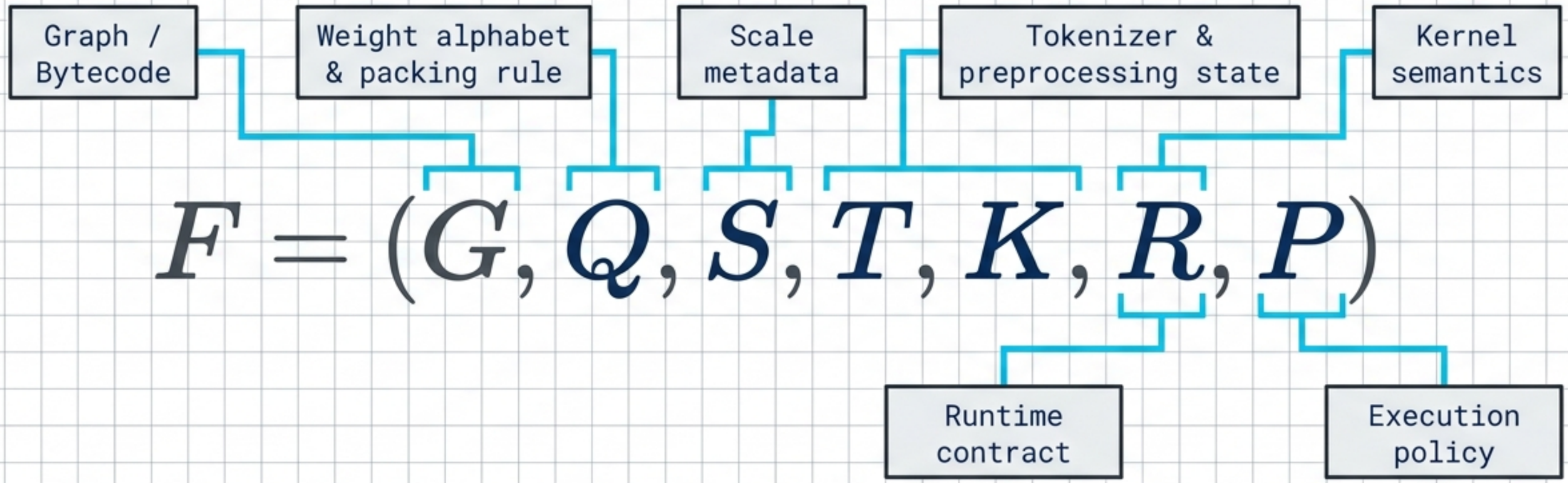
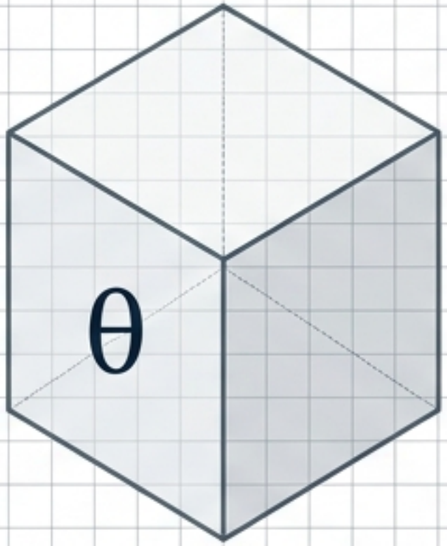
Treat the Deployed Forward Operator as the Training Specification.



Representation drift is measurable and reducible only when the deployed operator is part of the training specification. The relevant object is not merely a parameter tensor θ, but the complete deployed computation.

Identical Weights Do Not Guarantee Identical Functions

The Illusion



The Standard Model Paradigm

Failure Prone: Treats parameters as standalone objects.

