
Fly on Azure

Indice

1. Introduzione
    1. Microservices
    2. Servless computing
    3. Fly Language
2. Azure
    1. Architettura
        1. Azure AD
        2. Resource Manager
        3. Azure Storage
        4. Azure function
    2. Modalità di utilizzo
        1. API
        2. Portale
        3. CLI
        4. AZURE LIBRARIES
3. Artefatti
    1. Bash script
    2. Java
        1. Routine per il deploy
        2. Code di messaggi
        3. Trigger http
4. Azure vs AWS
5. Conclusioni



Introduzione
Microservices e
Servless computing

Tipi di servizi cloud

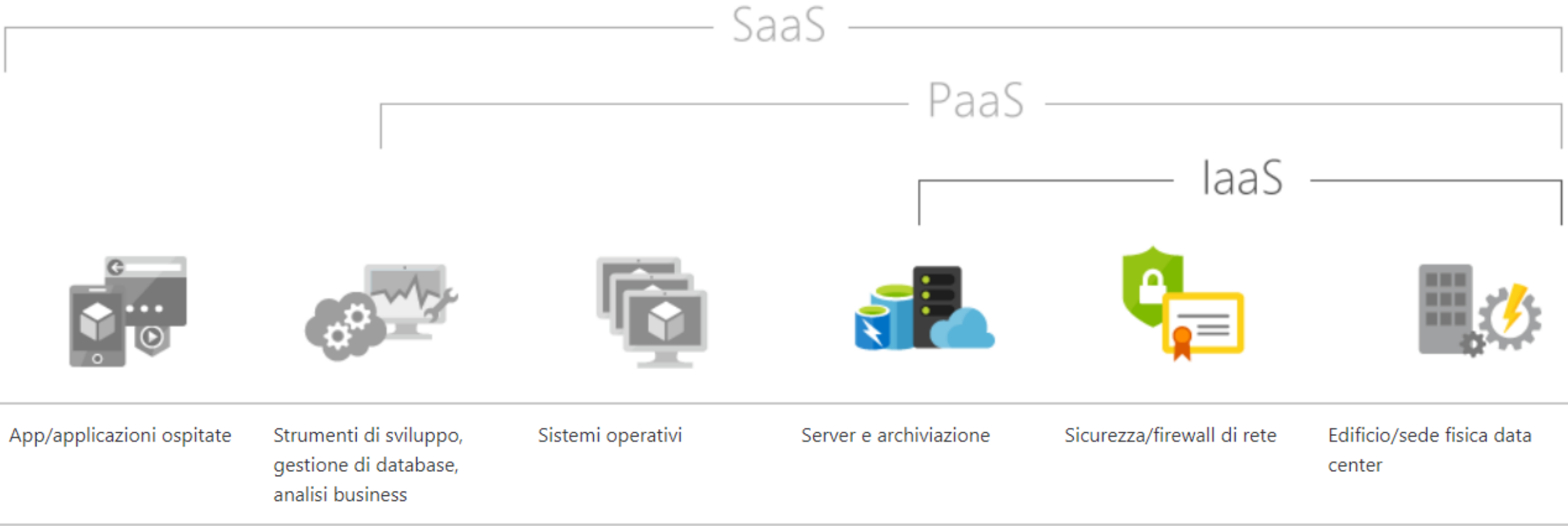
Modelli tradizionali

IaaS, PaaS e Saas



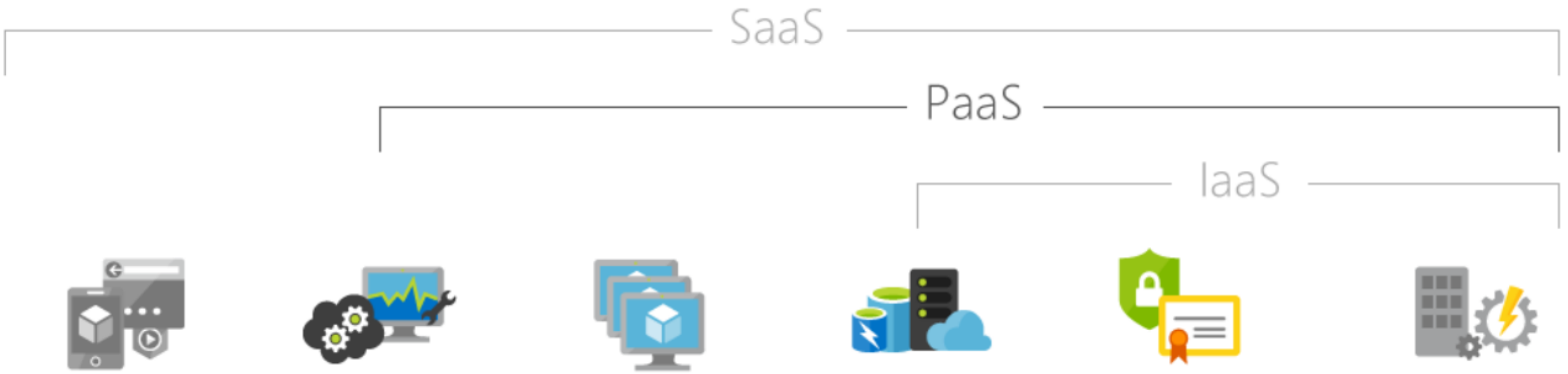
Tipi di servizi cloud

IaaS



Tipi di servizi cloud

