30th April 2026

REC Tariff Interoperability Implementation Working Group

recco

Agenda

AGENDA ITEM	PRESENTER
Welcome & Introduction	Chair
Confirmation of Agenda	Chair
Example Payload Walkthrough	Nizar Hamood
Update on Testing	John Aung
Change Log	Nizar Hamood
Introduction to TI Phase 1a Performance Assurance	Andrew Wallace
Questions from Previous IWG	Chris Stock
Q&A	Chair
AOB	Chair

Example Payload Walkthrough

Worked examples against the baselined API payload

The envelope every tariff shares

A response from GET /tariff/{mpid}/{tariff_id} carries three logical groups. The envelope identifies the tariff and its lifecycle. The keys describe who the tariff is for. The rates array is the pricing engine.

Envelope

Identity and lifecycle

name · description · comments
tariff_type (static or dynamic)
valid_from / valid_to
sellable_from / sellable_to
last_modified
percentage_green

Keys

Who the tariff applies to

fuel_type (E or G)
metering_point_energy_flow (I/E)
meter_type (array)
payment_method
gsp_group_id (electricity)
ldz_regions (gas)

Pricing

Standing charge + rates array

standing_charge (per day)
rates[] — one or more entries
Each rate carries its own
type_of_use, time window,
optional kWh band, and
unit_price

The core design fork — how rates[] behaves

tariff_type determines which set of fields inside each rate entry carry values, and which must be null. This discipline is what lets TI Users render prices correctly without needing supplier-specific handling.

STATIC

Rates are defined by day pattern and time window

tariff_days — e.g. [Monday..Friday]
time_from — e.g. 00:00:00
time_to — e.g. 06:00:00
valid_from — *null at rate level*
valid_to — *null at rate level*

Lifecycle lives on the envelope (valid_from / valid_to at the tariff level).

DYNAMIC

Rates are defined by explicit datetime windows

valid_from — e.g. 2026-04-02T00:00:00Z
valid_to — e.g. 2026-04-02T00:30:00Z
tariff_days — *not required*
time_from — *not required*
time_to — *not required*

Suppliers refresh dynamic rates on their own cadence — typically daily for next-day pricing.

Simple flat rate — the minimum viable tariff

FUEL **Electricity** · TYPE **static** · PAYMENT **quarterly_cash** · METER **S2A** · GSP **_B**

PAYLOAD · abbreviated — *Scenario 5b*


```
{
  "name": "peakwatt-standard-elec-2026",
  "tariff_type": "static",
  "valid_from": "2026-04-01T00:00:00.00Z",
  "valid_to": "2026-06-30T23:59:59.00Z",
  "sellable_from": "2026-03-20T00:00:00.00Z",
  "sellable_to": "2026-04-30T23:59:59.00Z",
  "fuel_type": "E",
  "payment_method": "quarterly_cash",
  "gsp_group_id": "_B",
  "metering_point_energy_flow": "I",
  "meter_type": ["S2A"],
  "standing_charge": "0.61200",
  "rates": [
    {
      "type_of_use": "all",
      "tariff_days": ["All"],
      "time_from": "00:00:00",
      "time_to": "23:59:59",
      "unit_price": "0.24500"
    }
  ]
}
```


Full payload omits: description, comments, last_modified, percentage_green.

WHAT TO LOOK AT

One rate covers everything

A single rates[] entry with tariff_days=["All"] and a 24-hour window.

type_of_use = "all"

No segmentation between EV / Not EV / other use. Every kWh is priced identically.

Numbers as strings

unit_price and standing_charge are JSON strings, to preserve decimal precision.

Sellable window can close

sellable_to is set — this tariff stops being offered on 30 April 2026 but remains valid for existing customers.

Static tariff with time-of-use and EV split

FUEL **Electricity** · TYPE **static** · PAYMENT **direct_debit** · METER **S1** · RATES **4**

PAYLOAD · abbreviated — *Scenario 1*

```
{
  "name": "greenflux-fix-ev-smart-2026",
  "tariff_type": "static",
  "fuel_type": "E",
  "gsp_group_id": "_A",
  "payment_method": "direct_debit",
  "standing_charge": "0.48210",
  "rates": [
    { "type_of_use": "EV",
      "tariff_days": ["Monday".."Friday"],
      "time_from": "00:00:00",
      "time_to": "06:00:00",
      "unit_price": "0.07500"
    },
    { "type_of_use": "EV",  time_from 06:00, unit_price 0.24150 },
    { "type_of_use": "EV", Sat–Sun all day, 0.07500 },
    { "type_of_use": "Not EV", all days, 0.24150 }
  ]
}
```

Abbreviated for slide space: 2 of 4 rate entries shown in full; lifecycle, comments and percentage_green omitted.

