

E-Gov Grievance Redressal System

A Scalable, Reactive Microservices Approach to
Transparent Governance

Presented By:

Yati
CHUBB
Java FSD
Amity University



Problem Statement

Current Challenges:

- Lack of transparency in grievance tracking.
- Delayed responses and SLA breaches without accountability.
- Manual and fragmented processes.
- Poor communication between citizens and government departments.

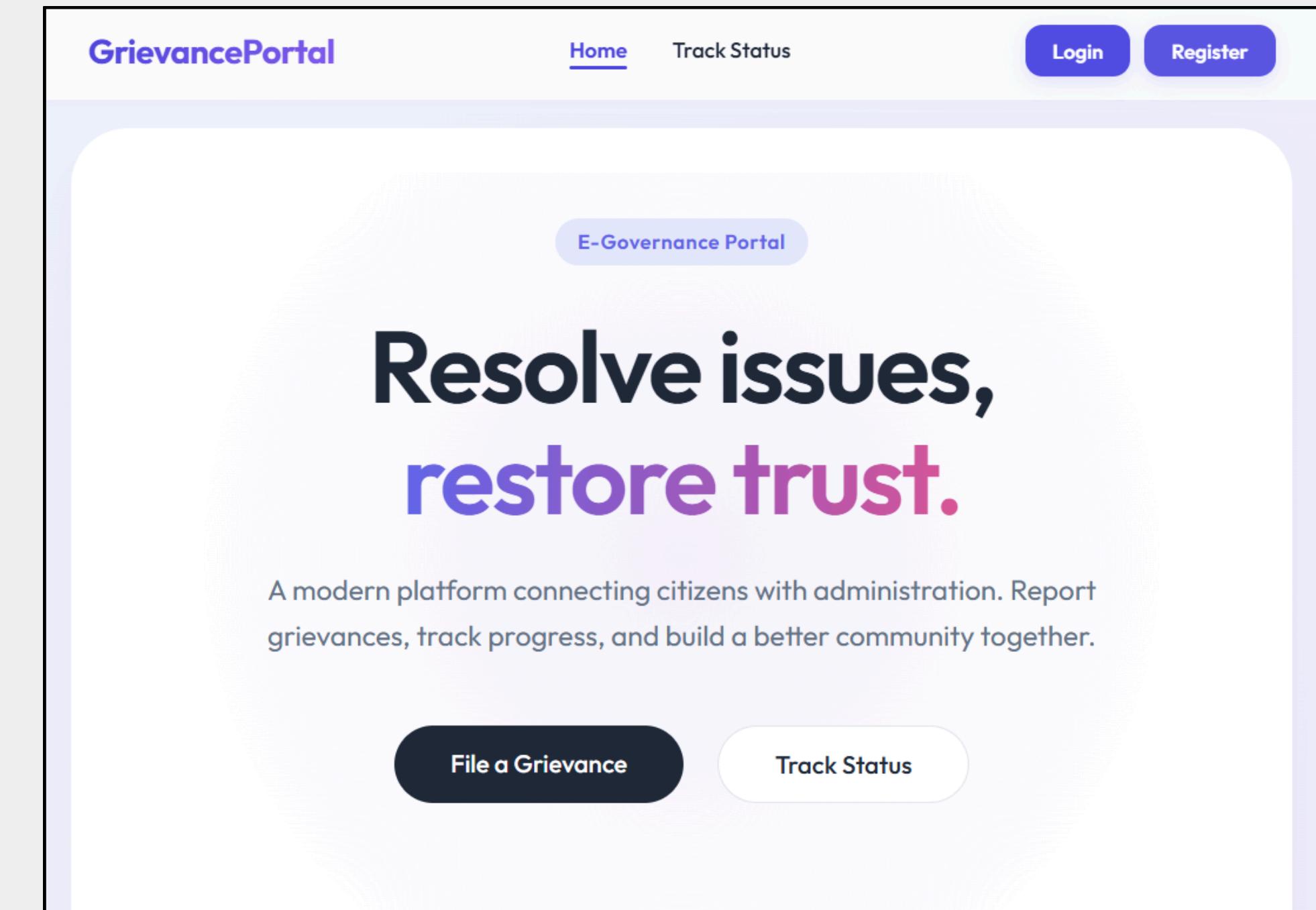
Need: A centralized, automated, and real-time system to handle public concerns efficiently.

Solution Overview

Project Goal: To provide a robust platform for citizens to file, track, and resolve grievances with escalation feature and real-time notifications.

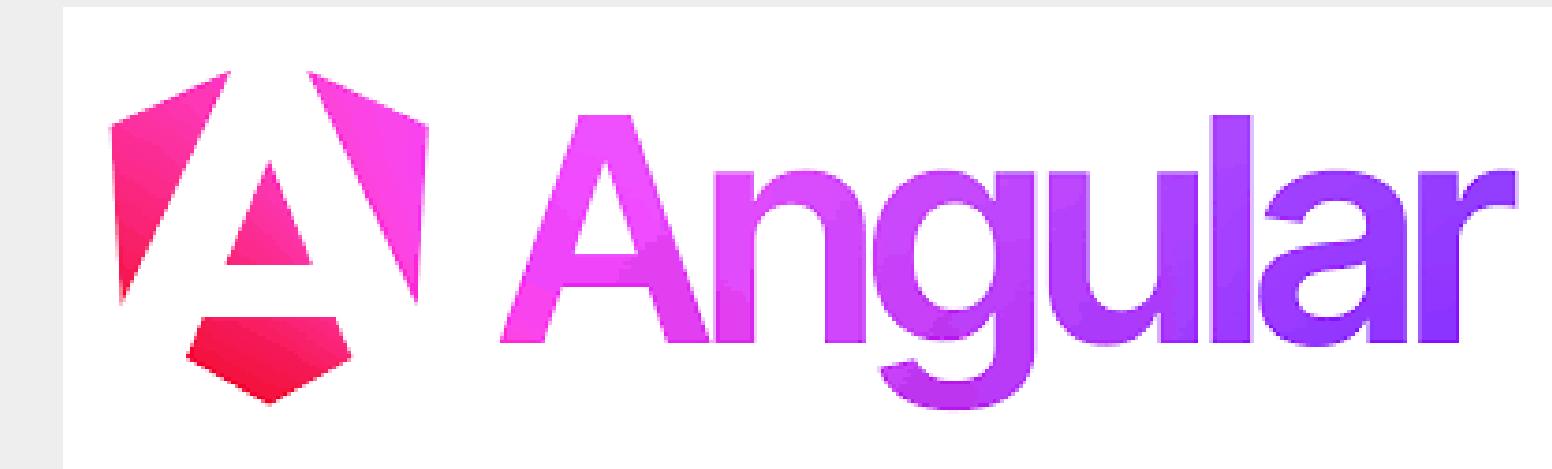
Key Pillars:

- Transparency: Real-time tracking for citizens.
- Accountability: Role-based access and SLA monitoring.
- Scale: Microservices architecture for high availability.
- Engagement: Automated email notifications.



