

# UC2 — Regulatory Data Exchange

Disclaimer: This is a working version and subject to change. For the latest information, please refer to the India Energy Stack GitHub.

Attested, Reusable Regulatory Filings (Create Once, Reuse Many) India Energy Stack (IES)

---

## Overview

Field	Value
Use Case ID	UC2
Use Case Name	Regulatory Data Exchange
Category	Regulatory & Compliance
Outcome Theme	Trustworthy, reusable, machine-verifiable filings
IES Role	Standards, specs, reference profiles, conformance (not an operational platform)

---

## Problem

Regulators require operational, commercial, and compliance datasets that are comparable across utilities, defensible in proceedings, and available in time to drive action. Today, reporting is largely document-first and re-created repeatedly:

- Redundant re-creation of the same data

Similar datasets are assembled multiple times for different requesters (SERC/CERC, auditors, ministries, programs), increasing cost and inconsistency.

- Weak non-repudiation and version ambiguity

“Which file was final?” leads to disputes, delays, reconciliation work, and reduced trust.

- Definition drift without machine-verifiable context

Same metric name can imply different formulas, time windows, exclusions, or adjustments, without an attached, executable definition.

- Ad-hoc handling of confidentiality and public disclosure

Some data must be private, some shareable under conditions, and some publishable—yet current mechanisms treat all outputs as similar “reports.”

---

## Regulatory Data Exchange in One Sentence

Regulatory Data Exchange standardizes regulatory exchange by defining attested, versioned Filing Objects (data + metadata + proofs) that can be validated, signed, receipted, discovered, and reused across many requesters—without re-creating the dataset each time.

---

## Core Concept: Filing Object as the Unit of Truth

A Filing Object is a machine-verifiable package that represents a specific regulatory submission for a specific period.

It contains (conceptually):

- Filing content (the dataset in a standard format)
- Semantic context (what it represents, period, units)
- Validation context (which rulebook and version was applied)
- Issuer attestation (authorized signatory proof)
- Integrity proof (hashes over content)
- Receipt/acceptance proof (regulator acknowledgment, timestamped)

Single Source of Truth (SoT) in UC2 is defined as: > The accepted Filing Object (plus its receipt and version chain), not a centralized database.

Corrections happen via a new version that explicitly links to the prior version.

---

## Public vs Private Disclosure (First-Class)

UC2 treats disclosure class as a first-class property of filings:

- Private Filing

Accessible only to regulator roles.

- Restricted Filing

Shareable with explicitly authorized requesters under defined policy.

- Public Disclosure

A Disclosure Object derived from an accepted Filing Object, containing only permitted fields/aggregations, and signed with a verifiable linkage to the source filing.

UC2 does not prescribe who hosts public data; it specifies how disclosures are derived and verified.

---

## Exchange Pattern (asynchronous packet + ACK + callback)

### 1) Standard packet

All messages use a common outer structure:

```
{
  "context": { /* routing + correlation */ },
  "message": { /* JSON-LD object(s) */ }
}
```

The requester sends a packet and receives an immediate ACK/NACK. The result payload is delivered later via the corresponding `on\_\*` callback.

This is the common pattern used in large-scale network protocols for interoperable marketplaces and data exchange: an immediate ACK for transport/format acceptance, then asynchronous business response with the full payload.

### 2) Immediate ACK/NACK envelope

```
{
  "context": { "message_id": "...", "transaction_id": "..." },
  "message": {
    "ack": { "status": "ACK" },
    "error": { "code": "...", "message": "..." }
  }
}
```

### 3) Actions (UC2 mapping)

UC2 uses a small set of actions, each with a paired callback.

Action	Callback	UC2 intent
`search`	`on_search`	Discover requests, filing types, rulebooks, or prior accepted filings (as permitted)
`init`	`on_init`	Submit a draft filing for pre-validation / pre-checks
`confirm`	`on_confirm`	Submit an attested filing for acceptance and receipt

`status`	`on_status`	Check processing / acceptance status; fetch receipt pointer
`update`	`on_update`	Submit a superseding version (correction)
`cancel`	`on_cancel`	Withdraw a draft / invalidate an in-flight submission (policy-dependent)

> Note: UC2 can be implemented over HTTP, message brokers, or other transports. The key is the packet shape, correlation IDs, and callback pairing.

