

The background features a complex, colorful mosaic pattern. On the left, there are concentric, semi-circular bands of small, multi-colored squares. On the right, the pattern transitions into a more sparse, scattered arrangement of similar colored squares and rectangles. A large, white rectangular box is positioned in the center-left, containing the title and author's name.

# Cloud Archive and Computation

**ROMOLO POLITI**



# Panoramica del corso



# Panoramica del Corso

---

## Cloud

Struttura del Cloud

Dati nel Cloud

Calcolo nel Cloud

## Dati

Dati e metadati

Archiviazione

DB Relazionali e non

## Calcolo

Recupero

Manipolazione

Visualizzazione

## Ambiente:

Virtualizzazione e container

Microservices

DevOps

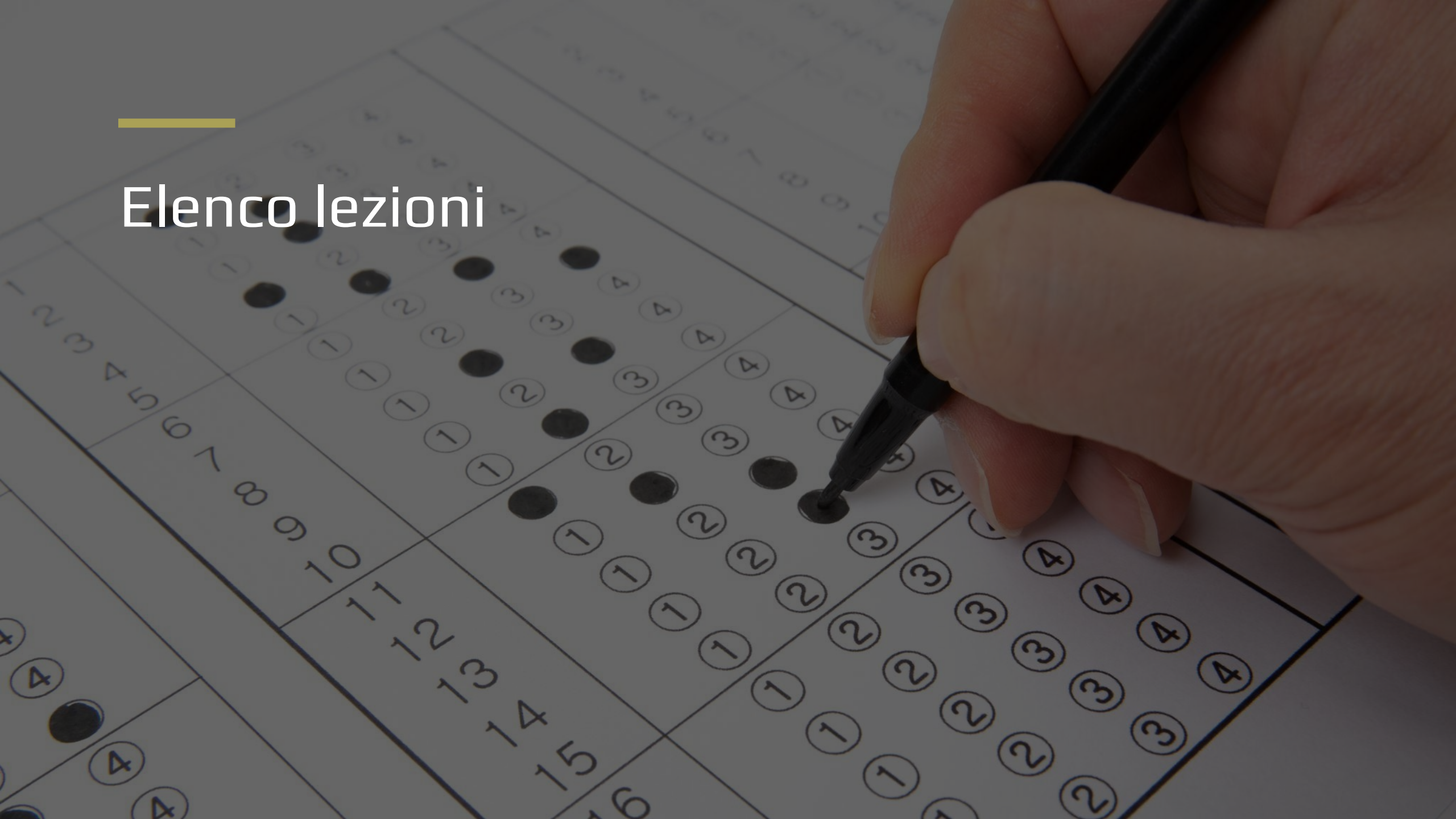
## Programmazione:

Fondamenti di programmazione

Python

Versioning e Documentazione

# Elenco lezioni



# Elenco lezioni

---

- [9 Maggio 2022](#)



# Lezione 9 Maggio 2022



# Tools

---

- La presentazione e gli esempi del corso sono su GitHub:  
<https://github.com/RomoloPoliti-INAF/PhDCourse2022>
- Per gli esempi utilizzeremo Python 10.3
- Come framework di sviluppo Microsoft Visual Studio Code  
<https://code.visualstudio.com>

# Struttura

---

- La lista degli argomenti mostrata in precedenza è stata costruita per categorie.
- Noi seguiremo un percorso guidato dagli esempi per meglio capire la filosofia che c'è dietro.
- Dopo l'introduzione alla programmazione svilupperemo un esempio di programma complesso (Macchina a Stati).
- In ultimo svilupperemo una WebApp e la prepareremo per la distribuzione su container
- Per alcuni argomenti non scenderemo nel dettaglio perché lo scopo del corso è dare una visione generale ed anche perché non c'è il tempo per approfondire tutti gli argomenti. In ogni modo molti dettagli sono nelle slide o nei link riportati.