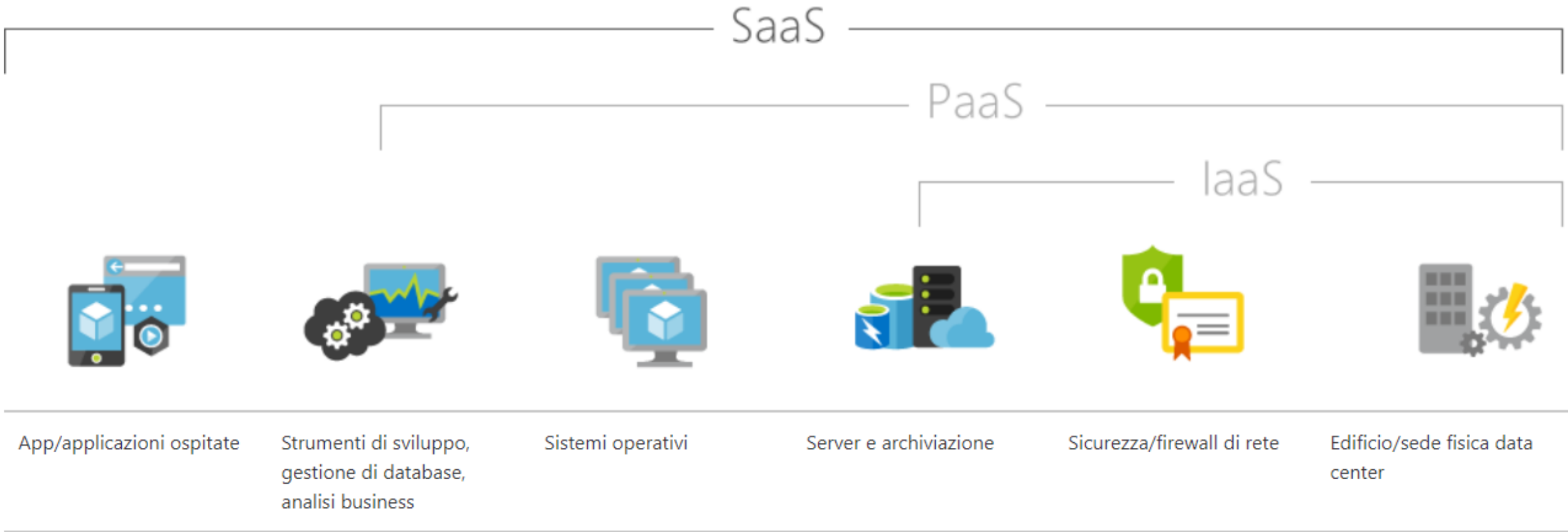
PaaS



App/applicazioni ospitate	Strumenti di sviluppo, gestione di database, analisi business	Sistemi operativi	Server e archiviazione	Sicurezza/firewall di rete	Edificio/sede fisica data center

Tipi di servizi cloud

SaaS



Microservices

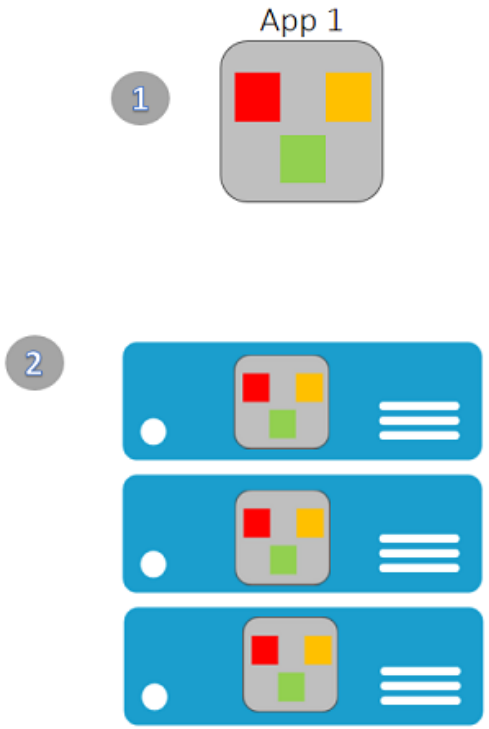
The term "Microservice Architecture" has sprung up over the last few years to describe a particular way of designing software applications as suites of independently deployable services. While there is no precise definition of this architectural style, there are certain common characteristics around organization around business capability, automated deployment, intelligence in the endpoints, and decentralized control of languages and data.

