Agentic Audiences Powered by the Universal Content Protocol (UCP)

A Technical Specification for Embeddings-Based Agent-to-Agent Advertising

Author: Evgeny Popov — Lead Technical Architect
Version: 5.1 (Final Corrected)
Date: February 2026
Status: Normative Specification
Supersedes: v5.0 (Release Candidate)
Key words: MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL in this document are to be interpreted as described in RFC 2119.

Executive Summary

Digital advertising is entering its agentic era—a paradigm where autonomous AI agents plan, negotiate, execute, and optimize campaigns at machine speed. This specification defines Agentic Audiences (formerly UCP), the open standard donated by LiveRamp to IAB Tech Lab in Q4 2025, as foundational data-plane infrastructure for agentic advertising.

UCP defines how intelligent agents exchange **embeddings**—compact, learned vector representations of 256–1024 dimensions—that encode identity, contextual, and reinforcement signals in a privacy-preserving, interoperable format. The architecture achieves RTB-compatible latency through binary quantization, enabling Hamming-distance search speeds of $< 1\text{ms}$ per million candidate vectors with a two-pass search architecture (binary candidate generation → float32 cosine reranking) completing in $< 4\text{ms}$ total.

This v5.1 Final Corrected release consolidates all normative content from the v4.4 and v5.0 drafts into a single, self-contained document with no placeholders, no cross-version references, and no abbreviated sections. Every appendix contains its full normative text. Key capabilities:

- **Normative vector space alignment** with mathematical acceptance functions and a Golden Test Set
- **Economic settlement** via AP2 Verifiable Digital Credentials
- **Hardware-rooted trust** through TEE attestation and IETF RATS
- **STRIDE threat modeling** with pseudocode verification flows
- **Privacy budget governance** for differential privacy
- **Spec-grade tool APIs** with idempotency, audit records, and HTTP status mappings
- **Agent Registry as trust fabric** with DID/X.509 identity, revocation propagation, and fail-closed semantics
- **BPF Psychographic Vector Contract** with proxy correlation bias testing

- **Normative Test Vectors** for privacy enforcement (GPP/TCF edge cases) and VAC conformance
- **Reproducible Benchmark Methodology** with exact index parameters, dataset characteristics, and hardware reference profiles
- **Governance & IPR Framework** establishing operational ownership of the Golden Test Set, Registry Root CA, and Projector signing authority under RAND-Z licensing

The critical interoperability challenge—agents using different foundation models producing vectors in incompatible latent spaces—is solved by the **Vector Alignment Contract (VAC)**, which defines model negotiation, signed projector artifacts, and graceful fallback to legacy AdCOM identifiers. If agents cannot verify vector space compatibility, similarity calculations produce mathematical noise; the VAC ensures this failure mode is detected and handled, making adoption incremental, resilient, and reversible.

Revision History

Version	Date	Author	Summary
4.0	January 2026	E. Popov	Initial public draft. Established four-plane embedding architecture (identity, contextual, reinforcement, behavioral), Vector Alignment Contract (VAC) with field table and projector specification, three normative tool APIs (AudienceCapability, AudienceValidation, CoverageCalculator), Economic Settlement framework with UsageRecord and AP2 integration, Hybrid Payload Architecture for backward compatibility.
4.1	January 2026	E. Popov	Security & Trust hardening. Added: STRIDE threat model (7 threats), TEE attestation verification pseudocode, DID/X.509 dual-identity agent model, `/lookup` API for Agent Registry with fail-closed semantics, Golden Test Set formal specification with diversity requirements, Mathematical Acceptance Function with LaTeX definitions, enhanced error taxonomy (`ALIGNMENT_BELOW_THRESHOLD`), HTTP status-to-reason code mapping, enhanced UsageRecord with `proof_refs`/`log_hash`.
4.2	February 2026	E. Popov	Normative Draft. Consolidated v4.0 + v4.1 into single document. Promoted all security requirements from SHOULD to MUST for L3+. Added BPF Psychographic Vector Contract (Appendix J) with proxy correlation bias testing.
4.4	February 2026	E. Popov	Complete VAC normative text. Full Section 2.3 with all seven sub-sections (Problem Statement through Handshake Protocol), including rendered Mermaid sequence diagram, Key Pinning Rules, and comprehensive GTS JSON Schema.
5.0	February 2026	E. Popov	Release Candidate. Added: Normative Privacy Test Vectors (Appendix G, §G.3). Reproducible Benchmark Recipe (Appendix E rewrite). Governance & IPR chapter

Version	Date	Author	Summary
			(Chapter 9). Implementer Checklists (Appendix K). Normative sequence diagrams for VAC Handshake and AP2 Billing Lifecycle.
5.1	February 2026	E. Popov	Final Corrected. Replaced all cross-version placeholder references in Appendices E–I with full normative text. Consolidated Section 2.3 from v4.4 source. All content is self-contained; no external document dependencies.

Change Tracking Convention

Changes from v4.2 to v5.0 are marked with **[v5.0-NEW]** in section headers. Content restored from v4.4 into v5.1 is marked with **[v5.1-RESTORED]**.

Part I — Paradigm & Governance

Chapter 1: The Agentic Advertising Paradigm

1.1 From Programmatic to Agentic

Programmatic advertising automated the mechanics of ad buying but left strategic decision-making in human hands. As detailed in the author's four-part series on the agentic transformation of digital advertising [3], this shift represents a qualitative leap: from systems where models serve as passive tools executing human strategies, to agentic systems that independently reason, learn, and evolve while pursuing defined objectives. The competitive implications are exponential—organizations that implement agentic systems learn faster, creating a compounding intelligence moat [3].

Agentic advertising enables AI-powered agents to autonomously plan strategies, execute campaigns, monitor performance, and optimize outcomes at a scale of 10^7+ decisions per second, billions of daily impression opportunities, and over 10^9 possible creative–audience–placement combinations—all within sub-100 ms decision windows.

The IAB Tech Lab Agentic Roadmap (January 2026) maps existing standards to agentic extensions:

Existing Standard	2026 Agentic Extension	Protocol Layer
OpenDirect	Agentic Direct	MCP + A2A
RTB (OpenRTB)	Agentic Bid	gRPC + Protobuf
Deals API	Agentic Deals	MCP + A2A
AdCOM	Agentic Ad Objects	JSON Schema
Ad Management API	Agentic Creative	MCP + A2A
ARTF	ARTF V2	gRPC + MCP

The **IAB Tech Lab Agent Registry**, launching March 1, 2026, catalogs agent types: Buyer, Seller, Creative, Audience, Measurement, Curation, Billing, Fraud, Signals, and Reporting agents. Each registered agent receives a verified identity, declared capabilities, and behavioral audit trail—critical infrastructure for the trust architecture described in Chapter 4. See Appendix I for the normative registry schema.

1.2 Why Embeddings Enable the Shift

The case for embeddings as foundational advertising infrastructure was first articulated in Popov's "Embeddings: The Next Frontier in Advertising" [1], which argued that dense vector representations would displace identifier-based signal exchange by simultaneously solving the latency, semantics, and privacy trilemma. UCP operationalizes this thesis as an open standard.

Traditional signal exchange suffers from three limitations in agentic contexts:

Verbosity and latency. A typical OpenRTB bid request with full user data, deal information, and contextual signals can exceed 10 KB. Passing this payload through multiple agent containers introduces serialization/deserialization overhead incompatible with sub-100 ms budgets.

Lack of semantic understanding. Segment ID "12345" carries no intrinsic meaning. An agent cannot compute similarity between segment IDs, cannot interpolate between audience definitions, and cannot generalize from observed segments to unseen but related audiences. Every semantic relationship must be pre-computed and stored in lookup tables.

Privacy exposure. Raw signal exchange—browsing histories, purchase records, location traces—creates re-identification risk. Even pseudonymous identifiers can be cross-referenced to reconstruct individual profiles.

Embeddings solve all three simultaneously: a 512-dimensional vector (2 KB in `float32`) encodes equivalent semantic information, captures similarity through vector geometry, and represents intent patterns rather than individual behaviors.

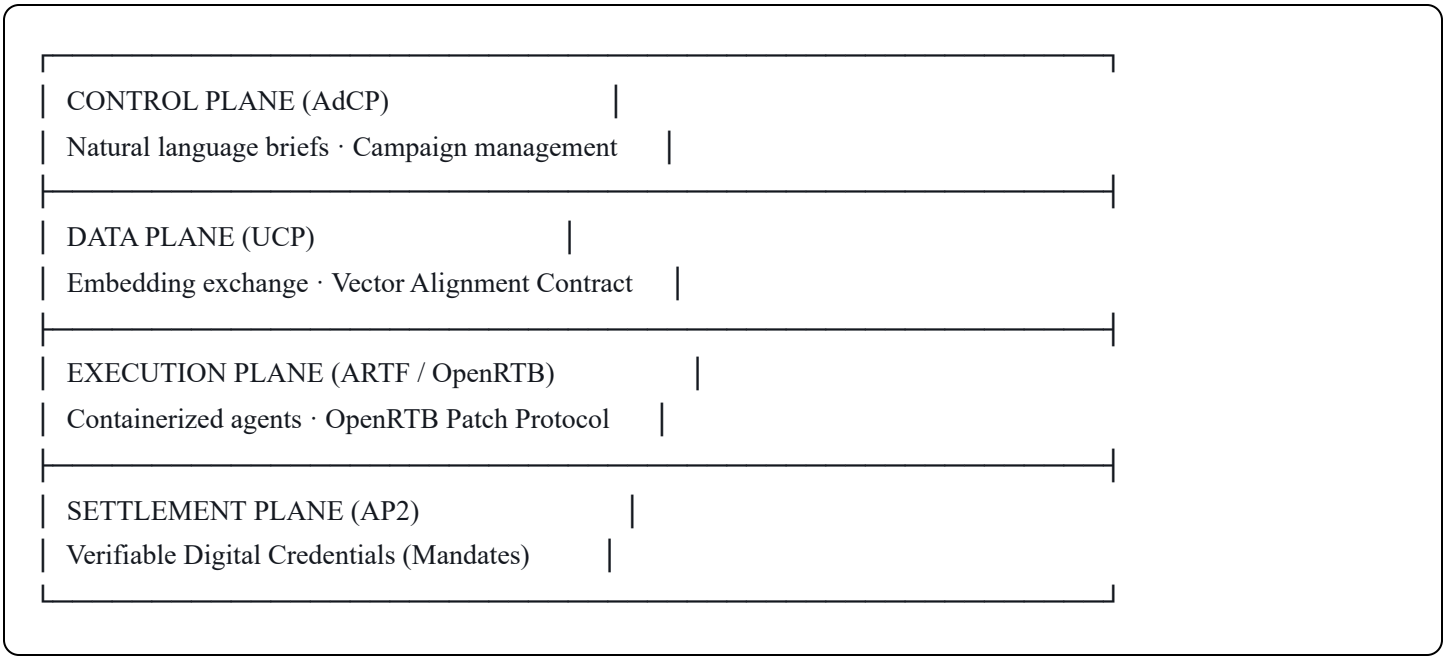
UCP defines four embedding types:

Embedding Type	Encodes	Freshness	Primary Use
Identity	Hashed IDs, behavioral segments, purchase history	Hours	Audience matching, frequency capping
Contextual	Page content, time of day, device type	Seconds	Real-time context alignment
Reinforcement	Impression outcomes, click probability, conversion signals	Days	Campaign optimization, bid strategy
Behavioral (BPF)	Psychographic dimensions: attention, emotion, cognition, intent, context, response	Hours	Predictive targeting (Chapter 6)

1.3 The Four-Plane Protocol Stack

The agentic advertising ecosystem is converging on a four-layer protocol architecture:

> **Control Plane — AdCP.** Built on MCP, AdCP enables natural-language campaign orchestration. The integration of MCP with advertising—explored in Popov's work on blending narrative frameworks with MCP-based agentic systems [4]—demonstrates how structured creative and emotional context can be encoded alongside transactional data. AdCP handles the *what* and *how*: Signals Activation, Media Buy, Creative, and Curation protocols.
>
> **Data Plane — UCP (Agentic Audiences).** UCP defines the *who* and *when*—how agents exchange embeddings encoding identity, contextual, and reinforcement signals. *Specified in this document.*
>
> **Execution Plane — ARTF + OpenRTB.** ARTF introduces containerized agents into the bidstream using the OpenRTB Patch Protocol for atomic, intent-declared mutations via gRPC (mandated) and MCP (supported).
>
> **Settlement Plane — AP2.** Google's Agent Payments Protocol uses Verifiable Digital Credentials (VDCs)—cryptographically signed Mandates—to authorize, authenticate, and audit agent-initiated transactions.



1.4 The Billing Agent & AP2 Lifecycle [v5.0-NEW]

When a Buyer and Seller Agent agree on a media buy via AdCP, the **Billing Agent** generates an *Intent Mandate* capturing campaign budget authority. As inventory is booked, *Cart Mandates* lock line items at agreed CPMs. Upon delivery confirmation, *Payment Mandates* trigger settlement. Each mandate is a VDC signed by the authorizing principal, creating a non-repudiable audit trail.

For computational tasks like `AudienceValidation`, the AP2 framework enables micro-settlements—the Billing Agent generates Cart Mandates for data access fees, embedding computation costs, and audience enrichment charges (see Appendix H for UsageRecord schema).

The following sequence diagram is normative. Implementations at L5 (Billing) MUST support all depicted message flows.

mermaid

```
sequenceDiagram
    participant BA as Buyer Agent
    participant SA as Seller Agent
    participant BilA as Billing Agent (AP2)
    participant Pay as Payment Rail

    Note over BA,Pay: Phase 1 — Intent
    BA->>BilA: Create Intent Mandate<br/>(campaign_id, budget_usd, currency, expiry)
    BilA-->>BA: Intent Mandate ID<br/>(mandate-intent-20260215-001)

    Note over BA,Pay: Phase 2 — Cart
    BA->>SA: AudienceValidation Request
    SA-->>BA: Validation Result + cost_usd
    BA->>BilA: Create Cart Mandate<br/>(intent_mandate_id, tool_name, cost_usd)
    BilA-->>BA: Cart Mandate ID<br/>(mandate-cart-val-20260215-001)

    Note over BA,Pay: Phase 3 — Delivery Confirmation
    SA->>BilA: Emit UsageRecord<br/>(cart_mandate_id, billable_units, nonce, log_hash)
    BilA->>BilA: Verify anti-fraud fields<br/>(nonce, attestation_ref, idempotency_key)

    Note over BA,Pay: Phase 4 — Payment
    BilA->>Pay: Create Payment Mandate<br/>(usage_record_id, verified_amount_usd)
    Pay-->>BilA: Payment Confirmation
    BilA-->>BA: Settlement Receipt
    BilA-->>SA: Settlement Receipt

    Note over BA,Pay: Dispute Path (optional)
    BA->>BilA: Dispute(usage_record_id, reason_code, evidence)
    BilA->>SA: Forward Dispute (3-day response window)
    alt Seller responds
        SA->>BilA: Counter-evidence
        BilA->>BilA: Evaluate evidence
    else No response by Day 10
        BilA->>BilA: Auto-resolve in Buyer favor
    end
    BilA-->>BA: Resolution
```

Normative flow rules:

1. Every Cart Mandate MUST reference a valid, non-expired Intent Mandate via `intent_mandate_id`.
2. The sum of all Cart Mandate amounts MUST NOT exceed the Intent Mandate `budget_usd`.
3. Payment Mandates MUST NOT be issued until delivery confirmation is received.

4. Each mandate (Intent, Cart, Payment) is a VDC signed by the authorizing principal, creating a non-repudiable audit trail per AP2.
5. For UCP tool calls (AudienceValidation, CoverageCalculator), the Billing Agent MUST generate Cart Mandates covering `compute_ms`, `vector_ops_count`, `enclave_ms`, and `bytes_processed` per the tariff in Appendix H.3.
6. Retroactive price changes MUST NOT be applied to already-committed Cart Mandates (Appendix H.3).

Chapter 1B: Conformance & Governance

1B.1 Compliance Levels

Implementations MUST declare which compliance level(s) they satisfy. Each level is cumulative—higher levels include all requirements of lower levels.

