



*Always Aware. Always On.*

# AERIONOS™ MANIFEST

---

The Engineering Handbook and Product Doctrine

Version 1.0 • July 2026

*Always Aware. Always On.*

**Developed by Boyer Creative, LLC**

## Document Status

|                    |                                                                                                               |
|--------------------|---------------------------------------------------------------------------------------------------------------|
| Document           | AerionOS™ Manifest                                                                                            |
| Version            | 1.0                                                                                                           |
| Status             | Founding governance document                                                                                  |
| Owner              | Boyer Creative, LLC                                                                                           |
| Approval authority | Founder approval; later Product Owner plus Engineering Lead                                                   |
| Applies to         | AerionOS Core, applications, providers, renderers, Customization Packs, deployments, and commercial offerings |

This Manifest is both an engineering standard and product doctrine. It defines what AerionOS is, what it must protect, how it may evolve, and which boundaries future products and customer configurations may not cross.

### Founding Statement

AerionOS exists to keep audiences informed and safe by transforming complex live information into clear, engaging, reliable experiences - while giving organizations the automation, presentation quality, and commercial flexibility once limited to major-market operations.

# Contents

- 1** Mission and Long-Term Promise
- 2** The Meaning of Always Aware. Always On.
- 3** Core Values and Non-Negotiables
- 4** Audience, Markets, and Product Scope
- 5** Platform Architecture
- 6** Broadcast Intelligence and Editorial Authority
- 7** Life-Safety Doctrine
- 8** Artificial Intelligence Doctrine
- 9** Reliability, Continuity, and Recovery
- 10** Module and API Standards
- 11** Content Providers and AerionOS Services
- 12** Presentation, Renderers, and Distribution
- 13** Customization Packs
- 14** Mission Control and Application Experience
- 15** Users, Roles, Permissions, and Audit
- 16** Security and Data Stewardship
- 17** Development, Testing, and Release Standards
- 18** Deployment and Shadow Mode
- 19** Commercialization and Licensing
- 20** Governance and Change Control
- Appendix A** Canonical Naming
- Appendix B** Brand Identity Standards
- Appendix C** Decision Tests
- Appendix D** Founding Roadmap

CHAPTER 1

---

# Mission and Long-Term Promise

Protect people first. Make live information understandable. Give every organization the power to present it beautifully and reliably.

## Mission

AerionOS™ is a Broadcast Intelligence Platform developed by Boyer Creative, LLC. Its first obligation is to keep audiences informed and safe. It does this by monitoring live information, interpreting meaning and urgency, applying deterministic policy, composing an appropriate experience, and delivering that experience across broadcast, streaming, web, radio, and digital signage.

AerionOS also exists to automate live content reliably, give smaller organizations major-market capabilities, and unify content, emergency information, sponsorship, and operations within one explainable platform.

## Long-Term Promise

In ten years, AerionOS should be the premier broadcast intelligence platform: a product organizations recognize, trust, and actively want for their station, business, venue, or web channel.

The platform must be broad enough to serve broadcasters, hospitality, public safety, schools, healthcare, campuses, corporate networks, and sponsored content channels without diluting its core responsibility to clarity, reliability, and audience safety.

## Product Position

AerionOS is not merely a weather graphics package, digital signage player, traffic system, or automation script. It is the intelligence and coordination layer between live inputs and live experiences.

## CHAPTER 2

# The Meaning of Always Aware. Always On.

The slogan is an engineering contract, not decoration.

## Always Aware

AerionOS continuously understands the state of content, hazards, schedules, systems, destinations, users, and operating conditions. Awareness includes both external signals and internal health.

- Monitor continuously rather than only when an operator asks.
- Normalize incoming information into common contracts.
- Know which event, mission, or item currently controls each output.
- Know whether a dependency is healthy, stale, degraded, or unavailable.
- Explain why an automated action was taken.

## Always On

AerionOS is designed for continuous 24/7 operation. It must preserve a usable output through failure, restart automatically when practical, and recover deliberately rather than returning blindly.

- No single asset failure should intentionally create a blank output.
- Local fallbacks should continue when internet access fails.
- Cloud and on-premise deployments must both be supported.
- Redundant servers and failover must be possible.
- Small installations may operate on a best-effort tier, but the architecture may never assume downtime is acceptable.