---

## `context` object (routing + correlation)

A minimal context for UC2:

```
{
  "domain": "energy.regulatory",
  "action": "confirm",
  "country": "IND",
  "city": "std:080",
  "timestamp": "2026-02-07T12:10:00Z",
  "transaction_id": "TXN-2026-02-07-0001",
  "message_id": "MSG-2026-02-07-0001",
  "requester_id": "regulator.example.in",
  "requester_uri": "https://regulator.example.in/uc2",
  "provider_id": "discom.example.in",
  "provider_uri": "https://discom.example.in/uc2",
  "ttl": "PT30S"
}
```

Fields like `action`, `transactionid`, and `messageid` enable deterministic correlation across async flows. A `\*` (requester side) and `\*` (provider side) pair enables multi-party exchange and routing.

The structure above mirrors commonly adopted open-network packet context conventions.

---

## JSON-LD base context (IES UC2 vocabulary)

UC2 uses a compact IES vocabulary. Implementations can host this at a stable URL; examples below inline it for clarity.

```
{
  "@context": {
    "ies": "https://ies.gov.in/vocab#",
  }
}
```

```

"schema": "https://schema.org/",
"xsd": "http://www.w3.org/2001/XMLSchema#",

"id": "@id",
"type": "@type",

"issuer": { "@id": "ies:issuer", "@type": "@id" },
"regulator": { "@id": "ies:regulator", "@type": "@id" },
"requester": { "@id": "ies:requester", "@type": "@id" },

"period": "ies:period",
"from": { "@id": "ies:from", "@type": "xsd:date" },
"to": { "@id": "ies:to", "@type": "xsd:date" },
"granularity": "ies:granularity",

"filingType": "ies:filingType",
"rulebook": { "@id": "ies:rulebook", "@type": "@id" },
"rulebookVersion": "ies:rulebookVersion",
"deadline": { "@id": "ies:deadline", "@type": "xsd:dateTime" },

"integrity": "ies:integrity",
"payloadHash": "ies:payloadHash",
"version": "ies:version",
"supersedes": { "@id": "ies:supersedes", "@type": "@id" },

"disclosureClass": "ies:disclosureClass",
"derivedFrom": { "@id": "ies:derivedFrom", "@type": "@id" },

"payload": "ies:payload",
"attachments": "ies:attachments"
}
}

```

JSON-LD's `@context` mechanism is what turns “a JSON blob” into a semantically portable object graph.

---

## Primitives (JSON-LD objects)

Each primitive is a JSON-LD object. These objects appear inside `message` in the standard packet.

### 1) `ies:Request` (RequestObject)

Purpose: regulator defines what dataset is required, for what period, by when, under which rulebook.

#### Schema (JSON-LD shape)

```

{
  "@context": "object|array",
  "@id": "string (URI/URN)",
  "@type": "ies:Request",
  "requester": "string (URI/URN of party)",
  "deadline": "dateTime",
  "rulebook": "string (URI/URN of rulebook)",
}

```

```

"rulebookVersion": "string",
"filingType": "string",
"period": { "from": "date", "to": "date", "granularity": "string" },
"payload": { "profile": "string", "requiredDimensions": ["..."], "requiredMeasures": ["..."] },
"notes": "string?"
}

```

### Example (`search` → `on\_search` returns available requests)

```

{
  "@context": [
    { "ies": "https://ies.gov.in/vocab#", "xsd": "http://www.w3.org/2001/XMLSchema#" },
    {
      "requester": { "@id": "ies:requester", "@type": "@id" },
      "rulebook": { "@id": "ies:rulebook", "@type": "@id" },
      "deadline": { "@id": "ies:deadline", "@type": "xsd:dateTime" }
    }
  ],
  "@id": "urn:ies:uc2:request:REQ-2026-02-GOA-001",
  "@type": "ies:Request",
  "requester": "urn:ies:party:SERC-GOA",
  "deadline": "2026-02-10T18:30:00Z",
  "rulebook": "urn:ies:uc2:rulebook:RB-POWER_PURCHASE-1",
  "rulebookVersion": "1.0.3",
  "filingType": "power_purchase_summary",
  "period": { "from": "2012-04-01", "to": "2013-03-31", "granularity": "annual" },
  "payload": {
    "profile": "IES.UC2.DatasetPayload.ObservationModel.v0.1",
    "requiredDimensions": ["year", "seller", "plant", "fuel", "state"],
    "requiredMeasures": ["energy_purchased_mu"]
  },
  "notes": "If plant coordinates are available include lat/long."
}

```

---

## 2) `ies:Filing` (FilingObject)

Purpose: issuer submits dataset payload (+ optional attachments) against a request.

