A Resilient Broadcast Signal for Coarse-Grain DER Coordination.

A Concept Note V1.1

Glen Kleidon *BSc, GradDipEd, GradDipAppComp, CTASC(1995), CTIAC(1999)*

Executive Summary (General Audience)

Rooftop PV and other small-scale distributed energy resources (DER) now materially influence daytime system outcomes in the National Electricity Market. While individual systems are constrained by fixed export limits and emergency backstop mechanisms exist, **aggregate DER behaviour remains largely unmanaged**, resulting in over-generation, negative pricing, and increasing reliance on corrective interventions.

This paper proposes a **lightweight, resilient broadcast mechanism** by which **Australian Energy Market Operator** (or a delegated operational authority) could publish **coarse-grain generation posture signals** to guide aggregate DER output toward system-useful levels.

The mechanism:

- Is **advisory, not dispatch**
- Scales to millions of devices
- Is highly resistant to denial-of-service attacks
- Degrades gracefully during outages
- Requires no telemetry, registration, or per-device control

At its core, the proposal treats DER coordination as a **distributed consensus problem**, not a command-and-control problem, allowing independent devices to converge naturally on a shared system target.

Executive Summary (Market Operator Audience)

The proposed mechanism uses **DNS TXT records** as a resilient, globally cached metadata channel to publish **time-bracketed DER generation targets** at a coarse electrical zone level (e.g. distribution zone substations).

Each zone publishes a small, signed metadata record containing:

- A current target level
- A short forward vector (e.g. +5 / +10 / +15 minutes)
- Explicit validity and timing information

Targets are expressed as **dimensionless numeric bands (0–100)**, leaving interpretation to local device logic. Simple devices may apply threshold-based curtailment; advanced systems may infer trends, confidence, and apply smoothing or optimisation.

Records are designed to fit within a **single UDP DNS response**, enabling:

- Extreme scalability
- Heavy caching
- Natural load shedding
- Robust operation during partial network failure

The system does **not** replace existing controls (export limits, backstop), but adds a missing middle layer: a **system-wide shaping signal** that restores predictability and value to distributed PV while improving grid stability.

Purpose

The purpose of this mechanism is to provide a **missing middle layer** between market-level intent and device-level autonomy.

Specifically, it aims to:

> Provide **system-level guidance** to shape aggregate DER behaviour at operational timescales (minutes), without dispatching individual assets.
>
> Restore **predictability and value** to distributed PV by reducing simultaneity-driven over-generation and curtailment.
>
> Improve **grid stability outcomes** (voltage management, ramping, reserve reliance) using a signal that is tolerant of delay, error, and partial adoption.

Avoid the cost, fragility, and regulatory complexity of real-time APIs, telemetry ingestion, or mandatory participation schemes.

The mechanism is intentionally **advisory**, **coarse-grain**, and **fail-soft**, aligning with the physical and operational realities of large-scale DER.

The mechanism (what happens, end-to-end)

Publish: AEMO broadcasts coarse targets

AEMO publishes one TXT record per zone (plus optional region defaults), updated on minutes timescale.

Example zone name

```
z470201.vic.pv-target.aemo.example
```

Example TXT payload (compact, v=1)

```
"v=1 ts=2025-12-29T01:00Z s=300 d=900 n=42 5=38 10=31 15=22 k=K1 sig=..."
```

Meaning:

- `n` is "now", and `5/10/15` are forward points in minutes
- values 0–100 define how strongly devices should be *in export-permitted posture* (higher = more export encouraged; 50=balanced, lower = more export restrained)
- `ts/s/d` bracket timing so devices can tolerate caching variance

Resolve: Devices fetch their zone target via normal DNS

Devices use their normal recursive resolver path. Caching occurs at multiple layers automatically (local gateway, ISP resolver, enterprise resolver).

If a device cannot determine zone reliably, it may:

1. use a region default (e.g., `vic.pv-target…`)
2. or use an installer-provided zone hint
3. or use a lightweight zone discovery record (see Appendix D)

Act: Local control law (simple and robust)

Devices convert the 0–100 posture signal to a local action. Examples:

- **Simple inverter controller**: clamp export limit based on posture band
- **DC-side controller**: proportionally disconnect discretionary strings
- **Battery/EV/VPP**: shift charge/discharge/export subject to local constraints

The key point: **no device receives per-device commands**. All devices observe the same target and apply local logic. Convergence emerges statistically.

Fail safely: cache + timestamps + decay

If DNS becomes unavailable:

- devices hold last valid vector for a short period
- then decay toward a neutral posture or revert to existing export-limit behaviour
- no cliff failures; no oscillatory "panic mode"

Why DNS TXT is a strong fit

DNS is a globally deployed, heavily cached, replicated metadata system built to be resilient under heavy load and attack. TXT records are small, cacheable, and easy to validate. Unlike bespoke APIs, DNS naturally sheds load via caching, tolerates partial outages via stale cache, and avoids session state. Using DNSSEC (or equivalent signing) provides authenticity and integrity without introducing TLS certificate management or large client stacks. For a coarse, advisory signal, DNS delivers high availability and graceful degradation with minimal complexity.

