1. System Architecture (Cost-Effective → Scalable)

Phase 1: MVP Architecture (Low Cost)

Goal: Launch fast, keep infra simple, avoid over-engineering.

```
Mobile App (Flutter)
        ↓
API Gateway (Nginx)
        ↓
Backend (Node.js / NestJS – Monolith)
        ↓
PostgreSQL (Main DB)
        ↓
Object Storage (Videos)
        ↓
CDN (Video Delivery)
```

Components Explained

- **Mobile App (Flutter):** Single codebase for Android & iOS
- **Backend (NestJS):** Auth, feed logic, unlock logic, admin APIs
- **PostgreSQL:** Users, series, episodes, progress
- **Object Storage:** Raw + processed videos
- **CDN:** Cheap global delivery (Cloudflare-style)
- **AI Worker (Single VM):** FFmpeg + Python scripts

Phase 2: Growth Architecture (Scalable)

```
Clients (Mobile / Web)
        ↓
API Gateway
        ↓
Microservices
 ├── Auth Service
 ├── Content Service
 ├── Monetization Service
 ├── Analytics Service
        ↓
Databases
 ├── PostgreSQL
```

```
 ├── Redis (Cache)
       ↓
Media Layer
 ├── Object Storage
 ├── CDN
       ↓
AI Processing Layer
 ├── Video Conversion Workers
 ├── Subtitle Workers
```

Why This Scales Well

- Stateless services
- AI workers auto-scale
- CDN handles traffic spikes
- DB can be read-replicated

Data Flow (End-to-End)

1. Creator uploads video
2. Stored in raw storage
3. AI worker converts to 9:16
4. Output stored in CDN bucket
5. Metadata saved in DB
6. Episode becomes live
7. User watches → progress tracked

---

2. Admin Dashboard UX (Web)

Primary Users: Internal team, studios, content managers

Admin Login

- Email / Password
- Role-based access (Admin, Editor)

Dashboard – Overview

Widgets: - Total Series - Total Episodes - Daily Views - Revenue Snapshot

---

Series Management Screen

UI Sections: - Create New Series - Series List Table

Create Series Fields: - Title - Language (Hindi / Urdu) - Genre (Multi-select) - Description - Poster Upload

---

Episode Upload Screen

Step-Based Flow: 1. Select Series 2. Upload Raw Video 3. Episode Number 4. Lock Type (Free / Coin / Ad) 5. Trigger AI Processing

Status Indicators: - Uploaded - Processing - Ready - Published

---

AI Processing Monitor

- Job Queue
- Processing Time
- Error Logs
- Manual Re-trigger button

Monetization Control

- Set coin price per episode
- Enable / disable ads
- Bulk pricing rules

Analytics Screen

- Episode completion rate
- Drop-off point graph
- Top series
- Revenue by series

---

3. Creator Onboarding Flow

Goal: Let studios & creators upload content without tech friction.

Step 1: Creator Registration

- Studio / Creator name
- Email & phone
- Country
- Content language

Step 2: Verification

- ID / Studio proof upload
- Manual approval (early stage)
- Auto-approval later

Step 3: Creator Dashboard

Sections: - My Series - Upload Episode - Earnings - Analytics

---

Step 4: Content Upload Flow

1. Select Series or Create New
2. Upload raw video (any format)
3. Choose episode length preference
4. Submit for AI processing

Step 5: Review & Publish

- Preview vertical output
- Edit title & description
- Publish episode

Step 6: Earnings & Insights

- Views per episode
- Ad revenue
- Coin unlocks
- Payout history

Creator Monetization Model (Future)

- Revenue share (ads)
- Exclusive content bonus
- Sponsored series deals

Why Creators Will Love This

- No manual editing needed
- Vertical AI conversion
- Clear monetization
- Local audience reach

This setup keeps initial costs low, allows manual control early, and smoothly transitions into a creator-driven scalable platform.