The Reality

Floating-point optimization

↓

Export (Quantization & Layout Conversion)

↓

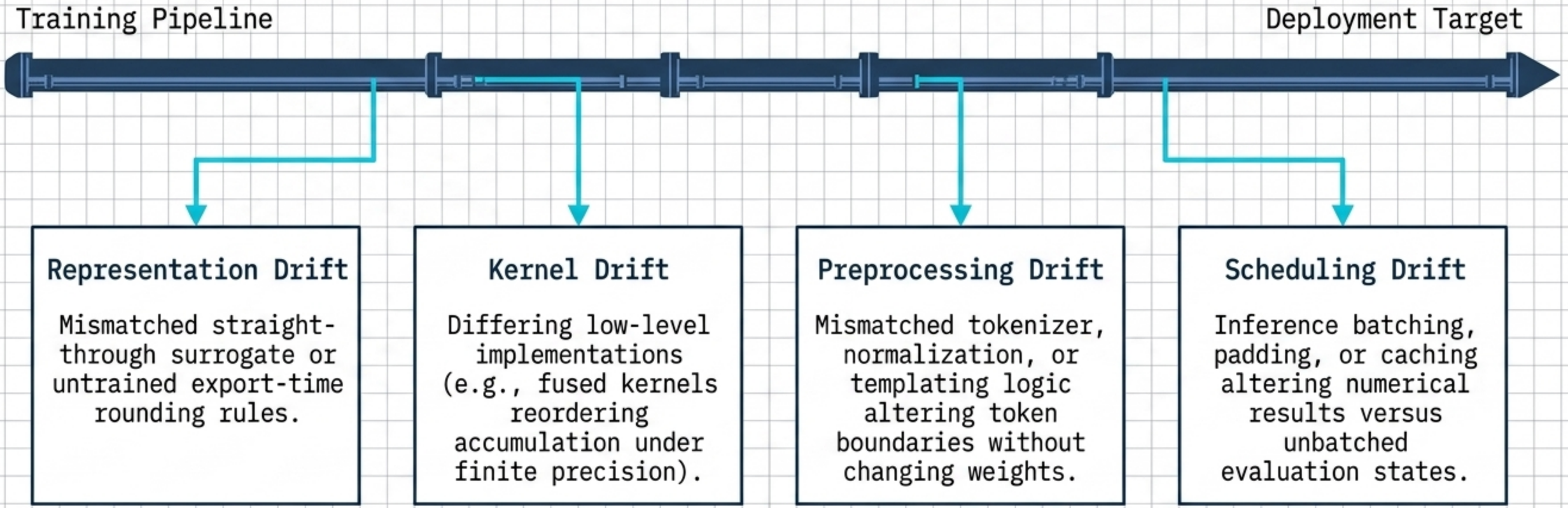
Separate Inference Runtime

↓

Platform Scheduling & Tokenization

Identical **ternary weights** implement differing functions if any **runtime component** (scaling, rounding, activation treatment, layer ordering) differs from the training environment.