What this is (and is not)

This is:

- An advisory, broadcast "shaping signal"
- Coarse-grain (minutes, zones)
- Vendor-neutral and scalable
- Compatible with existing export limits and backstop mechanisms

This is not:

- AEMO dispatch of household assets
- A telemetry system

- A per-device command channel
- A market settlement mechanism

Immediate outputs (what can be trialled quickly)

- Publish region defaults + a small set of pilot zones
- Implement a reference client library (ESP32 / inverter gateway)
- Demonstrate convergence behaviour with simulated DER populations
- Evaluate resolver behaviour (TTL vs timestamp) under normal and stressed conditions

Key Takeaway

This proposal reframes DER coordination as a **distributed consensus problem**, not a control problem.

By publishing a **coarse, time-bracketed generation posture signal** via a highly resilient broadcast medium, it becomes possible to shape aggregate DER behaviour at scale **without** dispatch, telemetry, or per-device control. DNS provides the necessary availability, scalability, and failure tolerance, while leaving interpretation and optimisation at the device level.

The result is a practical, low-risk pathway to:

- restore value to distributed PV,
- improve grid stability,
- and reduce reliance on blunt curtailment mechanisms,

using infrastructure that already exists and is proven at global scale.

Appendix A - DNS TXT Records as a Resilient Broadcast Medium

Purpose of this appendix

This appendix explains **why DNS TXT records are a suitable, secure, and resilient transport** for broadcasting coarse-grain DER generation posture signals, and how their operational characteristics align with the requirements of large-scale DER coordination.

It is intended to address common concerns such as:

- latency and propagation,
- denial-of-service risk,
- data integrity,
- cache behaviour,
- and suitability compared to conventional web APIs.

Why DNS is fundamentally different from APIs

Most control or coordination systems rely on **stateful, request–response services** (e.g. HTTPS APIs). These approaches scale poorly under extreme fan-out and are vulnerable to denial-of-service and cascading failures.

DNS operates on a different model:

Property	DNS
Communication model	Stateless query / cached response
Replication	Native, global, multi-layer
Load shedding	Automatic via caching
Failure mode	Stale data, not hard failure
Client complexity	Minimal
Attack surface	Narrow, well-understood

For a **coarse, advisory signal**, DNS's behaviour under stress is not a drawback-it is a design advantage.

TXT records as metadata containers

DNS TXT records are designed to carry **arbitrary, human-readable metadata**. They are widely used today for:

- email authentication (SPF, DKIM, DMARC),
- service discovery,
- policy publication,
- security assertions.

Key properties relevant to this proposal:

- TXT payloads are **small**
- They are **cacheable**
- They can be **signed and validated**
- They impose **no session state**

In this proposal, TXT records are used purely as **read-only broadcast metadata**, not as a control channel.

Caching behaviour and propagation

DNS caching occurs at multiple layers:

- device or gateway resolver
- ISP or enterprise recursive resolvers
- regional and global resolver infrastructure

Key implications:

- **Popular records are heavily cached**, dramatically reducing authoritative load.
- **Propagation delays are variable**, typically seconds to a few minutes.
- **Stale data is common and expected**.

This is acceptable-and desirable-for a signal that:

- operates on minute-scale buckets,
- is advisory rather than absolute,
- includes explicit timestamps and validity windows in the payload.

The proposal deliberately **does not rely on DNS TTL alone** to define signal validity. Instead:

- TTL is treated as a caching hint
- `ts` (timestamp) and `d` (duration) in the payload define semantic validity
- devices can detect and manage staleness locally

Single-packet UDP delivery and robustness

DNS is most robust when responses fit within a **single UDP packet**. This avoids:

- TCP fallback,
- increased latency,
- susceptibility to connection-based attacks.

The proposed TXT payloads are intentionally compact (≈100–200 bytes), allowing the full DNS response-including signatures-to fit comfortably within conservative UDP limits.

Benefits:

1. Works reliably through NATs and firewalls
2. Survives packet loss better than session-based protocols
3. Performs predictably under congestion
4. Public facing publisher resources (dedicated web hosting services) are not required

Denial-of-service resilience

DNS is one of the most attack-hardened systems in existence.

Key resilience properties:

- Anycasted authoritative servers
- Massive global caching
- Minimal per-query computation
- No per-client state

To suppress the signal, an attacker would need to:

- defeat authoritative DNS infrastructure,
- bypass or flush widespread caches,
- sustain attack long enough to prevent refresh,
- across many independent resolver networks.

By contrast, API-based systems often fail abruptly when endpoints are saturated.

Importantly, **the worst-case DNS failure mode is stale data**, not absence of data.

Integrity and authenticity (overview)

While confidentiality is not required for this signal, **authenticity and integrity are essential**.

DNS supports this via **DNSSEC**, which allows:

- cryptographic signing of records,
- verification by clients or gateways,
- detection of spoofed or tampered responses.

DNSSEC validation can be:

- performed directly by capable devices, or
- delegated to validating resolvers upstream.

This avoids:

- TLS certificate distribution,

- session negotiation,
- complex client stacks.

(Full details are covered in Appendix C.)

Failure modes and graceful degradation

DNS fails *softly* by design.

