

Rendiff FFmpeg API Service

Professional Scope Document v2.0

Document Classification: Confidential

Version: 2.0

Date: January 27, 2025

Prepared for: Rendiff Development Team

Document Owner: Senior Backend Development

Table of Contents

- 1. [Executive Summary](#)
 - 2. [Business Context](#)
 - 3. [Project Scope](#)
 - 4. [Technical Architecture](#)
 - 5. [API Specification](#)
 - 6. [Infrastructure Requirements](#)
 - 7. [Security & Compliance](#)
 - 8. [Performance Requirements](#)
 - 9. [Implementation Roadmap](#)
 - 10. [Risk Assessment](#)
 - 11. [Cost Analysis](#)
 - 12. [Success Criteria](#)
 - 13. [Appendices](#)
-

1. Executive Summary

1.1 Project Overview

Rendiff FFmpeg API Service is a cloud-native media processing platform that transforms FFmpeg's capabilities into a scalable, developer-friendly REST API. The service addresses the growing demand for programmatic video processing by providing a consolidated, intelligent API that simplifies complex media operations while maintaining enterprise-grade reliability.

1.2 Strategic Objectives

- **Democratize Access:** Enable developers to leverage professional media processing without FFmpeg expertise

- **Reduce Complexity:** Consolidate hundreds of FFmpeg options into intuitive API parameters
- **Ensure Scalability:** Support processing from startup MVPs to enterprise-scale operations
- **Maintain Quality:** Provide built-in quality metrics (VMAF, PSNR) for data-driven encoding decisions

1.3 Key Deliverables

1. **Consolidated REST API** with 12 endpoints replacing traditional 70+ endpoint designs
2. **Intelligent parameter system** supporting natural language-like requests
3. **Enterprise-grade infrastructure** supporting 1,000+ concurrent jobs
4. **Comprehensive SDK suite** for major programming languages
5. **Future-ready architecture** prepared for web UI integration (Part 2)

1.4 Investment Summary

- **Development Timeline:** 24 weeks
- **Initial Investment:** \$185,000
- **Monthly Operating Cost:** \$8,500 (at scale)
- **Break-even:** 850 paid users or 50 enterprise clients

2. Business Context

2.1 Market Analysis

The global video processing market is experiencing unprecedented growth:

- **Market Size:** \$7.1B (2024) → \$15.3B (2029) CAGR 16.7%
- **API Economy:** 83% of enterprises use APIs for digital transformation
- **Developer Demand:** 2.3M searches/month for "FFmpeg API" solutions

2.2 Competitive Landscape

Competitor	Strengths	Weaknesses	Renderiff Advantage
AWS MediaConvert	Scale, Reliability	Complex, Expensive	10x simpler API
Cloudinary	Easy to use	Limited video focus	Full FFmpeg power
FFmpeg.wasm	Free, Client-side	Performance limits	Server-side scale
Custom Solutions	Full control	Maintenance burden	Managed service

2.3 Target Market Segments

1. **Startup Developers** (Primary)
 - Need: Quick integration, predictable costs

- Volume: 10,000+ potential customers
- ARPU: \$50-200/month

2. **SaaS Platforms** (Secondary)

- Need: White-label video processing
- Volume: 1,000+ potential customers
- ARPU: \$500-5,000/month

3. **Enterprise** (Tertiary)

- Need: Scale, SLA, compliance
- Volume: 100+ potential customers
- ARPU: \$5,000-50,000/month

2.4 Unique Value Proposition

"The simplicity of a modern API with the power of FFmpeg - process video with one line of code"

3. Project Scope

3.1 In Scope

Phase 1 - Developer API (Current Focus)

- Consolidated REST API with intelligent parameter parsing
- Support for 30+ video/audio codecs
- Real-time progress tracking and webhooks
- Quality metrics (VMAF, PSNR, SSIM)
- Horizontal scaling architecture
- Developer SDKs (Python, JavaScript, Go)
- API documentation and sandbox environment