# Tipi di Cloud



# Tipi di Cloud

---

In Promise



Out Promise



Google Cloud

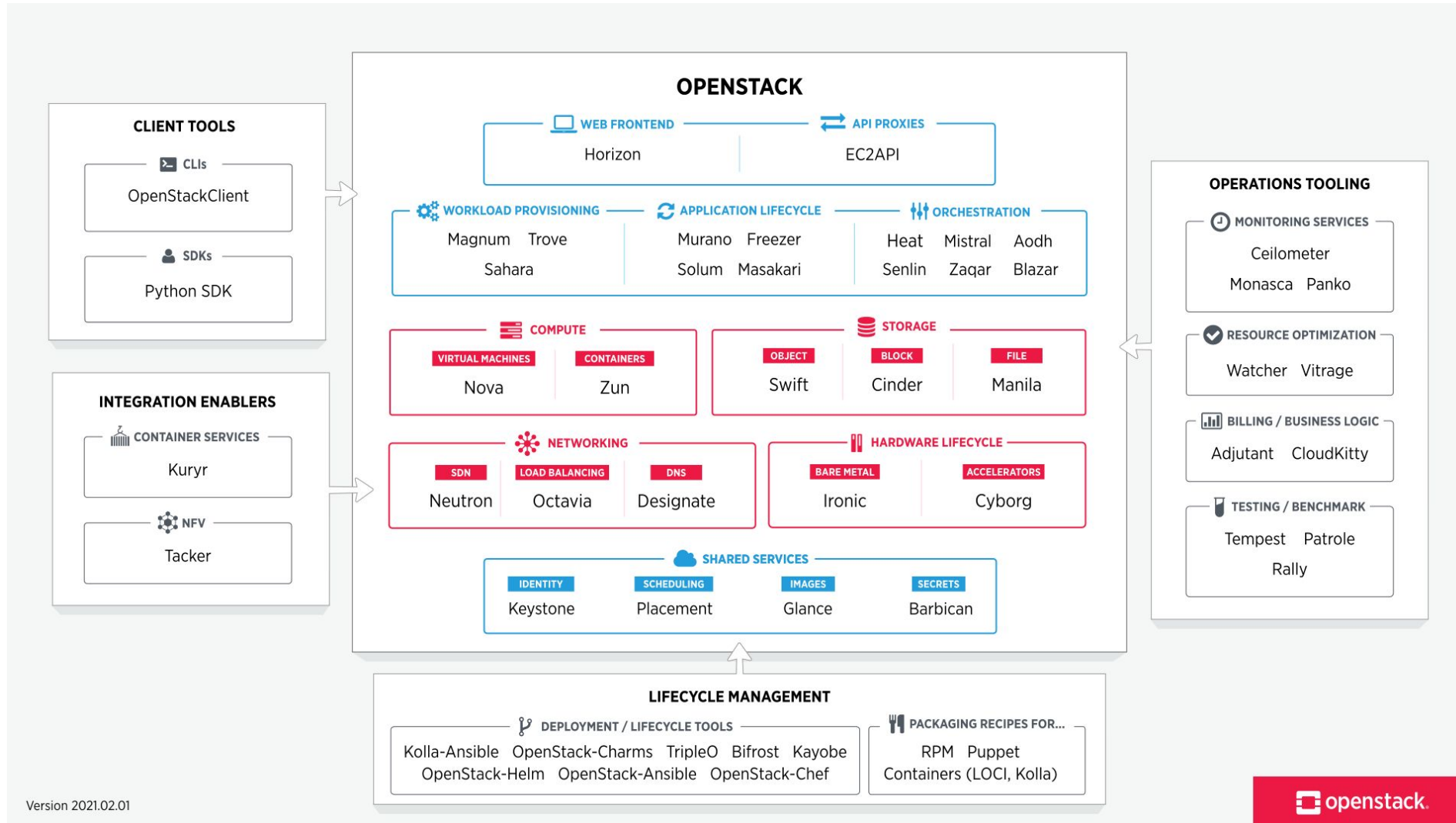




# Struttura del Cloud

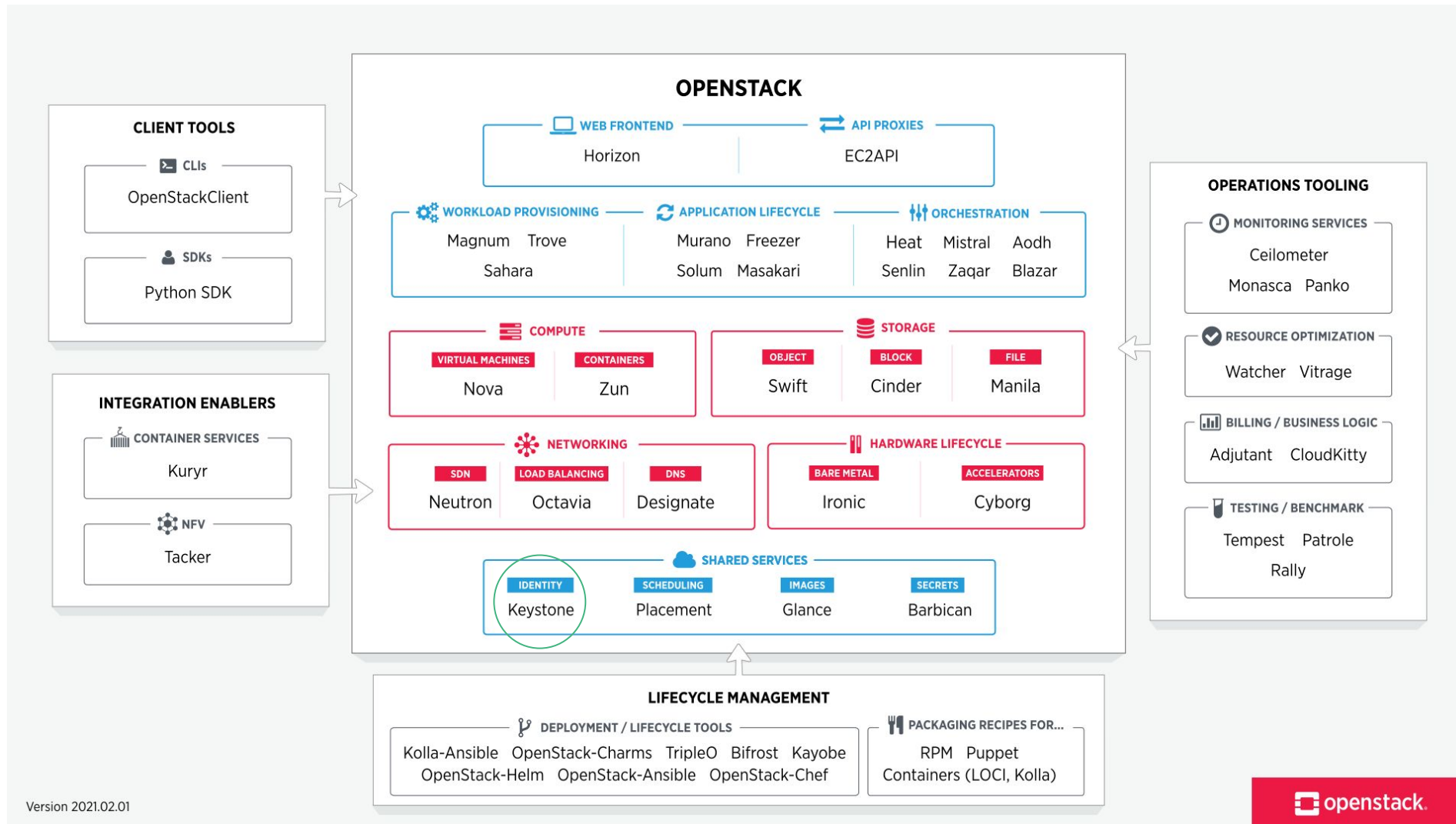


# Struttura di un cloud



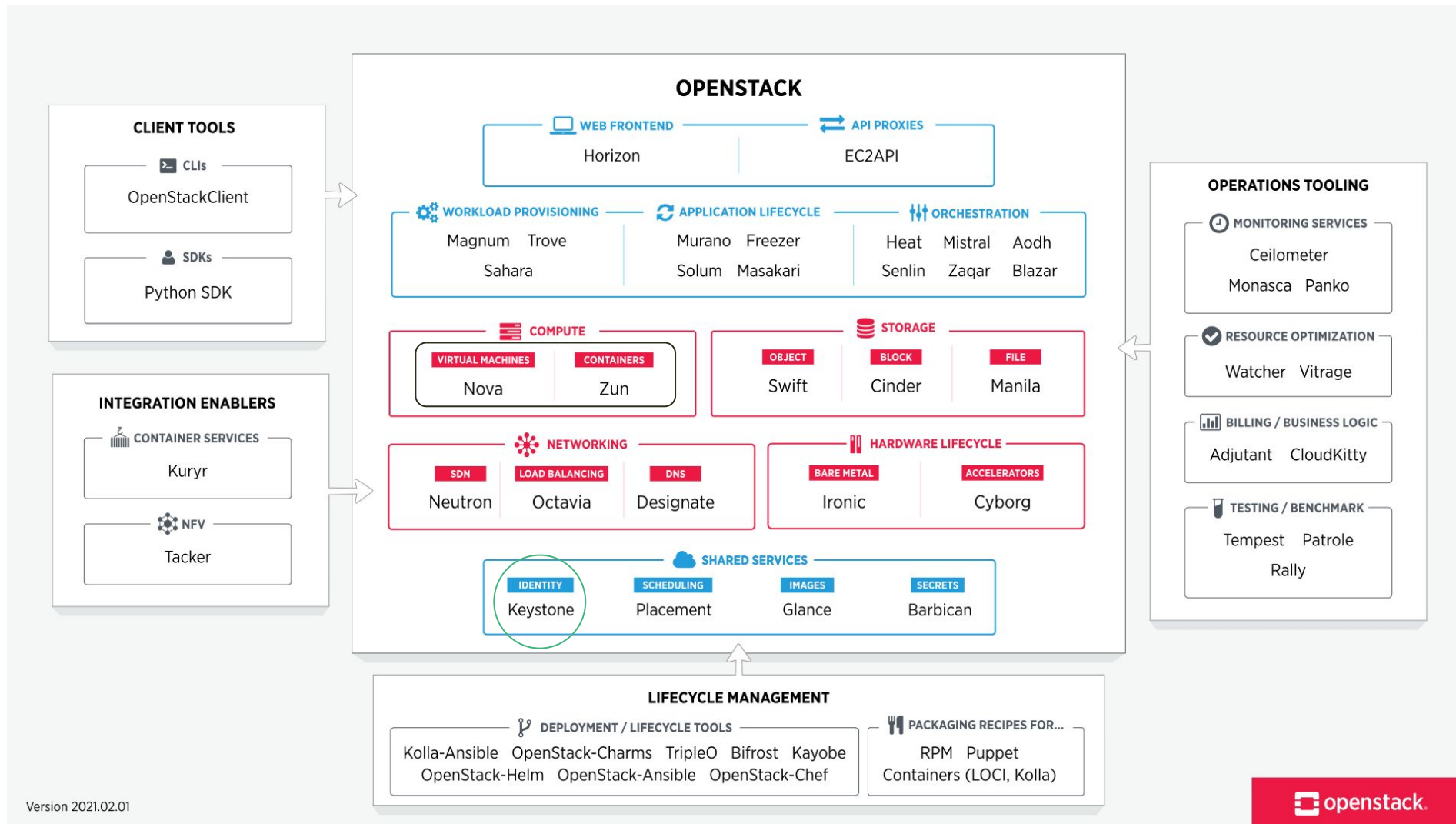
# Struttura di un cloud

- Identity



# Struttura di un cloud

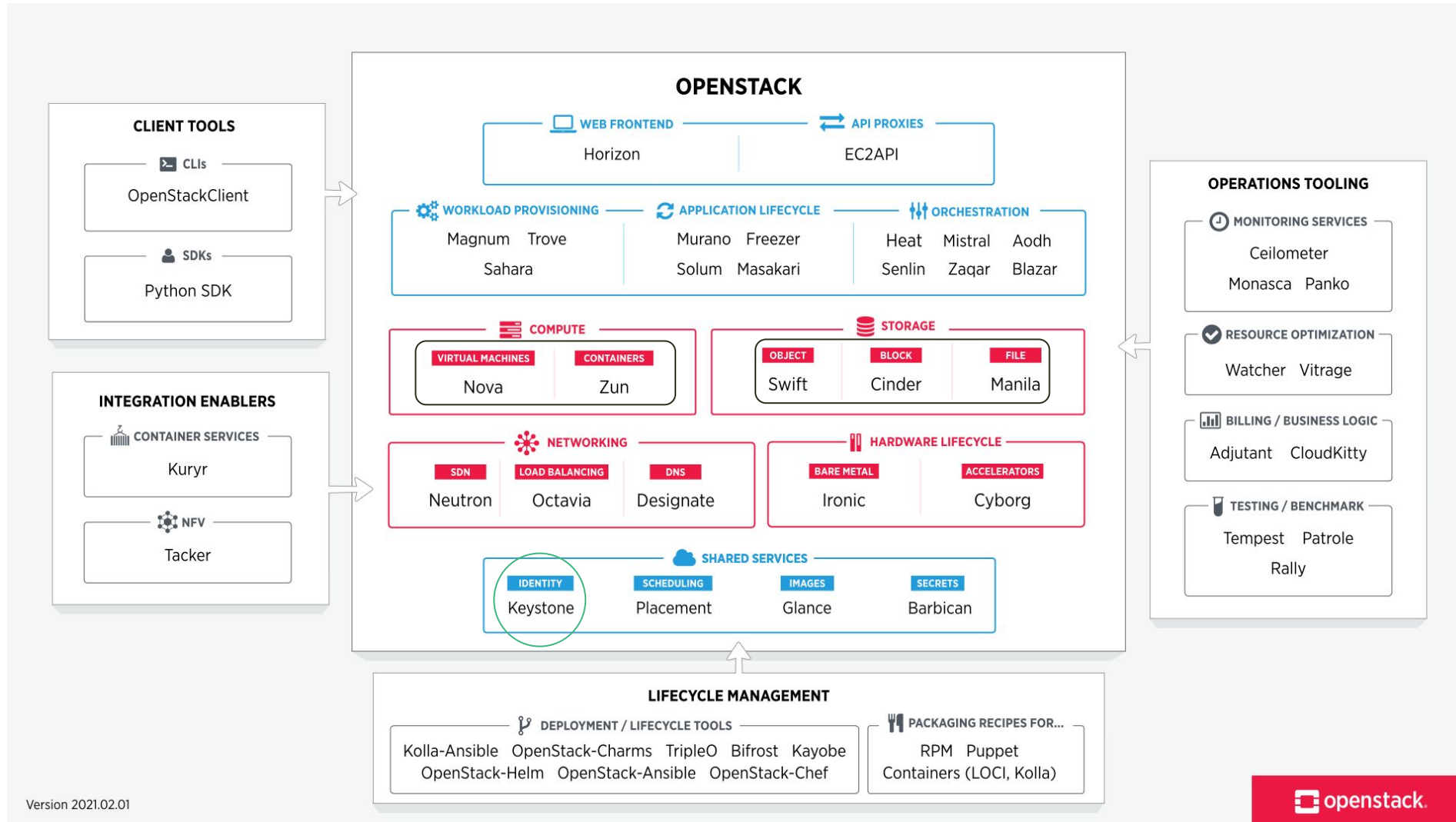
- Identity
- Compute
- e





# Struttura di un cloud

- Identity
- Compute
- Storage



# Componenti Principali

---




- IAM (Identity and Access Management)

# Componenti Principali

- IAM (Identity and Access Management)
  - verifica identità
  - lista di risorse dedicate
  - privilegi
  - Credito (cloud off premise)



Sign in with your Google Account



Email

Password

**Sign In**

☒ Stay signed in [Need help?](#)

# Componenti Principali

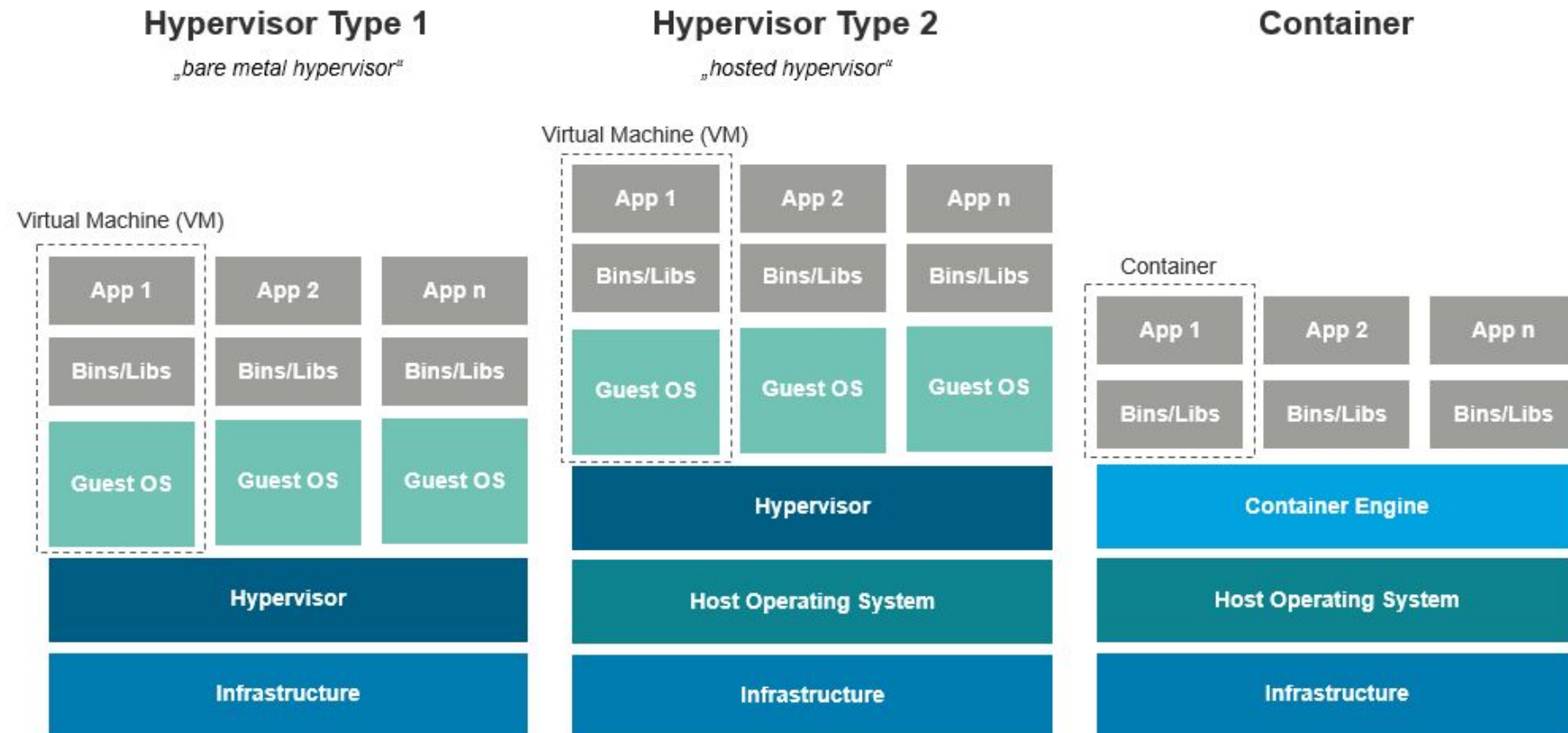


- IAM (Identity and Access Management)

# Componenti Principali



- IAM (Identity and Access Management)



