

VALID FOR THE ALPHA SERIES

HANDBOOK & SYSTEM ARCHITECTURE

How it works, security model, modules and operations

WORDPRESS WITHOUT THE PLUGIN STACK

First-Party Infrastructure • Security by Design • Privacy by Default • Recovery First

VisionGaiaTechnology

Technical documentation for the AstraeaOS WP Alpha Series

AstraeaOS WP - Handbook & System Architecture

Valid for the entire Alpha Series

DOCUMENT STATUS

This handbook describes the architecture, operating model and core functions of the AstraeaOS WP Alpha Series. Individual Alpha builds may extend, rename or harden functions. The document is therefore intentionally not tied to a specific Astraea version number.

AstraeaOS WP is a security-centered WordPress distribution and fork based on WordPress 7.1. Its guiding idea is **“WordPress without the Plugin Stack”**: universal infrastructure tasks are integrated as first-party subsystems in the Astraea kernel, while specialized business applications remain the domain of plugins.

This handbook is written for administrators, developers, security reviewers and technical decision-makers. It does not document every PHP class. Instead, it connects the user-facing operating model with the most important architectural decisions from the internal ARCHITECTURE.md.

ALPHA NOTICE

The Alpha Series is development software intended for test, staging and controlled evaluation environments. Security mechanisms reduce risk, but do not replace proper server hardening, external backups, monitoring or responsible administration.

Contents

01. What AstraeaOS WP is and why it exists
02. Architecture and boot model
03. Secure Genesis - installation with a security plan
04. Astraea Module Fabric
05. Security model: GeDefense, ThroneGuard, crypto and Step-Up
06. Vault, Update Guard and autonomous recovery
07. VLP Light and Dattrack
08. Astraea Mail Gateway
09. Performance Engine and Media Engine
10. Web infrastructure modules
11. Update Engine, migration and Compatibility Layer
12. Admin UI and Mission Control
13. Data flows and trust boundaries
14. Operations handbook
15. Plugin strategy and Alpha validity
16. Quick reference and glossary

01 What AstraeaOS WP is

AstraeaOS WP is not a conventional WordPress installation with a large bundle of preinstalled plugins. The distribution takes over universal infrastructure functions directly in its own kernel. Security, backup, privacy, mail, recovery and other modules therefore share common services, common state models and one security architecture.

Guiding idea: “WordPress without the Plugin Stack”

A typical WordPress site often grows by adding independent extensions for firewalling, backups, cookie consent, analytics, SMTP, caching, image optimization, redirects, SEO, forms, Cron management and maintenance mode. Each extension introduces its own update cycle, data model, authorization logic and potentially additional attack surface.

Astraea follows a different model: capabilities that almost every website needs as infrastructure are implemented first-party in the kernel. Plugins remain important, especially for specialist applications such as commerce, LMS platforms, booking systems and industry-specific integrations.

