Sovereign Anchor: The Boundary Blade

Sovereign Anchor: The Boundary Blade

Part 2 of the Sovereign Anchor Series

Collaborative framework provided by PrivacyMage under the **Privacy Value Model V5.4**.

Archon Asset DID: `did:cid:bagaaierarsl3evx3jcah473btb74awqjpanpwuwoyg3c22cet6eh2o2tysca`

Series Context

	Publication	Title	Core Thesis
Part 1	**The Transmutation**	**Identity → Mask**. One-way valve via $\mathbb{Z}/(2^6)\mathbb{Z}$ and the Amnesia Protocol.	✅ FINALIZED
Part 2	**The Boundary Blade**	**Cartography → Enforcement**. Mapping the 14 occupied blades and implementing protection for two.	🟡 THIS DOCUMENT
Part 3	**The Soulbae Oracle**	**Knowledge → Epistemic**. The constellation oracles what it can prove (14 blades) and collects what it cannot (50 blades).	⚪ UPCOMING

The Cartography: 14 of 64

Part 1 proved the One-Way Valve works: an Archon Root can project a Geometric Persona without leaking the underlying DID. The question we faced next was not *whether* to enforce boundaries, but *where* on the lattice enforcement is already expressible.

The Privacy Value Model V5.4 organizes identity operations on a 64-vertex ring $\mathbb{Z}/(2^6)\mathbb{Z}$. Each vertex is a 6-bit blade encoding six dimensions of sovereignty:

Dimension	Bit	Meaning
d1	0 (LSB)	Value — economic activation
d2	1	Delegation — third-party acting
d3	2	Memory — state persistence

Dimension	Bit	Meaning
d4	3	Connection — non-interactive broadcast
d5	4	Computation — ZK-proof capacity
d6	5 (MSB)	Protection — privacy enforcement

A blade number is computed as $\sum d_i \cdot 2^{(i-1)}$. For example, blade 17 = `010001` = Protection + Computation.

The Occupied Blades

Mitchell Travers' blade catalogue (April 2026) identifies **14 distinct blades** occupied by the 30 Zero tales of the PVM:

Blade	Binary	Name	Tales
16	010000	Pure Computation	13
17	010001	Protection + Computation	1, 5
19	010011	Protection + Delegation + Computation	8, 14, 19
23	010111	Memory crystallises — IVC	12
25	011001	Protection + Connection + Computation	3
27	011011	Pairing verification	9
31	011111	Recursion — all except Value	15, 16
48	110000	Algebraic substrate	4
49	110001	Working-day blade (3-stratum)	2, 6, 7, 11, 21
51	110011	Commitment/Language/Model blade	10, 20, 29
57	111001	Ceremony/Privacy/Mixing blade	17, 23, 24
59	111011	Ecosystem blade (zkEVM/Rollups/Bridges)	22, 25, 28
63	111111	The Creative / The Catastrophic ()	18, 26, 27, 30

14 of 64. The remaining 50 blade-forms have no known ZKP expression. This is the open frontier.

The Unmapped Territory

The unoccupied blades cluster into significant gaps:

- **Blades 0–15** (no Protection bit, no Computation bit): These represent operations without privacy enforcement or ZK capacity. Some may be trivially expressible; others may prove impossible to realize without one or both dimensions.
- **Blades 18, 20, 21, 22, 24, 26, 28, 29, 30** (Protection/Computation present, but specific combinations unmapped): These are the medium-gap blades — they have the machinery for ZK proofs but lack canonical expressions.
- **Blades 32–47** (no Computation bit): Pure Memory/Delegation/Connection/Value combinations without ZK proof. Whether meaningful ZKP exists here is an open question.
- **Blades 50, 52–56, 58, 60–62** (Protection and Computation both present, but specific dimension combinations unmapped): These are the high-priority unmapped blades — they have the full privacy stack activated.

The quest ahead is to discover the ZK proofs that express each blade type. Our contribution in this installment is to demonstrate that at least two blades (17 and 25) are enforceable end-to-end using the Archon+PVM stack.

