DEER DESIGN TICKETING SYSTEM – FILE STORAGE & FEEDBACK ARCHITECTURE (SCALABLE & SECURE DESIGN)

Core Components

- **Frontend (Web/UI):** Upload/download interface, design viewer, annotation tools.
- **Backend API Layer:** RESTful, stateless services for file handling and feedback.
- **Database:** Relational DB (e.g., PostgreSQL) for metadata and feedback.
- **File Storage:** Object storage (e.g., Amazon S3) for designs and annotation overlays.
- **Annotation Service:** Stores version-specific visual/text feedback on designs.

File Storage System

- **Structure:**
  - Files are tied to a ticketId and assigned a unique fileId.
  - Each upload creates a new version:
    `/tickets/{ticketId}/files/{fileId}/versions/{version}/filename.ext`
- **Security:**
  - Role-based access control (Designer, Client, Admin)
  - Signed URLs and virus scanning
- **Performance:**
  - CDN-backed file access
  - Multipart uploads for large files
- **Scalability:**
  - Stateless backend behind load balancer
  - Object storage (e.g., S3) scales independently
- **Availability:**
  - Cloud storage durability (e.g., 11 9s with S3)
  - Multi-zone deployment of API and DB replicas

Feedback & Annotation System

- **Version-Specific Feedback:**
  - Feedback references ticketId, fileId, and version to ensure precision
  - Supports annotations (text, highlights, drawings) with coordinate mapping
- **API Endpoints:**
  - `POST /tickets/:id/files/:fileId/versions/:version/feedback`
  - `GET /tickets/:id/files/:fileId/versions/:version/feedback`
- **Storage Format:**

```
{
  "ticketId": "abc123",
  "fileId": "file789",
  "version": 3,
  "authorId": "user456",
  "timestamp": "2025-04-29T12:10:00Z",
  "comment": "Move logo to the left",
  "coordinates": { "x": 120, "y": 250 },
  "type": "text"
}
```

- **Scalability:**
  - Feedback service as a microservice
  - Store annotations as structured data or overlays (JSON or vector format)
- **Performance:**
  - Lazy loading of annotations
  - Caching commonly viewed designs
- **Availability:**
  - Redundant DB with failover
  - Zone-resilient microservice deployment

Security, Performance & Availability Summary

Concern	Solution
Security	OAuth/JWT auth, RBAC, virus scanning, signed URLs
Performance	CDN, DB indexing, load balancing, async background tasks
Scalability	Stateless services, object storage, decoupled annotation processing
Availability	DB replication, multi-zone deployment, cloud-native file durability

Thiago Villa, 4/29/25