

Padova, 8/11/2016

La rivoluzione dei microservizi

Il nuovo paradigma di progettazione architetturale che cambierà i vostri sistemi IT rendendoli più flessibili e più dinamici



italianaSoftware

gruppoimola



La sfida



Velocità

sistema informativo
aziendale



processi



dati

La sfida consiste nell'avere un sistema informativo in grado di **cambiare velocemente** dove sia facile inserire nuovi processi e nuove sorgenti dati in **tempi e costi ridotti**

Com'è il sistema ora



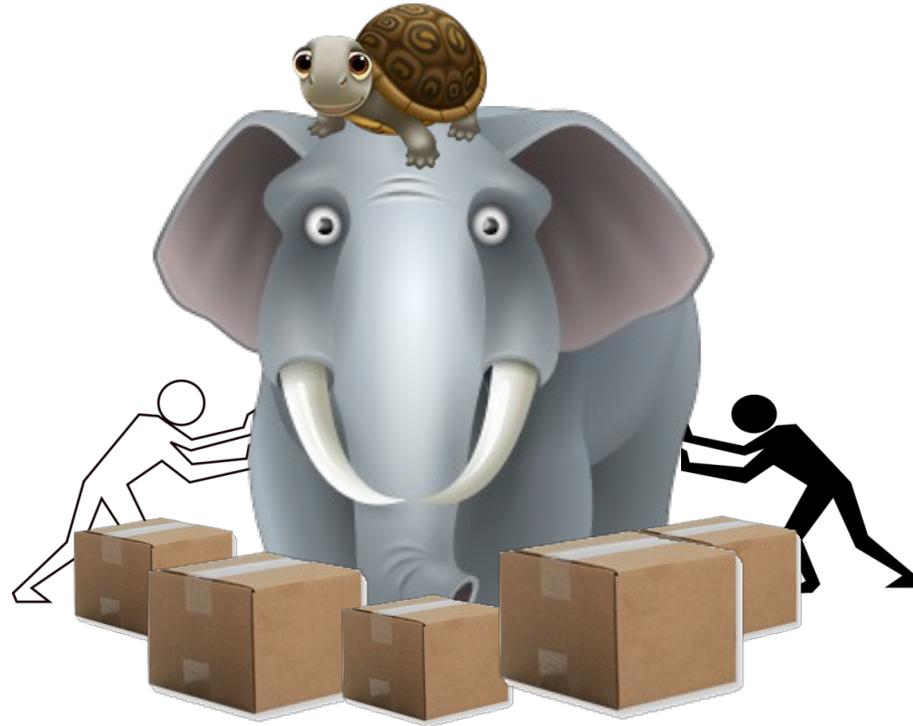
sistema informativo
aziendale



applicativi esterni

La maggior parte dei sistemi attualmente in uso è di tipo **monolitico** con processi e dati difficilmente modificabili se non con costi elevati. Tali sistemi sono spesso integrati con altri applicativi di fornitori esterni sui quali si ha scarso accesso

Cosa vuol dire che un sistema è monolitico



Significa che il sistema è formato da un blocco unico di software o da un insieme ristretto di blocchi unici (applicativi esterni) che presi singolarmente offrono tante funzionalità ma che sono **difficilmente collegabili insieme**. La natura del sistema **rallenta i processi aziendali**.

Come dovrebbe essere?



Un sistema altamente adattivo e flessibile dovrebbe essere composto da **tanti componenti interdipendenti** ciascuno dei quali offre funzionalità ben delineate

L'architettura è il cuore



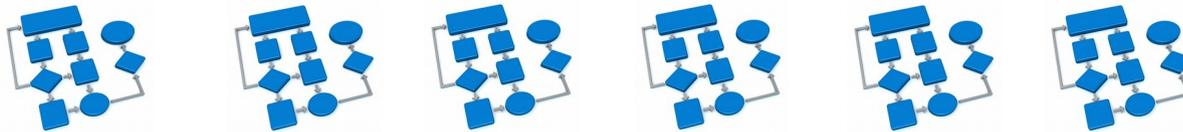
E' l'architettura ad essere al centro di tutto perché permette di rendere interoperabili i componenti. E' attraverso l'architettura che il sistema è in grado di offrire tutte le funzionalità.

Architettura a strati

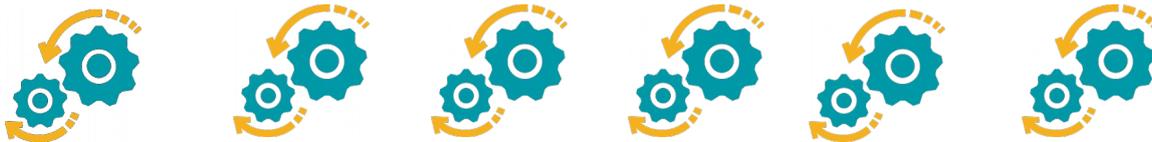


Applicativi

API



Processi di business



Processi base



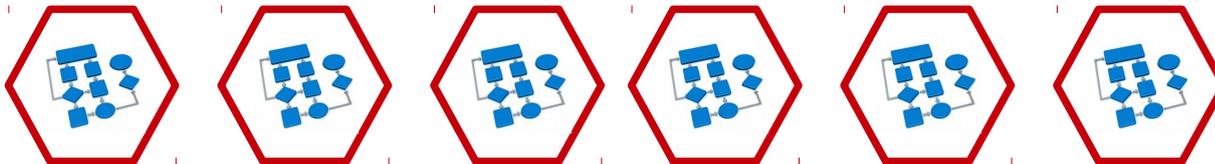
Sorgenti di dati

I microservizi come soluzione



Applicativi

API



MICROSERVIZI

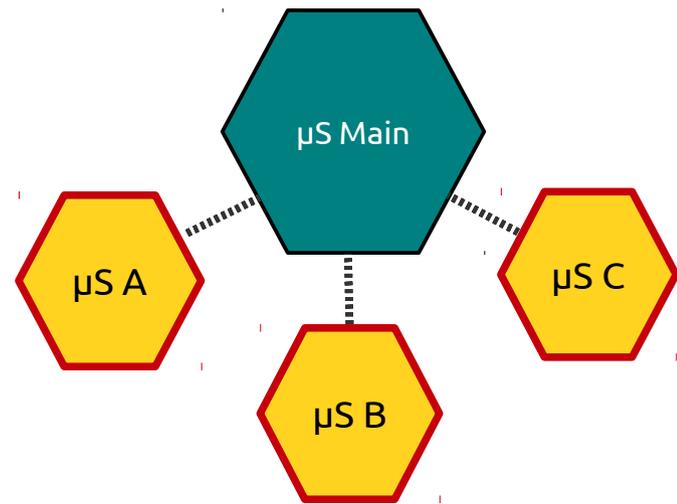
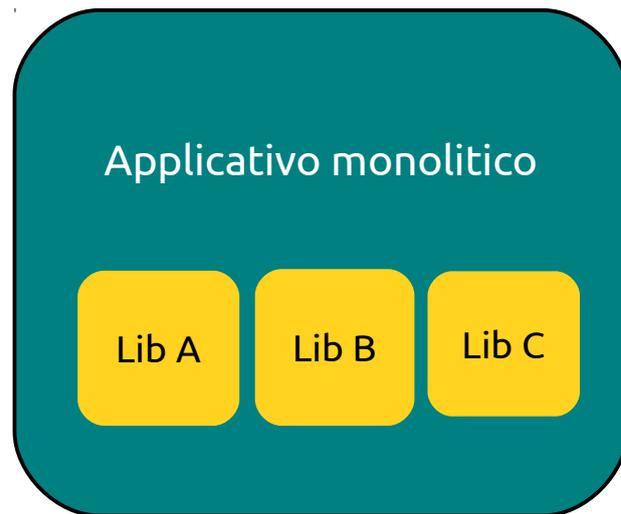


MICROSERVIZI

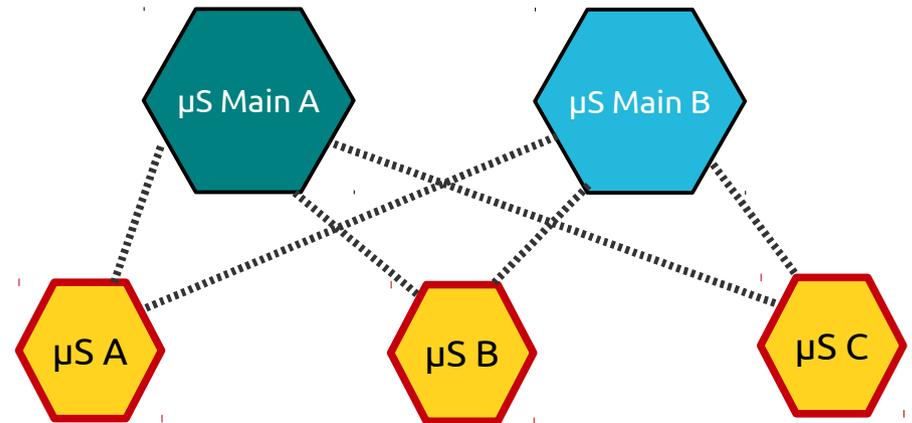
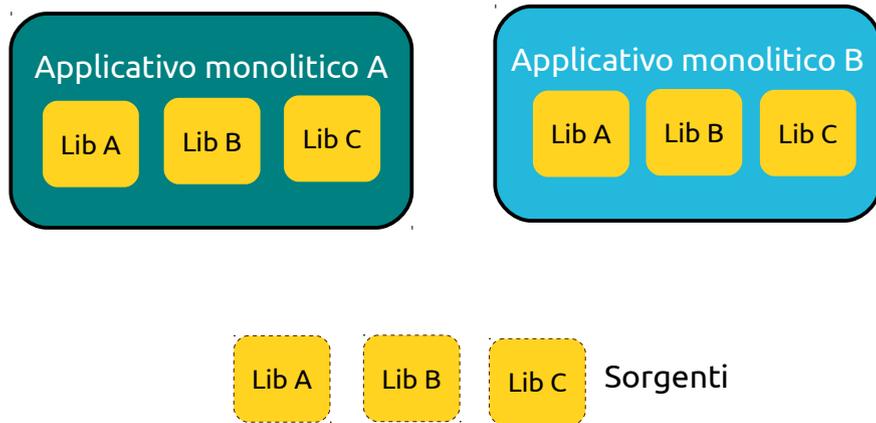