### Schema (JSON-LD shape)

```

{
  "@context": "object|array",
  "@id": "string (URI/URN)",
  "@type": "ies:Filing",
  "issuer": "string (URI/URN of party)",
  "filingType": "string",
  "period": { "from": "date", "to": "date", "granularity": "string" },
  "request": "string (URI/URN of Request)",
  "integrity": { "payloadHash": "string (sha256:...)" },
  "version": "string",
  "supersedes": "string (URI/URN) | null",
  "payload": "ies:DatasetPayload",
  "attachments": [
    { "name": "string", "mediaType": "string", "hash": "sha256:...", "url": "string?" }
  ]
}

```

```
]
}
```

### Example (`confirm` submits attested filing)

```
{
  "@context": {
    "ies": "https://ies.gov.in/vocab#",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "issuer": { "@id": "ies:issuer", "@type": "@id" },
    "request": { "@id": "ies:request", "@type": "@id" },
    "payloadHash": "ies:payloadHash",
    "from": { "@id": "ies:from", "@type": "xsd:date" },
    "to": { "@id": "ies:to", "@type": "xsd:date" }
  },
  "@id": "urn:ies:uc2:filing:FIL-2026-02-G0A-PP-2012-13-001",
  "@type": "ies:Filing",
  "issuer": "urn:ies:party:DISCOM-G0A",
  "filingType": "power_purchase_summary",
  "period": { "from": "2012-04-01", "to": "2013-03-31", "granularity": "annual" },
  "request": "urn:ies:uc2:request:REQ-2026-02-G0A-001",
  "integrity": { "payloadHash": "sha256:EXAMPLE_PAYLOAD_HASH" },
  "version": "1.0",
  "supersedes": null,
  "payload": {
    "@type": "ies:DatasetPayload",
    "profile": "IES.UC2.DatasetPayload.ObservationModel.v0.1",
    "dataset": { "datasetId": "power_purchase_summary", "title": "Power Purchase by Source/PL" },
    "schema": {
      "dimensions": [
        { "key": "year", "valueType": "date_year" },
        { "key": "seller", "valueType": "string" },
        { "key": "state", "valueType": "geo_state" },
        { "key": "plant", "valueType": "string" },
        { "key": "fuel", "valueType": "enum" }
      ],
      "measures": [
        { "key": "energy_purchased_mu", "valueType": "number", "unit": "MU" }
      ]
    }
  },
  "observations": [
    {
      "dims": { "year": 2012, "seller": "NTPC", "state": "Chhattisgarh", "plant": "KSTPS", "fuel": "Coal" },
      "measures": { "energy_purchased_mu": 1301.08 }
    }
  ],
  "attachments": [
    { "name": "supporting.pdf", "mediaType": "application/pdf", "hash": "sha256:EXAMPLE_ATTACHMENT_HASH" }
  ]
}
```

---

### 3) `ies:ValidationReport`

Purpose: deterministic validation result for a filing under a specific rulebook.

## Schema (JSON-LD shape)

```
{
  "@id": "string",
  "@type": "ies:ValidationReport",
  "rulebook": "string (@id)",
  "rulebookVersion": "string",
  "validatedAt": "dateTime",
  "target": "string (@id of Filing)",
  "result": "pass|fail",
  "checks": [ { "id": "string", "status": "pass|fail", "message": "string?" } ],
  "errors": [ { "code": "string", "message": "string", "path": "string?" } ]
}
```

---

## 4) `ies:Receipt`

Purpose: regulator acknowledgment binding acceptance status to the filing's `payloadHash`.

## Schema (JSON-LD shape)

```
{
  "@id": "string",
  "@type": "ies:Receipt",
  "regulator": "string (@id)",
  "issuedAt": "dateTime",
  "target": "string (@id of Filing)",
  "payloadHash": "string",
  "status": "accepted|rejected|accepted_with_observations",
  "notes": "string?",
  "signatures": [ { "kid": "string", "sig": "string" } ]
}
```

---

## 5) `ies:Disclosure`

Purpose: public/restricted derived dataset linked to an accepted filing.

