

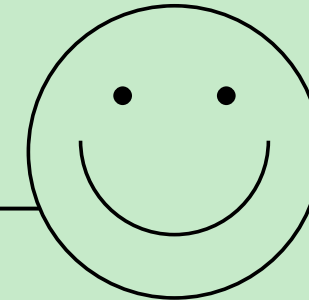
Manenti Edoardo

Simone Nicosanti

Andrea De Filippis

SDCC PROJECT 2K23

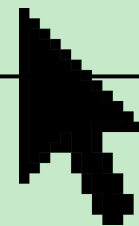
A.A. 2022/2023



STORAGE NEL CLOUD CONTINUUM

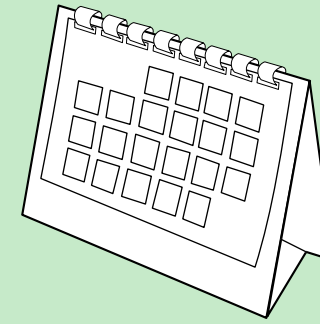


Presentazione del progetto  
*"SAE storage service"*



Let's get started

# AGENDA



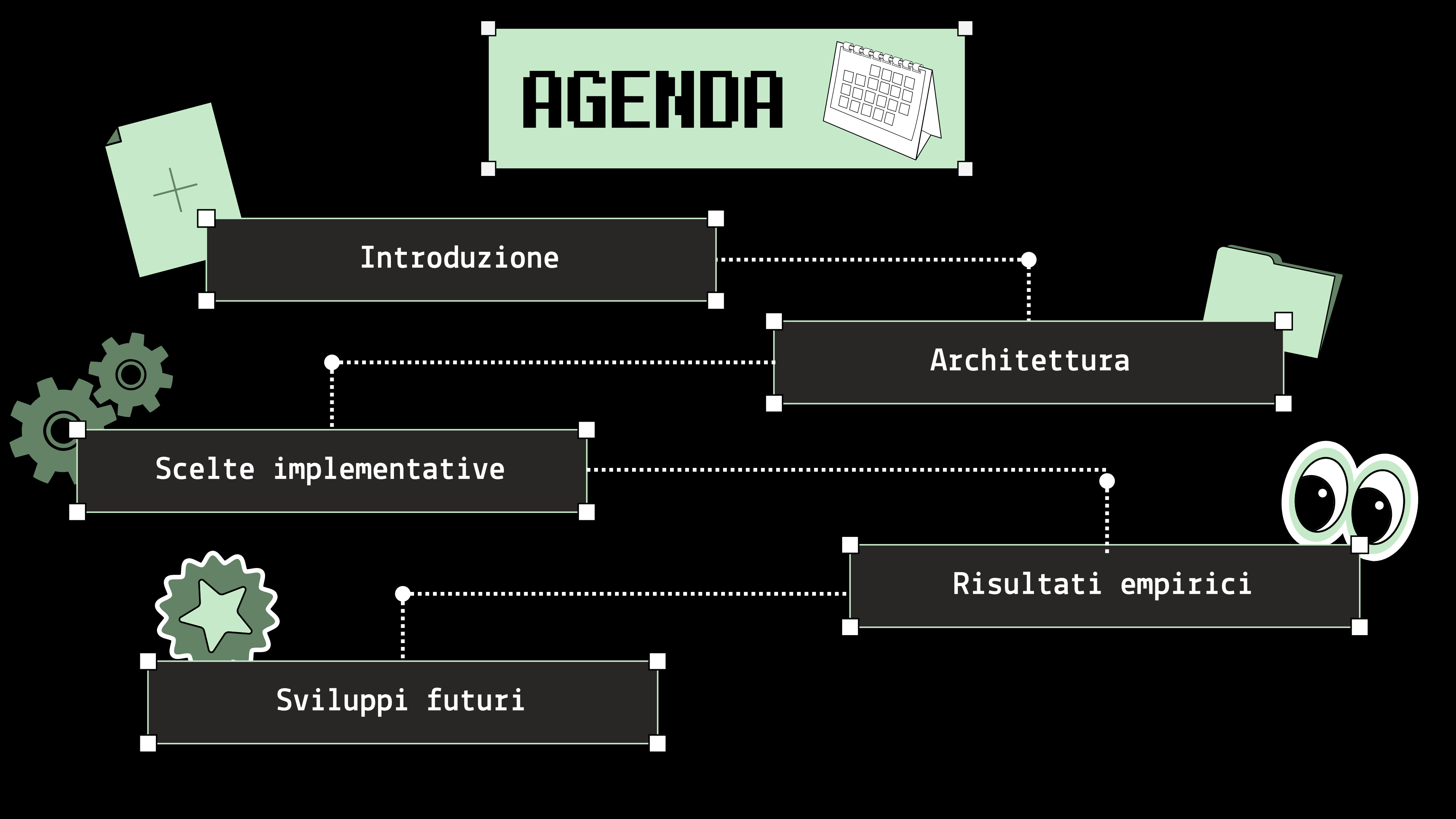
Introduzione

Architettura

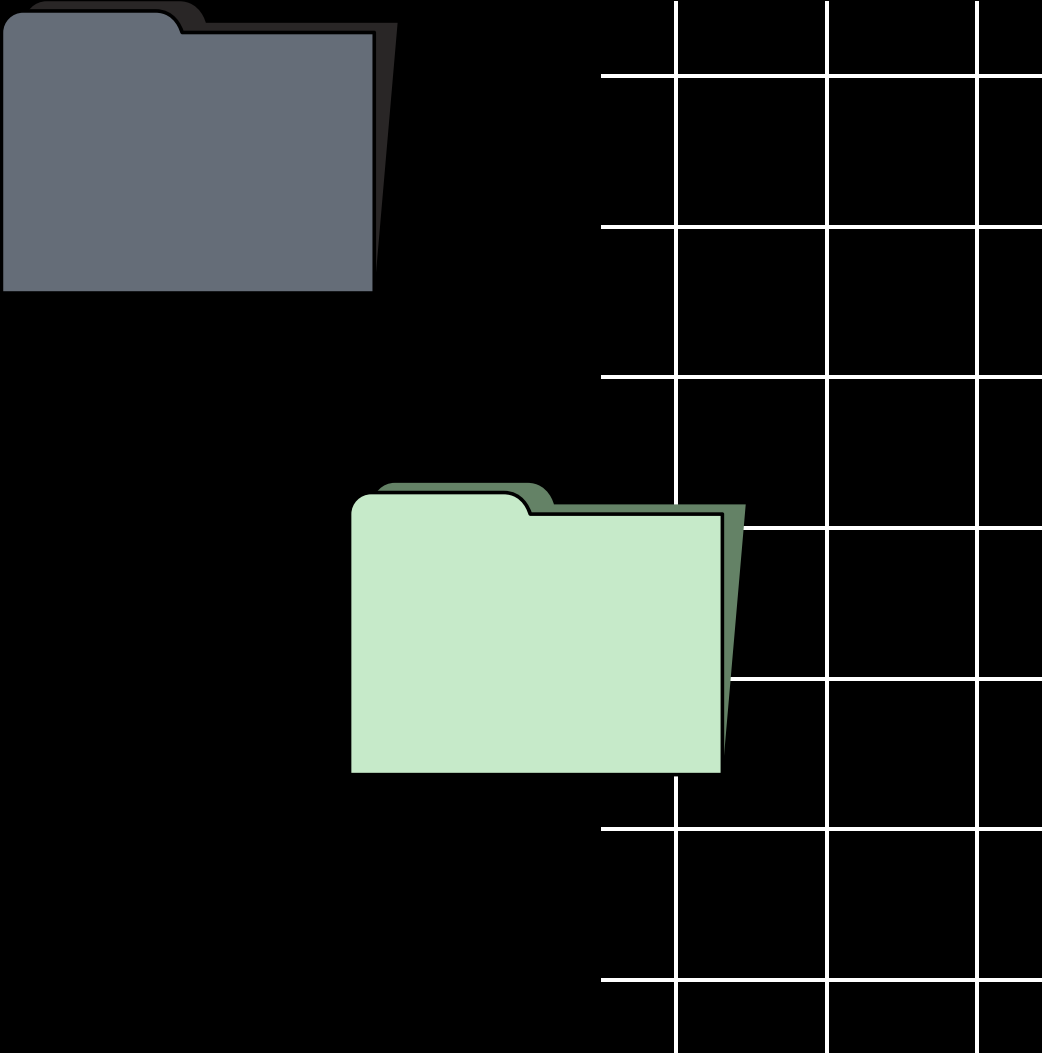
Scelte implementative

Risultati empirici

Sviluppi futuri



# INTRODUZIONE

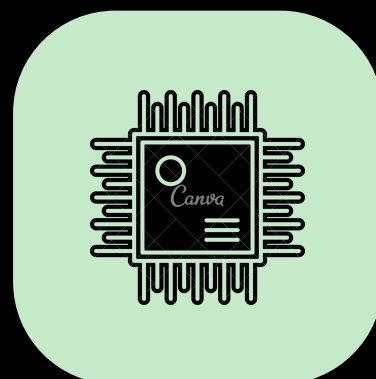


In questa presentazione entreremo nei dettagli del sistema sviluppato come progetto per l'esame di Sistemi Distribuiti.