Typical failure scenarios and outcomes:

Scenario	**Outcome**
Authoritative unreachable	Cached data served
Cache expired, no response	Device holds last known value
Extended outage	Device decays toward neutral
Partial propagation	Statistical smoothing across population

At no point does failure induce:

- oscillation,
- cliff curtailment,
- or loss of local protections.

This behaviour is **ideal for grid-relevant signals**, where stability is more important than precision.

Why DNS is appropriate *for this specific problem*

This proposal does **not** use DNS as a control system. It uses DNS as a **broadcast bulletin board**.

DNS is appropriate here because the signal is:

- coarse-grain,
- tolerant of delay,
- tolerant of staleness,
- advisory,
- identical for large populations.

In short:

DNS is not good for precise control.
DNS is excellent for resilient, global coordination cues.

That is exactly the requirement this mechanism addresses.

Summary

DNS TXT records provide:

- extreme scalability,
- inherent load shedding,
- graceful failure behaviour,
- minimal attack surface,
- and mature operational tooling.

For broadcasting a **coarse DER generation posture signal**, DNS offers a level of resilience and simplicity that would be difficult-and expensive-to reproduce with bespoke service architectures.

Appendix B - Packet Sizing and Single-UDP-Response Constraints

Purpose of this appendix

This appendix quantifies the packet-size constraints that matter for robust DNS delivery and shows why the proposed zone-level TXT record format can be kept within a **single UDP DNS response** under conservative assumptions, including signing.

The design goal is not "maximum possible DNS size", but **boringly reliable delivery** across common resolvers, NATs, and firewalls-especially under congestion or attack.

DNS size constraints that matter in practice

Why "single UDP response" is preferred

DNS is most robust when the answer fits in one UDP response because it avoids:

- truncation and TCP fallback,
- connection establishment overhead,
- stateful attack surfaces,
- operational variability across networks.

Typical size envelopes

Practical, conservative envelopes used in network engineering:

- **Path MTU (common Ethernet)**: ~1500 bytes
- **IP + UDP headers**: ~28 bytes
- **Remaining for DNS message**: ~1472 bytes (best case)

However, "best case" is not the target. Middleboxes, tunnels, and conservative resolvers can reduce the reliable payload size. A widely used conservative target is:

- **Keep DNS responses ≤ ~1,200 bytes** where possible

This reduces fragmentation risk and improves reliability across diverse network paths.

What contributes to DNS response size

A DNS response includes:

1. **DNS header** (fixed)
2. **Question section** (QNAME + QTYPE + QCLASS)
3. **Answer section** (TXT RR)
4. Optional:
    - **Authority / Additional** sections
    - **EDNS(0) OPT record** (often present)
    - **DNSSEC records** (e.g., RRSIG, DNSKEY in some cases)

For this proposal, the typical response for a single TXT query is:

- One question
- One TXT answer
- Possibly one OPT record (EDNS0)
- If signed: one RRSIG corresponding to the TXT RRset

Conservative byte budgeting (rule-of-thumb)

To ensure high reliability, the proposal targets:

- **Total DNS response**: **≤ 1,000 bytes** (preferred)
- **Hard ceiling**: **≤ 1,200 bytes** (conservative)
- **TXT payload (application data)**: **≤ 200–300 bytes**

This provides room for:

- name overhead,
- record framing,
- EDNS0,
- and a typical DNSSEC signature.

Worked sizing example (zone TXT without DNSSEC)

Assume:

- QNAME: `z470201.vic.pv-target.aemo.example`
- TXT payload:
  `"v=1 ts=2025-12-29T01:00Z s=300 d=900 n=42 5=38 10=31 15=22"`

Approximate sizing components:

Question section

- QNAME encoding: label lengths + bytes (typically 35–60 bytes for names of this size)
- QTYPE + QCLASS: 4 bytes
  Estimated question total: ~45–70 bytes

Answer section (TXT RR)

- NAME (may be compressed via pointer): typically 2 bytes pointer
- TYPE + CLASS + TTL: 8 bytes
- RDLENGTH: 2 bytes
- TXT RDATA:
  - 1 byte length + N bytes text (text split into ≤255-byte chunks if needed)
- Additional RR overhead: small

Estimated answer total:

- framing ~12–14 bytes + payload (~90–140 bytes) ≈ **~110–160 bytes**

DNS header

- 12 bytes

Total (no DNSSEC)

- ~12 + (45–70) + (110–160) = **~170–240 bytes**

This is extremely safe and leaves substantial headroom.

DNSSEC overhead (single TXT RRset)

If DNSSEC is used, the response typically includes an **RRSIG** for the TXT RRset.

RRSIG size depends on algorithm and key size. In practice:

- ECDSA signatures tend to be smaller than RSA
- RSA signatures can be larger, but still commonly workable

A conservative working allowance for a single RRSIG in typical deployments is:

- **RRSIG record + signature**: **~250–450 bytes**

(Exact size varies; the proposal's sizing target simply reserves ample space.)

Revised total with DNSSEC

- baseline (no DNSSEC): ~170–240 bytes
  - RRSIG: ~250–450 bytes
  - EDNS0 OPT record (often): ~11–40 bytes

Total (signed): **~430–730 bytes** (typical conservative range)

This remains comfortably below the 1,000–1,200 byte target.