Infrastructure & Operations

- Docker-based microservices architecture
- KrakenD API gateway implementation
- Valkey (Redis fork) for job queuing
- MinIO/S3 for object storage
- PostgreSQL for metadata
- Prometheus/Grafana monitoring

3.2 Out of Scope (Future Phases)

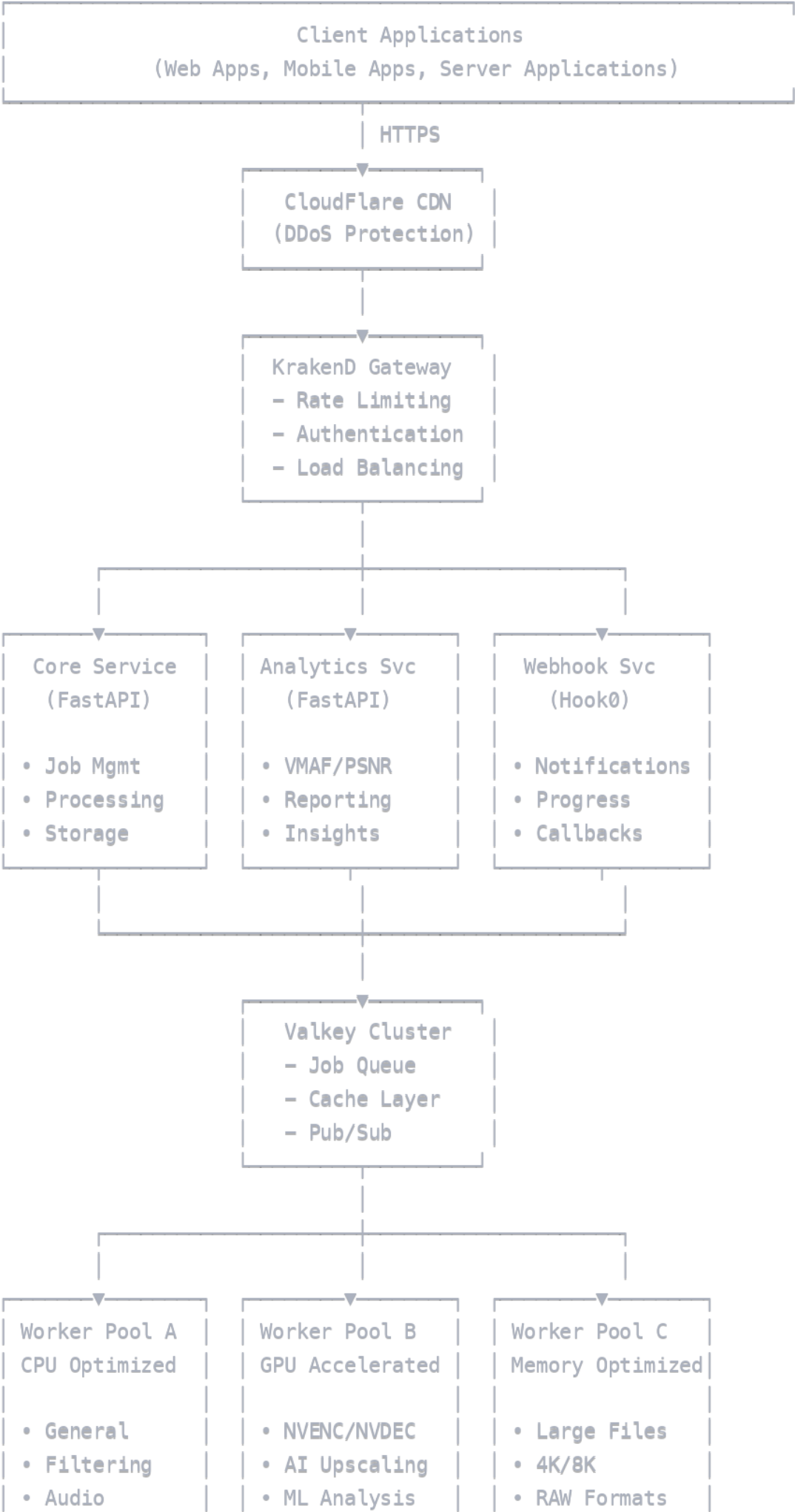
- Web UI for non-technical users (Part 2)
- Mobile applications
- Desktop applications
- Live streaming ingest
- Real-time video editing
- AI-based video enhancement
- Content delivery network (CDN)