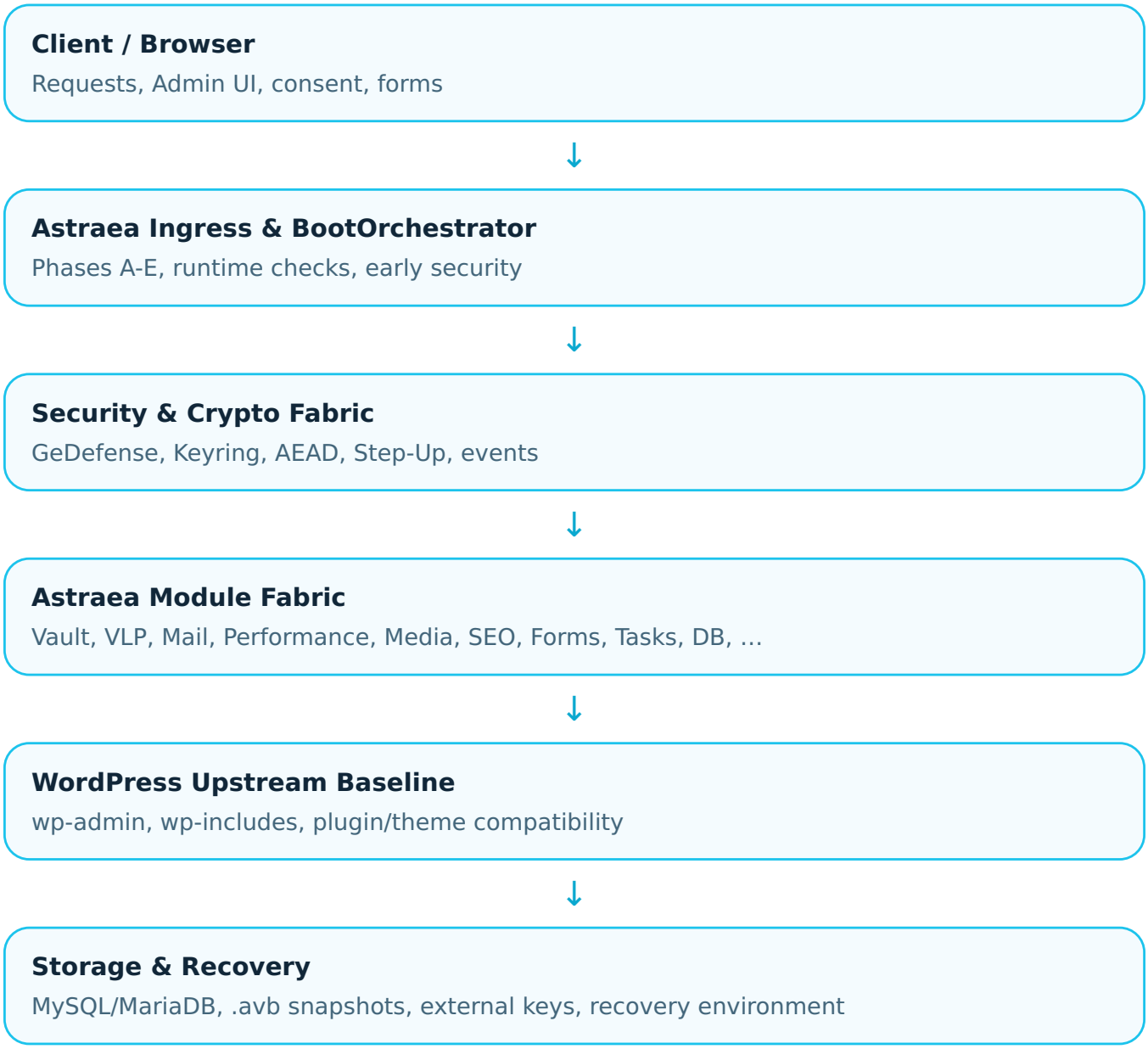
Area	Standard WordPress	AstraeaOS WP
Security	typically a third-party plugin	GeDefense as a kernel subsystem
Backup / rollback	backup plugin	Astraea Vault + Recovery Gate
Consent / analytics	cookie and analytics plugins	VLP Light + Dattrack
Mail	SMTP plugin	Astraea Mail Gateway
Performance	cache / optimization plugins	Performance Engine
Uploads / media	depends on plugins	Media Engine + FileGuard
Updates	upstream WordPress core updater	Astraea Update Engine + Core Update Guard
UI	classic wp-admin	Astraea Glass Shell + Mission Control

What AstraeaOS WP is explicitly not intended to be

- Not a page-builder replacement. Gutenberg remains the central content editing foundation.
- Not an all-purpose monolith. Specialized domain functionality should remain modular.
- No “100% secure” promise. Security states must be measured from evidence.
- Not a dependency collection. The core and Admin UI avoid Composer, npm and CDN runtime dependencies.

02 Architecture and boot model

Astraea modifies the upstream WordPress core as surgically as possible. Most functionality lives under `astraea-core/`. A small number of defined hook points execute security and recovery logic at times that ordinary plugins cannot reach.



The five boot phases

Phase	Timing	Primary responsibilities
A	before WordPress functions	Runtime checks, autoloader, Keyring, Cerberus L0, BootFailureDetector
B	WordPress core available	Argon2id overrides, password migration, FileGuard, header policy, GeDefense pre-flight
C	database and options available	schema migrations, security events, GeDefense full engagement
D	before MU/network/normal plugins	Vault incident check, auto-rollback, quarantine
E	normal lifecycle	Module Fabric, VLP, Mail, Performance, UI, CLI and extended services

> **WHY PHASE D MATTERS**
>
> A conventional backup plugin can fail to load if another plugin terminates the boot process first. Astraea Vault is positioned before the normal plugin load so that recovery actions can run before the problematic extension is loaded again.