Level	Name	Requirements
L1	Core	MUST implement Hybrid Payload (§2.5). MUST include `legacy_fallback` in all exchanges. MUST support `request_id` idempotency. MUST produce audit records per §3.0.
L2	Interop	L1 + MUST implement full VAC (§2.3) including handshake, GTS validation, and projector consumption. MUST register in Agent Registry. MUST pass VAC conformance tests.
L3	Secure	L2 + MUST run within a TEE. MUST present attestation documents per §4.2. MUST sign all projector artifacts (JWS/Ed25519). MUST use mTLS rooted in Registry CA.
L4	Privacy	L3 + MUST enforce GPP/TCF matrix per §4.4. MUST implement DP budget governance per §4.5. MUST attest consent enforcement inside TEE per §1B.4.
L5	Billing	L4 + MUST emit UsageRecords per Appendix H. MUST support AP2 mandate lifecycle. MUST implement dispute workflow per Appendix H.5.

1B.2 Certification Artifacts

To achieve certification at each level, an agent MUST submit the following test evidence:

Level	Required Artifacts
L1	Hybrid Payload test results, idempotency test log, sample audit records
L2	VAC conformance test output (`ucp-vac-validator` results against GTS v1.0), Registry entry JSON
L3	TEE attestation document, signed container image digest, projector signature verification log
L4	Privacy enforcement test results (`privacy-enforcement-tests.jsonl`), DP budget audit log sample
L5	UsageRecord samples, AP2 mandate chain (Intent → Cart → Payment), dispute workflow test evidence

1B.3 Schema Versioning Policy

All UCP schemas MUST follow Semantic Versioning 2.0:

Major (e.g., 4.x → 5.0): Breaking changes. Consumers MUST NOT assume backward compatibility. 90-day transition window.

Minor (e.g., 4.1 → 4.2): Additive changes only. New optional fields. Existing consumers MUST NOT break. Immediate adoption permitted.

Patch (e.g., 4.1.0 → 4.1.1): Bug fixes, typo corrections, clarifications. No schema changes. Immediate adoption.

Every schema MUST include `$schema`, `$id` (with version), and `version` fields. Implementations MUST accept unknown fields (forward compatibility) and MUST NOT reject payloads containing unrecognized optional properties.

1B.4 Audit Log Requirements

All agents at L1 or above MUST maintain audit logs with the following minimum fields:

Field	Type	Required
`event_id`	UUIDv7	**MUST**
`event_type`	enum	**MUST**
`timestamp`	ISO 8601 (ms precision)	**MUST**
`agent_id`	string	**MUST**
`counterparty_id`	string	**MUST**
`request_id`	UUIDv7	**MUST**
`tool_name`	string	**MUST**
`vac_status_code`	enum	**MUST**
`consent_state_hash`	SHA-256	**MUST**
`resolution_method`	enum (`embedding`, `segment`, `hybrid`)	**MUST**
`response_time_ms`	integer	**MUST**
`error_code`	enum or null	**MUST**

Retention: Audit logs MUST be retained for ≥ 90 days (operational). SHOULD be retained for 365 days (billing/regulatory).

Integrity: Entries MUST be stored in append-only storage. At L3+, entries MUST be hash-chained (each entry's hash includes the previous entry's hash) to enable tamper detection. The hash chain MUST use SHA-256.

Access: Audit logs MUST be available to the counterparty agent upon request within 24 hours (for dispute resolution per Appendix H.5). Logs MUST NOT contain raw user data—only hashed identifiers and aggregate metrics.

Part II — Protocol Architecture

Chapter 2: UCP Protocol Architecture

2.1 Technical Deep-Dive

UCP operates across three evolutionary phases:

Phase 1 — Agent Interoperability Layer (current): Existing LLM agents exchange structured marketing context using standardized JSON schemas. Focus on context engineering, schema alignment, and real-time messaging between buyer, seller, and measurement agents.

Phase 2 — Context Learning Layer: Deep learning models train on contextual and behavioral data exchanged through the protocol, learning to represent user journeys, ad impressions, conversions, and marketplace signals as dynamic embeddings.

Phase 3 — Embedding Intelligence Layer: Agents evolve from exchanging textual context to exchanging embeddings that encode understanding of user intent, campaign state, and performance. These embeddings act as transferable memory between agents sharing a compatible vector space.

UCP defines seven technical components:

1. **Protocol Interfaces** — APIs and schemas for exchanging context, signals, and results
2. **Context Management** — Structured representations of campaign, audience, and creative context
3. **Embedding Interoperability** — The Vector Alignment Contract (§2.3) ensuring cross-model compatibility
4. **Agent Coordination Flows** — Handshake, negotiation, and fallback protocols
5. **Privacy and Consent Controls** — GPP/TCF enforcement, differential privacy budgets (§4.4–4.5)
6. **Agentic Attestation** — TEE verification, IETF RATS integration (§4.2)
7. **Token Exchange and Settlement** — AP2 mandate lifecycle, UsageRecord billing (§1.4, Appendix H)

2.2 Embeddings Specification

Dimensionality tiers:

Tier	Dims	float32 Size	int8 Size	Use Case
Compact	256	1 KB	256 B	High-frequency RTB, mobile
Standard	512	2 KB	512 B	General-purpose exchange
Rich	768	3 KB	768 B	Complex audience modeling
Full	1024	4 KB	1 KB	Research, offline analytics

Quantization tiers:

Level	Representation	Metric	Use Case
float32	Full precision (32 bits/dim)	Cosine / Dot Product	Final ranking, reranking pass
float16	Half precision (16 bits/dim)	Cosine / Dot Product	Edge deployments
int8	Scalar quantized (8 bits/dim)	Dot Product (approx)	First-pass ANN candidate generation
binary	1-bit per dimension	Hamming Distance	Ultra-fast candidate generation ($< 1\text{ms}$/1M vectors)

Core embedding exchange schema:

```json
```

```
{
 "ucp_embedding": {
  "version": "5.1",
  "agent_id": "iab-reg-buyer-agent-acme",
  "timestamp": "2026-02-15T14:30:00Z",

  "embedding_spec": {
   "model_id": "ucp-foundation-v1.2",
   "model_family": "iab-techlab/ucp-distilled-6L-512d",
   "space_id": "ucp-space-v1.0",
   "dimensions": 512,
   "encoding": "float32",
   "quantization_available": ["float16", "int8", "binary"],
   "normalization": "l2",
   "distance_metric": "cosine"
  },

  "identity_embedding": {
   "vector": [0.0234, -0.1567, 0.0891, "...512 dims..."],
   "signal_types": ["hashed_id", "behavioral_segments", "purchase_history"],
   "confidence": 0.87,
   "freshness_seconds": 3600
  },

  "contextual_embedding": {
   "vector": [0.1123, 0.0456, -0.0789, "...512 dims..."],
   "signal_types": ["page_content", "time_of_day", "device_type"],
   "confidence": 0.92,
   "freshness_seconds": 1
  },

  "reinforcement_embedding": {
   "vector": [-0.0345, 0.2234, 0.0167, "...512 dims..."],
   "signal_types": ["impression_response", "click_probability"],
   "confidence": 0.79,
   "freshness_seconds": 86400
  },

  "legacy_fallback": {
   "segment_ids": ["IAB-602", "IAB-607", "IAB-612"],
   "taxonomy_version": "iab_audience_1.1"
  },
```

```
  "privacy": {
    "gpp_string": "DBACNYA~CPXxRfAPXxRfAAfKABENB-CgAAAAAAAAAAYg",
    "tcf_consent_string": "CPXxRfAPXxRfAAGABCENATEIAACAAAAAAAAAAA",
    "ttl_seconds": 86400,
    "permitted_uses": ["audience_matching", "frequency_capping"],
    "attestation": {
      "agent_registry_id": "iab-reg-buyer-001",
      "execution_environment": "tee",
      "tee_platform": "aws_nitro_enclave",
      "attestation_document": {
        "format": "ietf_rats_eat",
        "nonce": "a1b2c3d4e5f6",
        "pcr_hash": "sha384:e7f8a9b0c1d2",
        "timestamp": "2026-02-15T14:30:00Z"
      }
    }
  }
}
```

2.3 Normative Vector Alignment [v5.1-RESTORED]

2.3.1 The Vector Alignment Problem

In a decentralized agentic ecosystem, agents MAY use different foundation models. A Buyer Agent using BERT and a Seller Agent using RoBERTa produce vectors in incompatible latent spaces. Computing cosine similarity between these vectors produces mathematical noise. This is the most dangerous failure mode in agent-to-agent advertising: agents silently compute meaningless scores with no error signal.

The Vector Alignment Contract (VAC) solves this problem by defining a mandatory declaration, negotiation, and execution protocol that ensures all embedding exchanges occur within a shared, validated vector space.

Without VAC, two failure modes are possible:

1. **Silent degradation.** Agents compute cosine similarity between incompatible vectors. The resulting scores are mathematically valid but semantically meaningless. No error is raised. Campaign performance degrades with no diagnostic signal.
2. **Dimensional mismatch.** Agents attempt to compute similarity between vectors of different dimensionality. This produces a hard runtime error, which is detectable but still represents a complete transaction failure.

The VAC eliminates both failure modes by requiring agents to declare their embedding space, negotiate compatibility before any exchange, and fall back to legacy segment IDs when alignment cannot be achieved.

2.3.2 Vector Alignment Contract (VAC) — Field Table

Every agent embedding declaration MUST include the following fields in its `AudienceCapability` response:

Field	Type	Req.	Description
`model_id`	string	**MUST**	Unique model identifier following `{provider}-{name}-v{major}.{minor}` convention. Example: `ucp-foundation-v1.2`
`model_family`	string	**MUST**	HuggingFace-style model family identifier. Example: `iab-techlab/ucp-distilled-6L-512d`
`space_id`	string	**MUST**	Versioned vector space identifier following `{provider}-space-v{major}.{minor}`. Example: `ucp-space-v1.0`. Two agents sharing the same `space_id` are guaranteed compatible without a projector.
`dimensions`	integer	**MUST**	Vector dimensionality. Permitted values: 256, 384, 512, 768, or 1024.
`encoding`	enum	**MUST**	Numeric encoding of vector components. Values: `float32`, `float16`, `int8`, `binary`.
`normalization`	enum	**MUST**	Vector normalization applied before transmission. Values: `l2` or `none`. All vectors exchanged via UCP tools MUST use `l2` normalization.
`distance_metric`	enum	**MUST**	Similarity function. Values: `cosine`, `dot_product`, `euclidean`. For L2-normalized vectors, `cosine` and `dot_product` are equivalent.
`quantization_level`	enum	**SHOULD**	Declares available precision levels for the two-pass search architecture. Values: `pq64`, `pq32`, `binary`, `none`.

Compatibility rule: Two agents are **directly compatible** if and only if `space_id`, `dimensions`, `encoding`, and `normalization` all match. If any field differs, a projector is required (§2.3.3) or fallback to legacy segments is triggered (§2.5).

2.3.3 Projector Specification & Key Pinning Rules

When two agents declare different `space_id` values, a **projector**—a linear transformation matrix—may be used to map vectors from the source space into the target space. A projector MUST be a signed JSON artifact:

```
json
```

```
{
 "projector": {
  "source_space_id": "custom-bert-v1",
  "target_space_id": "ucp-space-v1.0",
  "matrix_dimensions": [512, 384],
  "matrix_data": "<base64-encoded row-major float32>",
  "calibration_metrics": {
   "cosine_rmse": 0.06,
   "pass_rate_at_0.10": 0.96,
   "mse": 0.038,
   "spearman_rho_k100": 0.93,
   "identity_pair_min_cosine": 0.92,
   "tail_divergence": 0.18,
   "gts_version": "gts-v1.0",
   "gts_pairs_tested": 1000
  },
  "signature": {
   "algorithm": "Ed25519",
   "value": "<JWS per RFC 7515>",
   "signed_by": "iab-reg-buyer-agent-acme",
   "key_id": "key-buyer-acme-2026-001"
  },
  "created_at": "2026-02-01T00:00:00Z",
  "expires_at": "2026-05-01T00:00:00Z"
 }
}
```


Key Pinning Rules:

1. Implementations MUST reject projectors with expired or invalid signatures.
2. Projector signing keys MUST rotate every 90 days.
3. The `signed_by` field MUST match a registered `agent_id` in the IAB Agent Registry (Appendix I).
4. The `key_id` MUST correspond to an active, non-revoked public key in the agent's registry entry.
5. Projectors MUST declare the `gts_version` against which calibration metrics were computed. Agents MUST reject projectors calibrated against a deprecated GTS version (§2.4).
6. The `matrix_dimensions` field encodes `[target_dims, source_dims]`. The projector matrix $P \in \mathbb{R}^{d_t \times d_s}$ transforms a source vector $\mathbf{s} \in \mathbb{R}^{d_s}$ into the target space via $\mathbf{t}' = P\mathbf{s}$. The result MUST be L2-normalized after projection.
7. Agents MUST NOT cache projectors beyond their `expires_at` timestamp. On expiry, the agent MUST re-fetch or re-negotiate.

2.3.4 Golden Test Set (GTS)

The Golden Test Set is the normative acceptance oracle for projector validation and cross-space alignment. Minimum GTS size: 1,000 pairs.

Category	Min Pairs	Description
`identity_pair`	200	Same concept encoded twice in both spaces. Expected similarity ≈ 1.0.
`high_similarity`	200	Related concepts. Expected similarity ∈ [0.80, 0.95]. Example: "auto_enthusiast" ↔ "vehicle_shopper".
`moderate_similarity`	200	Loosely related concepts. Expected similarity ∈ [0.40, 0.79].
`low_similarity`	200	Unrelated concepts. Expected similarity ∈ [0.00, 0.39].
`adversarial`	100	Edge cases: near-zero vectors, high-dimensional cluster boundaries, synthetic noise.
`protected_class_probe`	100	Pairs designed to detect bias leakage into protected demographic attributes.

Generation & Maintenance: The GTS MUST be published by IAB Tech Lab as a versioned, immutable artifact with a SHA-256 integrity hash. The GTS MUST be regenerated every 90 days or immediately upon any major `space_id` version change (§2.4). GTS generation MUST use seeded pseudorandom sampling (e.g., `numpy.random.seed(42)`) to ensure deterministic reproducibility. Two implementations running the same projector on the same GTS MUST produce identical metric outputs within float32 rounding tolerance (±1e-6). Vendors MAY extend the GTS with proprietary pairs but MUST NOT remove or modify normative pairs.

GTS Entry JSON Schema:

```json
```

```
{
 "$schema": "https://json-schema.org/draft/2020-12/schema",
 "$id": "golden-test-set-entry-v1.0",
 "title": "UCP Golden Test Set Entry",
 "type": "object",
 "required": [
  "pair_id", "concept_label", "source_vector",
  "target_vector", "expected_similarity", "category"
 ],
 "properties": {
  "pair_id": { "type": "string", "pattern": "^gts-[0-9]{4,}$" },
  "concept_label": { "type": "string" },
  "source_vector": {
   "type": "object",
   "required": ["space_id", "dims", "vector"],
   "properties": {
    "space_id": { "type": "string" },
    "dims": { "type": "integer", "enum": [256, 384, 512, 768, 1024] },
    "vector": { "type": "array", "items": { "type": "number" } }
   }
  },
  "target_vector": {
   "type": "object",
   "required": ["space_id", "dims", "vector"],
   "properties": {
    "space_id": { "type": "string" },
    "dims": { "type": "integer", "enum": [256, 384, 512, 768, 1024] },
    "vector": { "type": "array", "items": { "type": "number" } }
   }
  },
  "expected_similarity": { "type": "number", "minimum": -1.0, "maximum": 1.0 },
  "category": {
   "type": "string",
   "enum": ["identity_pair", "high_similarity", "moderate_similarity",
            "low_similarity", "adversarial", "protected_class_probe"]
  }
 }
}
```


2.3.5 Mathematical Acceptance Function

All metrics MUST be computed on L2-normalized vectors. For $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ where $\|\mathbf{a}\|_2 = \|\mathbf{b}\|_2 = 1$:

Definition 1 — Cosine Similarity:

$$\text{sim}(\mathbf{a}, \mathbf{b}) = \mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{d} a_i \, b_i$$

Definition 2 — Mean Squared Error:

$$\text{MSE}(P) = \frac{1}{|G|} \sum_{(\mathbf{s},\, \mathbf{t}) \in G} \|P\mathbf{s} - \mathbf{t}\|_2^2$$

where $P \in \mathbb{R}^{d_t \times d_s}$ is the projector matrix, G is the Golden Test Set.

