

Virtualizarea resurselor de calcul

Conf. Dr.Cristian KEVORCHIAN
Facultatea de Matematică și Informatică
ck@fmi.unibuc.ro

Centre de Date

- **Centrele de date mari(50000 de sisteme) sunt mai economice decât cele de dimensiuni medii(1000 de sisteme). Totalul centrelor de date în 2021 este de 3 milioane (1cd/100 de locuitori)**
- Costurile pentru networking in USD pe Mbits/lună sunt de 7.1 ori, iar cele de stocare Gbyte/luna sunt de 5.7 ori mai mari decât în centrele medii.
- În 2006 in SUA erau aprox 6000 de centre de date, care consumau energie electrica in valoare de 4.5 miliarde de USD, iar în 2011 se cheltuia dublu
- În 2021 in USA erau 2670 de centre de date
- **Microsoft avea în aprilie 2021 peste 4 milioane de servere în centrelle de date din întreaga lume**

Open Compute Project

- Reprezintă o comunitate globală de lideri în domeniul tehnologiei care cooperează pentru a elimina blocajele de la nivelul infrastructurii IT proprietare în scopul eficientizării producției de hardware prin creșterea flexibilității și scalabilității acesteia în scopul unei adaptări mai bune la cerințele de business.
- Proiectul **OLYMPUS** - dezvoltarea open-source a proiectelor hardware pentru cloud computing.
- Proiectul OLYMPUS permite operatorilor IT și datacentrelor să beneficieze de inovațiile dezvoltate de comunitate și să extindă modelele hardware care se dovedesc utile în contexte de business specifice



12U Shared Chassis
EIA Rack Mountable

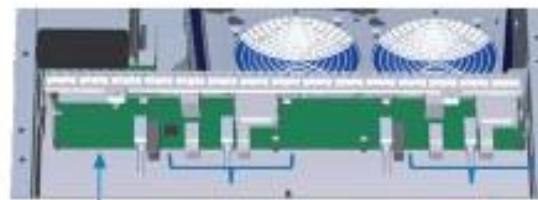
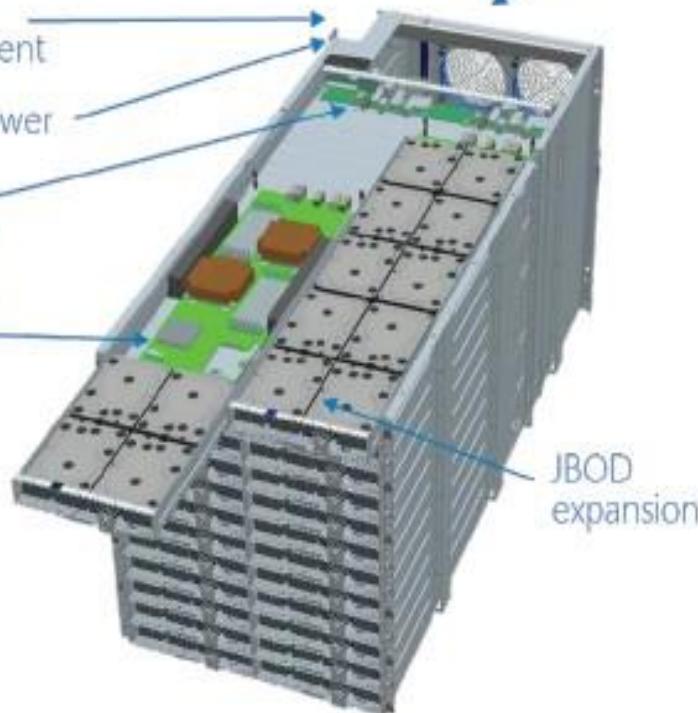
Shared management

Shared power

Signal backplane

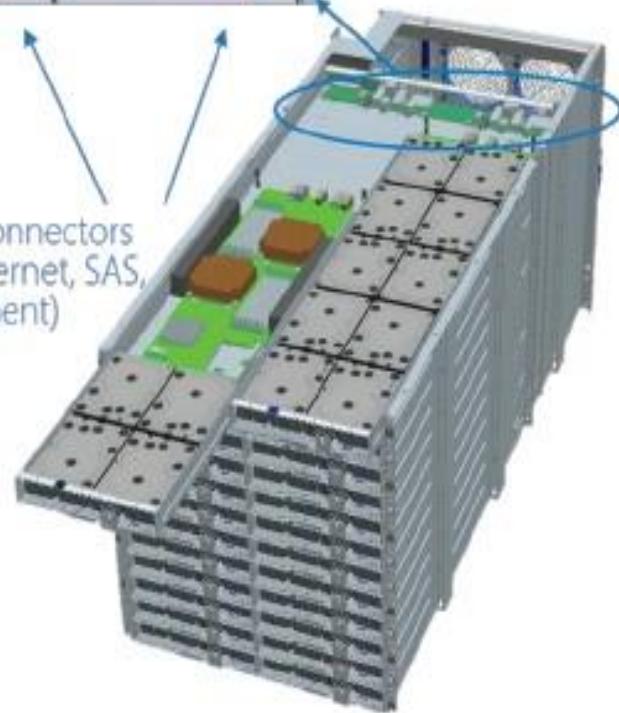
Compute blade

Shared fans



Signal Backplane

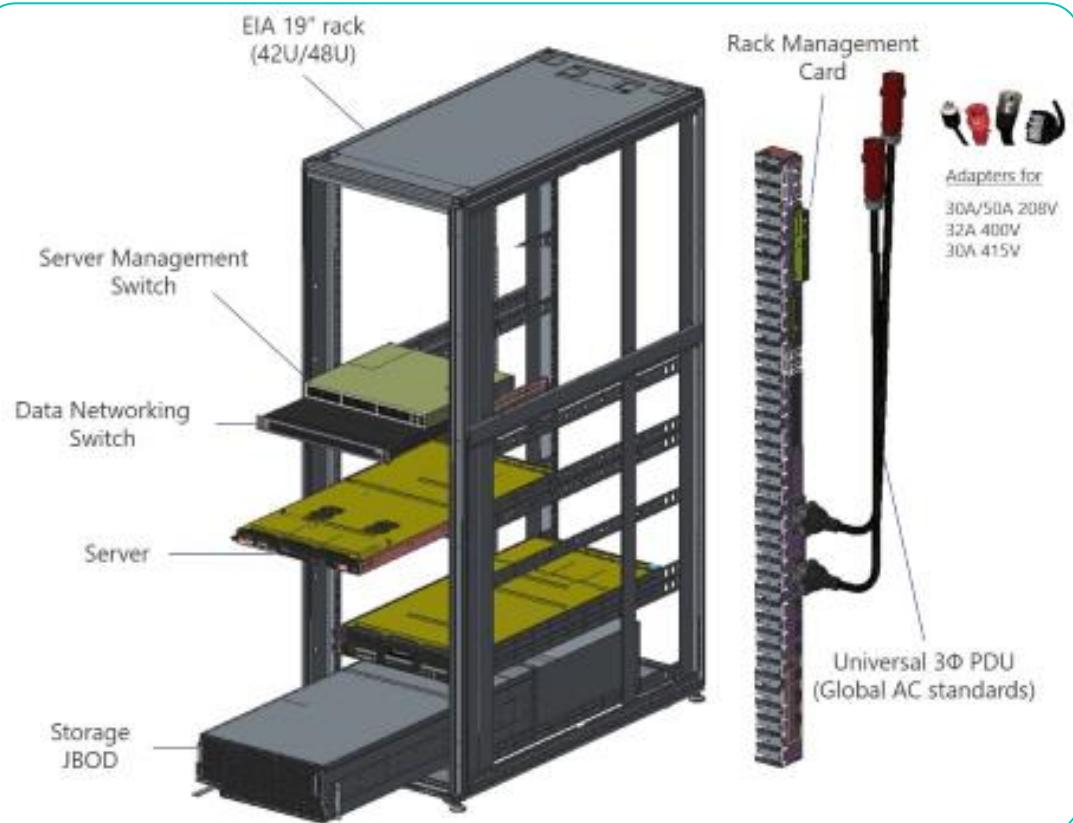
Blind-mated connectors
(12V Power, Ethernet, SAS,
Management)

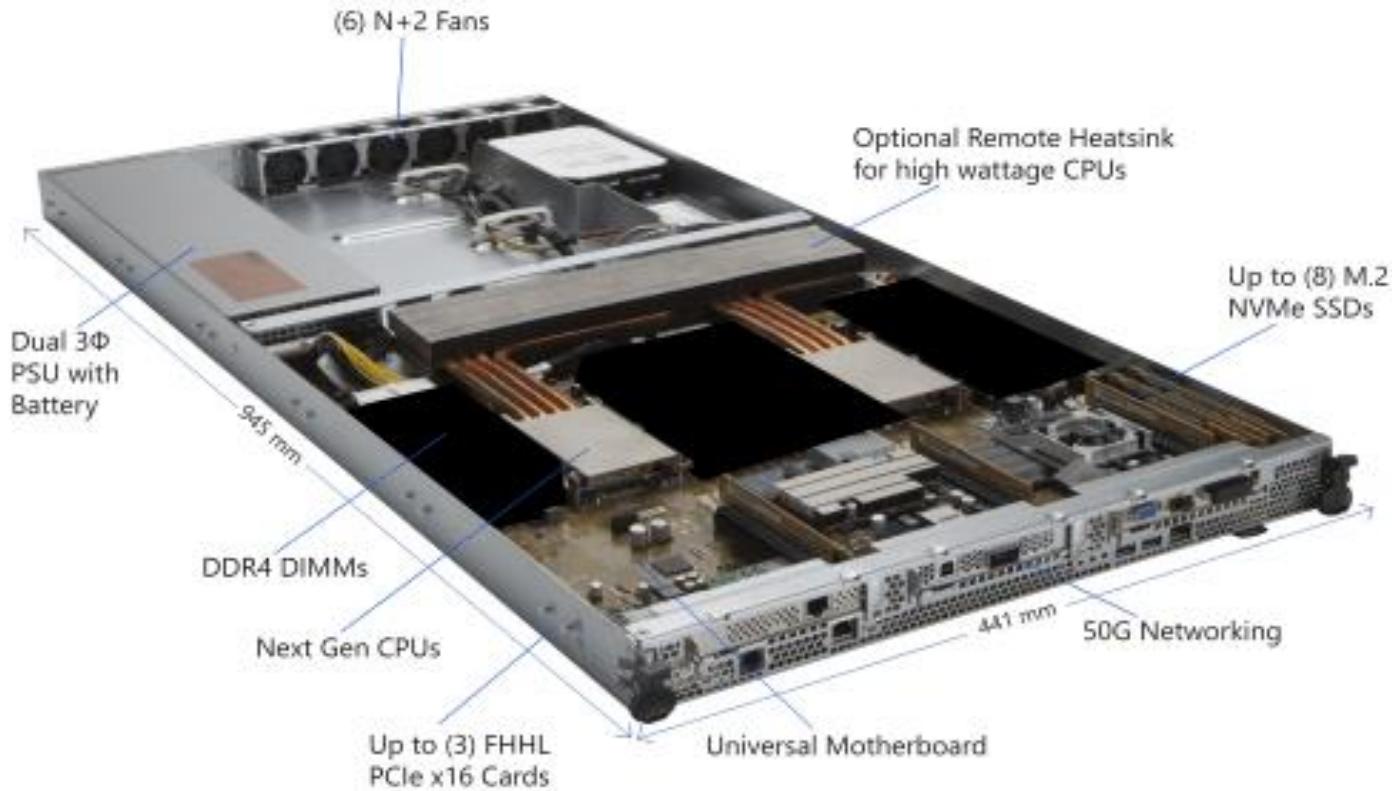


Arhitectura serverelor cloud se bazează pe o abordare modulară a șasiului de înaltă densitate, care permite partajarea eficientă a resurselor pe mai multe noduri de server.(JBOD-“Just a Bunch of Disks.”)

Un șasiu de 12U găzduiește 24 servere blade de consum

- Fiecare blade are **10 procesoare** Intel dual-core Xeon E5-2400.
- 96 servere/rack (3-4 șasiuri)
- X86 SoC (System-on-Chip) – management
- Aprox. 1770 km de cablu economisit la 120000 servere.
- Ignite 2017 – 3 milioane de servere (estimare: aprox 4 milioane in 2019)





Cloud Computing-Definiție

O Cloud este un mediu multi-tenant partajat caracterizat printr-un înalt nivel de automatizare, preferabil bazat pe o infrastructură virtualizată în care resursele IT livrate ca servicii pot fi provizionate și măsurate.

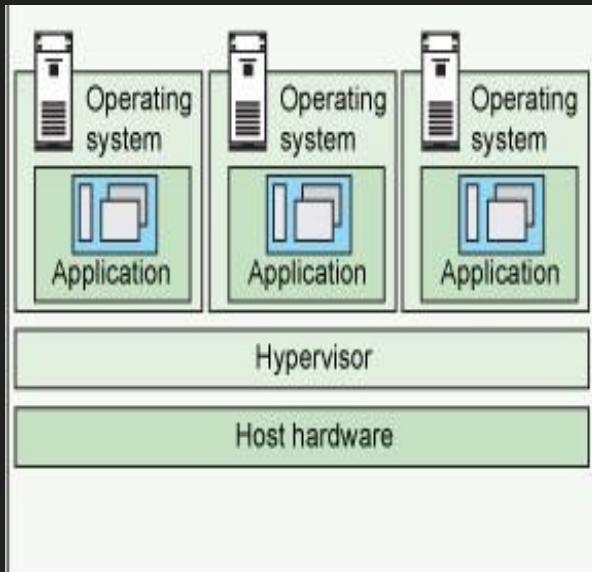
Virtualizare-Definitie

- Virtualizare reprezintă procesul prin care este creată o versiune virtuală (mai degrabă posibilă decât reală) a unor entități computationale, incluzând aici platforme hardware virtuale, dispozitive de stocare și resurse de rețea.
- Limbaje destinate descrierii structurilor hardware HDL(VHSIC-HDL, Very High Speed Integrated Circuit- Hardware Description Language) Verilog-Gateway Design Automation, 1985; RHDL-un HDL bazat pe Ruby etc.)

Virtualizarea

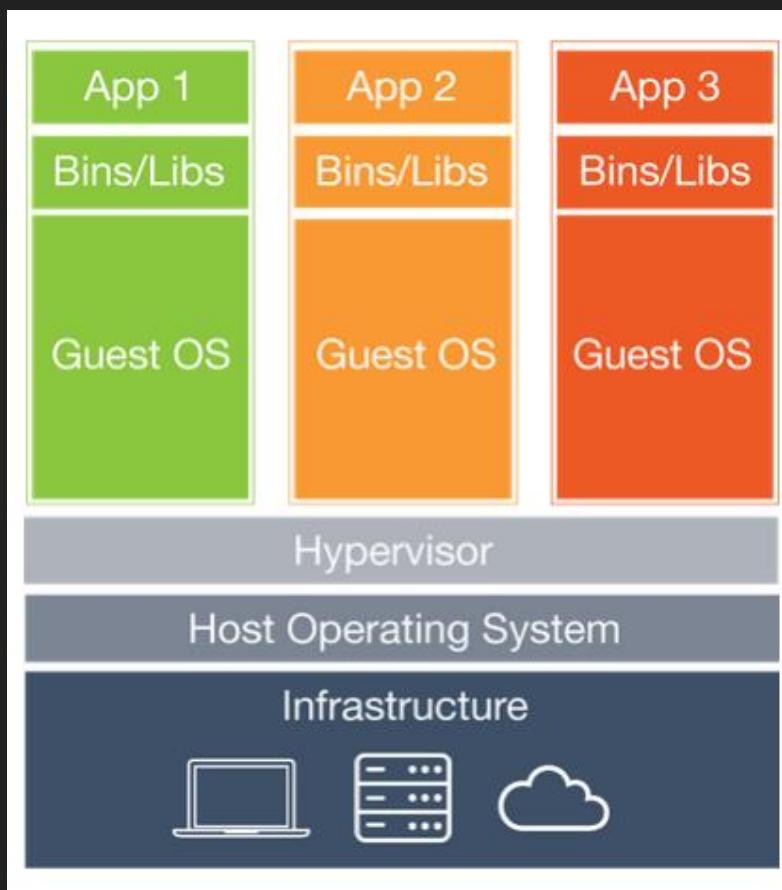
- **Virtualizare totală**- este un proces de natura computatională în care o mașină rulează pe o alta(datează din 1967 și a fost realizată pe un IBM CP-40)
- **Virtualizarea este principala metodă în baza căreia IT-ul se livrează ca serviciu**
- AMD-V și IVT(Intel Virtualization Technology) au permis obținerea unei virtualizări totale:
 - Partajarea unui sistem de calculatoare între mai mulți utilizatori
 - Emulare hardware pe o altă mașină
 - Izolarea utilizatorilor prin interfață de management

Hipervizor Tip I



- Hipervizoare-componentă software care permite partajarea resurselor asociate unei structuri de calcul fizice cu mai multe mașini virtuale:
- Hipervizor de tip I: este instalat după o strategie "bar-metal" în care prima componentă instalată pe mașina fizică este hipervizorul. Principalul avantaj al acestei arhitecturi este acela că hipervizorul comunică direct cu nivelul fizic partajat. Se spune că resursele fizice sunt **paravirtualizate** și dirijate către mașinile virtuale.
- Exemplu VMware ESX Server, Windows Hyper-V server,

Hipervizor Tip II



- Hipervizor de tip II(**hosted hypervisor**): Nu este instalat după un scenariu "bar-metal", contextul de lucru fiind un sistem de operare gazdă pentru hipervizor. Latența este minimală.
- Exemplu: un sistem de operare Windows Server 2016 poate găzdui un hipervizor VMware Workstation 10 cu pînă la 64 procesoare virtuale, storage 8Tb și 64Gb RAM sau Hyper-V V3.

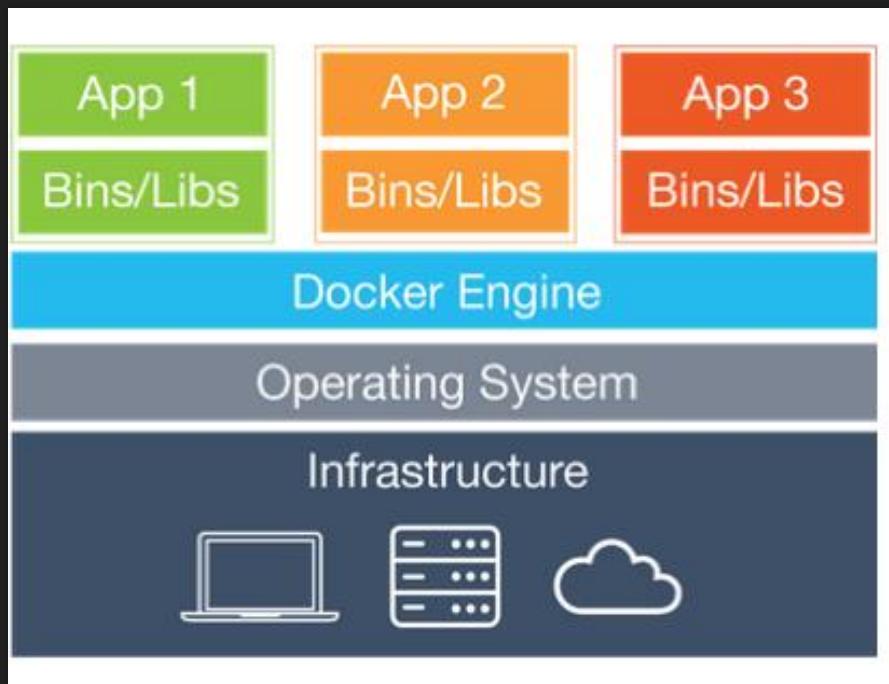
Paravirtualizare

- Permite lucrul cu sisteme de operare diferite pe același hardware.
- În virtualizare totală întreaga infrastructură hardware și software este virtualizată(BIOS, drivere,etc)
- În paravirtualizare managementul se realizează în conexiune cu un sistem de operare.
Paravirtualizarea este mai facilă virtualizării totale.
Exemplu:

Containere

Definiție.

- Containerele furnizează medii de runtime izolate destinate aplicațiilor utilizator. Întregul spațiu utilizator este expus ca un container și orice modificare a acestuia nu influențează celelalte containere.



Modele de implementare pentru Cloud Computing

- **Cloud Privat**-soluția este operată exclusiv de beneficiar și se poate exploata atât "on premises" cât și "off premises". Managementul poate fi realizat de organizație sau de un provider de cloud, "third-party"
- **Cloud Public**-serviciile de cloud sunt furnizate publicului larg sau organizațiilor și este deținut și administrat de organizații specializate în furnizarea de servicii de cloud
- **Cloud al Comunității**-modelul implică partajarea resurselor de către una sau mai multe organizații fiind susținut de o comunitate specifică caracterizată prin preocupari comune. Managementul poate fi realizat de o organizație sau de un provider de cloud "third-party"
- **Cloud Hibrid**-modelul cuprinde două sau mai multe tipuri de cloud și permite portabilitatea datelor și aplicațiilor între diferite tipuri de cloud.

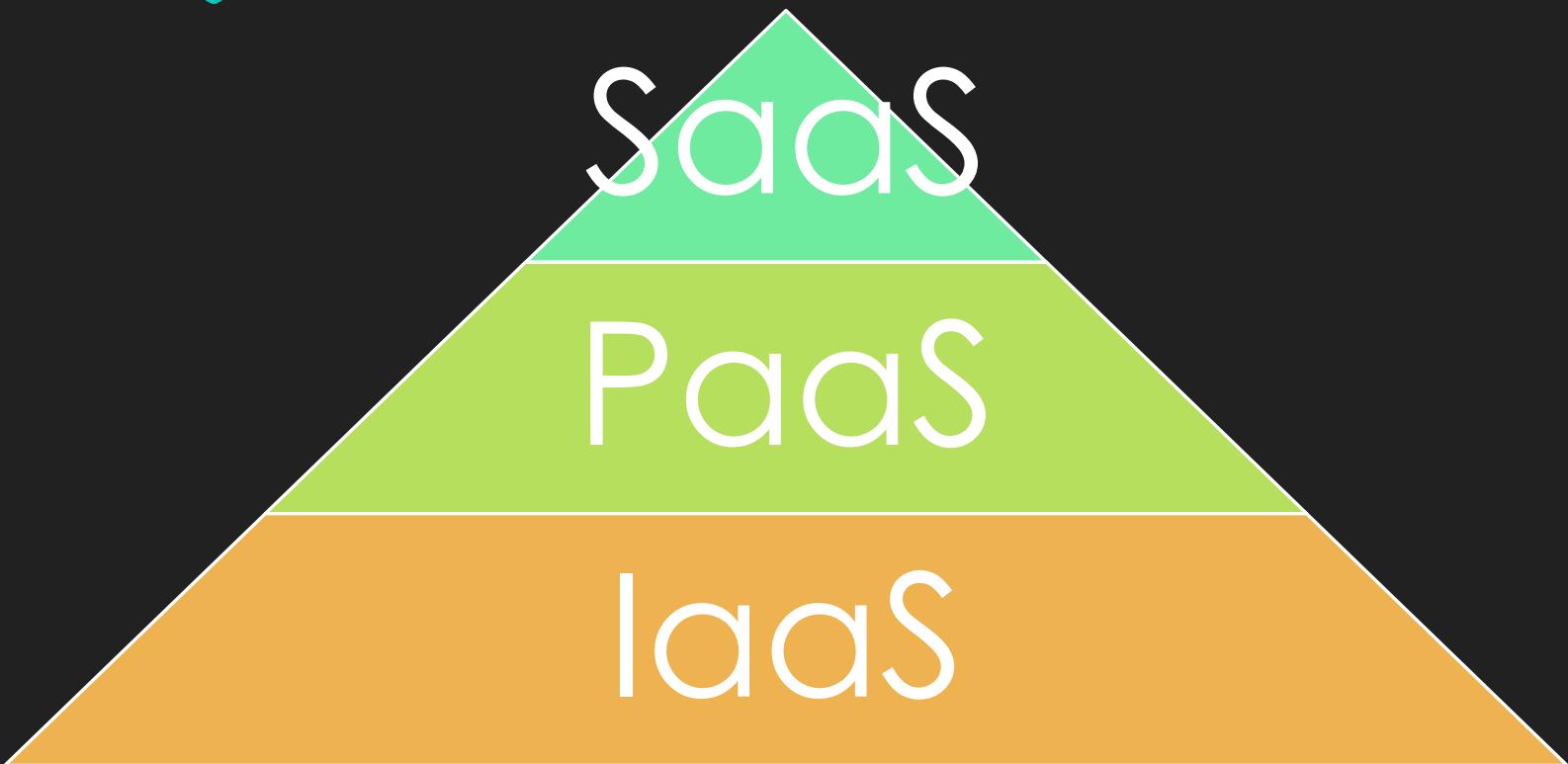
Caracteristici

- Furnizarea de servicii la cerere
- Resurse aggregate
- Elasticitate in alocarea resurselor
- Masurabilitatea resurselor alocate prin servicii
- Partajare multi-tenantă

Modele de servicii în Cloud

- IaaS (Infrastructure as a Service)-furnizează infrastructura IT ca serviciu către utilizatori-finali. Modelul permite utilizatorilor provizionarea proceselor, resurse de stocare și networking necesare utilizării de resurse software cum ar fi sistemele de operare, mediile de dezvoltare și aplicațiile. Utilizatorul are controlul acestor resurse și poate monitoriza volumul acestora.
- PaaS (Platform as a Service)-utilizatorul beneficiază de o platformă pentru dezvoltare software livrată ca serviciu peste infrastructura din cloud.
- SaaS (Software as a Software)-acest nivel furnizează funcționalități computationale ca serviciu. Aceste funcționalități pot fi accesate prin intermediul unui browser pe un thin client.

Cloud Computing nivelele de organizare



Resurse de Calcul

- Virtualizarea este realizată prin intermediul unui hipervizor care permite mașinilor virtuale să interacționeze cu resursele hardware.
- Resursele serverelor fizice sunt accesate de mașinile virtuale prin intermediul hipervizoarelor Hyper-V, VMware, sau Xen.
- Un server fizic poate gazdui mai multe mașini virtuale care pot accesa pachetul de resurse al suportului fizic în regim multi-tenant.

Servicii

- Serviciile în cloud vizează posibilitatea utilizării de componente disponibile în rețeaua vendorului de servicii
- Sufixul “as a service” implică:
 - Exploatarea fară bariere în a putea “consuma” serviciile
 - Scalabilitate extinsă
 - Capacitate de partajare extinsă(Multitenancy)
 - Posibilitate de a consuma serviciul de pe diferite platforme

Software as a Service

- SaaS este un model de distribuire a componentelor software stocate ca serviciu și accesate de consumatori prin Internet
- Pentru autori o garantie pentru protecție intelectuală
- Categorii de aplicații:
 1. Video conferință
 2. IT service management
 3. Contabilitate
 4. Web analytics
 5. Web content management
 6. CRM(Customer Resource Management)

- Aplicatii SaaS diferă de soluțiile de calcul distribuit. SaaS a fost dezvoltat în special pentru a folosi instrumente web(browser-ul) fapt ce îl face web-nativ.
- SaaS oferă posibilitatea livrării software-ului disponibil în comerț dintr-o locație centrală de unde clienții o pot accesa ori de câte ori au acces web.
- PaaS-SaaS este adesea folosit în combinație cu alte componente software. Când este utilizat ca o componentă a unui altă aplicații, acest lucru este cunoscut ca fiind un mash up sau un plugin.

Beneficii

- SaaS ușor de exploatat și întreținut.
- Ușor configurabil. Scaderea cheltuielilor cu personalul IT de înaltă calificare
- Securitate ridicată.
- Lățimea de bandă în permanentă creștere permite o exploatare performantă

Obstacole

- Probleme cu organizațiile care exploatează aplicații specifice
- Aplicații open-source ieftine rulate pe echipamente ieftine pot conduce la solutii avantajoase atit financiar cit si tehnic

Multumesc

CONF.DR. CRISTIAN KEVORCHIAN
UNIVERSITATEA DIN BUCURESTI
FACULTATEA DE MATEMATICA ŞI INFORMATICĂ

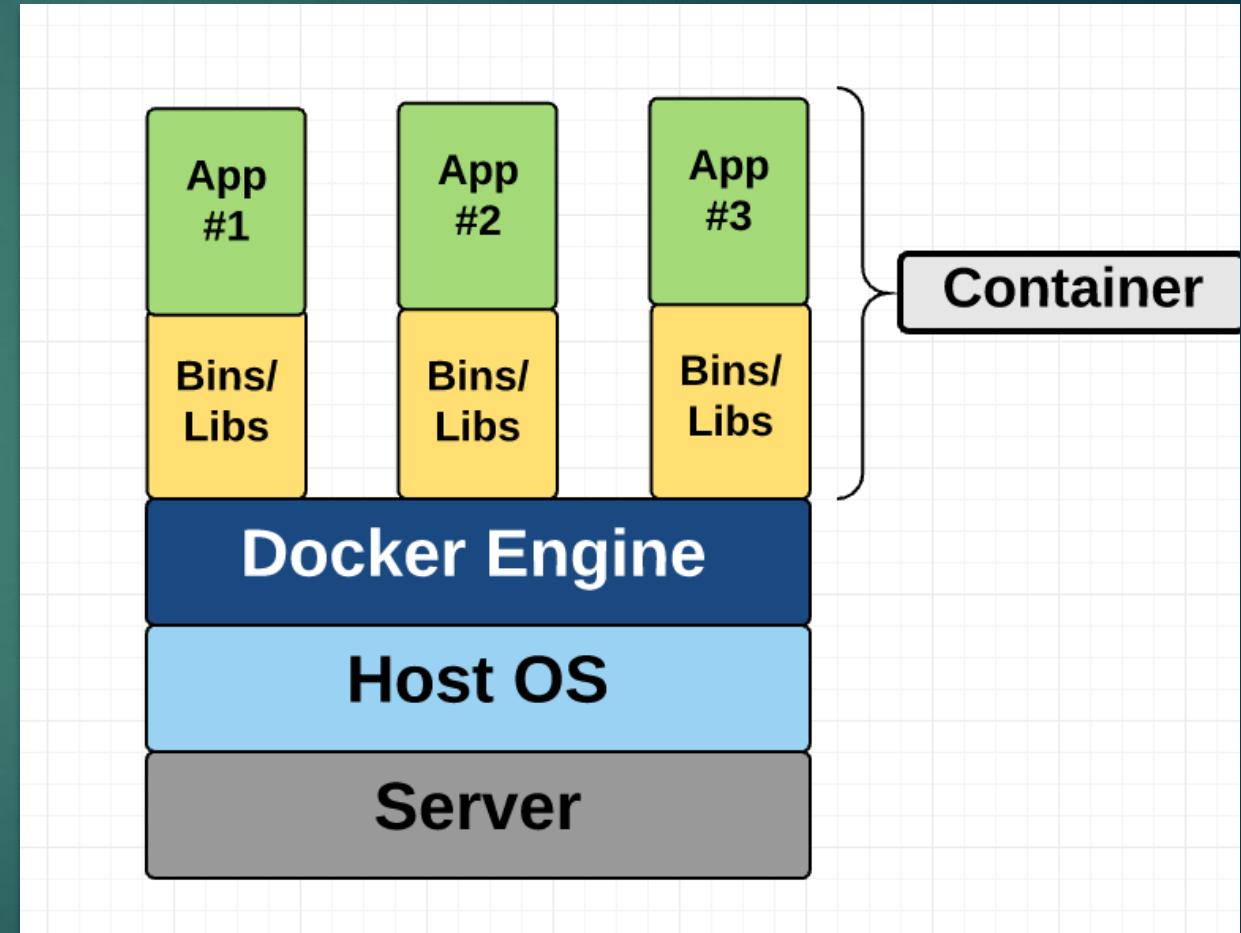
Containere

Virtualizarea Infrastructurii IT

- Virtualizarea platformelor și componentelor hardware este un proces de abstractizare logică necesară operationalizării diverselor sisteme de operare.
- ▶ Platforma abstractă de calcul rezultată în urma virtualizării ascunde caracteristicile fizice ale platformei de calcul inițiale(fizice) prezentând-o în schimb drept o platformă virtuală de calcul care conservă modelul computational al celei inițiale

Containere

- ▶ Containerele sunt cunoscute drept forme de virtualizare la nivelul sistemului de operare și reprezintă o abordare de complexitate redusă a virtualizării, care oferă un minim necesar de resurse de calcul unei aplicații pentru a funcționa corect într-un context computational dat. Într-un fel, ele pot fi considerate medii virtuale minimalistice care nu rulează pe un hypervisor.



Pe o singură
mașină-gazdă
pot rula mai
multe containere.

- ▶ Containerele izolează aplicațiile unele de altele pe un sistem de operare partajat.
- ▶ Aplicațiile containerizate rulează la nivelul superior de abstractizare a containere-lor, care, la rândul lor, rulează pe un sistem de operare (Linux sau Windows). Prin urmare, containerele au o prezență semnificativ mai mică decât imaginile unei mașini virtuale.
- ▶ Fiecare container poate găzdui o aplicație web sau un serviciu.

Containere-generalități

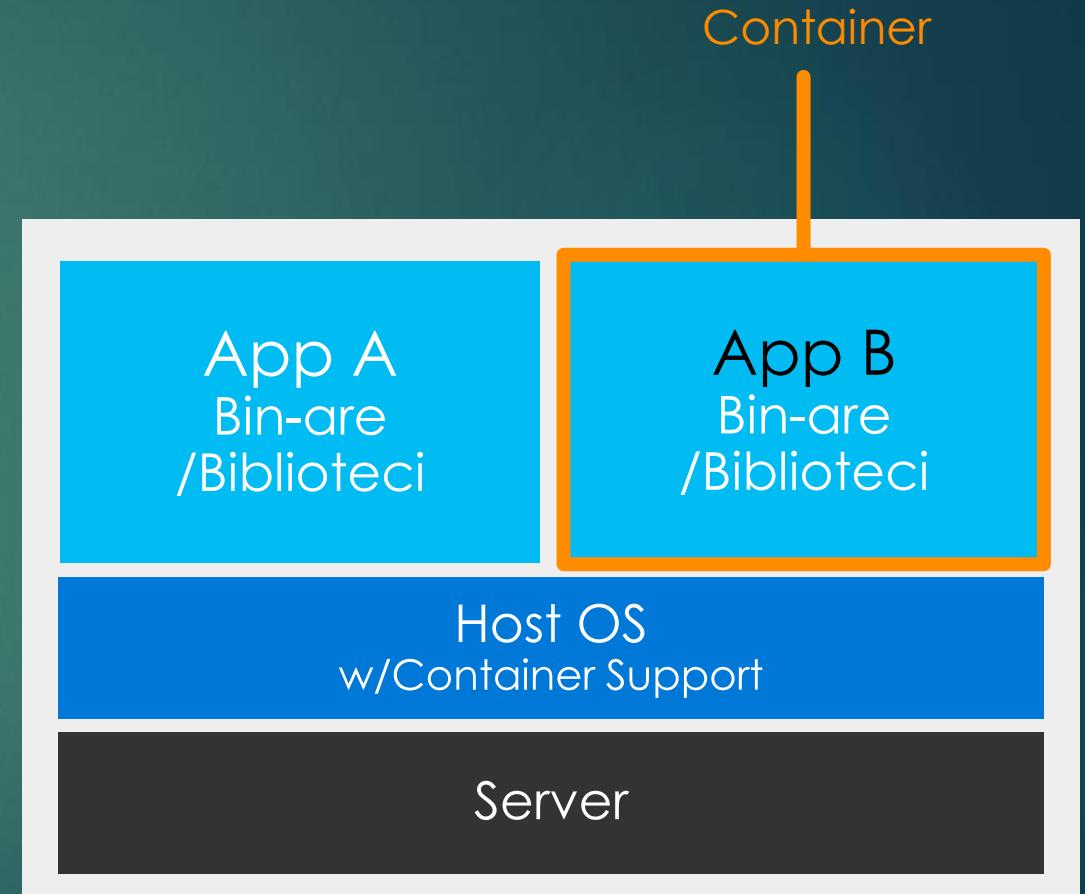
Dependențe: Fiecare aplicație are propriile dependențe care includ atât elemente software (servicii, biblioteci) cât și hardware (CPU, memorie, storage).