Sorgenti di dati

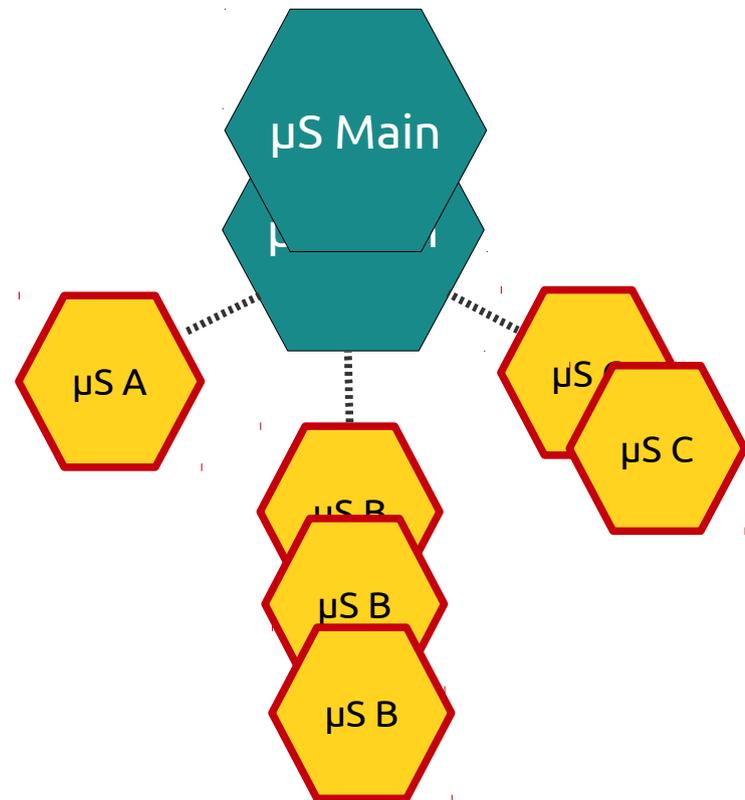
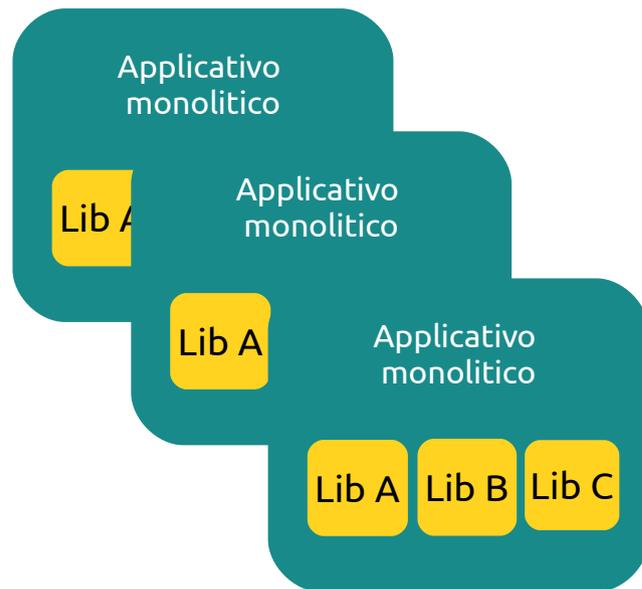
Microservices - Componibilità