WHAT TO LOOK AT

type_of_use segments the day

Separate rates for EV charging (cheap overnight, peak daytime) and household "Not EV" usage.

tariff_days as array

Each rate entry lists the days it applies. Allows split weekday / weekend pricing in one tariff.

Multiple entries, one tariff_id

The full rate schedule is expressed by 4 entries — collectively they cover every day × every type_of_use.

Same envelope as the simple case

Complexity lives entirely in rates[]. The supplier's envelope shape doesn't change.

Dynamic half-hourly tariff

FUEL **Electricity** · TYPE **dynamic** · PAYMENT **direct_debit** · METER **S2AD** · CADENCE **daily refresh**

PAYLOAD · abbreviated — *Scenario 2*


```
{
  "name": "voltstream-agile-apr-2026",
  "tariff_type": "dynamic",
  "fuel_type": "E",
  "gsp_group_id": "_C",
  "last_modified": "2026-04-01T16:00:00.00Z",
  "valid_to": null,
  "sellable_to": null,
  "standing_charge": "0.52380",
  "rates": [
    {
      "type_of_use": "all",
      "valid_from": "2026-04-02T00:00:00.00Z",
      "valid_to": "2026-04-02T00:30:00.00Z",
      "unit_price": "0.08120"
    },
    { ... 00:30 → 01:00, 0.07540 },
    { ... 04:00 → 04:30,  -0.0250 },
    { ... 16:00 → 16:30, 0.38750 },
    { ... 17:00 → 17:30, 0.42100 }
  ]
}
```


WHAT TO LOOK AT

Rate window is an explicit datetime

Each rate entry carries valid_from and valid_to — no tariff_days, no time_from/time_to.

Negative unit prices supported

During excess renewable generation the price can go below zero. Supplier's business rules, not a TI constraint.

Null valid_to on the envelope

The tariff is open-ended at the top level — but individual rate entries are always time-bounded.

48 entries per day is normal

Half-hourly pricing means ~48 rate entries per daily refresh. Payload size and caching strategy matter here.

Rates abbreviated — full response contains one entry per half-hourly slot. 2 of 48 shown in full.

Block tariff — unit price by consumption band

FUEL **Gas** · TYPE **static** · PAYMENT **direct_debit** · METER **S1** · LDZ **EM**

PAYLOAD · abbreviated — *Scenario 6*

```
{
  "name": "hearthstone-gas-block-em-2026",
  "tariff_type": "static",
  "fuel_type": "G",
  "ldz_regions": "EM",    // gas-specific geography
  "payment_method": "direct_debit",
  "standing_charge": "0.30120",
  "rates": [
    {
      "type_of_use": "all",
      "tariff_days": ["All"],
      "min_kWh": "0",       "max_kWh": "1000",
      "unit_price": "0.05890"
    },
    {
      "type_of_use": "all",
      "min_kWh": "1000",    // no max_kWh = open upper bound
      "unit_price": "0.07250"
    }
  ]
}
```

Abbreviated: lifecycle fields and comments string omitted.

WHAT TO LOOK AT

min_kWh and max_kWh define the band

First 1000 kWh at the lower rate; everything above at the higher tier. Measurement period is supplier-defined (here: quarterly).

Omitting max_kWh = open upper bound

Second tier has no max_kWh set, meaning the rate applies to all consumption above 1000 kWh.

ldz_regions is gas-specific

For gas tariffs the geography key is ldz_regions, not gsp_group_id. Suppliers populate one or the other by fuel type.

Measurement period is not in-payload

The TI payload carries the band thresholds; measurement window semantics are described in comments or supplier docs.

Design patterns that recur across every example

A single payload shape absorbs the full breadth of GB tariff complexity — flat, time-of-use, dynamic, block, and legacy register-based. Complexity lives in the rates array, not in the envelope.

Envelope is stable

Every tariff has the same top-level identity, keys, and lifecycle fields. Suppliers implement it once.

rates[] does the heavy lifting

All variety — ToU, dynamic, block, E7, EV splits — is expressed as different shapes of rate entries.

tariff_type is the discipline

static vs dynamic controls which fields inside each rate carry values and which stay null.

type_of_use carves usage

Separates EV from household, or export from import, without needing additional endpoints.