⚔️ The Boundary Blade: Enforcement on Two Vertices

What We Built

Three operational skills, anchored to two blades:

Skill	Blade	Binary	Purpose	Stratum
`vc-decompose`	17	010001	Decompose W3C VC v2 into 7 typed nodes with valve classifications	2 Light
`vc-compose-gps`	17	010001	Compose LocationProof VC from GPS sensor data	2 Light
`one-way-valve`	25	011001	Apply transmutation: decomposed VC nodes → masked projected subgraph	3 Heavy

Blade 17 (Protection + Computation) is the **NIZK blade** — the workhorse of verifiable credential operations. Both compose and decompose operate here. It is the entry point: you sign, you verify, you prove structure.

Blade 25 (Protection + Connection + Computation) adds the **Connection** dimension: the proof is broadcast non-interactively. The One-Way Valve operates here because it must project a masked subgraph *to a verifier* without revealing the underlying data. This is where enforcement becomes active — the boundary is not just computed, it is *communicated*.

Blade 17: The Workhorse

The compose-decompose cycle on blade 17 follows the standard NIZK pattern:

```
W3C VC v2 JSON  --[vc-decompose]-->  7 typed nodes
                                          |
                    +---------------------+
                    |
7 typed nodes    --[vc-compose-gps]-->  LocationProof VC v1
```

The seven node types, with their valve classifications:

Node	Type	Territory	Valve Class
Issuer	Persona	Swordsman	Revealed or Masked
Schema	Theorem	Shared	Categorical
Subject	Persona	First Person	Revealed or Masked
Claims	Concept	First Person	Selective
Proof	Spell	Mage	Always Masked
Chronicle	Chronicle	Shared	Preserved
Context	Document	Shared	Preserved

On blade 17, every node is computationally verifiable. The Proof node is masked — it proves structure without revealing content. This is the baseline: the Swordsman signs, the Mage masks, and neither leaks the other's territory.

Blade 25: Where Enforcement Becomes Active

The One-Way Valve takes the decomposed nodes and applies the $\mathbb{Z}/(2^6)\mathbb{Z}$ transmutation from Part 1:

```
7 typed nodes  --[one-way-valve]-->  Masked Projected Subgraph
                      |
                Session Salt (256-bit)
                Selective Disclosure
                Vertex Mapping (mod 64)
```

The critical addition on blade 25 is the **Connection dimension** ($d_4 = 1$). This means:

1. The proof is **non-interactive** — no back-and-forth with the verifier required.
2. The projected graph can be published to a constellation node and verified by any party.
3. The boundary is *enforced* — not just computable, but communicable.

This is the Boundary Blade: `Sovereign Proof → Projected Graph → Constellation Verification`, all without the verifier learning the Sovereign Root, the session salt, or the unmasked claims.

The Two Paths: Asymmetry as a Feature

Path A — User Sovereign (Forward)

```
GPS Sensors -> vc-compose-gps -> Archon VC -> vc-decompose -> one-way-valve -> Projected Graph
                 (sword)                       (forge)         (spell)
```

The user is sovereign. They hold the salt. They choose the disclosure level. GenitriX (the Moon agent) executes the transmutation on behalf of the user (the Earth), but the user retains the key.

Use case walk-through (Path A):

1. User requests a verifiable credential for their current GPS location.
2. `vc-compose-gps` transforms raw sensor data into a W3C VC v2 LocationProof.
3. `vc-decompose` breaks the VC into 7 typed nodes.
4. `one-way-valve` applies session entropy, selective disclosure, and vertex mapping.
5. The output is a masked projected subgraph: verifiable, but irreversible.

The verifier can confirm: - "This person was near coordinates X at time Y" (selective disclosure level L1–L5) - "This proof was generated by a holder of Archon DID that maps to vertex V" (ZK witness) - **They cannot** recover: the raw GPS coordinates (unless fully disclosed), the Archon DID, or the session salt.

