Tema1 DATC

On-Premise vs Cloud Software

În funcție de cerințele și nevoile unei organizații, se poate alege între sistemul privat: **On-Premise** -instalat local pe calculatoarele și serverele propriei companii- și **serviciile Cloud** -găzduite pe serverele vânzătorului și accesate prin browser-ul web-: Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS). Cele două platforme conțin atât dezavantaje, cât și avantaje:

- Din punct de vedere al securității, On-Premise poate fi mai sigur, datele fiind stocate în interiorul organizației, iar din punct de vedere al costului, acesta este mult mai scump în compație cu unul din serviciile Cloud; costurile de mentenanță sunt și ele foarte mari;
- Sistemele Cloud oferă un timp de funcționare mai rapid și mai puține întreruperi ale serviciului, precum și flexibilitate de expansiune mai mare decât ON-Premise – datele din sistemele Cloud pot fi accesate din orice browser din întreaga lume;
- Aplicațiile Cloud au nevoie în permanență de un semnal puternic și stabil de internet, iar acest lucru poate creea probleme în diverse locuri ale lumii;
- Utilizând On-Premise, există în permanență control asupra upgrade-urilor.

HTTP Protocol (Hyper Text Transfer Protocol) este un element de bază pentru realizarea aplicațiilor de tip Cloud și permite comunicarea între client și server. Principalele piese ale comunicării HTTP sunt: **URLs**, **verbs**, **status codes.**
Comunicarea se realizează printr-o pereche de mesaje request/response. Mesajul de **request** este transmis spre gazda printr-un **URLs**, protocol http și conține **verbs** - acțiuni pe care clientul ar dori ca serverul să le realizeze(ex: GET, POST, PUT, DELETE, etc). Mesajul de **response** conținte: **status codes** care informează clientul despre cum poate interpreta răspunsul dat de server(coduri între 1xx și 5xx) și **message body.**
Headerele pot fi: **General Headers** și **Entity Headers/Message Body** care furnizează informații despre conținutul mesajului. Pentru a urmării traficul și comunicarea HTTP, se poate utiliza **Fiddler** pentru Windows sau **Charles Proxy** pentru OSX.

JSON Web Tokens(JWT) este un standard care definește un mod sigur de transmitere a informațiilor între două părți. JWT este sigur deoarece este semnat utilizând o cheie secretă(algoritmul HMAC) sau o cheie publică/privată(RSA sau ECDSA). JWT este folositor pentru: **Autorizare** – fiecare cerere realizată de un user include JWT pentru a permite user-ului să acceseze rutele, serviciile și resursele care sunt permise cu acest token;
Schimbarea informațiilor – JWT permite o siguranță în transmiterea datelor - siguranța că ceea ce s-a transmis, a fost și primit. Structura JWT este realizată de trei părți separate prin punct:

- **Header**- constă în două părți: tipul token-ului și algortimul de *hashing*(ex. RSA);
- **Payload**- conține **claims(registered claims, public claims și private claims)** – sunt declarații despre o entitate(de obicei, utilizatorul) si date adiționale;
- **Signature**-pentru semnătură este necesar header-ul și payload-ul codificate dar și algoritmul specificat în header.