## CHAPTER 3

## Core Values and Non-Negotiables

Every feature is evaluated by the audience experience, safety, and operational confidence it creates.

### Audience First

The system exists for the people receiving information. Internal convenience never justifies confusing, late, or misleading output.

### Life Safety First

When presentation, revenue, convenience, and life safety conflict, life safety wins. Mandatory core safety protections cannot be disabled by a customer Customization Pack.

### Deterministic and Explainable

Broadcast-critical and life-safety actions are governed by explicit rules, versioned policies, and traceable data. The same validated input and configuration should produce the same result.

### Operator Authority

Automation may act first when seconds matter, but an authorized operator has final authority. Overrides must be deliberate, permission-controlled, visible, time-bounded where appropriate, and logged.

### Recoverable and Observable

Every important action must have a recovery path. Every important state must be observable. Every important decision must be auditable.

### Modular and Extensible

AerionOS Core remains neutral. Station, customer, sponsor, geography, and brand behavior belong in Customization Packs or service configuration.

### Professional and Beautiful

Clarity is operational safety. Visual quality is not superficial; it builds trust and comprehension.

## CHAPTER 4

# Audience, Markets, and Product Scope

One intelligence platform; many live-content environments.

## Priority Markets

- Local television and radio broadcasters.
- Digital signage and streaming channels.
- Hotels, restaurants, sports bars, and waiting rooms.
- Emergency management and public-information displays.
- Corporate, school, healthcare, and campus networks.

## Product Scope

AerionOS may manage weather, news, sports, traffic, closings, cameras, schedules, advertisements, sponsorships, emergency messages, public information, venue programming, and other live or scheduled content.

Weather is the first deep domain, not the permanent boundary of the platform.

## Market-Specific Experiences

The same Core may support a KNEB broadcast installation, a hotel channel, a sports-bar network, or an emergency operations display. The experience changes through applications, providers, renderers, and Customization Packs; Core intelligence and contracts remain shared.

## CHAPTER 5

# Platform Architecture

Separate intelligence from policy, policy from presentation, and presentation from rendering.

## Layered Model

- Core services coordinate lifecycle, command routing, scheduling, health, logging, and recovery.
- Engines interpret hazards, prioritize events, determine broadcast state, apply policy, and compose presentation plans.
- Providers supply or generate content. AerionOS may itself provide managed content according to the customer subscription.
- Presentation services resolve assets and execute plans.
- Renderers display technically appropriate output on browser, video, radio, signage, mobile, or future destinations.
- Applications provide human workflows such as Mission Control, Broadcast Workspace, Traffic Manager, Simulator, Diagnostics Center, and Customization Manager.
- Customization Packs contain customer-specific identity, geography, policy, assets, sponsorship, and preferences.

## Hard Boundaries

Presentation does not make editorial decisions. Renderers do not reinterpret policy. Providers do not command output. Customer configuration does not modify Core source code.

Production must never depend on development files. Development may consume copies or read-only feeds from production; production may not load mutable development resources.

## Common Contracts

Modules communicate through versioned APIs and the AerionOS Command Bus. Direct hidden coupling between modules is prohibited unless explicitly documented as a platform contract.

## CHAPTER 6

# Broadcast Intelligence and Editorial Authority

Automation should be fast; authority should remain human.

## Decision Hierarchy

AerionOS identifies the controlling event or mission, applies Core safety floors, evaluates the customer Customization Pack, composes the required actions, and exposes the reasons for those actions.

Authorized operators may override automation. A human override does not erase the underlying hazard or recommendation; the interface must continue to show what the system believes and what the operator changed.

## Customer Flexibility

Different customers, stations, locations, outputs, and time periods may use different automation policies. Overnight automation may differ from staffed daytime operation. Radio, television, web, and signage may respond differently to the same event.

## Commercial Considerations

Sponsored and paid content are valid platform content types. Revenue logic may affect scheduling and placement, but it may not suppress mandatory life-safety behavior or conceal the source and nature of emergency information.

## CHAPTER 7

# Life-Safety Doctrine

The platform must act decisively for the highest threats while preserving local editorial control.

