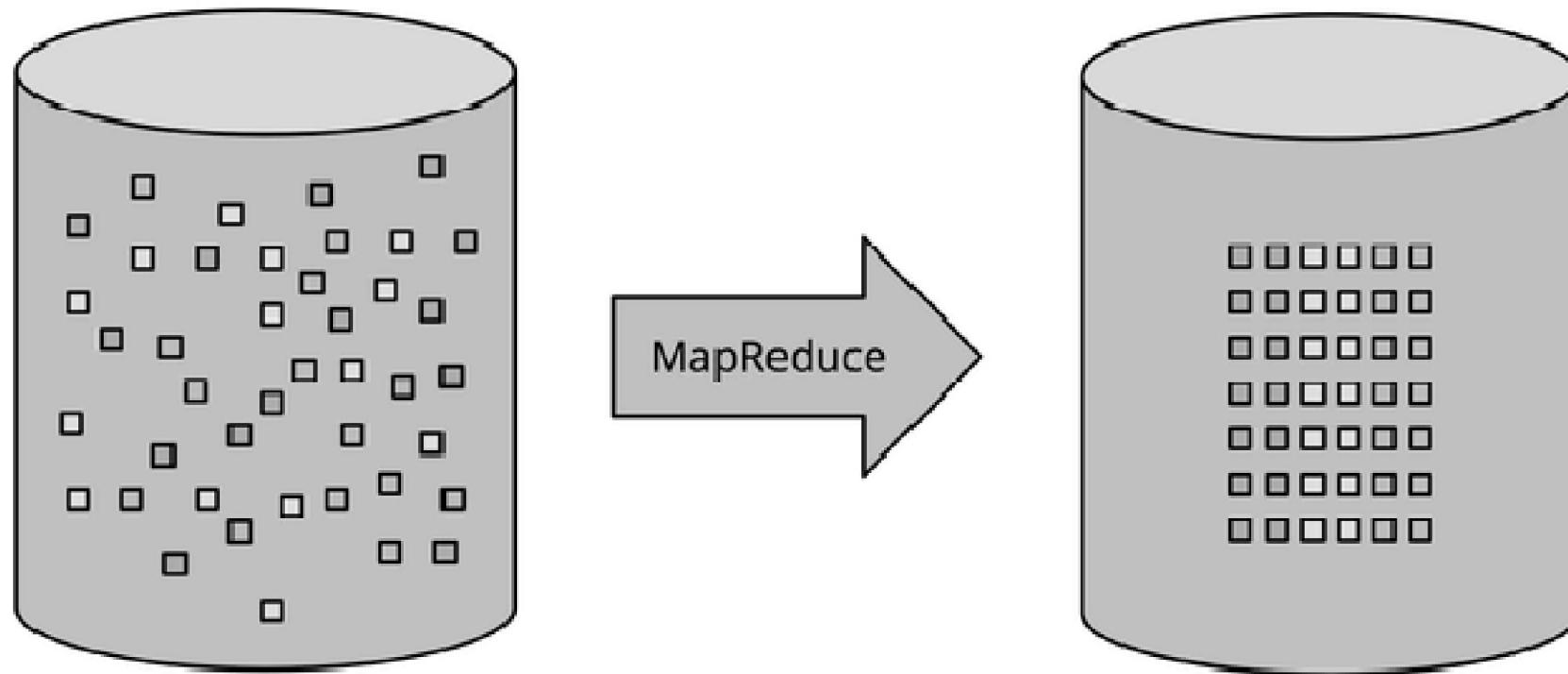


Sisteme Distribuite

Cursul 11

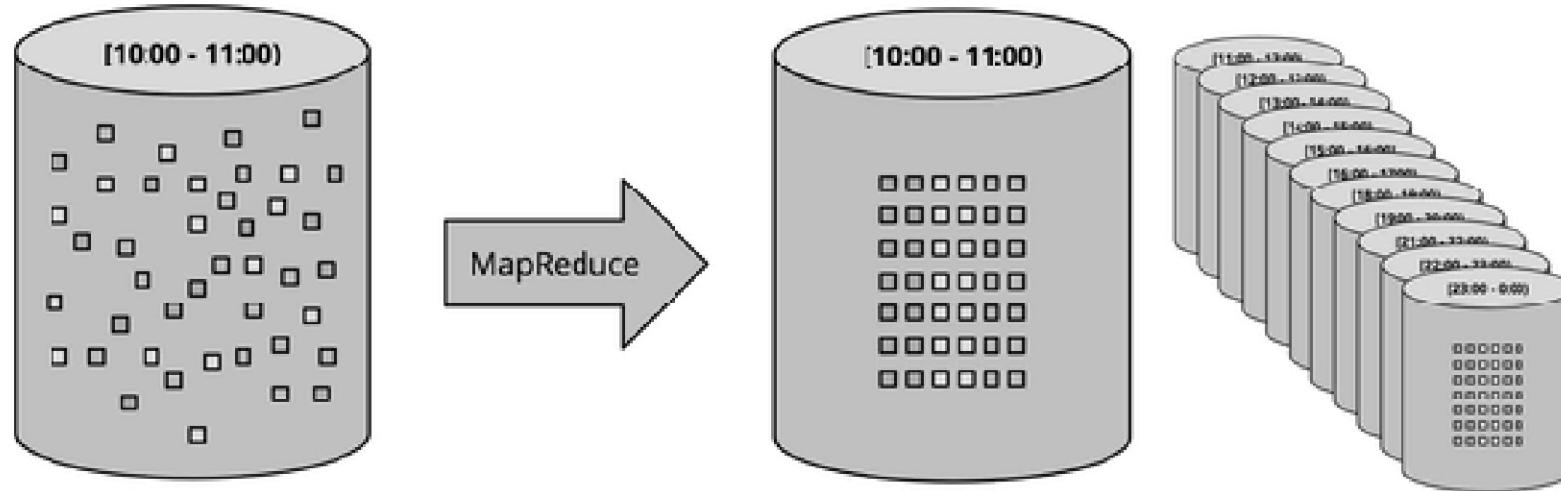
Mihai Zaharia

Modele de procesare a datelor



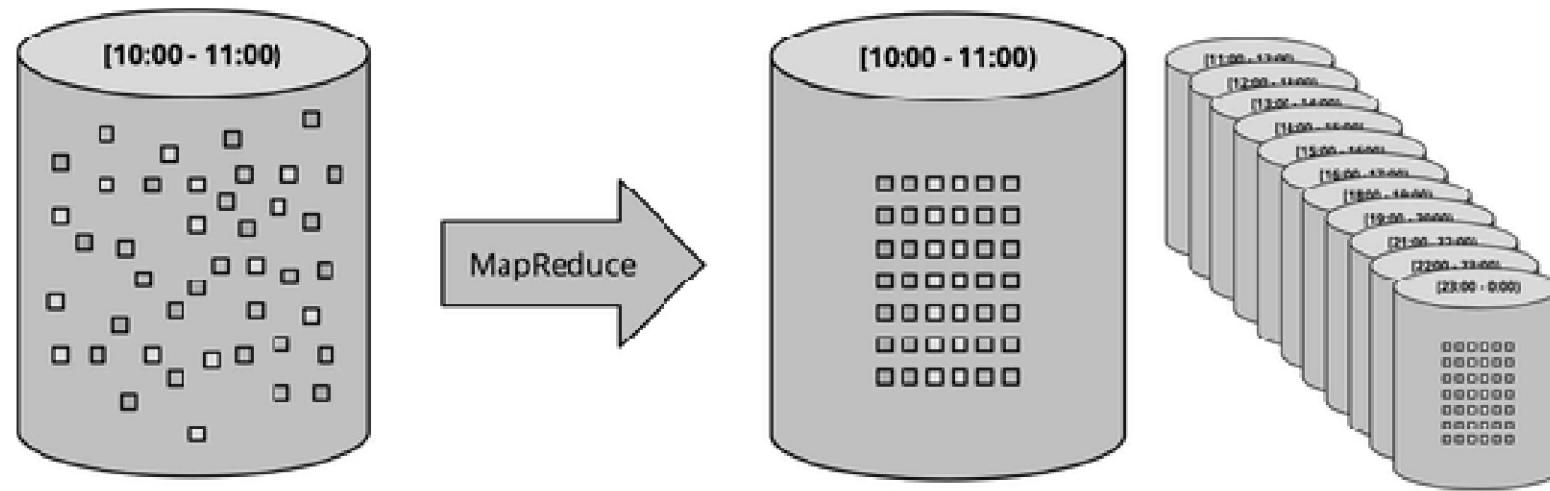
Seturi finite de date

Modele de procesare a datelor



fluxuri continue - procesare pe segmente/ferestre

Modele de procesare a datelor

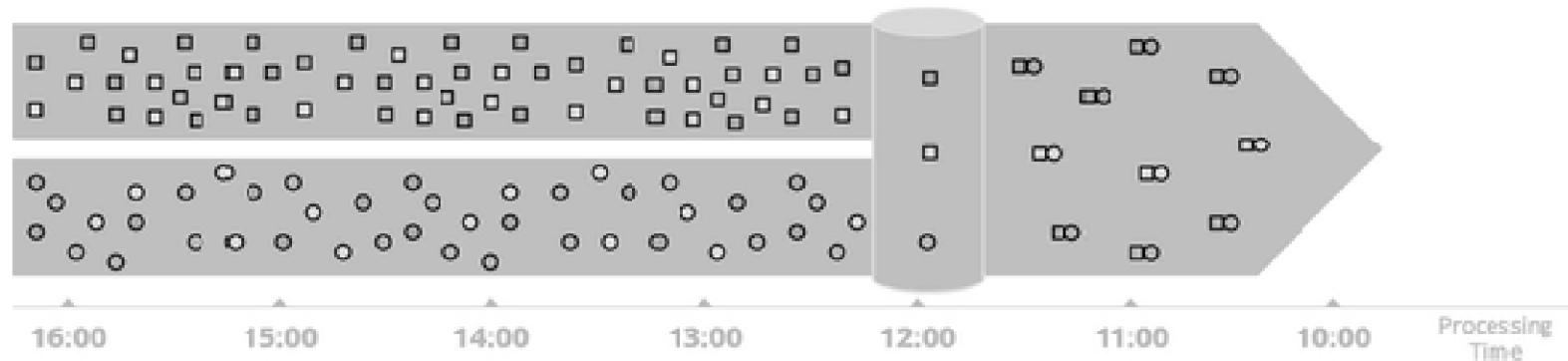


fluxuri continue - procesare pe sesiuni

Modele de procesare a datelor

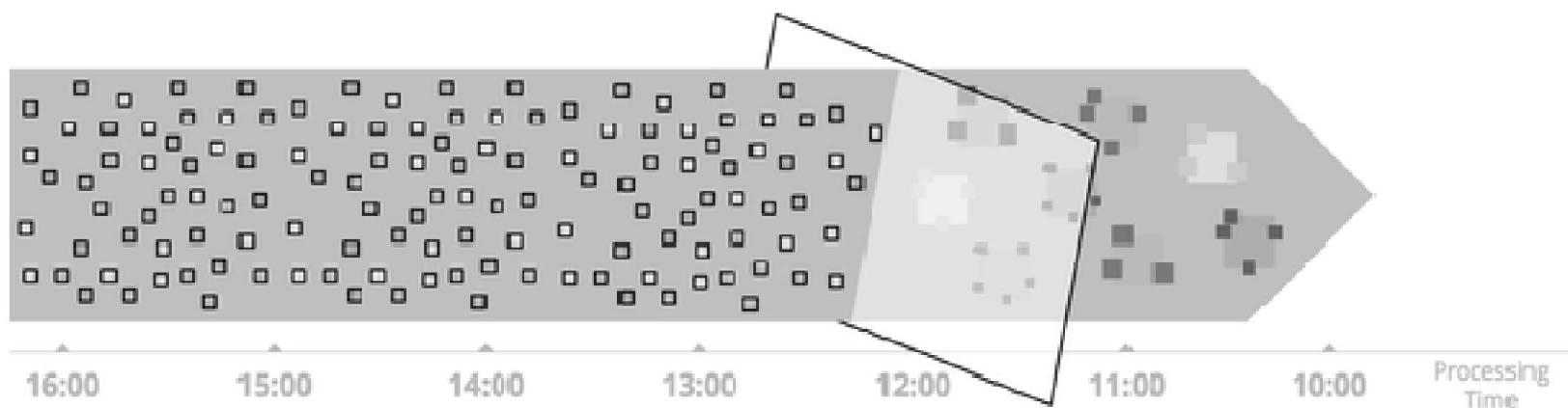
- Fluxuri de date continue:
 - time-agnostic,
 - approximation,
 - windowing by processing time,
 - windowing by event time.