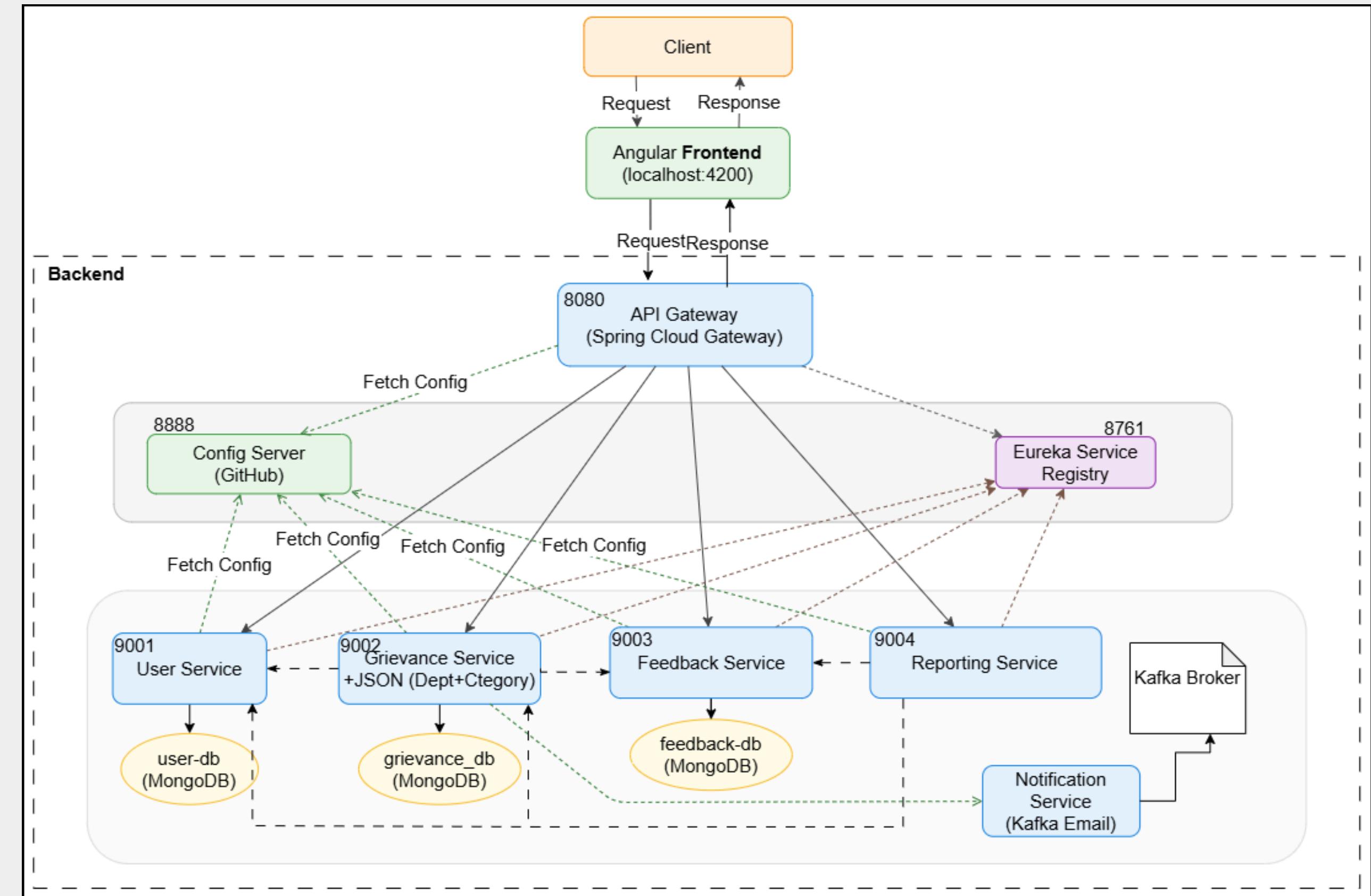
Technology Stack

- **Backend:** Java 21, Spring Boot 3.4, Spring Cloud (Gateway, Eureka, Config Server).
- **Reactive Programming:** Project Reactor (Mono/Flux), Spring WebFlux.
- **Frontend:** Angular 21
- **Database:** MongoDB 6.0 (NoSQL for flexible schema).
- **Messaging:** Apache Kafka (Event-driven status updates).
- **Infrastructure:** Docker, Docker Compose, Jenkins.