## Automatic Interruption Candidates

- Tornado Emergency.
- Tornado Warning.
- Destructive Severe Thunderstorm Warning.
- Severe Thunderstorm Warning.
- Flash Flood Emergency.
- Flash Flood Warning.
- Civil Emergency Message.
- Evacuation.
- 911 outage.
- Amber Alert.
- Locally defined emergency events.

## Mandatory Safety Floor

Some events - including the most extreme tornado, flash-flood, and civil emergency conditions - may carry mandatory Core behavior that cannot be fully disabled. The precise mandatory catalog must be versioned, legally reviewed where appropriate, and disclosed to customers.

Local policy may add events, expand geography, increase automation, or require additional destinations. It may not weaken a mandatory Core safety floor.

## Geography and Relevance

Automatic interruption must consider whether the output serves the affected location, market, venue, route, or audience. Geographic relevance is configuration-driven and explainable.

## Operator Intervention

An authorized operator may take final control, but the system must warn clearly when an override conflicts with a mandatory or strongly recommended safety action. Such overrides require elevated permissions and an audit reason.

## CHAPTER 8

# Artificial Intelligence Doctrine

AI assists intelligence; deterministic policy controls critical action.

## Permitted AI Roles

- Summarize complex feeds and documents.
- Recommend sequence, duration, and supporting content.
- Draft audience-ready language.
- Identify anomalies and possible duplicates.
- Suggest visual layouts, titles, maps, and sponsorship placement.
- Assist operators with diagnostics and recovery.

## Restricted AI Roles

AI may not be the sole authority for life-safety or broadcast-critical interruption. Takeover, abort, mandatory alerting, state transitions, permissions, and recovery boundaries must be enforced by deterministic code and explicit policy.

## Future Evolution

AI authority may expand for noncritical decisions as reliability is demonstrated, but critical control remains bounded by rule-based safeguards, permissions, test coverage, and transparent explanations.

## CHAPTER 9

# Reliability, Continuity, and Recovery

A degraded experience is acceptable; a silent or blank experience is not.

## Reliability Tiers

AerionOS may offer installation tiers ranging from best-effort small systems to redundant enterprise deployments. The feature set and service commitment may differ; the fundamental fail-safe philosophy does not.

## Required Capabilities

- Continuous health monitoring and heartbeat reporting.
- Automatic restart and state replay where safe.
- Fallback assets and fallback scenes.
- Local continuity when upstream internet or cloud services fail.
- Recovery checkpoints before disruptive actions.
- Support for redundant servers and failover.
- Cloud, on-premise, and hybrid deployment options.

## Recovery Doctrine

The system must know what was interrupted, whether it remains valid, and how it should resume. Commercials, live content, map sequences, audio, and manual cut-ins require distinct resume strategies.

# Module and API Standards

Every module must be discoverable, testable, observable, restartable, and versioned.

## Required Module Contract

- Stable module identifier and semantic version.
- Declared capabilities and dependencies.
- Initialize, start, heartbeat, health, recover, and shutdown lifecycle.
- Validated inputs and sanitized outputs.
- Structured logs and trace identifiers.
- Documented error states and fallback behavior.
- Automated unit and integration tests.

## API Standards

APIs use versioned JSON contracts over local interfaces, HTTPS, or WebSocket as appropriate. Schemas are authoritative. Breaking changes require a major version. Deprecations require a migration period and written rationale.

## Command Bus

Modules publish events and commands rather than silently manipulating one another. Priority, source, target, timestamp, trace, payload, and result must be captured for critical commands.

# Content Providers and AerionOS Services

Providers may be customer-operated, third-party, or delivered as managed AerionOS services.

## Provider Responsibilities

- Declare content type, provenance, freshness, health, and capabilities.
- Normalize source data into AerionOS contracts.
- Avoid embedding editorial or rendering decisions.
- Expose licensing, attribution, and usage restrictions.
- Fail visibly and provide fallback metadata.

## Managed Content

AerionOS may provide weather, news, sports, graphics, advertising, cameras, closings, or other feeds as part of subscription packages. Managed content is still presented through the same provider contracts so customers may substitute or combine their own sources.

## Sponsorship and Advertising

Advertising and sponsorship are first-class content providers. Their eligibility, frequency, exclusivity, targeting, and reconciliation must be policy-driven and auditable.

