

JusticeOS -- Reusable Platform Architecture

Generated March 30, 2026 | law.unykorn.org

JusticeOS -- Reusable Platform Architecture

A Turnkey Criminal Justice Defense System

1. VISION

JusticeOS is a multi-tenant legal defense technology platform that gives wrongfully convicted individuals, their families, and advocacy organizations the tools to fight back -- without needing a tech team or a law degree.

Built from the Marquis Delcampo case. Designed for everyone who comes after.

2. PLATFORM COMPONENTS

JusticeOS Platform

Client Case Engine Agent Portal Orchestrator - CRUD - Dashboard - Evidence - 350 agents - Documents - Timeline - 4 pools - Timeline - Custody - Priority q - Comms - Hash/IPFS - QA overlay

FastAPI Backend /cases /evidence /agents /docs /analytics

PostgreSQL Redis IPFS (cases, (cache, (evidence users, sessions, preservation) evidence) agent queues)

JusticeChain (L1 Blockchain) Case hashes | Evidence CIDs | Disclosures | Events

?
?
??
? Infrastructure: Docker Compose | Cloudflare | AWS ?
??

3. MULTI-TENANT DATA MODEL

Tenancy Strategy: Schema-per-case with shared infrastructure

```
-- Core tables (shared)
CREATE TABLE organizations (
  id UUID PRIMARY KEY,
  name TEXT NOT NULL,
  type TEXT NOT NULL, -- 'individual', 'family', 'advocacy_org', 'law_firm'
  created_at TIMESTAMPTZ DEFAULT NOW()
);

CREATE TABLE users (
  id UUID PRIMARY KEY,
  org_id UUID REFERENCES organizations(id),
  email TEXT UNIQUE NOT NULL,
  role TEXT NOT NULL, -- 'admin', 'editor', 'viewer', 'attorney'
  disclosure_acknowledged BOOLEAN DEFAULT FALSE,
  disclosure_date TIMESTAMPTZ,
  created_at TIMESTAMPTZ DEFAULT NOW()
);

-- Case tables (per-tenant isolation via org_id)
CREATE TABLE cases (
  id UUID PRIMARY KEY,
  org_id UUID REFERENCES organizations(id),
  case_name TEXT NOT NULL,
  defendant_name TEXT NOT NULL,
  jurisdiction TEXT NOT NULL,
  state TEXT NOT NULL,
  county TEXT,
  circuit TEXT,
  charge_statute TEXT,
  charge_description TEXT,
  sentence TEXT,
  case_number TEXT,
  agency_report_number TEXT,
  incident_date DATE,
  arrest_date DATE,
  sentencing_date DATE,
  appeal_deadline DATE,
  status TEXT DEFAULT 'active',
  template TEXT, -- 'wrongful_conviction', 'excessive_sentence', 'syg', 'brady', 'custom'
  created_at TIMESTAMPTZ DEFAULT NOW()
);
```

```
CREATE TABLE evidence (  
id UUID PRIMARY KEY,  
case_id UUID REFERENCES cases(id),  
org_id UUID REFERENCES organizations(id),  
title TEXT NOT NULL,  
description TEXT,  
source TEXT,  
evidence_type TEXT, -- 'video', 'document', 'photo', 'audio', 'digital', 'physical'  
sha256_hash TEXT NOT NULL,  
ipfs_cid TEXT,  
chain_tx_hash TEXT, -- JusticeChain transaction hash  
custody_status TEXT DEFAULT 'preserved',  
metadata JSONB,  
created_at TIMESTAMPTZ DEFAULT NOW()  
);
```

```
CREATE TABLE timeline_events (  
id UUID PRIMARY KEY,  
case_id UUID REFERENCES cases(id),  
event_date TIMESTAMPTZ NOT NULL,  
event_type TEXT,  
description TEXT NOT NULL,  
source TEXT,  
evidence_ids UUID[],  
chain_tx_hash TEXT,  
created_at TIMESTAMPTZ DEFAULT NOW()  
);
```

```
CREATE TABLE documents (  
id UUID PRIMARY KEY,  
case_id UUID REFERENCES cases(id),  
org_id UUID REFERENCES organizations(id),  
doc_type TEXT NOT NULL, -- 'motion', 'brief', 'memo', 'letter', 'report', 'tracker'  
title TEXT NOT NULL,  
content TEXT,  
template_source TEXT,  
status TEXT DEFAULT 'draft', -- 'draft', 'review', 'approved', 'filed'  
sha256_hash TEXT,  
created_at TIMESTAMPTZ DEFAULT NOW()  
);
```

```
CREATE TABLE attorney_contacts (  
id UUID PRIMARY KEY,  
case_id UUID REFERENCES cases(id),  
firm_name TEXT NOT NULL,  
attorney_name TEXT,  
tier INT,  
category TEXT,  
phone TEXT,  
email TEXT,  
website TEXT,
```

```
status TEXT DEFAULT 'not_contacted',
contacted_date DATE,
response_type TEXT,
notes TEXT,
created_at TIMESTAMPTZ DEFAULT NOW()
);
```

```
CREATE TABLE disclosures (
id UUID PRIMARY KEY,
user_id UUID REFERENCES users(id),
case_id UUID REFERENCES cases(id),
disclosure_type TEXT NOT NULL,
acknowledged BOOLEAN DEFAULT FALSE,
acknowledged_at TIMESTAMPTZ,
ip_address TEXT,
chain_tx_hash TEXT, -- On-chain record of acknowledgment
created_at TIMESTAMPTZ DEFAULT NOW()
);
```