Virtualizarea: Motorul containerelor este un sistem de management și orchestrare al acestora, echivalent cu virtualizarea prin care se pot izola dependențele pentru fiecare aplicație prin "ambalarea" lor în containere virtuale.

Partajarea SO gazda: Procesele în containere sunt izolate de celelalte containere în spațiul utilizatorilor, dar partajează kernel-ul cu gazda și cu celelalte containere.

Flexibilitate: Diferențele dintre sistemul de operare și infrastructura de bază se abstracționează, simplificând abordarea prin "deployment oriunde"..."..

Rapid: Containerele pot fi create aproape instant permitând o scalare elastică asociată cererii.



Containere

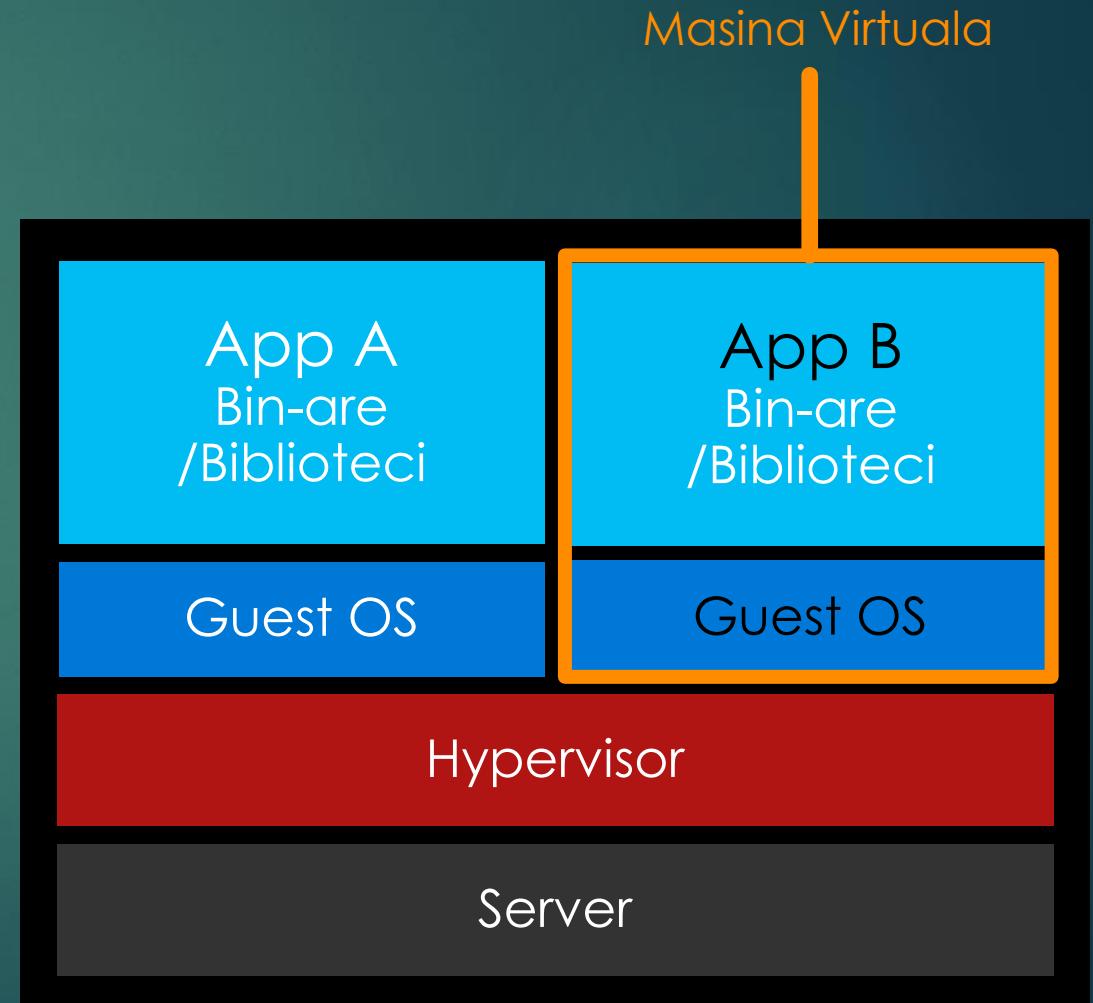
Diferențe față de masini virtuale

Dependențele: Fiecare app virtualizată include aplicația în sine, cerând binare și biblioteci dar și un SO "oaspete", care poate integra mai mulți GB de date.

SO independent: Fiecare MV poate avea un sistem de operare diferit de al altor MV, împreună cu un sistem de operare diferit de SO gazdă în sine.

Flexibilitate: MV pot fi migrate în alte locații pentru a balansa resursele utilizate și pentru menținerea locației fără "downtime".

Securitate: Nivel înalt de izolare securizată a resurselor pentru sarcinile cheie virtualizate.



Containere incluse în MV

Scenarii multiple de implementare a aplicațiilor

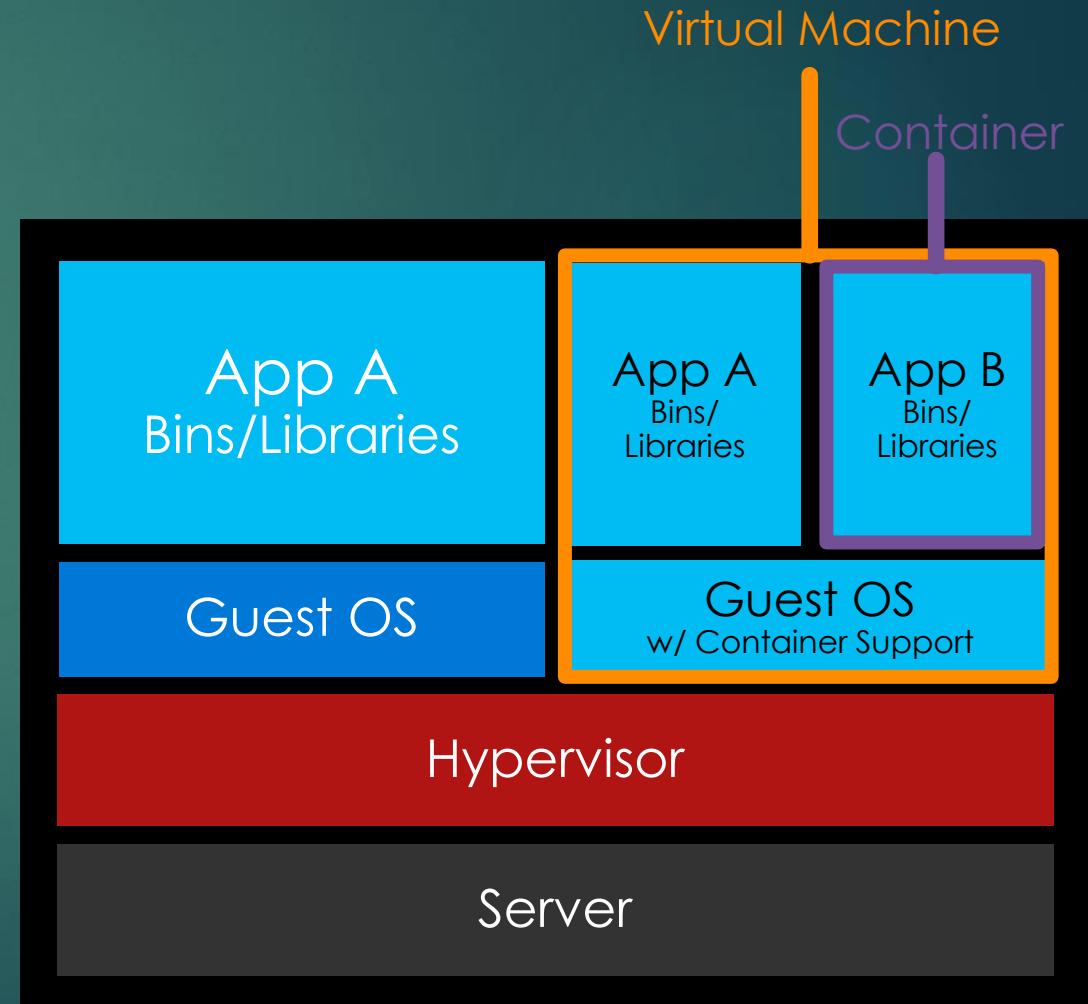
Containere în MV: Prin combinarea containerelor cu MV, utilizatorii pot implementa mai multe sisteme de operare în MV diferite, iar în interior pot implementa mai multe containere în cadrul acelor SO.

Prin combinarea containerelor cu MV, ar fi necesare mai puține MV pentru a suporta un număr mai mare de aplicații.

Mai puține MV ar avea ca rezultat o reducere necesarului de stocare.

Fiecare MV ar susține mai multe aplicații izolate, crescând astfel densitatea globală a soluției.

Flexibilitate: Rularea containerelor în interiorul MV permite implementarea de soluții precum "live migration" pentru utilizarea optimă a resurselor și întreținerea gazdei.



Containere Windows Server

Structura și funcționalitățile

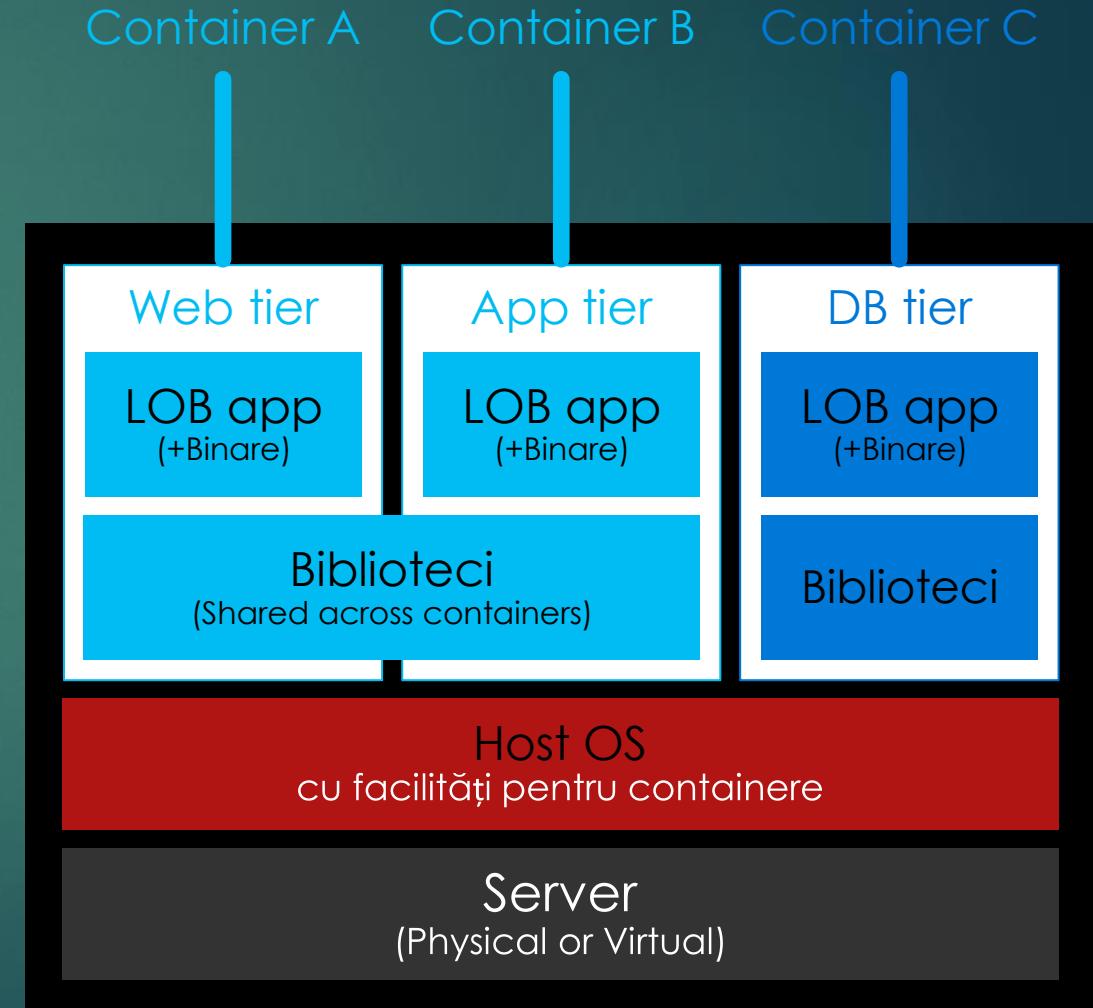
Build: Dezvoltatorii vor folosi instrumente cunoscute de dezvoltare, cum ar fi Visual Studio, Eclipse pentru a scrie aplicații și a rula în containere.

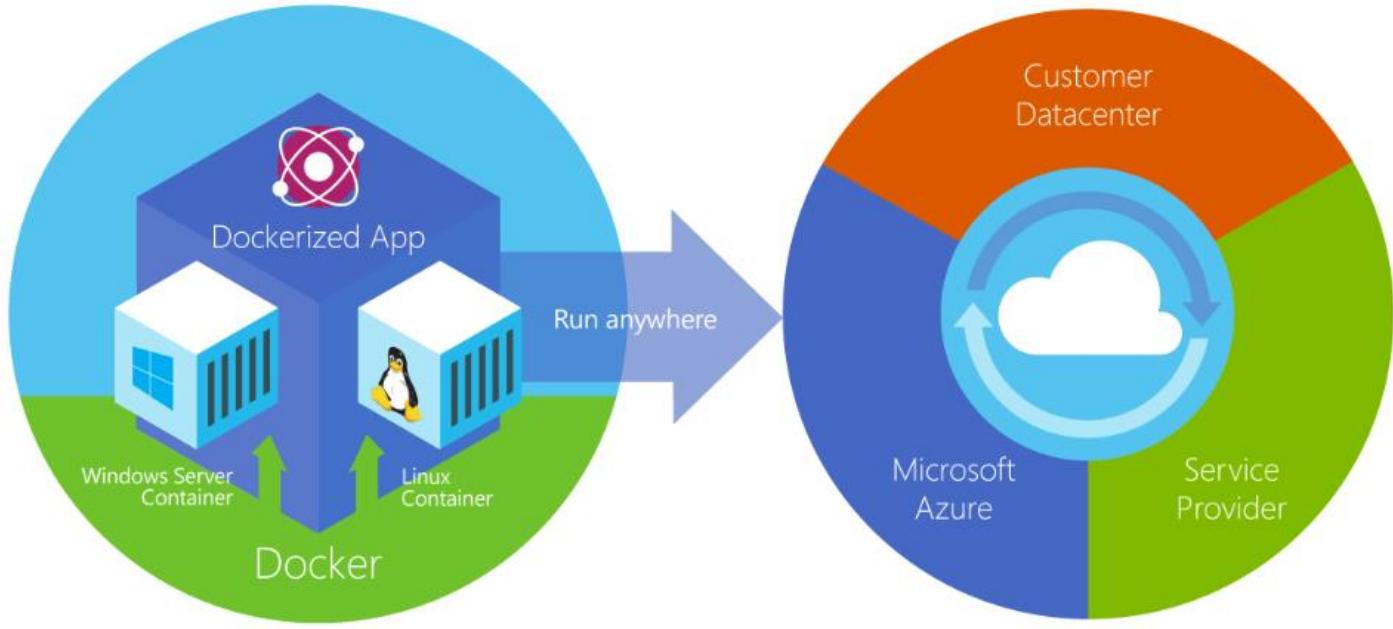
Prin construirea de aplicații modulare care folosesc containerele, modulele pot să fie actualizate independent – cu planuri independente.

Run: Funcționalitățile containerelor sunt built-in în Windows Server 2016.

Management-ul: Deployment-ul și managementul containerelor se realizează cu PowerShell sau Docker.

Resursele: Consumul de CPU și resursele de memorie pe container în funcție de capacitatea de stocare și banda transzitată.





Docker este un proiect open-source pentru automatizarea implementării aplicațiilor în containere portabile și care pot funcționa în cloud sau local. Docker este, de asemenea, o companie care promovează și dezvoltă această tehnologie, colaborând cu furnizori de cloud și SO.

Terminologie Docker

- ▶ **Imaginea containerului:** Un pachet cu toate dependențele și informațiile necesare pentru a crea un container. O imagine include toate dependențele (cum ar fi frameworkurile) plus informații de configurare a mediului de execuție asociat containerului. O imagine derivă din mai multe imagini de bază care sunt elemente de formare a sistemul de gestiune a fisierelor. O imagine este imutabilă odată ce a fost creată.
- ▶ **Container:** O instanță a unei imagini Docker. Un container este un mediu de runtime pentru o singura aplicație, proces sau serviciu. Când scalăm un serviciu sunt create o serie de instanțe ale unui container din aceeași imagine. Un job batch poate crea multiple containere din aceeași imagine prin transmiterea de parametri diferenți către fiecare instanță.
- ▶ **DockerFile:** Un fișier text ce conține instrucțiuni pentru generarea unei imagini Docker.
- ▶ **Build:** Construcția unei imagini bazate pe informațiile și contextul descris în DockerFile.

Containerele sunt centrate pe procese DevOps



Rezumând, putem afirma că:

Prin containerizarea aplicațiilor vechi utilizând varianta de containerizare Windows Server, obținem o consistență și un management îmbunătățit al echipelor de dezvoltatori și testeri, pe de-o parte și cele de implementare, pe de altă parte – într-un mediu unitar DevOps - fără a modifica aplicația..

Containere Hyper-V

Structură și funcționalități cheie

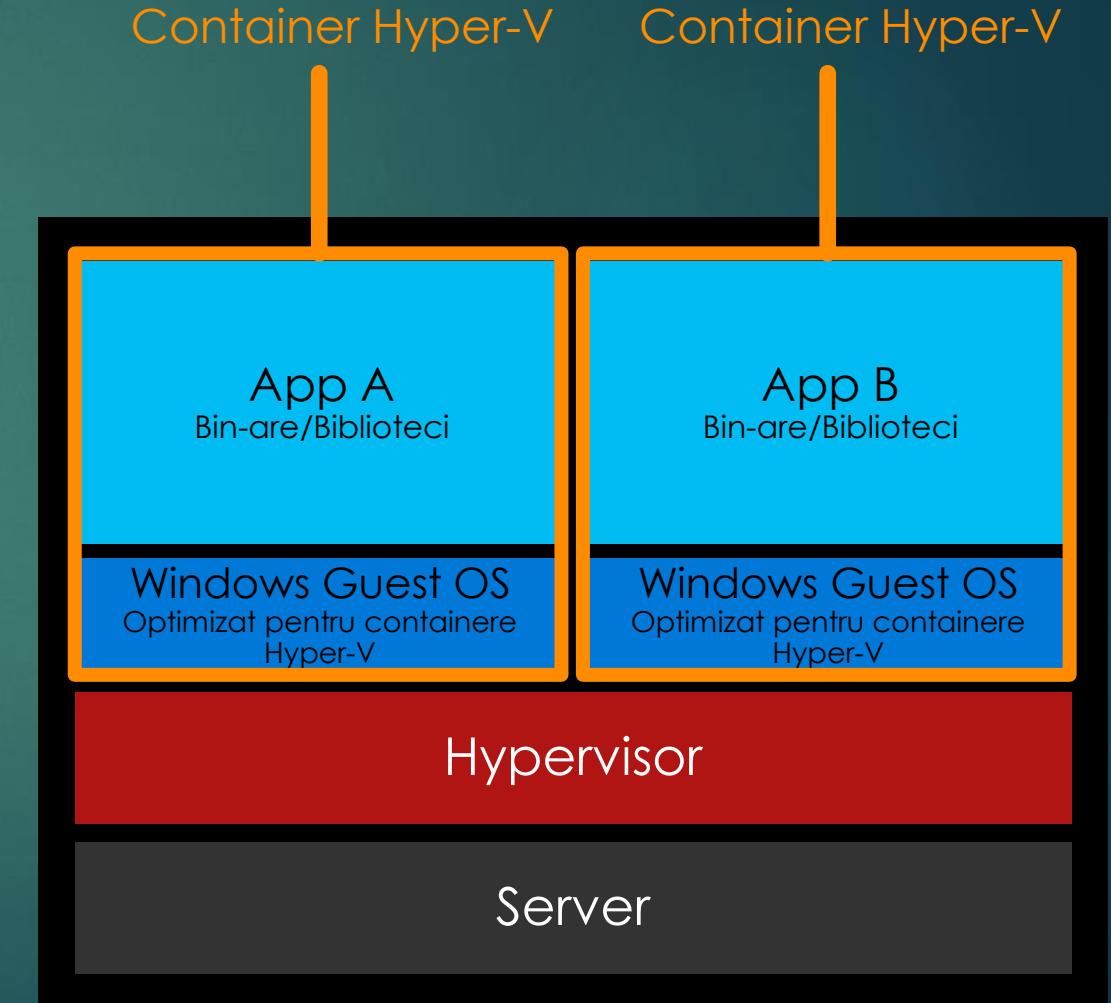
Consistență: Containerele Hyper-V utilizează aceleași API-uri ca și containerele Windows Server, asigurând coerență între seturile de instrumente de gestionare și implementare.

Compatibilitate: Containerele Hyper-V utilizează aceleași imagini ca și containerele Windows Server.

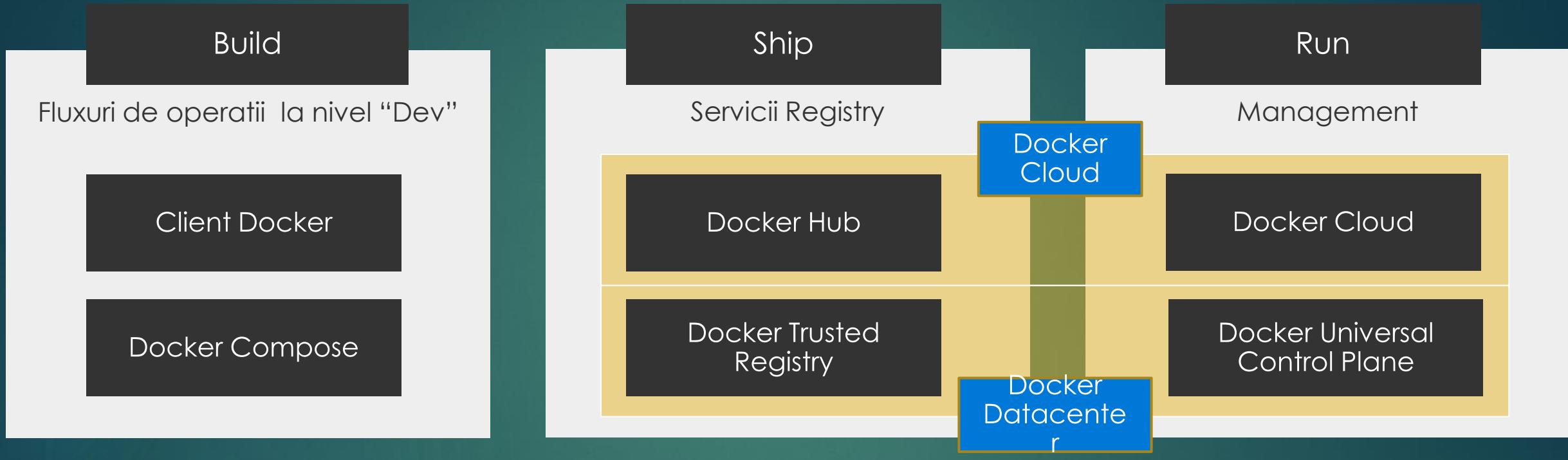
Izolare Puternică: **Fiecare container Hyper-V are propria copie dedicată a kernel-ului.**

Foarte sigur: Construit cu o tehnologie de virtualizare Hyper-V dovedită a fi foarte sigură.

Optimizată: Layer-ul de virtualizare și sistemul de operare au fost special optimizate pentru lucrul cu containere.



Componente Docker



Sistem de Operare

Infrastructură

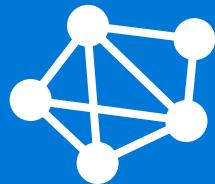
Plug-in-uri

Containerele reprezinta un excelent mediu pt.:

Calcul
Distribuit

$$f(x)$$

Scalare



Baze de
date



Task-uri

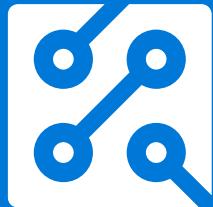


Web App



SO pentru lucrul cu containere

Nano Server



Bine optimizat



Aplicatii
"Born-in-the-cloud"

Server Core



Larg Compatibil



Aplicatii traditionale



Windows Server

Instrumente la dispozitie

Managementul containerelor



PowerShell



Docker



Others

Medii de dezvoltare



Altele...

Tehnologii bazate pe containere



Windows Server



Linux

Microsoft cloud



Azure



On Prem



Service Provider

Instrumente pentru Dev

Limbaje și Framework-uri pentru dezvoltare



Microsoft Cloud



Azure

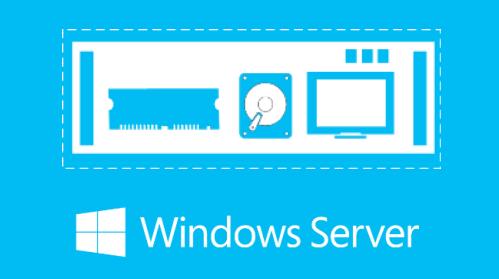


On Prem



Service Provider

Masini Virtuale



Windows Server



Linux

Service fabric



Tehnologii Bazate pe Containere



Windows Server



Linux

Comparării între tehnologii

Comparări între containere și VM

	Windows Server containere	Linux containere	Masini Virtuale
Imaginea de Bază	Acelasi cu gazda	Acelasi cu gazda	Oricare Windows/Linux
Securitate multi-tenant	Nu	Nu	Da
Managementul resurselor	Da	Da	Da
Densitate	Mare	Mare	Mica
Timpul de pornire	Scurt	Scurt	Lung
Amprenta pe disc	Mica	Mica	Mare
Compatibilitatea aplicațiilor	Medie	Medie	Mare

Sistemul de operare: Containerele partajeaza acelasi SO ca si sistemul gazda, dar poate rula in cadrul unei MV pentru a permite o flexibila utilizare a OS.

Securitate: Mașinile virtuale oferă un nivel mai ridicat de protecție împotriva amenințărilor, cum ar fi exploit-urile kernel-ului.

Proprietăți MV : În timp ce containerele au un timp de pornire mai rapid, mașinile virtuale beneficiază de caracteristici precum "live migration".

Compatibilitatea Aplic.: Pentru maximizarea beneficiului, aplicatiile ar trebui să fie proiectate, arhitecturate si

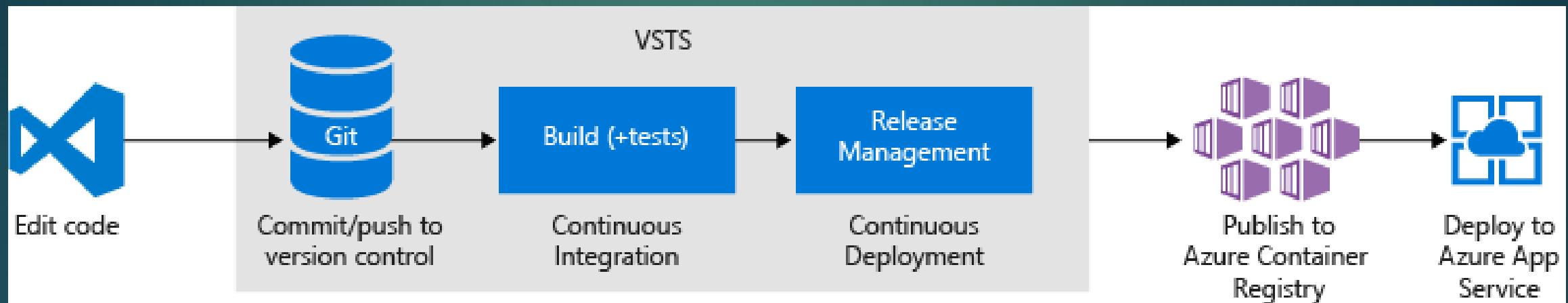
Livrare și Integrare Continua

- **Livrarea continuă** (LC) este un concept al ingineriei software în care echipele produc software în cicluri scurte, asigurându-se că software-ul dezvoltat este fiabil și livrabil pentru utilizare în orice moment. Acest fapt implică scrierea, testarea și deployment-ul de software la perioade foarte scurte.
- **Integrarea continuă**(IC) - este o practică în dezvoltarea software care cere dezvoltatorilor să integreze codul(2-3 ori/zi) într-un repository comun. Fiecare check-in este apoi verificat de un build automat, permitând echipelor de dezvoltatori să detecteze într-o etapă incipientă eventualele probleme.

Considerații pe marginea IC și LC

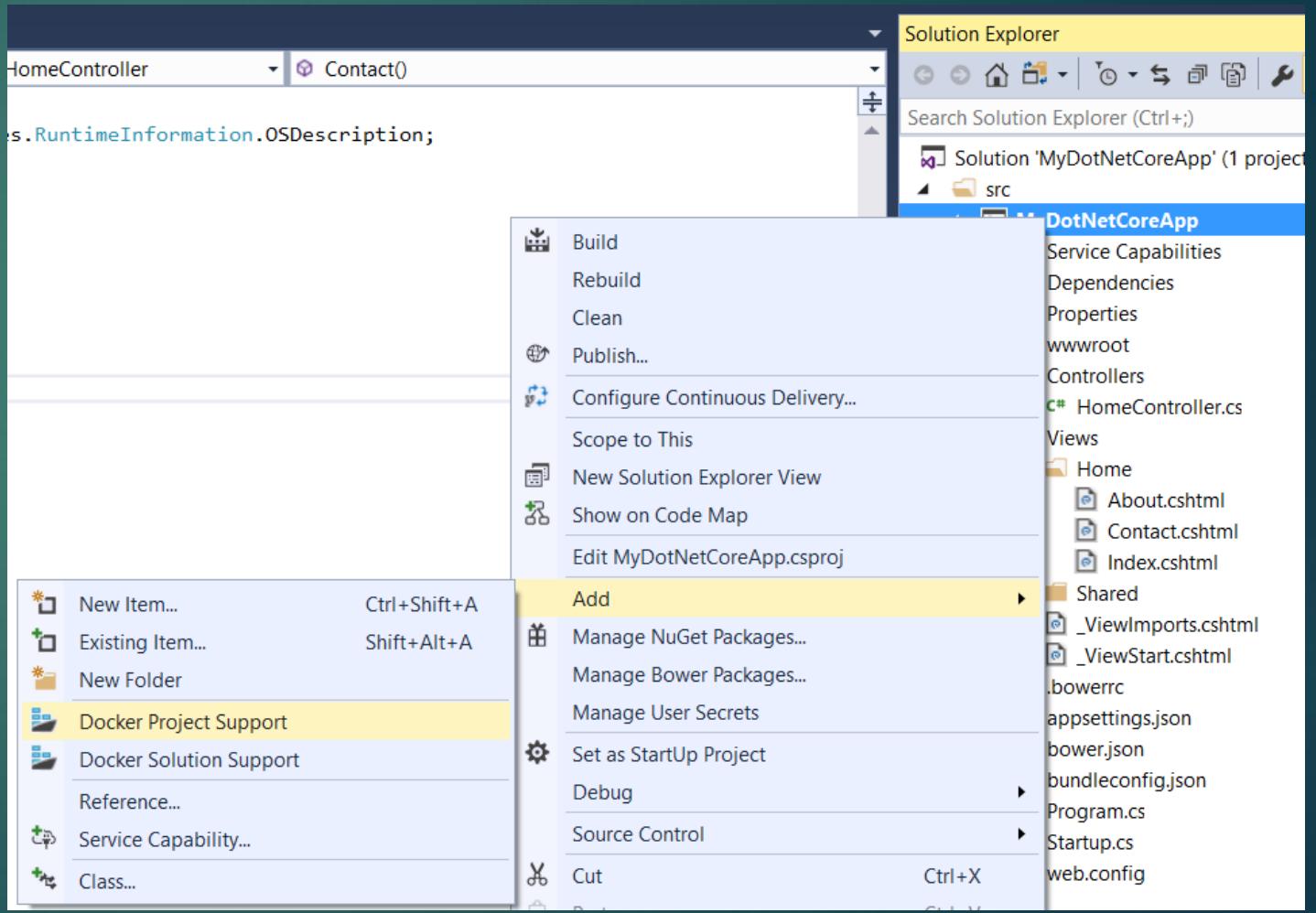
1. Livrarea continuă este o prelungire a integrării continue.
2. Este nevoie de o maturizare a proceselor ALM de aşa manieră încât să puteţi reacţiona în timpul cel mai scurt să schimbări semnificative ale produsului software.
3. Acest lucru se traduce prin necesitatea automatizării proceselor aşa încât să puteţi implementa realesuri la perioade foarte scurte

ALM pentru IC și LC



Configurarea Build-ului pentru LC

- container registry
- orchestrator (KUBERNETES)

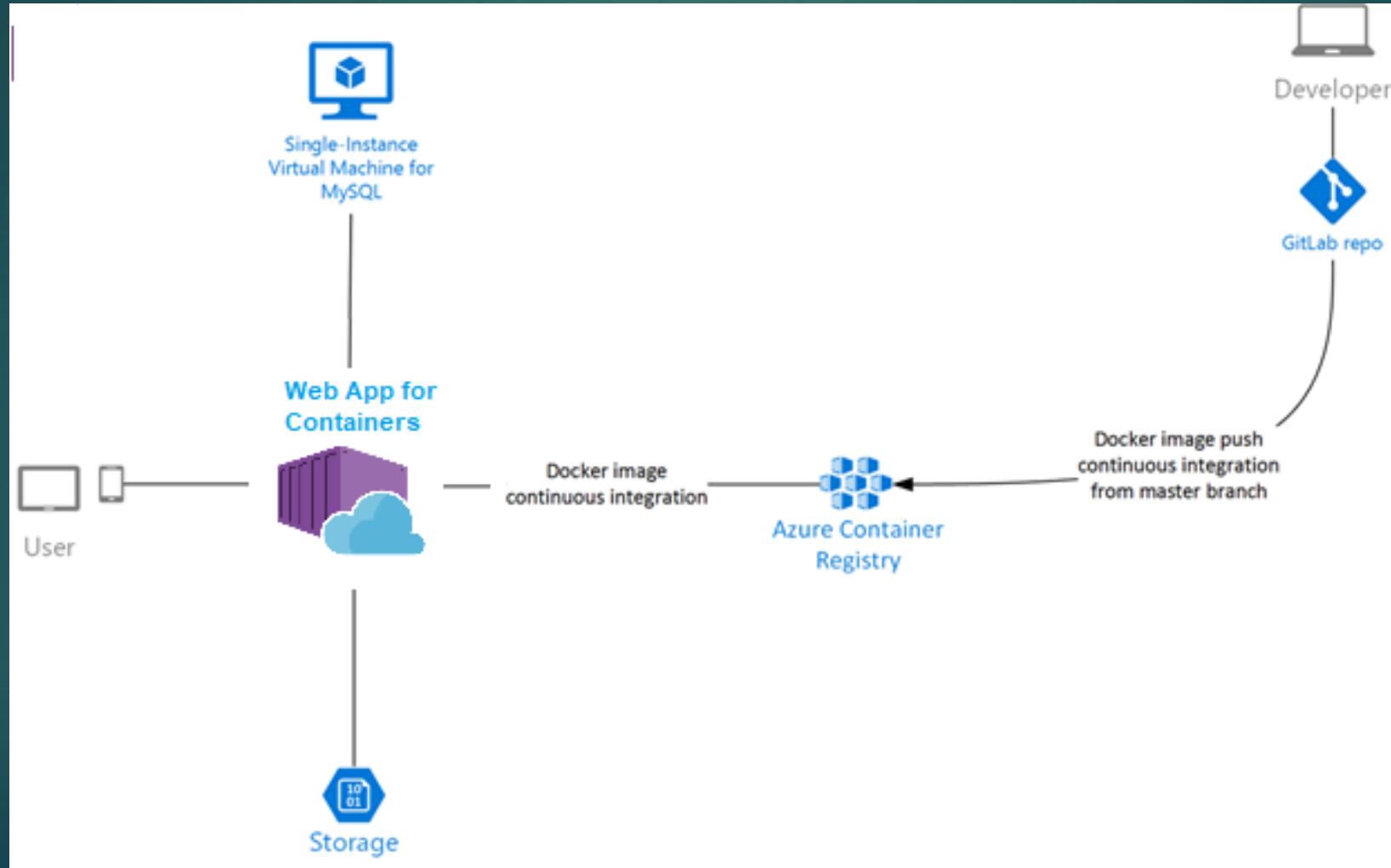


DEVOPS

DevOps (dezvoltare și operațiuni) - este un concept în dezvoltare de software utilizat pentru a desemna un tip de relație agilă între dezvoltare și operațiunile IT ale unei entități de business.

