UC3 — Energy Policy as Code

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

Machine-Readable Policy Packs and Deterministic Execution
India Energy Stack (IES)

General Use Case Discussion

Overview (from earlier draft)

Field	Value
Use Case ID	UC3
Use Case Name	Energy Policy as Code
Category	Policy & Governance
Outcome Theme	Uniform implementation, faster change propagation, audit-ready traces
IES Role	Standards/specs + trust anchors + conformance (not a policy portal or execution platform)

Problem

Policies (tariffs, subsidies, rebates, surcharges, eligibility rules, program rules) are issued as documents and then translated into system logic across DISCOMs and vendors. This translation introduces:

1. **Implementation drift**
 Same policy intent yields different outcomes across DISCOMs/vendors due to bespoke interpretations.

2. **Slow propagation of amendments**
 Changes take weeks/months to reach every billing/program platform consistently.

3. **Version ambiguity**
 Disputes arise around “what was in force, when, and for whom,” especially during audits and hearings.

4. **Weak decision explainability**
 It is hard to prove why a consumer or category received a specific charge/benefit, beyond opaque system logic.

UC3 in One Sentence

UC3 standardizes policies as **versioned, machine-readable Policy Packs** that can be **validated, attested, discovered, executed deterministically, and audited** consistently across participants.

Core Concept: Policy Pack as the Unit of Truth

A **Policy Pack** is a verifiable artifact that represents a policy (or computable subset) with:
- **Scope** (jurisdiction, categories, applicability)
- **Definitions** (units, windows, variables)
- **Rules** (eligibility + computations + overrides)
- **Effective dates and versioning**
- **Clause mapping** (references to source instrument)
- **Test vectors** (inputs → expected outputs for conformance)
- **Attestation** (authorized sign-off proof)

Single Source of Truth (SoT) in UC3 is the **attested Policy Pack version**, not a centralized policy portal.

Create Once, Reuse Many (Policy Reusability)

Once a Policy Pack version is published and attested:
- DISCOMs and vendors should **consume the same pack**,
- run the same conformance tests,
- produce consistent outputs,
- and attach policy-version references in decision traces and receipts.

This eliminates repeated “manual translation” of the same tariff/order into multiple systems.

Trust Anchors (IES-specified)

UC3 depends on explicit trust anchors (IES defines the profiles/interfaces; implementations can vary):

1. **Directory / Key Registry**
 Identifiers, public keys, endpoints, supported pack types, supported versions.

2. **Policy Pack Profiles**
 Required fields for pack envelope, versioning, effective dates, clause references, attestation fields.

3. **Policy-as-Code Rule Format + Execution Semantics**
 Deterministic evaluation rules and standard outputs/traces.

4. **Conformance Kit**
 Validator + test vectors to ensure two implementers produce identical outputs for the same pack version.

5. **Audit Receipt / Trace Profile**
 Standard “why” artifact: policy_version_id + permitted inputs + outputs + triggered clauses + timestamps + hashes.

Applicability of IES

IES applies because policy execution spans many organizations and platforms, where document-led, bespoke implementations do not scale.

IES contribution:
- Publish and discover **versioned Policy Packs**
- Validate packs via **standard validators + rulebooks**
- Ensure consistent behavior via **test vectors + conformance**
- Enable defensible outcomes via **receipts and traces**
- Support amendment/change control via **versioning + effective-date transitions**

Before IES vs After IES (Where IES Adds Value)

Dimension	Before IES (Typical today)	After IES (UC3)
Policy representation	Text orders/circulars interpreted manually	**Policy Packs**: machine-readable, versioned, clause-linked artifacts
Implementation consistency	Vendor/DISCOM-specific logic → drift	Deterministic execution semantics + conformance tests reduce drift
Change propagation	Amendments roll out slowly and unevenly	Versioned packs with effective dates enable faster, uniform rollout
Single source of truth	Multiple "translated" implementations exist	SoT = **attested Policy Pack version** referenced everywhere
Validation gate	Reviews are manual; errors ship into production	Standard validators + **test vectors** as a formal validation gate
Explainability ("why this outcome?")	Opaque system logic; poor traceability	Standard **trace/receipt** referencing policy version + triggered clauses
Audit readiness	Hard to reproduce decisions from past periods	Reproducible execution tied to a specific version and inputs
Interoperability	Each policy needs bespoke integration	Common pack/profile interfaces reduce bespoke work across ecosystem
Governance & sign-off	PDFs signed; logic changes not provably tied	**Attestation over pack hash** ties sign-off to executable content
IES role clarity	Not applicable	IES defines **formats, semantics, conformance** (not portals/workflows)

