



۱۳۰۷

دانشگاه صنعتی خواجه نصیرالدین طوسی

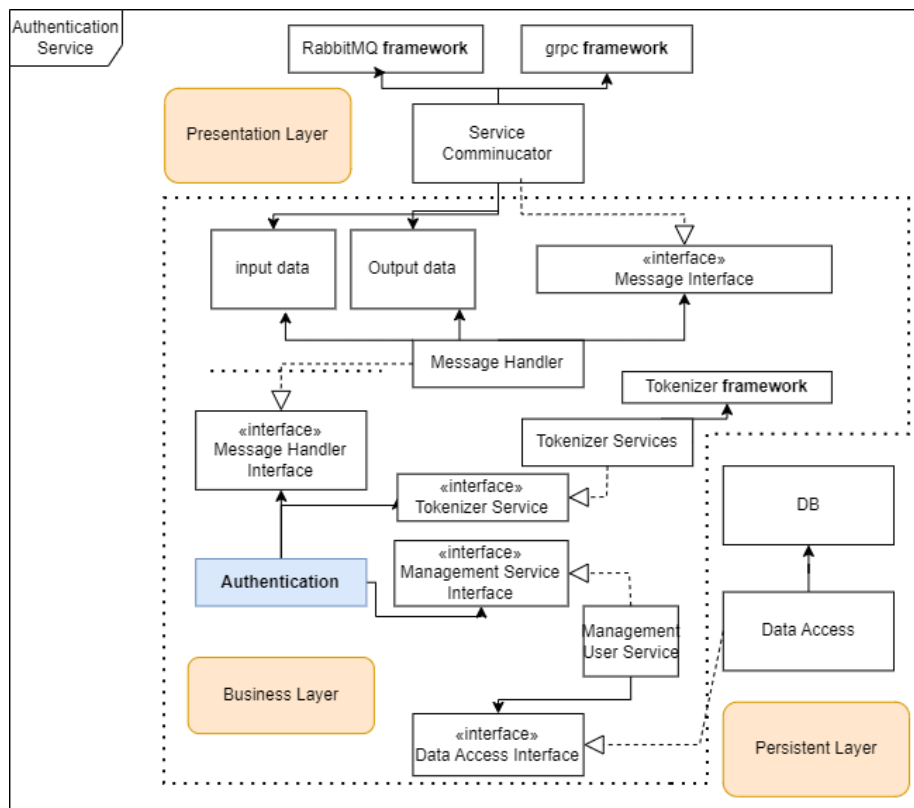
معماری نرم افزار - سرویس Authentication

سید حسین زراعتکار و نیما گمرکیان
دانشگاه خواجه نصیرالدین طوسی

۱۳ آذر ۱۴۰۴

۱.۰ میکروسرویس Authentication - Module View

در این بخش به بررسی معماری میکروسرویس Authentication خواهیم پرداخت. این میکروسرویس وظیفه احراز هویت کاربران و ادمین‌ها را به عهده دارد. پس از اهراز هویت به هر کاربر یک توکن داده خواهد شد که دارای زمان انقضا هست. هر سرویس دیگر متناسب با توکن‌های کاربران له آنها سرویس می‌دهد. در ادامه ماژول‌ها و وابستگی‌شان را در قالب دیاگرام در شکل ۱ ارائه کردیم.



