikacja · Licencja: otwarta (zob. §10)

OTEP to otwarty, neutralny wobec dostawców protokół opisujący cykl życia dowolnego śledzonego obiektu. Definiuje jeden model zdarzeń i jeden słownik statusów, tak aby zdarzenia śledzenia z dowolnego źródła — dostawy własną flotą, przewoźnicy zewnętrzni, etykiety przewoźników i nie tylko — mogły być wymieniane, rozumiane i rzutowane na standardy międzynarodowe bez ponownej integracji dla każdej ze stron.

Ten dokument jest specyfikacją: struktura, pola, tabele statusów, maszyna stanów, mapowania na standardy zewnętrzne oraz reguły zgodności pozwalające zbudować zgodną implementację. Jest niezależny od implementacji — opisuje protokół, a nie wewnętrzne mechanizmy konkretnego dostawcy. Słowa kluczowe RFC-2119 (MUST / SHOULD / MAY) mają charakter normatywny.

1. Co rozwiązuje OTEP

Śledzenie jest rozdrobnione: każdy przewoźnik inaczej nazywa pola i kody statusów, każdy kanał dostawy raportuje we własnej postaci, a połączenie z globalnymi standardami oznacza kolejne integracje od nowa. OTEP daje producentom i konsumentom jeden wspólny język: producent emituje zdarzenia OTEP raz, a każdy konsument OTEP je rozumie i może je rzutować na potrzebny mu standard.

2. Zasady projektowe

1. Oparte na zdarzeniach (event-sourced). Oś czasu zdarzeń jest źródłem prawdy; „bieżący status” jest zawsze rzutem — statusem najnowszego zdarzenia.
2. Co / Kiedy / Gdzie / Dlaczego. Każde zdarzenie jest ukształtowane wokół tych czterech wymiarów — tych samych, które dzielą GS1 EPCIS, IATA ONE Record i UN/CEFACT — dzięki czemu standardy te są rzutami wyjściowymi zdarzenia OTEP, a nie równoległymi modelami.
3. Źródła bez listy nie stanowią przeszkody. Źródło, które udostępnia tylko bieżący status (bez historii), jest obsługiwane przez syntezę jednego zdarzenia na każdą zaobserwowaną zmianę (§5).
4. Addytywne. OTEP jest udostępniany obok dowolnego istniejącego API śledzenia; jego przyjęcie nigdy nie wymaga niekompatybilnej zmiany w tym, czego konsumenci już używają.

3. Oś czasu OTEP

Oś czasu to koperta przenosząca śledzony obiekt oraz uporządkowaną listę zdarzeń.

