

Enterprise Mobile Product Factory: Comprehensive Reference Manual

Complete Technical Specifications & READMEs of All 163 Hardened Modules

GitHub: <https://github.com/sh15aa/Appreusable> **Developer:** foundershivanand@gmail.com

Module 01: Product Idea Engine

The **Product Idea Engine** ensures that every app built in the factory undergoes strict market validation, scoring, and lifecycle tracking before writing any code.

Key Features

- Strict Validation Rubric:** Evaluates market demand, competition density, willingness to pay, and technical feasibility.
- Finite State Machine:** Tracks app stages from `idea` -> `market_research` -> `mvp_definition` -> `launch` -> `scale`.
- Zero Hallucination Scorecard:** Computes composite viability scores (0-100) with pass/fail gates.

Usage

```
final idea = ProductIdea( id: 'app-001', name: 'FocusPulse', problem: 'Knowledge workers suffer from distraction without physical feedback', solution: 'Haptic-driven Pomodoro timer with biometric stress detection', targetAudience: 'Remote workers, software developers', monetization: MonetizationModel.subscription, platform: TargetPlatform.both, ); final score = IdeaValidator.evaluate(idea, marketDemand: 8, competition: 5, techFeasibility: 9); if (score.isViable) { ideaStateMachine.transition(idea.id, AppStage.mvpDefinition); }
```

Module 02: Tech Stack Standard

Enforces zero-tolerance architecture rules and Dart linter configurations for all apps in the factory.

Strict Rules

- Zero Dynamic/Raw Types:** `strict-casts`, `strict-inference`, and `strict-raw-types` are mandatory.
- No Raw Prints:** `avoid_print` is set to `error`. All logging must route through `mod_25_crash_monitoring` or secure logger.
- Unawaited Futures:** All asynchronous calls must be explicitly awaited or safely unawaited.
- Build Context Safety:** Direct usage of `BuildContext` across async gaps without `mounted` check triggers a compile error.

Module 03: App Starter Template

The foundation scaffold for any new app generated in the factory. It enforces Clean Architecture folders:

```
lib/ ■■■ core/ ■■■ config/ ■■■ errors/ ■■■ network/ ■■■ theme/ ■■■ utils/ ■■■ features/ ■■■ [feature_name]/ ■■■ data/ ■■■ datasources/ ■■■ models/ ■■■ repositories/ ■■■ domain/ ■■■ entities/ ■■■ repositories/ ■■■ usecases/ ■■■ presentation/ ■■■ controllers/ ■■■ screens/ ■■■ widgets/ ■■■ main.dart
```

Module 04: Core Mobile Architecture

Defines Clean Architecture primitives ensuring complete separation of concerns:

- `Result<Success, Failure>` functional pattern (zero unhandled exceptions)
 - `BaseUseCase<Type, Params>` for business rules
 - `BaseRepository` contracts
-

Module 05: App Configuration System

Manages per-app configuration and flavors:

- `Environment.dev`, `Environment.staging`, `Environment.production`
 - Immutable, strongly typed config objects
 - Strict validation preventing missing production keys
-

Module 06: Android Identity

Provides templates and scripts for hardening Android application identity, signing, and security:

- Secure `build.gradle.kts` configuration template
 - Obfuscation with ProGuard / R8 rules
 - `network_security_config.xml` with cleartext traffic disabled & certificate pinning
 - Automated keystore generation & validation scripts
-

Module 07: iOS Identity

Provides standardized configuration, security hardening, and entitlements for iOS:

- Strict `Info.plist` with App Transport Security (ATS) cleartext traffic blocked
 - Entitlements template with Keychain Sharing, Associated Domains, and Push Notifications
 - Fastlane Match signing workflow
-

Module 08: Environment System

Manages environment configuration across `.env.dev`, `.env.staging`, and `.env.prod`.

- Secrets are decrypted at runtime or injected via compile-time `--dart-define`
 - Validates that mandatory keys are present before app execution
-

Module 09: Authentication Module

Production-grade authentication engine designed with zero-trust principles:

- **PKCE OAuth2 Flow**: Code verifier and challenge generation
 - **Token Rotation**: Anti-replay token rotation engine
 - **Pluggable Identity Providers**: Email, Google, Apple, Anonymous
 - **Secure Token Storage Interface**: Backed by hardware keystore
-

Module 10: Authorization System

Enforces Zero-Trust authorization principles:

- The mobile app verifies claims locally for UI visibility, but server is always source of truth.
- RBAC role check: `UserRole.free`, `UserRole.subscriber`, `UserRole.admin`.
- Granular permissions: `Permission.createProject`, `Permission.exportData`, `Permission.accessAi`.