- Martin Fowler

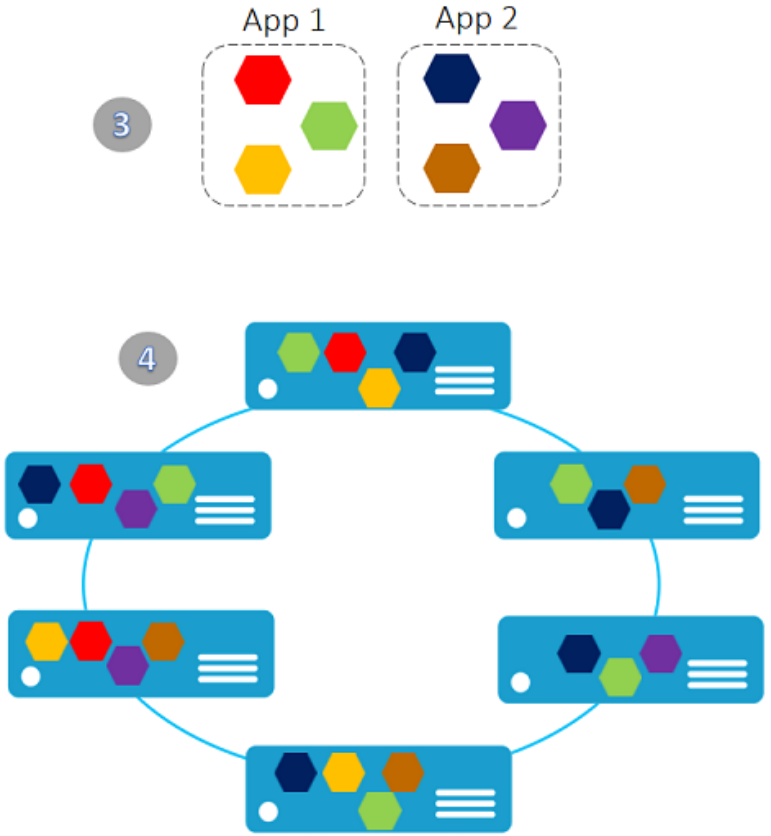


Decomposizione in servizi

Monolithic application approach

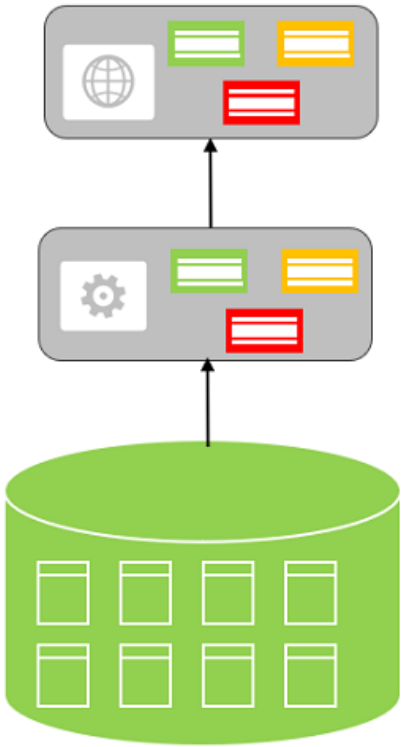


Microservices application approach

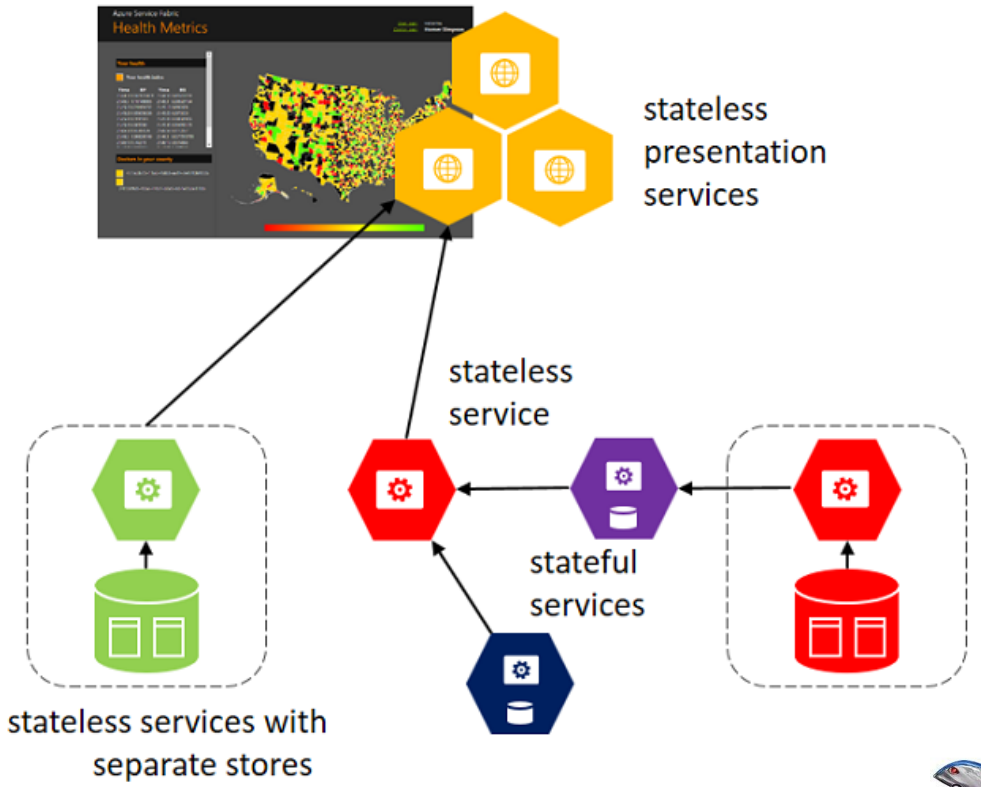


Gestione dei dati

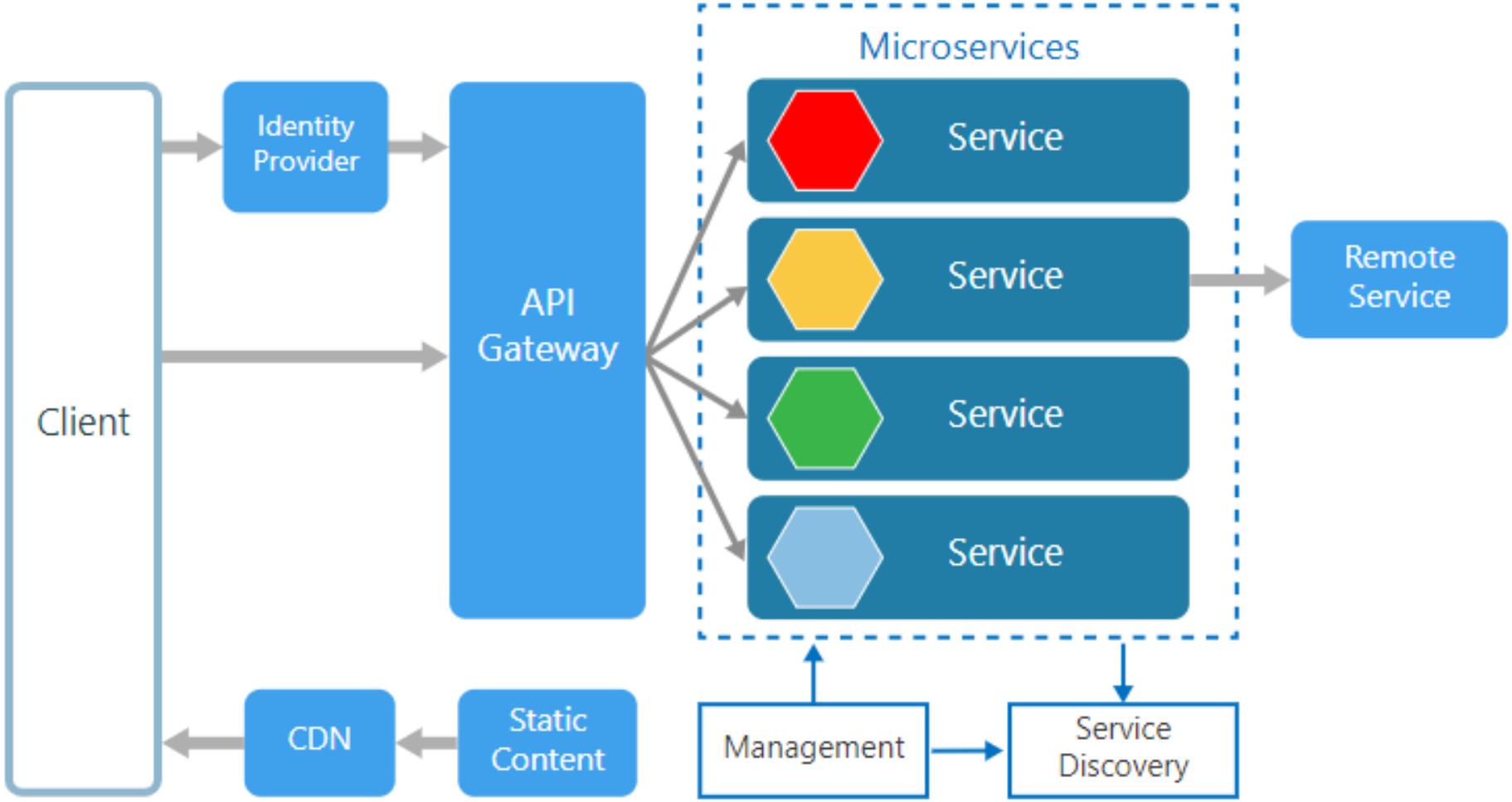
State in Monolithic approach



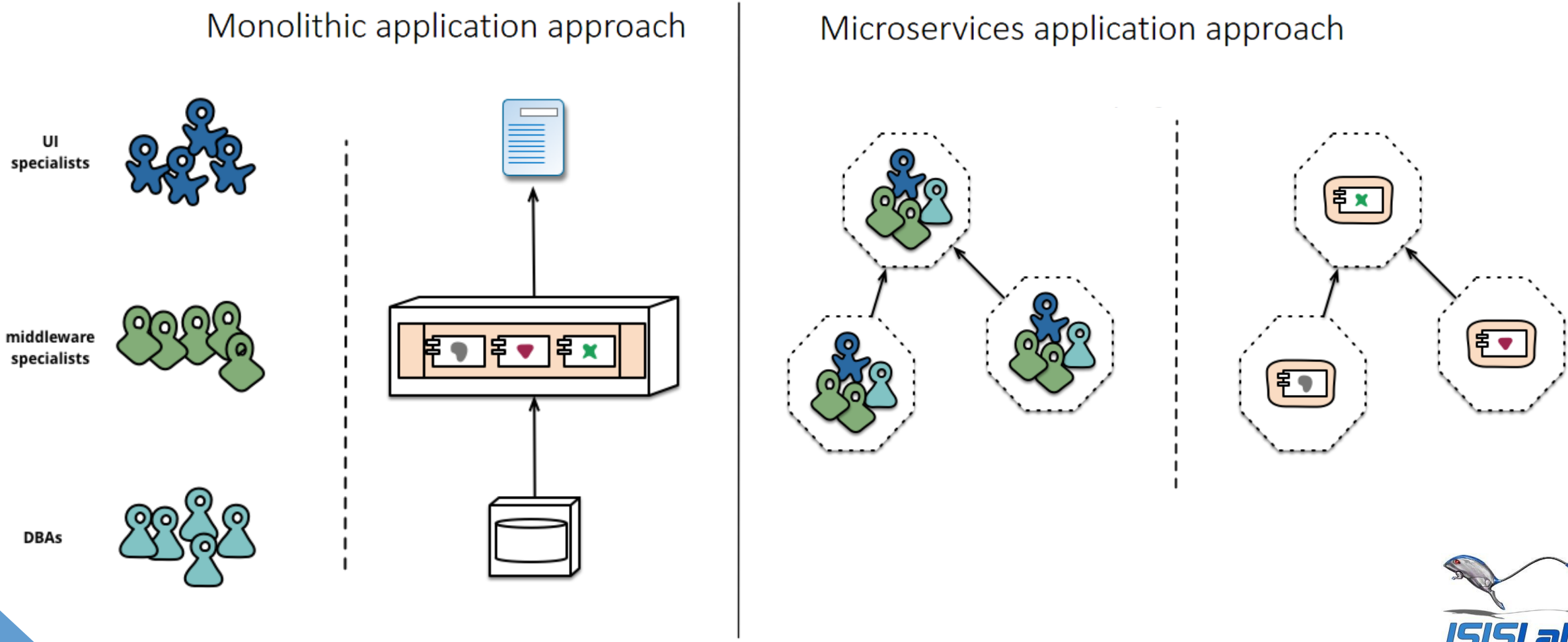
