

## Serverless computing. Azure Functions App

### TEMA 2 – DATC

Pentru a înțelege cât mai bine despre ce este vorba în acest document, putem începe cu cele mai simple întrebări, cum ar fi:

Ce este un *serverless computing*?

Serverless computing (sau pe scurt serverless), este un termen impropriu, deoarece nu poți avea o aplicație *serverless* fără a fi până la urmă găzduită pe un server. Martin Fowler, definește arhitectura *serverless* ca fiind :

[..] aplicații care depind în mod deosebit de servicii oferite de furnizori( supranumite si *Backed as a service* sau “*BaaS*”) sau depend de un cod sursă care este executat în containere temporare(*Function as a service* sau “*FaaS*”)

Se pot identifica două tipuri de arhitecturi serverless:

*BaaS*- aceasta a fost prima aplicare a termenului de arhitectura *serverless* ,înțelegând aplicații complexe care se conectează la baze de date accesibile prin internet, precum Prase,Auth0.

*FaaS*- Execuția logicii aplicației, fără a avea grija configurării sau administrării unor servere de gazduire sau serverelor de aplicații, pe containere *stateless* și invocă pe bază de evenimente. Amazon oferă servicii FaaS prin AWS Lambada, Microsoft prin Azure Functions prin Cloud Functions.

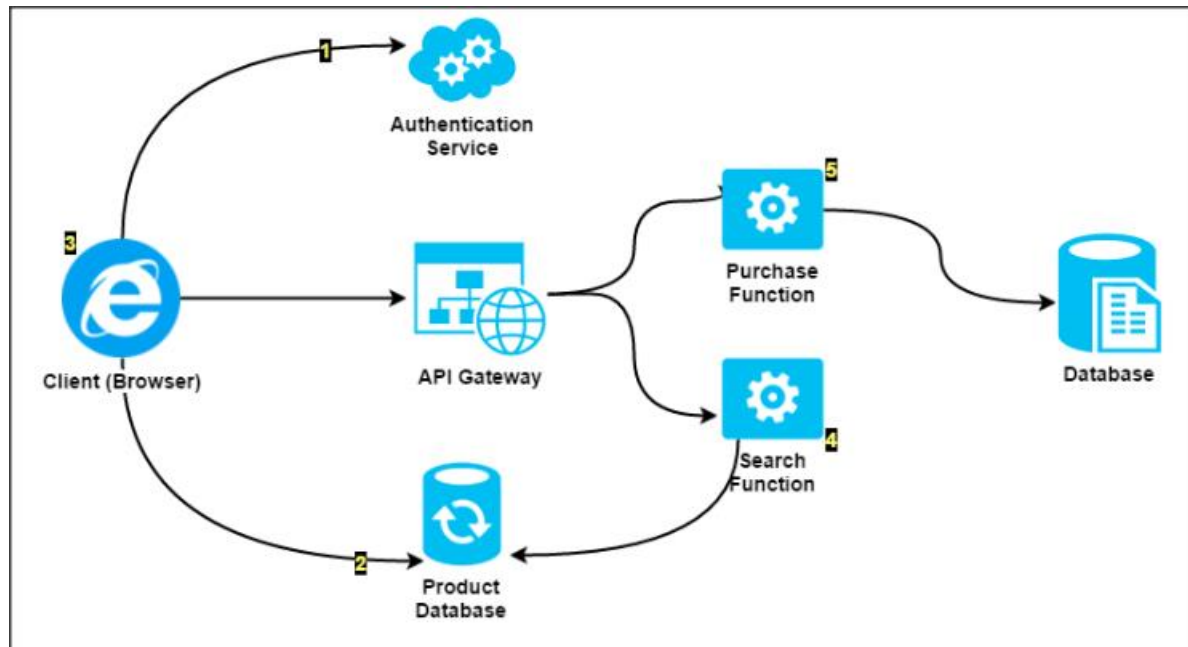
Cum se diferențiază o *arhitectură tradițională* față de *arhitectura serverless*?

O aplicație client tradițională bazată pe trei nivele, arată de obicei în felul următor:



În inaginea de mai sus observăm o aplicație web tipică.

O aplicație *serverless* ar arăta în felul următor:



Având această privire de ansamblu asupra arhitecturii *serverless*, putem să luăm ca exemplu practic, *Azure Functions*, serviciul FaaS oferit de Microsoft și să vedem cât de rapid poate fi folosit.