Modele de procesare a datelor



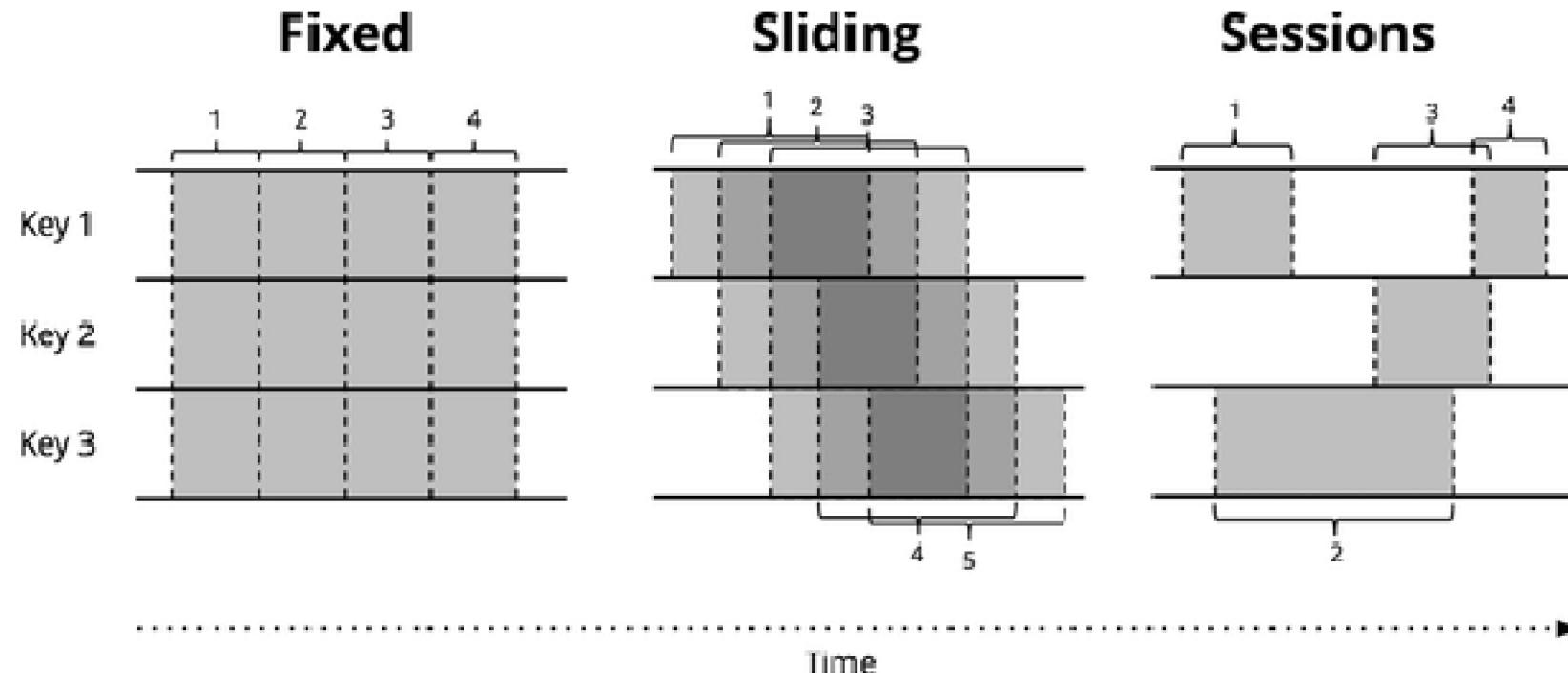
indiferent de timp - filtrarea

Modele de procesare a datelor



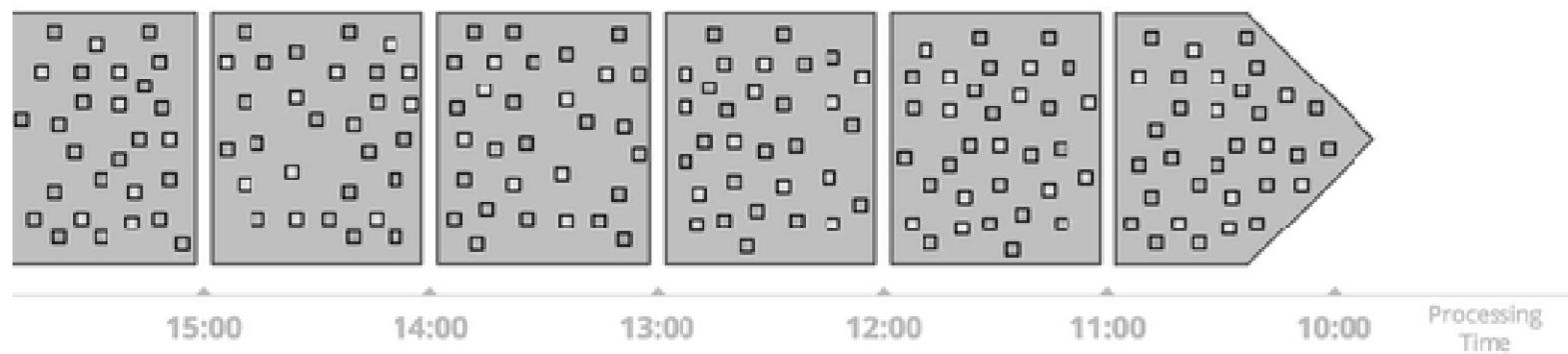
indiferent de timp - aproximarea

Modele de procesare a datelor



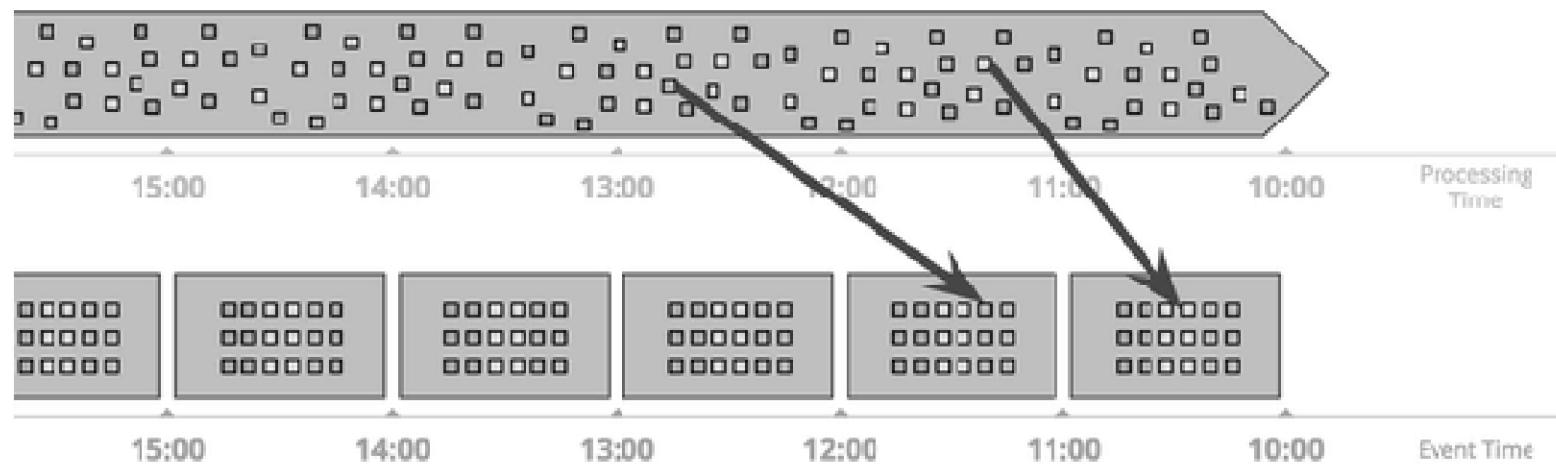
indiferent de timp - ferestre

Modele de procesare a datelor



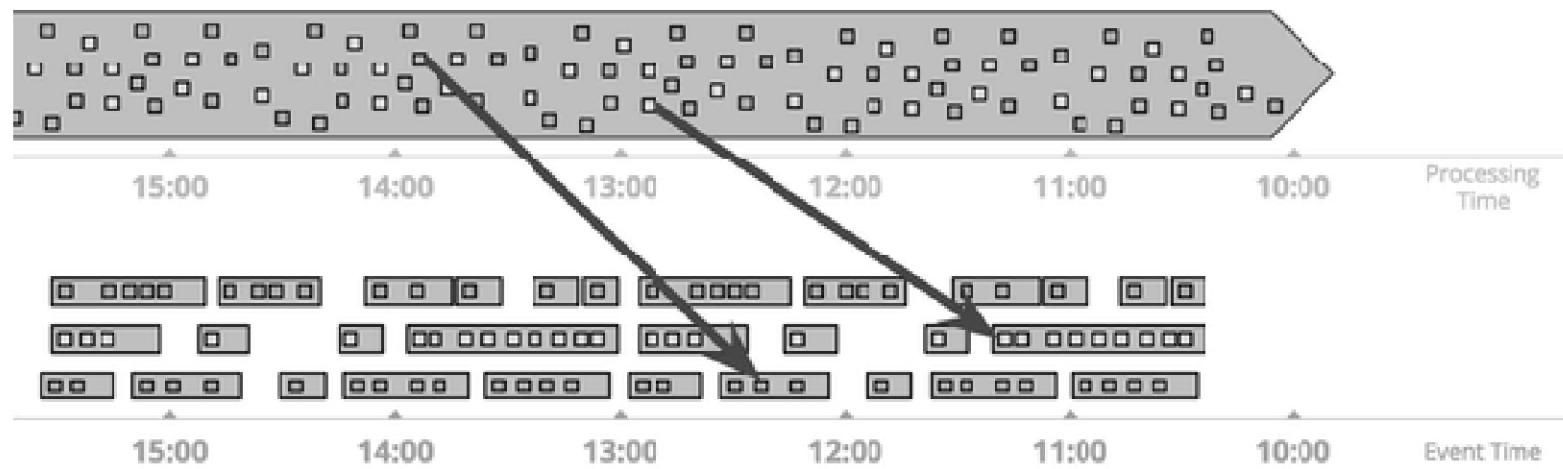
ferestre cu partitionare în funcție de timpul de procesare

Modele de procesare a datelor



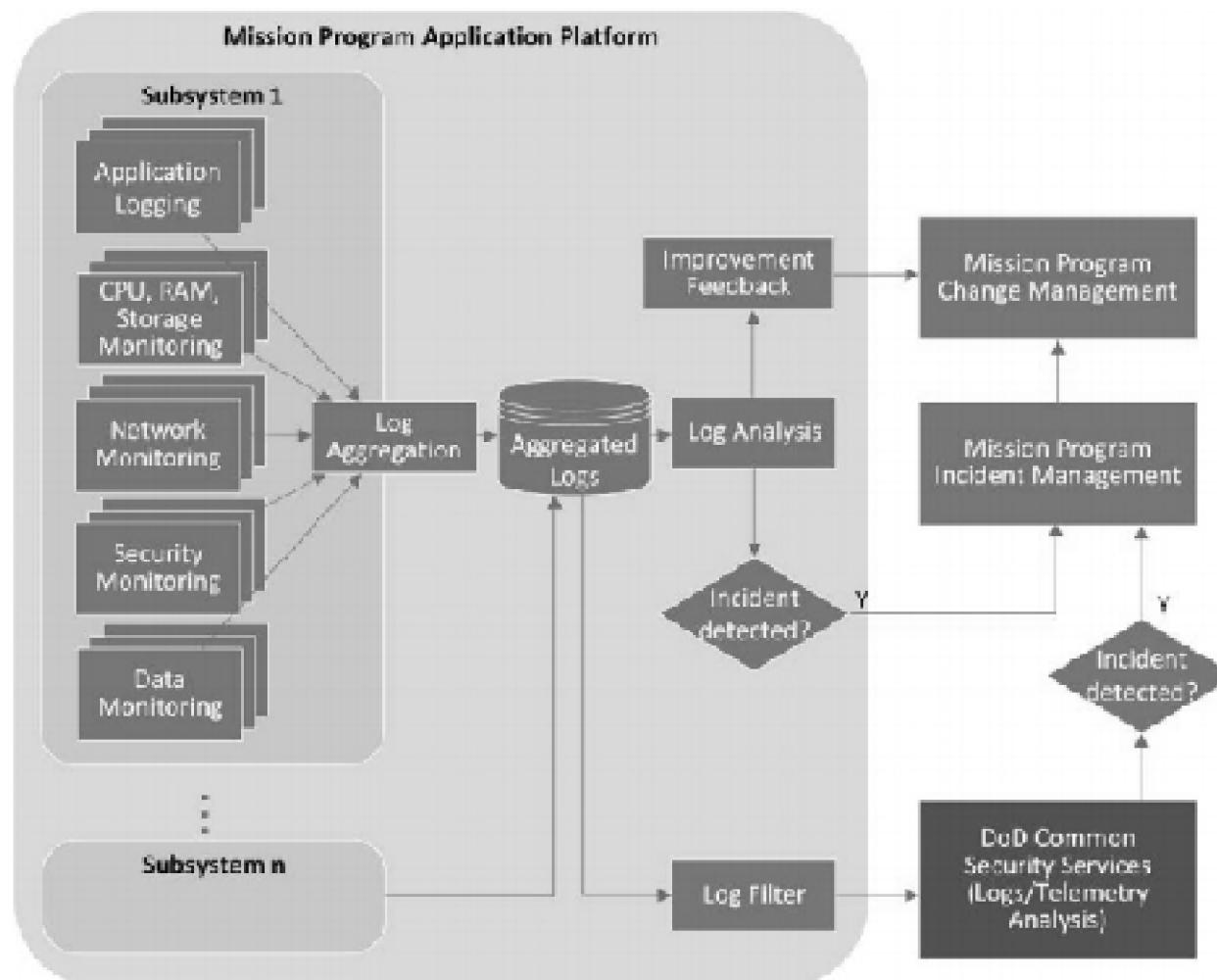
Ferestre - partitionare după timpul de apariție a evenimentelor

Modele de procesare a datelor



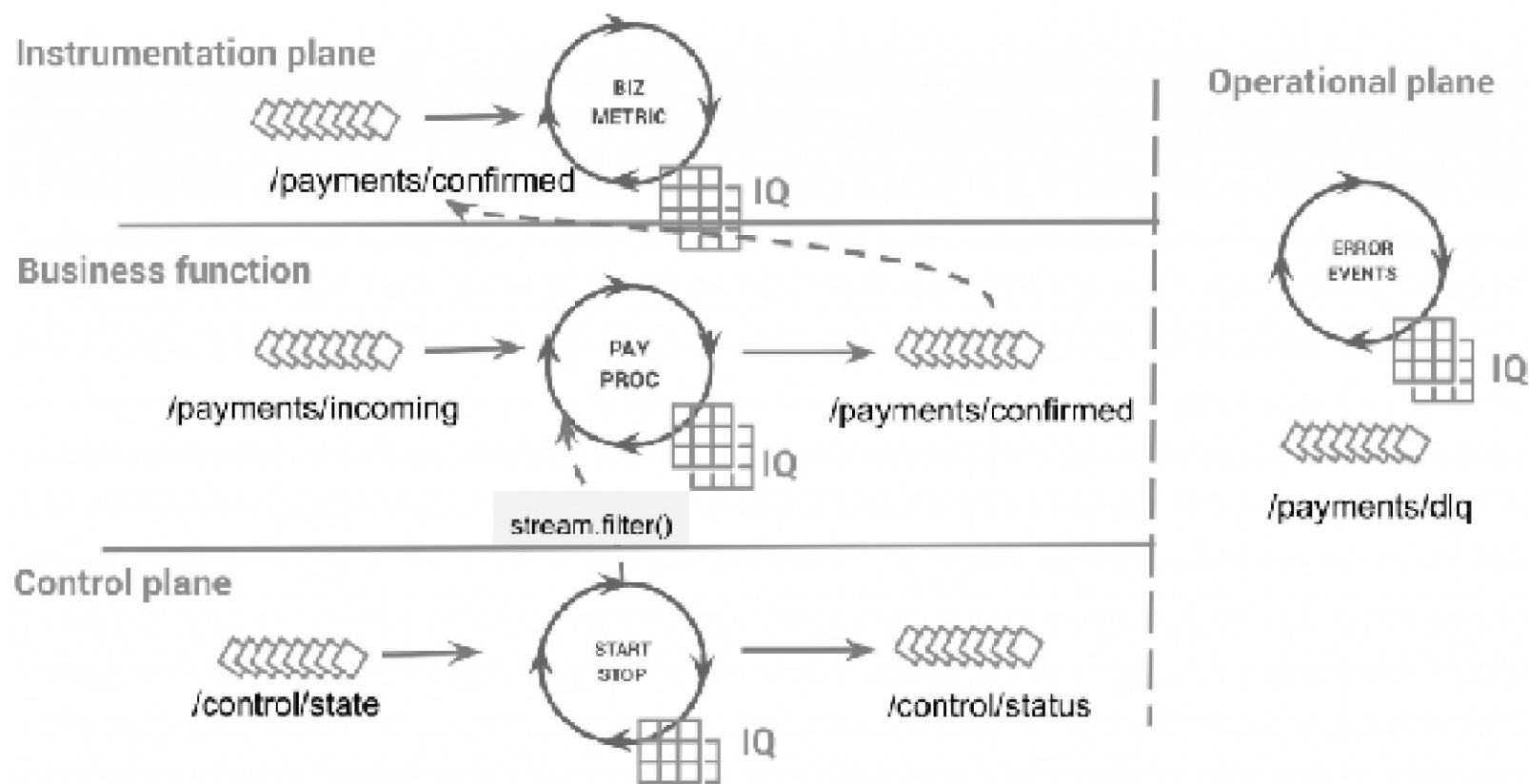
Ferestre - partitionare după timpul de apariție a evenimentelor

Monitorizare continuă



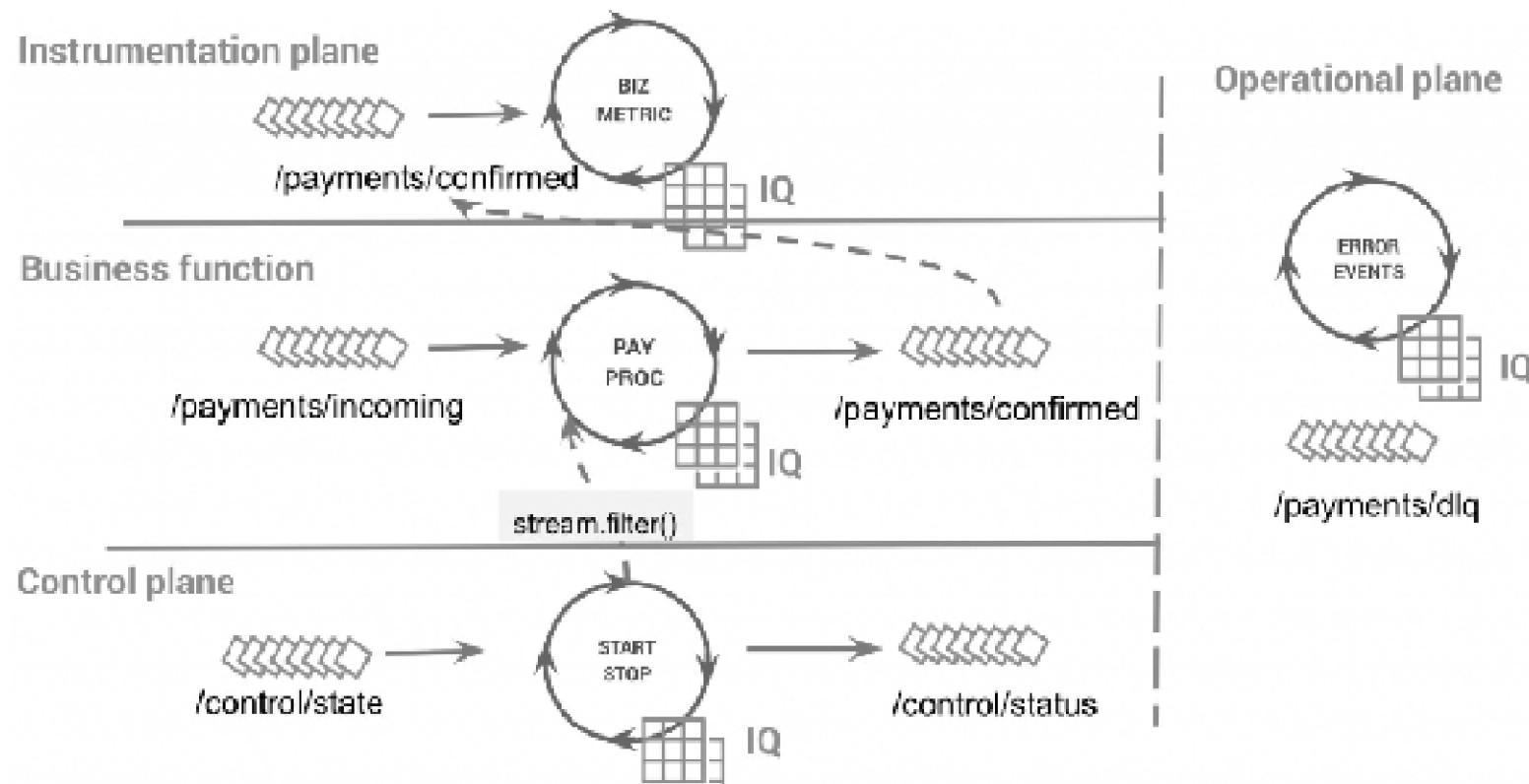
stil DevSecOps DoD

Afaceri și iar afaceri!