Module 11: Local Data System

Provides encrypted key-value and relational local persistence:

- SQLCipher / Encrypted SQLite interface
- Hardware Keystore / KeyChain backed 256-bit encryption key derivation
- In-memory test mock for rapid unit testing

Module 12: Offline-First Architecture

Implements the **Outbox Pattern** for robust offline mobile operations:

- Queues write mutations when device is disconnected
- Dispatches transactions with UUIDv4 idempotency keys when connection restores
- Conflict resolution strategies: `lastWriteWins`, `serverAuthoritative`, `clientWins`

Module 13: API Layer

Standardized HTTP client engine for the entire factory:

- Standardized API Response Envelope: `ApiResponse<T>` with `success`, `data`, `error`
- Retry logic with exponential backoff & jitter
- Authentication token injection & automatic 401 refresh handler

Module 14: Error-Handling System

Centralized error handling and security sanitization:

- Prevents sensitive stack traces, SQL errors, or backend addresses from leaking to the user.
- Maps low-level exceptions (HTTP 500, `SocketException`, `Timeout`) to user-friendly messages.

Module 15: State Management

Defines a strict state pattern to prevent ad-hoc boolean flags:

- `UiState<T>`: `Initial`, `Loading`, `Success`, `Empty`, `Error`, `Offline`, `Refreshing`
- Enforces exhaustive UI rendering via `when(...)` pattern matching.

Module 16: Design System

Internal mobile design system specification providing standard reusable UI primitives:

- [AppButton](#) (Primary, Secondary, Destructive, Ghost, with loading states)
 - [AppCard](#) (Elevated, Outlined, Flat)
 - [AppTextField](#) (With validation, clear button, obscure toggle)
 - [AppSkeleton](#) (Shimmer loading placeholder)
-

Module 17: Design Token System

Centralized design tokens ensuring visual harmony and ultra-fast re-theming across 100+ apps.

- [AppSpacing](#): [xs](#) (4), [sm](#) (8), [md](#) (16), [lg](#) (24), [xl](#) (32), [xxl](#) (48)
 - [AppRadius](#): [sm](#) (4), [md](#) (8), [lg](#) (16), [full](#) (999)
 - [AppElevation](#): standard elevation values
-

Module 18: Responsive UI

Adaptive layout breakpoints:

- [DeviceType.compact](#) (< 600 dp: standard phones)
 - [DeviceType.medium](#) (600 - 840 dp: foldables & small tablets)
 - [DeviceType.expanded](#) (> 840 dp: tablets & desktop)
-

Module 19: Accessibility (A11y)

Enforces accessibility guidelines across the factory:

- Minimum touch target: 48x48 dp
 - WCAG 2.1 AA and AAA contrast ratio verification
-

Module 20: Dark Mode & Dynamic Theme

Dynamic theme switcher supporting:

- [ThemePreference.system](#), [ThemePreference.light](#), [ThemePreference.dark](#), [ThemePreference.amoled](#)
-

Module 21: Device Features Module

Standardized native device bridges:

- [HapticService](#): Heavy, medium, light, selection feedback
 - [ClipboardService](#): Secure clipboard copy/paste with auto-clear after 60s for sensitive credentials
 - [BiometricService](#): Hardware-level biometric check contract
-

Module 22: Permission Architecture

Enforces the Golden Rule of Mobile Permissions:

- Never request permissions on cold launch.
 - Present a contextual rationale explaining the user value first.
 - Handle [granted](#), [denied](#), [permanentlyDenied](#), and [restricted](#) gracefully.
-

Module 23: Push Notifications

Handles notifications across Android (FCM) and iOS (APNs):

- Token registration & automatic refresh sync
 - Foreground, background, and cold start tap handling
 - Deep link routing via payload parameters
-

Module 24: Analytics

Privacy-centric event dispatcher:

- Strictly scrubs PII (emails, passwords, phone numbers, auth tokens)
 - Pluggable providers (PostHog, Mixpanel, Firebase, Amplitude)
 - Consent mode gatekeeper (requires user consent under GDPR)
-

Module 25: Crash Monitoring

Standardized error reporting:

- Captures unhandled asynchronous Dart zone errors
 - Adds structured breadcrumbs (navigation, button taps, network requests)
 - Scrubs sensitive authorization headers before submitting reports
-

Module 26: Performance Monitoring

Tracks mobile performance KPIs:

- Cold and warm startup duration
 - Screen transition rendering time
 - Network request latency
-

Module 27: AI Client Module

Client-side AI integration:

- Never includes sensitive raw AI provider keys in client binary.
 - Routes all AI requests through [mod_28_ai_cost_gateway](#).
 - Supports Server-Sent Events (SSE) streaming for typing animations.
-

Module 28: AI Cost Control & Gateway

Protects profitability and prevents abuse:

- Token quotas per user tier (Free: 5/day, Pro: 100/day, Unlimited: 500/day)
 - Rate limiting using Token Bucket algorithm
 - Automatic model fallback (Claude 3.5 Sonnet -> GPT-4o -> Gemini 1.5 Flash)
-

Module 29: Subscription System

Cross-platform subscription management:

- Tier abstraction: `Tier.free`, `Tier.monthlyPro`, `Tier.annualPro`, `Tier.lifetime`
 - Never relies on local boolean flags alone; validates with server receipts
-