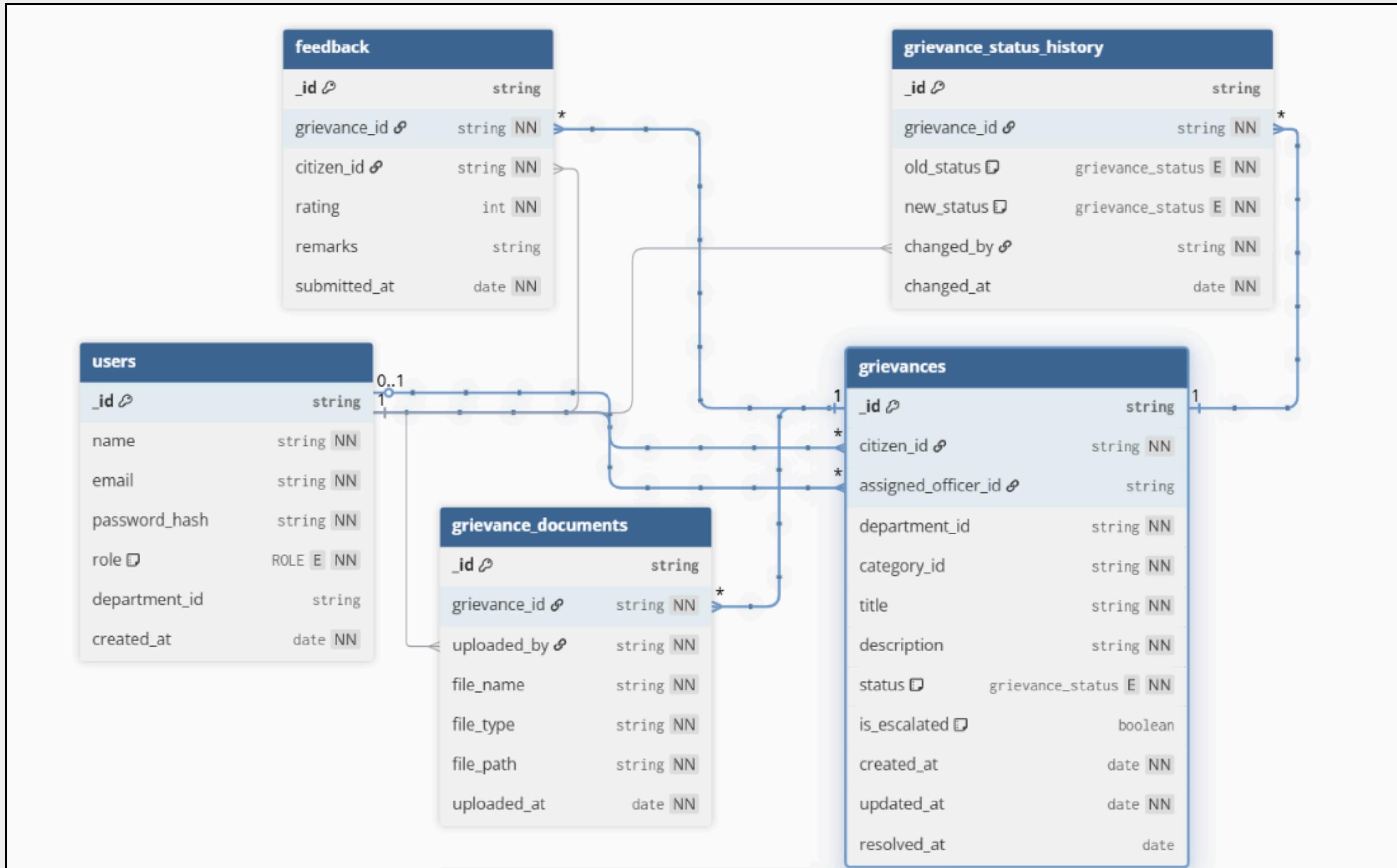


Microservices Architecture

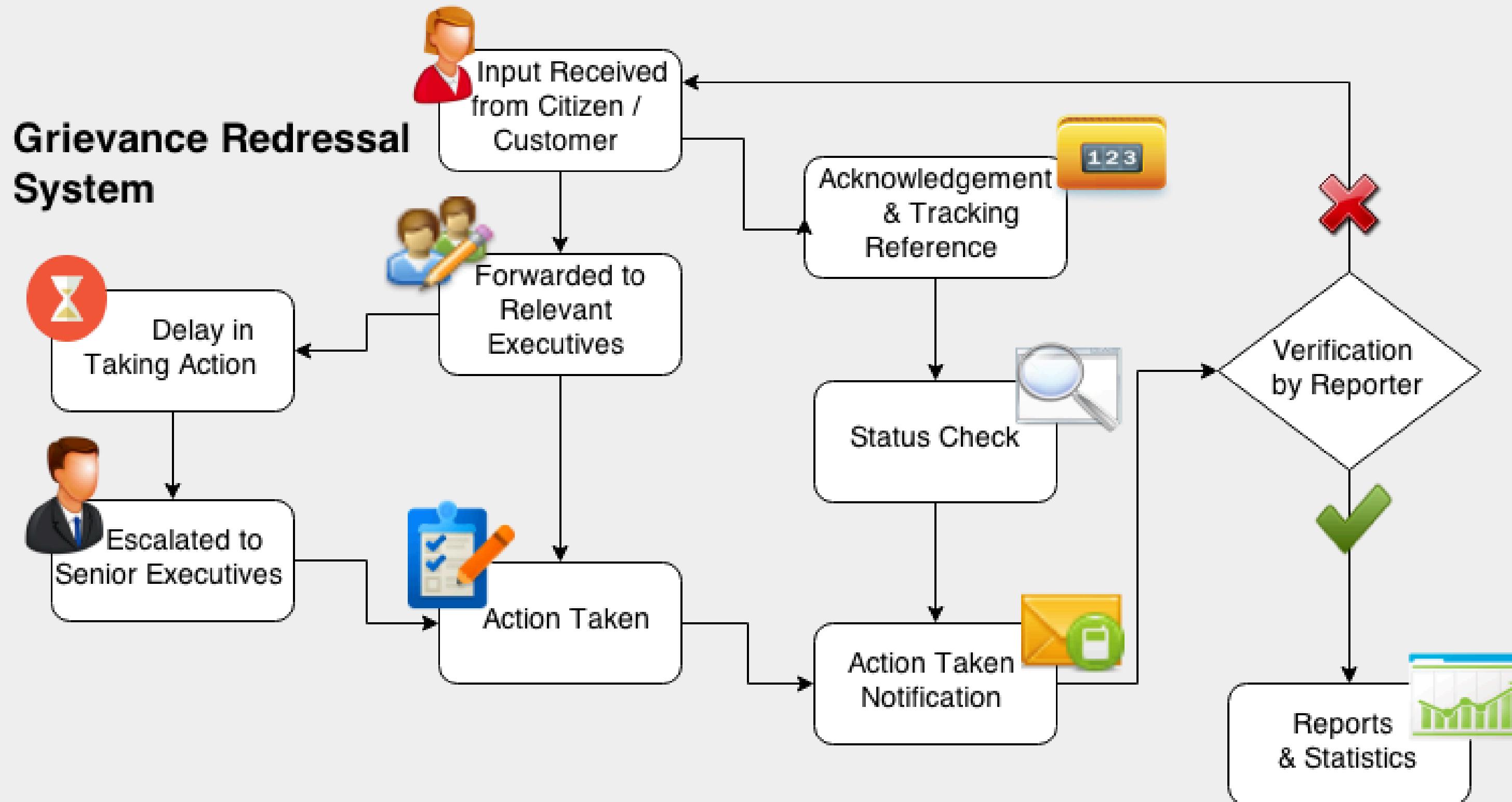
- **API Gateway:** Central entry point, JWT handling, routing
- **User Service:** Identity Management, RBAC
- **Grievance Service:** Core business logic, status management, file uploads.
- **Notification Service:** Kafka-driven
- **Reporting Service:** Analytics/summaries.
- **Eureka:** Service discovery



Database Design



Flow



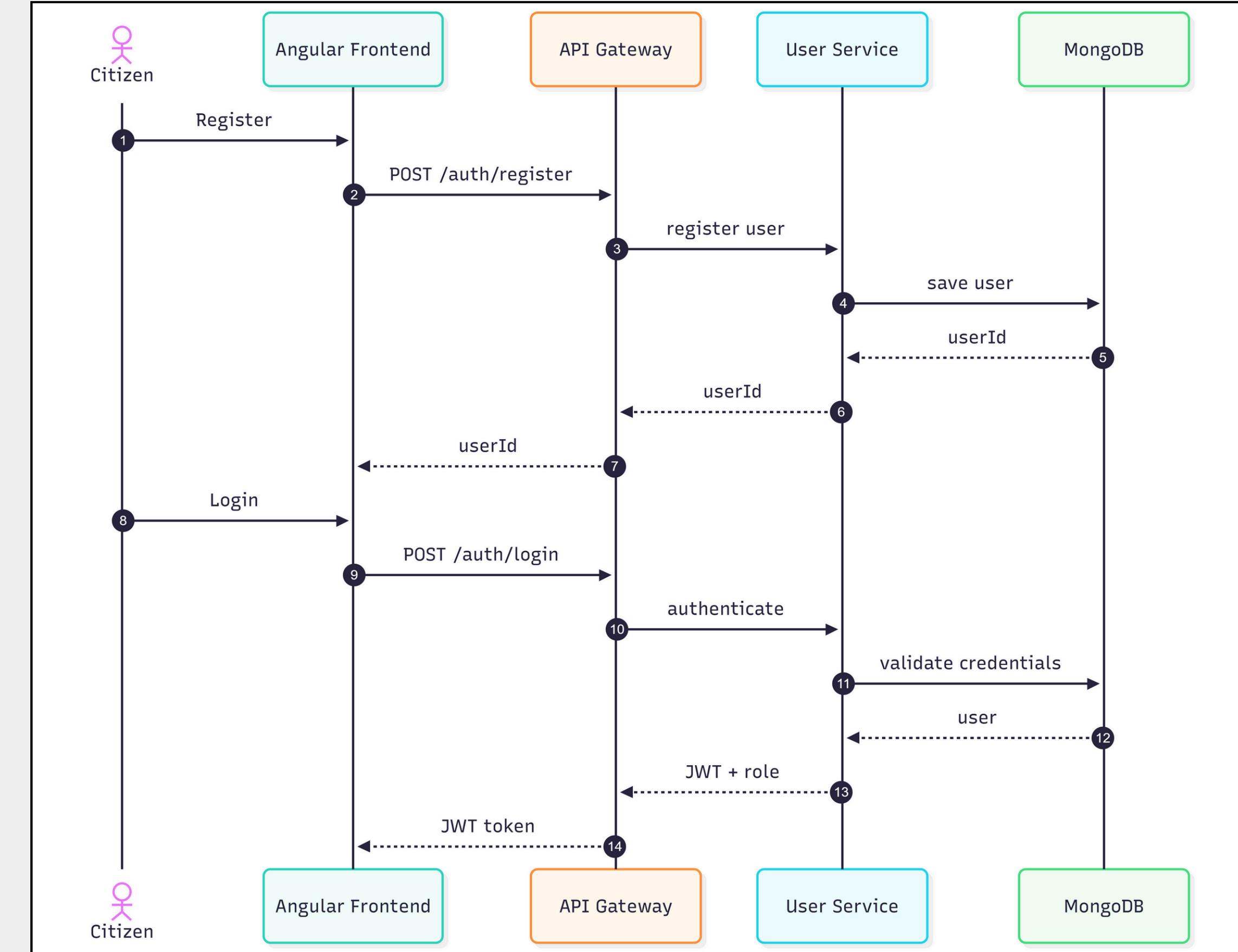
Roles & Responsibilities

Role	Responsibilities
Citizen	Register/Login , Lodge grievances, upload documents, view status, escalate delays, close/reopen grievances, give feedback
Officer	Review, process, and resolve grievances
Supervisor	Assign grievances, monitor escalations, view reports
Admin	Manage users and roles, view reports