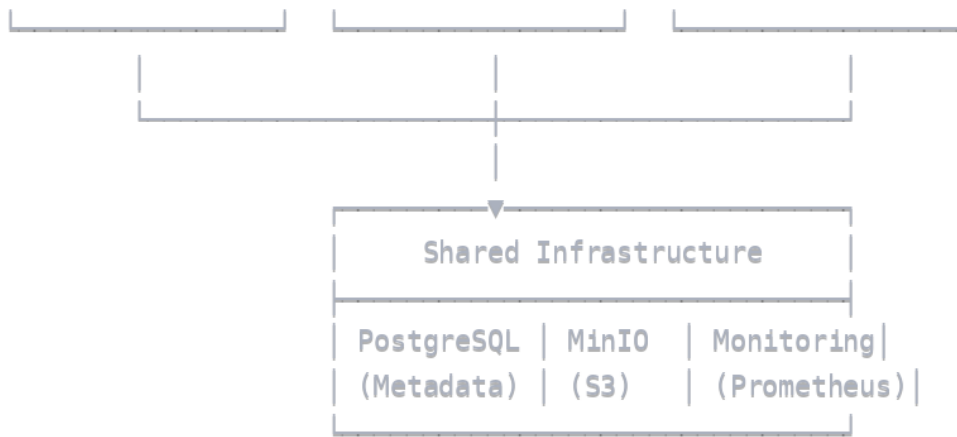
3.3 Technical Constraints

- Maximum file size: 100GB (initial), 1TB (future)
 - Concurrent jobs: 1,000 (initial), 10,000 (scaled)
 - Processing time: Maximum 10x real-time
 - Storage retention: 7 days (free), 30 days (paid)
 - Geographic availability: US/EU (initial), Global (future)
-

4. Technical Architecture

4.1 System Architecture





4.2 Service Components

4.2.1 Core Service

- **Technology Stack:** Python 3.12, FastAPI, SQLAlchemy
- **Responsibilities:**
 - API request handling and validation
 - Job orchestration and lifecycle management
 - Resource allocation and scheduling
 - User authentication and authorization
 - Usage tracking and billing

4.2.2 Worker Service

- **Technology Stack:** Python 3.12, FFmpeg 6.0, Celery
- **Variants:**
 - CPU Workers: Standard encoding tasks
 - GPU Workers: Hardware-accelerated encoding
 - Memory Workers: Large file processing
- **Capabilities:** Auto-scaling based on queue depth

4.2.3 Analytics Service

- **Technology Stack:** Python 3.12, FastAPI, VMAF SDK
- **Responsibilities:**
 - Quality metric calculation
 - Performance analytics
 - Usage pattern analysis
 - Cost optimization recommendations

4.3 Data Flow

- 1. **Request Phase:** Client → CDN → Gateway → Core Service
 - 2. **Processing Phase:** Core Service → Queue → Worker → Storage
 - 3. **Notification Phase:** Worker → Analytics → Webhooks → Client
 - 4. **Retrieval Phase:** Client → CDN → Storage → Client
-

5. API Specification

5.1 API Design Principles

- 1. **Consolidation:** Single endpoint for all conversions
- 2. **Intelligence:** Natural parameter interpretation
- 3. **Flexibility:** Simple to complex workflows
- 4. **Consistency:** Predictable request/response patterns
- 5. **Evolution:** Version-free through parameter extension

5.2 Core Endpoints

Endpoint	Method	Purpose
/api/v1/convert	POST	Universal conversion
/api/v1/analyze	POST	Media analysis
/api/v1/stream	POST	Streaming generation
/api/v1/estimate	POST	Cost/time estimation
/api/v1/jobs	GET	List jobs
/api/v1/jobs/{id}	GET	Job details
/api/v1/jobs/{id}	DELETE	Cancel job
/api/v1/jobs/{id}/progress	GET	Real-time progress
/api/v1/presets	GET	Available presets
/api/v1/capabilities	GET	Supported operations
/api/v1/usage	GET	Usage statistics
/api/v1/health	GET	Service health