# Presentation, Renderers, and Distribution

The presentation plan expresses intent; the renderer performs the technical work.

## Presentation Responsibilities

Screen Composer defines layout, hierarchy, timing, overlays, commercial behavior, transition intent, and content roles. Asset Registry resolves what is available and safe to use. Presentation Runtime stages and executes the plan.

## Renderer Availability

AerionOS is intended primarily as a SaaS platform, but renderer interfaces may be made available for on-premise, partner, or specialized deployments. Browser, video, radio, signage, mobile, and future renderers must preserve the editorial meaning of the plan.

## No Blank Output

Renderers must maintain a safe output when assets fail. A branded fallback, last-known-safe frame, or alternate content scene must replace unavailable content.

# Customization Packs

Customer identity and local behavior are configured, not forked.

## Purpose

Customization Packs replace the earlier station-only concept. They support broadcasters, businesses, venues, networks, schools, public agencies, and other deployments.

## Pack Contents

- Branding, logos, typography, and colors.
- Geography, service areas, and audience relevance.
- Hazard and editorial policy within Core safety limits.
- Sponsors, campaigns, and advertising rules.
- Assets, cameras, maps, feeds, and playlists.
- User-facing labels and navigation.
- Destination and renderer preferences.
- Operating modes and schedules.

## Prohibited Overrides

A Customization Pack cannot disable Core audit logging, bypass security, alter immutable safety floors, directly patch Core code, or make production depend on development resources.

# Mission Control and Application Experience

An operator should understand the system within ten seconds.

## Mission Control

Mission Control is the AerionOS Shell and primary application entry point. It provides immediate situational awareness, navigation, system health, active mission, hazards, current presentation, commercial status, recovery status, and recent events.

## Named Applications

- Broadcast Workspace - live presentation operations.
- Operator Console - hazard and system control.
- Diagnostics Center - health, logs, faults, and engineering tools.
- Customization Manager - customer identity, policy, geography, providers, assets, and preferences.
- Simulator - deterministic scenario and replay environment.
- Traffic Manager - commercial scheduling, break assembly, and reconciliation.
- Presentation Runtime - execution service.
- Broadcast Mission - traceable operational objective created by the Orchestrator.

## Experience Standards

Interfaces should blend broadcast confidence with executive clarity. Orbitron is the primary AerionOS headline typeface; Inter is the secondary body typeface. Individual outputs may use customer-selected fonts and colors.

Dark and light application modes should be supported. Green, amber, and red are reserved for system status, warnings, and critical conditions rather than decorative use.

# Users, Roles, Permissions, and Audit

Access follows responsibility; disruptive authority is never assumed.

## Canonical Roles

- Viewer - read-only situational awareness.
- Operator - routine operations and approved live controls.
- Supervisor - escalated editorial and operational authority.
- Administrator - configuration and user administration.
- Traffic - commercial scheduling and reconciliation.
- Sales - sponsorship and campaign visibility appropriate to assigned accounts.
- Content Editor - editorial content creation and approval workflows.
- Engineer - system, provider, renderer, and diagnostics authority.
- Corporate Administrator - multi-location governance, policy, licensing, and enterprise oversight.

## Least Privilege

Roles define defaults; deployments may create scoped permissions. A sales user should not restart a renderer. An operator should not silently alter enterprise safety floors.

## Audit Requirements

Logins, configuration changes, policy changes, overrides, destructive commands, approvals, commercial reconciliation, and deployment actions must be attributable to a user or system identity.

# Security and Data Stewardship

Trust depends on both uptime and responsible control.

## Security Principles

- Least privilege and role-based access.
- Encrypted transport and secure credential storage.
- Separation of customer data and configuration.
- Signed or verified release artifacts where practical.
- Dependency and vulnerability management.
- Tamper-evident audit records for critical actions.
- No secrets committed to source repositories or Customization Packs.

## Privacy

AerionOS should collect only the operational and account data necessary to provide the service. Retention, export, and deletion policies must be documented by product tier and jurisdiction.

## Data Provenance

Audience-facing facts must retain source provenance, timestamps, and freshness metadata so outdated or uncertain information can be identified.