Module 30: Purchase Security

Prevents in-app purchase piracy and tampering:

- Server-side cryptographic verification of Google Play and Apple App Store receipts
 - Webhook signature validation (HMAC SHA-256)
 - Anti-tamper validation prevents client-side binary modification
-

Module 31: Ads Module

Advertising infrastructure:

- Supports Banner, Interstitial, and Rewarded ads
 - Dynamic remote switch (`ads_enabled = false` for subscribers)
 - Frequency capping prevents user fatigue
-

Module 32: Remote Configuration

Enables instant app adjustments without App Store re-submission:

- Type-safe string, integer, boolean, and JSON getters
 - Fallback defaults ensure offline resilience
-

Module 33: Feature Flags

Enables gradual feature rollouts:

- Deterministic hashing via SHA-256 ensuring user remains in the same bucket
 - Percentage rollouts (10%, 50%, 100%)
 - Kill-switch support
-

Module 34: Deep Link System

Universal links and custom scheme router:

- Validates URI structure and sanitizes path and query params
 - Routes users to exact screens (e.g. [/product/:id](#), [/referral/:code](#))
-

Module 35: Sharing System

Standardizes social sharing across apps:

- Generates formatted share cards and attribution links
 - Prepares sanitized shareable text and assets
-

Module 36: Viral Loops Module

Built-in viral growth mechanics:

- Cryptographic referral code generation
 - Multi-tier referral reward entitlement unlock
-

Module 37: Onboarding Engine

Modular onboarding controller:

- Manages sequential slides, persona selection, and permission priming
 - Persists completion status
-

Module 38: App Lifecycle Guardian

Monitors application execution state:

- Cold start vs Warm start detection
 - Automatic flush of pending mutations and analytics before OS kills background process
-

Module 39: Security Hardening & Anti-Tamper

Hardens apps against attacks:

- Root / Jailbreak check indicators
 - Debugger attachment detection
 - Screenshot & screen recording prevention flags
-

Module 40: Data Validation & Sanitization

Sanitizes all user inputs before persistence or transmission:

- Strict email, phone, URL validation
 - Anti-XSS HTML entity escaping
 - Anti-SQLi parameter validation
-

Module 41: Database Architecture

Encrypted SQLite migration engine:

- Manages sequential schema migrations ([V1__init.sql](#), [V2__add_indexes.sql](#))
 - Atomic transaction wrapping ensures no partial migrations on crash
-

Module 42: File Storage & Upload

Uploads files directly to S3 / Cloud Storage via presigned URLs:

- File size validation and MIME-type verification
 - Never streams file uploads through application API servers directly
-

Module 43: Background Job System

Coordinates WorkManager (Android) and Background Tasks (iOS):

- Enforces network constraints ([requiresUnmeteredNetwork](#), [requiresCharging](#))
 - Executes periodic background sync safely without draining battery
-

Module 44: App Backend Template

Zero-trust isolated backend server template built with Python FastAPI:

- Strict JWT authentication and authorization
 - Rate limiting with SlowAPI
 - Isolated SQLite / PostgreSQL database connection
 - Structured JSON logging
-

Module 45: CI/CD Factory

Reusable GitHub Actions workflows and Fastlane templates:

- [ci_android.yml](#): Builds signed Android App Bundle (AAB), executes tests, uploads to Play Store Internal Track
 - [ci_ios.yml](#): Builds signed iOS IPA via Fastlane Match, uploads to Apple TestFlight
-

Module 46: Automated Testing Suite

Utilities for test-driven development:

- Mock HTTP response dispatcher
 - Standard test fixtures
-

Module 47: Golden User Journey

Defines Maestro / Patrol end-to-end automation scripts for the primary conversion funnel:

Module 48: Device Test Matrix

Standardized test matrix configuration for Firebase Test Lab / AWS Device Farm:

Ensures testing across 12 distinct device archetypes (small phones, mid-range, phablets, tablets, Android 11 to 14, iOS 16 to 18).

Module 49: Release Checklist Pre-Flight

Automated Python pre-flight auditor script that verifies all 32 critical release criteria before store submission.

Module 50: ASO Factory

App Store Optimization (ASO) metadata engine:

- Generates localized titles (max 30 chars), subtitles (max 30 chars), and descriptions
 - Analyzes keyword density and character limits for Google Play and Apple App Store
-

Module 51: Screenshot Factory

Automates App Store and Google Play screenshot generation:

- Standard 5-slide layout narrative:
 1. Main Value Proposition
 2. Core Action / Feature
 3. Differentiator / Magic Moment
 4. Rich Results & Analytics
 5. Premium / Superpowers
 - Fastlane Snapfile configuration and templating
-

Module 52: Review Monitoring & AI Categorization

Harvests and classifies store reviews using rule-based and AI sentiment analysis:

- Categories: [Bug](#), [FeatureRequest](#), [PricingComplaint](#), [Praise](#), [Confusion](#)
 - Generates actionable backlog issues with priority scores
-

