# Product Requirements Document (PRD) for Ticket Service

Contents

[**Product Requirements Document (PRD) for Ticket Service 1**](#_Toc204967516)

[**1. Overview 2**](#_Toc204967517)

[**2. Goals 2**](#_Toc204967518)

[**3. Scope 2**](#_Toc204967519)

[**In-Scope: 2**](#_Toc204967520)

[**Out-of-Scope: 2**](#_Toc204967521)

[**4. Functional Requirements 2**](#_Toc204967522)

[**4.1 Ticket Management 2**](#_Toc204967523)

[**4.2 Event Publishing 3**](#_Toc204967524)

[**4.3 Notification Integration 3**](#_Toc204967525)

[**5. Non-Functional Requirements 3**](#_Toc204967526)

[**6. System Diagrams 3**](#_Toc204967527)

[**6.1 Ticket-Service Workflow Design 3**](#_Toc204967528)

[**6.2 Sequence Diagram 5**](#_Toc204967529)

[**6.3. Data Flow Diagram (Level 0) 5**](#_Toc204967530)

[**6.4 Data Flow Diagram (Level 1) 6**](#_Toc204967531)

[**7. Ticket Lifecycle Workflow 6**](#_Toc204967532)

[**7.1 Workflow (Steps) 6**](#_Toc204967533)

[**7.2 Ticket Lifecycle – DFD 7**](#_Toc204967534)

[**7.3 Ticket Lifecycle- DFD level 1 7**](#_Toc204967535)

[**8. Optional Kafka Integration at Each Step (event emitted): 8**](#_Toc204967536)

## 1. Overview

The Ticket Service is a core microservice responsible for managing Ticket creation, updates, assignments, and status transitions. It integrates with Apache Camel and Kafka for asynchronous processing and event-driven notification.

## 2. Goals

* Enable users to create, assign, and track Tickets.
* Publish events (Ticket created, updated) to Kafka.
* Trigger notifications through the Notification Service.
* Ensure message routing and transformation via Apache Camel.

## 3. Scope

### In-Scope:

* REST APIs for Ticket operations
* Kafka integration
* Apache Camel middleware
* Role-based access (e.g., admin, user)
* Basic frontend integration readiness

### Out-of-Scope:

* Frontend implementation
* Direct notification logic (delegated to Notification Service

## 4. Functional Requirements

### 4.1 Ticket Management

* Create Ticket
* Assign Ticket
* Update Ticket status
* Delete Ticket
* View Ticket by ID or user

### 4.2 Event Publishing

* Publish events to Kafka:
  + Ticket.created
  + Ticket.updated
  + Ticket.assigned

### 4.3 Notification Integration

* Produce Kafka messages consumed by Notification Service

## 5. Non-Functional Requirements

* Scalability: Handle up to 10,000 Tickets/day
* Latency: Ticket creation response < 500ms
* Security: OAuth2 + JWT for APIs
* Reliability: At-least-once Kafka delivery
* Availability: 99.9%