# Componenti Principali

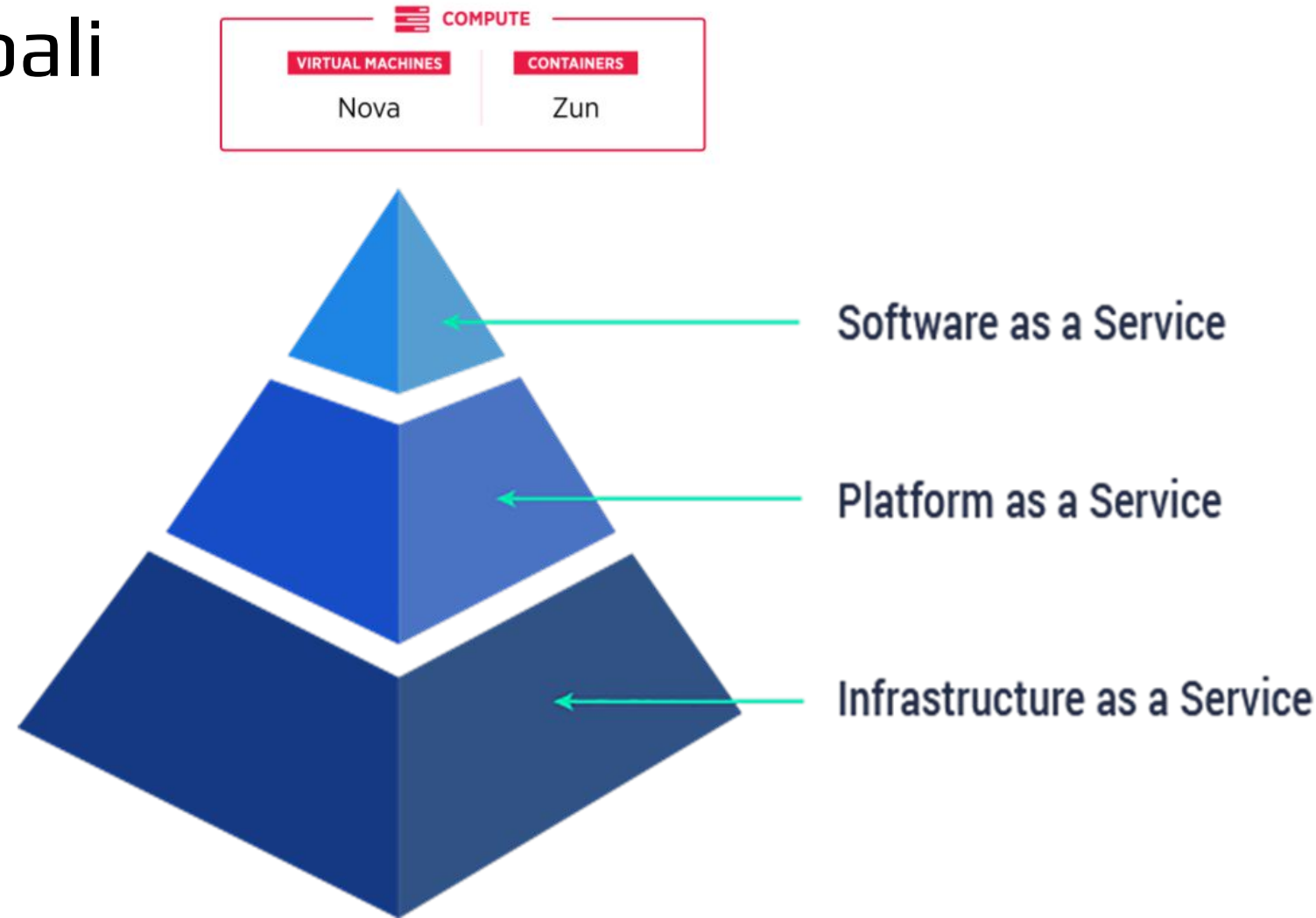


- IAM (Identity and Access Management)
- IaaS
- PaaS
- SaaS



# Componenti Principali

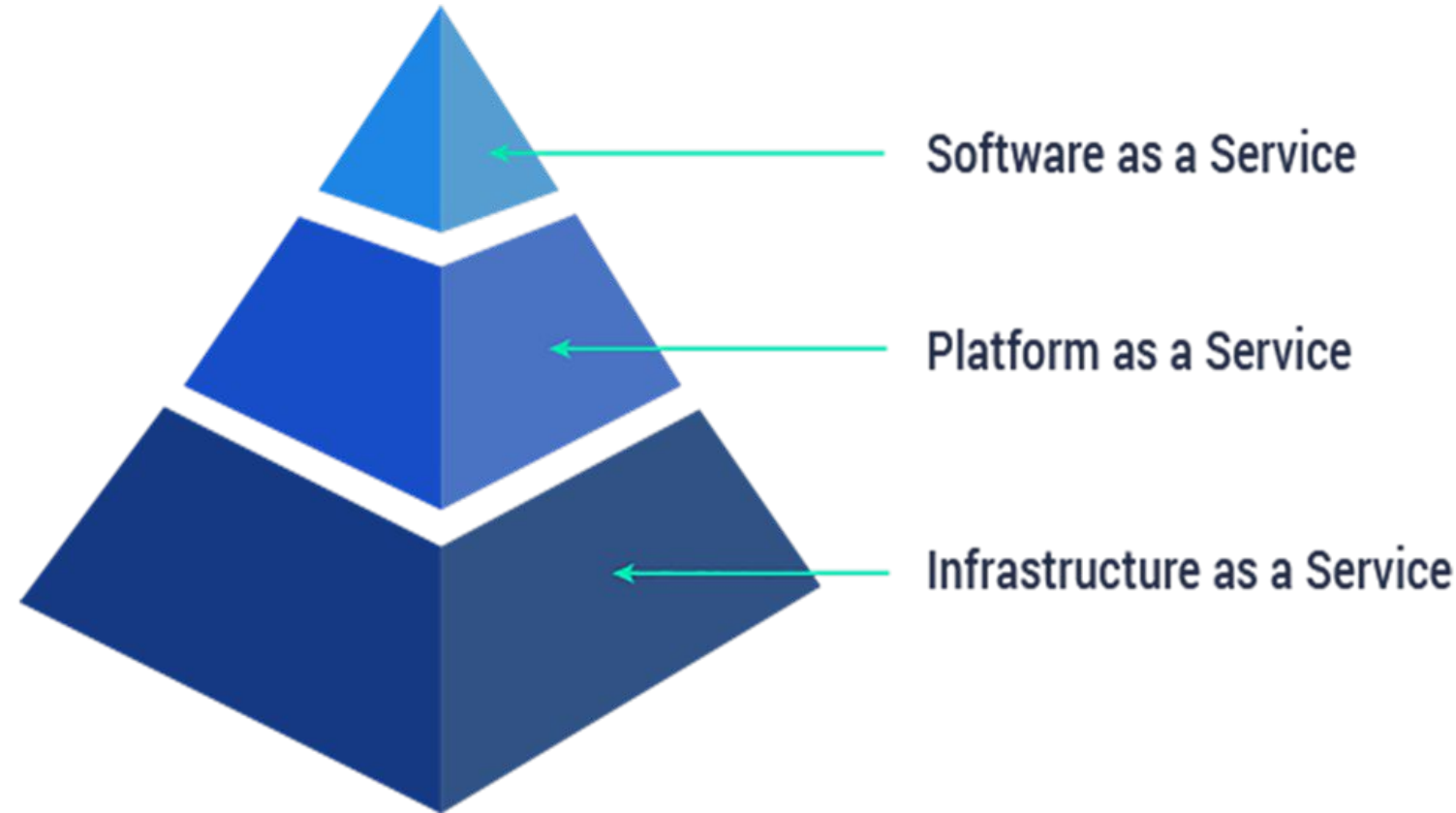
- IAM (Identity and Access Management)
- IaaS
- PaaS
- SaaS



# Componenti Principali



- IAM (Identity and Access Management)
- IaaS
- PaaS
- SaaS

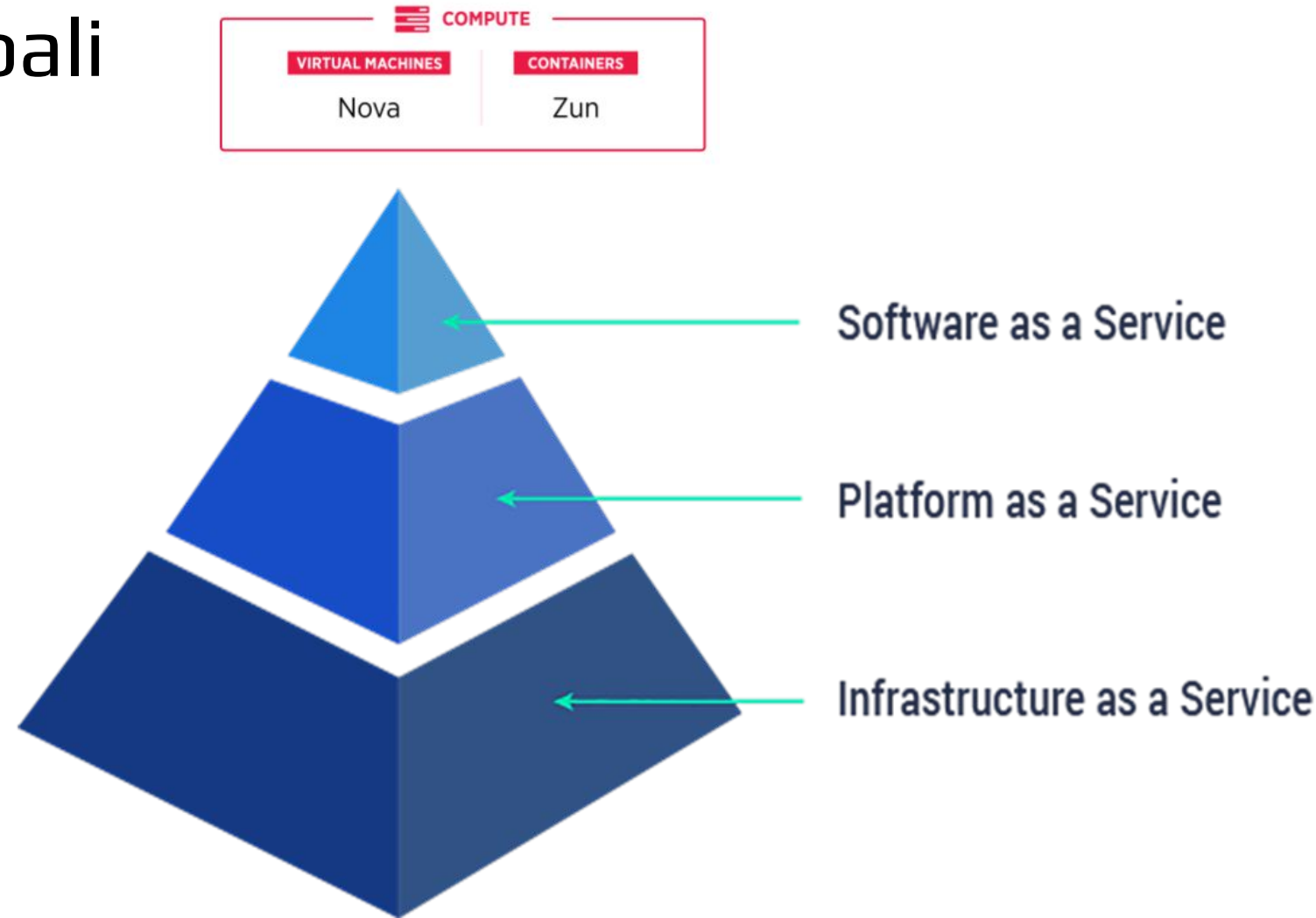


## IaaS

Il provider offre un hardware virtuale (CPU, RAM, spazio e schede di rete) e quindi la flessibilità di un'infrastruttura fisica, senza l'onere per l'utente, della gestione fisica dell'hardware

# Componenti Principali

- IAM (Identity and Access Management)
- IaaS
- PaaS
- SaaS

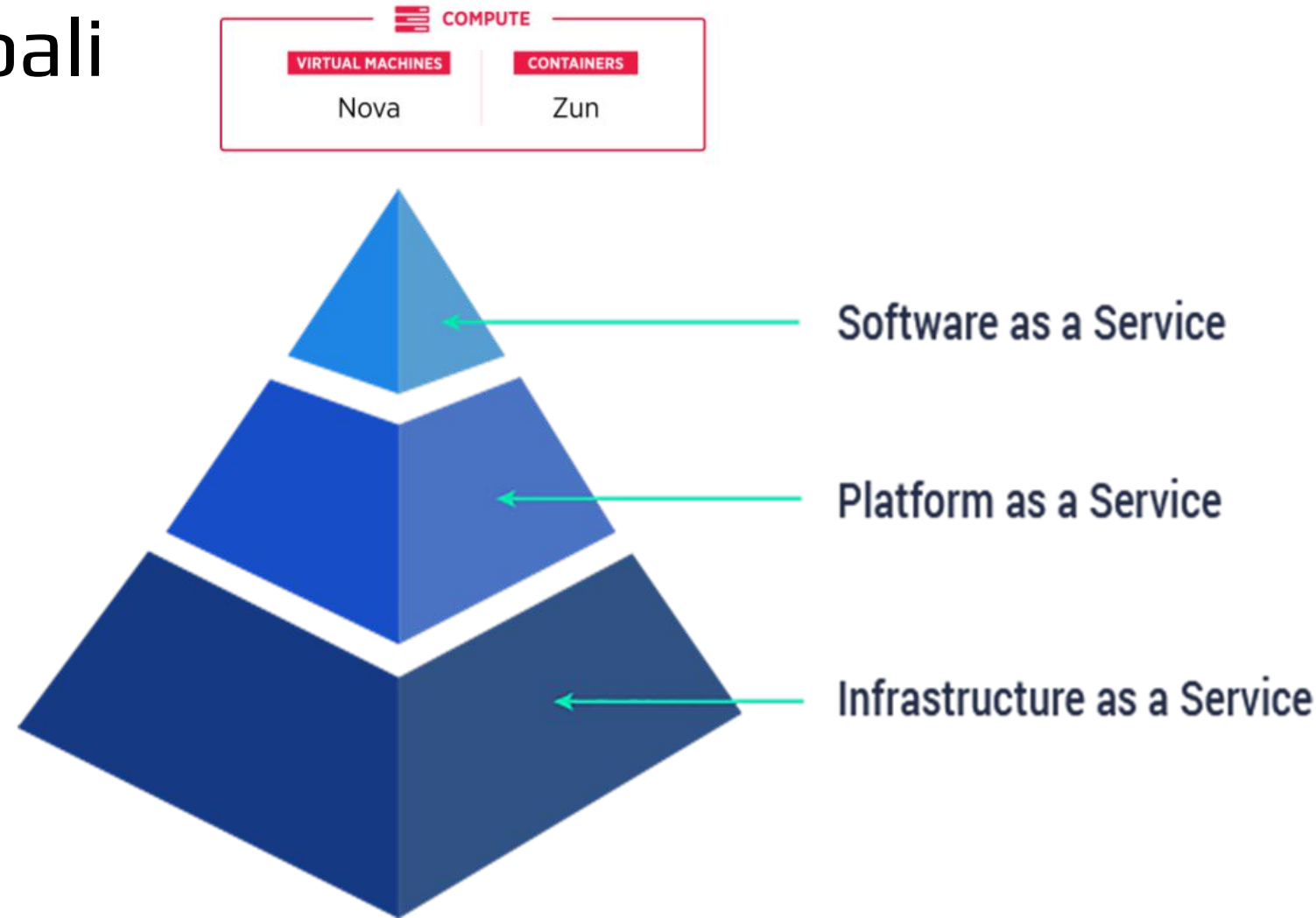


## PaaS

Il provider si occupa dell'infrastruttura hardware, mentre l'utente dovrà installare il sistema operativo e occuparsi di sviluppare la sua applicazione

# Componenti Principali

- IAM (Identity and Access Management)
- IaaS
- PaaS
- SaaS

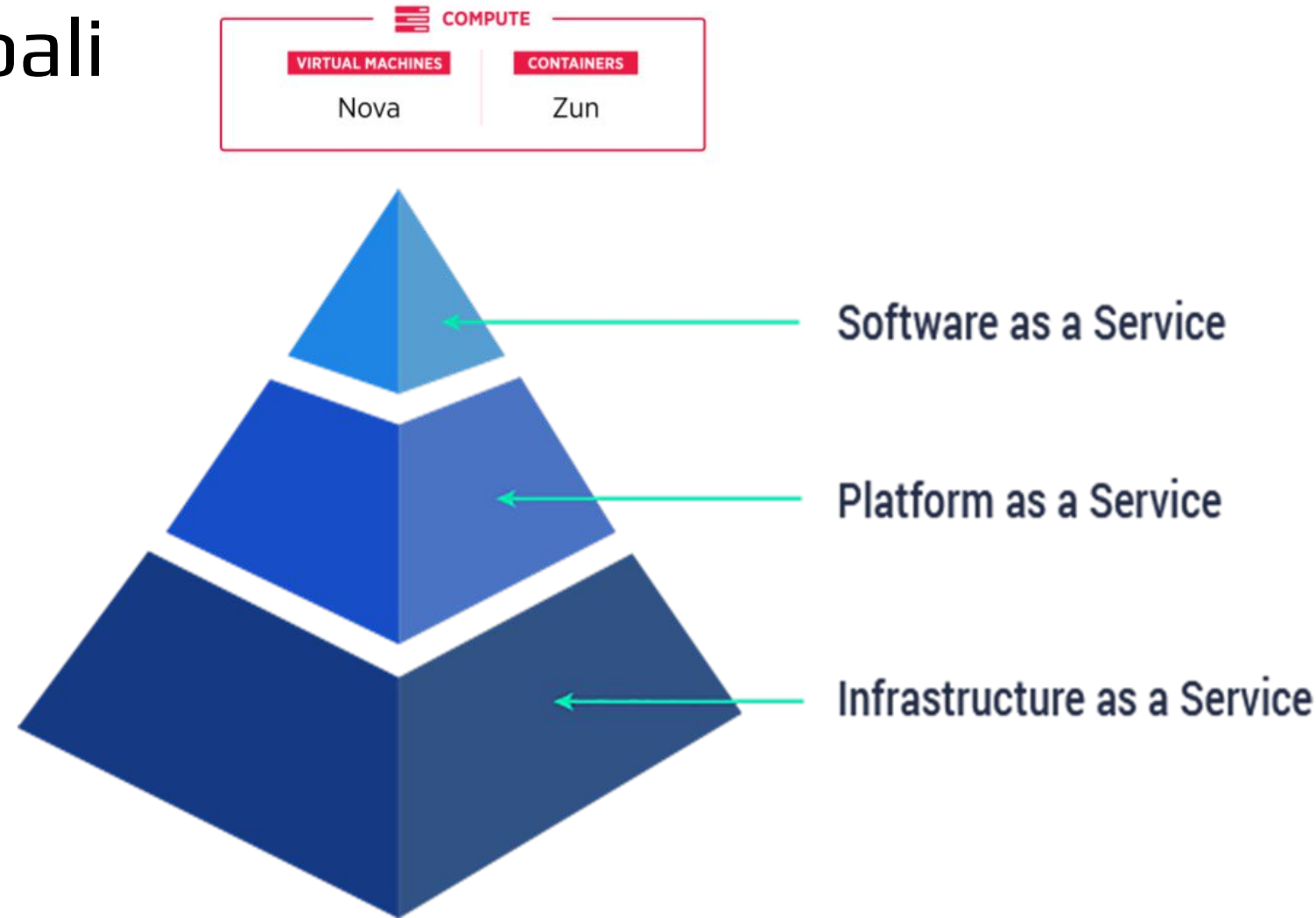


## SaaS

L'utente finale non ha bisogno di nessuna conoscenza informatica per utilizzare l'applicazione o i servizi erogati. I servizi sono utilizzabili semplicemente con una connessione internet e un browser.