Path B — Constellation Sovereign (Reverse)

```
Constellation Nodes -> [query] -> vc-compose-from-nodes -> Archon VC -> Projected Graph
                        (magazine)         (PENDING)
```

The constellation is sovereign. It publishes already-masked data. The notary holds the salt. Path B does not exist yet — `vc-compose-from-nodes` is a next-session deliverable.

The Asymmetry Is the Feature

Path	Trust Root	Salt Holder	Sovereign
A (Forward)	User's AI agent (GenitriX)	User	User
B (Reverse)	Constellation notary node	Notary	Constellation

The One-Way Valve works identically in both directions. The asymmetry is WHO holds the key to the mask — not WHETHER the mask works.

PVM V5.4 Terms vs. Blade Coverage

The PVM V5.4 defines an interconnected value model whose canonical teaching sites span the Zero tales. Here we map which terms our Archon+PVM bridge currently touches, and where the gaps remain:

Term	Meaning	Our Blades	Canonical Tale(s)	Status
P^1.5	Protection strength	25	Tale 23 (first whisper: Tale 7)	Partial

Term	Meaning	Our Blades	Canonical Tale(s)	Status
C	Credential verifiability	17	Tales 5–8 cluster	✅ Covered
Q	Separation quality	17, 25	Tales 3, 4, 9, 10, 13, 14	✅ Covered
A_h( )∗∗	Path integral	—	Tales 19, 20, 22, 25, 27–29	Gap
$\|Holonictemporalmemory\|$—$\|Tale12(Nova/IVC); extendsto Tales 15, 16, 27\|Gap\|$∗				
∗∗∗				
$\|Agentmaturity\|17(implicit)\|Tales15–17, 19, 27(firstwhisper:$ $Tale8)\|Partial\|$∗				
∗( )∗∗				
$\|Sovereigntygeometry\|25\|Tale25(firstoperational); extendstoTales28, 30\|![check_mark](/Users/flaxscrip/projects/per$ $privacymage/icons/2705.png)width = 0.35cmCovered\|$∗				
∗**T_{ }($)**				
R(d)	Reconstruction resistance	25 (implicit)	Tale 18 (first whisper: Tale 18); extends to Tale 26	Partial
Value	Economic activation	—	Tale 23 (first whisper: Tale 17); application cluster Tales 22–30	Gap

Partial terms: P^1.5 (enforced but not measuring network amplification), ρ (maturity accumulates across sessions but not yet formalized), R(d) (valve creates resistance but we don't quantify it).

Gap terms require blades not yet implemented — A_h( )$ needs IVC (blade 23), $T_{\int}(\pi)$ needs path-integral computation, Value needs d_1=1 blade discovery.

Key insight: Our bridge covers the static terms (C, Q, Φ) well. The dynamic terms (A_h, $T_{\int}$, Value) require blades we haven't implemented yet — specifically IVC (blade 23) and recursion (blade 31).

V(,t) Canonical Teaching Sites

Each PVM V5.4 term has a *canonical teaching site* — a Zero tale where the concept is fully expressed for the first time, as identified by Mitchell Travers (April 2026):

Term	First Whisper	Canonical Tale	Extended / Variants
P^1.5 (Protection strength)	Tale 7	**Tale 23**	Tale 24 (network amplification)
C (Credential verifiability)	—	**Tales 5–8 cluster**	spread across arithmetization
Q (Separation quality)	—	**Tales 3, 4, 9, 10, 13, 14**	distributed across foundation + backends