Fuel-specific geography

Electricity uses gsp_group_id, gas uses ldz_regions. One is populated, the other omitted.

Numbers as strings

unit_price, standing_charge, min_kWh / max_kWh are JSON strings for decimal precision.

Nulls are expressive

valid_to = null means open-ended. sellable_to = null means still being sold. percentage_green = null means not disclosed.

Arrays allow inclusivity

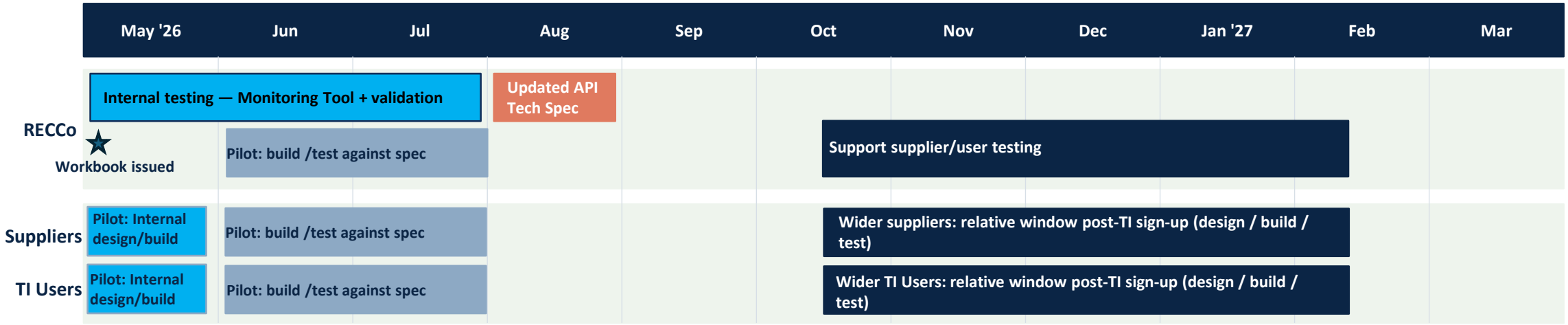
meter_type, tariff_days, and the rates array itself all accept multiple values where real tariffs demand it.

Update on Testing

Plan on a page, pilot testing, internal testing, test case examples, and what we need

TI Plan on a Page

RECCo deliverables carry fixed calendar dates. Individual supplier/user windows are expressed relative to TI programme sign-up, so suppliers/users can plan their own design, build and test activity around existing change commitments.



Fixed RECCo dates

Test workbook version, pre-live ping windows and pilot-group anchor dates are absolute.

Relative supplier, user windows

Post-signup: N days to design, N days to build, N days to test. Integrates with supplier / users change plans.

Maximum warning

Dates published now so suppliers can schedule against other changes (system procurement, etc).

Pilot Testing

Suppliers test their own hosted TI endpoints against the TI API Technical Specification, record evidence in a RECCo-issued test workbook. TI Users test End-to-End with RECCo and Suppliers

What suppliers do

- Internal development against the TI API Technical Specification
- Execute the mandatory test cases against their own endpoints
- Log defects and internally agree work-off plans
- Retain the completed workbook as evidence
- Allow TI User and RECCo to hit their end points for testing

What RECCo provides

- A draft versioned test case workbook
- Issue final test case workbook post Pilot testing
- TI Monitoring Tool
- Validation Service / simulator
- TI Supplier register
- Clarifications via the questions log / FAQ
- Review of evidence on request

What TI Users do

- Internal development against the TI API Technical Specification
- Testing with RECCo
- Testing with Suppliers

Internal testing and internal supplier testing run

RECCo — Internal Testing

What we test inside RECCo before suppliers start

TI Monitoring Tool

Endpoint discovery, scheduling, validation against spec, retry behaviour.

Validation service

Payload conformance rules, enum checks, schema enforcement, simulator.

Database write-back

Monitoring results persisted and surfaced to reporting layer.

Suppliers — Endpoint Testing

What you test on your own hosted endpoints

GET /tariffs/{mpid}

List response, schema, filters, error handling.

GET /tariff/{mpid}/{tariff_id}

Detail response, required fields, static vs dynamic semantics.

Non-functionals

Latency, availability, absence of 5xx errors on the above endpoints.