Module 53: App Analytics Dashboard

Aggregates product KPIs across the 100-app portfolio:

- DAU, MAU, Retention (D1, D7, D30)
 - MRR, ARPU, Churn Rate, LTV
 - Exportable to SQLite / Postgres / JSON
-

Module 54: Product Health Scorecard

Algorithmic scorecard evaluating the operational health of any app in the portfolio:

- Stability (Crash-free rate)
- Retention (D1, D7)
- Monetization (Conversion %)
- Rating (Store average)

Module 55: App Factory Database

Central registry tracking all 100+ applications across their lifecycle:

- Product metadata, bundle IDs, backend URLs, store statuses
- SQLite database schema with migrations

Module 56: Automated Product Creation CLI (factory-cli)

Command-line interface to scaffold a new independent production mobile app with 1 command:

```
./factory-cli new --id app-042 --name "VoiceNotes AI" --bundle com.factory.voicenotes
```

Module 57: AI Agent Team Prompts & Specs

System prompts and operational playbooks for the 15 specialized AI agent roles:

1. Research Agent
2. Competitor Agent
3. PRD Agent
4. UX Agent
5. UI Agent
6. Architecture Agent
7. Coding Agent
8. Backend Agent
9. Security Agent
10. Testing Agent
11. ASO Agent
12. Content Agent
13. Analytics Agent
14. Review Agent
15. Growth Agent

Module 58: 16-Phase Gatekeeper Pipeline

Orchestrates the 16 phases of app development with strict entry and exit criteria:

- Phase 1: Discovery
- Phase 2: Validation
- Phase 3: PRD
- Phase 4: UX
- Phase 5: UI
- Phase 6: Architecture
- Phase 7: Backend
- Phase 8: Mobile Development
- Phase 9: Testing
- Phase 10: Security
- Phase 11: Beta
- Phase 12: Store Submission
- Phase 13: Launch
- Phase 14: ASO
- Phase 15: Analytics
- Phase 16: Iteration

Module 59: Architecture Boundary Linter

Enforces boundary separation:

- Prohibits cross-app imports (`app-001` cannot import `app-002`)
- Ensures shared modules do not import app-specific feature code

Module 60: Master Portfolio Registry & Monorepo

Orchestrates the 100+ app portfolio monorepo:

- Melos / Dart workspace management
- Bulk testing, linting, and dependency upgrading across all 60 modules and 100 apps

Module 61: Isolation Guardian

Enforces **Rule 61: The Most Important Factory Rule:**

```
FACTORY ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ ■■■ App 1 App 2 App 3 ■■■ Backend Backend Backend ■■■ DB DB
DB NOT: GIANT BACKEND ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ ■■■ App 1 App 2 App 100
```

Security & Architectural Invariants

1. **Isolated Source Code:** Each app has its own self-contained directory under `apps/<app_id>`.
2. **Isolated Backend:** Each app has its own dedicated backend service (`apps/<app_id>/backend/`).
3. **Isolated Database:** No shared connection strings, credentials, or databases between products.
4. **Isolated Store Identity:** Independent Bundle IDs, Keystores, and Provisioning Profiles.

Module 62: One-Click Product Synthesizer

Implements **Rule 62: The 'One-Click' Future**.

Transforms a high-level product concept into a complete, isolated, production-grade product bundle:

1. PRD (Product Requirements Document)
2. Architecture Blueprint
3. Database Schema (SQLite / SQLCipher)
4. OpenAPI 3.0 API Specification
5. Screen List & User Flows
6. UI Generation Prompts
7. Test Plan & Golden Journey
8. Security Checklist
9. ASO Plan (Apple + Google)
10. Scaffolded Flutter App + Isolated FastAPI Backend

Module 63: Master Checklist & Progress Tracker

Implements **Rule 63: The Complete Factory Layers & Master 360+ Task Checklist**.

Tracks an application's progress across all 10 Layers and 16 Phases with 360+ verifiable checkboxes.

The 10 Factory Layers

1. **Layer 1:** Idea / Research
2. **Layer 2:** Product Management
3. **Layer 3:** Design System
4. **Layer 4:** Mobile Development
5. **Layer 5:** Backend / Database
6. **Layer 6:** AI / Automation
7. **Layer 7:** Testing / Security
8. **Layer 8:** CI/CD / Release
9. **Layer 9:** ASO / Marketing
10. **Layer 10:** Analytics / Growth

Usage

```
# Initialize tracking for App #037 ./tracker-cli init --app app-037 # View progress & scorecard ./tracker-cli status
--app app-037 # Complete a specific task ./tracker-cli complete --app app-037 --task TASK_042 # Export full Markdown
checklist for GitHub Issues ./tracker-cli export --app app-037 > apps/app-037/CHECKLIST.md
```

Module 64: Threat Modeling Module

Automates STRIDE / DREAD threat modeling for every mobile feature before production:

Asset -> Trust Boundary -> Entry Point -> Threat -> Attack Path -> Impact -> Mitigation -> Test

Module 65: Security Boundary Module

Enforces the core rule: **The mobile application must always be treated as potentially compromised.**

Separates untrusted mobile runtime artifacts from trusted backend boundaries.