Once logged in, everything the user can see or do depends on their role

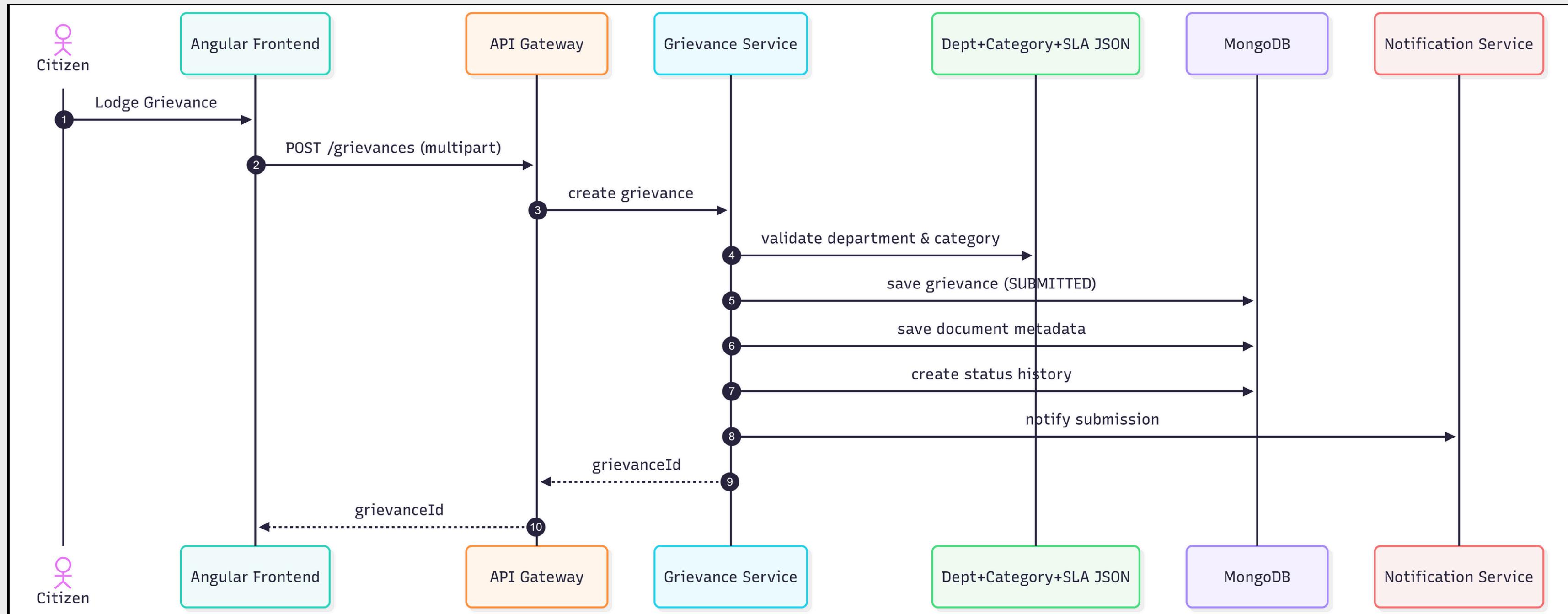
User Registration & Login

- Register → user enters details → password encrypted → ROLE = CITIZEN
- Admin create users → Officer / Supervisor / Admin
- Login → credentials verified → JWT (userId + role)
- Access → JWT sent with requests → role checked → allow/deny



Grievance Creation

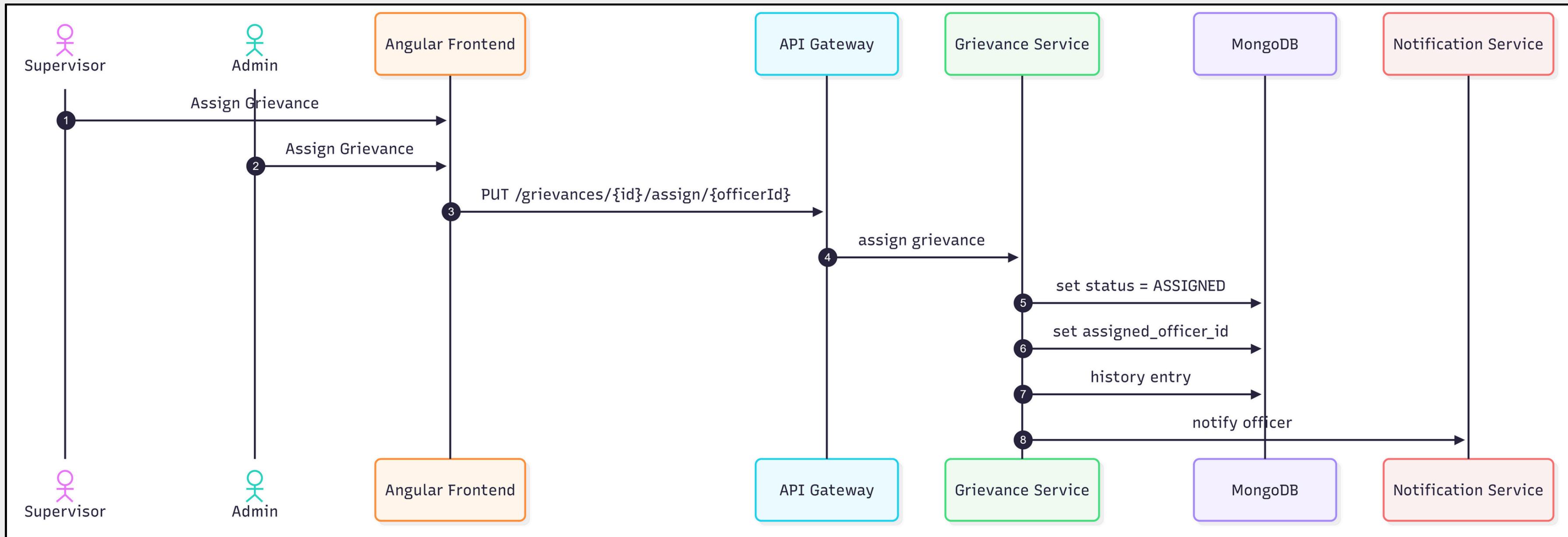
Citizen → Lodge Grievance → Enter title, description, department, category → Uploads documents



System → validate → generates id → sets status SUBMITTED → stores document metadata → creates status history → sends email notification

Grievance Assignment

Supervisor/Admin → views SUBMITTED grievances → Assigns grievance → Officer



System → validates officer belongs to department → updates status → ASSIGNED → saves → assigned_officer_id → logs status history → email