5.3 Universal Conversion Examples

5.3.1 Simple Conversion

json

POST /api/v1/convert

```
{  
  "input": "https://storage.example.com/raw-video.mov",  
  "output": "mp4"  
}
```

5.3.2 Platform-Optimized

json

POST /api/v1/convert

```
{  
  "input": "https://storage.example.com/content.mp4",  
  "preset": "youtube-4k"  
}
```

5.3.3 Advanced Pipeline

json

POST /api/v1/convert

```
{
  "input": "https://storage.example.com/source.mov",
  "output": {
    "format": "mp4",
    "video": {
      "codec": "h265",
      "quality": "high",
      "resolution": "1080p",
      "fps": 30
    },
    "audio": {
      "codec": "aac",
      "bitrate": "192k",
      "normalize": true
    }
  },
  "operations": [
    {
      "type": "trim",
      "start": "00:01:00",
      "duration": 300
    },
    {
      "type": "watermark",
      "image": "https://cdn.example.com/logo.png",
      "position": "bottom-right",
      "opacity": 0.7
    }
  ],
  "optimize_for": "streaming",
  "enable_metrics": ["vmaf", "psnr"]
}
```

5.4 Response Standards

5.4.1 Success Response

json

```
{
  "status": "success",
  "data": {
    "job_id": "job_7f3e4d2a1b9c",
    "status": "processing",
    "created_at": "2025-01-27T10:00:00Z",
    "estimated_completion": "2025-01-27T10:15:00Z",
    "progress_url": "/api/v1/jobs/job_7f3e4d2a1b9c/progress",
    "webhook_url": "https://hooks.hook0.com/...",
    "cost_estimate": {
      "credits": 150,
      "usd": 1.50
    }
  }
}
```

5.4.2 Error Response

json

```
{
  "status": "error",
  "error": {
    "code": "INVALID_CODEC_COMBINATION",
    "message": "VP9 codec cannot be used with MP4 container",
    "details": {
      "parameter": "output.video.codec",
      "value": "vp9",
      "allowed_values": ["h264", "h265"],
      "suggestion": "Use 'webm' format or change codec to 'h264'"
    },
    "documentation": "https://docs.rendiff.dev/errors/INVALID_CODEC_COMBINATION",
    "request_id": "req_8g4f5e3b2c0d"
  }
}
```

6. Infrastructure Requirements

6.1 Computing Resources

Development Environment

- 3x API servers (4 vCPU, 8GB RAM)
- 5x Worker nodes (8 vCPU, 16GB RAM)

- 1x GPU node (NVIDIA T4)
- Load balancer
- Development databases

Production Environment (Initial)

- 10x API servers (8 vCPU, 16GB RAM)
- 50x CPU workers (16 vCPU, 32GB RAM)
- 10x GPU workers (NVIDIA A10G)
- 3x Database servers (PostgreSQL cluster)
- 3x Valkey nodes (cluster mode)
- MinIO cluster (100TB initial)

Production Environment (Scaled)

- Auto-scaling API servers (10-50 instances)
- Auto-scaling workers (50-500 instances)
- Multi-region deployment
- 1PB+ storage capacity
- Global CDN integration

6.2 Technology Stack

Category	Technology
Language	Python 3.12
API Framework	FastAPI 0.109
Gateway	KrakenD 2.6
Queue	Valkey 7.2
Database	PostgreSQL 16
Object Storage	MinIO / S3
Media Processing	FFmpeg 6.0
Webhooks	Hook0
Monitoring	Prometheus + Grafana
Container	Docker 25.0
Orchestration	Kubernetes 1.29
CI/CD	GitHub Actions

6.3 Network Architecture

- **Bandwidth:** 10 Gbps dedicated (initial), 100 Gbps (scaled)
 - **CDN:** CloudFlare Enterprise
 - **Regions:** US-East, US-West, EU-Central (initial)
 - **Latency Target:** <50ms API response globally
-