Outside IES scope:
- Legal drafting and adjudication
- Internal DISCOM approvals/governance workflows
- Vendor release management and UI portals
- Payment/settlement integrations beyond computed outputs and traces

Stakeholders

Stakeholder	Role	What they get
Policy Owner / Regulator (SERC/CERC, Govt)	Defines and publishes packs	Faster rollout, fewer disputes, audit-ready evidence

DISCOMs	Consume packs; execute rules	Uniform decisions, quicker amendments, explainability
Technology Providers / Vendors	Implement and integrate	Standard artifacts, less bespoke work, conformance clarity
Consumers / Participants	Receive decisions	Predictable outcomes and reasons
Auditors / Programs / Ministries	Verify outcomes	Reproducible rule execution linked to versioned packs

Key Outcomes

- **Primary:** Uniform, deterministic implementation of tariff/policy logic across ecosystems
- **Secondary:** Faster change propagation with clear effective dates and version transitions
- **Tertiary:** Strong auditability via verifiable receipts/traces tied to pack versions

Actors

- **Regulator / SERC**: issues tariff orders, defines applicability, and publishes official sources
- **DISCOM / Utility**: applies policy for billing and settlements; may publish utility-level overlays/signals
- **Market / Platform Provider**: consumes executable policies to compute charges and settle outcomes
- **Consumer / Prosumer**: sees explainable tariff computation and price signals
- **Auditor / Govt**: verifies that applied tariff matches policy pack

Primitives & API Spec

Common envelope

Envelope schema (key → type)

```
{
  "header": "object",
  "body": "object"
}
```

CommonHeader (shared by ALL top-level primitives)

All top-level primitives in this spec MUST use the following common header.
Only the **body** changes by primitive.

CommonHeader schema (exactly as required; key → type)

```
{
  "policy_name": "string, required",
  "policy_record_id": "string, required, system-generated",
  "issuing_authority": "string, required",
```

```
  "jurisdiction": "JurisdictionFilter, required",
  "instrument_type": "enum: Tariff Order | Notification | Circular | Regulation | Other",
  "instrument_reference": "string, required",
  "issuance_date": "date, required",
  "effective_from": "date, required",
  "effective_to": "date or 'until revised', required",
  "status": "enum: Draft | In consultation | Active | Retired",
  "source_url": "string, required",

  "signing_officer": "string, optional",
  "legal_references": ["array of strings, optional, e.g. 'Section 42, Electricity Act 2003'"],
  "supersedes": {
    "policy_record_id": "string, optional",
    "policy_name": "string, optional"
  },
  "transition_rules": "string or object, optional",
  "last_updated": "date, optional"
}
```

JurisdictionFilter (composable applicability filters)

`jurisdiction` MUST be represented as **filters** (not only a flat list). This allows policies to apply to:
- **Geographies** (state/district/utility area/custom polygons), and
- **Target audiences / consumer segments** (category/slab/eligibility rules).

Schema (key → type)

```
{
  "geography": "GeoFilter, required",
  "consumer_segment": "ConsumerSegmentFilter, optional",
  "expression": "object (JSON-LD expression), optional"
}
```

Notes
- `expression` is an optional **policy expression** field for advanced applicability (e.g., combined constraints), expressed as a JSON-friendly, programmable form. JSON-LD is a recommended option for plug-and-play expression possibilities.

GeoFilter

Represents where a policy applies. MUST support GeoJSON; MAY also carry schema.org-aligned location metadata.

Schema (key → type)

```
{
  "geojson": "object (GeoJSON Feature or FeatureCollection), required",
  "schema_org_location": "object (schema.org Place/AdministrativeArea), optional"
}
```

Minimum supported GeoJSON shapes
- `Feature` with `geometry.type` in: `Point | Polygon | MultiPolygon`
- `FeatureCollection` of the above

ConsumerSegmentFilter

Represents who the policy applies to (consumer segment / eligibility).