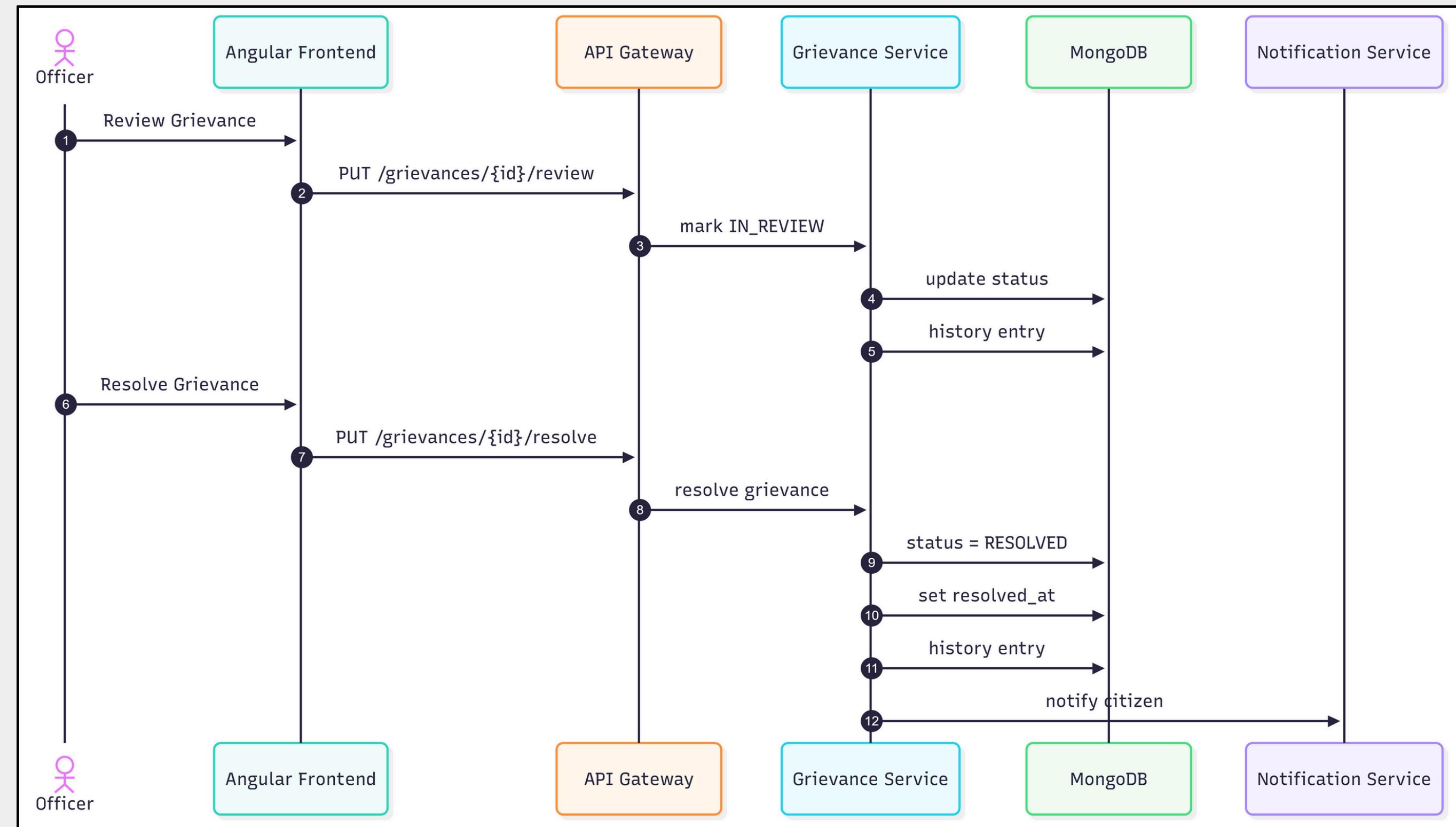
Grievance Review & Resolution

Officer → logs in → views assigned grievances

- Reviews details & documents
- Updates status → IN REVIEW
- Resolves grievance
- sets status RESOLVED