Three endpoints are in scope for testing

Phase 1A supplier/user testing targets three endpoints, two tariff endpoints that suppliers host and one central register endpoint which RECCo host.

IN SCOPE

`GET /suppliers`

Central TI Supplier Register. Operated by RECCo

`GET /tariffs/{mpid}`

List endpoint. 10 mandatory test cases: basic response, schema, six filters, combined filters, invalid enum handling.

`GET /tariff/{mpid}/{tariff_id}`

Detail endpoint. 9 mandatory test cases: required fields, rates structure, null handling, static vs dynamic, 404 behaviour.

`Non-functionals`

3 mandatory test cases: P95 latency, availability across multiple test periods, absence of 5xx errors.

23 mandatory test cases across four categories (final version issued post Pilot testing)

This is the scope we are proposing for IWG review. Detailed test steps will be shared for a deeper review at the next touchpoint. TI User test cases to be added.

23 Mandatory test cases	**4** Categories	**2** Endpoints in scope	**1** Test workbook issued by RECCo

Category	Sub-category coverage	Cases
GET /tariffs/{mpid}	Basic response · Schema validation · Filter: fuel_type, tariff_type, payment_method, direction, gsp_group_id, ldz_region · Combined filters · Invalid enum error	10
GET /tariff/{mpid}/{tariff_id}	Basic response · Required fields · Rates structure · Null handling (valid_to / sellable_to) · String types (standing_charge / unit_price) · List consistency · Unknown tariff_id 404 · Dynamic valid_from/valid_to fields · Price update freshness	9
Non-functionals	P95 latency within threshold · Availability across multiple test periods · No 5xx errors under normal operation	3
Pre-live handshake	Supplier endpoint responds correctly to the TI Monitoring Tool validation ping	1

TC-TAR-003 — Filter: fuel_type

TC ID	CATEGORY	SUB-CATEGORY	PRIORITY	PARTY
TC-TAR-003	GET /tariffs/{mpid}	Filter — fuel_type	Mandatory	Supplier

TEST STEPS

What the supplier executes

1. Issue: GET /tariffs/{mpid}?fuel_type=E
2. Verify HTTP 200 is returned
3. Verify every returned object has fuel_type='E'
4. If gas tariffs exist in the portfolio, repeat with fuel_type=G
5. Verify every object in that response has fuel_type='G'

EXPECTED RESULT

What a PASS looks like

- **HTTP 200**
- All returned tariffs match the requested fuel_type exactly
- No cross-contamination between fuel types
- Empty array (not an error) if no tariffs of that type exist

TC-DET-008 — Dynamic tariff valid_from/valid_to fields

TC ID	CATEGORY	SUB-CATEGORY	PRIORITY	PARTY
TC-DET-008	GET /tariff/{mpid}/{tariff_id}	Dynamic — valid_from/valid_to fields	Mandatory	Supplier

TEST STEPS

Why this matters: static vs dynamic discipline

1. Get a list of dynamic tariffs: GET /tariffs/{mpid}?tariff_type=dynamic
2. Issue GET /tariff/{mpid}/{tariff_id} for that dynamic tariff
3. Verify tariff_type = 'dynamic'
4. **For each entry in rates[], verify:**

- valid_from — valid ISO 8601 datetime (NOT null)
- valid_to — valid ISO 8601 datetime (NOT null)
- days_of_week — null
- time_from — null
- time_to — null

EXPECTED RESULT

What a PASS looks like

- Dynamic rates use valid_from and valid _to, both non-null
- Static-specific fields (days_of_week, time_from, time_to) are null for dynamic tariffs
- Tariff type discipline is preserved: no mixed semantics within a single rate entry

Why we test this. Mixed static/dynamic semantics break TI User price rendering and undermine the two-endpoint design.

NEXT STEPS

What we need from the IWG

01 Feedback on test case SCOPE

The Supplier test case workbook will be issued towards the end of next week. Does the category coverage look right? Any obvious gaps in what we're proposing to test against the two endpoints? Written feedback welcomed within one week.

02 Volunteers for the pilot group

We are short on Phase 1A pilot participants. Pilot involvement is much lower burden than Phase 1B and is a strong signal to the market.

03 Timeline constraints to flag

If your organisation has known change-freeze or major delivery windows that would clash with the testing slots above, tell us now so we can plan around them.

04 Confirm the next touchpoint

Detailed test workbook will be shared for the deeper review. Please nominate a reviewer in your organisation.

