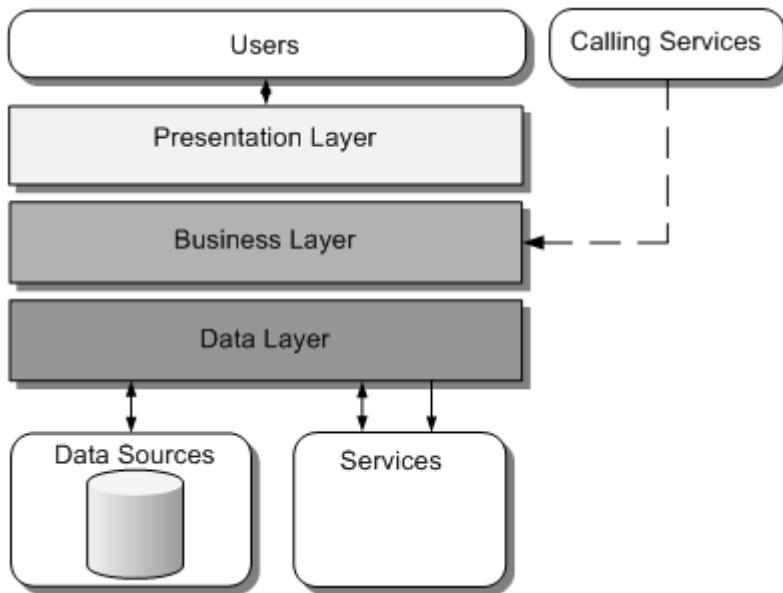


Layered architecture

- The **Intelligent Email Processing & Response Automation System** processes emails, analyzes content (possibly using NLP), generates responses, and stores data.

This type of system clearly separates:

- User interaction
- Business logic (processing & automation)
- Data storage



Presentation Layer

Granularity: High-level component

Handles:

- **Admin dashboard**
- **User login**
- **Viewing processed emails**
- **Viewing generated responses**

Business Logic Layer (Application Layer)

Granularity: Medium-grained service modules

Contains core processing modules:

- **Email Fetching Module**
- **NLP Processing Module**

- Intent Classification Module
- Response Generation Module
- Rule Engine
- Workflow Controller

Data Layer (Database Layer)

Granularity: Fine-grained data access components

Handles:

- Email storage
- User data
- Logs
- Templates
- Configuration settings

Justification Based on Granularity

Coarse-Grained Components (Top Level)

- Presentation Layer
- Business Logic Layer
- Data Layer

Medium-Grained Components (Inside Business Layer)

- EmailProcessor
- NLPAnalyzer
- AutoReplyGenerator
- TemplateManager
- AdminController

Fine-Grained Components (Inside Modules)

- Methods like:
- fetchEmail()
- classifyIntent()
- generateReply()
- saveToDatabase()

These are detailed operations.

1. Scalability

- **Supports modular growth:**
 - **New modules (Spam Detection, Sentiment Analysis, Priority Classification) can be added in the Business Layer.**
- **Enables independent scaling:**
 - **Upgrade server resources without changing UI.**
 - **Optimize database separately.**
- **Allows future transition to Microservices if system expands.**
- **Suitable for moderate-to-large email traffic systems.**

2. Maintainability

- **Clear separation of concerns:**
 - **Presentation Layer → User Interface**
 - **Business Layer → Email processing logic**
 - **Data Layer → Database operations**
- **Changes in one layer do not affect other layers.**
- **Easier debugging and code updates.**
- **Promotes clean code structure.**
- **Reduces long-term maintenance cost.**

3. Performance

- **No network communication overhead (unlike Microservices).**
- **Faster processing due to direct method calls between layers.**
- **Centralized database access improves query efficiency.**
- **Suitable for real-time or near real-time email processing**

4. Security

- **Database is not directly accessible from UI.**
- **All data access controlled via Business Layer.**
- **Easier implementation of:**
 - **Authentication**
 - **Authorization**
 - **Role-based access control**
- **Improved protection against data misuse**

5. Testability

- **Each layer can be tested independently.**
- **Business logic can be unit-tested without UI.**
- **Database operations can be tested using mock data.**
- **Improves reliability and software quality.**

6. Simplicity & Development Efficiency

- **Easier to implement compared to Microservices.**
- **Lower deployment complexity.**
- **Suitable for academic and mid-scale projects.**
- **Faster development lifecycle.**

FINAL JUSTIFICATION

- **Provides strong modularity and separation of concerns.**
- **Offers good scalability without high architectural complexity.**
- **Ensures high maintainability and easier debugging.**
- **Delivers better performance compared to distributed architectures.**
- **Supports secure and structured data handling.**

APPLICATION COMPONENTS OF THE SYSTEM

1. Presentation Layer Components

- **User Interface Module**
 - Dashboard for viewing processed emails
 - View generated responses
 - Manual override option
- **Admin Panel**
 - Manage response templates
 - View system logs
 - Configure automation rules
- **Authentication Module**
 - Login
 - Role-based access control

2. Business Logic Layer Components

Email Fetching Component

- Connects to email server (IMAP/SMTP)
- Retrieves incoming emails

Email Preprocessing Component

- Cleans email content
- Removes noise (HTML tags, signatures)

NLP Processing Component

- Tokenization
- Intent classification
- Keyword extraction

Spam Detection Component

- Identifies unwanted emails

Priority Classification Component

- Marks emails as high/medium/low priority

Response Generation Component

- Selects template
- Generates automated reply

Rule Engine Component

- Applies predefined automation rules

Workflow Controller

- Coordinates entire processing pipeline

3. Data Layer Components

Email Database

- Stores received emails

User Database

- Stores user credentials and roles

Template Repository

- Stores predefined response templates

Logs & Audit Database

- Stores activity logs

Data Access Objects (DAO)

- Handles database CRUD operations

4. External Integration Components

- **Mail Server Interface**
 - Connects to external email services
- **Third-Party NLP Library Integration**
 - Used for text processing

Summary of Application Components

◆ **High-Level Components**

- Presentation Layer
- Business Logic Layer
- Data Layer

◆ **Core Functional Components**

- Email Fetching
- NLP Processing
- Spam Detection
- Response Generation

- Rule Engine
- Workflow Controller

◆ **Supporting Components**

- Authentication
- Database Management
- Logging System