Definition 3 — Pairwise Cosine Preservation:

$$\delta_i = \left| \text{sim}(P\mathbf{s}_i, \mathbf{t}_i) - \text{sim}(\mathbf{s}_i, \mathbf{t}_i) \right|$$

Definition 4 — Pass Rate:

$$\text{PassRate}(\tau) = \frac{|\{i : \delta_i \leq \tau\}|}{|G|}$$

Acceptance Thresholds:

Metric	Formula	Threshold	Level	Disposition
Cosine sim (identity pairs)	$\text{sim}(P\mathbf{s}, \mathbf{t})$	≥ 0.90 for ALL identity pairs	**MUST**	Reject projector if ANY identity pair fails
Pass rate	$\text{PassRate}(0.10)$	≥ 0.95	**MUST**	$\geq$ 95% of ALL GTS pairs within $\delta \leq 0.10$
MSE	$\text{MSE}(P)$	< 0.05	**MUST**	Reject if exceeded
Rank preservation	Spearman ρ at $K = 100$	≥ 0.90	**SHOULD**	Log warning if below
Tail divergence	$\max_i \delta_i$	≤ 0.25	**SHOULD**	Flag for review

Pass Semantics: A projector P is **VALID** if and only if ALL identity pairs satisfy $\text{sim}(P\mathbf{s}, \mathbf{t}) \geq 0.90$, AND $\text{PassRate}(0.10) \geq 0.95$, AND $\text{MSE}(P) < 0.05$. If ANY condition fails:

`ALIGNMENT_BELOW_THRESHOLD`.

2.3.6 Error Taxonomy

Code	Meaning	Action
`MODEL_MATCH`	Same `space_id`, `dimensions`, `encoding`, `normalization`.	Proceed directly.
`MODEL_MISMATCH`	Different `space_id` values.	Initiate projector negotiation.
`PROJECTOR_REQUIRED`	Mismatch; no projector declared.	Fallback to legacy segments or terminate.
`ALIGNMENT_BELOW_THRESHOLD`	Projector failed acceptance thresholds.	Return rich error. Fallback.
`FALLBACK_TRIGGERED`	Legacy AdCOM segment exchange activated.	Proceed using `legacy_fallback`.
`CONSENT_BLOCKED`	Privacy framework blocks exchange.	MUST NOT exchange embeddings.

Rich error response for `ALIGNMENT_BELOW_THRESHOLD`:

```
json
```

```
{
 "error": {
  "code": "ALIGNMENT_BELOW_THRESHOLD",
  "request_id": "01945f3a-0001-7d1e-8a3f-9c0d1e2f3a4b",
  "timestamp": "2026-02-15T14:30:00Z",
  "measured_metrics": {
   "identity_pair_min_cosine": 0.87,
   "pass_rate_at_0.10": 0.91,
   "mse": 0.062,
   "spearman_rho_k100": 0.88
  },
  "required_thresholds": {
   "identity_pair_min_cosine": 0.90,
   "pass_rate_at_0.10": 0.95,
   "mse": 0.05
  },
  "failures": [
   "identity_pair_min_cosine: 0.87 < 0.90 (MUST)",
   "pass_rate: 0.91 < 0.95 (MUST)",
   "mse: 0.062 > 0.05 (MUST)"
  ],
  "gts_version": "gts-v1.0",
  "projector_id": "proj-custom-bert-to-ucp-v1",
  "next_steps": {
   "projector_required": true,
   "fallback_allowed": true,
   "retrain_recommendation": "Increase training epochs or use more GTS-aligned training pairs"
  }
 }
}
```


2.3.7 Vector Alignment Handshake Protocol [v5.0-NEW]

The following diagram is normative. Implementations MUST support all depicted message flows.

```
mermaid
```

```
sequenceDiagram
    participant BA as Buyer Agent
    participant SA as Seller Agent
    participant REG as IAB Agent Registry

    Note over BA,REG: Phase 1 — Declaration
    BA->>REG: GET /v1/agents/{seller_agent_id}
    REG-->>BA: AgentRegistryEntry<br/>(supported_models, space_id, projectors)
    BA->>SA: AudienceCapability Request<br/>(model_id, space_id, dimensions, encoding)

    Note over BA,REG: Phase 2 — Negotiation
    SA->>SA: Compare space_id, dimensions,<br/>encoding, normalization
    alt MODEL_MATCH
        SA-->>BA: vac_status: MODEL_MATCH
    else MODEL_MISMATCH (Projector Available)
        SA->>SA: Retrieve projector for<br/>source_space → target_space
        SA->>SA: Validate projector signature<br/>(Ed25519, key_id, expiry)
        SA->>SA: Run projector against GTS<br/>(1000+ pairs, 6 categories)
        SA->>SA: Compute acceptance metrics<br/>(MSE, PassRate, identity cosine)
        alt Projector VALID
            SA-->>BA: vac_status: MODEL_MATCH<br/>(via projector)
        else Projector INVALID
            SA-->>BA: vac_status: ALIGNMENT_BELOW_THRESHOLD<br/>(rich error with measured_metrics)
        end
    else MODEL_MISMATCH (No Projector)
        SA-->>BA: vac_status: PROJECTOR_REQUIRED
    end

    Note over BA,REG: Phase 3 — Execution or Fallback
    alt Alignment Achieved
        BA->>SA: Embedding exchange in agreed space
        SA-->>BA: Validation/Coverage results
    else Fallback
        BA->>SA: legacy_fallback (AdCOM segment IDs)
        SA-->>BA: Segment-based results
    end
```

Phase 1 — Declaration. The initiating agent declares its Model Family, quantization level, and projector capabilities in the `AudienceCapability` request. The agent SHOULD pre-fetch the counterparty's capabilities from the Agent Registry.

Phase 2 — Negotiation. The receiving agent compares the four compatibility fields. If all match: `MODEL_MATCH`. If any differs and a projector exists: apply P, validate against GTS, compute metrics. If the

projector passes: `MODEL_MATCH` via projection. If it fails: `ALIGNMENT_BELOW_THRESHOLD`. If no projector: `PROJECTOR_REQUIRED`.

Phase 3 — Execution or Fallback. On success, agents exchange embeddings. On failure, agents activate `legacy_fallback` (§2.5).

---

§2.4 Model Lifecycle Governance

Versioning: `model_id` MUST follow `{provider}-{name}-v{major}.{minor}`. `space_id` MUST follow `{provider}-space-v{major}.{minor}`. Major versions are NOT comparable without a projector. Minor versions MUST maintain cosine RMSE ≤ 0.05.

Event	Timeline	Requirement
New minor version	Immediate	Agents SHOULD adopt within 30 days
New major version	90-day transition	Registry MUST support both versions concurrently
Old major deprecated	+90 days after new major	Agents MUST NOT emit deprecated vectors
Emergency deprecation	Immediate	Registry sets `revoked: true`; agents MUST stop within 24 hours

Rollback: If alignment degrades post-update, the agent MUST revert within 1 hour and file `ALIGNMENT_DEGRADATION` with the Registry.

---

§2.5 Hybrid Payload Architecture

Every UCP exchange MUST carry `legacy_fallback` with AdCOM segment IDs. The `resolution_policy` field controls migration posture:

Policy	Behavior
`prefer_segments_try_embedding`	Use segments; log embedding scores for A/B comparison.
`prefer_embedding_fallback_segments`	Use embeddings; fall back to segments on failure.
`embedding_only`	No fallback. Transaction fails if alignment cannot be achieved.

```
json
```

```
{
  "intent": "activateSegments",
  "op": "add",
  "path": "/user/ext/ucp_embedding",
  "value": {
    "embedding": {
      "model_id": "ucp-foundation-v1.2",
      "space_id": "ucp-space-v1.0",
      "dimensions": 512,
      "encoding": "float32",
      "vector": [0.0234, -0.1567, 0.0891, "...512 dims..."]
    },
    "legacy_segments": {
      "ids": ["IAB-602", "IAB-607"],
      "taxonomy": "iab_audience_1.1"
    },
    "resolution_policy": "prefer_embedding_fallback_segments",
    "embedding_timeout_ms": 10
  }
}
```

Three-stage migration: Stage 1 (`prefer_segments_try_embedding`) → Stage 2 (`prefer_embedding_fallback_segments`) → Stage 3 (`embedding_only`).

Part III — Tool APIs

Chapter 3: Spec-Grade Tool APIs

3.0 Common Requirements

Idempotency: All requests MUST include `request_id` (UUIDv7). Servers MUST return identical responses for duplicate `request_id` within 300 seconds.

Audit Records: Every invocation MUST produce:

```
json
```

```
{
 "audit_record": {
  "request_id": "01945f3a-7b2c-7d1e-8a3f-9c0d1e2f3a4b",
  "tool_name": "AudienceValidation",
  "timestamp": "2026-02-15T14:30:00.123Z",
  "caller_agent_id": "iab-reg-buyer-agent-acme",
  "responder_agent_id": "iab-reg-seller-agent-premium-pub",
  "vac_status_code": "MODEL_MATCH",
  "consent_state": { "gpp_validated": true, "tcf_purposes_checked": [1, 3, 4] },
  "resolution_method": "embedding",
  "response_time_ms": 12,
  "billing": { "ap2_cart_mandate_id": "mandate-cart-val-20260215-001", "cost_usd": 0.002 },
  "privacy_budget_consumed": { "epsilon": 0.0, "tool_exempt": true }
 }
}
```


Error Codes (closed enumeration):

Code	Scope	Meaning
`MODEL_MISMATCH`	VAC	No common embedding model
`PROJECTOR_REQUIRED`	VAC	Mismatch; projector negotiation needed
`ALIGNMENT_BELOW_THRESHOLD`	VAC	Projector failed quality thresholds
`FALLBACK_TRIGGERED`	VAC	Legacy segment exchange activated
`CONSENT_BLOCKED`	Privacy	GPP/TCF blocks this operation
`BUDGET_EXHAUSTED`	Privacy	DP epsilon budget depleted
`ATTESTATION_FAILED`	Security	TEE attestation verification failed
`AGENT_NOT_REGISTERED`	Registry	Agent ID not found
`AGENT_REVOKED`	Registry	Agent capability revoked
`REQUEST_EXPIRED`	Transport	Timestamp older than 60 seconds
`DUPLICATE_REQUEST`	Idempotency	`request_id` already processed
`VALIDATION_FAILED`	Tool	Input schema validation error
`INTERNAL_ERROR`	Tool	Unrecoverable server error
`RATE_LIMITED`	Transport	Throttle; check `retry_after_seconds`

HTTP Status → Reason Code Mapping:

HTTP	Code(s)	Meaning
200	*(success)*	Normal
400	`VALIDATION_FAILED`	Schema error
401	`ATTESTATION_FAILED`	mTLS/attestation missing
402	`BUDGET_EXHAUSTED`	DP epsilon depleted
403	`CONSENT_BLOCKED`, `AGENT_REVOKED`	Forbidden
404	`AGENT_NOT_REGISTERED`	Unknown agent
409	`DUPLICATE_REQUEST`	`request_id` reused
422	`ALIGNMENT_BELOW_THRESHOLD`, `MODEL_MISMATCH`, `PROJECTOR_REQUIRED`	Semantic error
429	`RATE_LIMITED`	Throttled
500	`INTERNAL_ERROR`	Server error

3.1 AudienceCapability

Task: Discover and negotiate embedding model compatibility.

Request:

```json
```

```
{
  "tool": "AudienceCapability",
  "version": "5.1",
  "request": {
    "request_id": "01945f3a-0001-7d1e-8a3f-9c0d1e2f3a4b",
    "timestamp": "2026-02-15T14:30:00Z",
    "requesting_agent_id": "iab-reg-buyer-agent-acme",
    "vac": {
      "preferred_models": [{
        "model_id": "ucp-foundation-v1.2",
        "model_family": "iab-techlab/ucp-distilled-6L-512d",
        "space_id": "ucp-space-v1.0",
        "dimensions": 512,
        "encoding": "float32",
        "normalization": "l2",
        "distance_metric": "cosine"
      }],
      "projector_support": {
        "can_provide": true,
        "can_consume": true,
        "supported_target_spaces": ["ucp-space-v1.0"]
      },
      "fallback_strategy": {
        "type": "adcom_segment_exchange",
        "taxonomy_version": "iab_audience_1.1",
        "segment_ids": ["IAB-602", "IAB-607"]
      }
    },
    "requirements": {
      "latency_requirement_ms": 50,
      "require_tee_attestation": true,
      "privacy_frameworks": ["gpp", "tcf_2.2"]
    },
    "consent": {
      "gpp_string": "DBACNYA~CPXxRfAPXxRfAAfKABENB-CgAAAAAAAAAAYg",
      "tcf_consent_string": "CPXxRfAPXxRfAAGABCENATEIAACAAAAAAAAAAA"
    }
  }
}
```


Response (Model Match):

```
json
```

```
{
 "tool": "AudienceCapability",
 "version": "5.1",
 "response": {
  "request_id": "01945f3a-0001-7d1e-8a3f-9c0d1e2f3a4b",
  "responder_agent_id": "iab-reg-seller-agent-premium-pub",
  "vac_status": {
   "code": "MODEL_MATCH",
   "agreed_model": { "model_id": "ucp-foundation-v1.2", "space_id": "ucp-space-v1.0", "dimensions": 512, "encoding": "fl
   "projection_applied": false,
   "fallback_activated": false
  },
  "capabilities": {
   "audience_types": [
    { "type": "first_party_deterministic", "reach_estimate": 45000000 },
    { "type": "modeled_lookalike", "reach_estimate": 120000000 }
   ],
   "performance": { "avg_response_ms": 12, "p99_response_ms": 45 }
  },
  "attestation": { "tee_platform": "aws_nitro_enclave", "enclave_image_hash": "sha384:9a8b7c6d5e4f...", "verification_statu
  "cache_control": { "max_age_seconds": 300 }
 }
}
```


Response (Consent Blocked):


```
json

{
 "error": {
  "code": "CONSENT_BLOCKED",
  "reason": "GPP targeted_advertising_opt_out=true",
  "request_id": "01945f3a-0001-7d1e-8a3f-9c0d1e2f3a4b",
  "timestamp": "2026-02-15T14:30:00Z"
 }
}
```


3.2 AudienceValidation

Task: Verify audience quality, authenticity, and compliance.

Request:

json


```
{
  "tool": "AudienceValidation",
  "version": "5.1",
  "request": {
    "request_id": "01945f3a-0002-7d1e-8a3f-9c0d1e2f3a4b",
    "timestamp": "2026-02-15T14:30:01Z",
    "buyer_agent_id": "iab-reg-buyer-agent-acme",
    "audience_definition": {
      "type": "hybrid",
      "embedding": {
        "model_id": "ucp-foundation-v1.2",
        "space_id": "ucp-space-v1.0",
        "dimensions": 512,
        "encoding": "float32",
        "vector": [0.0234, -0.1567, 0.0891, "...512 dims..."]
      },
      "legacy_fallback": { "segment_ids": ["IAB-602", "IAB-607"], "taxonomy_version": "iab_audience_1.1" }
    },
    "validation_criteria": { "min_recency_hours": 72, "min_match_rate": 0.65, "min_embedding_similarity": 0.70 },
    "consent": {
      "gpp_string": "DBACNYA~CPXxRfAPXxRfAAfKABENB-CgAAAAAAAAAAYg",
      "tcf_consent_string": "CPXxRfAPXxRfAAGABCENATEIAACAAAAAAAAAAA"
    },
    "billing": { "ap2_intent_mandate_id": "mandate-intent-2026Q1-acme-001", "max_cost_usd": 0.01 }
  }
}
```


Response (Happy Path):

json

```
{
  "tool": "AudienceValidation",
  "version": "5.1",
  "response": {
    "request_id": "01945f3a-0002-7d1e-8a3f-9c0d1e2f3a4b",
    "validation_status": "valid",
    "resolution_method": "embedding",
    "quality_metrics": {
      "match_rate": 0.78,
      "embedding_similarity_score": 0.86,
      "segment_overlap": ["IAB-602", "IAB-607", "IAB-612"],
      "data_recency": { "median_hours": 24, "p95_hours": 68 },
      "provenance": { "data_source": "first_party", "collection_method": "deterministic", "last_refresh": "2026-02-14T18:00:00
    },
    "privacy_compliance": { "gpp_validated": true, "tcf_purposes_checked": [1, 2, 3, 4, 7], "data_processing_basis": "consent"
    "billing_result": { "cost_usd": 0.002, "ap2_cart_mandate_id": "mandate-cart-val-20260215-001" },
    "cache_control": { "max_age_seconds": 60 }
  }
}
```


AudienceValidation Error Scenarios:

Condition	HTTP	Code	Example reason
Agent not in Registry	404	AGENT_NOT_REGISTERED	Agent unknown
Agent revoked	403	AGENT_REVOKED	Revoked: conformance failure
GPP blocks identity	403	CONSENT_BLOCKED	targeted_advertising_opt_out=true
DP epsilon exhausted	402	BUDGET_EXHAUSTED	epsilon_remaining=0.0
Model not supported	422	MODEL_MISMATCH	ucp-space-v2.0 unsupported
TEE failed	401	ATTESTATION_FAILED	Enclave hash mismatch
Duplicate request	409	DUPLICATE_REQUEST	Already processed
Dim mismatch	400	VALIDATION_FAILED	Expected 512, got 384
Overloaded	429	RATE_LIMITED	Retry-After: 5

3.3 CoverageCalculator

Task: Estimate reach, frequency, and overlap with differential privacy.