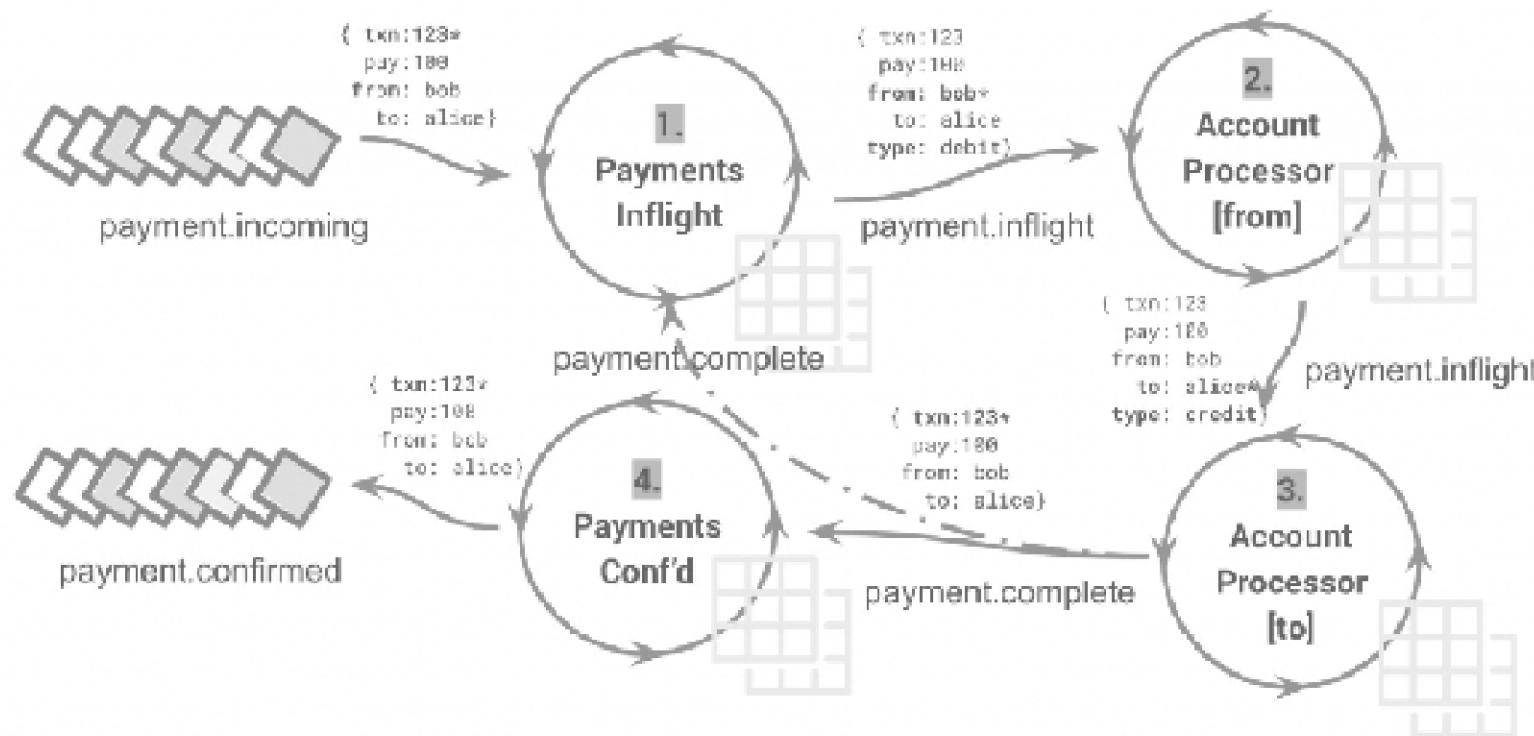


Cele patru planuri ale unei arhitecturi bazate pe fluxuri de evenimente

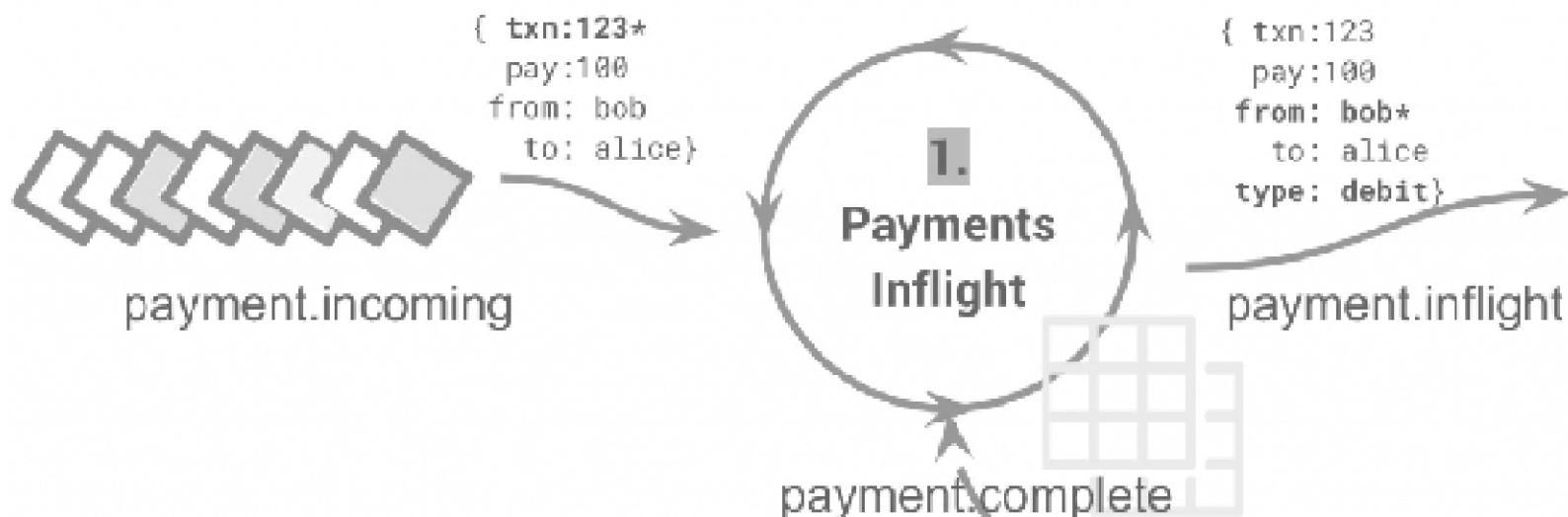
Detalii



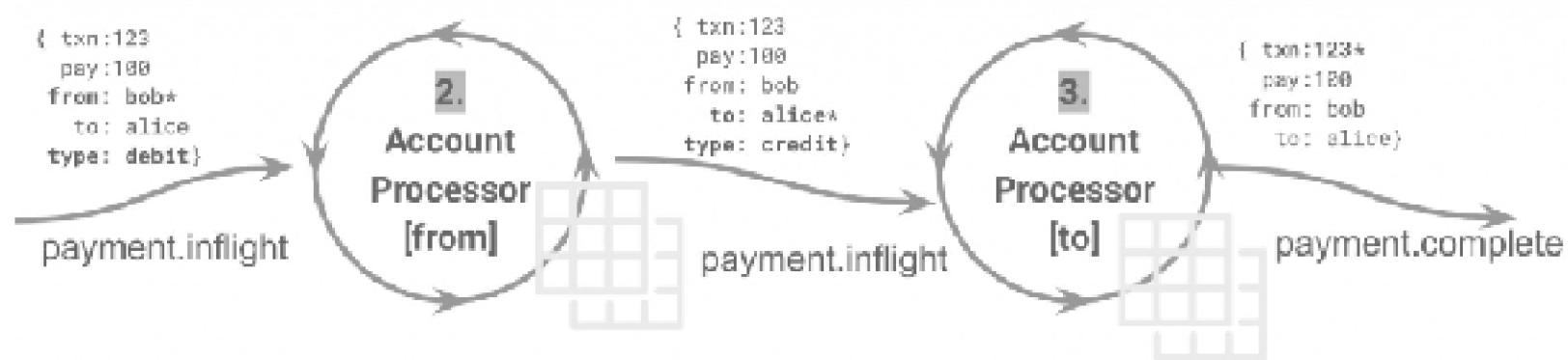
Fluxul de rezolvare al plășilor



Analiza fluxului de evenimente

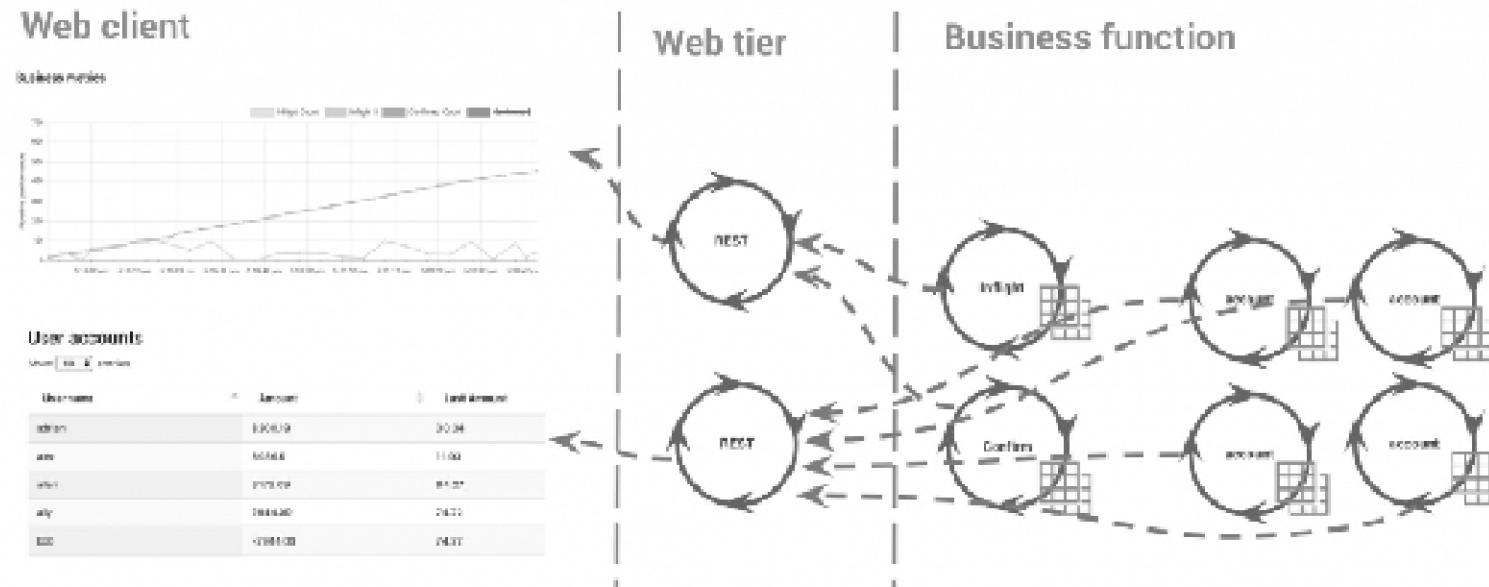


Analiza procesorului de cont

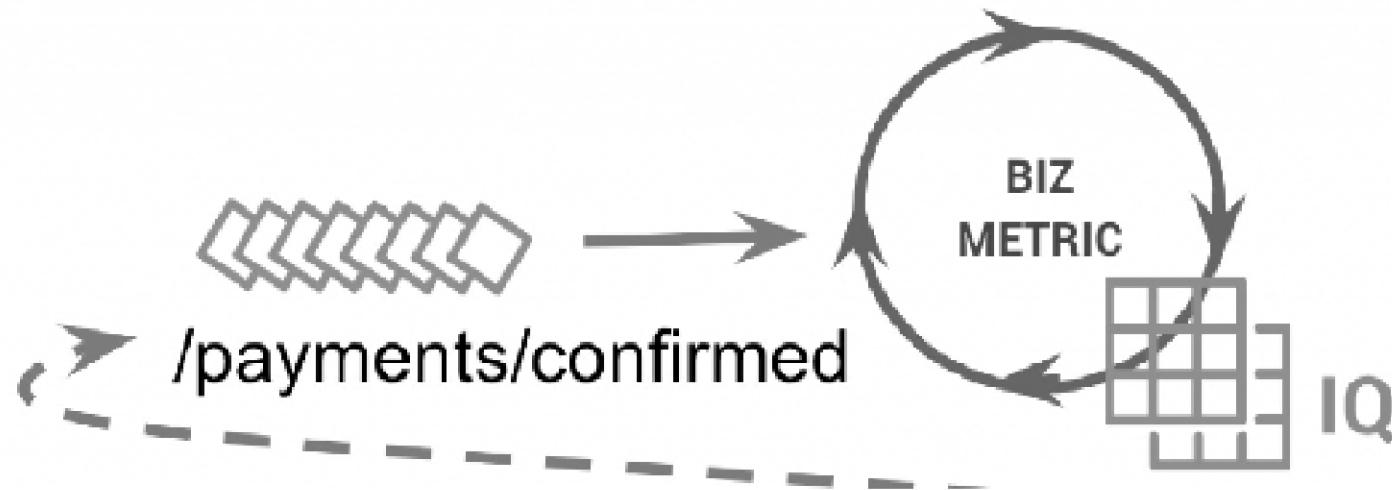


- X

Detalii pentru confirmarea balanței

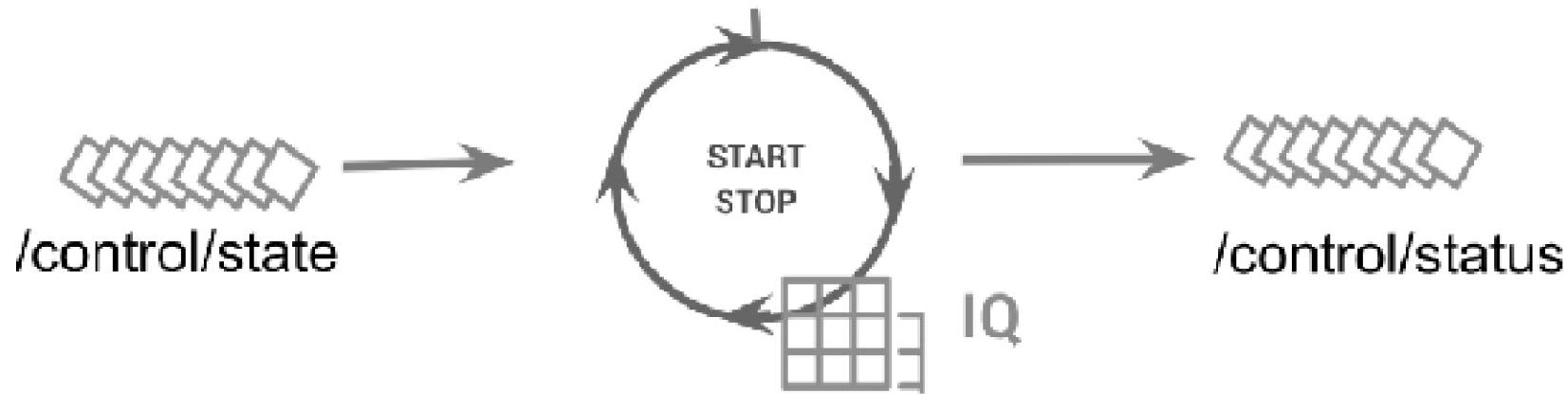


Detalii la planul de instrumentație



- X

Controlul Execuției

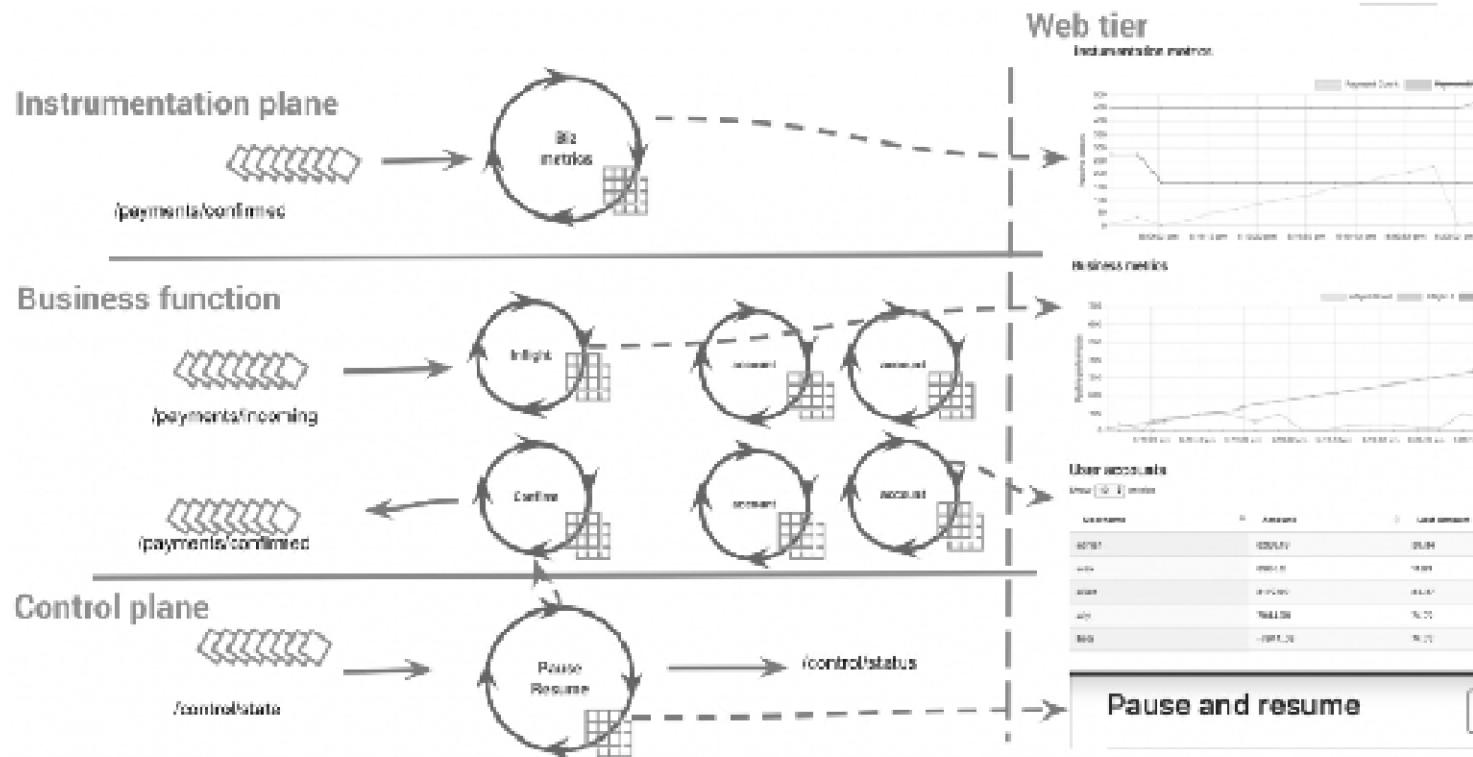


- X

Planul operațional

- Logica Aplicației
- Jurnale de erori
- Jurnale de securitate
