Exercise 2

Task 1 :Research the SaaS company personio.com. What functionality does its software offer to its clients? Please draw an structural architecture that models this HR Software. Think about proper components and interfaces to existing 3rd party services. Describe your architecture and categorise your work within existing architectural styles.

- **Personio** is a **SaaS company** offering **HR software** for **small and medium-sized enterprises (SMEs)**.

Its main functionalities include:

- **Applicant Tracking System (ATS):** Supports managing recruitment and onboarding new employees.

- **Employee Data Management:** Collcets and centralizes all employee information and documents

- **Time and Attendance Tracking:** Records working hours, processes leave requests, and generates detailed reports.

- **Performance Management:** Support goal setting, feedback, and evaluations.

- **Payroll and Benefits Management:** Automate payroll, taxes, and benefits.

- **Learning and Development:** Organizes training programs and monitors learning progress

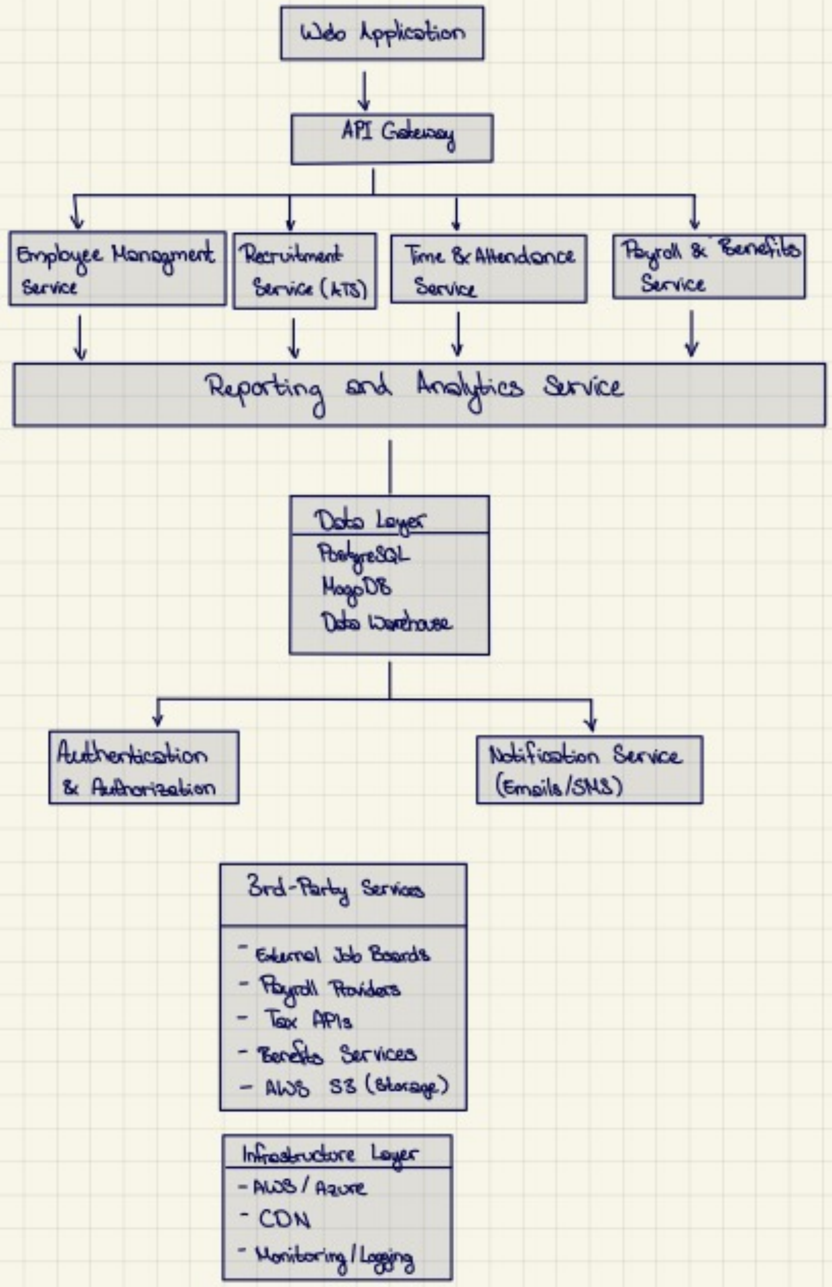
- **HR Analytics and Reporting:** Captures Analyze HR data through customizable dashboards.

Structural Architecture:

1. User Interface:
○ Web Application
2. Application Layer:
○ Employee Management Service
○ Payroll Service
○ Recruitment Service
○ Time and Attendance Service
○ Reporting and Analytics Service
3. Integration Layer:
○ API Gateway
○ Authentication and Authorization Service
○ Notification Service (Email, SMS)
4**. Data Layer:**
○ Relational Database (e.g., PostgreSQL)
○ NoSQL Database (e.g., MongoDB)
○ Data Warehouse (for analytics and reporting)
5. **Third-Party Services:**
○ Tax Calculation Service
○ Benefits Management Service
○ External Job Boards
○ Payroll Providers
○ Cloud Storage Services (e.g., AWS S3)
6. Infrastructure:
○ Cloud Hosting (e.g., AWS, Azure)
○ CDN (Content Delivery Network)
○ Monitoring and LoggingArchitecture Description and Categorization:

- Components: The architecture is composed of modular services that handle different
HR functions
- Interfaces: The API Gateway manages communication between the application and third-party services
- Architectural Style: Each HR function is encapsulated in its own service, allowing independent development, deployment, and scaling

Exercises 2 – Task 1



Task 2: Read the paper Yahyavi.pdf pages 1-20 (see CampUAS). Describe this architectural style and compare it to the styles mentioned in the lecture.