Request:

json


```
{
  "tool": "CoverageCalculator",
  "version": "5.1",
  "request": {
    "request_id": "01945f3a-0003-7d1e-8a3f-9c0d1e2f3a4b",
    "timestamp": "2026-02-15T14:30:02Z",
    "buyer_agent_id": "iab-reg-buyer-agent-acme",
    "target_audience": {
      "embedding": { "model_id": "ucp-foundation-v1.2", "space_id": "ucp-space-v1.0", "dimensions": 512, "vector": [0.0234,
      "legacy_fallback": { "segment_ids": ["IAB-602"], "taxonomy_version": "iab_audience_1.1" },
      "geographic_constraints": { "countries": ["US"], "dma_ids": ["501"] }
    },
    "channels": ["ctv", "display"],
    "flight_dates": { "start": "2026-03-01", "end": "2026-03-31" },
    "budget_usd": 250000,
    "privacy_budget": { "epsilon_per_query": 0.1, "epsilon_campaign_cap": 2.0, "epsilon_remaining": 1.3, "on_budget_exhaus
    "consent": {
      "gpp_string": "DBACNYA~CPXxRfAPXxRfAAfKABENB-CgAAAAAAAAAAYg",
      "tcf_consent_string": "CPXxRfAPXxRfAAGABCENATEIAACAAAAAAAAAAAA"
    }
  }
}
```


Response (Happy Path):

json

```
{
 "tool": "CoverageCalculator",
 "version": "5.1",
 "response": {
  "request_id": "01945f3a-0003-7d1e-8a3f-9c0d1e2f3a4b",
  "coverage_estimate": {
   "total_unique_reach": 8500000,
   "reach_by_channel": {
    "ctv": { "unique_reach": 4200000, "avg_frequency": 3.2, "cpm": 28.50 },
    "display": { "unique_reach": 5100000, "avg_frequency": 4.8, "cpm": 6.20 }
   },
   "cross_channel_overlap": { "ctv_and_display": 0.32 },
   "confidence_interval": { "lower_95": 7200000, "upper_95": 9800000 }
  },
  "methodology": { "model": "probabilistic_hll_sketch", "privacy_method": "differential_privacy", "epsilon_consumed": 0.1
  "cache_control": { "max_age_seconds": 300 }
 }
}
```


Response (Budget Exhausted):


```
json

{
 "error": {
  "code": "BUDGET_EXHAUSTED",
  "reason": "epsilon_remaining=0.0; identity signals blocked",
  "request_id": "01945f3a-0003-7d1e-8a3f-9c0d1e2f3a4b",
  "timestamp": "2026-02-15T14:30:02Z",
  "privacy_budget_state": { "epsilon_campaign_cap": 2.0, "epsilon_consumed": 2.0, "epsilon_remaining": 0.0 }
 }
}
```


3.4 Proposed Additional Tools

SemanticMatchingEngine — Two-pass quantized search (binary Hamming → float32 cosine reranking) for audience–content alignment.

DynamicSegmentOptimizer — Adaptive segment refinement using reinforcement embeddings with `max_embedding_drift` constraint.

CrossContextBridge — Translation between taxonomies and embedding spaces.

PredictivePerformanceEstimator — Pre-bid performance prediction with AP2 mandate verification.

Part IV — Security, Privacy & Trust

Chapter 4: Security, Privacy & Trust

4.1 STRIDE Threat Model

#	Category	Threat	Mitigation (MUST)
T1	Spoofing	Projector spoofing	Projectors MUST be JWS-signed (Ed25519). Consumers MUST verify AND run full GTS validation locally.
T2	Tampering	Semantic poisoning	Reject vectors with any dimension $> 4\sigma$ from population stats. Log anomalies.
T3	Repudiation	Unsigned transactions	All payloads MUST include `request_id` (UUIDv7) and agent signature. Audit records MUST be append-only.
T4	Info Disclosure	Embedding inversion	DP with $\varepsilon \leq 2.0$ MUST be applied. Cohort floor: 100 users. TEE MUST verify DP runs inside enclave.
T5	DoS	Replay attacks	`request_id` dedup within 300s. `timestamp` MUST be within 60s. Reject duplicates.
T6	Elevation	Registry manipulation	Append-only audit log. All mutations signed by operator key. Clients MUST verify response signatures.
T7	Spoofing	Stale capability claims	Capabilities MUST include `last_updated`. Reject if > 3600s old. Registry lookup MUST for high-risk ops.

4.2 TEE Attestation Verification

```
python
```

```python
def verify_attestation(attestation_doc, agent_id, registry_client, nonce_store):
    """Returns (verified: bool, reason_code: str)"""
    # Step 1: Nonce freshness (MUST be within 60s)
    nonce_record = nonce_store.get(attestation_doc.nonce)
    if not nonce_record or (now() - nonce_record.issued_at).seconds > 60:
        return (False, "ATTESTATION_NONCE_STALE")

    # Step 2: Certificate chain (MUST terminate at known TEE root CA)
    root_cas = load_trusted_tee_roots()  # AWS Nitro, Intel TDX, Azure CC
    if not validate_chain(attestation_doc.certificate_chain, root_cas):
        return (False, "ATTESTATION_CHAIN_INVALID")

    # Step 3: Enclave image hash (MUST match Registry entry)
    entry = registry_client.lookup(agent_id)
    if not entry:
        return (False, "AGENT_NOT_REGISTERED")
    if entry.revocation.revoked:
        return (False, "AGENT_REVOKED")
    if attestation_doc.enclave_image_hash != entry.attestation.expected_enclave_hash:
        return (False, "ATTESTATION_IMAGE_MISMATCH")

    # Step 4: PCR values (MUST match expected measurements)
    for pcr in ["pcr0", "pcr1", "pcr2"]:
        if attestation_doc.pcr_values[pcr] != entry.attestation[f"expected_{pcr}"]:
            return (False, "ATTESTATION_PCR_MISMATCH")

    # Step 5: Signature verification
    pub_key = extract_public_key(attestation_doc.certificate_chain[0])
    if not verify_signature(attestation_doc.payload_bytes, attestation_doc.signature, pub_key):
        return (False, "ATTESTATION_SIGNATURE_INVALID")

    # Step 6: Emit audit record (MUST)
    emit_audit_record("ATTESTATION_VERIFIED", agent_id, sha384(attestation_doc.payload_bytes))
    return (True, "ATTESTATION_VERIFIED")
```

4.3 Signing Requirements

Artifact	Method	Rotation
Container images	Cosign / Notary v2	180 days
Projector matrices	JWS (Ed25519) per RFC 7515	90 days

Artifact	Method	Rotation
Attestation documents	TEE hardware key	Per-boot
Registry entries	Operator key (Ed25519)	365 days

4.4 GPP/TCF Enforcement Matrix

Operation	Sale Opt-Out	Targeted Ad Opt-Out	No TCF Purpose 1	No TCF Purpose 4
Identity embedding exchange	**MUST** block	**MUST** block	**MUST** block	**MUST** block
Contextual embedding exchange	MAY proceed	MAY proceed	**MUST** block	MAY proceed
Reinforcement signal feedback	**MUST** block	**MUST** block	**MUST** block	**MUST** block
CoverageCalculator (aggregate)	MAY proceed	MAY proceed	**MUST** block	MAY proceed

Consent propagation: The initiating agent MUST include `gpp_string` and `tcf_consent_string` in every request. The receiving agent MUST independently parse and validate — MUST NOT trust the caller's assertion.

Consent freshness: Agents MUST include `consent.captured_at`. Receivers MUST configure `max_age_seconds` (default: 86400). Stale consent MUST be treated as `CONSENT_BLOCKED`.

Fail-closed principle: If consent strings are absent, malformed, or unparseable, agents MUST NOT assume consent. Default: BLOCK all operations.

4.5 Differential Privacy Budget Governance

```json
{
  "privacy_budget": {
    "campaign_id": "campaign-q1-brand-2026",
    "epsilon_cap": 2.0,
    "epsilon_consumed": 0.8,
    "epsilon_remaining": 1.2,
    "window_start": "2026-02-15T00:00:00Z",
    "window_end": "2026-03-15T00:00:00Z",
    "mechanism": "gaussian",
    "sensitivity": 1.0
  }
}
```

Sequential Composition: $\varepsilon_{\text{total}} = \varepsilon_1 + \varepsilon_2 + \cdots + \varepsilon_n$. Before each query: $\varepsilon_{\text{remaining}} = \varepsilon_{\text{cap}} - \varepsilon_{\text{consumed}} \geq \varepsilon_{\text{query}}$. If false: BUDGET_EXHAUSTED.

Exhaustion Degradation:

Remaining ε	Behavior
$> 50\%$ of cap	Normal operation. All signal types permitted.
25–50% of cap	SHOULD log BUDGET_WARNING. MAY reduce precision.
1–25% of cap	MUST block reinforcement signals. Identity and contextual MAY continue.
0 (exhausted)	MUST block identity AND reinforcement. Contextual MAY continue. Legacy fallback MUST activate.

Hash-Chained Verification Records:

```json
{
  "dp_ledger_entry": {
    "entry_id": "dp-20260215-0007",
    "prev_hash": "sha256:abc123...",
    "entry_hash": "sha256:def456...",
    "request_id": "01945f3a-0003-7d1e-8a3f-9c0d1e2f3a4b",
    "tool_name": "CoverageCalculator",
    "epsilon_consumed": 0.1,
    "epsilon_remaining_after": 1.2,
    "mechanism": "gaussian",
    "sensitivity": 1.0,
    "noise_sigma": 10.0,
    "timestamp": "2026-02-15T14:30:02Z",
    "attested_inside_tee": true,
    "enclave_signature": "<Ed25519-signature>"
  }
}
```

$$\texttt{entry\_hash} = \text{SHA-256}(\texttt{prev\_hash} \| \texttt{request\_id} \| \varepsilon_{\text{consumed}} \| \texttt{timestamp})$$

Part V — Taxonomies, Prediction & Implementation

Chapter 5: Taxonomies & Migration

5.1 Comparative Analysis

Aspect	AdCOM	AdCP	UCP
Role	Domain objects	Control plane	Data plane
Signal format	Segment IDs, key-value	Natural language + JSON	Dense vector embeddings
Latency	Sub-100 ms	Asynchronous	Sub-100 ms
Privacy	GPP/TCF passthrough	Audit trails	TEE-attested embeddings with DP
Failure mode	Silent (no error on mismatch)	Explicit errors	VAC Handshake + Hybrid Fallback
Semantic capability	None (discrete IDs)	Limited (structured text)	Full (continuous vector geometry)

5.2 Migration via Hybrid Payload

See §2.5 for the three-stage progression: `prefer_segments_try_embedding` → `prefer_embedding_fallback_segments` → `embedding_only`. This ensures zero-risk adoption with measurable A/B comparison at each stage.

Chapter 6: Behavioral Prediction Framework 2.0

6.1 Theoretical Foundation: From Relevance Realization to Predictive Infrastructure

This chapter extends the Behavioral Prediction Framework introduced in Popov [2], which established a psychologically informed model for predicting user propensity to engage across the advertising funnel. The original framework demonstrated how behavioral signals—including attention patterns, contextual relevance, and cognitive engagement markers—could be structured into a unified prediction architecture that outperforms demographic-only targeting models. UCP provides the infrastructure layer that makes this framework operational at agentic scale.

The theoretical foundation is grounded in **relevance realization**—the cognitive science concept that intelligence is not data processing but *dynamic salience filtering*—as explored in the author's DMEXCO column on rethinking marketing relevance [5]. The key insight is that humans do not process all available information equally; they dynamically filter for relevance based on context, goals, and prior experience. Advertising systems that mirror this cognitive architecture—selectively attending to the most salient signals

rather than processing exhaustive feature sets—achieve superior prediction accuracy with lower computational cost.

The Framework maps 15+ model categories across three operational phases:

Pre-Campaign Planning [6]: Agentic systems autonomously discover audiences by exploring the embedding space. Instead of requiring human planners to define segments ("women 25–34 interested in fitness"), the agent navigates the vector space to find clusters of high-intent users that no human would have identified. This is the "exploration" phase where embedding geometry reveals non-obvious audience structures.

Real-Time Execution [7]: Learning policies replace rigid rules in the sub-100ms bidding window. Instead of evaluating static business rules ("bid $2.50 for segment X"), the agent computes real-time similarity between the user's behavioral embedding and the campaign's ideal-outcome embedding, generating a bid price that reflects *predicted marginal value* rather than *category membership*.

Post-Campaign Learning: Reinforcement signals create compounding intelligence. Every impression outcome (view, click, conversion, or absence thereof) updates the reinforcement embedding, shifting the agent's understanding of what "works" for this campaign in this market at this time. Over thousands of iterations, the embedding space self-organizes to reflect causal patterns that no human planner could specify.

UCP embeddings transform this framework by providing models with richer, more efficient, and more transferable input representations. Pre-agentic models consumed raw features—age, gender, browsing history, purchase records—requiring bespoke feature engineering per model and per data source. With UCP, models consume and produce embeddings that encode multi-signal intent in a unified vector space. A single 512-dimensional identity embedding replaces hundreds of categorical and numerical features, while capturing non-linear interactions between them.

6.2 The Psychographic Vector Space

6.2.1 Why 512 Dimensions?

The BPF Psychographic Vector is a 512-dimensional real-valued vector that encodes cohort-level behavioral tendencies observed during advertising interactions. It is explicitly **not** a user profile—it represents a *behavioral state* that is transient, context-dependent, and privacy-preserving by construction.

The choice of 512 dimensions reflects a careful balance between expressiveness and operational constraints. At 256 dimensions, empirical testing shows degraded separation between nuanced behavioral states (e.g., "high-intent browsing" versus "comparison shopping" collapse into overlapping regions). At 1024 dimensions, the marginal information gain does not justify the doubling of storage, transmission, and computation costs—particularly given the sub-100ms latency budget of RTB environments. At 512 dimensions in float32 representation, the vector occupies exactly 2,048 bytes—compact enough for inclusion in OpenRTB bid requests while expressive enough to encode the six cognitive domains defined below.

6.2.2 Dimension-to-Domain Mapping

The 512-dimensional space is partitioned into six semantically meaningful domains:

Dimension Range	Semantic Domain	Encoded Signals	Why It Matters
0–63	**Attention**	Dwell time distribution, scroll depth, gaze proxy, viewport engagement, tab-switch frequency	Predicts *whether the user will notice* the ad.
64–127	**Emotional Engagement**	Sentiment trajectory, creative resonance patterns, emotional valence shifts, arousal indicators	Predicts *whether the ad will create an emotional response*.
128–191	**Cognitive Load**	Content complexity correlation, decision latency, comparison behavior depth, information-seeking patterns	Predicts *how much mental effort the user will invest*.
192–255	**Intent Strength**	Funnel stage indicators, comparison shopping signals, price sensitivity markers, urgency signals	Predicts *how close the user is to a decision*.
256–383	**Contextual Affinity**	Category preference trajectories, temporal patterns, seasonal sensitivity, cross-device context	Predicts *what content environment will maximize receptivity*.
384–511	**Response History**	Ad interaction patterns, conversion propensity, frequency response curves, creative fatigue indicators	Predicts *what the user will do after seeing the ad*.