Scopul DevOps este de a schimba și de a îmbunătăți relația prin promovarea unei mai bune comunicări și colaborări între aceste două componente importante care sprijină business-ul.

Aplicații WEB Pentru containere



Container registry

- ▶ "Registry" este o aplicație stateless server, foarte scalabilă, care stochează și distribuie "agini Docker". Este open-source, sub licența deschisă Apache.
- Container Registry Privat
- Container Registry Public

"AZURE Container Registry"

- ▶ **Azure Container Registry** permite să stocarea și gestiunea de imagini pentru toate tipurile de implementări de containere.
- ▶ **"Azure Container Instances"** permite implementarea de containere Docker pe infrastructura Azure fără a utiliza mașini virtuale sau pentru a adopta servicii de nivel superior.
- ▶ O **îmagine Docker** este construită dintr-o serie de straturi. Fiecare strat reprezintă o instrucțiuțe a fișierului Docker. Fiecare strat, cu excepția ultimului, este numai pentru citire.
- ▶ Exemplu:

FROM UBUNTU:15.04

COPY . / app

RUN make / app

CMD python /app/app.py

Containere

- ❖ O instanță a unei imagini este un container.
- ❖ O imagine, este formată dintr-o familie de starturi aşa cum apar descrise în Docker file.
- ❖ Dacă pornim această imagine, avem un container care rulează această imagine.(mai multe containere pot rula asociat aceleiași imagini).
- ▶ O lista a imaginilor se poate obține cu "docker images", dar și containerele cu "docker ps -a"
- ▶ Deci, o instanță care rulează o imagine este un container.

Configurare CD

Configure Continuous Delivery

Create the Team Services and Azure resources needed to continuously deliver your app

User account:

Team project:

Repository:

Azure subscription:

Container registry:

Azure container service:

Enabling continuous delivery will:

Create a build definition for your repository

Create a release definition that runs on each successful build, and then delivers to the 'Dev' environment

Initiate a build-and-release run now, and whenever you push to this repository

[Review Azure Container Service pricing information](#)

[What is continuous delivery?](#)

This process may take several minutes to complete

OK

Cancel

Cloud Storage

Conf.dr. Cristian KEVORCHIAN

Facultatea de Matematică și
Informatică

ck@fmi.unibuc.ro

Storage as a Service

- Stocarea ca serviciu este un model de business în care o companie furnizoare închiriază spațiu de stocare în infrastructura proprie unei alte companii sau persoane fizice.
- Avantajele oferite de “Storage as a Service” constau în reducerea costurilor de: personal, hardware suport de stocare dar și o capacitate de scalare superioară în condiții de disponibilitate ridicată.

Abstracțiile care fundamentează Azure Storage

- **BLOB** - Oferă o interfață simplă pentru stocarea fișierelor împreună cu metadatele asociate acestora.
- **Fișiere** – Disponibilizate pentru acces distribuit
- **Tabele** - Oferă stocare structurată de tip noSQL masiv scalabilă. O tabelă este o familie de entități care conțin o mulțime de proprietăți. O aplicație poate gestiona entitățile și poate interoga oricare dintre proprietățile stocate într-o structură relațională.
- **Cozi** - Oferiți o stocare și o livrare sigură a mesajelor pentru o aplicație pentru a construi fluxul de lucru(cuplat și scalabil) între diferitele părți (roluri) ale aplicației .

Contul din Azure Storage

Contul de stocare, permite lucrul cu:

- BLOB-uri
- Entități
- Mesaje

în limitele a 5PiB(pebibytes 2^{50} bytes) în
5200 de conturi de storage/subscriptie.

Windows Azure Storage

Abstarctiile serviciului de stocare

Blob-uri

Sistem de gestiunea fisierelor in
cloud

Table

Stocare structurata masiv
scaabila

Queues

Stocare si livrare de mesaje

File Storage

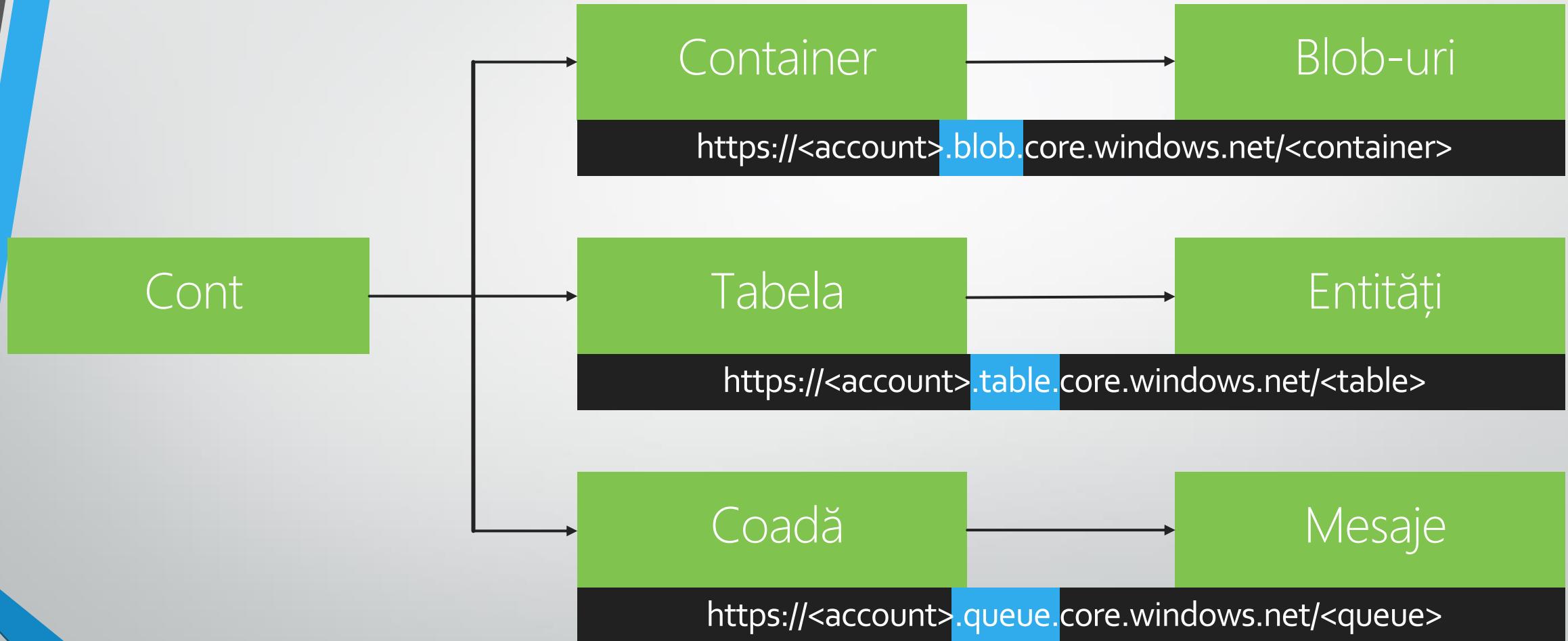
Volume NTFS pentru aplicatii in
Windows Azure

Utilizare facila

Simplu de utilizat REST API-urile
si Librarie-iile

Exista NTFS API-uri for Windows
Azure Drivere

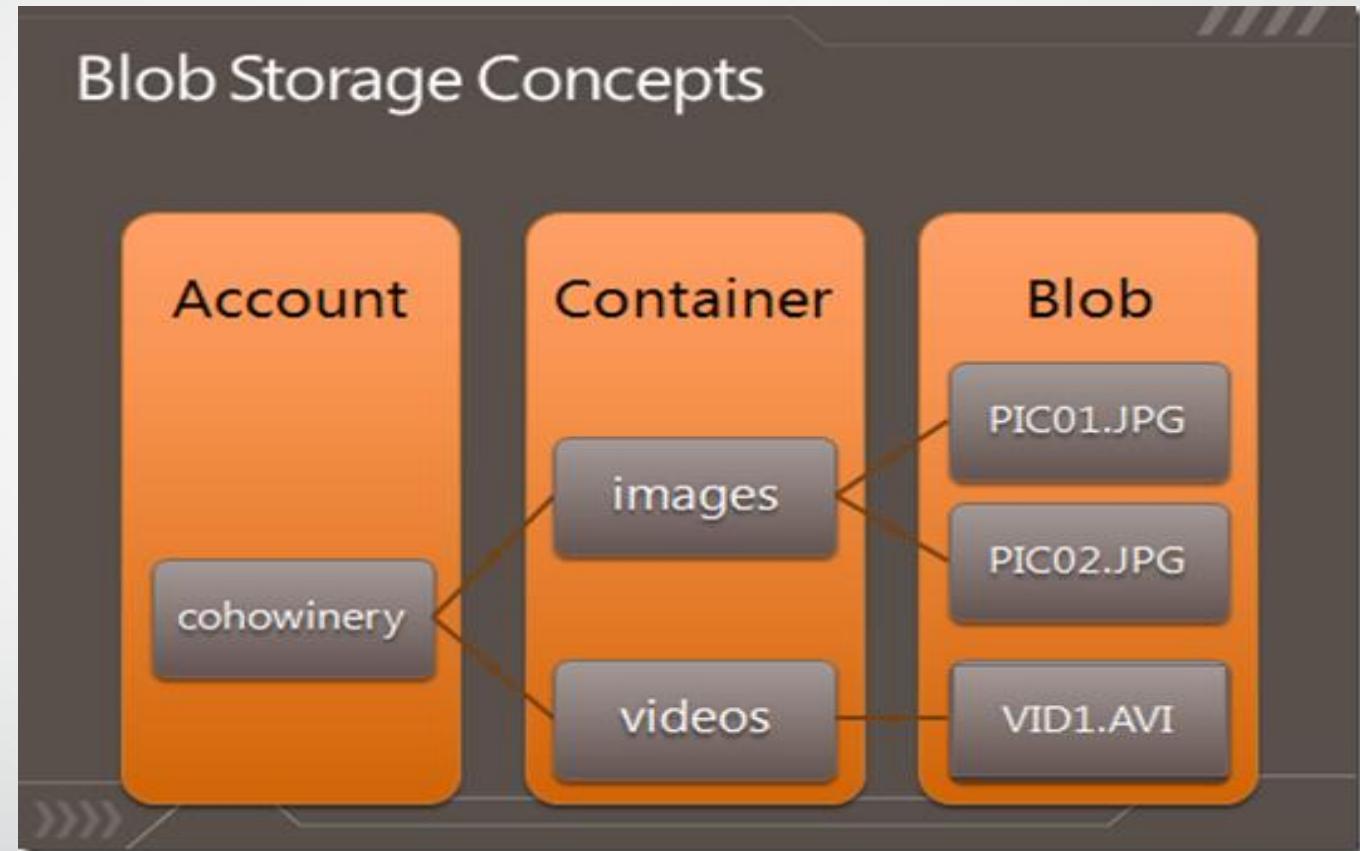
Concepte-Windows Azure Data Storage



BLOB-uri în Windows Azure

Conceptele de stocare ale sistemului Windows Azure BLOB sunt:

- Cont de stocare
- Container
- BLOB-uri



BLOB-uri

- Blob-urile sunt stocate în Containerele Blob
- Blob-urile pot avea **metadate** asociate de forma **<nume, valoare>** și au o dimensiune de până la 8KB/blob.
- URL-ul pentru un BLOB este structurat după cum urmează:
 - `http://<cont>.blob.core.windows.net/<container>/<numeBLOB>`

BLOB-uri Windows Azure

Un sistem de gestiunea fisierelor durabil si scalabil in cloud.

Container Blob

- Oferă o grupare de seturi de bloburi.
- Numele containerului este cuprins de cont.
- Politicile de distribuire sunt setate la nivelul containerului, unde un container poate fi setat ca privat sau să fie accesibil publicului.
- Când un container este setat la "**Public**", întregul său conținut poate fi citit de oricine fără a fi necesară autentificarea. Atunci când un container este privat, este necesară autentificarea pentru a accesa blob-urile din container.
- Containerele pot avea, de asemenea, metadate asociate cu acestea. Metadatele sunt sub formă de perechi <nume, valoare> și au o dimensiune de până la 8KB pe container.
- Există, posibilitatea de a lista toate bloburile din container.

Contul de stocare

- accesul la spațiul de stocare Windows Azure se realizează printr-un cont de stocare.
- cel mai înalt nivel al spațiului de nume pentru accesarea bloburilor
- un cont poate avea numeroase containere Blob

Tabele Windows Azure

Windows Azure Tables reprezintă un sistem de stocare structurat și scalabil non-relațional, de tip cheie-valoare, destinat stocării unor volume mari de date nestructurate.

Deoarece stocarea datelor într-o paradigmă relațională, cum ar fi SQL Server-cu un design bazat pe normalizarea structurilor, fiind optimizată pentru stocarea de date, aşa încât interogările să fie ușor de realizat, soluțiile de stocare non-relațională, cum ar fi volumele de date livrate în timp real sunt optimizate pentru extragerea simplă și adaugare rapidă.

Tabele Windows Azure

Storage Structurat și Scalabil

- Încarcă tabele cu miliarde de intrări și TB de date
- Schema Flexibilă(NoSQL)
- Modelul de Date
 - a. O tabelă este o familie de intrări(linii)
 - b. O entitate este o familie de proprietăți(coloane)

API ușor de utilizat

- Protocol OData
- Serviciul de date WCF – clase .NET și LINQ

Tabele Windows Azure

- O entitate este o abstractizare a unei structuri de date identificată printr-un set de proprietăți ale acesteia.
- O entitate poate fi identificată cu o relație între proprietatile(coloane) unei anumite structuri
- este mai mult decât o colecție de proprietăți și valori care au fost stocate împreună într-un tabel.
- Spre deosebire de SGBDR, entitățile în cadrul unei tabele nu trebuie să aibă aceeași structură sau schemă.
- Acest lucru înseamnă că am putea avea o entitate care stochează proprietăți ale unui produs în aceeași tabelă cu o entitate care stochează proprietăți despre opțiunile produsului

Tabele Windows Azure

- Există cîteva reguli referitoare la entități: fiecare entitate poate avea pînă la 252 proprietăți dar mărimea entității nu poate depăși 1 MB.
- Entitățile asociate Table au asociate următoarele tipuri de date: **Byte array, Boolean, DateTime, Double, GUID, Int32, Int64 și String** (pînă la 64KB).
- Există următoarele trei proprietăți pentru fiecare entitate: **PartitionKey, RowKey și TimeStamp**.
 - RowKey este unic identificator al entitatii în cadrul partiției.
 - **Timestamp** reprezinta ora ultimei modificări a entității și este gestionată de sub sistemul Storage.
 - Dimensiunea admisa 200TB

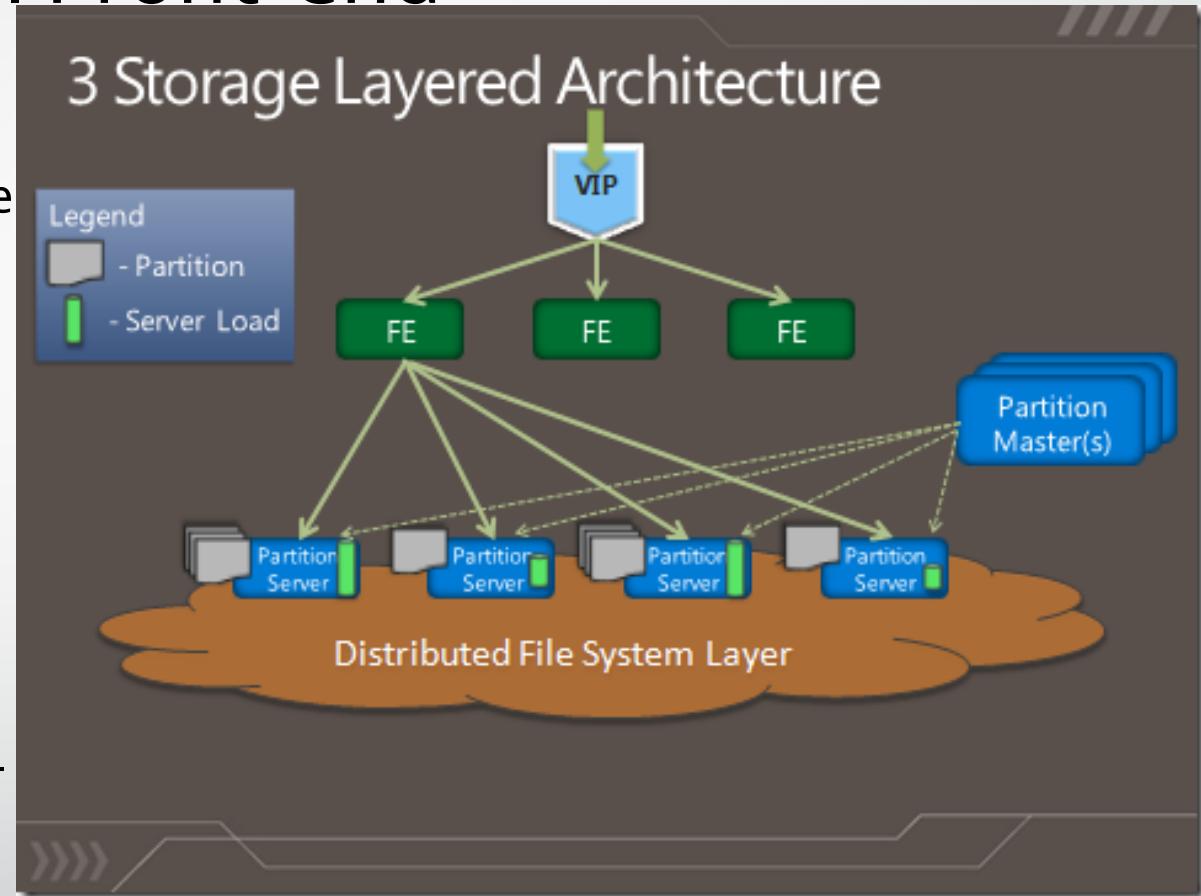
Servicii de Stocare În Cloud(StaaS)

- STaaS este un model de cloud computing destinat stocării datelor în format digital de o manieră logică, stocarea fizică fiind realizată pe sisteme distribuite de servere geolocate, iar mediile fizice sunt în mod uzual deținute și administrate de o companie furnizoare de servicii cloud. Acești furnizori de stocare în cloud răspund pentru păstrarea datelor și accesibilitatea acestora, precum și de protejarea și funcționarea mediului fizic. Utilizatorii achiziționează sau închiriază capacitate de stocare de la furnizorii de servicii pentru a stoca date necesare utilizatorilor privati, companiilor sau aplicațiilor deținute de acestia.

Arhitectura serviciului Azure Storage

Nivelul Front-end

- Front-End Layer(FE)-Acest layer preia cererile, autentifică și autorizează solicitările și apoi le direcționează spre un server de partii din layerul de partii.
- Front-end-ul stabilește către ce server de partii să trimită fiecare cerere, deoarece fiecare server de front-end gestionează în cache o hartă de partii.
- Harta partiiilor realizează evidență partiiilor pentru accesarea serviciului (Blob-urilor, Table-urilor sau Queue-urilor) și ce server de partii controlează accesul la fiecare parte din sistem.



Arhitectura serviciului Azure Storage

Nivelul Partiție

- Nivelul partiție - gestionează partajarea tuturor obiectelor de date din sistem.
- Obiectele au asociată o cheie de partiționare.
- Un obiect aparține unei singure partiții și fiecare partiție este alocată de un singur server de partiții.
- Aceasta este nivelul care gestionează ce partiție este alocată unui server de partiții dat. În plus, acesta asigură echilibrarea încărcării partițiilor de pe servere pentru a răspunde nevoilor de trafic ale Blob-urilor, Table-urilor și Queue-urilor.
- **Un singur server de partiții poate gestiona mai multe partiții.**

Arhitectura serviciului Azure Storage

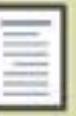
Nivelul de Distribuție și Replicare al Fisierelor

- Nivelul DFS(Distributed and replicated File System)- este nivelul care gestionează stocarea efectivă pe disc și este responsabil de distribuirea și replicarea datelor pe mai multe servere pentru a le menține durabilitatea.
- Un concept cheie pentru a înțelege aici este faptul că datele sunt stocate de stratul DFS, dar toate serverele DFS sunt (și toate datele stocate în stratul DFS este) accesibile de pe oricare dintre serverele de partii.

Opțiuni în “data storage”

- **Baze de date cheie-valoare** = stochează un singur obiect serializat pentru fiecare pereche cheie-valoare.
- **Azure BLOB storage**=este o bază de date cheie- valoare care funcțio-nează drept sistem de gestiunea fisierelor în cloud bazat pe un sistem de identificare prin perechi (director, nume fisier) nu elemente de conținut.
- **Azure Table Storage**=este o bază de date cheie-valoare. Fiecare valoare fiind o entitate. Instrument software scalabil dar dificil de interogat.
- **Baze de date coloana**=stocheaza date cheie / valoare care permit structurarea datelor ce vor fi stocate în colecții de coloane aferente schemei bazei.

Variante de implementare

Relational	Key/Value	Column Family	Document	Graph
 <ul style="list-style-type: none">• Azure SQL Database• SQL Server• Oracle• MySQL• SQL Compact• SQLite• Postgres	 <ul style="list-style-type: none">• Azure Blob Storage• Azure Table Storage• Azure Cache• Redis• Memcached• Riak	 <ul style="list-style-type: none">• Cassandra• HBase	 <ul style="list-style-type: none">• MongoDB• RavenDB• CouchDB	 <ul style="list-style-type: none">• Neo4J

Microsoft Azure

- Windows Azure reprezintă fundamentalul platformei de cloud Microsoft Azure
- Este un "Sistem de operare in cloud" și furnizează servicii esențiale pentru cloud

A cloud operating system is a type of operating system designed to operate within cloud computing and virtualization environments. A cloud operating system manages the operation, execution and processes of virtual machines, virtual servers and virtual infrastructure, as well as the back-end hardware and software resources.

A cloud operating system may also be called a virtual operating system.

TEHNOPEDIA

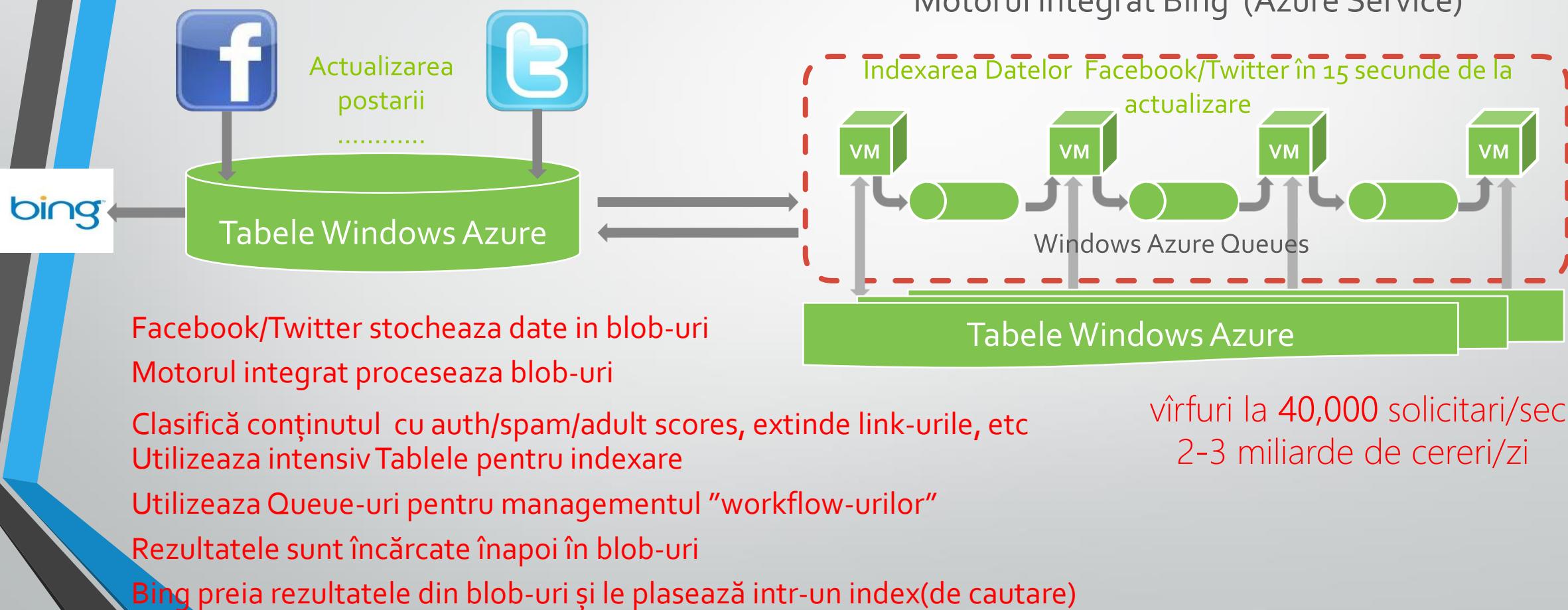
- Calculabilitate virtualizată
- **Stocare scalabilă**
- Management automat
- SDK pentru dezvoltatori

Abstractizarea datelor

- BLOB – Furnizează o interfață simplă pentru stocarea de fisiere împreună cu metadatele asociate.
- **Tabele – Furnizează stocare structurată;**
O tabela este o familie de entități care se caracterizează printr-o familie de proprietăți.
- Cozi – Furnizează storage și livrează mesaje pentru aplicații.

Windows Azure Storage

Bing, facebook/twitter integreaza motoare de cautare in timp real



Windows Azure Storage Account

Utilizatorii creaza un cont global, unic de stocare.

Se alege o locatie primara pentru gazduirea contului de storage.

North Central US

South Central US

Northern Europe

Western Europe

East Asia

South East Asia

Cozi de mesaje in Windows Azure

Livrare de mesaje

Semantica Programului – Un mesaj poate fi procesat o singura data

Plasarea mesajului într-o coadă

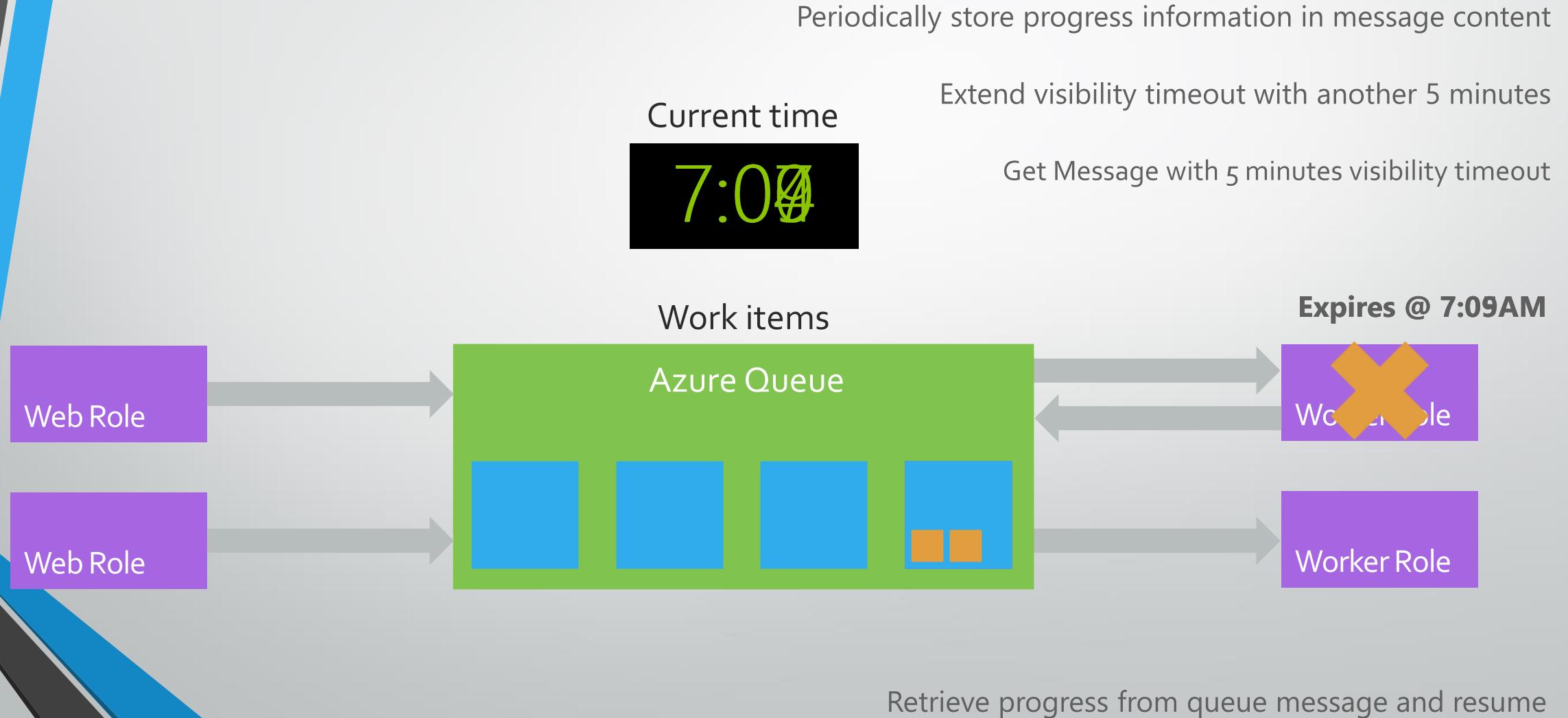
Preluarea mesajului îl face invizibil în coadă pentru un timp specificat.

Stergerea mesajului în procesul de prelucrare implica eliminarea lui din coada.

Dacă procesul worker cade atunci mesajul devine vizibil în coada pentru alt proces worker.

