1

Personio is a software company from Germany that helps small and medium-sized businesses manage their human resources (HR). The software makes HR tasks easier with features like recruitment, employee management, payroll, and performance tracking.

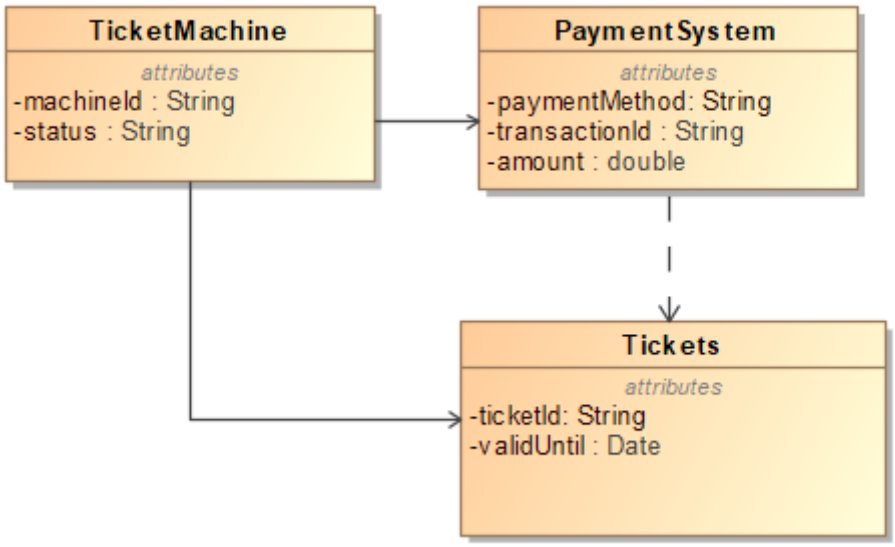
Personio's system is designed in different layers to handle tasks smoothly. The User Interface Layer has a web app for HR admins and an employee self-portal, plus mobile apps for iOS and Android. The Services Layer includes services like recruitment, employee management, payroll etc which cover the main HR functions. The Data Storage Layer keeps important data like employee details, contracts. Finally, the Integration Layer connects Personio to other services using an API, allowing integrations with tools like Slack, Microsoft Teams.

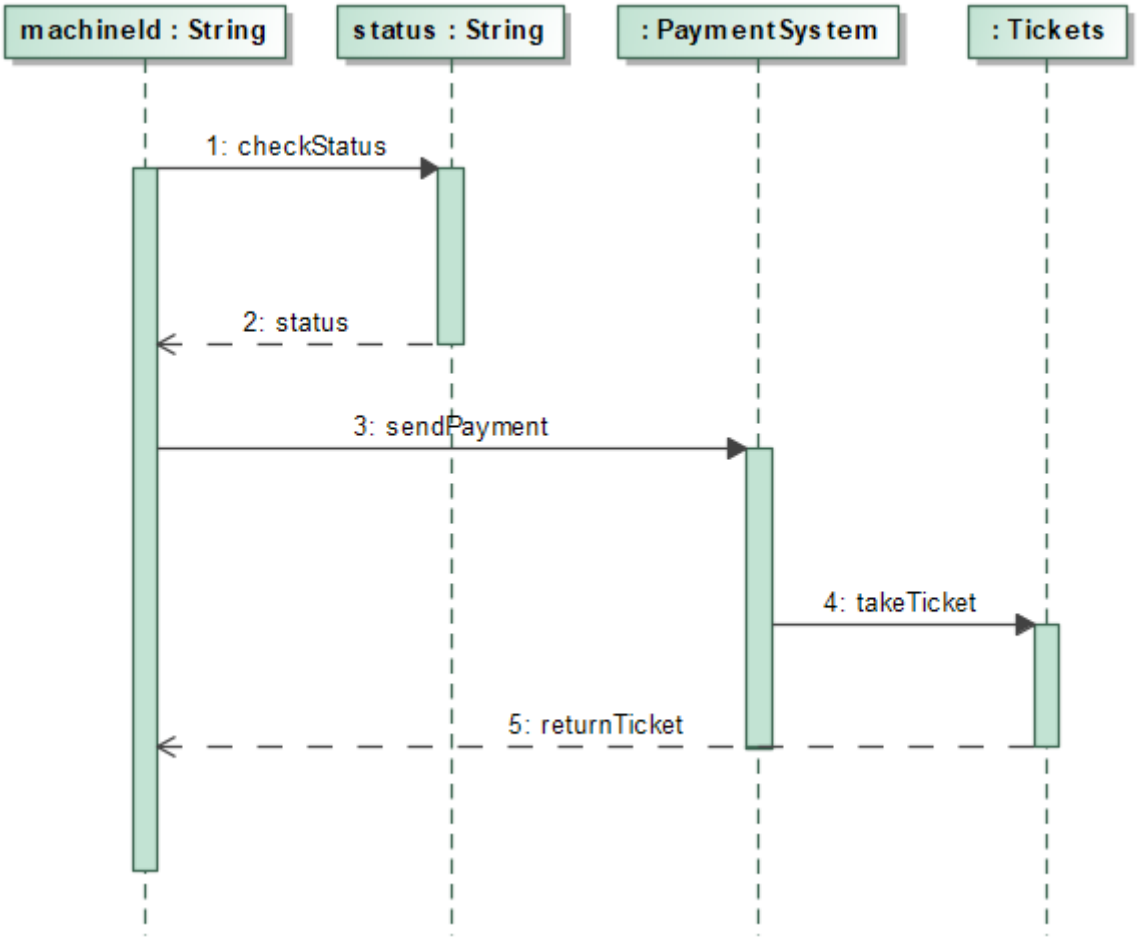
The way Personio is built includes a few common software styles. One of them is Service-oriented Architecture. Each part of the software works separately, which makes it easier to fix problems and update features. Also, Event-Driven Architecture take apart. The software responds to different events, like a new hire, and triggers the right actions without waiting for commands from other parts of the system.

