MnV -Spec-Driven Development Document

Project: AI Content Generation Pipeline for MnV

Version: 1.0

Date: 28-Aug-2025

Authors: MnV Architect

# 1. System Overview

* \*\*Purpose\*\*: Automate the generation, production, distribution, and monetization of AI-generated content (short stories → narrated videos → uploaded to platforms).
* \*\*Scope\*\*: End-to-end pipeline covering content generation, video assembly, platform uploads, analytics, optimization, and monetization.
* \*\*Exclusions\*\*: Manual editing, live interaction, or custom creative writing beyond AI output.
* \*\*Stakeholders\*\*: Content creators, social media managers, platform admins, dev/ops teams.
* \*\*Constraints\*\*: Dependent on multiple third-party APIs, Internet + GPU required for image/video generation, Costs per video: $0.26–0.85.

# 2. Functional Requirements

* FR1: Content Generation – Auto-generate stories (300–500 words), narration, images, and music. Retry logic (3 retries on API failure).
* FR2: Video Production – Assemble videos in 16:9, 9:16, 1:1 formats with transitions, overlays, and color correction.
* FR3: Platform Distribution – Upload to YouTube, Instagram, TikTok, Facebook with retry logic and logs.
* FR4: Monetization – Support creator funds, ad revenue, affiliate links, eBooks, Patreon/Ko-fi integration.
* FR5: Analytics & Optimization – Track engagement, revenue, suggest strategy changes (ML-based).

# 3. Non-Functional Requirements

* Performance: Pipeline ≤10 min/video; story generation ≤30s.
* Reliability: ≥99% uptime; auto-retry on failures.
* Scalability: Horizontal scaling on Kubernetes/Docker.
* Security: API key encryption, rate limiting.
* Compliance: GDPR, COPPA, copyright moderation.
* Maintainability: ≥80% test coverage, linting.
* Cost: Alert if >$1/video.

# 4. Data Models

* Story: {id, content, genre, theme, length, created\_at}
* Video: {id, story\_id, format, path, status}
* Upload: {id, video\_id, platform, url, metrics}
* Analytics: {id, upload\_id, engagement\_json, revenue, timestamp}

# 5. Interfaces & APIs

* Internal APIs: POST /generate/story, POST /assemble/video, POST /upload/{platform}, GET /analytics/{video\_id}.
* External APIs: OpenAI, Claude, Mistral, ElevenLabs, Mubert, DALL-E, YouTube, Instagram, TikTok, Facebook.

# 6. Architecture

* Hybrid Architecture: Next.js frontend, FastAPI backend, Airflow/Prefect orchestration, PostgreSQL + Redis, Prometheus + Grafana monitoring, Docker/Kubernetes deployment.

# 7. Testing Strategy

* Unit Tests: Core modules.
* Integration Tests: End-to-end with mocks.
* E2E Tests: Full pipeline runs.
* Acceptance Tests: Verify FR1–FR5.
* Performance Tests: Simulate 100 videos/day.
* Security Tests: API keys, rate limits.

# 8. Implementation Roadmap

* Phase 1 (Weeks 1–2): Core pipeline with YouTube uploads.
* Phase 2 (Weeks 3–4): Multi-platform + basic analytics.
* Phase 3 (Weeks 5–8): Monetization + optimization.
* Phase 4 (Ongoing): Risk mgmt, scaling, cost optimization.

# 9. Risk Mitigation

* Content Risks: Filters, copyright checks, moderation queue.
* Technical Risks: API fallback chains, retries, multi-region deployment.
* Business Risks: Diversify platforms, adjust to policy shifts.

# 10. Success Metrics

* Content: Completion >40%, engagement >5%.
* Technical: Error rate <2%, recovery <5 min.
* Business: Revenue/video ≥$0.50, MRR growth ≥10%/mo.