DSCE

DSCE PACT v3 API Technical Specification (Version 1.0.0)

Release, 30 September 2026

This version:	https://api.dsce.com.tw/docs/guide/pact-v3-api-spec.html
PDF version:	https://api.dsce.com.tw/docs/guide/pact-v3-api-spec.pdf
Base specification:	Technical Specifications for PCF Data Exchange (Version 3.0.3)
Related documents:	PACT v3 API 使用指南（Traditional Chinese） · DSCE API 使用指南
Feedback:	Contact the DSCE platform administrator.
Editor:	DSCE Platform Team, Digital Transformation Research Institute (DTRI), Institute for Information Industry (III)

© 2026 III, Digital Transformation Research Institute (DTRI). All Rights Reserved.

Abstract

This document specifies the PACT Technical Specifications v3.0.3 implementation provided by the DSCE host system (數位減碳雲資料交換平台, the PCF data exchange platform operated by III DTRI). It covers authentication, the ProductFootprint data model as accepted and returned by DSCE, support for Data Model Extensions including the *Product Footprint Extension for CBAM*, the HTTP REST API actions, event-based communication, and platform-specific P2P actions.

> **IMPORTANT:** Access to the PACT v3 API (all `/api/3/*` endpoints) is **disabled by default** and MUST be enabled for each account by a DSCE platform **administrator**. Holding valid credentials or a valid access token is not sufficient. See § 5.2 API v3 Enablement.

Status of this Document

This is version 1.0.0 of the DSCE PACT v3 API Technical Specification. Where this document is silent, the behavior defined in [PACT-V3] applies. Where DSCE deviates from or extends [PACT-V3], the deviation is stated explicitly.

Table of Contents

1 Introduction
- 1.1 Scope
- 1.2 Relationship to PACT v3.0.3
- 1.3 Access Control

2 Terminology

3 Conformance
4 Data Model
4.1 Data Types
4.2 ProductFootprint
4.3 DataModelExtension
4.4 CBAM Extension
4.5 Error
5 HTTP REST API
5.1 Host System
5.2 API v3 Enablement
5.3 Authentication
5.4 Error Handling
5.5 Action ListFootprints
5.6 Action GetFootprint
5.7 Action CreateFootprint
5.8 Action DeprecateFootprint
5.9 Action Events
5.10 P2P Actions
6 Data Ownership and Access Scope
7 Examples
Appendix A: Changelog
References

1. Introduction

1.1. Scope

DSCE is a multi-tenant host system for the exchange of Product Carbon Footprint (PCF) data. This specification describes the native PACT v3 interface of DSCE, intended for system integrators of approved partner organizations. General-purpose, permission-controlled integration (including acting on behalf of customers) is provided by the separate DSCE API and is out of scope of this document.

1.2. Relationship to PACT v3.0.3

DSCE implements the data model (§ 4) and the actions ListFootprints, GetFootprint and Events (§ 5) of [PACT-V3]. In addition DSCE provides:

- Actions for data owners to **create** and **deprecate** footprints (§ 5.7, § 5.8).
- Validation of Data Model Extensions, including the CBAM extension (§ 4.3, § 4.4).
- Platform-specific P2P actions (§ 5.10).

1.3. Access Control

> **IMPORTANT:** The PACT v3 API is available only to accounts for which a DSCE **administrator** has enabled the *API v3* permission (`isApiV3Enabled = 1`). Users and solution providers cannot enable this permission themselves. Requests from accounts without this permission are rejected with HTTP `403`.

2. Terminology

Host system
: The DSCE platform, which serves multiple data owners and data recipients.

Data owner
: The company that created a ProductFootprint. On the PACT v3 API the data owner is always the authenticated caller.

Data recipient
: A company that has been granted access to a ProductFootprint by its data owner.

Administrator
: A DSCE platform operator who manages accounts and permissions, including enabling the PACT v3 API.

Registered company
: A company with an account on DSCE. Each registered company has a unique `companyId` URN, for example `urn:company:III-DTRI:00000257`.

Internal caller

An account designated by DSCE for strict extension validation (see § 4.4.4). All other accounts are external callers.

3. Conformance

The key words “MUST”, “MUST NOT”, “SHOULD”, “SHOULD NOT” and “MAY” in this document are to be interpreted as described in [RFC2119]. Requirements on the host system describe the behavior of DSCE; requirements on the client describe what callers must do for their requests to be accepted.