Per prima cosa vediamo i building blocks e l'architettura di base...

# BUILDING BLOCKS

9%



EDGE

Nodo con risorse limitate e capacità di caching che comunica con il cloud ed altri dispositivi a formare una rete non strutturata che costituisce un layer intermedio tra i client e il cloud.



CLIENT

Client che, una volta autenticato, può richiedere di inserire o eliminare file nella rete.

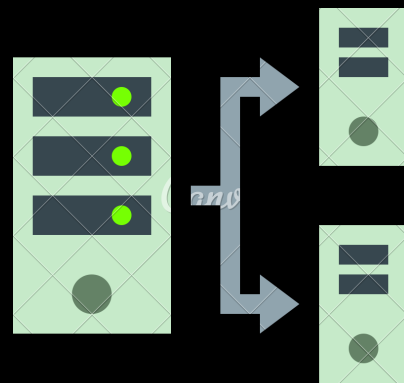


S3 CLOUD

Bucket S3 su amazon web services su cui sono salvati i file dell'intero sistema.

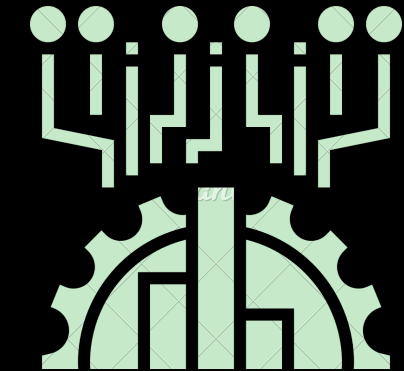
# BUILDING BLOCKS

14%



**BALANCER**

Nodo che autentica i client e ne gestisce le richieste redistribuendole equamente tra i vari edge.



**REGISTRY**

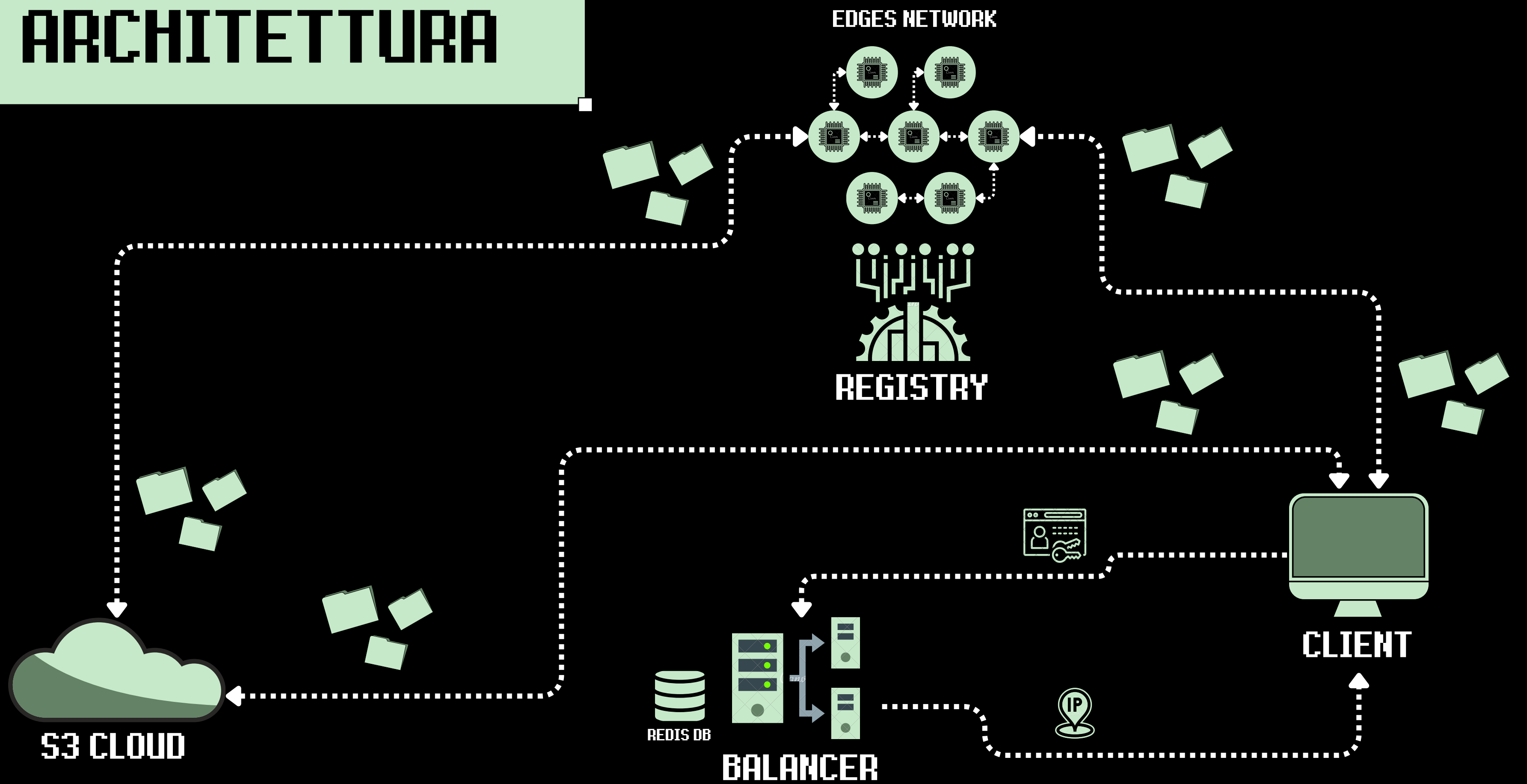
Nodo che gestisce la rete di edge in maniera centralizzata con controlli periodici sulle componenti connesse.