Windows Azure Queue

Update Message Example





DEMO

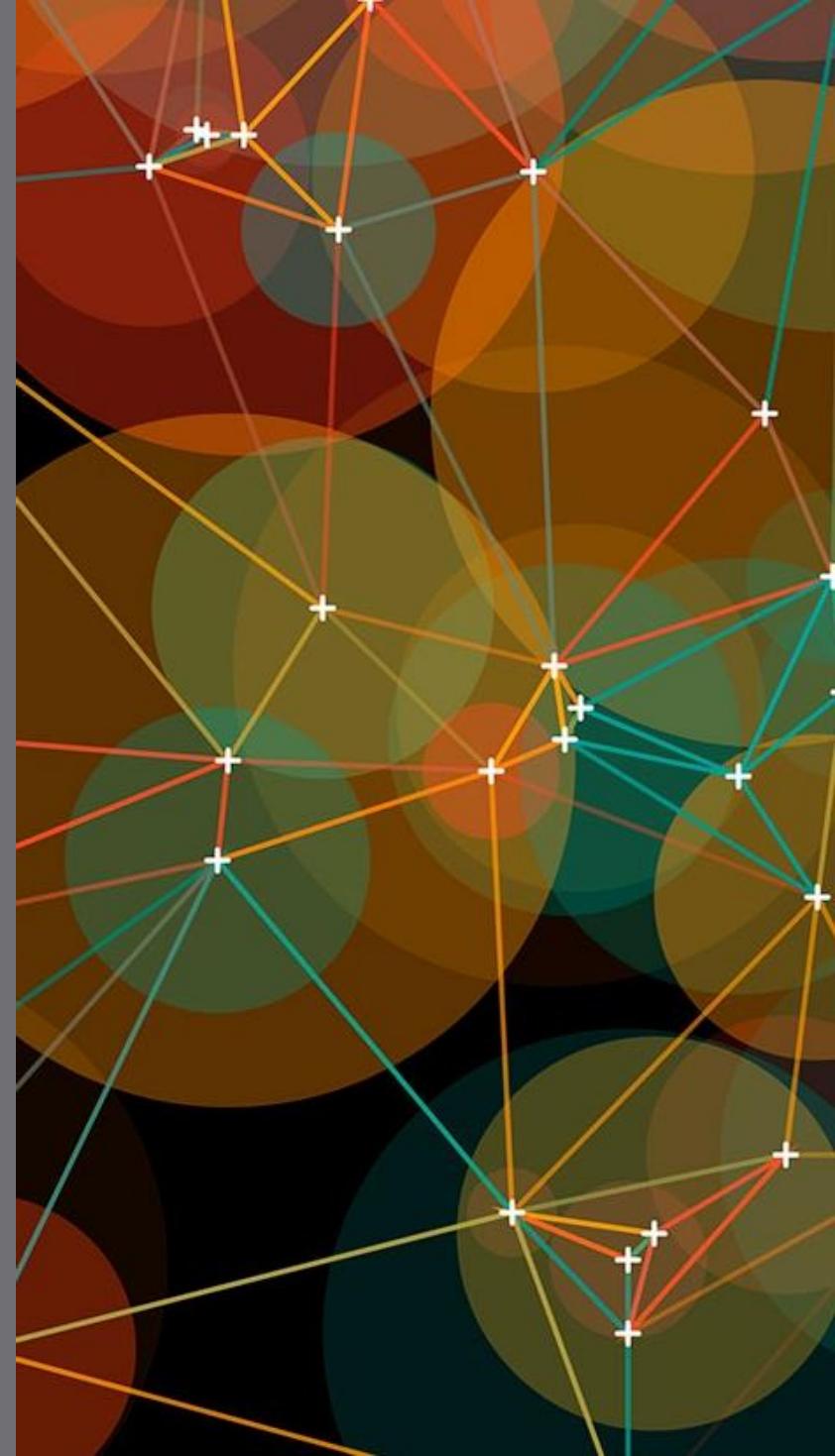
”DB as a Service” și „Data Lake”

Conf.dr. Cristian Kevorchian

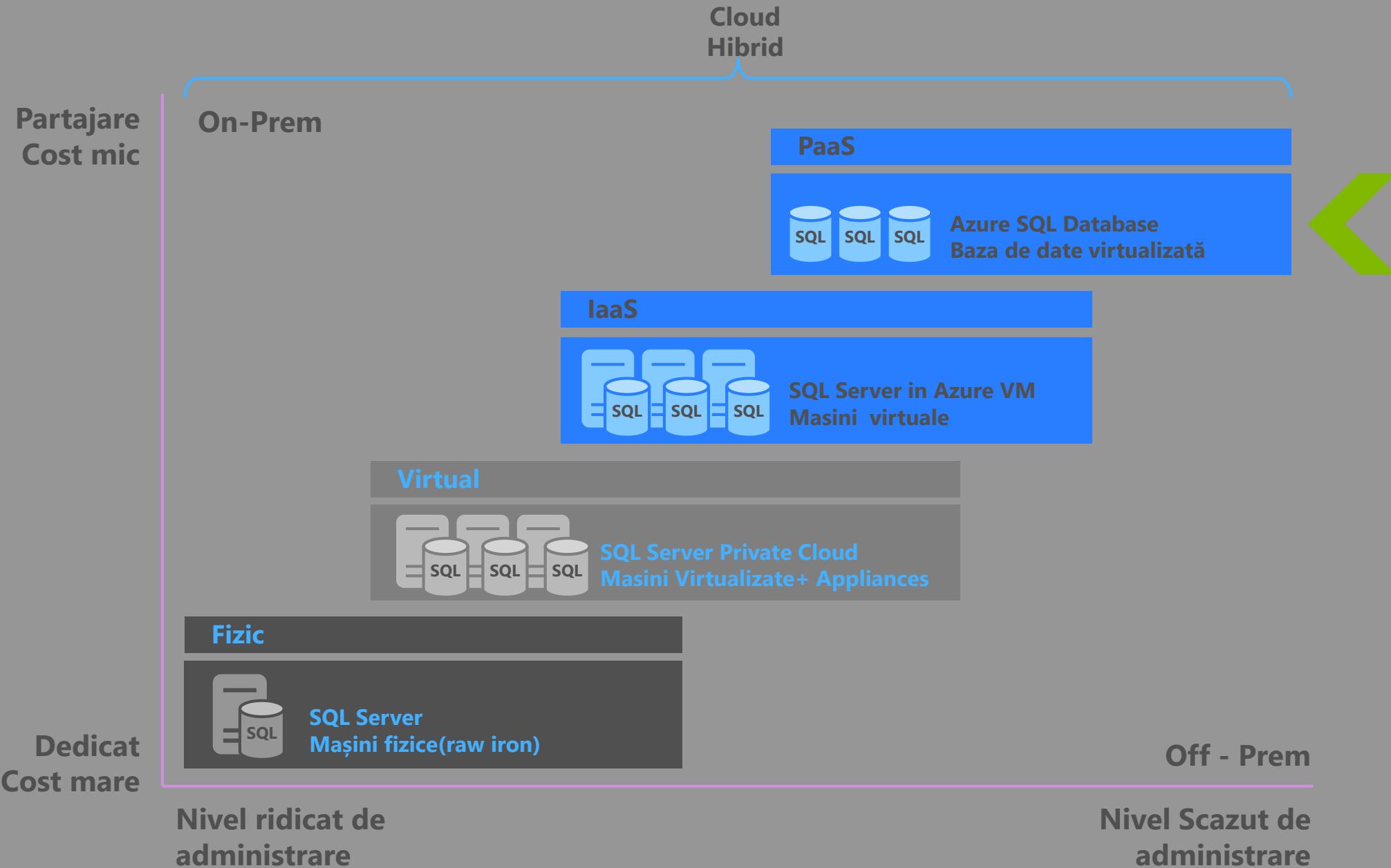
Facultatea de Matematică și Informatică

ck@fmi.unibuc.ro

Azure SQL Server



Microsoft SQL - Baza de date ca serviciu



Azure SQL Database

Baza-de-date livrată ca serviciu, complet gestionată de Microsoft

Proiectată pentru aplicații în cloud cu efort minim de administrare

Scalabilitate

Nivel de performanță predict.
Scale up/down & out/in
Vizualizare prin dashboard-uri
a metricilor BD

Business Continuity și Protecția datelor

Self-service restore
Disaster recovery
Compliance-enabled

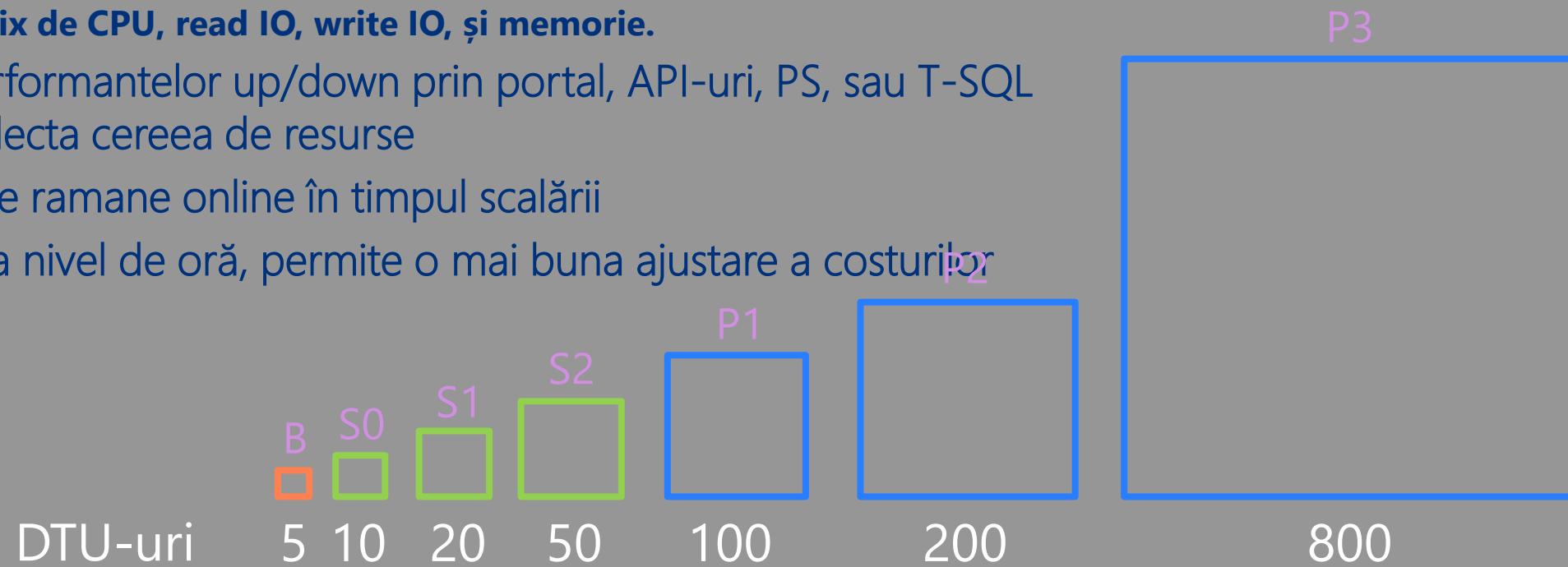
Self-managed and Familiar

Self-managed
API-uri programabile
Instrumente de lucru și
limbaje familiare(e)

Platformă de baze de date enterprise

Scalabilitate

- Basic, Standard și Premium furnizează nivelele de performanță și implicit de preț.
- Performanța este exprimată prin DTU(Database Throughput Unit)
- **DTU este un mix de CPU, read IO, write IO, și memorie.**
- Scalarea performantelor up/down prin portal, API-uri, PS, sau T-SQL pentru a reflecta cererea de resurse
- Baza de date ramane online în timpul scalării
- Facturarea la nivel de oră, permite o mai bună ajustare a costurilor



Performante usor scalabile în funcție de cerințele de business

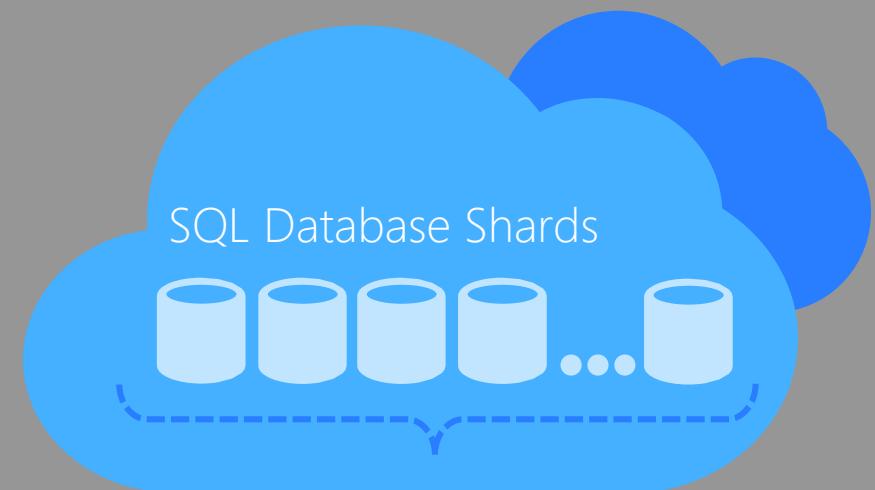
Scalare elastică

Scalarea a mii de baze de date utilizând modelul bazelor de date partitionate(sharded)

Suportă adăugarea, împărțirea și reunirea partțiilor în funcție de mișcarea datelor

Clientii pot reuni rezultatele interogării din mai multe partitii.

Operațiuni de gestiune asincronă (întreținerea indexului, DDL, DML)



Geo-replicare Asincronă

Geo-replicare standard (standard și premium)

Opțional generarea unei replici secundare(non-readable) intr-o regiune pereche

Replica este taxata la consturi reduse

Activarea replicii se face de Microsoft la apariția unei catastrofe.

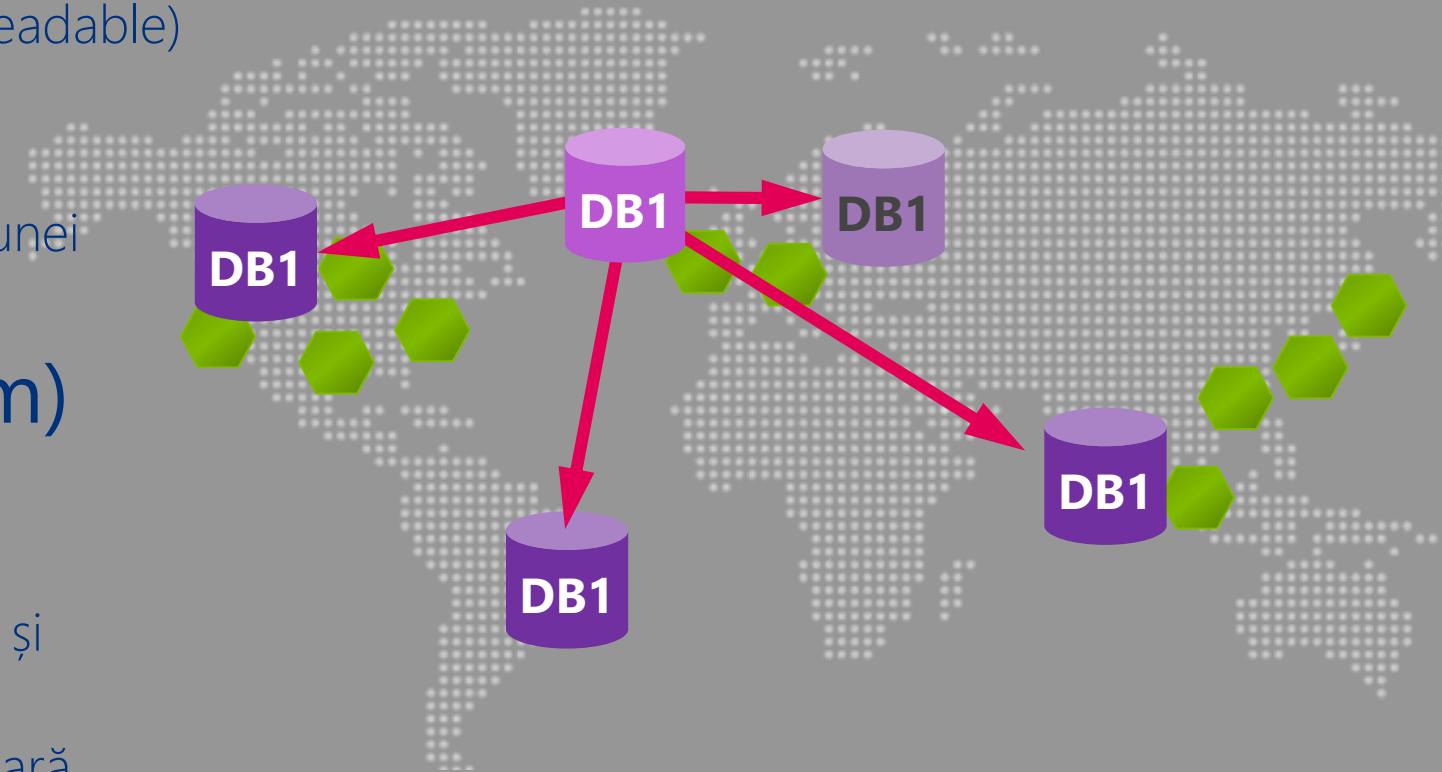
Geo-replicare activă (Premium)

Până la patru replici în zone secundare

Control complet peste locația secundara

Suportă load balancing, actualizarea applicațiilor și scenario de relocare

Poate fi combinată cu o replica din zona secundară



Geo-replicarea minimizează discontinuitatea business-ului

“Zero-admin” și “Self-managed”

Zero-administrare

Infrastructura virtualizată elimină aproape tot efortul de întreținere, inclusiv aplicarea patch-urilor.

Platforma HA tolerantă-la-erori nu necesită monitorizare

Generarea de back-ul automat

Self-service management

Realizat prin Azure management portal, T-SQL, REST APIs, PowerShell
Provision, copy, delete, restore, configure geo-replication, auditare,
export/import, etc.

Mentenanță și tolerantă-la-erori prin servicii built-in.

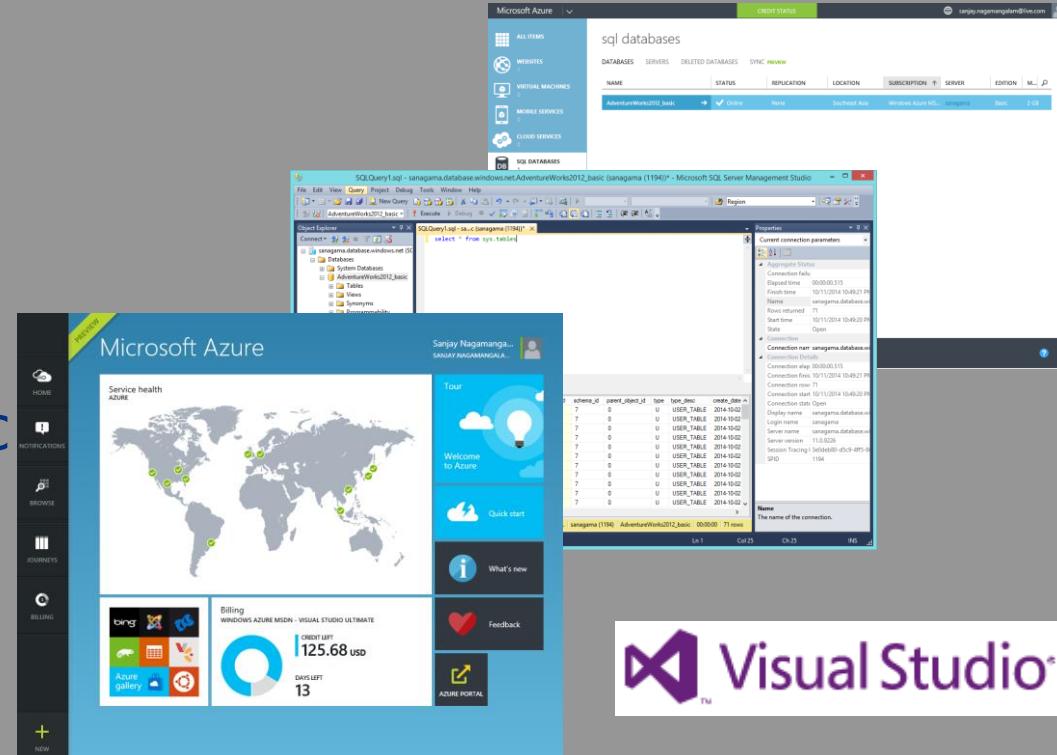
Dezvoltarea de aplicații

T-SQL, REST API-uri, PowerShell

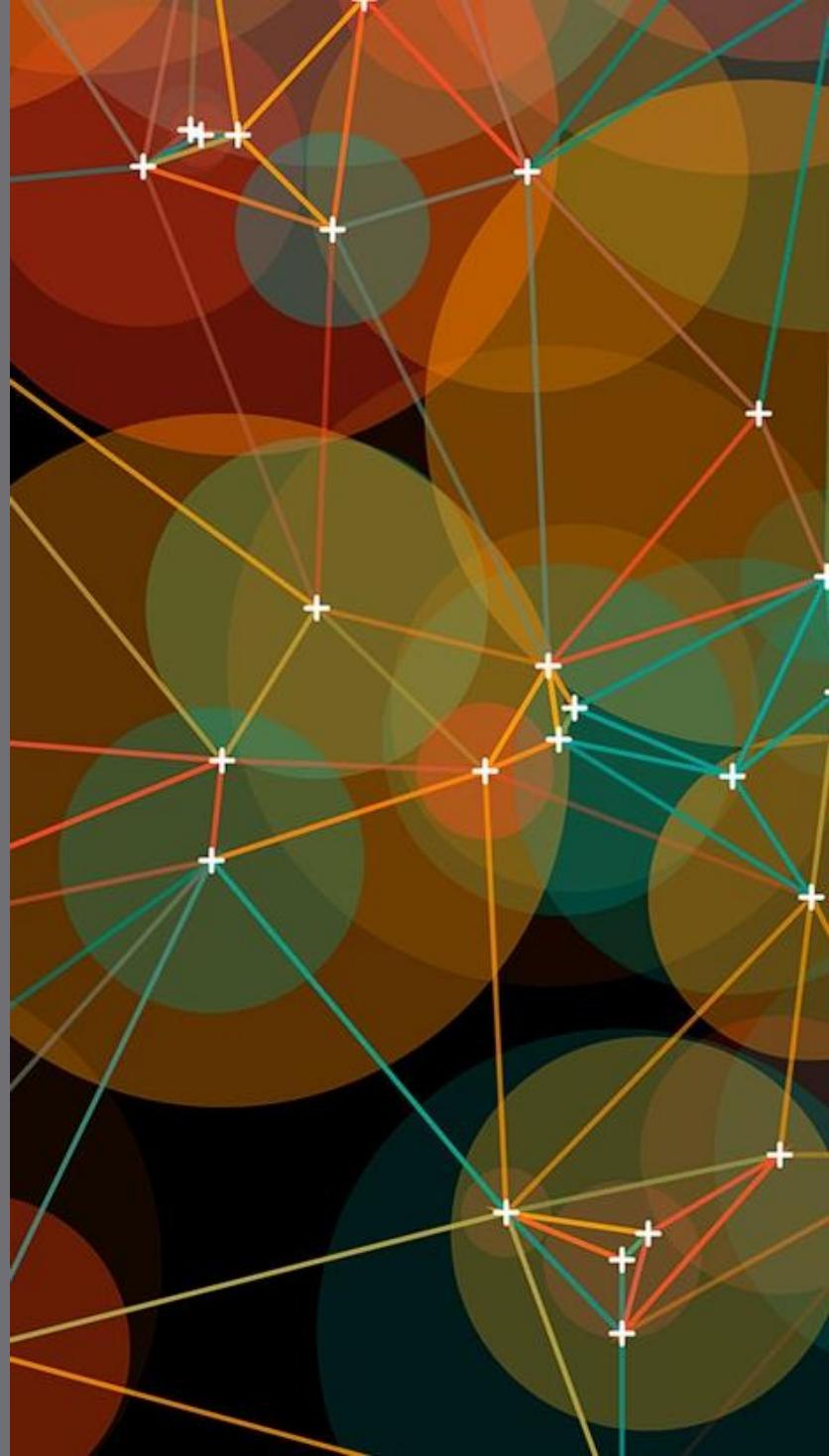
SQL Server Management Studio (SSMS),
Visual Studio

Suportă platforme și tehnologii dintre care
amintim .NET, Java, Ruby on Rails, Node.js etc

Azure Machine Learning Services

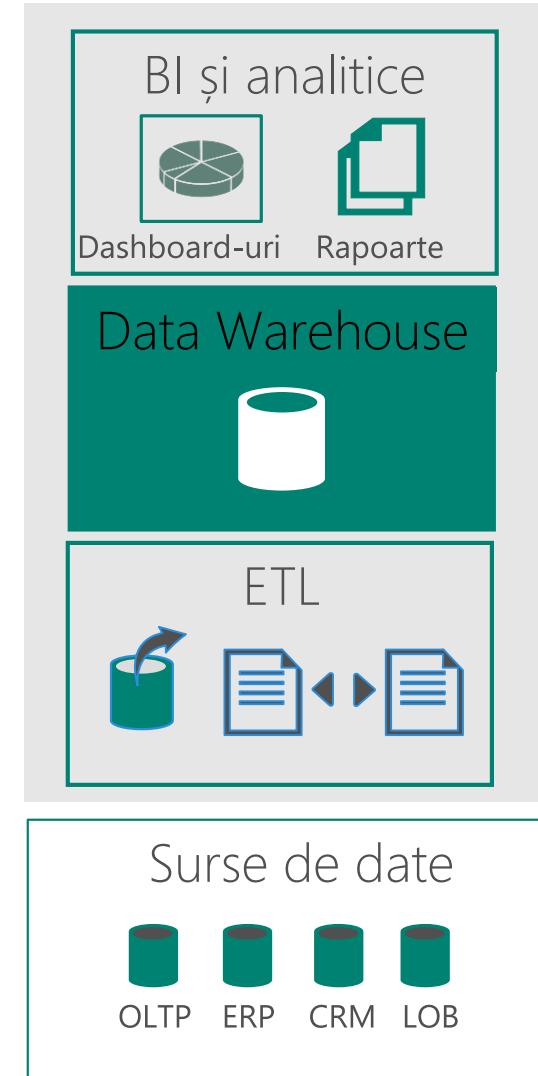


Data Warehouse



Data Warehouse Traditional

... data warehouse a atins cel mai important punct al evolutiei sale. Cel mai elaborat sistem de gestiune a datelor din IT se transformă de o manieră radicală.



Big Data este motorul transformării

"Big data reprezintă o familie de active informaționale care se caracterizează prin "high-volume", "high-velocity" și/sau "high-variety" care implică costuri-efective, forme inovative de procesare a informației care conduc la o cunoaștere îmbunătățită a proceselor interne, a modului de luare a deciziilor, și a automatizării proceselor."

Gartner, Big Data Definition*

* Gartner, Big Data (Stamford, CT.: Gartner, 2016), URL: <http://www.gartner.com/it-glossary/big-data/>

Big Data implică modificări radicale



Characteristicile datelor

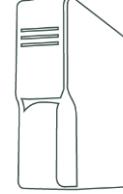


Costuri

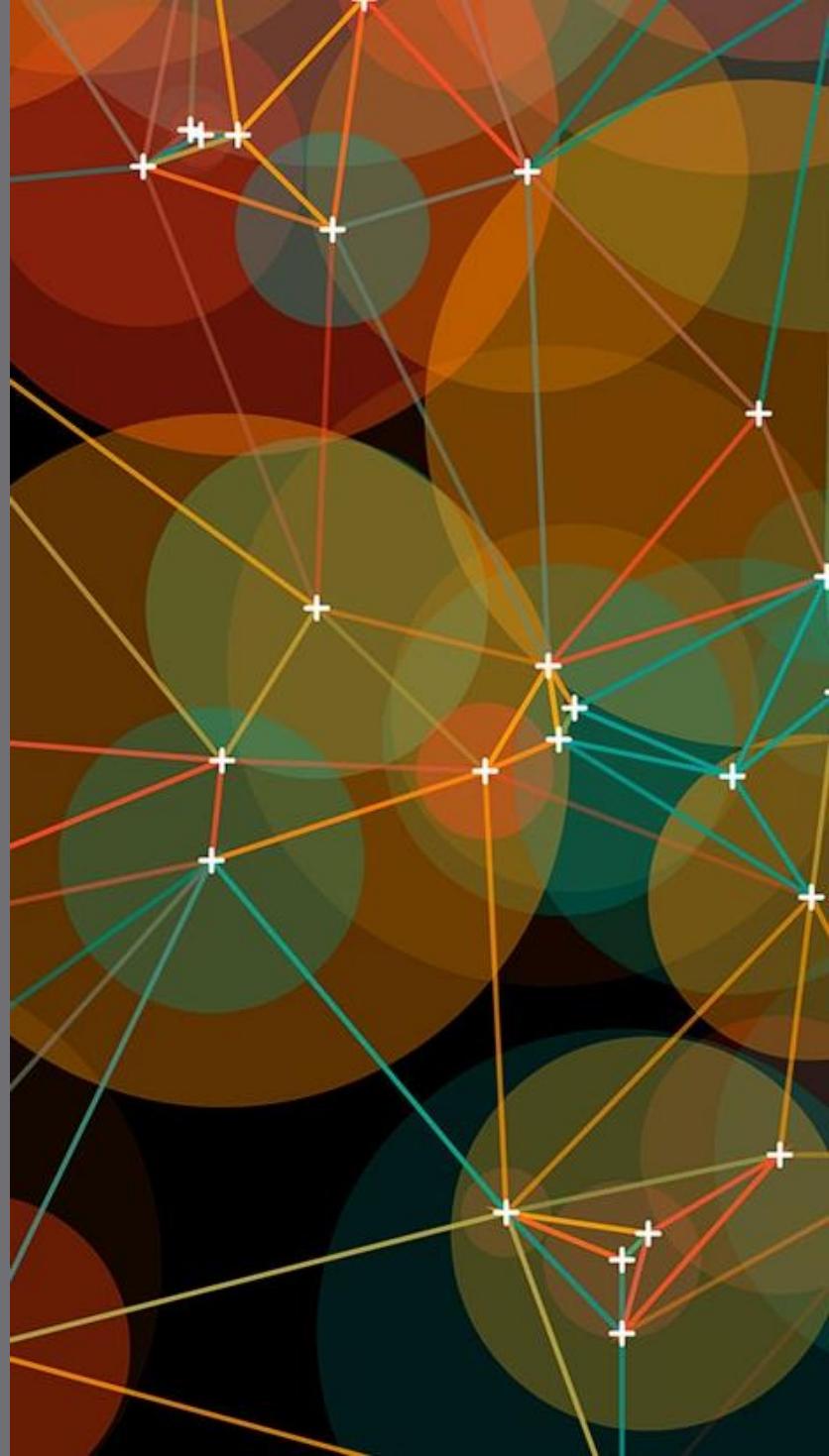


Cultură

Big Data induce modificări radicale

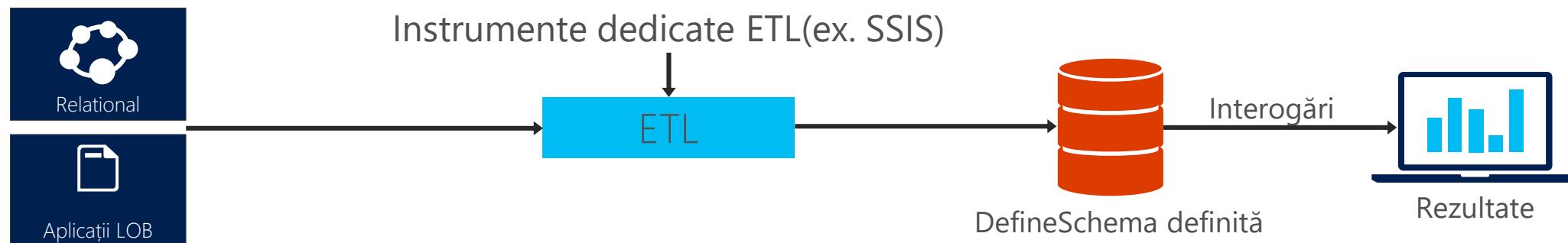
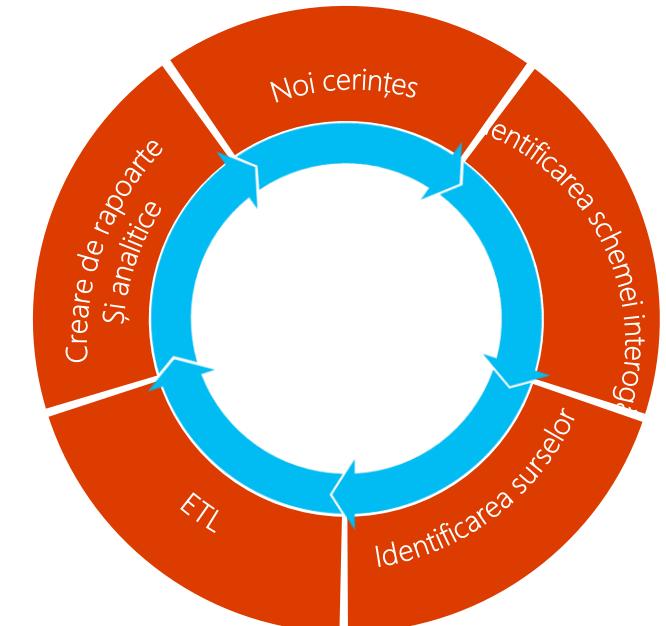
	Tradițional	Big Data
Caracteristicile datelor	 Relațional (cu dependență de schemă)	 Date (cu schemă adaptivă)
Costuri	 Costuri mari (capacitate mare de calcul și stocare)	 leftine (stocarea și capacitate de calcul)
Cultură	 Vizualizare rapoarte (utilizarea algebrei relaționale)	 Acțiuni inteligente (utilizând alg. relațională și ML)

Data Lake



Analitice asociate business-ului tradițional

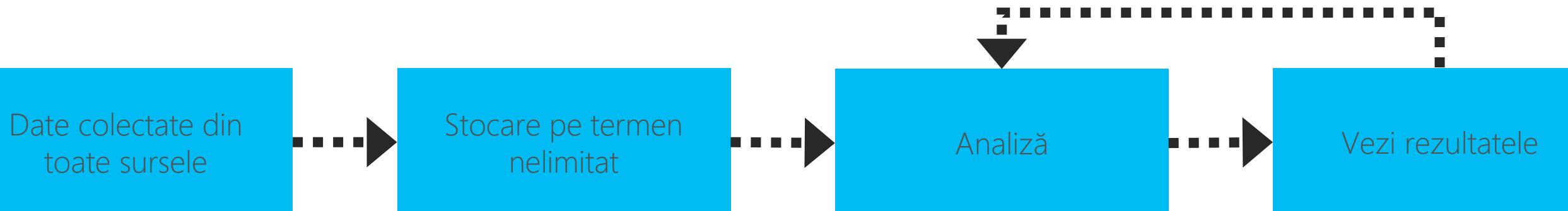
1. Se pornește de la cerințele end-user-ului pentru a identifica rapoartele și analizele dorite.
2. Definirea corespunzătoare a schemei bazei de date și interogărilor.
3. Identificarea surselor de date
4. Crearea unui ETL(Extract-Transform-Load) pentru extragerea datelor cerute și transformarea acestora schemei țintă ('schema-on-write')
5. Crearea de rapoarte și analize.