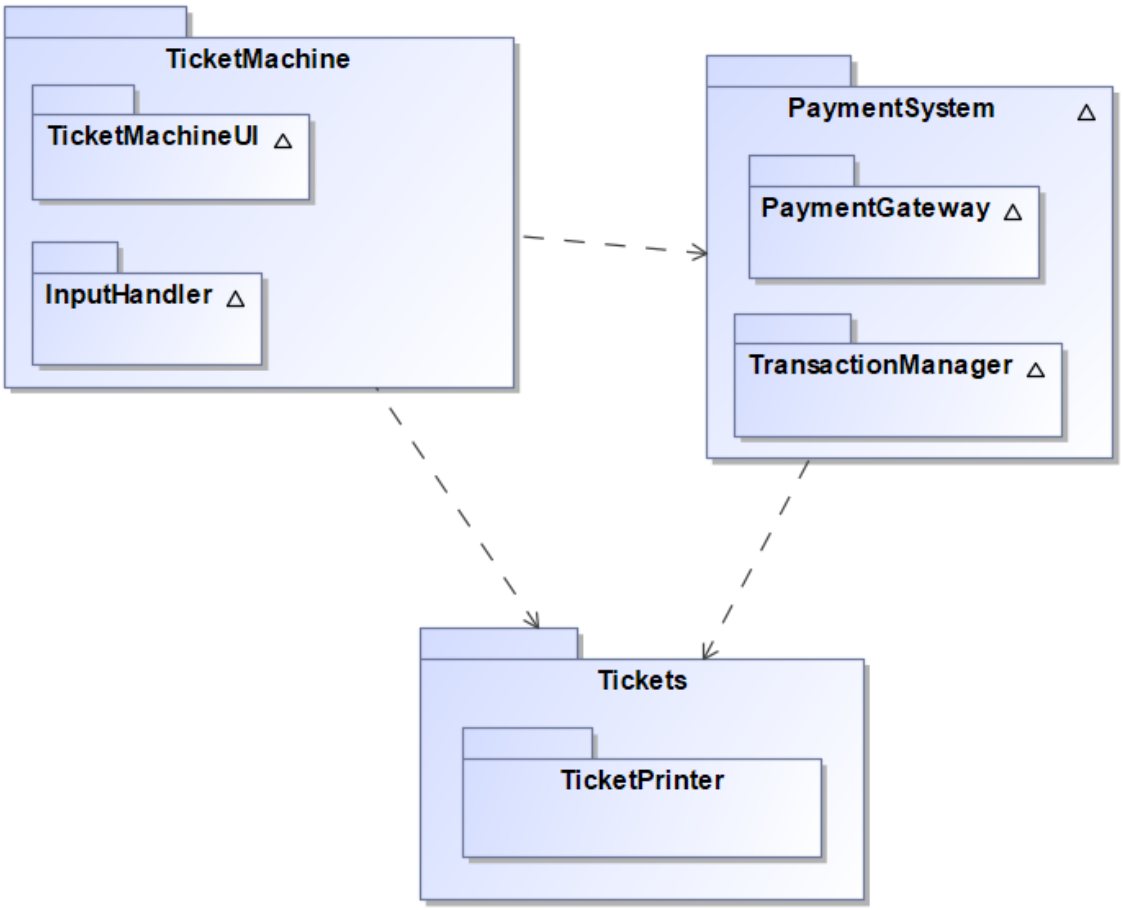
2

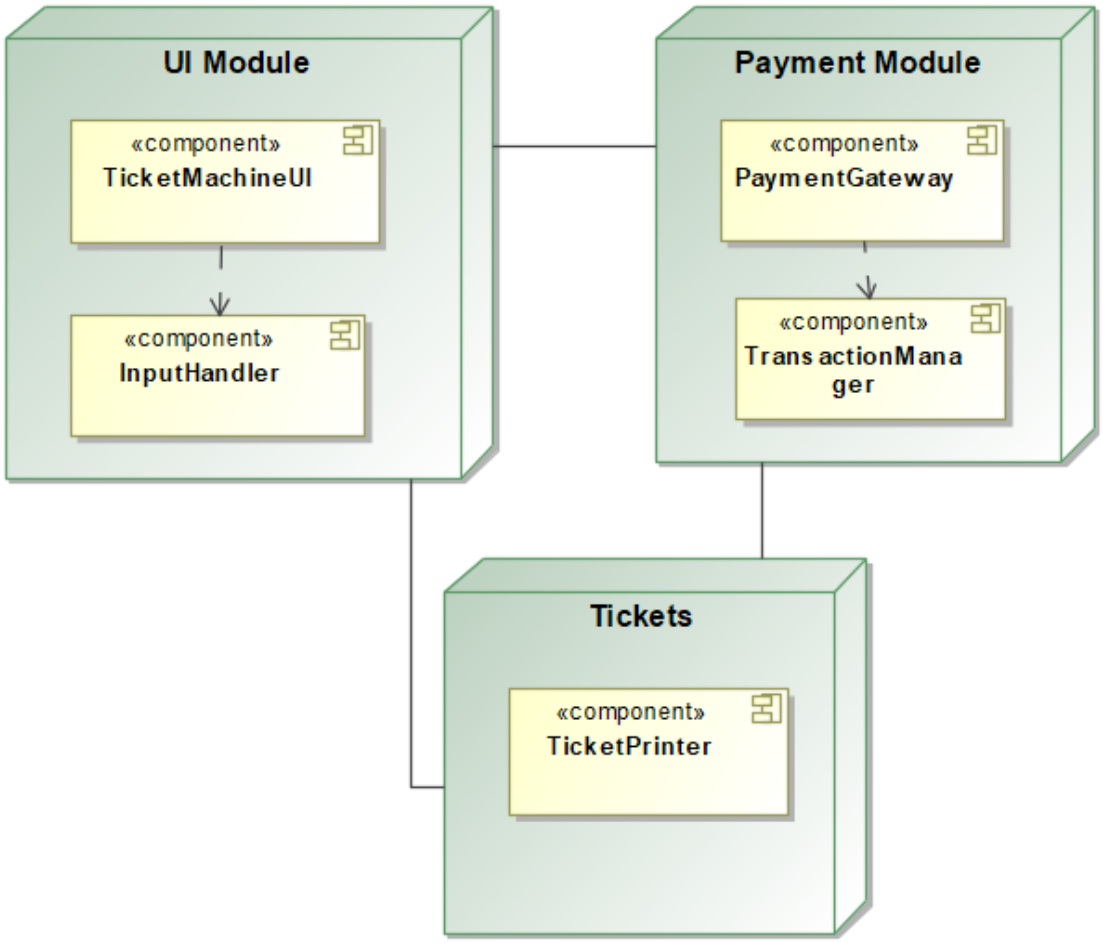
In this style, players' computers connect to each other directly. Each player shares the work, like saving game data or sending updates. There is no big central server. This makes games cheaper and more scalable. But it also makes it harder to stop cheating and keep everything in sync. If I compare; It is like Peer-to-Peer in Distributed Systems. Also it uses Publish-Subscribe from Event-based systems (players get updates from nearby players). However, It is not like Dataflow or Repository because there's no central data store and It is not like Hierarchical systems because there are no layers or managers.

3




machineId : String
status : String
: PaymentSystem
: Tickets
1: checkStatus
2: status
3: sendPayment
4: takeTicket
5: returnTicket


TicketMachine
TicketMachineUI
InputHandler
PaymentSystem
PaymentGateway
TransactionManager
Tickets
TicketPrinter



4

a) Whistleblowing system:

Data-centric, Repository Pattern. Because you need a safe, central place to store all the reports. It should be secure, easy to manage, and easy to search later."

b) Video conferencing system:

Event-based Pattern. Because many people talk and listen at the same time. Events (like someone speaking) need to go quickly to all others."

c) GPS tracker for cats:

Distributed systems, Client-Server Pattern. Because the cat's device (client) sends its location to a server, and the owner’s app can ask the server to get the cat’s latest position."