

Allegato 6: La gestione degli utenti nell'architettura COBRA

Introduzione

Lo sviluppo secondo l'architettura a Microservizi, proposta inizialmente da Martin Fowler, ha l'obiettivo principale di suddividere un sistema in piccoli blocchi costituenti con lo scopo di rispondere a specifiche esigenze che sono qui sottoelencate:

- 1) Resilienza del sistema alla caduta temporanea di uno o più servizi che lo compongono
- 2) Rapidità di risposta dei servizi mediante la distribuzione e la replica degli stessi su differenti
 1. hardware
- 3) Migliore adattabilità alle tecnologie più efficaci per la risposta a specifiche esigenze
- 4) Facilità di evoluzione del sistema mediante sostituzione e/o integrazione di nuovi servizi nel
 2. sistema
- 5) Sviluppo parallelo delle componenti del sistema

La struttura di una architettura a Microservizi prevede la realizzazione di una pletora di servizi che possano essere facilmente installati e distribuiti pur restando fortemente disaccoppiati. Questi servizi devono essere il più possibile autonomi all'interno del proprio dominio applicativo e non hanno dipendenze dirette con altri servizi grazie ad una comunicazione basata su messaggi asincroni veicolati per mezzo di una coda invece che su chiamate incrociate alle rispettive API.

Schema logico

Lo schema sottostante descrive la visione di insieme dell'architettura tecnica della piattaforma COBRA. La soluzione così disegnata non è direttamente dipendente con la struttura hardware sulla quale risulta ospitata. Infatti, l'intera architettura può trovare collocazione in una singola macchina oppure più propriamente su molteplici hardware dedicati garantendo così una migliore tolleranza al fault e una migliore prestazione di ciascuna parte. Alcuni elementi chiave del sistema possono essere oggetto di ridondanza e clustering allo scopo di evitare situazioni di singlepoint-of-failure su elementi chiave del sistema che siano vitali per la comunicazione dei componenti e quindi potenzialmente cruciali per la sua resilienza.

Token Issuer Authority

Il "Token Issuer" è il servizio principale dell'architettura del sistema che si occupa di centralizzare l'identificazione degli utenti dei vari servizi e di conseguenza funge da single-sign-on rispetto alle applicazioniche insistono su di essi. Esso è in grado di registrare le utenze, identificarle per mezzo di una login e in seguito emettere un token in standard JWT riconoscibile dalle varie componenti che ne possono desumere identità e permessi dell'utente cui esso si riferisce. Il token issuer è in grado anche di gestire il refresh di token scaduti e di interrompere la validità di un token in anticipo rispetto alla scadenza.

Authentication Web UI

L'applicazione in questione è l'interfaccia utente del token issuer e si occupa delle fasi di identificazione dell'utente per mezzo di username e password. Esso opera secondo uno standard OAUTH2.0 mediante Implicit o explicit grant. Le applicazioni che intendono usufruire dell'autenticazione dovranno indirizzare i propri utenti all'interfaccia web di questa applicazione indicando un url di redirect al quale sarà notificato il token JWT emesso in seguito alla login.

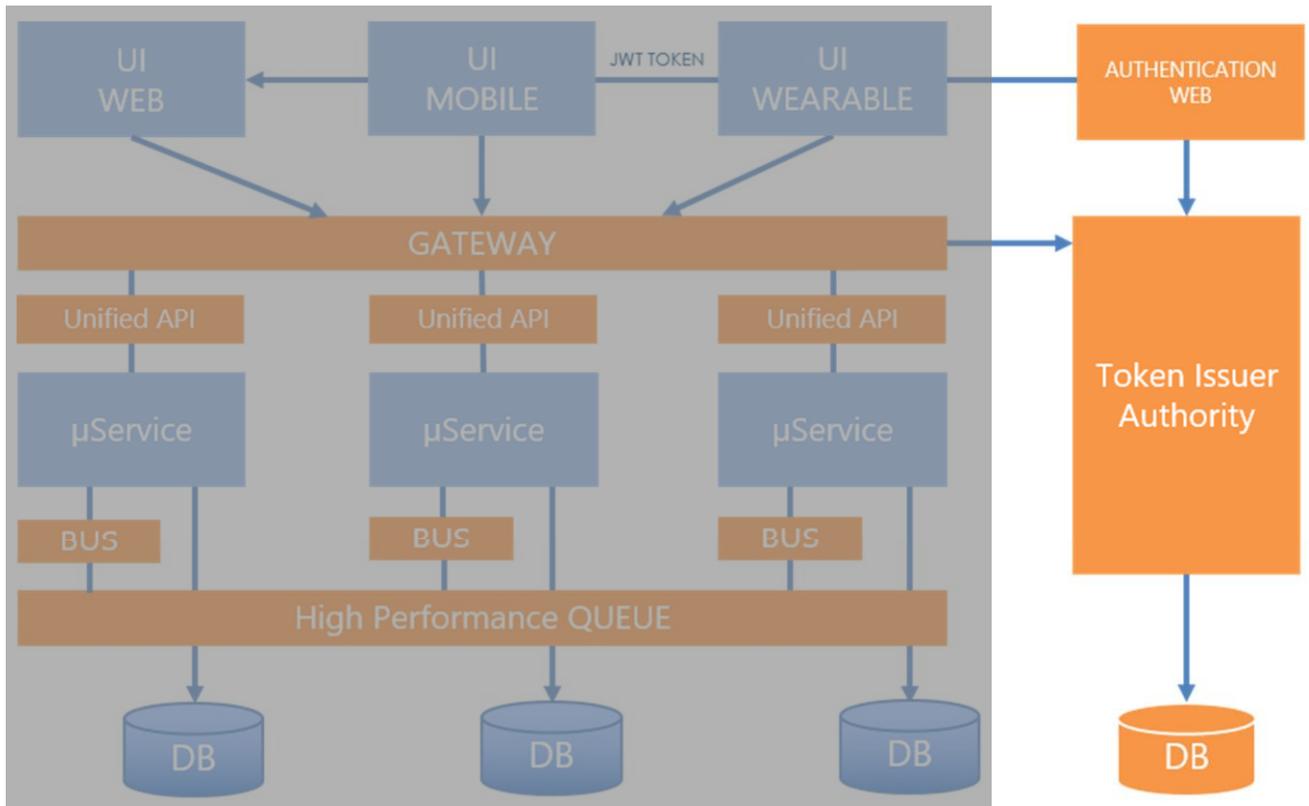


Figura 1: Gestione degli utenti nell'architettura COBRA