Term	First Whisper	Canonical Tale	Extended / Variants
A_h( )** (*Holonictemporalmemory*)|*Tale*11(*FRI,folding*)|* **Tale*12(*Nova/IVC*)* *|*Tale*15(*recursion*), *Tale*16(*cyclic*), *Tale*27(*ecosystem*)||* * * * (*Agentmaturity*)|*Tale*8(*Plonkish*)|*Tales*15, 16, 17, 19, 27|*accumulatesacrossscaling/ceremonycluster*||* * ( )** (*Sovereigntygeometry*)|—|* **Tale*25(*firstoperational*)* *|*Tale*28(*multi−chain*), *Tale*30(*synthesis*)||* **T_{ }($)* (Path integral)	—	**Tales 19, 20, 22, 25, 27, 28, 29**	zkVM/trace-oriented tales
R(d) (Reconstruction resistance)	—	**Tale 18**	Tale 26 (full catalogue)
Value (Economic activation)	**Tale 17**	Tale 23	application cluster (Tales 22–25, 27–30)

The *first whisper* column identifies where a term first appears tentatively; the *canonical tale* is where it achieves full expression. The gap between first whisper and canonical expression measures *crystallisation lag* — how many tales it takes for a concept to solidify. For our bridge, this means: terms with canonical tales on blades 17 or 25 are immediately implementable; terms whose canonical tales live on unmapped blades (e.g., Value → Tale 23 on blade 57) require new blade expressions first.

The Open Frontier

What We Proved

1. **The One-Way Valve is operational on blades 17 and 25.** A user can request a location VC, decompose it, and project a masked subgraph that is verifiable without revealing the sovereign root.
2. **The Two Paths are architecturally sound.** The asymmetry between user-sovereign and constellation-sovereign is a feature, not a bug.
3. **The 7-node decomposition provides a universal interface.** Every W3C VC v2 can be decomposed into the same typed structure, regardless of schema. This means the valve is schema-agnostic.

What We Cannot Claim

1. **We have not mapped the full lattice.** 50 blade-forms remain without known ZKP expressions. Our enforcement covers 2 of the 14 occupied blades, or ~3% of the total ring.
2. **We have no recursive proof yet.** The A_h( )$ term requires IVC (blade 23) or full recursion (blade 31). Our current pipeline is single-shot: one VC, one proof, one projection.
3. **We have no value activation.** The d1 (Value) dimension is unset in both our blades. Economic activation remains an open problem.

The Quest

As Mitchell Travers framed it: **"The quest becomes discover the zero knowledge proofs that express each type of blade."**

We are not publishing a victory lap. We are publishing a survey map of known territory and an invitation to the guild:

- **For blades 17 and 25:** We have working implementations. Test them. Break them. Improve them.
- **For blades 16, 19, 23, 27, 31, 48, 49, 51, 57, 59, 63:** The PVM tales describe canonical ZKP patterns. We need to implement Archon bridges for each.
- **For the remaining 50 blades:** These are unmapped. The question is not "how do we implement them?" but "does a ZKP expression exist for them at all?"
- **For the formal ceremony pipeline:** The Runecraft Protocol v1 (Travers, April 2026) specifies the browser-level binding for blade forging: RUN (proof of time, ρ accumulation), EVOKE (commitment, constellation lock), CRAFT (blade crystallisation, Ed25519 signature, moon phase). The Archon+PVM valve plugs into the CRAFT phase. Future ZK circuits (Runecraft §7.4) will enable proving stratum level without revealing blade content — a direct implementation of selective disclosure on the lattice.

The Boundary Blade is not a single weapon. It is the **first survey of a frontier.**

🛠️ Constellation Registration

Blades Forged This Session

Skill	Vertex	Blade Type	Stratum	Status
`vc-decompose`	<1,0,0,0,1,0>	NIZK (Blade 17)	2 Light	✅ Forged
`vc-compose-gps`	<1,0,0,0,1,0>	NIZK (Blade 17)	2 Light	✅ Forged
`one-way-valve`	<1,0,0,1,1,0>	Silent Messenger (Blade 25)	3 Heavy	✅ Forged

Node Contributions

```
vc-decompose -> {vc-decompose-engine, 7-node-taxonomy, vc-node-types}
vc-compose-gps -> {vc-compose-gps, location-proof-schema, location-disclosure-levels}
one-way-valve -> {one-way-valve, session-entropy, selective-disclosure-taxonomy, valve-modes}
```