Schema (key → type)

```
{
  "category": "string, optional (e.g., 'LT_domestic'|'HT_industrial'|...)",
  "slab_or_eligibility": "object, optional (e.g., sanctioned_load ranges, consumption band, subsidy el
  "voltage_level": "string?, optional",
  "metering": "array<string>?, optional",
  "expression": "object (JSON-LD expression), optional"
}
```

Rules
- Top-level objects MUST be `{ header: CommonHeader, body: ... }`.
- Anything that is **tariff-specific / state-specific / utility-specific** lives **inside** `body` (IES does not standardize every tariff table).
- Nested types inside `body` (e.g., `TariffPlan`, ToD overlays, dynamic signals) are **body-only** and MUST NOT include `header`.

PolicyRecord

Purpose: registry metadata for discovery + provenance; points to effective versions (EPO artifacts).

Schema (key → type)

```
{
  "header": "CommonHeader",
  "body": {
    "published_effective_versions": "array<PublishedEffectiveVersion>"
  }
}
```

PublishedEffectiveVersion (key → type)

```
{
  "policy_pack_version_id": "string",
  "version": "string (e.g., '1.0.0')",
  "effective_from": "date",
  "effective_to": "date or 'until revised'",
  "artifact": {
    "media_type": "string (e.g., 'application/json')",
    "url": "string",
    "hash": "string? (e.g., 'sha256:...')"
  }
}
```

Example

```
{
  "header": {
    "policy_name": "FY 2025-26 Tariff — LT Domestic",
```

```
    "policy_record_id": "IN.XX.SERC.TARIFF.2025-26.LT_DOMESTIC",
    "issuing_authority": "XX State Electricity Regulatory Commission",
    "jurisdiction": {
      "geography": {
        "geojson": {
          "type": "Feature",
          "properties": { "label": "XX (illustrative boundary)" },
          "geometry": { "type": "Polygon", "coordinates": [[[0,0],[0,1],[1,1],[1,0],[0,0]]] }
        },
        "schema_org_location": { "@type": "AdministrativeArea", "name": "XX" }
      },
      "consumer_segment": {
        "category": "LT_domestic"
      }
    },
    "instrument_type": "Tariff Order",
    "instrument_reference": "Tariff Order for FY 2025-26",
    "issuance_date": "2025-03-31",
    "effective_from": "2025-04-01",
    "effective_to": "2026-03-31",
    "status": "Active",
    "source_url": "https://example.org/serc-xx/tariff-order-2025-26.pdf",
    "signing_officer": "Secretary, XX SERC",
    "legal_references": ["Section 42, Electricity Act 2003"],
    "supersedes": {
      "policy_record_id": "IN.XX.SERC.TARIFF.2024-25.LT_DOMESTIC",
      "policy_name": "FY 2024-25 Tariff – LT Domestic"
    },
    "transition_rules": "New rates apply from 01-Apr-2025 for all bills with consumption after this da
    "last_updated": "2025-04-02"
  },
  "body": {
    "published_effective_versions": [
      {
        "policy_pack_version_id": "PPV-IN.XX.SERC.TARIFF.2025-26.LT_DOMESTIC-0001",
        "version": "1.0.0",
        "effective_from": "2025-04-01",
        "effective_to": "2026-03-31",
        "artifact": {
          "media_type": "application/json",
          "url": "https://registry.example.org/v1/epos/EPO-IN.XX.SERC.TARIFF.2025-26.LT_DOMESTIC.v1.js
          "hash": "sha256:EXAMPLE_EPO_HASH"
        }
      }
    ]
  }
}
```


EffectivePolicyObject (EPO)

Purpose: minimal executable policy payload for apps (billing / plan selection / settlement).

Schema (key → type)


```
{
  "header": "CommonHeader",
  "body": {
```

```
    "pricing_model": "TariffPlan | ToDOverlay | DynamicPriceSignal",
    "policy_expression": "object (JSON-LD expression), optional",
    "clause_mapping": "array<ClauseRef>?",
    "test_vectors": "array<TestVector>?"
  }
}
```

Note on tariff records: the *actual tariff tables* (slabs, charges, ToD slots, event rates) are represented as **body models** below.
IES keeps these models generic enough to cover many policies, without forcing a bespoke top-level header field per tariff parameter.