6.2.3 Worked Examples

Example 1 — "Rationality" (Cognitive Load domain, dims 128–191). Segment IDs cannot express "this user tends to make analytical, comparison-driven decisions with low emotional influence." A segment ID like IAB-602 (Automotive Enthusiasts) captures *what* the user is interested in but not *how* they process information. In vector space, a high-rationality user has strong activations in dimensions 128–155 (content complexity correlation, decision latency, comparison depth) and weak activations in dimensions 64–95 (emotional engagement). This geometric relationship—expressible as a direction in 512-space—enables the agent to select analytical creative (specification tables, feature comparisons) over emotional creative (lifestyle imagery).

Example 2 — "Novelty Seeking" (cross-domain construct). Novelty seeking cannot be localized to a single domain—it manifests across Attention (rapid attention shifts, dim 48–63), Emotional Engagement (high arousal response to unfamiliar stimuli, dim 96–127), and Intent (early-funnel exploration behavior, dim 192–210). In vector space, this cross-domain pattern is a *direction* that spans multiple subvector ranges. The cosine similarity between a user's vector and the "novelty seeking" direction vector produces a single scalar score, computed in one operation. In segment-based systems, this would require manually constructing and maintaining a composite segment across three taxonomies.

6.3 The Safe Mode Protocol

6.3.1 Architecture: Coarse, Standard, and Fine-Grained Vectors

Not all agents deserve the same level of behavioral signal. The Safe Mode Protocol defines three resolution tiers:

Mode	What Is Shared	Dims Transmitted	Payload Size	Use Case
`coarse`	6 aggregate domain scores (one scalar per domain)	6	24 bytes	Low-trust contexts, no explicit consent, first-time interactions
`standard`	Full 512-dimensional vector	512	2,048 bytes	Normal consented operation, agents with valid Registry entry
`fine_grained`	512-dim + per-domain confidence + uncertainty	512 + 6 + 3	~2,120 bytes	High-trust partnerships, TEE attestation verified per §4.2

The **default is always** `coarse`. This is a normative requirement: implementations MUST NOT default to `standard` or `fine_grained` regardless of configuration convenience. If a consent signal is lost, corrupted, or misinterpreted, the system degrades to minimal disclosure rather than maximum exposure.

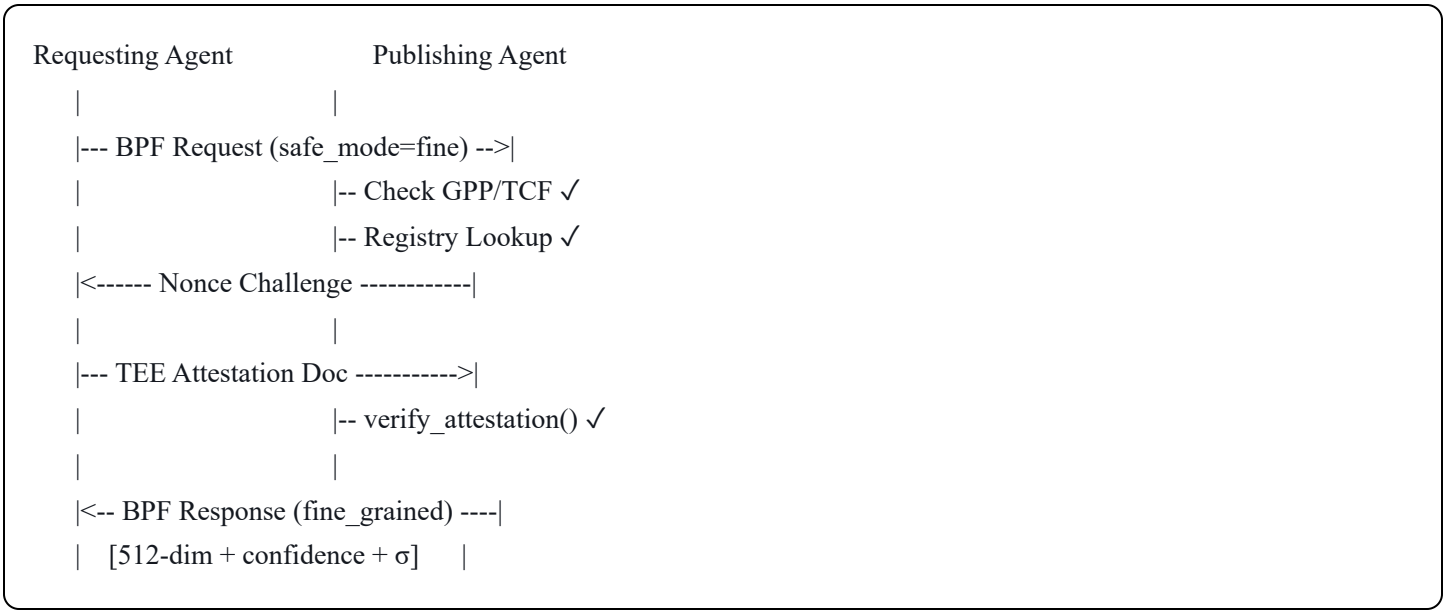
6.3.2 Escalation via TEE Attestation

An agent requiring `fine_grained` access must complete a three-step escalation:

Step 1 — Consent Verification. The requesting agent presents GPP and TCF consent strings. The publishing agent evaluates against the enforcement matrix (§4.4). If `targeted_ad_opt_out` is set or TCF Purpose 1 lacks legal basis, escalation is denied; response remains `coarse`.

Step 2 — Registry Verification. The publishing agent resolves the requesting agent's `agent_id` against the Agent Registry (Appendix I). The agent must have `status: active`, `revocation.revoked: false`, and capabilities including BPF vector consumption. If the registry entry is older than 3,600 seconds, a fresh lookup MUST be performed.

Step 3 — TEE Attestation Challenge. The publishing agent issues a fresh nonce (valid for 60 seconds) and requires a TEE attestation document. Verification follows §4.2 pseudocode. Only upon ALL steps passing does the publisher escalate to `fine_grained`.



If any step fails, the publishing agent MUST NOT silently downgrade to `standard`. It MUST return an explicit error response with the appropriate reason code so the requesting agent can decide whether to retry, fall back to `coarse`, or abort.

6.3.3 Coarse Mode as Privacy Floor

In `coarse` mode, the six aggregate scores are computed as L2-norms of each domain subvector, normalized to [0, 1]. This provides enough signal for basic optimization without exposing fine-grained dimensional structure. The coarse-to-fine escalation is **per-request, not per-session**—preventing "consent drift."

6.4 Bias & Fairness Guardrails

6.4.1 The Fundamental Risk

Psychographic vectors encode behavioral patterns. Behavioral patterns correlate with demographics. Demographics include protected classes. Any system optimizing on psychographic vectors faces unavoidable risk of *proxy discrimination*—targeting or excluding users based on protected attributes not through explicit intent but through mathematical correlation.

6.4.2 The Proxy Correlation Test

For each dimension d in the 512-dimensional vector and each protected-class proxy variable $\mathbf{p}_k$, compute the Pearson correlation coefficient:

$$r_{d,k} = \frac{\sum_{i=1}^{N}(x_{d,i} - \bar{x}_d)(p_{k,i} - \bar{p}_k)}{\sqrt{\sum_{i=1}^{N}(x_{d,i} - \bar{x}_d)^2 \cdot \sum_{i=1}^{N}(p_{k,i} - \bar{p}_k)^2}}$$

where $N \geq 10{,}000$ observations drawn via stratified sampling.

For a 512-dimensional vector tested against 4 proxy variables (the minimum required set: race, gender, religion, health), this produces $512 \times 4 = 2{,}048$ correlation coefficients. Disposition:

| $|r|$ Range | Disposition Code | Required Action | |:---|:---|:---| | < 0.10 | `BIAS_PASS` | Proceed. Dimension safe for all predictions. | | 0.10–0.20 | `BIAS_WARN` | Downweight dimension by factor $(1 - |r|)$. Log warning. | | 0.20–0.30 | `BIAS_QUARANTINE` | Zero out dimension. Log quarantine. Alert model owner. | | ≥ 0.30 | `BIAS_BLOCK` | Block entire BPF vector. File incident. Retrain required. |

6.4.3 Worked Example: Validating "Rationality"

A vendor publishes a BPF model where dimensions 128–191 encode "Rationality" (Cognitive Load domain). The bias audit produces:

| Dimension | Proxy Variable | $|r|$ | Disposition | |:---|:---|:---:|:---| | dim_142 (comparison_depth) | race | 0.03 | `BIAS_PASS` | | dim_155 (info_seeking) | religion | 0.07 | `BIAS_PASS` | | dim_161 (decision_latency) | gender | 0.14 | `BIAS_WARN` — downweight by 0.86 | | dim_177 (complexity_preference) | health | 0.04 | `BIAS_PASS` |

Dimension 161 receives `BIAS_WARN`: the model has learned a weak correlation between decision latency and gender. The dimension remains usable but with 14% reduced influence. This is flagged in the model card and reported in the bias audit log. If the correlation were 0.22 instead, the dimension would be quarantined (zeroed out entirely).

6.4.4 Structural Fairness Requirements

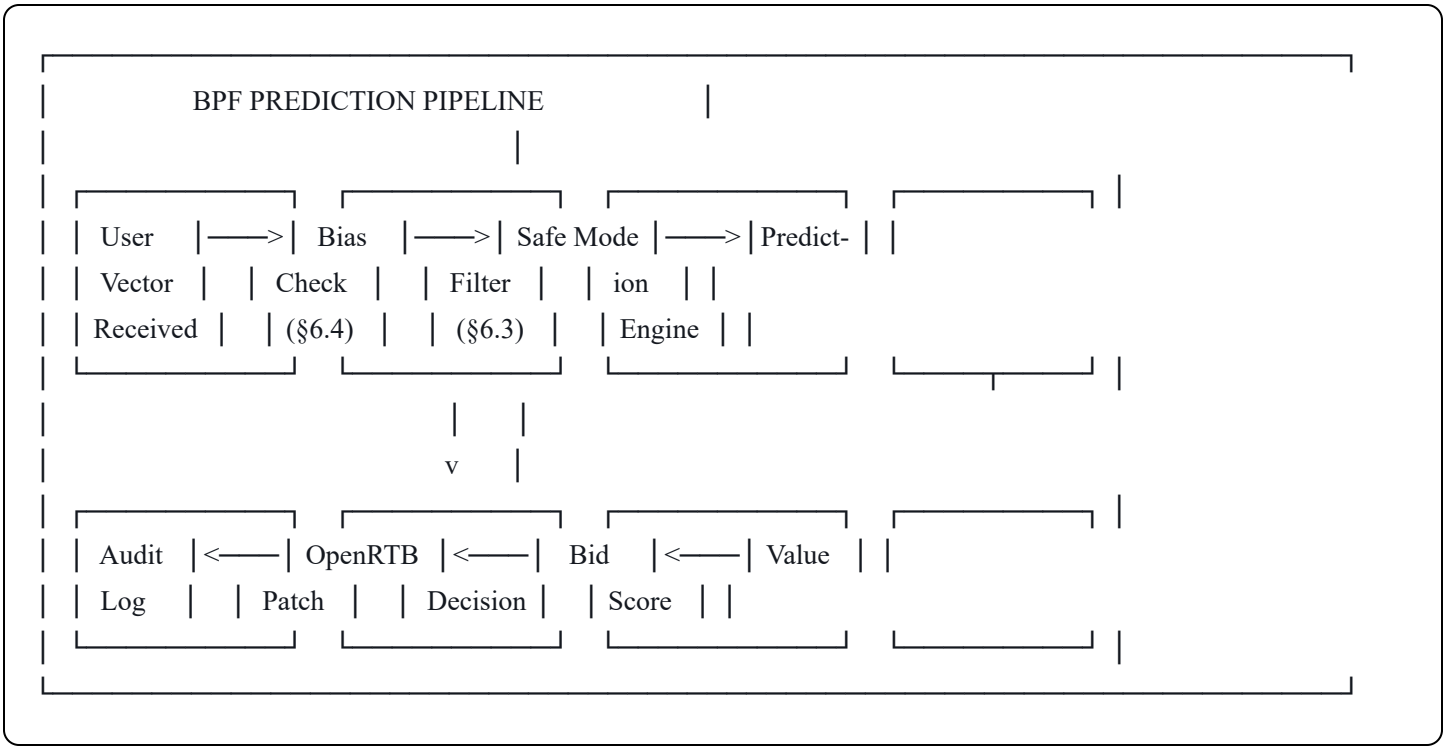
Demographic Parity: For any prediction threshold applied to BPF-based scores, the ratio of positive prediction rates between any two demographic groups must not exceed 1.25.

Equalized Odds: The maximum difference in True Positive Rate (TPR) or False Positive Rate (FPR) between any two demographic groups must not exceed 0.05.

Both tests MUST be re-run every 90 days or after any major model update, and results MUST be documented in a model card registered with the Agent Registry.

6.5 Operational Workflow: The Complete Prediction Loop

6.5.1 End-to-End Flow



6.5.2 Step-by-Step Execution

Step 1 — User Vector Received (0–2 ms). The agent receives a bid request containing a BPF payload within the UCP `user_embedding` field.

Step 2 — Bias Check (0.5–1 ms, cached). The agent verifies the `bias_audit` field: `last_test_date` within 90 days, `max_proxy_correlation` below 0.10, `disposition` is `BIAS_PASS`. In production, results are cached per `bpf_contract_version` and `space_id`.

Step 3 — Safe Mode Filter (0.5 ms). Based on consent state and trust level, the Safe Mode Protocol determines vector resolution. If only `coarse` authorized, dims 0–511 are replaced with 6 aggregate domain scores.

Step 4 — Prediction (5–10 ms). The model consumes the (possibly filtered) vector alongside contextual and reinforcement embeddings. Output: click probability, conversion probability, lifetime value estimate.

Step 5 — Value Scoring (0.5 ms). The agent computes expected value: $\text{EV} = P(\text{click}) \times P(\text{conv}|\text{click}) \times \text{LTV} - \text{cost}$.

Step 6 — Bid Decision (0.5 ms). If EV > threshold: generate bid. Construct OpenRTB Patch mutation.

Step 7 — Audit Log (0.5 ms, async). Emit audit record with `resolution_method`, `safe_mode_level`, `bias_check_result`, and `prediction_confidence`.

Total latency: 7.5–15 ms — well within the 50 ms RTB container budget.

6.5.3 Prediction Signal Schema

```
json
```

```
{
 "behavioral_prediction_signal": {
  "version": "2.0",
  "user_embedding": {
   "model_id": "ucp-foundation-v1.2",
   "space_id": "ucp-space-v1.0",
   "dimensions": 512,
   "identity_vector": [0.0234, -0.1567, "...512 dims..."],
   "contextual_vector": [0.1123, 0.0456, "...512 dims..."],
   "reinforcement_vector": [-0.0345, 0.2234, "...512 dims..."]
  },
  "predictions": {
   "click_probability": {
    "value": 0.032,
    "confidence": 0.89,
    "causal_factors": ["contextual_affinity_high", "intent_stage_mid_funnel"]
   },
   "conversion_probability": {
    "value": 0.0045,
    "expected_value_usd": 12.40
   },
   "lifetime_value": {
    "predicted_12m_usd": 285.00,
    "churn_risk_30d": 0.12
   },
   "optimal_frequency": {
    "current": 3,
    "optimal_range": [4, 7],
    "fatigue_threshold": 9
   }
  },
  "counterfactuals": {
   "creative_change_impact": { "estimated_lift": 0.15, "confidence": 0.72 },
   "timing_change_impact": { "estimated_lift": 0.08, "confidence": 0.81 }
  },
  "privacy_attestation": {
   "differential_privacy_epsilon": 1.0,
   "aggregation_level": "cohort_100plus",
   "tee_verified": true
  }
 }
}
```

6.6 Embeddings Interoperability — The Vector Alignment Problem

Buy-side agents, sell-side agents, and signal providers (measurement companies, data vendors) each train their own embedding models. If Agent A's "high intent" vector and Agent B's "high intent" vector inhabit different latent spaces, cosine similarity between them produces mathematical noise, not semantic meaning. This is the core challenge that Chapter 2's Vector Alignment Contract (VAC) solves.