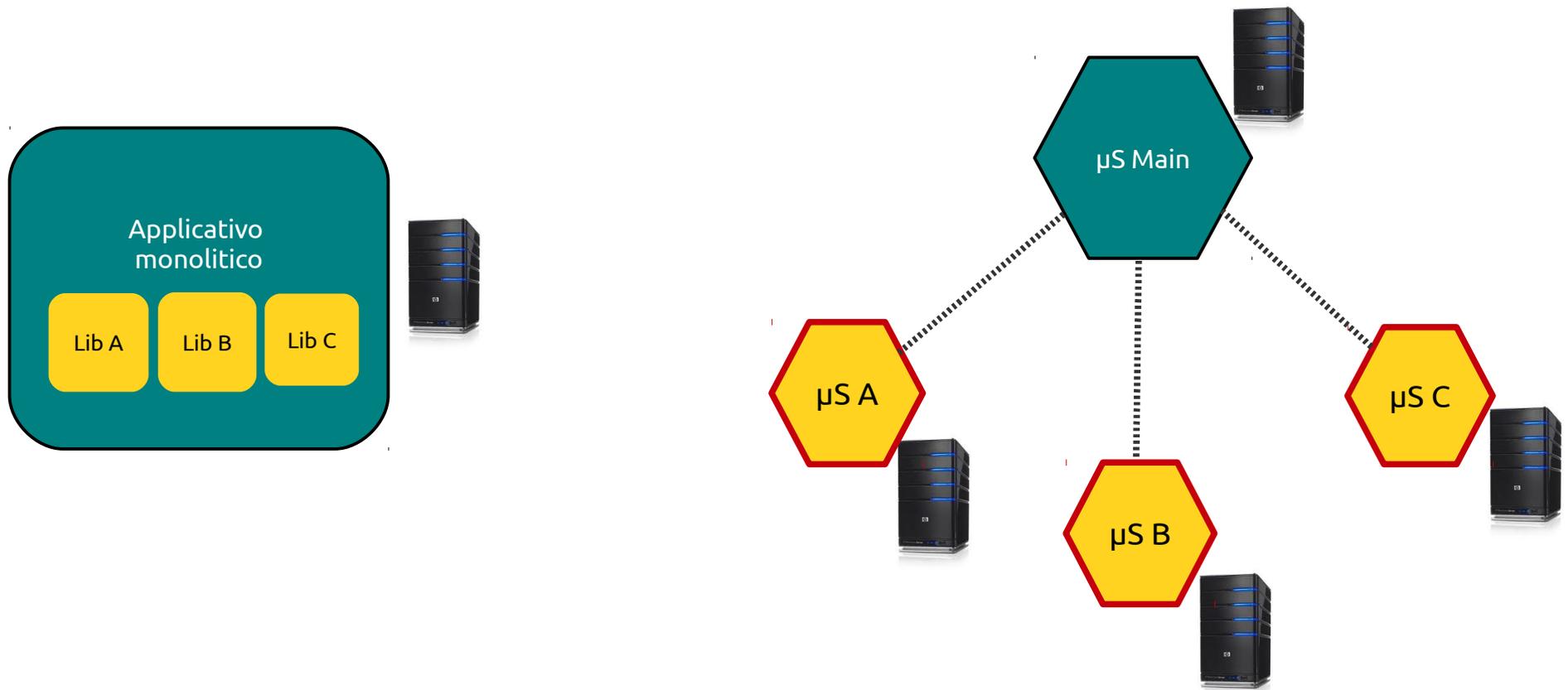
Microservices - Riusabilità



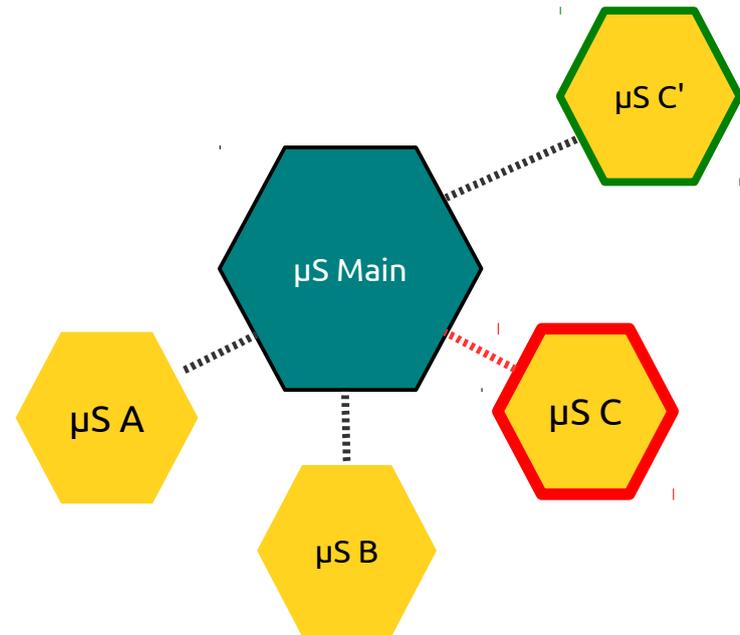
Microservices - Scalabilità



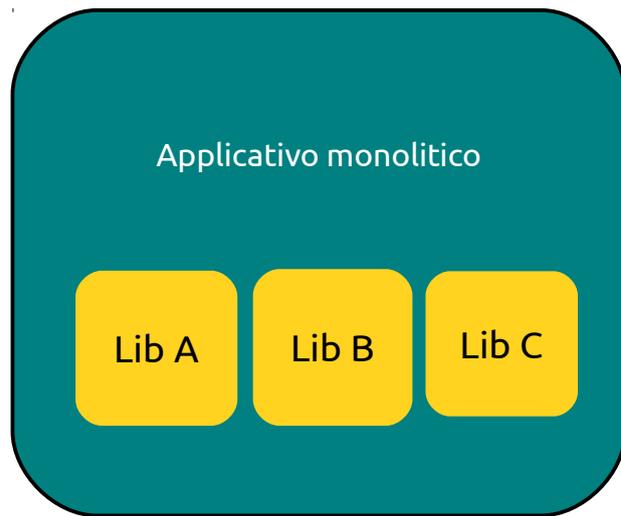
Microservices - Deployment



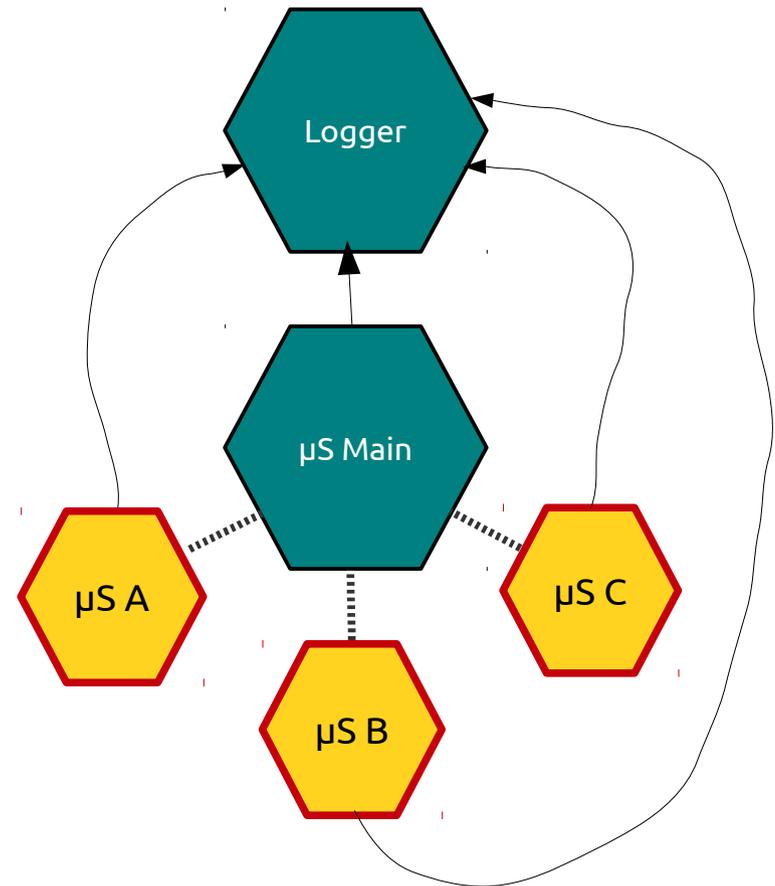
Microservices - Resilienza



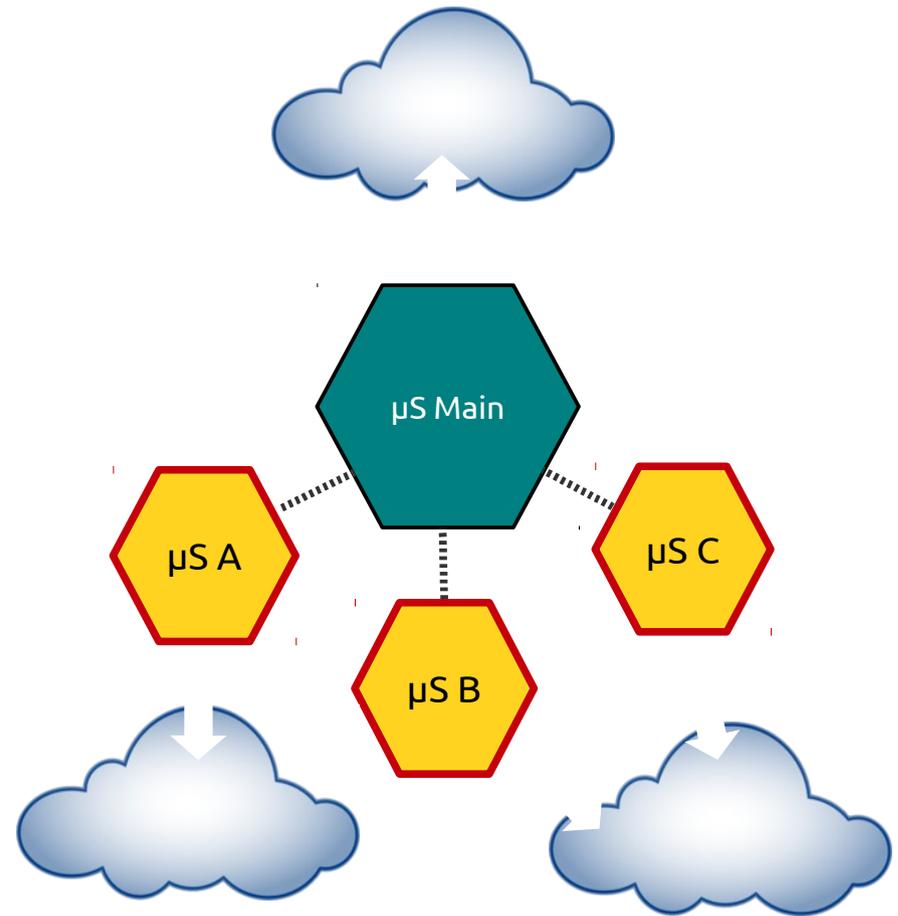
Microservices - Monitoring



LOG



Microservices - Cloud



Chi Siamo

- Startup nata nel 2008 2° classificata alla competizione Innovami Start Cup di Imola
- Ideatori di Jolie, linguaggio di programmazione orientato alla creazione e all'orchestrazione di MicroServizi
- dal 2016 parte di Gruppo Imola

gruppo**imola**



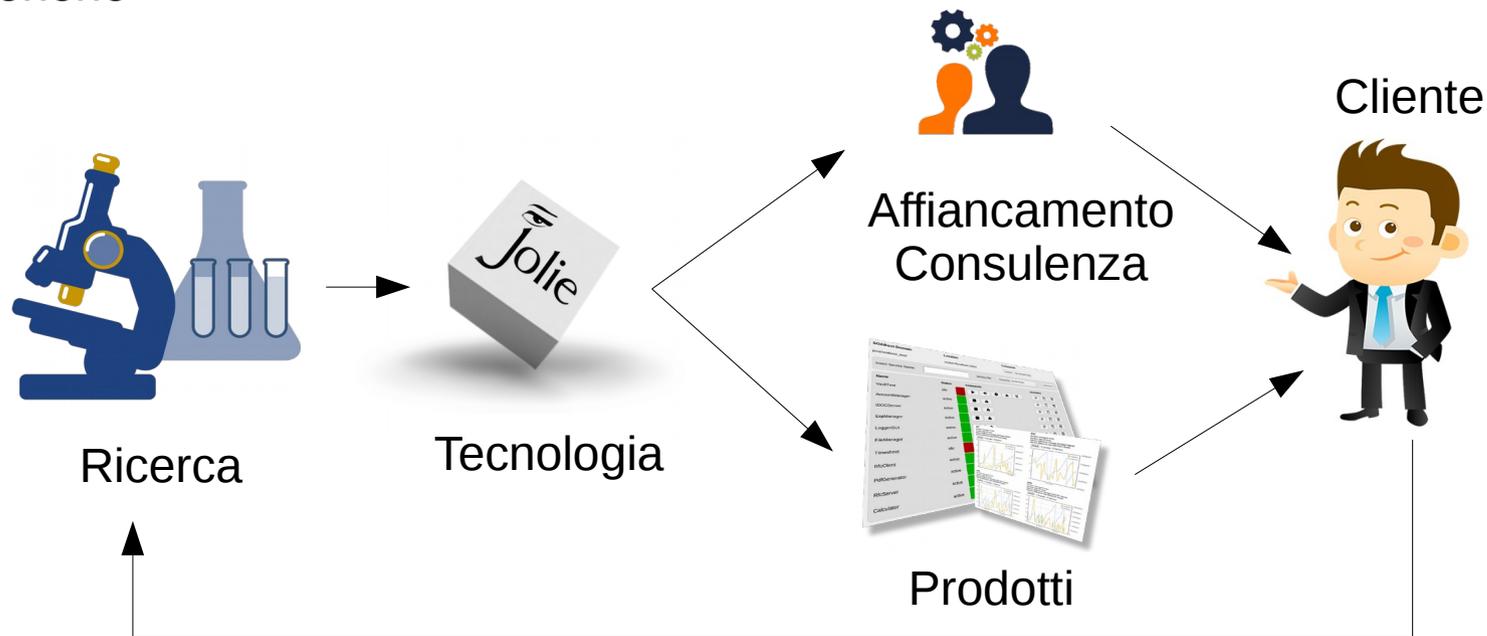
italianaSoftware

gruppoimola



MISSION

- Fornire la tecnologia ed il supporto per la costruzione di Architetture a MicroServizi, fondamentali nella programmazione di servizi web complessi
- Progettare Architetture estremamente flessibili e implementabili anche su realtà piccole e medie, che trovano un limite invalicabile nelle attuali tecnologie e nei costi necessari a sostenerle



Cosa offriamo



- **Introduzione ed assistenza a Jolie:**
 - Formazione
 - Consulenza
- **Strumenti facilitatori per la realizzazione di sistemi microservices basati su Jolie:**
 - Jolie Enterprise
 - JSAP – suite di microservices per l'integrazione con SAP
- **Sviluppo:**
 - Sviluppo di microservizi ed interfacce web

Jolie



Java Orchestration Language Interpreter Engine, nasce come alternativa al linguaggi di orchestrazione per web service WS-BPEL

- Semplificandone la gestione
 - Non richiede la presenza di motori di esecuzione dedicati
 - La definizione non è definita mediante xml ma è scritta in linguaggio nativo eseguito dall'interprete
- Estendendone le funzionalità
 - Possibilità di integrare logica applicativa Java
 - Possibilità di utilizzare il linguaggio per implementare microservizi
 - Orchestrazione di Web Service Soap, Rest, Jolie

Perché un linguaggio?

ANNO 2016

Ho comprato uno smartphone nuovo con
sistema operativo Android e connessione
ad internet

Perché un linguaggio?

ANNO 1800?

Ho comprato uno **smartphone** nuovo con
sistema operativo Android e **connessione**
ad **internet**

Perché un linguaggio?

ANNO 1800?

Ho acquistato un oggetto grande quanto un libro di piccole dimensioni la cui copertina straordinariamente lucida e variopinta è in grado di mostrare immagini così ben definite da apparire quasi rappresentazioni della realtà, ma questo è nulla poiché è sufficiente sfiorare tale oggetto con un dito per innescare un meccanismo quasi diabolico in grado di fare udire la mia voce ad una persona all'altro capo del mondo in tempo reale così che io possa comunicare con lei una volta acquisito un numero che mi permette di riconoscerla tra tutti. L'oggetto è nuovo con **sistema operativo** Android e **connessione ad internet**

Il linguaggio fa guadagnare TEMPO e SPAZIO

oggetto grande quanto un libro di piccole dimensioni la cui copertina straordinariamente lucida e variopinta è in grado di mostrare immagini così ben definite da apparire quasi rappresentazioni della realtà, ma questo è nulla poiché è sufficiente sfiorare tale oggetto con un dito per innescare un meccanismo quasi diabolico in grado di fare udire la mia voce ad una persona all'altro capo del mondo in tempo reale così che io possa comunicare con lei una volta acquisito un numero che mi permette di riconoscerla tra tutti.

smartphone

Il linguaggi di programmazione

MENO ERRORI UMANI



MACCHINA

```
01010100010100010010
01001001010001010000
10100100100101001010
10100101001001010010
01010100010100010010
01001001010001010000
10100100100101001010
10100101001001010010
01010100010100010010
01001001010001010000
10100100100101001010
10100101001001010010
01010100010100010010
01001001010001010000
10100100100101001010
10100101001001010010
01010100010100010010
01001001010001010000
10100100100101001010
10100101001001010010
01010100010100010010
01001001010001010000
```

ASSEMBLER

```
ORG 2050H
MOV DPTR,#CNT
MOV A,#INIT
MOVX @DPTR,A
MOV SP,#2FH
CLR RS0
CLR RS1
ACALL SERIAL
MOV
DPTR,#HEADING
ACALL STRING
ORG 2050H
MOV DPTR,#CNT
MOV A,#INIT
MOVX @DPTR,A
MOV SP,#2FH
CLR RS0
```

C

```
int main()
{
    int n, sum = 0, c, value;

    printf("Enter the number of integers ");
    scanf("%d", &n);

    printf("Enter %d integers\n",n);

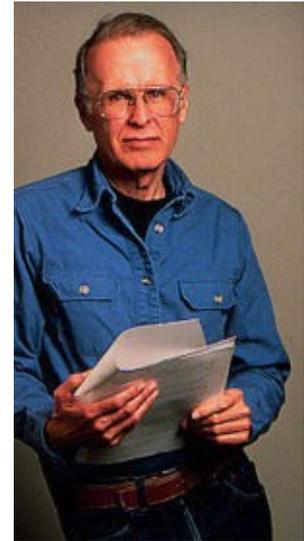
    for (c = 1; c <= n; c++)
    {
        scanf("%d",&value);
        sum = sum + value;
    }

    printf("Sum of entered integers = %d\n",sum);

    return 0;
}
```

Un vantaggio concreto

- Il primo compilatore completo fu creato dall'equipe di John. W. Backus presso i laboratori IBM nel 1957. Era un compilatore FORTRAN. Ci vollero ben
- **18 anni uomo** per realizzarlo.
- Al tempo non era stata ancora sviluppata la **teoria dei linguaggi**.
- Oggi in molte università, viene data la creazione di un compilatore come esercizio d'esame.