ClauseRef (key → type)

```
{
  "clause_id": "string",
  "label": "string?",
  "source_url": "string?",
  "notes": "string?"
}
```

TestVector (key → type)

```
{
  "name": "string",
  "inputs": "object",
  "expected": "object"
}
```

Example

```
{
  "header": {
    "policy_name": "FY 2025-26 Tariff — LT Domestic",
    "policy_record_id": "IN.XX.SERC.TARIFF.2025-26.LT_DOMESTIC",
    "issuing_authority": "XX State Electricity Regulatory Commission",
    "jurisdiction": {
      "geography": {
        "geojson": {
          "type": "Feature",
          "properties": { "label": "XX (illustrative boundary)" },
          "geometry": { "type": "Polygon", "coordinates": [[[0,0],[0,1],[1,1],[1,0],[0,0]]] }
        },
        "schema_org_location": { "@type": "AdministrativeArea", "name": "XX" }
      },
      "consumer_segment": {
        "category": "LT_domestic"
      }
    },
    "instrument_type": "Tariff Order",
    "instrument_reference": "Tariff Order for FY 2025-26",
    "issuance_date": "2025-03-31",
    "effective_from": "2025-04-01",
    "effective_to": "2026-03-31",
    "status": "Active",
    "source_url": "https://example.org/serc-xx/tariff-order-2025-26.pdf",
    "legal_references": ["Section 42, Electricity Act 2003"],
```

```
    "last_updated": "2025-04-02"
  },
  "body": {
    "pricing_model": {
      "plan_type": "tariff_plan",
      "consumer_applicability": {
        "category": "LT_domestic",
        "voltage_level": "LT",
        "metering": ["single_phase", "smart_optional"]
      },
      "components": {
        "fixed_charge": { "basis": "per_month", "amount_inr": 120 },
        "energy_charge": {
          "type": "slabbed",
          "unit": "INR/kWh",
          "slabs": [
            { "from_kwh": 0, "to_kwh": 100, "rate": 4.50 },
            { "from_kwh": 101, "to_kwh": 200, "rate": 6.00 },
            { "from_kwh": 201, "to_kwh": null, "rate": 7.20 }
          ]
        }
      },
      "overlays": []
    },
    "clause_mapping": [
      { "clause_id": "TARIFF-LT-DOM-ENERGY-01", "label": "Energy charges (slabs)", "source_url": "http
    ],
    "test_vectors": [
      {
        "name": "LT domestic 250 kWh",
        "inputs": { "kwh": 250 },
        "expected": { "fixed_charge_inr": 120, "energy_charge_inr": 100*4.5 + 100*6.0 + 50*7.2 }
      }
    ]
  }
}
```

TariffPlan

Purpose: fixed/demand/energy components. Energy charges can be flat or slabbed.
IES note: tariff-specific slabs/charges are in `body` and differ by policy; IES does not define per-state slab tables at the platform level.

Schema (key → type)

```
{
  "plan_type": "string = 'tariff_plan'",
  "consumer_applicability": {
    "category": "string",
    "voltage_level": "string?",
    "metering": "array<string>?"
  },
  "components": {
    "fixed_charge": "FixedCharge?",
    "demand_charge": "DemandCharge?",
    "energy_charge": "EnergyCharge"
  },
```

```
  "overlays": "array<ToDOverlay>?"
}
```

FixedCharge

```
{
  "basis": "string (e.g., 'per_month'|'per_connection')",
  "amount_inr": "number"
}
```

DemandCharge (optional)

```
{
  "basis": "string (e.g., 'per_kW'|'per_kVA')",
  "rate_inr": "number",
  "billing_demand_rule": "string?"
}
```

EnergyCharge

```
{
  "type": "string = 'flat'|'slabbed'",
  "unit": "string (e.g., 'INR/kWh')",
  "rate": "number? (required if flat)",
  "slabs": "array<{from_kwh:number, to_kwh:number|null, rate:number}>? (required if slabbed)"
}
```

Example

```
{
  "plan_type": "tariff_plan",
  "consumer_applicability": { "category": "LT_domestic", "voltage_level": "LT" },
  "components": {
    "fixed_charge": { "basis": "per_month", "amount_inr": 120 },
    "energy_charge": {
      "type": "slabbed",
      "unit": "INR/kWh",
      "slabs": [
        { "from_kwh": 0, "to_kwh": 100, "rate": 4.50 },
        { "from_kwh": 101, "to_kwh": 200, "rate": 6.00 },
        { "from_kwh": 201, "to_kwh": null, "rate": 7.20 }
      ]
    }
  },
  "overlays": []
}
```

ToDOverlay

Purpose: slot-based adjustment applied to a component (typically energy charge).