### *Azure Functions*

*Azure Functions* este o soluție care permite într-un mod foarte facil executarea unor bucăți de cod de mici dimensiuni ("funcții") în internet. Poți scrie codul care rezolvă o anumită problemă de business, fără a te îngrijora de detaliile legate de infrastructura pe care se va executa. Codul sursă poate fi scris într-o varietate de limbaje de programare, cum ar fi: C#, Python, PHP, etc.

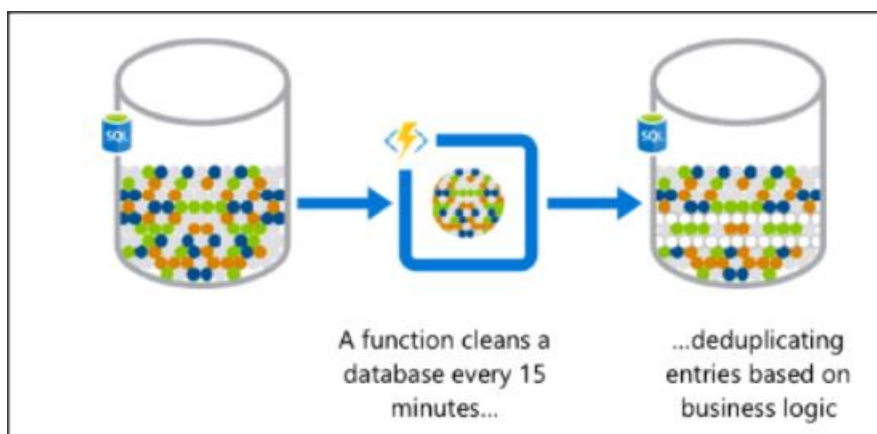
Acestea ar fi principalele beneficii ale *Azure Functions*:

- *Selecția de limbaje de programare* - poți scrie funcții folosind: C#, batch, PHP, Python sau chiar să execuți aproape orice tip de executabil.
- *Plătești cât consumi* - plătești doar pentru timpul folosit pentru executarea codului.
- *Gestionarea dependențelor* - Poți folosi *nuget* sau *npm* pentru a reutiliza librăriile favorite.
- *Securitate integrată* - Funcțiile expuse prin HTTP pot fi securizate folosind furnizorii OAuth cum ar fi Azure Active Directory, Facebook, Google, Twitter sau folosind contul Microsoft.

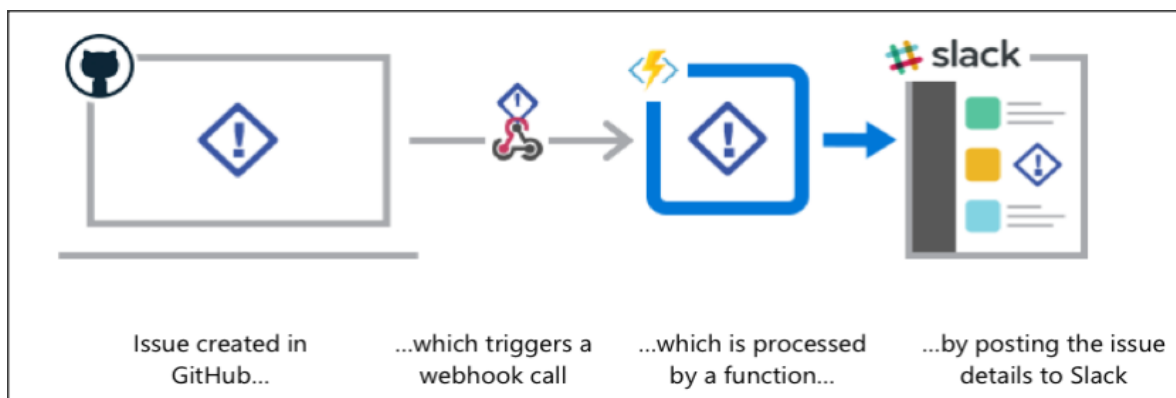
- *Integrare simplificată* - Este posibilă integrarea cu celelalte servicii oferite de Microsoft în Azure sau de alți furnizori de servicii SaaS.
- *Dezvoltare flexibilă* - codul sursă poate fi editat direct în portal sau poate să fie rezultatul unui *pipeline* de integrare continuă având la bază un cod sursă găzduit în GitHub sau Visual Studio Team Services.
- *Codul sursă este disponibil oricui* - codul sursă al mediul în care este executat Azure Functions este disponibil pe GitHub.