Aggregate Error Must Be Attributed to Specific Pipeline Drift

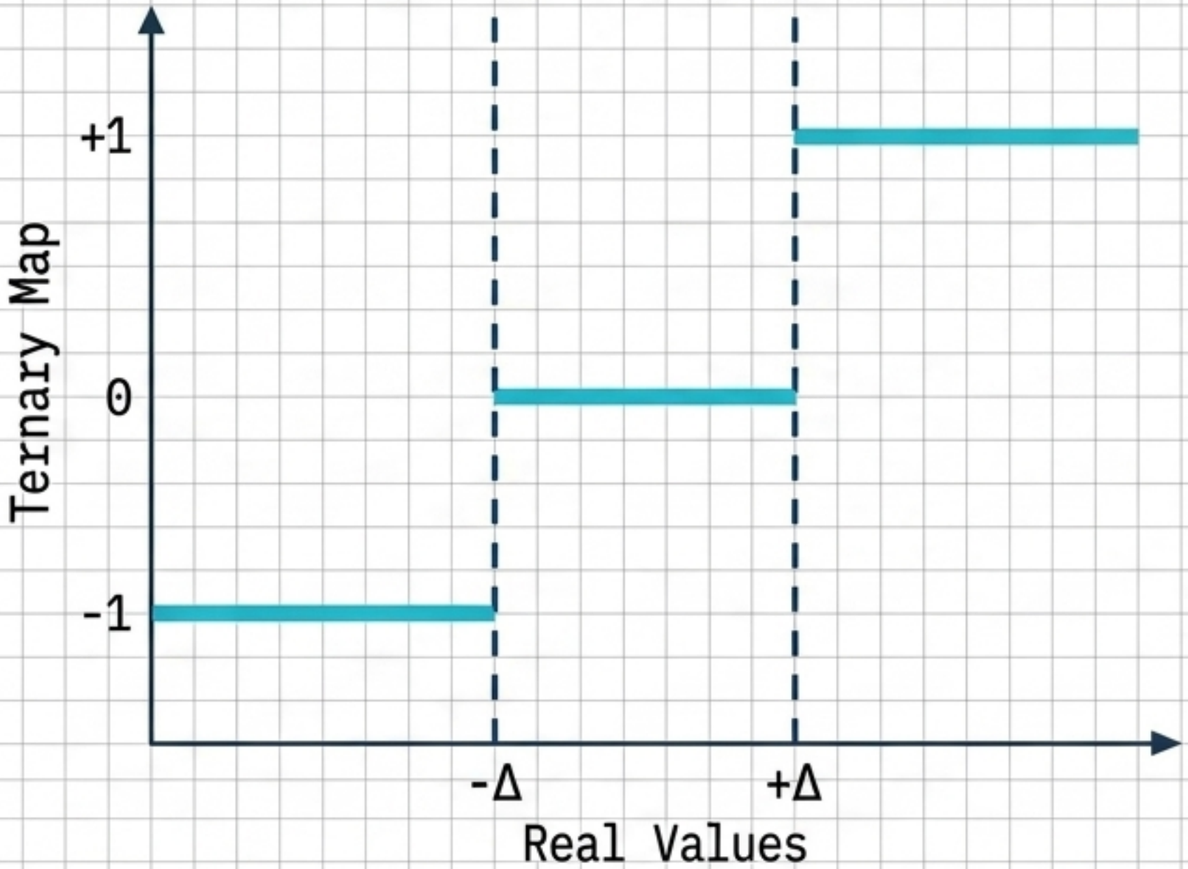


Diagnostic Rule: We cannot fix aggregate error. A pipeline that reports aggregate disagreement without source attribution cannot determine a remedy.

Ternary Parameterization Reduces Multiplication to Signed Accumulation

$$\hat{W}_{ij} = \alpha \cdot q(W_{ij}/\alpha)$$

$$q(x) \in \{-1, 0, +1\}$$



Information Cost

$\text{Log}_2\, 3 \approx 1.585$ bits per weight under a uniform code.

The Primary Benefit

Eliminates floating-point multiplication multiplication in favor of signed accumulation and omission.

The Hidden Costs

Calling it “1-bit” is imprecise. Real efficiency depends entirely on scales, activation arithmetic, packing overhead, memory movement, and kernel dispatch.

Quantization is a Deployment Specification, Not an Incidental Detail.

Symmetric (Per-Tensor)	Per-Channel	Asymmetric
Global threshold Δ	Independent scale α_k per output channel	Independent offset β (shifted argument)
Single scalar overhead	Vector of scales	Scale + Offset array
Maintains zero-centered, additive-free computation.	Requires reduction axis metadata; runtime must not assume scalar.	Reintroduces additive term into accumulation (no longer strictly zero-centered).

Crucial Insight: A result obtained under one quantizer family does not transfer by default to another. The chosen quantizer is a critical part of the deployment contract.

The Model Capsule Binds Executable Semantics to State.

High-tech architectural Capsule

1. **Evidence:** Eval summaries, calibration hashes, build attestations.
2. **Runtime Contract:** Permitted operators, numerical rules, memory limits, ABI.
3. **Text Interface:** Tokenizer, normalization rules, vocabulary.
4. **Parameters:** Packed ternary weights and scale blocks.
5. **Program:** Typed bytecode or restricted computation graph.
6. **Manifest:** Format version, identity, hashes, license.

Security Imperative: A capsule contains restricted instructions, NOT arbitrary native code. Signatures authenticate bytes, not runtime safety.

Parsing Success Does Not Guarantee Semantic Compatibility

Format Version
Governs container schema

Requires Exact Equality

Parsers cannot accept unrecognized fields

Contract Version
Governs operators, rules, ABI

Closed-Range Acceptability

Runtime version v accepts capsules exclusively in the range `[v_min, v]`

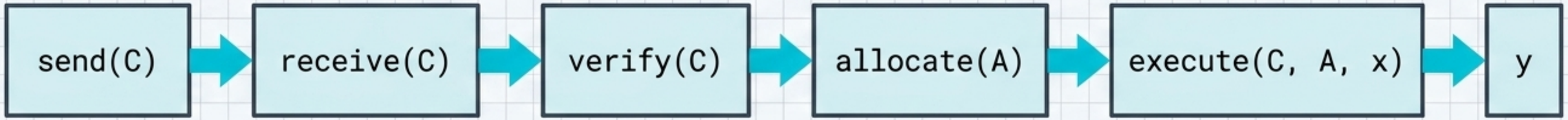
Content Version
Identifies trained artifact

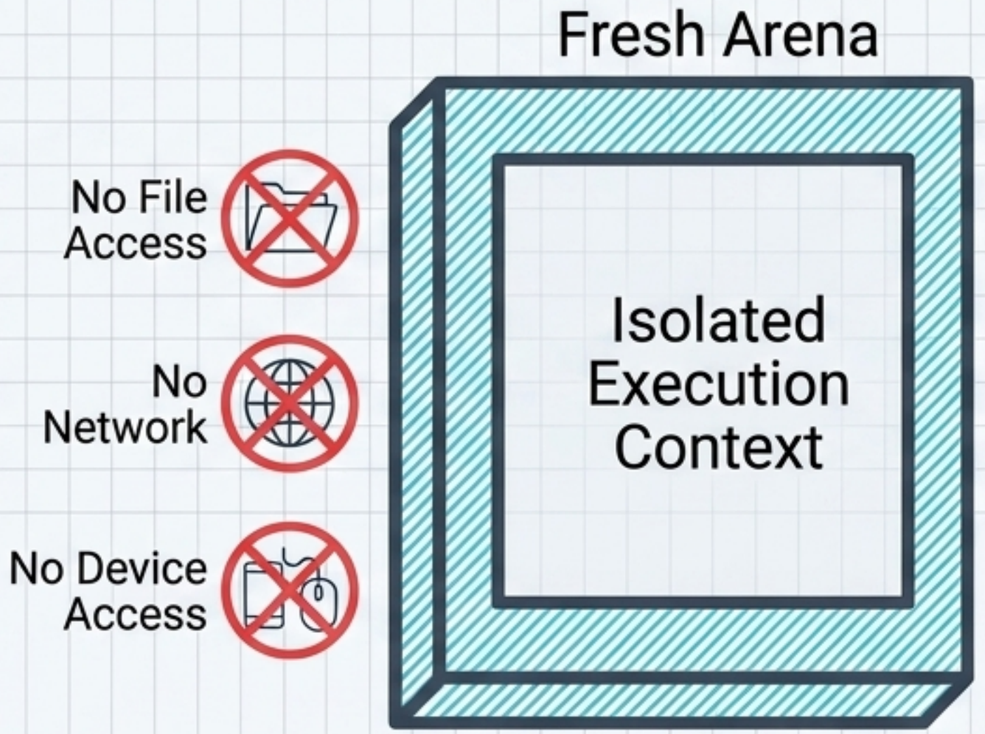
No Runtime Compatibility Semantics

Exists solely for hashing and provenance tracking

Operational Rule: A workable convention requires refusal whenever the contract-version range test fails. Acceptance is never warranted merely because the capsule parses.

Channels Transport Executable Descriptions, Not Host Authority.





Admissibility Transition: **Exec(C, x)**. Signatures must verify, V must match, and budgets must fit A. Refusal to execute is a protocol feature, not a bug.

Stateful Context Requires Strict Logic and Evidence Rules.