John. W. Backus

Il linguaggio fa guadagnare TEMPO e SPAZIO

oggetto grande quanto un libro di piccole dimensioni la cui copertina straordinariamente lucida e variopinta è in grado di mostrare immagini così ben definite da apparire quasi rappresentazioni della realtà, ma questo è nulla poiché è sufficiente sfiorare tale oggetto con un dito per innescare un meccanismo quasi dialettico in grado di fare udire la mia voce ad una persona all'altro capo del mondo in tempo reale così che io possa comunicare con lei una volta acquisito un numero che mi permette di riconoscerla tra tutti.

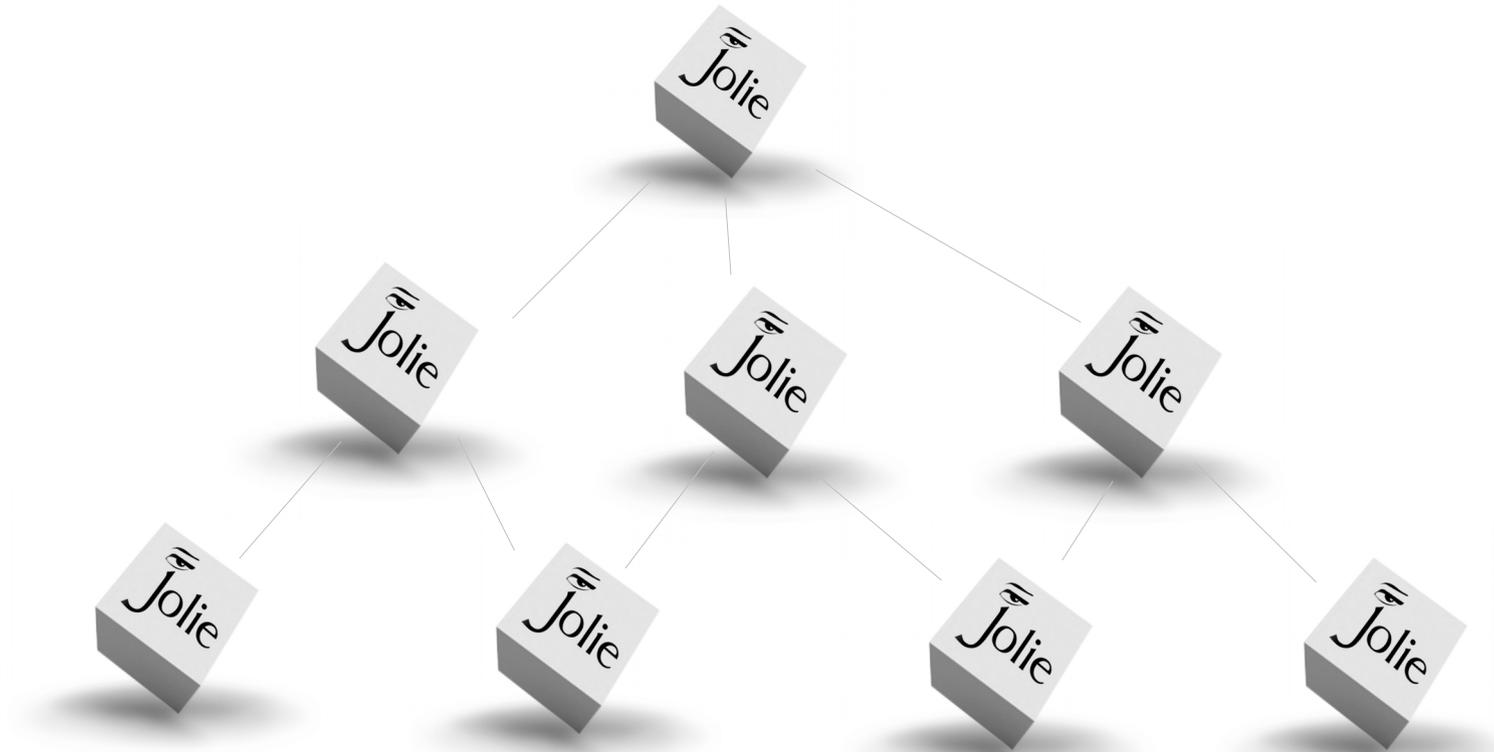
18 ANNI
UOMO

smartphone
TEORIA DEI
LINGUAGGI

I tre vantaggi competitivi di Jolie

- 1) Paradigmaticità:** il programmatore **scrive e pensa** direttamente a microservizi.
- 2) Autosufficienza:** non sono necessari tool aggiuntivi per la messa in esecuzione di un microservizio Jolie
- 3) Indipendenza:** un microservizio jolie non è legato a nessun protocollo di comunicazione in particolare

Paradigmaticità



Il programmatore programma e progetta l'architettura a microservizi in unico passaggio poiché il linguaggio fornisce tutti gli strumenti base per la creazione e manipolazione dei microservizi.

Conseguenza importante: durante la programmazione il programmatore si disinteressa di come dovranno essere eseguiti i microservizi

Dall'applicazione monolitica...



...esecuzione come singola
applicazione



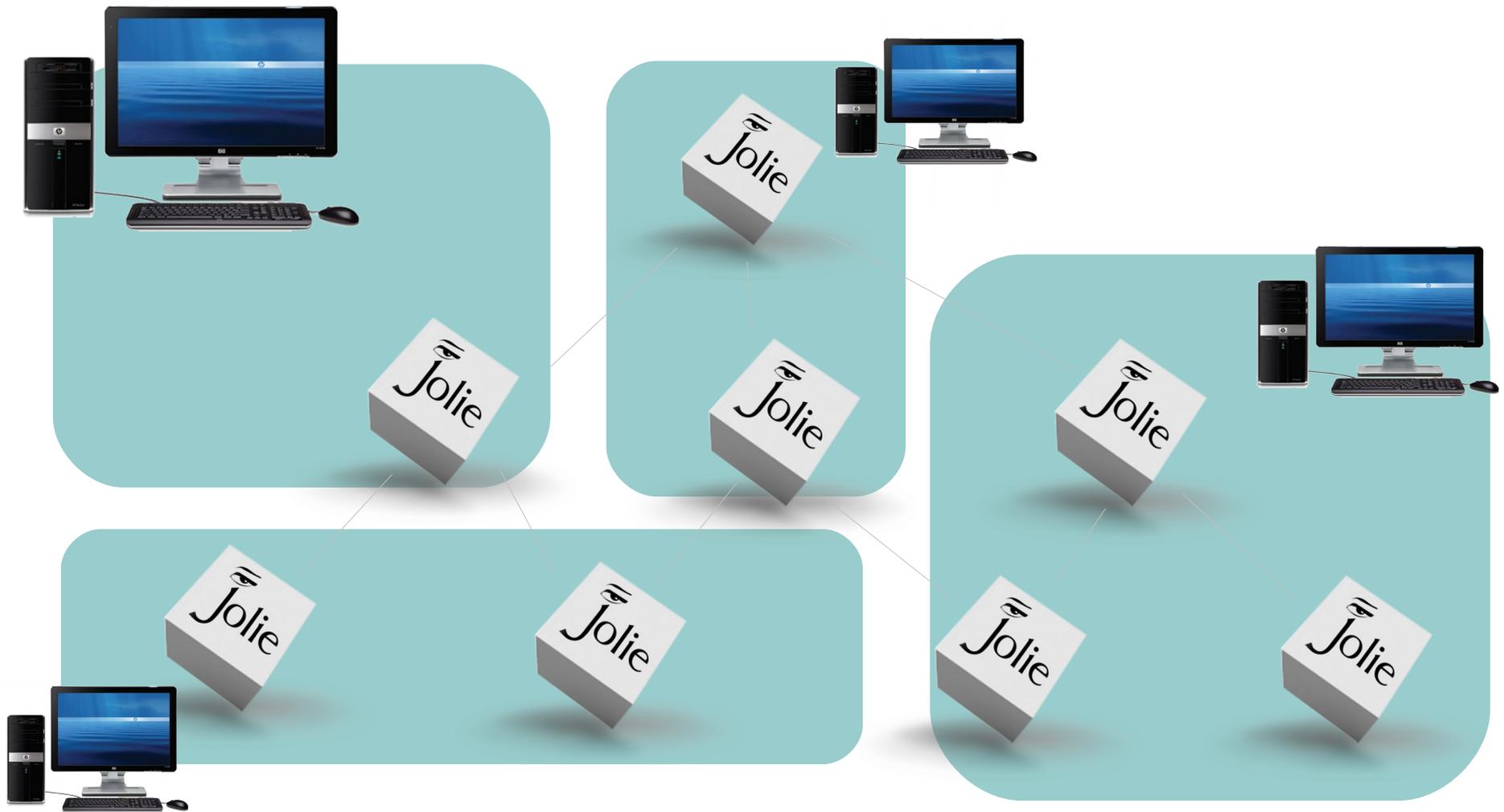
...verso...



