Project Design Phase II

Technology stack

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

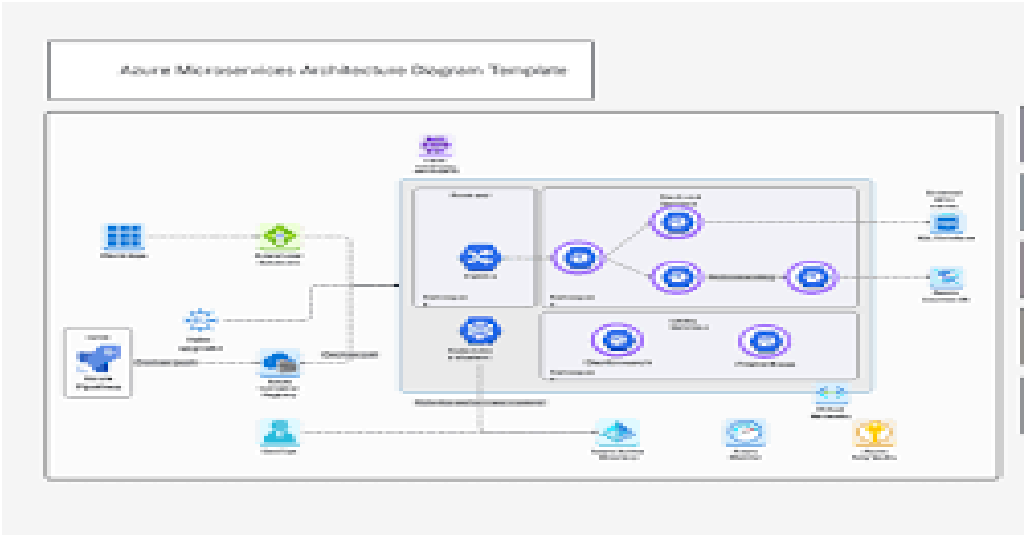
Streamlining Ticket Assignment for Efficient Support Operations

This section outlines the specific technologies chosen to implement the Automated Ticket Assignment solution, ensuring robustness, scalability, and seamless integration with the existing enterprise environment.

Component / Layer	**Technology / Tool**	**Rationale for Selection**
Core Ticketing Platform	**ServiceNow** (or Zendesk/Jira Service Management)	Acts as the **Source of Truth** for all incidents and agent profiles (CMDB). The solution is built using native **Business Rules** and **Script Includes** for deep integration.
ML/AI Routing Engine	**Python (Scikit-learn / TensorFlow)**	Python is the industry standard for **NLP** and **Machine Learning**. Libraries like Scikit-learn or TensorFlow are used to train the classification model for accurate ticket categorization.
Assignment Service (Microservice)	**Node.js** or **Python (Flask/Django)**	Used for developing the lightweight, high-performance microservice that orchestrates the assignment logic and API calls. **Node.js** is often preferred for its non-blocking I/O, ideal for low-latency assignment.
Message Queue / Event Bus	**Apache Kafka** (or AWS SQS/Azure Service Bus)	**Decouples** the assignment logic from the ticketing platform. Kafka handles high volumes of "Ticket Created" events, preventing system slowdowns during peak load.

Component / Layer	Technology / Tool	Rationale for Selection
Agent Profile & Status DB (CMDB)	**PostgreSQL** or **ServiceNow CMDB**	Used to store structured data like Agent Skills, Certifications, and **real-time capacity metrics**. PostgreSQL is robust for complex load-balancing queries.
Data Storage / Training	**AWS S3 / Azure Data Lake**	Highly scalable and cost-effective cloud storage for housing years of historical ticket data used to **train and re-train the ML models** periodically.
API Integration	**REST APIs (JSON/XML)**	Used for all communication between the Assignment Service and the Ticketing Platform (e.g., fetching agent status, executing the final assignment).
Monitoring & Visualization	**Grafana / Kibana** (or native Ticketing Platform Dashboards)	Provides real-time dashboards for tracking critical KPIs like **TFAT (Ticket First Assignment Time)** and **AAR (Assignment Accuracy Rate)**, ensuring model performance is maintained.
Version Control	**Git / GitHub (or GitLab/Bitbucket)**	Essential for collaborative development, code tracking, and managing different versions of the assignment scripts and ML models.

Technical Architecture Diagram:


Azure Microservices Architecture Diagram Template