## 6. System Diagrams

### 6.1 Ticket-Service Workflow Design

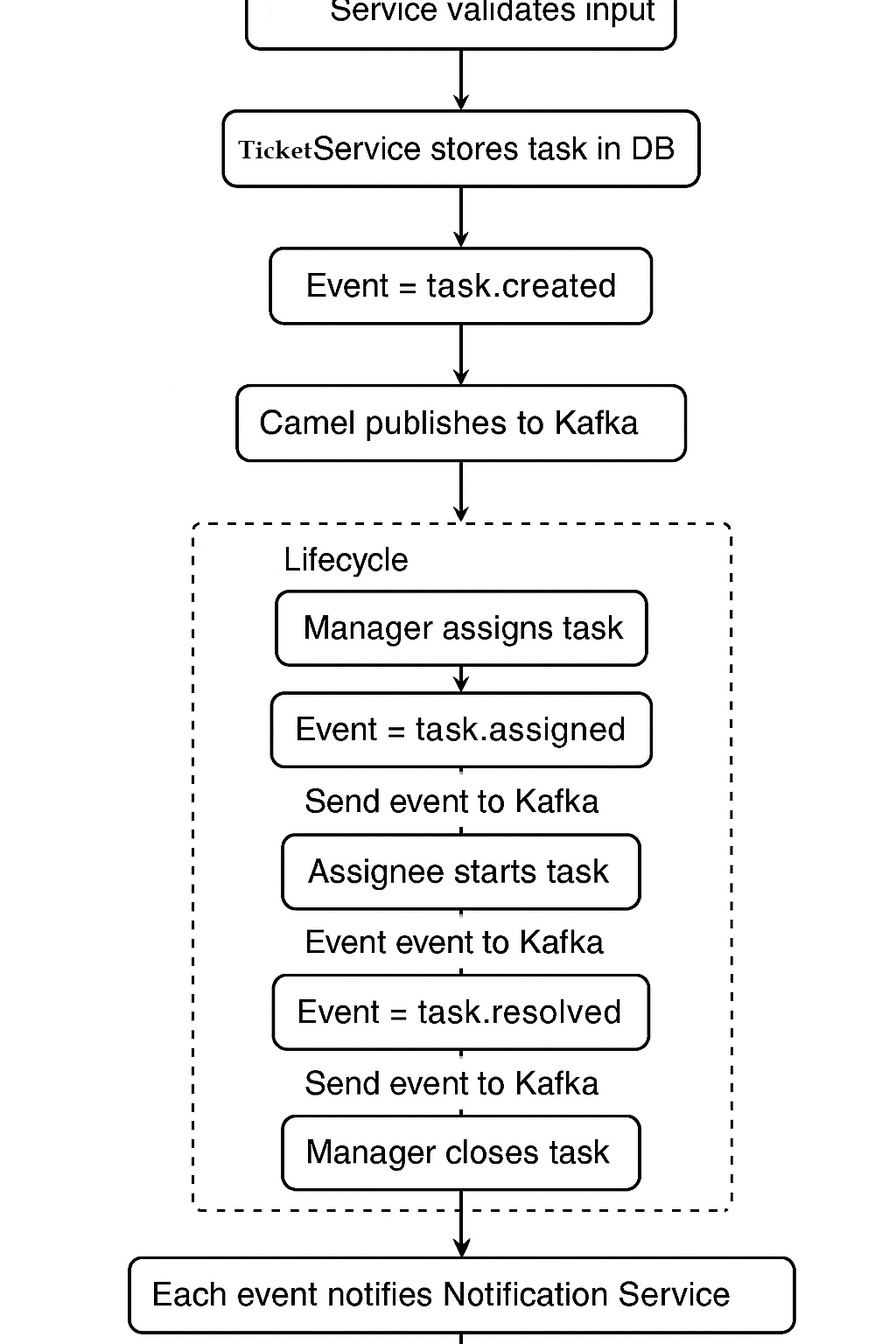


Fig 1: Ticket service and workflow management

### 6.2 Sequence Diagram

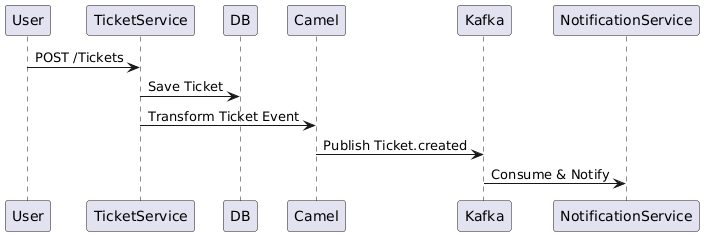


Fig 1: Sequence diagram – Ticket Service

### 6.3. Data Flow Diagram (Level 0)

A diagram of a task

AI-generated content may be incorrect.

Fig 2: DFD – Ticket service(level 0)

## 6.4 Data Flow Diagram (Level 1)

A diagram of a diagram

AI-generated content may be incorrect.

Fig 3: DFD – Ticket service( level 1)

6.5 Ticket Workflow full view

A diagram of a process

AI-generated content may be incorrect.

## 7. Ticket Lifecycle Workflow

**States:**

1. **Created** – Ticket is created by user
2. **Assigned** – Ticket is assigned to a responsible user
3. **In-Progress / Responded** – Work is being done or acknowledged
4. **Resolved** – Ticket is marked completed
5. **Closed** – Final verification and closure

### 7.1 Workflow (Steps)

1. **User** creates a Ticket → System sets status = Created
2. **Admin/Manager** assigns it to a user → status = Assigned
3. **Assignee** starts or responds → status = In-Progress or Responded
4. **Assignee** resolves the Ticket → status = Resolved
5. **Manager** or **System** closes the Ticket → status = Closed

### 7.2 Ticket Lifecycle – DFD

A diagram of a work flow

AI-generated content may be incorrect.

