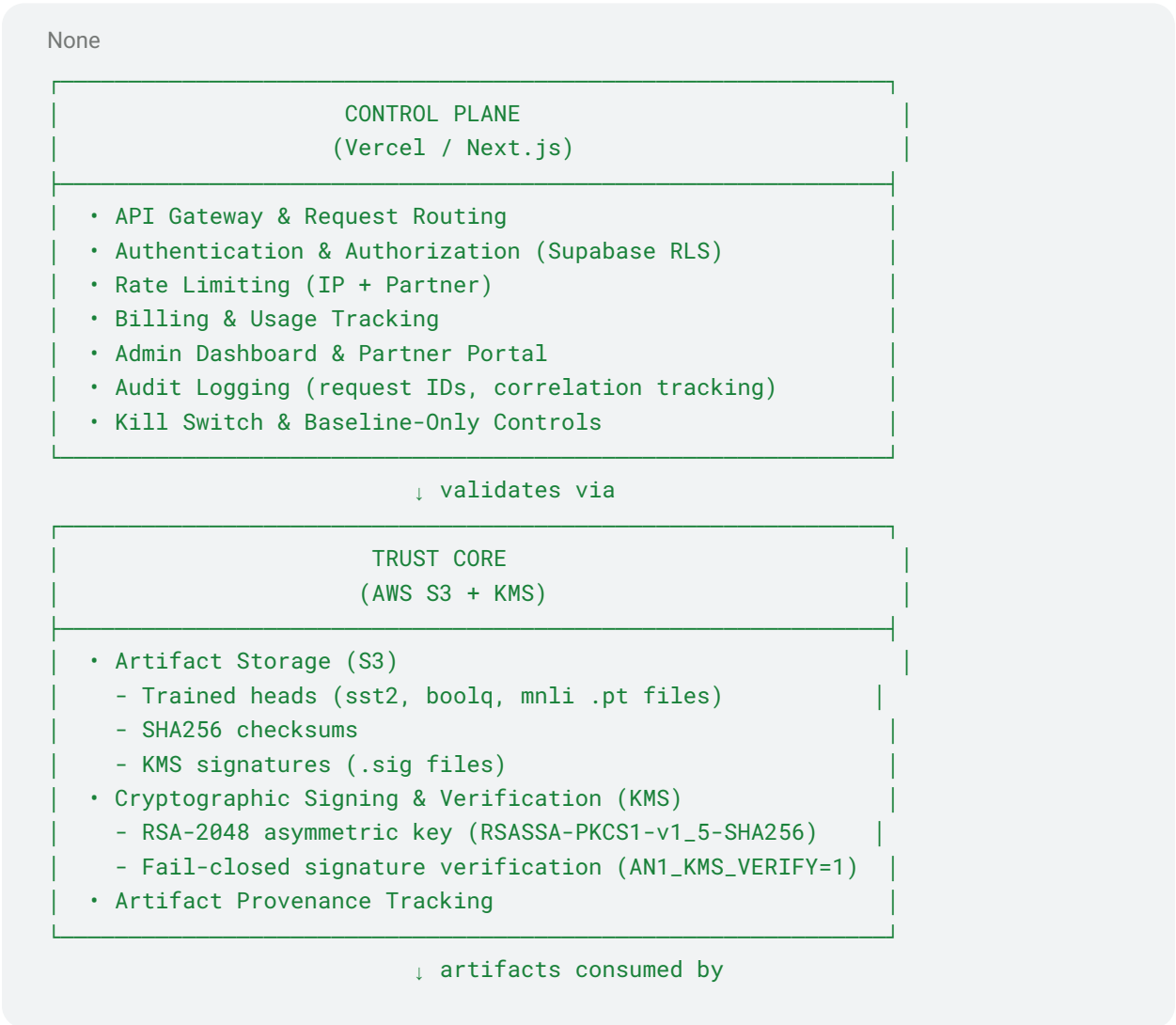
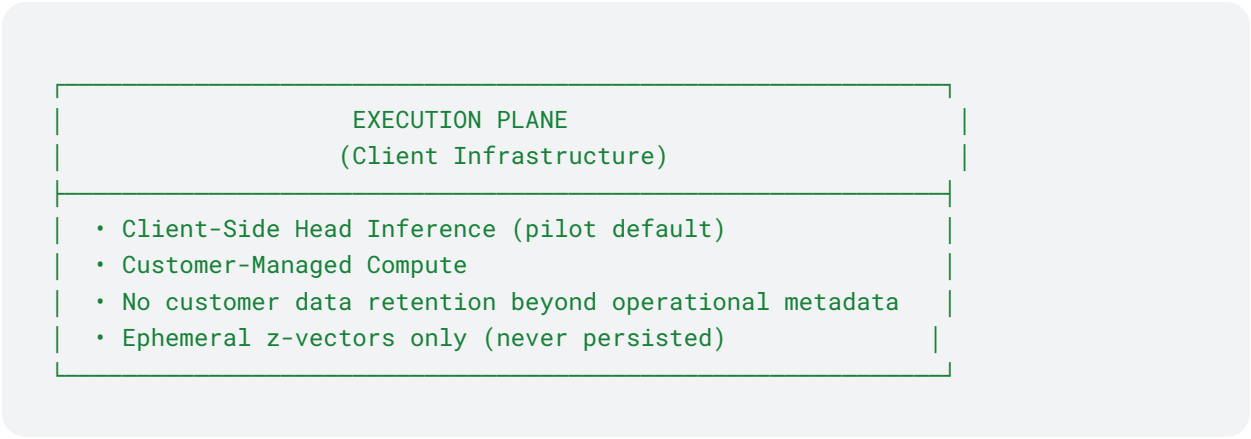
AN1 Architecture for Technical Reviewers