Module 66: Zero-Trust API Module

Every API request undergoes a multi-layer verification chain:

Request -> TLS -> AuthN -> Token Validation -> User Lookup -> AuthZ -> Object Ownership -> Input Validation -> Rate Limit -> Execution

Module 67: Object-Level Authorization (BOLA Defense)

Guards against OWASP API1: Broken Object Level Authorization.

Every resource retrieval verifies that the requesting identity is the legitimate owner or tenant.

Module 68: Property-Level Authorization (BOPLA Defense)

Guards against OWASP API3: Broken Object Property Level Authorization.

Uses strict allow-lists for user-editable fields, completely preventing mass-assignment exploits.

Module 69: Function-Level Authorization

Guards against OWASP API5: Broken Function Level Authorization.

Enforces strict segregation of USER, ADMIN, SUPERADMIN, SUPPORT, and SERVICE privileges.

Module 70: Admin Security Module

Hardens internal management dashboards:

- Mandatory MFA / WebAuthn Passkeys
 - Maximum 15-minute idle session timeout
 - Step-up re-authentication required for destructive operations (drop DB, refund, price change)
-

Module 71: Break-Glass Access

Emergency elevated access protocol:

- Temporary time-bounded access (maximum 1 hour)
 - Requires explicit justification and dual authorization
-

- Generates immutable audit log
-

Module 72: Secrets Management

Enforces zero secrets in source code, APKs, or IPAs:

- Integration with AWS Secrets Manager / HashiCorp Vault / GCP Secret Manager
 - Client receives only ephemeral, scoped tokens
-

Module 73: Secret Rotation

Tracks secret age, rotation schedule, and automated revocation:

- Maximum secret lifespan: 90 days
 - Automated alerts when secrets exceed 75 days without rotation
-

Module 74: Key Management Module

Separates cryptographic keys by purpose:

- Data Encryption Keys (DEK)
 - Key Encryption Keys (KEK)
 - App Signing Keys
 - Webhook Signing Keys
 - OAuth Client Secrets
-

Module 75: Database Security Module

Database hardening rules:

- TLS enforced on all connections
 - Least-privileged database user roles (read-only replicas, write-only queues)
 - Query timeout enforced (5000ms max)
-

Module 76: Database Tenancy Isolation

Ensures complete logical or physical isolation for all 100+ apps:

- App A's backend and database cannot query App B's tables
-

Module 77: Backup & Disaster Recovery Verification

> "A backup that has never been successfully restored is not a proven backup."

Automates scheduled restore drills to a temporary isolated sandbox database.

Module 78: Disaster Recovery (RPO / RTO)

Defines and audits Recovery Point Objective (RPO) and Recovery Time Objective (RTO):

- Target RPO: ≤ 15 minutes
 - Target RTO: ≤ 60 minutes
-

Module 79: Multi-Region / Availability Module

Orchestrates multi-region failover and CDN edge caching for 1M+ user scale:

- Edge caching for static assets
 - Health checks with automatic failover between Primary and Secondary regions
-

Module 80: Autoscaling Module

Multi-metric autoscaler for backend containers:

- Scales on: CPU, Memory, Requests/Sec, Queue Depth, DB Connection Pool Saturation
-

Module 81: Load Shedding Module

Under extreme traffic spikes ($>95\%$ capacity), gracefully degrades non-critical features:

- Disables expensive AI generation & video rendering
 - Preserves authentication, reads, and core data operations
-

Module 82: Circuit Breaker Module

Prevents cascading outages when external providers fail (OpenAI, Stripe, Firebase):

- States: **CLOSED** (Normal), **OPEN** (Tripped - fast fail), **HALF_OPEN** (Trial recovery)
-

Module 83: Retry Safety & Idempotency

Prevents double-charging, duplicate orders, or double-spent AI credits:

- Mandates **Idempotency-Key** header on all mutating POST/PUT requests
 - Returns cached response if idempotency key has already been processed
-

Module 84: Rate-Limiting Matrix

Multi-dimensional rate limiting:

- Login: 10 attempts / 10 min per IP
 - AI Generation: 20 requests / min per User
 - File Upload: 5 uploads / min per Device
 - Password Reset: 3 requests / hour per Account
-

Module 85: Abuse-Prevention Module

Detects non-hacking abuse:

- Credential stuffing
- Referral farming (multiple accounts using same device fingerprint)
- Coupon & promotional credit abuse

Module 86: Bot / Automation Defense

Selective friction engine:

- Never puts CAPTCHAs everywhere
- Evaluates device integrity signals, request timing velocity, and headless browser signatures

Module 87: Replay-Attack Protection

Guards against captured request replay:

- Nonce verification with sliding timestamp window (300 seconds)

Module 88: Session Security Module

Multi-device session manager:

- Session listing, remote revocation, and automatic logout of all devices on password change