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

Zdarzenie 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"
  }
}
```

Każde pole z wyjątkiem `subject`, `occurred_at`, `status_code` i `source` jest opcjonalne — częściowe źródła wypełniają to, co posiadają. Skany w węzłach/hubach ustawiają `location` na obiekt skanujący. Wysokoczęstotliwościowa telemetria GPS NIE jest zdarzeniem OTEP; zdarzenie jest emitowane przy zmianie statusu lub skanie w węźle.

4. Słownik

4.1 Kody statusów

20 kanonicznych kodów cyklu życia. `phase` jest zawsze wyprowadzalna z kodu.

code	phase	terminalny	POD	znaczenie
`information_submitted`	pre_shipment			otrzymano informacje o zamówieniu
`booking_confirmed`	pre_shipment			potwierdzono przewoźnika/rezerwację
`awaiting_pickup`	pre_shipment			gotowe do odbioru
`out_for_pickup`	pickup			w drodze po odbiór
`picked_up`	pickup		✓	odebrano od nadawcy
`pickup_failed`	exception			próba odbioru nieudana
`pickup_rescheduled`	exception			odbiór zostanie powtórzony
`received`	inbound			przyjęto w obiekcie
`arrival_scan`	inbound			skan przyjazdu w węźle
`in_transit`	transit			w ruchu
`package_outbound`	transit			opuściło obiekt
`removed_from_route`	exception			zdjęto z trasy

code	phase	terminalny	POD	znaczenie
`route_cancelled`	exception			trasa anulowana
`out_for_delivery`	out_for_delivery			w pojeździe
`delivered`	delivered	✓	✓	dostarczono do odbiorcy
`delivery_failed`	exception			próba dostawy nieudana
`delivery_rescheduled`	exception			zostanie powtórzone / ponowna dostawa
`return_to_sender`	return	✓		powrót do nadawcy
`rejected_by_recipient`	return	✓		odbiorca odmówił
`cancelled`	return	✓		zamówienie anulowane

4.2 Fazy

`pre_shipment` · `preparing` · `pickup` · `inbound` · `in_custody` · `transit` · `out_for_delivery` · `delivered` · `exception` · `return` . (`preparing` i `in_custody` są zarezerwowane dla profili niepaczkowych — §8.)

4.3 Typy źródeł i typy czasu

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

4.4 Przyczyny incydentów

Gdy zdarzenie znajduje się w fazie `exception` , SHOULD przenosić `incident_reason` z tego znormalizowanego słownika:

- 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. Maszyna stanów i źródła zwracające tylko status

Fazy postępują naprzód; wyjątki przerywają je i rozwiązują się powrotnie. Stany terminalne (`delivered` , `return_to_sender` , `rejected_by_recipient` , `cancelled`) zamykają obiekt.

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

Reguły:

- Konsumenci MUST porządkować zdarzenia według `occurred_at` i MUST tolerować przybycie poza kolejnością.
- Przejścia do stanu terminalnego są idempotentne; powtórzenia są deduplikowane.
- Po statusie terminalnym producent MUST NOT emitować dalszych zdarzeń poza udokumentowanym przepływem RMA / ponownego otwarcia.

- Źródło udostępniające tylko bieżący status MUST syntetyzować jedno zdarzenie na każdą zaobserwowaną zmianę (ze stabilnym kluczem deduplikacji), zamiast pomijać historię. Z czasem migawki kumulują się w oś czasu.

6. Mapowania na standardy zewnętrzne

Cztery wymiary OTEP odpowiadają pole w pole głównym standardom, więc każdy z nich jest rzutem wyjściowym zdarzenia 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

Wartości dla poszczególnych kodów (EPCIS CBV `bizStep` / `disposition`, ONE Record `eventCode`, kod UN/CEFACT) są publikowane w odczytywalnym maszynowo zbiorze kodów. Poza tymi standardami międzynarodowymi oś czasu OTEP może być również rzutowana na ślady OpenTelemetry, obserwacje OGC SensorThings oraz popularne platformy handlowe (AfterShip, Shopify, Amazon, Walmart, BigCommerce, Magento, WooCommerce, Etsy).

> Pewność: wartości EPCIS CBV to stabilne, standardowe URN. Wartości kodów ONE Record i UN/CEFACT stanowią najlepsze dopasowanie dla ostatniej mili i SHOULD być zweryfikowane wobec oficjalnych list kodów przed użyciem zewnętrznym. „Dostarczono do odbiorcy” nie ma dokładnego CBV bizStep — używane jest najbliższe dopasowanie (`receiving` + `received`) lub URN rozszerzenia słownika użytkownika.

7. API OTEP

OTEP jest konsumowany przez niewielką, tylko do odczytu powierzchnię HTTP; wszystkie punkty końcowe są publiczne.