Shared core services

- CryptoService and Keyring for encrypted data and key rotation
- StepUpAuthService for sensitive administrative actions
- SecurityEventManager / SecurityEventBridge for evidence-based events
- AtomicCounter for race-resistant rate limiting
- Random/CSPRNG for tokens, salts and keys
- ModuleRegistry and ModuleDependencyGraph for module state and dependencies
- BootFailureDetector for crash-loop protection

03 Secure Genesis - installation with a security plan

The Astraea installer is more than a visual redesign of the standard WordPress setup. After the WordPress installation core completes, Secure Genesis takes control and compiles a defined security baseline so that critical protections are consistently prepared before the first normal login.

Installation sequence

Step	Purpose
1. System Preflight	Check PHP, 64-bit runtime, Sodium, OpenSSL, Argon2id, database version, HTTPS and runtime conditions
2. Database & Core	Connect the database and prepare prefix and baseline
3. Site Identity	Site title, language, timezone and base settings
4. Astraea Master	Create the first privileged Master account
5. Security Profile	Choose Balanced, Hardened, Maximum or Custom
6. ThroneGuard	Provision the privilege boundary and recovery key
7. GeDefense Matrix	Compile security modules and policies
8. Network / Privacy / Communication	Prepare network surface, VLP and optionally Mail
9. Security Compilation	Apply configuration and run real probes
10. Verified Launch	Set installation state to READY and hand over to the first runtime boot

Master and recovery model

ThroneGuard separates the highly privileged Master role from ordinary administrator rights. Secure Genesis generates a high-entropy recovery key intended to be displayed only once. Only an Argon2id verifier is persisted - not the recovery key in plaintext.

> **OPERATIONAL REQUIREMENT**
>
> Store the recovery key securely outside the WordPress installation. Astraea is not designed around a hidden vendor backdoor that can recreate a lost key.

Transactional installation state

```
CORE_INSTALLED
    ↓
SECURITY_APPLYING
    ↓
SECURITY_VERIFYING
    ↓
READY

Failure path: SECURITY_FAILED / Resume
```

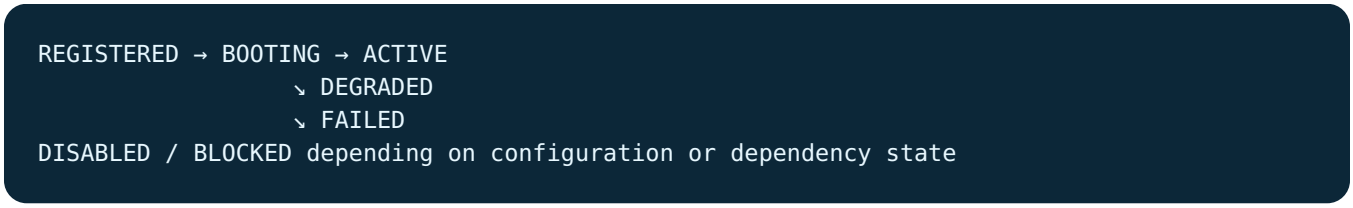
“WordPress is installed” is therefore not treated as equivalent to “Astraea security setup is complete.” An interrupted security compilation pass should be resumable and must not masquerade as a successful installation.

04 Astraea Module Fabric

The Module Fabric is the central lifecycle and dependency controller for first-party subsystems. Modules have an identity, version, boot phase, dependencies, states and health probes. Dependencies are resolved as a directed acyclic graph instead of being documentation only.

Module	Core responsibility	Typical boot
Recovery Gate	Crash-loop detection and emergency console	Phase A
GeDefense	WAF, perimeter, XDR, ThroneGuard	Phase C
Vault	Snapshots, restore, update guard	Phase D
VLP Light	Consent, gatekeeper, Dattrack	Phase E
Mail Gateway	SMTP, StrictTLS, XOAUTH2	Phase E
Performance	Page cache, browser policy, profiling	Phase E
Media	Upload security, SVG, image formats	Phase E
Redirects	301/302, slug watcher, 404 monitoring	Phase E
SEO	Canonicals, OpenGraph, JSON-LD	Phase E
Forms	Secure forms, optional AEAD storage	Phase E
Tasks	WP-Cron inspection and controlled trigger	Phase E
Database	Dry-run cleanup + Vault snapshot	Phase E
Maintenance	HTTP 503 maintenance mode	Phase E
Identity	Sessions, login audit, password status	Phase E
Update	Signed Astraea core updates	Phase E
Compatibility	Granular exceptions	Phase E
Migration	Transition from legacy plugin stacks	Phase E

Module states



Health probes are expected to report evidence. If a condition cannot be verified, **UNKNOWN** or **DEGRADED** is more accurate than an invented **HEALTHY**.