State in Microservices approach



Tipica architettura a microservizi



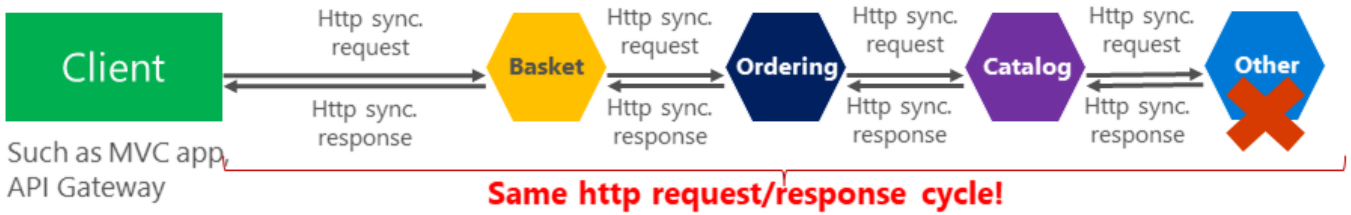
Team organizzati sulle funzionalità di business



Comunicazione tra microservizi

Anti-pattern

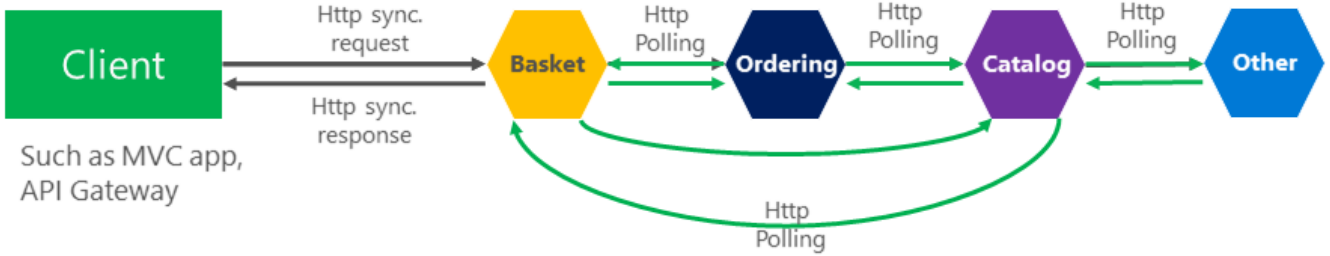
Synchronous
all request/response cycle



Asynchronous
Comm. across internal microservices
(EventBus: like **AMQP**)