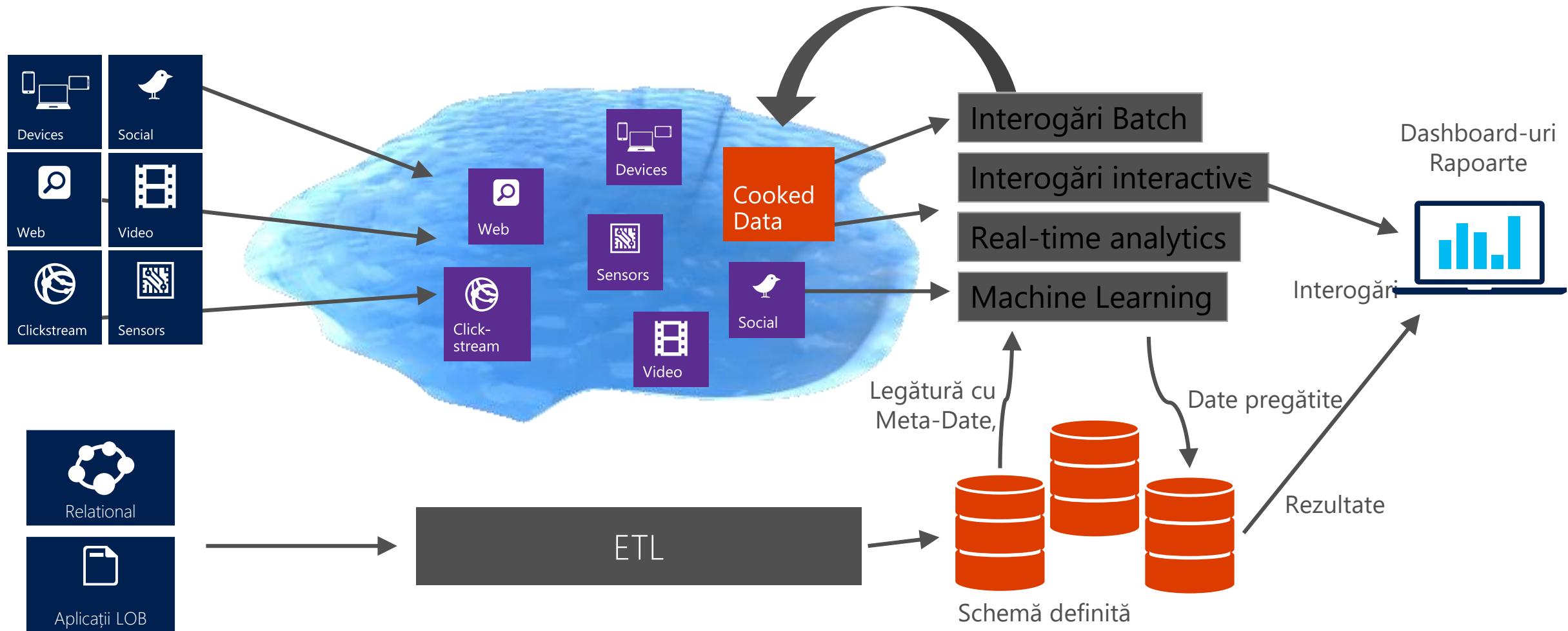
Toate datele care nu sunt necesare imediat sunt eliminate sau arhivate

Big Data: Toate datele sunt valoroase

- Toate datele au o valoare potențială
- Date tezaurizate
- Fără schemă definită—încărcate în format nativ
- Schema este impusă și transformările se fac la momentul interogării(*schema-on-read*).
- Aplicațiile și utilizatorii interpretează datele aşa cum consideră de cuvîntă



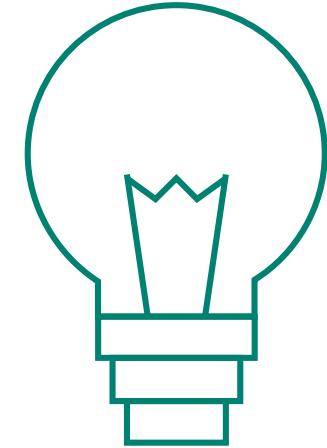
Data lake și warehouse



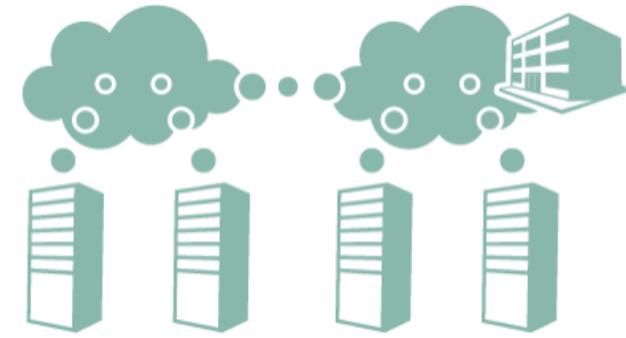
... Big Data nu este un lucru banal



Obținerea de
competențe

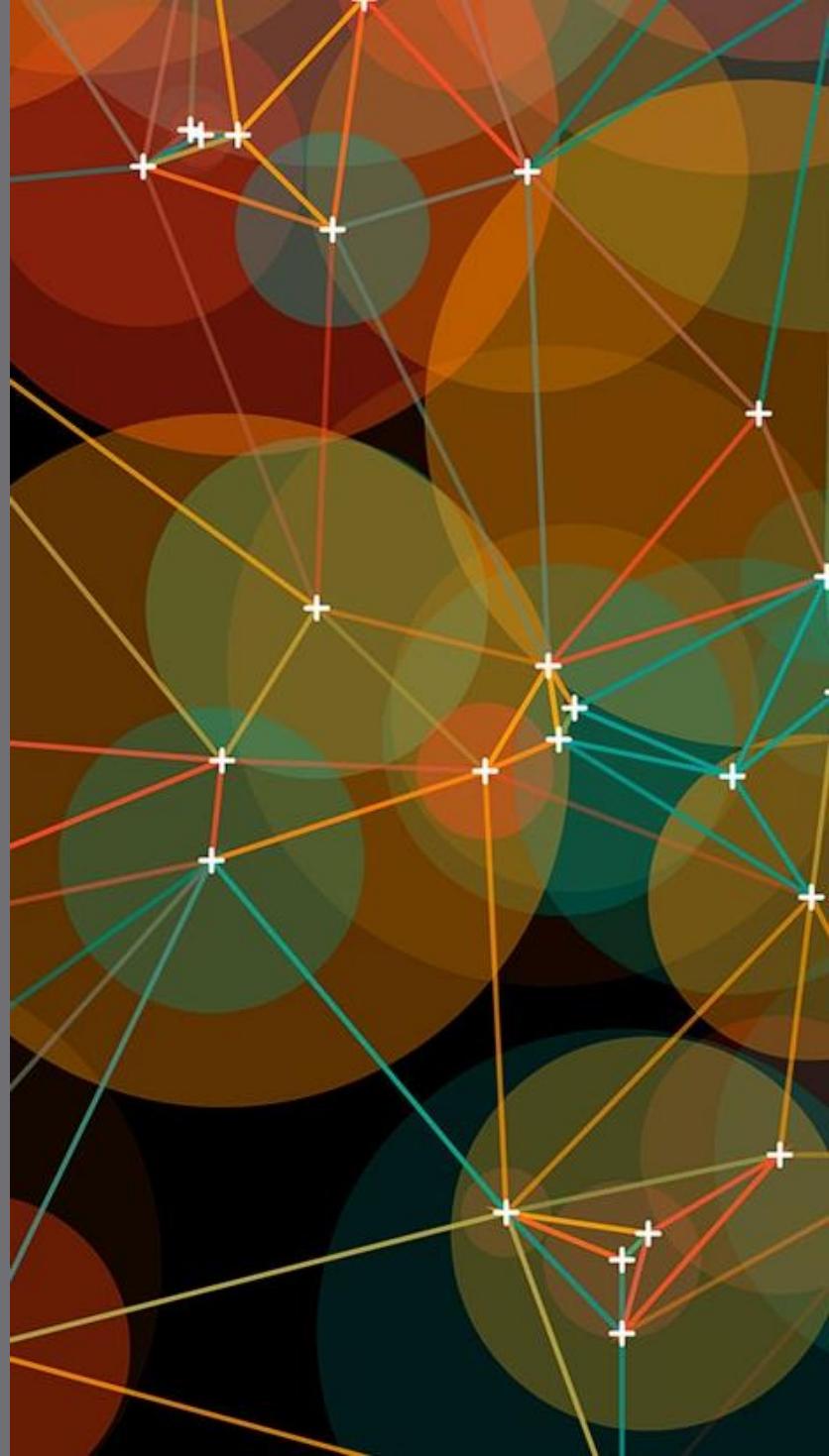


Determină modul
în care putem obține
valoare



Integrarea cu
Investițiile IT existente

Pattern-uri pentru Big Data



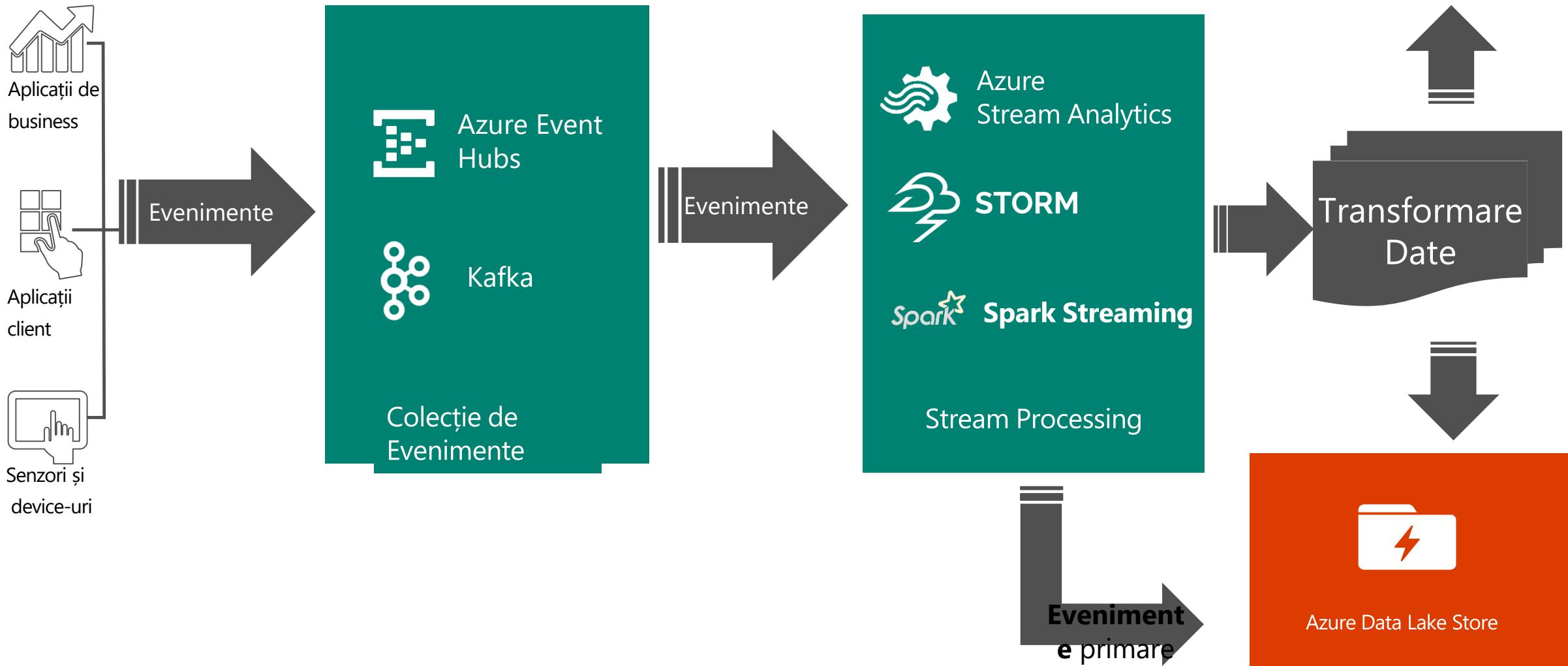
Big Data Analytics – Fluxuri de date



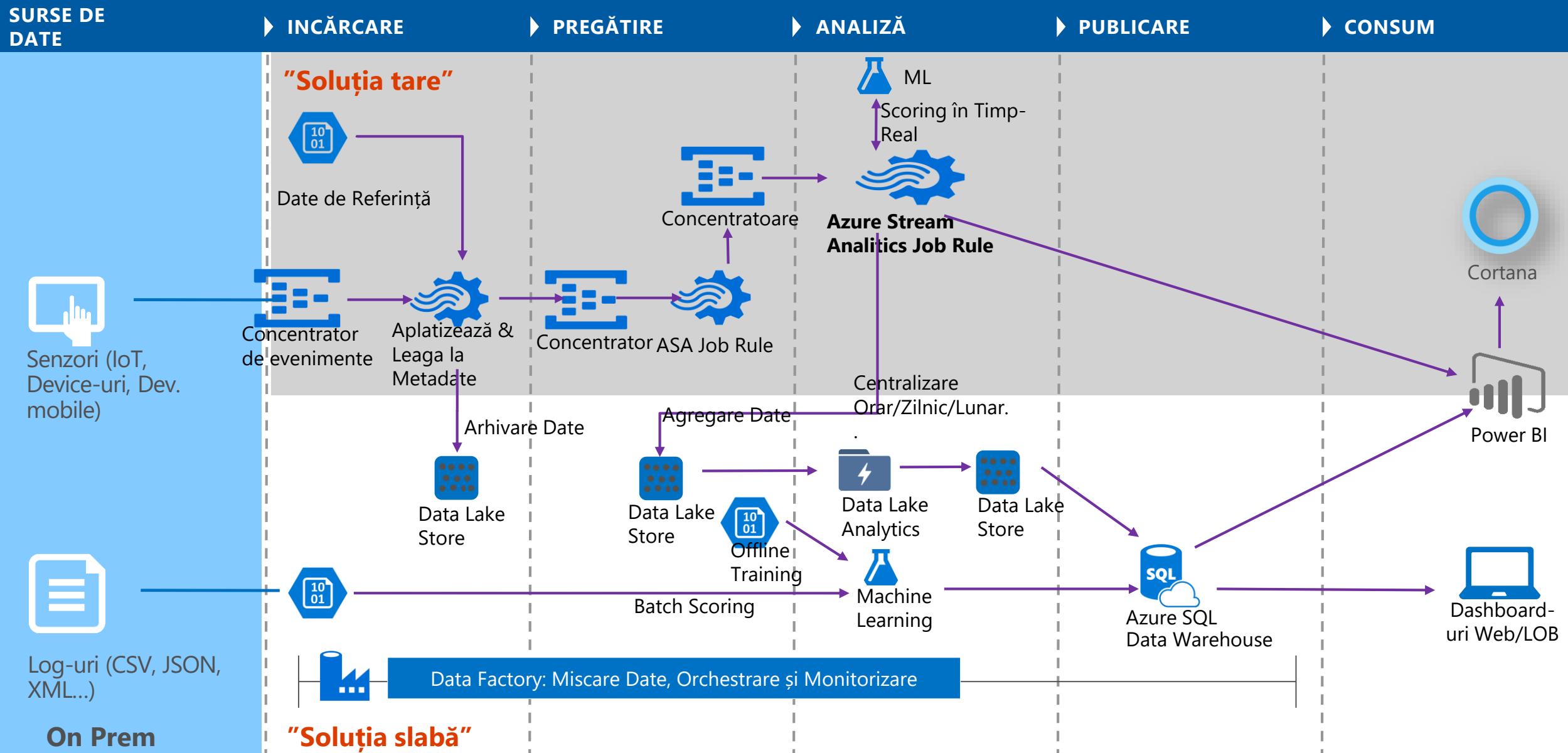
Pattern-uri de Încărcare Date furnizate de Evenimente



Dashboard-uri în tim real

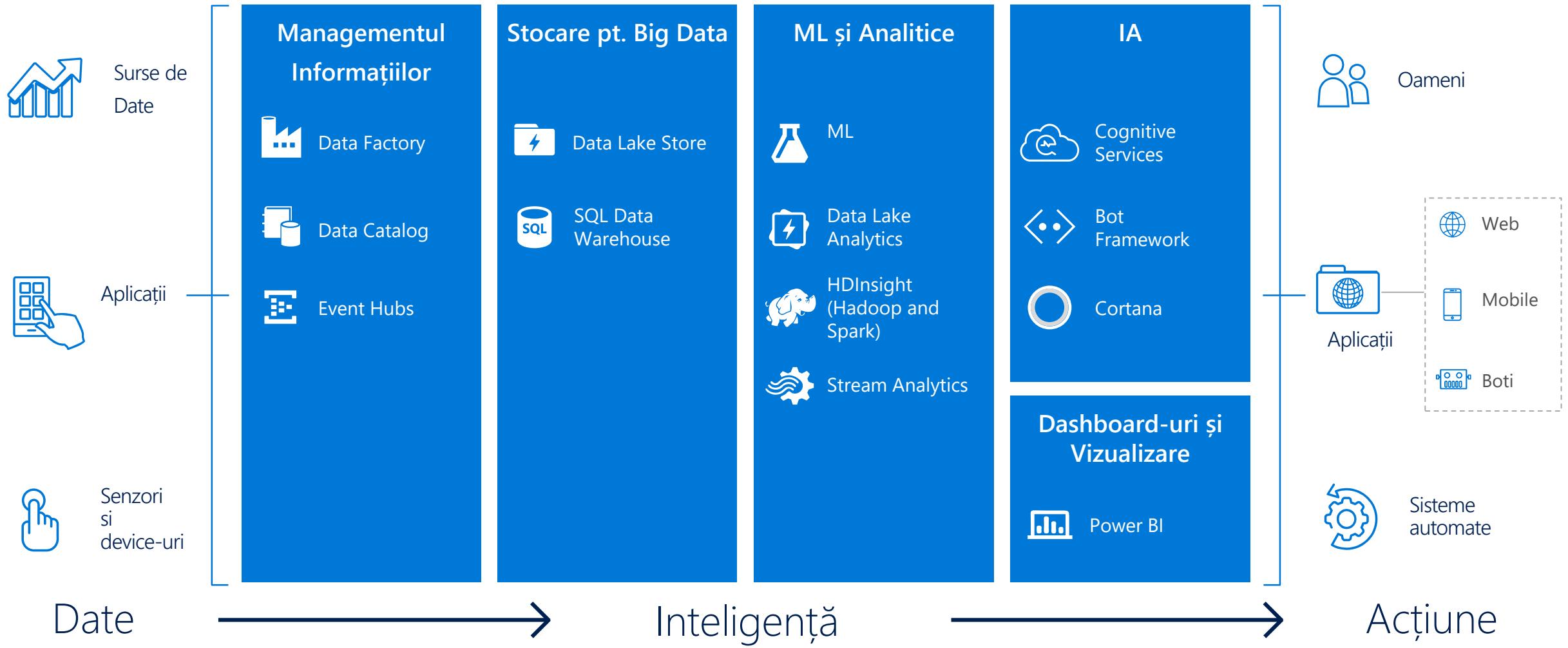


Arhitectură Lambda

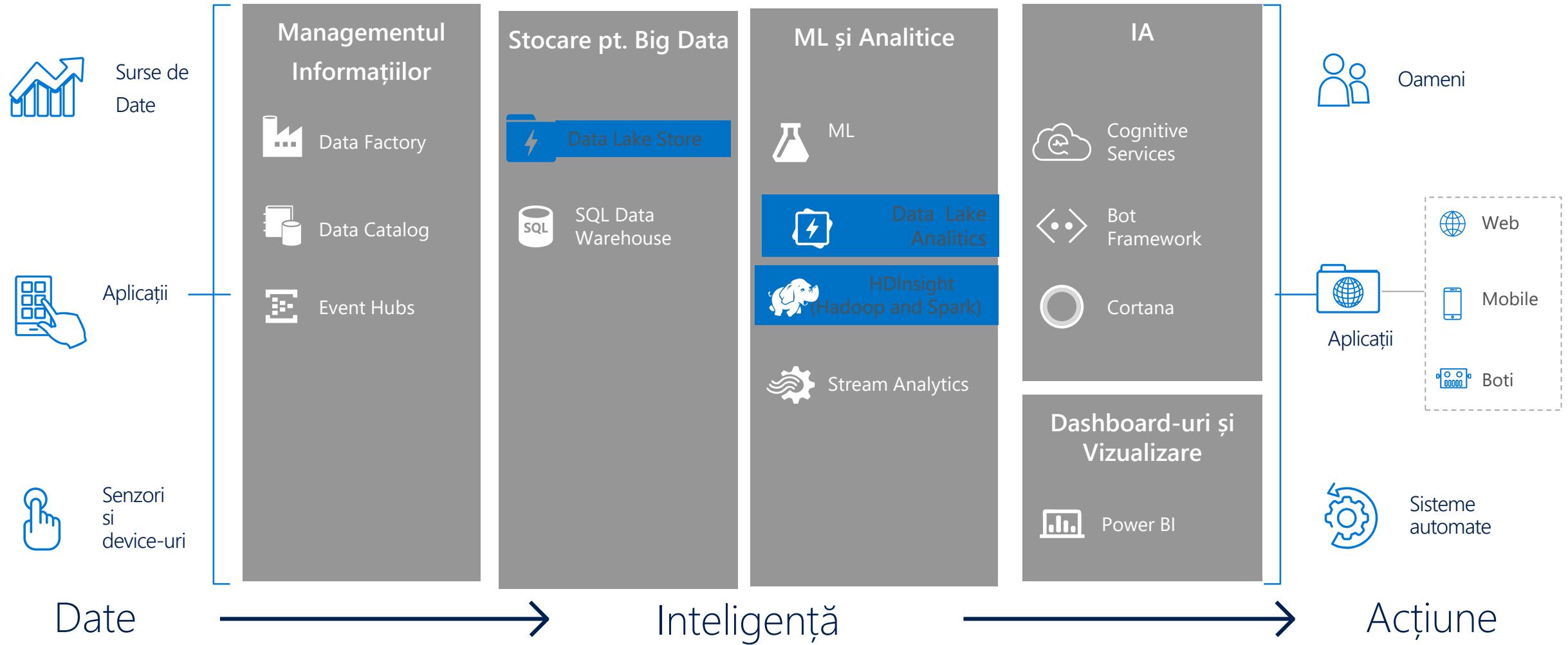


Trecerea la big data – o cale dificilă

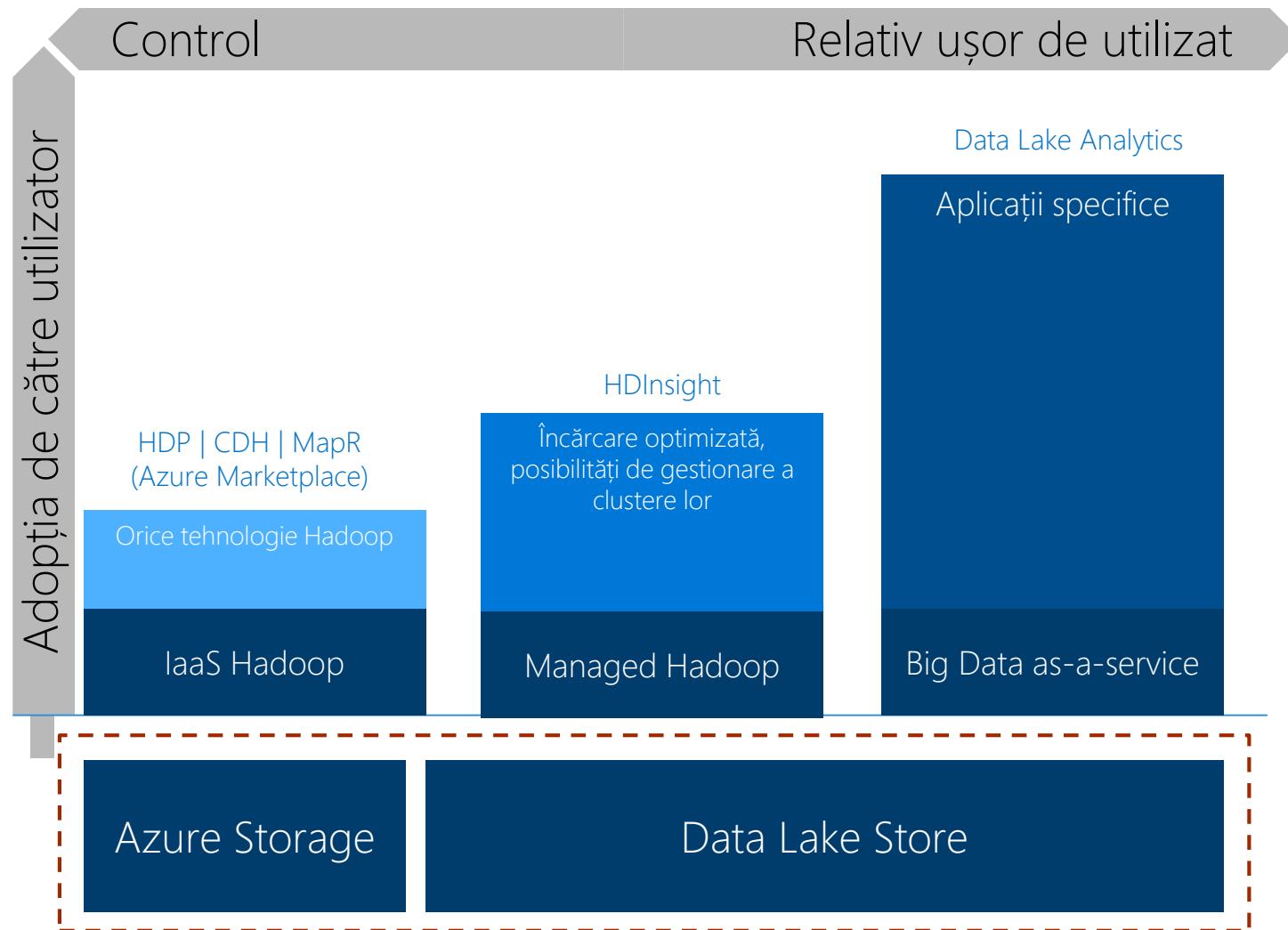
- Selectarea unei stive open source
- Selectați mai multe componente din zecile care sunt disponibile în baza volumului de lucru.
- Achiziție de hardware, networking, etc
- Instalare și management, de ex. cu Ambari
- Instalarea componentelor Big Data
- Instalarea serviciilor opționale big data
- Adauga serviciile de securizare și autentificare
- Configurare și testarea cluster
- <câte și mai câte ...>



Unde Big Data este o soluție fundamentală



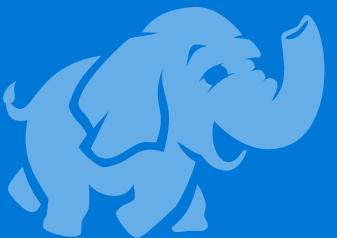
Democratizarea Big Data



Construit pentru cloud pentru a accelera ritmul inovării pe o platformă calcul modernă

Azure HDInsight

Hadoop și Spark
ca Serviciu în Azure



Fully-managed Hadoop and Spark
pentru cloud

**100% Open Source Platforma de date de
la Hortonworks**

Clusters disponibil **relativ ușor**

Familie **de instrumente BI pentru analiză**,
sau medii open source pentru **data science
interactiv**

63% costuri mai mici ale TCO decât
utilizare Hadoop on-prem

Azure Data Lake Store

Un hyper-scalabil
storage pentru analitice
Big Data



Hadoop File System (HDFS) adaptat la cloud

Scalare fără limite

Încarcă **orice date** în format nativ.

Optimizat pentru lucrul cu analitice
performante

Azure Data Lake Analytics

Serviciu distribuit de analitice



Serviciu distribuit de analitice bazat pe Apache YARN

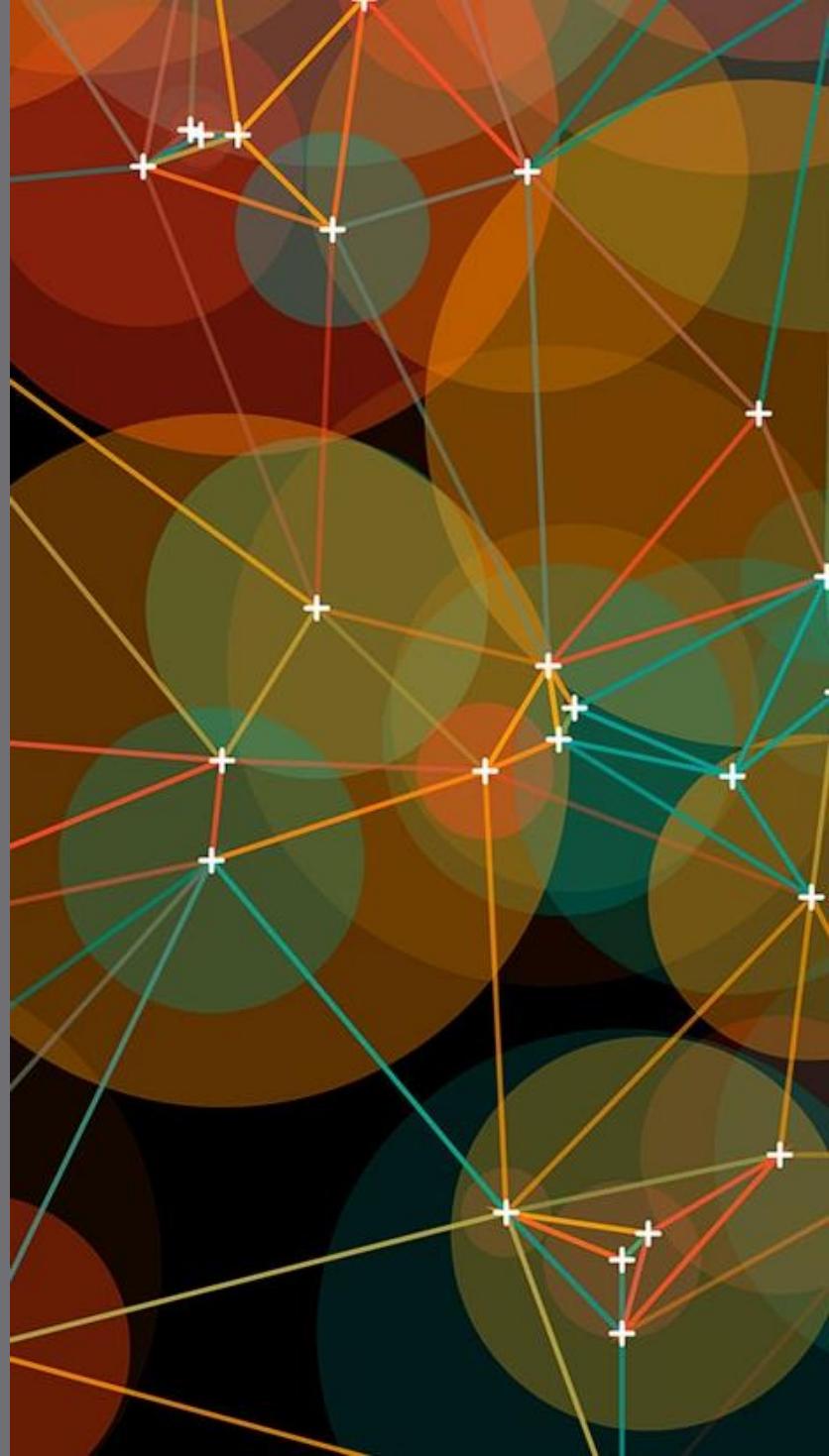
Sistem elastic de interogare permite utilizatorilor să se concentreze pe problematica de business —nu pe configurații de mediu

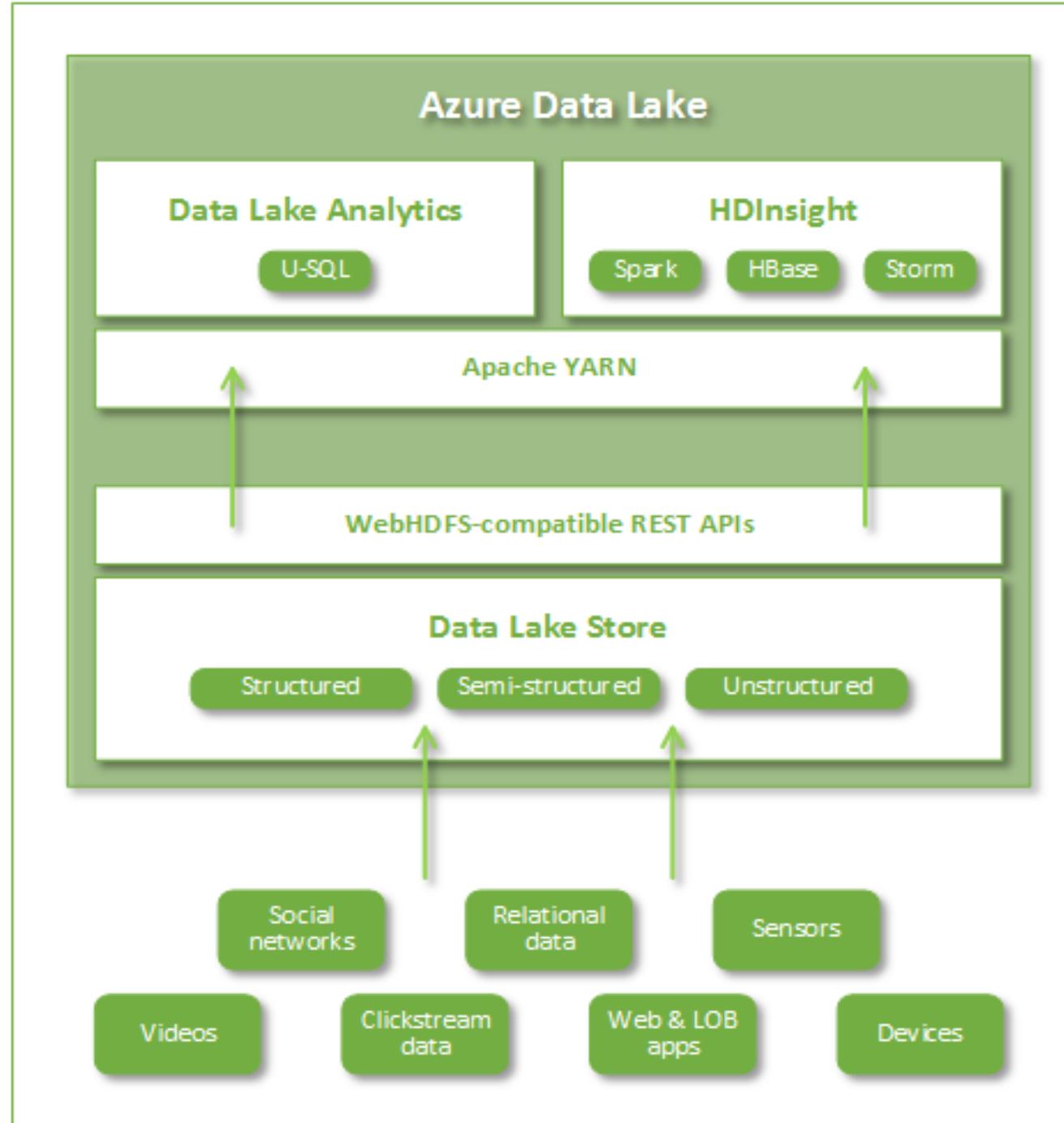
Include U-SQL—un limbaj care unifică **beneficiile SQL puterea C#**

Integrat cu Visual Studio pentru dezvoltare, debug, și optimizare de cod

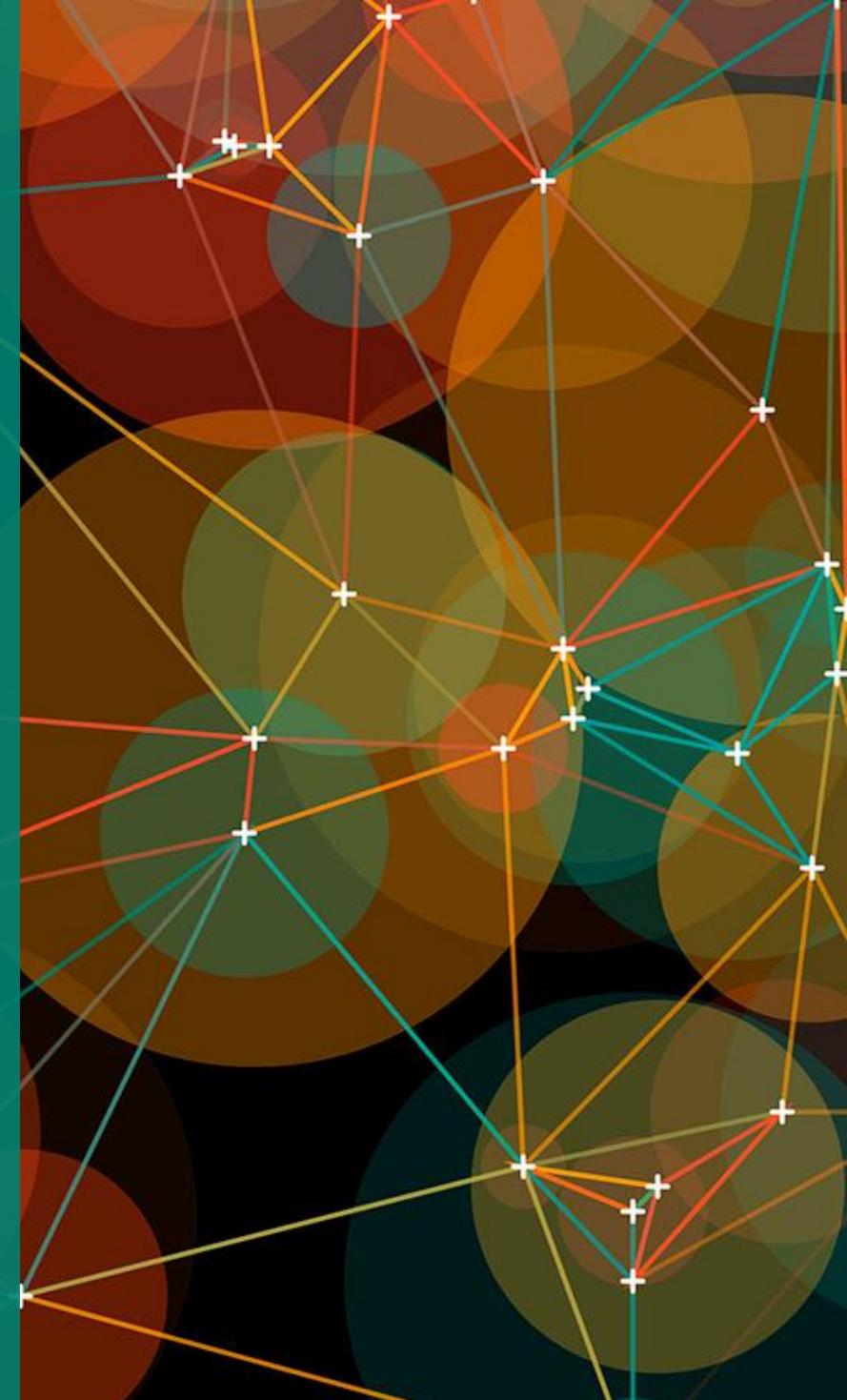
Interogări federalizate dealungul mai multor surse de date din Azure.