### Ce putem face cu Azure Functions

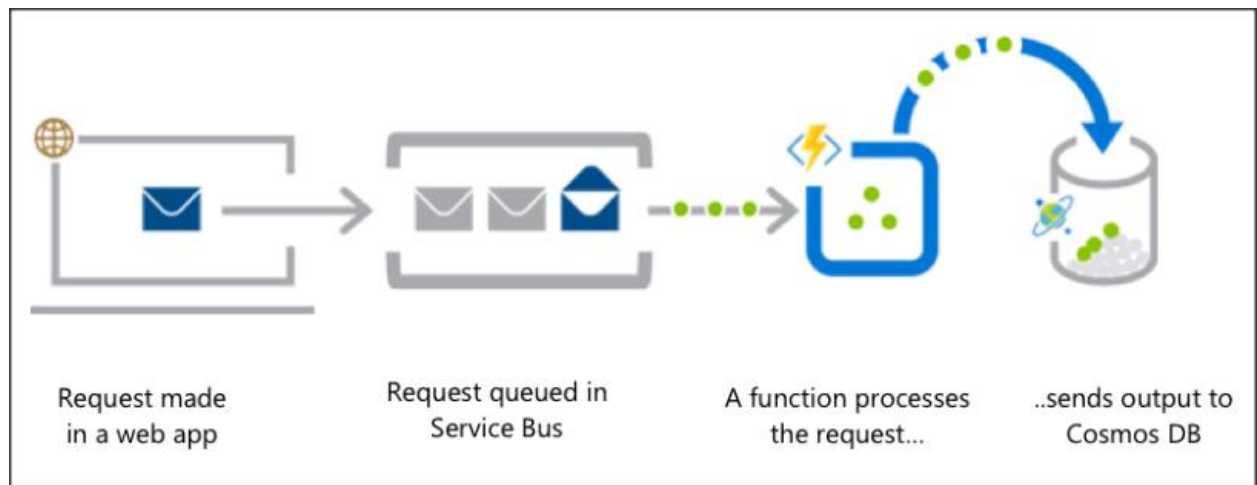
1. *Procesare dependentă de timp*- executarea unei funcții de *ștergere* sau procesarea de task-uri după un anumit orar.



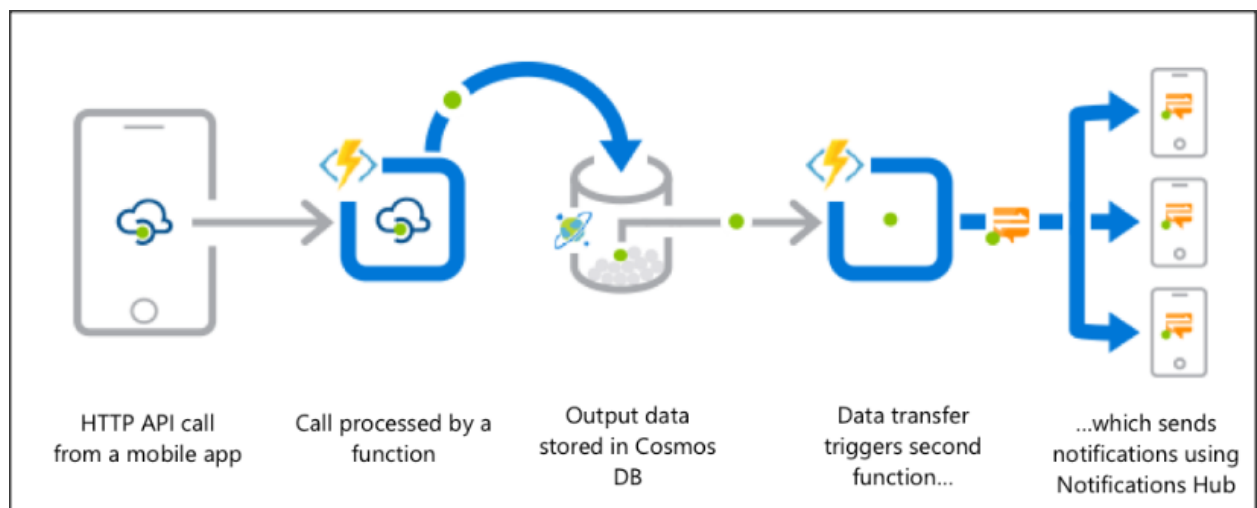
2. *Procesarea evenimentelor provenite din SaaSuri* - Azure Functions oferă posibilitatea integrării mecanismelor de declanșare bazate pe evenimente ce au loc într-o alta platformă de tip SaaS, cum ar fi un nou fișier salvat în OneDrive ca declanșator al funcției.