## Schema (JSON-LD shape)

```
{
  "@id": "string",
  "@type": "ies:Disclosure",
  "derivedFrom": "string (@id of Filing)",
  "receipt": "string (@id of Receipt)",
  "payloadHash": "string",
  "disclosureClass": "public|restricted",
  "issuedAt": "dateTime",
  "payload": "ies:DatasetPayload"
}
```

---

## Packets: end-to-end examples

### A) Discover active requests (`search` → `on\_search`)

Request

```
{
  "context": {
    "domain": "energy.regulatory",
    "action": "search",
    "country": "IND",
    "city": "std:080",
    "timestamp": "2026-02-06T10:00:00Z",
    "transaction_id": "TXN-2026-02-06-0007",
    "message_id": "MSG-2026-02-06-0007",
    "requester_id": "discom.example.in",
    "requester_uri": "https://discom.example.in/uc2",
    "provider_id": "regulator.example.in",
    "provider_uri": "https://regulator.example.in/uc2",
    "ttl": "PT30S"
  },
  "message": {
    "intent": {
      "filingType": "power_purchase_summary",
      "period": { "from": "2012-04-01", "to": "2013-03-31" }
    }
  }
}
```

Immediate ACK

```
{
  "context": { "transaction_id": "TXN-2026-02-06-0007", "message_id": "MSG-2026-02-06-0007" },
  "message": { "ack": { "status": "ACK" } }
}
```

Callback (`on\_search`)

```
{
  "context": {
    "domain": "energy.regulatory",
    "action": "on_search",
    "country": "IND",
    "city": "std:080",
    "timestamp": "2026-02-06T10:00:02Z",
    "transaction_id": "TXN-2026-02-06-0007",
    "message_id": "MSG-2026-02-06-0008",
    "requester_id": "discom.example.in",
    "requester_uri": "https://discom.example.in/uc2",
    "provider_id": "regulator.example.in",
    "provider_uri": "https://regulator.example.in/uc2"
  },
  "message": {
    "requests": [
      {
        "@id": "urn:ies:uc2:request:REQ-2026-02-G0A-001",
        "@type": "ies:Request",

```

```

"@context": { "ies": "https://ies.gov.in/vocab#" },
"requester": "urn:ies:party:SERC-GOA",
"deadline": "2026-02-10T18:30:00Z",
"rulebook": "urn:ies:uc2:rulebook:RB-POWER_PURCHASE-1",
"rulebookVersion": "1.0.3",
"filingType": "power_purchase_summary",
"period": { "from": "2012-04-01", "to": "2013-03-31", "granularity": "annual" }
}
]
}
}
}
---

```

## API surface (transport-level)

The packet pattern above can be transported over HTTP endpoints like these (illustrative):

### Discovery

Method	Path	Purpose
GET	`.well-known/ies/uc2`	endpoints, supported filing types, rulebook catalog pointers

### Transaction endpoints (async)

Method	Path	Action(s)
POST	`.v1/search`	`search`
POST	`.v1/init`	`init`
POST	`.v1/confirm`	`confirm`
POST	`.v1/status`	`status`
POST	`.v1/update`	`update`
POST	`.v1/cancel`	`cancel`

### Callback endpoints (async)

Method	Path	Callback(s)
POST	`.v1/on_search`	`on_search`
POST	`.v1/on_init`	`on_init`
POST	`.v1/on_confirm`	`on_confirm`
POST	`.v1/on_status`	`on_status`
POST	`.v1/on_update`	`on_update`

POST	`/v1/on_cancel`	`on_cancel`
------	-----------------	-------------

This “paired request/callback” design matches how asynchronous, server-to-server transaction networks standardize interoperability.

---

## Decentralization Position

UC2 does not require a centralized ledger. A “single source of truth” can be achieved via:

- issuer signatures,
- content hashes,
- version chaining,
- receipts and timestamp proofs,
- and portable verification rules.

Optional witness/timestamp services may exist, but are not mandated by UC2.

---

## Implementation notes

- IDs (`@id`): use URNs/URLs that are stable and resolvable (where appropriate). JSON-LD treats these as node identifiers, enabling referencing without copying.
- Validation: keep deterministic rule execution; bind results and receipts to `payloadHash`.
- Versioning: corrections are new `ies:Filing` objects with `supersedes` linking to the prior `@id`.

---

## What UC2 Does NOT Do

- Define DISCOM internal extraction pipelines (CIS/MDMS/ERP etc.)
- Mandate centralized storage, ledger, or repository
- Replace regulator portals or MIS systems
- Define regulatory policy intent (only the machine-verifiable representation of requirements and validation)

---