# Componenti Principali

- IAM (Identity and Access Management)
- IaaS
- PaaS
- SaaS

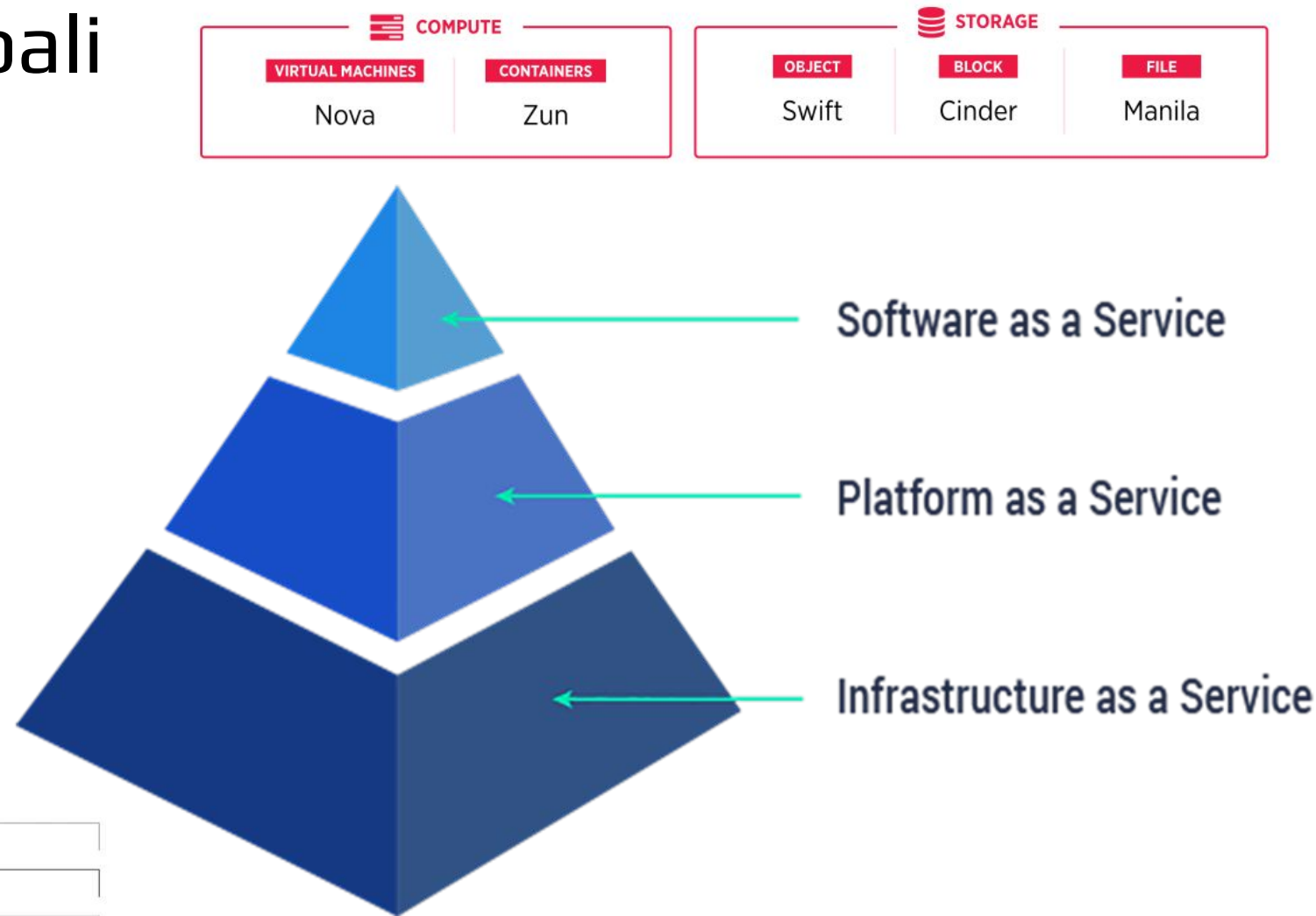
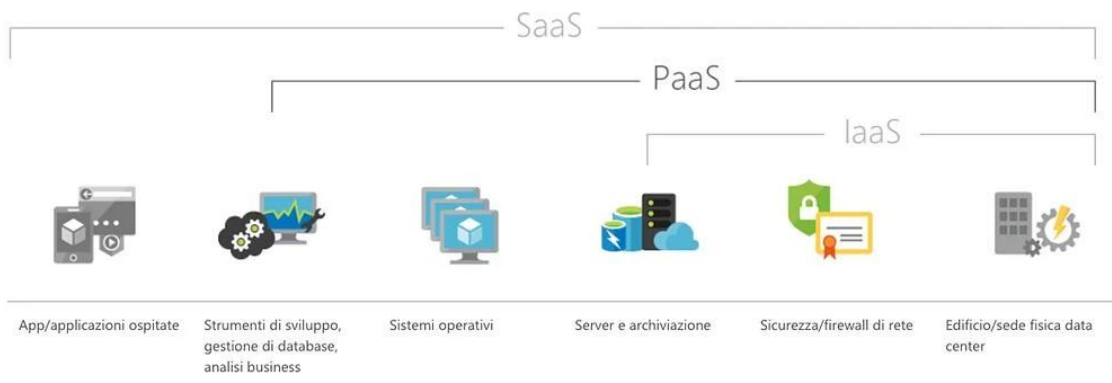


## SaaS

L'utente finale non ha bisogno di nessuna conoscenza informatica per utilizzare l'applicazione o i servizi erogati. I servizi sono utilizzabili semplicemente con una connessione internet e un browser.

# Componenti Principali

- IAM (Identity and Access Management)
- IaaS
- PaaS
- SaaS

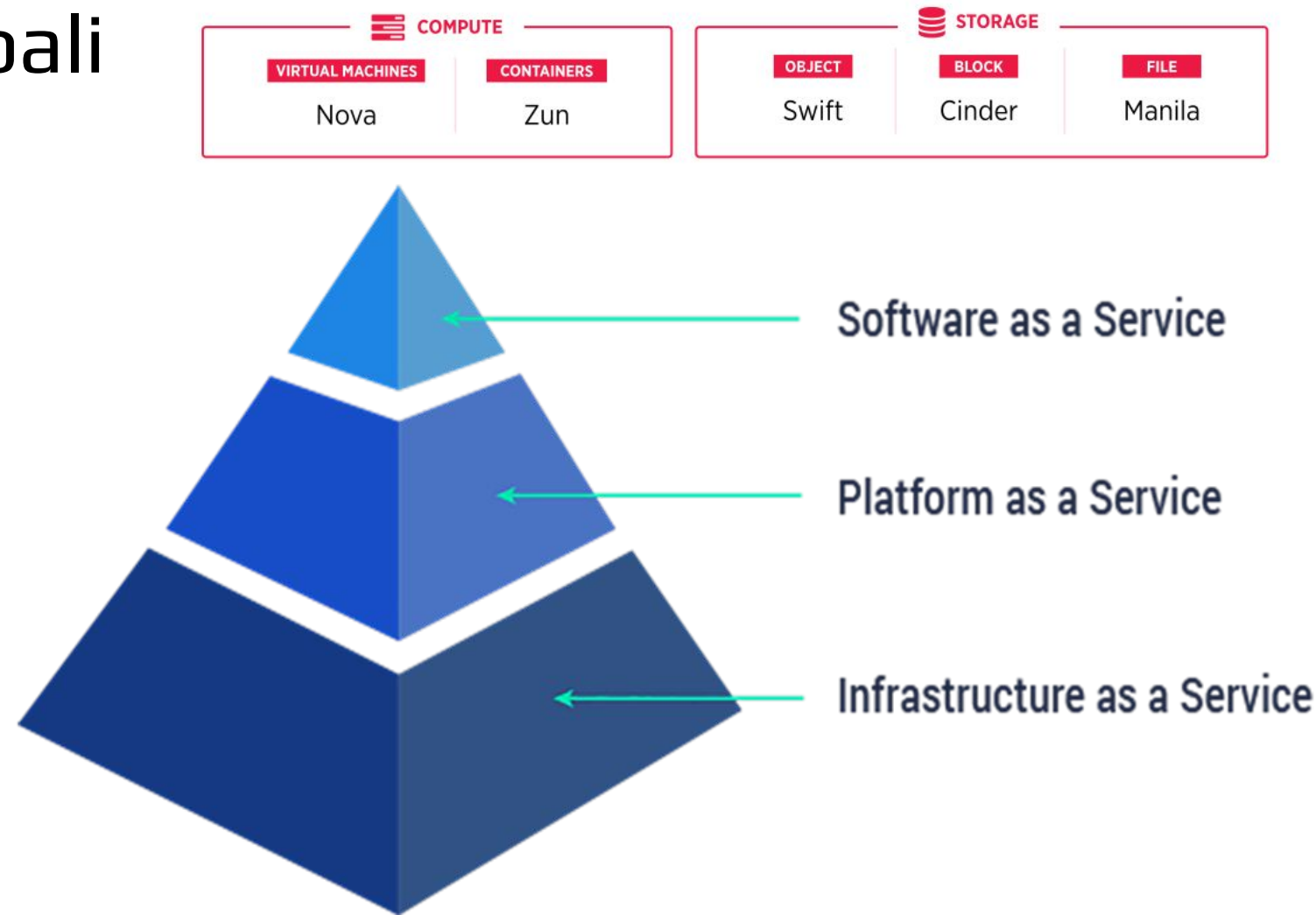




# Componenti Principali

- IAM (Identity and Access Management)
- IaaS
- PaaS
- SaaS
- DaaS

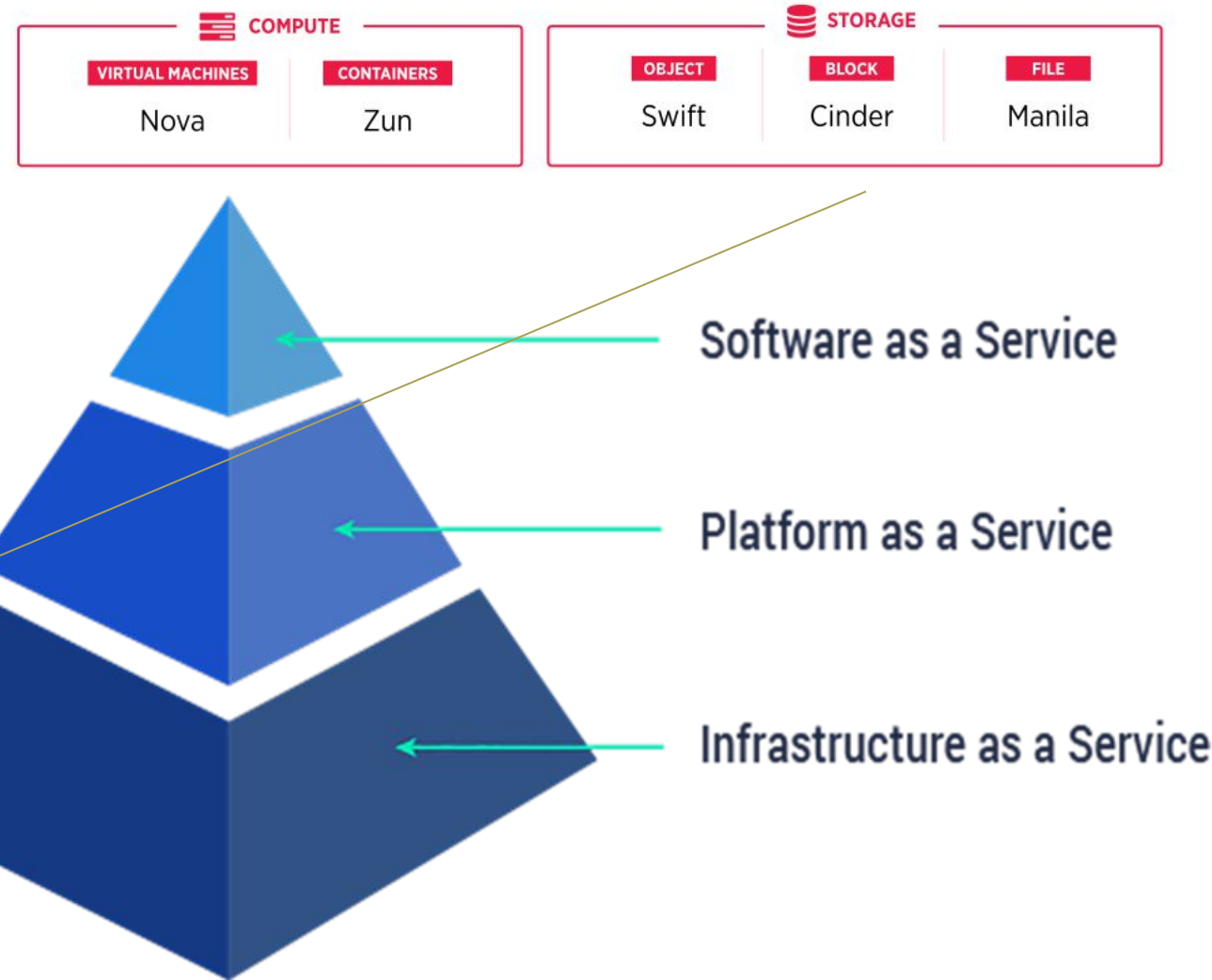
Data as a Service



# Componenti Principali

- IAM (Identity and Access Management)
- IaaS
- PaaS
- SaaS
- DaaS

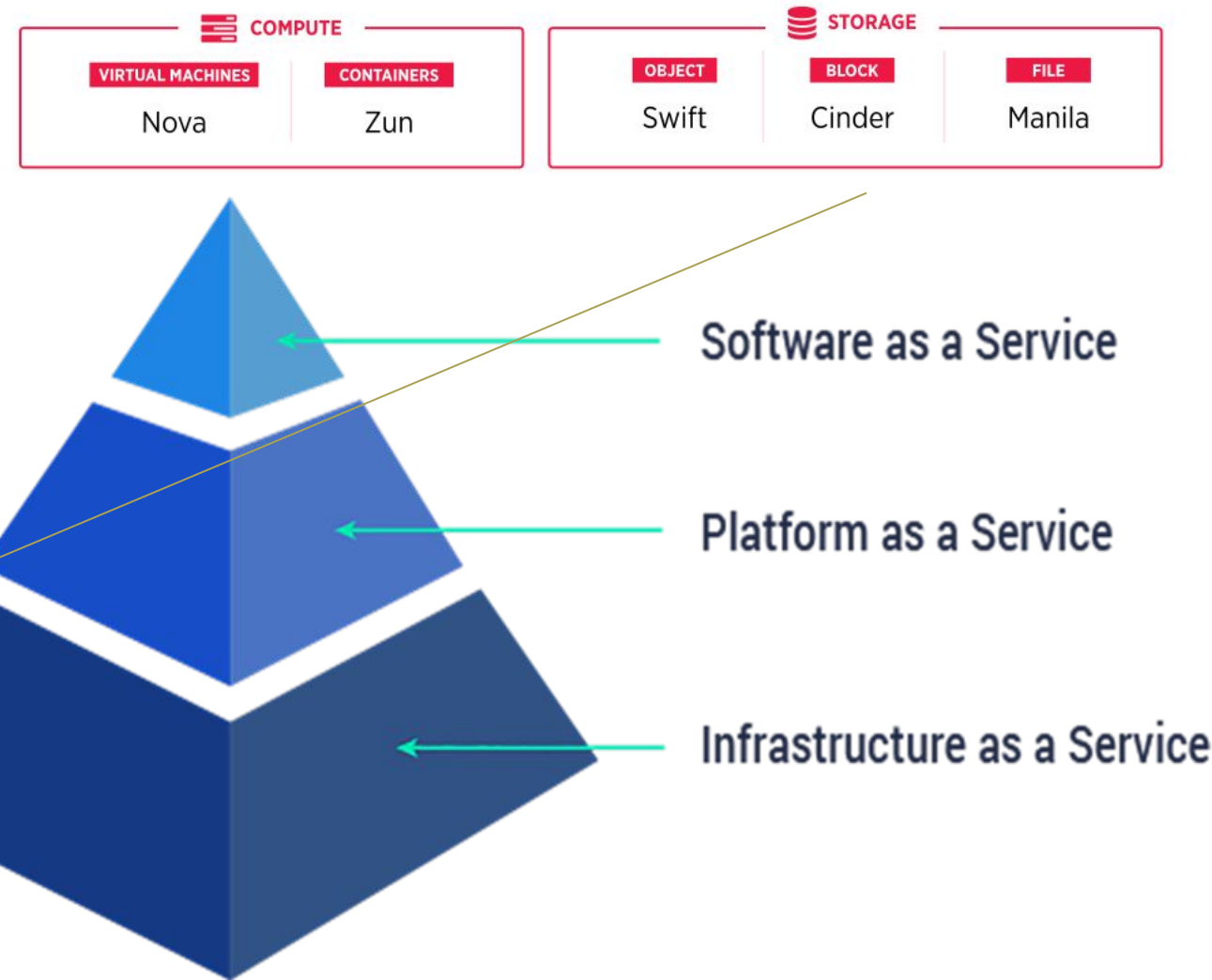
Data as a Service



# Componenti Principali

- IAM (Identity and Access Management)
- IaaS
- PaaS
- SaaS
- DaaS

Data as a Service



<https://www.bmc.com/blogs/saas-vs-paas-vs-iaas-whats-the-difference-and-how-to-choose/>



# Dati e Metadati



# Definizione di Dato

---

Un dato (dal latino datum che significa dono, cosa data) è una descrizione elementare codificata di un'informazione, un'entità, di un fenomeno, di una transazione, di un avvenimento o di altro.