Why "one blob per region" is risky

A "regional blob" record that enumerates many zones grows linearly with the number of zones included. This creates three practical risks:

1. **UDP truncation and TCP fallback**
   - Large responses exceed safe UDP envelopes and trigger truncation.
2. **Middlebox drop / fragmentation**
   - Fragmented UDP DNS responses are more likely to be dropped.
3. **Cache inefficiency**

If any zone value changes, all clients must re-fetch a large payload.

For these reasons, the recommended pattern is:

1. **One DNS name per zone** (small records)
2. Optional **region default record** (also small)
3. Optional **discovery/mapping records** (kept small)

This provides horizontal scaling and retains single-packet reliability.

Payload design guidance (to stay safely inside UDP)

To maintain safe single-packet delivery even when signed:

Recommended constraints

1. TXT payload: **≤ 200 bytes** preferred, **≤ 300 bytes** acceptable
2. Avoid embedding long certificates or verbose JSON
3. Use compact key/value fields
4. Limit forward vector length (e.g., now + 3 points)

Example compact payload

```
"v=1 ts=20251229T0100Z s=300 d=900 n=42 5=38 10=31 15=22"
```

If signatures must be represented inside TXT (not recommended if using DNSSEC), keep them short and use key IDs. However, the preferred approach is to rely on DNSSEC rather than embedding signatures in the TXT data.

Summary

The proposed zone-level TXT records:

1. are small enough to fit well within conservative single-UDP DNS response limits,
2. remain robust even with DNSSEC signatures included,
3. avoid TCP fallback and fragmentation risks,
4. and scale cleanly to thousands of zones nationally.

The key design choice enabling this is **record granularity**:

small independent records per zone, rather than aggregated blobs.

Appendix C - Authenticity and Integrity (DNSSEC Model)

Purpose of this appendix

This appendix describes how **authenticity and integrity** are provided for the broadcast posture signal, why **confidentiality is not required**, and how the security model avoids the operational and scaling risks of session-based approaches (e.g. HTTPS APIs).

The intent is to show that the proposal achieves **cryptographic trust with minimal complexity**, and that its failure modes are safe and bounded.

Security objectives and threat model

Security objectives

- **Authenticity**: devices can verify the signal originated from the authorised publisher.
- **Integrity**: devices can detect tampering or corruption in transit.
- **Availability**: the signal remains obtainable under load or attack.
- **Fail-soft behaviour**: loss or delay does not cause abrupt or unsafe actions.

Explicitly out of scope

- Confidentiality (the signal is not sensitive).
- Non-repudiation (the signal is advisory and time-bounded).
- Per-device authentication or authorisation.

Primary threats considered

- DNS spoofing / cache poisoning
- On-path modification
- Replay of stale data
- Denial-of-service against publication endpoints

Why DNSSEC is the appropriate integrity mechanism

DNS natively supports **DNS Security Extensions (DNSSEC)**, which provide cryptographic signing of DNS records.

Key properties:

- Signatures are attached to DNS data (RRsets), not sessions.
- Validation can occur at resolvers or endpoints.
- There is no per-client state.
- Trust is rooted in the DNS hierarchy.

DNSSEC is standardised and maintained by the **Internet Engineering Task Force**, and is widely deployed for security-critical uses.

DNSSEC basics (at a high level)

In DNSSEC:

- The authoritative zone signs its records using a private key.
- A corresponding public key is published in DNS (DNSKEY).
- A chain of trust links the zone to a known trust anchor.
- Clients or validating resolvers verify signatures (RRSIGs).

For this proposal:

- Each **TXT RRset** (per zone) is signed.
- Devices accept the record **only if signature validation succeeds**.
- Invalid or unverifiable data is treated as unavailable (fail-soft).

Validation models

Two validation models are supported without changing the protocol.

Resolver-side validation (preferred for simple devices)

- ISP, gateway, or enterprise resolvers perform DNSSEC validation.
- Devices receive already-validated answers.
- Devices rely on the resolver's AD (Authenticated Data) bit or equivalent trust indicator.

This minimises device complexity and is already common practice.

Device-side validation (optional for advanced systems)

- Devices perform DNSSEC validation locally.
- Suitable for gateways, aggregators, or controllers with sufficient resources.
- Provides independence from resolver trust.

Both models are compatible and interoperable.

Key management and rotation

Key management principles

- Keys are long-lived and rotated infrequently (months to years).
- Overlapping validity periods are used during rotation.
- Zone signing keys (ZSK) and key signing keys (KSK) may be separated following standard practice.

Operational simplicity

- No device certificates
- No TLS stacks
- No session negotiation
- No per-vendor trust onboarding

Rotation events are handled entirely within DNS infrastructure and do not require device reconfiguration.

Replay and staleness handling

DNSSEC guarantees **authenticity**, not freshness. This is by design.

To handle replay and staleness safely, the proposal includes:

- Explicit **timestamp (`ts`)** in the TXT payload
- Explicit **duration (`d`)** defining semantic validity
- Device-side logic to:
    - reject values outside a reasonable time window,
    - hold last known good values briefly,
    - decay toward neutral posture thereafter.