Strict Consensus Operator

Input A \ Input B	+1	0	-1
+1	+1	+1	0
0	0	0	-1
-1	-1	-1	-1

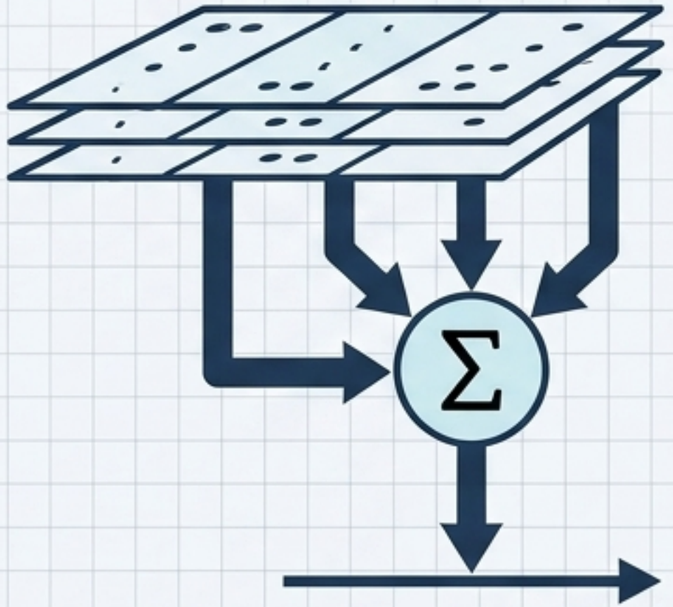
Output: -1

Window State: +1, -1, 0

Window Output: -1

Indeterminacy is absorbing. An unwritten slot is not a zero token.