**REDIS DB**

DB ospitato dal nodo del load balancer che permette di mantenere le utenze autorizzate in persistenza.

# ARCHITETTURA

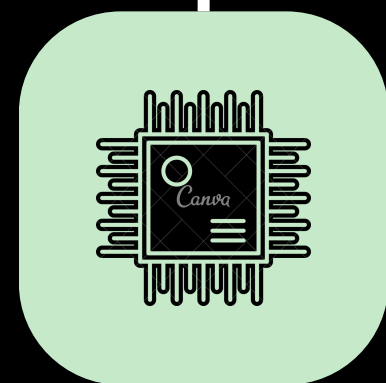


# ENTRATA DI UN EDGE

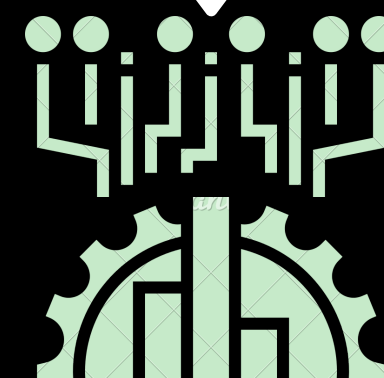
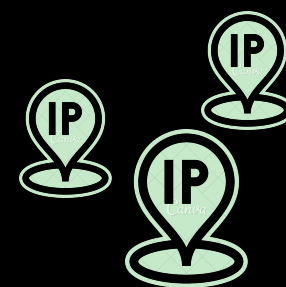
23%

*Hello, i want to enter in the network!*

*Hey there! Here are your neighbours...*



**NEW EDGE !!!**



**REGISTRY**



**ELABORAZIONE**

Scelta dei vicini a cui connettersi e aggiornamento degli edge attivi.



**ELABORAZIONE**

Tentativo di connessione ai vicini restituiti, inizio meccanismo di ping verso di essi e invio di heartbeat periodici verso il registry.

# PING

TRA VICINI

27%

Connecting to neighbour...

OK!

Connection request received, connecting...

OK!



## A FEW MOMENTS LATER...

Hey neighbour! Are you there?

Yes! Still here...

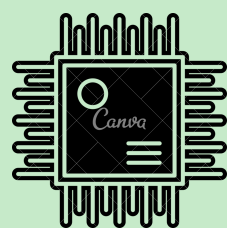
I am already connected to this guy... all good!



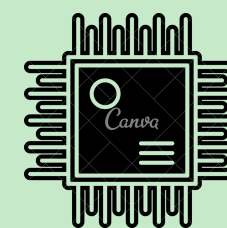
## N SECONDS LATER...

Hey neighbour! Are you there?

Yes! Still here...



EDGE



NEIGHBOUR



# PING

TRA VICINI

32%



I don't know you...  
Let's connect!

