Essentia

Discover Your Perfect Fragrance...

Perfume discovery is often fragmented across multiple platforms with limited personalization. Existing catalog systems focus only on inventory management, lacking community interaction or intelligent filtering. Essentia addresses this gap by offering an integrated, microservices-based web platform that combines catalog management, advanced search, and community engagement. Key challenges include ensuring data scalability, fast parallel retrieval, and secure user interactions between services for catalog, authentication, and media, helping users find their own Essentia...

Goal

Perfume discovery is often fragmented across multiple platforms with limited personalization. Existing catalog systems focus only on inventory management, lacking community interaction or intelligent filtering. Essentia addresses this gap by offering an integrated, microservices-based web platform that combines catalog management, advanced search, and community engagement. Key challenges include ensuring data scalability, fast parallel retrieval, and secure user interactions between services for catalog, authentication, and media, helping users find their own Essentia...

Proposed Solution

Essentia's architecture follows a modular microservices model designed for scalability, security, and maintainability. The solution is divided into three layers: Frontend, backend, and infrastructure interconnected through secure HTTPS REST APIs.

The UML Class Diagram models the core entities and relationships: User, Perfume, and Review. It defines how authentication, catalog management, and user-generated reviews interact through dedicated services (AuthService, PerfumeService, and ReviewService). This conceptual structure ensures logical data integrity and clear responsibility boundaries.

The Architecture Diagram illustrates the system's workflow: the frontend (React/Next.js) sends requests through an API Gateway, which routes them to backend services, a Spring Boot Auth Service and a FastAPI Catalog Service, connected to separate databases (MySQL for user data and PostgreSQL for perfume data) and Cloudinary for image storage.

Finally, the Deployment Diagram demonstrates how Essentia integrates CI/CD pipelines (GitHub Actions) with Vercel and Fly.io, enabling automated builds and independent deployment of each service.

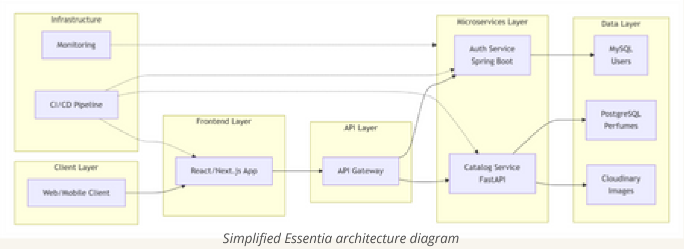
Together, these diagrams validate a technically coherent, scalable, and secure system design ready for implementation.

Experiments / Methodology

Essentia's validation was carried out through conceptual experimentation and model-driven analysis, as the project focused on the design and architecture stage.
We performed systematic evaluations to ensure the coherence, scalability, and feasibility of the proposed solution.
Each experiment simulated core user processes such as Perfume Search, User Authentication, and Admin Catalog Management to confirm logical consistency, data isolation, and service interaction across system layers.

ID	Focus	Method	Expected Result
V1	Workflow validation	UML Activity Diagram	Correct process flow
V2	Architectural structure	Architecture Diagram	Scalable microservices design
V3	System deployment	Deployment Diagram	Independent service deployment

The diagrammatic validation confirmed that Essentia's architecture is logically consistent, modular, and scalable, providing a solid foundation for future implementation and testing phases.



Simplified Essentia architecture diagram

Conclusion

Essentia's design successfully demonstrates how a microservices architecture can transform perfume discovery into a personalized, community-driven experience. Through rigorous workflow modeling and architectural validation, we have established a scalable foundation that supports efficient catalog management, secure user interactions, and future-enhanced capabilities, proving that thoughtful design paves the way for impactful digital solutions in niche fragrance markets.