1. **Users/Clients**: Interact with the system to submit documents for processing.
2. **API Gateway**:
   * Routes incoming requests to the appropriate microservices.
   * Queries the **Service Registry** to discover available instances of microservices and route requests accordingly.
3. **Load Balancer**:
   * Distributes traffic across multiple instances of the API Gateway for high availability and fault tolerance.
4. **Message Queue**:
   * Decouples document ingestion from processing.
   * Ensures reliable message delivery.
   * Tools: RabbitMQ, Apache Kafka, AWS SQS.
5. **Conversion Microservices**:
   * Handle specific document types (e.g., PDF, DOCX).
   * Scalable and can be replicated for high availability.
   * **Circuit Breakers**: Protect microservices from cascading failures by detecting failures and halting requests when a service is in a failure state.
   * **Bulkhead**: isolate different types of processing tasks (e.g., document conversion vs. metadata handling).
6. **Result Storage & Retrieval**:
   * Stores processed documents and metadata.
   * Uses SQL databases for structured data and object storage (e.g., AWS S3) for large files.
7. **Caching Layer**:
   * Caches frequently accessed data to improve performance.
   * Tools: Redis, Memcached.
8. **Monitoring & Logging**:
   * Aggregates logs and monitors system performance.
   * Tools: ELK Stack (Elasticsearch, Logstash, Kibana), Prometheus with Grafana.