Design & Decision Change Log

Governance, current state, and how we engage with the IWG

A single, open record of design decisions and change requests

The Design & Decision Change Log is the authoritative record of every change to the TI API baseline — whether proposed, accepted, delayed, or pending review. The principle is simple: everything is captured, nothing lives behind closed doors.

What it is

- The single source of truth for TI API design decisions
- A chronological record: CHG-001, CHG-002, CHG-003 and onwards
- Aligned to the versioned API Technical Specification
- Maintained by RECCo, reviewed with the IWG

What it captures

- The change itself and the category it sits in
- Rationale and business driver
- Who decided, what the decision was, and the status
- Impact and trade-offs — especially for breaking changes

Why we keep it

- Open and auditable across the market
- Supports suppliers planning their build and test effort
- Captures dissenting views and deferred items honestly
- Feeds straight into versioning of the API spec

Every change is classified, decided, and tracked

Each entry in the log carries a category, a decision status, and a rationale. Breaking changes are flagged explicitly. Items awaiting a decision are visible, so the IWG can see exactly what's moving and what's parked.

APPROVED

Decision made. Implementation confirmed. Reflected in the next version of the API Technical Specification.

IN DISCUSSION

Decision under active consideration. Rationale and trade-offs documented while alternatives are weighed.

PENDING REVIEW

Captured but not yet worked. Refreshed monthly against the backlog so nothing quietly disappears.

BREAKING CHANGE FLAG

Every change carries an explicit breaking / non-breaking label

Breaking changes are implemented when needed. Where a breaking change can be safely delayed to a later Phase or API version, the rationale for deferring is captured alongside the entry — so suppliers see the full picture, not just the outcome.

MONTHLY REVIEW CADENCE

The log is refreshed every month

To Be Reviewed items are revisited monthly. Approved and In Discussion items move forward. New entries from bilateral calls, consultation responses and IWG sessions are added as they arise.

Read-only access for IWG, single editor in RECCo

Openness is the principle; controlled editing is the mechanism. The IWG sees everything, but the log has a single set of hands on the keys to keep the record coherent.

WHAT THE IWG GETS

Full visibility of every entry

Open, In Discussion, Approved, and TBR items — the complete record, not a curated summary.

Read-only file, refreshed monthly

Shared as a PDF or locked Excel. Guaranteed not to drift between refreshes.

Full history and rationale

Every change carries its category, decision, and trade-off — nothing reduced to a one-liner.

Links to the underlying sources

Confluence pages, meeting notes, and consultation threads referenced alongside each entry.

HOW TO FEED IN

Raise change requests in writing

Email RECCo or raise via your IWG representative. Each request becomes a new CHG entry.

Comment at the IWG session

In-discussion items are open for challenge. Recorded in meeting notes and reflected in the log.

Bilateral calls are welcomed

Material points from bilaterals (e.g. Suppliers) are captured into the log by RECCo.

Editorial control stays with RECCo

One set of hands on the record to keep entries coherent, versioned, and free of duplicates.

Three changes tracked today

The full record will be circulated alongside the IWG papers. The entries below are a snapshot as at today.

2	Tracked changes
1	Approved
1	In Discussion
0	To Be Reviewed

ID	Category	Change	Rationale	Status
CHG-001	API Payload Structure	Baselined current payload structure with days/hours as one object, hours nested with valid-from/valid-to and single rate per period.	Need a stable baseline before suppliers begin building against their tariffs.	APPROVED
CHG-002	Economy 7 Handling	Example proposed: E7 as parent element with rates nested by hours, include type-of-use field.	Simplifies tariff representation for multi-rate tariffs like E7.	IN DISCUSSION

Discipline: changes go here, everything else is routed elsewhere

Not every message from a supplier is a change request. Keeping the log tight is what makes it useful. The recent Centrica exchange is a good worked example of how we triage.

BELONGS IN THE LOG

In-scope change requests

Proposed amendments to the API spec baseline, with or without a decision yet taken.

Breaking-change flags

Anything affecting supplier build — even when the decision is to defer.

Design decisions

Choices made by RECCo team that lock in a direction (e.g. CHG-001 baseline).

Deferred items

Changes that will be revisited later; parked as To Be Reviewed, not lost.

ROUTED ELSEWHERE

Clarification questions → FAQ / Questions log

E.g. Supplier: 'how are sellable_from / sellable_to updated?' — a design question, not a change.

