# Broker Service Microservices Analysis Document

## 1. Overview

This document provides a detailed breakdown of the Broker Service into dedicated microservices, including their responsibilities, utilized fields, data sources, caches, and views. The content is based on Swagger API definitions and internal service design patterns.

## 2. Microservices Breakdown

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| New Microservice | Key Responsibility | View/Collection | Fields | Cache Key | Remarks |
| desk-broker-service | Desk-to-Broker permission mapping | desk\_broker\_permission\_view | desk, brokerId, entitlementLevel | desk:{deskId} | Used in entitlement APIs |
| account-broker-service | Account-to-Broker mapping | account\_permission\_view | accountId, brokerId, role, status | account:{accountId} | High-volume access |
| product-entitlement-service | Broker’s entitlement per product | product\_entitlement\_view | brokerId, productCode, accessLevel | broker:{brokerId} | Used in UI dropdowns |
| role-config-service | Roles, feature access per broker | role\_feature\_mapping | role, feature, brokerId | role:{roleId} | Mostly static config |
| broker-master-service | Broker master data (name, region, type) | broker\_master | brokerId, brokerName, region, type | broker:{brokerId} | Low cardinality, infrequent updates |