- Informații pentru restaurare
- Cozile de mesaje pierdute

Modelul de instalare



Fără servere?

Origine

Unde se află în contextul tehnologic?

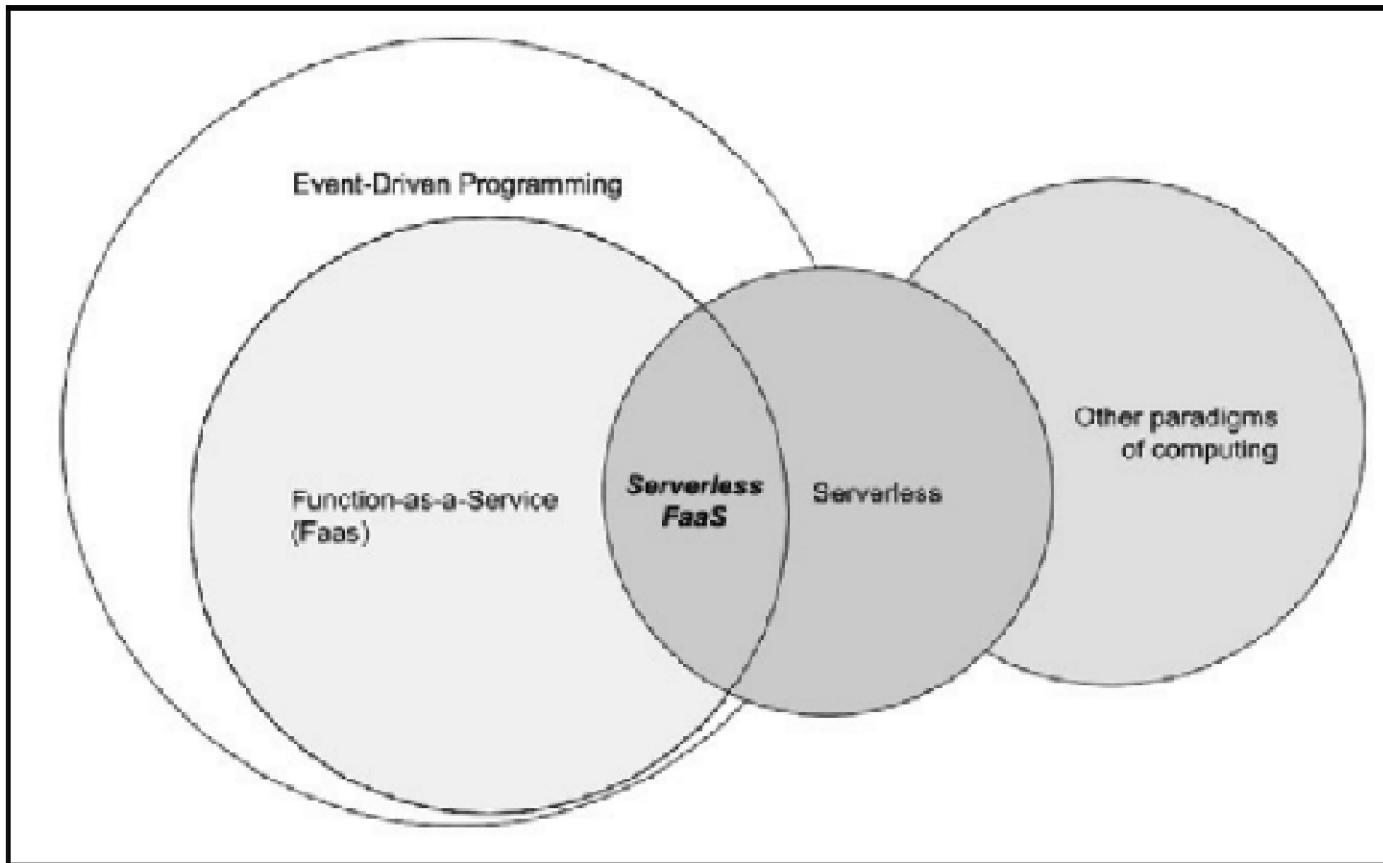
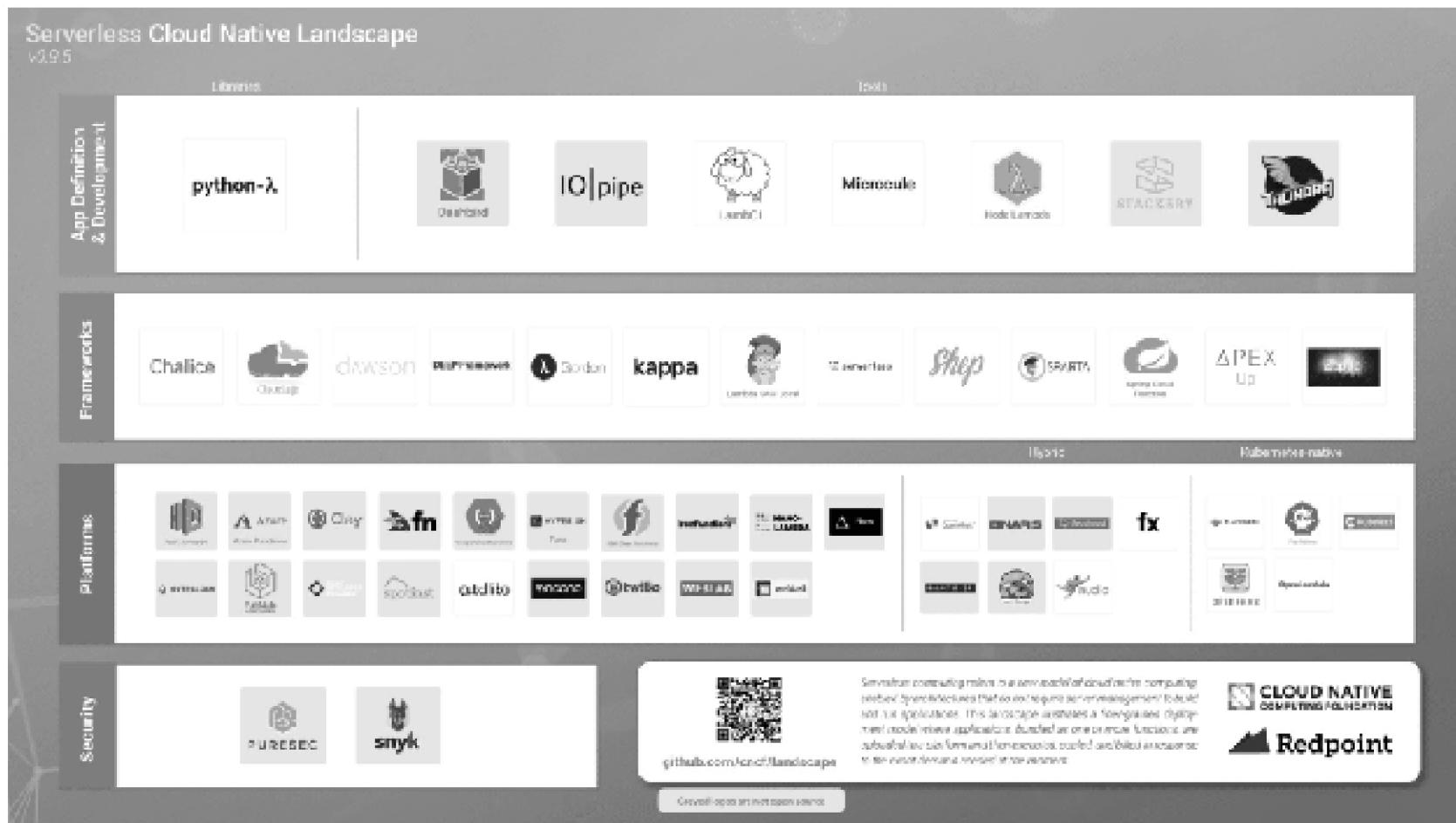


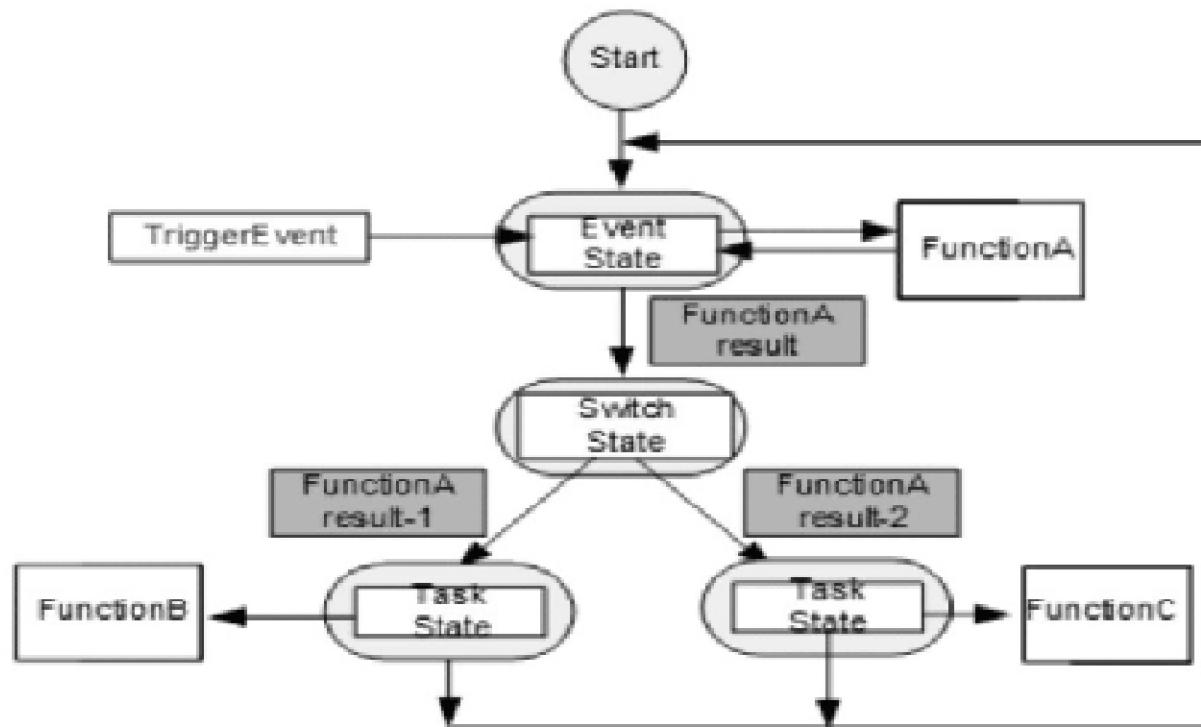
Diagrama Venn de plasare a conceptului

Serverless Working Group - CNCF



- Harta aplicațiilor serverless în nor

Modelul de functionare al FaaS



- X

Care este totuși diferența între PaaS și FaaS?

“If your PaaS can efficiently start instances in 20ms that run for half a second, then call it serverless”

Adrian Cockcroft

VP Cloud Architecture Strategy at AWS

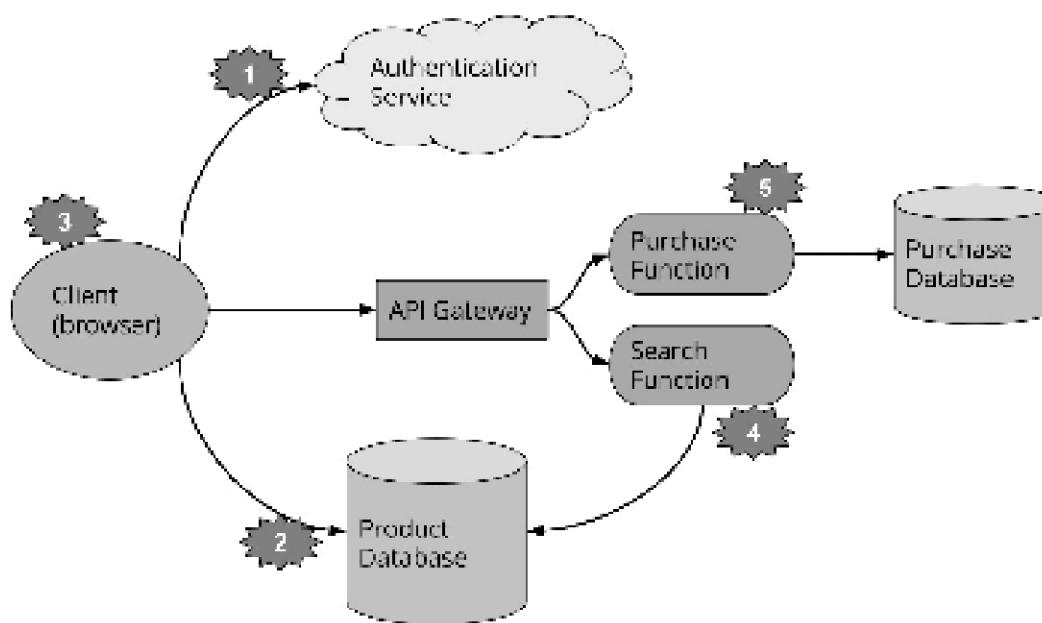
Este aşa de important?