The DSCE PACT v3 implementation passed the WBCSD PACT v3 conformance testing (notified 29 October 2025).

4. Data Model

The data model is that of [PACT-V3] § 4. This section lists DSCE-specific rules only.

4.1. Data Types

Type	DSCE rule
`string<decimal>`	MUST be a JSON string containing a decimal number, e.g. `"2.0"`. JSON numbers and scientific notation (e.g. `"1.5e0"`) are rejected. Values are stored and returned exactly as submitted, without rounding.
`string<uuid>`	Identifiers generated by DSCE are RFC 4122 version 4 UUIDs in lowercase, e.g. `"bc30e6e5-d7c3-4ad8-91f6-5d8be04f9627"`. Identifiers created before 30 September 2026 may be uppercase; comparisons are case-insensitive.
`string<urn>`	See [RFC8141]. DSCE requires the lowercase prefix `urn:`.
`string<date-time>`	ISO 8601 in UTC, e.g. `"2026-01-01T00:00:00Z"`; fractional seconds are accepted.

4.2. ProductFootprint

ProductFootprint objects follow [PACT-V3] § 4.6 with the following host-system rules:

Property	DSCE rule
`id`	Assigned by DSCE on creation (see § 5.7). A value supplied by the client is ignored; clients MUST use the `id` returned in the response.
`specVersion`	MUST be `3.x` (e.g. `"3.0.0"`).
`companyName`	If omitted, DSCE fills in the registered name of the caller's company.
`companyIds`	If omitted, DSCE sets it to the caller's `companyId`. If supplied, it MUST be a JSON array; DSCE ensures the caller's `companyId` is its first element. Additional identifiers of the caller (e.g. DUNS or facility URNs) are kept. It MUST NOT contain the `companyId` of another registered company, otherwise the request is rejected with `400`.
`status`	`Active` on creation; `Deprecated` after DeprecateFootprint.
`version`	Incremented by DSCE when a footprint is deprecated.
`pcf.*`	Numeric properties are `string<decimal>` (see § 4.1) and are stored without rounding.
`extensions`	Array of DataModelExtension, validated as described in § 4.3 and § 4.4.

4.3. DataModelExtension

As defined in [PACT-V3] § 4.6.1 and [DATA-MODEL-EXTENSIONS] § 6, the value of `extensions` is an **array**; each member is a DataModelExtension **object**. DSCE validates every member against the following

rules, for all callers. A violation rejects the whole request with `400` before any data is stored.

Name	Description
`specVersion`	**required** string. Semantic Versioning `major.minor.patch`.
`dataSchema`	**required** string<uri>. Identifies the extension; DSCE selects extension-specific validation by this value, regardless of `specVersion`.
`documentation`	optional string<uri>.
`data`	**required** JSON object.

Extensions whose `dataSchema` is not recognized by DSCE are validated against the rules above only and are stored and returned unchanged.

4.4. CBAM Extension

DSCE supports the [CBAM-EXTENSION] (specVersion `0.9.0`), identified by:

```
https://cbam.corporate-harmony.org/schemas/pact-extension.json
```

4.4.1. CbamExtension (`data`)

Name	Description
`directEmissions`	**required** DirectEmissions. Direct emissions of the reported product or its precursors.
`indirectEmissions`	optional IndirectEmissions. Indirect emissions of the reported product or its precursors.

4.4.2. DirectEmissions

Name	Description
`determinationType`	**required** string. `01` actual data; `02` estimated values including Commission default values; `03` actual data not available.
`determinationTypeElectricity`	optional string. `01` actual data; `02` default values.
`applicableReportingTypeMethodology`	**required** string. `TOM01` Commission rules; `TOM02` other methods; `TOM03` Commission default values.
`applicableReportingMethodology`	optional string, max. 4000 characters.
`specificEmbeddedEmissions`	**required** string<decimal>. GHG emissions intensity (t-CO_2e/t).
`otherSourceIndication`	optional string, max. 4000 characters.
`emissionFactorSourceElectricity`	optional string. `01` or `02`.
`emissionFactor`	optional string<decimal> (t-CO_2e/MWh).
`electricityImported`	optional string<decimal>.

Name	Description
`totalEmbeddedElectricityImported`	optional string<decimal>.
`measurementUnit`	**required** string. `EMU1` (tonnes).
`emissionFactorSourceValue`	optional string, max. 5000 characters.
`conditionalityFulfillment`	optional string, max. 5000 characters.