7. Security & Compliance

7.1 Security Framework

7.1.1 API Security

- **Authentication:** API key with HMAC signatures
- **Authorization:** Role-based access control (RBAC)
- **Encryption:** TLS 1.3 minimum
- **Rate Limiting:** Tiered by plan (10/100/1000 req/min)

7.1.2 Data Security

- **Encryption at Rest:** AES-256
- **Encryption in Transit:** TLS 1.3
- **Key Management:** AWS KMS / HashiCorp Vault
- **Access Control:** Principle of least privilege

7.1.3 Application Security

- **Input Validation:** Strict file type verification
- **Malware Scanning:** ClamAV integration
- **Content Moderation:** Optional NSFW detection
- **Sandboxing:** Isolated processing environments

7.2 Compliance Requirements

- **SOC 2 Type II:** Target Year 2
- **GDPR:** Privacy by design, data portability
- **CCPA:** California privacy compliance
- **HIPAA:** Healthcare tier (optional)
- **PCI DSS:** Payment processing compliance

7.3 Operational Security

- **Monitoring:** 24/7 security monitoring

- **Incident Response:** 15-minute SLA
 - **Backup:** 3-2-1 backup strategy
 - **Disaster Recovery:** RTO 4 hours, RPO 1 hour
-

8. Performance Requirements

8.1 Service Level Objectives (SLOs)

Metric	Target	Measurement
API Availability	99.9%	Monthly
API Response Time	<200ms	P95
Job Success Rate	>99%	Daily
Processing Speed	2x RT	Average
Quality Degradation	<2% VMAF	Per job

8.2 Scalability Targets

- **Concurrent Jobs:** 1,000 (Launch) → 10,000 (Year 1)
- **Daily Volume:** 100TB (Launch) → 1PB (Year 1)
- **API Requests:** 1M/day (Launch) → 100M/day (Year 1)
- **Geographic Coverage:** 3 regions (Launch) → Global (Year 1)

8.3 Performance Optimization

- **Caching Strategy:** Multi-layer (CDN, API, Database)
 - **Database Optimization:** Read replicas, partitioning
 - **Queue Optimization:** Priority queues, job batching
 - **Network Optimization:** Direct peering, edge locations
-

9. Implementation Roadmap

9.1 Phase 1: Foundation (Weeks 1-4)

Objective: Establish core infrastructure and development environment

- Set up development environment
- Deploy KrakenD gateway
- Configure Valkey cluster
- Implement basic API structure

- Create CI/CD pipeline
- **Deliverable:** Working development environment

9.2 Phase 2: Core API (Weeks 5-8)

Objective: Implement primary conversion functionality

- Build /convert endpoint with basic codecs
- Implement job management system
- Create worker pool architecture
- Add progress tracking
- Develop error handling
- **Deliverable:** MVP with basic conversion

9.3 Phase 3: Advanced Features (Weeks 9-12)

Objective: Add quality metrics and advanced operations

- Integrate VMAF/PSNR analysis
- Implement streaming formats (HLS/DASH)
- Add preset system
- Build estimation endpoint
- Create webhook integration
- **Deliverable:** Feature-complete API

9.4 Phase 4: Scale & Optimize (Weeks 13-16)

Objective: Prepare for production scale

- Implement auto-scaling
- Add GPU acceleration
- Optimize performance
- Create caching layer
- Build monitoring dashboard
- **Deliverable:** Production-ready system

9.5 Phase 5: SDK & Documentation (Weeks 17-20)

Objective: Enable developer adoption

- Develop Python SDK
- Create JavaScript SDK

- Build API documentation
- Create interactive sandbox
- Develop example applications
- **Deliverable:** Complete developer toolkit

9.6 Phase 6: Launch Preparation (Weeks 21-24)

Objective: Production deployment and launch

- Production deployment
 - Security audit
 - Load testing
 - Beta program
 - Launch marketing
 - **Deliverable:** Public launch
-

