# Functional Requirement

## 3.1 Real-Time Synchronization  
The system shall synchronize contact and email data with third-party tools (e.g., CRM systems) within 5 seconds of a change being detected or initiated.  
  
## 3.2 Intelligent Email Archiving  
The system shall automatically archive emails using a combination of:  
- Predefined rules for retention and categorization.  
- Customizable user-defined tags.  
- AI-driven content analysis for intelligent filing.  
The system shall allow users to create and modify archiving rules and tags via the administrative interface.  
  
## 3.3 Secure and Compliant Storage  
The system shall store archived emails in a manner that complies with the following standards:  
- GDPR (General Data Protection Regulation)  
- HIPAA (Health Insurance Portability and Accountability Act)  
- SOX (Sarbanes-Oxley Act)  
Compliance shall be verified by automated logs and audit trails.  
  
## 3.4 Optional Delayed Synchronization  
The system shall support optional delayed synchronization in high user activity scenarios, where more than 100 simultaneous actions are detected within a 1-minute interval. The system shall queue and process data in batches to reduce latency and prevent system overload.

# External Description

# 5 Constraints  
  
## 5.1 Legal Constraints  
- The system shall comply with GDPR (C-Legal-001).  
- The system shall comply with HIPAA (C-Legal-002).  
  
## 5.2 Hardware Constraints  
- The system shall support deployment on servers with at least 8 GB of RAM and 500 GB of storage (C-Hardware-001).  
  
## 5.3 Interface Constraints  
- The system shall support RESTful API integration (C-Interface-001).  
- The system shall provide a web-based user interface with responsive design (C-Interface-002).  
- The system shall support integration with Microsoft Outlook and Google Workspace (C-Interface-003).  
  
## 5.4 Design Constraints  
- The system shall be modular to allow for future feature expansion (C-Design-001).  
- The system shall be developed using a microservices architecture (C-Design-002).  
- The system shall support multi-tenancy for enterprise deployments (C-Design-003).  
  
## 5.5 Other Constraints  
- The system shall be developed using Python and Java, with containerization via Docker (C-Other-001).