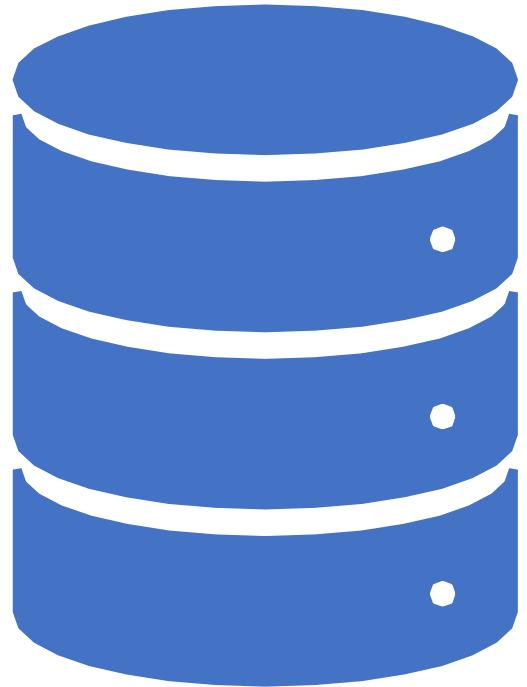
Sintetic





DEMO



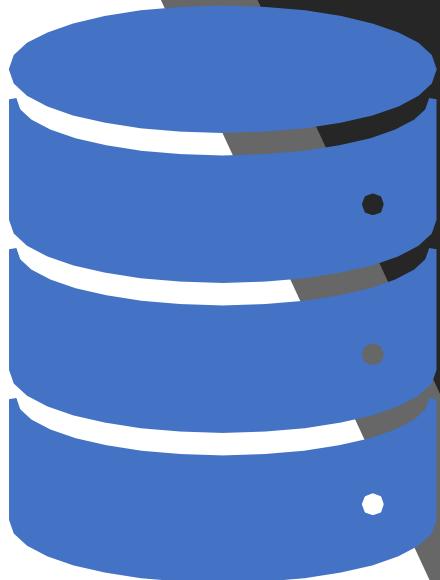


Arhitecturi Big Data

Conf.dr. Cristian KEVORCHIAN

Facultatea de Matematică și Informatică
Universitatea din București

BIG DATA



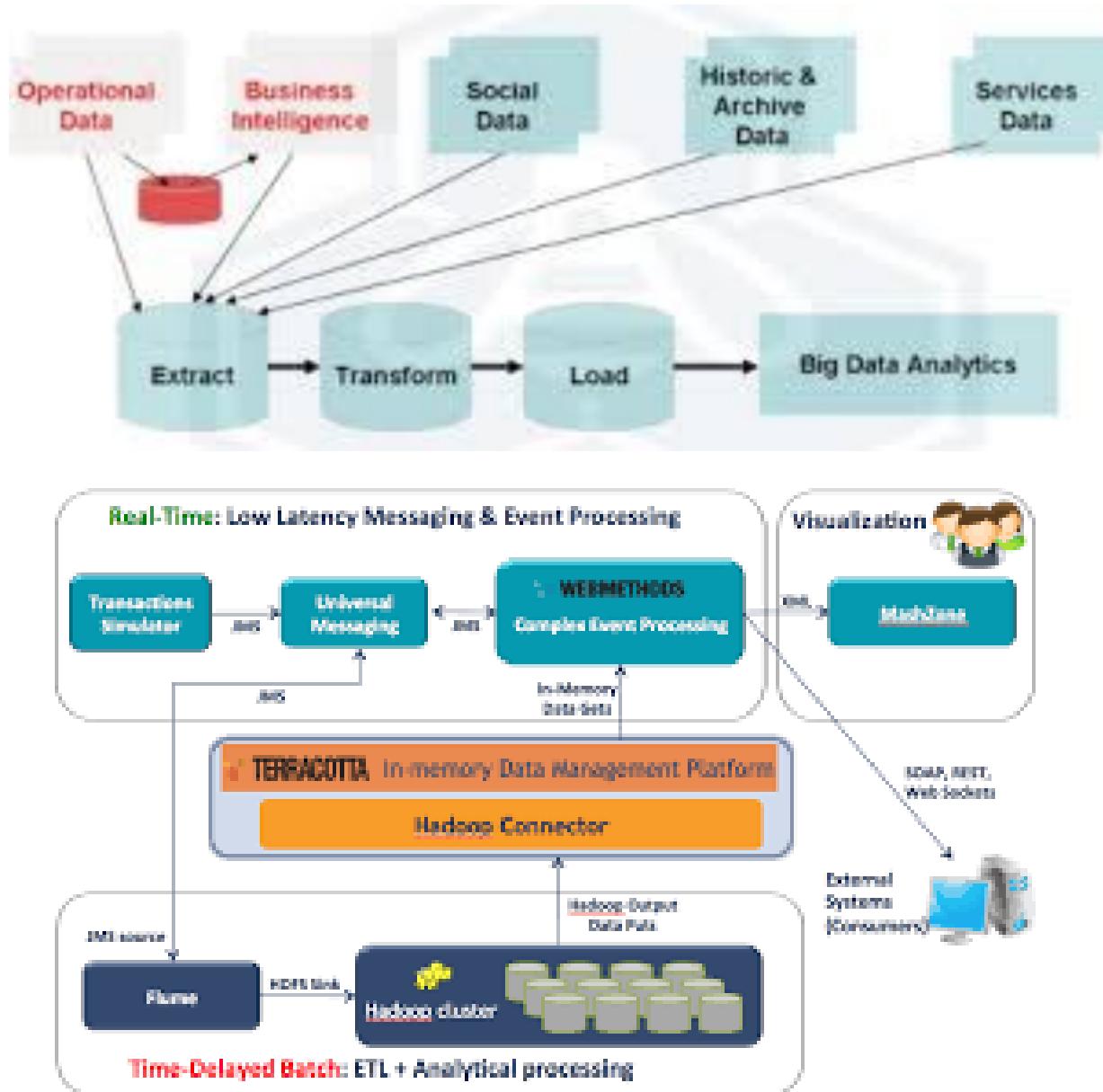
- Termenul “big data” caracterizează seturi de date de dimensiuni mari și complexitate ridicată pentru care aplicațiile tradiționale de procesare și analiză a acestora sunt inadecvate
- Încadrarea datelor în categoria “Big Data” poate fi făcută de la sute de Gb de date la zeci de petabyte.
- Din ce în ce mai mult, acest termen vizează în principal valoarea de business generată de analize complexe, decât volumul și complexitatea datelor utilizate.
- Costul stocării a scăzut dramatic, în timp ce mijloacele prin care sunt colectate datele continuă să se diversifice.
- Unele date sunt colectate și prelucrate în timp-real, iar altele sunt prelucrate lent ca date istorice.



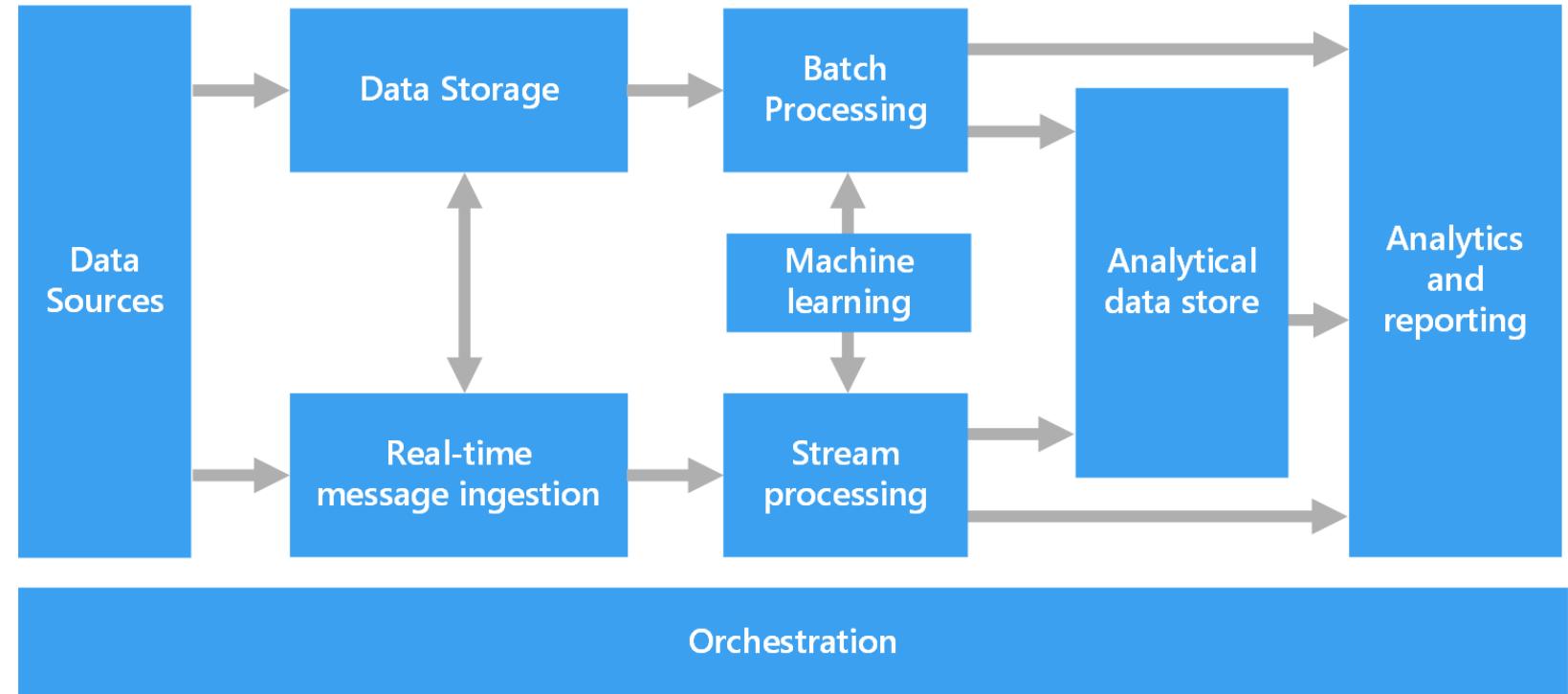
Soluții Big Data :

- Soluțiile Big Data implică una sau mai multe sarcini din următoarele:
 - Procesarea batch(serială) a volumelor mari de date statice .
 - Procesarea în timp-real a volumelor mari de date dinamice.
 - Explorarea interactivă a volumelor mari de date.
 - Analiza predictivă și machine learning.
- Trecerea la arhitecturi Big Data este necesară când:
 - Stochează și procesează date în volume prea mari pentru SGBD clasice
 - Transformă date nestructurate pentru analiză și raportare
 - Preia, procesează și analizează nelimitate stream-uri de date în timp real cu latențe neglijabile.

Batch vs Real-Time



Componentele unei arhitecturi Big Data



Sursa: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/>

Cele mai multe arhitecturi Big Data includ

- **Surse de date:** Toate soluțiile big data includ una sau mai multe surse de date:
 - Aplicații "data store", cum ar fi bazele de date relaționale.
 - Fișiere statice produse de aplicații cum ar fi fisiere log ale serverelor web.
 - Surse de date în timp-real, cum ar fi echipamentele IoT.
- **Stocarea datelor:** Datele asociate operațiilor de procesare batch sunt stocate prin intermediul unor sisteme de fișiere distribuite care pot cuprinde volume mari de date prezentate în diferite formate. O asemenea formă de stocare poartă numele de **data lake**. Aceste forme de implementare includ **Azure Data Lake Store** și containerele de BLOB-uri din **Azure Storage**



Procesarea datelor în sisteme Big Data

- **Procesare batch** – Datorită volumului mare de date, de multe ori o soluție Big Data trebuie să proceseze fișiere de date necesare execuției job-urilor batch, costisitoare din perspectiva timpului, pentru filtrare, agregare și pregătire pentru analiză. Opțiunile includ lansarea de job-uri **U-SQL** în **Azure Data Lake Analytics**, utilizând **Hive**, **Pig** sau job-uri **Map/Reduce** în clusterul **HDInsight** sau **Java**, **Scala** sau **Python** în clusterul **HDInsight Spark**
- **Preluarea în timp-real a mesajelor** – Dacă soluția include surse de date utilizate în timp-real, arhitectura trebuie să includă o soluție de preluare și stocare în timp-real a mesajelor pentru procesarea stream-urilor. Acest segment include opțiuni cum ar fi **Azure Event Hub**, **Azure IoT Hub** și **Kafka**.

Analiza Datelor Stocate

- Multe soluții în Big Data implică pregătirea anterioară a datelor pentru analiză și apoi transferul acestora către prelucrare într-un format structurat care poate fi interogat utilizând instrumente analitice.
- Datele analitice folosite pentru a răspunde acestor întrebări pot fi prezentate ca un **model de date Kimball** asociat unui data warehouse relațional, aşa cum se procedează în cele mai multe soluții de business intelligence (BI) tradiționale.
- Alternativ, datele ar putea fi prezentate printr-o tehnologie NoSQL cu latență redusă, cum ar fi **HBase**, sau o bază de date interactivă **Hive** care oferă o abstractizare a metadatelor peste fișierele de date din depozitul de date distribuite.
- Azure SQL Data Warehouse oferă un serviciu pentru stocarea date bazat pe cloud. **HDInsight** acceptă interactiv Hive, HBase și Spark SQL, care pot fi, de asemenea, folosite pentru a oferi date spre analiză.

Procesarea Fluxurilor de Mesaje

- După captarea mesajelor în timp-real, soluția arhitecturală trebuie să le includă o formă de procesare prin: filtrare, agregare dar și alte variante de pregătire a datelor pentru analiză.
- Datele procesate sunt apoi scrise într-un canal de ieșire.
- Azure Stream Analytics oferă un serviciu de procesare al fluxurilor de mesaje bazat pe interogări SQL care rulează permanent peste fluxuri nelimitate. De asemenea, pot fi utilizate tehnologii de streaming cum ar fi Storm și Spark Streaming într-un cluster HDInsight.

Aspecte privind Compatibilitatea

Azure Service Bus ~= RabbitMQ(Pivotal, 2013)

Azure Event Hubs ~= Apache Kafka

- Arhitectura Apache Kafka este foarte diferită de Azure Service Bus. În general, dacă performanța efectivă este un **must**, atunci Kafka este soluția potrivită.
- Dacă se dorește un serviciu managed echivalent dar nu asa sofisticat, atunci Azure Event Hub sau Amazon Kinesis pot fi opțiuni de luat în considerație.

Analiza și Raportarea

- Scopul soluțiilor software în Big Data este de a oferi informații prin intermediul analizelor și a realiza livrabile sub forma unor rapoarte complexe.
- Pentru a permite utilizatorilor să analizeze datele, arhitecturile Big Data include o componentă de modelare a datelor, sub forma unui cub multidimensional OLAP sau a unui model tabular de date în **Azure Analysis Services**.
- De asemenea, democratizarea BI-ul prin utilizarea tehnologiilor de modelare și vizualizare din Power BI sau Microsoft Excel facilitează efectuarea de analize și generarea de raporte de către specialistii în data science și analiștii de business.
- Pentru aceste scenarii, multe servicii Azure suportă notebook-uri analitice, cum ar fi **Jupyter**, care le permit acestor utilizatori să-și folosească abilitățile existente cu **Python** sau **R**. Pentru explorarea de date pe scară largă, puteți utiliza Microsoft R Server, independent sau cu Spark.

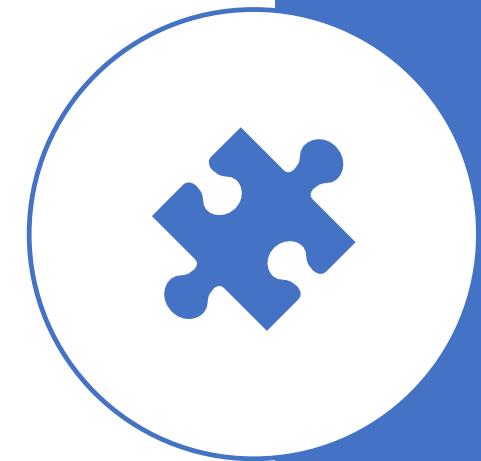
Servicii Azure pentru arhitecturi Big Data

- Azure include multe servicii care pot fi utilizate într-o arhitectură Big Data. Acestea se încadrează în două categorii:
 1. Serviciile managed: Azure Data Lake Store, Azure Data Lake Analytics, Azure Data Warehouse, Azure Stream Analytics, Azure Event Hub, Azure IoT Hub și Azure Data Factory.
 2. Tehnologiile open source bazate pe platforma Apache Hadoop, inclusiv HDFS, HBase, Hive, Pig, Spark, Storm, Oozie, Sqoop și Kafka. Aceste tehnologii sunt disponibile pe Azure în serviciul Azure HDInsight.
- Aceste opțiuni nu se exclud reciproc, iar multe soluții combină tehnologiile open source cu serviciile Azure.



Orchestrarea

- Cele mai multe soluții relative la ciclul de viață al datelor includ:
 - operații repetitive de procesare a datelor
 - includerea acestora în fluxuri de lucru în scopul transformării datelor sursă
 - transferarea datelor către diverse destinații de utilizare,
 - încărcarea datele procesate în data warehouse(Kimball) sau
 - Transferare rezultatelor direct în rapoarte sau grafice .
- Pentru a automatiza aceste fluxuri de lucru, puteți utiliza o tehnologie de orchestrare precum Azure Data Factory sau Apache Oozie și Sqoop



Tehnologie Lambda-Generalități

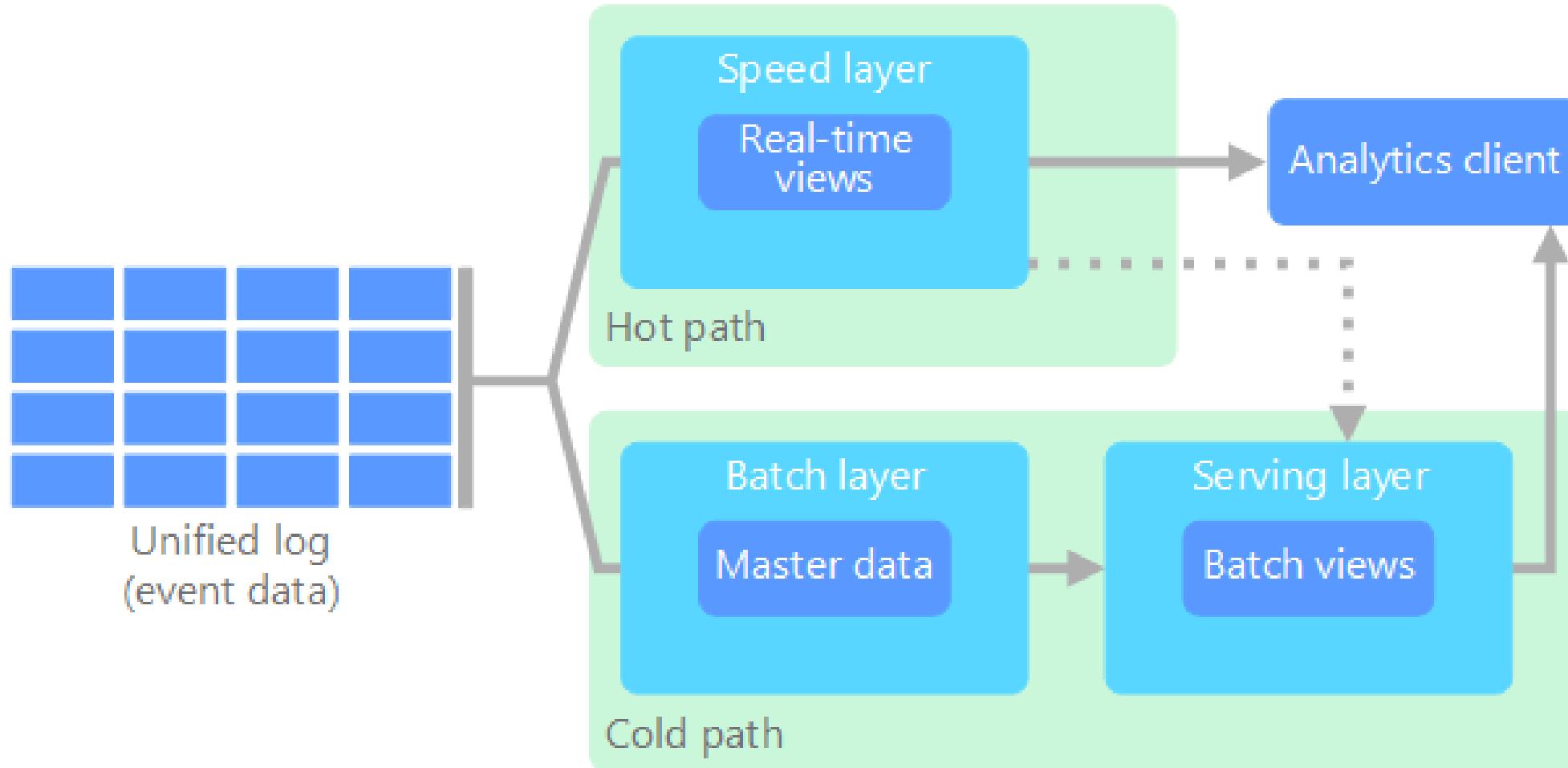
- Când operăm volume foarte mari de date, timpul de interogare poate fi foarte lung și inadecvat proceselor de business.
- Aceste interogări nu pot fi efectuate în timp real și necesită deseori tehnici de procesare paralelă cum ar fi MapReduce.
- Rezultatele sunt stocate separat de datele brute și folosite pentru interogare.
- Dezavantajul acestei abordări este faptul că introduce latență - dacă procesarea durează câteva ore, o interogare poate să întoarcă rezultate după câteva ore.
- Ideal, dorim rezultate în timp real (chiar cu o anumită pierdere de precizie) și combinăm aceste rezultate cu rezultatele din analizele istorice(batch).

Arhitecturi Lambda-Definiție

O lambda-arhitectură este o denumire generică pentru o arhitectură scalabilă, tolerantă-la-erori destinată procesării datelor în scenarii de prelucrare batch cu latențe reduse. (Nathan Marz, <http://lambda-architecture.net>)

- Batch layer(cale rece) - stochează toate datele primite în forma lor brută și efectuează prelucrarea loturilor de date. Rezultatul acestei procesări este stocat ca un **batch view**.
- Speed layer(calea fierbinte) - analizează datele în timp real. Acest nivel al arhitecturii este proiectat pentru o latență scăzută, în detrimentul preciziei.

Arhitectura Lambda



Sursa: <https://docs.microsoft.com/en-us/azure/architecture/data-guide/big-data/>

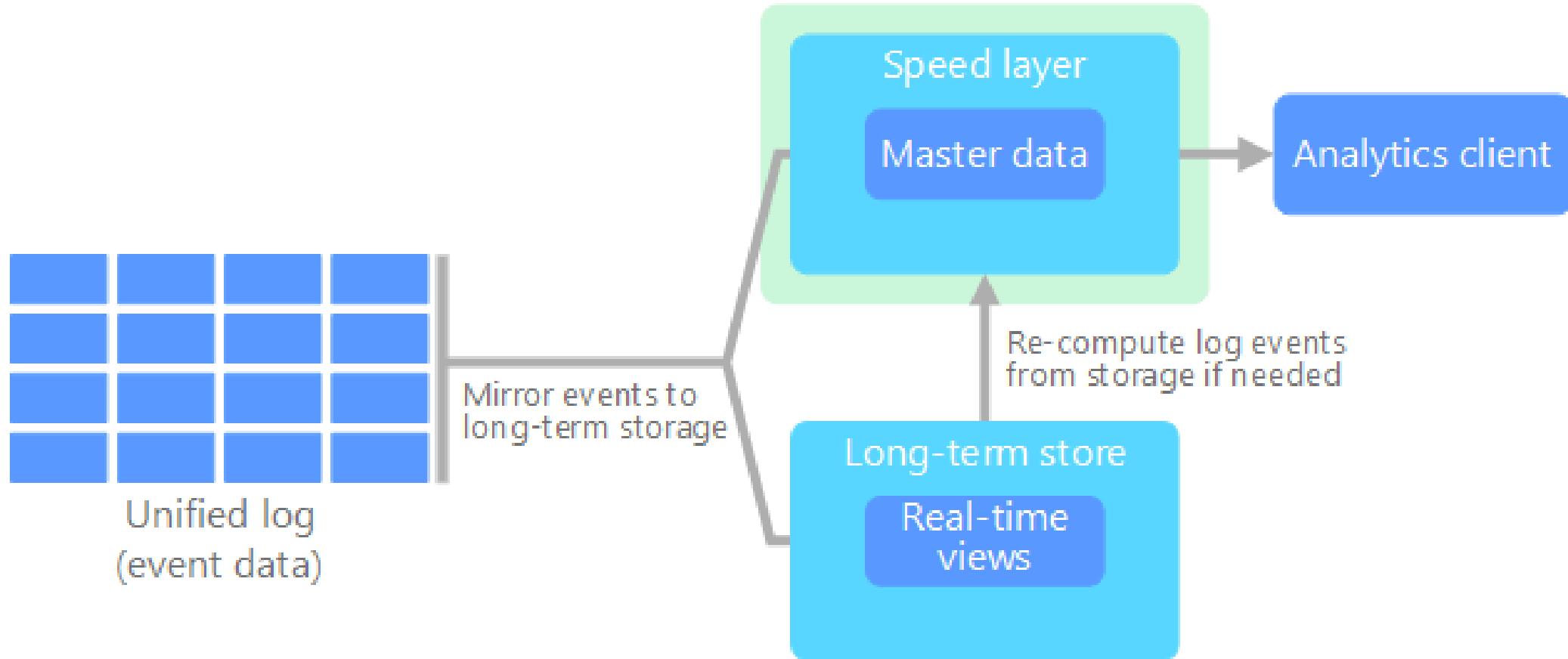
- Datele brute stocate prin "batch layer" sunt imuabile. Datele primite sunt întotdeauna adăugate la cele existente, iar datele anterioare nu sunt suprascrise niciodată.
- Orice modificare a valorii unei date este stocată ca o nouă înregistrare a cu amprentă-de-timp. Acest lucru permite o recompunere în orice moment al ciclului de viață al datelor colectate.
- Abilitatea de a recalcula orice "batch view" din datele originale este foarte importantă, deoarece permite crearea unor sisteme "batch view" în evoluție.

- **Batch layer** se alimentează într-un **Serving Layer** care indexează **Batch View** pentru o interogare eficientă. **Speed Layer** actualizează **Serving Layer** cu actualizări incrementale pe baza celor mai recente date.
- Datele preluate de "hot path" sunt restricționate de un anumit nivel al latenței impus de **Speed Layer** asa încât să poată fi prelucrate cât mai repede cu putință.
- Uneori este necesar un compromis între un anumit nivel de acuratețe a datelor în favoarea prelucrării cât mai rapide
- Datele preluate în "cold path", nu sunt supuse acelorași cerințe de latență redusă. Acest lucru permite calcularea cu precizii superioare a unor volume mari de date consumatoare de timp de procesare.

Arhitecturile Kappa

- Un dezavantaj al arhitecturii lambda este complexitatea sa ridicată.
- Logica de procesare apare la două ninele diferite – "cold path" și "hot path" - folosind diferite frameworkuri. Aceasta duce la dubla implementare a logicii de calcul dar și a complexității managementului arhitecturii pe ambele căi.
- Arhitectura **kappa** a fost propusă de **Jay Kreps** ca o alternativă la arhitectura lambda. Are aceleași obiective de bază ca arhitectura lambda, dar cu o distincție importantă: toate fluxurile de date se dirijază printr-o singură cale, folosind un sistem de procesare a fluxului.

Arhitetura Kappa



Kappa

- Există unele asemănări cu "batch layer" al arhitecturii lambda, prin faptul că datele asociate evenimentelor captate sunt imuabile.
- Datele sunt preluate ca flux-uri de evenimente într-un log unificat distribuit și tolerant la erori. Aceste evenimente sunt ordonate, iar starea curentă a unui eveniment este modificată numai de un nou eveniment care este atașat.
- Similar cu speed layer-ul arhitecturii lambda, toate procesările de evenimente sunt efectuate pe fluxul de intrare și persistent ca vizualizare în timp real.

Azure Service Bus

- Azure Service Bus este un sistem generic de mesagerie bazat pe cloud computing pentru conectarea la orice categorie de aplicații, servicii și dispozitive oriunde s-ar afla.
- Datorită faptului că este un sistem de mesagerie care se integrează cu arhitecturi multi-nivel;
- În acest context arhitectura pe mai multe nivele se limitează la sensul clasic în care separarea este în cea mai mare parte de natură logică, dar și foarte utilă acolo unde separarea este și fizică.
- Este util să se genereze arhitecturi scalabile.

Scenariu

- Dacă sistemul proiectat se instalează local într-o regiune, de ex. Australia de Est și presupunem că numărul utilizatorilor înregistrează o rată de creștere foarte mare distribuiți în Oceania și Europa.
- Pentru a oferi performanță bună utilizatorilor, este de preferat ca implementarea sistemului să fie realizată în centrele de date din Europe & Australia de Est.
- Cu toate acestea, pot apărea probleme de coerență a sistemului de baze de date. Ne putem baza pe funcția de replicare a SQL Server cu toate că acestea se pot dovedi a fi scumpe.

- În acest scenariu Service Bus, ca system de messaging între aplicații și servicii, poate fi de un real sprijin.
- Dacă plasăm funcția Service Bus + Role Worker între nivelul Service / API și nivelele de acces / bază de date; apoi în loc să se actualizeze direct baza de date, informațiile sunt transmise pe magistrala de servicii.
- Worker Role va prelua mesajul din coada de mesaje și va realiza atât actualizarea datelor din baza de date a regiunii respective (de exemplu, Europa), cât și dirijarea aceluiasi mesaj în Service Bus-ul celeilalte regiuni (de exemplu, Australia de Est) - al cărui Worker Rol va prelua acest mesaj și va actualiza baza de date a regiunii Australia de Est.

Azure Cloud Service

- Azure Cloud Services reprezintă o resursă Azure clasică, inițial introdusă în cloud-ul celor de la Microsoft în 2008.
- Scopul tehnologiei a fost acela de a face posibilă scalabilitatea aplicațiilor web dar și de execuția aplicațiilor din categoria "worker role" pe Windows.
- În timp ce Azure a avansat cu noile tehnologii de scalare pentru VM, serviciile clasice de tip ACS rămân în continuare o variantă de deployment pentru vechile medii de lucru din Azure.

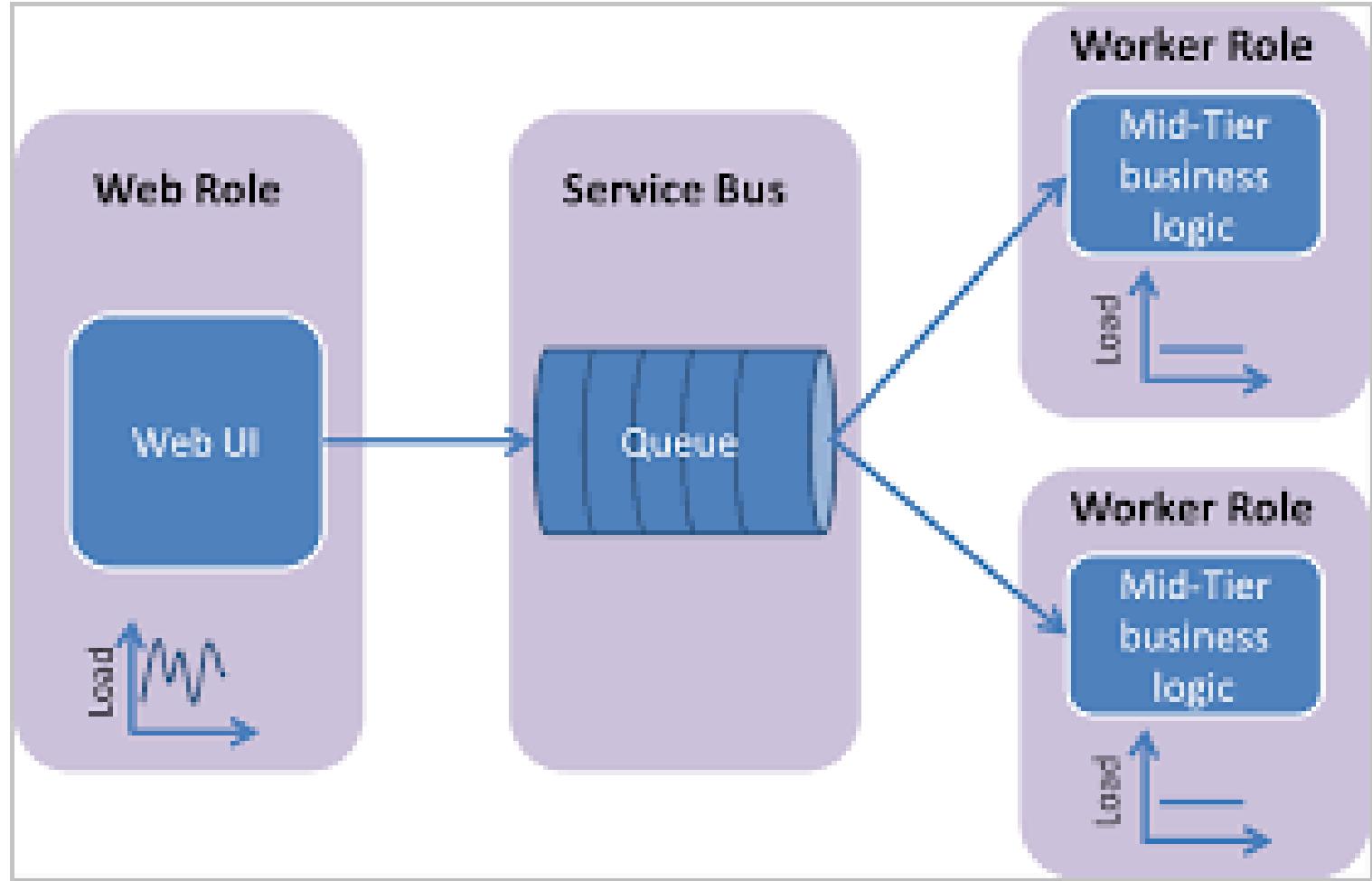
“Web Role” și “Worker Role”

- Web Role este un rol Cloud Service in Azure care este configurat si adaptat sa ruleze aplicatii web dezvoltate în limbaje/tehnologii de programare care sunt suportate de Internet Information Services (IIS), cum ar fi ASP.NET, PHP, WCF(Windows Communication Foundation) și Fast CGI.
- Worker Role este un rol in Azure care execută task-uri la nivel de aplicații și servicii, care în general nu cer IIS. In Worker Role, IIS nu este instalat implicit. Ele sunt utilizate în principal pentru a efectua procese în background suportând împreună cu rolurile web efectuarea de task-uri, cum ar fi comprimarea automată a imaginilor încărcate, rularea scripturilor atunci când se schimbă ceva în bază de date, obținerea de mesaje noi din coadă și procese și multe altele.