The VAC's capability discovery (§2.3.2), projector negotiation (§2.3.3), and mathematical acceptance thresholds (§2.3.5: PassRate ≥ 0.95, MSE < 0.05, identity cosine ≥ 0.90) are what make BPF vectors interoperable across the buy, sell, and signal sides.

Three-Party Alignment Scenario: A **Buyer Agent** (DSP) uses `buyer-space-v2.0`. A **Seller Agent** (SSP/publisher) produces BPF vectors in `bpf-space-v1.0`. A **Signal Agent** (measurement company, e.g., an ACR-based TV measurement provider) produces reinforcement embeddings in `signal-space-v1.0`. During the VAC handshake, the three agents discover their incompatible spaces and negotiate learned projection matrices: $P_{B \to U}$ (buyer-to-UCP), $P_{S \to U}$ (seller-to-UCP), and $P_{\Sigma \to U}$ (signal-to-UCP). Each projector is validated against the GTS. Once all three projectors pass the acceptance thresholds, the agents compute meaningful cross-party similarity scores in the shared `ucp-space-v1.0` alignment space. The Hybrid Payload (§2.5) ensures that if any projector fails validation, all three agents can fall back to AdCOM segment IDs without transaction failure.

Chapter 7: Implementation Guide

7.1 Latency Budget (Hybrid Mode)

Step	Budget (ms)	Notes
Receive bid request	2	gRPC deserialization
Extract hybrid context	1	Parse embedding + legacy payload
VAC handshake check	0.5	Cached model compatibility
ANN first pass (binary)	0.8	Hamming distance, 1000 candidates
ANN rerank (float32)	2.4	Cosine similarity, top 50
Model inference	5–10	CTR/CVR prediction
Build OpenRTB Patch	1	RFC 6901 JSON Pointer paths
Serialize response	1	gRPC serialization
Total (embedding path)	**13.7–18.7**	Within 50 ms container budget

Step	Budget (ms)	Notes
Total (fallback path)	**8–12**	Faster — skip ANN search

7.2 Vector Storage Architecture

Recommended: HNSW (Hierarchical Navigable Small World) graphs for $O(\log n)$ search with >95% recall. Product Quantization (PQ) compresses 512-dim float32 vectors (2 KB) to 64 bytes. Dual-index architecture: a primary immutable index rebuilt daily and a secondary append-only index for real-time updates, merged during daily rebuilds.

7.3 Implications for Measurement Companies

For companies operating at the intersection of TV measurement and data-as-a-service, ACR-derived viewership data is ideally suited for UCP embeddings. A measurement agent could provide real-time reinforcement embeddings capturing household-level TV exposure patterns, enabling cross-screen optimization with unprecedented fidelity.

Chapter 8: Future Directions

8.1 Standards Convergence

The agentic advertising ecosystem converges around a clear architecture: MCP and A2A for agent communication, gRPC for high-performance execution, OpenRTB + AdCOM for transaction semantics, UCP for signal exchange, AP2 for economic settlement, and IETF RATS for trust verification. The IAB Agent Registry (March 1, 2026) will be critical infrastructure for all layers.

AP2 integration trajectory: As AP2 matures from its September 2025 launch, advertising-specific extensions will emerge. The three mandate types (Intent, Cart, Payment) map naturally to campaign authorization, line-item booking, and delivery-based settlement. The 60+ AP2 launch partners (Mastercard, PayPal, Coinbase, Adyen, American Express) provide the financial network connectivity that agent-to-agent advertising settlement requires.

8.2 Research Challenges

Base UCP Encoder standardization. An industry-standard encoder (the "ImageNet moment" for advertising embeddings) would dramatically reduce projector complexity. The `iab-techlab/ucp-distilled-6L-512d` reference model is the first step, but broader adoption requires multilingual support, cross-vertical training data, and federated fine-tuning protocols.

Embedding alignment across organizations. When multiple agents use different embedding models, projected alignment techniques (learning linear transformations between spaces) and shared foundation models are both

active research areas. The VAC provides the protocol infrastructure; the research challenge is improving projector quality.

Adversarial robustness. Embedding-based systems may be vulnerable to adversarial perturbations—small changes to embeddings that cause large changes in matching outcomes. Robust embedding training and anomaly detection (§4.1, T2) are critical for trust.

Causal inference in embedding spaces. Current embedding similarity captures correlation, not causation. Integrating causal AI methods—enabling counterfactual reasoning like "would this user have converted without this ad?"—represents the frontier of agentic optimization.

Federated embedding governance. Who owns derived embeddings? How are embedding models audited for bias? What constitutes "data" under privacy law when embeddings cannot be reversed?

Chapter 9: Governance & Operational Ownership [v5.0-NEW]

9.1 Scope

This chapter defines the organizational governance structure for normative artifacts, cryptographic trust anchors, and intellectual property rights governing the UCP specification and its operational infrastructure. All governance provisions in this chapter are normative for L2+ implementations.

9.2 Golden Test Set Ownership & Lifecycle

Owner: IAB Tech Lab, in collaboration with the UCP Working Group.

Governance model:

1. The Golden Test Set (GTS, §2.3.4) is a normative artifact owned and published by IAB Tech Lab.
2. The UCP Working Group—comprising representatives from IAB Tech Lab member organizations—is responsible for GTS content curation, bias review, and version approval.
3. GTS versions are published as immutable, versioned artifacts (e.g., `gts-v1.0`, `gts-v1.1`) with SHA-256 integrity hashes.
4. **Regeneration cadence:** Every 90 days, or immediately upon a major `space_id` change (§2.4).
5. **Approval process:** New GTS versions require a simple majority vote of the Working Group quorum (≥50% of voting members). Emergency regenerations (triggered by `space_id` changes) require only Working Group Chair approval, with ratification at the next regular meeting.
6. **Seed publication:** Every GTS version MUST be generated with a published seed for deterministic reproducibility (§2.3.4). The seed, generation script, and source distributions MUST be published alongside the GTS artifact.

7. **Backward compatibility:** A new GTS version MUST NOT invalidate projectors that passed the previous version, unless accompanied by a major `space_id` change. If a new GTS causes >5% of previously-passing projectors to fail, the Working Group MUST issue a 30-day grace period.

9.3 Registry Root Certificate Authority

Owner: IAB Tech Lab operates the Registry Root CA.

Trust hierarchy:

```
IAB Tech Lab Registry Root CA (offline, Ed25519)
  │
  ├── Registry Intermediate CA (online, Ed25519, 365-day rotation)
  │     ├── Agent Identity Certificates (per-agent, X.509, §Appendix I)
  │     └── Registry Response Signing Key (§I.3)
  │
  └── Emergency Revocation Key (offline, 2-of-3 threshold signing)
```

Operational requirements:

1. The Root CA private key MUST be stored offline in an HSM. Ceremony-based access only, minimum two authorized signatories.
2. The Intermediate CA signs Agent Identity Certificates and Registry response signatures. It MUST rotate every 365 days. The new certificate MUST be published ≥ 30 days before the old one expires.
3. Agent Identity Certificates are valid for the lesser of 365 days or the agent's registration period.
4. The Emergency Revocation Key is a 2-of-3 threshold key held by IAB Tech Lab's CTO, VP of Engineering, and an independent Working Group Chair.
5. CRLs MUST be published at `https://registry.iabtechlab.com/.well-known/crl.pem` and refreshed every 60 minutes.
6. OCSP responder availability: 99.9% uptime SLA.

9.4 Projector Signing Authority

Model: Federated signing with Registry-validated identity.

1. Agents sign their own projector artifacts (§2.3.3) using Ed25519 keys registered in the Agent Registry.
2. The Registry validates signing identity (X.509 chain → Intermediate CA → Root CA) but does NOT validate projector quality. Quality validation is the consuming agent's responsibility via GTS testing (§2.3.4–2.3.5).

3. Projector signing keys MUST rotate every 90 days. The Registry MUST reject projectors signed with expired keys.
4. If a signing agent's registration is revoked, all projectors signed by that agent MUST be treated as invalid within the SLO defined in §I.7 (≤120 seconds end-to-end).
5. **Cross-organizational projectors:** When Agent A publishes a projector mapping from Agent B's space to a target space, Agent A MUST sign the projector AND Agent B MUST countersign (dual-signature).

9.5 Working Group Structure

Role	Responsibility	Selection
Working Group Chair	Meeting facilitation, GTS approval, emergency decisions	Elected annually by WG members
Spec Editor	Normative text, schema maintenance, release management	Appointed by IAB Tech Lab
Security Reviewer	STRIDE model updates, TEE attestation validation	Rotating, from member organizations
Privacy Reviewer	GPP/TCF matrix updates, DP budget governance	Rotating, from member organizations
Certification Lead	Conformance test suite, benchmark validation	IAB Tech Lab staff

Meeting cadence: Biweekly during active spec development; monthly during maintenance periods.

9.6 Intellectual Property Rights — RAND-Z

This specification is developed under the **Reasonable and Non-Discriminatory, Zero-Cost (RAND-Z)** IPR policy of IAB Tech Lab.

1. **Patent commitment:** All contributors grant a royalty-free, worldwide, non-exclusive, irrevocable license under any Essential Claims to implement this specification.
2. **Scope:** The RAND-Z commitment covers all normative content including schemas, algorithms, protocol flows, and test vectors. It does NOT extend to specific implementations, optimizations, or proprietary extensions beyond normative requirements.
3. **Disclosure obligation:** Contributors MUST disclose any patents or patent applications that may contain Essential Claims within 60 days of becoming aware of them.
4. **Defensive termination:** The RAND-Z license granted by any contributor may be terminated with respect to any party that initiates patent litigation alleging that implementation of this specification infringes that

party's patents.

9.7 Specification Status

This specification is a **Normative Specification** (Final Corrected). It will advance to **Final Specification** upon demonstration of ≥2 independent implementations passing L2 conformance tests using the `ucp-vac-validator` tool against GTS v1.0.

Part VI — Appendices

Appendix A: Glossary

Term	Definition
AdCOM	Advertising Common Object Model. IAB Tech Lab standard for domain objects.
AdCP	Ad Context Protocol. MCP-based control plane for campaign orchestration.
ANN	Approximate Nearest Neighbor. Algorithm for fast vector similarity search.
AP2	Agent Payments Protocol. Google's VDC-based settlement framework.
ARTF	Agentic RTB Framework. Containerized agent execution plane.
BPF	Behavioral Prediction Framework. Psychographic vector contract (Ch. 6, App. J).
DP	Differential Privacy. Mathematical framework for privacy-preserving computation.
GTS	Golden Test Set. Normative acceptance oracle for projector validation (§2.3.4).
HNSW	Hierarchical Navigable Small World. Graph-based ANN index algorithm.
MCP	Model Context Protocol. Anthropic's agent communication standard.
PQ	Product Quantization. Vector compression technique.
TEE	Trusted Execution Environment. Hardware-isolated compute enclave.
UCP	Universal Content Protocol. This specification.
VAC	Vector Alignment Contract. Interoperability protocol for embedding exchange (§2.3).
VDC	Verifiable Digital Credential. Cryptographic mandate per AP2.

Appendix B: Specification References

Author's Prior Work

[1] Popov, E. "Embeddings: The Next Frontier in Advertising." LinkedIn, 2025. https://www.linkedin.com/pulse/embeddings-next-frontier-advertising-evgeny-popov-xvyde/

[2] Popov, E. "Behavioral Prediction Framework for Advertising." LinkedIn, 2025. https://www.linkedin.com/pulse/behavioral-prediction-framework-advertising-evgeny-popov-vhaue/

[3] Popov, E. "Four-Part Series: The Agentic Transformation of Digital Advertising." Medium, 2025. https://medium.com/@ev.popov/four-part-series-the-agentic-transformation-of-digital-advertising-6d20b1eb48cd

[4] Popov, E. "Blending Freytag's Pyramid and Plutchik's Wheel with MCP and Agentic AI." LinkedIn, 2025. https://www.linkedin.com/pulse/blending-freytags-pyramid-plutchiks-wheel-mcp-agentic-evgeny-popov-knnie/

[5] Popov, E. "Rethinking Marketing Relevance — From AI to Personalized Communication." DMEXCO Column, 2025. https://dmexco.com/stories/dmexco-column-rethinking-marketing-relevance-from-ai-to-personalized-communication/

[6] Popov, E. "Part 2: Pre-Campaign Planning in the Age of Agentic Intelligence." Medium, 2025. https://medium.com/@ev.popov/part-2-pre-campaign-planning-in-the-age-of-agentic-intelligence-from-static-assumptions-to-ba0febdbbe54

[7] Popov, E. "Part 3: Real-Time Execution Through Learning Policies." Medium, 2025. https://medium.com/@ev.popov/part-3-real-time-execution-through-learning-policies-the-millisecond-battlefield-where-strategy-b3b326eebcbd

Industry Specifications

[8] IAB Tech Lab — Agentic Audiences (UCP). https://github.com/IABTechLab/user-context-protocol

[9] IAB Tech Lab — Agentic RTB Framework v1.0. https://iabtechlab.com/standards/artf/

[10] IAB Tech Lab — Buyer Agent Reference. https://github.com/IABTechLab/buyer-agent

[11] IAB Tech Lab — ARTF Reference Implementation. https://github.com/IABTechLab/agentic-rtb-framework

[12] IAB Tech Lab — AdCOM v1.0. https://github.com/InteractiveAdvertisingBureau/AdCOM

[13] Ad Context Protocol (AdCP). https://github.com/adcontextprotocol/adcp

[14] IAB Tech Lab — Agentic Roadmap, January 2026. https://iabtechlab.com/

[15] IAB Tech Lab — Agent Registry, launching March 1, 2026.

[16] Google — Agent Payments Protocol (AP2). https://ap2-protocol.org/specification/

[17] IAB Tech Lab — GPP Specification. https://github.com/InteractiveAdvertisingBureau/Global-Privacy-Platform

[18] IAB Europe — TCF v2.2. https://iabeurope.eu/transparency-consent-framework/

[19] IETF — RATS Architecture (RFC 9334). https://www.rfc-editor.org/rfc/rfc9334

[20] IETF — Entity Attestation Token (EAT). https://datatracker.ietf.org/doc/draft-ietf-rats-eat/

[21] RFC 6901 — JSON Pointer. https://www.rfc-editor.org/rfc/rfc6901

Appendix C: Embedding Model Specifications

Parameter	Value	Notes
Base Model	iab-techlab/ucp-distilled-6L-512d	Canonical public interface
Dimensions	512	Standard tier
Normalization	L2	MUST for all exchange vectors
Training	6-layer distilled Transformer	Contrastive learning
Projection	Required for non-standard models	Linear, ≥ 0.90 quality
Update	Quarterly (base), daily (fine-tune)	Via Agent Registry
ANN index	PQ-64	64 bytes per vector
Binary	Supported	512 bits for Hamming

Appendix D: Release Checklist

#	Artifact	Format
1	ucp-spec-v5.1.md	Markdown
2	schemas/vac-contract.json	JSON Schema

#	Artifact	Format
3	schemas/audience-capability.json	JSON Schema
4	schemas/audience-validation.json	JSON Schema
5	schemas/coverage-calculator.json	JSON Schema
6	schemas/usage-record.json	JSON Schema
7	schemas/agent-registry-entry.json	JSON Schema
8	schemas/audit-record.json	JSON Schema
9	schemas/privacy-budget.json	JSON Schema
10	schemas/bpf-vector.json	JSON Schema
11	schemas/error-envelope.json	JSON Schema
12	schemas/golden-test-set-entry.json	JSON Schema
13	schemas/alignment-error-response.json	JSON Schema
14	schemas/agent-identity-did.json	JSON Schema
15	tests/vac-conformance-vectors.jsonl	JSONL (50+ vectors)
16	tests/golden-test-set-v1.0.jsonl	JSONL (1000+ pairs)
17	tests/privacy-enforcement-tests.jsonl	JSONL
18	tests/bias-proxy-vectors.jsonl	JSONL
19	tools/ucp-bench	CLI (Go)
20	tools/ucp-vac-validator	CLI (Python)
21	tools/ucp-bias-checker	CLI (Python)
22	examples/happy-path-workflow.json	JSON
23	examples/fallback-workflow.json	JSON
24	security/threat-model.md	Markdown

Appendix E: Benchmark Methodology — Recipe for Reproduction [v5.0-REWRITE]

E.1 Purpose

This appendix provides a fully reproducible benchmark methodology. All latency and recall claims in this specification (§7.1, §2.2) are contextualized against the Hardware Reference Profile defined here. Implementers MUST use this methodology when certifying RTB compatibility. Third parties MAY use alternative hardware but MUST disclose deviations and normalize results against the reference profile.

