

Video Converter Android Constitution

Core Principles

I. Component-Driven Development (NON-NEGOTIABLE)

Every feature is implemented following atomic design principles with React Native components; Components must be self-contained, reusable, and follow single responsibility principle; Clear TypeScript interfaces required for all component props with comprehensive documentation; No inline styles - all styling through Tailwind CSS/NativeWind

II. Maximum Productivity and TypeScript Excellence

Development process focused on modern React Native patterns with TypeScript 5.0+ strict configuration; Functional programming approach with custom hooks for reusable logic; ES6+ features mandatory (destructuring, async/await, template literals); Zero **any** types allowed - strict type safety enforced

III. Test-Coverage (NON-NEGOTIABLE)

All features must be covered by Jest unit tests and React Native Testing Library component tests; Minimum 80% code coverage for utilities and services; TDD approach: Tests written → User approved → Tests fail → Then implement; Performance tests for video processing workflows mandatory

IV. Integration Testing

Focus areas requiring integration tests: Video conversion end-to-end workflows, File system operations, Background processing, Component interaction patterns; Test with real video files and various Android devices; Create test dataset snapshots for repeatable testing

V. Architecture & Performance Standards

Follow atomic design folder structure (atoms/molecules/organisms/templates); FFmpeg Kit integration with chunked processing and memory management; Background processing with real-time progress tracking; App launch time < 2 seconds, memory usage < 200MB baseline

Technology Stack & Design System

Required Technologies

React Native 0.73+ with New Architecture (Fabric/TurboModules); TypeScript 5.0+ with strict configuration; NativeWind for Tailwind CSS styling; FFmpeg Kit React Native for video processing; React Navigation v6 with type-safe navigation; React Native File System (RNFS) for file operations

Design System Standards

Color palette: Primary Blue #2f6690, Secondary Blue #3a7ca5, Light Gray #d9dcd6, Dark Blue #16425b, Light Blue #81c3d7; Typography scale with consistent font sizes and weights; 8px base spacing unit;

Component variants standardized (size, color, state); WCAG 2.1 AA accessibility compliance

Development Workflow & Quality Gates

Code Quality Requirements

ESLint and Prettier configuration enforced; Husky git hooks for pre-commit quality gates; JSDoc comments for all functions and complex logic; Comprehensive README files for each major module; Architecture Decision Records (ADRs) for significant changes

Git Workflow & Versioning

Semantic versioning (MAJOR.MINOR.PATCH); Branch strategy: main/develop/feature/hotfix/release; Pull request requirements: code review, automated testing, documentation updates; Manual testing on physical Android devices required

Governance

Constitution supersedes all other development practices; All PRs/reviews must verify compliance with component-driven development and TypeScript excellence; Complexity must be justified with performance implications documented; Security reviews required for video processing and file system changes; Amendments require team consensus, documentation, and migration plan

Version: 1.0.0 | **Ratified:** September 17, 2025 | **Last Amended:** September 17, 2025