...eseguiti in due macchine

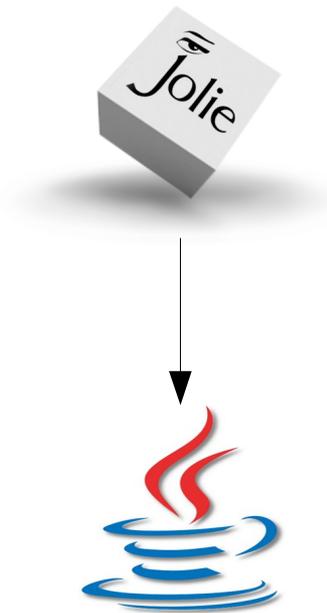


...i microservices



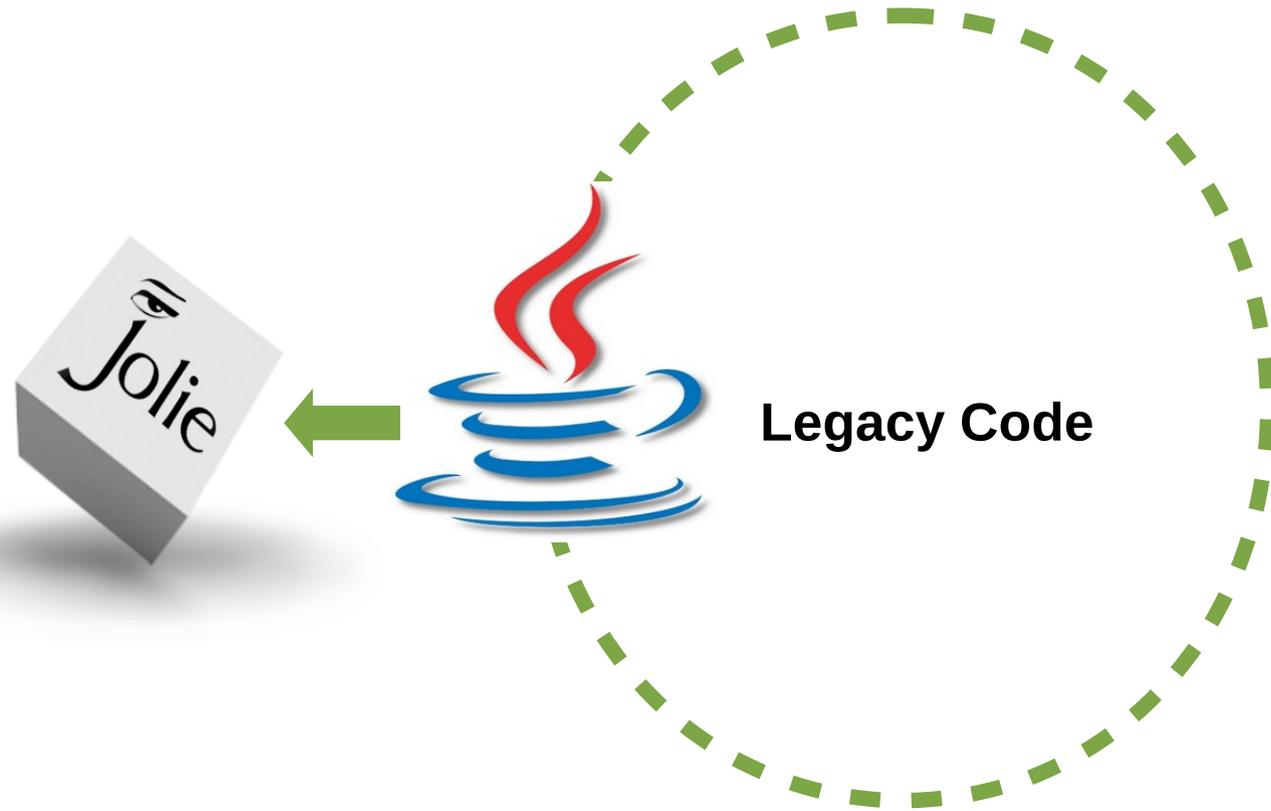
Autosufficienza

Un servizio Jolie necessita solamente di una **Java Virtual Machine** per funzionare



Non sono necessari framework
aggiuntivi per il suo funzionamento
No application server, No web server

Importazione di codice Legacy



Indipendenza



Si possono
sviluppare protocolli
custom specifici



Perché usare Jolie?

Perché è veloce!

- Jolie è lo strumento ideale per IT di qualsiasi dimensioni che abbiano l'ambizione di portarsi in casa tutto o parte dello sviluppo della propria architettura interna.



PROBLEMA

Ho un flusso da inserire nel mio sistema e mi servirebbe averlo entro brevissimo tempo



con Jolie posso iniziare lo sviluppo immediatamente dal problema



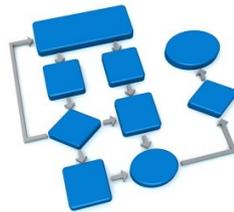
VANTAGGIO

Posso permettermi di fare un'analisi di partenza più approssimata e di refinarla durante il processo



Perché usare Jolie?

Perché è generale!



PROBLEMA

Ho necessità di orchestrare servizi già esistenti per portare a termine un flusso di controllo



con Jolie posso programmare **orchestratori** allo stesso modo con il quale programmo microservizi semplici!



VANTAGGIO

Non devo cambiare tecnologia e non mi servono competenze speciali per risolvere il problema



Perché usare Jolie? Perché è leggero!



PROBLEMA

Ho sviluppato la logica per il mio flusso, ora come faccio a metterlo in esecuzione?



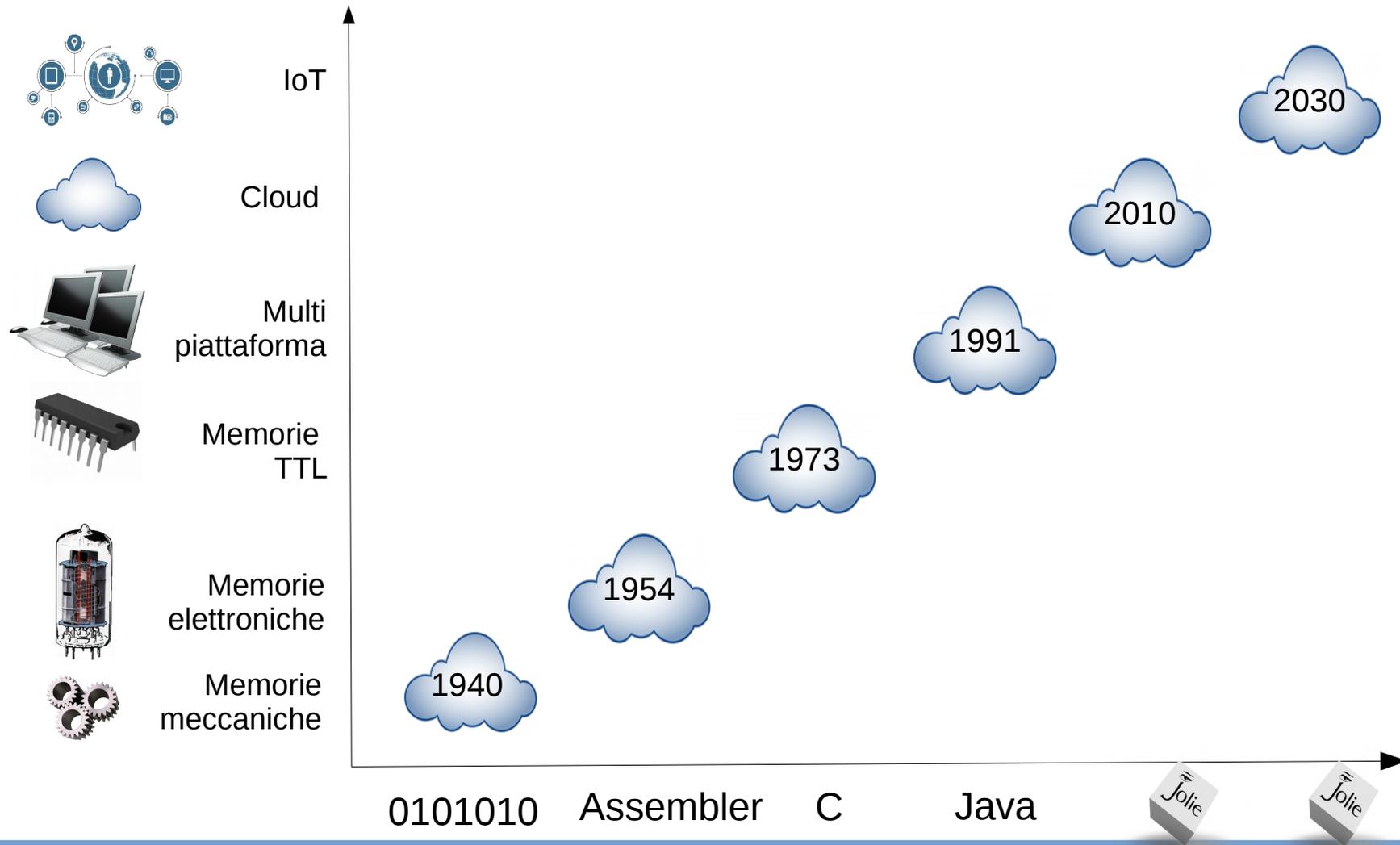
con Jolie è sufficiente avere installata una JVM e l'interprete di Jolie per poter lanciare tutti i microservizi che desideri su una qualsiasi macchina