Fig 3: Work-flow -Ticket Service

### 7.3 Ticket Lifecycle- DFD level 1(Component Base)

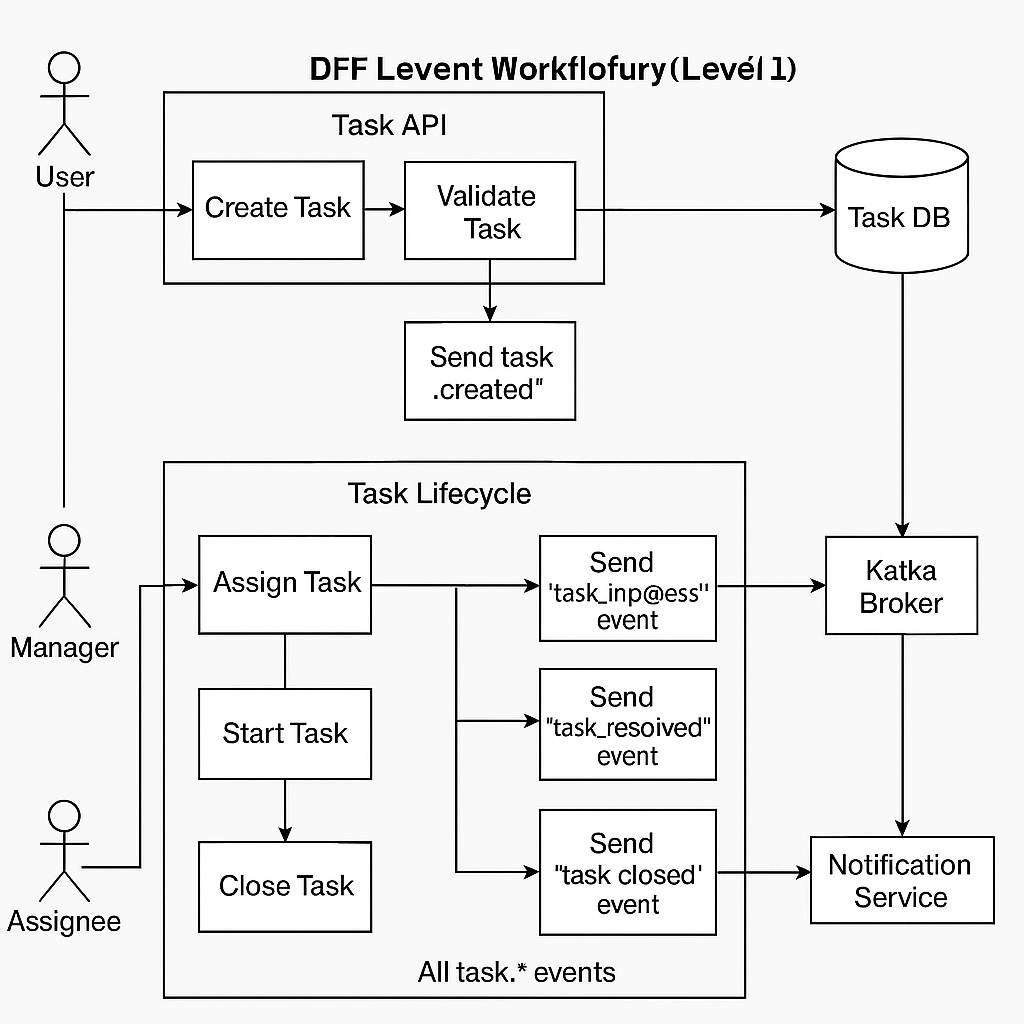


Fig 4: Work-flow -Ticket Service – Extended design

## 8. Optional Kafka Integration at Each Step (event emitted):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | Action | Kafka Event | Consumer | | Create Ticket | Ticket.created | Notification | | Assign Ticket | Ticket.assigned | Notification | | Submit Response | Ticket.responded | Logging / Workflow | | Resolve Ticket | Ticket.resolved | Notification | | Close Ticket | Ticket.closed | Audit / Reporting | |

# 9. Database design

A screenshot of a computer

AI-generated content may be incorrect.

|  |  |  |
| --- | --- | --- |
| Kaushik Nandy | Created | V1.0 |
|  |  |  |
|  |  |  |
|  |  |  |