- și da și nu
- depinde la ce am nevoie de el
- limitări
 - timp
 - memorie
- tehnologie proprietară
- soluții publice
- confidențialitate ?

Avantaje:

- induc încărcări mici
- decuplare maximă
- scalabilitate
- eficiente economic

Exemplu simplu - multilevel clasic



"favoritul" meu - magazinul on-line

Exemple simple - bazat pe mesaje



- arhitectură orientată pe mesaje - clasică



- arhitectură orientată pe mesaje - cu FaaS

Matricea pentru selecția FaaS

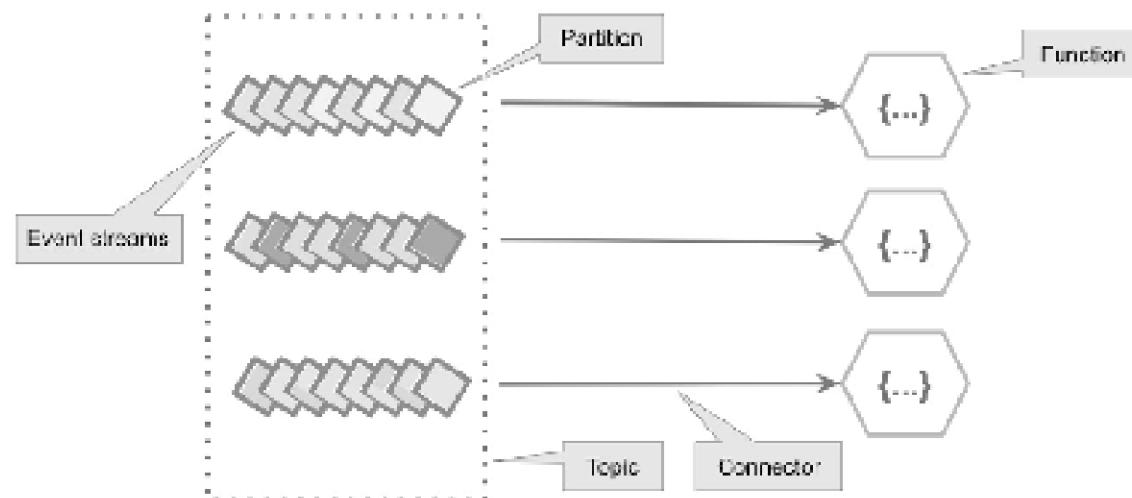
Atribut	Procesare nativă pe flux	Procesare FaaS
Model evenimente	Asincron	Limită functii (în AWS 1,000 soft)
Întârziere/T-reactie	Scăzut (<10 ms)	Mare (cu inițializare: 5s), (după inițializare: 25–300 ms)
Încărcare	Înaltă	Medie spre înaltă
Elasticitate	controlată extern per nod	Nativă, (per instanță funcție)
Respectare flux	Da	Nu
Operații fără stare (filtrare, îmbogățire, transformare)	Da	Da

Matricea pentru selecția FaaS

Atribut	Procesare nativă pe flux	Procesare FaaS
Operații cu stare (fereastră, compunere)	Da	dezvoltate manual
Cost	necesită în totdeauna un procesor să funcționeze	numai când este ceva de procesat
Runtime	VM, container, server	furnizor FaaS
Limitări în timpul execuției (runtime)	manuală	dependent de furnizor: 500 MB storage, 128 MB → 3,008 MB memorie

Evenimentul primul și FaaS

- furnizor de fluxuri
- kafka
- coregrafia și încâlceala
- nivelurile app
- conector



FaaS Când Da și Când Nu pentru procesarea flux

- Cand am vrevoie de viteză mare de reacție - DA
- Încărcare mare - DA
- Operații fără stare - DA
- Operatii cu stare orientate pe flux - NU
- Alte operații cu stare - DA
- Corectitudinea fluxului - NU
- Modele pe flux - NU

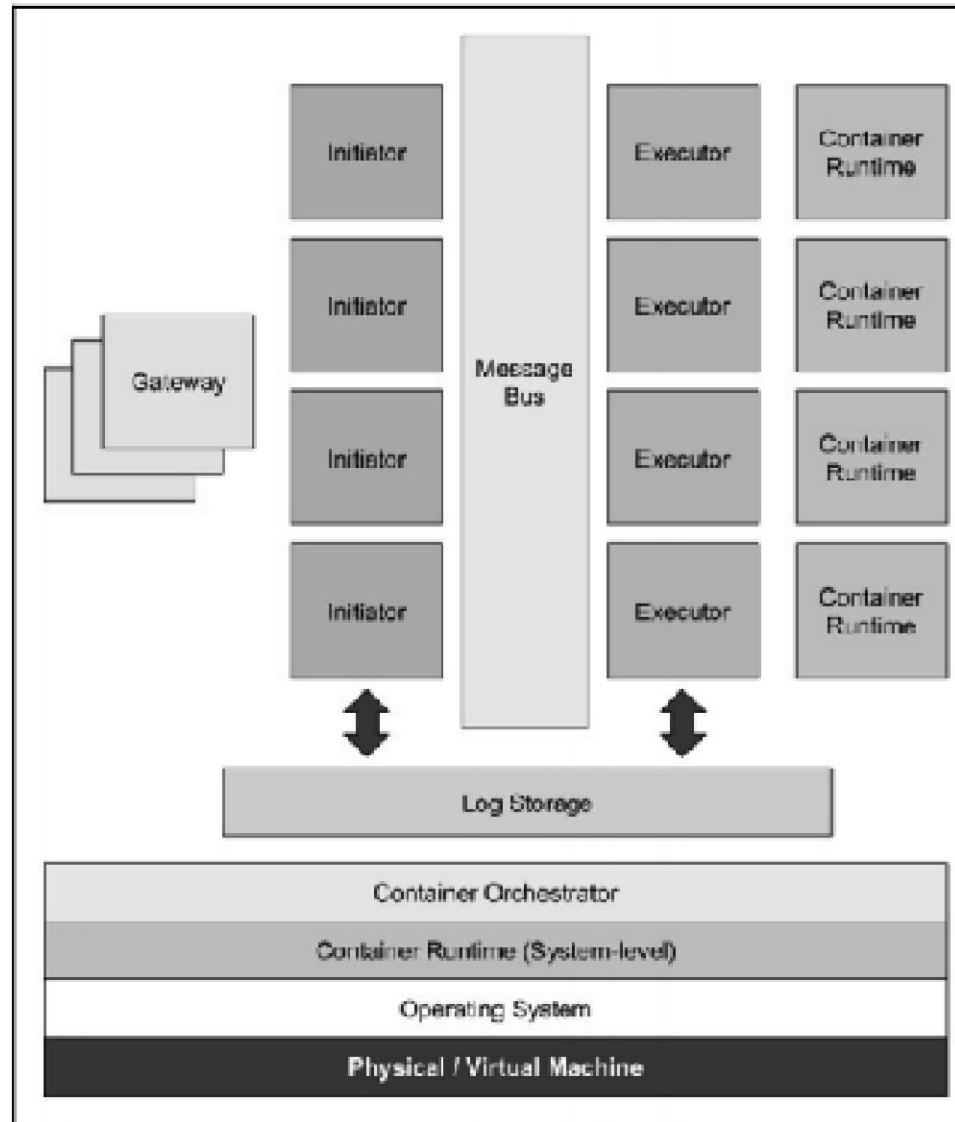
FaaS și procesarea pe flux

- *"AWS Lambda and serverless architectures are well-suited for stream processing workloads which are often event-driven and have spiky or variable compute requirements"*
 - Amazon, 2018

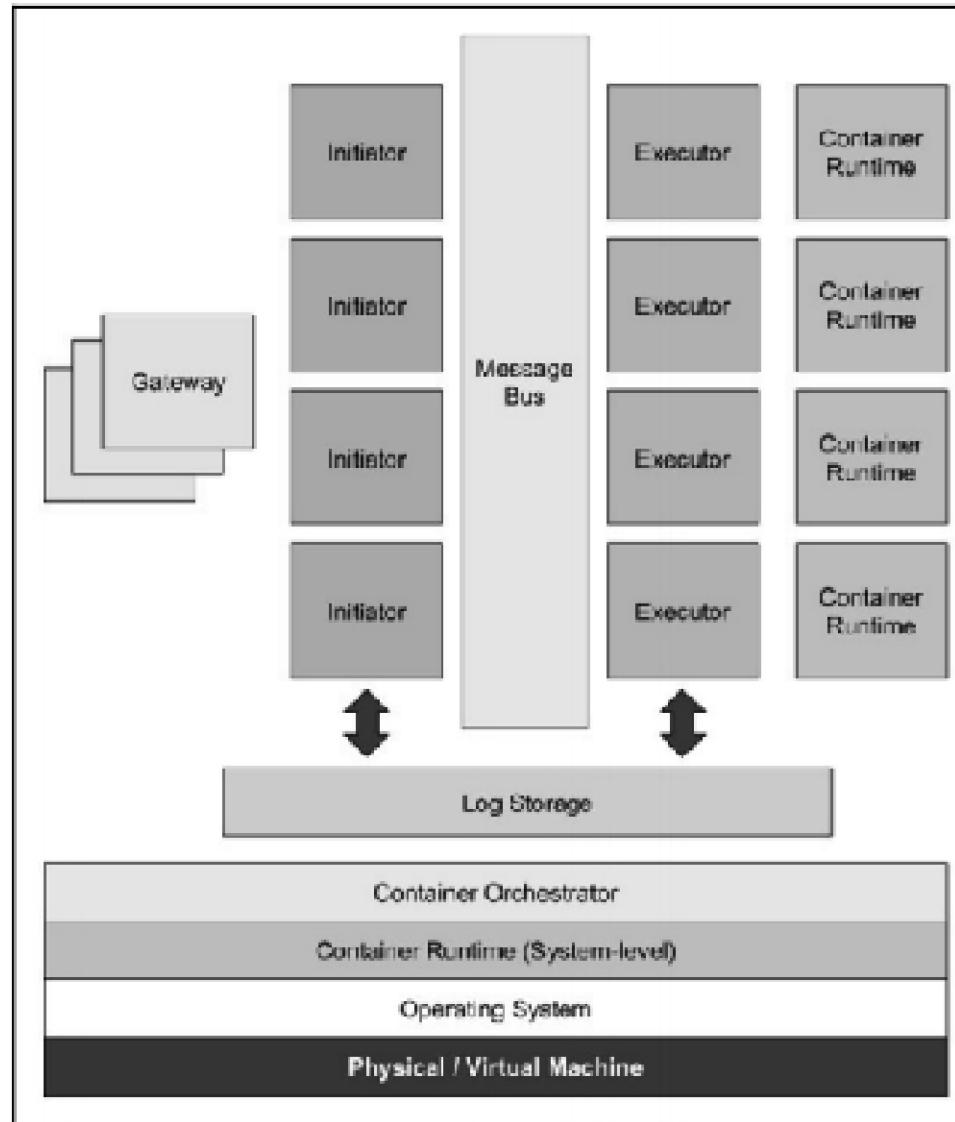
Performanțe FaaS în procesarea pe flux

- date dinamice
- procesorul cu timpi fixi
- cache-ul ?
- pornirea initială (la rece)?