VANTAGGIO

Dal punto di vista sistemistico ho meno problemi da dover gestire

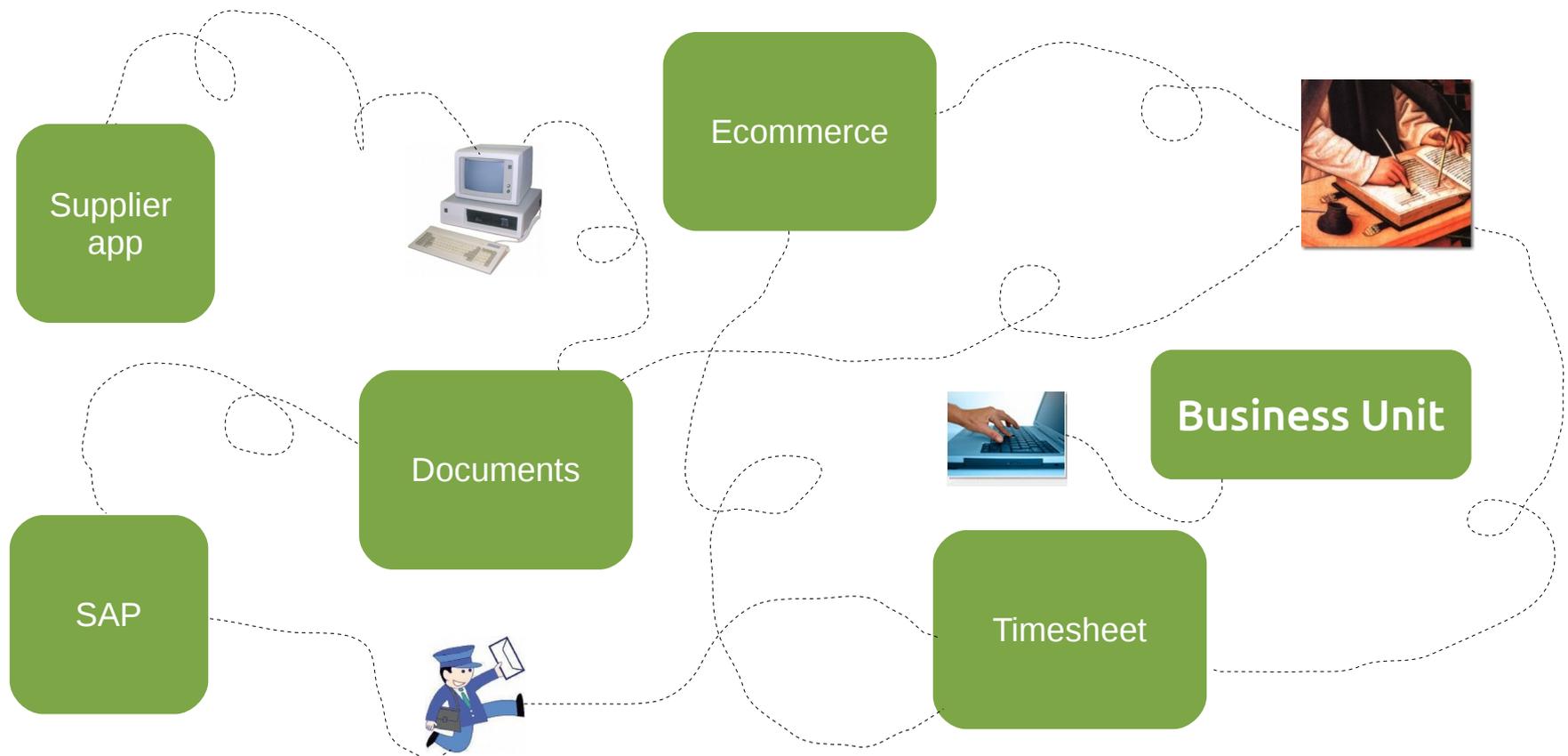
L'interdipendenza tra linguaggi e supporti



Integrazione di sistemi con Jolie

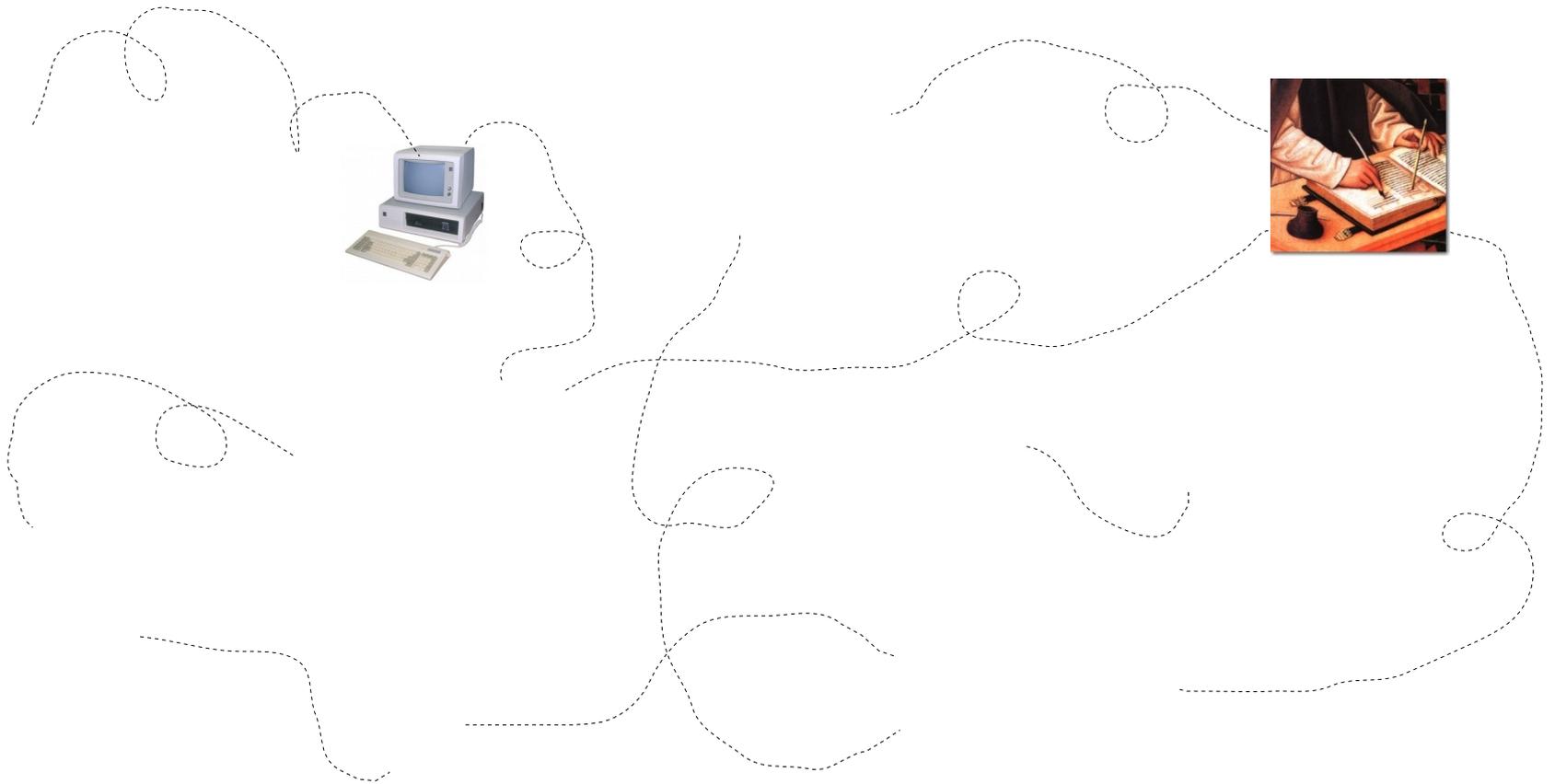
Integrare sistemi

La prima classe di problemi risolvibili con l'utilizzo di Jolie sono quelli di integrazione



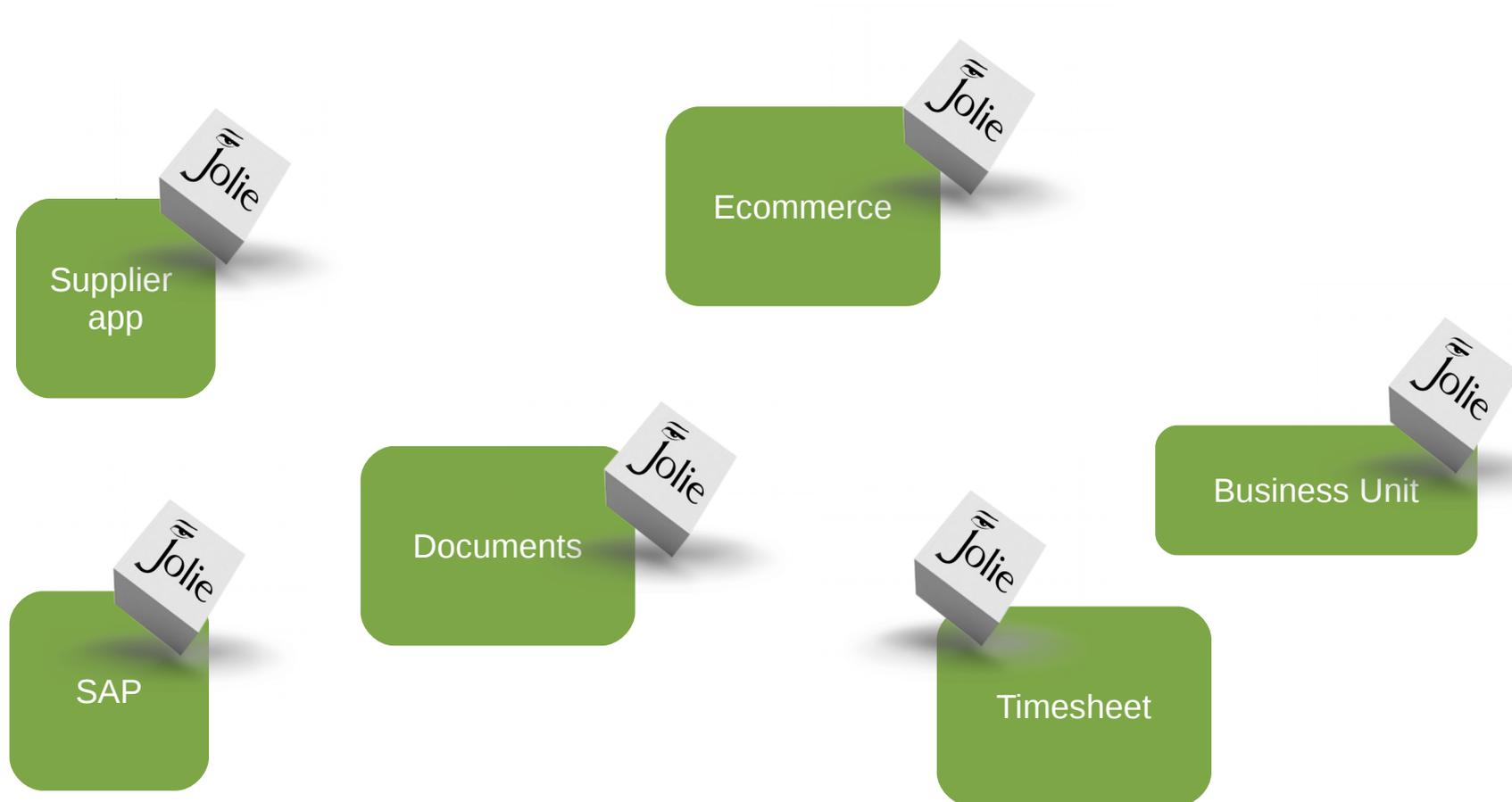
Cosa significa integrare sistemi?