This ensures:

- replayed old records are detected,
- stale cache entries do not cause unsafe behaviour,
- availability is prioritised over precision.

Denial-of-service considerations

DNSSEC does **not** prevent denial-of-service; availability is achieved through DNS's inherent properties:

- caching at multiple layers,
- anycast authoritative infrastructure,
- small, stateless responses.

Crucially:

- DNSSEC validation failure does **not** induce unsafe behaviour.
- Devices simply revert to last known good state and decay.

This contrasts with API-based systems where authentication failure often equals total service loss.

Why TLS / HTTPS is not used

TLS-secured APIs introduce:

- session state,
- certificate lifecycle management,
- per-client scaling pressure,
- higher susceptibility to volumetric and application-layer attacks.

For a **broadcast, identical-for-many, advisory signal**, DNSSEC provides:

- stronger availability characteristics,
- simpler trust semantics,
- fewer failure modes.

The proposal deliberately avoids mechanisms that create hard dependencies on continuous connectivity.

What happens if validation fails

If a device cannot validate a record:

- it treats the signal as unavailable,
- continues operating on the last valid posture,
- applies decay logic toward a normal unconstrained situation

At no point does invalid data cause:

- forced curtailment,
- sudden export increase,
- or override of existing protections.

This behaviour is consistent with the fail-soft design goals.

Summary

The authenticity and integrity model:

- uses **standard DNSSEC**, not bespoke cryptography,
- avoids session-based trust mechanisms,
- tolerates delay and partial failure,
- and aligns with DNS operational norms.

By combining DNSSEC with explicit payload timestamps and conservative device behaviour, the proposal achieves **cryptographic trust without sacrificing availability or scalability**.

Appendix D - Zone Definition, Discovery, and Fallback

Purpose of this appendix

This appendix defines **what a "zone" is**, why administrative boundaries (e.g. postcodes or similar metric) are insufficient for electrical coordination, and how devices can **reliably determine an appropriate zone** using lightweight discovery and robust fallback-without requiring registration, telemetry, or precise network topology knowledge.

What a "zone" represents (and what it does not)

Zone definition (for this proposal)
A *zone* is a **coarse electrical area** with broadly similar distribution-network constraints at operational timescales (minutes). The natural unit is a **distribution zone substation (ZSS)** or an equivalent DNSP-defined aggregation.

Zones are intended to:

- Align with voltage/thermal constraints that matter for DER shaping
- Be **stable over time**
- Be independent of customer identity or market registration

Zones are **not** intended to:

- Identify individual feeders or LV transformers
- Track customer locations precisely
- Represent administrative, postal, or municipal boundaries

Why postcodes are insufficient as zones

Australian postcodes are designed for **mail routing**, not electrical topology. Known issues include:

- Large geographic coverage (often >100 km)
- Non-contiguous service areas
- Satellite towns sharing a postcode but served by different substations
- Boundaries that change for administrative reasons unrelated to the grid

Using postcodes *directly* as electrical zones would therefore:

- Create false aggregation
- Dilute constraint relevance
- Reduce the usefulness of the signal

However, postcodes remain **useful as a discovery hint** (see Appendix D Zone Discovery).

Recommended zone granularity and scale

Recommended granularity

- Primary: **Distribution zone substation (ZSS)** level
- Optional aggregation where DNSPs prefer (e.g. small ZSS groups)

Expected scale

- NEM-wide: approximately **2,000–3,000 zones**
- Typical device resolves **one** zone
- Region-level defaults provide coverage where zone resolution is uncertain

This scale balances:

- Electrical relevance
- DNS operational safety
- Simplicity for device implementers

Zone identifiers and naming

Zones are identified by **opaque, stable IDs** assigned by the publisher (e.g. **Australian Energy Market Operator** or a delegated DNSP authority).

Key properties of zone IDs

- No semantic meaning implied by the number
- Stable across years
- Independent of postcodes or addresses

Example DNS name

`z470201.qld.pv-target.aemo.example`
Where:

- `z470201` is a zone identifier
- `qld` provides optional regional scoping
- `pv-target` denotes the signal class

The DNS name itself is the *binding* between zone and signal.

Zone discovery (optional, lightweight)

Devices with limited information may not know their zone a priori. To avoid hardcoding topology, the proposal allows **optional discovery records**.

Discovery pattern

`<postcode>.zones.<region>.pv-target.aemo.example`
Example

`4702.zones.qld.pv-target.aemo.example`
Example TXT response

```
"v=1 z=z470201:Gindie,z470205:Anakie,z470209:Comet"
```
Notes:

- The list is **small** (typically ≤10 zones)
- It represents *candidate zones*, not certainty
- TTL can be short; payload remains tiny

Zone selection by devices

When multiple candidate zones are returned, devices select one using **local heuristics**, such as:

- Installer configuration (preferred)
- PC-Suburb Name match
- Resolver locality (e.g. DNS resolver geography)
- Historical stability (stickiness)

The design goal is **consistency**, not perfect accuracy. Occasional misclassification is acceptable because:

- Signals are coarse and advisory
- Errors are damped statistically
- Region defaults exist as a safety net

Region-level defaults (mandatory fallback)

To ensure safe operation under uncertainty, a **region-level default record** is always published.

