APIS v2.1 — Reference Proof: Herman

First conforming APIS v2.1 agent passport. This is the evidence artifact for the published spec: not a diagram, a receipt. Every claim below is reproducible from public artifacts.

Summary

Field	Value
DID	did:passport:syndicate:herman
Realm	syndicate
Principal	AetherPro Technologies LLC (ORGANIZATION, US)
Tier	2.5 (DNS-anchored; resolved by attestation_resolver.py on an OVH KVM node)
Issuer key	apis-realm-issuer-2026-05-09 (ES256), authorized by a root-signed delegation
Passport signature	ES256, verified against the published issuer JWKS at mint time (2026-06-24)
Anchor	herman._apis.syndicateai.co (DNS TXT)
DNS provider	**Namecheap** — registrar-managed zone, no Cloudflare, no DNS API
Key fingerprint	sha256/hBN9-eo9H_igHiwalPt1Ft7kPUA4YRNGvXLIIcnY7iU
Mandate	scope: govcon:scan, web:fetch, telegram:respond, model:inference, report:write
model_scope	allow {grm-2.6-plus, qwen3.6-27b, qwen3.6-35b-a3b}; deny {claude-fable-5, claude-mythos-5}; policy deny-overrides

The chain proven

```
Alliance root (Ed25519, air-gapped)
  -> signs -> issuer delegation
    -> authorizes -> realm issuer key apis-realm-issuer-2026-05-09
(ES256)
```

```
      -> signs -> agent passport  did:passport:syndicate:herman
        -> bound to -> DNS TXT anchor  herman._apis.syndicateai.co
```

All anchors are published at `https://passportalliance.org/.well-known/`: `alliance-root.jwk`, `apis-issuer-jwks.json`, `apis-issuer-delegation.json`, `apis-issuer-delegation-payload.json`.

Reproduce it yourself

```
# 1. published chain validates (root -> issuer delegation)
python3 - <<'PY'
import json,base64,urllib.request
from cryptography.hazmat.primitives.asymmetric.ed25519 import
Ed25519PublicKey
B="https://passportalliance.org/.well-known"; f=lambda
u:urllib.request.urlopen(u,timeout=20).read()
b=lambda s:base64.urlsafe_b64decode(s+'='*(-len(s)%4))
root=json.loads(f(B+"/alliance-root.jwk"));
deleg=json.loads(f(B+"/apis-issuer-delegation.json"))
payload=f(B+"/apis-issuer-delegation-payload.json")
Ed25519PublicKey.from_public_bytes(b(root['x'])).verify(b(deleg['signa
ture']),payload)
print("delegation signature VALID; authorizes",deleg['payload']
['issuer_key_id'],"for realm",deleg['payload']['realm'])
PY

# 2. DNS anchor resolves and matches the passport fingerprint
dig +short TXT herman._apis.syndicateai.co
```

Verification output (re-run from public artifacts)

```
APIS v2.1 — Herman reference proof, public re-verification
run at: 2026-06-29T17:12:37.452684Z

[1] alliance root JWK (anchor):     kid apis-root-v1.0-202602 alg
EdDSA
[2] issuer JWKS:                    kid apis-realm-issuer-2026-05-09
alg ES256
[3] delegation authorizes issuer:   apis-realm-issuer-2026-05-09 for
realm syndicate
[4] root->issuer delegation signature over published payload bytes:
VALID
[5] DNS anchor herman._apis.syndicateai.co: v=APIS1; t=2.5; k=ec-p256;
p=sha256/hBN9-eo9H_igHiwalPt1Ft7kPUA4YRNGvXLIIcnY7iU
[6] anchor p= matches passport fingerprint: True

RESULT: published chain (root -> issuer delegation -> issuer key) is
valid,
        and Herman's DNS anchor publishes the matching key
fingerprint.
        (Passport JWT ES256 signature was verified against this JWKS
at mint time, 2026-06-24, on the issuer host.)
```

Honest scope

- Tier 2.5 security reduces to DNS-account security; DNSSEC protects the record in transit, not the legitimacy of its contents (see DELTA §3, §3A).
- The passport JWT signature was verified against the published JWKS at mint time on the issuer host. The full bearer JWT is intentionally **not** published — its signature is the credential. The public proof above does not require it.
- This demonstrates conformance of one mint. It is a reference proof, not a claim that every implementation or tier is productized.