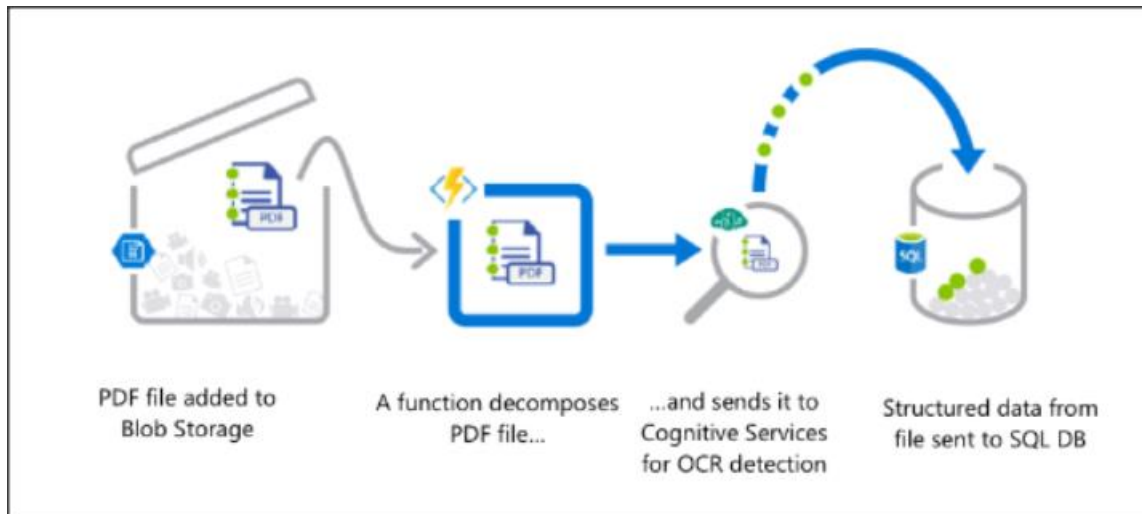
3. *Aplicații web bazate pe arhitectura serverless* - Azure Functions poate fi motorul unei aplicații web de tip *single page application* (SPA).



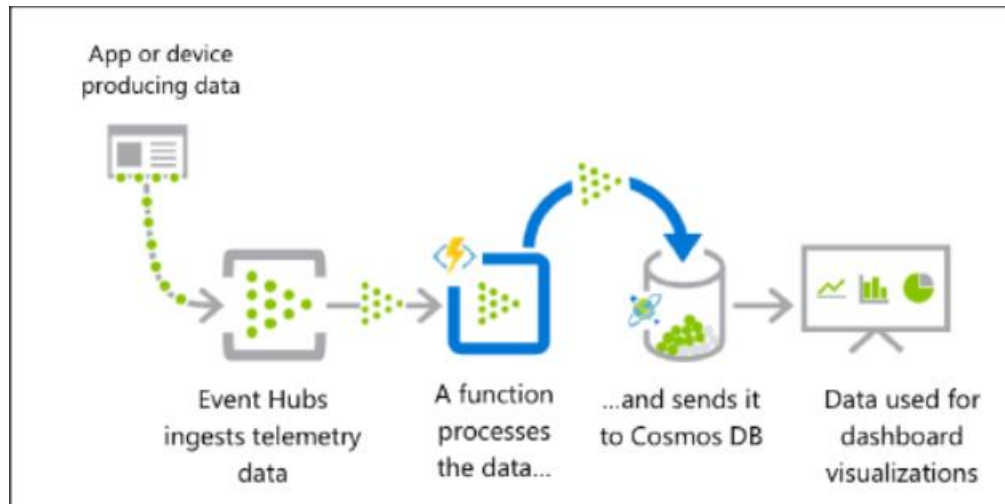
4. *Backend pentru aplicații mobile* - o aplicație pe un dispozitiv mobil ar putea captura o imagine care mai apoi ar putea apela o funcție Azure pentru a putea încărca imaginea într-un *blob storage*. O a doua funcție Azure ar putea fi declanșată de acea încărcare așa încât să redimensioneze imaginea la valori care să fie potrivite cu dispozitivele mobile.



5. *Procesare în timp real* - de exemplu, dispozitive IoT ar putea transmite mesaje către Azure Stream Analytics, care mai apoi să apeleze o funcție Azure, pentru a procesa mesajul respectiv.



6. *Mesagerie în timp real realizată programatic* - Integrarea Bot Framework cu o funcție Azure pentru a procesa un mesaj provenit de la Cortana Analytics.



În concluzie, pentru a mai scurta din numărul de pagini, am omis de a mai scrie despre cum creez o funcție în Azure sau ce tehnologii stă la baza Azure Functions. Am încercat să mă rezum la informațiile care au fost mai importante.