05 Security model: GeDefense, ThroneGuard, crypto and Step-Up

Defense in depth

Astraea deliberately distributes security across multiple layers so that no single filter carries the entire defense. GeDefense combines early network/IP decisions with WAF and DPI logic, browser confinement, RASP-adjacent controls and XDR correlation.

Component	Role
Cerberus	early IP/perimeter drop before complex PHP logic
Zeus	pre-boot WAF and request normalization
Aegis	deep packet inspection for common web attack patterns
Titan	security headers and CSP policy
Hades	optional login/admin concealment and 404 mimicry
ThroneGuard	Master role, privilege separation and recovery key
Styx	egress and SSRF controls
Morpheus	RASP/sandbox-adjacent protections
Gorgon	telemetry/rule synchronization under defined policies
Scanner	file triage: MALWARE, SUSPICIOUS, POLICY, INFO
TRINITY XDR	correlates individual events into incidents

ThroneGuard: administrator is not automatically Master

The Master role is intended to prevent every ordinary administrator session from automatically gaining the same control over critical security functions. Highly sensitive operations are additionally protected through Step-Up authentication.

Session-bound Step-Up authentication

Sensitive actions require fresh re-authentication inside the current browser session. Verification is bound to the WordPress session token and is normally time-limited. Typical examples include restores, snapshot deletion, module toggles, SMTP secrets, database cleanup and VLP registry changes.

Cryptographic foundation

Astraea uses shared AEAD services instead of implementing independent encryption in every module. The architecture provides Libsodium XChaCha20-Poly1305 and AES-256-GCM. Keys are versioned and have lifecycle states such as ACTIVE, DECRYPT_ONLY, RETIRED and REVOKED.

HKDF domain separation ensures that different data classes do not share the same cryptographic context. Examples include MAIL_TRANSPORT, VLP_CONSENT, FORMS_SUBMISSION, VAULT_SNAPSHOT, DATTRACK_ANALYTICS and SECURE_OPTIONS.

> **IMPORTANT**
>
> “Encrypted at rest” only means something when key management, rotation, authenticity checks and failure behavior are equally sound. Astraea therefore treats Keyring, Master Key, AAD and fail-closed behavior as shared platform services.

06 Vault, Update Guard and autonomous recovery

Astraea Vault

Vault is not only a backup archive. It is part of the continuity and recovery model. Snapshots are created as streams and stored in `.avb` containers. The architecture uses envelope encryption: backup data receives dedicated data keys that are then protected by passphrase- or service-key mechanisms.

- full or targeted database and file snapshots
- AES-256-GCM container I/O with integrity validation
- VerifyService for container and SHA-256 verification
- pre-update snapshots for plugins, themes and Astraea core
- pre-cleanup snapshots before database maintenance
- rollback and quarantine for critical failures

Recovery Gate before plugins

```
Boot starts
  ↓
Phase D: Vault checks incident/crash state
  ↓
Problem detected?
  ├─ No  → load plugins
  └─ Yes → rollback / quarantine / recovery
```

Autonomous recovery console

`/astraea-recovery/` provides a dedicated emergency path that relies on as little of the normal WordPress lifecycle as possible. After recovery-key authentication an administrator can disable plugins or modules, control maintenance state, reset boot counters and inspect diagnostic state.

> **RECOVERY PRINCIPLE**
>
> The recovery subsystem should not depend on the same components that may currently be preventing the normal application from booting.

07 VLP Light and Dattrack

Consent as a kernel function

VLP Light (VisionLegalPro Light Edition) integrates consent management and privacy gating directly into Astraea. The purpose is to block external or non-essential resources before consent and to make the service registry centrally manageable.

- server-side gate using WP_HTML_Tag_Processor
- client-side DOM/property gate for dynamically inserted resources
- HMAC-authenticated consent receipts in an HttpOnly cookie
- editable service registry and scanner for external resources
- privacy center in the Astraea Glass design

Dattrack Light

Dattrack is the integrated self-hosted analytics layer and is intended to activate only inside the configured consent logic. The design avoids long-term storage of raw IP addresses, uses rotating pseudonymous identifiers and encrypts analytics payloads using a dedicated Astraea crypto context.