Diferențe între "Web Role" și "Worker Role"

- Principala diferență dintre cele două constă în următoarele:
 - un "Web Role" instalează automat și găzduiește aplicația utilizator în IIS
 - un "Worker Role" nu utilizează IIS și rulează aplicația în mod autonom, aceasta fiind instalată și livrată prin intermediul platformei de **Servicii Azure**, ambele pot fi gestionate în același mod și pot fi instalate pe același instanță Azure.
- În majoritatea scenariilor, instanțele **Web Role** și **Worker Role** lucrează împreună și sunt adesea folosite de o aplicație simultan. De exemplu, o instanță de Web Role ar putea accepta solicitări din partea utilizatorilor, apoi le transmită unei instanțe de Worker Role pentru procesare.

Web Role și
Worker Role



HDInsight: Hadoop și Hive

Conf.dr. Cristian Kevorchian
Facultatea de matematică și informatică

cristian.kevorchian@unibuc.ro

Big Data

MapReduce, Streaming, ML(Machine Learning),
MPP(Massively Parallel Processing)



"Scale Out" pentru
"Pay As You Go"



Schema pentru
Read Nu Write



BASE NU ACID

Prea Mare, Complex, sau Scump pentru mediile IT
exploataate curent

Noi probleme

Care sunt sentimentele de natură socială legate de un brand sau produs.

FEED-URI DE DATE ÎN TIMP REAL

ANALITICE
SOCIAL & WEB

Optimizarea flotei de transport bazată pe vreme și pattern-uri de trafic.

O predicție superioară a evenimentelor de business

ANALITICE AVANSATE

Tehnologii Big Data

Relational / Baze de Date Structurate(SQL Server și PDW)

Enterprise BI și Instrumente Analitice (Office, Analysis Services)

Multi-Dimensional și Tabular / Baze de Date Structurate (Servicii de analiză)

Hadoop
MapReduce
Stocare Distribuită & Procesare Date Multi-Structurate (HDInsight)

Surse de Date Multiple

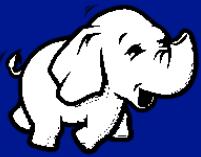
Procese Orientate Eveniment

SSBI(Self Service BI) și Instrumente Analitice (Office, Analysis Services)

Cloud (Azure) Flexibilitate+ Optiuni On-Premises

Apache Hadoop, Hortonworks, Hdinsight

Orchestrare de tehnologii



Apache Hadoop



Hortonworks



HDInsight

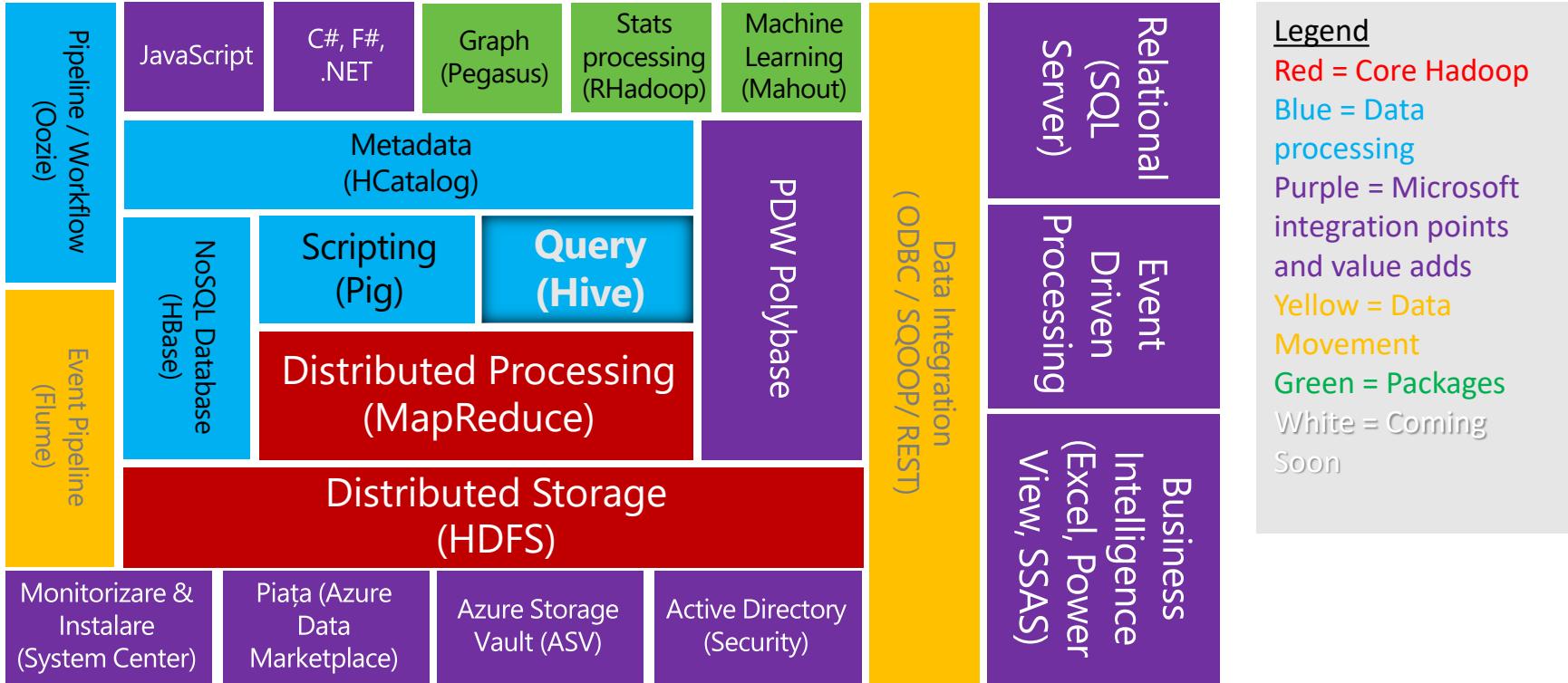
Comunitate Open Source
"Noi Consumăm Cod
Noi Contribuim la
realizarea Codului
Același "Core Code" în
toate distribuțiile"

Partener Microsoft
Important Actor în
Open Source Hadoop
Relație de "Trust" în
Comunitatea Open
Source

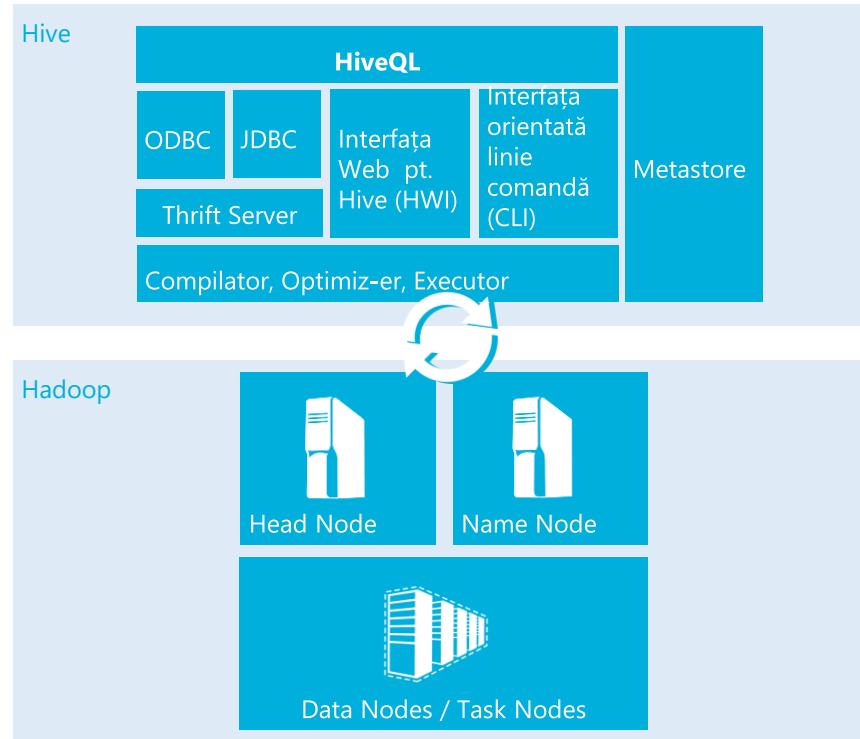
Serviciul HDInsight,
Server HDInsight pe
platforma Hortonworks

Funcționalități
adiționale

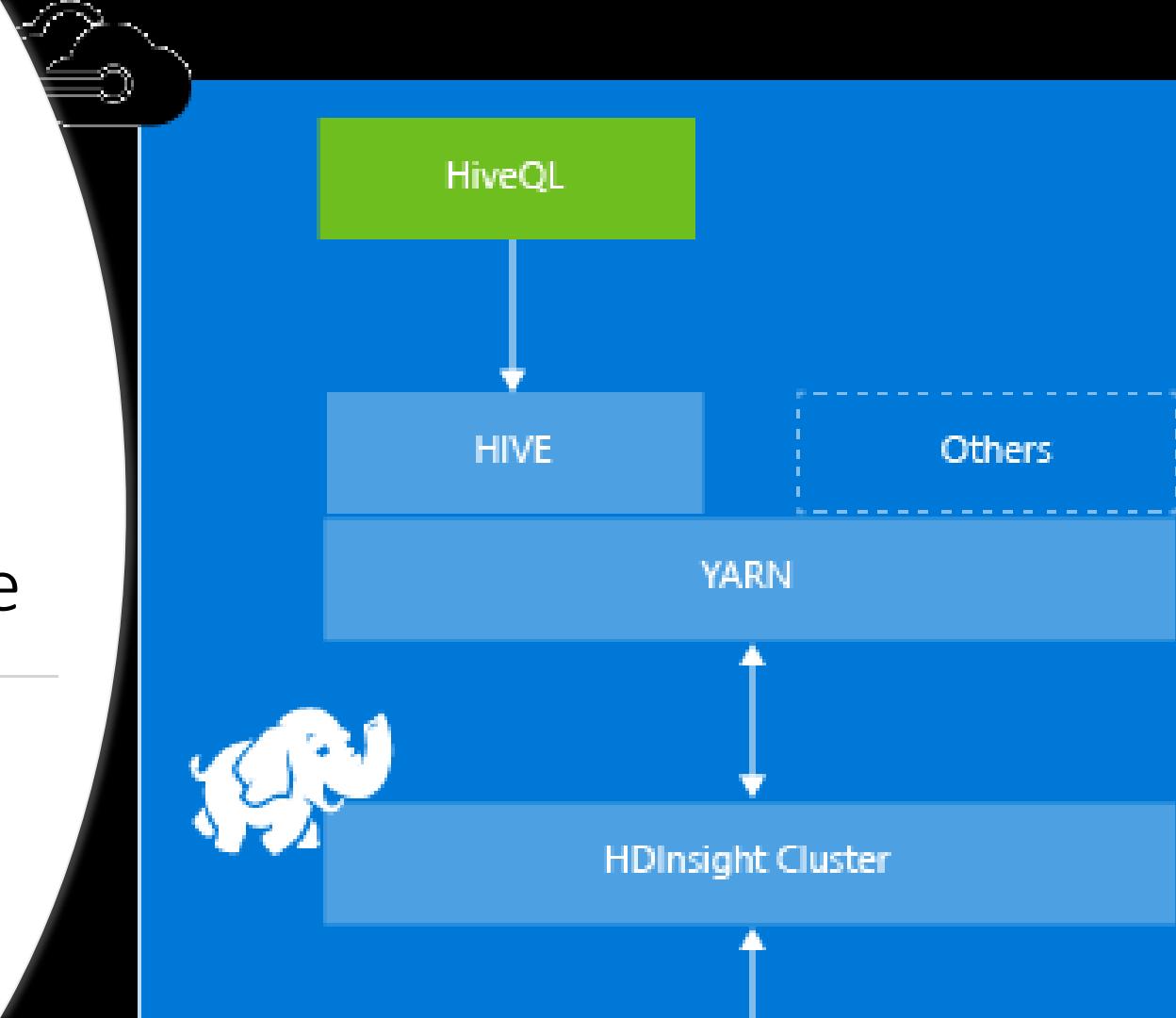
Arhitectura HDInsight / Hadoop



Arhitectura HIVE



Hadoop 2.0 Arhitectura Hive



Hive for Hadoop

- Facilitează lucrul cu instrumentele BI prin ODBC
- Structură fără model relațional complet.
- Utilizarea funcționalităților HiveQL
- Acces simplificat la datele Hadoop

Caracteristici Hive

- Orientat batch
- Orientat Data Warehouse
- Seturi de date
- Genereaza/executa MapReduce
- Indexare limitată, fără statistici(încarca statisticile asociate tabelelor în Derby), fara cache
- Optimizare in programare
- Numai adaugare

Hive pentru SQL Pro

- Asocierea proiectelor cu tehnologia adecvată
- Noi oportunități de business și tehnologice
- Căutare, arhivare, prototip, pre-agregare,
- algoritmi rafinați, etc



SQL/AS încă
necesare
pentru....

- Actualizări, OLTP, ACID
- Submultimi, indexari/agregări, optimizări, caching
- Aplicații, date, structuri, infrastructură deja existentă
- Fiecare interogare trebuie să fie rapidă

HIVE Fundamente

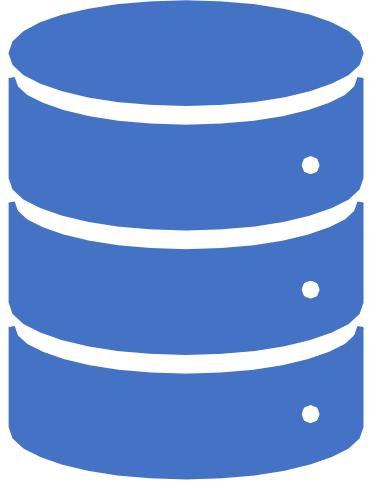
Creare Tabelă

Tabela ne-partitionată

```
CREATE EXTERNAL TABLE tabelaUnPart (type string COMMENT 'tip de tabela')
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
LOCATION '/user/demo/ckTabela';
```

Partitionată

```
CREATE EXTERNAL TABLE tabelaPart (type string COMMENT 'type of sectiune tabela')
COMMENT 'SQL Sectiune tabela'
PARTITIONED BY (an string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE;
ALTER TABLE tabelaPart ADD PARTITION (Year = '1')
LOCATION '/user/demo/cktabela1';
ALTER TABLE tabelaPart ADD PARTITION (Year = '2')
LOCATION '/user/demo/ckTabela2';
```



Detalii Hive Table

- DATA TYPES
- EXTERNAL / INTERNAL
- PARTITIONED BY | CLUSTERED BY | SKEWED BY
- Terminators
- ROW FORMAT DELIMITED | SERDE
- STORED AS
- Fields/Collection Items/Map Keys
- TERMINATED BY
- LOCATION

MetaDate

- Metadatele sunt încărcate în baze de date MetaStore cum ar fi:
 - Derby
 - SQL Azure
 - SQL Server
- Vizualizarea Schemei
 - SHOW TABLES 'ta.*';
 - DESCRIBE tabelaunpart;
 - DESCRIBE tabelaunpart.type;
 - DESCRIBE EXTENDED tabelaunpart;
 - DESCRIBE FORMATTED tabelaunpart;
 - SHOW FUNCTIONS "x.*";
 - SHOW FORMATTED INDEXES ON tabelaunpart;

Tipuri de Date

Primitive

- Numerice: Int, SmallInt, TinyInt, BigInt, Float, Double
- Caracter: String
- Speciale: Binary, Timestamp

Colectii

- STRUCT<Oras:String, Judet:String> | Struct ('Cluj', 'Cluj')
- ARRAY <String> | Array ('Cluj', 'Cluj')
- MAP <String, String> | Map ('City', 'Cluj', 'State', 'Cluj')
- UNIONTYPE <BigInt, String, Float>

Proprietăți

- Lungime stringului nu conteaza
- NULL depinde de SerDe

Stocare – Externă și Internă

```
CREATE EXTERNAL TABLE tabelaUnPart(...)  
LOCATION '/user/demo/tabela';  
LOCATION 'hdfs:///user/demo/tabela';  
LOCATION 'asv://user/demo/tabela';
```

- Utilizăm EXTERNAL cînd datele au fost utilizate în afara Hive.
- Este necesar ca datele sa persiste și după DROP TABLE
 - Utilizăm o locatie indicata de utilizator cum ar fi ASV(Azure Storage Volume)
 - Hive should not own data and control settings, dirs, etc.
 - Use INTERNAL when you want Hive to manage the data and storage for short term usage
 - Creating table based on existing table (AS SELECT)

Storage – Partition și Bucket

- CREATE EXTERNAL TABLE tabelaPart (...) PARTITIONED BY (an string) CLUSTERED BY (type) into 256 BUCKETS
- Partitia
 - Directoare pentru fiecare combinație distinctă de valori partitionate
 - Cheia de partitionare nu poate fi definită în tabela.
 - Permite eliminarea de partii
 - Useful in cauterile pe nivale
 - Poate conduce la o slaba performanta daca partitia nu este referita in querry
- Bucket-uri
 - Splitarea datelor bazate pa hash-ul unei coloane
 - Un fisier HDFS per bucket în cadrul partitiei
 - Performanta poate fi imbunatatita pentru agregări și joinuri
 - Esantionare
 - set hive.enforce.bucketing = true;

Storage – Formatul Fisierului

CREATE EXTERNAL TABLE tabelaPart (...)
ROW FORMAT DELIMITED
FIELDS TERMINATED by '\001'
STORED AS TEXTFILE, FISIERERC, FISIERE
de SECVENTE, AVRO(structure complexe
de date cu MapReduce)

- Format
 - În general se sterg headerele înaintea încărcării fișierelor
 - TEXTFILE este comun, util cind datele sunt alfanumerice și partajate
 - Format de stocare extensibil.
 - Extensibil pe disk/reprezentare in-memory



Încărcare— SerDes

```
CREATE EXTERNAL TABLE ClientSerDeUtil(...)  
  ROW FORMAT SERDE  
  'com.cloudera.hive.serde.JSONSerDe'  
  LOCATION ....
```

- SerDes(Serializare/Deserializare)
 - Crează la nivel de utilizator Java Serializare/Deserializare
 - Include input/output parsing, optimizare
 - În mod ușual suprascrie CREATE TABLE DDL
 - SerDes uzuale: CSV, XML, JSON
 - bibliotecă: org.apache.hadoop.hive.serde2

Storage

-HDFS si

ASV

- ASV://containername@account.blob.core.windows.net/user/demo/
 - HDFS:///user/demo/
-
- Format de stocare
 - HDFS este Hadoop Distributed File System
 - ASV este Azure Storage Vault utilizînd un API îñ top HDFS
 - ASV permite reutilizarea peste clustere si cu alte aplicatii
 - ASV disponibilizeaza date rapid către clustere HDInsight

CREARE INDEX

- CREATE INDEX tabelaPart_idx
 ON TABLE tabelaPart (type)
 AS 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler'
 WITH DEFERRED REBUILD
 IN TABLE tabelaPart_index;
• ALTER INDEX tabelaPart_idx ON tabelaPart REBUILD;
 - Chei
 - Fara chei
 - Datele asociate indexului reprezinta o alta tabela
 - Se impune REBUILD pentru a include noile date
 - SHOW FORMATTED INDEXES on ckTable;
 - Indexarea poate fi utila la
 - Creare unui număr mare de partitii mici
 - GROUP BY



Creare View-uri

- CREATE VIEW tabelaUnAn (type)
- AS SELECT type
- FROM tabelaPart
- WHERE year = 1
- ORDER BY type;
- Secvență Simplă de Cod
 - SELECT * FROM tabelaUnAn;
DESCRIBE FORMATTED tabelaUnAn;
- Puncte cheie
 - Nu este materializata
 - Poate avea ORDER BY sau LIMIT

Interogari

Interogări

- ```
SELECT c.state_fips, c.county_fips, c.population
 FROM census c
 WHERE c.median_household_income > 100000
 GROUP BY c.state_fips, c.county_fips, c.population
 ORDER BY county_fips
 LIMIT 100;
```
- Puncte cheie
  - Cash minimal, statistici, sau optimizări
  - În general citeste întregul volum de date pentru fiecare interogare
- Performanță
  - Ordinea coloanelor, tablelele pot face diferență în zona performantei
  - Eliminarea partiiilor filtrarea rangului

# Sortare

- ORDER BY
  - Sortul final realizat cu un reducer poate ridica probleme.
- SORT BY
  - Sortarea în cadrul unui reducer poate aduce performante mai bune
- DISTRIBUTE BY
  - Determină modul de interconectare a datelor implica distribuția reduce-rilor.
- SORT BY + DISTRIBUTE BY = CLUSTER BY
  - Poate copia comportamentul ORDER BY, cu performante mai bune în context de distribuire.

# Join-uri

---

- Tipuri de Join în HIVE
  - Equality
  - OUTER - LEFT, RIGHT, FULL
  - LEFT SEMI
- Nu permite lucrul cu join-uri de tip:
  - Non-Equality
  - IN/EXISTS sub-queries (rewrite as LEFT SEMI JOIN)
- Caracteristici
  - Job-uri MapReduce multiple cu exceptia cazului in care avem aceleasi coloane in join in toate tabelele
  - Plaseaza cele mai mari tabele ultimele pentru economie de memorie
  - Joinurile sunt solutionate de la stînga la dreapta in ordinea interogării.
  - JOIN ON este complet evaluată inaintea pornirii lui WHERE.

# HIVE EXPLICATII

- EXPLAIN SELECT \* FROM tabelaPart;
- EXPLAIN SELECT \* FROM tabelaPart WHERE year > 1;
- EXPLAIN EXTENDED SELECT \* FROM tabelaPart;
- Caracteristici
- Nu executa interogarea
- Prezinta parsing-ul
- Lista stage-urilor, fișierelor temp, dependințelor, modes, output operators, etc.
- ABSTRACT SYNTAX TREE:
  - (TOK\_QUERY (TOK\_FROM (TOK\_TABREF (TOK\_TABNAME tabelaPart)))) (TOK\_INSERT (TOK\_DESTINATION (TOK\_DIR TOK\_TMP\_FIL • E)) (TOK\_SELECT (TOK\_SELEXPR TOK\_ALLCOLREF))))
- STAGE DEPENDENCIES:
  - Stage-0 is a root stage
- STAGE PLANS:
  - Stage: Stage-0
  - Fetch Operator
  - limit: -1

# Configurare

# Configurare HIVE

---

- cd %hive\_home%\bin
- <install-dir> curent: C:\Hadoop\hadoop-1.1.0-SNAPSHOT
- Hive configurare implicita<install-dir>/conf/hive-default.xml
- Configurare<install-dir>/conf/hive-site.xml
- Hive configurare director HIVE\_CONF\_DIR variabile de mediu
- Log4j configurare <install-dir>/conf/hive-log4j.properties
- Log: c:\Hadoop\hive-0.9.0\logs\hive.log

# Utilizare HIVE

---

- BI în Big Data
- “Polenizare încrucișată” a capabilităților SQL!
- Corelatii-incrucisate cu Hadoop, join-uri, filtre,etc
- Permite încărcarea rezultatelor intermediare pentru accelerarea/simplificarea interogărilor
- Procesare orientată pe procese batch
- O abordare insight E2E poate fi mult mai rapida
- Proiecte potrivite la tehnologii potrivite

# Pasul Următor

---

- De citit
  - [http://sqlblog.com/blogs/lara\\_rubbelke/archive/2012/09/10/big-data-learning-resources.aspx](http://sqlblog.com/blogs/lara_rubbelke/archive/2012/09/10/big-data-learning-resources.aspx)
  - [Programming Hive Book](#)
  - <http://blogs.msdn.com/cindygross>
- Descarca Microsoft HDInsight Server <http://microsoft.com/bigdata> (On-Prem CTP)
- Gindeste cum poate Big Data fi utilizat in viata de zi cu zi.
- Incearca sa gindesti proiectul cit mai critic

# Referințe pentru Big Data

Hadoop: The Definitive Guide by Tom White

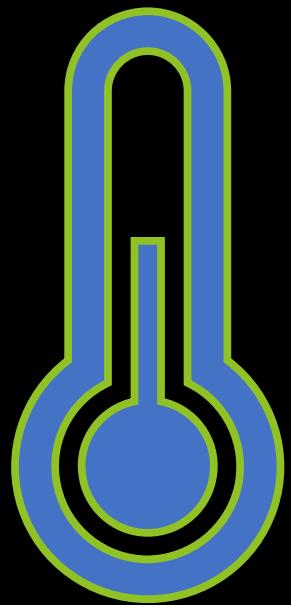
SQL Server Sqoop <http://bit.ly/rulsjX>

Hive <http://hive.apache.org>

Excel2Hadoop via Hive ODBC <http://tinyurl.com/7c4qjjj>

Hadoop pe Azure Videos <http://tinyurl.com/6munnx2>

Microsoft Big Data <http://microsoft.com/bigdata>



# SERVICIUL AZURE DATABRICKS

CONF.DR. CRISTIAN KEVORCHIAN  
UNIVERSITATEA DIN BUCUREŞTI  
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

THIS IS YOUR MACHINE LEARNING SYSTEM?

YUP! YOU POUR THE DATA INTO THIS BIG  
PILE OF LINEAR ALGEBRA, THEN COLLECT  
THE ANSWERS ON THE OTHER SIDE.

WHAT IF THE ANSWERS ARE WRONG?

JUST STIR THE PILE UNTIL  
THEY START LOOKING RIGHT.



## Developer Services



Visual Studio Team Services



Azure DevTest Labs



VS Application Insights\*



HockeyApp



Developer Tools

## Management & Security



Azure Portal



Scheduler



Automation



Log Analytics



Key Vault



Security Center\*

### Compute

- Virtual Machines
- Virtual Machine Scale Sets
- Cloud Services
- Batch
- RemoteApp
- Service Fabric
- Azure Container Service

### Web & Mobile

- Web Apps
- Mobile Apps
- Logic Apps\*
- API Apps
- API Management
- Notification Hubs
- Mobile Engagement
- Functions\*

### Data & Storage

- SQL Database
- DocumentDB
- Redis Cache
- Storage: Blobs, Tables, Queues, Files and Disks
- StorSimple
- Search
- SQL Data Warehouse\*
- SQL Server Stretch Database\*

### Analytics

- Data Lake Analytics\*
- Data Lake Store\*
- HDInsight
- Machine Learning
- Stream Analytics
- Data Factory
- Data Catalog
- Power BI Embedded\*

### Internet of Things & Intelligence

- Azure IoT Suite
- Azure IoT Hub
- Event Hubs
- Cortana Intelligence Suite
- Cognitive Services\*

### Media & CDN

- Media Services
- Content Delivery Network

### Identity & Access Management

- Azure Active Directory
- B2C\*
- Domain Services\*
- Multi-Factor Authentication

## Hybrid Integration



BizTalk Services



Service Bus

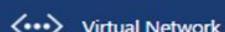


Backup



Site Recovery

## Networking



Virtual Network



ExpressRoute



Traffic Manager



Load Balancer



Azure DNS\*

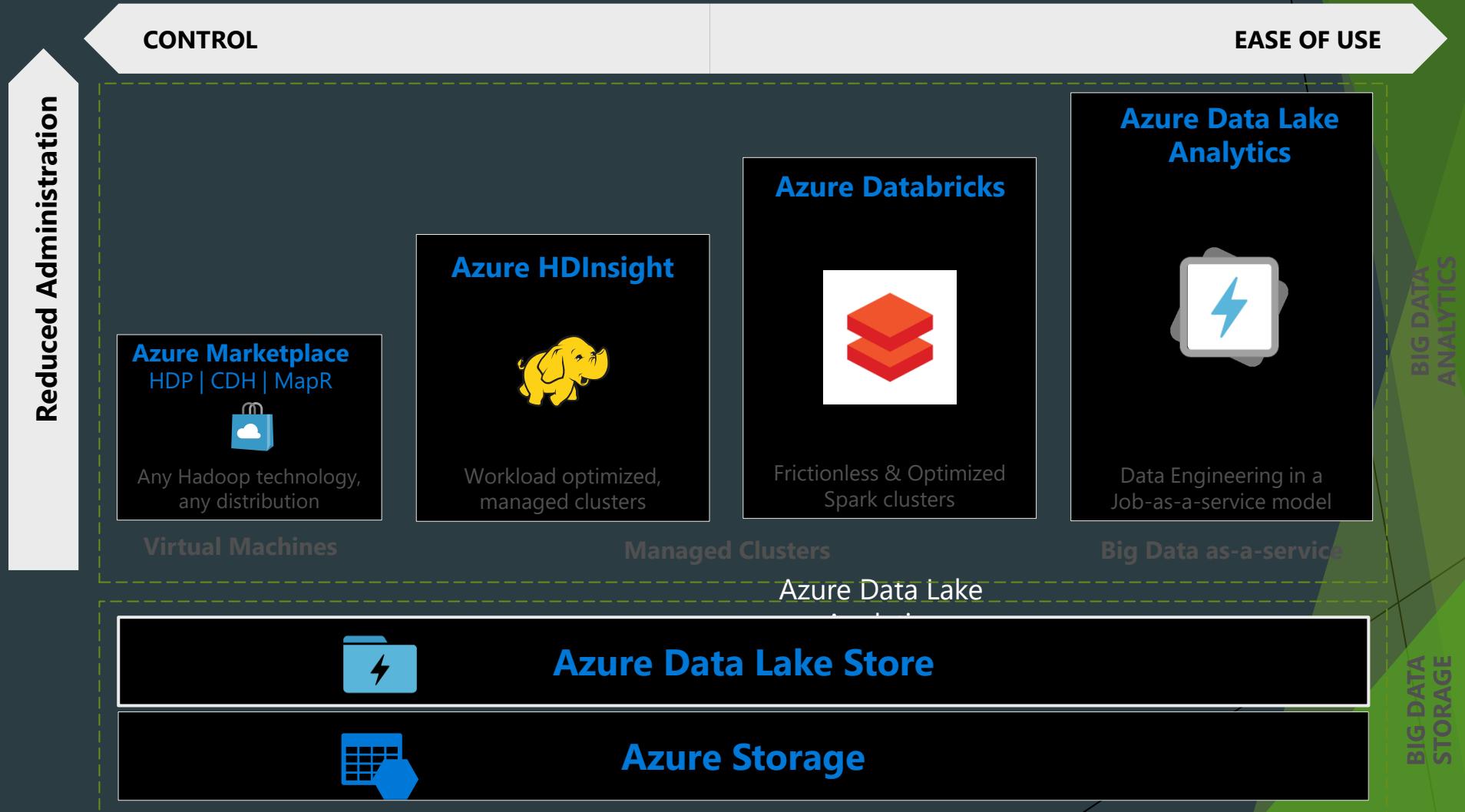


VPN Gateway

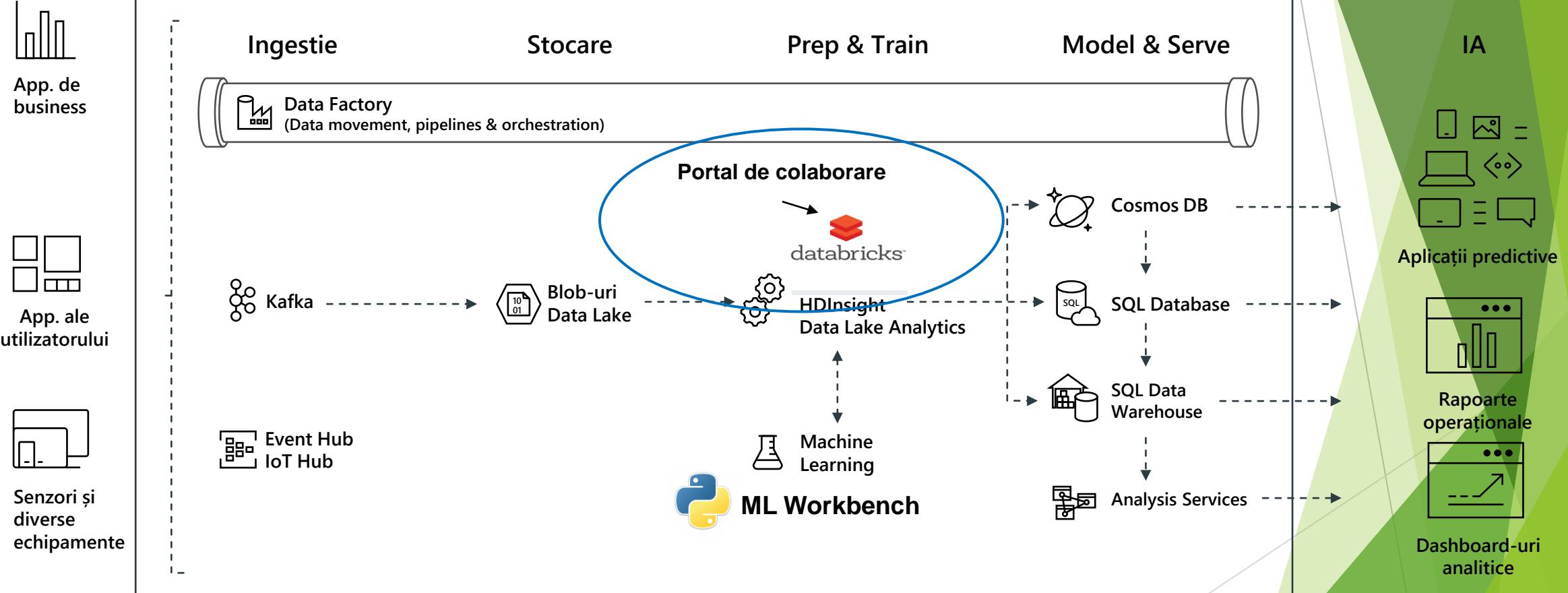


Application Gateway

# BIG DATA IN CLOUD(AZURE)



# BIG DATA & SISTEME DE ANALITICE



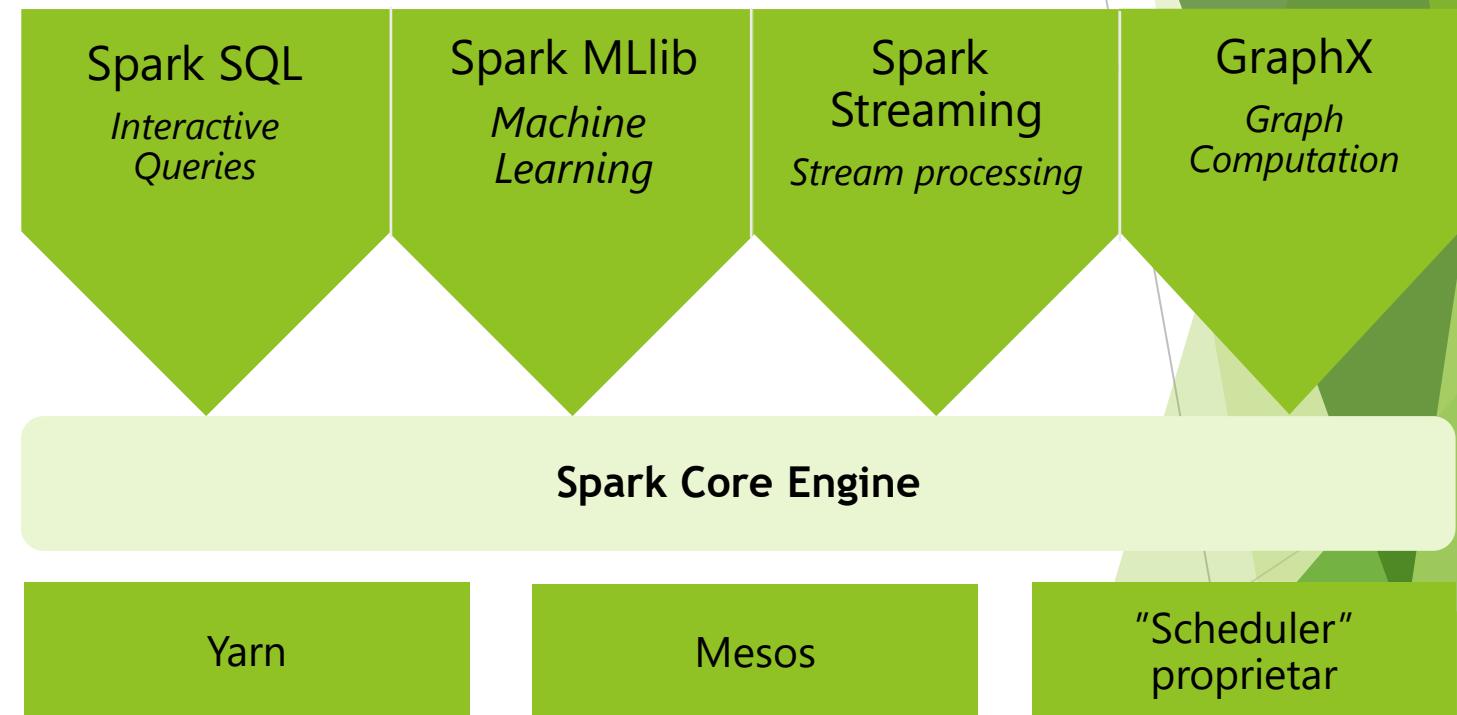
# Azure Databricks Şı Apache Spark

# A P A C H E S P A R K

Sistem open source, paralel inclusând un framework pentru procesare date pentru  
"Big Data Analytics"