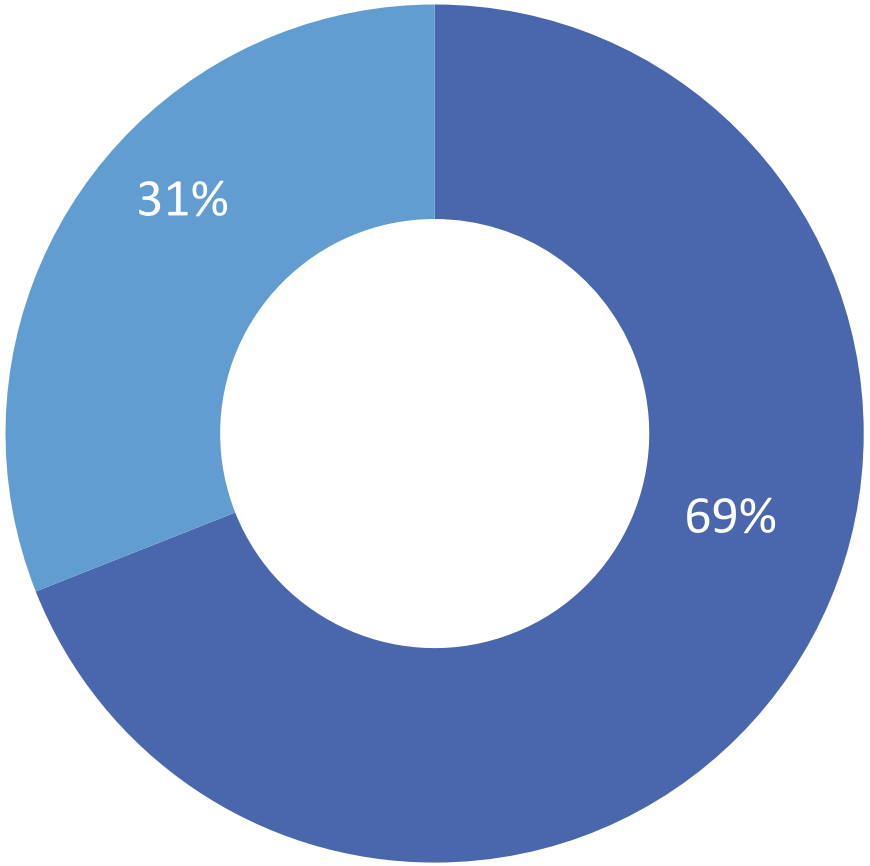


"Asynchronous"
Comm. across internal microservices
(Polling: **Http**)



ISISLab

Trend in crescita



Il 69% degli intervistati utilizza i microservizi per strutturare nuove applicazioni e ri-progettare applicazioni legacy

fonte

https://www.redhat.com/fr/blog/state-microservices

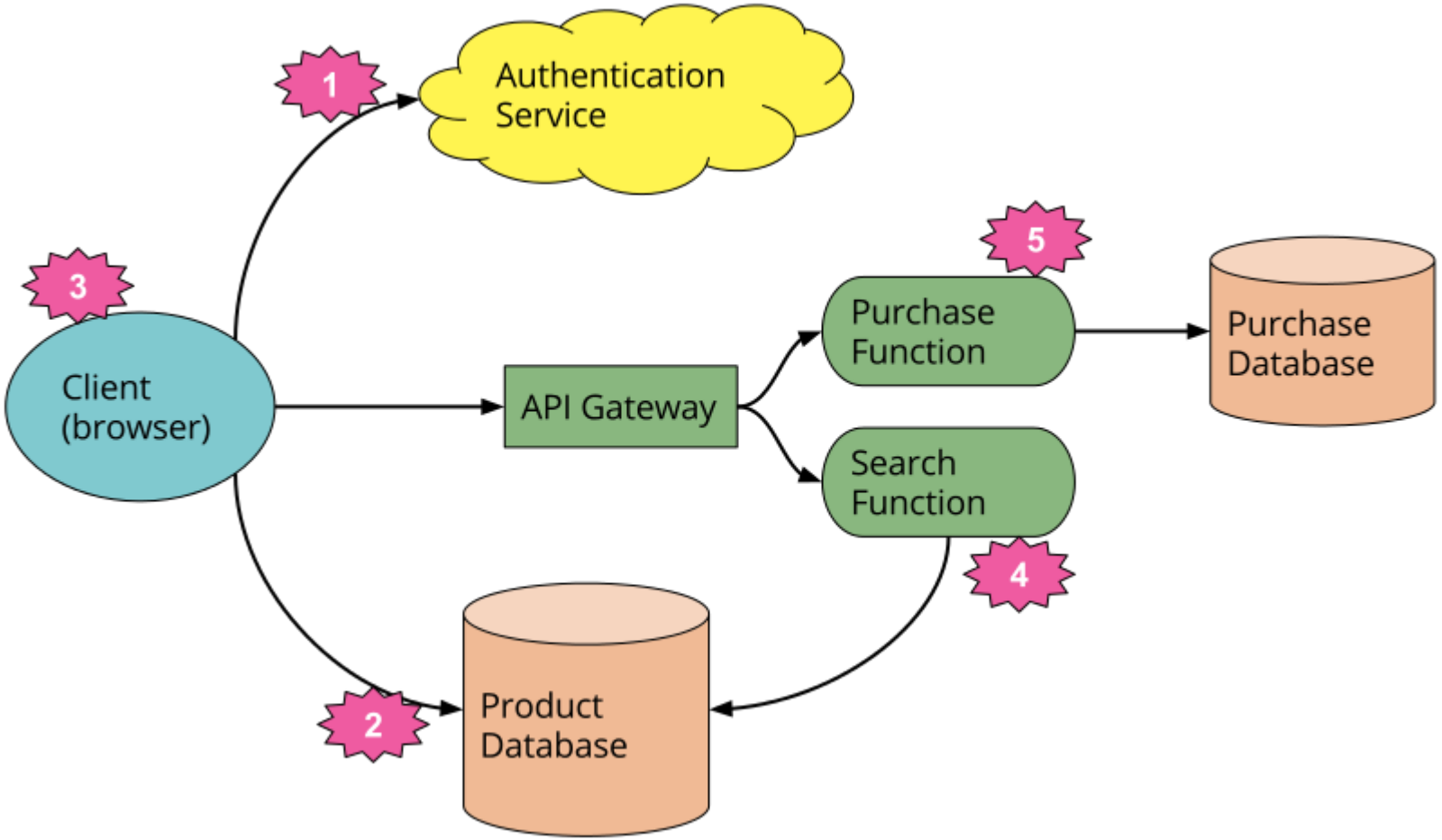
Serverless Model

Serverless architectures are application designs that incorporate third-party "Backend as a Service" (BaaS) services, and/or that include custom code run in managed, ephemeral containers on a "Functions as a Service" (FaaS) platform.