E.2 Hardware Reference Profile

Component	Reference Specification	Notes
Instance type	AWS `c6i.2xlarge`	Compute-optimized, 3rd-gen Intel Xeon (Ice Lake)
vCPU	8 vCPU (x86_64)	AVX-512 MUST be available and enabled
Memory	16 GB DDR4	Sufficient for 10M × 512d float32 index in-memory
Storage	gp3 EBS, 3000 IOPS	Index MUST be fully loaded into memory before benchmarking
Network	Up to 12.5 Gbps	Not a factor for local index benchmarks
OS	Amazon Linux 2023 or Ubuntu 22.04 LTS	Kernel ≥ 5.15
Compiler flags	`-O2 -mavx512f -mavx512bw`	AVX-512 intrinsics MUST be enabled

Deviation Disclosure: If benchmarking on non-reference hardware, the report MUST include: instance type, CPU model and ISA extensions, memory capacity, and a normalization factor computed as `(reference_p50 / observed_p50)` on the standard 10K query set.

E.3 Index Parameters (HNSW)

Parameter	Value	Rationale
`M`	16	Graph connectivity. Balances recall and memory. Industry standard for 512d vectors.
`efConstruction`	200	Build-time search width. Higher values improve index quality at cost of build time.

Parameter	Value	Rationale
efSearch	50	Query-time search width. Tuned to achieve Recall@50 ≥ 0.92 within latency budget.
Distance metric	L2 (internal)	Cosine similarity equivalent on L2-normalized vectors.
Quantization (first pass)	PQ-64	Product Quantization with 64 sub-quantizers (64 bytes/vector).
Quantization (rerank)	float32	Full-precision reranking of top-K candidates from PQ pass.

Two-Pass Search Architecture:

1. **Pass 1 (Candidate Generation):** PQ-64 compressed index. Hamming distance or asymmetric distance computation. Returns top-1000 candidates. Target: $< 1\text{ms}$.
2. **Pass 2 (Reranking):** Load float32 vectors for top-1000 candidates. Compute exact cosine similarity. Return top-50. Target: $< 3\text{ms}$.

E.4 Dataset Characteristics

Property	Value
Index size	10,000,000 vectors
Dimensions	512
Distribution	Gaussian mixture with 500 clusters (synthetic, seed=42)
Query set	10,000 vectors (separate from index, same distribution)
Ground truth	Exact brute-force top-50 for each query
Sparsity	Dense (all dimensions populated)

E.5 Pass/Fail Criteria ("RTB Compatible")

Metric	Target	Level
ANN search p50	≤ 2 ms	**MUST**
ANN search p95	≤ 5 ms	**MUST**

Metric	Target	Level
ANN search p99	≤ 10 ms	**MUST**
Full tool response p50	≤ 15 ms	**MUST**
Full tool response p95	≤ 30 ms	**MUST**
Full tool response p99	≤ 50 ms	**MUST**
Recall@50 (from 10M index)	≥ 0.92	**MUST**
Recall@100 (from 10M index)	≥ 0.95	**SHOULD**

An implementation is "RTB Compatible" if and only if ALL MUST targets pass over ≥ 10,000 sequential queries, Recall@50 ≥ 0.92 over the full test set, and no single query exceeds 100 ms (hard ceiling).

E.6 Warmup Protocol

Before measurement begins, the benchmark harness MUST execute ≥ 1,000 warmup queries that are excluded from results. This eliminates JIT compilation and cache cold-start artifacts. Latency is measured as wall-clock time via monotonic clock (e.g., `clock_gettime(CLOCK_MONOTONIC)`). Results MUST report p50, p95, p99, and max.

E.7 CLI Harness

```bash
ucp-bench run \
  --index-path ./10M.hnsw \
  --queries-path ./10K.jsonl \
  --dims 512 \
  --quantization pq64 \
  --ef-search 50 \
  --top-k 50 \
  --warmup-queries 1000 \
  --output-format json \
  --output-path ./results.json \
  --seed 42
```

Input (`10K.jsonl`, one object per line):

```json
```

{"query_id": "q-0001", "vector": [0.0234, -0.1567, 0.0891, "...512 dims..."], "ground_truth_ids": ["v-8821", "v-3301", "...up t

Output (`results.json`):

```json
```

```
{
 "run_id": "bench-20260215-001",
 "spec_version": "5.1",
 "hardware_profile": {
  "instance_type": "c6i.2xlarge",
  "vcpu": 8,
  "memory_gb": 16,
  "avx512_enabled": true
 },
 "index_params": {
  "M": 16,
  "efConstruction": 200,
  "efSearch": 50,
  "quantization_pass1": "pq64",
  "quantization_pass2": "float32"
 },
 "dataset": {
  "index_size": 10000000,
  "dims": 512,
  "distribution": "gaussian_mixture_500",
  "seed": 42,
  "query_count": 10000,
  "warmup_queries": 1000
 },
 "latency": {
  "p50_ms": 1.8,
  "p95_ms": 4.2,
  "p99_ms": 8.1,
  "max_ms": 42.3
 },
 "recall": {
  "at_50": 0.934,
  "at_100": 0.961
 },
 "queries_run": 10000,
 "pass": true,
 "failures": []
}
```


E.8 Certification Submission

Benchmark results submitted for L1+ certification MUST include the full `results.json` output. The IAB Tech Lab certification body will cross-reference `hardware_profile` and `index_params` against reference values.

Deviations MUST be accompanied by the normalization factor described in §E.2.

Appendix F: Security Threat Model & TEE Verification

F.1 Threat Classes

This appendix provides the expanded STRIDE-based threat model. The summary table is in §4.1; full mitigations are below.

#	Threat	Vector	Impact	Mitigation (MUST)
T1	**Embedding Poisoning**	Malicious agent sends adversarial vectors to manipulate matching	Mis-targeting, wasted spend	Validate incoming embeddings against distribution bounds. Flag vectors $> 3\sigma$ from population mean on any dimension.
T2	**Replay Attack**	Attacker re-sends captured payloads	Budget theft, stale targeting	`request_id` (UUIDv7) + `timestamp`. Reject requests older than 60s. Reject duplicate `request_id` within TTL.
T3	**Embedding Inversion**	Adversary reconstructs raw features from embeddings	Privacy breach	DP ($\varepsilon \leq 2.0$) MUST be applied. Cohort floor: 100 users. TEE MUST verify DP runs inside enclave.
T4	**Membership Inference**	Adversary determines if specific user is in segment	Privacy breach	MUST NOT expose per-user embeddings. Minimum cohort size: 100.
T5	**Projector Spoofing**	Malicious projector subtly distorts matching	Mis-targeting	Projectors MUST be JWS-signed. Consumers MUST verify AND run full GTS validation locally.
T6	**Registry Manipulation**	Attacker modifies agent entries	Capability escalation	Append-only audit log. All mutations signed by operator key. Clients MUST verify response signatures.
T7	**Stale Capability Claims**	Agent presents outdated capabilities	Trust violation	`last_updated` MUST be present. Reject if > 3600s old. Registry lookup MUST for high-risk ops.

F.2 TEE Verification

See §4.2 for the full normative verification pseudocode and §4.3 for signing requirements and rotation schedules.

F.3 Container Security

All agent containers at L3+ MUST be signed with Cosign or Notary v2. The signed digest MUST be registered in the Agent Registry. Receivers MUST verify the container signature before accepting any tool invocations.

Appendix G: Privacy Enforcement

G.1 GPP/TCF Decision Matrix

See §4.4 for the normative enforcement matrix governing identity embedding exchange, contextual embedding exchange, reinforcement signal feedback, and CoverageCalculator operations under Sale Opt-Out, Targeted Ad Opt-Out, No TCF Purpose 1, and No TCF Purpose 4 conditions.

G.2 Differential Privacy Budget Governance

See §4.5 for the normative privacy budget schema, sequential composition accounting, exhaustion degradation ladder, and hash-chained verification records.

G.3 Normative Test Vectors [v5.0-NEW]

Implementations at L4+ MUST pass all test vectors below. Implementations at L1–L3 SHOULD pass these vectors for forward compatibility.

G.3.1 GPP US National Test Vectors

Vector ID	GPP String	Field Tested	Opt-Out Value	Operation	Identity
PTV-001	DBACNYA~CPXxRfAPXxRfAAfKABENB-CgAAAAAAAAAAYg	Baseline (all consented)	None	AudienceValidation	ALLOW
PTV-002	DBACNYA~CPXxRfAPXxRfAAfKABENB-CgABAAAAAAAAYg	SaleOptOut (field 3)	1	AudienceValidation	**BLOCK**
PTV-003	DBACNYA~CPXxRfAPXxRfAAfKABENB-CgAABAAAAAAAYg	SharingOptOut (field 4)	1	AudienceValidation	**BLOCK**
PTV-004	DBACNYA~CPXxRfAPXxRfAAfKABENB-CgAAAABAAAAAAAYg	TargetedAdOptOut (field 5)	1	AudienceValidation	**BLOCK**
PTV-005	DBACNYA~CPXxRfAPXxRfAAfKABENB-CgAAAAABAAAAAAAYg	SensitiveData[0] (racial/ethnic)	1	AudienceValidation	**BLOCK**
PTV-006	DBACNYA~CPXxRfAPXxRfAAfKABENB-CgAAAAAAAABAAAAYg	KnownChild[0]	1	AudienceValidation	**BLOCK**
PTV-007	DBACNYA~CPXxRfAPXxRfAAfKABENB-CgABBAAAAAAAYg	Sale + Sharing OptOut	Both 1	CoverageCalculator	**BLOCK**
PTV-008	DBACNYA~CPXxRfAPXxRfAAfKABENB-CgABBABAAAAAAAYg	Sale + Sharing + Targeted	All 1	AudienceValidation	**BLOCK**

G.3.2 TCF v2.2 Test Vectors

Vector ID	TCF String	Purpose Tested	Consented?	Operation
PTV-101	CPXxRfAPXxRfAAGABCENATEIAACAAAAAAAAAAA	All purposes (1–10)	Yes	AudienceValic
PTV-102	CPXxRfAPXxRfAAGABCENATEIAABAAAAAAAAAAA	Purpose 1 (Store/access)	**No**	AudienceValic
PTV-103	CPXxRfAPXxRfAAGABCENATEIAACAAAAAABAAAAA	Purpose 4 (Personalized ads)	**No**	AudienceValic
PTV-104	CPXxRfAPXxRfAAGABCENATEIAACAAAAAABAAAAA	Purpose 4	**No**	CoverageCalc
PTV-105	CPXxRfAPXxRfAAGABCENATEIAABAAAAAABAAAAA	Purpose 1 + Purpose 4	Both **No**	AudienceValic

G.3.3 Combined GPP + TCF Test Vectors

Vector ID	Scenario	Expected Outcome
PTV-201	GPP SaleOptOut=1, TCF all consented	GPP more restrictive → Identity: **BLOCK**
PTV-202	GPP all consented, TCF Purpose 1 not consented	TCF more restrictive → All signals: **BLOCK**
PTV-203	GPP TargetedAdOptOut=1, TCF Purpose 4 not consented	Both restrict → Identity **BLOCK**, Contextual **BLOCK** (most restrictive wins)
PTV-204	GPP KnownChild=1, TCF all consented	Child protection override → All signals: **BLOCK**
PTV-205	GPP all consented, TCF all consented	Full consent → All signals: ALLOW

G.3.4 Edge Cases

Vector ID	Scenario	Expected Behavior
PTV-301	GPP empty/missing, TCF present and fully consented	ALLOW — absence of GPP does not block
PTV-302	GPP consented, TCF empty/missing	Jurisdiction-dependent. EU traffic: **BLOCK**. US-only: ALLOW.
PTV-303	Both GPP and TCF empty/missing	**BLOCK** — fail-closed per §4.4
PTV-304	GPP string malformed (unparseable)	**BLOCK** — MUST NOT assume consent
PTV-305	TCF string version 1.x (deprecated)	**BLOCK** — only TCF v2.2+ supported
PTV-306	`consent.captured_at` older than `max_age_seconds`	**BLOCK** — stale consent
PTV-307	`consent_hash` does not match SHA-256(gpp‖tcf)	**BLOCK** — integrity failure, possible tampering

Normative requirement: Implementations MUST produce deterministic BLOCK/ALLOW outcomes for each test vector.

Appendix H: Economic Settlement — UsageRecord

H.1 Billable Units

Unit	Description	Applies To
`compute_ms`	Compute time consumed	All tools
`vector_ops_count`	Vectors compared	SemanticMatchingEngine
`enclave_ms`	TEE compute time	All TEE-mode tools
`bytes_processed`	Data volume processed	CoverageCalculator

H.2 UsageRecord Schema

```
json
```

```
{
 "usage_record": {
  "record_id": "ur-20260215-001",
  "idempotency_key": "01945f3a-0002-7d1e-8a3f-9c0d1e2f3a4b",
  "tool_name": "AudienceValidation",
  "timestamp": "2026-02-15T14:30:01Z",
  "buyer_agent_id": "iab-reg-buyer-agent-acme",
  "seller_agent_id": "iab-reg-seller-agent-premium-pub",
  "billable_units": {
   "compute_ms": { "quantity": 12, "unit_price_usd": 0.0001, "total_usd": 0.0012 },
   "vector_ops_count": { "quantity": 512, "unit_price_usd": 0.000001, "total_usd": 0.000512 },
   "enclave_ms": { "quantity": 12, "unit_price_usd": 0.0002, "total_usd": 0.0024 },
   "bytes_processed": { "quantity": 4096, "unit_price_usd": 0.0000001, "total_usd": 0.0004 }
  },
  "pricing": { "total_cost_usd": 0.004512 },
  "ap2_cart_mandate_id": "mandate-cart-val-20260215-001",
  "sla": { "max_response_ms": 50, "actual_response_ms": 12, "sla_met": true },
  "proof": {
   "proof_refs": ["audit://seller.example/logs/01945f3a-0002"],
   "log_hash": "sha256:abcdef1234567890",
   "attestation_ref": "iab-reg-seller/attestation/2026-02-15"
  },
  "dispute": {
   "disputable_until": "2026-02-22T14:30:01Z",
   "dispute_status": null,
   "dispute_evidence_required": ["proof_refs", "log_hash", "attestation_ref"]
  }
 }
}
```


H.3 Pricing Discovery

json

```
{
 "tariff": {
  "seller_agent_id": "iab-reg-seller-agent-premium-pub",
  "effective_from": "2026-02-01T00:00:00Z",
  "effective_until": "2026-03-01T00:00:00Z",
  "tools": {
   "AudienceValidation": {
    "pricing_model": "per_call",
    "unit_price_usd": 0.002,
    "sla_tiers": {
     "standard": { "max_response_ms": 50, "price_multiplier": 1.0 },
     "premium": { "max_response_ms": 20, "price_multiplier": 1.5 }
    }
   },
   "CoverageCalculator": {
    "pricing_model": "per_call_plus_data",
    "base_price_usd": 0.005,
    "per_mb_usd": 0.001
   }
  },
  "currency": "USD",
  "signature": "<JWS-signed-by-seller>"
 }
}
```


Retroactive price changes MUST NOT be applied to already-committed Cart Mandates.