Un dato (in informatica) può avere dimensione da 1 bit (booleano) sino a migliaia di byte.

# Definizione di Metadato

---

il **metadato** è, letteralmente, "(dato) per mezzo di un (altro) dato", è un'informazione che descrive un insieme di dati.

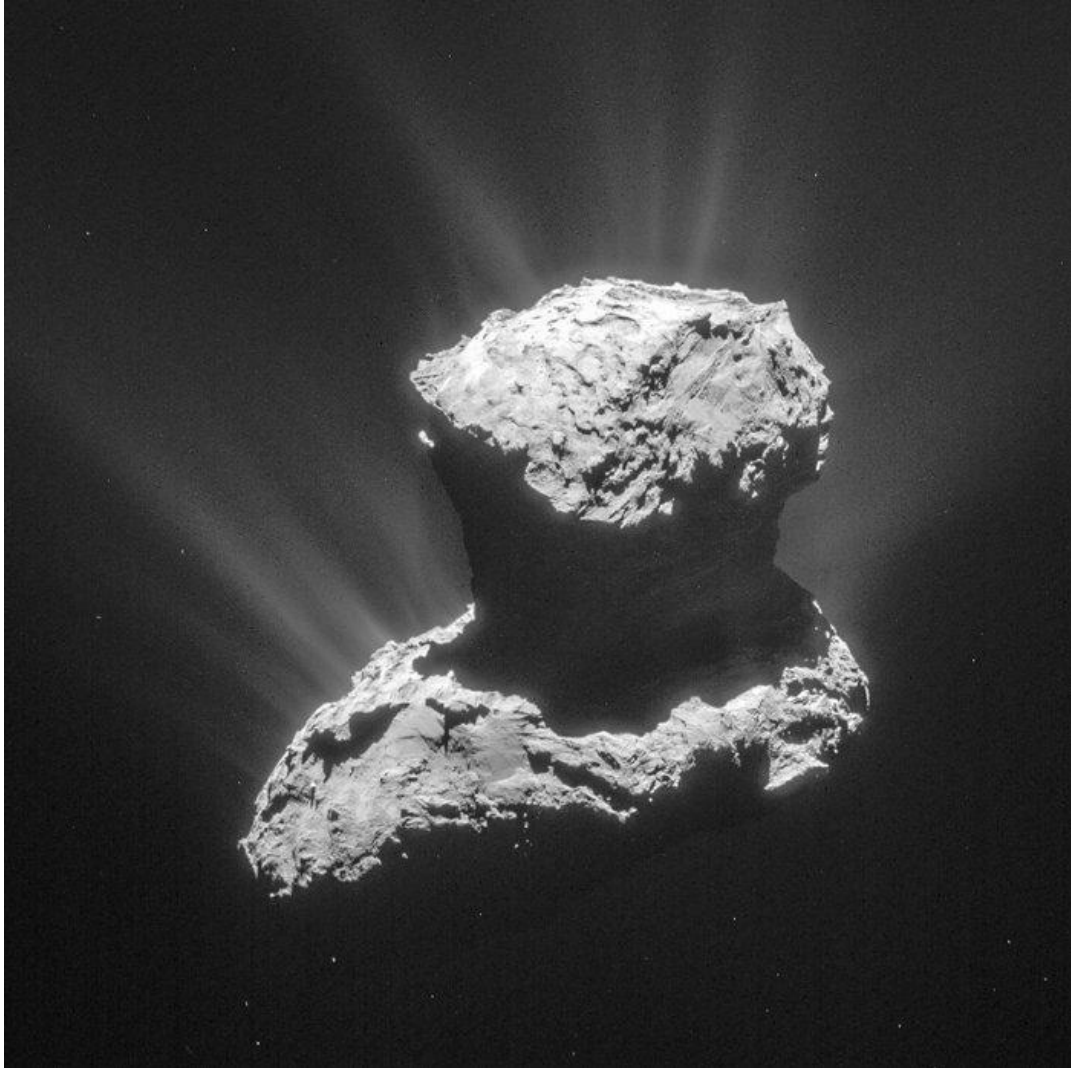
Un esempio tipico di metadati è costituito dalla scheda del catalogo di una biblioteca, la quale contiene informazioni circa il contenuto e la posizione di un libro, cioè dati riguardanti più dati che si riferiscono al libro. Un altro contenuto tipico dei metadati può essere la fonte o l'autore dell'insieme di dati descritto, oppure le modalità d'accesso con le eventuali limitazioni.

Un metadato può essere anche un dato aggiunto all'insieme delle informazioni per altri scopi. Ad esempio, se alla scheda del libro della biblioteca aggiungo un ID, ossia un identificatore univoco, quest'ultimo è un metadato.



# Dato e Metadato in Planetologia

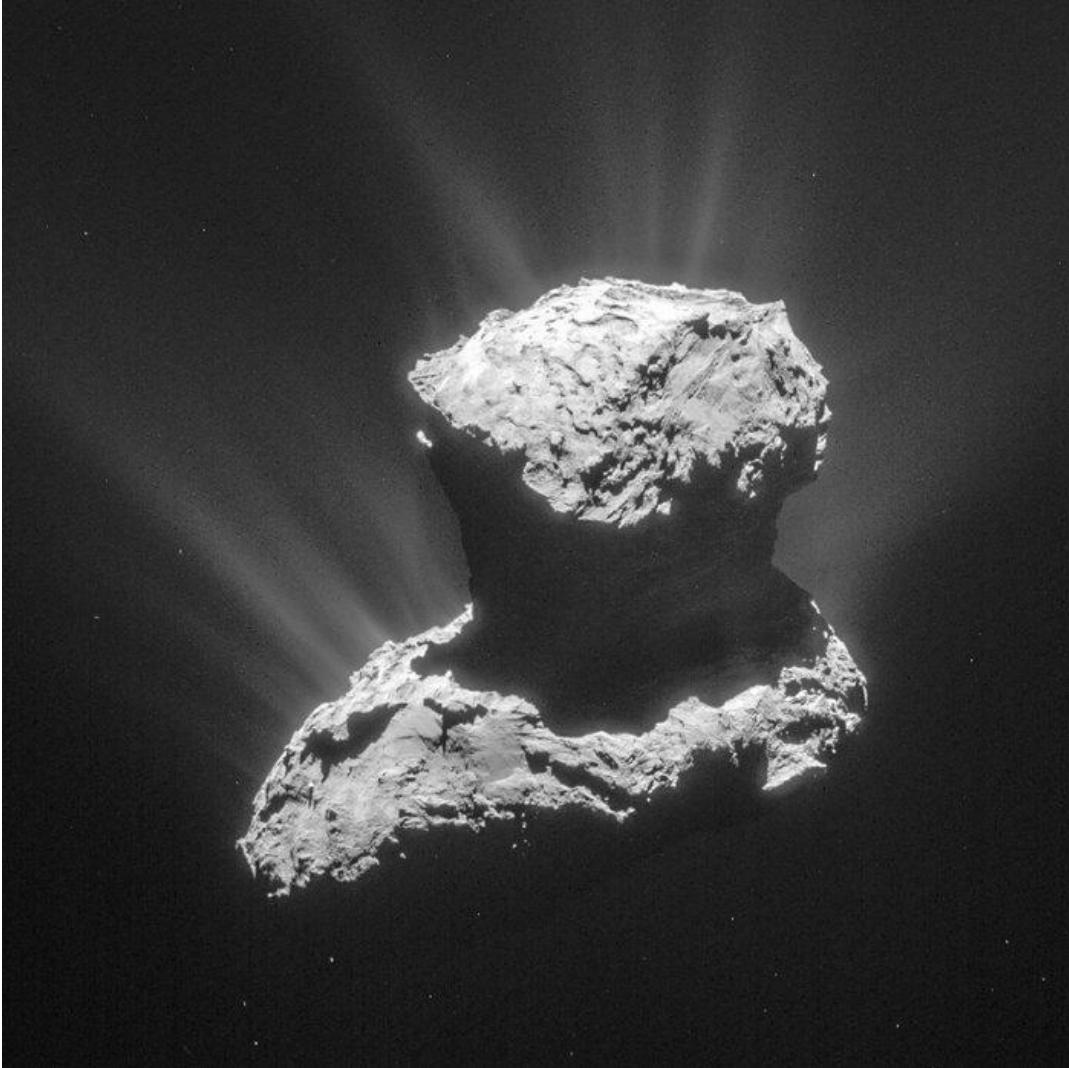
---



# Dato e Metadato in Planetologia

---

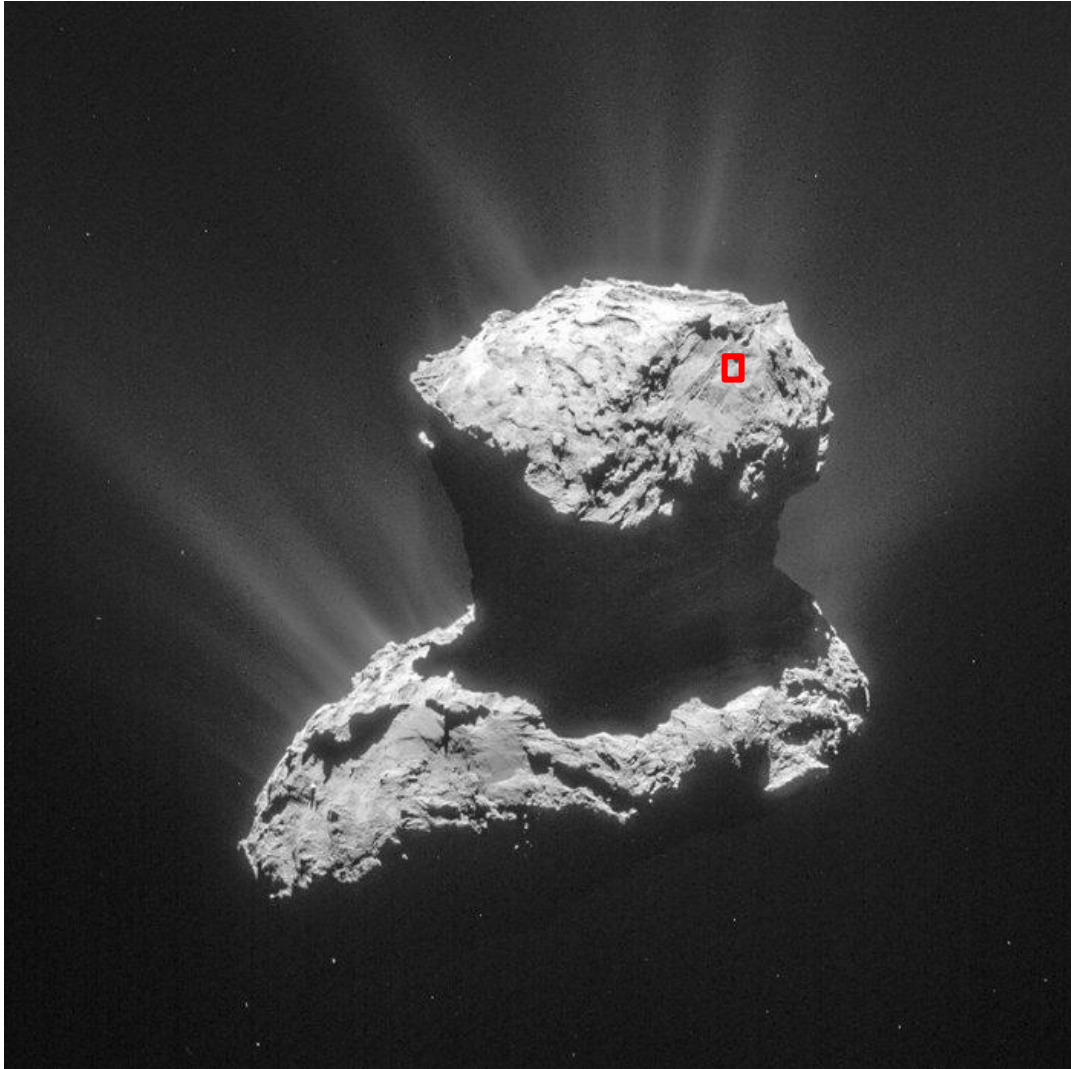
Quale è il dato in questo caso?



