

**PROJECT DESIGN PHASE**  
**SOLUTION ARCHITECTURE**

Date	01 NOVEMBER 2025
Team ID	NM2025TMID01728
Topic Name	Streamlining Ticket Assignment for efficient support operation
Maximum Marks	4 Marks

## Solution Architecture:

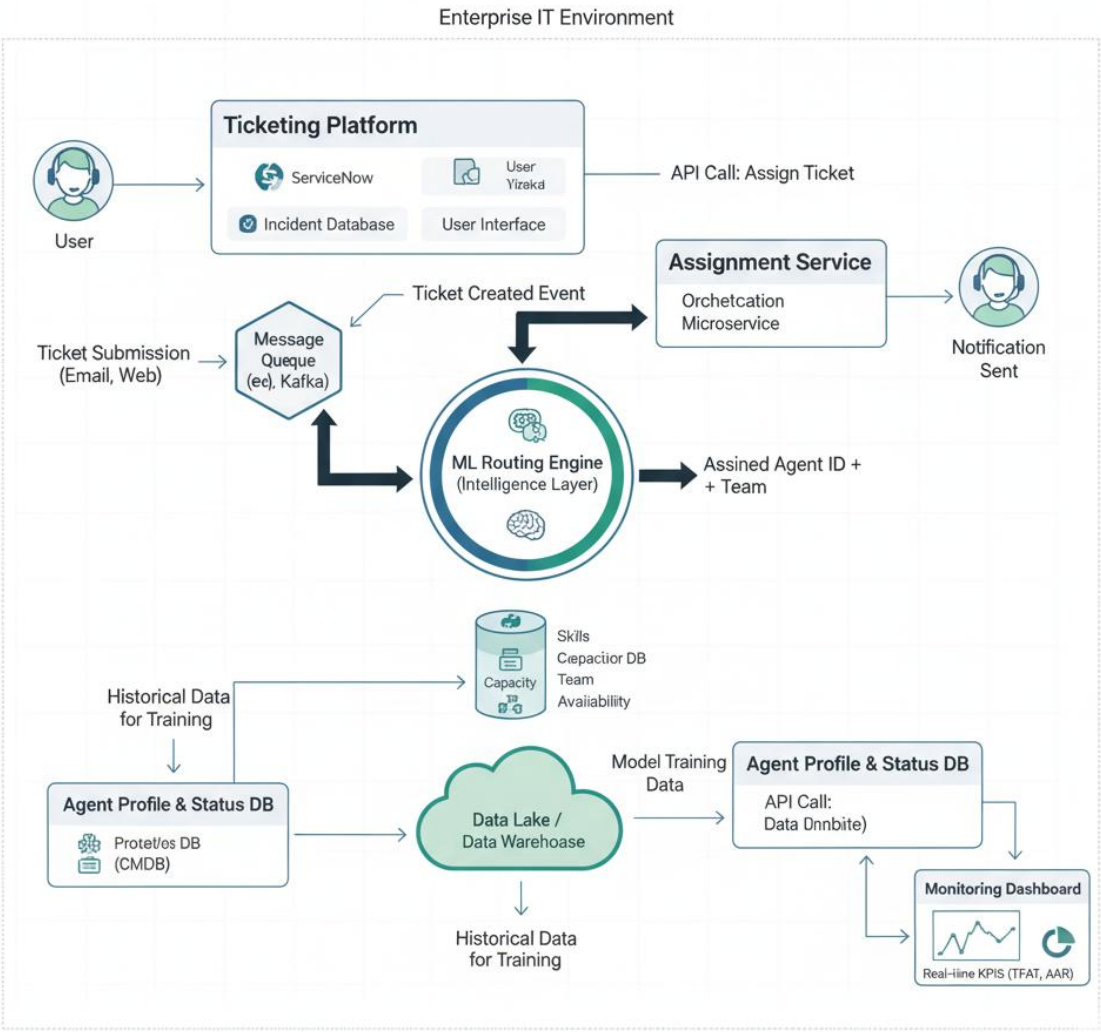
The proposed solution architecture is designed for a highly scalable and decoupled system that minimizes latency in the critical assignment process while maximizing accuracy. It leverages an **Event-Driven Architecture (EDA)** where the creation of a new ticket triggers the intelligent routing workflow.

### Architectural Flow: Ticket Processing (Event-Driven)

The system operates through the following steps when a user submits a ticket:

1. **Ticket Submission:** A new ticket is created by the user (via email, web portal, or API) in the core **Ticketing Platform**.
2. **Event Trigger:** The creation event is immediately published to a **Message Queue** (or an internal service bus).
3. **ML Router Ingestion:** The **Assignment Service** consumes the event and passes the ticket details (Title, Description, Category, Priority) to the **ML Routing Engine**.
4. **Intelligence Layer (ML Engine):**
  - **NLP Processing:** The engine uses Natural Language Processing to extract key entities, intent, and sentiment from the text.
  - **Classification:** It runs the data through a **Trained ML Model** (e.g., a Classification Model) to determine the accurate **Group/Team** and **Priority**.
  - **Agent Matching:** The engine consults the **Agent Profile & Status Database** (CMDB) to identify agents with matching skills (Skill-Based Routing) and available capacity (Load Balancing).
5. **Assignment Logic:** The Assignment Service receives the optimal Agent ID and Team from the ML Engine.
6. **Assignment Execution:** The Assignment Service calls an internal API on the **Ticketing Platform** to set the `Assigned To` field and update the status in a single, atomic transaction.
7. **Confirmation & Alert:** The Ticketing Platform triggers notifications to the newly assigned agent and sends an immediate acknowledgement to the customer.

Solution Architecture: Automated Ticket Assignment for Efficient Support



Key Architectural Components

Component	Function / Technology	Purpose in Assignment
Ticketing Platform	ServiceNow, Zendesk, Jira Service Management, etc.	Core system for ticket creation, storage, and history. Acts as the <b>source of truth</b> and the final destination for the assignment outcome.
API Gateway	NGINX, API Management Tool	Manages incoming ticket creation APIs, security, and authentication before passing requests to the core platform.

Component	Function / Technology	Purpose in Assignment
Message Queue	Kafka, RabbitMQ, AWS SQS/Azure Service Bus	<b>Decouples</b> the ticket creation process from the heavy routing logic. Ensures assignment processing doesn't slow down ticket submission and handles high volume/retries.
Assignment Service	Microservice (e.g., Python/Flask or Node.js)	The central brain. Consumes the ticket event, orchestrates the ML routing, and executes the final assignment transaction.
ML Routing Engine	Python ML Framework (e.g., Scikit-learn, TensorFlow)	<b>Intelligence Layer:</b> Hosts the trained classification model for Category/Priority and the custom logic for Skill-Based Routing and Load Balancing.
Agent Profile & Status DB (CMDB)	Relational Database (Postgres, MySQL) or dedicated CMDB	Stores agent skills, certifications, working hours, and real-time workload (number of open, in-progress tickets) for load balancing.
Data Lake/Data Warehouse	Snowflake, AWS S3, Azure Data Lake	Stores all historical ticket data, resolution data, and agent performance logs. <b>Feeds and trains the ML Routing Model</b> continuously.
Monitoring Dashboard	Grafana, Kibana, PowerBI	Provides real-time metrics on <b>Ticket First Assignment Time (TFAT)</b> and <b>Assignment Accuracy Rate (AAR)</b> to monitor performance and identify model drift.