> **LEGAL NOTICE**
>
> Technical privacy-by-design controls are not general legal advice and do not automatically guarantee GDPR compliance. Controllers still need to verify purposes, legal bases, notices and local requirements.

08 Astraea Mail Gateway

The Mail Gateway can replace a separate SMTP plugin on many installations. It plugs into the existing WordPress mail path while enforcing Astraea transport and secret policies.

Area	Implementation
Providers	presets for common providers plus Custom SMTP
Transport	STARTTLS / SMTPS with StrictTLS
TLS	TLS 1.2/1.3 with certificate and hostname validation
Modern Auth	XOAUTH2 without an additional OAuth library
Secrets	AEAD-encrypted profiles, tokens and secrets
SSRF	endpoint policies against loopback, private networks and metadata targets
Diagnostics	SMTP probe and transport status
Journal	minimal delivery metadata only; no message content

Typical data flow

```
wp_mail()
  ↓
TransportManager
  ↓
EncryptedConfigStore → CryptoService(MAIL_TRANSPORT)
  ↓
Endpoint Policy / TLS Policy
  ↓
StrictSMTP / XOAUTH2
  ↓
SMTP Provider
```

Credential changes and test actions are intended to require Step-Up authentication. The Compatibility Layer may allow narrowly scoped exceptions, but such exceptions should be explicit, auditable and used only when needed.

09 Performance Engine and Media Engine

Performance Engine

The Performance Engine is designed to replace common caching and optimization plugins for standard sites without blindly applying historical “combine everything and minify everything” heuristics.

- disk page cache with atomic writes and invalidation
- consent-aware cache variance for VLP state
- browser cache policies and immutable directives for versioned assets
- Heartbeat control
- autoload profiling and budgets
- Legacy Pruner for policy-disabled surfaces such as XML-RPC and pingbacks
- request and database profiling in Mission Control

Media Engine

The Media Engine moves upload security into a defined pipeline: do not trust browser-provided MIME types, inspect magic bytes, constrain active content, control metadata and expose modern formats only if the server actually supports them.

Function	Protection / benefit
Magic-byte validation	makes extension and MIME spoofing harder
SVG sanitization	constrains active XML/SVG constructions
EXIF policy	can remove GPS and camera metadata
GD re-encode	normalizes supported raster images
WebP / AVIF capability	enabled only when server-side codec support exists

10 Further web infrastructure modules

Redirect Manager

Manages 301/302 redirects, watches slug changes and provides 404 telemetry. Regex rules are bounded, redirect loops and chains are checked, and privacy-conscious monitoring avoids unnecessary raw-IP retention.

SEO Essentials

Astraea integrates technical SEO fundamentals rather than keyword “traffic-light” scoring. It covers canonicals, meta descriptions, OpenGraph/Twitter Cards, JSON-LD and sitemap extensions. Conflict detection is intended to avoid duplicate output when a known external SEO plugin is active.

Forms Light

Forms Light covers standard forms through a secure submission pipeline. CSRF protection, honeypot, atomic rate limiting, upload checks and optional encrypted storage reuse shared Astraea services.

```
Form Submit
  ↓
CSRF / Nonce
  ↓
GeDefense / Rate Limit / Honeypot
  ↓
Validation + Upload Pipeline
  ↓
CryptoService(FORMS_SUBMISSION) optional
  ↓
Mail Gateway optional
```

Task Center

Makes WP-Cron visible: scheduled hooks, origin, next run, overdue tasks and unusually high frequencies. Manual execution is limited to validated hooks that are actually scheduled.

Database Maintenance

Uses the pattern **Analyze → Dry Run → Vault Snapshot → Step-Up → Cleanup**. Typical targets include revisions, spam, trash, expired transients and selected orphaned data. If a required snapshot cannot be created, destructive cleanup should fail closed.

Maintenance Mode

Provides an SEO-safe HTTP 503 response with Retry-After, administrator/token bypass and a local glassmorphism page without CDN dependencies.

Identity Center

Shows active sessions, login audit data and password status. Other sessions can be terminated selectively. Features such as passkeys/WebAuthn should only be reported as implemented once the complete authentication flow is genuinely present.

