Integrinow DMS v2: Comprehensive Briefing Document

Executive Summary

Integrinow DMS v2 is a high-security, multi-tenant document management system designed to establish a defensible chain of custody for sensitive customer documents. The system's core architecture ensures that no document reaches durable storage without undergoing a mandatory security pipeline consisting of malware scanning, integrity hashing, encryption, and human review.The platform utilizes a modern asynchronous stack—Flask, Celery, PostgreSQL, MinIO, and HashiCorp Vault—to balance responsive user interaction with heavy background processing. Key security features include envelope encryption, WORM (Write Once, Read Many) storage, a hash-chained audit log for tamper evidence, and a unique source-code integrity verification system. While the current version serves as a robust functional baseline, several components, such as the automated processing stage, remain placeholders pending specific customer requirements.

1. System Roles and Access Control

The system operates on a strict multi-tenant model where access is governed by roles and organization-based scoping.| Role | Scope | Key Capabilities || ------ | ------ | ------ || **Customer** | Org-specific | Upload and download approved documents for their own organization. || **Processor** | Cross-org | Upload documents on behalf of any organization; limited visibility. || **Reviewer** | Cross-org | Access the review queue to approve or reject pending documents. || **Admin** | Full System | Full operational access, including audit logs and backup/restore. |

Access Security

- **Authentication:** Managed via JWT (JSON Web Tokens) with a 60-minute access window and 30-day refresh.
- **Rate Limiting:** Login attempts are restricted to 5 per minute per IP/username to mitigate credential-stuffing attacks.
- **Enforcement:** Organization scoping is enforced at the database query level; customers attempting to access documents outside their org_id receive a 403 Forbidden error.

2. Document Lifecycle and Pipeline

The system enforces a rigid, forward-only state machine. Documents cannot skip stages or return to previous states.

The Fixed State Machine

1. **RECEIVED:** Upload URL issued; file landing in quarantine.
2. **AV_SCANNING:** Malware scanning in progress.
3. **VALIDATED:** Malware-free; integrity hash (SHA-256) recorded.
4. **PROCESSING:** Automated transformations (currently a placeholder).
5. **NEEDS_REVIEW:** Encrypted and stored in the vault, awaiting human approval.
6. **APPROVED / REJECTED:** Terminal states based on reviewer decision.

7. **ARCHIVED:** Terminal state for older approved documents.

Automated Processing Pipeline

Triggered asynchronously via Celery after a client confirms an upload, the pipeline follows these steps:

- **Malware Detection:** Files are streamed to ClamAV via the INSTREAM protocol. Infected files are immediately rejected and deleted from quarantine.
- **Integrity Record:** A SHA-256 hash is taken of the raw bytes *before* any processing to record exactly what was uploaded.
- **Placeholder Transformation:** Currently, plain text/Markdown files receive a timestamped "placeholder stamp." Binary files (PDF, Office) are left untouched to prevent corruption until real processing logic (e.g., OCR) is defined.
- **Vault Storage:** Processed files are encrypted and moved from quarantine to WORM-protected vault storage.

3. Technical Architecture and Security

The technical stack is designed for isolation and high-assurance data protection.

Core Stack

- **Application:** Flask 2.3.3 behind an Nginx reverse proxy (TLS termination).
- **Storage:** MinIO (S3-compatible). The "Vault" bucket uses **Object Lock (GOVERNANCE mode)** to prevent deletion or modification.
- **Encryption:** Managed via HashiCorp Vault.
- **Envelope Encryption:** Each document has a unique Data Encryption Key (DEK). The DEK is "wrapped" (encrypted) by Vault's transit engine before being stored in the database.
- **Plaintext Security:** The raw DEK never touches the database, and the Master Key (KEK) never leaves Vault.

Source Integrity Verification

Integrinow DMS v2 includes a "startup-check" mechanism to prevent the execution of tampered code:

- A manifest of SHA-256 fingerprints for all source files is signed using an Ed25519 key in Vault.
- The running application only holds "verify" permissions, not "sign" permissions.
- At startup, the app re-hashes its own files and compares them against the signed manifest. Any mismatch in critical files (.py, .sh) results in an immediate system exit.

Audit Logging

The system maintains a tamper-evident, append-only audit log.

- **Hash Chaining:** Each entry contains a hash of itself plus the hash of the preceding entry.

- **Concurrency Control:** A Postgres advisory lock ensures that concurrent background workers do not fork the chain during simultaneous logging events.
- **Verification:** Admins can trigger a chain walk to verify that no historical entry has been altered or deleted.

4. Operational Management

Backup and Restore

The system provides enterprise-grade data protection through an encrypted packaging system:

- **Scope:** Captures the PostgreSQL database and both MinIO buckets (Quarantine and Vault).
- **Security:** Backups are envelope-encrypted using the same Vault-backed transit engine as the documents.
- **Off-site Replication:** Supports multi-cloud redundancy. Backups can be automatically replicated to AWS S3, Google Cloud Storage, or Azure Blob Storage.
- **Destructive Restore:** Restoration is intentionally manual and destructive, requiring explicit confirmation ({"confirm": "RESTORE"}) to overwrite current data.

Deployment Options

The system supports three primary deployment topologies:

1. **Development:** Host-run application with Docker-managed infrastructure.
2. **Containerized:** A fully containerized stack using docker-compose.
3. **Orchestrated:** Kubernetes deployment via Helm charts.