Example

```
vic.pv-target.aemo.example
```
Devices use the region default when:

- Zone discovery fails
- Zone records are unavailable
- Validation fails
- Configuration is incomplete

Region defaults provide:

- A known-safe posture
- Very high cache hit rates
- Predictable behaviour during outages

Behaviour under incorrect or stale zone mapping

The system is explicitly tolerant of imperfect mapping.

If a device:

- selects a neighbouring zone,
- or falls back to region default,

then:

- its behaviour remains bounded,
- its contribution is statistically smoothed across the population,
- no device-level hazard is introduced.

This tolerance is a **deliberate design choice** aligned with coarse-grain coordination.

Governance and evolution of zones

Zones are expected to:

- Be defined conservatively at rollout
- Change rarely
- Be versioned if changes are unavoidable

If zones must change:

- Old zone names may be retained temporarily
- Devices can be guided via discovery records
- Region defaults continue to provide coverage

This avoids forcing firmware updates or installer action.

Summary

Zone definition and discovery are designed to:

- Reflect electrical reality without excessive granularity
- Avoid reliance on administrative boundaries
- Minimise device complexity
- Tolerate uncertainty and partial adoption

By combining **stable zone identifiers**, **optional discovery**, and **region-level defaults**, the proposal ensures that every device can obtain a meaningful signal without requiring precise topology knowledge or central registration.

Appendix E - Reference Device Behaviour and Failure Modes

Purpose of this appendix

This appendix defines a **reference client behaviour** for devices consuming the broadcast posture signal. It is not prescriptive; rather, it establishes **safe defaults**, **interoperable semantics**, and **bounded failure modes** so that diverse devices (inverters, DC-side controllers, batteries, EVSEs, gateways, VPP platforms) can participate without creating instability.

The emphasis is on:

- predictable convergence,
- graceful degradation,
- and compatibility with existing protections.

Design goals for device behaviour

Device behaviour should:

1. **Be monotonic and bounded**
   No abrupt increases in export; no oscillatory responses.
2. **Prefer availability over precision**
   Stale-but-authentic data is preferable to no data.
3. **Remain subordinate to local protections**
   Existing export limits, voltage protections, and emergency backstop controls always take precedence.
4. **Avoid tight coupling to freshness**
   Devices must tolerate variable propagation and caching behaviour.

Interpreting the posture signal (0–100)

The posture value is **dimensionless** and intentionally abstract. Devices map it to actions using local policy.

Recommended interpretation bands (illustrative)

Posture (0–100)	Suggested interpretation
80–100	Export encouraged (subject to local limits)
60–80	Normal export
40–60	Mild restraint
20–40	Strong restraint

Posture (0–100)	Suggested interpretation
0–20	Minimum export / discretionary load shifting

Notes:

- Devices may use **continuous curves** rather than bands.
- Devices should implement **rate limits** (e.g., % change per minute).
- No device is expected to hit an exact target; **population statistics** do the work.

Using the forward vector (now, +5, +10, +15)

The forward points allow devices to:

- anticipate ramps,
- avoid step changes,
- schedule discretionary actions.

Guidance

- Treat `n` ("now") as the immediate posture.
- Use forward points as *trend indicators*, not commitments.
- If some forward points are missing, interpolate conservatively.
- Do not extrapolate beyond the declared duration (`d`).

Reference state machine

A simple, robust state machine is recommended.

States

1. **VALID** - authenticated signal within validity window
2. **STALE** - signal authenticated but outside preferred freshness window
3. **UNAVAILABLE** - no authenticated signal available

Transitions and actions

- **VALID**
  - Apply posture using local control law
  - Respect local limits and protections
  - Update internal timers
- **STALE**
  - Hold last applied posture
  - Gradually decay toward default behaviour
  - Do not increase against a downward trend
- **UNAVAILABLE**
  - Continue toward default posture
  - Revert to static configuration if decay completes

This can be implemented with minimal state and no clocks beyond basic timekeeping.

Behaviour on loss of signal (graceful disengagement)

When the broadcast posture signal becomes unavailable, delayed, or stale, devices should disengage from the signal **gradually and predictably**, returning to their normal operating behaviour.

The guiding principle is:

Loss of signal should result in a slow return to unconstrained local operation, not an abrupt change in export.

Reference behaviour

When a valid signal is no longer available, a device should:

1. **Hold its current operating point initially**
   Avoid immediate changes while signal availability is uncertain.
2. **Gradually reduce the influence of the broadcast signal**
   Over time, the device progressively relaxes any export shaping that was being applied due to the signal.
3. **Transition smoothly to unconstrained operation**
   In the absence of a valid signal, the device returns to its normal configuration, subject only to existing export limits, protections, and market arrangements (e.g. FiT eligibility).
4. **Avoid abrupt or synchronised step changes**
   All transitions should be smooth and rate-limited to prevent aggregate surges.

Design guidance

- The disengagement rate is implementation-specific (e.g. linear or exponential).
- Typical time constants may range from several minutes to tens of minutes.
- Faster disengagement may be appropriate where participation incentives are explicitly time-limited.
- Existing protections and regulatory constraints always take precedence.