Module 89: Token Security Module

Cryptographic token lifecycle:

- Access tokens: 15-minute lifespan
- Refresh tokens: Rotated on every use; reuse detection revokes entire token family

Module 90: Device-Side Secure Storage

Hardware-backed secure storage:

- Android: EncryptedSharedPreferences with Android Keystore 256-bit AES-GCM
- iOS: Keychain Services with `kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly`

Module 91: App Integrity Module

Server-side verification of mobile environment attestation:

- Google Play Integrity API
- Apple App Attest / DeviceCheck
- Backend rejects requests if binary or device integrity fails

Module 92: Tamper & Reverse-Engineering Resilience

Multi-layered client-side defense:

- Debugger detection ([ptrace](#) / [sysctl](#))
- Root & Jailbreak file indicators
- Integrity signals forwarded to backend for risk scoring

Module 93: Network Security Module

Enforces TLS 1.3, strict certificate validation, WebSocket security, and absolute ban on cleartext HTTP traffic.

Module 94: Webhook Security Module

Hardens webhook receivers (Stripe, RevenueCat, Firebase):

- HMAC SHA-256 signature verification
- Max timestamp drift: 300 seconds
- Webhook idempotency ledger

Module 95: File-Upload Security Module

Never trusts extension or [Content-Type](#):

- Inspects magic bytes (JPEG: [FF D8 FF](#), PNG: [89 50 4E 47](#), PDF: [25 50 44 46](#))
- Generates randomized UUID filenames
- Direct upload via presigned S3/GCS URLs with 15-minute expiration

Module 96: SSRF Protection Module

Guards against OWASP API7: Server-Side Request Forgery.

Blocks user-supplied URLs from accessing private IPv4/IPv6 ranges or AWS/GCP metadata endpoints ([169.254.169.254](#)).

Module 97: Input Security Module

Comprehensive input validation firewall:

- SQL injection
- NoSQL injection
- Command injection
- Path traversal
- Prompt injection defense for LLMs

Module 98: Output Security Module

Production response shaping:

- Scrubs password hashes, internal database IDs, stack traces, and cloud provider metadata from all JSON responses.
-

Module 99: Error-Leak Prevention

Ensures production errors return a generic message + UUID correlation ID:

[Something went wrong.](#) [Reference: ERR-9F82A](#)

Detailed stack traces are logged only in internal telemetry.

Module 100: Security Logging Module

Structured JSON logging for security events:

- Login success / failure
 - Password reset
 - Privilege changes
 - Rate limit triggers
 - Never logs raw passwords, full credit card numbers, or auth tokens.
-

Module 101: Audit Log System

Immutable audit ledger:

[WHO](#) | [WHAT](#) | [WHEN](#) | [WHERE](#) | [TARGET](#) | [RESULT](#) | [CORRELATION_ID](#)

Module 102: Security Monitoring / SIEM

Aggregates auth logs and API metrics to detect security anomalies (e.g. 50 failed logins in 1 minute).

Module 103: Security Alerting Module

Multi-channel alerting (Slack, PagerDuty, Email) when security thresholds are breached.

Module 104: Incident Response Module

Step-by-step incident response playbook:

[DETECT](#) -> [VERIFY](#) -> [CONTAIN](#) -> [ERADICATE](#) -> [RECOVER](#) -> [ROTATE CREDENTIALS](#) -> [PATCH](#) -> [POST-MORTEM](#)

Module 105: Automatic Emergency Controls

Remotely freeze operations in case of an attack:

- Disable AI generation
 - Freeze payments
 - Block abusive IP/account
 - Force user re-authentication
-

Module 106: Force-Upgrade System

Protects users against critical client vulnerabilities:

- [min_supported_version](#): Hard block if older
 - [recommended_version](#): Soft prompt
 - [blocked_versions](#): Explicit list of revoked releases
-

Module 107: Remote Kill Switch

Secure remote feature toggles:

- Kill switches must be authenticated via dual-operator signatures
 - Cached locally with TTL fallback
-

Module 108: Supply-Chain Security

Audits dependencies across all 60+ modules:

- Checksum lockfile verification ([pubspec.lock](#), [package-lock.json](#), [poetry.lock](#))
 - Automated CVE vulnerability scanning
-

Module 109: Software Bill of Materials (SBOM)

Generates CycloneDX / SPDX SBOM JSON for every production build:

- Documents all direct & transitive dependencies, licenses, and hashes
-

Module 110: Dependency Policy & Evaluation

Pre-install vetting for new packages:

- License check (MIT, Apache 2.0, BSD allowed; GPL/AGPL rejected)
 - Maintainer activity within last 12 months
-

Module 111: Secure Build Environment

Hardens CI/CD runners:

- Ephemeral containers with zero persistent storage
 - Secrets injected only into environment memory, never written to disk
-

Module 112: Reproducible Builds

Addresses OWASP MASWE: Non-reproducible builds.

Ensures identical inputs yield byte-identical build artifacts.