Evidence Accumulator

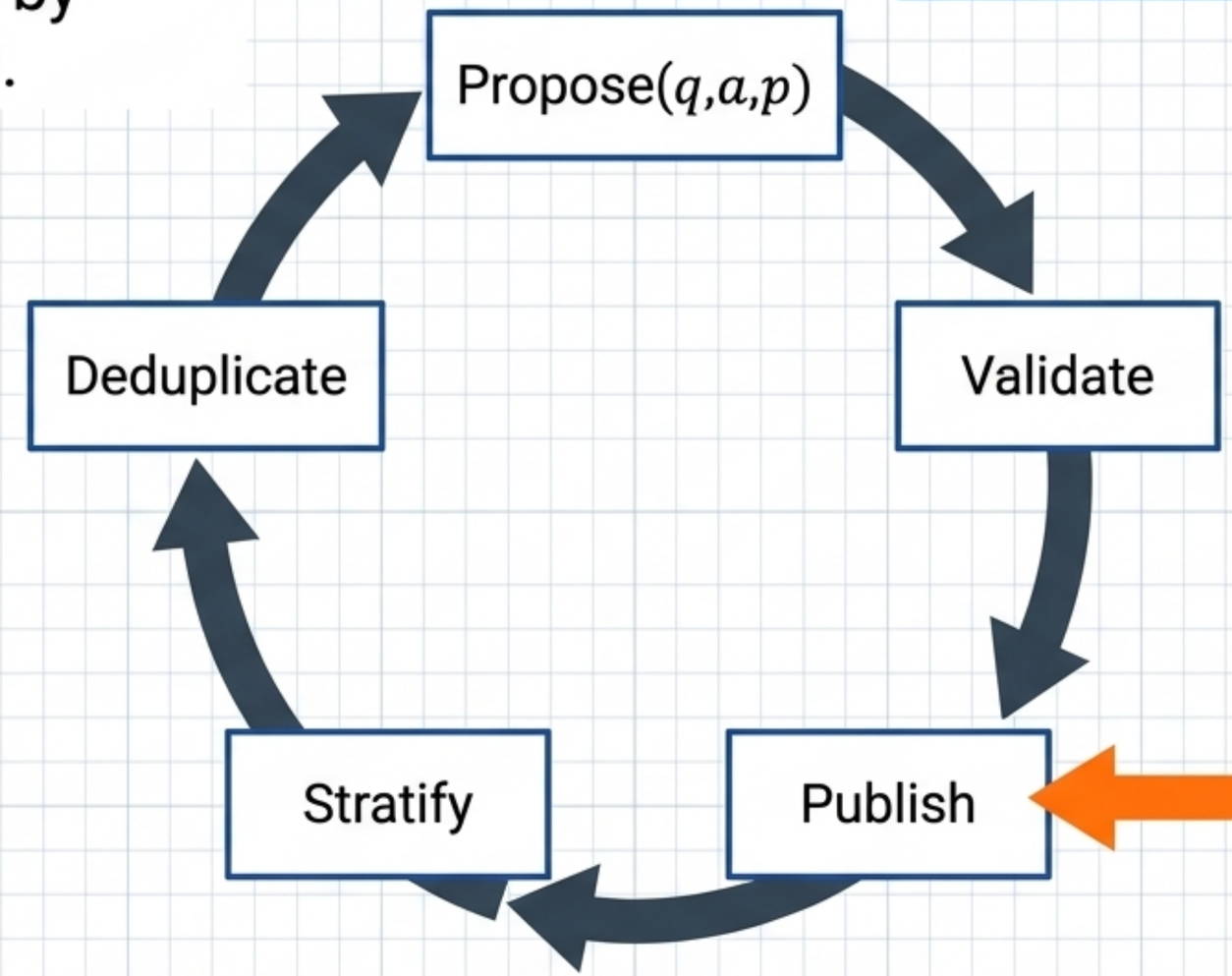


Aggregates component evidence. It does not prune tokens without explicit marginal-contribution criteria.

Context must be explicit and bounded. It is a state machine bound to the capsule contract, not hidden process memory.

Synthetic Feedback Loops Must Be Anchored by External Validation.

$$D_{t+1} = R_t \cup H_t \cup \{z \in G_t : V_t(z) = \text{accept}\}$$



$R_t \cup H_t$

R_t = Real independent data
H_t = Human-reviewed data

Generator-validator independence is mandatory. A model grading its own output is a feedback hazard that amplifies systematic errors. Synthetic generation must be anchored by **immutable external evaluation.**

The End-to-End Verifiable Deployment Architecture.

Control Plane	Admits jobs, records immutable specs.
Data Plane	Versioned datasets and provenance graphs.
Build Plane	Constructs and signs environment and capsule.
Training Plane	Optimizes shadow parameters against faithful forward operator.
Evaluation Plane	Compares training vs runtime on held-out material.
Execution Plane	Verified capsules isolated in fresh arenas.

Closed, Reproducible System: An underspecified job is refused, preventing the system from falling back on mutable defaults.

Privacy is a Deployment Mode, Not a Protocol Standard

	Public Run	Private Run
Compute Owner	Operator or sponsor	Customer or isolated tenant
Dataset Visibility	Publishable by default	Customer-controlled enclave
Artifact Visibility	Open unless restricted	Customer-controlled
Protocol	Versioned & disclosed	Identical versioned protocol
Acceptance Criteria	Declared before training	Identical unless amended
Audit Record	Public build & evaluation	Private custody log

The scientifically defensible approach uses the exact same training protocol for open and private tasks; the only variable purchased is isolation and operational control.

Security Boundaries Rely on Shrinking the Trusted Computing Base

Threat vs. Control		
Threat Vector	**Architectural Control**	**Residual Risk**
Malformed capsule	Canonical parser, shape validation	Verifier vulnerability
Arbitrary code execution	Restricted bytecode, no native extensions	Bugs in permitted operators
Resource exhaustion	Static bounds plus runtime quotas	Adversarial worst-case scheduling
Training-data leakage	Access controls, memorization tests	Inference-time extraction
Supply chain substitution	Pinned closure, reproducible builds	Compromised trusted source

