Dispute Resolution Engine

Non-functional Requirements

Quality Requirements

Reliability

The Dispute Resolution Engine must maintain high uptime due to the nature of the services provided. Clients require continuous access to the system to respond promptly to time-sensitive communications. A reliable system with high uptime not only benefits clients but also increases overall throughput.

- Stimulus Source: System Users/Clients
- Stimulus: Attempt to access the Dispute Resolution Engine
- Response: The system should maintain high uptime to ensure continuous access.
- Response Measure: 99.9% system uptime, fewer than 1 hour of downtime per year
- Environment: Production environment
- Artifact: Dispute Resolution Engine

Scalability and Reliability

Scalability and performance are critical for the Dispute Resolution Engine (DRE). Disputes are common, and the system must handle many cases simultaneously, along with multiple active users per case. High performance is essential for scalability, as a well-performing system can accommodate more users effectively.

- Stimulus Source: System Users/Clients
- Stimulus: High volume of disputes and multiple active users per case
- Response: The system should handle multiple cases simultaneously and perform efficiently for each user.
- Response Measure: System supports 1000 concurrent users and 500 disputes without performance degradation
- Environment: Production environment
- Artifact: Dispute Resolution Engine API, Backend & Deployment

Usability

The Dispute Resolution Engine is designed to be user-friendly, with simple and intuitive interfaces to facilitate easy navigation. The goal is to enable users to operate the system effectively without extensive training or technical knowledge, catering to users from a wide range of domains.

- Stimulus Source: System Users/Clients
- Stimulus: Attempt to navigate and use the system
- Response: The system should provide a simple and intuitive user interface requiring minimal training.
- Response Measure: Lighthouse reports above 90/100, average request responses under 2 seconds

- Environment: Production environment
- Artifact: Dispute Resolution Engine Frontend

Security

The Dispute Resolution Engine handles sensitive information, including user credentials and confidential dispute communications. Ensuring the protection of this data is crucial to maintain user trust and uphold the integrity of the system.

- Stimulus Source: Malicious Actors/Unauthorized Users
- Stimulus: Attempt to access sensitive information
- Response: The system should ensure the protection of personal and dispute-related information.
- Response Measure: Unauthorized access attempts identified and flagged, all OWASP checks passed
- Environment: Production environment
- Artifact: Dispute Resolution Engine Frontend & API

Maintainability

The Dispute Resolution Engine must be easy to update and extend, allowing the addition of new legal processes, dispute types, and other features. This adaptability ensures the system remains relevant and aligned with the latest legal standards and practices.
- Stimulus Source: Development Team
- Stimulus: Requirement to update or extend the system
- Response: The system should allow for easy addition of new legal processes, dispute types, and other features.
- Response Measure:
    - Frontend uses a component library.
    - Backend API is fully extensible with trivial addition/removal of endpoints.
- Example: When we integrated the OpenAI API into the system to provide summaries for archived disputes, we had to incorporate an external service into our existing structure. Despite this being a new type of integration, we followed the same structure used throughout our API to manage requests, provide responses, and handle errors. This approach ensured that the API remained consistent and maintainable, demonstrating the system's ability to integrate new functionalities seamlessly.
- Environment: Development environment
- Artifact: Dispute Resolution Engine