Scope-extending asks → Future-scope parking

E.g. Supllier: non-domestic pricing attributes (DUoS ID, EAC, Exit Zone, AQ) — outside Phase 1A scope.

Guidance and onboarding → Supplier documentation

Operational 'how do I' questions handled via the RECCo supplier-facing materials.

Items that may become changes → Triaged first

E.g. non-contiguous sellable periods — currently a query; promoted to the log if confirmed as a gap.

Introduction to Performance Assurance

RECCo is developing proposals to assure Supplier performance against their TI requirements

Introduction to Performance Assurance

- RECCo developing proposals to assure Supplier performance against their TI requirements.

- Aim to mitigate risk that poor Supplier performance could lead to:
  - Higher bills for consumers
  - Reduction in wider benefits e.g. to Net Zero
  - Reduced future take-up of optimisation and associated benefits

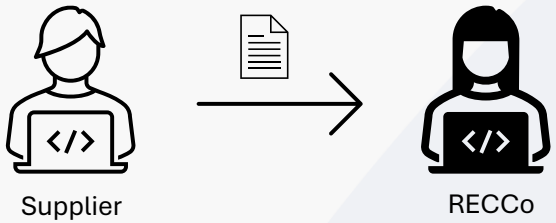
- Removed Dec 2025 Consultation proposed that Suppliers should provide a daily API with monitoring data.

- Suppliers to provide monthly report and updated Annual Statement of compliance :
  - To be defined in an update to the REC Performance Assurance Report Catalogue.
  - I0323 raised with Party Impact Assessment and Consultation later this year.
  - To provide efficiencies, monthly report includes Supplier SLC requirements.

- Supplier performance will also be assured using the TI Monitoring Tool - mimics a TI User by sending/receiving TI API messages.

- Spreadsheet with proposals provided to IWG for review and **welcome comments by 8 May** (tariffinteroperability@retailenergycode.co.uk)

Performance Assurance – Supplier reporting requirements



Supplier Monthly Report

- No. of updates to Tariff Pricing Data in month and % made < 60mins before change takes effect.
- No., duration and reason category for suspending access to all TI Users/individual TI Users.
- API availability against 99.9% SLA.
- Steady state API response time against the '*at least 95% < 1 second*' SLA.
- Peak API response time against '*at least 95% < 5 second*' SLA.
- No. of seconds in month where peak volume > 5,000 requests/second.
- No. of Get Tariffs Requests received and successful responses sent*.
- Number of Get Tariffs Details Requests received and successful responses sent*.
- Message rejection volumes, by reason.

Annual Statement

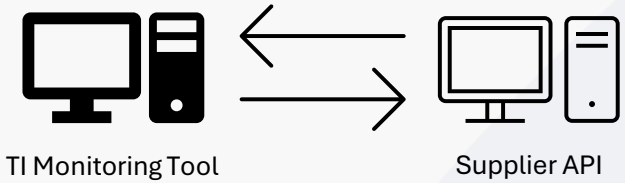
- Confirmation that all tariff data within scope is available to TI Users.
- Confirmation that arrangements in place for TI Users and Customer to raise queries.

Ad hoc/annual report

- Structure of any tariffs not captured within scope of the TI API*

*SLC reporting requirements

Performance Assurance – TI Monitoring Tool



TI Monitoring Tool

- Poll each Supplier every few minutes.
- Randomised so that checks are not predictable.
- Separate API call made for each Supplier MPID.
- Check Get Tariffs Response and Get Tariff Details Response message in each cycle.
- Validate Supplier response payloads against the API schema to ensure conformity.

Questions from Previous IWG

THEME	DETAIL	RESPONSE
Data Model & Schema Clarity	Clarity on fields, IDs, and structure	Schema refined for consistency and flexibility; guidance to be updated where needed
Implementation Support	Need for practical tools and examples	YAML and OpenAPI specs will be provided; further support such as simulators under development
Testing Approach	How testing and environments will work	Test Plan in development, including sandbox and simulators; feedback being incorporated
Security & Assurance	Data accuracy and risk management	Supplier responsibility supported by assurance and monitoring approach
Scope & Phasing	What is included in Phase 1a and 1b	Phase 1a and b focused on Public and Customer specific Electricity Tariff pricing data; future phases will expand scope
Access & User Roles	Who can access what data	Open API access for public data; RTI required for customer-specific data in Phase 1b
Operational Readiness	How the service will run in BAU	Operational processes (onboarding, monitoring, support) in development

Questions

AOB