# Dato e Metadato in Planetologia

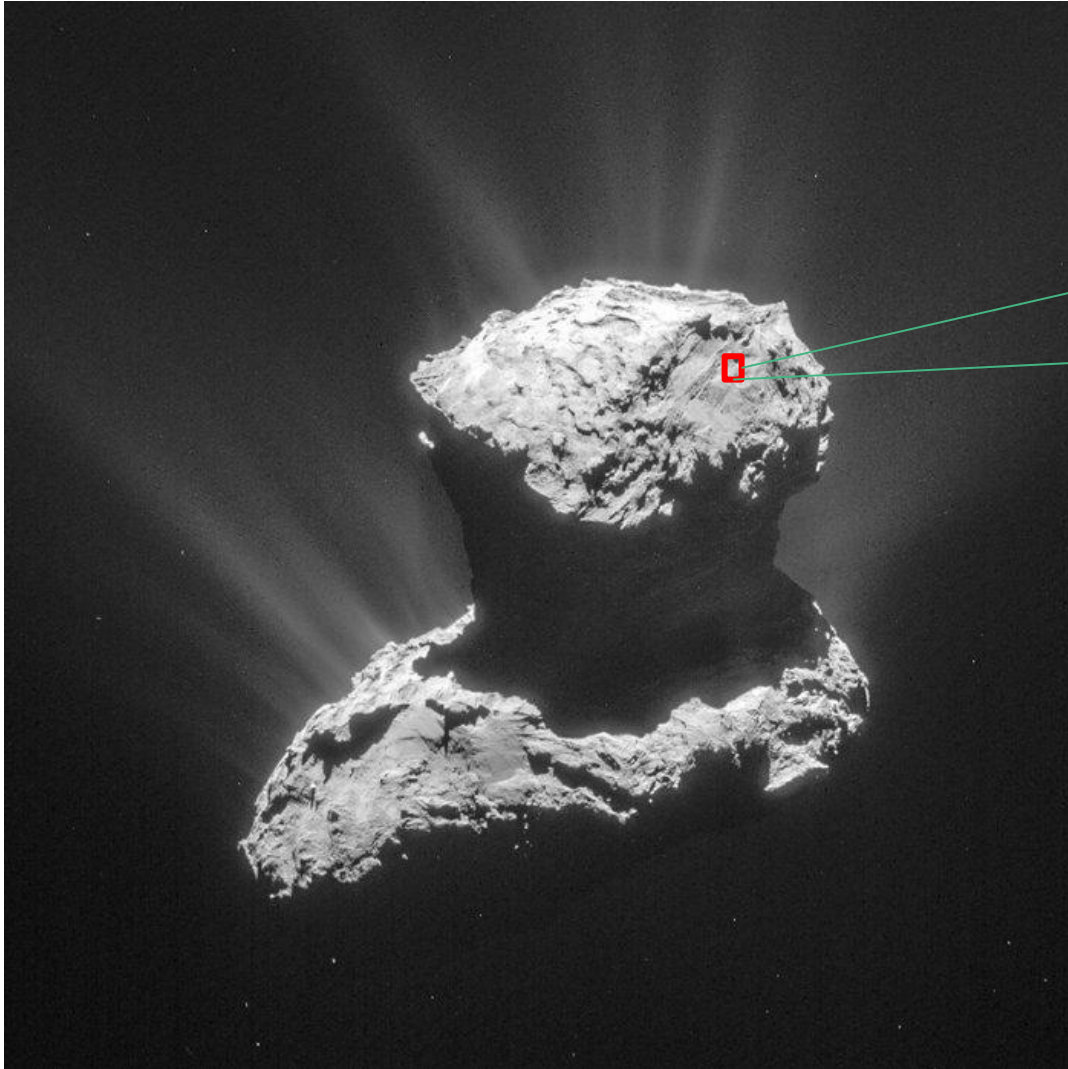
---

Quale è il dato in questo caso?



# Dato e Metadato in Planetologia

Quale è il dato in questo caso?



Il pixel è rappresentato come un numero in virgola mobile a 32 bit.

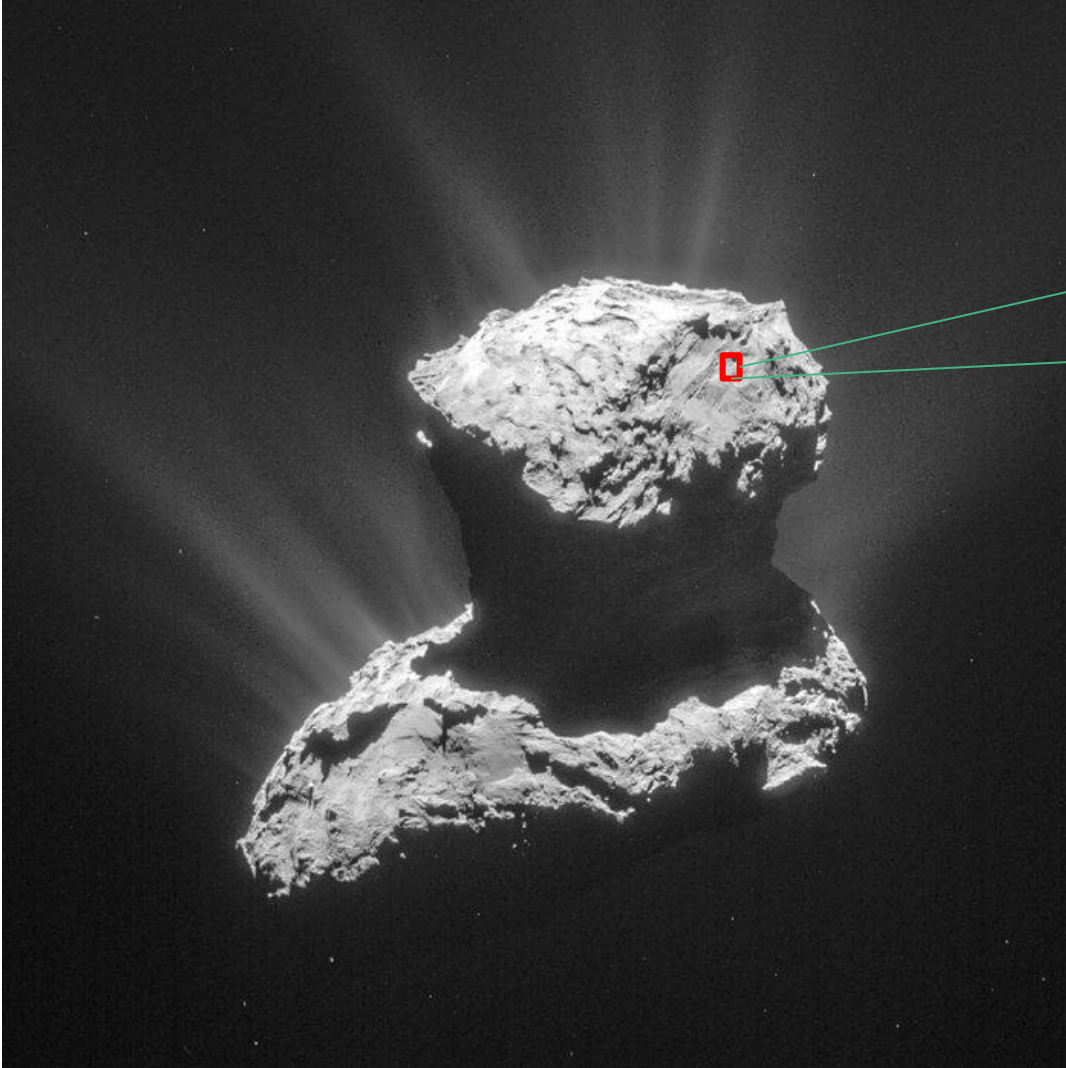


# Dato e Metadato in Planetologia

Quale è il dato in questo caso?

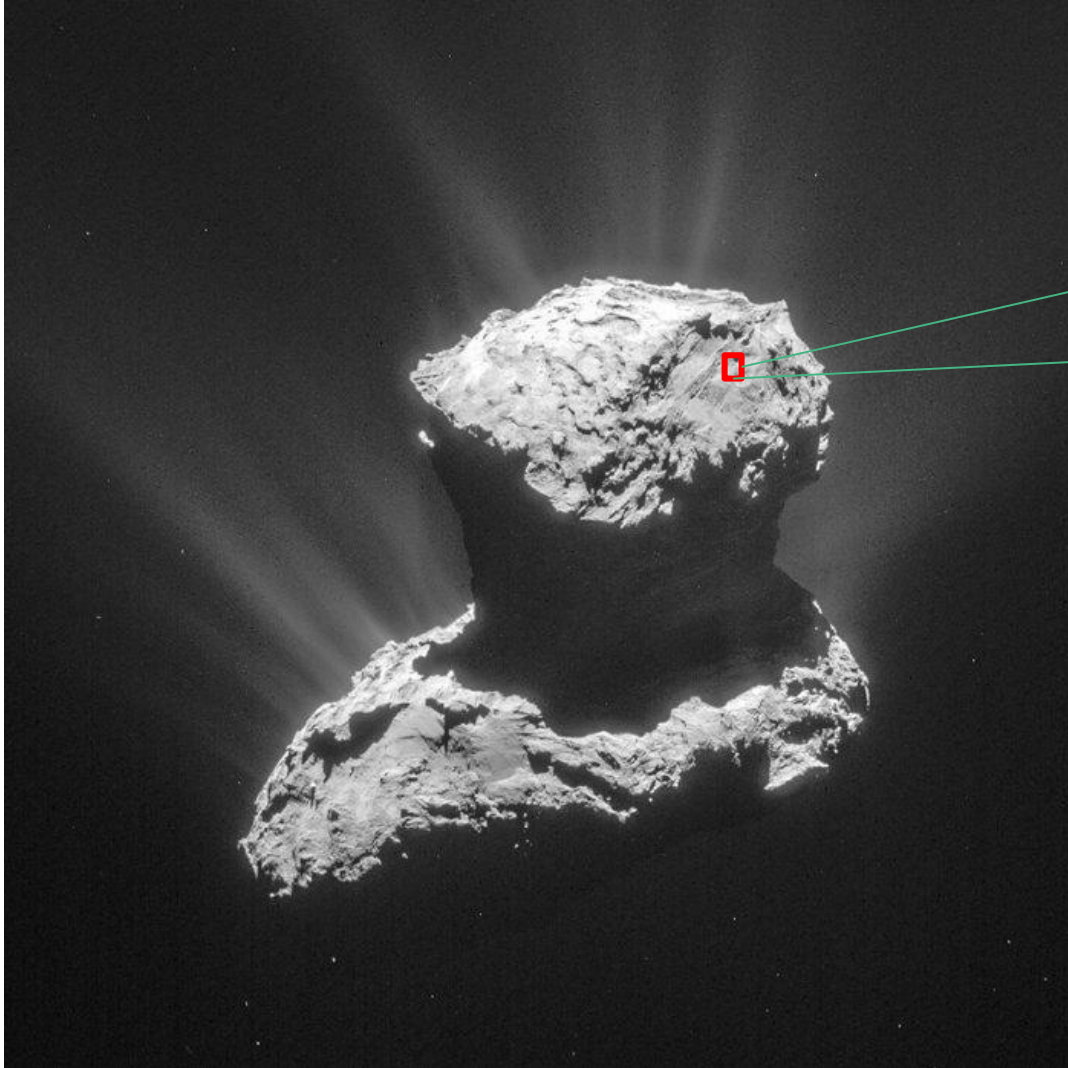
Il pixel è rappresentato come un numero in virgola mobile a 32 bit.

Ha significato?



# Dato e Metadato in Planetologia

Quale è il dato in questo caso?



Il pixel è rappresentato come un numero in virgola mobile a 32 bit.

Ha significato? **No**

# Dato e Metadato in Planetologia

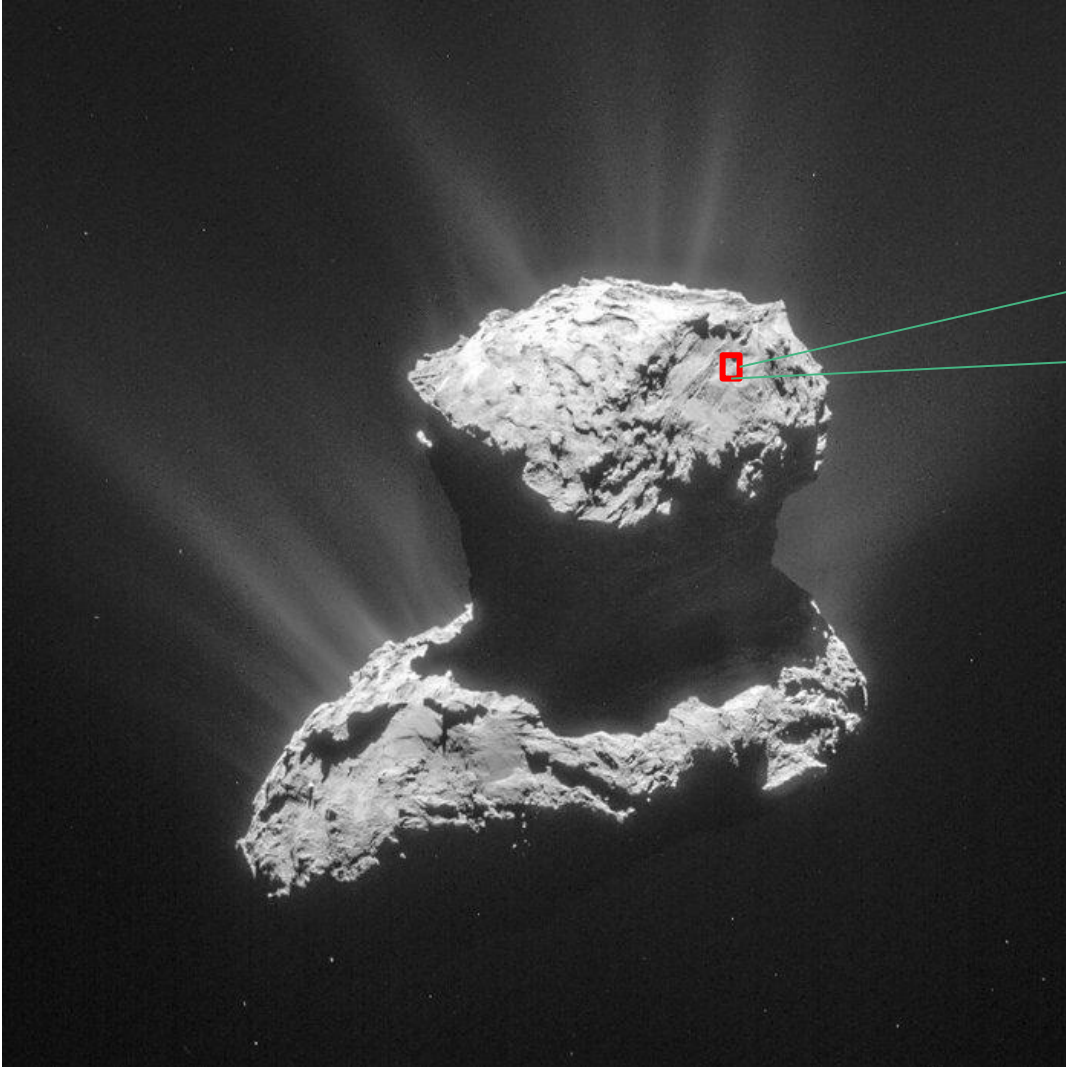
Quale è il dato in questo caso?

Il pixel è rappresentato come un numero in virgola mobile a 32 bit.

Ha significato? **No**

Si ha necessità di conoscere

- illuminazione,
- posizione della cometa,
- posizione dello spacecraft,
- tempi di esposizione,
- modalità di acquisizione,
- georeferenziazione del pixel





# Dati nel Cloud





# Dati nel Cloud

---



# Differenza tra **Object Storage** e **File Storage**

---

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una directory file gerarchica.

# Differenza tra **Object Storage** e **File Storage**

---

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una directory file gerarchica.

*Unix and Unix Like*

# Differenza tra **Object Storage** e **File Storage**

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una directory file gerarchica.

Unix and Unix Like

- Tutti i nomi dei *file* sono "Case Sensitive". Ciò vuol dire che vivek.txt Vivek.txt VIVEK.txt sono tre file differenti.
- Per i nomi di file si possono usare lettere maiuscole, minuscole ed i simboli "." (*dot*), e "\_" (*underscore*).
- Possono essere usati anche altri caratteri speciali come " " (*blank space*) ma hanno un uso complesso (devono essere quotati) e se ne sconsiglia l'uso.
- In pratica il nome di un file può contenere qualsiasi carattere escluso "/" (*root folder*) che è riservato come separatore tra file e folder nel *pathname*.
- Non può essere usato il carattere *null*.
- L'uso del "." non è necessario ma aumenta la leggibilità specialmente se usato per identificare l'estensione.
- Il nome del file è unico all'interno di un folder.
- In un folder non possono coesistere un folder ed un file con lo stesso nome.