11 Update Engine, migration and Compatibility Layer

Astraea Update Engine

Because Astraea is a WordPress fork, the standard upstream core update path must not overwrite the Astraea kernel. Core Update Guard therefore blocks normal core updates, while the Astraea Update Engine is designed for signed Astraea distributions.

```
Vault pre-update snapshot
  ↓
Release manifest + package
  ↓
Verify Ed25519 signature
  ↓
Verify SHA-256
  ↓
Path-jailed staging / ZipSlip protection
  ↓
Pre-swap integrity check
  ↓
Activate
  ↓
Health probe
  ↓
Commit or rollback
```

Release authenticity must be distinguished from file integrity: a SHA-256 hash detects changes, but without a trusted signature it does not prove that a package actually came from VisionGaiaTechnology.

Migration Wizard

The Migration Wizard analyzes existing WordPress installations and detects infrastructure plugins whose responsibilities Astraea can already provide natively. A Vault snapshot is required before changes. Plugins are not deleted automatically; migration is designed around non-destructive deactivation and a way back.

Legacy category	Astraea target
Security plugin	GeDefense
Backup plugin	Vault
Cookie / consent	VLP Light
Analytics	Dattrack
SMTP	Mail Gateway
Cache / performance	Performance Engine
Redirects	Redirect Manager
Basic SEO	SEO Essentials
Standard forms	Forms Light

Compatibility Layer

Astraea does not attempt to buy compatibility by globally disabling security. It follows the principle of the **minimum effective exception**: relax only the exact capability a theme or plugin needs, label the exception and audit it.

> **COMPATIBILITY IS A TRADE-OFF**
>
> Flags such as legacy XML-RPC, relaxed REST authentication or a PHP mail fallback may be useful in specific environments, but can weaken the original Astraea baseline. They should be documented, justified and reviewed regularly.

12 Admin UI and Mission Control

Astraea places a consistent glassmorphism shell on top of the WordPress administration base. The goal is not to reimplement the entire backend, but to create one interface across core screens and first-party modules.

- Astraea top bar and categorized sidebar
- NavigationAdapter with CONTENT, DESIGN, SYSTEM, ASTRAEA and EXTENSIONS areas
- Mission Control dashboard with security and system telemetry
- Command Palette through Ctrl+K
- Notice Center with toasts and drawer
- Legacy Surface Adapter for classic WordPress tables, postboxes and plugin screens
- Glass Login Portal and Secure Genesis installer

UI area	Example
Mission Control	system status, Security HUD, performance
GeDefense Suite	WAF, scanner, XDR, Titan
Vault	snapshots, restore, key actions
VLP Light	consent, services, scanner, Dattrack
Mail Gateway	provider, TLS, OAuth, diagnostics
Module Fabric	active state, dependencies, health
Recovery	standalone emergency interface

Design system

The UI uses local design tokens for color, spacing, typography and shadows. Layers such as glass surface L1/L2/L3 define visual depth. Assets are local; Google Fonts and CDN JavaScript are not required.

13 Data flows and trust boundaries

Public page request

```
Browser Request
  ↓
Phase A/B security baseline
  ↓
Phase C GeDefense
  ↓
Phase D Recovery Gate
  ↓
Plugins / Theme
  ↓
Phase E Modules
  ↓
VLP / Performance / Headers
  ↓
Response
```

Privileged administrator action

```
Admin starts critical action
  ↓
Capability + Nonce
  ↓
Step-Up session verification
  ↓
optional Vault snapshot
  ↓
Operation
  ↓
Health / Security Event
```

Form submission

Form data moves through CSRF, rate-limit, validation, upload and crypto layers. Sensitive payloads should not be written to generic logs. Encryption at rest uses a dedicated domain context.

API architecture

Astraea uses WordPress AJAX, admin-post and selected REST endpoints. Administrative mutations are protected with nonces and capabilities, with Step-Up added for sensitive operations. Public endpoints such as consent, Dattrack or CSP reporting require their own rate limits and minimized payloads.

Filesystem trust boundary

File operations are designed around path jails, realpath validation, symlink awareness and bounded staging areas. Update, Vault, Media, Forms and Recovery are especially filesystem-adjacent and must preserve path boundaries consistently.

14 Operations handbook

After first installation

- Store the recovery key offline and document the recovery process organizationally.
- Open Mission Control and verify all security-critical modules report ACTIVE/HEALTHY or understandable DEGRADED states.
- Configure Vault, create at least one complete snapshot and verify it.
- Configure the Mail Gateway and run SMTP diagnostics / a test message.
- Review the VLP Service Registry, run the scanner and validate consent texts from a technical and legal perspective.
- Enable page caching only after functional testing and exclude dynamic areas as necessary.
- Use Compatibility flags only when a concrete problem can be reproduced.

