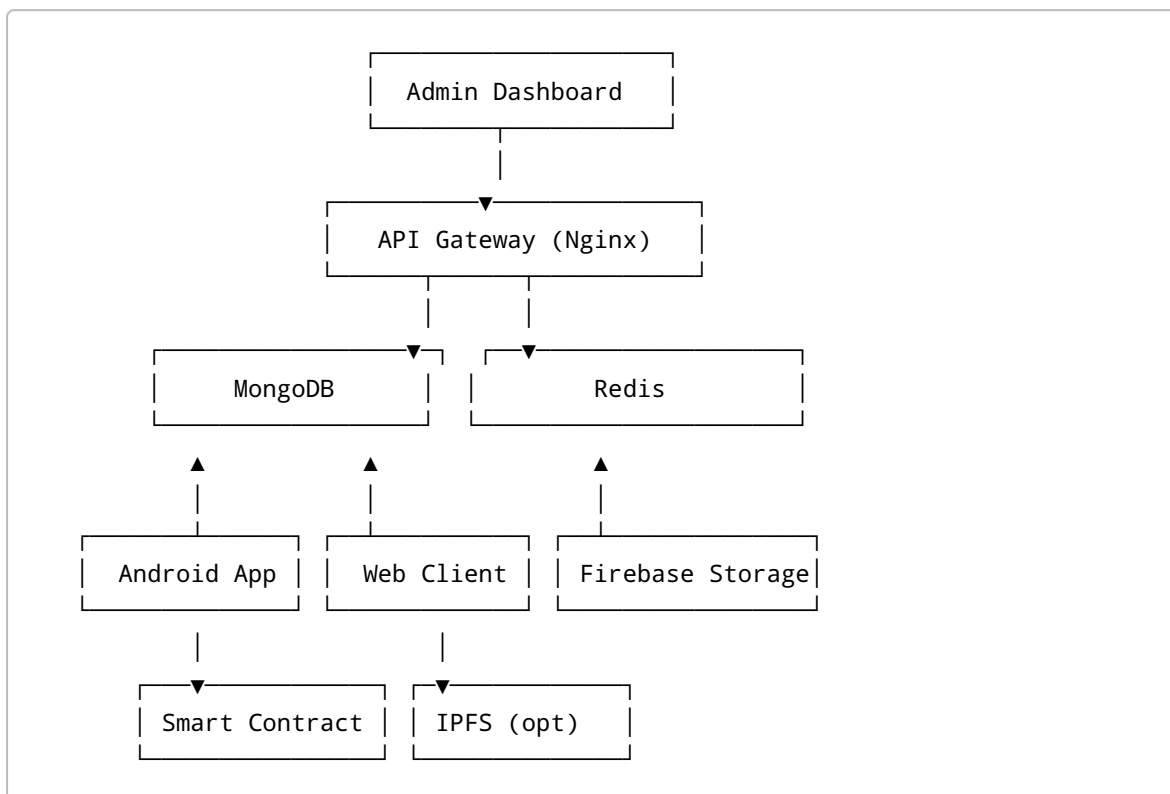


TrueCred System Design (Modular & Industry-Grade)

Design Principles

- **Modular:** Clearly separated concerns across mobile, web, backend, blockchain
- **Hybrid Architecture:** Decentralized proofs with centralized services
- **Mobile-First:** Core flows start from Android app
- **Multi-Role:** Distinct flows for users, institutions, companies, and admins

1. Main System Architecture Overview



2. Mobile App Architecture (Android - Java)

TrueCred Android App

- └─ AuthModule (Login, Register, JWT)
- └─ ProfileModule (Edit, View, Add Skills, Work, Certs)
- └─ CredentialModule (Upload, OCR Extract, Verify)
- └─ ResumeModule (Auto-generate, Download)
- └─ JobModule (List, Apply, Filter)
- └─ FreelanceModule (Browse gigs, Bid)
- └─ QRScanModule (Scan to Verify)

- └─ MessagingModule (Chat, Alerts)
- └─ Network Layer (Retrofit + Interceptors)

Local Storage: SQLite (Resume cache), SharedPreferences (Session)

3. 🏠 Web Platform Architecture (React)

```
React Web App
└─ AuthContext (Firebase, JWT)
└─ DashboardLayout
  │   └─ ProfileBuilder
  │   └─ Resume Generator
  │   └─ Job/Freelance Boards
  │   └─ Certificate/Exp Upload & Status
└─ Institution Panel
  │   └─ Incoming Requests, View Docs, Approve/Reject
└─ Recruiter Panel
  │   └─ Post Job, View Applicants, Verify Exp
└─ AdminPanel
  │   └─ Reports, Moderation, Analytics
└─ State: Redux Toolkit
└─ Styling: Tailwind CSS
```

4. Server & API Infrastructure (Node.js + Express)

Services

```
└─ AuthService
  │   └─ /login, /register, /refresh
└─ ProfileService
  │   └─ /profile, /public/:id
└─ VerificationService
  │   └─ /cert/upload, /exp/upload, /request, /status
└─ ResumeService
  │   └─ /resume/gen
└─ JobService
  │   └─ /jobs/list, /apply
└─ FreelanceService
  │   └─ /freelance/list, /post
└─ BlockchainService
  │   └─ /hash/store, /hash/verify
└─ QRService
  │   └─ /verify/:hash
```

```
├─ AdminService
├─ /admin/users, /reports
```

Infra Tools

- **MongoDB Atlas** for data
- **Redis** for sessions & caching
- **Firebase** for file storage
- **ethers.js/web3.js** for blockchain integration
- **Dockerized** microservices
- **CI/CD**: GitHub Actions, Railway/Vercel

5. Blockchain Subsystem (Polygon Mumbai)

Smart Contracts

- `verifyCertificate(address user, bytes32 certHash)`
- `verifyExperience(address user, bytes32 expHash)`
- `isVerified(bytes32 hash)`

Workflow

- Hash created from metadata (cert/experience string)
- Signed by institution or company
- Stored on-chain via smart contract
- Verifiable through resume QR or user profile

Stored On-Chain: Hash, verifier address, timestamp **Off-Chain:** Document (Firebase/IPFS), personal info

6. 🏠 GitHub Repo Structure (Monorepo Layout)

```
truecred/
├─ contracts/           # Solidity contracts
├─ backend/             # Node.js services (Express + MongoDB)
│   └─ api/
│       └─ config/
├─ web/                 # React frontend
├─ android/             # Android app (Java)
├─ utils/               # Common hashing, validators
├─ scripts/             # Deployment & automation
└─ README.md
```

Let me know if you want diagrams, database schemas, or data flow mapping next.