10. Risk Assessment

10.1 Technical Risks

Risk	Probability	Impact	Mitigation
FFmpeg compatibility issues	Medium	High	Extensive testing matrix
Scaling bottlenecks	Medium	High	Progressive load testing
Codec licensing complexity	High	Medium	Legal review, alternatives
Storage costs explosion	Medium	High	Intelligent retention
GPU availability	Low	Medium	Multi-cloud strategy

10.2 Business Risks

Risk	Probability	Impact	Mitigation
Slow adoption	Medium	High	Free tier, great docs
Competition	High	Medium	Unique features
Pricing pressure	Medium	Medium	Value-based pricing
Support burden	Medium	Low	Self-service tools

10.3 Operational Risks

Risk	Probability	Impact	Mitigation
DDoS attacks	Medium	High	CloudFlare protection
Data breach	Low	Critical	Security framework
Service outage	Low	High	Multi-region deployment
Vendor lock-in	Medium	Medium	Open standards

11. Cost Analysis

11.1 Development Costs

Category	Cost	Duration
Senior Backend Dev	\$15,000	6 months
DevOps Engineer	\$12,000	4 months
Frontend Dev (Part 2)	\$10,000	3 months
QA Engineer	\$8,000	3 months
Technical Writer	\$5,000	1 month
Total Personnel	\$185,000	

11.2 Infrastructure Costs (Monthly)

Component	Launch	Scale
Compute (API)	\$500	\$2,000
Compute (Workers)	\$2,000	\$10,000
GPU Instances	\$1,000	\$5,000
Storage (S3)	\$500	\$3,000
Bandwidth	\$1,000	\$5,000
Database	\$300	\$1,000
Monitoring	\$200	\$500
Third-party APIs	\$500	\$2,000
Total Monthly	\$6,000	\$28,500

11.3 Revenue Projections

Plan	Price	Month 6	Month 12
Free Tier	\$0	5,000	20,000
Starter	\$49	200	1,000
Professional	\$199	50	300
Enterprise	\$999+	5	30
MRR		\$24,850	\$138,700
Annual Run		\$298,200	\$1,664,400

11.4 Break-even Analysis

- **Fixed Costs:** \$185,000 (development)
 - **Monthly Costs:** \$8,500 (average)
 - **Break-even:** Month 8 at projected growth
 - **Profitability:** Month 12 onwards
-

12. Success Criteria

12.1 Technical Success Metrics

- ☐ API response time < 200ms (P95)
- ☐ 99.9% uptime achieved
- ☐ Support for 50+ formats/codecs
- ☐ Process 100TB in first month
- ☐ Zero critical security incidents

12.2 Business Success Metrics

- ☐ 1,000 registered developers (Month 6)
- ☐ 100 paying customers (Month 6)
- ☐ \$25,000 MRR (Month 6)
- ☐ 4.5+ developer satisfaction score
- ☐ <2 hour average support response

12.3 Operational Success Metrics

- ☐ <5% job failure rate
- ☐ <10% monthly churn rate
- ☐ 90%+ automated support resolution
- ☐ 50%+ organic traffic growth
- ☐ 3x CAC recovery within 12 months

13. Appendices

Appendix A: Technical Specifications

- Detailed FFmpeg codec support matrix
- Performance benchmarks by operation type
- API request/response schemas
- Error code reference

Appendix B: Market Research

- Competitor feature comparison
- Pricing analysis
- Developer survey results
- Use case studies

Appendix C: Legal Considerations

- Codec licensing requirements
- Terms of service template
- Privacy policy requirements
- Data processing agreements

Appendix D: Operational Runbooks

- Deployment procedures
- Scaling playbooks
- Incident response plans
- Disaster recovery procedures

Document Control

Version	Date	Author	Changes
1.0	2025-01-27	Backend Team	Initial draft
2.0	2025-01-27	Backend Team	Consolidated API design

Next Review Date: February 10, 2025

Approval Required From: CTO, VP Engineering, VP Product

This document contains confidential and proprietary information of Rendiff. Distribution is limited to authorized personnel only.