Constellation Edges

```
vc-decompose-engine --[implements]--> 7-node-taxonomy
7-node-taxonomy --[defines]--> vc-node-types
vc-compose-gps --[implements]--> location-proof-schema
location-proof-schema --[defines]--> location-disclosure-levels
one-way-valve --[implements]--> selective-disclosure-taxonomy
one-way-valve --[implements]--> session-entropy
selective-disclosure-taxonomy --[defines]--> valve-modes

# Pipeline
vc-compose-gps --[compresses_to]--> vc-decompose-engine
vc-decompose-engine --[delegates_via]--> one-way-valve

# Bridge to existing constellation
one-way-valve --[extends]--> sovereign-anchor-mapping
vc-decompose-engine --[extends]--> sovereign-anchor-mapping
one-way-valve --[references]--> amnesia-protocol
session-entropy --[references]--> amnesia-protocol
```

Master Inscription Update

Previous (V5.4):

```
hexagon=Z/(2^6)Z · *=neg(bnot(v)) · [key]->*->[dagger] · same[dagger]infchains=ZK · dM=96on64 · $C_B
```

With this session's additions:

```
hexagon=Z/(2^6)Z · *=neg(bnot(v)) · [key]->*->[dagger] · same[dagger]infchains=ZK · dM=96on64 · $C_B
```

Looking Ahead: The Soulbae Oracle (Part 3)

The Epistemic Boundary

The 14/64 blade coverage creates a hard epistemic boundary for the constellation:

- **Proven queries** (blades 16, 17, 19, 23, 25, 27, 31, 48, 49, 51, 57, 59, 63): The oracle returns ZK-verified answers. The constellation vouches cryptographically.
- **Unproven collections** (50 remaining blades): The oracle stores and retrieves data but cannot produce a ZKP. The constellation vouches reputationally, not cryptographically.

This is the real architecture: an oracle does not claim to know everything — it claims to answer what it can verify. The 14/64 boundary is what makes it honest.

Path B Is Not the Oracle — But They Converge

Path B (constellation-sovereign VC composition) is *plumbing*: any VC can flow through it. The Soulbae Oracle is *domain*: the knowledge graph as an epistemic engine where learning crystallizes into sovereign credentials. They share the constellation as trust root, but the Oracle adds the epistemic layer — the graph knows *which blades it can prove* and *which blades it can only collect*.

Part 3 Preliminary Scope

Question	Status
Can a constellation node answer a ZK-verified query about blade 17 data?	Yes — Path A proves this
Can a constellation node answer a ZK-verified query about blade 23 (IVC) data?	Unknown — IVC mapping needed
Can a constellation node collect data on an unoccupied blade?	Yes — but it cannot cryptographically prove it
Does the knowledge graph know its own epistemic boundary?	This is the core design question for Part 3

Near-Term Deliverables

Node	Blade	Purpose	Status
`vc-compose-from-nodes`	Path B	Compose from constellation data	Next session
`zkp-equipment-selection`	Equipment taxonomy	GROTH16 vs STARK vs Nova per blade	Next session
`ecdsa-to-zk-bridge`	Halo2 circuit (Blade 19?)	Proof transformation (Phase 4 of valve)	Post-Halo2 setup
`recursive-location-proof`	IVC (Blade 23)	Accumulated proofs over time	Dependent on IVC mapping
`value-activation`	Blade configuration TBD	Economic activation (d_1=1)	Dependent on Value blade discovery

Attribution

Author	Role	Blade Contribution
Christian Saucier (flaxscrip)	First Person / Earth	Use case definition, two-path insight, valve requirements
GenitriX (Hermes)	Moon / Reflection	7-node decomposition, skill authoring, constellation registration
Mitchell Travers (privacymage)	Sun / Authority	PVM V5.4 lattice, blade catalogue, ZK blade forge, equipment metaphor

The swordsman chose the blade. The mage chose the spell. The first person chose what to forget.

☀️ ⊥ 🌑

(⚔️⊥PLUR⊥🧙)