Module 113: Artifact Signing & Provenance

Signs production artifacts with SHA-256 and records build provenance (Git commit, build ID, signer identity).

Module 114: CI/CD Pipeline Security

Prevents pipeline poisoning:

- Step-level least-privilege permissions (`permissions: read-all`)
 - Secrets masked in console output
-

Module 115: Git Branch Protection

Enforces GitHub branch protection policies:

- Mandatory PR reviews
 - Passing status checks (SAST, Tests, Linter)
 - Signed commits required
-

Module 116: Static Application Security Testing (SAST)

Scans Dart and Python code for hardcoded secrets, insecure random generators, and missing SSL checks.

Module 117: Dynamic Application Security Testing (DAST)

Automated API fuzzer testing for:

- SQL injection payloads
 - BOLA ID manipulation (`/api/v1/user/1` vs `/api/v1/user/2`)
 - Rate limit thresholds
-

Module 118: Penetration Testing

Standard operating procedure & checklist for mobile and API penetration testing.

Module 119: OWASP MASVS / MASTG / MASWE Test Suite

Conforms to OWASP Mobile Application Security Verification Standard (MASVS v2.0) and MASWE 1.0 (78 catalogued weaknesses).

Module 120: Privacy Engineering & Data Mapping

Maintains an automated data map:

[Data Item](#) -> [Why Collected](#) -> [Where Stored](#) -> [Who Accesses](#) -> [Retention Period](#) -> [Deletion Mechanism](#)

Module 121: Data Classification

Classifies all data into 5 tiers:

1. [PUBLIC](#)
 2. [INTERNAL](#)
 3. [CONFIDENTIAL](#)
 4. [SENSITIVE](#)
 5. [HIGHLY_SENSITIVE](#) (Credentials, Biometric hashes, Payment tokens)
-

Module 122: Data Retention Engine

Automates scheduled data purging:

- Access logs: purged after 90 days
 - Soft-deleted accounts: purged after 30 days
 - Temporary upload files: purged after 24 hours
-

Module 123: Account-Deletion Pipeline

Apple Guideline 5.1.1 & GDPR Article 17 (Right to Erasure) compliant pipeline:

[Request Deletion](#) -> [Verify](#) -> [Anonymize Primary DB](#) -> [Delete Files](#) -> [Revoke Sessions](#) -> [Cancel Subscriptions](#)

Module 124: Child & Age Safety

COPPA & Age-Appropriate Design Code (AADC) compliance:

- Age gate before registration
 - No behavioral ads or tracking for users under 13
-

Module 125: Regulatory & Store Compliance Engine

Continuously audits compliance:

- Google Play Data Safety
 - Apple App Privacy Nutrition Labels
 - GDPR / CCPA privacy policies
-

Module 126: Third-Party Risk Module

Audits all external SaaS & SDK integrations (OpenAI, Stripe, Firebase, Mixpanel):

- Documents data egress, country of processing, and compliance posture.
-

Module 127: Third-Party Failure Isolation

Ensures that if an external provider (Analytics, AI, Push) goes down, the core mobile app remains operational.

Module 128: Cost-Abuse Protection

Hard limits to prevent massive cloud/AI bills:

- Per-user daily cost caps (\$1.00/day for free users, \$15.00/day for pro users)
 - Global kill-switch when daily spend reaches \$500.00
-

Module 129: Billing Anomaly Detection

Monitors sudden spikes in API, SMS, or AI consumption (e.g. 5x increase over historical baseline).

Module 130: Cache Security

Guards Redis / Memcached against cross-user cache contamination:

- Prefixes all keys with `tenant_id:user_id:`
 - Enforces TTL on all cached records
-

Module 131: Queue Security

Protects message queues (RabbitMQ / SQS / Redis Streams):

- Message signature verification
 - Poison pill dead-letter queue (DLQ) after 3 failed attempts
-

Module 132: Job Idempotency

Ensures background workers never process duplicate tasks (e.g. sending 2 emails, processing 2 invoices).

Module 133: Observability Pillars

Unified telemetry engine:

- Structured JSON Logs
 - Prometheus / Datadog Metrics
 - OpenTelemetry Distributed Traces
-

Module 134: Distributed Tracing

Propagates [traceparent](#) and correlation IDs across Flutter -> API Gateway -> Backend -> Database.

Module 135: Database Performance Engine

Optimized for 1M user scale:

- Connection pooling limits (max 20 per pod)
 - Slow query threshold logger (> 200ms)
 - Mandatory index verification on foreign keys
-

Module 136: Pagination & Query Limits

Never permits unbounded [SELECT *](#):

- Mandatory cursor pagination
 - Hard limit of 100 items per page
-

Module 137: Search Abuse Protection

Protects search endpoints against ReDoS and resource exhaustion:

- Max query length: 100 characters
 - Rejects wildcards at query start ([*query](#))
-

Module 138: API Versioning & Deprecation

Manages API lifecycle:

- Returns [Sunset: <date>](#) and [Deprecation: true](#) HTTP headers for deprecated endpoints.
-

Module 139: API Inventory & Shadow Endpoint Detection

OWASP API9 compliance:

- Catalog of all registered production endpoints
- Automated alerts if traffic is observed on unregistered endpoints.