> **NOTE:** [CBAM-EXTENSION] marks `determinationType` and `specificEmbeddedEmissions` as optional for electricity goods. DSCE currently requires them for all goods.

4.4.3. IndirectEmissions

Name	Description
`determinationType`	**required** string. `01`, `02` or `03`.
`emissionFactorSource`	optional string. `01` other; `02` Commission based on IEA data.
`emissionFactor`	optional string<decimal> (t-CO_2e/MWh).
`specificEmbeddedEmissions`	**required** string<decimal> (t-CO_2e/t).
`measurementUnit`	**required** string. `EMU1`.
`electricityConsumed`	optional string<decimal> (MWh/t).
`electricitySource`	**required** string. `SOE01` direct technical link; `SOE02` (bilateral) power purchase agreement; `SOE03` grid.
`otherSourceIndication`	optional string, max. 4000 characters.
`emissionFactorSourceValue`	optional string, max. 512 characters.

4.4.4. Consistency and Enforcement

The sum of the direct and indirect `specificEmbeddedEmissions` MUST be consistent with the ProductFootprint:

```
direct.specificEmbeddedEmissions + indirect.specificEmbeddedEmissions
  = pcf.pcfExcludingBiogenicUptake ÷ pcf.productMassPerDeclaredUnit
```

A violation of any rule in this section is handled as follows:

Caller	Behavior
Internal caller	Rejected with `400`; nothing is stored.
External caller	Accepted; the violation is logged by DSCE.

Clients SHOULD always submit data conforming to this section. [CBAM-EXTENSION] further requires `companyIds` to represent the precursor manufacturing facilities and `productIds` to identify the precursor for

the delivery destination.

4.5. Error

As [PACT-V3] § 4.13. All error responses of `/api/*` are JSON Error objects, never HTML pages.

Name	Description
`code`	**required** string. One of `BadRequest`, `AccessDenied`, `TokenExpired`, `NotFound`, `InternalError`, `NotImplemented`; DSCE additionally uses `PaymentRequired` (see § 5.4).
`message`	**required** string. Human-readable error message.

5. HTTP REST API

5.1. Host System

The base URL of the production host system is:

```
https://api.dsce.com.tw/api
```

All PACT v3 actions are served under `<base-url>/3/`. Authentication actions are served under `<base-url>/auth/` and are shared with the DSCE API. All requests MUST use HTTPS.

Method	Path	Action
`POST`	`/auth/token`	Obtain access token
`GET`	`/3/footprints`	ListFootprints
`GET`	`/3/footprints/{id}`	GetFootprint
`POST`	`/3/footprints`	CreateFootprint (DSCE)
`PATCH`	`/3/footprints/{id}`	DeprecateFootprint (DSCE)
`POST`	`/3/events`	Events
`GET`	`/3/events`, `/3/events/{id}`	List / get events (DSCE)
`GET/POST`	`/3/p2p/*`	P2P actions (DSCE)

5.2. API v3 Enablement

> **IMPORTANT:** The API v3 permission can **only be enabled by a DSCE administrator**. It is disabled for every account by default.

Before calling any `/api/3/*` action, all of the following MUST hold:

1. The client holds a valid access token (§ 5.3).
2. The account has been enabled for API v3 by an administrator.
3. The account itself is enabled.

Enablement procedure:

1. The organization requests API v3 access from the DSCE platform (directly or through its solution provider), stating the integration purpose and the counterparties.
2. An administrator opens the member in the DSCE back office (*Member Management*) and switches *API v3* to *Enabled*.
3. The permission takes effect immediately; existing access tokens remain valid.
4. An administrator MAY disable the permission at any time; subsequent requests are rejected immediately.

EXAMPLE 1

Response for an account without API v3 permission:

```
HTTP/1.1 403 Forbidden
Content-Type: application/json

{
  "code": "AccessDenied",
  "message": "PACT v3 API access is not enabled for this account."
}
```

5.3. Authentication

DSCE uses the OAuth 2.0 client credentials grant [RFC6749] and Bearer tokens [RFC6750], as specified in [PACT-V3] § 5.5.

5.3.1. Client Credentials

A `client_id` and `client_secret` are issued through **`POST`** `/auth/credentials` or the DSCE web portal. The `client_secret` is shown only once. A lost secret can be reset with **`PATCH`** `/auth/credentials/{clientId}` (Bearer token required); the previous secret becomes invalid immediately. Issuing credentials does not grant API v3 access (§ 5.2).

