

Advanced Flutter Architecture, State Management and Deployment Training Course

13 – 24 December 2026
Toronto

AGILE LEADERS
Training Center

AGILE LEADERS
Training Center

Advanced Flutter Architecture, State Management and Deployment Training Course

Ref: 1897_241383 **Date:** 13 – 24 December 2026 **Location:** Toronto **Fees:** 16000 Euro

Overview

Advanced Flutter Architecture, State Management and Deployment Training Course is a two-week course for mobile software engineers, application architects, development leads, and technical delivery specialists who leave with a production-ready Enterprise Flutter Architecture Blueprint. The course resolves architectural drift, uncontrolled state propagation, and pipeline vulnerabilities by teaching structured Clean Architecture, predictable reactive state management, and hardened delivery workflows. Participants establish testable layered codebases, integrate secure hardware-backed storage, and implement end-to-end continuous deployment. Agile Leaders Training Center delivers this advanced Flutter architecture, state management and deployment course.

Who Should Attend

- Mobile engineering teams responsible for cross-platform application structure and codebase scalability
- Software architecture leads responsible for enforcing Clean Architecture boundaries and dependency injection
- Mobile technical leads responsible for complex state synchronization, caching, and reactive data streams
- Application security engineers responsible for mobile code hardening, key storage, and vulnerability mitigation
- DevOps and release engineers responsible for automated mobile build pipelines and store distribution

The course assumes participants can build foundational Flutter applications with Dart and consume REST APIs; it leaves out introductory programming syntax, UI design basics, and developer account administrative enrollment.

Departments and Industries

The course supports mobile engineering governance and delivery across software-driven enterprises and technical operations.

- Digital banking and financial technology services
- Healthcare systems and patient portal engineering
- Telecommunications and consumer self-service platforms
- Enterprise mobile application development units
- Software engineering and cloud architecture departments

Learning Objectives

By the end of this course, participants will be able to:

- Build modular Flutter codebases adhering to Clean Architecture layers and dependency injection principles
- Compare Bloc, Cubit, and Riverpod patterns to select optimal state management solutions
- Apply cryptographic storage mechanisms and secure key management across iOS and Android runtimes
- Analyze network and runtime vulnerabilities against OWASP MASVS v2.1.0 security requirements
- Evaluate static analysis, security linting, and automated testing results across continuous integration pipelines
- Deploy release-ready builds to digital storefronts using Fastlane and automated delivery pipelines

Course Agenda

Day 1: Clean Architecture and Layered Boundaries

- Presentation, Domain, and Data Layer Boundary Mapping
- Domain Entity and Value Object Modeling with Dart
- Data Transfer Object and Entity Mapping Matrix
- Repository Interface Contracts and Functional Error Handling
- Service Locator Pattern Configuration with get_it and injectable

Day 2: Reactive Repositories and Local Caching

- Reactive Stream Controller Setup in Data Source Layers
- Encrypted Hive and SQLite Relational Cache Architectures
- Optimistic UI Update and Cache Invalidation Protocol
- Offline-First Synchronization Engine and Conflict Resolution
- Remote Data Source Integration with Dio Interceptor Pipelines

Day 3: State Management with Bloc and Cubit

- Event-to-State Transformation Pipeline Design in Bloc
- Cubit State Emission and Transient State Encapsulation
- BlocObserver Middleware Implementation for Global Telemetry
- Widget Rebuild Optimization and BuildWhen Predicate Filters
- MultiBlocProvider Scoping and HydratedBloc State Persistence

AGILE LEADERS
Training Center

Day 4: Declarative State Architecture with Riverpod

- Riverpod 2.0 Provider Hierarchy and AutoDispose Strategies
- Notifier and AsyncNotifier Lifecycle Management
- Family Provider Parameterization and Scoped Cache Eviction
- StateNotifier Transition to Code-Generated Riverpod Patterns
- WidgetRef Listening Mechanics and Selective Rebuild Diagnostics

Day 5: Mobile Security and OWASP MASVS Baselines

- OWASP MASVS v2.1.0 Storage and Cryptography Category Audit
- OWASP MASTG Test Case Execution for Mobile Binary Profiling
- Secure In-Memory Data Sanitization and Memory Scrubbing
- Platform Channel Bridging for Native Biometric Authentication
- Secure Clipboard and App Switcher Screen Masking Handlers