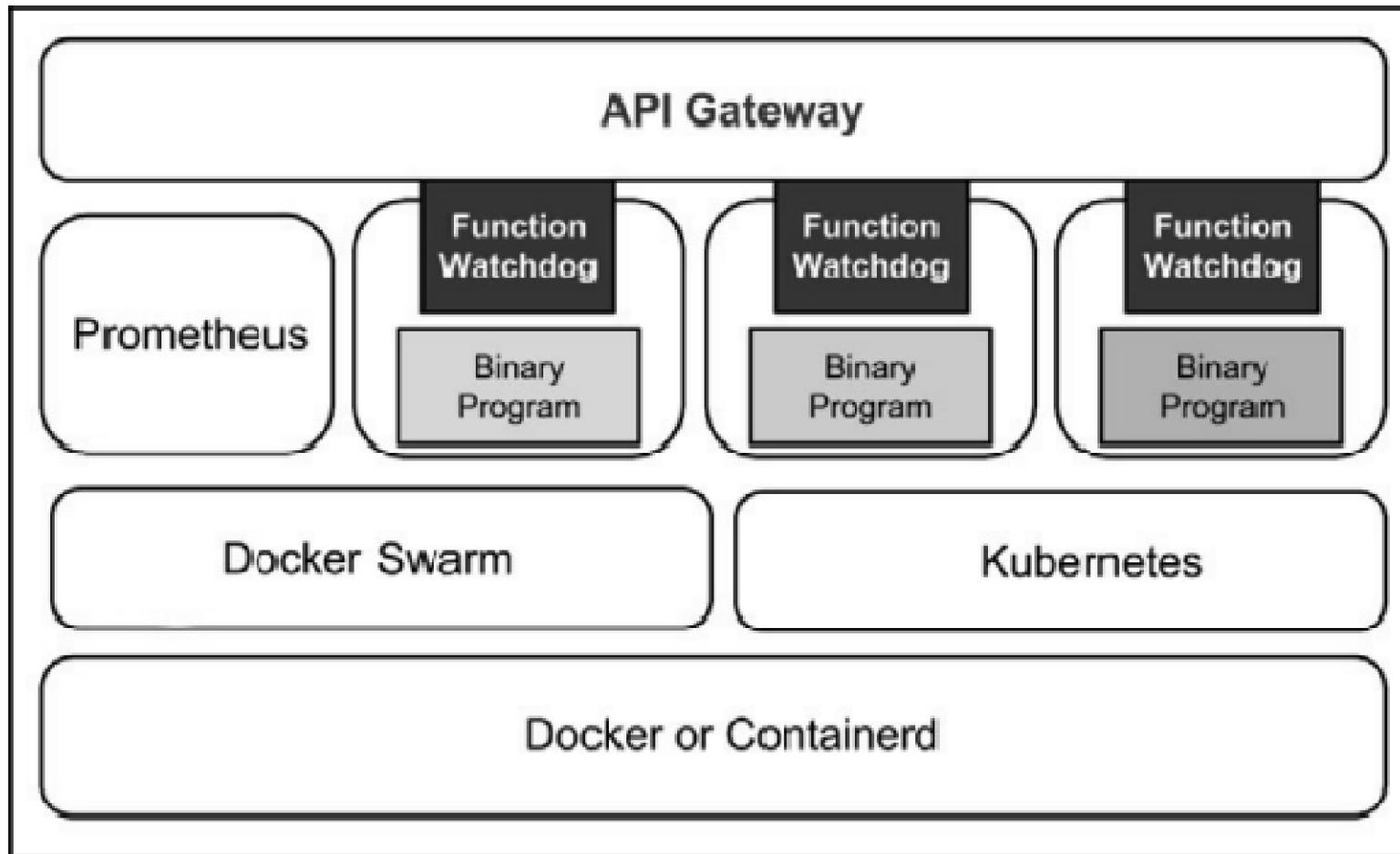
Arhitectura FaaS - nivelurile Sistem



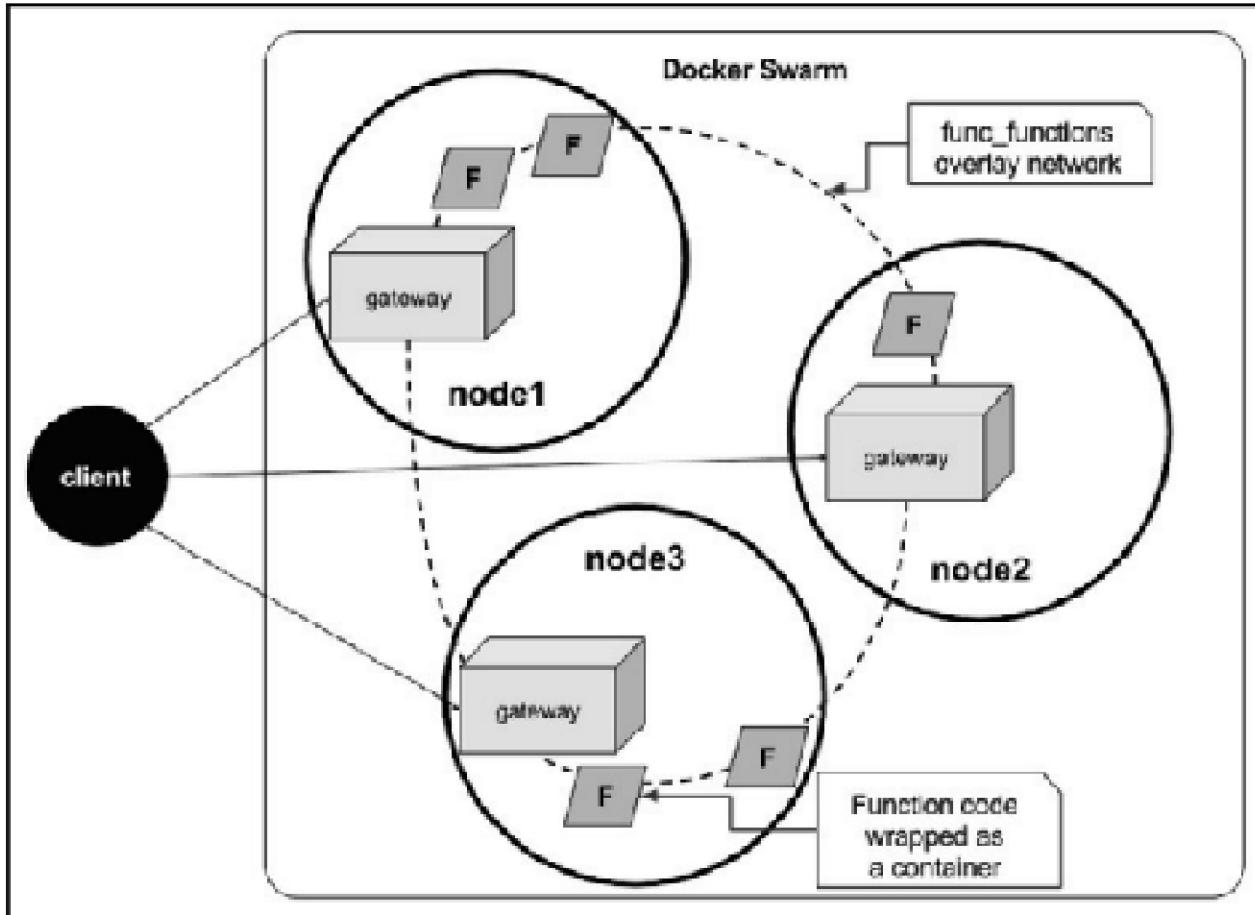
Arhitectura FaaS - nivelurile FaaS



Componentele OpenFaaS

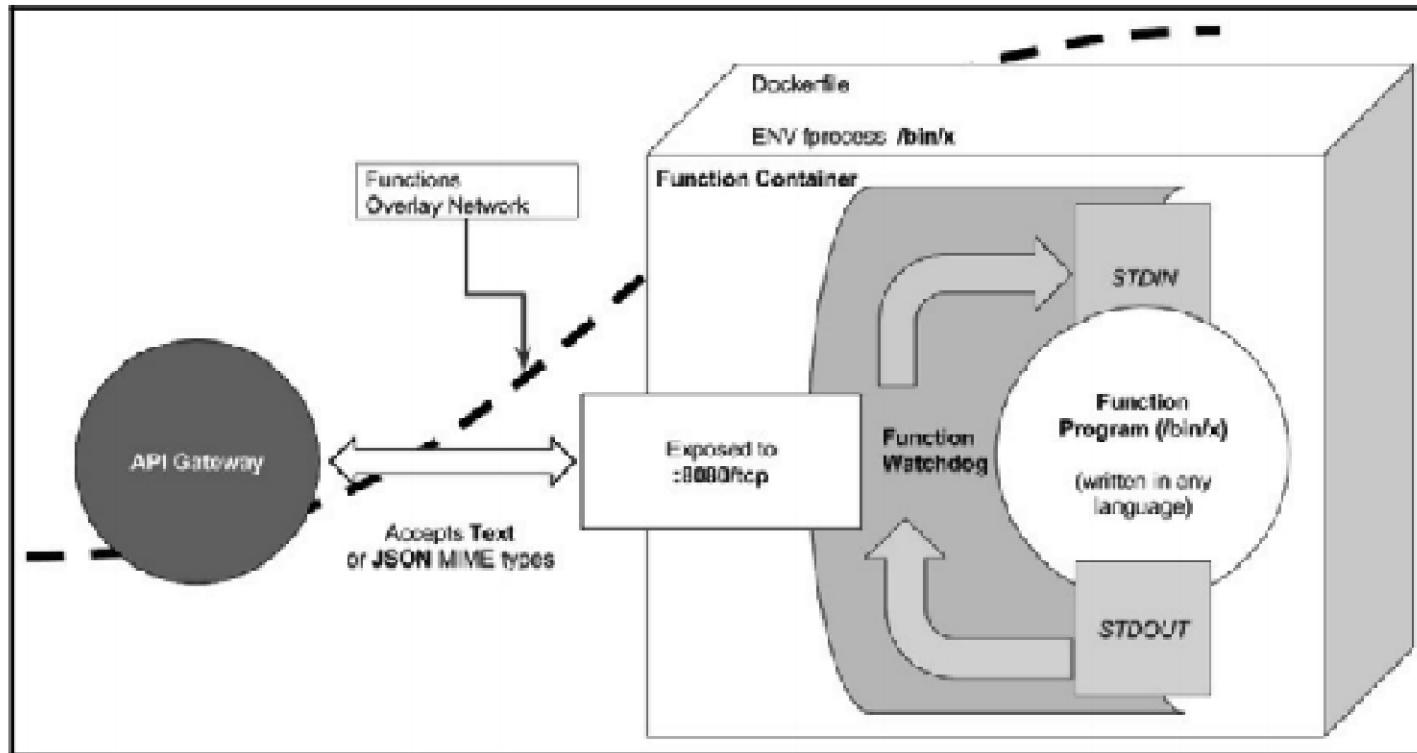


Arhitectura OpenFaaS



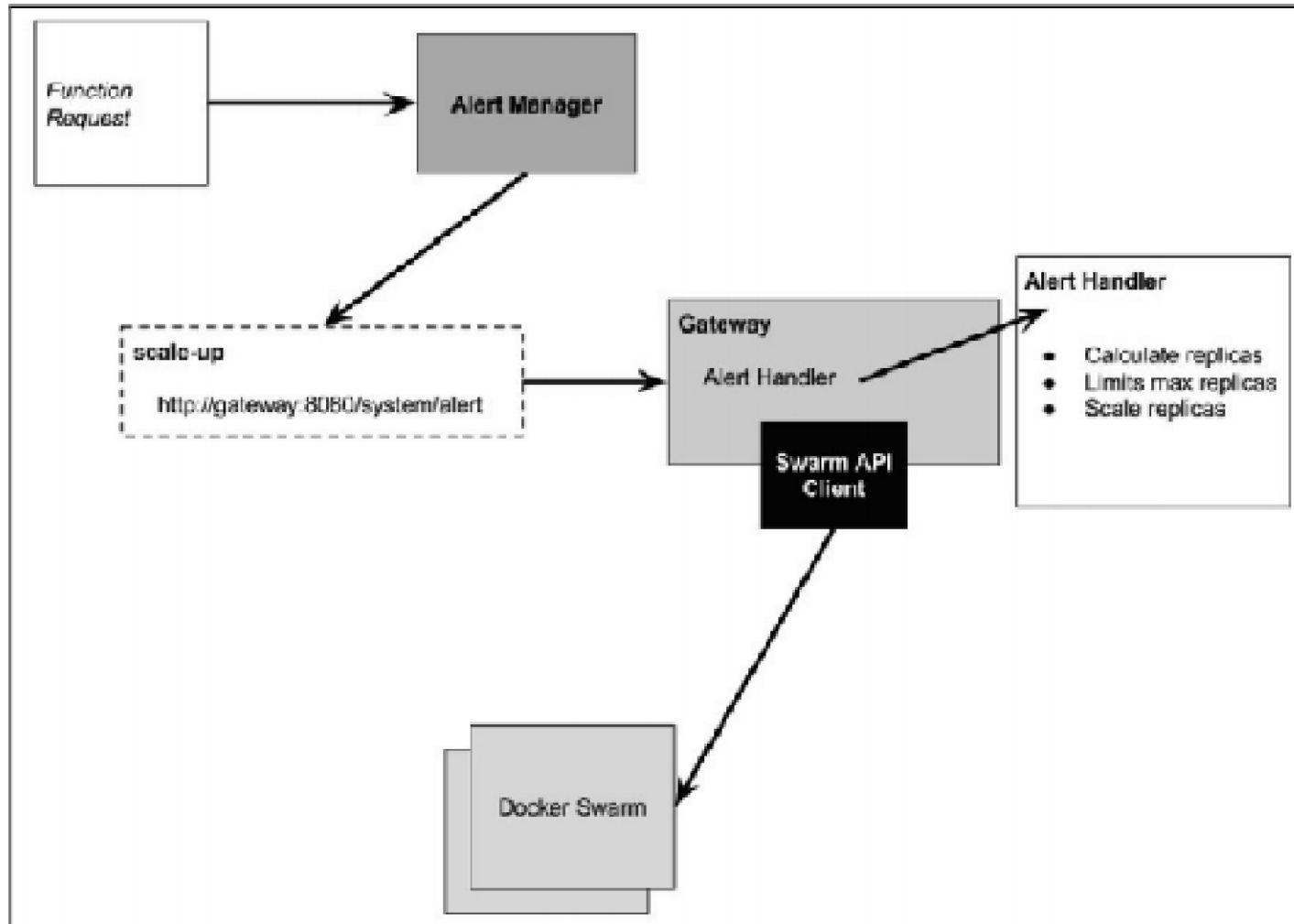
infrastructura internă a OpenFaaS executată pe Docker