Row-Level Security

Every query filtered by `org_id`. Users can only see data for their organization. Attorneys granted cross-case access only with explicit permission.

4. CASE TEMPLATES

When a new case is created, the system auto-generates a starter kit based on the case template:

Template: Wrongful Conviction

Auto-generated:

- ??? Case dossier skeleton
- ??? Evidence index (blank, ready to fill)
- ??? Timeline template
- ??? Motion templates:
 - ? ??? Rule 3.850 (post-conviction)
 - ? ??? Rule 3.800(a) (illegal sentence)
 - ? ??? Brady violation motion
- ??? Attorney referral list (filtered by jurisdiction)
- ??? Deadline calculator (from sentencing date)
- ??? Non-attorney disclosure (REQUIRED)
- ??? Law library (jurisdiction-specific)

Template: Excessive Sentencing

Auto-generated:

- ??? Sentencing analysis worksheet
- ??? Enhancement challenge checklist
- ??? Rule 3.800(a) motion template
- ??? Scoresheet audit template
- ??? Jurisdiction sentencing guidelines
- ??? Attorney referral list

??? Appeal deadline calculator

Template: Stand Your Ground / Self-Defense

Auto-generated:

- ??? SYG immunity motion template
- ??? Self-defense analysis worksheet
- ??? Occupied vehicle doctrine checklist
- ??? Evidence inventory (focused on video, witnesses)
- ??? SYG hearing preparation checklist
- ??? Jurisdiction-specific SYG statutes
- ??? Attorney referral list

Template: Police Misconduct / Brady Violation

Auto-generated:

- ??? Brady checklist
- ??? Public records request templates (police agencies)
- ??? IA complaint template
- ??? Evidence preservation demands
- ??? Officer background research checklist
- ??? Civil rights firm referral list
- ??? §1983 claim assessment

5. JURISDICTION MODULES

The law library and motion templates are jurisdiction-specific. The Marquis case builds out the Florida module. Future modules follow the same structure:

Florida Module (Built)

law-library/florida/

- ??? statutes/ # FL Statutes relevant to criminal defense
- ??? case-law/ # FL case law by topic
- ??? rules/ # FL Rules of Criminal/Appellate Procedure
- ??? motions/ # FL-specific motion templates
- ??? quick-reference/ # FL courtroom reference sheets
- ??? research-apis/ # FL court database search scripts
- ??? agencies/ # FL agency contact info, FOIA procedures

Future Modules (Same Structure)

- * Georgia, Texas, California, New York, Illinois
- * Federal (AEDPA, §2254, §1983)

Each module contains:

- * State-specific statutes
- * State-specific case law
- * State-specific procedural rules
- * State-specific motion templates
- * State-specific court database APIs
- * State-specific agency contacts

6. CLIENT PORTAL -- UI/UX

Dashboard

JusticeOS -- Case: State v. [Defendant] ?

Home ? Docs ? Time ? Evid ? Legal ? Reach ? Settings ?

CASE STATUS ? CRITICAL DEADLINES ? ?

* Active ? ? Appeal: May 10, 2026 ? ?

* Stage: Appeal ? ? Rule 3.850: TBD ? ?