H.4 Anti-Fraud Controls

Field	Purpose	Req.
`nonce`	One-time replay prevention (UUIDv4)	**MUST**
`attestation_ref`	TEE proof of compute	**MUST** (L3+)
`log_hash`	SHA-256 of audit log entry	**MUST**
`idempotency_key`	Matches triggering `request_id`	**MUST**
`timestamp_range`	Computation start/end time	**SHOULD**

Receivers MUST reject UsageRecords where `nonce` matches a previously seen nonce within 24 hours, `idempotency_key` does not correspond to a known tool invocation, or `timestamp_range.end - timestamp_range.start`

exceeds 5× the declared `compute_ms`.

H.5 Dispute Workflow

Step	Actor	Deadline	Action
1	Seller	Immediate	Emit UsageRecord with all anti-fraud fields
2	—	Immediate	Dispute window opens
3	Buyer	7 days	Submit `dispute_evidence` package
4	Seller	3 days after dispute	Provide counter-evidence
5	System	Day 10	Auto-resolve in Buyer's favor if no response
6	Third party	Day 14	Escalate to arbitrator per AP2 framework

Dispute Reason Codes: `SLA_BREACH` · `DUPLICATE_CHARGE` · `UNAUTHORIZED_CHARGE` · `QUALITY_FAILURE` · `ATTESTATION_MISMATCH`

H.6 Billable Unit Tolerances

Unit	Acceptable Variance	Verification Method
`compute_ms`	±10%	Cross-reference audit record `response_time_ms`
`vector_ops_count`	Exact match	Deterministic from input dimensions
`enclave_ms`	±5%	TEE-attested timing log
`bytes_processed`	±1%	Input payload size verification

Appendix I: Agent Registry Specification

I.1 AgentRegistryEntry Schema

```
json
```

```
{
 "agent_registry_entry": {
  "agent_id": "iab-reg-seller-agent-premium-pub",
  "agent_type": "seller",
  "organization": {
   "name": "Premium Publisher Inc.",
   "iab_member_id": "IAB-ORG-4521"
  },
  "status": "active",
  "agent_identity": {
   "primary": {
    "type": "x509",
    "certificate_chain_pem": ["-----BEGIN CERTIFICATE-----"],
    "subject_cn": "iab-reg-seller-agent-premium-pub.iabtechlab.com"
   },
   "alternative": {
    "type": "did",
    "did": "did:web:premium-pub.com:agents:seller-001"
   }
  },
  "capabilities": {
   "supported_tools": ["AudienceCapability", "AudienceValidation", "CoverageCalculator"],
   "supported_models": [{
    "model_id": "ucp-foundation-v1.2",
    "space_id": "ucp-space-v1.0",
    "dimensions": [256, 512],
    "conformance_test_passed": "2026-02-01"
   }],
   "projector_support": {
    "can_provide_for": ["ucp-space-v1.0"],
    "can_consume_from": ["ucp-space-v1.0"]
   }
  },
  "keys": {
   "signing_key": {
    "algorithm": "Ed25519",
    "public_key_pem": "-----BEGIN PUBLIC KEY-----",
    "key_id": "key-seller-2026-001",
    "expires_at": "2026-08-15T00:00:00Z"
   }
  },
  "attestation": {
   "tee_platform": "aws_nitro_enclave",
```

```
    "expected_pcr0": "sha384:e7f8a9b0c1d2",
    "expected_enclave_hash": "sha384:9a8b7c6d5e4f",
    "container_image_digest": "sha256:abc123def456"
   },
   "revocation": { "revoked": false, "revocation_reason": null },
   "lifecycle_policy": {
    "deprecation_window_days": 90,
    "min_supported_version": "ucp-space-v1.0",
    "rollback_supported": true
   },
   "reputation": { "conformance_test_streak": 47, "uptime_30d_pct": 99.97 }
  }
}
```


I.2 Required Verification Steps

1. Resolve `agent_id` → `AgentRegistryEntry` via `GET /v1/agents/{agent_id}`.
2. Verify `status == "active"` and `revoked == false`.
3. Verify mTLS certificate matches `agent_identity`.
4. Verify `supported_models` includes a compatible model.
5. If TEE required: verify attestation per §4.2.
6. If **ANY** step fails: **MUST NOT** proceed.

I.3 `/lookup` API

Request:

```http
GET /v1/agents/iab-reg-seller-agent-premium-pub HTTP/1.1
Host: registry.iabtechlab.com
Accept: application/json
If-None-Match: "v5.1-sha256:abc123"
Authorization: Bearer <mtls-derived-token>
```

Response:

```http
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: max-age=300, stale-while-revalidate=60
ETag: "v5.1-sha256:def456"
X-Registry-Signature: <base64-Ed25519-signature-of-body>
X-Registry-Signed-At: 2026-02-15T14:30:00Z
```

Headers (MUST): `Cache-Control: max-age=300` · `X-Registry-Signature` (Ed25519) · `X-Registry-Signed-At` (ISO 8601).

Clients MUST verify `X-Registry-Signature`. Stale responses (300–3600s) MAY be used for low-risk ops. Beyond 3600s: treat as unreachable.

Batch requests:

```
http

POST /v1/agents/batch HTTP/1.1
Content-Type: application/json

{"agent_ids": ["iab-reg-seller-agent-premium-pub", "iab-reg-seller-agent-news-co"]}
```

Maximum batch size: 50. Missing agents represented as `{"agent_id": "...", "status": "not_found"}`.

I.4 Revocation Propagation

Registry MUST publish via polling (ETag) and webhook. Agents on webhook MUST process within 60 seconds. Polling agents MUST check every 300 seconds during active transactions.

I.5 Caching Tiers & Freshness

Tier	Max Age	Stale-While-Revalidate	Use Case
Hot	60s	30s	Active bidding sessions
Warm	300s	60s	Default for most operations
Cold	3600s	300s	Low-frequency lookups

Agents MUST use Hot tier for any counterparty with whom they are actively exchanging embeddings.

I.6 Outage Modes

Mode	Condition	Behavior
Fail-Closed	Registry unreachable AND no cached state, OR cache > 3600s	MUST NOT process embeddings. MAY use `legacy_fallback`. Return `AGENT_NOT_REGISTERED`.
Brownout	Registry unreachable BUT valid cache < 3600s	High-risk ops (identity, AudienceValidation): MUST fail-closed. Low-risk ops (contextual, CoverageCalculator aggregate): MAY proceed. Log `REGISTRY_STALE`.
Degraded	Registry reachable but slow ($> 2\times$ normal p99)	Extend cache TTLs by $2\times$. MUST NOT skip verification for high-risk ops.

I.7 Revocation SLOs

SLO	Target	Measurement
Webhook delivery latency	≤ 5 seconds from Registry commit	p99 measured by Registry
Agent processing time	≤ 60 seconds from webhook receipt	Agent-reported
End-to-end propagation	≤ 120 seconds	Registry commit → agent enforcement
Polling detection	≤ 300 seconds	Worst case for polling-only agents
Cache invalidation on revocation	Immediate upon webhook receipt	Agent MUST evict from all tiers

If an agent receives a revocation webhook for a counterparty during an active transaction, the agent MUST complete any in-flight request but MUST NOT initiate new embedding exchanges with the revoked counterparty.

Appendix J: BPF Psychographic Vector Contract

J.1 Vector Semantics

Dim Range	Domain	Signals
0–63	Attention	Dwell time, scroll depth, gaze proxy

Dim Range	Domain	Signals
64–127	Emotional	Sentiment, creative resonance, arousal
128–191	Cognitive	Content complexity, decision latency, comparison depth
192–255	Intent	Funnel stage, comparison shopping, urgency
256–383	Contextual	Category preference, temporal patterns, cross-device
384–511	Response	Ad interaction, conversion propensity, frequency response

J.2 Confidence & Drift

Every BPF vector MUST include overall and per-domain confidence, `drift_flag` (set `true` when `drift_score` exceeds `drift_threshold`), and uncertainty method.

```json
{
  "bpf_payload": {
    "bpf_contract_version": "1.0",
    "space_id": "bpf-space-v1.0",
    "safe_mode": "standard",
    "vector": [0.0234, -0.1567, "...510 more..."],
    "confidence": {
      "overall": 0.82,
      "per_domain": { "attention": 0.91, "emotional": 0.74, "cognitive": 0.68, "intent": 0.85, "contextual": 0.89, "response": 0.78
    },
    "uncertainty": { "method": "mc_dropout", "n_samples": 50, "std_dev_mean": 0.04 },
    "drift_flag": false,
    "drift_score": 0.03,
    "bias_audit": { "last_test_date": "2026-02-01", "max_proxy_correlation": 0.07, "disposition": "BIAS_PASS", "test_sample_
    "audit_hooks": { "request_id": "01945f3a-0005-7d1e", "sender_agent_id": "iab-reg-seller", "tee_attested": true }
  }
}
```

J.3 Permissible & Forbidden Constructs

Permissible:

Construct	Domain	Justification
Decision latency	Cognitive	Observable behavioral pattern
Content complexity preference	Cognitive	Derived from content interaction
Ad format response curve	Response	Campaign optimization signal
Scroll depth distribution	Attention	Engagement measurement

Forbidden (MUST NOT encode):

Construct	Reason
Individual identity	Privacy; BPF is cohort-level only
Protected class membership	Discrimination risk
Health condition indicators	Sensitive data per GDPR Art. 9
Financial status proxies	Fair lending regulations
Political affiliation	Electoral interference risk

J.4 Bias & Fairness Tests (MUST)

Before deploying a BPF model, the publisher MUST run demographic parity tests (max disparity ratio ≤ 1.25), equalized odds tests (max TPR/FPR gap ≤ 0.05), and document results in a model card registered with the Agent Registry. Tests MUST re-run every 90 days or after any major model update. See §6.4 for full Pearson correlation test methodology and worked example.

J.5 Safe Modes

Mode	Dimensions Shared	Use Case
coarse	6 aggregate domain scores	Low-trust or no-consent contexts
standard	Full 512-dim vector	Normal consented operation
fine_grained	512-dim + per-domain confidence + uncertainty	High-trust, TEE-attested

Default: ALWAYS coarse. Escalation per §6.3.

J.6 Audit Hooks

Every BPF vector exchange MUST log: `request_id`, `sender_agent_id`, `receiver_agent_id`, `safe_mode` used, `consent_state` (GPP + TCF), `drift_flag` state, `bias_test_date` of producing model, and `tee_attested` (boolean).

Appendix K: Implementer Checklists [v5.0-NEW]

K.1 Level 1 — Core Compliance Checklist

#	Requirement	Spec Reference	Verification Method
☐	Include `legacy_fallback` in ALL UCP exchanges	§2.5	Schema validation: `legacy_fallback` field present and non-empty
☐	Support `request_id` (UUIDv7) idempotency	§3.0	Send duplicate `request_id`; verify identical responses
☐	Produce audit records for every tool invocation	§1B.4, §3.0	Inspect audit log; verify all 12 required fields present
☐	Reject requests with `timestamp` older than 60s	§3.0, T5	Submit stale request; verify `REQUEST_EXPIRED`
☐	Reject duplicate `request_id` within 300s TTL	§3.0	Submit duplicate; verify `DUPLICATE_REQUEST`
☐	Include `gpp_string` and `tcf_consent_string` in all requests	§4.4	Schema validation on all outgoing requests
☐	Implement common error envelope format	§3.0	Verify error responses match schema
☐	Support `resolution_policy` field in hybrid payload	§2.5	Test all three policy values
☐	Retain audit logs ≥ 90 days	§1B.4	Verify log retention policy
☐	Hash-chain audit log entries (SHA-256)	§1B.4	Verify `entry_hash` includes `prev_hash`

K.2 Level 3 — Secure Compliance Checklist

Includes all L1 and L2 requirements.

#	Requirement	Spec Reference	Verification Method
☐	**[L2]** Implement full VAC handshake	§2.3.2–2.3.7	Passes `ucp-vac-validator` against GTS v1.0
☐	**[L2]** Register in Agent Registry	§I.1–I.2	Valid `AgentRegistryEntry` retrievable via `/v1/agents/{agent_id}`
☐	**[L2]** GTS: identity pair cosine ≥ 0.90 for ALL pairs	§2.3.5	`ucp-vac-validator` output
☐	**[L2]** GTS: PassRate(0.10) ≥ 0.95	§2.3.5	`ucp-vac-validator` output
☐	**[L2]** GTS: MSE(P) < 0.05	§2.3.5	`ucp-vac-validator` output
☐	Run all operations within TEE	§4.2, §1B.1	TEE attestation document with valid certificate chain
☐	Present attestation with valid nonce (≤ 60s)	§4.2	Automated nonce freshness and chain validation
☐	Enclave image hash matches Registry entry	§4.2 (Step 3)	Cross-reference against `expected_enclave_hash`
☐	PCR values match Registry entry	§4.2 (Step 4)	Automated verification per §4.2 pseudocode
☐	Sign all projectors with JWS/Ed25519	§2.3.3	Verify signature using signer's public key
☐	Rotate projector signing keys every 90 days	§2.3.3	`expires_at` ≤ 90 days from `created_at`
☐	Use mTLS rooted in Registry CA	§9.3	Certificate chain terminates at IAB Tech Lab Root CA
☐	Sign container images (Cosign/Notary v2)	§4.3	Verify container signature
☐	Reject expired/invalid projector signatures	§2.3.3	Submit expired projector; verify rejection
☐	Invalidate cached projectors from revoked signers	§9.4	Simulate revocation; verify cache eviction ≤ 120s

K.3 Level 5 — Billing Compliance Checklist

Includes all L1–L4 requirements.

#	Requirement	Spec Reference	Verification Method
☐	**[L4]** Enforce GPP/TCF matrix for all operations	§4.4	Passes all Normative Privacy Test Vectors (§G.3)
☐	**[L4]** Implement DP budget governance with hash-chain	§4.5	Verify `entry_hash = SHA-256(prev_hash‖request_id‖ε‖timestamp)`
☐	**[L4]** Attest consent enforcement inside TEE	§1B.4, §4.4	Attestation includes `consent_enforcement: true`
☐	**[L4]** Pass privacy enforcement test suite	§G.3	All PTV vectors return expected outcomes
☐	Emit UsageRecords for every billable invocation	§H.1–H.2	Verify all billable units populated
☐	UsageRecords include all anti-fraud fields	§H.4	Schema validation against `usage-record.json`
☐	Support AP2 Intent → Cart → Payment lifecycle	§1.4, App. H	End-to-end test: create Intent, book Cart, confirm, trigger Payment
☐	Cart sum does not exceed Intent budget	§1.4	Attempt Cart exceeding budget; verify rejection
☐	Implement dispute workflow per §H.5	§H.5	Submit dispute; verify 7-day window and escalation
☐	Support tariff discovery per §H.3	§H.3	Verify signed tariff is retrievable and JWS-valid
☐	No retroactive price changes on committed Carts	§H.3	Update tariff; verify pre-existing Carts retain original pricing
☐	Billable unit tolerances within spec	§H.6	Cross-reference quantities against audit log

Conformance Test Vectors

Implementations MUST pass these deterministic test vectors:

#	Scenario	Expected Result
1	Identical vectors, same space	$\cos = 1.0$, MODEL_MATCH
2	Orthogonal vectors, same space	$\cos = 0.0$
3	Different space_id, no projector	PROJECTOR_REQUIRED → FALLBACK_TRIGGERED
4	Projector applied, rank preservation = 0.84	ALIGNMENT_BELOW_THRESHOLD
5	GPP targeted_ad_opt_out = true	CONSENT_BLOCKED
6	Projector MSE = 0.062	ALIGNMENT_BELOW_THRESHOLD, MSE in measured_metrics
7	Registry unreachable, high-risk operation	MUST fail-closed
8	BPF dimension $	r
9	Stale capability entry (4,000s old)	Reject, re-lookup required

---

This specification extends the author's prior work on embeddings infrastructure [1], behavioral prediction [2], the agentic transformation series [3][6][7], MCP-native creative intelligence [4], and relevance realization theory [5]. The Vector Alignment Contract, Golden Test Set, Hybrid Payload Architecture, TEE Attestation integration, privacy budget governance, and BPF Psychographic Vector Contract are original proposals intended to advance the IAB Tech Lab community discussion.

END OF DOCUMENT — Agentic Audiences (UCP) v5.1 Final Corrected Specification