Recommended update procedure

Step	Control
1	check release channel and signature status
2	verify the current Vault snapshot
3	apply the update only through the Astraea Update Engine
4	check health probes and Mission Control
5	spot-check frontend, login, Gutenberg, forms and mail
6	use recovery / Last Known Good when faults occur

If a site fails after a plugin update

Do not immediately delete files manually. First inspect Recovery Gate / Vault incident state. If Astraea can correlate the failure with a plugin, the recovery mechanism should perform a controlled rollback or quarantine before loading that plugin again.

Triaging security events

A security event is evidence, not automatically a confirmed attack. The malware scanner distinguishes MALWARE, SUSPICIOUS, POLICY and INFO. An unusual file must not be classified as malware without sufficient evidence.

Class	Meaning
MALWARE	strong active malicious-code evidence; may block/quarantine
SUSPICIOUS	notable, but insufficient for a malware claim
POLICY	violates a local security policy
INFO	context information without a malware claim

Backup discipline

Astraea Vault reduces recovery risk but does not replace an offsite-backup strategy. Important environments should regularly copy encrypted snapshots away from the web server and perform restore tests.

15 Plugin strategy and Alpha validity

Plugin categories Astraea can often eliminate

Plugin category	First-party capability
Security / WAF	GeDefense
Backup / rollback	Vault
Cookie / consent	VLP Light
Self-hosted analytics	Dattrack Light
SMTP	Mail Gateway
Caching / performance	Performance Engine
Image / upload security	Media Engine
Redirects	Redirect Manager
Technical SEO	SEO Essentials
Contact / standard forms	Forms Light
Cron management	Task Center
DB cleanup	Database Maintenance
Maintenance plugin	Maintenance Mode
Session security	Identity Center

Functions that should remain plugins

- WooCommerce and specialized e-commerce
- LMS / course platforms
- booking and reservation systems
- CRM / ERP domain logic
- industry-specific integrations
- complex page builders or design systems when genuinely required

Validity

> **VALID FOR THE ALPHA SERIES**
>
> This handbook intentionally carries no Astraea version number. It describes the common architecture and operating philosophy of the Alpha Series. Where details conflict, the installed build, its release notes and current health/compatibility notices take precedence.

The Alpha Series may evolve substantially. Update, recovery, performance, compatibility and security policies in particular may change as a result of audits and field testing. That is expected during this development stage.

16 Quick reference and glossary

Important paths

```
/astraea-core/          First-party kernel and modules
/astraea-recovery/      Autonomous emergency console
/wp-admin/              Upstream admin + Astraea surgical hooks/UI
/wp-includes/           WordPress baseline + runtime anchor
/wp-content/            Themes, plugins, uploads, caches
```

Glossary

Term	Meaning
AEAD	Authenticated Encryption with Associated Data; encryption plus integrity/ authenticity protection
AAD	additional authenticated context data that does not need to be encrypted
Argon2id	memory-hard password hashing / key-derivation function
CSPRNG	cryptographically secure pseudorandom number generator
GeDefense	Astraea security kernel combining perimeter, WAF, XDR and identity components
Keyring	versioned key inventory with lifecycle states
Module Fabric	registry, dependency graph and lifecycle management of first-party modules
Recovery Gate	early boot layer before plugins for recovery actions
Step-Up	fresh re-authentication for critical actions
ThroneGuard	Master privilege separation and recovery-key model
VLP Light	privacy/consent kernel including Dattrack
Vault	encrypted snapshot, restore and rollback engine
XDR	correlation and response across multiple security event sources
Zero External Dependencies	no Composer/npm/CDN requirement for the Astraea runtime

Technical basis

This system description is condensed from the internal architecture map of the Alpha Series. The complete ARCHITECTURE.md additionally contains file references, API mappings, dashboard routes, database tables and detailed component relationships.

> **DOCUMENT END**
>
> This handbook is valid for the Alpha Series and is intended to evolve with the architecture. Production decisions should always be based on the actual build, current release notes, test results and independent security reviews.

AstraeaOS™ is a trademark of VisionGaiaTechnology. This handbook does not claim a registered trademark status.