Functia de monitorizare - watchdog



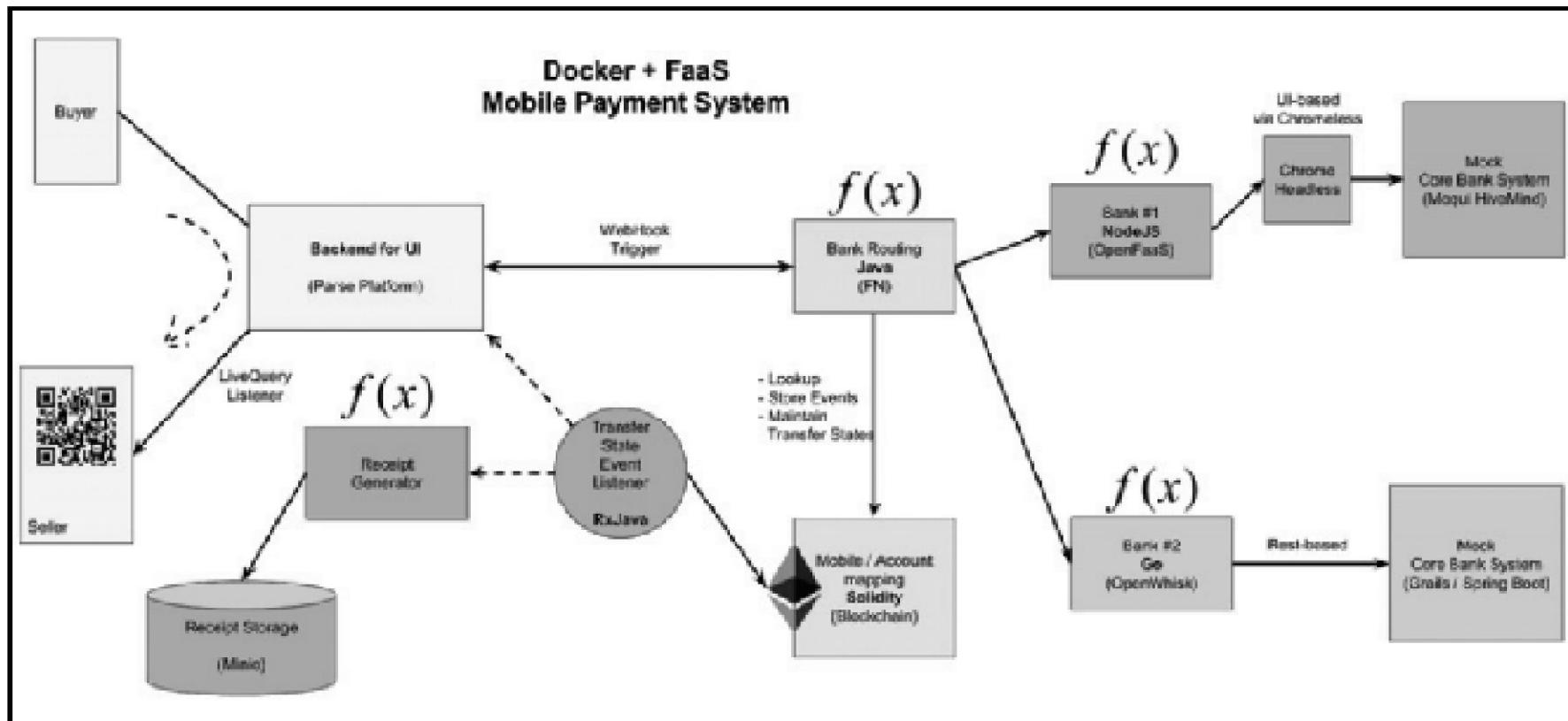
fwatchdog

Scalare FaaS în Docker

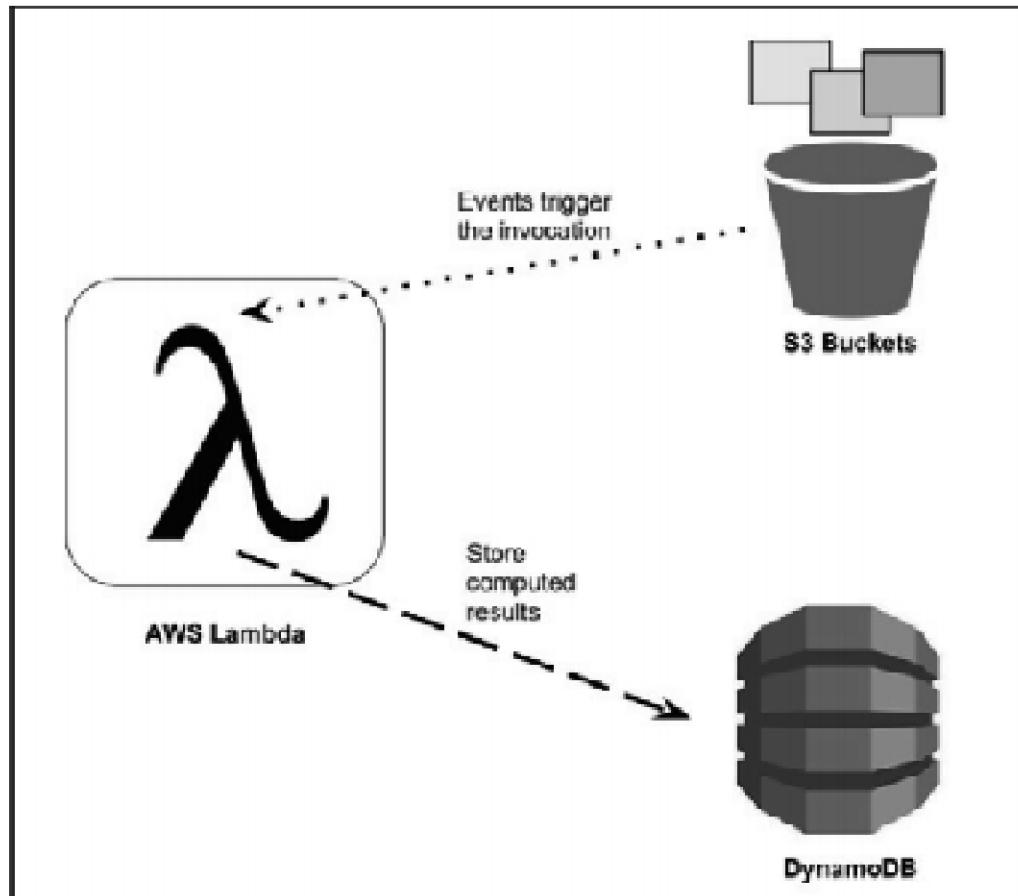


Alert Handler URL - `http://gateway:8080/system/alert`

Exemplu de caz de utilizare pentru FaaS



AWS Lambda

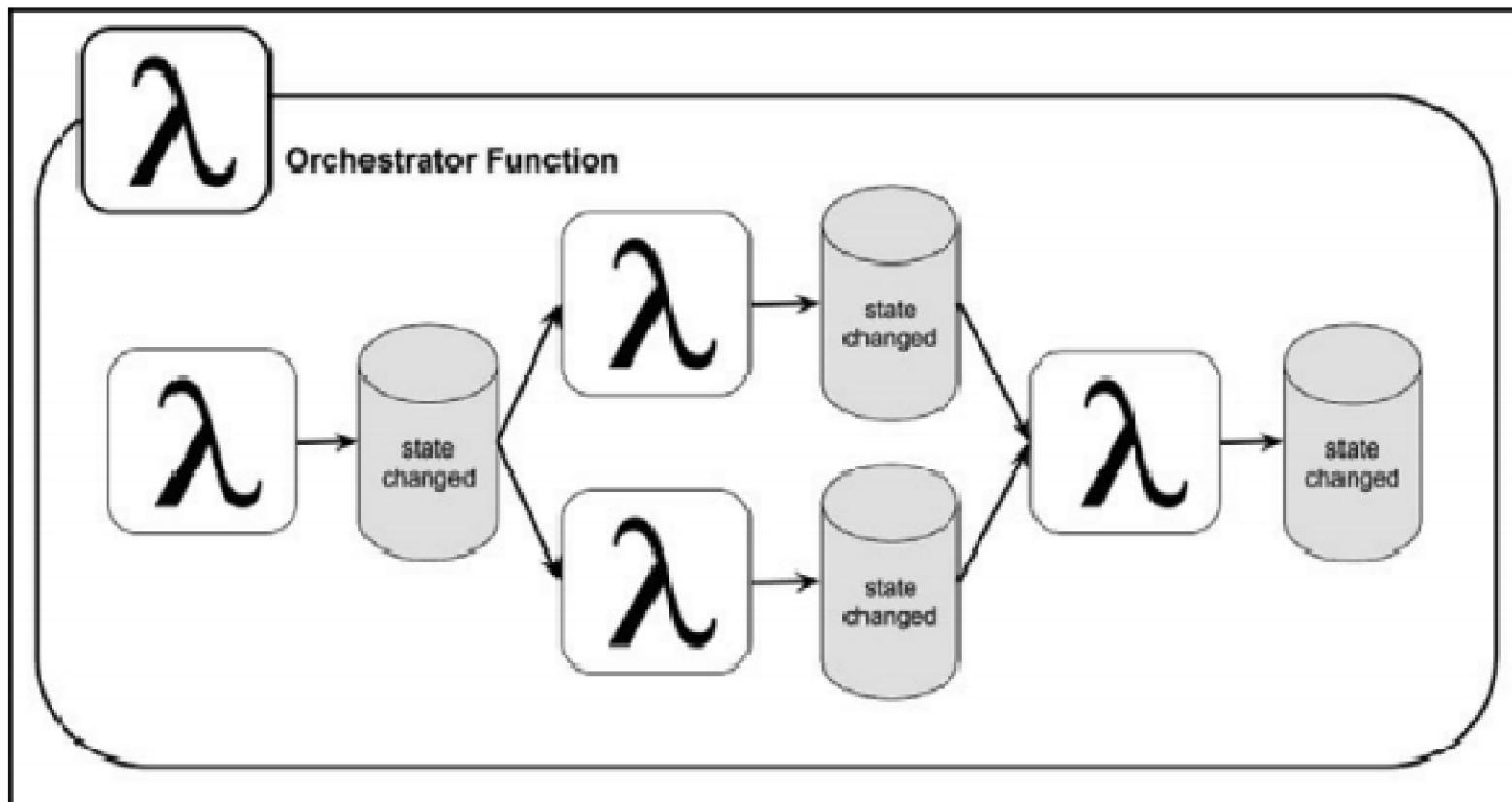


Exemplu simplu de utilizare

Oprirea execuției unei f. Lambda AWS

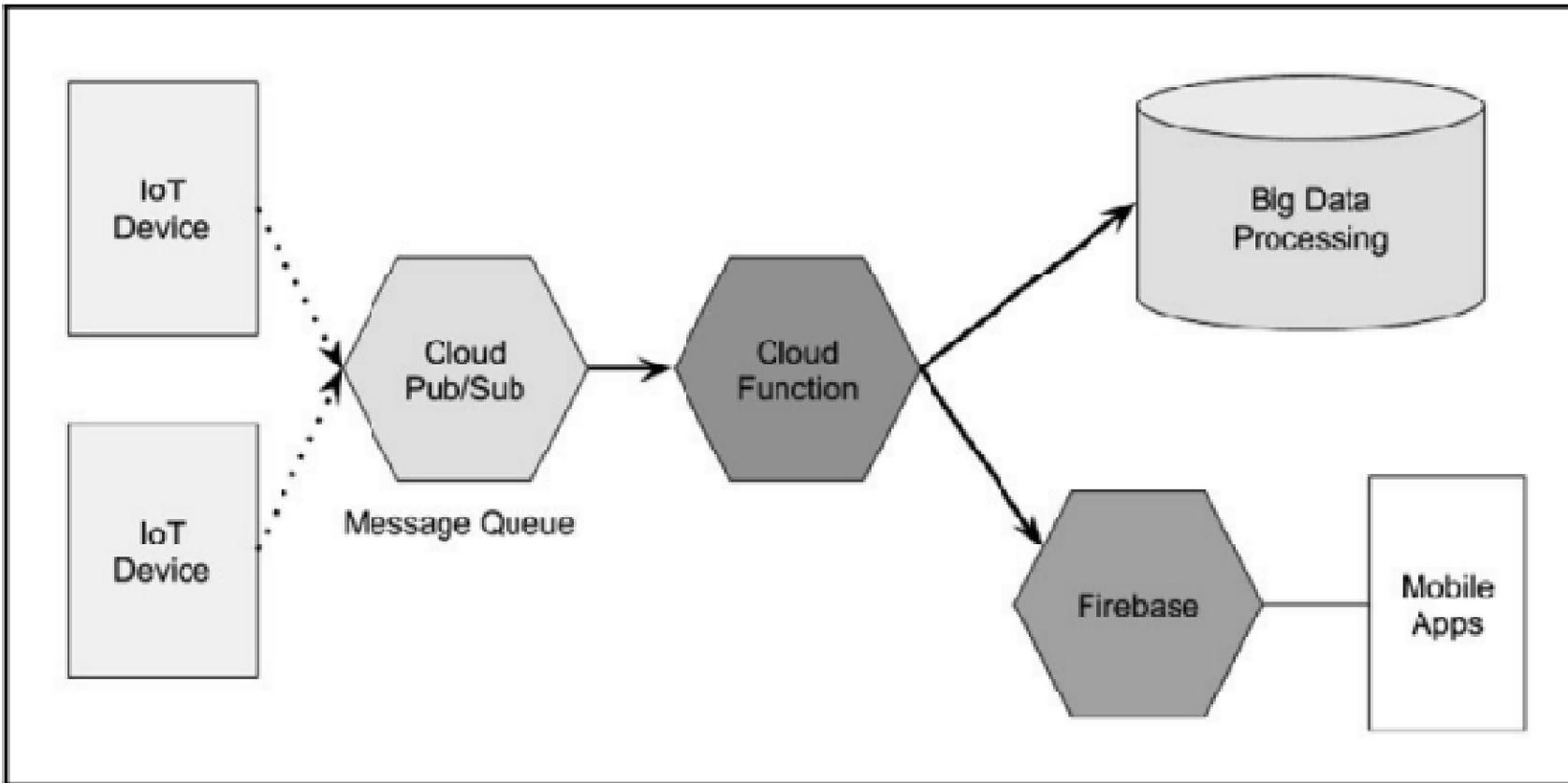
- funcția în container
- condiții de oprire
 - de timp
 - oprire la cerere
 - implicită
 - eroare

Funcții durable Azure



orchestrator bazat pe funcții durable în Azure

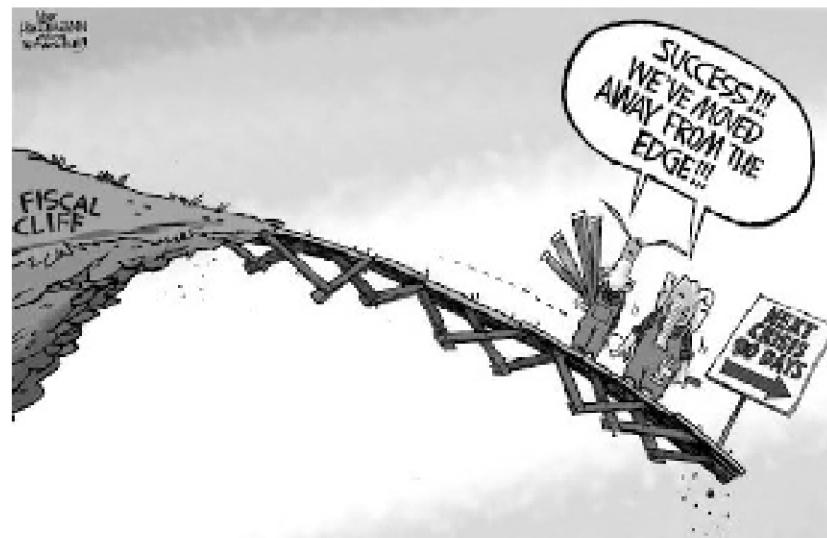
FaaS în norul Google



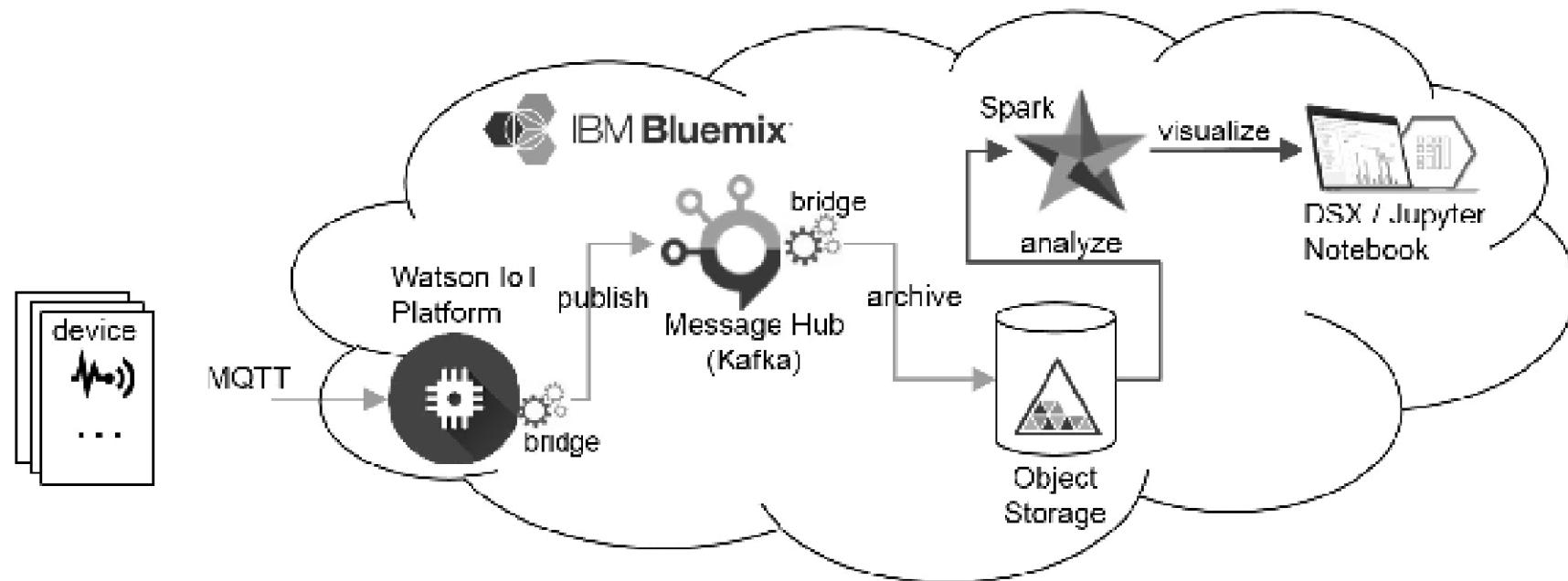
exemplu simplu de IoT implementat utilizând Google FaaS

Ce este edge computing?

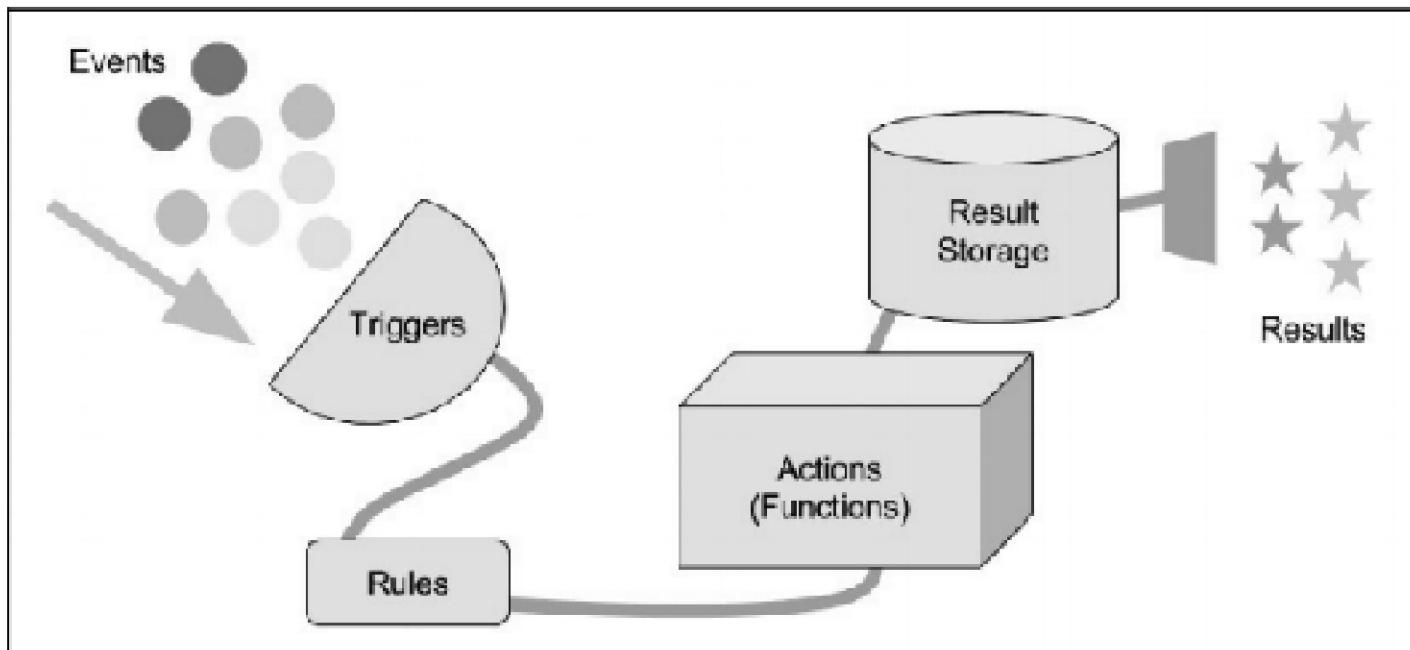
- rețele radio de comunicare de mare viteză
- microcontrolere de mare performanță
- IoT --> IoE
- aplicații în timp real pe mobile