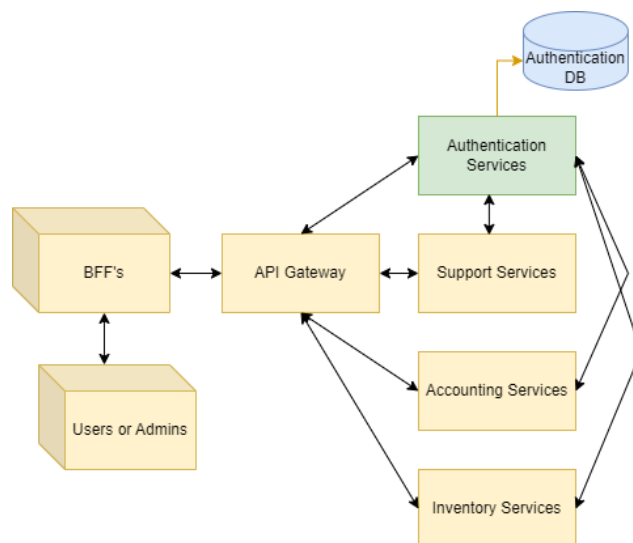
شکل ۱: دیاگرام وابستگی ماژول‌ها

این ماژول از طریق دو framework با نام‌های **grpc** و **RabbitMQ** با دیگر سرویس‌ها و **bff** تعامل می‌کند. برای تعاملات بین سرویس به دلیل سرعت بالاتر از **grpc** و برای تعاملات با **bff** از **RabbitMQ** استفاده کردیم. برای پیاده‌سازی متدها و مدیریت پیام‌های این دو سرویس یک ماژول پیاده‌سازی کردیم که بتواند کارهایی که فریم‌ورک قرار است برای ما انجام دهد را مدیریت کند که این ماژول باید **interface** مربوط به **message** را پیاده‌سازی کند تا **business logic** ما بتواند بدون وابستگی به ماژول **low level** با بقیه ارتباط برقرار کند. در قسمت **business logic** ما یک ماژول به منظور هندل کردن رخدادهای دریافتی یا ارسال پیام‌ها پیاده‌سازی شده است به نام **Message Handler** این ماژول نیز به علت سطح پایین بودن باید برای مورد استفاده قرار گرفتن توسط **Authentication** که بالاترین سطح را دارد با استفاده از معکوس‌سازی وابستگی و استفاده از **interface** پیاده‌سازی شود. از طرفی دیگر ما برای افزودن کاربران (ادمین و کاربر معمولی) ماژولی به نام **Man-agement User Service** داریم که عملیات **verification**، **registration**، **update** را انجام می‌دهد. این ماژول نیز به دلیل نیاز به استفاده از دیتابیس با **interface** و معکوس‌سازی وابستگی با دیتابیس مورد نظر کار می‌کند. در آخر ماژول **Tokenizer service** با استفاده از کتابخانه‌ها و **framework** ها توکن‌ها را ایجاد کرده و آپدیت و انقضای توکن‌ها از این طریق مدیریت می‌شوند. برای انقضای توکن در واسط مربوط به **Tokenizer** ما یک متد برای ایونت انقضا داشته و در طرف ماژول **mangement** یک هندلر تابع و در **authentication** این‌ها را مدیریت می‌کنیم. یکی از نکات مهم این است که برای تعاملات با سایر سرویس‌ها ما باید از معماری تراکنش **saga**

استفاده کنیم تا قبل از اعمال تراکنش احراز هویت بر اساس توکن کاربر بررسی شود.

۲.۰ میکروسرویس Authentication - Component & Connector View (C&C)

در این قسمت به بررسی نمای دیگر از معماری سرویس Authentication به نام Connection And Component View می‌پردازیم. نمای کلی ارتباطات این سرویس با سرویس‌های دیگر و BFF در شکل ۲ ارائه شده است. بر اساس نیاز موجود هر کاربر با هر سطح دسترسی یا درخواست احراز هویت اولیه می‌دهد که در آن صورت مسیر از



شکل ۲: نمای C&C

کاربر به BFF، API Gateway و در آخر به سرویس احراز هویت می‌رسد و اما در صورت درخواست سرویس دیگر مرحله آخر به سرویس مورد نظر رسیده و سرویس مورد نظر درخواست تایید برای توکن را به سرویس احراز هویت ارسال کرده و بر اساس تایید یا رد پاسخ کار را ادامه می‌دهند. پاسخ نیز از سرویس مورد نظر به کاربر برگشت داده می‌شود و اگر پاسخ سرویس احراز هویت باشد، مقدار توکنی را برمی‌گرداند.