## Spark unifică:

- **Procesări Batch**
- **SQL Interactiv**
- **Procesare real-time**
- **Machine Learning**
- **Deep Learning**
- **Graph Processing**



DATA BRICKS

## Azure Databricks este un serviciu în Azure.

- Bazele de date Azure sunt integrate cu serviciile Azure Databricks
- Serviciile de stocare Azure permit accesarea datelor din Azure Blob Storage și Azure Data Lake Store
- Azure Active Directory permite autentificarea utilizatorilor, eliminând necesitatea menținerii a două familii de utilizări separate în Databricks și Azure.
- Azure SQL DW și Azure Cosmos DB permit combinarea datelor structurate și nestructurate pentru lucru cu analitice
- Apache Kafka pentru HDInsight permite lucru cu fluxuri de date
- Azure Power BI pentru vizualizarea datelor



# STRUCTURA AZURE DATABRICKS

## Azure Databricks



IoT / streaming data



Cloud storage



Data warehouses



Hadoop storage

### Mediu colaborativ



DATA ENGINEER



DATA SCIENTIST



BUSINESS ANALYST

### Deployment de Job-uri de Producție & Workflow-uri



MULTI-STAGE PIPELINES



JOB SCHEDULER



NOTIFICATION & LOGS

### Motoare de Runtime Optimizezate pentru Databricks



DATABRICKS I/O



APACHE SPARK



SERVERLESS



Rest APIs



Machine learning models



BI tools



Data exports



Data warehouses

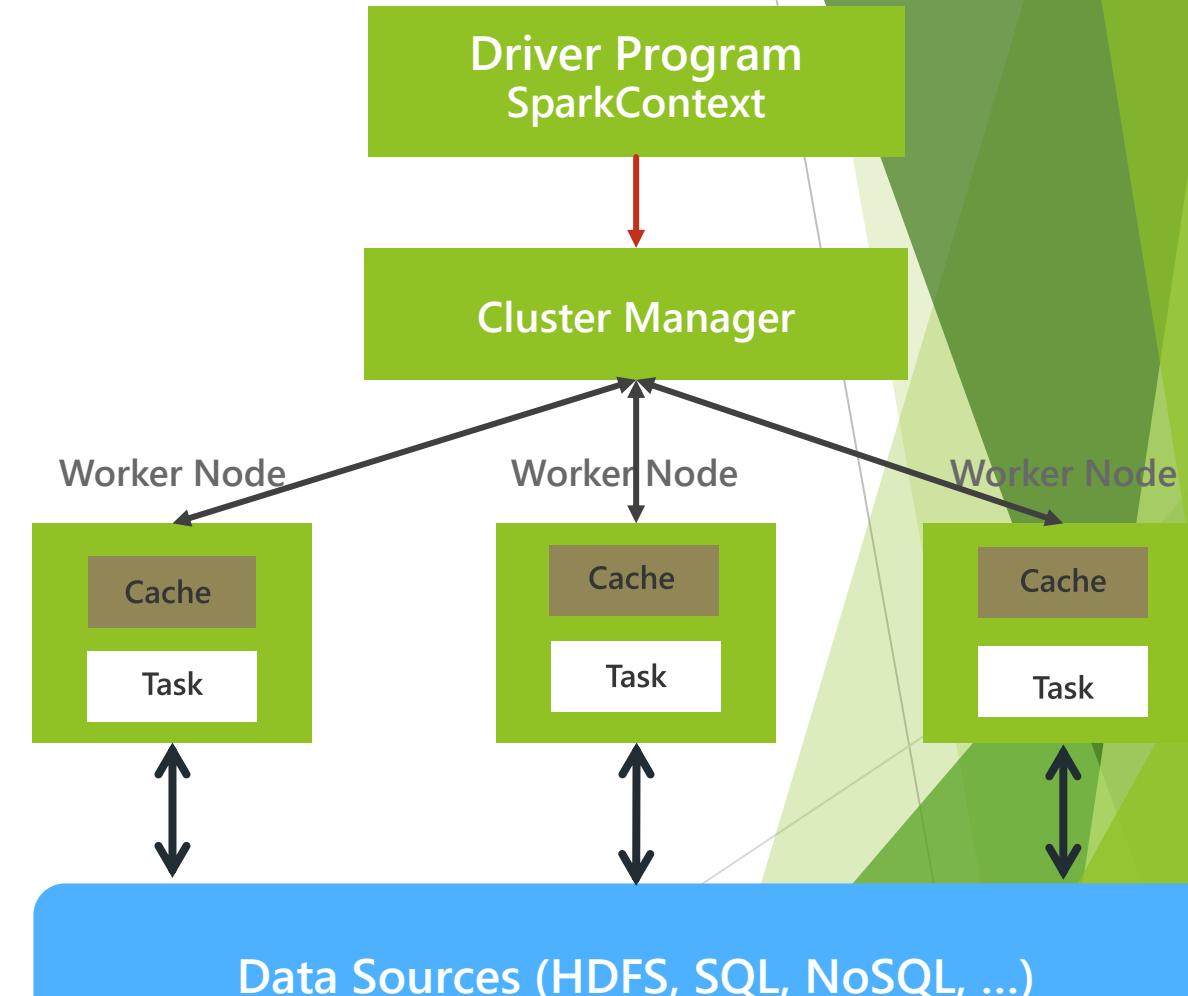
Productivitate îmbunătățită

Build securizat

Scalare fără limite

- Spark Driveste un proces JVM care găzduiește SparkContext pentru o app Spark.
- Spark Context este end-point-ul serviciului Spark(execution engine) și componenta centrală a unei app Spark
- Rezultatele operațiilor sunt colectate de driver
- "Worker node"-urile citesc și scriu date din/în Sursele de Date incluzând HDFS.
- Un "worker node" (cache) transformă datele în RDD-uri (Resilient Distributed Data sets).
- "Worker node"-urile și "Driver Node"-urile sunt executate ca MV in cloud public (AWS, Google and Azure).

## ARHITECTURA CLUSTERULUI SPARK



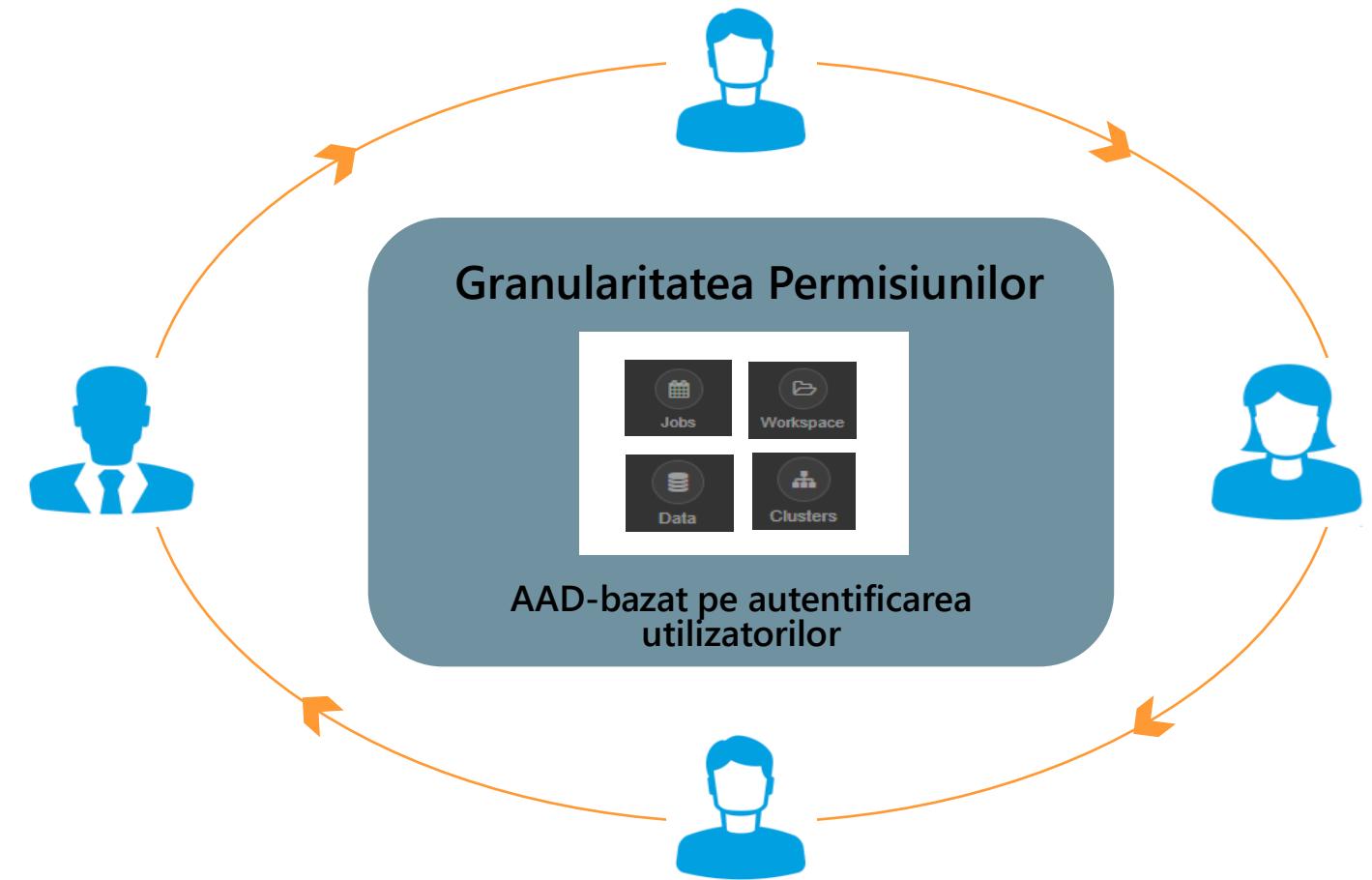
# Spark Context

```
Cmd 11

1 dataPath = "/databricks-datasets/Rdatasets/data-001/csv/ggplot2/diamonds.csv"
2 diamonds = spark.read.format("com.databricks.spark.csv") \
3 .option("header", "true") \
4 .option("inferSchema", "true") \
5 .load(dataPath)
6
7 # inferSchema means we will automatically figure out column types
8 # at a cost of reading the data more than once
```

SparkContext reprezintă punctul de intrare al funcționalităților în Spark. Cel mai important pas al oricărei aplicații driver în Spark este de a genera SparkContext. Acesta, permite aplicației utilizatorului să acceseze cluster-ul Sparc cu ajutorul Resource Manager. Managerul de resurse poate fi unul dintre : Spark Standalone, YARN, Apache Mesos.

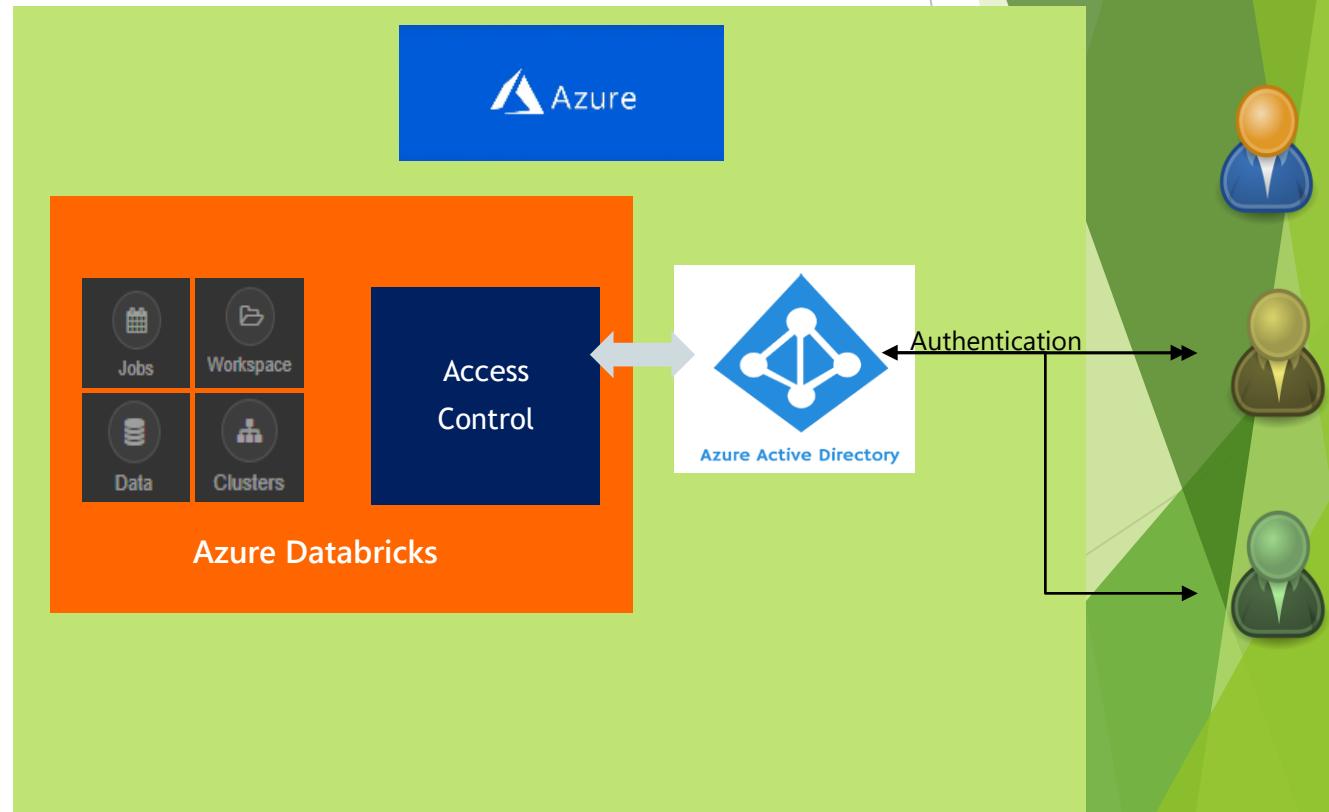
În Azure Databricks se pot partaja în deplină siguranță artefacte cum ar fi Clusteré, Notebook-uri, Job-uri și Workspace-uri



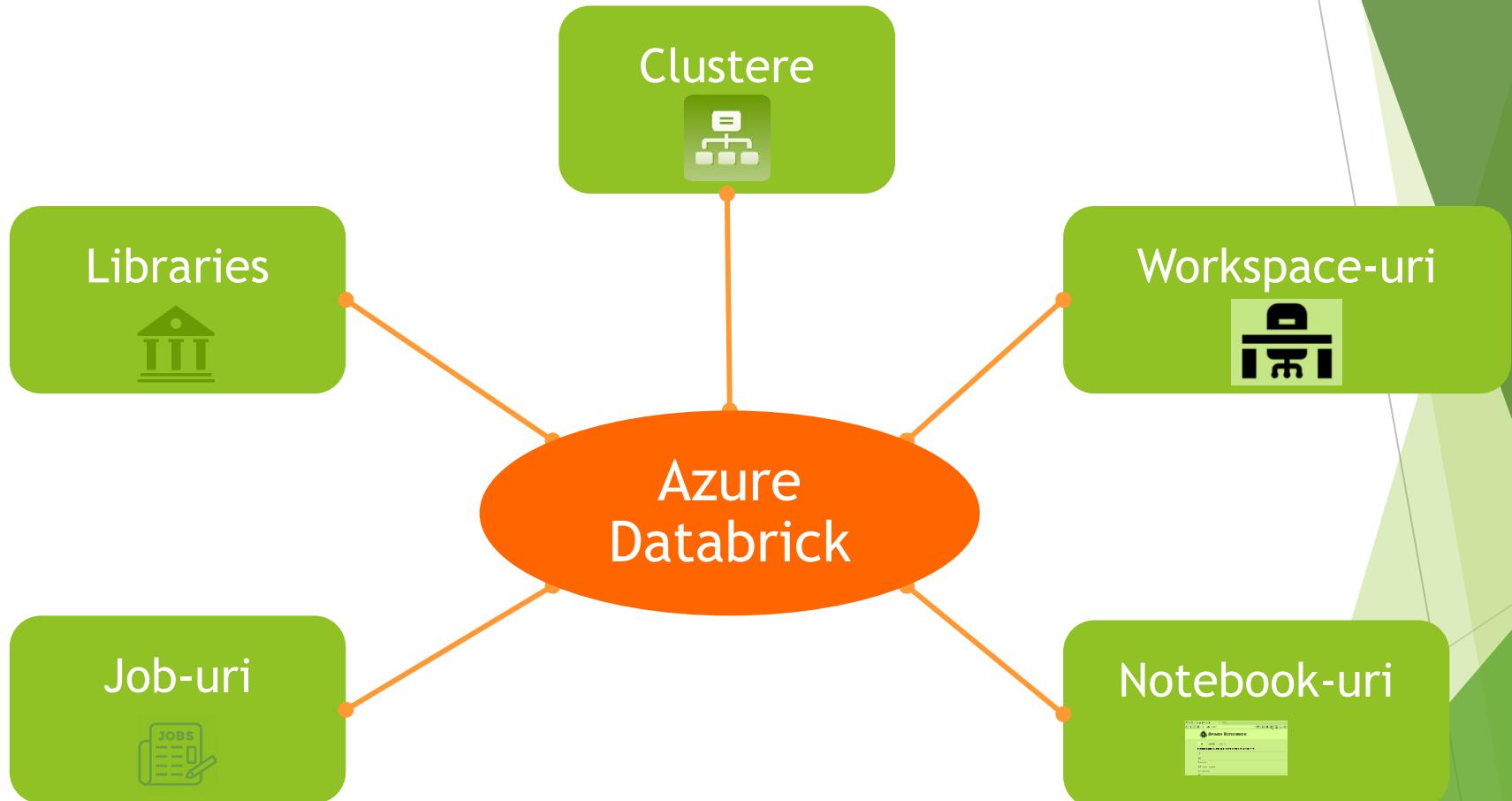
Azure Databricks este un mediu sigur  
pentru colaborare

# AZURE DATABRICKS SE INTEGREAZĂ CU AAD

- Nu este nevoie de definit utilizatori — nivelul permisiunilor la nivelul Databricks nu este necesar.
- Databricks a realizat delegarea și autentificarea către AAD prin SSO(single-sign on ).
- *Notebook-urile, și output-urile asociate, sunt încărcate în contul Databricks.* Totuși, asigură faptul că numai utilizatorii autorizați pot avea access.



# ARTEFACTE SPECIFIQUE AZURE DATABRICKS



# Workspace Colaborativ

## UN MEDIU DE LUCRU INTUITIV

Access la date prin intermediul notebook-urilor interactive bazate pe limbaje cum ar fi R, Python, Scala, and SQL

## COLABORARE

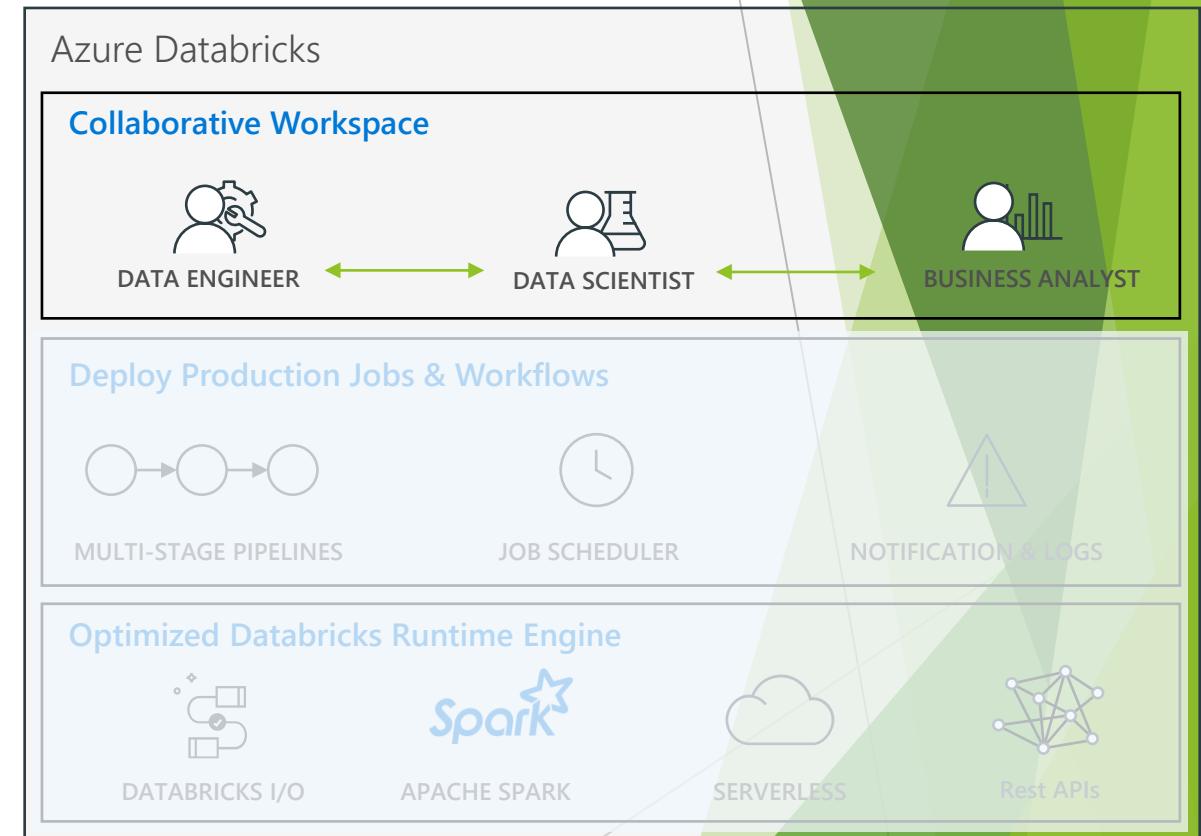
Notebook-ul poate fi modificat în timp real cu posibilitatea de a urmări modificările prin istoricul detaliat al variantelor în GitHub sau Bitbucket

## VISUALIZĂRI

Vizualizarea statisticilor printr-o gamă largă de instrumente de tip point-and-click sau prin utilizarea opțiunilor bazate pe scripturi precum matplotlib, ggplot și D3

## DASHBOARD

Integrare cu PowerBI pentru a analiza și disemina cunoștințele sintetizate în pattern-uri.



# Job-uri de producție & Workflow-uri

## PROGRAMATOR DE JOB-URI

Executa job-uri pentru pipeline-urile de producție pentru o programare data

## WORKFLOW-URI ASOCIAȚE NOTEBOOK-URILOR

Creaza pipeline-uri multi-stage cu controlul structurii sursei limbajului de programare

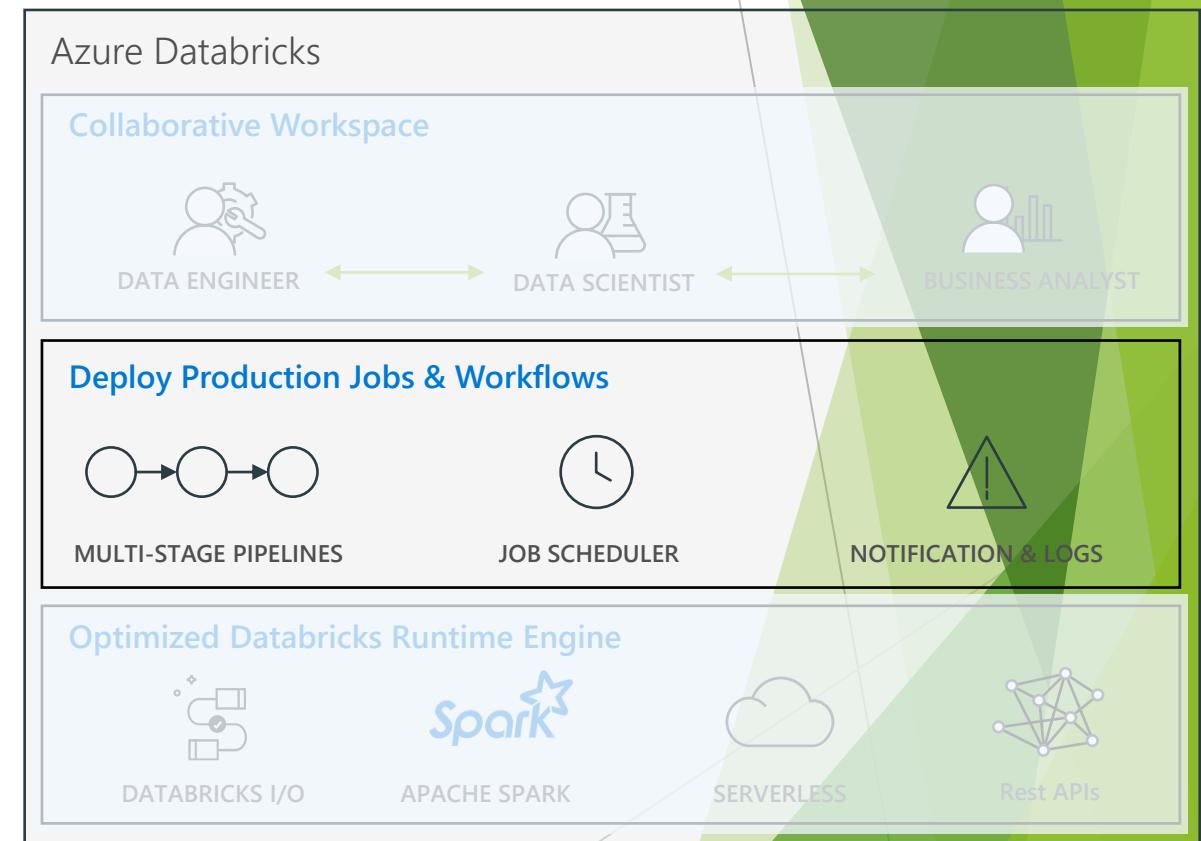
## RULEAZĂ NOTEBOOK-URI CA JOB-URI

Plasează notebook-urile sau JAR-ul in job-uri Spark reziliente de o maniera extrem de simpla

## NOTIFICATIARI SI LOG-URI

## SE INTEGREAZA NATIV CU AZURE SERVICES

Integrare cu: Azure SQL Data Warehouse, Cosmos DB, Azure Data Lake Store, Azure Blob Storage, si Azure Event Hub



# DRE(Databricks Runtime Engine) optimizat

## PERFORMANTE I/O OPTIMIZATE

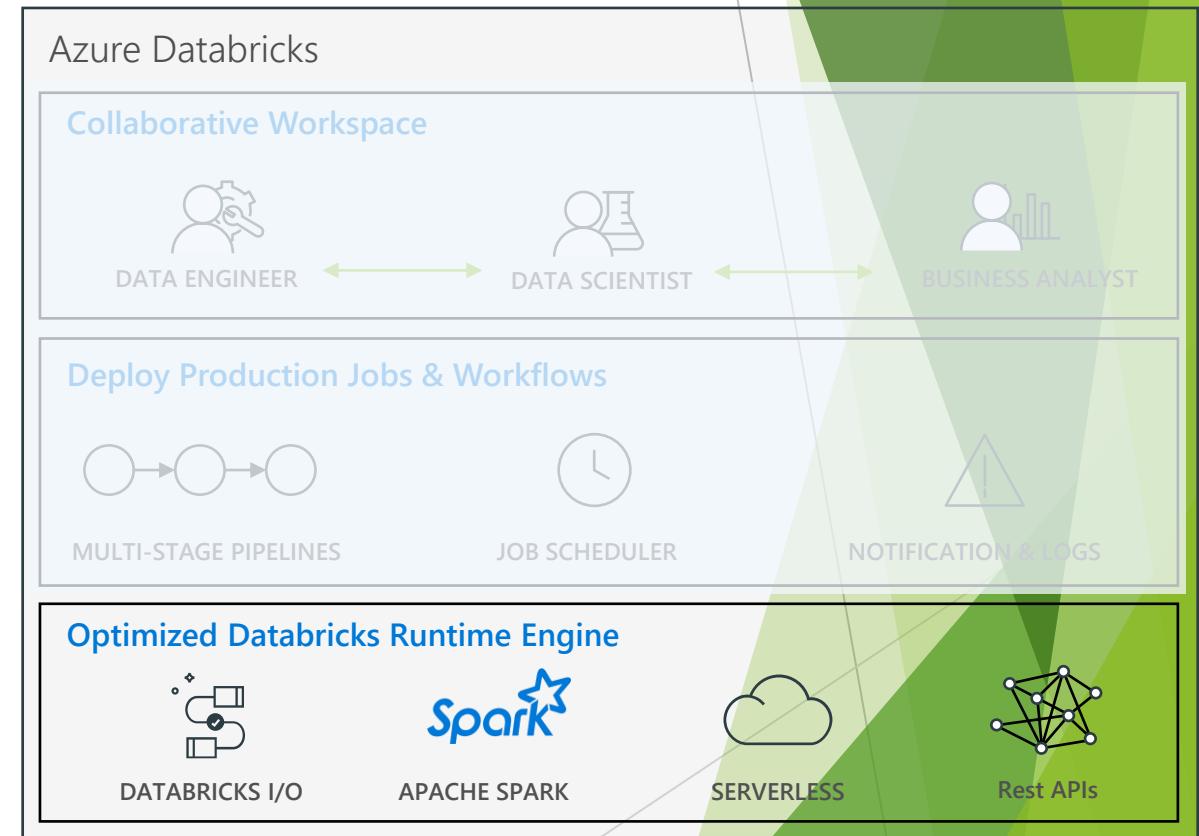
Databricks I/O (DBIO) optimizat trece viteza proceselor la un nivel superior odată cu portarea Spark in cloud

## PLATFORMĂ FULL-MANAGED ÎN AZURE

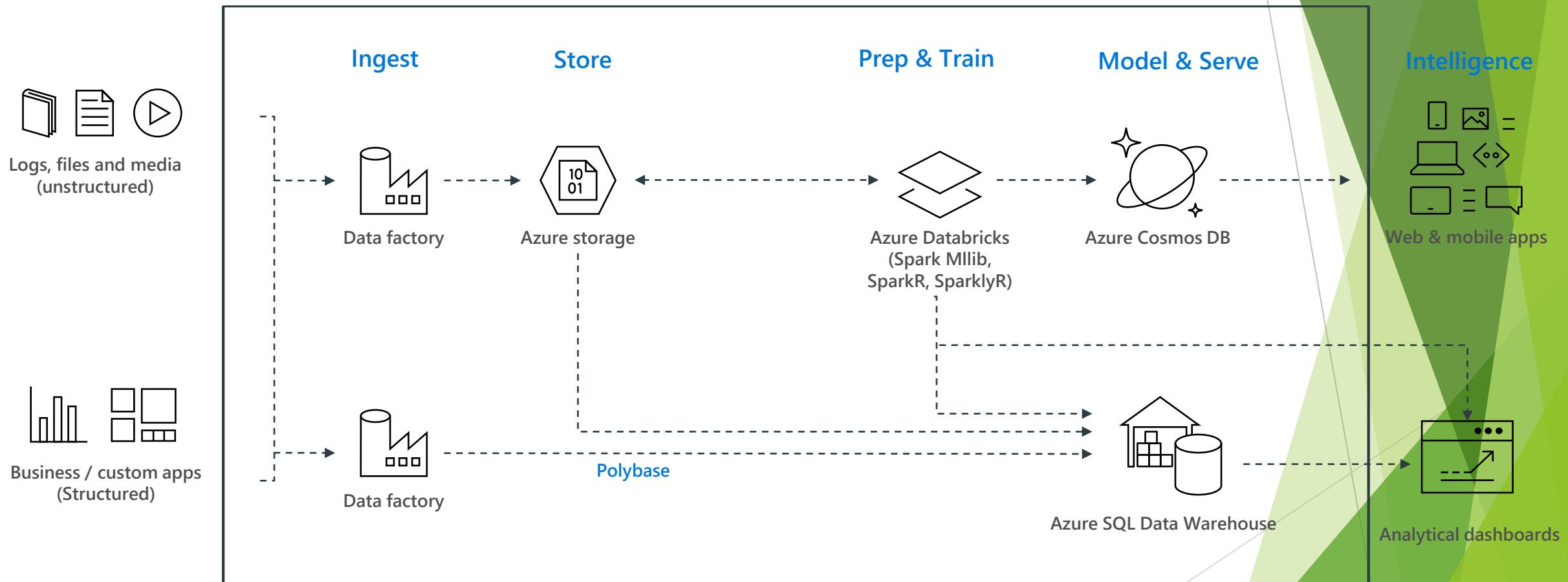
Serviciu complet gestionat ce produce reducerea complexității soluțiilor Big Data și ML într-un context SERVERLESS elastic proiectat a reduce complexitatea operatională.

## OPEREAZĂ CU SCALARE MASIVĂ

Fără limite globale



# Analitice pentru Big Data



# DATABRICKS ACCESS CONTROL

Access control can be defined at the user level via the Admin Console

## Databricks Access Control

Access Control can be defined for Workspaces, Clusters, Jobs and REST APIs

|                          |                                                                                                     |
|--------------------------|-----------------------------------------------------------------------------------------------------|
| Workspace Access Control | Defines who can view, edit, and run notebooks in their workspace                                    |
| Cluster Access Control   | Allows users to attach to, restart, and manage (resize/delete) clusters.                            |
| Jobs Access Control      | Allows Admins to specify which users have permissions to create clusters                            |
| REST API Tokens          | Allows owners of a job to control who can view job results or manage runs of a job (run now/cancel) |

DEMO