Document Purpose: One-page reference for enterprise security and architecture teams evaluating AN1.

System Architecture: Three-Plane Design

AN1 is designed as a **secure control plane + trust core** for a distributed execution architecture. It is not a monolithic hosted inference service—it is a control system for client-side execution with cryptographic provenance and fail-closed enforcement.





Note: RSA-2048 is used for artifact signing to balance strong security guarantees with fast verification in latency-sensitive inference paths.

Security Boundaries

What Lives Where

Plane	Service	Data	Security Model
Control	Vercel	API keys (hashed), usage metadata, audit logs	RLS-enforced Supabase, service-role only
Trust Core	AWS S3 + KMS	Trained heads, signatures, checksums	KMS-signed artifacts, fail-closed verification
Execution	Client infra	z-vectors, inference results	Customer-managed, ephemeral only

Data Flow (Read-Only Path)

1. Partner authenticates via API key → Control Plane validates (constant-time comparison)
2. Control Plane checks kill switch, baseline-only mode, artifact availability
3. Control Plane loads artifact from S3 → Trust Core verifies KMS signature
4. If signature valid → inference runs on artifact
5. If signature invalid → request fails (fail-closed)
6. Metadata logged (latency, cost, request ID) → Customer data never retained

---

Trust Core Enforcement

Current Production Posture

Environment Variables (Production):

```shell
AN1_STRICT_ARTIFACTS=1      # No fallback to demo weights
AN1_KMS_VERIFY=1            # Cryptographic signature verification required
AN1_KMS_KEY_ID=<key-id>     # AWS KMS key for artifact signing
AN1_BASELINE_ONLY=0         # (Can be toggled to 1 to disable all inference)
AN1_KILL_SWITCH=0           # (Can be toggled to 1 for emergency stop)
```