5.3.2. Obtaining an Access Token

```
POST /api/auth/token HTTP/1.1
Host: api.dsce.com.tw
Authorization: Basic <base64(client_id:client_secret)>
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials
```

EXAMPLE 2

Successful response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "access_token": "eyJhbGciOiJSUzI1NiIs...",
  "expires_in": 1800,
  "token_type": "Bearer",
  "scope": "profile email"
}
```

Access tokens expire after **1800 seconds**. Clients MUST obtain a new token after expiry.

5.3.3. Using the Access Token

```
GET /api/3/footprints HTTP/1.1
Host: api.dsce.com.tw
Authorization: Bearer <access_token>
```

The authentication scheme name is case-insensitive (`Bearer` or `bearer`). **`POST`** `/auth/tokenVerify` returns the validity and the `client_id` of a token.

5.4. Error Handling

HTTP status	code	Situation
`400`	BadRequest	Missing or malformed parameters; extension or CBAM validation failure; `companyIds` containing another registered company; deprecating an already deprecated footprint.
`401`	TokenExpired	Token missing (*Missing or malformed Authorization header.*), invalid (*The access token is invalid.*) or expired (*The specified access token has expired.*). Per [PACT-V3] § 5.5.3, invalid and expired tokens both return 401.
`402`	PaymentRequired	The account is disabled (DSCE-specific).
`403`	AccessDenied	**API v3 not enabled by an administrator** (§ 5.2).
`404`	NotFound	Resource does not exist, or is outside the caller's access scope (existence is not disclosed); unknown route.
`405`	BadRequest	HTTP method not supported for the path.
`500`	InternalError	Unexpected server error; authorization server unreachable.

5.5. Action ListFootprints

```
GET /api/3/footprints?limit=10&offset=0 HTTP/1.1
```

Returns the footprints within the caller's access scope (§ 6), newest first, as `{ "data": [ ProductFootprint, ... ] }`.

Parameter	Description
`limit`	optional Page size, default 10. Non-positive or non-numeric values fall back to the default.
`offset`	optional Zero-based offset, default 0. Negative values are treated as 0.
`productId`	optional URN; repeatable.
`companyId`	optional URN; repeatable.
`geography`	optional Region or subregion.

Parameter	Description
`classification`	optional Product classification URN.
`validOn`, `validAfter`, `validBefore`	optional string<date-time>; filters on the validity period.
`status`	optional `Active` or `Deprecated`.

Pagination links are returned in the `Link` response header with relations `first`, `prev`, `next` and `last`.

5.6. Action GetFootprint

```
GET /api/3/footprints/{id} HTTP/1.1
```

Returns `{ "data": ProductFootprint }`. Deprecated footprints are returned as well. A footprint that does not exist or is outside the caller's access scope returns `404 NotFound`.

> **NOTE:** [PACT-V3] § 5.7 specifies `403` when the recipient is not allowed to access an existing footprint. DSCE returns `404` so as not to disclose the existence of footprints of other companies.

5.7. Action CreateFootprint

```
POST /api/3/footprints HTTP/1.1
Content-Type: application/json
```

Creates a ProductFootprint owned by the caller. Processing order:

1. Assign a new `id` and apply the `companyName` / `companyIds` rules (§ 4.2).
2. Validate the ProductFootprint against [PACT-V3] § 4.
3. Validate `extensions` (§ 4.3), then extension-specific rules (§ 4.4).
4. Store the footprint and its extensions in a single transaction.

Any failure in steps 1–3 returns `400` and stores nothing. On success the response is `200` with the assigned identifier:

```
{ "id": "bc30e6e5-d7c3-4ad8-91f6-5d8be04f9627" }
```

5.8. Action DeprecateFootprint

```
PATCH /api/3/footprints/{id} HTTP/1.1
```

Sets `status` to `Deprecated` and increments `version`, see [PACT-V3] § 7. Only the data owner can deprecate a footprint. Deprecating an already deprecated footprint returns `400`; an unknown footprint returns `404`. Extensions are retained.

5.9. Action Events