Rationale

The broadcast posture signal is **voluntary, advisory, and temporary**. It is intended to shape aggregate behaviour when available, not to impose ongoing constraints when unavailable.

A gradual return to unconstrained operation ensures that:

- participants are not penalised for signal loss,
- operator faults or outages do not suppress legitimate export,
- and system operators have time to respond using established mechanisms.

Interaction with existing controls

The broadcast posture signal is **advisory** and must not bypass:

- Inverter export limits
- Voltage and frequency protections
- DNSP-imposed dynamic limits
- Emergency backstop mechanisms

Precedence order (highest to lowest)

1. Safety protections (hard limits)
2. Regulatory / DNSP constraints
3. Local configuration
4. Broadcast posture signal

This ordering ensures zero regression in safety or compliance.

Partial adoption and mixed populations

The mechanism assumes **partial and uneven adoption**.

Properties under partial adoption:

- Non-participating devices are unaffected
- Participating devices still converge statistically
- Increased adoption improves signal effectiveness but is not required for safety

There is no "minimum participation threshold" for safe operation.

Behaviour under incorrect or misclassified zone

If a device consumes a neighbouring or incorrect zone signal:

- The posture remains bounded
- Errors are damped by population averaging
- Region defaults provide a safe baseline

No device-level hazard arises from occasional misclassification.

Replay, validation failure, and staleness

Devices should:

- Reject records that fail authenticity validation
- Detect replay via timestamp (`ts`) and duration (`d`)
- Treat invalid data as **UNAVAILABLE**

At no point should invalid data:

- force immediate curtailment,

- cause sudden export increases,
- or disable existing protections.

Implementation simplicity

A compliant implementation can be achieved with:

- a small DNS client,
- basic timekeeping,
- a simple state machine,
- and a local control mapping.

No persistent connectivity, registration, or telemetry is required.

Summary

The reference behaviour defined here ensures that:

- devices act conservatively under uncertainty,
- loss of signal produces gradual, predictable outcomes,
- existing safety and regulatory mechanisms remain authoritative,
- and aggregate behaviour converges smoothly without central control.

Appendix F - Standards, Policy, and Appropriateness of DNS TXT Usage

Purpose of this appendix

This appendix addresses whether the proposed use of **DNS TXT records as a broadcast metadata channel** is consistent with DNS standards, operational guidance, and typical policy or contractual constraints.

It is intended to pre-empt concerns that the approach:

- misuses DNS,
- violates standards or best practice,
- or introduces unacceptable policy or governance risk.

DNS TXT records: intended purpose and scope

DNS TXT records are explicitly defined as **generic text containers** with no prescribed semantics. Their purpose is to allow publication of **arbitrary metadata** without requiring protocol changes.

In practice, TXT records are widely used for:

- policy publication (e.g. email sender policy),
- security signalling and verification,
- service capability advertisement,
- application coordination metadata.

There is **no restriction in DNS standards** on the semantic meaning of TXT payloads, provided they remain:

- small,
- cacheable,
- non-interactive,
- and tolerant of staleness.

The proposed mechanism aligns with this intended flexibility.

Relevant standards and guidance

The **Internet Engineering Task Force** does not prohibit application-level signalling via DNS TXT records. Instead, guidance focuses on **how DNS should be used safely**.

Key principles derived from IETF guidance and operational best practice include:

1. **DNS is not a real-time control channel**
   DNS responses may be cached, delayed, or served stale.

2. **TTL is a caching hint, not a correctness guarantee**
   Applications must tolerate variability in resolver behaviour.
3. **Small, infrequently changing records scale best**
   Large or rapidly changing payloads are discouraged.
4. **Identical answers for many clients are preferred**
   Personalised or per-client responses reduce cache effectiveness.
5. **Integrity must be protected where correctness matters**
   DNSSEC or equivalent mechanisms should be used if tampering is a concern.

This proposal conforms to these principles when update frequency is interpreted in the DNS-operational sense: records change at a cadence that preserves cache effectiveness, avoids per-client variation, and tolerates resolver-level staleness.

In DNS guidance, "infrequent change" does not imply static records, but rather change rates that allow effective caching and avoid per-client or per-request variation. Minute-scale updates to small, identical TXT records are common in modern DNS practice and remain well within operational norms, particularly where applications are explicitly tolerant of propagation delay and staleness.

Alignment of the proposal with DNS best practice

DNS Best-Practice Consideration	**Proposal Alignment**
Small payloads	TXT payloads ~100–200 bytes
Low update frequency	Minute-scale cadence
Cache-friendly	One record per zone, identical answers
Staleness tolerance	Explicit timestamps and validity windows
Integrity protection	DNSSEC (or equivalent signing)
Stateless operation	No sessions, no client state

The proposal uses DNS **as a broadcast metadata bulletin**, not as a command, control, or transactional system.

No contractual or policy prohibition

There is **no known contractual restriction** imposed by:

- domain registries,
- DNS operators,
- ISPs,
- or cloud DNS providers

that limits TXT records to specific application classes (e.g. email-only use).

TXT records are routinely used for:

- internal control metadata,
- security assertions,
- orchestration hints,
- verification and discovery mechanisms.

As long as usage remains:

- non-abusive,
- cache-friendly,
- and within normal query volumes,

it is considered operationally acceptable.