Architectural Style:

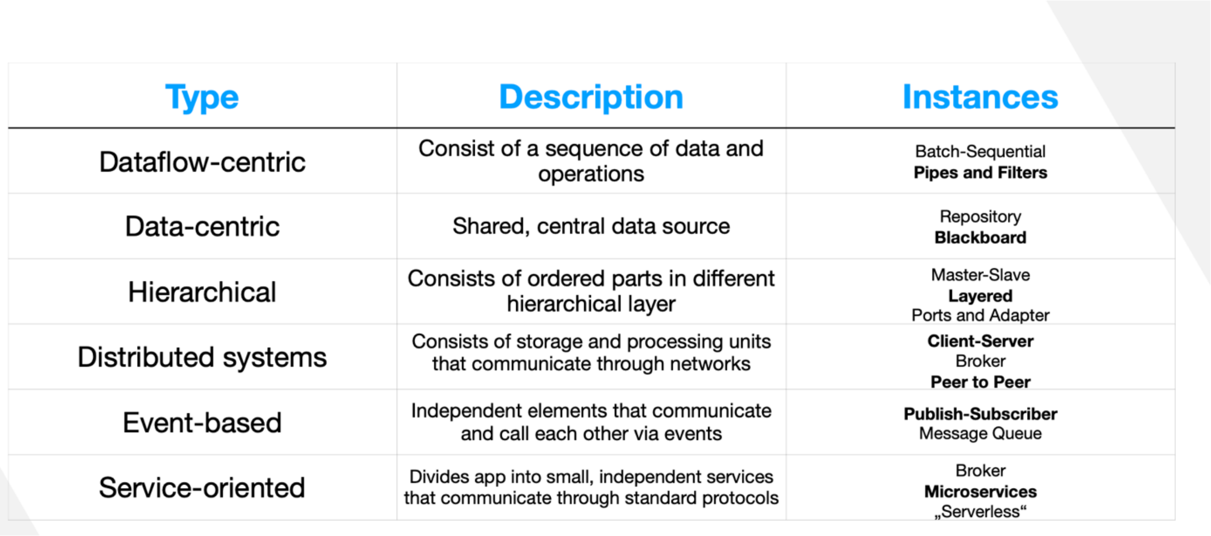
The paper discusses a Peer-to-Peer (P2P) Architecture, characterized by:
• **Decentralization:** No central server; each peer acts as both client and server.
• **Scalability:** New nodes can be added easily
• **Fault Tolerance**: The system can handle failures of individual peers without affecting the overall network. – The System continues to work even if nodes fail

Comparison to Other Styles:

• **Client-Server Architecture:** Centralized with distinct roles for clients and servers, whereas P2P is decentralized

• **Microservices Architecture:** Composed of loosely coupled services focusing on specific business capabilities, whereas P2P focuses on distributing tasks among peers

Monolithic Architecture: Single unified codebase, whereas P2P is distributed with each peer handling tasks independently



Type	Description	Instances
Dataflow-centric	Consist of a sequence of data and operations	Batch-Sequential **Pipes and Filters**
Data-centric	Shared, central data source	Repository **Blackboard**
Hierarchical	Consists of ordered parts in different hierarchical layer	Master-Slave **Layered** Ports and Adapter
Distributed systems	Consists of storage and processing units that communicate through networks	**Client-Server** Broker **Peer to Peer**
Event-based	Independent elements that communicate and call each other via events	**Publish-Subscriber** Message Queue
Service-oriented	Divides app into small, independent services that communicate through standard protocols	Broker **Microservices** „Serverless“

Task 3: Illustrations of the 4 Architectural Views for a Ticket Machine System

Logical View:

- Components: User Interface, Ticket Processing, Payment Processing, Ticket Printing
- Modules: Ticket Selection, Printing Service

Development View:

- Layers: Presentation Layer (UI), Logic Layer (Ticket and Payment Processing), Data Layer (Database for ticketing data)

Process View:

- Processes: User interacts with the UI, system processes ticket request, processes payment, prints ticket, logs transaction

Physical View:

- Deployment: Touchscreen interface, printer, payment terminal

Task 4: Suggested Architectural Patterns

1. **Whistleblowing System on the Internet:**
- Recommended Pattern: Anonymous Communication Pattern
- Reason: Ensures user anonymity and data protection, essential for whistleblowing platforms. Utilizes encryption and anonymization techniques.

2. Video Conferencing System:

- Recommended Pattern: Event-Driven Architecture
- Reason: Manages real-time communication efficiently, supports high concurrency, and scales to handle many simultaneous video streams.

3**. GPS-Tracker for Cats:**

- Recommended Pattern: Microservices Architecture
- Reason: Modular and scalable, notification services, and integration with various devices.