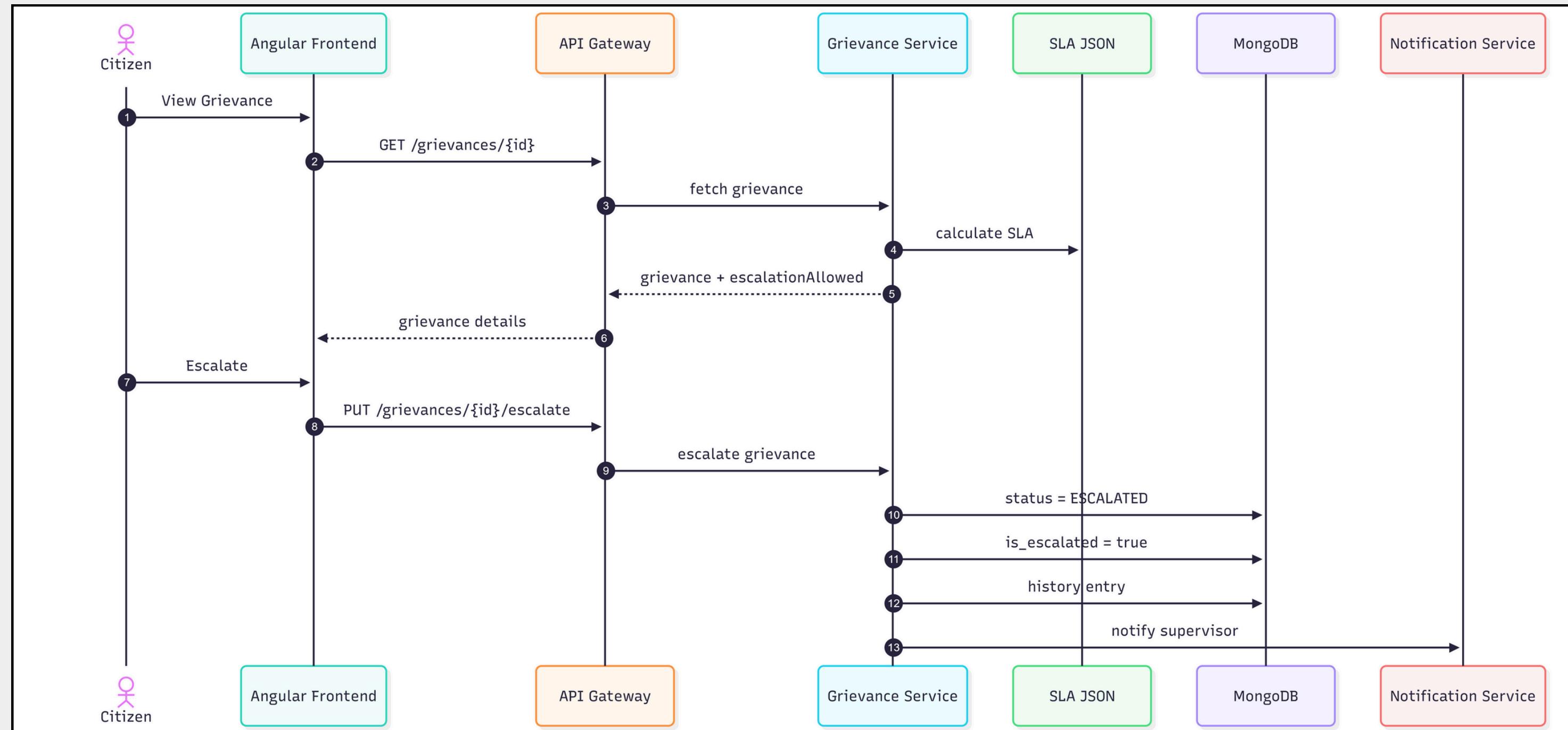
System → saves resolved_at

- logs history
- notifies citizen



SLA Evaluation & Escalation

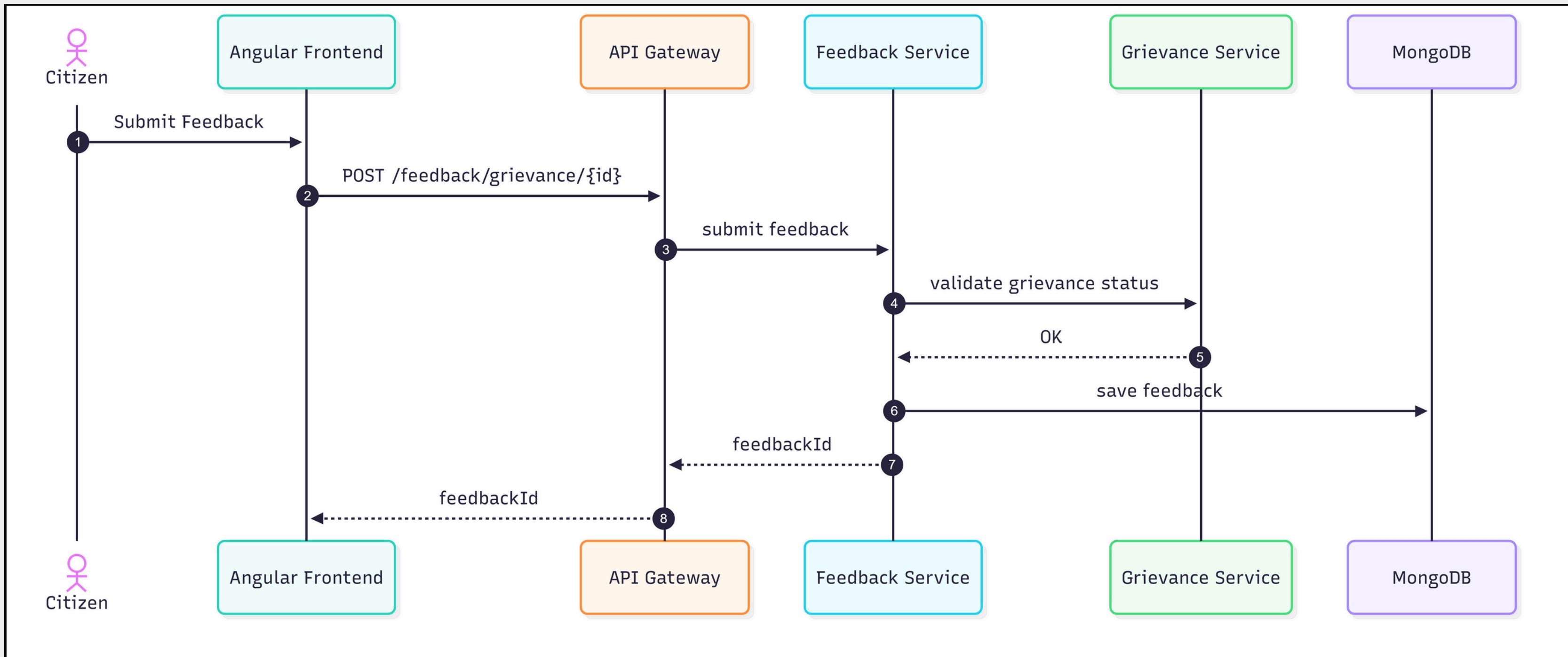
- SLA → defined per category (JSON)
- Escalation (Allowed if: not RESOLVED/CLOSED, SLA exceeded, not already escalated)



On escalate → status ESCALATED → `is_escalated = true` → history logged → supervisors notified

Feedback Submission

- Citizen → submits feedback after RESOLVED/CLOSED
- Includes rating (1-5) + optional remarks



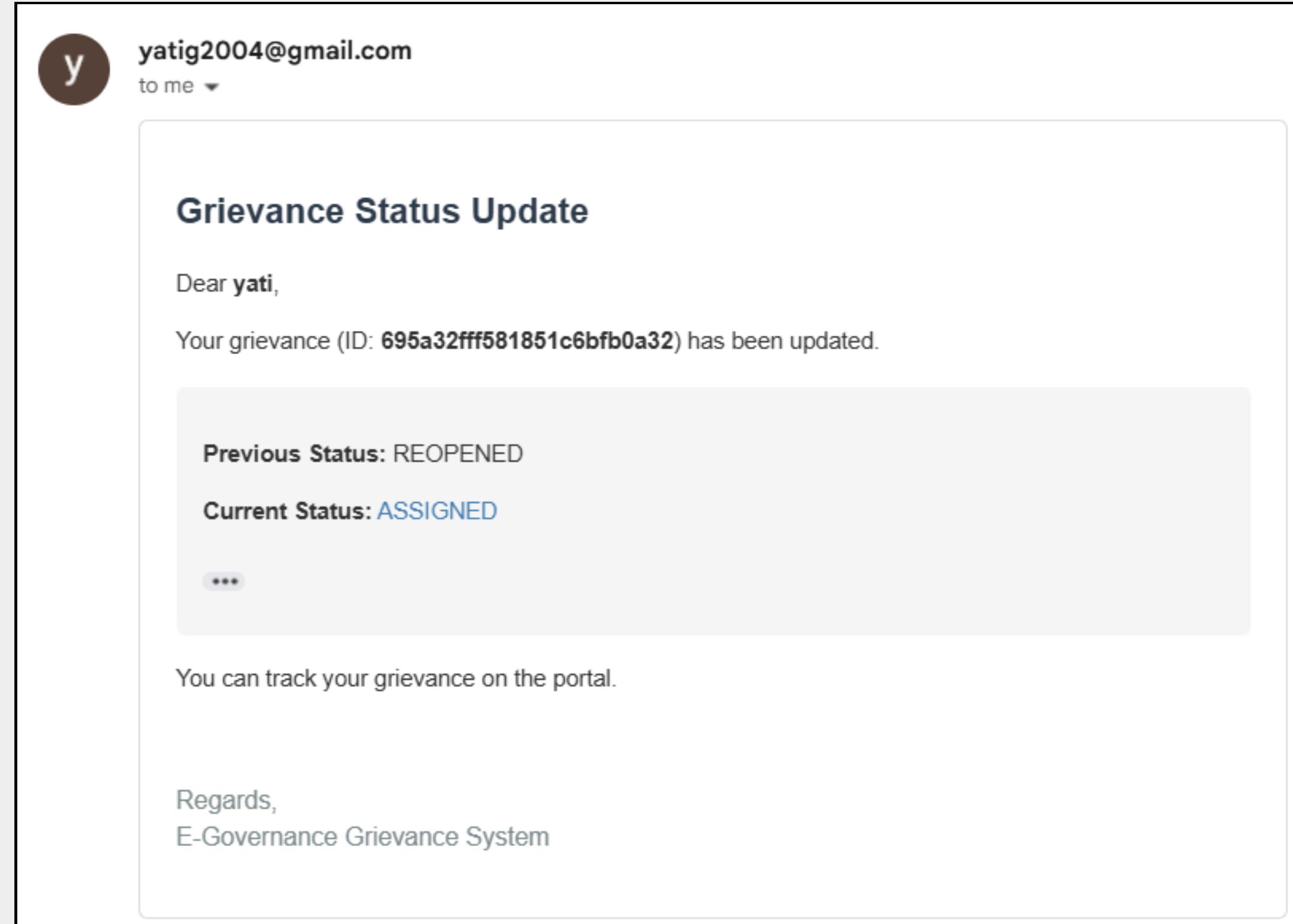
Notifications

Notifications are sent for:

- grievance submission,
- assignment,
- in review,
- resolution,
- escalation,
- reopened

The Notification Service:

