

Project Design Phase

Solution Architecture

Date	30 June 2025
Team ID	LTVIP2025TMID30150
Project Name	Streamlining Ticket Assignment for Efficient Project Management
Maximum Marks	4 Marks

Solution Architecture:

The solution architecture for the **ServiceNow Ticket Assignment Optimization Project** is designed to streamline and automate the assignment of service tickets. It uses rule-based logic enhanced with workload analytics and machine learning (optional), integrated within ServiceNow via scoped apps or platform capabilities.

❖ Architecture Objectives

- Automate ticket assignment to reduce manual overhead and improve response time
- Provide workload-based distribution to avoid team overload
- Integrate predictive logic (optional) based on historical ticket data
- Support SLA compliance with proactive escalation and reassignment
- Ensure scalability and adaptability for various departments and teams

Component	Department
User Interface	ServiceNow front-end for agents/admins to view assigned tickets and dashboard.
Assignment Rule Engine	Built-in ServiceNow assignment rules or custom script includes rule conditions.
Workload Monitor	Dashboard module showing live workload stats per team/agent.
ML Prediction Engine	(Optional) Trained model hosted externally or via MID Server for ticket routing.
Ticket Database	Existing ServiceNow database with ticket metadata and history.
SLA Watcher Module	Monitors ticket age and reassigns or escalates based on SLA thresholds.
Integration Layer	Connects ServiceNow with external prediction API or internal logic modules.

Architecture Flow

1. User submits a ticket via the ServiceNow portal
2. Rule engine checks criteria (e.g., urgency, category, requester dept.)
3. Assignment decision is made based on rules or ML model output
4. Workload monitor ensures no overload; ticket is routed accordingly

5. SLA Watcher keeps track of ticket progress; escalates if needed
6. Assigned agent receives the ticket in their dashboard

Technology Stack

1. **Frontend:** ServiceNow portal (Now Experience UI / Classic UI)
2. **Backend:** ServiceNow Flow Designer / Business Rules / Script Includes
3. **Data Storage:** ServiceNow tables (e.g., incident, task, sys_user)
4. **Optional ML Layer:** Python-based model hosted on AWS Lambda / Flask API
5. **Visualization:** Performance Analytics dashboard, ServiceNow Reports

Development Phases

1. • Problem & Workflow Definition
2. • Rule Configuration in ServiceNow
3. • Dashboard and Workload Metrics Integration
4. • ML Model Training (if used)
5. • API Integration with ServiceNow
6. • Testing and Deployment

Scalability Considerations

- Modular rule design to support different departments (HR, IT, Facilities)
- Extensible API layer for AI/ML logic
- Global scalability via cloud deployment of ML and real-time analytics
- SLA logic can be customized per business unit or ticket type.

Solution Architecture Diagram:

