Sapere utile

IFOA
Istituto Formazione Operatori Aziendali

BIG DATA e Analisi dei Dati

Lezione 1.1 - Introduzione

Mauro Bellone,
Robotics and AI researcher

bellonemauro@gmail.com
www.maurobellone.com

Obiettivo

- ✓ Fornire una presentazione del Corso e dei suoi contenuti
- ✓ Fornire una introduzione alle tecnologie big data, quando, come e perchè si utilizzano o implementano
- ✓ Introduzione ai data base relazionali e non relazionali

About me:

Mauro Bellone,
Robotics and AI researcher

bellonemauro@gmail.com
www.maurobellone.com

TAL TECH

ECHOLIGHT

fluicell®
Pioneers in Single Cell Biology

Schedule of classes

Lec 1 – 22/7

- Introduzione al Big Data
- HPC - High performance computing -
- Definizione delle soluzioni architetturali per il trattamento di big-data
- Hadoop framework, common utilities - ecosistema e distribuzioni

Lec 2 – 27/7

- Hadoop Distributed File System (HDFS)
- Hadoop mapReduce

Lec 3 – 30/7

- Hadoop YARN - resource negotiators
- YARN framework
- Hadoop tools
- HDFS Tutorial: Architettura, Operazioni di Lettura & scrittura tramite Java API

Schedule of classes

Lec 4 – 07/09

- Elementi di rilevanza statistica dei dati
- Tecniche di visualizzazione
- Esempi python

Lec 5 – 10/09

- Introduzione agli algoritmi di data mining
- Elementi di classificazione
- Classificazione - tutorial python

Lec 6 – 14/09

- Regressione lineare e logistica
- Regressione - tutorial python
- Clusterizzazione per dati non annotati
- Clusterizzazione - tutorial python

Schedule of classes

Lec 7 – 17/09
- Introduzione a spark
- Resilient distributed system (RDD)
- introduzione alle reti neurali
- Il percettrone - funzioni di attivazione dei neuroni
- Reti neurali lineari

Lec 8 – 21/09
- Funzioni di costo
- Tutorial learning con CIFAR10 su rete lineare
- Reti lineari convoluzionali
- Database distribuiti e data loading

Lec 9 – 24/09
- Tecniche di annotazione
- Supervised learning
- Back propagation
- Semi-supervised learning

Schedule of classes

Lec 10 – 27/09
- Unsupervised learning
- Reinforcement learning
- costruzione di livelli neurali con pytorch

Lec 11 – 01/10
- Deep learning
- Transfer learning
- Tutorial su learning supervisionato con rete convoluzionale

Lec 12 – 05/10
- Computer vision
- Reti neurali ricorsive
- Reti neurali con memoria
- Speech recognition
- Natural language processing – tutorial pytorch
- Conclusione del corso

Letteratura raccomandata

- White, T., 2012. Hadoop: The definitive guide. " O'Reilly Media, Inc."
  - https://github.com/tomwhite/hadoop-book/
- Severance, C. and Dowd, K., 2018. High Performance Computing.
- Trobec, R., Slivnik, B., Bulic, P. and Robic, B., 2018. Introduction to Parallel Computing: From Algorithms to Programming on State-of-Art Platforms. Springer.
- http://hadoop.apache.org/docs/current/index.html
- LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), pp.436-444.
- Eli Stevens, Luca Antiga, Thomas Viehmann, “Deep Learning With Pytorch: Build, Train, and Tune Neural Networks Using Python Tools” 2020
- https://github.com/EbookFoundation/free-programming-books/blob/master/books/free-programming-books.md

Prerequisiti

- Elementi di programmazione orientata agli oggetti (classi, metodi, attributi etc.)
- Conoscenza delle strutture di rete (IP, porte, configurazioni di rete etc.)
- Conoscenze base sull’uso del terminal in ambiente linux
- Matematica di base

Background?

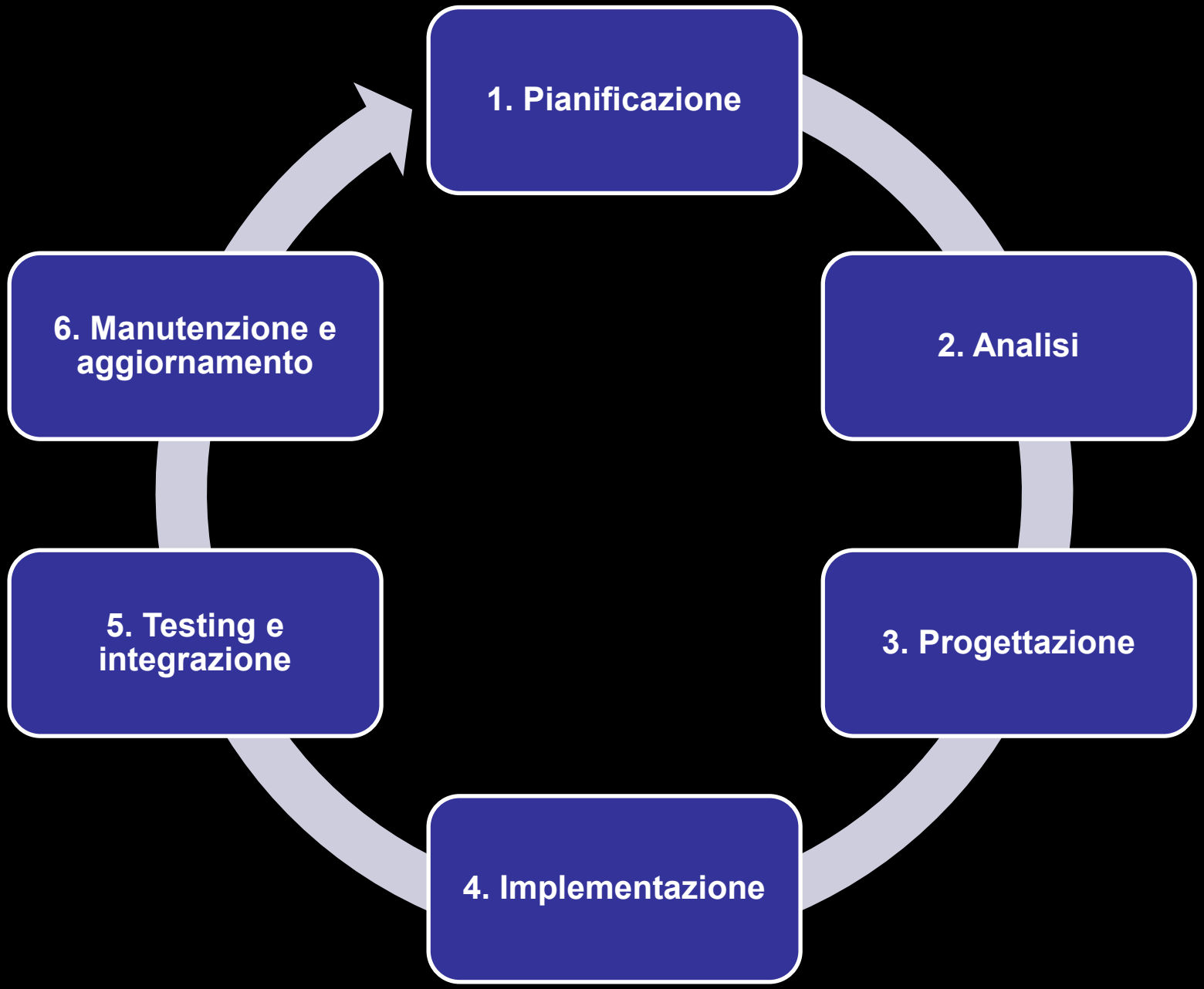
Titolo di studio: es. Ing. Informatico/Meccanico/ …

Anno Accademico di laurea – es. 2020

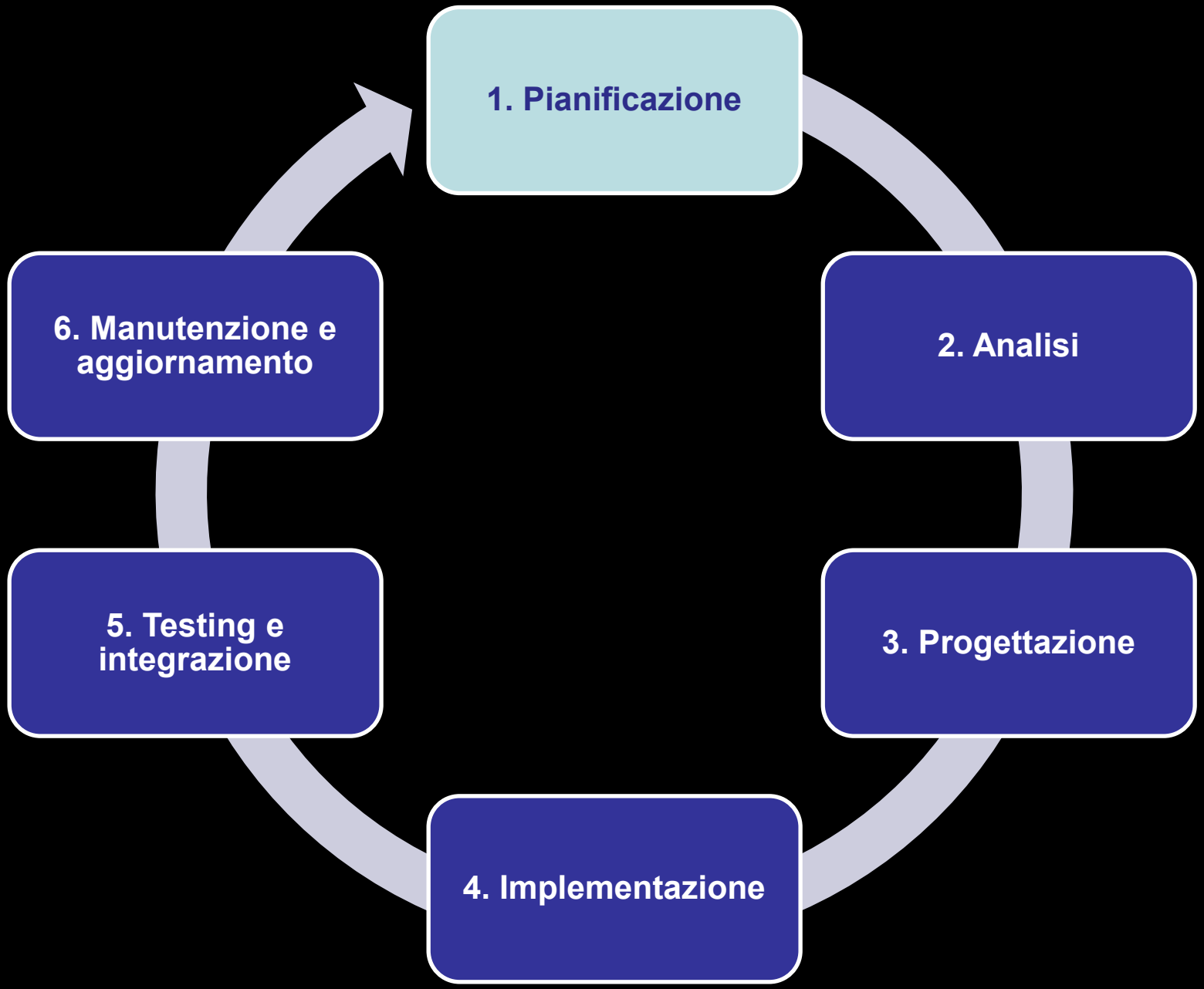
Ruolo lavorativo attuale: es. sviluppatore Java / gestore di rete / …

Lavori precedenti (se applicabile): es. sviluppatore applicazioni web,

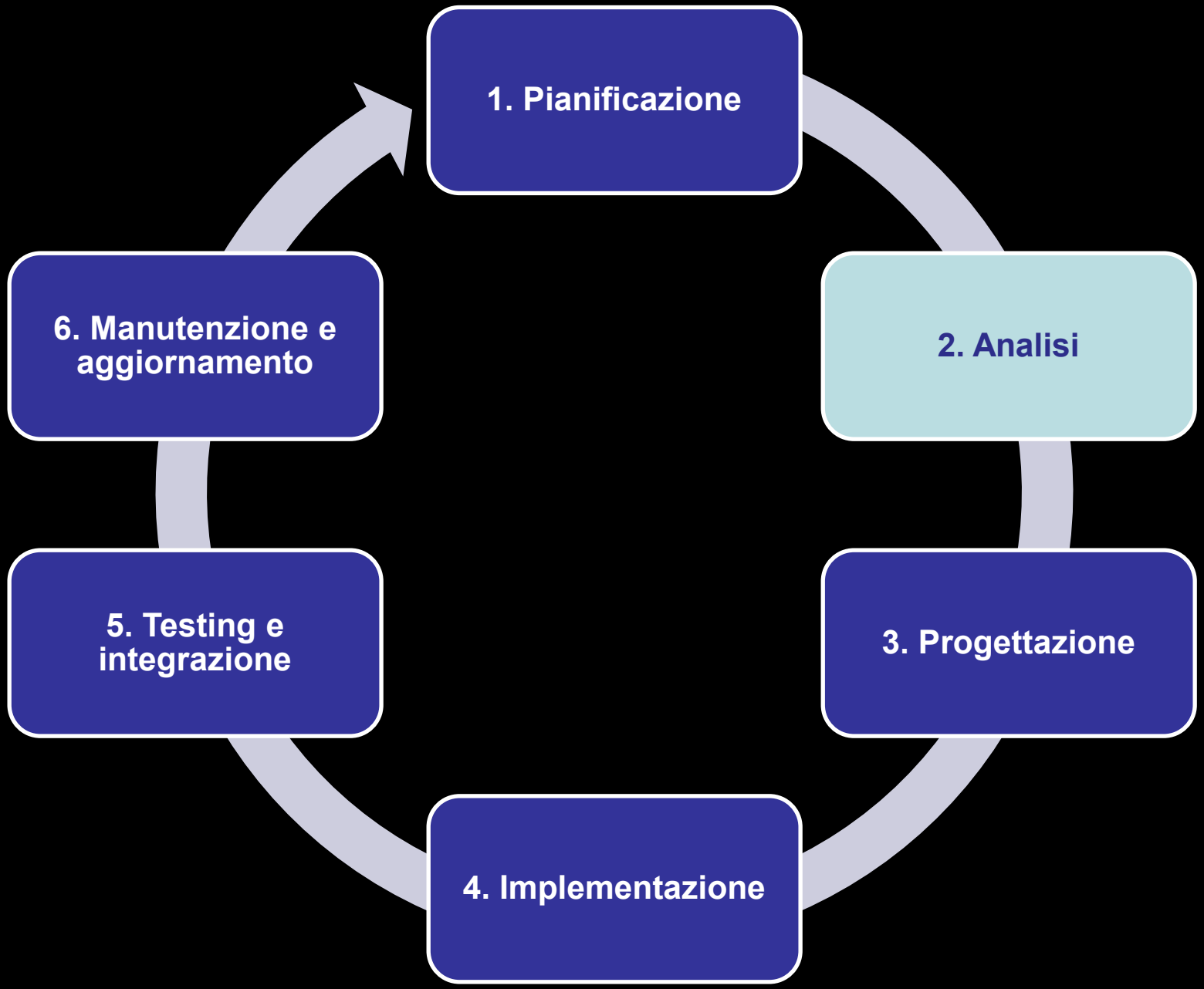
Ciclo del software



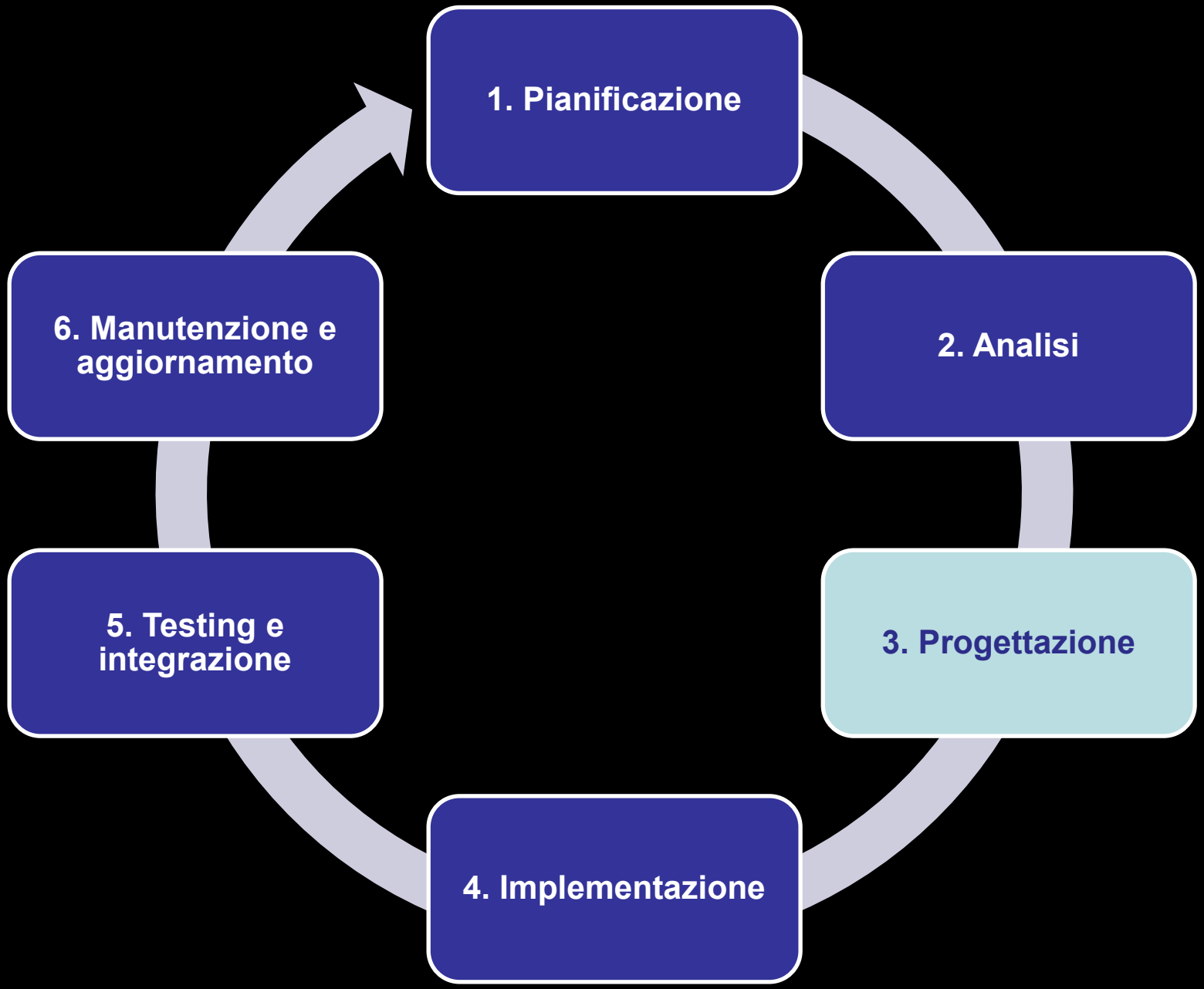
Ciclo del software



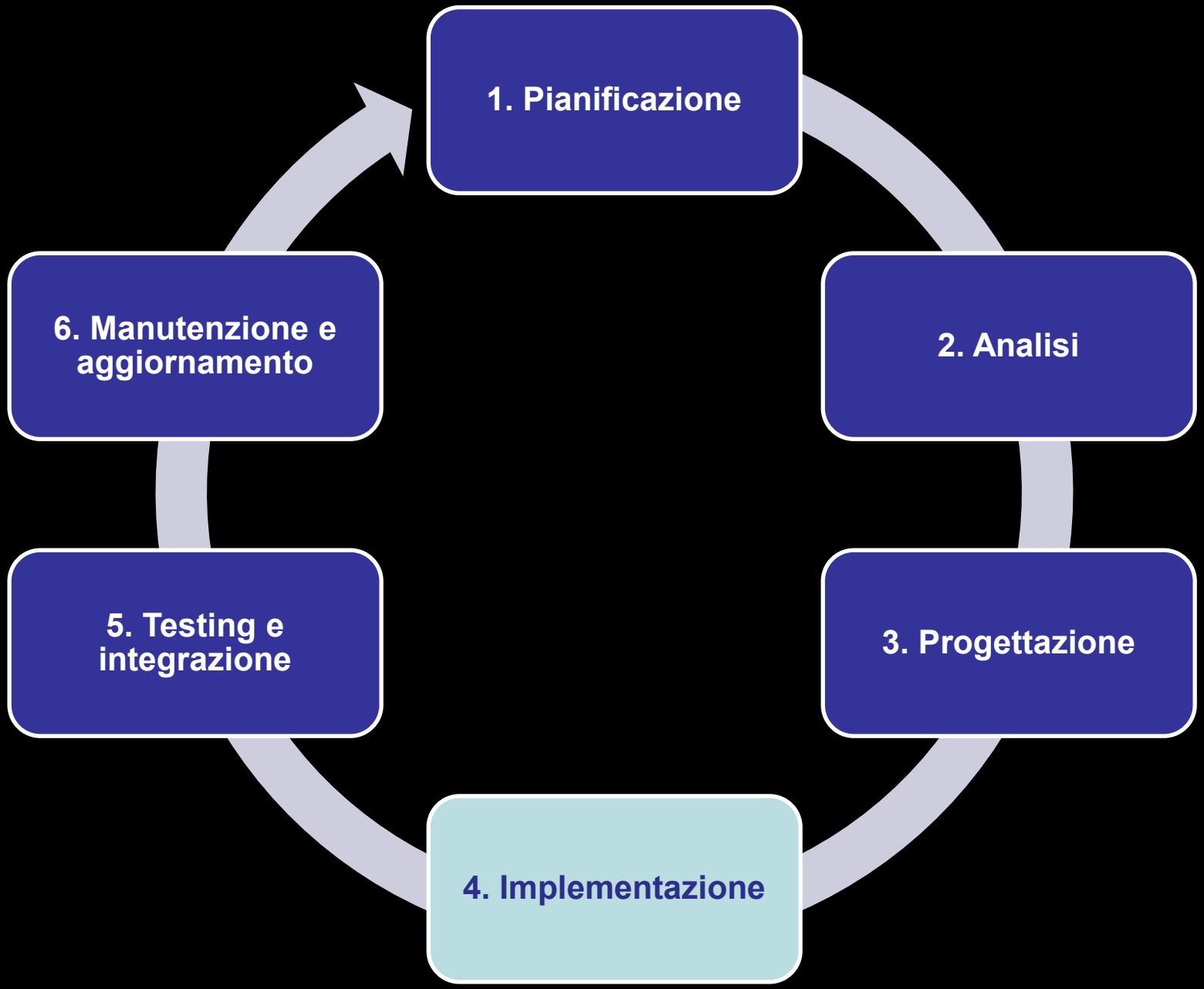
Ciclo del software



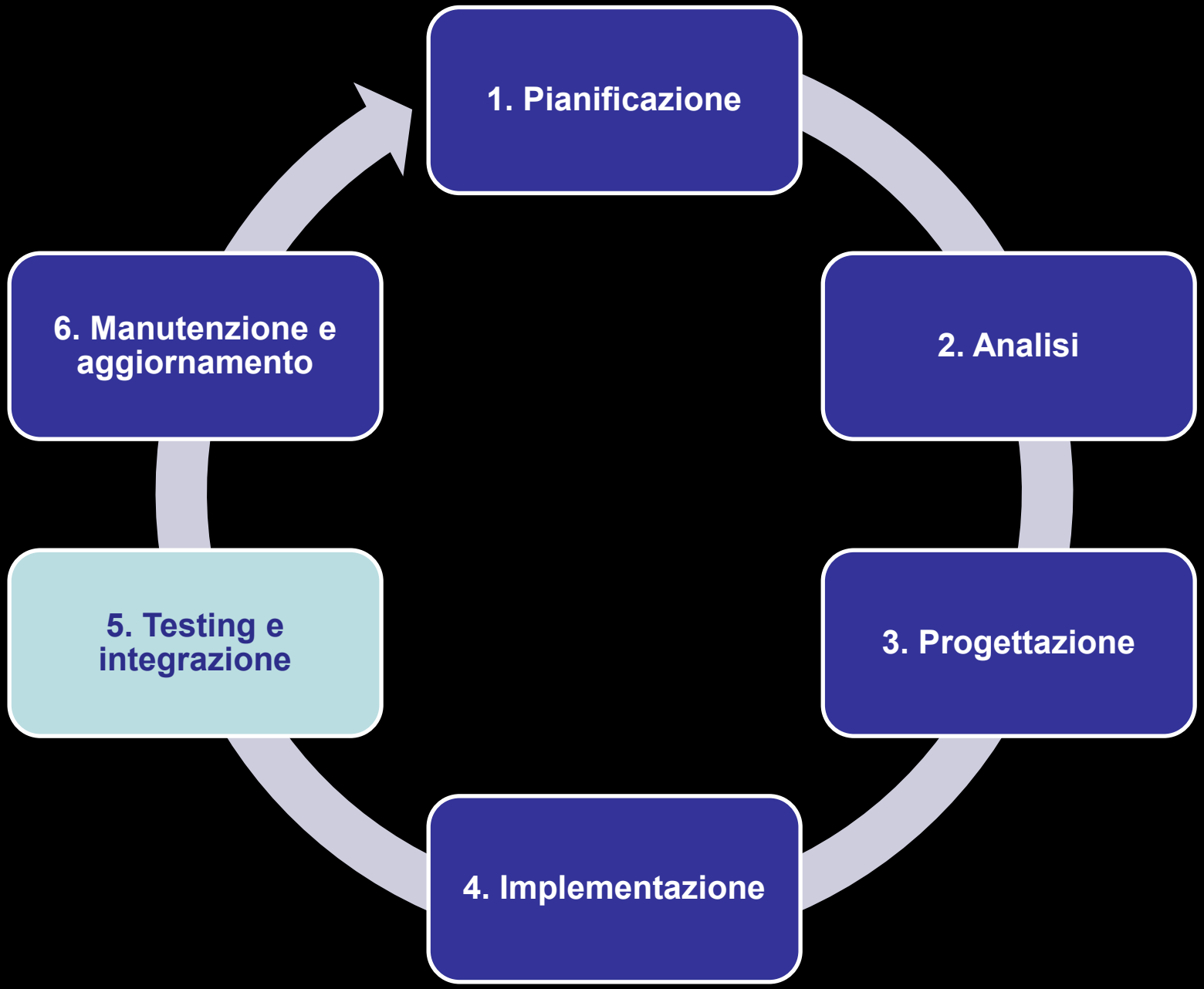
Ciclo del software



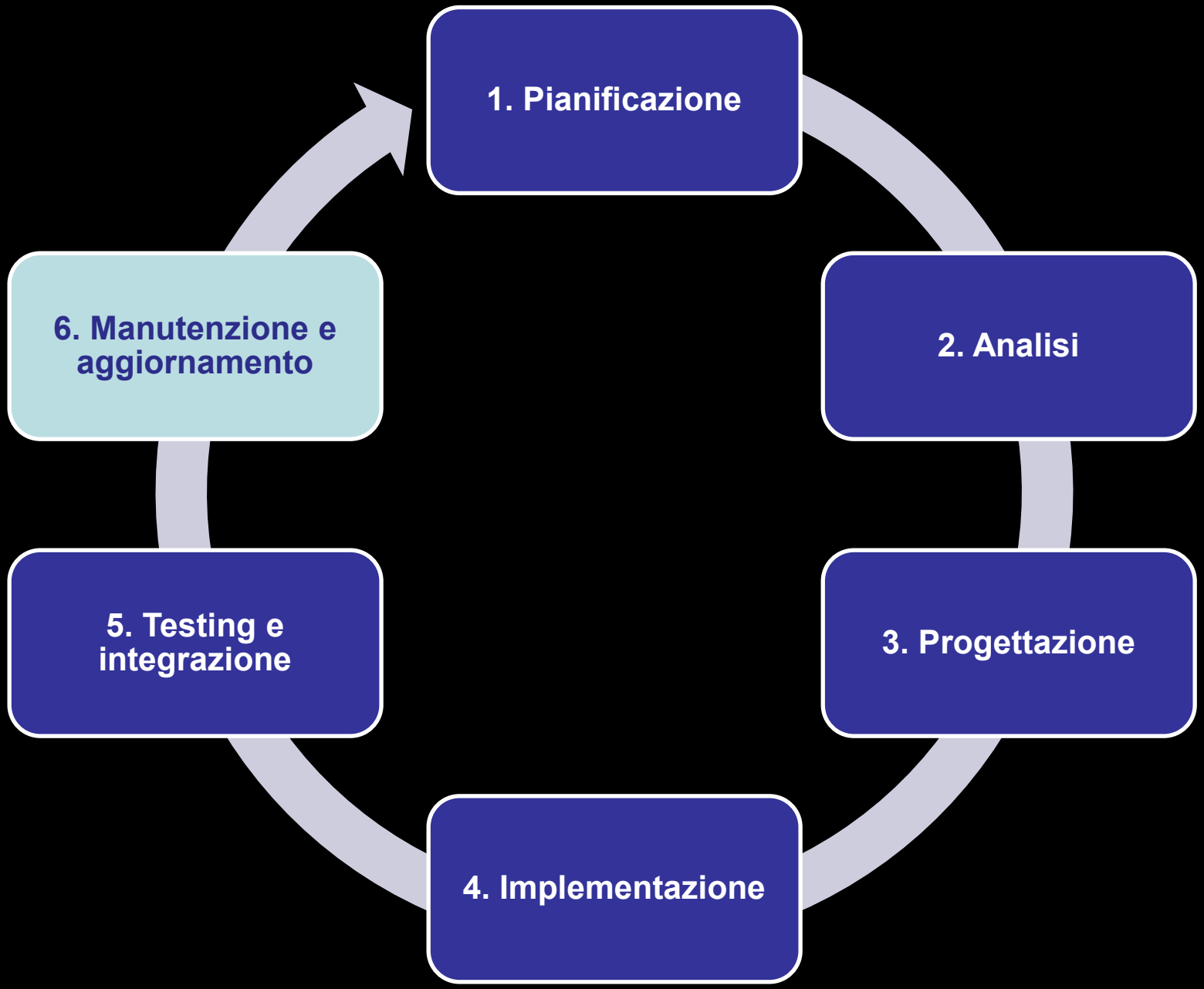
Ciclo del software



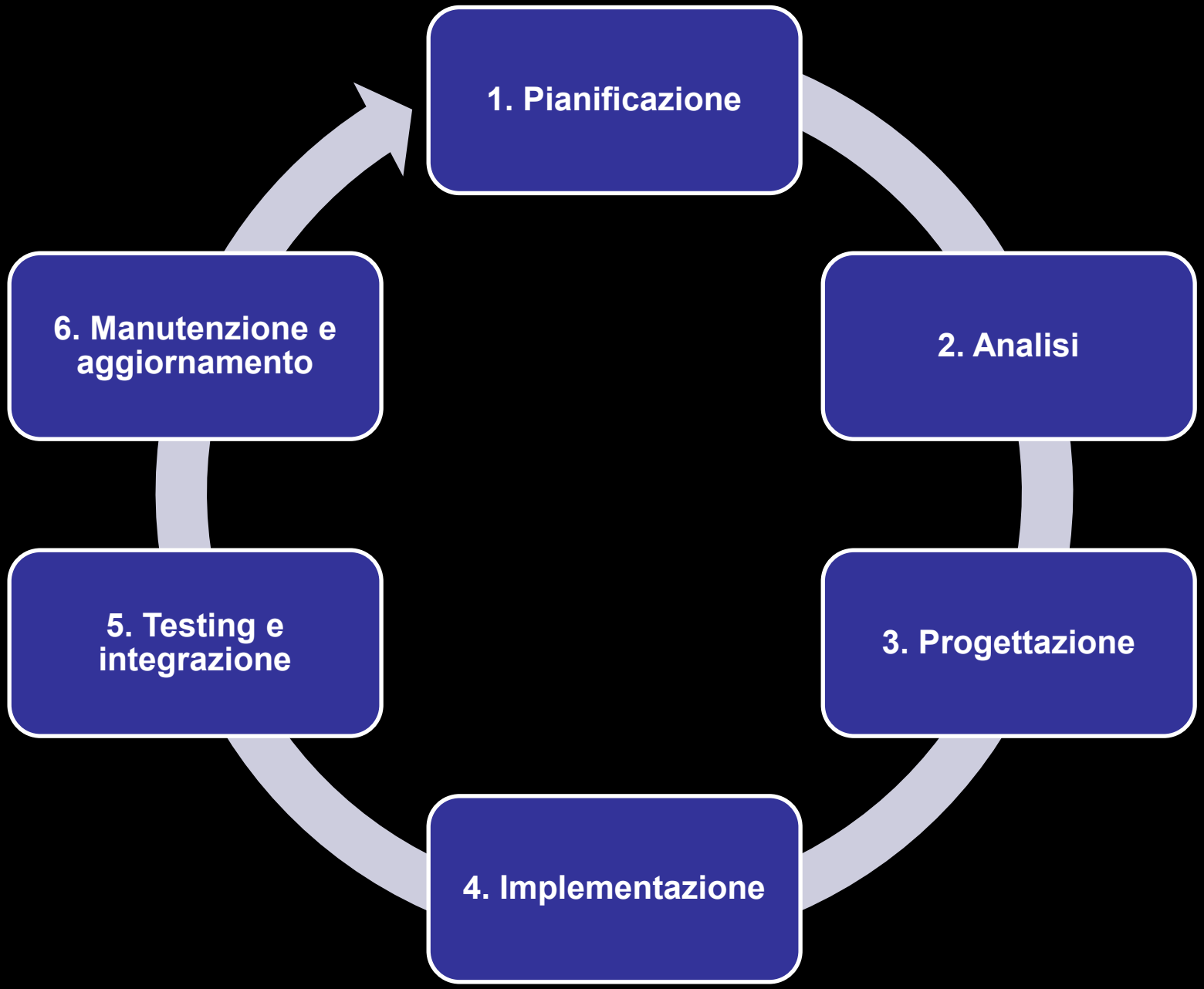
Ciclo del software



Ciclo del software



Ciclo del software



Ogni giorno il nostro smartphone produce dati nell'ordine degli exabytes (10^{18})

Ogni giorno il nostro smartphone produce dati nell'ordine degli exabytes (10^{18})

x 5.000.000.000 di utenti !!!!

Ogni minuto su internet:

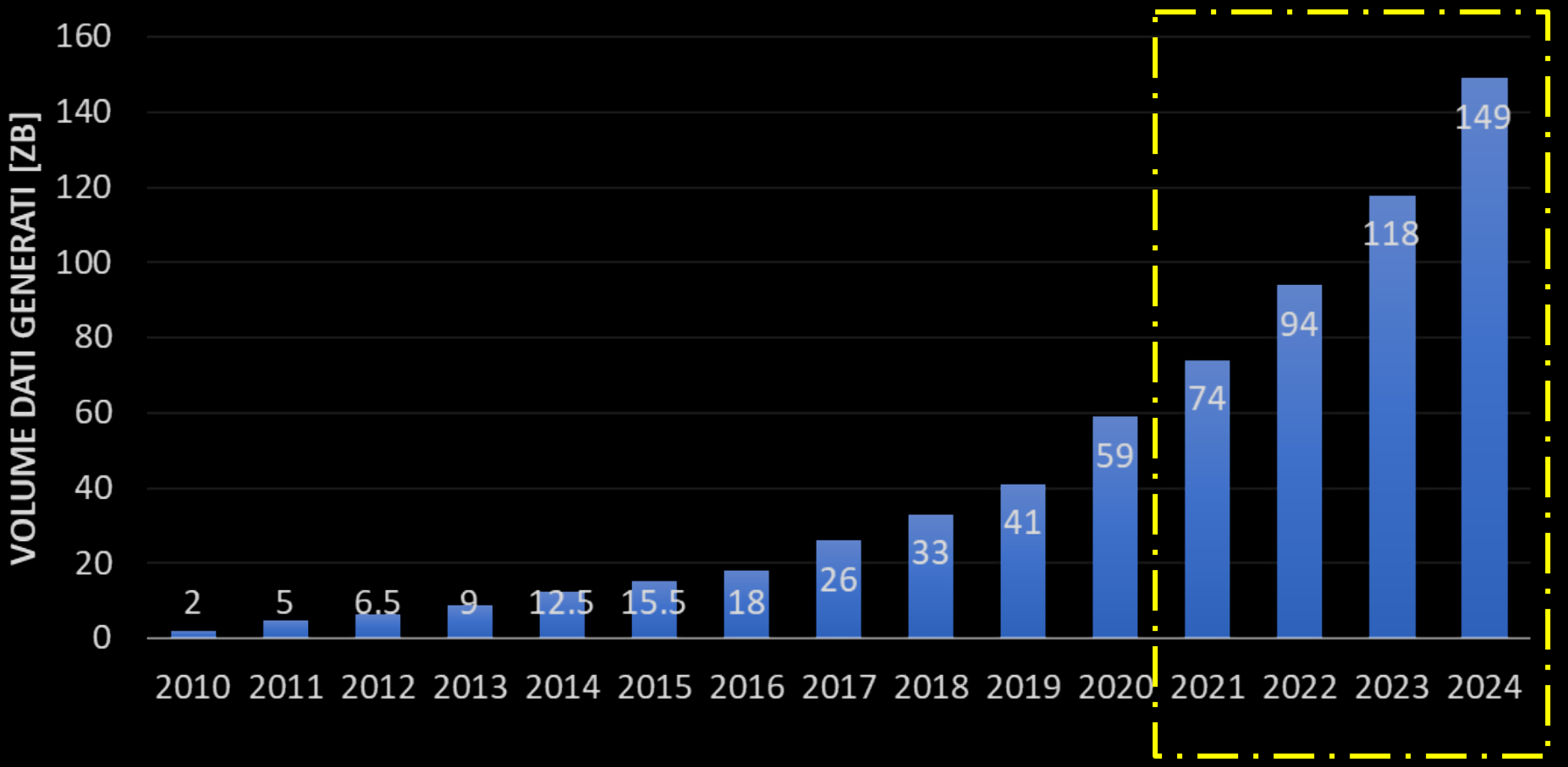
2.1 milioni snapchat

3.8 milioni di ricerche su google

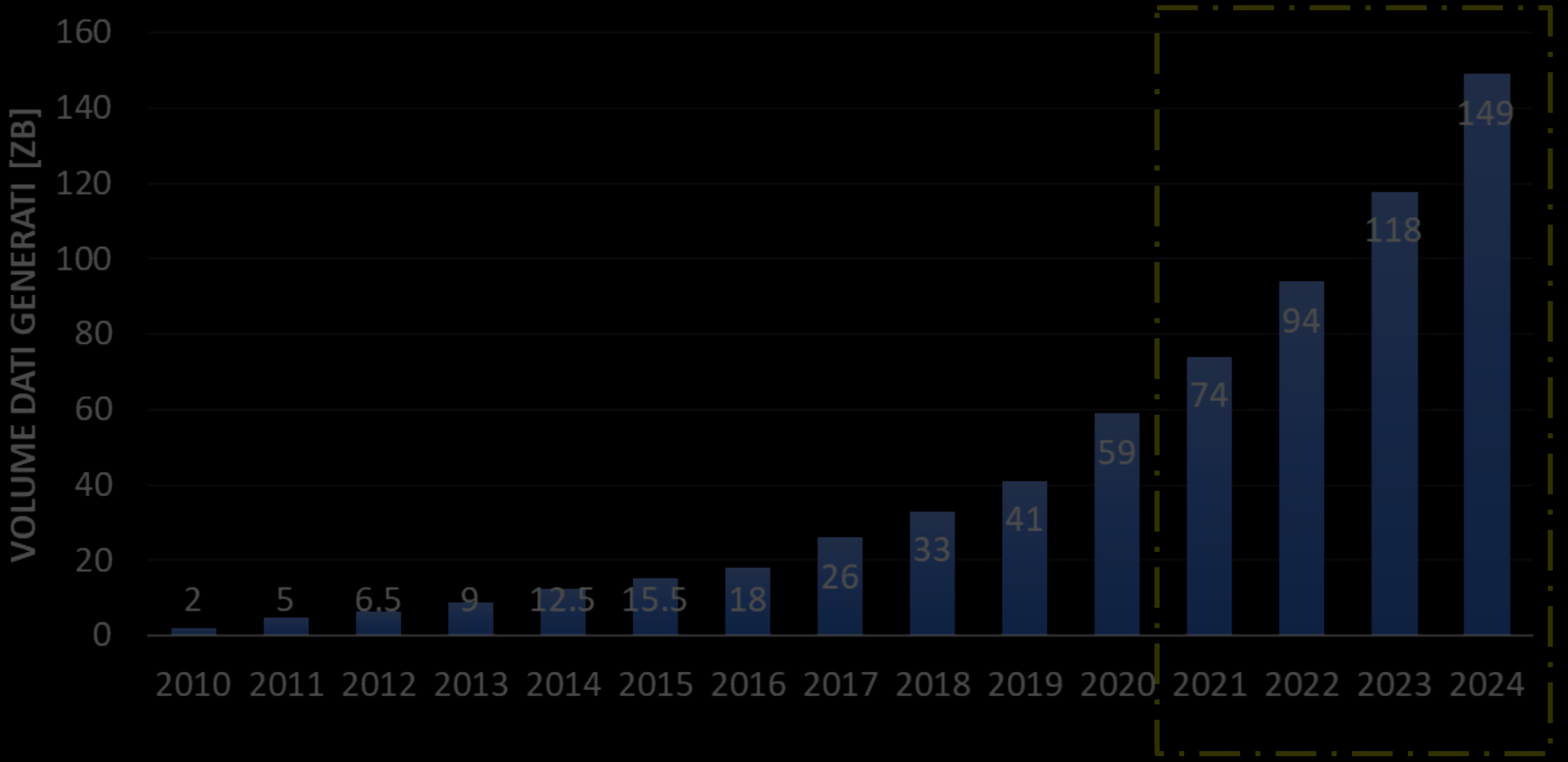
1.0 milioni di login su facebook

4.5 milioni di video sono guardati su youTube

188 milioni di e-mail inviate



source: https://datawrapper.dwcdn.net/EmEY2/2/

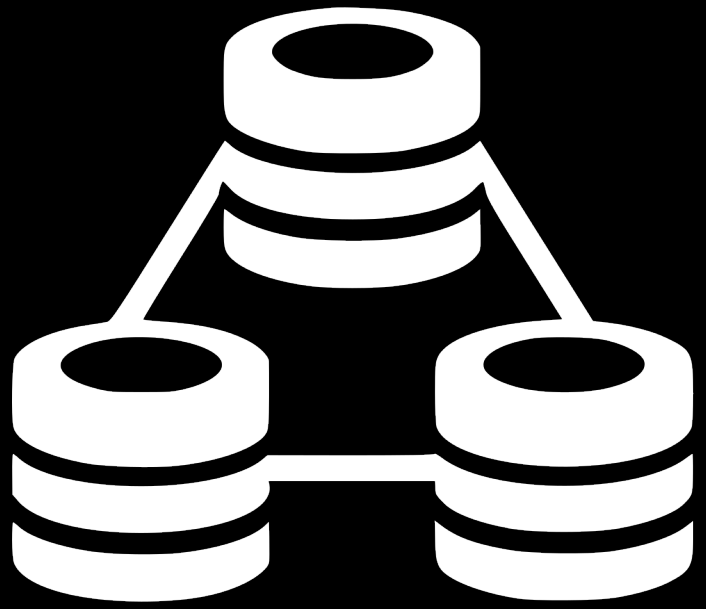


Zettabyte = 10^{21} = 1 000 000 000 000 000 000 000

source: https://datawrapper.dwcdn.net/EmEY2/2/

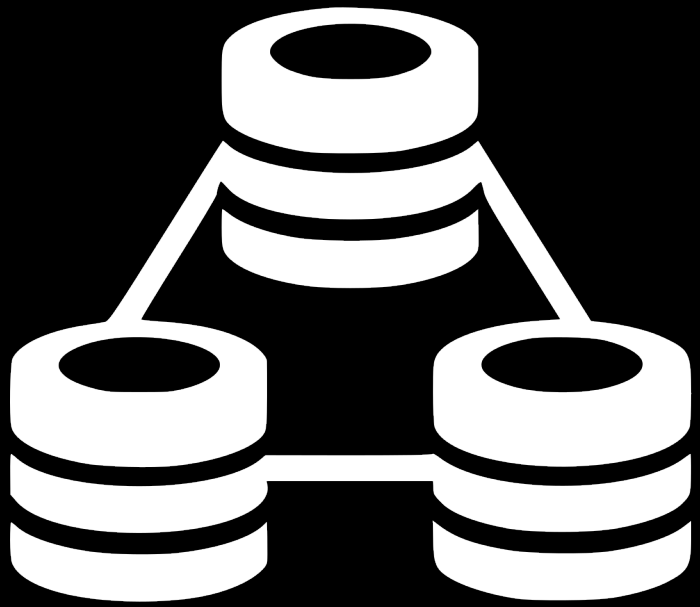
Big data

Con Big data is intende un insieme di dataset che non possono essere processati usando tecniche computazionali classiche.



Big data

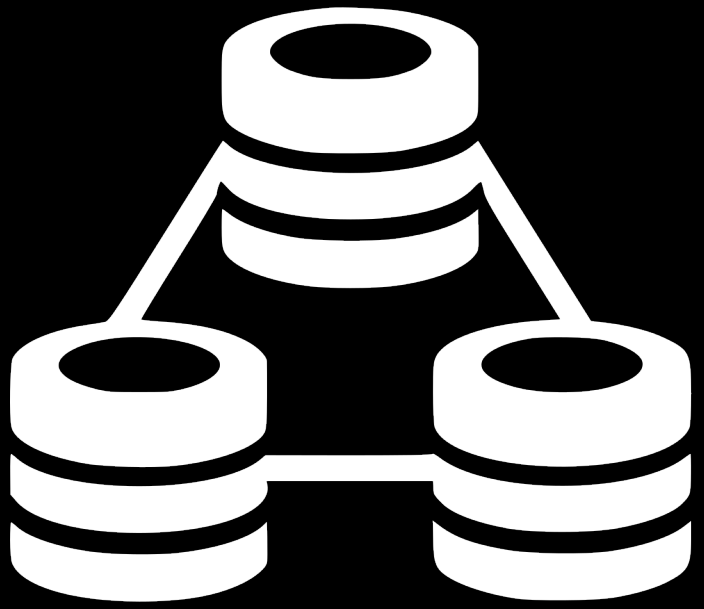
Con Big data is intende un insieme di dataset che non possono essere processati usando tecniche computazionali classiche.



Big data

Con Big data is intende un insieme di dataset che non possono essere processati usando tecniche computazionali classiche.

Non ci riferiamo ad un singolo tool o tecnica, ma ad un insieme di strumenti, tecniche e frameworks.



Big data

Con Big data is intende un insieme di dataset che non possono essere processati usando tecniche computazionali classiche.

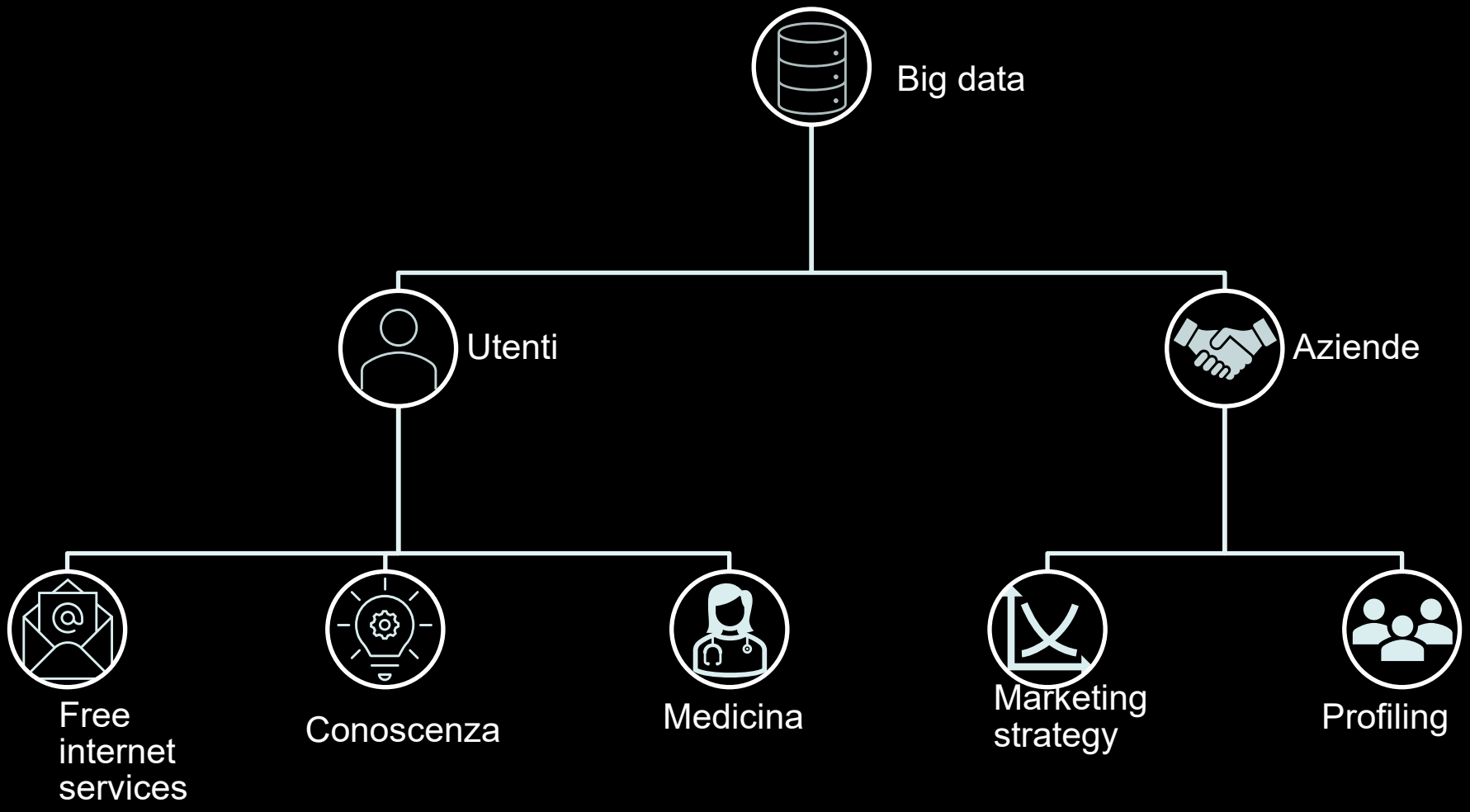
Non ci riferiamo ad un singolo tool o tecnica, ma ad un insieme di strumenti, tecniche e frameworks.

HPC- High performance computing
Aumentare la capacità di calcolo in maniera distribuita per risolvere problemi complessi che richiedono l'accesso ad una grande mole di informazione (big data)

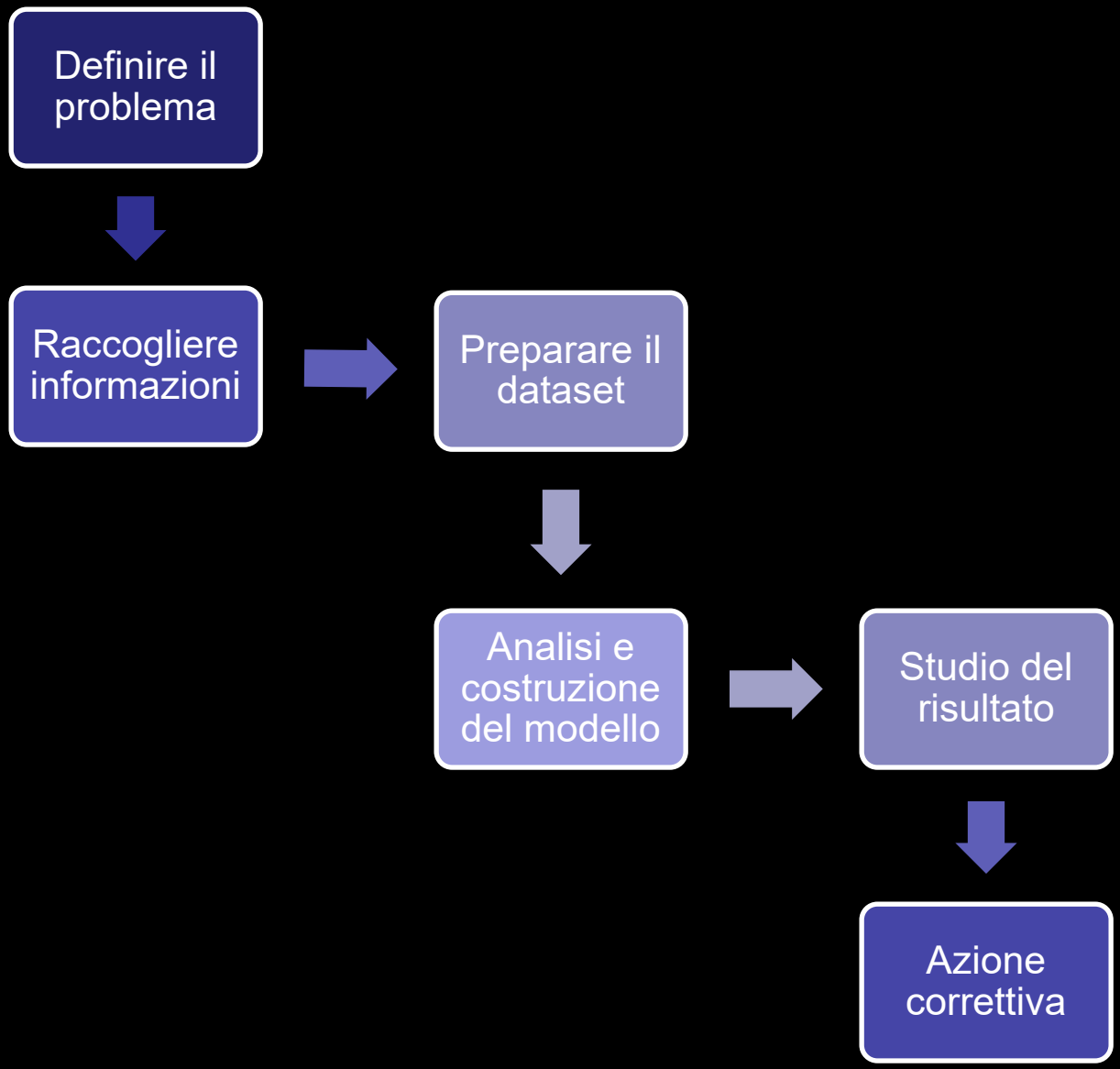
Esempi di utilizzo dei big data

- Social Media Data − Social media come Facebook e Twitter hanno una grande quantità di informazioni ben catalogata postata dagli utenti nel mondo.
- Stock Exchange Data − Lo stock exchange contiene una grande mole di dati su acquisti/vendite di azioni e prodotti finanziari in tutto il mondo. Le decisioni prese dai diversi attori del settore hanno un enorme valore commerciale.
- Power Grid Data − La rete elettrica e la sua gestione ha la necessità di gestire informazioni rilevanti riguardo al consumo di energia e ai profili energetici egli utenti.
- Transport Data − Dati di trasporti e spostamento che includono modelli, capacità dei mezzi, distanza percorsa, prezzi dei biglietti e disponibilità di mezzi.
- Search Engine Data − I motori di ricerca raccolgono e rendono disponibili dati da molti database diversi.
- Scatole nere − Le scatole nere registrano grandi quantità di dati durante il volo che includono conversazioni tra I membri del personale di bordo e tutto ciò che succede a bordo.

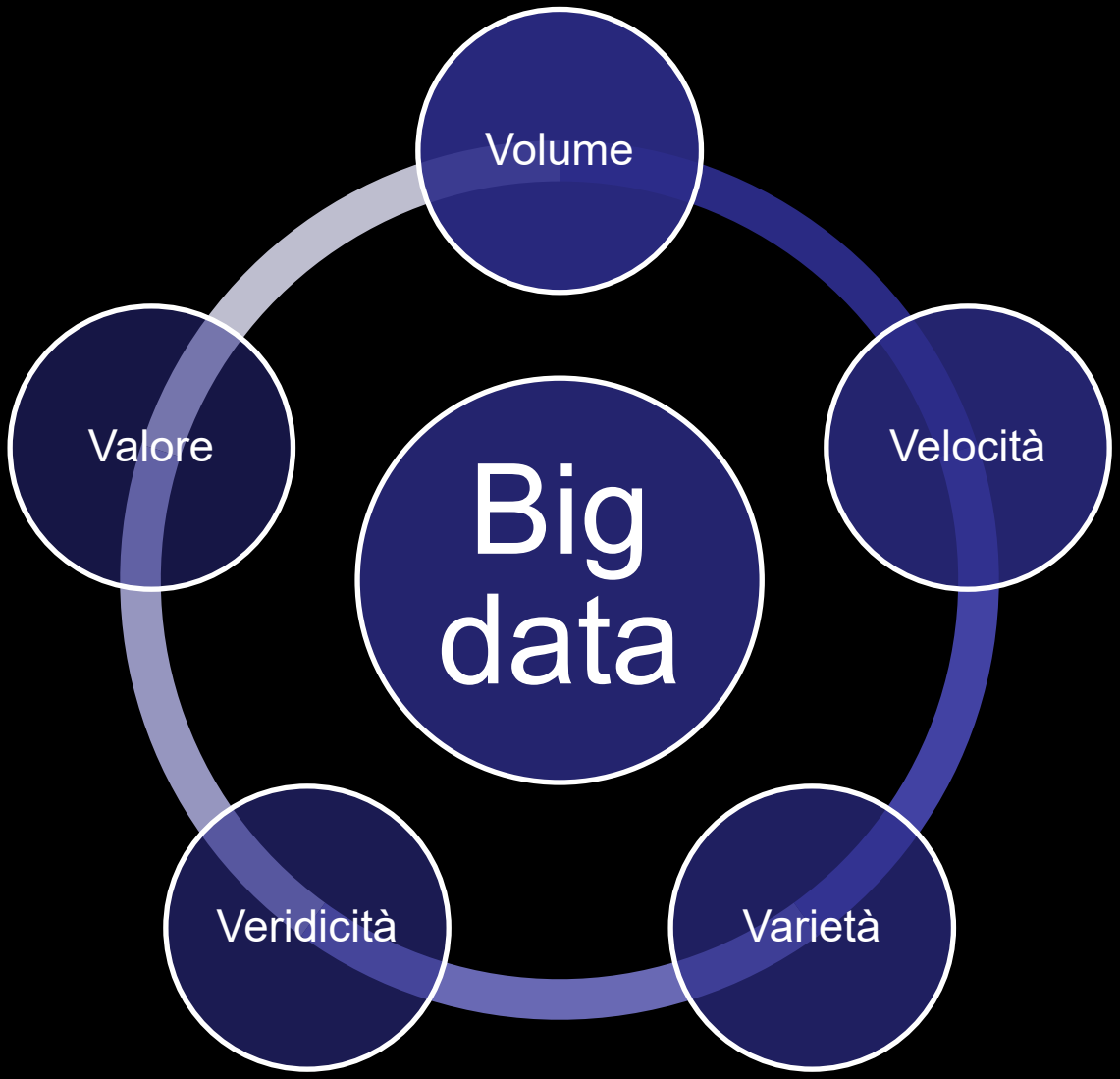
Benefici dei Big Data



Approccio alle tecnologie sul big data

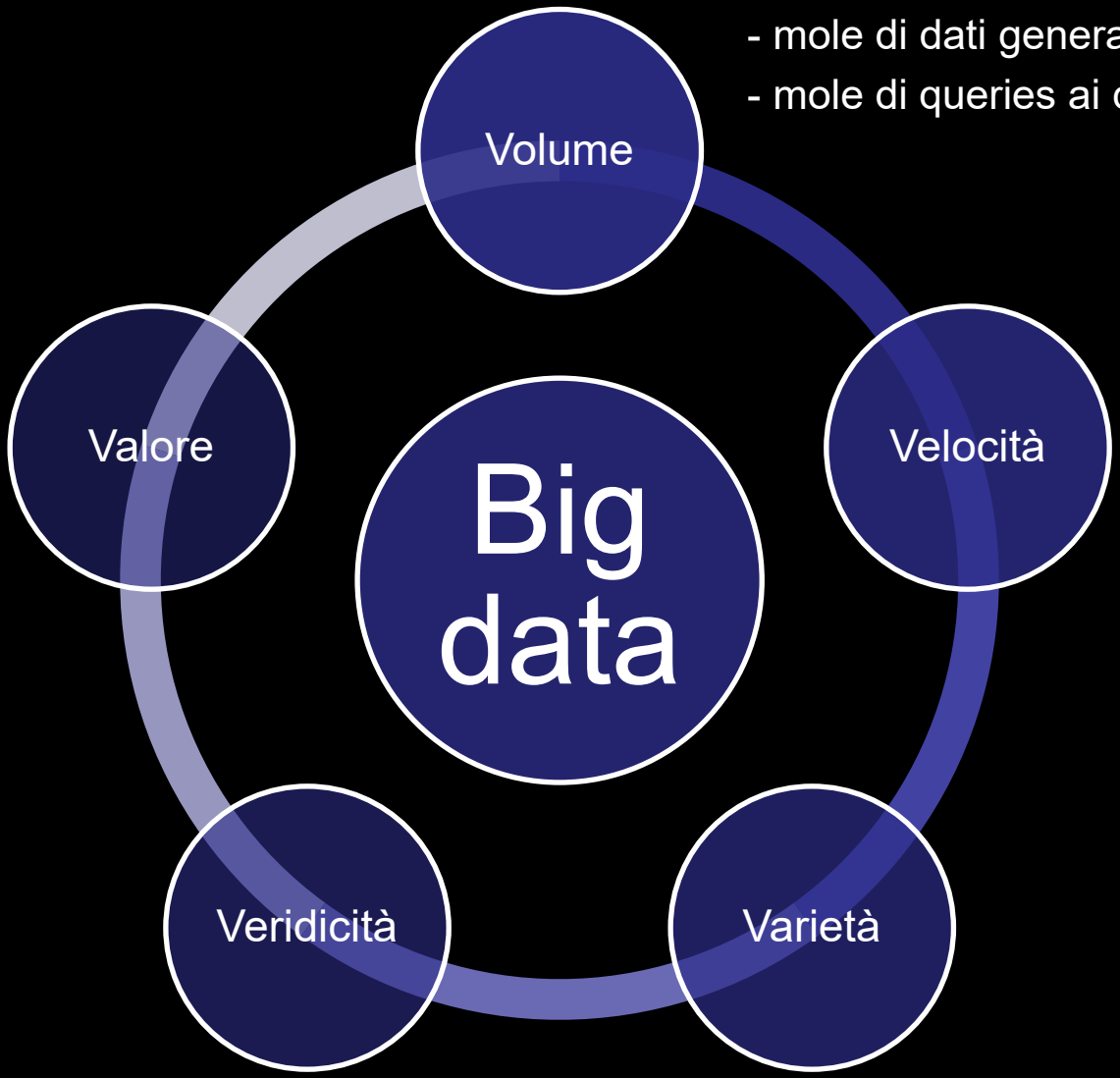


5-V

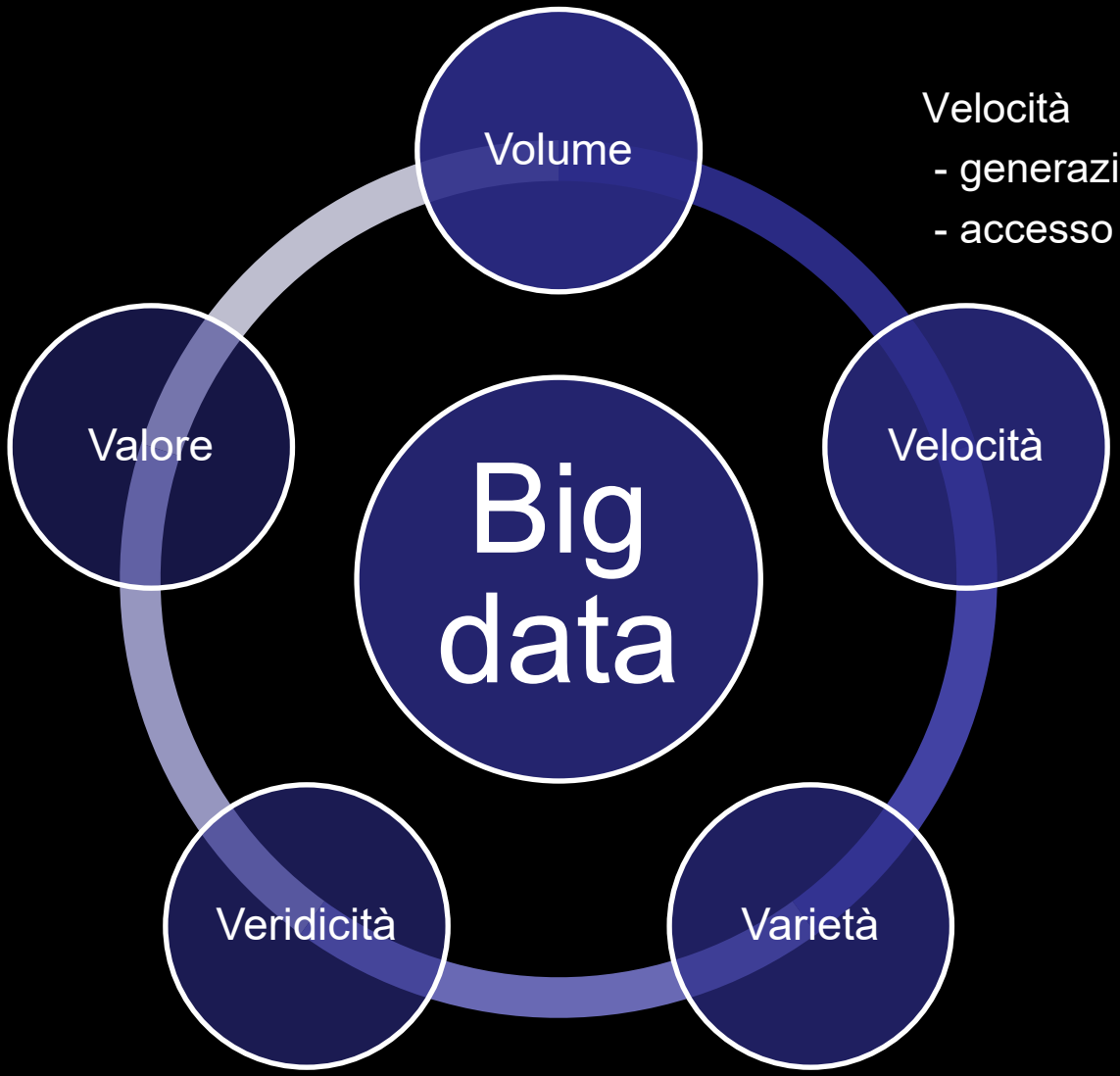


5-V

Volume
- mole di dati generati
- mole di queries ai dati

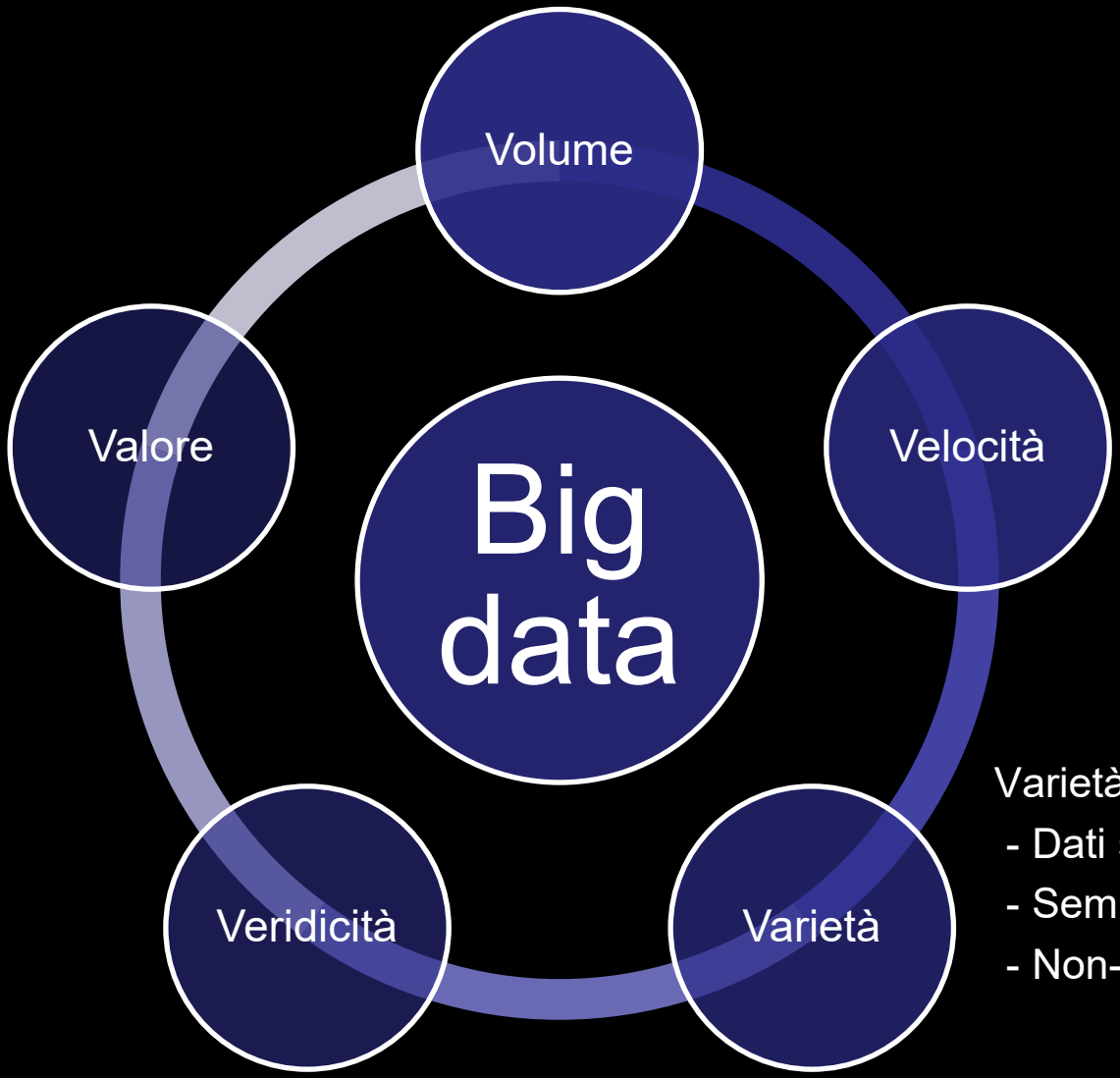


5-V



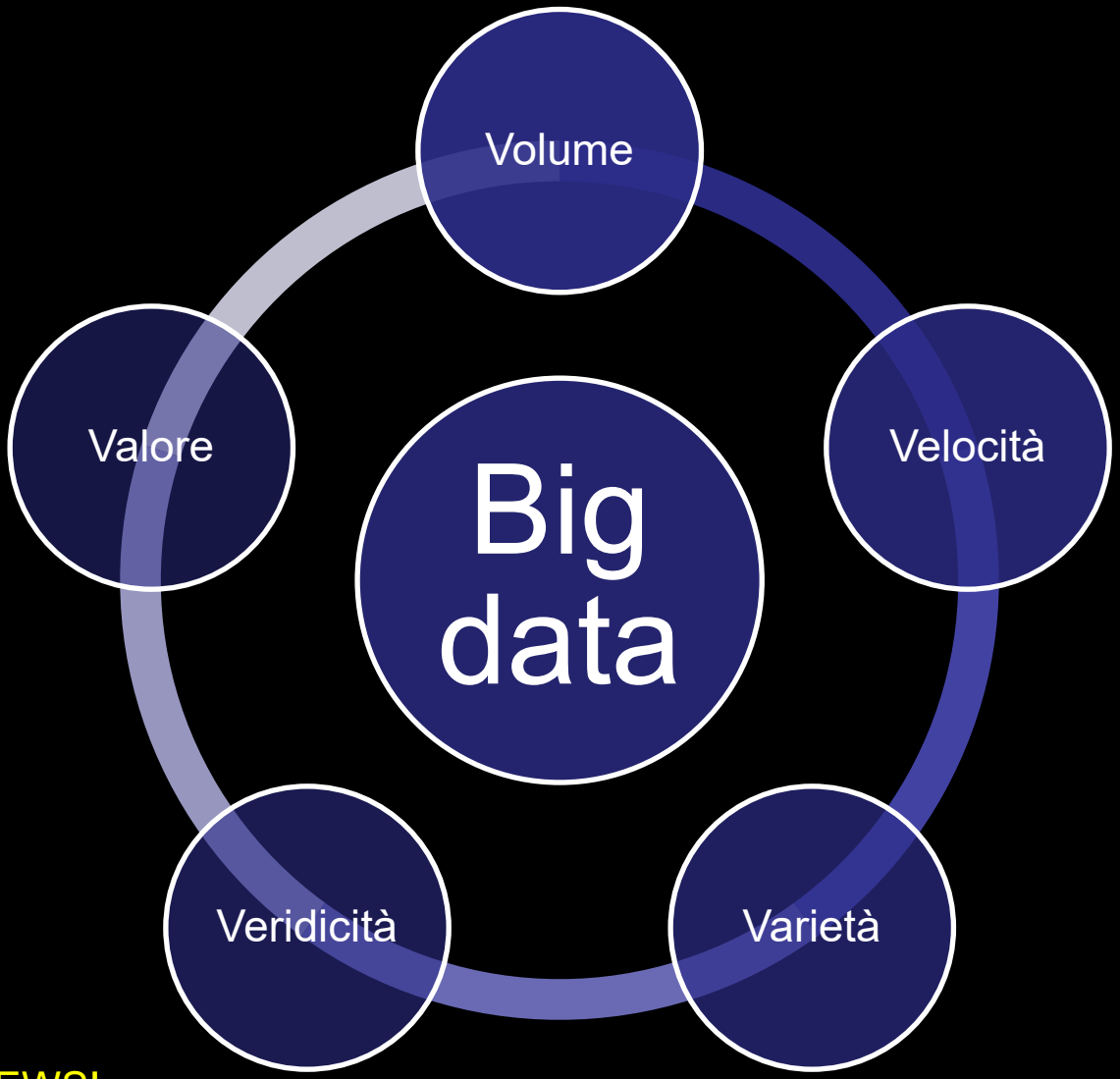
Velocità
- generazione
- accesso

5-V



Varietà:
- Dati strutturati
- Semi strutturati
- Non-strutturati

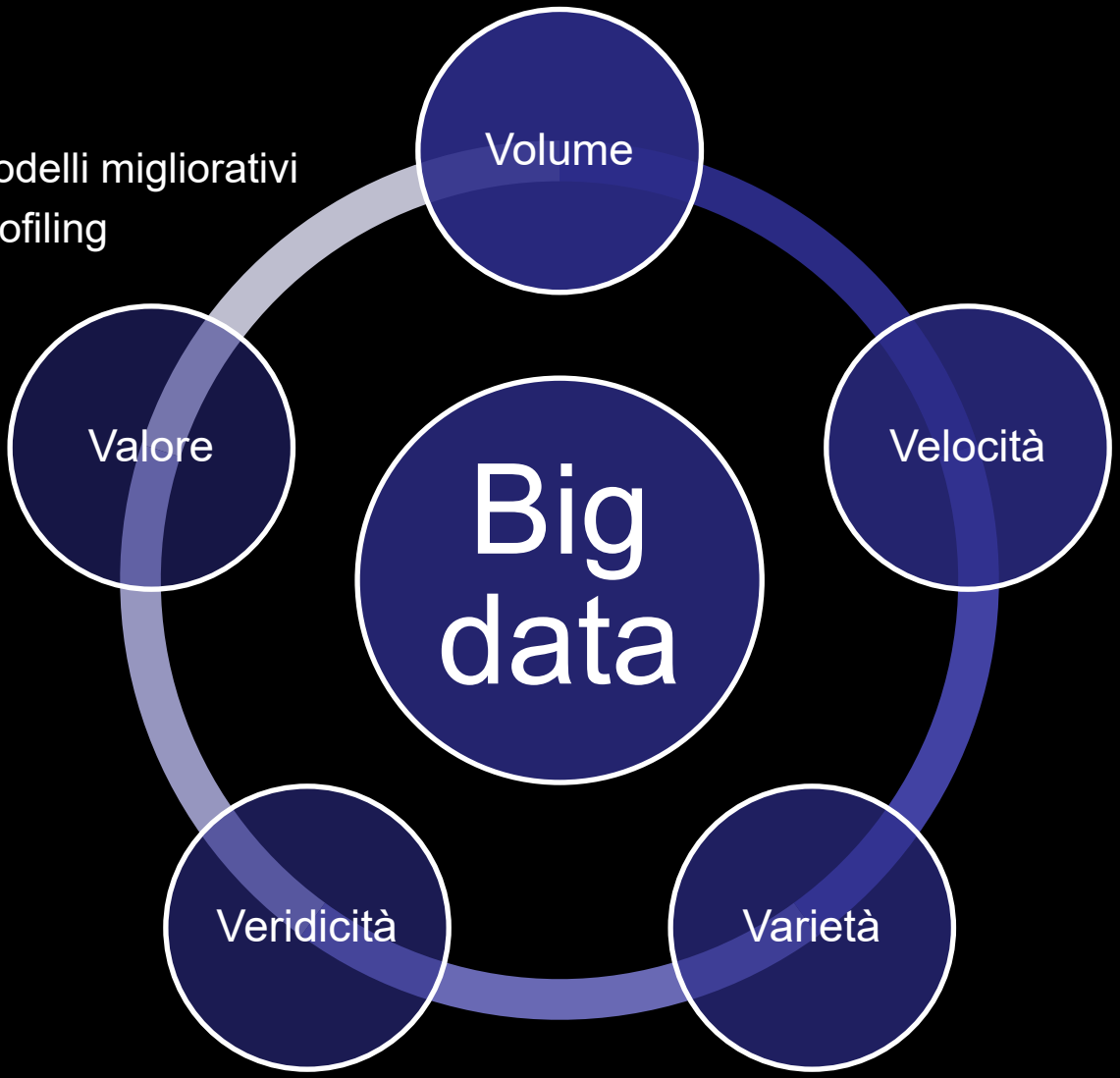
5-V



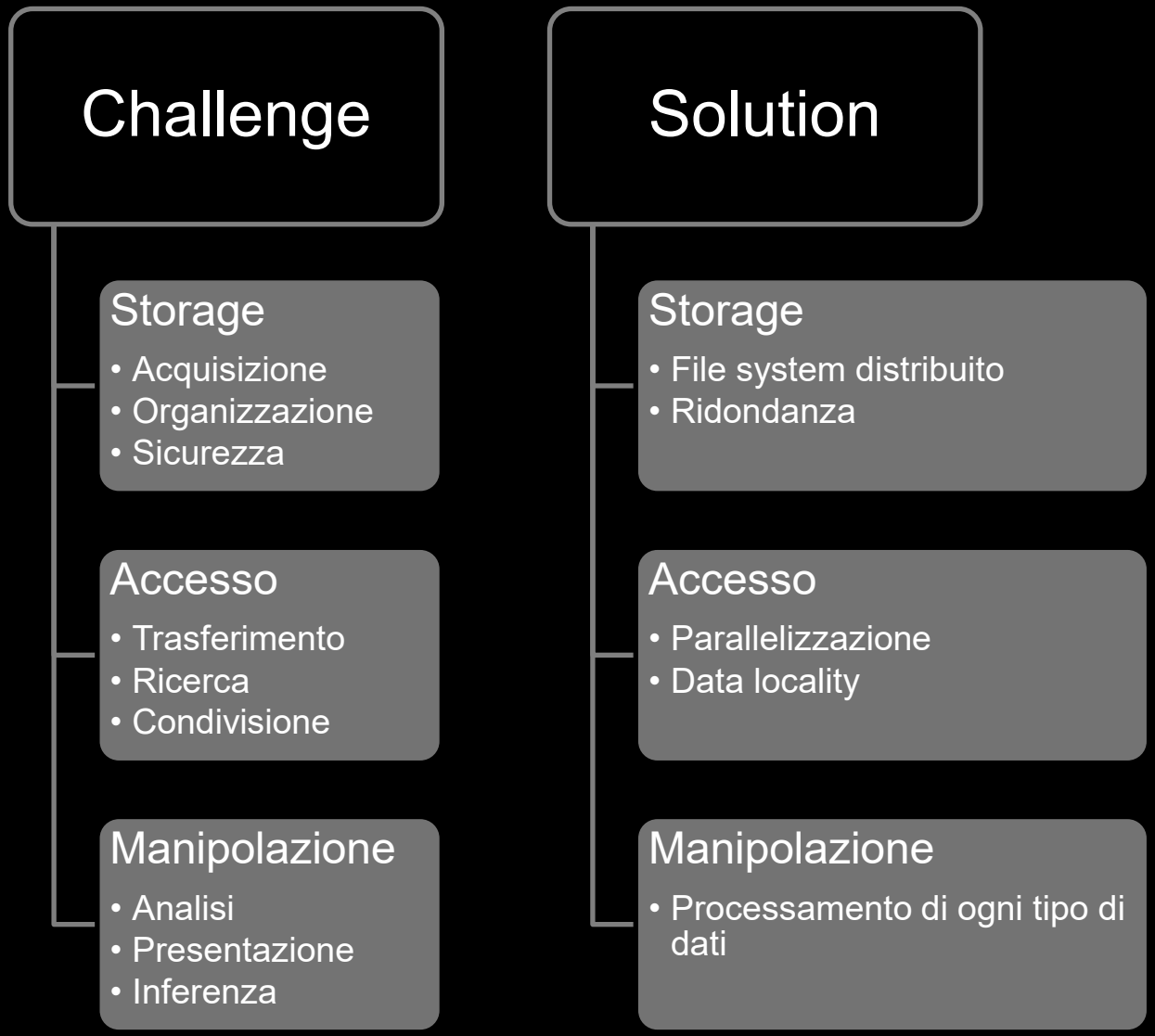
Veridicità:
- Dati accurati
- Veritieri
- NO FAKE NEWS!

5-V

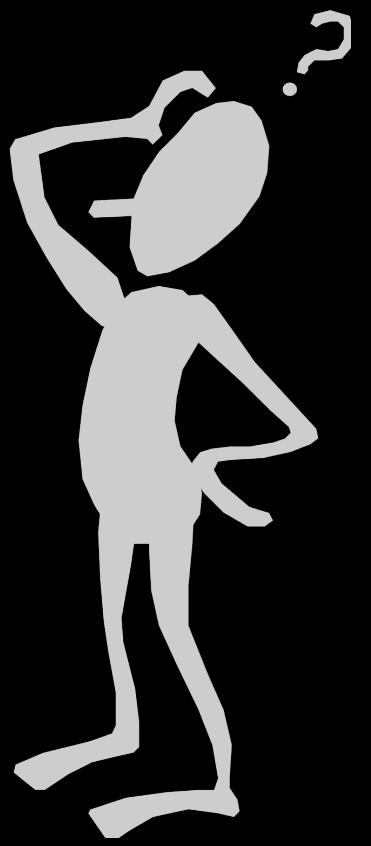
Valore:
- Generare modelli migliorativi
- Customer profiling



Big Data challenges and solutions



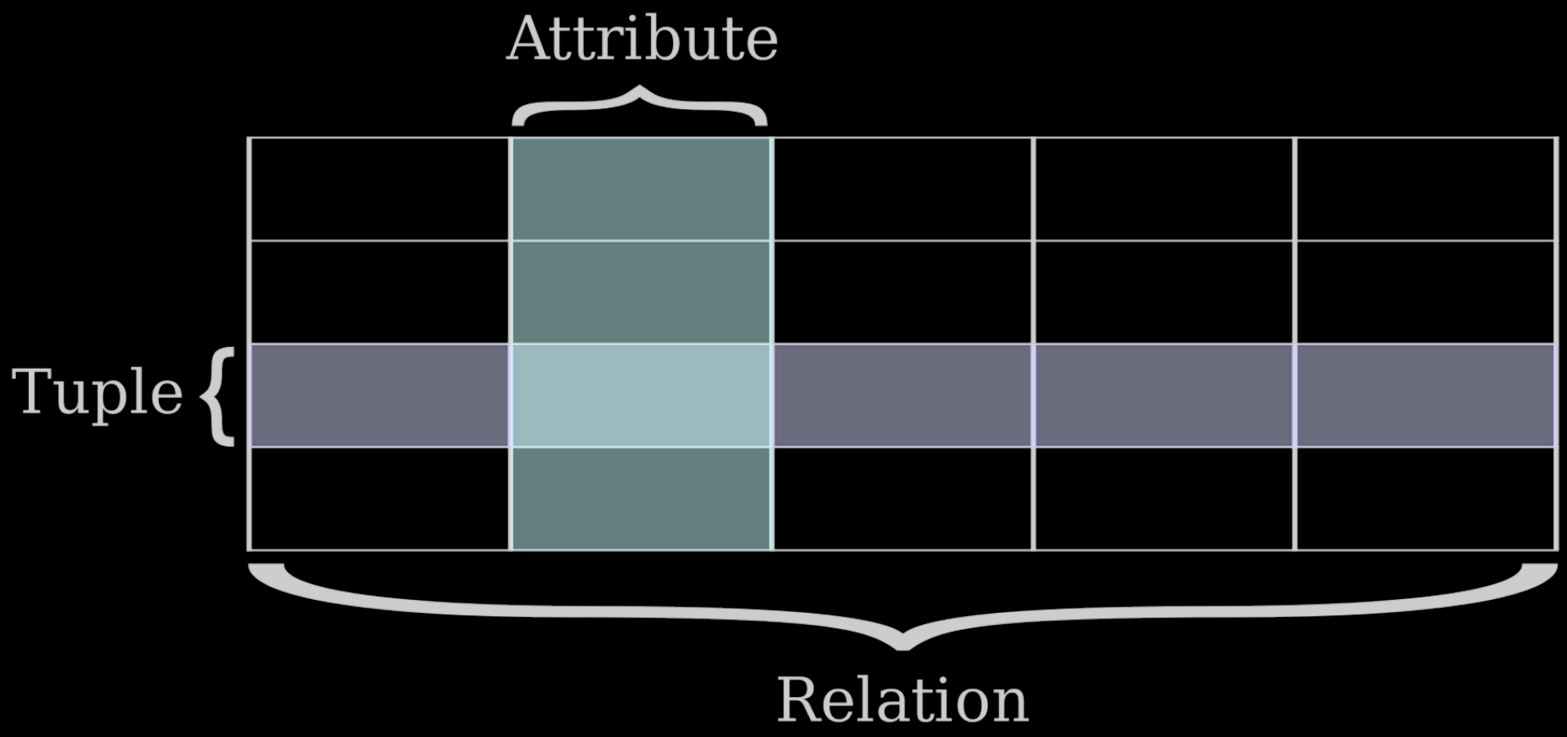
Com’è possibile memorizzare una mole così grande di informazione?



Database relazionale (SQL) Vs. NoSQL (non-relazionale)

Un database relazionale (anni 60) è basato su un modello tabulare tra attributi e voci.

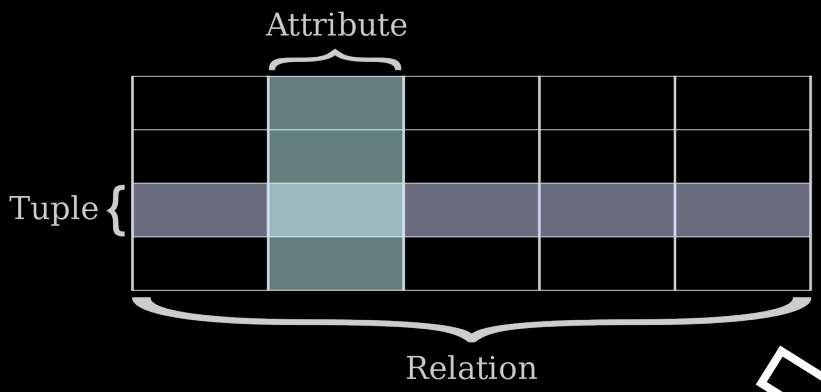
Tipicamente è gestito tramite un sistema di gestione relazionale con l'opzione di usare un sistema di interrogazione strutturato (SQL – Structured query language)



Database relazionale (SQL) Vs. NoSQL (non-relazionale)

Un database relazionale (anni 60) è basato su un modello **tabulare** tra attributi e voci.

Tipicamente è gestito tramite un sistema di gestione relazionale con l'opzione di usare un sistema di interrogazione strutturato (SQL – Structured query language)



auto	peso	prezzo
Fiat panda	2	10
Alfa 147	3	12
Volvo x90	4	14

Database relazionale (SQL) Vs. NoSQL (non-relazionale)

Memorizzazione dati senza schema relazionale

Key	value
2345677	any
.	.
.	.
.	.

Database relazionale (SQL) Vs. NoSQL (non-relazionale)

Memorizzazione dati senza schema relazionale

- ✓ Facile scalabilità del Sistema

Key	value
2345677	any
.	.
.	.
.	.

Database relazionale (SQL) Vs. NoSQL (non-relazionale)

Memorizzazione dati senza schema relazionale

- ✓ Facile scalabilità del Sistema
- ✓ Minore uso di memoria

Key	value
2345677	any
.	.
.	.
.	.

Database relazionale (SQL) Vs. NoSQL (non-relazionale)

Memorizzazione dati senza schema relazionale

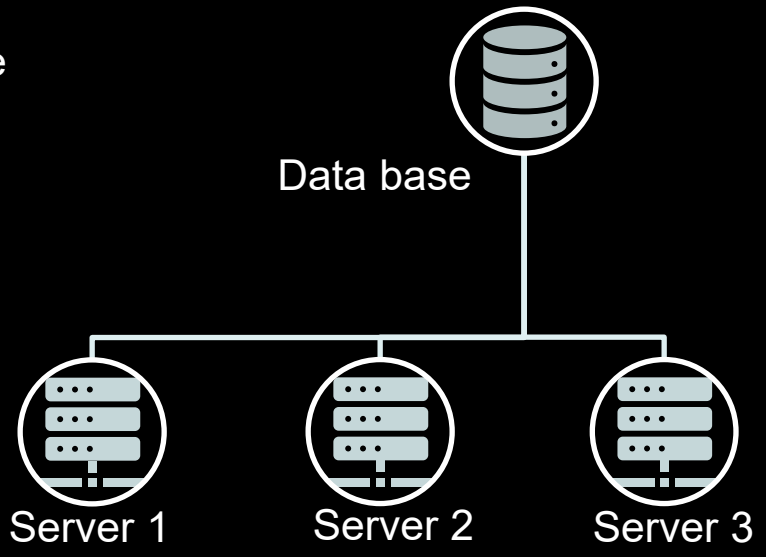
- ✓ Facile scalabilità del Sistema
- ✓ Minore uso di memoria
- ✓ Nessuna relazione tra i dati

Key	value
2345677	any
.	.
.	.
.	.

Database relazionale (SQL) Vs. NoSQL (non-relazionale)

Memorizzazione dati senza schema relazionale

- ✓ Facile scalabilità del Sistema
- ✓ Minore uso di memoria
- ✓ Nessuna relazione tra i dati
- ✓ Semplice da distribuire su diversi server



Key	Hash	value
2345677	1	any
.	2	.
.		.
.	100	.

Database relazionale (SQL) Vs. NoSQL (non-relazionale)

Memorizzazione dati senza schema relazionale

- ✓ Facile scalabilità del Sistema
- ✓ Minore uso di memoria
- ✓ Nessuna relazione tra i dati
- ✓ Semplice da distribuire su diversi server
- ✓ Semplice interrogazione (tramite la key!)

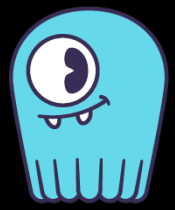
Key	Hash	value
2345677	1	any
.	2	.
.		.
.	100	.

RDBMS vs. MapReduce

	RDBMS	MapReduce
Dati	GB	PB
Accesso	Interattivo e batch	batch
Pattern di accesso	Lettura e scrittura multiple	Lettura multipla, scrittura singola
Transazioni	ACID	None
Struttura	Schema-on-write	Schema-on-read
Integrità	Alta	Bassa
Scalabilità	Non lineare	Lineare

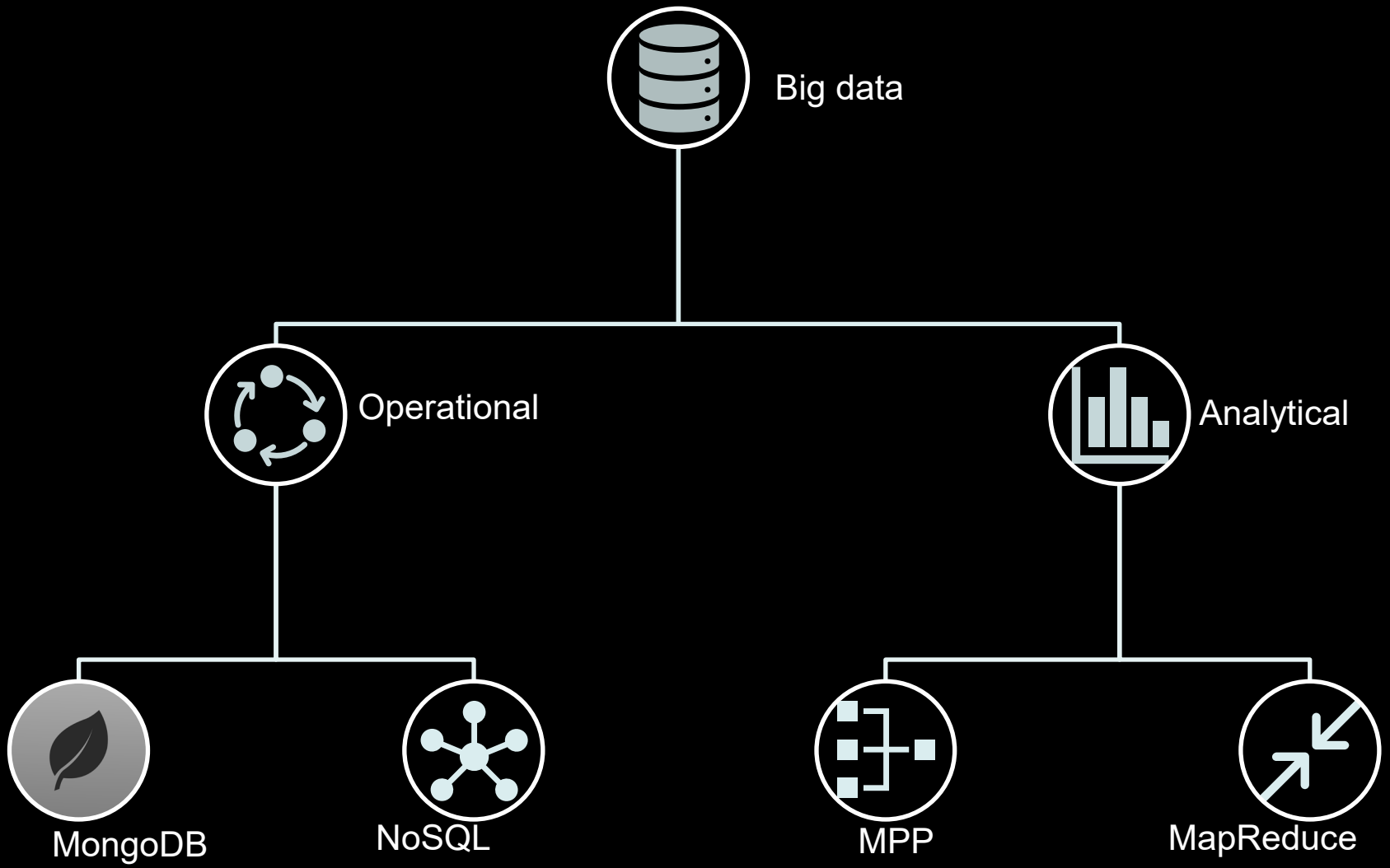
Database relazionale (SQL) Vs. NoSLQ (non-relazionale)

- Cloud based
  - DynamoDB
  - BigTable
  - CosmosDB
- Self-hosted
  - Cassandra
  - Scylla
  - CouchDB
  - MongoDB

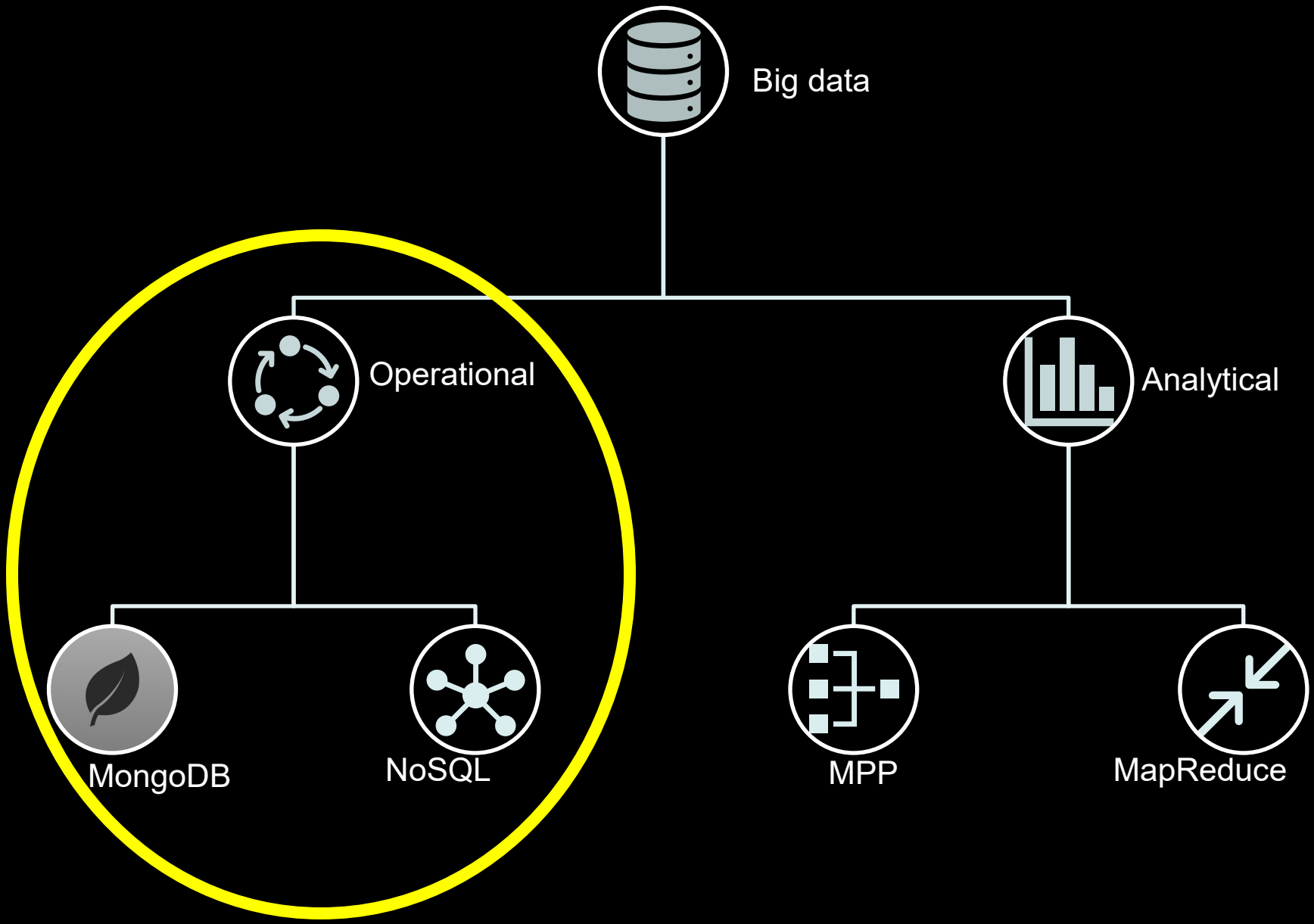


SCYLLA

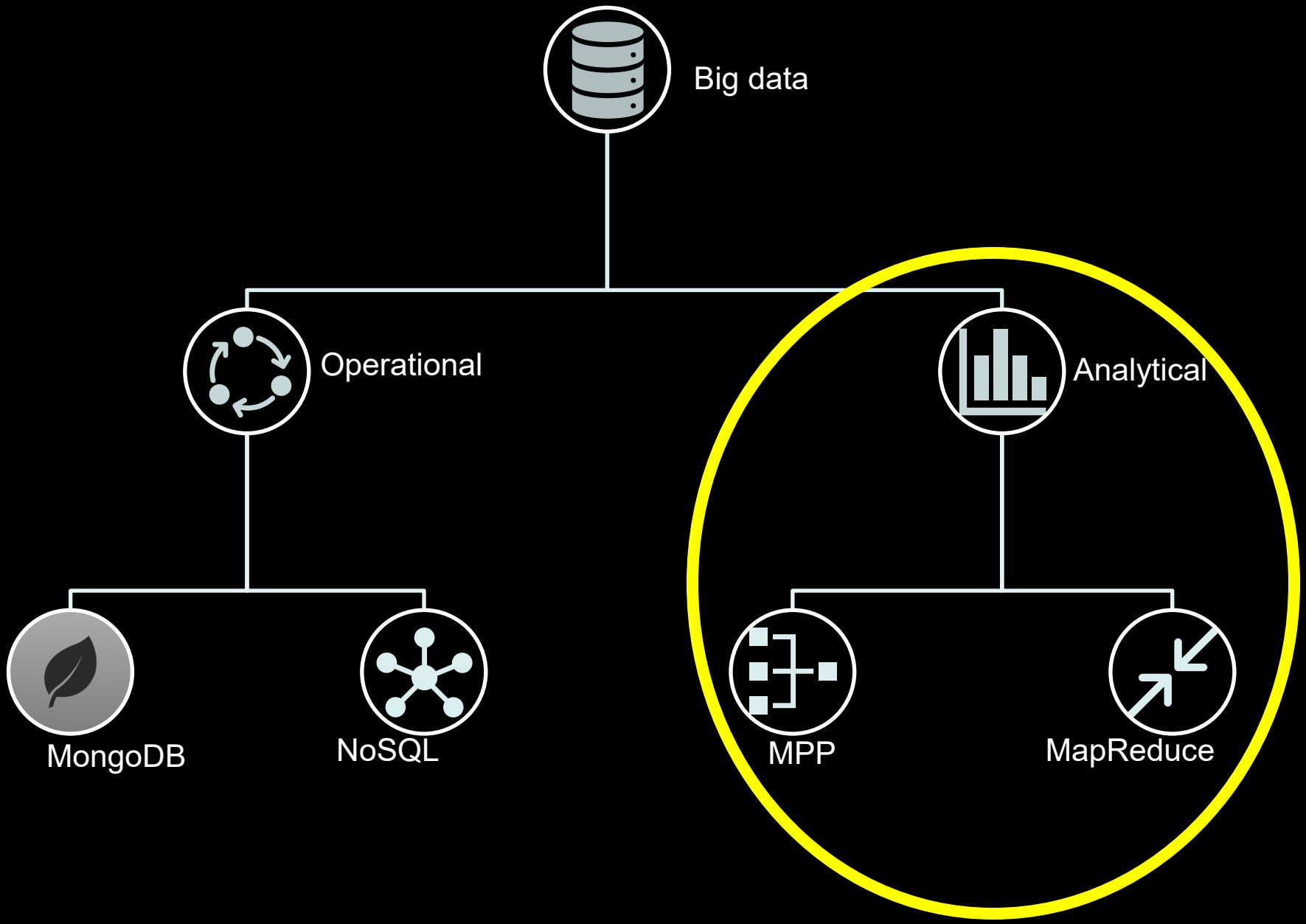
Big Data Technologies



Big Data Technologies



Big Data Technologies



Operational vs. Analytical Systems

	Operationale	Analitico
Latenza	1 ms - 100 ms	1 min - 100 min
Concorrenza	1000 - 100,000	1 - 10
Pattern di accesso	Lettura e scrittura	Lettura
Queries	Selettive	Non-selettive / iterative
Scopo dei dati	Operazionale	Retrospettiva
Utente finale	Customer	Data Scientist
Tecnologia	NoSQL	MapReduce, MPP Database

Sapere utile

IFOA
Istituto Formazione Operatori Aziendali

BIG DATA e Analisi dei Dati

Lezione 1.2 – High perfomance computing

Mauro Bellone,
Robotics and AI researcher

bellonemauro@gmail.com
www.maurobellone.com

Obiettivo

- ✓ Comprensione dei layout hardware in HPC
- ✓ Capire come si può parallelizzare una applicazione tipicamente seriale
- ✓ Capire quali sono le parti che sono importanti per le performance

Computazione parallela

Modello concettuale di collegamento di unità computazionali per risolvere un task complesso

High perfomance computing

High performance computing (HPC), o in italiano sistemi di calcolo ad elevate prestazioni, si riferisce alla pratica di aggregare potenza computazionale per risolvere problemi di calcolo complessi in maniera distribuita.

High perfomance computing

High performance computing (HPC), o in italiano sistemi di calcolo ad elevate prestazioni, si riferisce alla pratica di **aggregare** potenza computazionale per risolvere problemi di calcolo complessi in maniera **distribuita**.

High perfomance computing

High performance computing (HPC), o in italiano sistemi di calcolo ad elevate prestazioni, si riferisce alla pratica di aggregare potenza computazionale per risolvere problemi di calcolo complessi in maniera distribuita.

Aumentare la capacità di calcolo in maniera distribuita per risolvere problemi complessi che richiedono l’accesso ad una grande mole di informazione (big data)

High perfomance computing

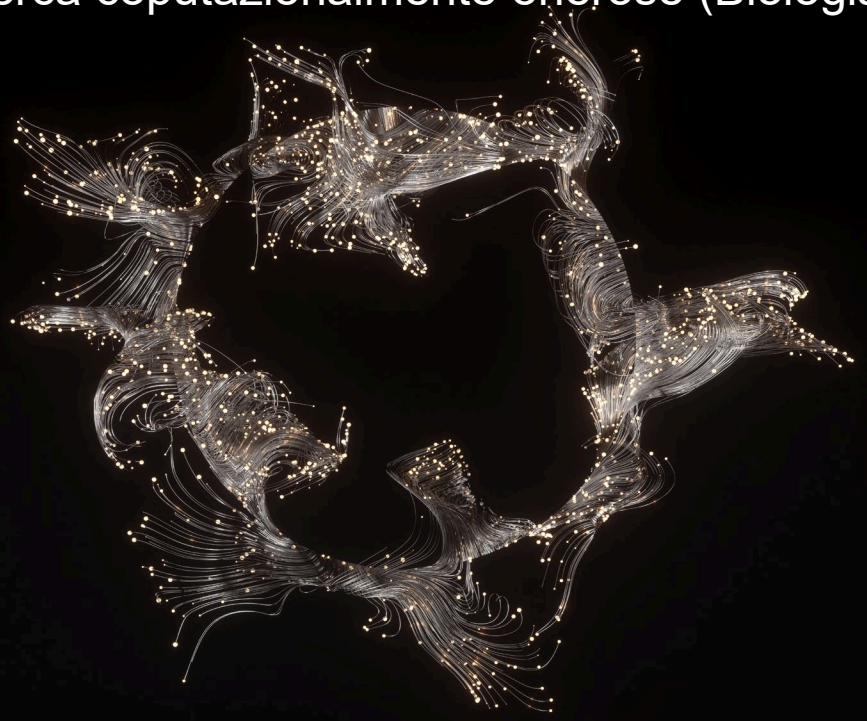
- ✓ La computazione parallela e l’HPC sono strettamente correlate

High perfomance computing

- ✓ La computazione parallela e l'HPC sono strettamente correlate
- ✓ La comprensione delle diverse architetture di programmazione parallela permette di capire come allocare risorse in maniera corretta

High perfomance computing

- ✓ La computazione parallela e l'HPC sono strettamente correlate
- ✓ La comprensione delle diverse architetture di programmazione parallela permette di capire come allocare risorse in maniera corretta
- ✓ Ci permette di capire meglio come usare technologie HPC in altre aree di ricerca coputazionalmente onerose (Biologia, medicina, AI, etc.)

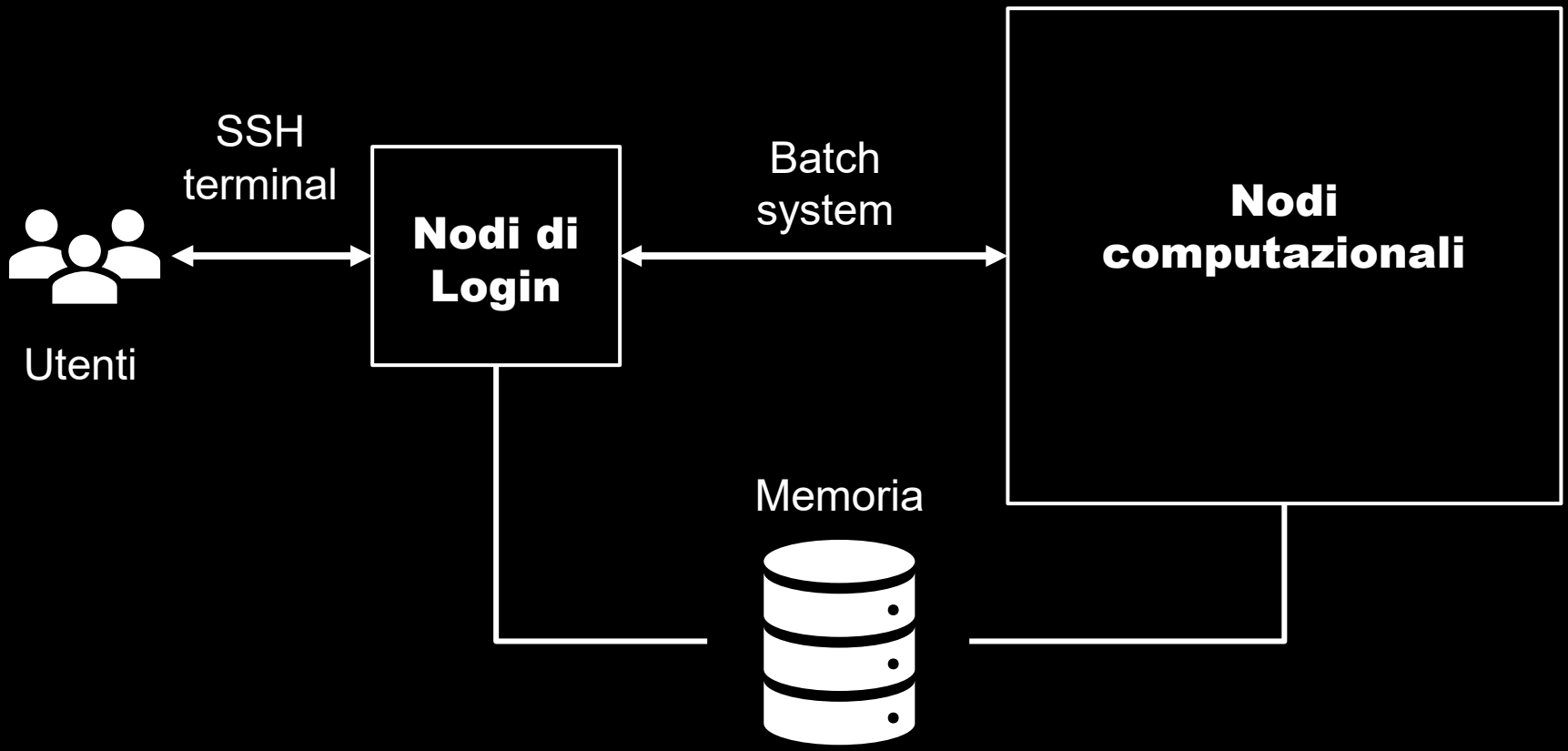


HPC e fortemente orientato alle prestazioni

Tutto è orientato alle prestazioni

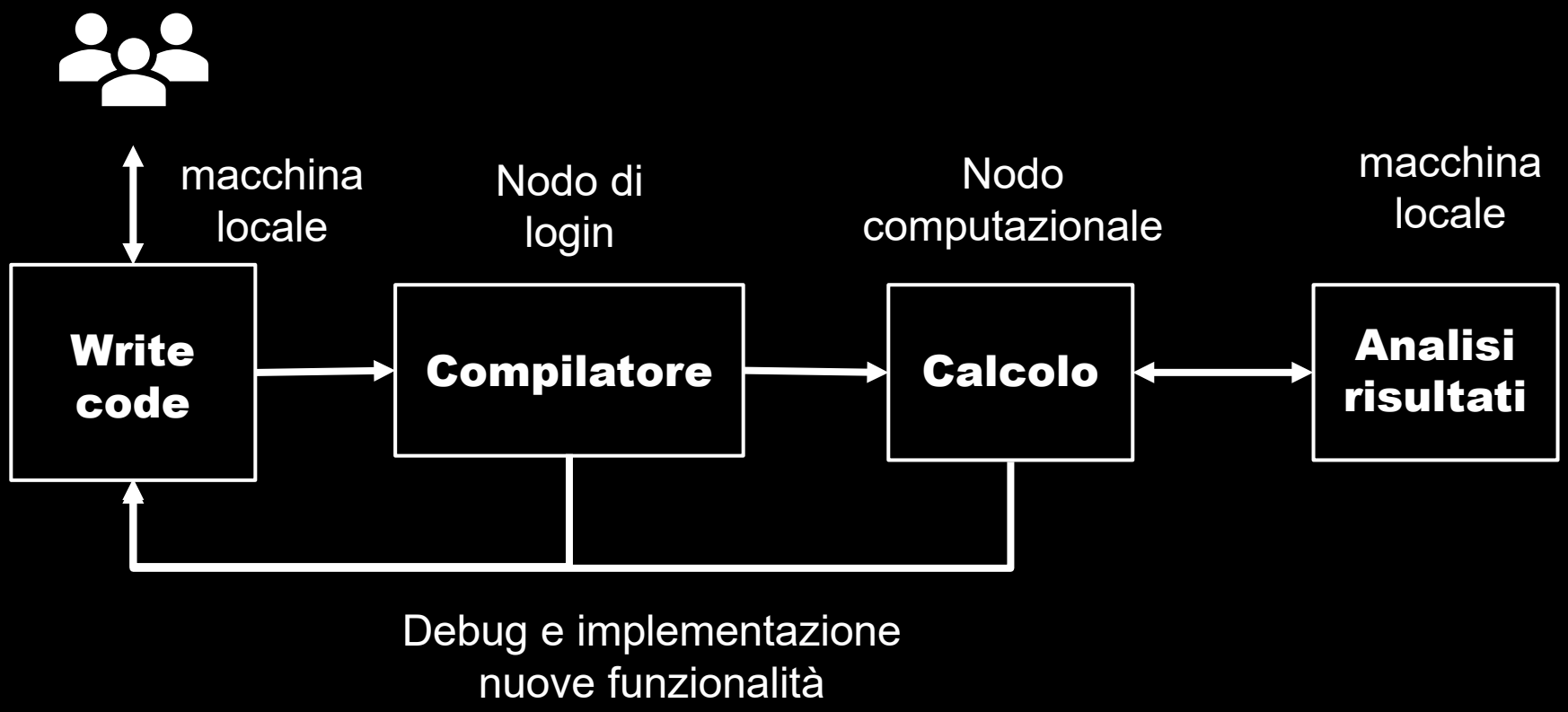
- NO connessione diretta – submit jobs via batch scheduler
- NO GUI – Connessione remota tramite SSH
- Lo stesso sistema è condiviso su molti utenti
- Le risorse sono costantemente monitorate in termini di CPU usage e Disk usage

HPC system layout



Gli utenti caricano/scaricano dati, e sottomettono
il loro task computazionale tramite SSH terminal

Usage flow


macchina locale
Nodo di login
Nodo computazionale
macchina locale
Write code
Compilatore
Calcolo
Analisi risultati
Debug e implementazione nuove funzionalità

Secure Socket Shell - SSH.

Secure socket shell è un protocolo di rete client-server con crittografia che consente la connessione e il trasferimento sicuro tra computer remoti usando una interfaccia di testo (NO GUI).

Sito ufficiale: https://www.openssh.com/

Manuale: https://man.openbsd.org/ssh

Secure Socket Shell - SSH.

Secure socket shell è un protocolo di rete client-server con crittografia che consente la connessione e il trasferimento sicuro tra computer remoti usando una interfaccia di testo (NO GUI).

Per connettersi al server remoto:

```
ssh remote_user_name@host_ip_address
```

A questo punto si ha il completo controllo della macchina remota.

Secure Socket Shell - SSH.

Secure socket shell è un protocolo di rete client-server con crittografia che consente la connessione e il trasferimento sicuro tra computer remoti usando una interfaccia di testo (NO GUI).

Per connettersi al server remoto:

```
ssh remote_user_name@host_ip_address
```

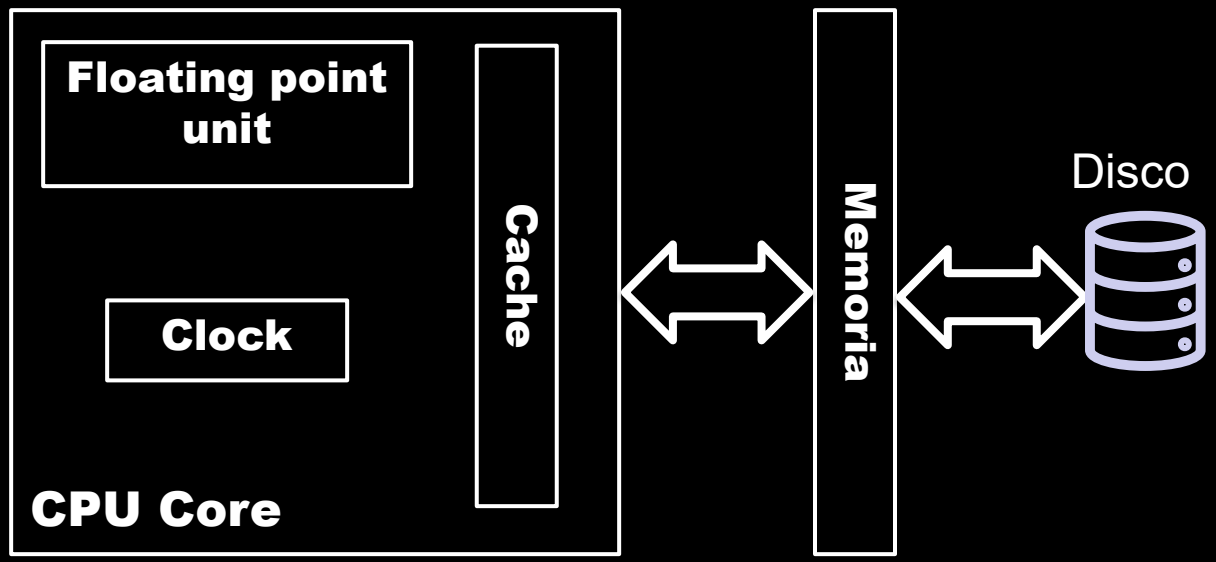
A questo punto si ha il completo controllo della macchina remota.

Il commando più utilizzato è la secure copy:

```
scp [option] user@SOURCE_HOST:file user@DESTINATION_HOST:file
```

Usando l'opzione `-r` è possibile copiare cartelle in maniera ricorsiva

Anatomia di un calcolatore classico



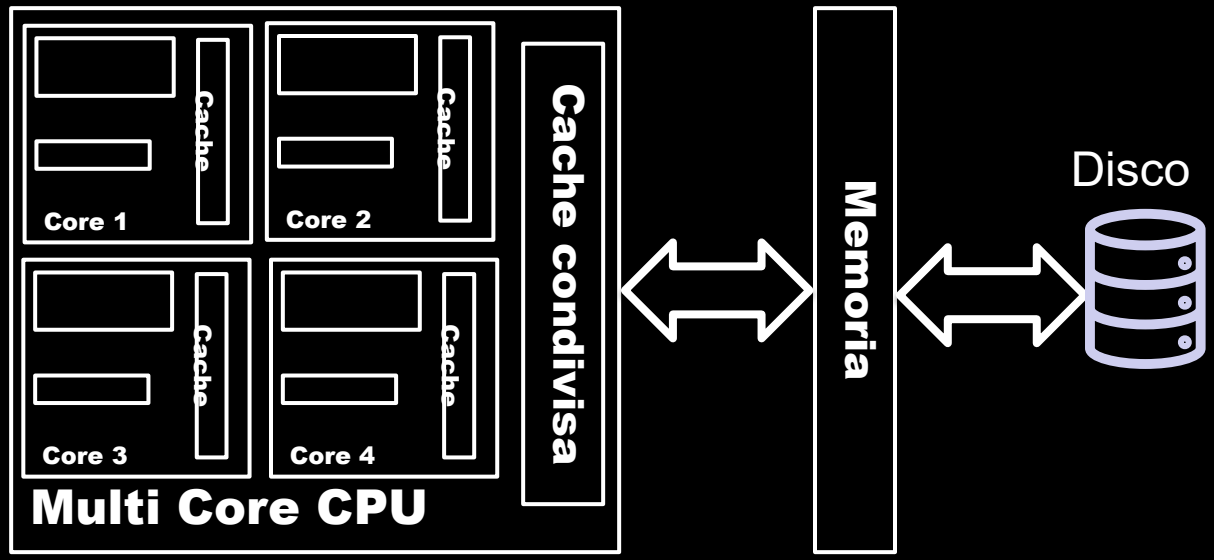
Il collo di bottiglia da decenni è la comunicazione tra CPU e memoria

Performance

Le performance di un single-core PC dipendono da:

- Velocità del clock – numero di operazioni per secondo
- Floating point unit – quanti operandi possono essere processati contemporaneamente
- Latenza della memoria
- Larghezza di banda della memoria
- Velocità di accesso ai dischi

Anatomia di un calcolatore classico - multicore



I cores condividono parte della memoria per le elaborazioni

Single Instruction stream, Multiple Data stream (SIMD) instructions + multicore

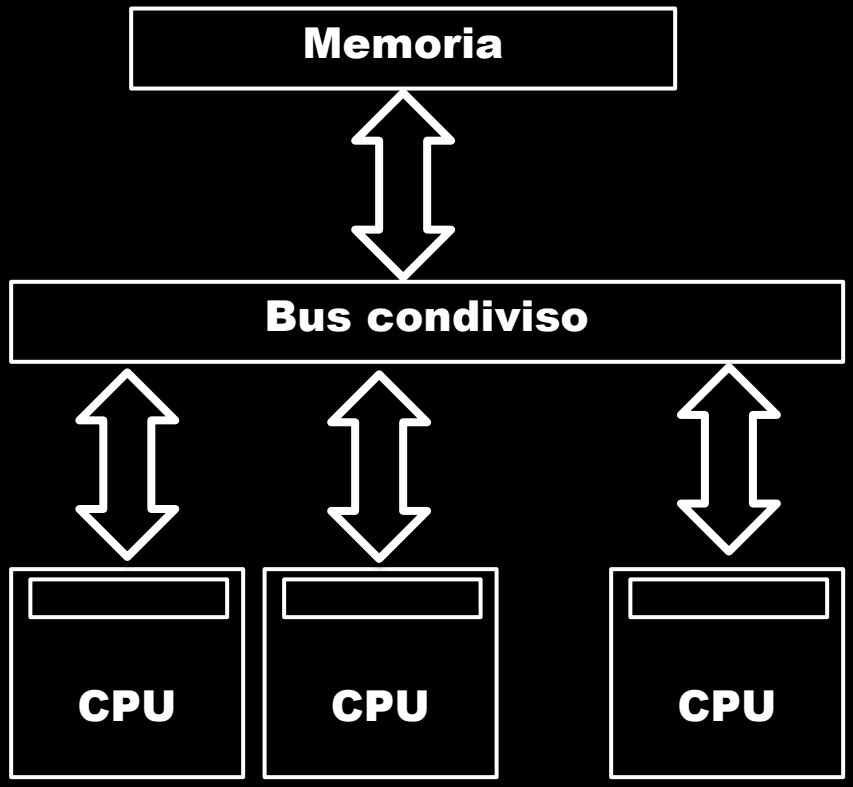
Il collo di bottiglia aumenta!!!

Architetture di high-performance computing

- Architetture a memoria condivisa
- Architetture distribuite
- Architetture ibride a memoria condivisa/distribuita

Shared memory

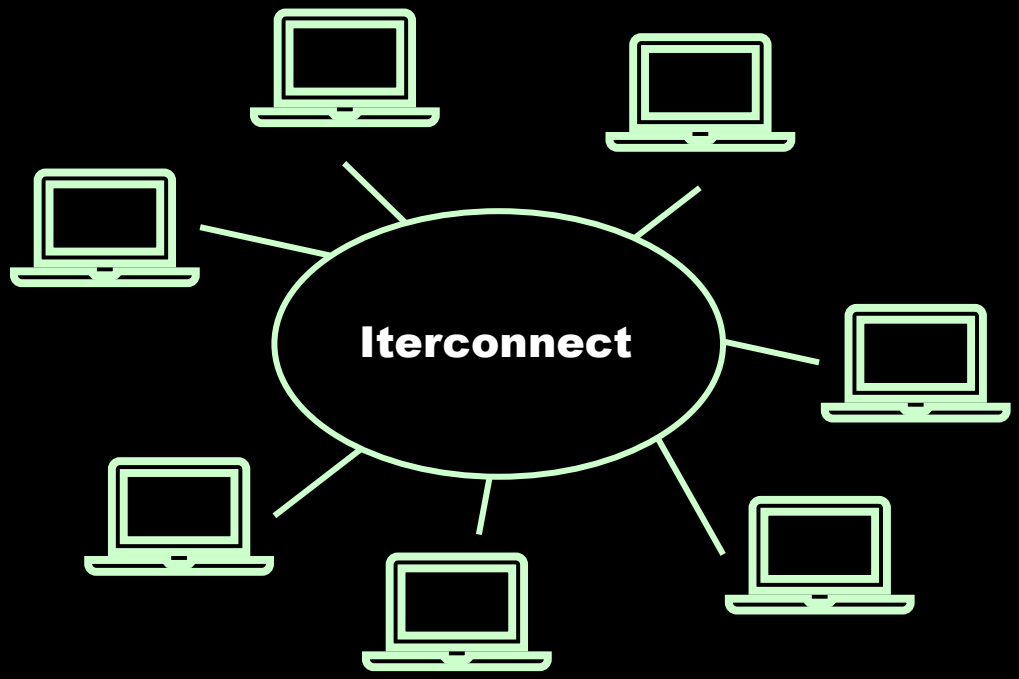
Architetture a memoria condivisa – semplici da usare, difficili da costruire



Tutti i cores hanno accesso alla stessa memoria

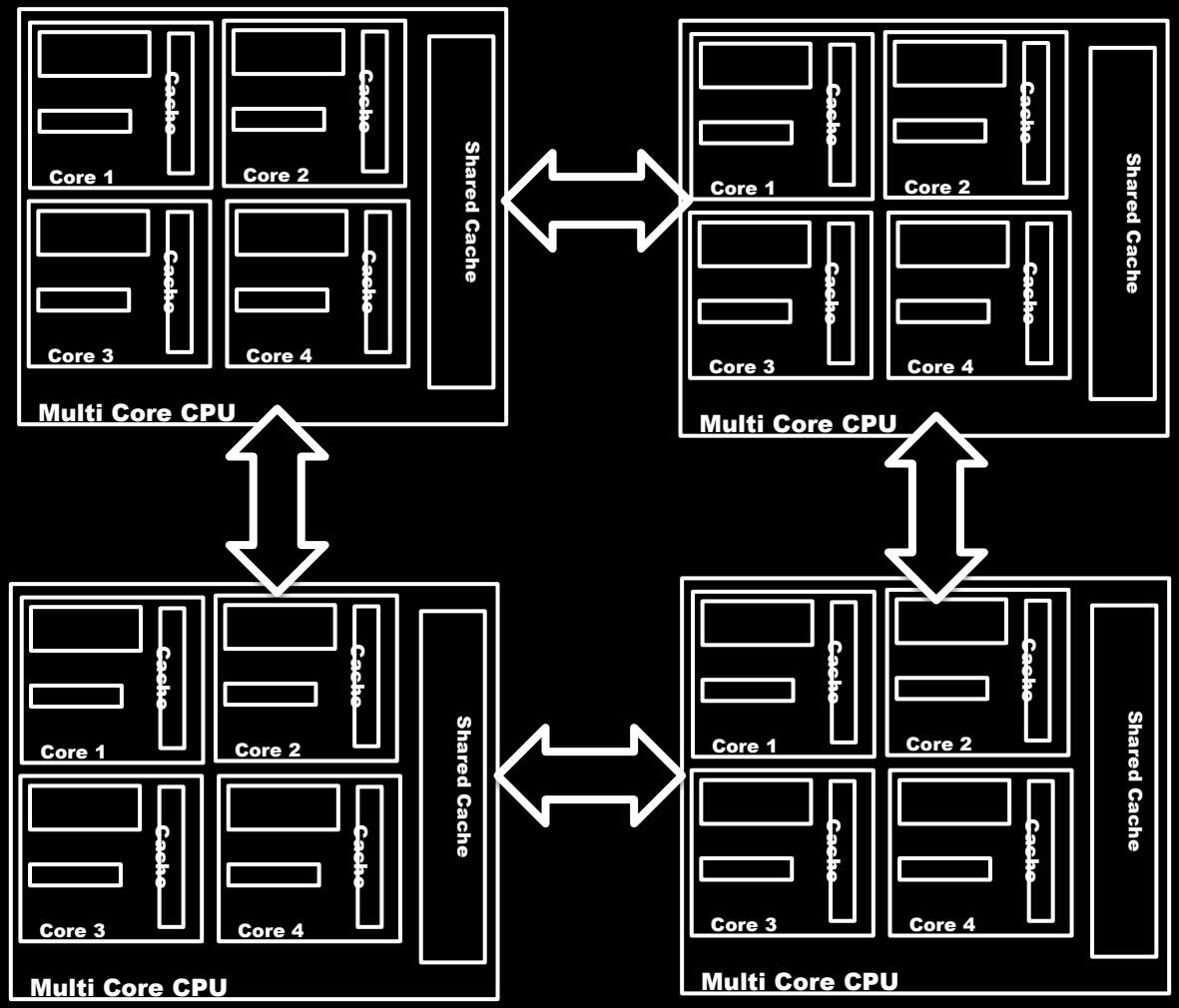
Distributed computers

Ogni computer connesso è chiamato «nodo» e sono dotati di sistemi operativi individuali



Architetture ibride

Le architetture ibride sono essenzialmente composte da sistemi multicores interconnessi tra loro



Sistemi computazionali

Unità di accelerazione sono incluse in molti sistemi HPC

- Numero di acceleratori per nodo
- Nodi connessi tramite interfaccia di connessione di rete

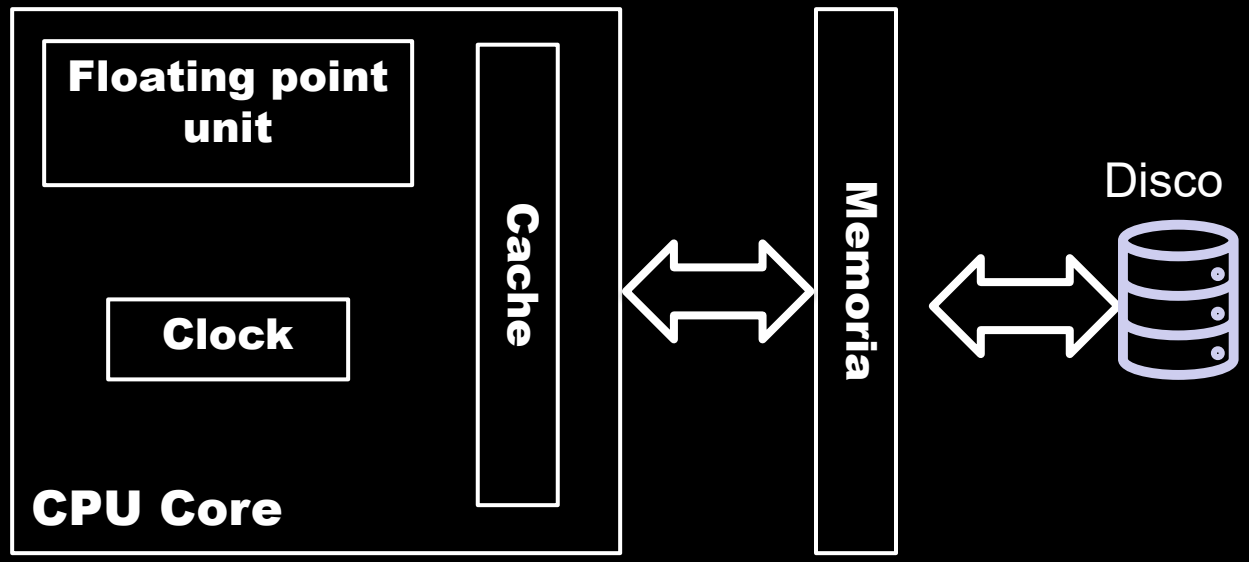
La comunicazione tra acceleratori dipende dall'hardware

- NVIDIA GPU supportano la comunicazione diretta
- AMD GPU comunicano tramite il microprocessore

La comunicazione tramite processore include una serie di computazioni aggiuntive copie dati e accessi in memoria che rallentano l sistemi

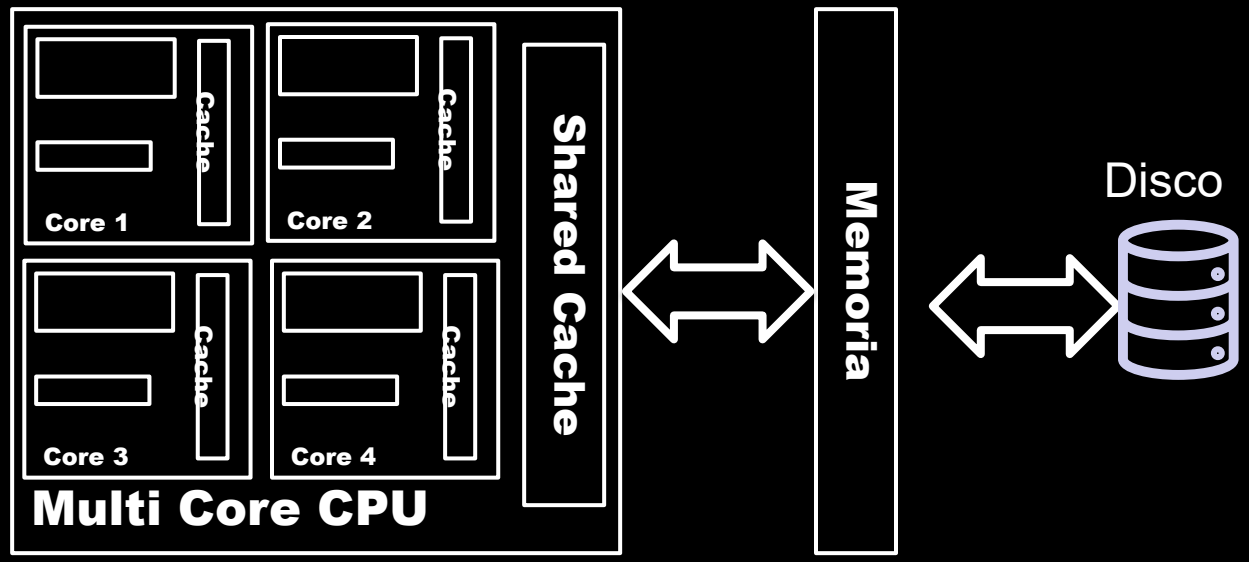
Parallelizzazione su CPU VS GPU

Entrambe hanno FPU cores che fanno somme e moltiplicazioni, la principale differenza è che le GPU fanno la stessa operazione (eseguono la stessa istruzione di basso livello) allo stesso momento su molticores diversi con input diversi



Parallelizzazione su CPU VS GPU

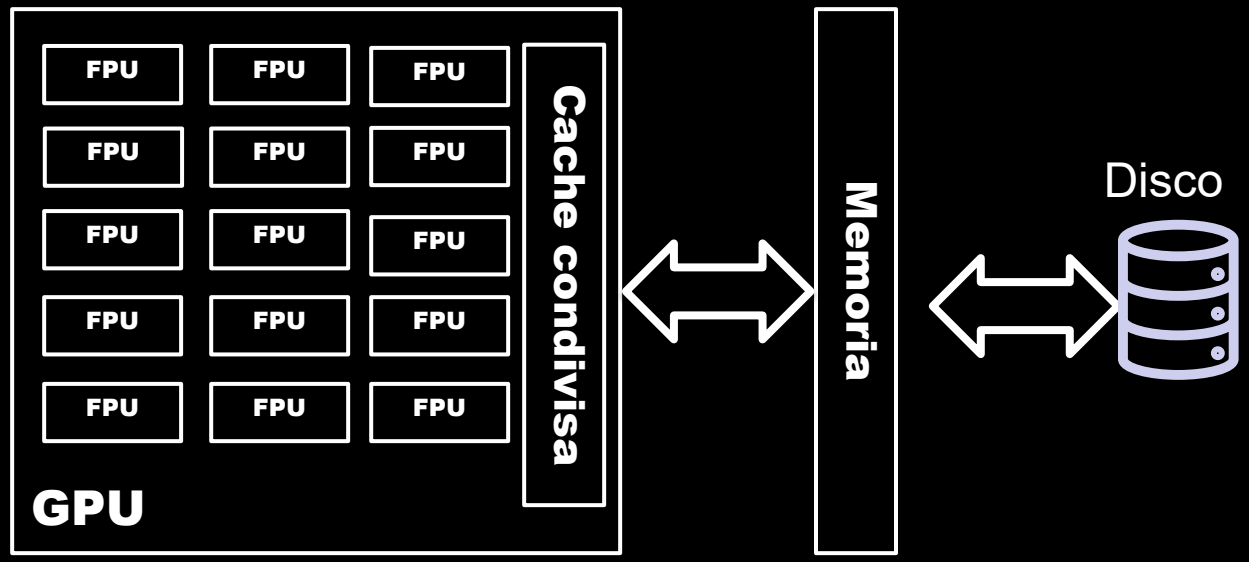
Entrambe hanno FPU cores che fanno somme e moltiplicazioni, la principale differenza è che le GPU fanno la stessa operazione (eseguono la stessa istruzione di basso livello) allo stesso momento su molticores diversi con input diversi



Obiettivo: bassa latenza

Parallelizzazione su CPU VS GPU

Entrambe hanno FPU cores che fanno somme e moltiplicazioni, la principale differenza è che le GPU fanno la stessa operazione (eseguono la stessa istruzione di basso livello) allo stesso momento su molticores diversi con input diversi



Obiettivo: alta larghezza di banda

Di fatto tutte le FPU eseguono la stessa istruzione macchina allo stesso tempo sulla cache condivisa

Efficienza computazionale

$$\text{Efficiency} = \frac{\#\text{operazioni}}{\#\text{max operazioni nel tempo T}}$$

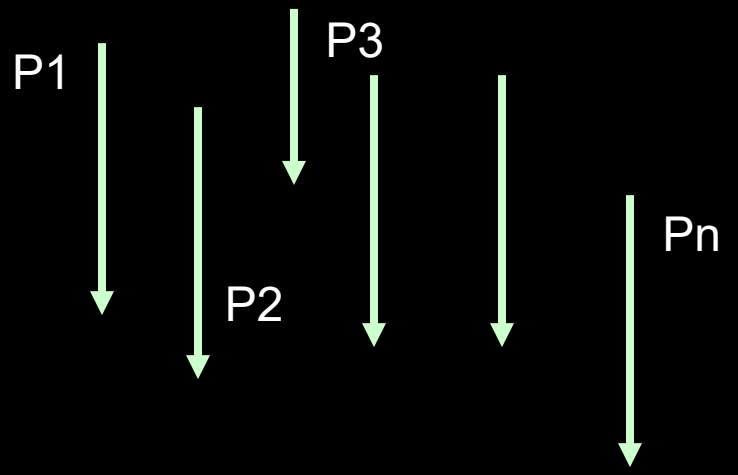
Processi e threads

Il processo è una esecuzione sequenziale di una sequenza di istruzioni

Processi eseguibili contemporaneamente sono comunemente chiamati threads

Ogni processo concorre alle risorse e accede alla memoria che gli è stata dedicate dal Sistema operative che si occupa della allocazione delle risorse condivise

La principale differenza tra processo e thread è l'accesso alla memoria, I processi accedono solo alla memoria a loro dedicata, I threads sono processi che condividono memoria



Process scheduling

Il sistema operativo ha la responsabilità di interrompere un processo per garantire le risorse ad un altro processo concorrente

- Lo specifico processo al quale è garantito l'accesso è definito dallo scheduling policy manager
- Le interruzioni possono avvenire a cadenza temporale specifica

Process scheduling

Il sistema operativo ha la responsabilità di interrompere un processo per garantire le risorse ad un altro processo concorrente

- Lo specifico processo al quale è garantito l'accesso è definito dallo scheduling policy manager
- Le interruzioni possono avvenire a cadenza temporale specifica

L'hardware può supportare in maniera nativa lo scheduling di risorse ed è comunemente chiamato SMT o symmetric multi threding e può apparire al Sistema operativo come un core aggiuntivo da utilizzare (core logico).

Il process scheduling comporta anche un peso computazionale

Scheduling policy

Hardware:

- MAX throughput
- MAX CPU/GPU usage etc.
- MIN memory access

Software:

- FCFS First come first served o anche First-in first-out (FIFO)
- Fixed priority
- Round robin
- Time table-driven
- Dynamic priority scheduling
  - Earliest deadline first (EDF)
  - Least laxity first

Scheduling policy

Hardware:

- MAX throughput
- MAX CPU/GPU usage etc.
- MIN memory access

Software:

- FCFS First come first served o anche First-in first-out (FIFO)
- Fixed priority
- Round robin
- Time table-driven
- Dynamic priority scheduling
  - Earliest deadline first (EDF)
  - Least laxity first

Scheduling policy

Hardware:

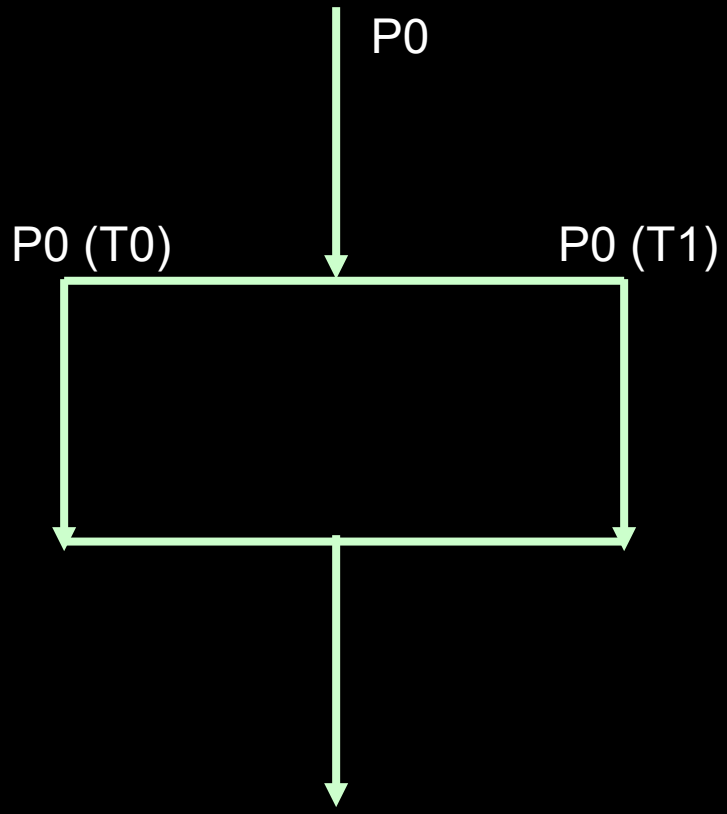
- MAX throughput
- MAX CPU/GPU usage etc.
- MIN memory access

Software:

- FCFS First come first served o anche First-in first-out (FIFO)
- Fixed priority
- Round robin
- Time table-driven
- Dynamic priority scheduling
  - Earliest deadline first (EDF)
  - Least laxity first

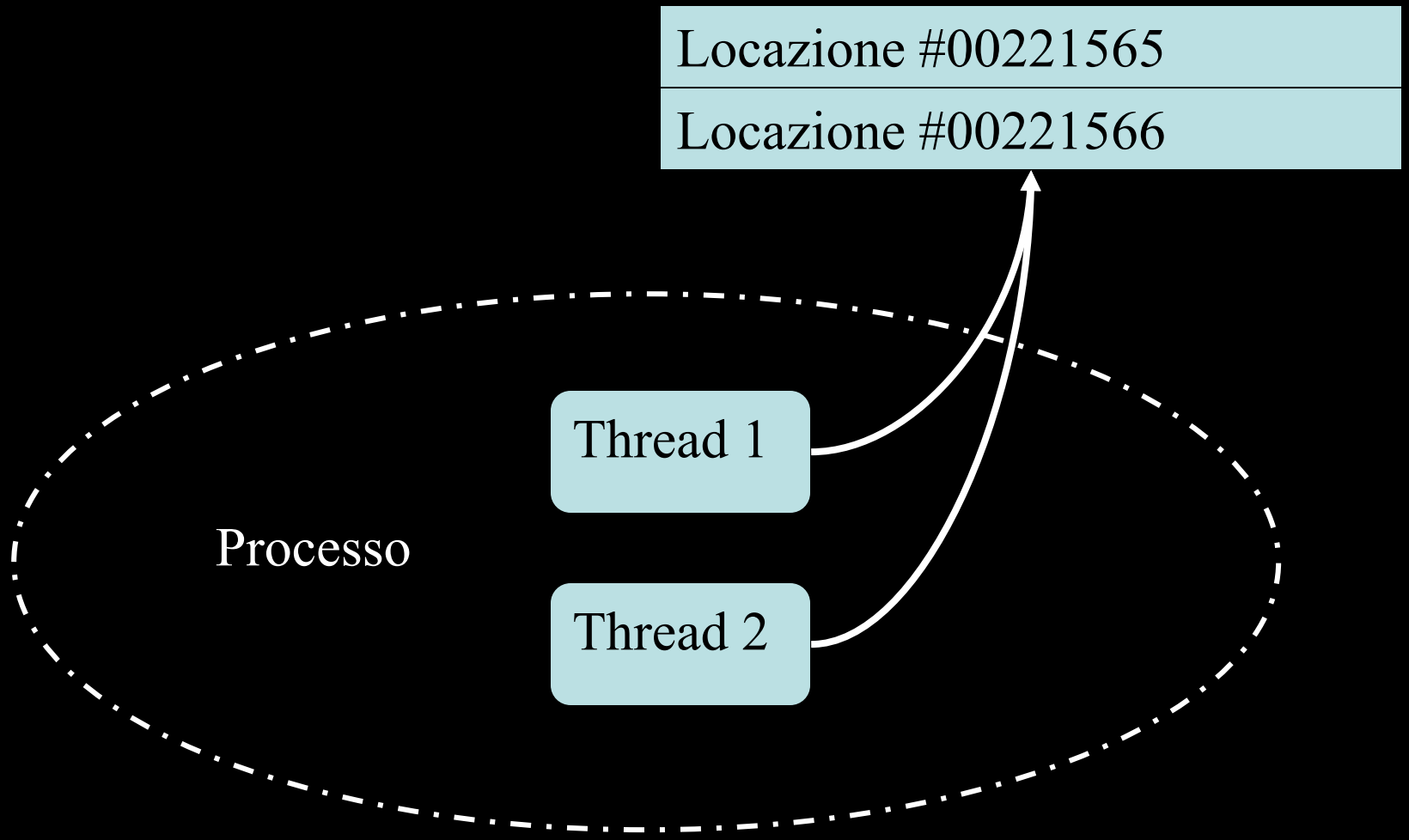
Threads

Per la maggior parte delle applicazioni ogni processo ha un singolo thread, ma per sistemi multiprocessore è sempre più comune che un processo contenga più threads al fine di parallelizzare delle operazioni e velocizzarne l'esecuzione



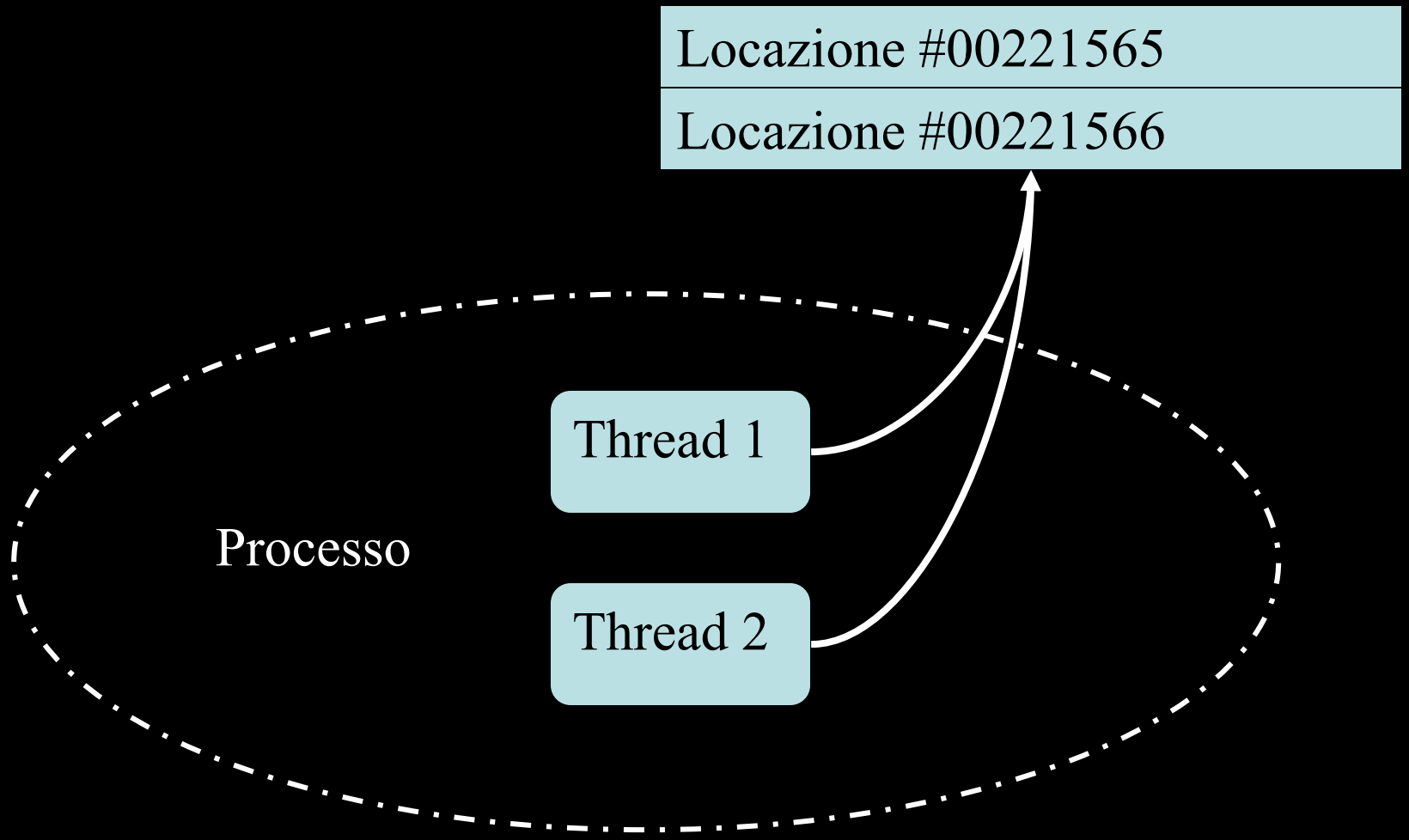
Threads

- Tutti i threads in un processo accedono alla stessa memoria



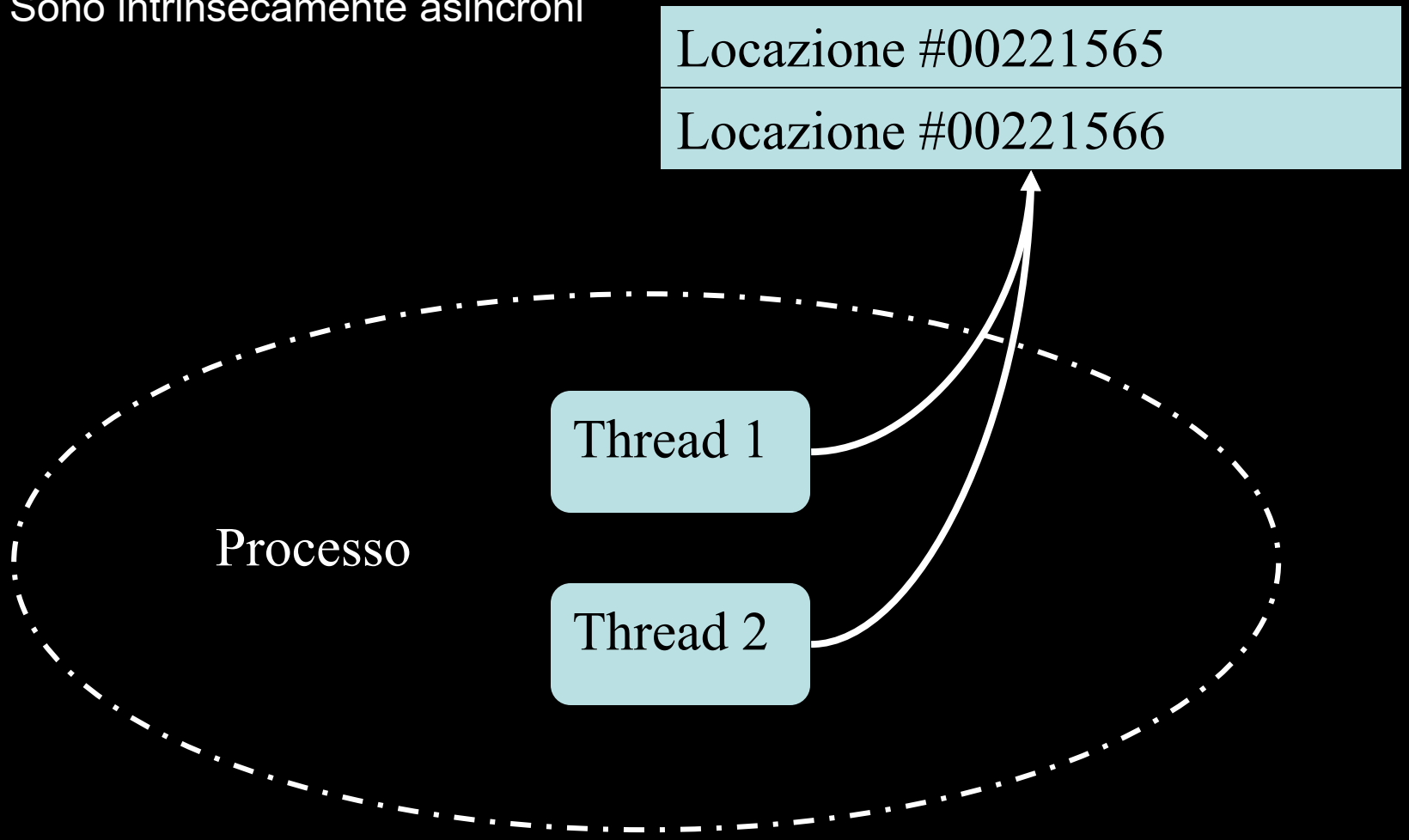
Threads

- Tutti i threads in un processo accedono alla stessa memoria
- Possono operare in parallelo sugli stessi dati per velocizzare l'applicazione



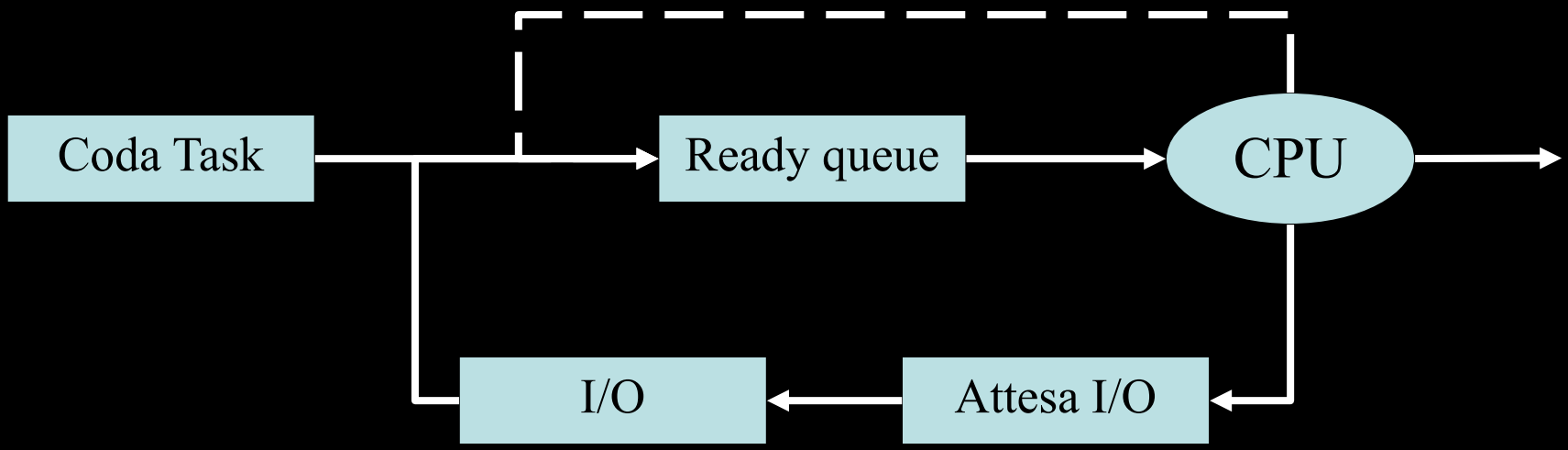
Threads

- Tutti i threads in un processo accedono alla stessa memoria
- Possono operare in parallelo sugli stessi dati per velocizzare l'applicazione
- Sono intrinsecamente asincroni



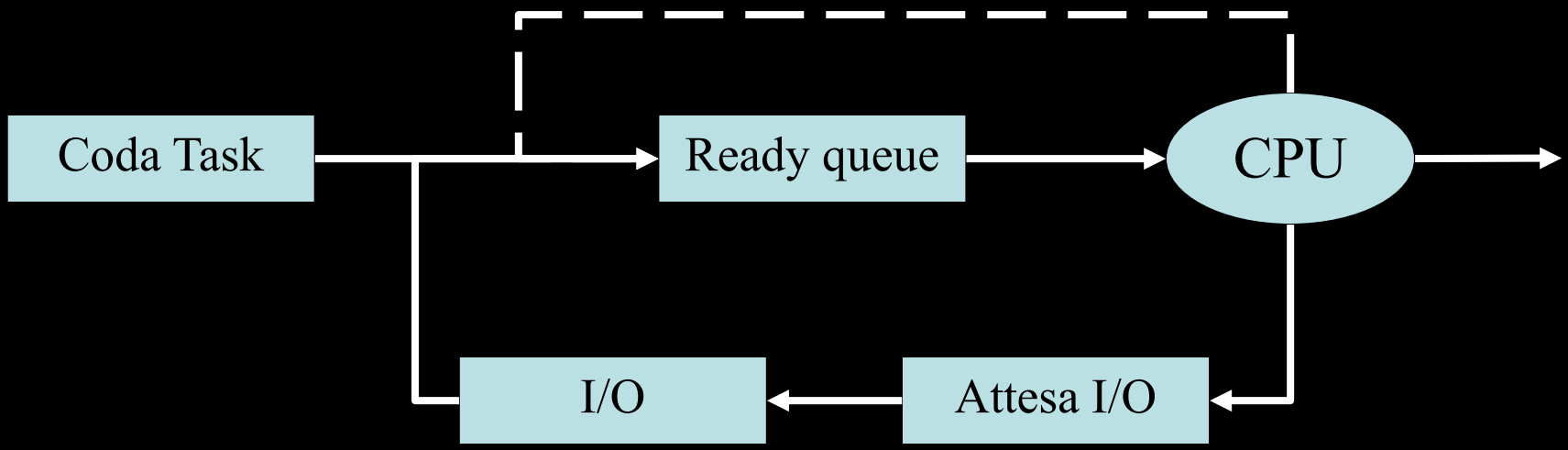
Threads

- Tutti i threads in un processo accedono alla stessa memoria
- Possono operare in parallelo sugli stessi dati per velocizzare l'applicazione
- Sono intrinsecamente asincroni
- Lo scheduling policy manager del OS tiene traccia dei threads in esecuzione



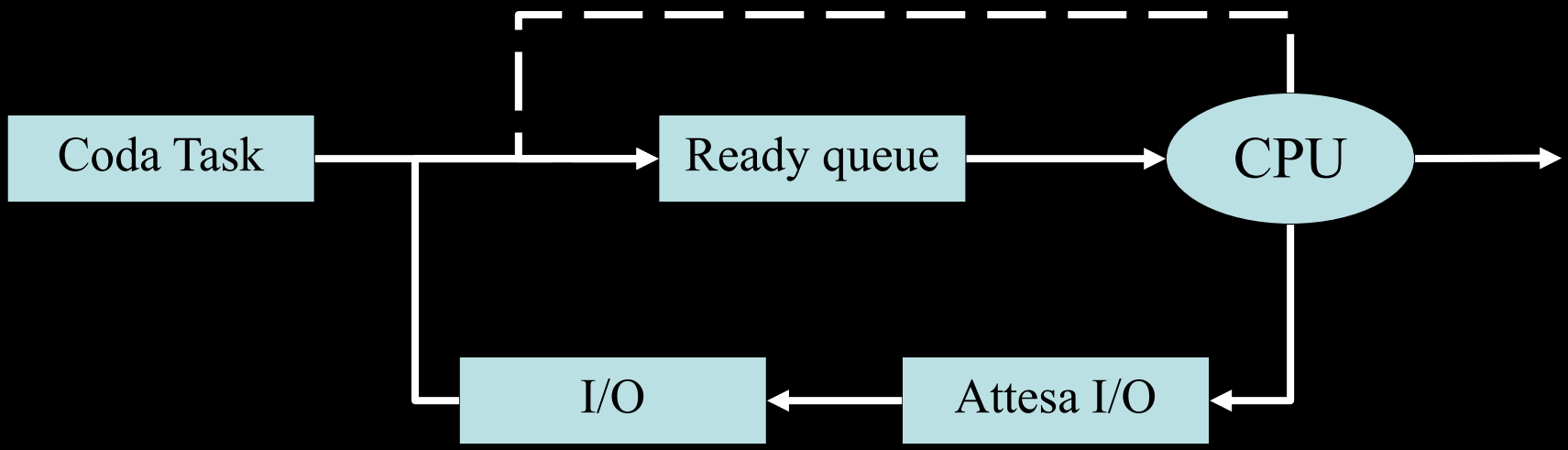
Threads

- Tutti i threads in un processo accedono alla stessa memoria
- Possono operare in parallelo sugli stessi dati per velocizzare l'applicazione
- Sono intrinsecamente asincroni
- Lo scheduling policy manager del OS tiene traccia dei threads in esecuzione
- Tipicamente ogni thread è eseguito su un core (non sempre)



Threads

- Tutti i threads in un processo accedono alla stessa memoria
- Possono operare in parallelo sugli stessi dati per velocizzare l'applicazione
- Sono intrinsecamente asincroni
- Lo scheduling policy manager del OS tiene traccia dei threads in esecuzione
- Tipicamente ogni thread è eseguito su un core (non sempre)
- Switchare tra threads può essere più rapido che switchare tra processi



Secure Socket Shell - SSH.

Tutorial ----

Futuro per HPC

- ✓ Aumento dei dati da processare (big data)
- ✓ Aumento della larghezza di banda memoria – CPU
- ✓ Aumento del numero di cores sulle macchine standalone
- ✓ Aumento della parallelizzazione delle applicazioni (nuovi algoritmi)
- ✓ Aumento del software di gestione e manipolazione dati

Sapere utile

IFOA
Istituto Formazione Operatori Aziendali

BIG DATA e Analisi dei Dati

Tutorial 1.1 – ssh e monitoraggio risorse

Mauro Bellone,
Robotics and AI researcher

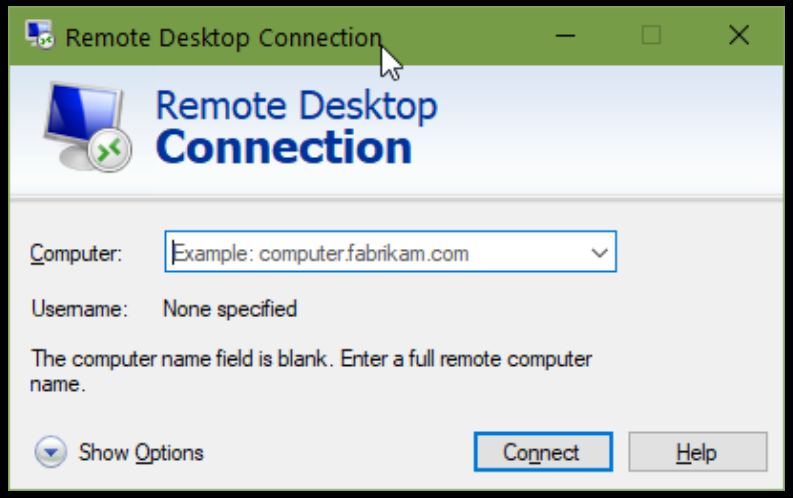
bellonemauro@gmail.com
www.maurobellone.com

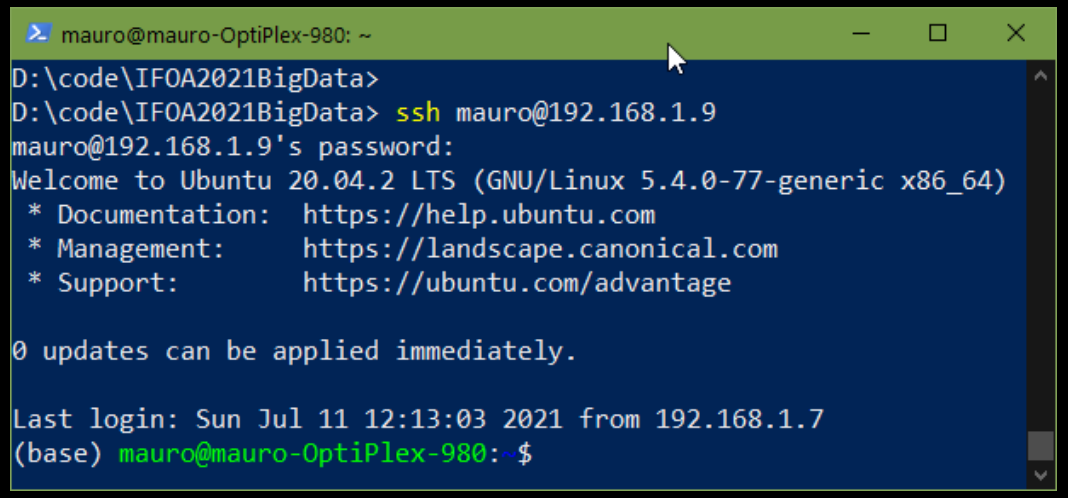
Obiettivo

- ✓ Acquisire familiarità con l’utilizzo di ssh
- ✓ Acquisire familiarità con il monitoraggio risorse su cluster remoto

Connessione remota - Secure Socket Shell (SSH)

- Remote desktop protocol (RDP)
- Secure Socket Shell (SSH)





Secure Socket Shell - SSH.

Secure socket shell è un protocolo di rete client-server con crittografia che consente la connessione e il trasferimento sicuro tra computer remoti usando una interfaccia di testo (NO GUI).

Sito ufficiale: https://www.openssh.com/

Manuale: https://man.openbsd.org/ssh

Secure Socket Shell - SSH.

Secure socket shell è un protocolo di rete client-server con crittografia che consente la connessione e il trasferimento sicuro tra computer remoti usando una interfaccia di testo (NO GUI).

Per connettersi al server remoto:

```
ssh remote_user_name@host_ip_address
```

A questo punto si ha il completo controllo della macchina remota.
Un comando comunemente utilizzato è la secure copy:

```
scp [option] user@SOURCE_HOST:file user@DESTINATION_HOST:file
```

Usando l'opzione `-r` è possibile copiare cartelle in maniera ricorsiva

Monitoraggio delle risorse tramite HTOP

HTOP è un software open source per il monitoraggio delle risorse su terminal

Se HTOP non è installato

```
sudo apt-get install htop
```

Per eseguire semplicemente eseguire il comando

```
htop
```

Website: https://htop.dev/

Esecuzione codice CPU overflow e memory overflow

Esecuzione

Esecuzione codice thread

Esecuzione

http://cpp.sh/

Sapere utile

IFOA
Istituto Formazione Operatori Aziendali

BIG DATA e Analisi dei Dati

Lezione 1.3 – Ecosistema hadoop

Mauro Bellone,
Robotics and AI researcher

bellonemauro@gmail.com
www.maurobellone.com

Obiettivo

- ✓ Fornire una introduzione sul framework Hadoop
- ✓ Comprensione dell’approccio Hadoop al big data
- ✓ Descrizione dei principali tools aggiuntivi per l’elaborazione di big data

Approccio tradizionale

Approccio tradizionale

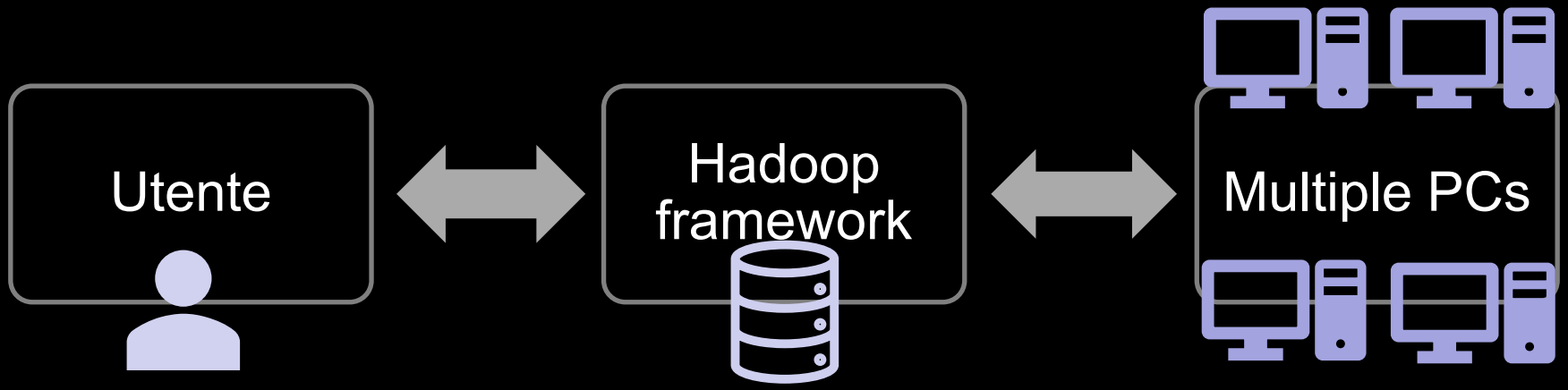
Limiti:

- Capacità limitata
- Mancanza di scalabilità
- Collo di bottiglia nel processamento dati

Approccio di google

Dividere un task da eseguire su molti PC di piccole dimensioni invece di usare un server centrale

Approccio hadoop



Usando questo approccio Douglas Cutting e il suo team di sviluppo hanno implementato un sistema opensource chiamato Hadoop

Hadoop

The Origin of the Name "Hadoop"

The name Hadoop is not an acronym; it's a made-up name. The project's creator, Doug Cutting, explains how the name came about:

> The name my kid gave a stuffed yellow elephant. Short, relatively easy to spell and pronounce, meaningless, and not used elsewhere: those are my naming criteria. Kids are good at generating such. Googol is a kid's term.

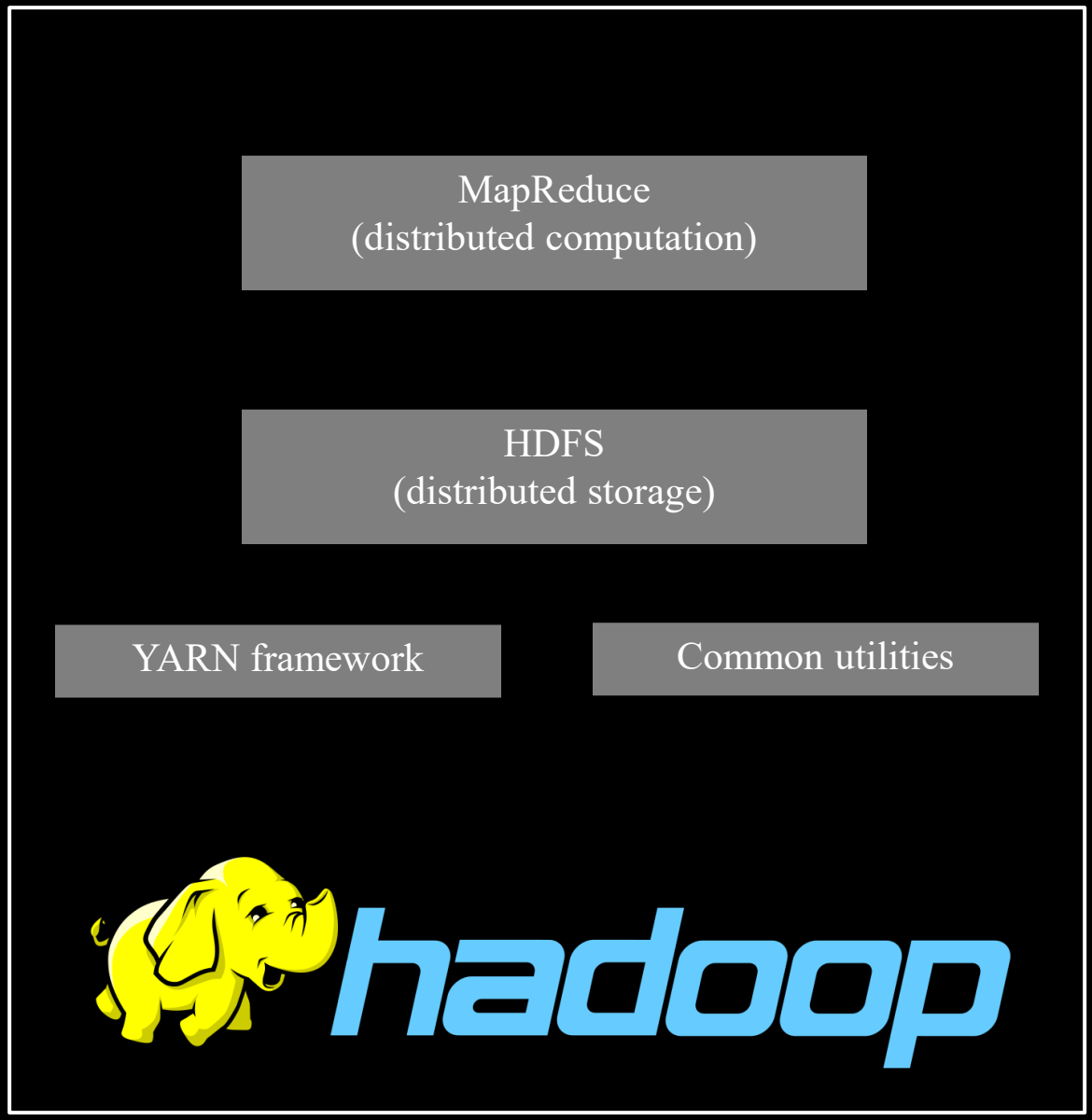
Projects in the Hadoop ecosystem also tend to have names that are unrelated to their function, often with an elephant or other animal theme ("Pig," for example). Smaller components are given more descriptive (and therefore more mundane) names. This is a good principle, as it means you can generally work out what something does from its name. For example, the namenode[8] manages the filesystem namespace.

Hadoop in a nutshell

Hadoop è un framework open-source Apache scritto in java che permette il processamento distribuito di grandi dataset su cluster di computer che usano modelli di programmazione semplici.

Il framework Hadoop funziona in un ambiente che fornisce la memorizzazione e la computazione distribuita su cluster ed è progettato per essere facilmente scalabile a migliaia di computers anche offrendo computazione locale.

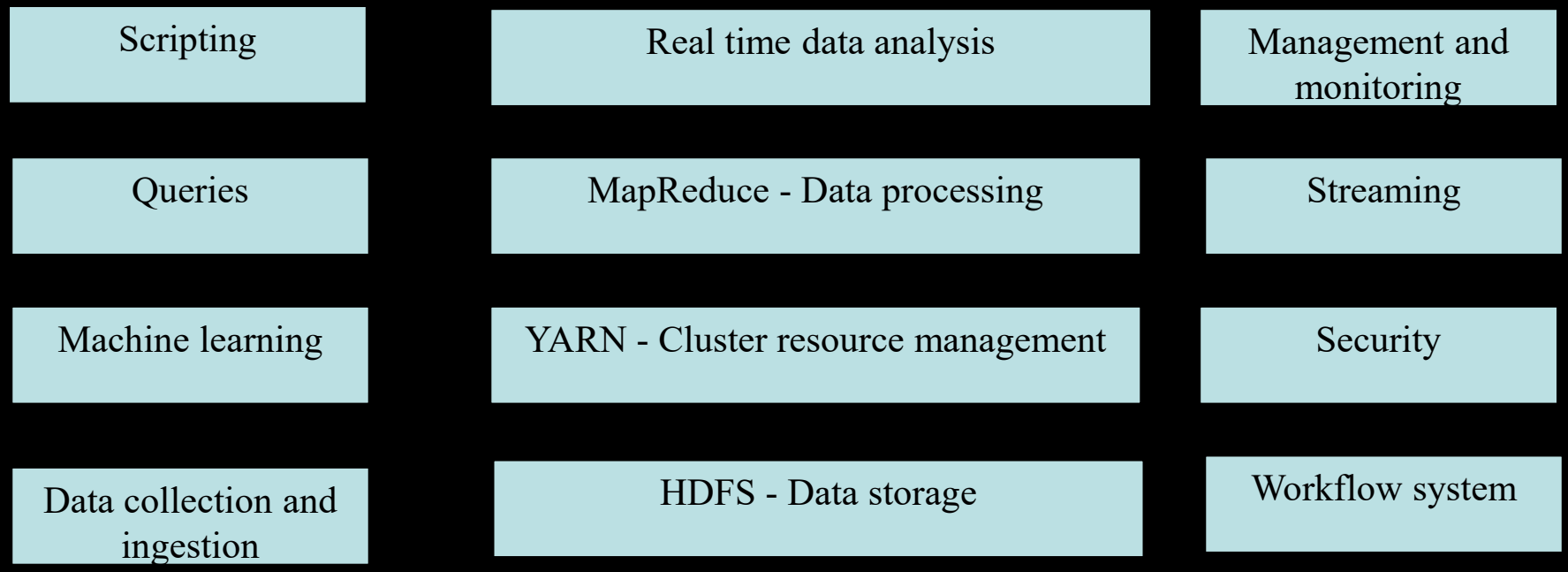
Componenti di Hadoop



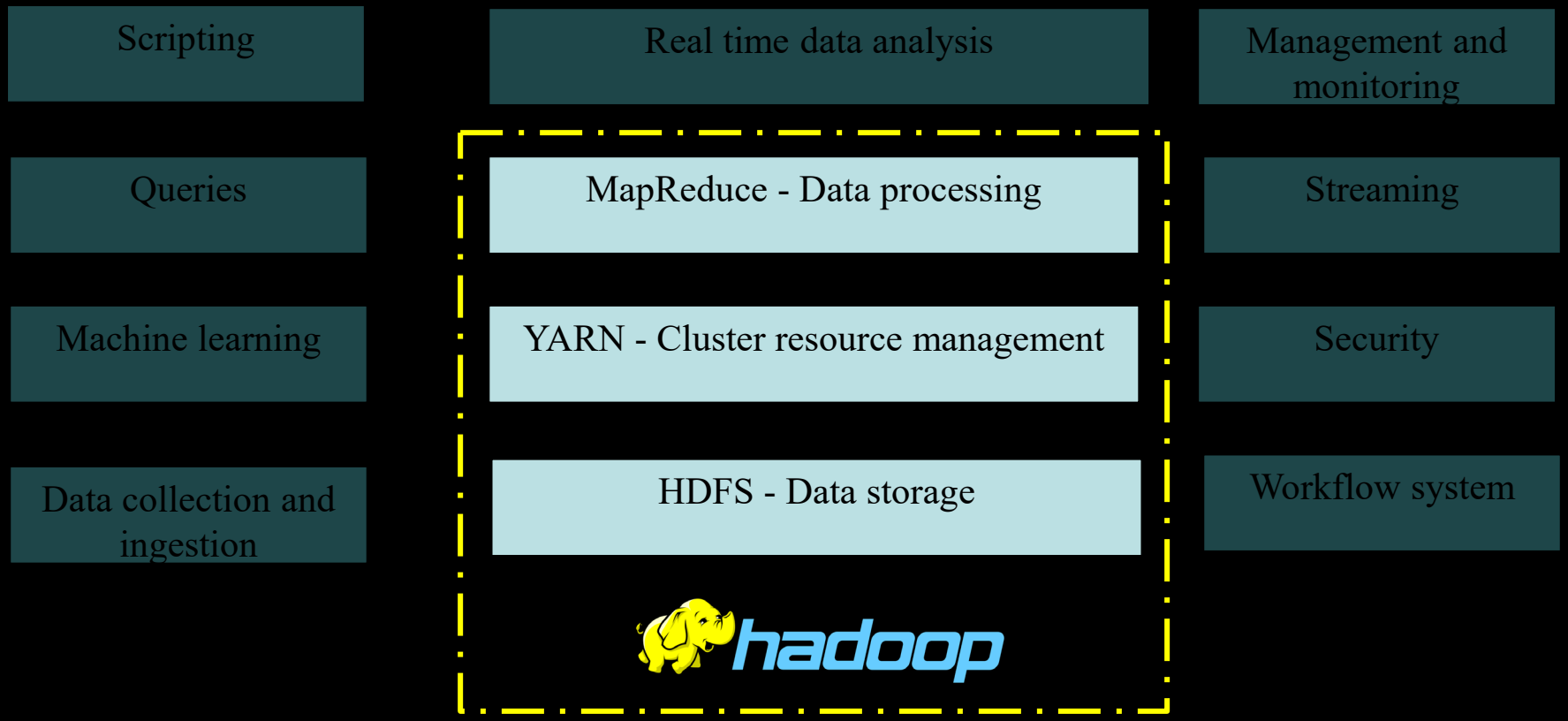
Vantaggi di Hadoop

- Permette di scrivere e testare semplicemente sistemi distribuiti
- È efficiente e distribuisce I dati automaticamente utilizzando la parallelizzazione dei CPU cores
- FTHA - fault-tolerance and high availability
- I server si possono aggiungere e rimuovere in maniera dinamica senza interruzione di servizio
- Compatibile con tutte le machine java-based
- E’ open source

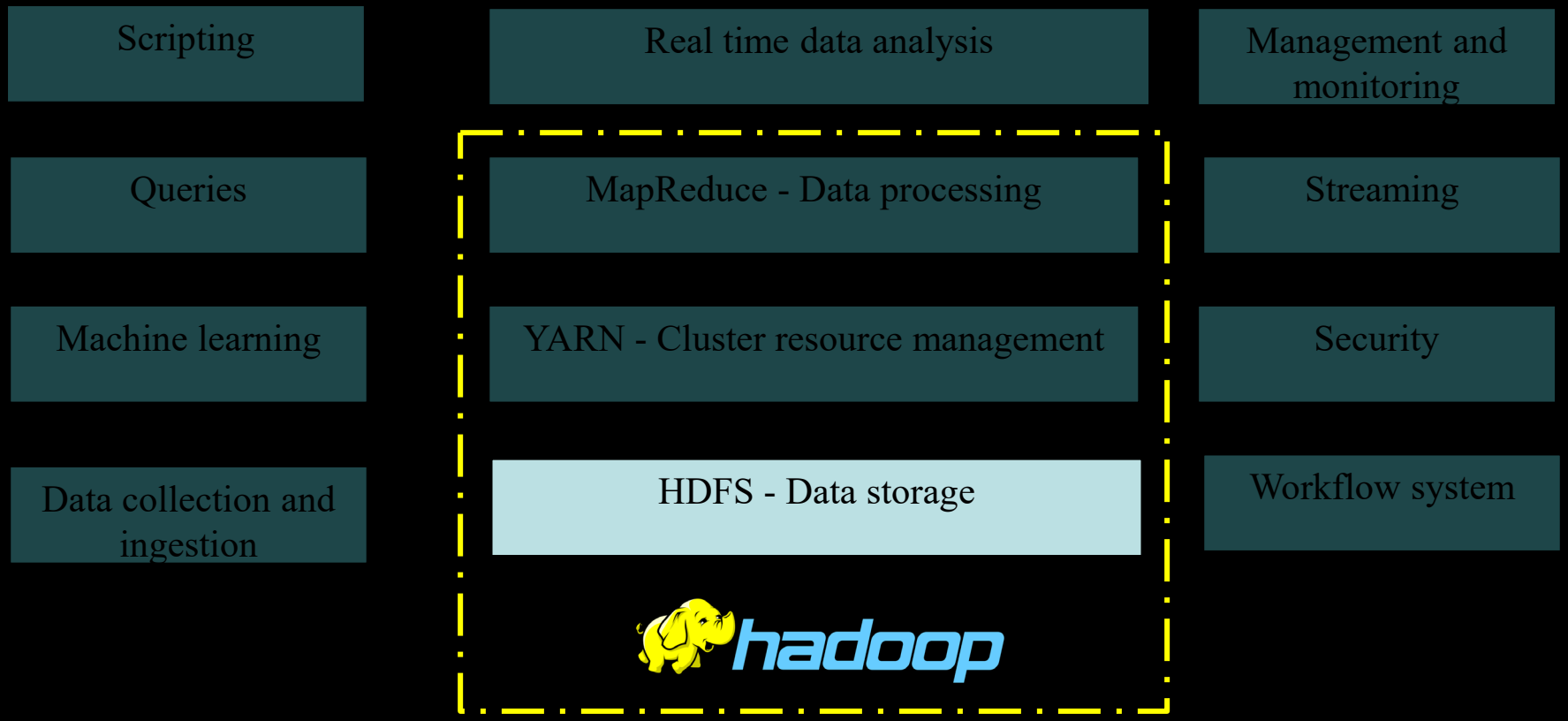
Hadoop - Ecosystem



Hadoop - Ecosystem

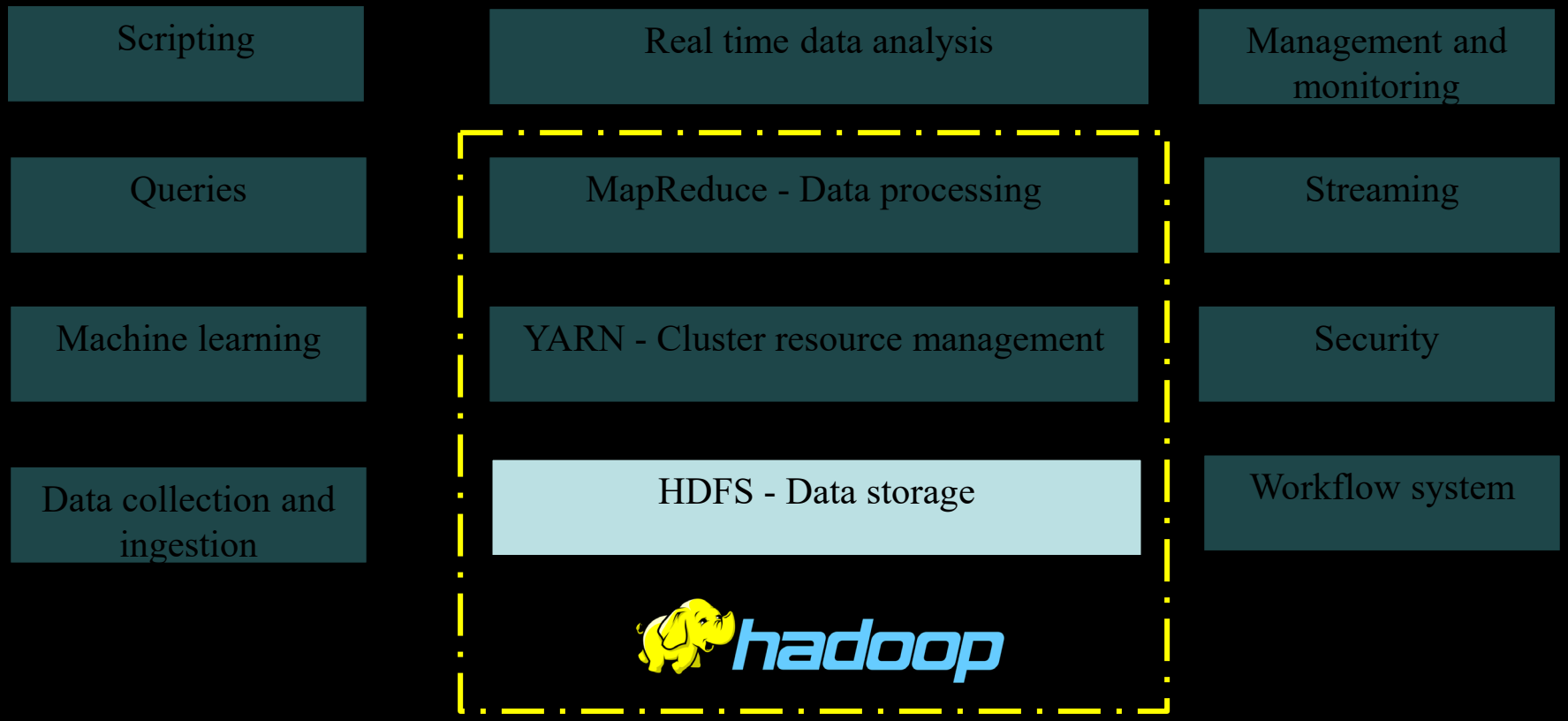


Hadoop – Data storage

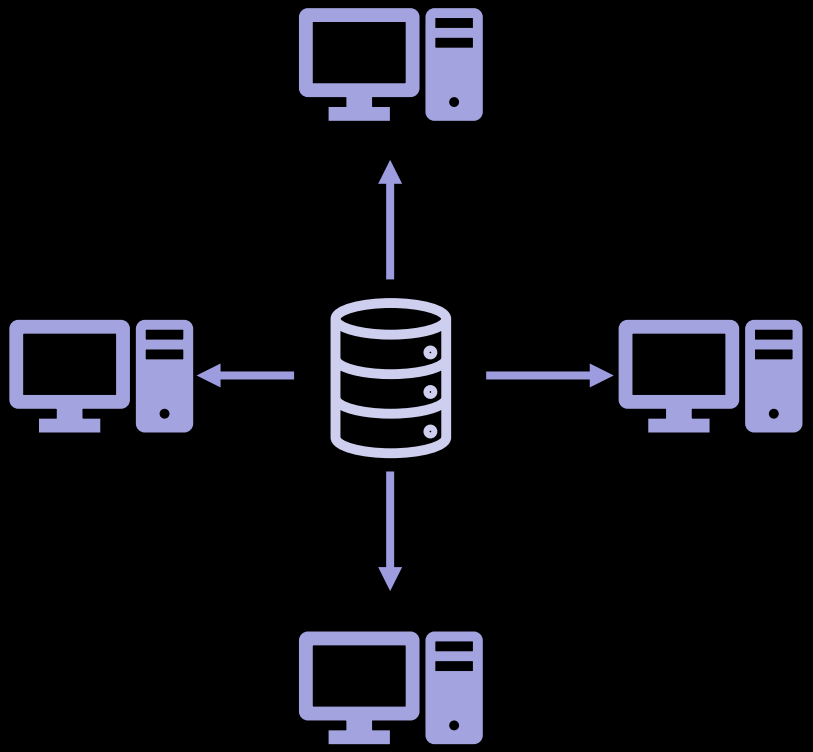


Hadoop – Data storage

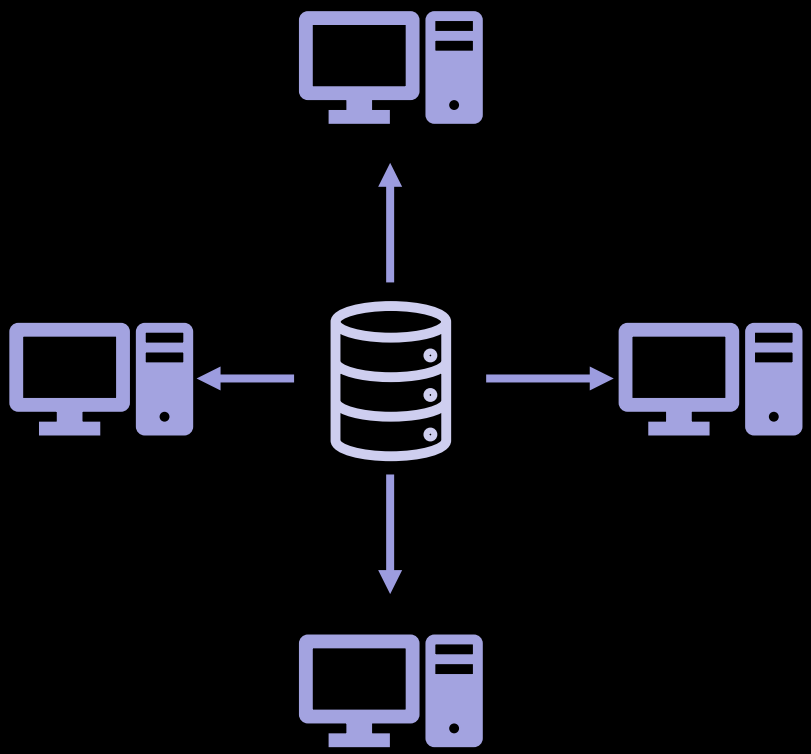
HDFS – Hadoop distributed file system

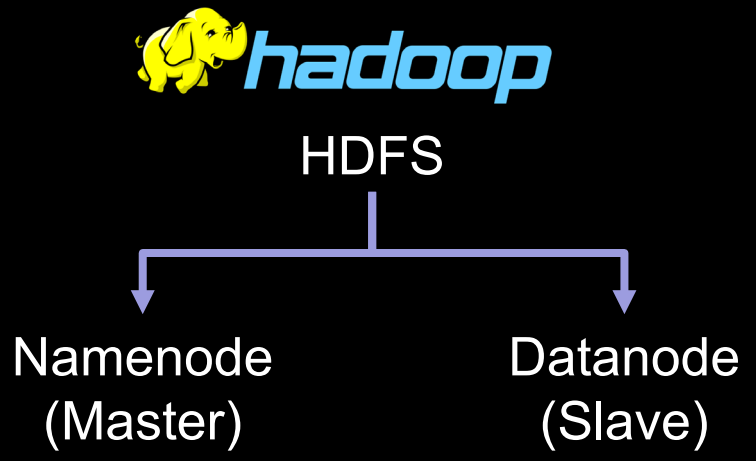


HDFS – Hadoop distributed file system

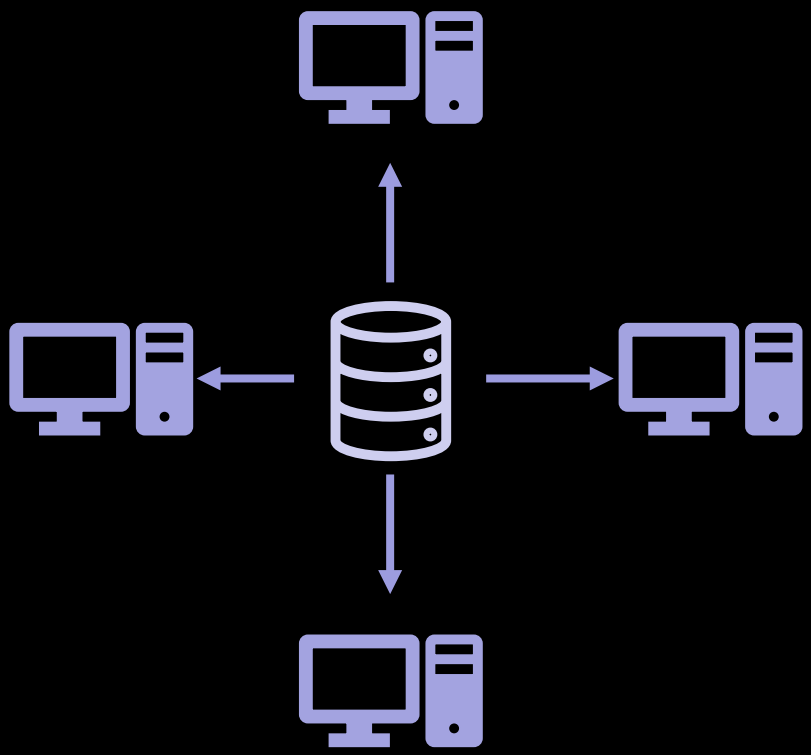


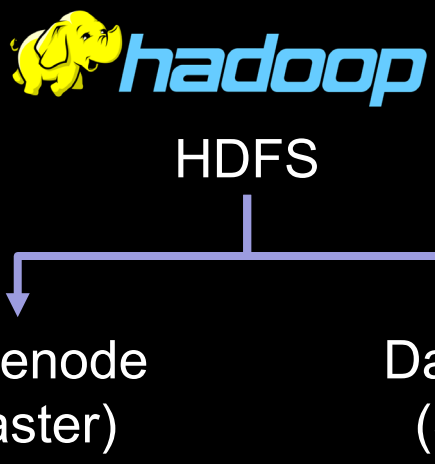
HDFS – Hadoop distributed file system

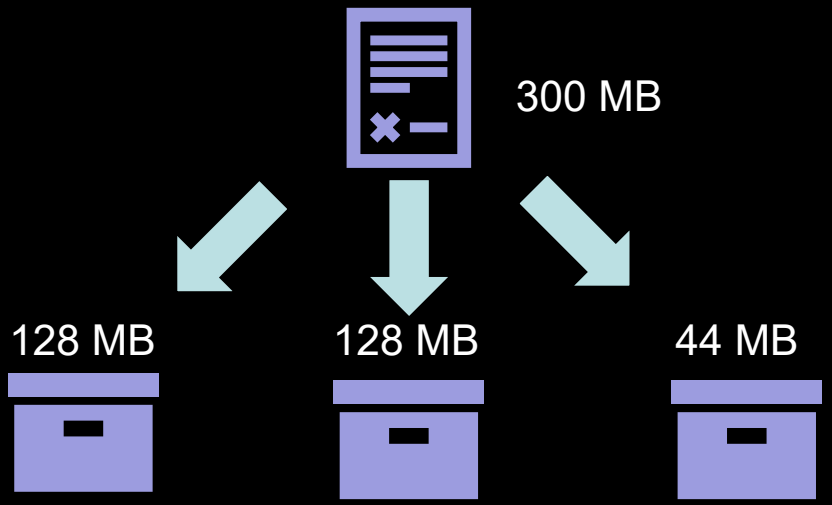




HDFS – Hadoop distributed file system

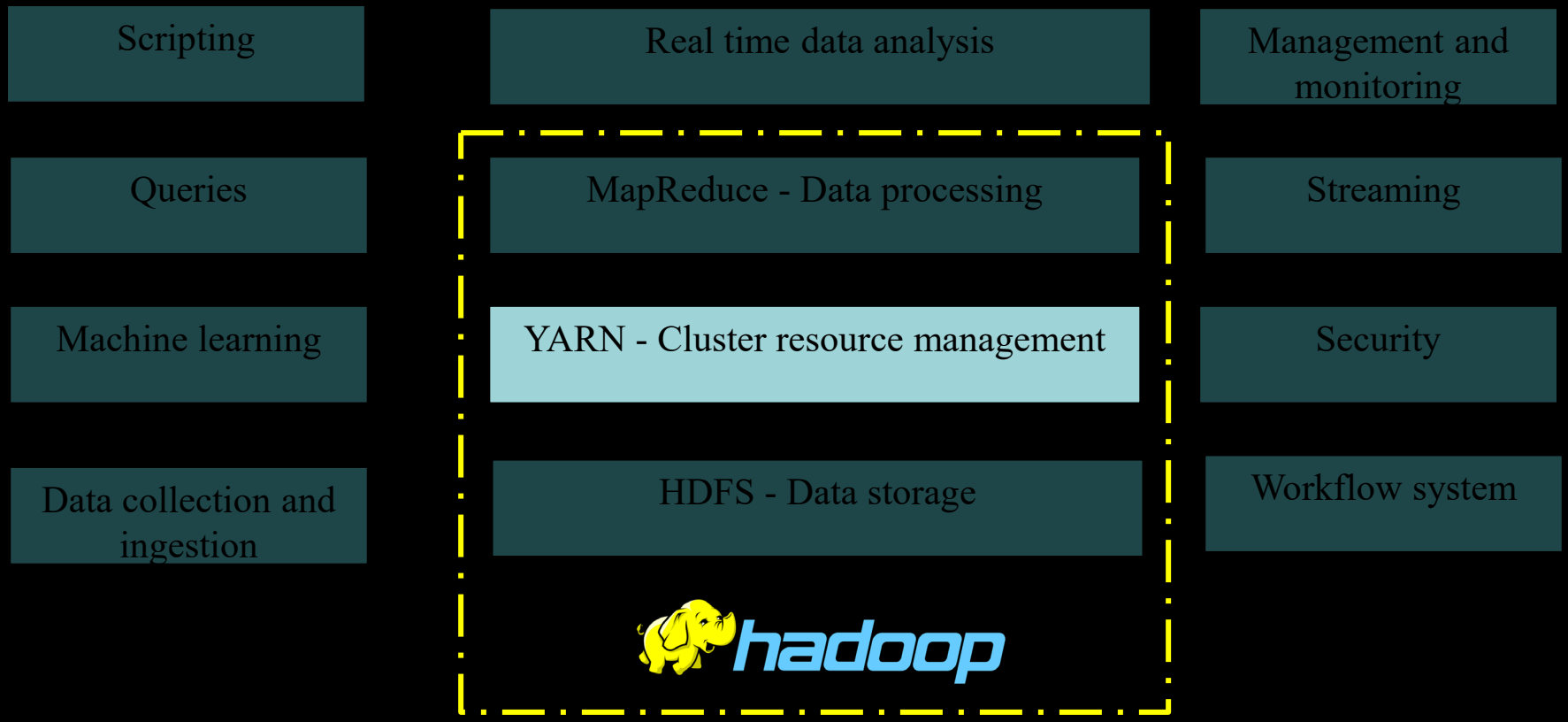




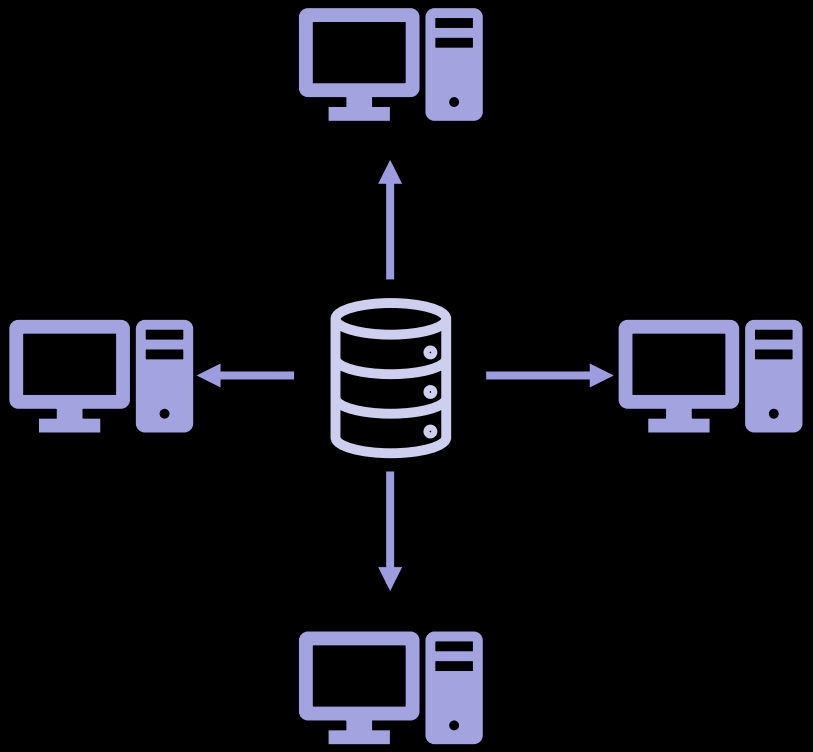


By default i dati sono divisi in blocchi di 128 MB

YARN – Yet Another Resource Negotiator

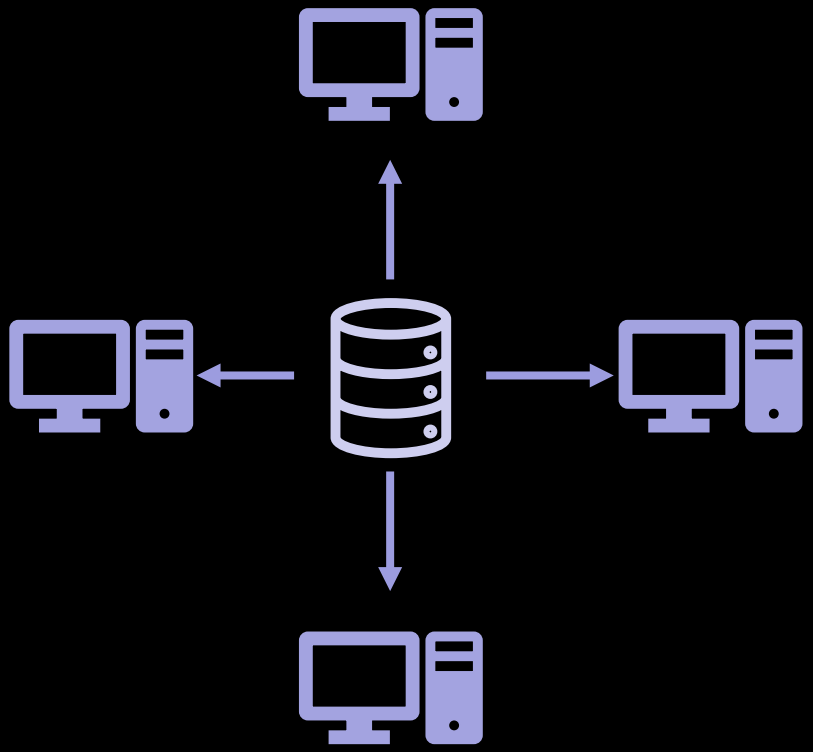


YARN – Yet Another Resource Negotiator



Gestione delle risorse

YARN – Yet Another Resource Negotiator



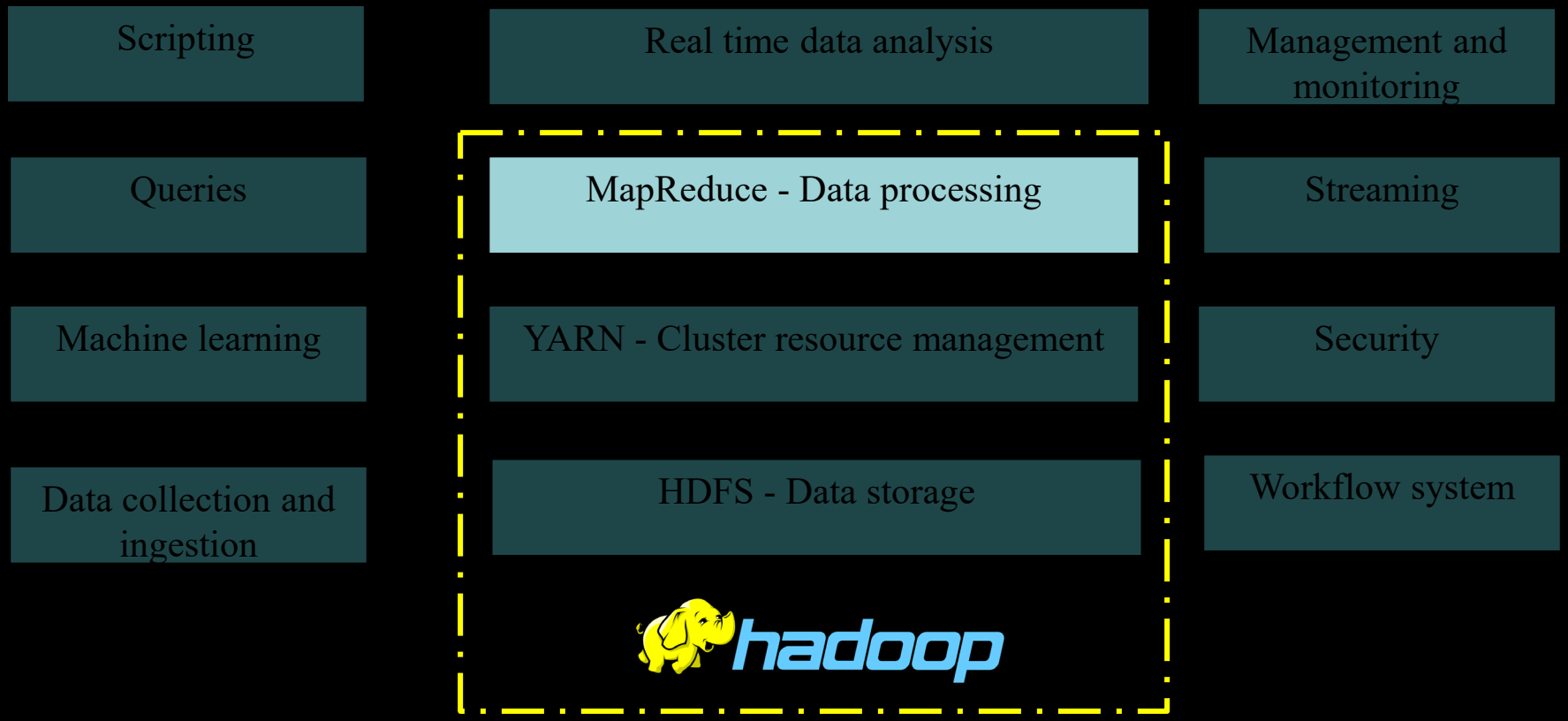
Gestione delle risorse

YARN

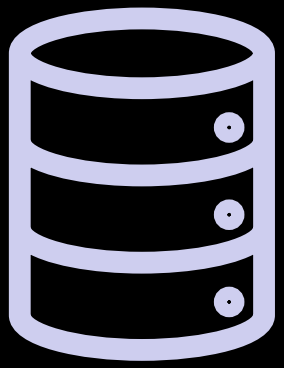
ResourceManager
(master)

NodeManager
(slave)