```
POST /api/3/events HTTP/1.1
Content-Type: application/cloudevents+json
```

Receives PACT events encoded as CloudEvents [CLOUDEVENTS], as specified in [PACT-V3] § 5.8. The host system validates the event, returns `200`, and processes it asynchronously.

type	Purpose	Receiver(s) determined by DSCE
`org.wbcsd.pact.ProductFootprint.RequestCreatedEvent.3`	Request footprints	Registered companies listed in `data.companyId`, and owners of footprints matching `data.productId`
`org.wbcsd.pact.ProductFootprint.PublishedEvent.3`	Announce new or updated footprints	Recipients authorized for the footprints in `data.pfIds` that are owned by the sender
`org.wbcsd.pact.ProductFootprint.RequestFulfilledEvent.3`	Fulfil a request	Sender of the original request (`data.requestEventId`)
`org.wbcsd.pact.ProductFootprint.RequestRejectedEvent.3`	Reject a request	Sender of the original request (`data.requestEventId`)

For a PublishedEvent, every element of `data.pfIds` MUST be a UUID; otherwise the event is rejected with `400`.

5.9.1. Responses to RequestCreatedEvent

DSCE answers a RequestCreatedEvent asynchronously with a RequestFulfilledEvent (footprints the requester is authorized to access) or a RequestRejectedEvent, delivered to `<source>/3/events` after authenticating at `<source>/auth/token`.

> **IMPORTANT:** DSCE delivers responses **only to hosts registered in the platform allowlist**. The recipient host MUST be registered with a DSCE administrator in advance. Events from unregistered hosts are still accepted, but no response is delivered.

5.9.2. Listing and Retrieving Events

```
GET /api/3/events HTTP/1.1
GET /api/3/events/{id} HTTP/1.1
```

Returns the events in which the caller's company is the sender or a receiver, newest first. Each event contains only `type`, `specversion`, `id`, `source`, `time` and `data`. An event unrelated to the caller returns `404`. These actions are DSCE extensions to [PACT-V3].

5.10. P2P Actions

Platform-specific actions for supplier-to-manufacturer PCF exchange. Suppliers are identified by their Taiwan business tax ID (`taxId`).

Method	Path	Description
GET	`/3/p2p/pcfRequest/{taxId}`	List the products of a supplier.
GET	`/3/p2p/productMapping/{pfpId}`	Get the supplier that owns a footprint.
POST	`/3/p2p/pcfDeposit`	Record a PCF hash deposit and the manufacturer–supplier relationship. Body: `taxId`, `pfpId`, `pcfHash`, `content`.
GET	`/3/p2p/pcfQuery/{taxId}`	Get the PCF data provided by a supplier.
GET	`/3/p2p/pcfVerify/{pcfHash}`	Verify that a supplier's hash matches the deposited data.

6. Data Ownership and Access Scope

Resource	Rule
Footprints created	Always owned by the caller's company. Acting on behalf of another company is not supported on this API (use the DSCE API).
Footprints readable	Footprints owned by the caller, and footprints whose owner has granted access to the caller.
Events readable	Events sent or received by the caller's company (§ 5.9.2).
Anything else	`404 NotFound`; existence is not disclosed.

7. Examples

EXAMPLE 3

CreateFootprint request with a CBAM extension. `companyName` and `companyIds` are omitted and filled in by DSCE. 1.5 + 0.5 = 2.0 ÷ 1.0, so the CBAM consistency rule holds.

```
POST /api/3/footprints HTTP/1.1
Host: api.dsce.com.tw
Authorization: Bearer <access_token>
Content-Type: application/json

{
  "specVersion": "3.0.0",
  "version": 1,
  "created": "2026-09-30T06:00:00Z",
  "status": "Active",
  "validityPeriodStart": "2026-01-01T00:00:00Z",
  "validityPeriodEnd": "2027-01-01T00:00:00Z",
  "productDescription": "Hot-rolled steel coil",
  "productIds": ["urn:gtin:4712345060507"],
  "productClassifications": ["urn:pact:productclassification:un-cpc:4112"],
  "productNameCompany": "HR Coil A",
  "comment": "",
  "pcf": {
    "declaredUnitOfMeasurement": "kilogram",
    "declaredUnitAmount": "1.0",
    "productMassPerDeclaredUnit": "1.0",
    "pcfExcludingBiogenicUptake": "2.0",
    "pcfIncludingBiogenicUptake": "2.0",
    "fossilGhgEmissions": "2.0",
    "fossilCarbonContent": "0.0",
    "biogenicCarbonContent": "0.0",
    "ipccCharacterizationFactors": ["AR6"],
    "crossSectoralStandards": ["ISO14067"],
    "productOrSectorSpecificRules": [],
    "boundaryProcessesDescription": "Cradle-to-gate",
    "referencePeriodStart": "2025-01-01T00:00:00Z",
    "referencePeriodEnd": "2026-01-01T00:00:00Z",
    "geographyCountry": "TW",
    "exemptedEmissionsPercent": "0",
    "exemptedEmissionsDescription": "",
    "allocationRulesDescription": "Mass allocation",
    "primaryDataShare": "50"
  },
  "extensions": [
    {
      "specVersion": "0.9.0",
      "dataSchema": "https://cbam.corporate-harmony.org/schemas/pact-extension.json",
      "data": {
        "directEmissions": {
          "determinationType": "01",
          "applicableReportingTypeMethodology": "TOM01",
          "specificEmbeddedEmissions": "1.5",
          "measurementUnit": "EMU1"
        },
```