- Mike Roberts



Tipica architettura serverless



Vantaggi

Infrastruttura

Non si ha l'onere di gestire un server

Scalabilità

Offerta dal cloud provider

Risorse

Le risorse vengono allocate dinamicamente

Costi

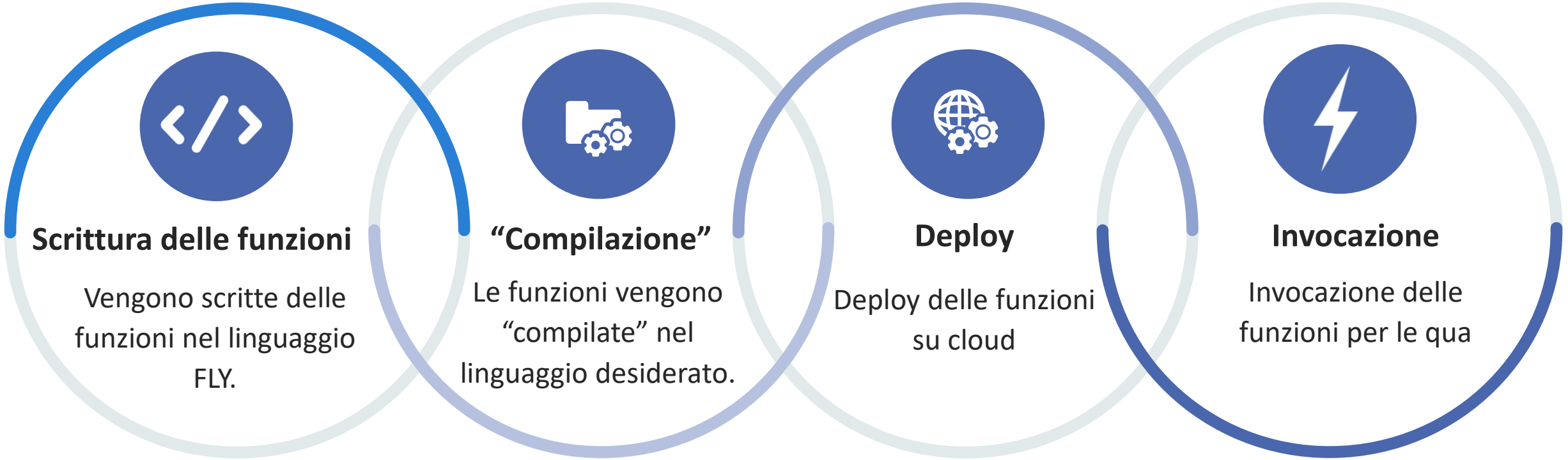
Tariffe sul tempo di esecuzione del codice

Il mio tirocinio

Integrare nel linguaggio FLY l’ambiente di Microsoft Azure come backend per l’esecuzione delle funzioni Fly

ISISLab

FLY Language



Scrittura delle funzioni

Vengono scritte delle funzioni nel linguaggio FLY.

“Compilazione”

Le funzioni vengono “compilate” nel linguaggio desiderato.

Deploy

Deploy delle funzioni su cloud

Invocazione

Invocazione delle funzioni per le qua

Eseguire le funzioni Fly su modello FaaS



Azure
Architetture e Servizi

















Cloud Platform

Indica una piattaforma che eroga un set di servizi cloud per creare gestire e distribuire applicazioni su internet.

Modelli disponibili

- IaaS
- PaaS
- SaaS
- FaaS

Azure AD

Active Directory

Azure AD è il servizio di Azure per la gestione dell'identità e dell'accesso controllato. Azure AD introduce diversi concetti fondamentali per la comprensione del funzionamento della piattaforma. Azure AD permette anche la gestione in gruppi semplificando l'accesso alle app e alle risorse sul cloud.



Concetti fondamentali

1. Tenant
2. Directory
3. Account
4. Ruoli

Autenticazione

1. Password
2. App Microsoft Authenticator
3. Token hardware OATH

Protocollo

Azure AD si basa sul protocollo Oauth 2.0 e lo standard OpenID Connect che definiscono i meccanismi per ottenere e usare token di accesso per accedere a risorse protette

ISISLab

Azure RM

Resource Manager

Questo servizio di Azure permette di gestire le risorse necessarie in maniera efficiente e organizzata: raggruppando le risorse in gruppi, questo permette di definire delle caratteristiche comuni a tutto il gruppo, oltre a semplificare la fatturazione e la gestione delle risorse.



Definizione

Raccoglie la definizione dell'infrastruttura e delle dipendenza tra le risorse utilizzate.

Organizzazione

Organizza la gestione delle risorse, facilitandone la vision d'insieme.

Accesso Controllato

Garantisce l'accesso controllato alle risorse sfruttando le funzionalità di Azure AD.

ISISLab

Azure Storage

Azure Storage è il servizio offerto da Azure per l'archiviazione cloud.

Azure Storage è organizzato mediante account di archiviazione, tramite i quali è possibile creare tutti gli oggetti di archiviazione.

L'account storage rappresenta uno spazio di nomi univoco.



Blobs

Permette di archiviare dati non strutturati e di accedere in larga scala tramite un'interfaccia REST.

Queues

Le code consentono un recapito dei messaggi di tipo FIFO a uno o più consumer concorrenti.

Files

Permette di archiviare file tramite un' interfaccia REST o SMB.

ISISLab

Azure Function

Azure Function è il servizio fornito da Azure per il servless computing, questo servizio permette di eseguire blocchi di codice, funzioni, quando un determinato «trigger» viene innescato.

Per creare una function in Azure si ha la necessità di creare un «App per funzioni» a cui associarle, questo perché l'app per funzione gestisce la condivisione delle risorse, oltre a raggruppare le funzioni come un'unità logica.