Verification Commands:

```shell
# Confirm KMS verification is enabled
curl https://animacore.ai/api/status | jq '.security.kms_verification_enabled'
# Expected: true

# Confirm strict artifacts mode
curl https://animacore.ai/api/status | jq
'.configuration.strict_artifacts_only'
# Expected: true

# Verify artifact signature
aws s3 cp s3://an1-artifacts/heads/sst2_head.pt.sig - | base64
aws kms verify --key-id <KEY_ID> --message fileb://sst2_head.pt \
  --signature fileb://sst2_head.pt.sig --signing-algorithm
RSASSA_PKCS1_V1_5_SHA_256
# Expected: SignatureValid: true
```

Operational Controls

Admin Controls (Real-Time)

- **Kill Switch:** Stops all inference immediately (`AN1_KILL_SWITCH=1`)
- **Baseline-Only Mode:** Disables all non-baseline inference (`AN1_BASELINE_ONLY=1`)
- **Artifact Health:** Status endpoint shows S3 artifact availability and KMS signature status

- **Rate Limiting:** 60 req/min per IP, configurable per partner
- **Audit Logs:** All requests logged with correlation IDs, partner IDs, artifact versions

Key Management

- **Root:** Ultimate admin (emergency access only)
- **Human Admin Role:** Day-to-day key management (policy updates, rotation)
- **Service Account (`an1-vercel-signer`):** Runtime operations only (Sign, Verify, DescribeKey)

Production Readiness Checklist

Pilot-Ready (Current State)

- ☑ Control plane deployed on Vercel with RLS-enforced database
- ☑ Artifact storage on S3 with SHA256 checksums
- ☑ KMS key created with proper role separation
- ☑ Fail-closed artifact loading (no fallback weights)
- ☑ Rate limiting, audit logging, correlation IDs
- ☑ Kill switch and baseline-only controls
- ☑ Status endpoint for health monitoring

Enterprise-Ready (Next Phase)

- ☐ All artifacts signed with KMS (`.sig` files in S3)
- ☐ `AN1_KMS_VERIFY=1` enabled in production
- ☐ CloudWatch log mirroring for immutable audit trail
- ☐ Lambda artifact validation on S3 uploads
- ☐ Automated key rotation policy
- ☐ SOC 2 Type II audit (if required)

What This Architecture Enables

1. **Cryptographic Provenance:** Every inference can be traced to a KMS-signed artifact
2. **Fail-Closed Enforcement:** System refuses to run on unsigned or invalid artifacts
3. **Zero Customer Data Retention:** z-vectors are ephemeral, never persisted (only operational metadata retained)
4. **Instant Kill Switch:** Global inference stop without code deployment

5. **Clean Migration Path:** Execution plane can move to AWS (Lambda/ECS) without changing control plane
6. **Managed Execution (Roadmap):** Anima Cloud is an optional future deployment mode and is not required for pilot evaluation

---

Common Questions

Q: Is this a "real" SaaS system?
A: Yes. It is a SaaS control plane for distributed execution. The control plane handles auth, billing, audit, and artifact distribution. Execution happens client-side (pilot) or can be moved to managed compute (future).

Q: What happens if KMS signature verification fails?
A: The request fails immediately with HTTP 500. No inference runs. This is fail-closed by design.

Q: Can artifacts be tampered with?
A: No. Every artifact is signed with a KMS key. Tampering invalidates the signature, causing fail-closed rejection.

Q: What data is retained about customers?
A: Only metadata: request timestamps, latency, cost, partner ID, artifact version. No z-vectors, no inference results, no customer representations.

Q: How do you prove this to auditors?
A:

1. `curl /api/status` shows KMS verification enabled
2. Code inspection shows fail-closed logic in `lib/an1/headLoader.ts`
3. Database schema shows no tables for z-vectors or results
4. Audit logs show request IDs but no payload data

Contact for Technical Deep-Dive

For architecture walkthroughs, security reviews, or integration planning:
Email: ops@animacore.ai
Documentation: https://animacore.ai/pilot
Status Endpoint: https://animacore.ai/api/status
We welcome technical scrutiny and are prepared to answer detailed questions about our security posture, operational controls, and architectural decisions.