IBM, ServerLess și IoT-ul

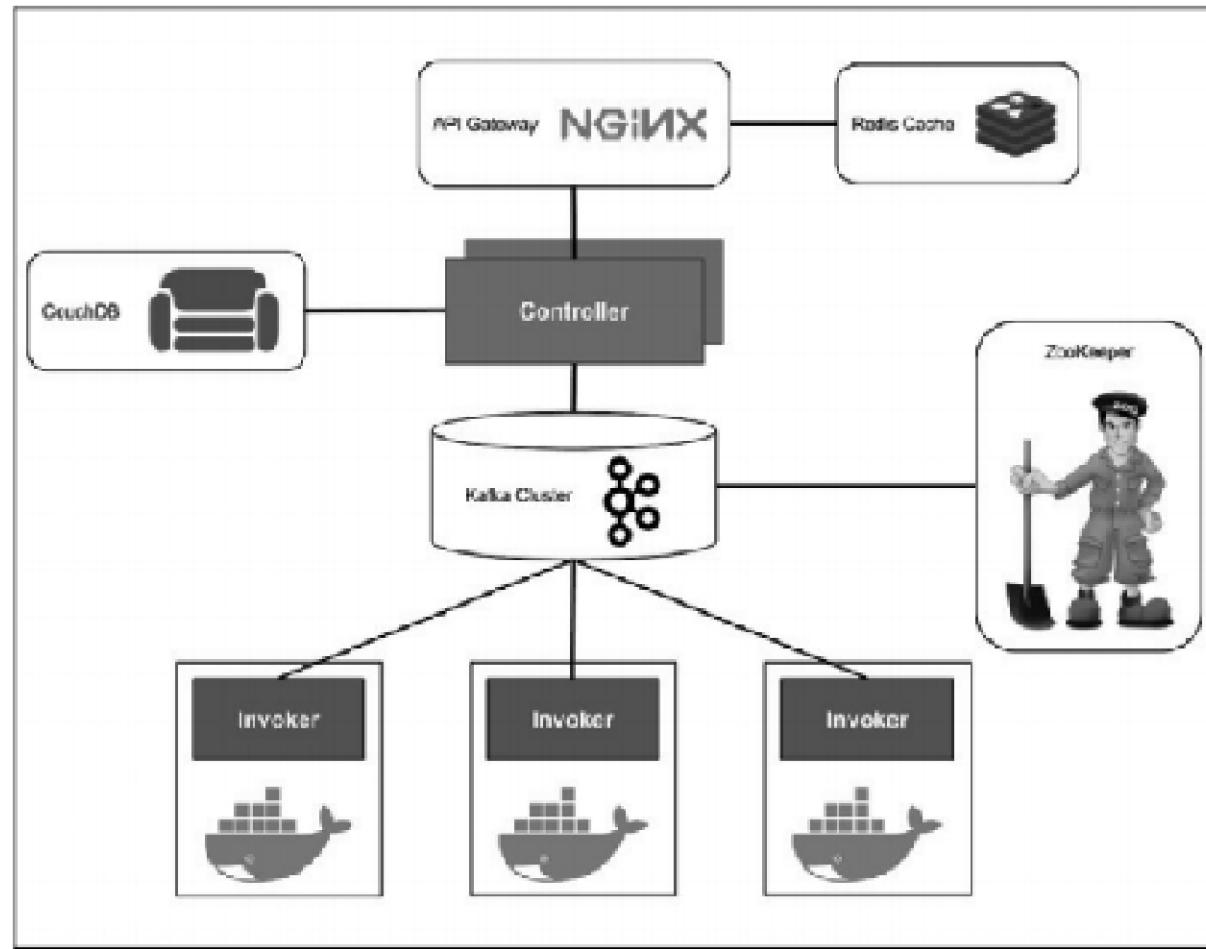


Arhitectura OpenWisk



- fluxul de evenimente din OpenWisk

OpenWisk&Docker



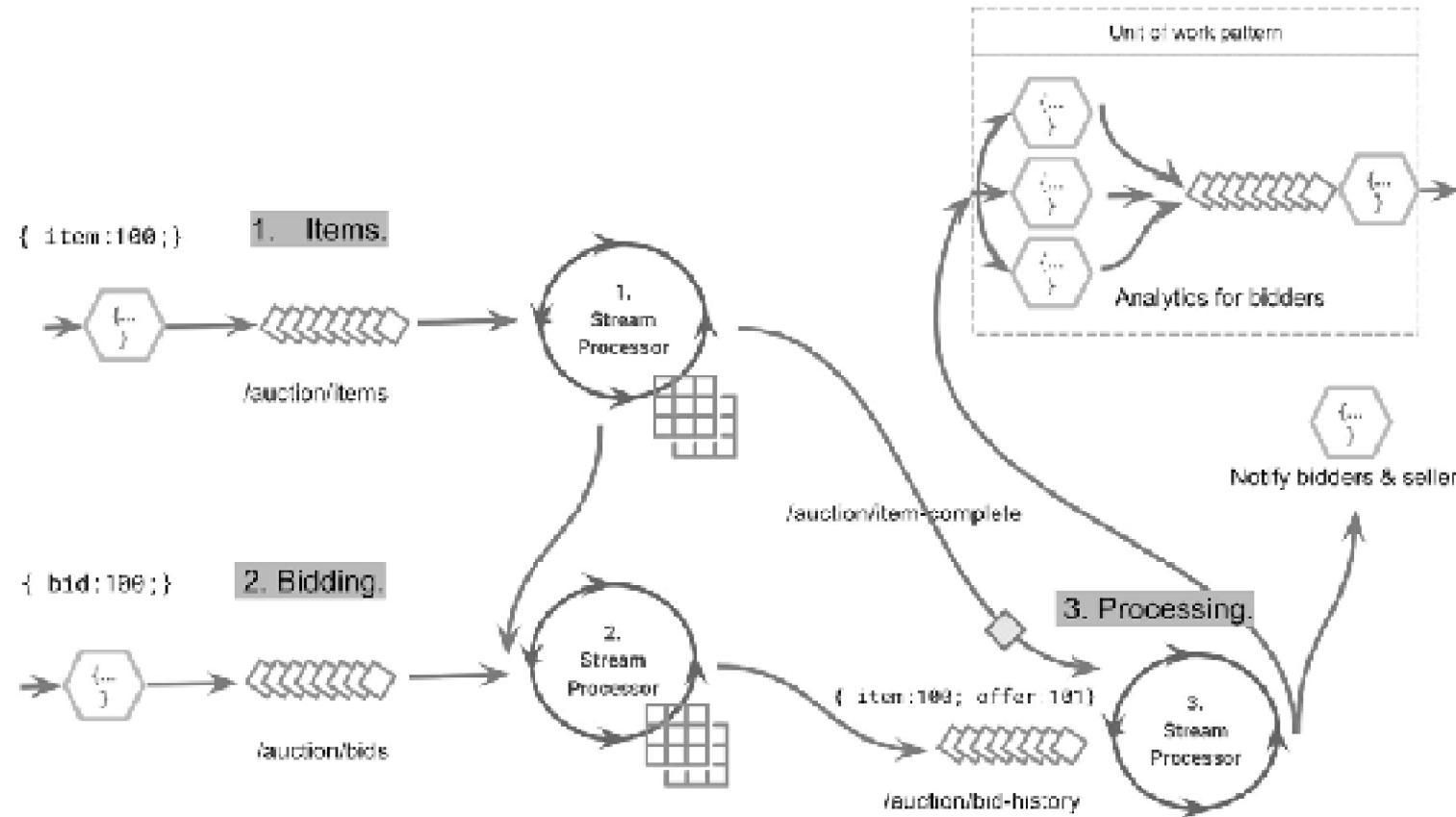
- X

Exemplu de utilizare FaaS în flux

- hand made
- mix tehnologic
- posibilități
 - AWS Lambda cu Kinesis
 - Kafka Connect cu AWS Lambda Sink Connector



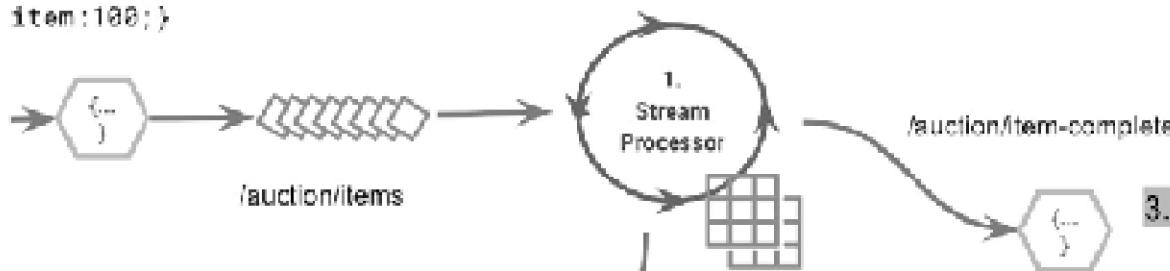
Exemplu de utilizare FaaS în flux



Exemplu de utilizare FaaS în flux

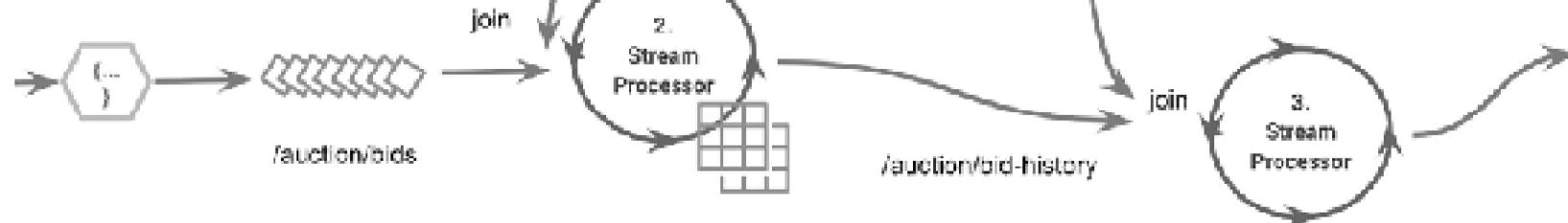
1. Validation

{ item:100; }



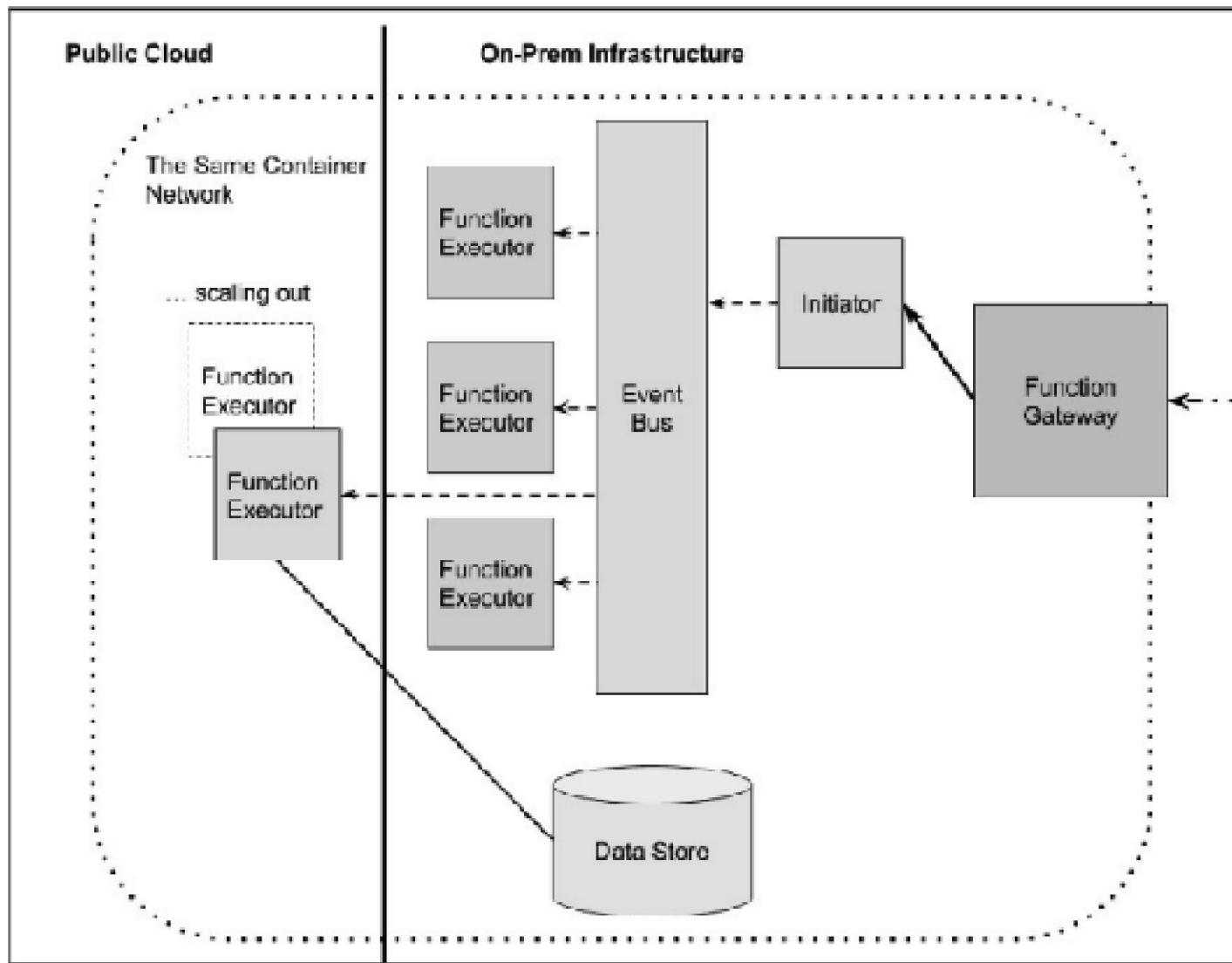
2. Validation

{ bid:100; }



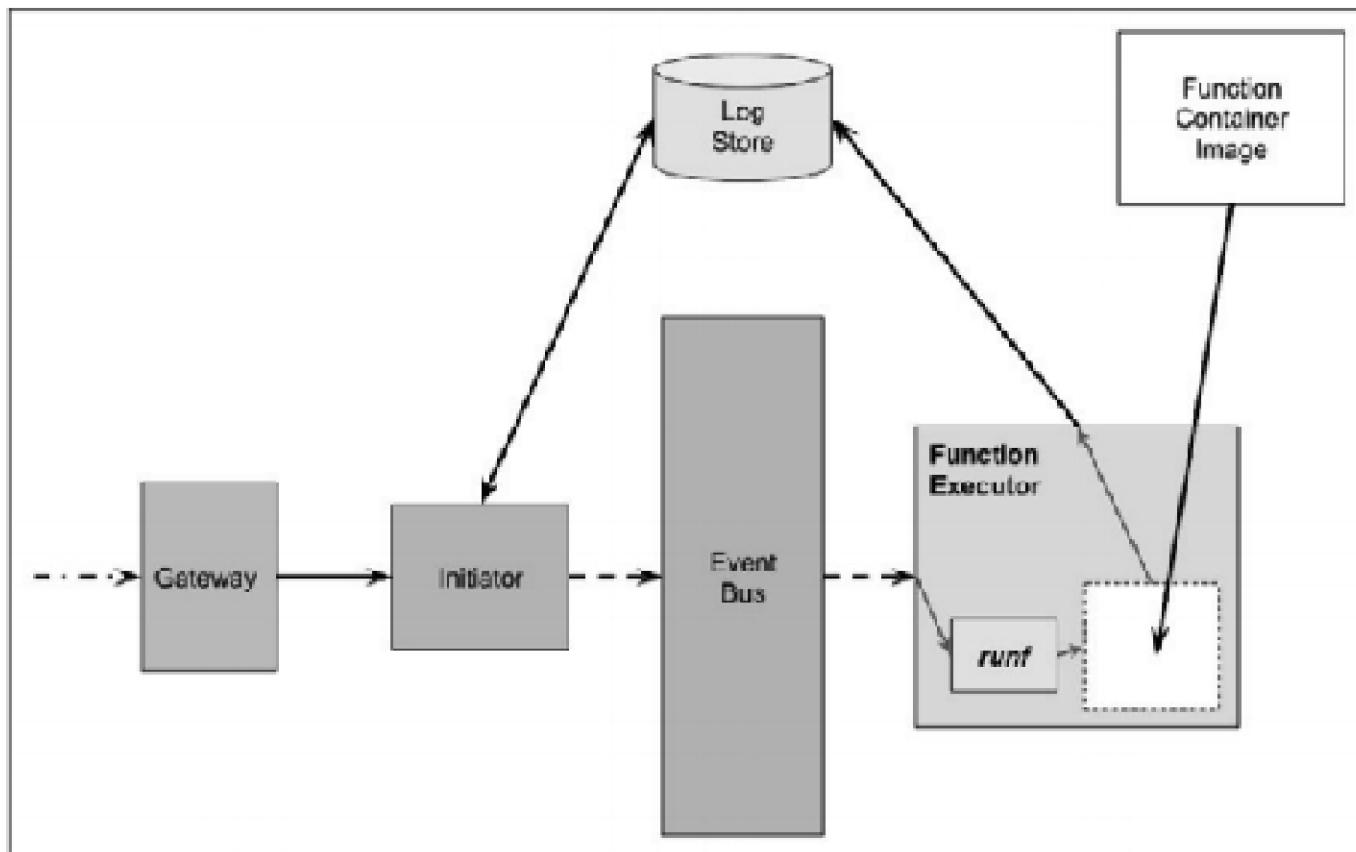
- X

Ce urmează după serverless?



- X

Viitorul FaaS poate?



FaaS bazat pe RunF