Schema (key → type)

```
{
  "overlay_type": "string = 'tod'",
  "applies_to_component": "string? (e.g., 'energy_charge')",
  "time_zone": "string? (e.g., 'Asia/Kolkata')",
  "time_slots": "array<{name:string, from:string(HH:MM), to:string(HH:MM)}>",
  "adjustment_mode": "string = 'multiplier'|'adder'|'percentage'",
  "adjustments": "array<{slot:string, mode:string, value:number}>",
  "applicability_conditions": "object?"
}
```

Example

```
{
  "overlay_type": "tod",
  "applies_to_component": "energy_charge",
  "time_zone": "Asia/Kolkata",
  "time_slots": [
    { "name": "peak_evening", "from": "18:00", "to": "22:00" },
    { "name": "solar_hours", "from": "09:00", "to": "17:00" }
  ],
  "adjustment_mode": "multiplier",
  "adjustments": [
    { "slot": "peak_evening", "mode": "multiplier", "value": 1.10 },
    { "slot": "solar_hours", "mode": "multiplier", "value": 0.90 }
  ],
  "applicability_conditions": { "requires_metering": ["smart", "tod"] }
}
```

DynamicPriceSignal

Purpose: time-series pricing (day-ahead, real-time, event-based) bound to a base tariff.

Schema (key → type)

```
{
  "plan_type": "string = 'dynamic_price_signal'",
  "signal": {
    "signal_kind": "string = 'day_ahead'|'real_time'|'critical_peak_event'",
    "granularity_minutes": "number",
    "currency": "string",
    "unit": "string",
    "time_zone": "string?",
    "points": "array<{ts:string(date-time), value:number}>?"
  },
  "binding_rule": {
    "apply_to": "string",
    "merge_strategy": "string = 'overlay'|'replace'",
    "base_tariff_policy_record_id": "string?"
  }
}
```

Example

```
{
  "plan_type": "dynamic_price_signal",
  "signal": {
```

```
    "signal_kind": "day_ahead",
    "granularity_minutes": 60,
    "currency": "INR",
    "unit": "INR/kWh",
    "time_zone": "Asia/Kolkata",
    "points": [
      { "ts": "2026-02-01T00:00:00+05:30", "value": 6.25 },
      { "ts": "2026-02-01T01:00:00+05:30", "value": 5.90 }
    ]
  },
  "binding_rule": {
    "apply_to": "energy_charge",
    "merge_strategy": "overlay",
    "base_tariff_policy_record_id": "IN.XX.SERC.TARIFF.2025-26.LT_DOMESTIC"
  }
}
```

API spec (simplified)

Discovery

Method	Path	Purpose
GET	`/.well-known/ies/policy-registry`	endpoints + supported schemas

Beckn protocol interface (reuse Beckn; no new protocol)

Policy discovery and resolution SHOULD be implemented using standard Beckn **Discovery** APIs (search → on_search) and (optionally) **Item detail** APIs (select → on_select). This keeps the interaction compatible with Beckn gateways/registries and allows multiple policy registries to participate without bespoke integration.

Beckn roles
- **BAP**: Policy Consumer App (utility system, market platform, auditor tool, etc.)
- **BPP**: Policy Registry Provider (publishes policy catalogs + policy packs)
- **BG (optional)**: Beckn Gateway for discovery broadcast/routing
- **Registry (optional)**: Beckn Registry for participant lookup

API surface (Beckn)

Beckn API	Direction	Purpose
`search`	BAP → BG/BPP	Discover policies matching filters (jurisdiction, segment, kind, effective date)
`on_search`	BPP → BAP (via BG if used)	Return a catalog of matching policies (as items)

`select` *(optional)*	BAP → BPP	Request the full policy record / policy pack for a chosen policy item
`on_select` *(optional)*	BPP → BAP	Return the chosen policy with requested resolution (link or embedded pack)

Notes:
- **Applicability filters** (GeoJSON + consumer segment) are carried as structured data inside Beckn **Tags** on `message.intent` (for search) and on returned `item.tags` (for on_search). This avoids breaking Beckn interoperability while supporting richer filters than the base schema.
- If a network requires it, the same “detail” resolution can be modeled via `init/on_init`, but `select/on_select` is usually sufficient for “view item details” flows.