# Development, Testing, and Release Standards

Reliable live systems are built through discipline, not confidence.

## Versioning

AerionOS Core follows semantic versioning: MAJOR.MINOR.PATCH, with alpha, beta, and release-candidate prereleases. Applications, modules, and Customization Packs version independently while declaring compatible Core ranges.

## Testing

- Unit tests for module behavior.
- Contract and schema validation.
- Integration tests across the canonical pipeline.
- Simulator scenarios for hazards, cancellations, failures, commercial interruptions, and recovery.
- Visual regression tests for presentation and application UI.
- Shadow-mode observation before production.
- Rollback validation before release.

## Documentation

Every major feature includes vision, requirements, architecture, implementation notes, tests, operating guidance, and release notes. Code without an operational explanation is incomplete.

## Quality Gate

A release must not be promoted because it appears to work once. It must be repeatable, observable, recoverable, and compatible with the declared deployment tier.

# Deployment and Shadow Mode

New intelligence must prove itself beside production before it controls production.

## Environment Separation

Development, test, shadow, staging, and production environments remain distinct. Production never imports mutable development files.

## Shadow Mode

Shadow Mode is required before deployment. It receives representative or mirrored inputs, executes the real decision pipeline, and records what it would have done without affecting on-air or customer output.

## Promotion Path

- Develop and validate locally.
- Run deterministic simulator suites.
- Deploy to isolated shadow environment.
- Compare decisions and output against current operations.
- Resolve discrepancies and complete operator acceptance.
- Schedule controlled launch with rollback.
- Observe and document post-launch behavior.

## Rollback

Every production release requires a tested rollback path, preserved configuration, and a known-safe prior artifact.

# Commercialization and Licensing

Build internally, prove operational value, then commercialize without compromising the doctrine.

## Initial Strategy

AerionOS begins as a private internal system. Commercial offerings follow after stable internal operation and measurable proof.

## Potential Models

- Monthly SaaS subscription.
- Per-screen subscription.
- Per-station or per-location license.
- Managed service.
- Setup fee plus monthly support.
- Sponsor-supported content network.

## Product Packaging

Packages may combine software, managed content, hosting, monitoring, support, storage, provider access, renderer capacity, and Customization services.

## Commercial Guardrails

Pricing or sponsorship must not create hidden editorial influence over mandatory emergency behavior. Customer terms must distinguish AerionOS Core, applications, managed content, third-party feeds, and custom development.

# Governance and Change Control

The Manifest evolves deliberately and never silently.

## Current Authority

Founder approval is required for Manifest changes. As the organization develops, authority transitions to the Product Owner plus Engineering Lead, with founder-defined governance.

## Change Requirements

- Versioned revision.
- Written rationale and impact analysis.
- Identification of affected modules, products, customers, and migration needs.
- Review of safety, reliability, security, commercial, and compatibility consequences.
- Approval recorded in the document history.

## Immutable Principles

Customer Customization Packs cannot override Core safety principles. Commercial pressure cannot remove explainability, auditability, production isolation, or mandatory safety floors.

## Interpretation Test

When the Manifest does not provide a direct answer, choose the option that best protects the audience, preserves operator understanding, maintains reliability, and avoids hidden coupling.

APPENDIX A

---

# Canonical Naming

**AerionOS™:** The platform and Core product family.

**Broadcast Intelligence Platform:** Official descriptor.

**Mission Control:** The AerionOS Shell and primary application entry point.

**Broadcast Workspace:** Live presentation operations.

**Operator Console:** Hazard, mission, and system control.

**Customization Manager:** Customer pack, branding, policy, providers, and local configuration.

**Customization Pack:** Customer-specific package formerly referred to as a Station Pack.

**Diagnostics Center:** Health, logs, faults, replay, and engineering tools.

**Broadcast Mission:** A traceable operational objective created by the Orchestrator.

**Presentation Runtime:** The service that stages and executes presentation plans.

**Traffic Manager:** Commercial scheduling, break management, and reconciliation.

**Simulator:** Deterministic testing and event replay environment.

APPENDIX B

# Brand Identity Standards

Official product identity: AerionOS™ - Broadcast Intelligence Platform - Always Aware. Always On.  
Developed by Boyer Creative, LLC.