Module 140: Edge Security Headers

Injects industry-standard HTTP security headers:

- `Strict-Transport-Security: max-age=63072000; includeSubDomains; preload`
- `X-Content-Type-Options: nosniff`
- `X-Frame-Options: DENY`
- `Content-Security-Policy: default-src 'none'`

Module 141: DDoS Resilience

Cloudflare / AWS Shield edge defense:

- Layer 7 rate-shedding rules
- Blocks geo-IP anomalies and known malicious autonomous systems (ASNs)

Module 142: Network Segmentation

VPC 3-tier network topology:

1. Public Subnet: Load balancer only
2. Private Subnet: API containers & background workers
3. Isolated DB Subnet: PostgreSQL / Redis (no internet access)

Module 143: Cloud IAM Least Privilege

Generates least-privilege IAM policies:

- Application pods receive only read access to specific S3 buckets
- Zero wildcard `*` permissions

Module 144: Environment Isolation

Strict audit preventing dev/staging from communicating with production databases or buckets.

Module 145: Production Data Protection

Never dumps real customer records into developer laptops:

- Synthetic test data generator producing realistic mock users and transactions.

Module 146: Support-Tool Security

Hardens customer support dashboards:

- Masks PII (emails masked as `j***@gmail.com`, phone numbers as `+1 ***-***-1234`)

- Support agents cannot view raw passwords or execute arbitrary database updates.
-

Module 147: Social-Engineering Defense

Standard operating procedure for account recovery:

- Mandates out-of-band identity verification before password reset or email changes.
-

Module 148: Fraud Detection

Flags high-risk transactions:

- Payment velocity checks (> 3 failed card attempts in 10 minutes)
 - Disposable email domain blocking
-

Module 149: Release Rollback System

1-command atomic rollback:

- Reverts Kubernetes / Docker container tags to prior known-good SHA within 30 seconds.
-

Module 150: Canary Releases

Progressive rollout engine:

- Steps: 1% -> 5% -> 25% -> 50% -> 100%
 - Automated rollback if canary error rate exceeds 0.5%
-

Module 151: Feature-Gated Migrations

Dual-schema backward compatibility:

- Phase 1: Add new column (nullable)
 - Phase 2: Dual-write from new app version
 - Phase 3: Backfill old rows
 - Phase 4: Make column non-null
-

Module 152: Security Regression Suite

> "A vulnerability should never be fixed only once."

Every past vulnerability is converted into a permanent automated regression test.

Module 153: Security Scorecard per App

14-dimension security scorecard release gate:

Module 154: Security Release Gate

Pipeline gatekeeper blocking production deployment unless 100% of critical security checks pass.

Module 155: 1M User Scale Gate

Simulates 10k, 50k, 100k, and 1M registered user loads:

- Peak RPS capacity testing
 - Database read/write IOPS under load
-

Module 156: Capacity Planning Engine

Calculates required compute pods, database storage, and egress bandwidth based on projected MAU.

Module 157: Chaos & Failure Testing

Simulates real-world infrastructure failures:

- Simulates sudden database dropouts
 - Simulates Redis cache eviction
 - Verifies system recovers automatically within 10 seconds
-

Module 158: Business Continuity Module

Playbooks for major external failures:

- Cloud provider outage
 - Stripe / RevenueCat payment gateway outage
 - OpenAI / AI provider outage
-

Module 159: Vulnerability Disclosure Program (VDP)

Provides standard [security.txt](#) and responsible disclosure guidelines for security researchers.

Module 160: Security Patch Pipeline

Automates patching shared factory modules across all 100+ apps:

[Vulnerability Alert](#) -> [Patch Shared Module](#) -> [Run Regressions](#) -> [Bulk Build Affected Apps](#) -> [Release](#)

Module 161: Blast-Radius Control

Never auto-updates all 100 apps at once when a shared module changes:

- Enforces semantic versioning on shared packages
- Gradual rollout of shared components to 5 canary apps before widespread deployment

Module 162: Factory Security Monitor

Central CLI & dashboard aggregating security posture across the 100-app portfolio:

- Compliance status
- Expiring certificates & keystores
- Outdated dependencies

Module 163: Final Production Architecture

The unified synthesis of the complete 163-module production system:

[illegible]

Security Philosophy

ASSUME BREACH ↓ NEVER TRUST THE CLIENT ↓ VERIFY EVERY REQUEST ↓ LEAST PRIVILEGE EVERYWHERE ↓ ISOLATE EVERY PRODUCT ↓
LIMIT EVERY EXPENSIVE ACTION ↓ LOG SECURITY EVENTS ↓ DETECT ANOMALIES FAST ↓ CONTAIN AUTOMATICALLY ↓ RECOVER FROM
BACKUPS ↓ PATCH + REGRESSION TEST