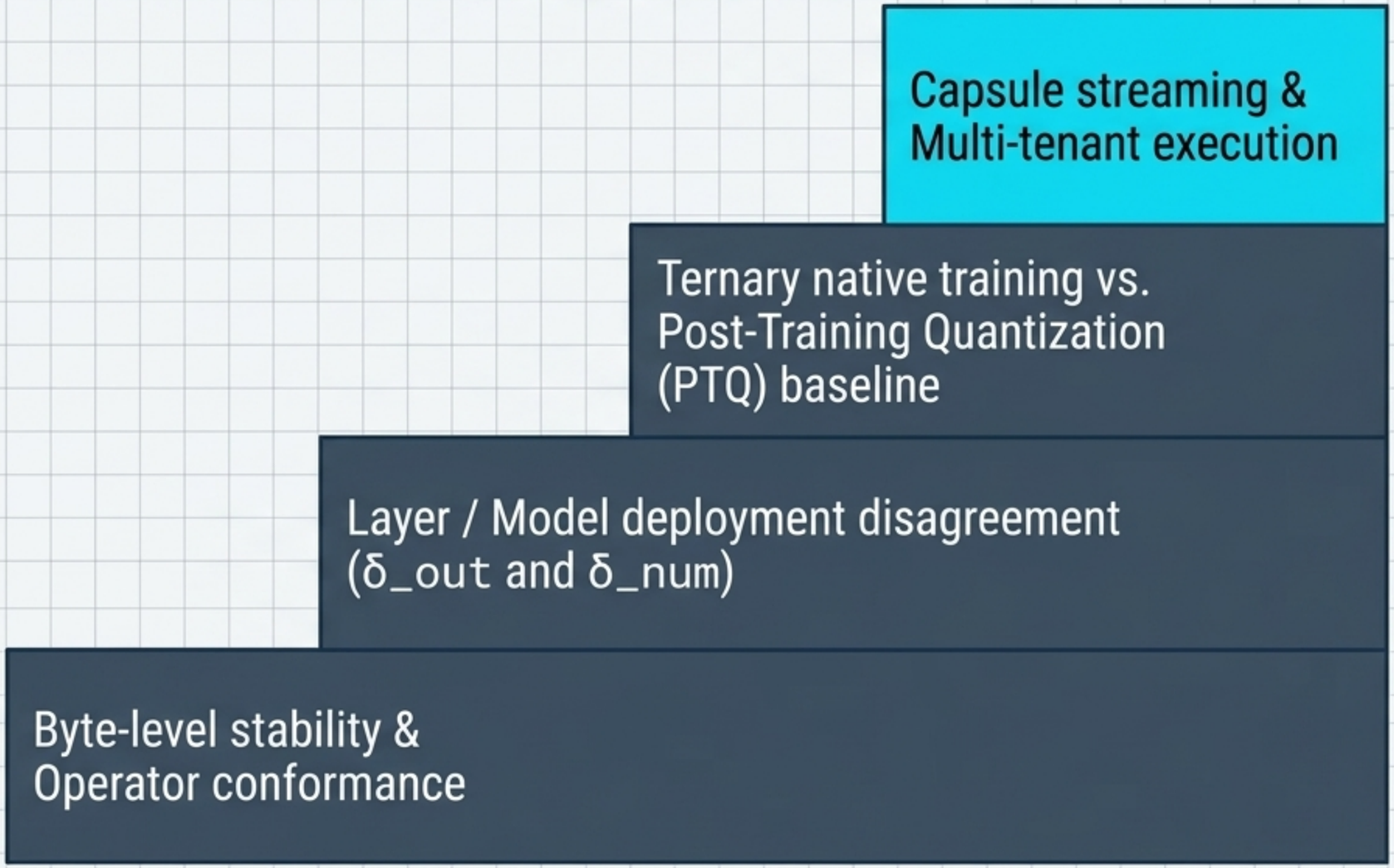
Verification Overhead Tradeoff

$$c_v / (c_v + c_x)$$

c_v = parse/verify cost, c_x = execution cost

Shrinking the TCB is only useful if the parsing overhead doesn't destroy latency for streaming workloads.

Engineering Claims Require Pre-Registered, Falsifiable Sequences



Demonstrations are not science.

Quality and efficiency must be reported jointly.

A speedup measured at a lower accuracy or altered batch size is not a controlled comparison.

Use Wilson or Clopper-Pearson intervals for low-rate disagreement.

Scientific Commitments Over Metaphysical Terminology

IF native training fails to reduce output drift relative to tuned export baseline… **THEN** reject or revise.

IF claimed efficiency disappears at matched task quality… **THEN** reject or revise.

IF the silver-label loop improves internal validator but degrades external evaluation… **THEN** reject or revise.

IF private operation depends on undeclared operator access… **THEN** reject or revise.

The ultimate value of this architectural framework is that its claims are testable, measurable, and falsifiable—stripping away hype for reproducible systems engineering.