- is stateless,
- is triggered only by internal services



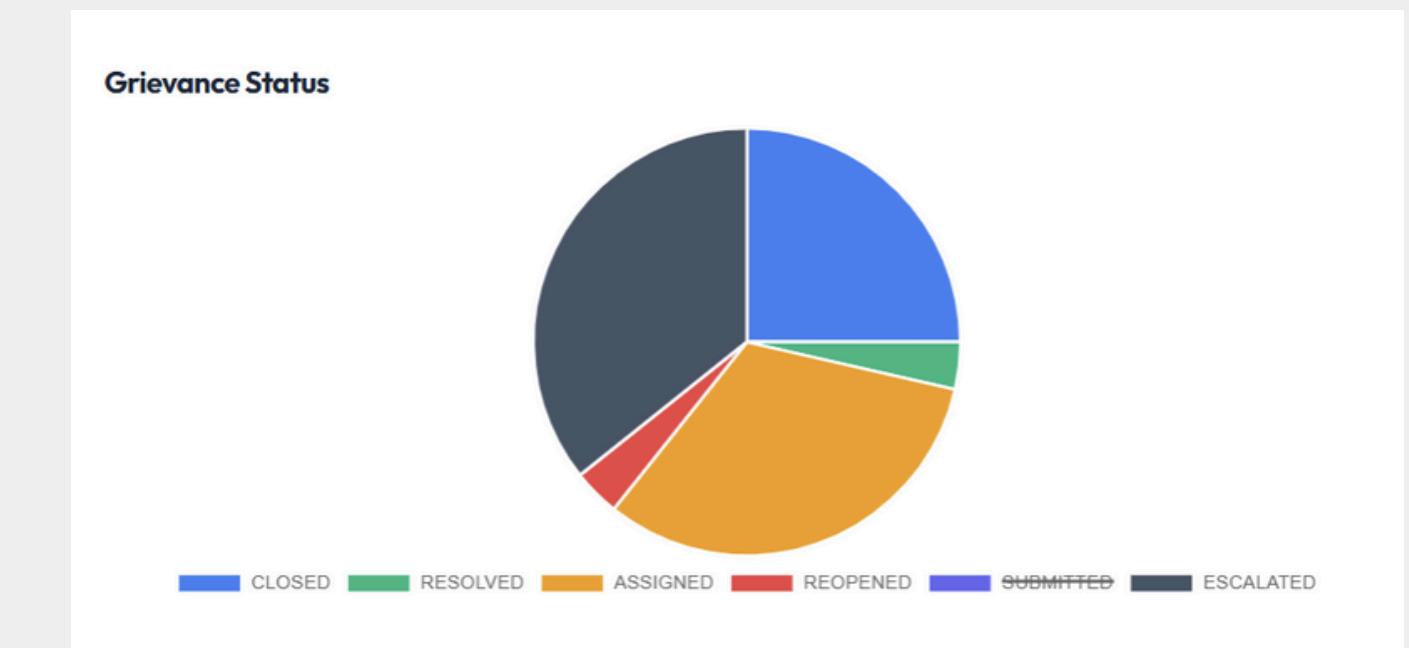
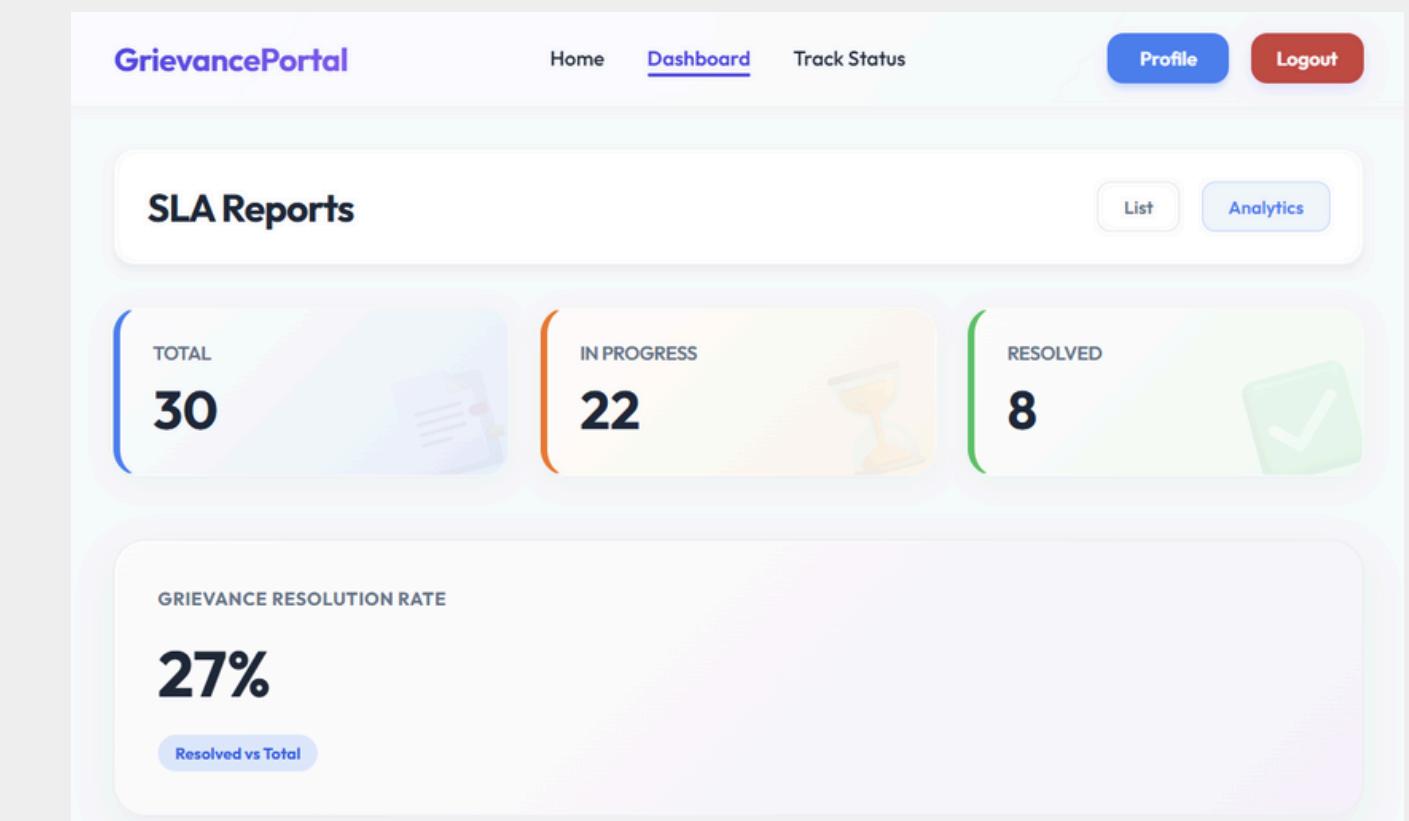
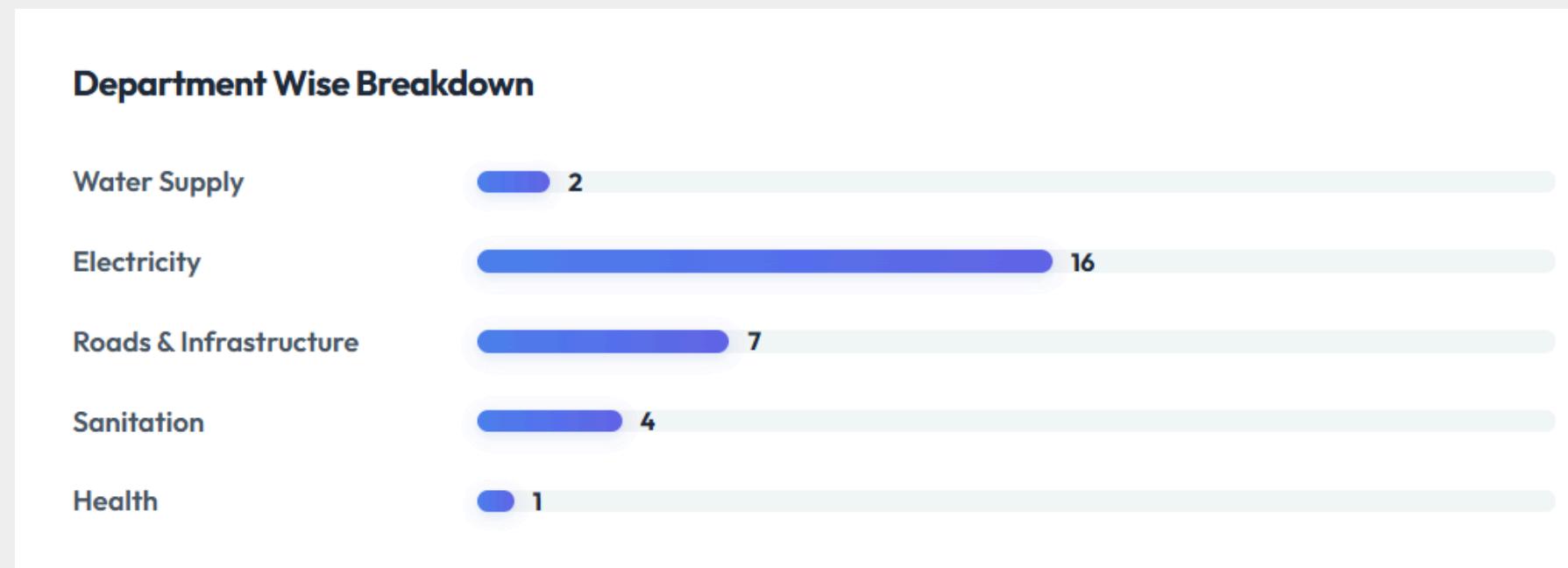
Reporting & Analytics

Users can access dashboards showing:

- grievance counts by status,
- department-wise performance,
- average resolution time, etc

Reporting service:

- is strictly read-only,
- aggregates data from other services.