Beckn `search` request (policy discovery)

```
{
  "context": {
    "domain": "energy.policy",
    "country": "IND",
    "city": "std:080",
    "action": "search",
    "core_version": "1.0.0",
    "bap_id": "policy-consumer.example.org",
    "bap_uri": "https://policy-consumer.example.org/beckn",
    "transaction_id": "3c1b5e1e-4b8f-4b2b-9b44-2e0c1a5a8c11",
    "message_id": "8e43c8d8-62d0-4b47-9ab0-0d9a4c1c0b7f",
    "timestamp": "2026-02-03T10:30:00Z",
    "ttl": "PT30S"
  },
  "message": {
    "intent": {
      "item": {
        "descriptor": { "name": "Tariff Order" },
        "tags": [
          {
            "descriptor": { "name": "policy_filters" },
            "list": [
              { "descriptor": { "code": "kind" }, "value": "Tariff Order" },
              { "descriptor": { "code": "effective_on" }, "value": "2026-02-01" },
              { "descriptor": { "code": "issuing_authority" }, "value": "KERC" },

              { "descriptor": { "code": "jurisdiction.geography.geojson" }, "value": "{...GeoJSON Feat
              { "descriptor": { "code": "jurisdiction.geography.schema_org_location" }, "value": "{ "@

              { "descriptor": { "code": "jurisdiction.consumer_segment.category" }, "value": "LT_domes
              { "descriptor": { "code": "jurisdiction.consumer_segment.slab_or_eligibility" }, "value"

              { "descriptor": { "code": "jurisdiction.expression" }, "value": "{...JSON-LD expression.
            ]
          }
        ]
      }
```

```
    }
  }
}
```

Beckn on_search response (catalog of policies)

```
{
  "context": {
    "domain": "energy.policy",
    "country": "IND",
    "city": "std:080",
    "action": "on_search",
    "core_version": "1.0.0",
    "bpp_id": "policy-registry.example.org",
    "bpp_uri": "https://policy-registry.example.org/beckn",
    "transaction_id": "3c1b5e1e-4b8f-4b2b-9b44-2e0c1a5a8c11",
    "message_id": "b7cc2f7b-3af2-4e71-9e6d-7d70f3d5d1d0",
    "timestamp": "2026-02-03T10:30:01Z"
  },
  "message": {
    "catalog": {
      "providers": [
        {
          "id": "kerc-policy-registry",
          "descriptor": { "name": "KERC Policy Registry" },
          "items": [
            {
              "id": "KERC-TO-2025-001",
              "descriptor": { "name": "KERC Tariff Order FY2025-26" },
              "tags": [
                {
                  "descriptor": { "name": "policy_meta" },
                  "list": [
                    { "descriptor": { "code": "policy_record_id" }, "value": "KERC-TO-2025-001" },
                    { "descriptor": { "code": "instrument_type" }, "value": "Tariff Order" },
                    { "descriptor": { "code": "instrument_reference" }, "value": "KERC-TO-2025-26/1234
                    { "descriptor": { "code": "effective_from" }, "value": "2025-04-01" },
                    { "descriptor": { "code": "effective_to" }, "value": "until revised" },
                    { "descriptor": { "code": "source_url" }, "value": "https://kerc.karnataka.gov.in/

                    { "descriptor": { "code": "policy_pack.format" }, "value": "application/ld+json" }
                    { "descriptor": { "code": "policy_pack.url" }, "value": "https://policy-registry.e
                    { "descriptor": { "code": "policy_pack.digest" }, "value": "sha256:..." }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  }
}
```

Beckn `select` / `on_select` (optional policy pack resolution)
Use this when the BAP needs the **full** policy record / policy pack (instead of just metadata).

`select` **request**: choose `item.id` and add a tag such as `resolution = policy_pack | full_record` and `embed = true|false`.

`on_select` **response**: return the `order.items[0]` with either:
- `policy_pack.url` (recommended for large packs), or
- an embedded `policy_pack` payload in tags (only if size permits).

Signals

Method	Path	Purpose
GET	`/v1/signals/{policy_pack_version_id}?from=…&to=…`	fetch dynamic price points

Simulation (recommended)

Method	Path	Purpose
POST	`/v1/simulate-bill`	compute bill breakdown

Error envelope

```
{
  "error": {
    "code": "POLICY_NOT_FOUND",
    "message": "No policy matched the query",
    "details": { "policy_record_id": "…", "date": "…" }
  }
}
```