L'integrazione di sistemi significa razionalizzare ed ingegnerizzare con un approccio sistematico e preciso l'insieme disomogeneo di soluzioni nate per lo scambio di dati tra applicativi diversi



L'utilizzo di Jolie

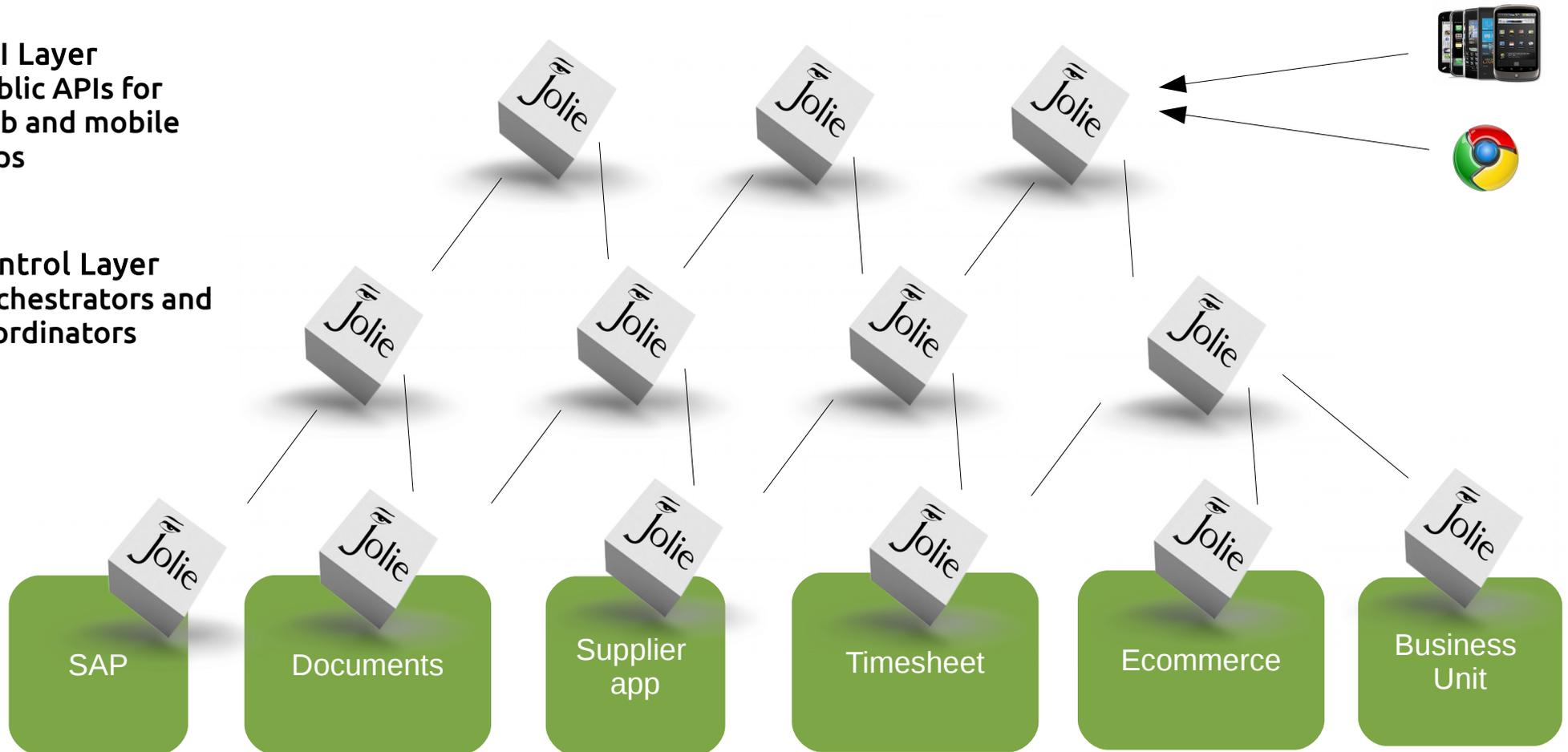
Ogni applicazione viene vista come un microservizio Jolie



Come appare un sistema integrato complesso

API Layer
Public APIs for
web and mobile
apps

Control Layer
Orchestrators and
coordinators



Prodotti

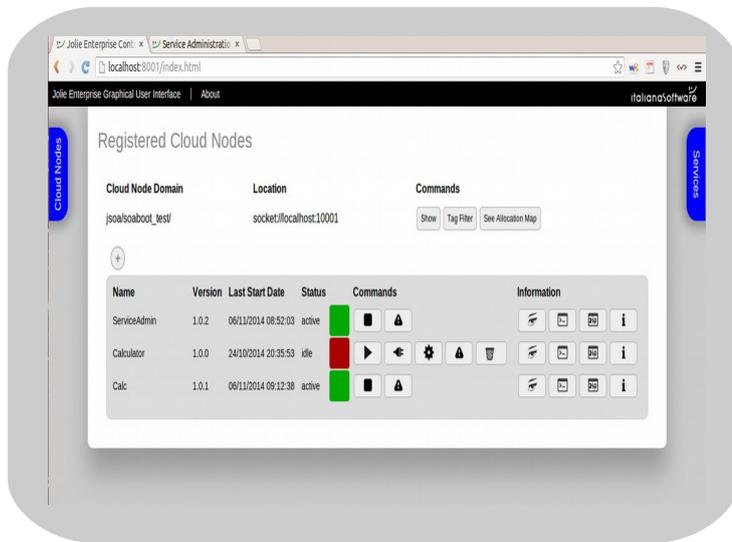
Jolie Enterprise



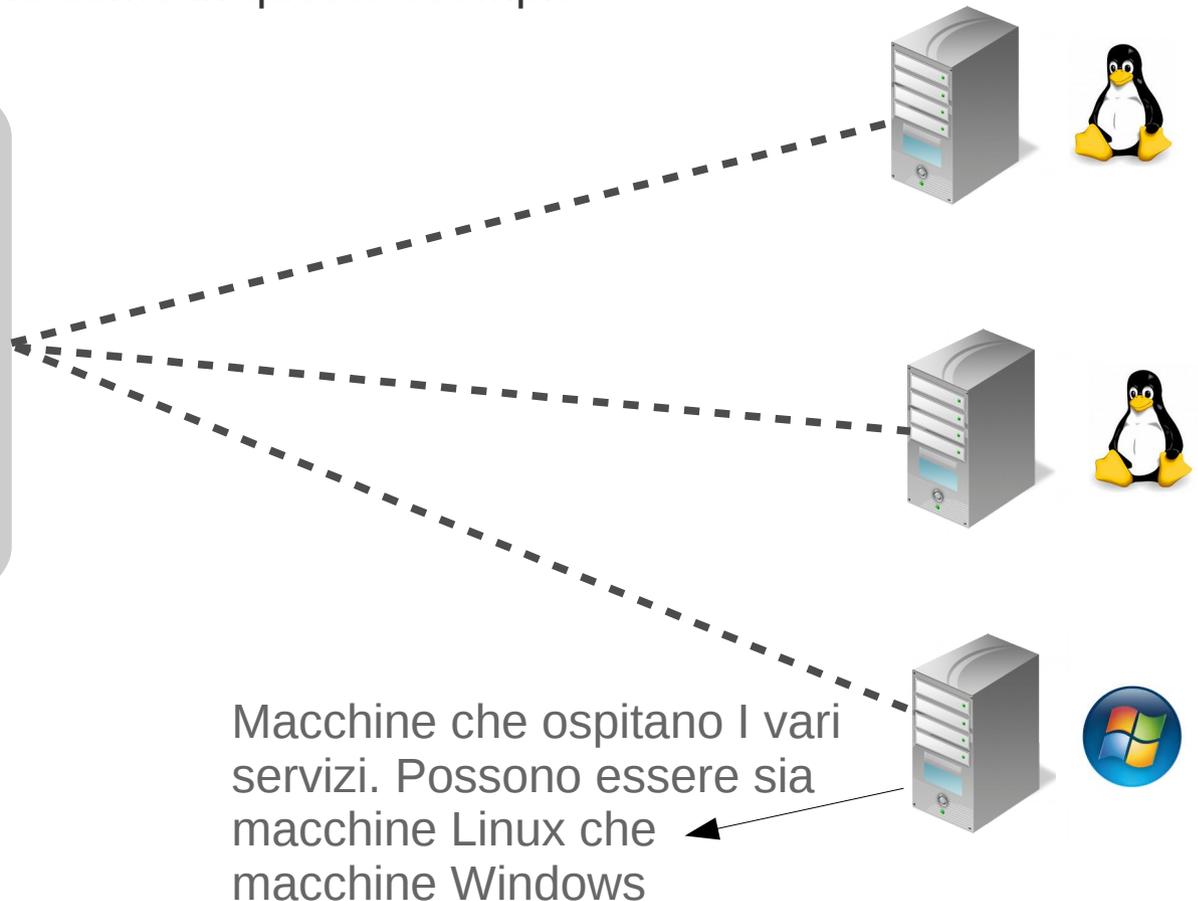
Jolie Enterprise è un tool che ti permette in modo agevole e semplice di fare deploy di microservlets tenendone monitorate le attività e ed il funzionamento

Tanti microservizi in macchine diverse

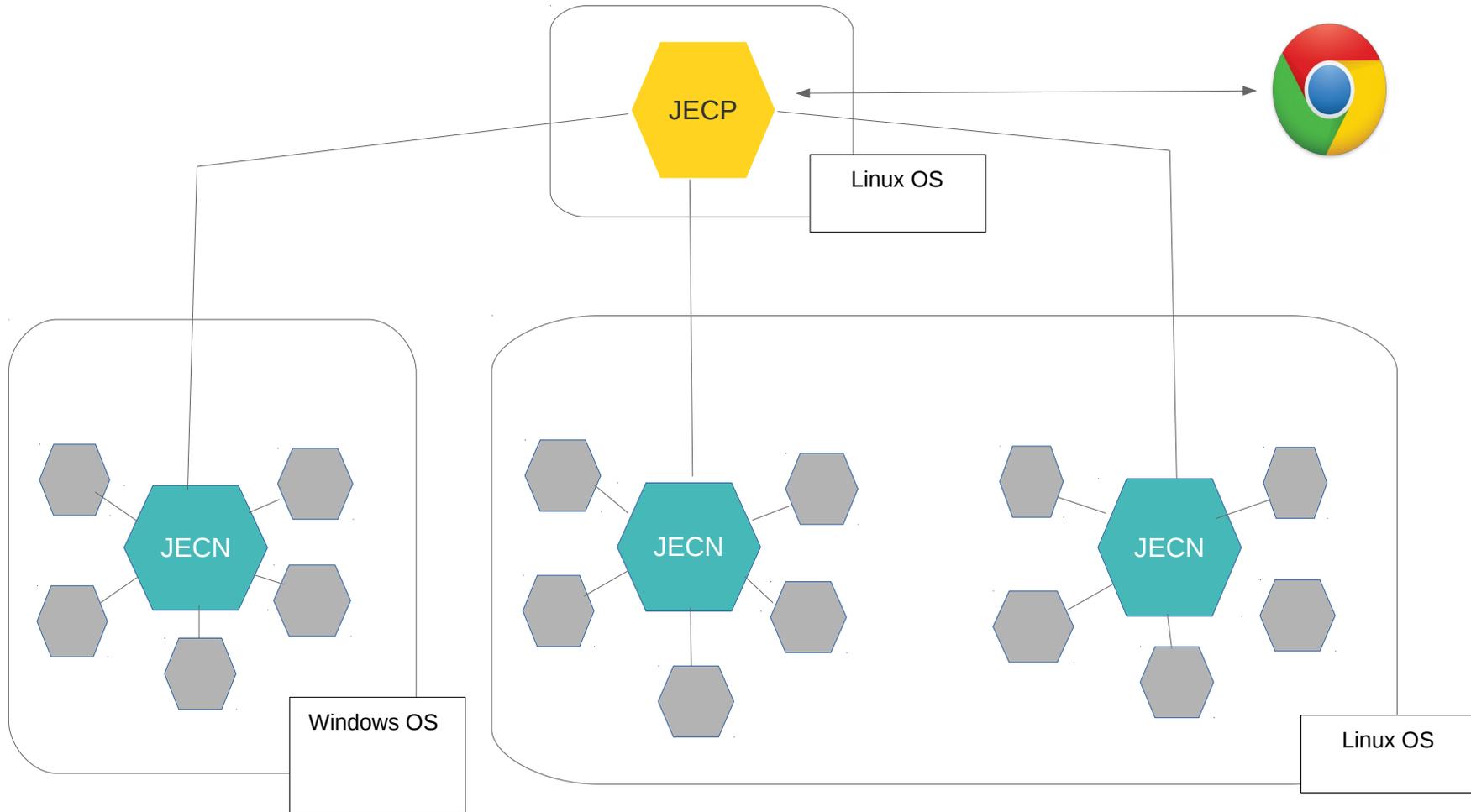
Il sistema può essere eseguito su un'unica macchina o su più macchine come in questo esempio