Verb	Ścieżka	Zwraca
GET	`/api/v1/otep/trackings/{tracking_number}`	oś czasu dla numeru śledzenia
GET	`/api/v1/otep/trackings/{tracking_number}/events`	tylko zdarzenia
POST	`/api/v1/otep/trackings/batch`	wiele numerów śledzenia w jednym wywołaniu
GET	`/api/v1/otep/trackings/{tracking_number}?view=public`	publiczny bezpieczny widok: bez nazwisk kierowców i wewnętrznych identyfikatorów, ograniczony do firmy domeny, także numery przeprowadzek i magazynowania
GET	`/api/v1/client/otep/trackings/{identifier}`	(token API) oś czasu i szczegóły własnej przesyłki: paczka, przeprowadzka, magazynowanie
GET	`/api/v1/client/otep/locker-storage-rentals/{id}`	(token API) oś czasu przechowania w paczkomacie, według id rekordu
GET	`/oteptrack`	publiczna strona śledzenia: paczka, przeprowadzka, magazynowanie; szczegóły po weryfikacji
POST	`/api/v1/otep/validate`	kontrola zgodności dla przesłanej osi czasu (§9)

Dostępne jest również zapytanie GraphQL udostępniające tę samą oś czasu.

Negocjacja treści

Ta sama oś czasu jest serializowana do dowolnej żądanej reprezentacji, za pomocą parametru zapytania `?format=` lub profilu `Accept`:

Żądanie	Reprezentacja
`?format=otep` (domyślnie)	natywna oś czasu OTEP
`?format=epcis`	GS1 EPCIS 2.0 (JSON-LD `ObjectEvent`s)
`?format=onerecord`	IATA ONE Record (`LogisticsEvent`s)
`?format=uncefact`	status transportu UN/CEFACT
`?format=otlp`	ślady OpenTelemetry
`?format=sensorthings`	obserwacje OGC SensorThings
`?format=aftership` | `shopify` | `amazon` | `walmart` | `bigcommerce` | `magento` | `woocommerce` | `etsy`	postać śledzenia/realizacji danej platformy

Zdarzenia, którym nie można przypisać kodu w danym rzucie, są pomijane i zliczane (nigdy po cichu odrzucane).

8. Profile domenowe

OTEP jest protokołem ogólnym, a nie paczkowym. Warstwa protokołu (koperta zdarzenia, kręgosłup faz, maszyna stanów) jest uniwersalna; konkretne kody statusów należą do profilu zadeklarowanego na osi czasu poprzez `profile`. Kody z §4 to profil **`parcel`**. Inne domeny — przeprowadzki, dostawa jedzenia, magazynowanie i nie tylko — dodają własne zestawy kodów w ramach swojego profilu, w przestrzeni nazw `otep:<profile>:<code>`, z których każdy mapuje się na ten sam kręgosłup faz. Dodanie profilu jest rozszerzeniem, a nie zmianą protokołu.

9. Specyfikacja zgodności (normatywna)

Producent lub konsument jest zgodny z OTEP, gdy każde zdarzenie, które emituje lub akceptuje, spełnia te tabele i reguły.

9.1 Koperta osi czasu

Pole	Typ	Wym.	Ograniczenia
`otep_version`	string	MUST	semver, np. `0.1`
`profile`	string	MUST	zarejestrowany profil
`subject`	object	MUST	§9.2
`current_status`	string|null	SHOULD	kod statusu (§4)
`current_phase`	string|null	SHOULD	MUST być równe fazie `current_status`, jeśli oba obecne
`delivered`	boolean	SHOULD	`true` wtedy i tylko wtedy, gdy `current_status` = `delivered`
`events`	array	MUST	obiekty zdarzeń (§9.3), porządkowalne według `occurred_at`

9.2 `subject`

Co najmniej JEDNO z `tracking_number` / `order_id` / `package_id` MUST być obecne.

Pole	Typ	Ograniczenia
`tracking_number`	string	niepuste
`order_id` / `package_id`	integer|null	
`external_tracking_number`	string|null	
`gs1_sscc`	string|null	18 cyfr
`piece_id`	string|null	

9.3 Obiekt zdarzenia

#	Pole	Typ	Wym.	Ograniczenia

1	`occurred_at`	string	MUST	ISO-8601 z przesunięciem
2	`recorded_at`	string|null	SHOULD	ISO-8601
3	`time_type`	string	MAY (domyślnie `actual`)	`actual` | `estimated` | `scheduled`
4	`status_code`	string|null	MUST[1]	kod z §4.1
5	`phase`	string|null	SHOULD	MUST być równe fazie `status_code`
6	`incident_reason`	string|null	SHOULD[2]	przyczyna z §4.4
7	`description`	string|null	MAY	czytelne dla człowieka
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] Zakodowane zdarzenie MUST przenosić kod statusu z §4.1. Surowy skan, którego nie potrafisz jeszcze sklasyfikować, MAY ustawić `status_code = null`, ale MUST zachować natywny kod w `source.external_event_code` i MUST być zliczony, nigdy odrzucony. [2] Zdarzenie w fazie `exception` SHOULD przenosić `incident_reason`. [3] Zdarzenia ze `status_code` ∈ {`delivered`, `picked_up`} SHOULD przenosić `pod`.