Piani di consumo

1. Consumption plan
2. App service plan

Trigger

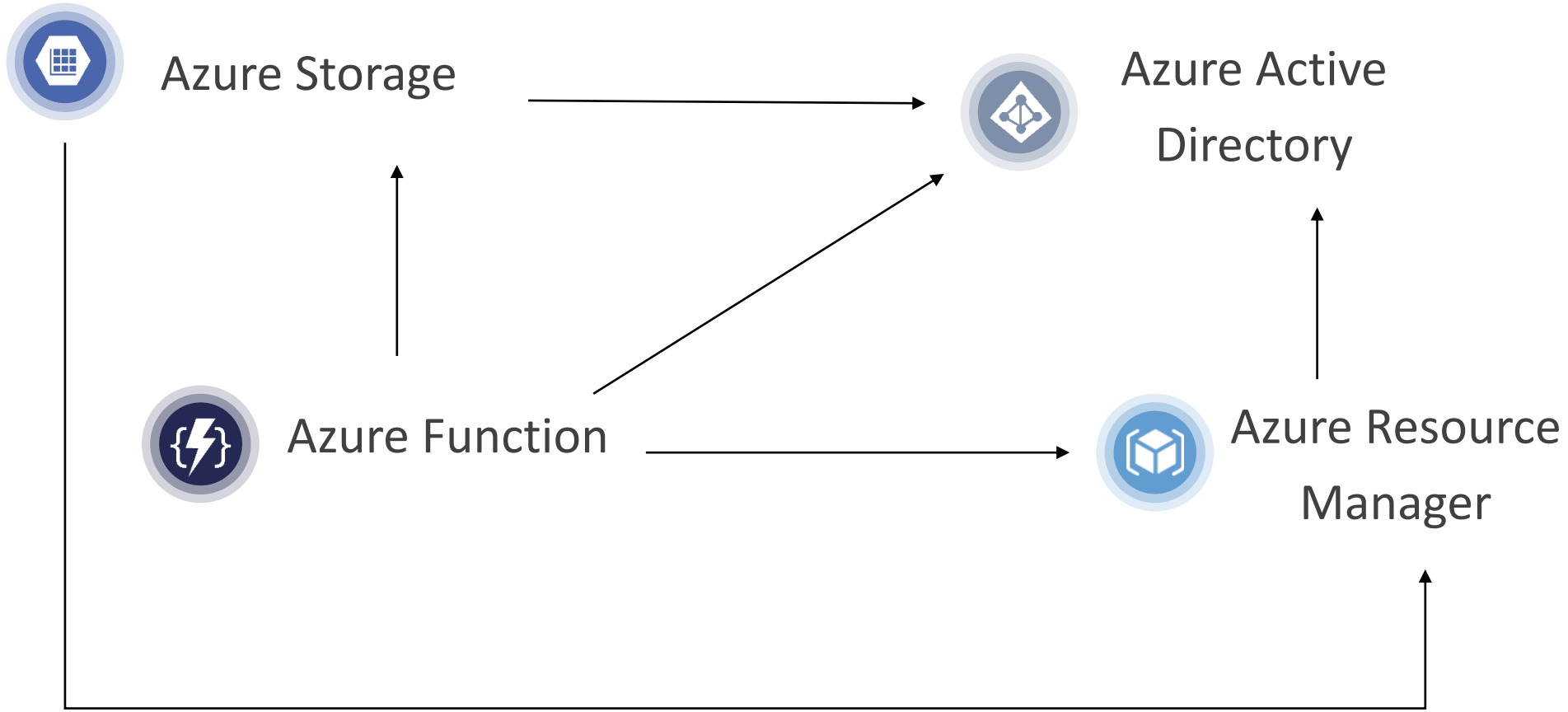
1. HTTP
2. Timer
3. Cosmos DB
4. Blob
5. Queue

Durable function

Estensione delle Azure Function che permette la scrittura di funzioni stateful in ambienti servless.

ISISLab

Dipendenze tra i servizi utilizzati



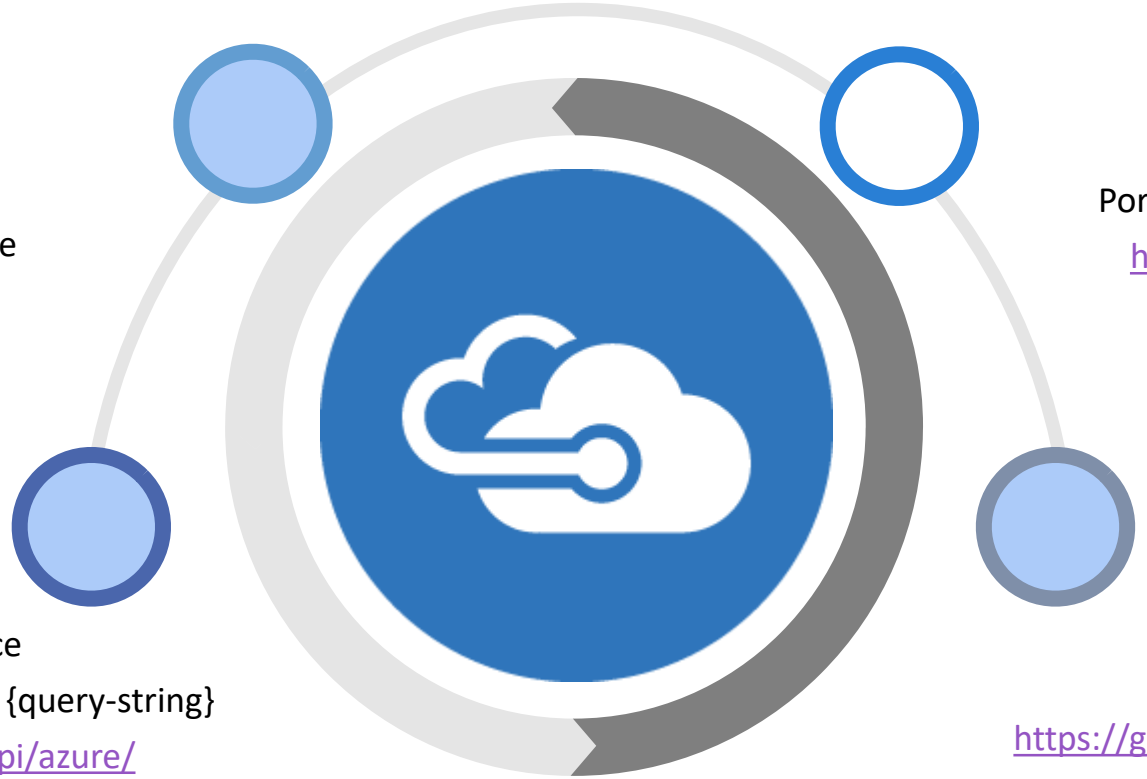
Modalità di utilizzo

CLI

Interfaccia a linea di commando
Es. az storage create –name account_name
https://docs.microsoft.com/en-us/cli/

API

Application Programming Interface
URI-scheme} :// {URI-host} / {resource-path} ? {query-string}
https://docs.microsoft.com/en-us/rest/api/azure/



Portale

Portale Web / App per mobile
https://portal.azure.com/

Libraries

Librerie di codice
Es. azure-libraries-for-java
https://github.com/Azure/azure-libraries-for-java

ISISLab

Artefatti
Utilizzo di Azure

Primo step

Creare un insieme di artefatti per automatizzare i processi necessari al deploy e l’esecuzione di funzioni su azure.

ISISLab

Implementazione in Java



TRIGGER HTTP

Interfacce per invocare una funzione tramite richiesta http.

CODE DI MESSAGGI

Interfacce per utilizzare una coda di Azure per comunicare con una funzione.