↓
Pannello di controllo verso cui confluiscono tutti i dati di tutti i servizi



L'architettura bassa



Il pannello di controllo

Registered Cloud Nodes

Cloud Node Domain: jsoa/soaboot_test/ Location: socket://localhost:10001 Commands: Show Tag Filter See Allocation Map

(+)

| Name | Version | Last Start Date | Status | Commands | Information |
|--------------|---------|---------------------|--------|----------|-------------|
| ServiceAdmin | 1.0.2 | 06/11/2014 08:52:03 | active | | |
| Calculator | 1.0.0 | 24/10/2014 20:35:53 | idle | | |
| Calc | 1.0.1 | 06/11/2014 09:12:38 | active | | |

Funzionalità: avvio, stop, configurazione, test

Logging e Monitoring

Jolie Service Log & Stat Console | Archive About italianaSoftware

Page: < 0 > Records in a page: 50 Total Records: 11

| index | Domain | Service | Scope | id | Type | Priority | Operation | Message | Timestamp |
|-------|--------|------------|-------|------------------|---------|----------|-----------|---------|----------------------------|
| 1 | temp | calculator | main | OperationEnded | SUCCESS | NORMAL | sum | | 06/11/2014 09:12:54.796 |
| 2 | temp | calculator | main | OperationStarted | REQUEST | NORMAL | sum | null | 06/11/2014 09:12:54.794 |
| 3 | temp | calculator | main | OperationEnded | SUCCESS | NORMAL | check | | 06/11/2014 09:12:38.776 |
| 4 | temp | calculator | main | OperationStarted | REQUEST | NORMAL | check | null | 06/11/2014 09:12:38.774 |
| 5 | temp | calculator | main | MonitorAttached | | NORMAL | | null | 06/11/2014 09:12:36.712 |
| 6 | temp | calculator | main | MonitorAttached | | NORMAL | | null | 06/11/2014 09:10:24.932 |
| 7 | temp | calculator | main | MonitorAttached | | NORMAL | | null | 06/11/2014 09:08:14.840 |
| 8 | temp | calculator | main | MonitorAttached | | NORMAL | | null | 06/11/2014 08:56:32.829 |

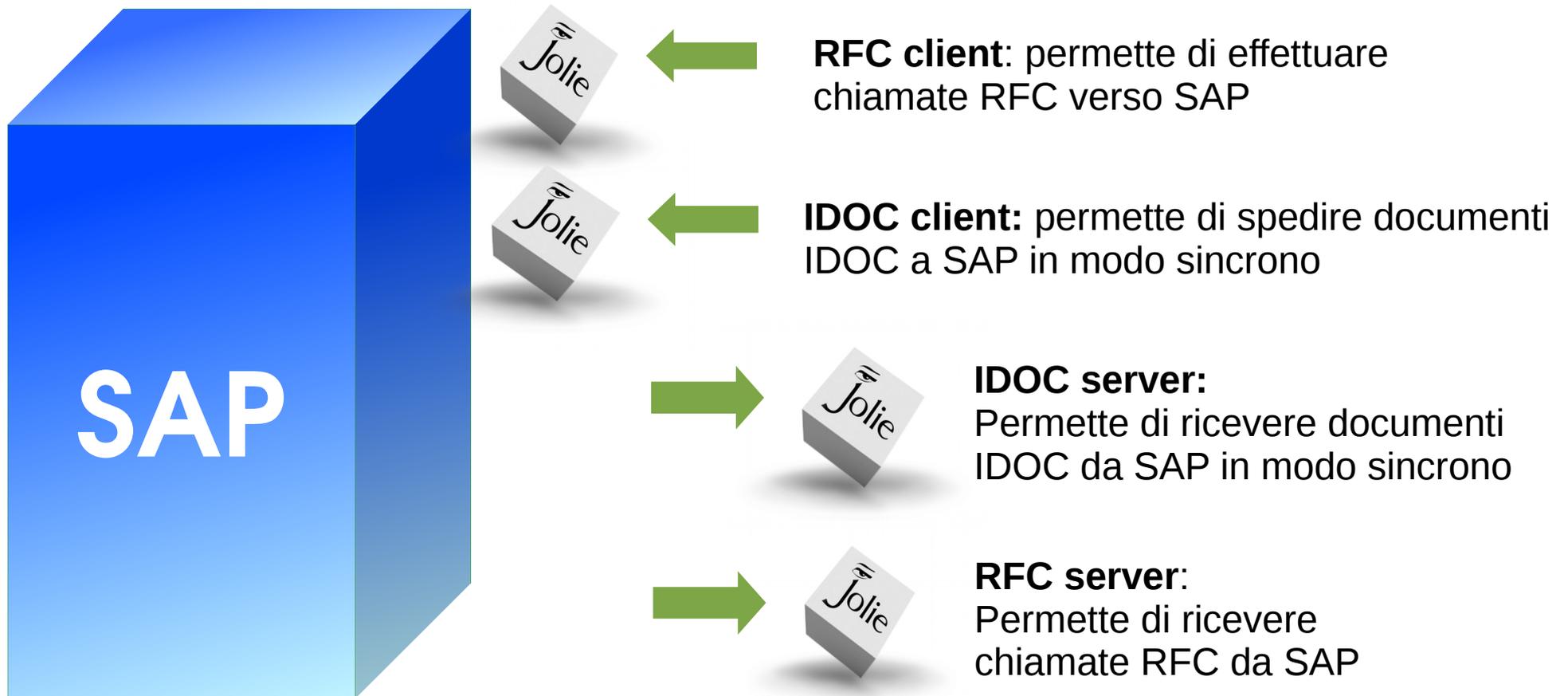
Funzionalità: consultazione log, statistiche tempi e consumo memoria, possibilità di associare eventi

Perché usare Jolie Enterprise?

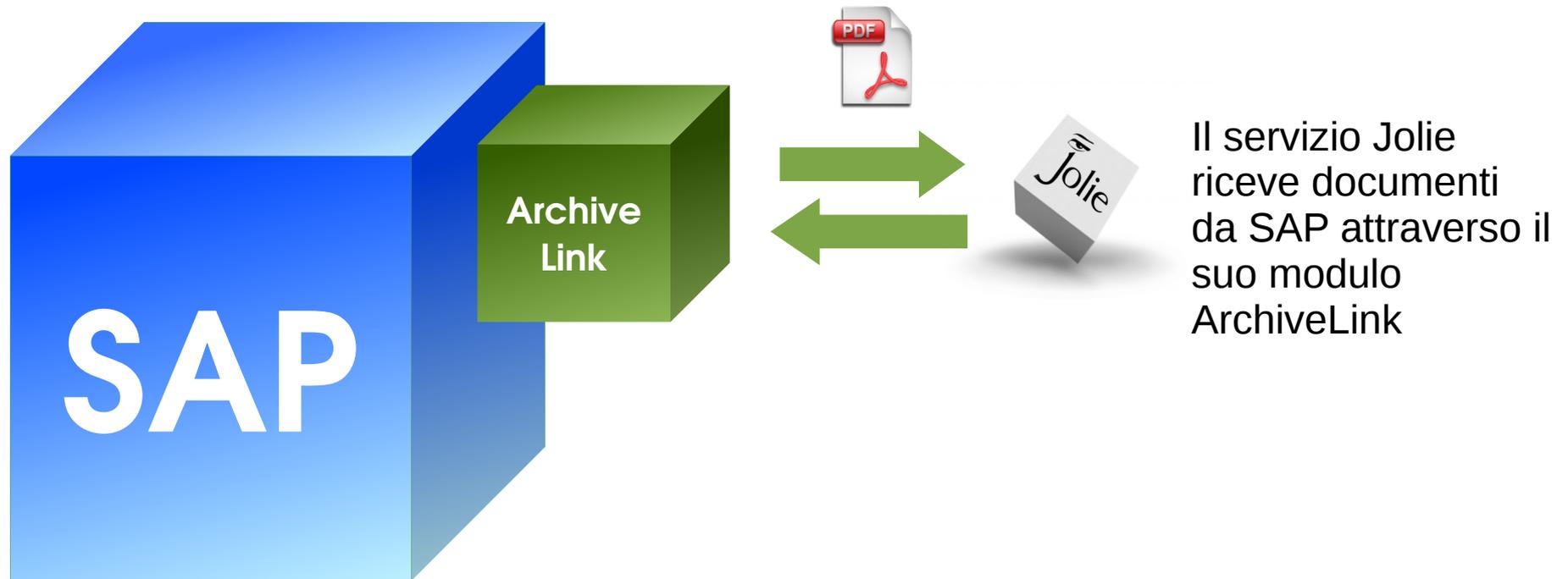


- Possibilità di **amministrare** i microservizi attivi sui diversi cloud node
- Soluzione di **monitoring** built-in senza necessità di avere strumenti in più
- **Non vincolante** dal punto di vista della messa in esecuzione dei microservizi
- **Multiplatforma**

Il connettore JSAP



Il connettore per SAP ArchiveLink



italianaSoftware

*Un sistema software complesso non è mai un'entità statica ma è
soggetto a continue modifiche per potersi adattare agli stimoli
esterni... come il pensiero umano*