Day 6: Cryptographic Storage and Hardware Key Management

- Flutter Secure Storage Integration with iOS Keychain Access
- Android Keystore Master Key Cryptographic Binding Protocol
- Hardware-Backed AES-256 GCM Data Encryption Implementation
- Sensitive Token Rotation and Revocation Workflow Engine
- Device Jailbreak and Root Detection Signal Interceptors

Day 7: Network Security, Certificate Pinning, and Tamper Resistance

- TLS Fingerprint and Public Key Pinning Implementation in Dio
- Dynamic Network Payload Encryption and Signature Verification
- Code Obfuscation and Symbol Stripping via Flutter Build Flags
- Anti-Tamper Integrity Checks and Binary Signature Validation
- Reverse Engineering Defense and Debugger Attachment Detection

Day 8: Automated Testing Suites and Quality Governance

- Unit Testing of Domain Use Cases with Mocktail Stubs
- BlocTest and Stream Verification for Asynchronous State
- Golden Widget UI Snapshot Testing and Difference Reporting
- Static Application Security Testing and Security Lint Rules
- End-to-End Integration Testing with Patrol Framework Drivers

Day 9: Mobile DevOps and CI/CD Automation

- Fastlane Match Keystore and Provisioning Profile Automation
- GitHub Actions Workflow Configuration for Flutter Multi-Flavors
- Automated Semantic Versioning and Changelog Generation
- Store Metadata, Screenshot, and Binary Upload Automation
- Secure Pipeline Secrets and Environment Artifact Management

Day 10: Production Architecture Delivery and Capstone

- Exercise: Audit Application Codebases for Clean Layer Violations
- Exercise: Refactor Nested State Management to Reactive Bloc Flow
- Exercise: Harden Local Storage and Network Pinning Configurations
- Exercise: Construct an Automated Fastlane Multi-Environment Pipeline
- Capstone Exercise: Build and Deliver a Hardened Enterprise Flutter Architecture Blueprint

Practical Exercises

Suggested activities translate architectural, security, and deployment principles into reliable enterprise mobile solutions.

- Suggested activity: refactor a tightly coupled Flutter application into decoupled Clean Architecture layers with dependency injection
- Suggested activity: build a fault-tolerant reactive state pipeline using Bloc and Riverpod caching
- Suggested activity: implement hardware-backed encryption, certificate pinning, and integrity checks aligned with OWASP MASVS v2.1.0
- Suggested activity: construct an automated continuous integration pipeline deploying signed builds via Fastlane

FAQs

Who suits the Advanced Flutter Architecture, State Management and Deployment Training Course and what does it assume?

The course suits mobile engineers and technical leads building scalable, production-grade applications. Participants should already construct basic Flutter apps and consume APIs; the course leaves out basic Dart syntax, UI layout fundamentals, and developer account administrative setups.

How does advanced Flutter architecture differ from foundational mobile app development courses?

Foundational development courses focus on basic UI widgets and simple API calls. Advanced Flutter architecture focuses on Clean Architecture layering, reactive state patterns with Bloc and Riverpod, hardware-backed cryptography, OWASP MASVS hardening, and automated deployment pipelines.

AGILE LEADERS
Training Center

Why is Clean Architecture implemented in advanced Flutter applications?

Clean Architecture isolates business logic from UI frameworks and external data sources. This separation enables independent testability, prevents regressions during package upgrades, and allows enterprise teams to scale complex codebases across multiple distributed developers cleanly.

How are state management decisions made between Bloc and Riverpod in advanced Flutter architecture?

Bloc suits event-driven enterprise workflows requiring strict traceability, observable transitions, and explicit state auditing. Riverpod offers lightweight, compile-safe dependency injection and caching, making it ideal for reactive data binding with minimal boilerplate code.

What role does OWASP MASVS play in advanced Flutter deployment?

OWASP MASVS provides standard mobile security verification requirements. Advanced Flutter deployments utilize it to audit cryptographic storage, enforce TLS certificate pinning, detect rooted devices, and prevent reverse engineering before distributing release binaries to production app stores.

Conclusion

Participants take back a production-ready Enterprise Flutter Architecture Blueprint that establishes decoupled presentation, domain, and data layers. The blueprint implements robust state management frameworks and hardware-backed cryptographic storage across multi-platform targets. It establishes automated continuous integration workflows that secure and accelerate enterprise mobile app releases.