ROUTINE PER IL DEPLOY

1. Creazione di un progetto remote su Azure
    1. Creazione resource group
    2. Creazione account storage
    3. Creazione piano di consume
    4. Creazione dell'app per funzioni
2. Invocazione dello shell script per il deploy

ISISLab

Bash script



Creazione del progetto locale

In input il codice della funzione
e le librerie necessarie

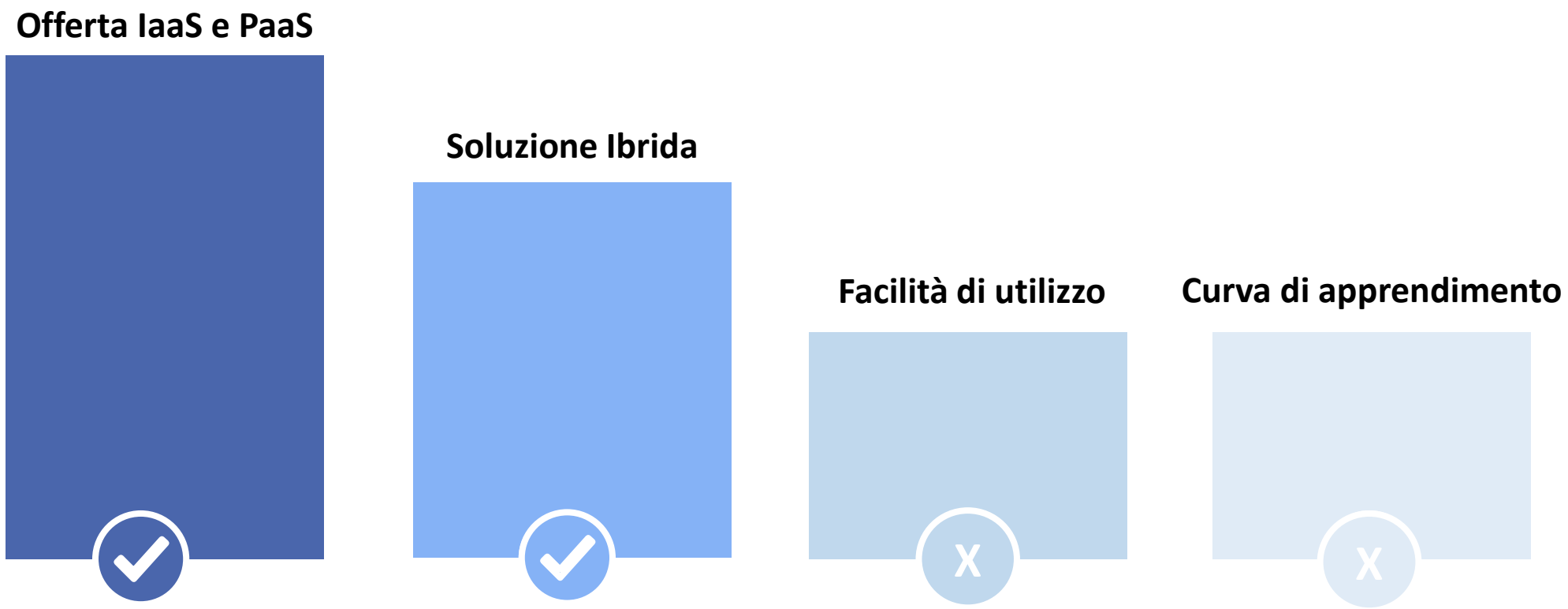
Deploy della funzione

Deploy del progetto locale sull'app
per funzioni indicate in input

Azure vs AWS

Pro e contro

Azure vs AWS

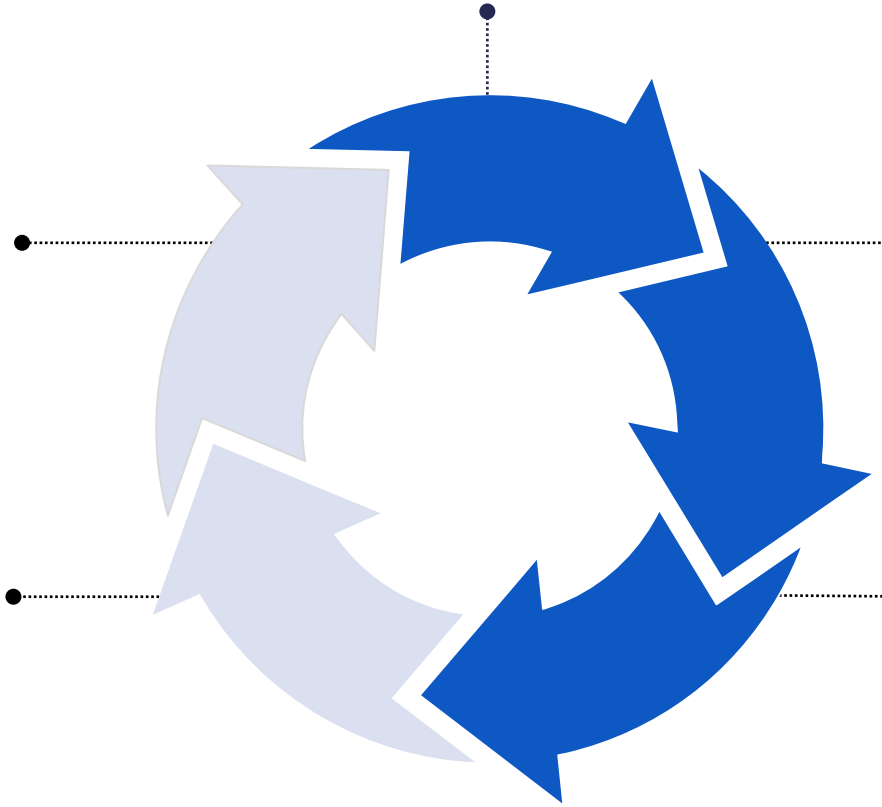

Offerta IaaS e PaaS
Soluzione Ibrida
Facilità di utilizzo
Curva di apprendimento
X
X

Conclusioni
Stato attuale e prossimi steps

Conclusioni

Sintassi del linguaggio FLY, concetti fondamentali (microservices, servless computing etc)

Studio iniziale



Studio della piattaforma Azure

Primi artefatti

Shell script e software java per automatizzare il deploy di una funzione su azure

Integrare il deploy su azure in Fly

Valutare l’esecuzione di funzioni FLY mediante Azure

ISISLab

Thanks