- Nome
- Percorso (path)
- Tipo
- Dimensione
- Proprietario (UID, GID)
- Permessi
- Marcature Temporali
  - creazione
  - modifica

# Differenza tra **Object Storage** e **File Storage**

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una directory file gerarchica.

Unix and Unix Like

Max 255 caratteri

- Tutti i nomi dei *file* sono "Case Sensitive". Ciò vuol dire che vivek.txt Vivek.txt VIVEK.txt sono tre file differenti.
- Per i nomi di file si possono usare lettere maiuscole, minuscole ed i simboli "." (*dot*), e "\_" (*underscore*).
- Possono essere usati anche altri caratteri speciali come " " (*blank space*) ma hanno un uso complesso (devono essere quotati) e se ne sconsiglia l'uso.
- In pratica il nome di un file può contenere qualsiasi carattere escluso "/" (*root folder*) che è riservato come separatore tra file e folder nel *pathname*.
- Non può essere usato il carattere *null*.
- L'uso del "." non è necessario ma aumenta la leggibilità specialmente se usato per identificare l'estensione.
- Il nome del file è unico all'interno di un folder.
- In un folder non possono coesistere un folder ed un file con lo stesso nome.

- Nome
- Percorso (path)
- Tipo
- Dimensione
- Proprietario (UID, GID)
- Permessi
- Marcature Temporali
  - creazione
  - modifica

# Differenza tra **Object Storage** e **File Storage**

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una directory file gerarchica.

*Unix and Unix Like*

Il set di nomi richiesto per specificare un particolare file in una gerarchia di folder è detto percorso del file o *path*.

percorso e nome del file formano il cosiddetto *pathname*.

Il percorso può essere assoluto o relativo:

- nel path assoluto si specifica tutto il percorso dall'inizio del disco (/ ,root):  
`/u/politi/projectb/plans/1dft`
- nel path relativo si può indicare il percorso a partire dal folder in cui ci si trova.  
`projectb/plans/1dft`

Un path relativo non può iniziare con /.

Simboli speciali:

- `.` indica il folder corrente
- `..` indica il folder di livello superiore

- Nome
- **Percorso (path)**
- Tipo
- Dimensione
- Proprietario (UID, GID)
- Permessi
- Marcature Temporali
  - creazione
  - modifica

# Differenza tra **Object Storage** e **File Storage**

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una directory file gerarchica.

Unix and Unix Like

Max 1024 caratteri

I set di nomi richiesto per specificare un particolare file in una gerarchia di folder è detto percorso del file o *path*.

percorso e nome del file formano il cosiddetto *pathname*.

Il percorso può essere assoluto o relativo:

- nel path assoluto si specifica tutto il percorso dall'inizio del disco (/ ,root):  
/u/politi/projectb/plans/1dft
- nel path relativo si può indicare il percorso a partire dal folder in cui ci si trova.  
projectb/plans/1dft

Un path relativo non può iniziare con /.

Simboli speciali:

- . indica il folder corrente
- .. indica il folder di livello superiore

- Nome
- Percorso (path)
- Tipo
- Dimensione
- Proprietario (UID, GID)
- Permessi
- Marcature Temporali
  - creazione
  - modifica

# Differenza tra **Object Storage** e **File Storage**

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una directory file gerarchica.

*Unix and Unix Like*

Il tipo di file viene identificato dal primo carattere della stringa dei permessi.

```
-rwxrwxrwx 1 romolo romolo      658 apr 30 09:56 manage.py
```

i tipi possono essere:

- file regolare
- d** directory
- l** symbolic link
- c** Character file device
- b** block device
- s** local socket
- p** named pipe

- Nome
- Percorso (path)
- **Tipo**
- Dimensione
- Proprietario (UID, GID)
- Permessi
- Marcature Temporali
  - creazione
  - modifica



# Differenza tra Object Storage e File Storage

Il **file storage** è il formato di storage  
maggiormente conosciuto: i dati

Univ

```
-rwxrwxr-- 10 root root 2048 Jan 13 07:11 afile.exe
?UUUGGGG00S 00 UUUUUU GGGGGG ##### ^-- date stamp and file name are obvious ;-)
```

^ ^ ^ ^ ^  
| | | | |  
| | | | | \--- File Size  
| | | | | \----- Group Name (for example, Users, Administrators, etc)  
| | | | | \----- Owner Acct  
| | | | | \----- Link count (what constitutes a "link" here varies)  
| | | | | \----- Alternative Access (blank means none defined, anything else varies)  
| \--\--\-- \----- Read, Write and Special access modes for [U]ser, [G]roup, and [O]thers (everyone else)  
| \----- File type flag

```
-rwxrwxrwx 1 romolo romolo 658 apr 30 09:56 manage.py
```

i tipi possono essere:

- file regolare
- d** directory
- l** symbolic link
- c** Character file device
- b** block device
- s** local socket
- p** named pipe

- Percorso (path)
- **Tipo**
- Dimensione
- Proprietario (UID, GID)
- Permessi
- Marcature Temporali
  - creazione
  - modifica

\_\_\_\_\_

Univ

•

rwyrwyrwyr	1	remote	remote	650	apr 20	00:56	manage.py
------------	---	--------	--------	-----	--------	-------	-----------

	Character	Effect on files	Effect on directories
Read permission (first character)	-	The file cannot be read.	The directory's contents cannot be shown.
	r	The file can be read.	The directory's contents can be shown.
Write permission (second character)	-	The file cannot be modified.	The directory's contents cannot be modified.
	w	The file can be modified.	The directory's contents can be modified (create new files or folders; rename or delete existing files or folders); requires the execute permission to be also set, otherwise this permission has no effect.
Execute permission (third character)	-	The file cannot be executed.	The directory cannot be accessed with <code>cd</code> .
	x	The file can be executed.	The directory can be accessed with <code>cd</code> ; this is the only permission bit that in practice can be considered to be "inherited" from the ancestor directories, in fact if <i>any</i> folder in the path does not have the <code>x</code> bit set, the final file or folder cannot be accessed either, regardless of its permissions; see <a href="#">path_resolution(7)</a> for more information.
	s	The <code>setuid</code> bit when found in the <b>user</b> triad; the <code>setgid</code> bit when found in the <b>group</b> triad; it is not found in the <b>others</b> triad; it also implies that <code>x</code> is set.	
	S	Same as <code>s</code> , but <code>x</code> is not set; rare on regular files, and useless on folders.	
	t	The sticky bit; it can only be found in the <b>others</b> triad; it also implies that <code>x</code> is set.	
	T	Same as <code>t</code> , but <code>x</code> is not set; rare on regular files, and useless on folders.	

\_\_\_\_\_

Univ

- Percorso (path)

- **Tipo**

- Dimensione
- Proprietario (UID, GID)
- Permessi
- Marcature Temporali
  - creazione
  - modifica

# Differenza tra **Object Storage** e **File Storage**

---

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una directory file gerarchica.

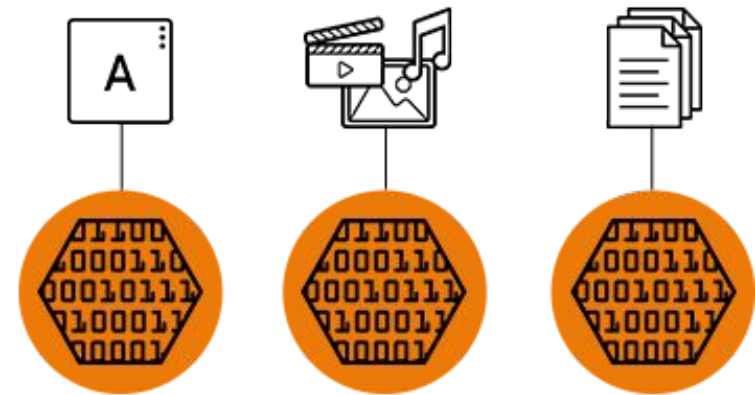
L'**object storage** è un formato di storage in cui i dati sono archiviati in unità separate chiamate oggetti. Ciascuna unità ha un identificatore univoco, o chiave, che ne permette l'individuazione indipendentemente dalla posizione in cui sono memorizzate in un sistema distribuito.

# Differenza tra **Object Storage** e **File Storage**

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una directory file gerarchica.

L'**object storage** è un formato di storage in cui i dati sono archiviati in unità separate chiamate oggetti. Ciascuna unità ha un identificatore univoco, o chiave, che ne permette l'individuazione indipendentemente dalla posizione in cui sono memorizzate in un sistema distribuito.

Nello storage di oggetti, i dati vengono frammentati in unità discrete chiamate appunto oggetti e conservati in un unico repository invece che come file all'interno di cartelle o come blocchi su server.





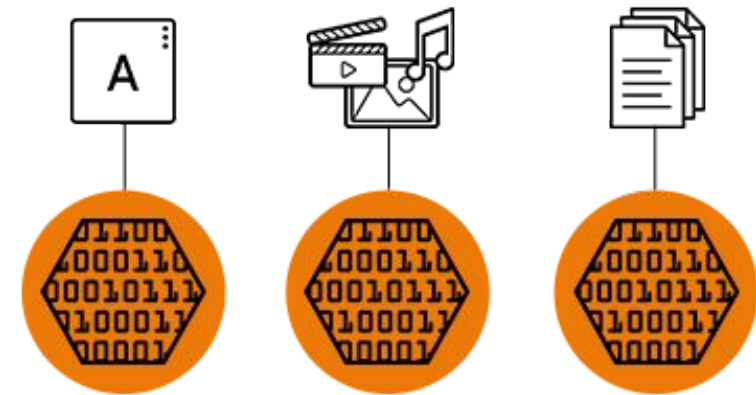
# Differenza tra **Object Storage** e **File Storage**

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una directory file gerarchica.

L'**object storage** è un formato di storage in cui i dati sono archiviati in unità separate chiamate oggetti. Ciascuna unità ha un identificatore univoco, o chiave, che ne permette l'individuazione indipendentemente dalla posizione in cui sono memorizzate in un sistema distribuito.

Nello storage di oggetti, i dati vengono frammentati in unità discrete chiamate appunto oggetti e conservati in un unico repository invece che come file all'interno di cartelle o come blocchi su server.

I volumi dello storage di oggetti operano come unità modulari: ognuno è un repository indipendente che conserva al suo interno i dati, un identificativo univoco che permette di individuare un oggetto in un sistema distribuito e i metadati che descrivono i dati. I metadati sono importanti e includono dettagli come l'età, privacy/sicurezza e limitazioni all'accesso.



# Differenza tra **Object Storage** e **File Storage**

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una **directory file** gerarchica.

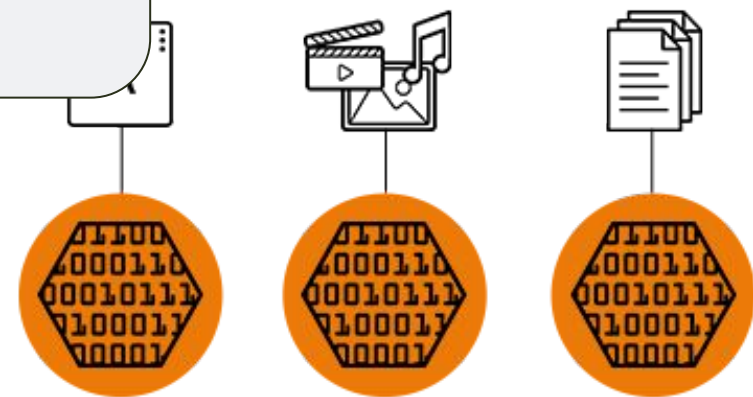
Nello storage di file i dati sono archiviati in unità discrete chiamate file, organizzati in repository invece che come file all'interno di cartelle o come blocchi su server.

I volumi dello storage di oggetti operano come unità modulari: ognuno è un repository indipendente che conserva al suo interno i dati, un identificativo univoco che permette di individuare un oggetto in un sistema distribuito e i metadati che descrivono i dati. I metadati sono importanti e includono dettagli come l'età, privacy/sicurezza e limitazioni all'accesso.

L'**object storage** è un formato di storage in cui i dati sono archiviati in unità separate chiamate oggetti. Ciascuna unità ha un identificatore univoco, o chiave, che ne permette l'individuazione.

La posizione in cui si trova l'oggetto è memorizzata nel sistema distribuito.

I **metadati** dello storage di oggetti possono essere estremamente dettagliati e capaci di archiviare informazioni sul luogo in cui un video è stato girato, sul tipo di fotocamera che è stato utilizzato e sugli attori che compaiono in ogni fotogramma.



# Differenza tra **Object Storage** e **File Storage**

---

Il **file storage** è il formato di storage maggiormente conosciuto: i dati vengono archiviati in file con cui è possibile interagire, contenuti in cartelle all'interno di una directory file gerarchica.

L'**object storage** è un formato di storage in cui i dati sono archiviati in unità separate chiamate oggetti. Ciascuna unità ha un identificatore univoco, o chiave, che ne permette l'individuazione indipendentemente dalla posizione in cui sono memorizzate in un sistema distribuito.

Il **block storage** suddivide i dati in componenti separati composti da blocchi di dati di dimensioni fisse, ognuno dotato di un identificatore univoco. Il block storage permette al sistema di storage sottostante di recuperarlo indipendentemente dalla posizione in cui viene memorizzato.





# Calcolo nel Cloud



# Calcolo nel Cloud

---

## PaaS

Ho un ambiente di lavoro del tutto simile al pc ma con risorse molto più elevate.

### **Schema di sviluppo**

Sviluppo in locale con dataset limitati.

Trasferimento su cloud dei codici/dati

Esecuzione su larga scala.

(Elevato numero di CPU/GPU, RAM, FLOPs, etc)

# Calcolo nel Cloud

---

## PaaS

Ho un ambiente di lavoro del tutto simile al pc ma con risorse molto più elevate.

### **Schema di sviluppo**

Sviluppo in locale con dataset limitati.

Trasferimento su cloud dei codici/dati

Esecuzione su larga scala.

(Elevato numero di CPU/GPU, RAM, FLOPs, etc)

## SaaS

Sviluppo un codice che come GUI, solitamente, una pagina web.

### **Schema di sviluppo**

Sviluppo in locale con opzioni di debug attive

Trasferimento su server e messa online.

# Calcolo nel Cloud

---

## PaaS

Ho un ambiente di lavoro del tutto simile al pc ma con risorse molto più elevate.

### Schema di sviluppo

Sviluppo in locale con dataset limitati.

Trasferimento su cloud dei codici/dati

Esecuzione su larga scala.

(Elevato numero di CPU/GPU, RAM, FLOPs, etc)

## SaaS

Sviluppo un codice che come GUI, solitamente, una pagina web.

### Schema di sviluppo

Sviluppo in locale con opzioni di **debug attive**

Trasferimento su server e messa online.

# Calcolo nel Cloud

## PaaS

Ho un ambiente di lavoro del tutto simile al pc ma con risorse molto più elevate.

### Schema di sviluppo

Sviluppo in locale con dataset limitati.

Trasferimento su cloud dei codici/dati

Esecuzione su larga scala.

## SaaS

Sviluppo un codice che come GUI, solitamente, una pagina web.

### Schema di sviluppo

Sviluppo in locale con opzioni di **debug attive**

Trasferimento su server e messa online.