* Attorneys: 2/19 ? ? Records: 3 pending ? ?

RECENT ACTIVITY ? ? AGENT STATUS ? ?

- Motion draft ready ? ? Legal: 150 active ? ?

- New evidence added ? ? Outreach: 65 active ? ?

- Attorney responded ? ? Media: 50 standby ? ?

Platform: 85 active ? ?

Key Features

- Documents Tab: All case documents with status (draft/review/approved/filed)
- Timeline Tab: Interactive case timeline with evidence links
- Evidence Tab: Evidence inventory with hash verification and custody chain
- Legal Tab: Motion templates, law library access, filing tracker
- Outreach Tab: Attorney contact status, follow-up reminders
- Settings: User management, disclosure status, blockchain records

7. API ENDPOINTS (Expanded)

Case Management

POST /api/v1/cases # Create new case

GET /api/v1/cases/{id} # Get case details

PUT /api/v1/cases/{id} # Update case

GET /api/v1/cases/{id}/dashboard # Dashboard data

Evidence

POST /api/v1/cases/{id}/evidence # Add evidence (auto-hash + IPFS)

GET /api/v1/cases/{id}/evidence # List evidence

POST /api/v1/cases/{id}/evidence/{eid}/verify # Verify hash integrity

Timeline

POST /api/v1/cases/{id}/timeline # Add timeline event
GET /api/v1/cases/{id}/timeline # Get full timeline

Documents

POST /api/v1/cases/{id}/documents # Generate document from template
GET /api/v1/cases/{id}/documents # List documents
PUT /api/v1/cases/{id}/documents/{did} # Update document status

Attorney Outreach

GET /api/v1/cases/{id}/outreach # Outreach status
POST /api/v1/cases/{id}/outreach/contact # Mark firm contacted
POST /api/v1/cases/{id}/outreach/packet # Generate case packet

Agents

GET /api/v1/agents # List all agents
GET /api/v1/agents/pool/{pool} # List agents in pool
POST /api/v1/agents/{id}/task # Assign task to agent
GET /api/v1/agents/{id}/status # Get agent status

Disclosures

POST /api/v1/disclosures/acknowledge # Record disclosure acknowledgment
GET /api/v1/disclosures/{user_id} # Get user's disclosure history

Blockchain

POST /api/v1/chain/record # Record event on JusticeChain
GET /api/v1/chain/verify/{hash} # Verify on-chain record

Analytics

GET /api/v1/analytics/case/{id} # Case progress metrics
GET /api/v1/analytics/agents # Agent performance metrics
GET /api/v1/analytics/outreach # Outreach campaign metrics

8. DEPLOYMENT

Development

docker-compose.yml (extended)

```
services:  
  api:  
    build: ./services/api  
    ports: ["8080:8080"]
```

depends_on: [postgres, redis, ipfs]

orchestrator:

build: ./services/agents

depends_on: [api, redis]

postgres:

image: postgres:16

ports: ["5440:5432"]

volumes: ["pgdata:/var/lib/postgresql/data"]

redis:

image: redis:7

ports: ["6390:6379"]

ipfs:

image: ipfs/kubo:latest

ports: ["5001:5001", "8081:8080"]

justicechain:

build: ./services/chain

ports: ["7333:7333"]

web:

build: ./services/web

ports: ["3000:3000"]

depends_on: [api]

Production (AWS)

- * ECS Fargate containers
- * RDS PostgreSQL (Multi-AZ)
- * ElastiCache Redis
- * S3 + CloudFront for static assets
- * Pinata/Infura for IPFS pinning
- * Cloudflare for DNS + CDN + SSL
- * JusticeChain nodes on dedicated EC2 instances

9. OPEN-SOURCE STRATEGY

Phase 1 (Month 3): Core tools open-sourced

- * Motion template library (all jurisdictions)
- * Deadline calculators
- * Research API scripts
- * Evidence hash verification tool

Phase 2 (Month 6): Platform core open-sourced

- * Case management engine
- * Agent orchestrator framework
- * Client portal (no agent integration)

Phase 3 (Month 12): Full platform

- * Complete JusticeOS platform
 - * JusticeChain contracts
 - * All jurisdiction modules
 - * Deployment guides
 - * License: AGPL-3.0 (copyleft -- ensures derivatives remain open)
-

Classification: INTERNAL -- Architecture Document

Version: 1.0