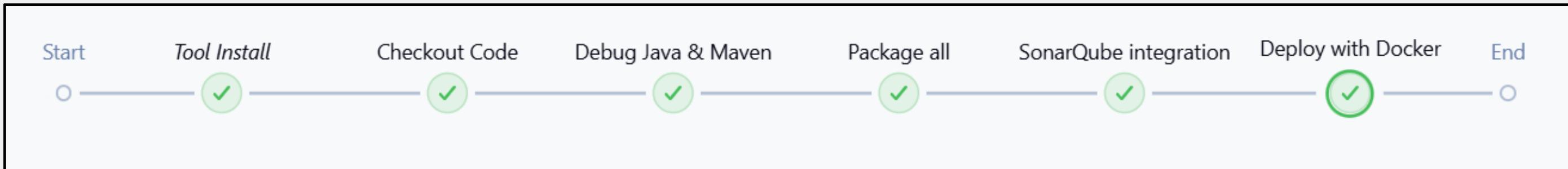
Jenkins (using SCM , Manual Build)

The screenshot shows the Jenkins console output for build #38 of the EGovGrievance job. The pipeline started by an SCM change and completed successfully. The log output includes:

```
Started by an SCM change
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\EGovGrievance
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)

[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

```
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



Docker Containers

The screenshot shows the Docker Desktop interface with the "Containers" tab selected. The main area displays a table of running containers, each with a name, container ID, image, ports, CPU usage, and actions. A tooltip for the "feedback-ser" container highlights its port mapping to `http://localhost:9006`.

	Name	Container ID	Image	Port(s)	CPU (%)	Actions
	egovgrievance	-	-	-	6.72%	3
	mongo	728f1c41ad3f	mongo:6	27018:27017	0.33%	2
	zookeeper	fe03ddfa8eb9	confluentinc	2181:2181	0.19%	1
	kafka	9bf9086f5486	confluentinc	9092:9092	0.76%	4
	config-server	aa07fbffea6a	egovgrievance	8888:8888	0.19%	4
	service-regist	c0a3b4dc1a57	egovgrievance	8761:8761	2.65%	3
	user-service	157ebae5e797	egovgrievance	9002:9002	0.11%	3
	grievance-ser	36fe2ab91548	egovgrievance	9004:9004	0.15%	3
	feedback-ser	65ac7382fcd2	egovgrievance	9006:9006	0.16%	3
	notification-s	8444045086f5	egovgrievance	9006:9006	0.4%	3
	reporting-ser	99de1281111d	egovgrievance	9007:9007	0.4%	3
	api-gateway	91ac17676178	egovgrievance	8080:8080	0.1%	3

Showing 12 items

Engine running | RAM 5.31 GB CPU 4.63% Disk: 14.07 GB used (limit 1006.85 GB) > Update available

SonarQube

The screenshot shows the SonarQube web interface for the project "EGov Grievance Redressal System". The URL in the browser is `sonarcloud.io/summary/overall?id=Yati-21_EGovGrievanceBackend&branch=main`. The interface includes a navigation bar with links for "My Projects", "My Issues", "Explore", "Upgrade", and various notifications. On the left, there's a sidebar with project details like "E Gov Grievance Redres...", "Project", "Private", and "Overview". The main content area displays the "Summary" tab, which includes sections for "New Code" (1 failed), "Overall Code", "Security", "Reliability", "Maintainability", "Accepted Issues", "Coverage", and "Duplications". The "Overall Code" section shows 0 open issues for security, reliability, and coverage, while maintainability has 30 open issues. The "Coverage" section notes 90.8% coverage with 1.1k lines to cover. The "Duplications" section shows 0.6% with 4.7k lines. A "Security Hotspots" section is also present.

sonarcloud.io/summary/overall?id=Yati-21_EGovGrievanceBackend&branch=main

SonarQube cloud

My Projects My Issues Explore

Upgrade

E Gov Grievance Redres... Project

Private

Overview

Main Branch

Pull Requests 2

Branches 1

Information

Administration >

Yati-21 > EGov Grievance Redressal System > main

Summary Issues Security Hotspots More

New Code 1 failed Overall Code

Security Reliability Maintainability

0 Open issues 0 Open issues 30 Open issues

Accepted Issues Coverage Duplications

0 90.8% 0.6%

No conditions set on 1.1k Lines to cover

No conditions set on 4.7k Lines

Security Hotspots

JacocoReports

api-gateway						
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Lines	
com.egov.apigateway.security	95%	67%	30	n/a	3	
com.egov.apigateway	37%	n/a	1	n/a	1	
com.egov.apigateway.config	100%	n/a	0	n/a	0	
Total	23 of 524	95%	32 of 98	67%	3	

user-service						
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Lines	
com.egov.user.service	90%	81%	13	57	15	151
com.egov.user	37%	n/a	1	2	2	3
com.egov.user.controller	100%	n/a	0	13	0	32
com.egov.user.exception	100%	n/a	0	10	0	23
com.egov.user.security	100%	n/a	0	5	0	18
com.egov.user.model	100%	n/a	0	1	0	5
com.egov.user.config	100%	n/a	0	4	0	4
Total	58 of 933	93%	11 of 58	81%	14	236

grievance-service						
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Lines	
com.egov.grievance.service	89%	74%	42	145	33	474
com.egov.grievance.controller	91%	n/a	2	21	2	40
com.egov.grievance.config	96%	83%	3	21	1	41
com.egov.grievance	37%	n/a	1	2	2	3
com.egov.grievance.model	100%	n/a	0	1	0	8
com.egov.grievance.exception	100%	100%	0	10	0	15
Total	214 of 2,222	90%	41 of 170	75%	48	581

feedback-service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Lines
com.egov.feedback.service	86%	75%			
com.egov.feedback	37%	n/a			
com.egov.feedback.exception	100%	n/a			
com.egov.feedback.controller	100%	n/a			
com.egov.feedback.config	100%	n/a			
Total	27 of 267	89%	2 of 8	75%	

notification-service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Lines
com.egov.notification.service	88%	n/a			
com.egov.notification.consumer	91%	64%			
com.egov.notification	37%	n/a			
com.egov.notification.config	100%	n/a			
Total	34 of 381	91%	5 of 14	64%	

reporting-service

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed Lines
com.egov.reporting.service	96%	81%			
com.egov.reporting	37%	n/a			
com.egov.reporting.exception	95%	50%			
com.egov.reporting.client	100%	50%			
com.egov.reporting.controller	100%	n/a			
com.egov.reporting.config	100%	n/a			
Total	14 of 435	96%	6 of 22	72%	

Conclusion

What the System Achieves:

- ✓ Centralized grievance tracking
- ✓ Clear ownership and accountability
- ✓ SLA-based escalation without schedulers
- ✓ Full audit trail
- ✓ Citizen transparency
- ✓ Scalable and maintainable architecture

Thank you!