Regulatory and governance considerations

From a governance perspective, this approach is **less intrusive** than many alternatives:

- No inbound control of devices
- No telemetry ingestion
- No per-device addressing
- No market participation requirement
- No dependency on continuous connectivity

The signal is best characterised as **public operational guidance**, analogous to:

- forecasts,
- advisories,
- or published constraint outlooks,

rather than dispatch or instruction. This distinction materially reduces regulatory and legal risk.

F.7 What would *not* be appropriate DNS usage

For clarity, this proposal explicitly avoids practices that would raise standards or policy objections:

- Using DNS as a high-frequency control loop
- Encoding large datasets or bulk state
- Personalising responses per device
- Requiring strict freshness or delivery guarantees
- Treating DNS as a command channel

Avoiding these patterns is central to the design.

Precedent and relative risk

Many existing DNS TXT use cases are **more demanding** than this proposal, including:

- security-critical email authentication,
- automated certificate issuance,
- identity and zero-trust verification,
- abuse and reputation signalling.

Compared to these, a **coarse, advisory DER posture signal** represents:

- lower security impact,
- lower correctness sensitivity,
- higher tolerance of delay and error.

Summary

There is **no standards-based, contractual, or policy barrier** to using DNS TXT records for the proposed purpose.

When used within established operational constraints-small payloads, low update rates, cache tolerance, and integrity protection-DNS TXT records are a **well-accepted mechanism for resilient metadata publication**.

This proposal adheres to those constraints and deliberately avoids known anti-patterns, making DNS a technically and governance-appropriate choice for broadcasting coarse-grain DER coordination signals.

Appendix G - Future Uses and Extensibility of the Architecture

Purpose of this appendix

This appendix outlines **potential future uses** of the proposed broadcast posture architecture that are **explicitly out of scope for the initial implementation**, but naturally enabled by the design.

These uses are described to:

- demonstrate architectural headroom,
- show that the mechanism is not single-purpose,
- and avoid future redesign pressure.

Nothing in this appendix is required for initial deployment.

Design principle: extensibility without dependency

A core strength of the architecture is that it:

- publishes **relative, advisory signals**,
- decouples signal semantics from device implementation,
- and allows devices to opt into additional behaviours voluntarily.

Future enhancements must preserve:

- identical baseline signals for the general fleet,
- fail-soft behaviour,
- no implied capacity obligation,
- no per-device dispatch or registration.

Differential responsiveness as a future capability

One potential extension is support for **differential responsiveness**, where certain participants can **track posture changes more rapidly** than the general fleet due to site characteristics or control sophistication.

Examples include:

- high-capacity DC-coupled PV systems,
- systems with controllable batteries or discretionary loads,
- sites using advanced DC-side control (e.g. string-level shedding),
- locations where faster shaping has outsized network benefit.

Importantly, this concerns **how quickly** a participant responds - not **how much** power it supplies.

Responsiveness as a dynamics hint, not a capacity signal

Under this architecture, enhanced responsiveness would be expressed only as a **recommended response dynamics parameter**, not as:

- an MW or MWh obligation,
- a guaranteed response,
- or a dispatch instruction.

For example, a zone posture record may include optional fields such as:

```
tau   = recommended response time constant (general fleet)
tau_f = recommended response time constant (fast-capable devices)
```

or equivalently:

```
resp   = 1 (baseline responsiveness)
resp_f = 2 (twice the baseline response rate)
```

Devices that are not capable of fast response simply ignore the additional parameter.

Device-side opt-in and local governance

Participation in enhanced responsiveness is:

- **voluntary**,
- **device-local**, and
- determined by installer configuration or device capability.

A device may choose to treat itself as "fast-capable" only if it can:

- respond smoothly without instability,
- respect existing export limits and protections,
- avoid oscillatory behaviour.

No device is expected to self-identify or register centrally.

Operational value for system operators

If adopted in future, differential responsiveness could provide system operators with:

- faster aggregate convergence during sharp ramps,
- additional stabilisation leverage near constrained zones,
- improved handling of forecast error without invoking emergency mechanisms.

This would act as a **pre-emptive stabilisation tool**, complementing (but not replacing) existing services such as export limits, constraint equations, or ancillary services.

Relationship to incentives and markets

Any incentives associated with enhanced responsiveness would be:

- defined outside the signalling protocol,
- optional,
- and subject to separate regulatory and market design processes.

The broadcast posture mechanism itself remains:

- non-market,
- non-dispatch,
- and advisory.

This separation avoids embedding market semantics into the protocol.

Backward compatibility and safety

Critically:

- baseline devices remain fully functional,
- absence of enhanced parameters has no adverse effect,
- and failure or removal of future extensions degrades gracefully.

The architecture therefore supports incremental capability growth without forcing coordinated upgrades.

Summary

The broadcast posture architecture is deliberately designed to support **future extensions** without compromising its core principles.

Differential responsiveness is one example of how:

- the same signal can influence different classes of devices differently,
- without introducing dispatch semantics,
- capacity obligations,
- or additional DNS complexity.

By keeping such capabilities optional and advisory, the architecture remains robust, fair, and adaptable as DER capability evolves.