Data processing - MapReduce

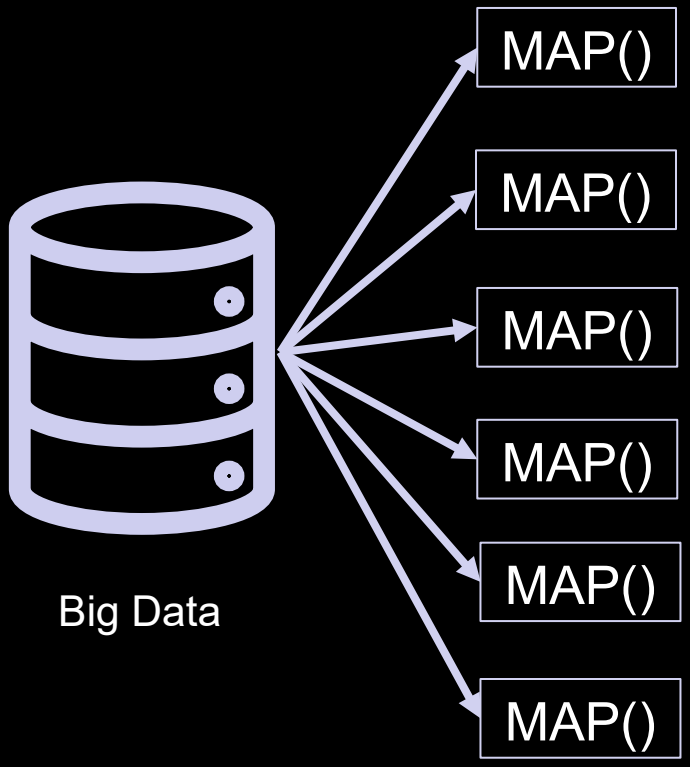


Data processing - MapReduce

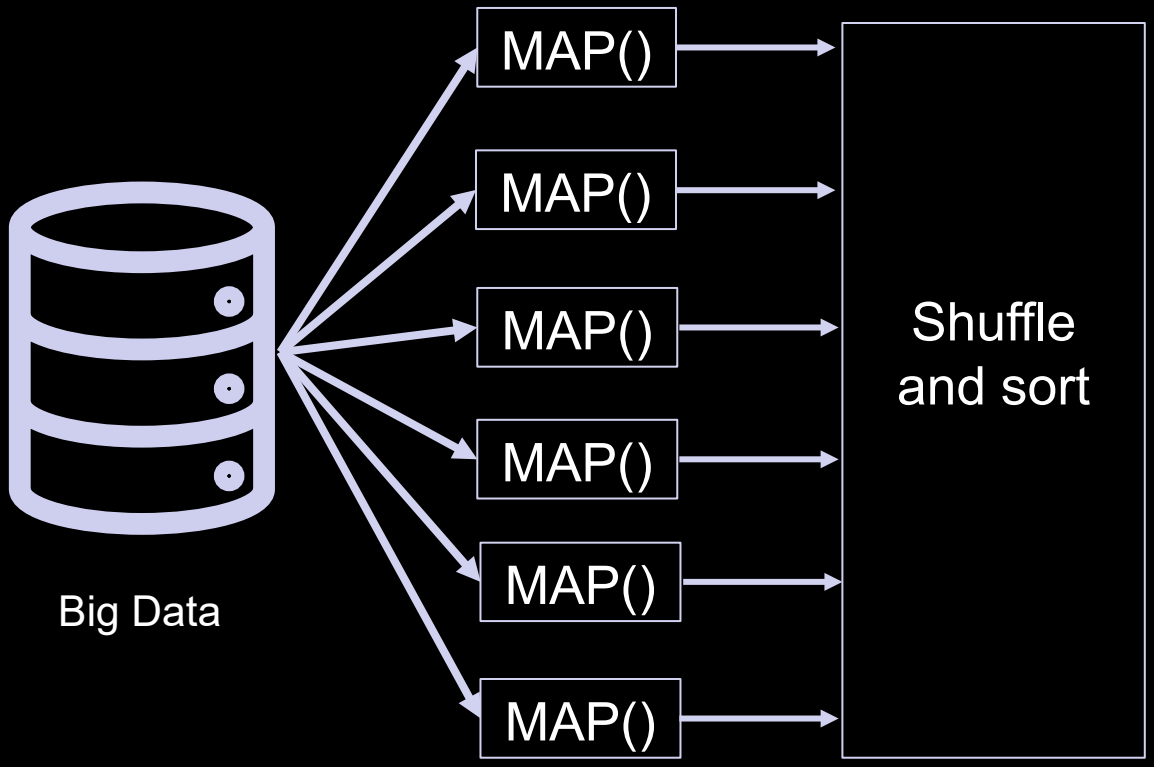


Big Data

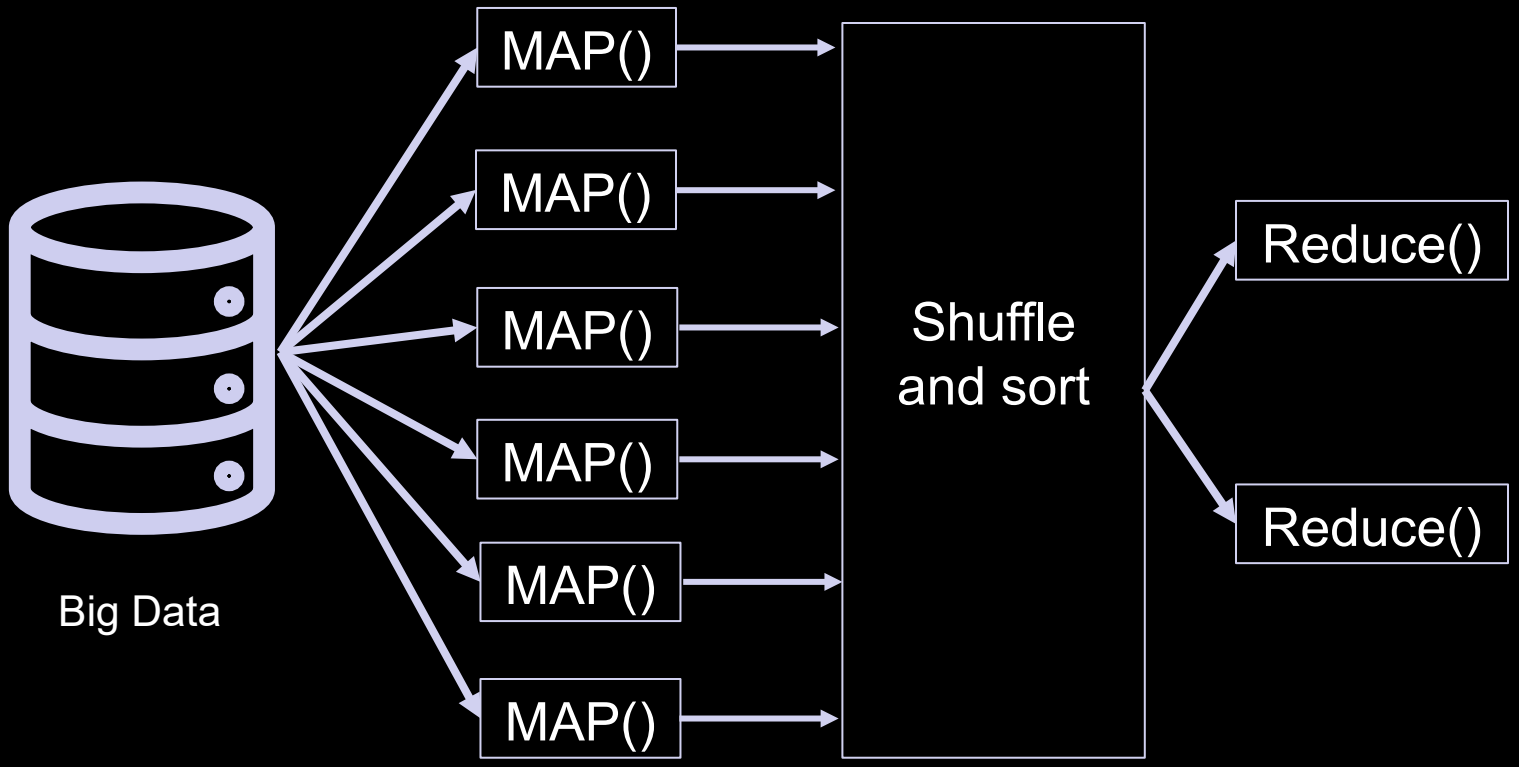
Data processing - MapReduce



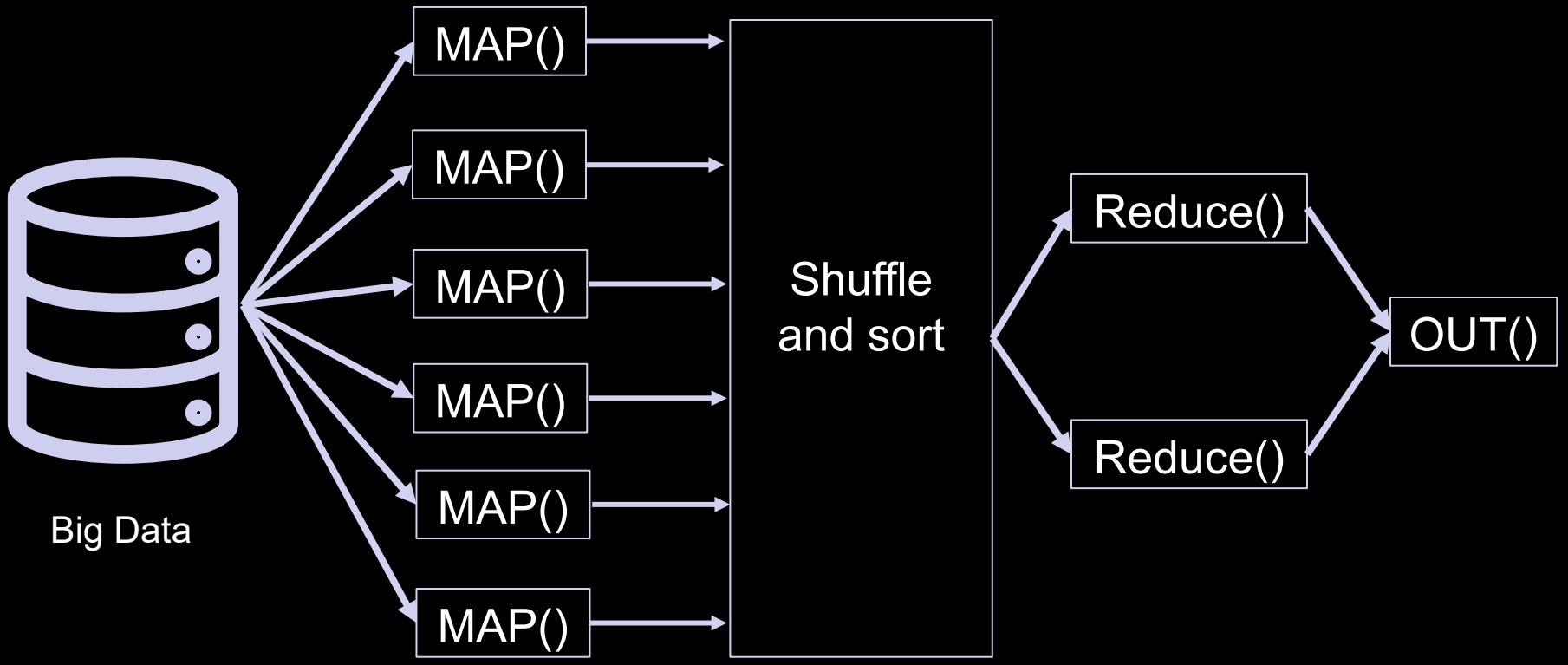
Data processing - MapReduce



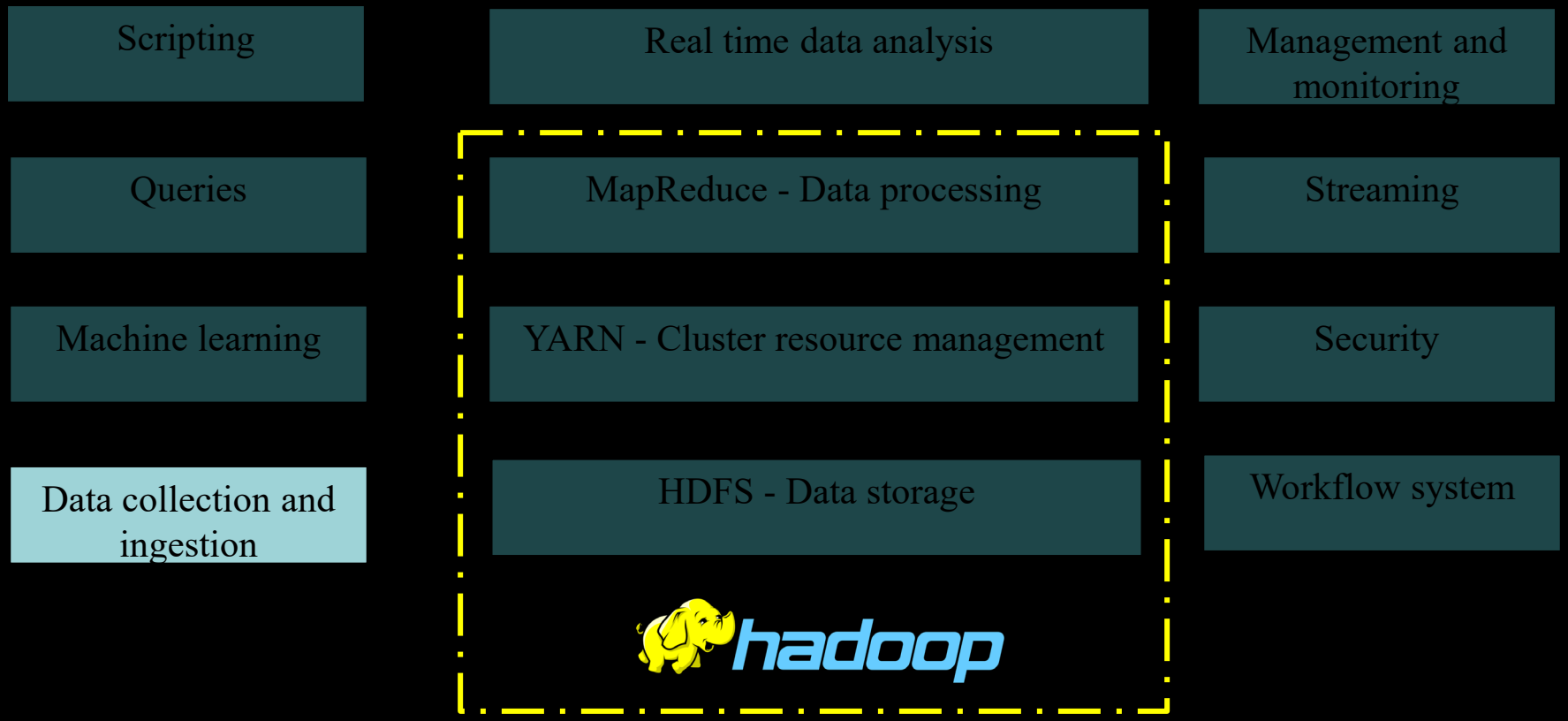
Data processing - MapReduce



Data processing - MapReduce



Data collection and ingestion



Data collection and ingestion

Sqoop è usato per trasferire dati tra hadoop e un datastore esterno

https://sqoop.apache.org/

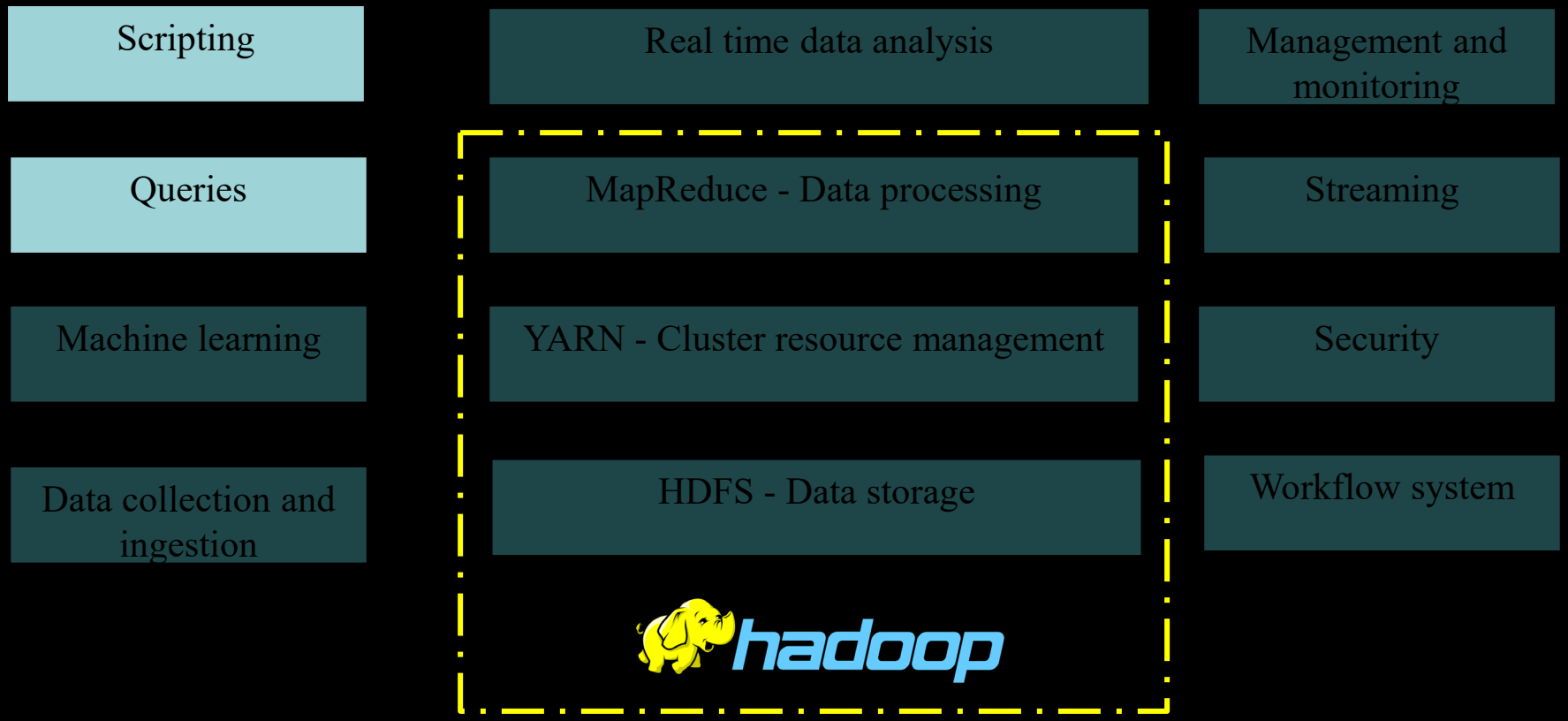
Data collection and ingestion

Sqoop è usato per trasferire dati tra hadoop e un datastore esterno

Flume è un servizio distribuito per collezionare, aggregare e spostare grandi quantità di dati.

https://flume.apache.org/

Scripting and queries



Scripting and queries

PIG è usato per analizzare dati in hadoop. Fornisce una intergaccia di alto livello per il data processing e per eseguire numerose operazioni sui dati.

Piattaforma **ETF** – Extract Transfer and Load

https://pig.apache.org/

Scripting and queries

PIG è usato per analizzare dati in hadoop. Fornisce una intergaccia di alto livello per il data processing e per eseguire numerose operazioni sui dati.

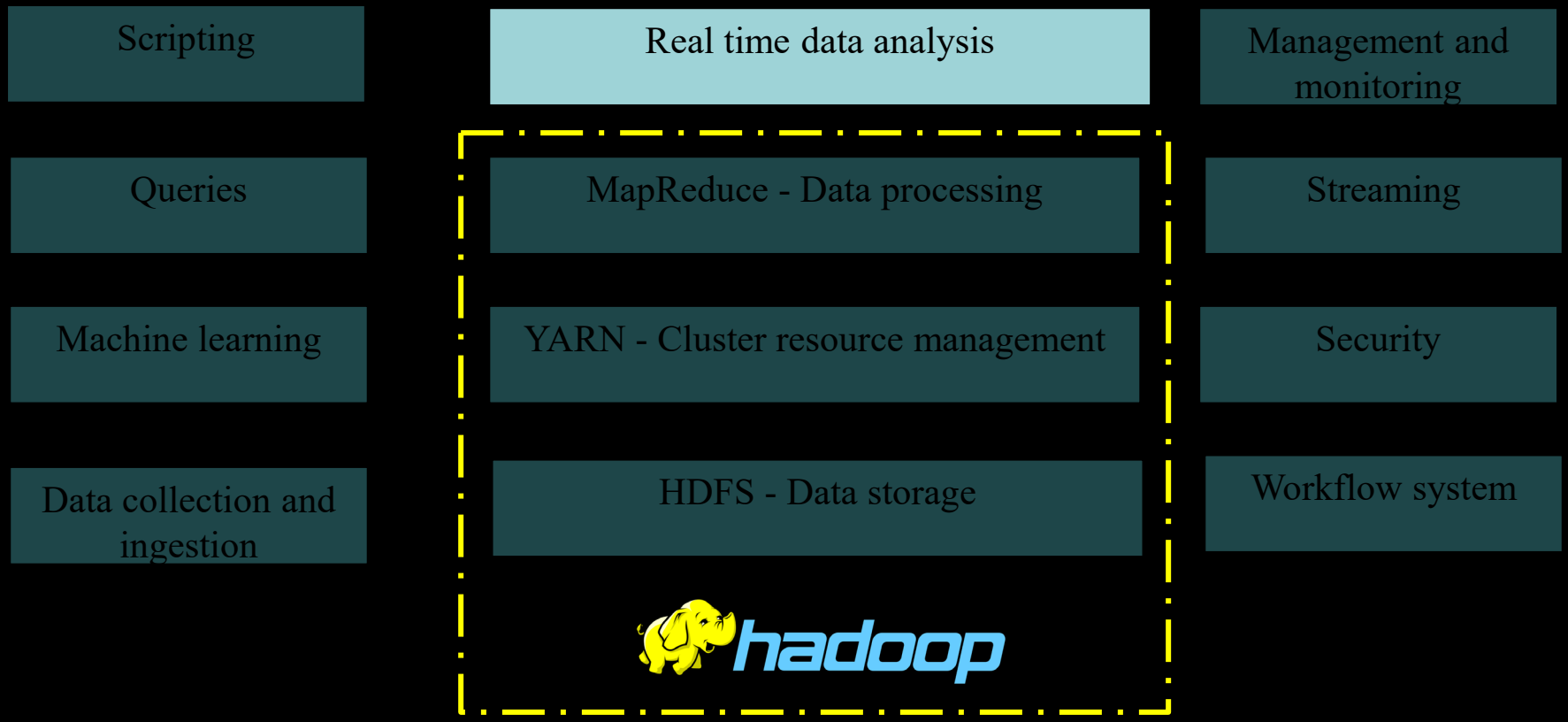
Piattaforma **ETF** – Extract Transfer and Load

https://pig.apache.org/

HIVE semplifica la lettura, scrittura e controllo di grandi dataset che sono allocati in uno storage distribuito usando SQL

https://hive.apache.org/

Spark – Real time data analysis



Spark – Real time data analysis

SPARK è un motore di calcolo distribuito per il processamento e l'analisi di grandi quantità di dati in tempo reale.

https://spark.apache.org/

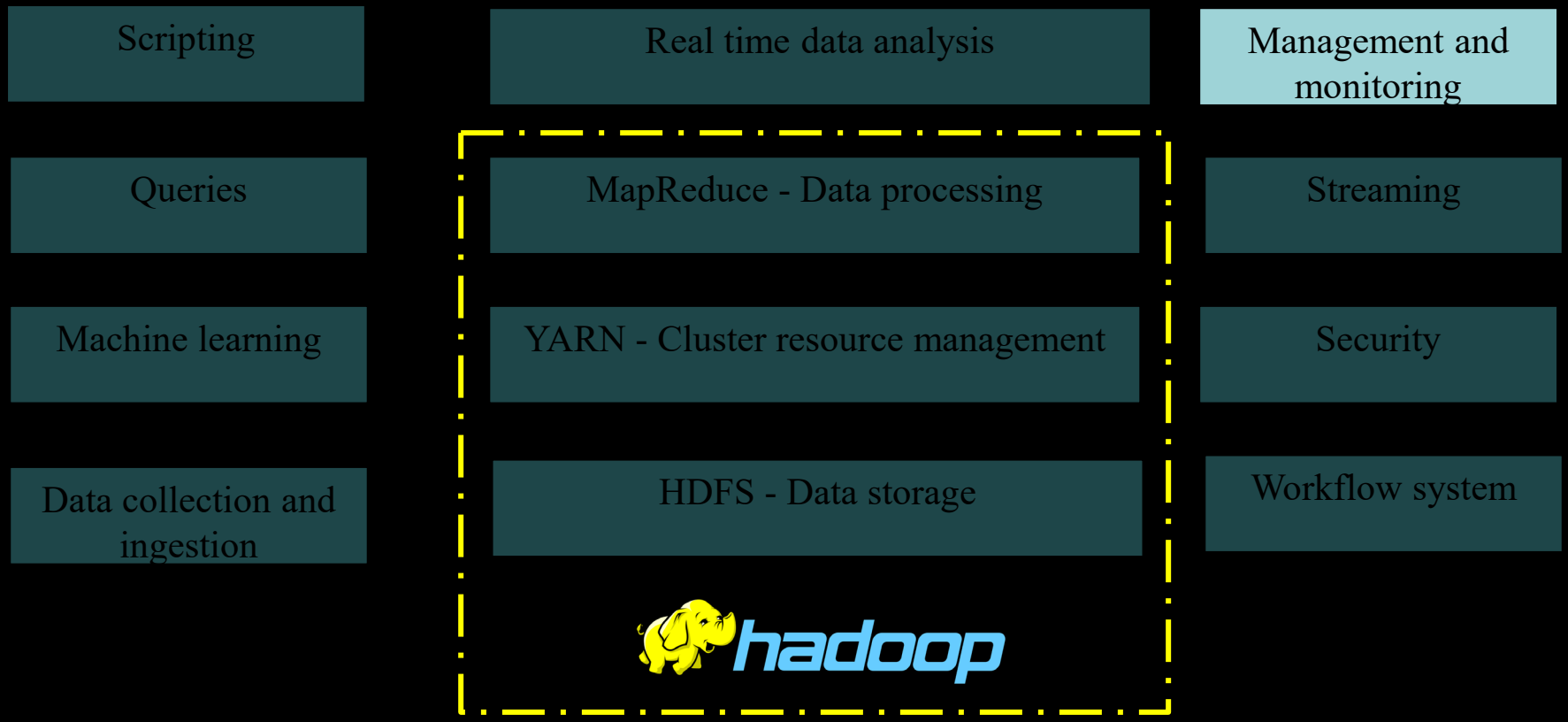
Spark – Real time data analysis

SPARK è un motore di calcolo distribuito per il processamento e l'analisi di grandi quantità di dati in tempo reale.

https://spark.apache.org/

- ✓ Velocità 100x
- ✓ Effettua calcoli per dati direttamente in memoria
- ✓ È usato per processare dati in tempo reale come azioni e dati bancari

Ambari – management and monitoring



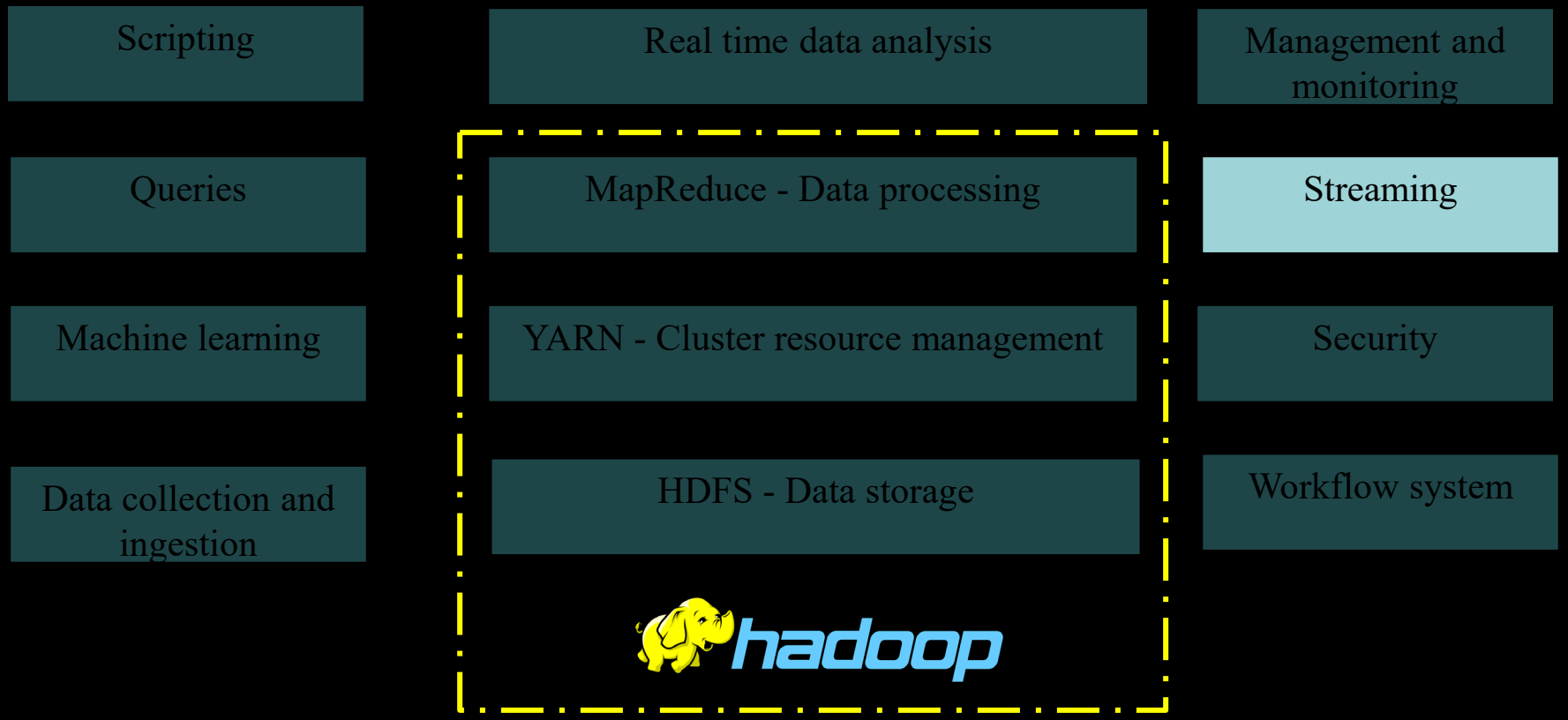
Ambari – management and monitoring

Ambari è un sistema open source responsabile di tenere traccia delle applicazioni in esecuzione e il loro stato.

https://ambari.apache.org/

- ✓ Monitora e gestisce le risorse su un server hadoop
- ✓ Fornisce un servizio di gestione centralizzato per avviare, stoppare e configurare servizi Hadoop
- ✓ Monitoraggio su protocollo web

Streaming


Scripting
Real time data analysis
Management and monitoring
Queries
MapReduce - Data processing
Streaming
Machine learning
YARN - Cluster resource management
Security
Data collection and ingestion
HDFS - Data storage
Workflow system
hadoop

Streaming

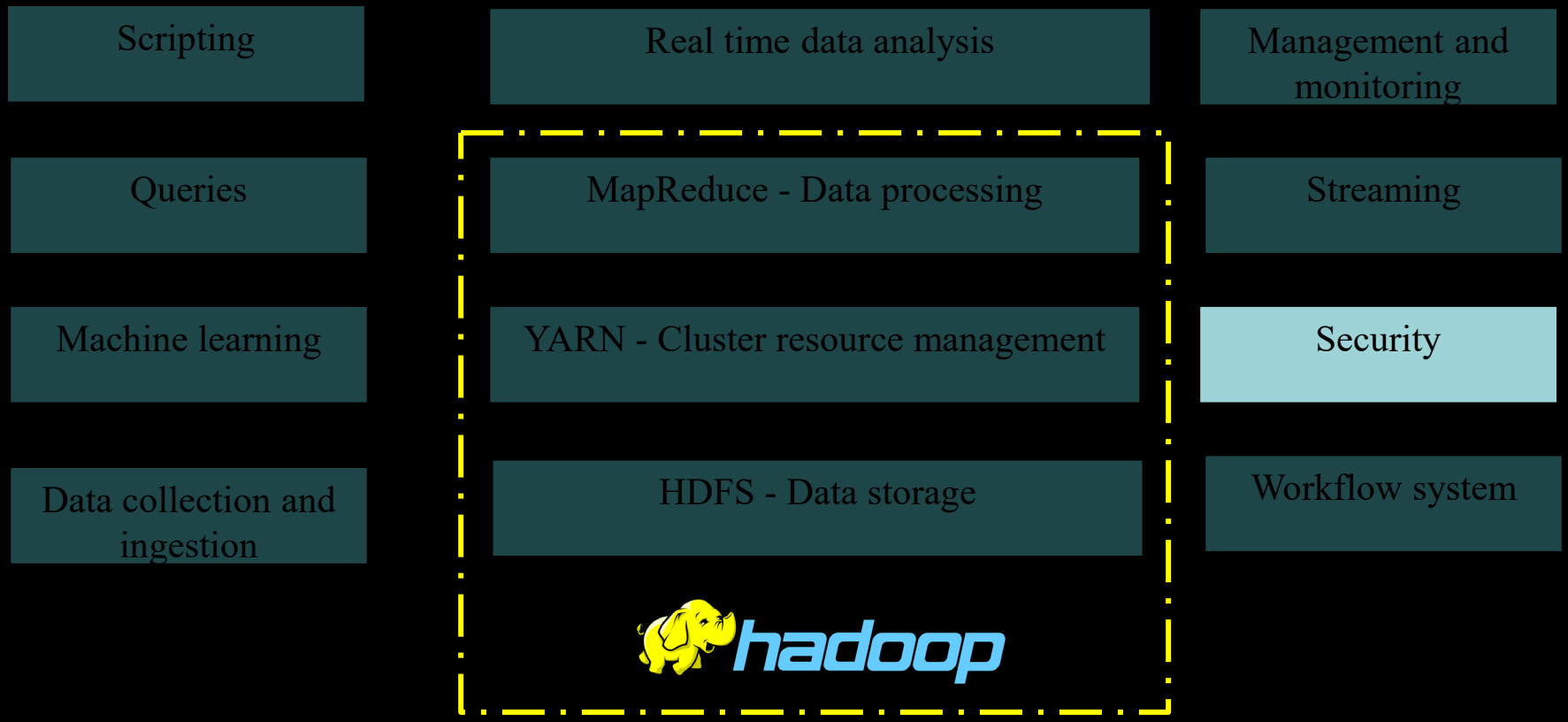
KAFKA è una piattaforma di streaming distribuito per memorizzare e processare flussi di dati

https://kafka.apache.org/

STORM è motore di processo che lavora dati in streaming in tempo reale (velocità molto alta)

https://storm.apache.org/

Sicurezza

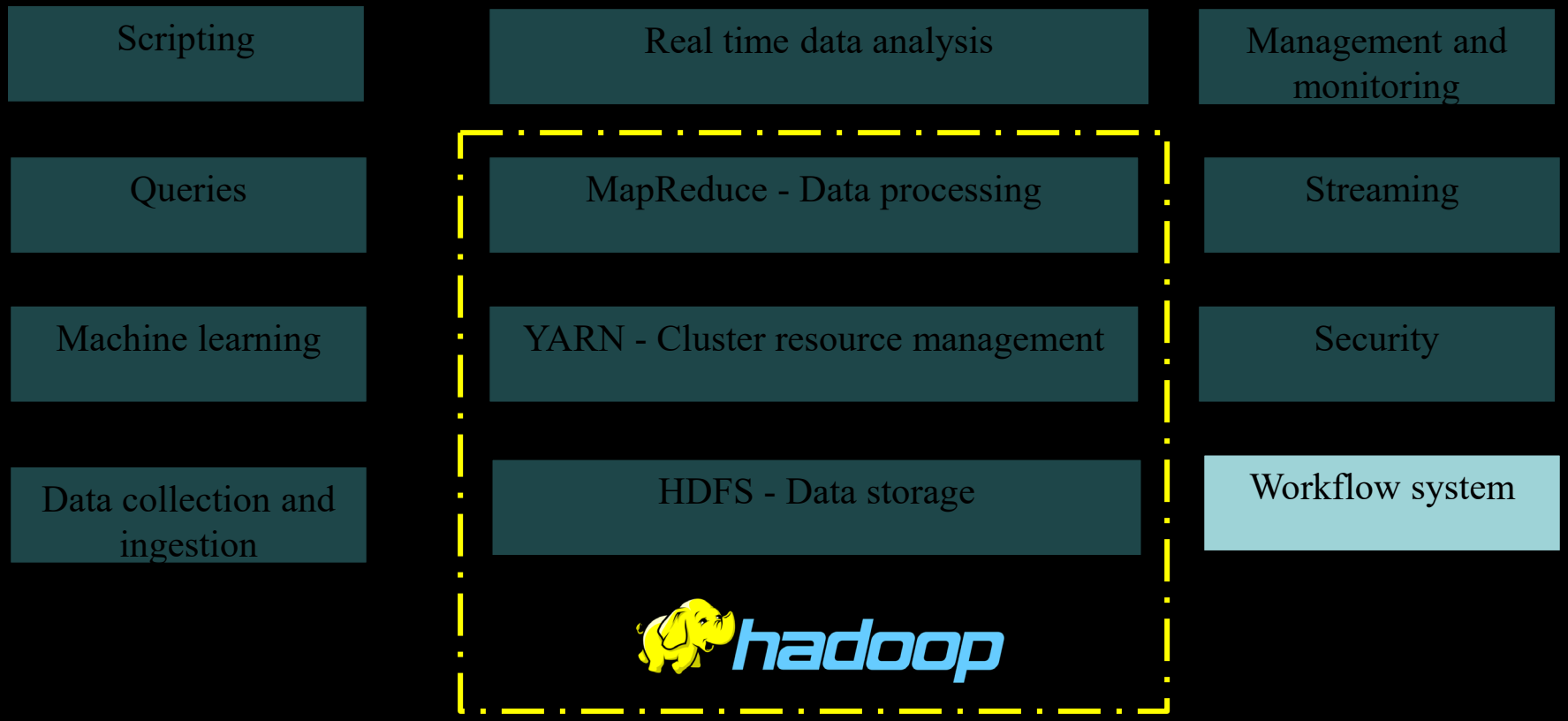

Scripting
Real time data analysis
Management and monitoring
Queries
MapReduce - Data processing
Streaming
Machine learning
YARN - Cluster resource management
Security
Data collection and ingestion
HDFS - Data storage
Workflow system
hadoop

Security

Ranger è un framework che permette di monitorare e controllare la sicurezza dei dati sulle piattaforme Hadoop

KNOX è una applicazione gateway per interaggire con il resto delle applicazioni o interfacce (API) che usano hadoop

Workflow


Scripting
Real time data analysis
Management and monitoring
Queries
MapReduce - Data processing
Streaming
Machine learning
YARN - Cluster resource management
Security
Data collection and ingestion
HDFS - Data storage
Workflow system
hadoop

Workflow

Oozie è uno scheduler del workflow per controllare i jobs in esecuzione in hadoop tramite DAGs o Directed Acyclical Graphs

Hadoop - Ecosystem

Scripting

Real time data analysis

Management and monitoring

Queries

MapReduce - Data processing

Streaming

Machine learning

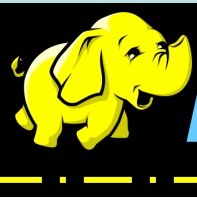
YARN - Cluster resource management

Security

Data collection and ingestion

HDFS - Data storage

Workflow system



L’arte di gestire il big data

source: Image from: CloudTweaks

Quando si NON ha la necessità di lavorare con il big data

- Dati nell’ordine dei 100GB non richiedono un approccio big data

Quando si NON ha la necessità di lavorare con il big data

- Dati nell'ordine dei 100GB non richiedono un approccio big data
- Quando ci sono molti dataset eterogenei molto piccoli da processare individualmente

Quando si NON ha la necessità di lavorare con il big data

- Dati nell'ordine dei 100GB non richiedono un approccio big data
- Quando ci sono molti dataset eterogenei molto piccoli da processare individualmente
- Assenza di un flusso

Quando si NON ha la necessità di lavorare con il big data

- Dati nell'ordine dei 100GB non richiedono un approccio big data
- Quando ci sono molti dataset eterogenei molto piccoli da processare individualmente
- Assenza di un flusso
- No business intelligence plan

Quando NON usare Hadoop

- ✓ Real time analytics

Quando NON usare Hadoop

- ✓ Real time analytics
- ✓ Hadoop non di adatta bene a sostituire una piattaforma esistente

Quando NON usare Hadoop

- ✓ Real time analytics
- ✓ Hadoop non di adatta bene a sostituire una piattaforma esistente
- ✓ Molti dataset di piccole dimensioni

Quando NON usare Hadoop

- ✓ Real time analytics
- ✓ Hadoop non di adatta bene a sostituire una piattaforma esistente
- ✓ Molti dataset di piccole dimensioni
- ✓ Quando la sicurezza è un aspetto fondamentale

Quando NON usare Hadoop

- ✓ Real time analytics
- ✓ Hadoop non di adatta bene a sostituire una piattaforma esistente
- ✓ Molti dataset di piccole dimensioni
- ✓ Quando la sicurezza è un aspetto fondamentale
- ✓ Quando manca un sistema hardware dedicato

Quando usare Hadoop

- ✓ Dati nell’ordine dei TB-PB e diversità dei dati

Quando usare Hadoop

- ✓ Dati nell’ordine dei TB-PB e diversità dei dati
- ✓ Si pianifica una crescita del flusso dati

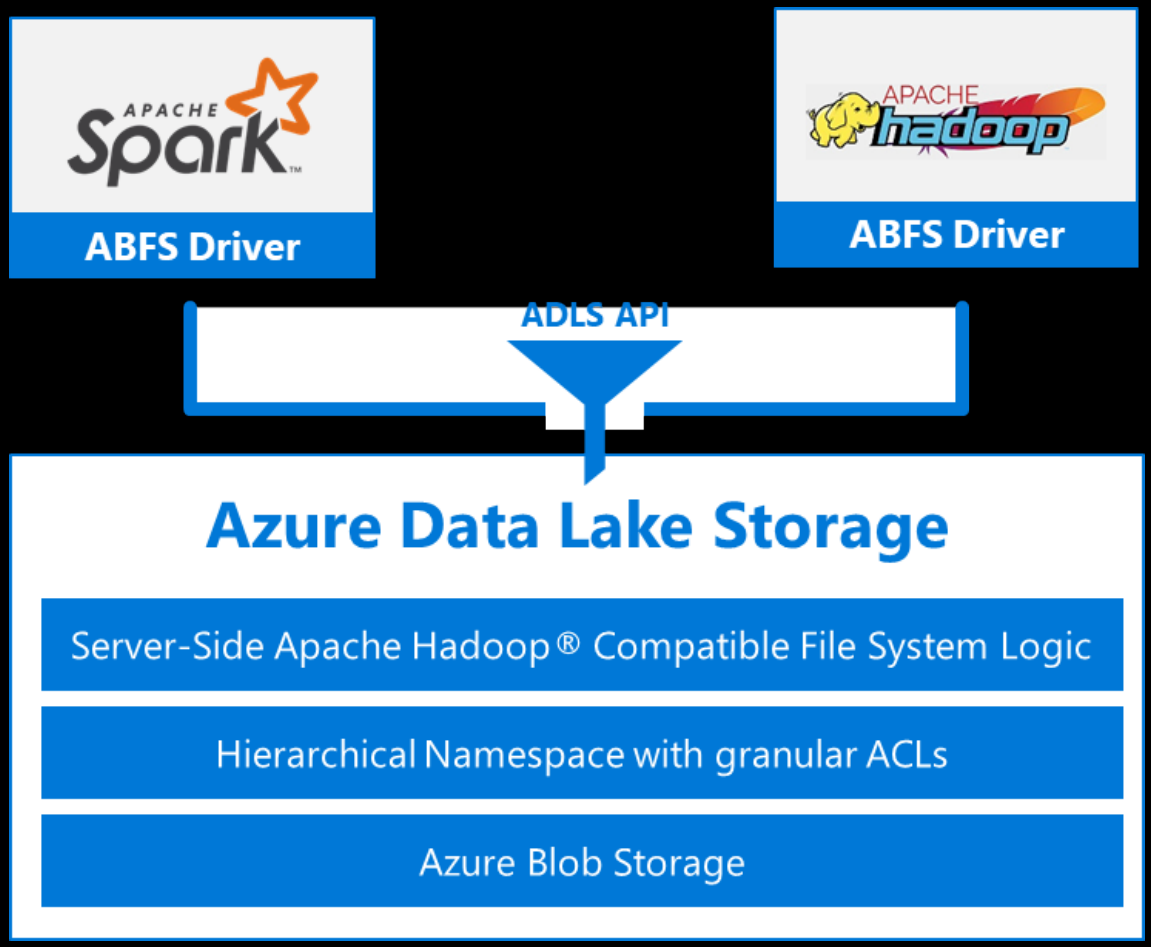
Quando usare Hadoop

- ✓ Dati nell’ordine dei TB-PB e diversità dei dati
- ✓ Si pianifica una crescita del flusso dati
- ✓ Quando sono usati molti framework per gestire i big data

Cluster Hadoop VS cloud service

- ✓ Forniscono piattaforme già integrate con numerose funzionalità
- ✓ Costo variabile in base all'utilizzo
- ✓ Alta flessibilità e adattabilità
- ✓ NO costo hardware
- ✓ NO costo di manutenzione (hardware e software)

Cluster Hadoop VS cloud service



1: https://azure.microsoft.com/en-us/blog/under-the-hood-performance-scale-security-for-cloud-analytics-with-adls-gen2/