| Token       | Hex     | Use                                        |
|-------------|---------|--------------------------------------------|
| Aerion Navy | #0A1A2F | Primary dark background and executive tone |
| Aerion Blue | #1E6BFF | Primary action and brand emphasis          |
| Sky Blue    | #37C1FF | Information and secondary highlight        |
| Aerion Teal | #00C2BC | Intelligence, connectivity, and accent     |
| Slate Gray  | #6B7280 | Secondary text and neutral status          |
| Soft Gray   | #F4F6F8 | Light-mode background                      |

The primary gradient moves from Aerion Navy to Aerion Blue to Sky Blue to Aerion Teal. Orbitron is the preferred headline typeface; Inter is the preferred body and interface typeface. Individual output channels may use Customization Pack typography and color systems.

AERIONOS™ BRAND IDENTITY



ABOUT THE BRAND

AerionOS™ is a Broadcast Intelligence Platform that transforms complex live information into clear, engaging experiences for television, radio, streaming, and digital signage.

AerionOS Alert Manager™ is the intelligent engine that monitors, interprets, and delivers critical weather and emergency information in broadcast-ready language.



PRIMARY LOGO (FULL COLOR)

DARK BACKGROUND

LIGHT BACKGROUND

MONOCHROME

APP ICON / FAVICON

COLOR PALETTE

|             |             |          |             |            |           |
|-------------|-------------|----------|-------------|------------|-----------|
| #0A1A2F     | #1E6BFF     | #37C1FF  | #00C2BC     | #6B7280    | #F4F6F8   |
| AERION NAVY | AERION BLUE | SKY BLUE | AERION TEAL | SLATE GRAY | SOFT GRAY |

GRADIENT (PRIMARY MARK)

TYPOGRAPHY

PRIMARY FONT – HEADLINES

**Orbitron**

ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789

SECONDARY FONT – BODY

**Inter**

Aa Bb Cc Dd Ee Ff Gg Hh Ii Jj Kk Ll Mm Nn Oo Pp Qq Rr Ss Tt Uu Vv Ww Xx Yy Zz 0123456789

APPLICATION EXAMPLES

OUR MISSION

To deliver intelligent, reliable, and beautifully presented information that keeps communities informed and safe.

PRODUCT FAMILY

© 2026 Boyer Creative, LLC. All Rights Reserved.

APPENDIX C

---

## Decision Tests

### Does this protect or confuse the audience?

Choose the clearer, safer audience outcome.

### Can an operator understand the system within ten seconds?

If not, simplify the experience or expose the missing explanation.

### Does this belong in Core or a Customization Pack?

If it varies by customer, geography, brand, sponsor, or editorial preference, it belongs outside Core.

### Is AI deciding a broadcast-critical action alone?

Move the final authority into deterministic policy.

### Could one failure produce a blank output?

Add a fallback or safe retained output.

### Can this action be explained and audited?

If not, it is not ready for production.

### Does production rely on development?

Remove the dependency before deployment.

### Can the interrupted operation recover safely?

Define and test a resume strategy before release.

APPENDIX D

---

# Founding Roadmap

## Phase 1: Platform Foundation

- Hazard Catalog
- Broadcast State Engine
- Station Policy
- Event Queue and Priority Manager
- Screen Composer
- Asset Registry
- Command Bus
- Presentation Runtime
- Recovery Manager
- Orchestrator
- Operator Console
- Simulator and integration adapters

## Phase 2: Product Experience

- Mission Control Shell
- Broadcast Workspace
- Customization Manager
- Diagnostics Center
- Real provider adapters
- Shadow output renderer

## Phase 3: KNEB Shadow Deployment

- Import existing KNEB assets and feeds
- Mirror weather and traffic inputs
- Run side-by-side with KNEB.TV LIVE
- Compare live and shadow decisions
- Complete operator acceptance

## Phase 4: Production Launch

- Controlled KNEB cutover
- Rollback readiness
- 24/7 monitoring
- Post-launch refinement

## Phase 5: Commercial Platform

- Multi-tenant SaaS services
- Subscription packaging
- Managed content products
- Hospitality and signage editions
- Licensing and partner renderer program

# AerionOS™

**Broadcast Intelligence Platform**

*Always Aware. Always On.*