```
        "indirectEmissions": {
          "determinationType": "01",
          "specificEmbeddedEmissions": "0.5",
          "measurementUnit": "EMU1",
          "electricitySource": "SOE01"
        }
      }
    }
  ]
}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{ "id": "bc30e6e5-d7c3-4ad8-91f6-5d8be04f9627" }
```

EXAMPLE 4

CBAM consistency violation by an internal caller (direct emissions changed to `"3.0"`):

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "code": "BadRequest",
  "message": "CBAM extension 驗證失敗（pfpId=…）：pcf_sum_mismatch: direct(3.0) + indirect(0.5) = 3.5 tCO2e/t，與 pcfExcludingBiogenicUptake(2.0) ÷ productMassPerDeclaredUnit(1.0) = 2 不一致，差異 1.5"
}
```

EXAMPLE 5

Invalid envelope: `extensions` sent as an object instead of an array:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{ "code": "BadRequest", "message": "extensions 必須為 JSON array（PACT v3 §4.8）" }
```

EXAMPLE 6

RequestCreatedEvent:

```
POST /api/3/events HTTP/1.1
Host: api.dsce.com.tw
Authorization: Bearer <access_token>
Content-Type: application/cloudevents+json

{
  "type": "org.wbcsd.pact.ProductFootprint.RequestCreatedEvent.3",
  "specversion": "1.0",
  "id": "71f32fda-5ab6-4855-a88f-c085f31a523b",
  "source": "https://api.recipient.example/api",
  "time": "2026-09-30T06:31:44Z",
  "data": {
    "productId": ["urn:gtin:4712345060507"],
    "companyId": ["urn:company:III-DTRI:00000257"]
  }
}
```

NOTE: Validation messages are currently returned in Traditional Chinese. Clients SHOULD rely on the HTTP status and `code`, not on the text of `message`.

Appendix A: Changelog

Version	Date	Changes
1.0.0	2026-09-30	Initial English specification. Covers: administrator-controlled API v3 enablement; CBAM extension validation; `companyIds` ownership rule; RFC 4122 v4 identifiers; decimal values stored without rounding; event access scope (sender and receiver); response-delivery allowlist; distinguished 401 messages; JSON errors for all `/api/*` routes.

References

Normative References

[PACT-V3]

WBCSD. Technical Specifications for PCF Data Exchange (Version 3.0.3) <https://wbcsd.github.io/tr/data-exchange-protocol/latest/>.

[DATA-MODEL-EXTENSIONS]

WBCSD. Technical Specification for Data Model Extensions <https://wbcsd.github.io/data-model-extensions/spec/>.

[CBAM-EXTENSION]

T. Okada. Product Footprint Extension for CBAM <https://cbam.corporate-harmony.org/schemas/pact-extension.html>. Living Document, 27 February 2026.

[CLOUDEVENTS]

CloudEvents Specification v1.0 <https://github.com/cloudevents/spec>.

[RFC2119]

S. Bradner. Key words for use in RFCs to Indicate Requirement Levels <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC4122]

P. Leach et al. A Universally Unique IDentifier (UUID) URN Namespace <https://www.rfc-editor.org/rfc/rfc4122>.

[RFC6749]

D. Hardt. The OAuth 2.0 Authorization Framework <https://www.rfc-editor.org/rfc/rfc6749>.

[RFC6750]

M. Jones; D. Hardt. The OAuth 2.0 Authorization Framework: Bearer Token Usage <https://www.rfc-editor.org/rfc/rfc6750>.

[RFC8141]

P. Saint-Andre; J. Klensin. Uniform Resource Names (URNs) <https://www.rfc-editor.org/rfc/rfc8141>.