Le opzioni di debug **non devono mai essere attive** sul server di produzione, questo perchè il debug, per sua natura, apre alla possibilità di Hacking/Cracking del codice e della macchina

# Sistemi di archiviazione

---

## Cosa

- archiviazione di file (file storage)
- archiviazione di oggetti (object storage/database)

# Sistemi di archiviazione

---

## Cosa

- archiviazione di file (file storage)
- archiviazione di oggetti (object storage/database)

## Come

- archiviazione gerarchica/relazionale
- archiviazione piana

# Sistemi di archiviazione

---

## Cosa

- archiviazione di file (file storage)
- archiviazione di oggetti (object storage/database)

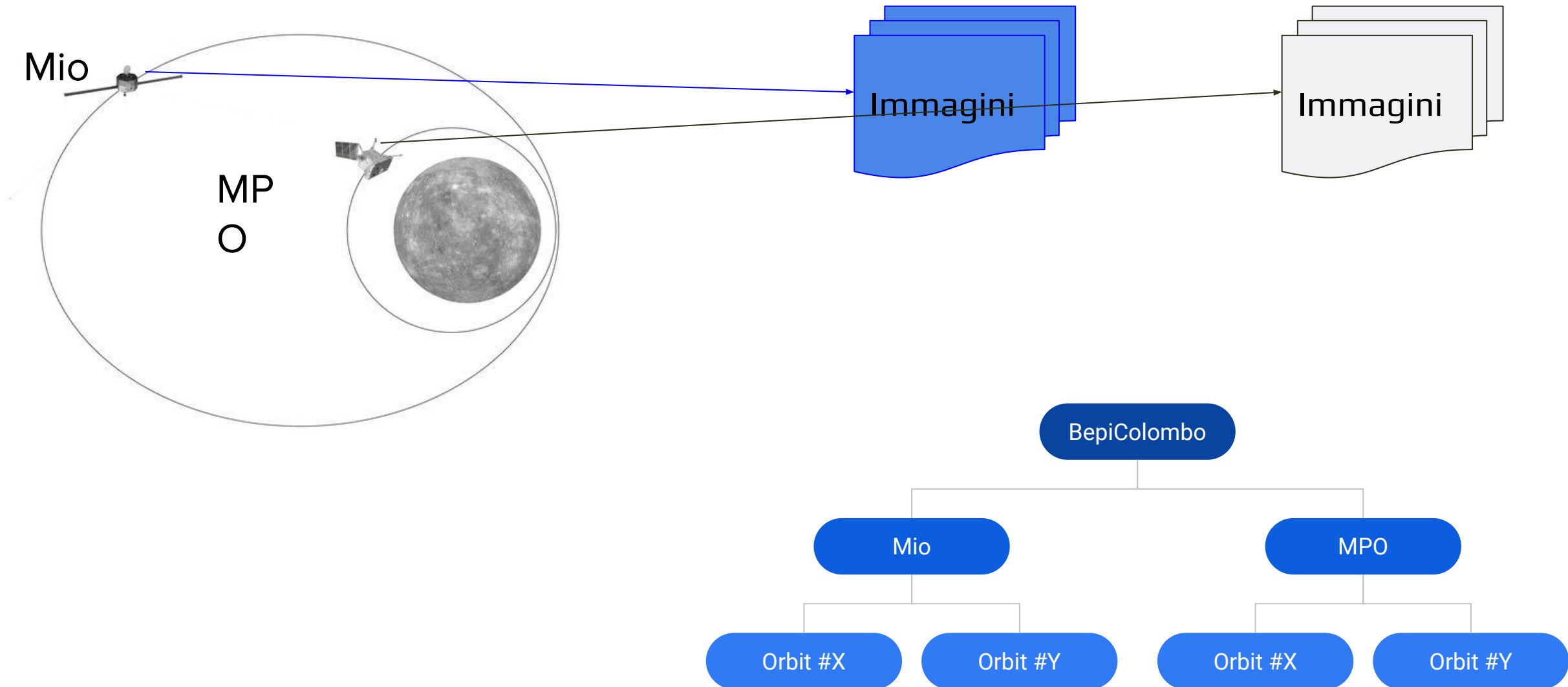
## Come

- archiviazione gerarchica/relazionale
- archiviazione piana

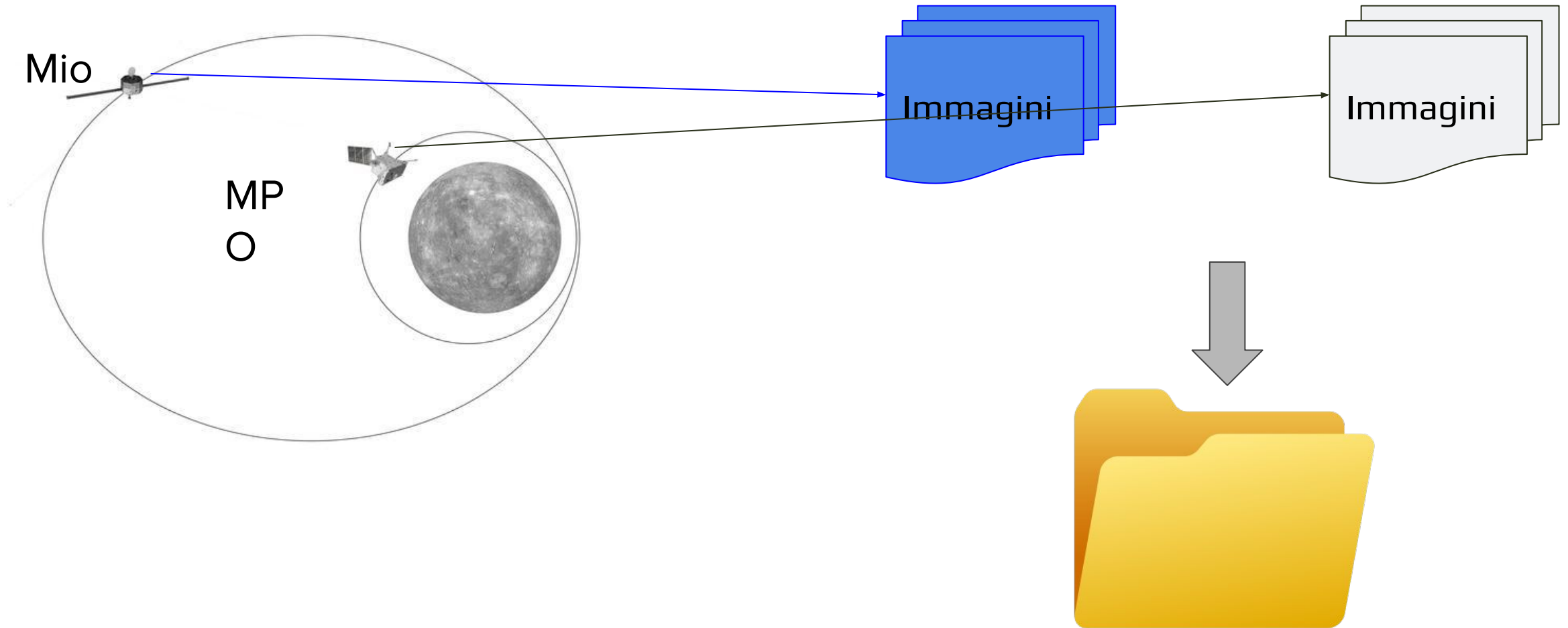
## Archiviazione Ibrida



# Esempio Archiviazione gerarchica



# Esempio Archiviazione gerarchica



# Quali dei due è più efficiente?

---

La risposta è...

# Quali dei due è più efficiente?

---

La risposta è... un'altra domanda

# Quali dei due è più efficiente?

---

La risposta è... un'altra domanda

Chi deve “vedere l'archivio”?

# Quali dei due è più efficiente?

---

La risposta è... un'altra domanda

Chi deve “vedere l'archivio”?

**Macchina**

Archiviazione piana

# Quali dei due è più efficiente?

---

La risposta è... un'altra domanda

Chi deve “vedere l'archivio”?

**Macchina**

Archiviazione piana

**Uomo**

Archiviazione gerarchica

# Concetto di Data Preservation

---

Nella gestione dei dati (*Data Management*), il ***Data Preservation*** è il processo di che permette di mantenere l'accesso ai dati, in modo che questi possano essere trovati, compresi ed utilizzati in futuro.

- short-term
- medium-term
- long-term



# Concetto di Data Preservation

---

Nella gestione dei dati (*Data Management*), il **Data Preservation** è il processo di che permette di mantenere l'accesso ai dati, in modo che questi possano essere trovati, compresi ed utilizzati in futuro.

- short-term
- medium-term
- long-term

# Concetto di Data Preservation

---

Nella gestione dei dati (*Data Management*), il **Data Preservation** è il processo di che permette di mantenere l'accesso ai dati, in modo che questi possano essere trovati, compresi ed utilizzati in futuro.

- short-term
- medium-term
- long-term

## Conservazione a breve termine.

Accesso ai materiali digitali per un periodo di tempo definito durante il quale è previsto l'uso ma che non si estende oltre il prevedibile futuro e/o fino a quando non diventa inaccessibile a causa dei cambiamenti tecnologici.

# Concetto di Data Preservation

---

Nella gestione dei dati (*Data Management*), il **Data Preservation** è il processo di che permette di mantenere l'accesso ai dati, in modo che questi possano essere trovati, compresi ed utilizzati in futuro.

- short-term
- medium-term
- long-term

## Conservazione a lungo termine

Accesso continuo ai materiali digitali, o almeno alle informazioni in essi contenute, a tempo indeterminato.

# Archiviazione Ibrida

---

## **Long Term Preservation**

- File Storage
- Archiviazione Gerarchica

## **STP**

- Object Storage
- Archiviazione Piana
- Database Relazionale

---

# Database Relazionale

---

# Cos'è un Database relazionale

---

Il termine **relational database management system** (**RDBMS**, sistema per la gestione di [basi di dati relazionali](#)) indica un *database management system* basato sul [modello relazionale](#), introdotto da [Edgar F. Codd](#).

Oltre a questi, anche se meno diffusi a livello commerciale, altri sistemi di gestione di basi di dati che implementano [modelli dei dati](#) alternativi a quello relazionale: [gerarchico](#), [reticolare](#) e [a oggetti](#).

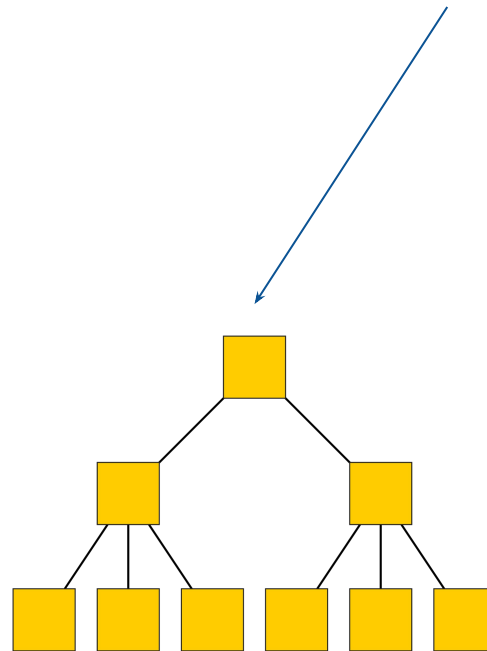
([Wikipedia](#))

# Cos'è un Database relazionale

Il termine **relational database management system** (**RDBMS**, sistema per la gestione di [basi di dati relazionali](#)) indica un [database management system](#) basato sul [modello relazionale](#), introdotto da [Edgar F. Codd](#).

Oltre a questi, anche se meno diffusi a livello commerciale, altri sistemi di gestione di basi di dati che implementano [modelli dei dati](#) alternativi a quello relazionale: [gerarchico](#), [reticolare](#) e [a oggetti](#).

([Wikipedia](#))

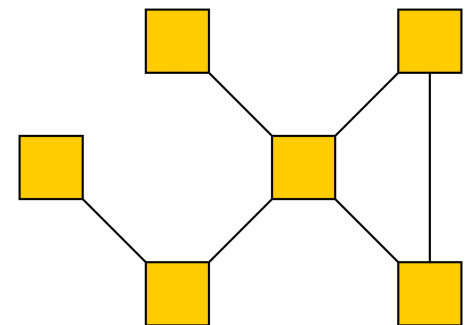
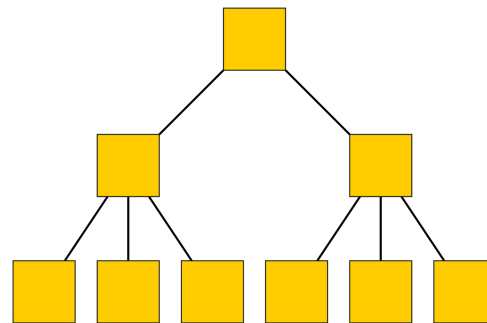


# Cos'è un Database relazionale

Il termine **relational database management system** (**RDBMS**, sistema per la gestione di **basi di dati relazionali**) indica un *database management system* basato sul **modello relazionale**, introdotto da **Edgar F. Codd**.

Oltre a questi, anche se meno diffusi a livello commerciale, altri sistemi di gestione di basi di dati che implementano **modelli dei dati** alternativi a quello relazionale: **gerarchico**, **reticolare** e **a oggetti**.

([Wikipedia](#))





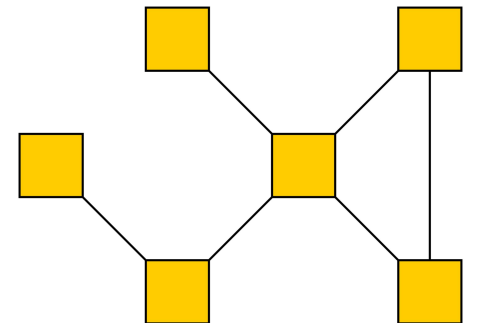
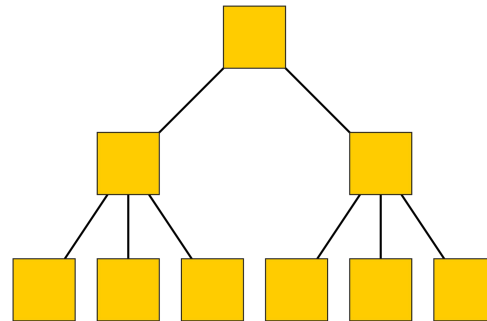
# Cos'è un Database relazionale

Il termine **relational database management system** (**RDBMS**, sistema per la gestione di **basi di dati relazionali**) indica un *database management system* basato sul **modello relazionale**, introdotto da **Edgar F. Codd**.

Oltre a questi, anche se meno diffusi a livello commerciale, altri sistemi di gestione di basi di dati che implementano **modelli dei dati** alternativi a quello relazionale: **gerarchico**, **reticolare** e **a oggetti**.

([Wikipedia](#))

Tra i vari database di relazionali e DB ad oggetti PostgreSQL è quello con la più elevata diffusione



# Cos'è un Database relazionale

Il termine **relational database management system (RDBMS)**, sistema per la gestione di **basi di dati relazionali**) indica un *database management system* basato sul **modello relazionale**, introdotto da **Edgar F. Codd**.

Oltre a questi, anche se meno diffusi a livello commerciale, altri sistemi di gestione di basi di dati che implementano **modelli dei dati** alternativi a quello relazionale: **gerarchico**, **reticolare** e **a oggetti**.

([Wikipedia](#))

Tra i vari database di relazionali e D ad oggetti PostgreSQL è quello con la più elevata diffusione

## Nota per programmatore:

in Python esistono diverse librerie che permettono l'accesso ai dati di un DB. Tra queste alcune convertono il db relazionale in classi python