9.4 `location`

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

9.5 `source`

Pole	Typ	Wym.	Ograniczenia
`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]	Twój surowy kod statusu
`raw`	object|null	MAY	oryginalny ładunek

[4] MUST być obecne, gdy `status_code` jest null, tak aby natywny kod nigdy nie został utracony.

9.6 Reguły

1. Czas. `occurred_at` MUST parsować się jako ISO-8601. Znormalizuj numeryczne epoki i `.NET /Date(ms)/` przy przyjmowaniu; nie emituj tych form.
2. Porządkowanie / deduplikacja. Konsumenci MUST sortować według `occurred_at`, tolerować przybycie poza kolejnością i deduplikować na (`subject`, `status_code`, `occurred_at`).
3. Maszyna stanów. Po statusie terminalnym nie emituj dalszych zdarzeń poza udokumentowanym przepływem RMA / ponownego otwarcia.
4. Brak cichej utraty. Zdarzenia, których nie można zakodować, MUST być zliczane, nigdy odrzucane.
5. Wyprowadzanie. `phase`, `current_status`, `current_phase`, `delivered` są rzutami — jeśli są obecne, MUST być spójne z osią czasu zdarzeń.

9.7 Poziomy zgodności i walidacja

- Level 1 — emituje kopertę (§9.1), zdarzenia z wymaganymi polami (§9.3), prawidłowe kody statusów (§4.1), prawidłowe fazy (§4.2) oraz przestrzega maszyny stanów (§5).

- Level 2 — dodatkowo emituje co najmniej jeden rzut na standard zewnętrzny (§6) oraz, gdzie to zasadne, kody specyficzne dla profilu (§8).

Zweryfikuj swoje wyjście, wysyłając POST z osią czasu na `POST /api/v1/otep/validate`. Każdy `errors` traktuj jako blokujący; zaadresuj `warnings`. Odczytywalny maszynowo JSON Schema oraz kompletny zbiór kodów (każdy kod statusu, faza i mapowanie zewnętrzne) są publikowane na potrzeby walidacji offline.

10. Otwartość i zarządzanie

OTEP jest otwartą specyfikacją, dostępną za darmo do implementacji przez dowolną stronę.

- Normatywne vs informacyjne. Normatywne: koperta zdarzenia, kręgosłup faz, słownik statusów, maszyna stanów oraz mapowania pól na standardy zewnętrzne. To, w jaki sposób implementujący wiąże OTEP ze swoimi wewnętrznymi systemami, jest jego własną sprawą i pozostaje poza zakresem tego dokumentu.
- Stabilne identyfikatory. Kody są adresowane jako `otep:<profile>:<code>`, fazy jako `otep:phase:<name>`. Po opublikowaniu w wydanej wersji znaczenie identyfikatora jest niezmienne.
- Wersjonowanie. Wersjonowanie semantyczne. Dodawanie kodów/profili jest zmianą MINOR (kompatybilną wstecz); zmiana znaczenia istniejącego kodu jest zmianą MAJOR i SHOULD być unikana. Wersja protokołu podróżuje z każdą osią czasu (`otep_version`).
- Rozszerzanie. Nowe profile i kody są proponowane wobec tej specyfikacji, a nie odgałęziane (forkowane), tak aby niezależni implementujący się zbiegali. Eksperymentalne kody MAY używać prefiksu `x-` (`otep:parcel:x-my_code`) do czasu rejestracji.
- Licencja. Specyfikacja ma zostać wydana na otwartej licencji — do ustalenia, oczekuje na zatwierdzenie.

11. Interoperacyjność dostawców — przynieś własny standard

OTEP zaprasza innych dostawców do wniesienia własnego standardu zdarzeń śledzenia, tak aby OTEP mógł współdziałać z nim w obu kierunkach:

- Rzutuj OTEP → swój standard. Zdefiniuj mapowanie z osi czasu OTEP na swój format, ponownie wykorzystując słownik statusów OTEP. To czysta transformacja — oś czasu na wejściu, Twoja struktura na wyjściu — więc mapowanie pisze się raz i każdy producent OTEP może emitować Twój format.
- Mapuj swój standard → OTEP. Dostarcz powiązanie (crosswalk) z Twojego słownika statusów na kody OTEP (§4) oraz, w razie potrzeby, profil (§8). Twoje surowe kody są zachowywane w `source.external_event_code`; niezmapowane kody są zliczane, nigdy odrzucane.

Nawet jeśli nie możesz przyjąć OTEP bezpośrednio, możesz dodać pojedynczy znormalizowany `otep_status` do własnych odpowiedzi API oraz udostępnić swoje kody zdarzeń śledzenia do mapowania crosswalk. Proponuj mapowania wobec tej specyfikacji (zamiast forkowania), aby implementujący się zbiegali; nowe formaty podłączają się do tej samej negocjacji treści `?format=` i nigdy nie psują istniejącego konsumenta.