# Business Requirements Final Release

# BRD — Microservice Product Search & Filter Database APIs

Document ID: ████  
████ | Date: ████  
Sponsor: ████  
BA: ████

## Executive Summary

This initiative delivers microservice APIs for product search and filtering in ████The APIs decouple data retrieval from legacy dependencies, improve performance, and enable real-time data pipelines. Business outcomes include faster product discovery, reduced reliance on third-party vendors, and streamlined system clarity.

## Business Objectives

* Streamline data retrieval via in-house database APIs.
* Reduce third-party search service dependency by ≥20%.
* Achieve ≤300 ms median query response time.
* Deliver scalable BPMN-documented data architecture.
* Enhance user experience with accurate and responsive filtering.

## Scope

In: Product search API, Filter API, data retrieval pipelines, BPMN flows, MySQL deployment.  
Out: Payment, shipping, and fulfillment services.

## Stakeholders

* Product: ███████
* Engineering: Microservices, Data Platform
* Business Analyst: Requirements and BPMN documentation
* QA: Functional and performance testing

## Business Requirements

* BR-1: Enable keyword and attribute-based product search.
* BR-2: Provide multi-facet filters (price, size, color, rating).
* BR-3: Ensure real-time inventory integration.
* BR-4: Maintain modular APIs with documented BPMN flows.
* BR-5: Reduce external API costs by shifting retrieval in-house.
* BR-6: Expose structured analytics events for search and filter usage.

## Success Metrics

* Search latency P50 ≤300 ms, P95 ≤800 ms.
* 10% increase in add-to-cart from filtered results.
* 99.95% uptime SLA.
* Zero-result rate <5%.

# Functional Specification Document — Microservice Product Search & Filter APIs

Document ID: ████  
████ | Date: ████

## Architecture

Services: search-service, filter-service, indexer-service.  
Each service with dedicated MySQL schema.  
BPMN flows model retrieval pipelines and user interaction.  
APIs exposed through REST/JSON, managed via gateway.  
Integration with Aris for process modeling.

## API Endpoints

Search: GET /v1/search?q=█████████████████████&sort=&page=  
Response includes product list, metadata, and facets.  
  
Filter: GET /v1/filters?category\_id=&selected []=  
Returns applicable filters and counts.

## Functional Requirements

* FR-1: Query parsing with typo tolerance.
* FR-2: Faceted filtering with multi-select.
* FR-3: Real-time inventory joins.
* FR-4: BPMN-documented pipelines for traceability.
* FR-5: Configurable sort orders (price, rating, relevance).
* FR-6: Ingestion pipelines from catalog to MySQL with <60s SLA.

## Non-Functional Requirements

* Latency: P50 ≤300 ms.
* Uptime: 99.95%.
* Scalability: Up to 5k products.
* Security: █████████████████████.
* Observability: Structured logs, Prometheus metrics.