# PING

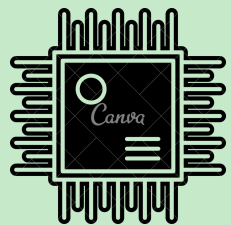
TRA VICINI

36%

*Hey neighbour! Are you there?*

*I'm asking you for the 2nd time... Are you there?*

*Third time's the charm... Are you there?*



EDGE

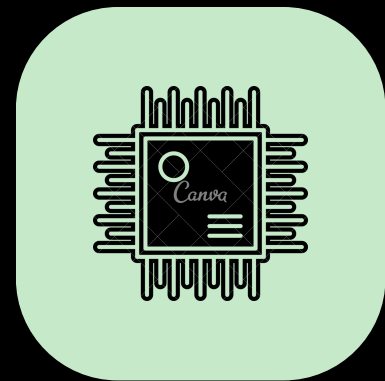


NEIGHBOUR

Neighbour not responding...  
assuming he's dead...

# HEARTBEAT

VERSO IL  
REGISTRY



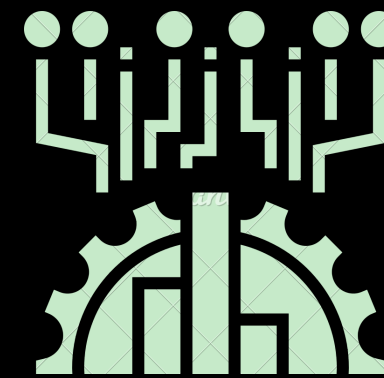
EDGE

*I'm alive!*



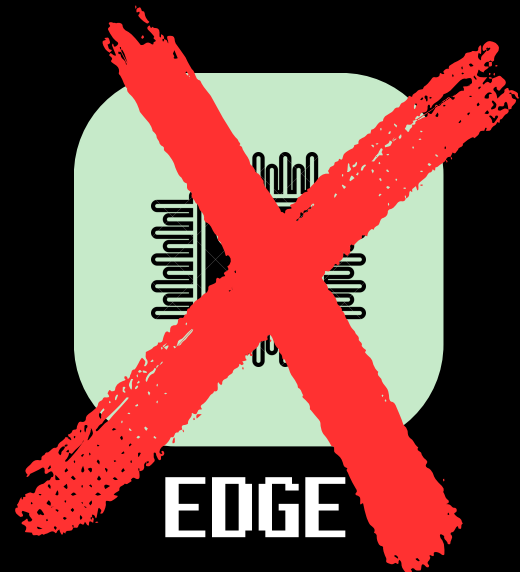
N SECONDS LATER...

*I'm alive!*



REGISTRY

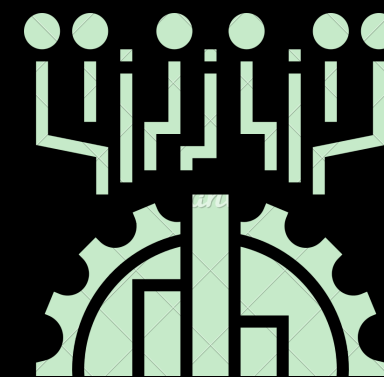
Not hearing from edge for a  
while... assuming he's dead...



EDGE



T SECONDS LATER...

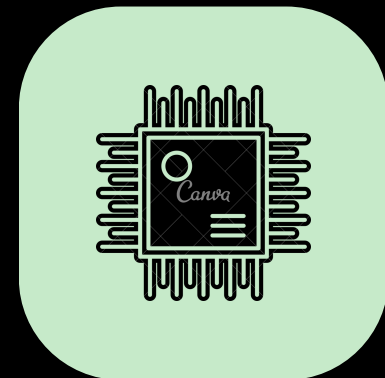


REGISTRY

# COSTRUZIONE GRAFO

NEL REGISTRY

45%

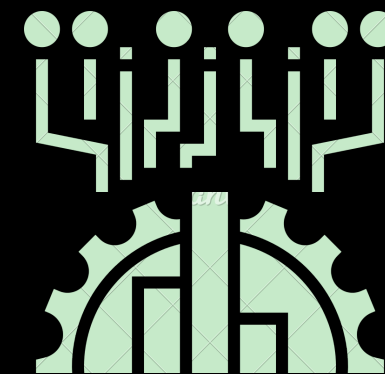
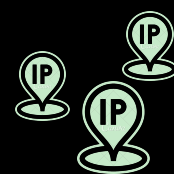


EDGE

*Which are your neighbours?*



*These are my neighbours!*



REGISTRY



## ELABORAZIONE

Raccolgo le informazioni sui vicini per ciascun edge attivo e ricostruisco il grafo delle connessioni.

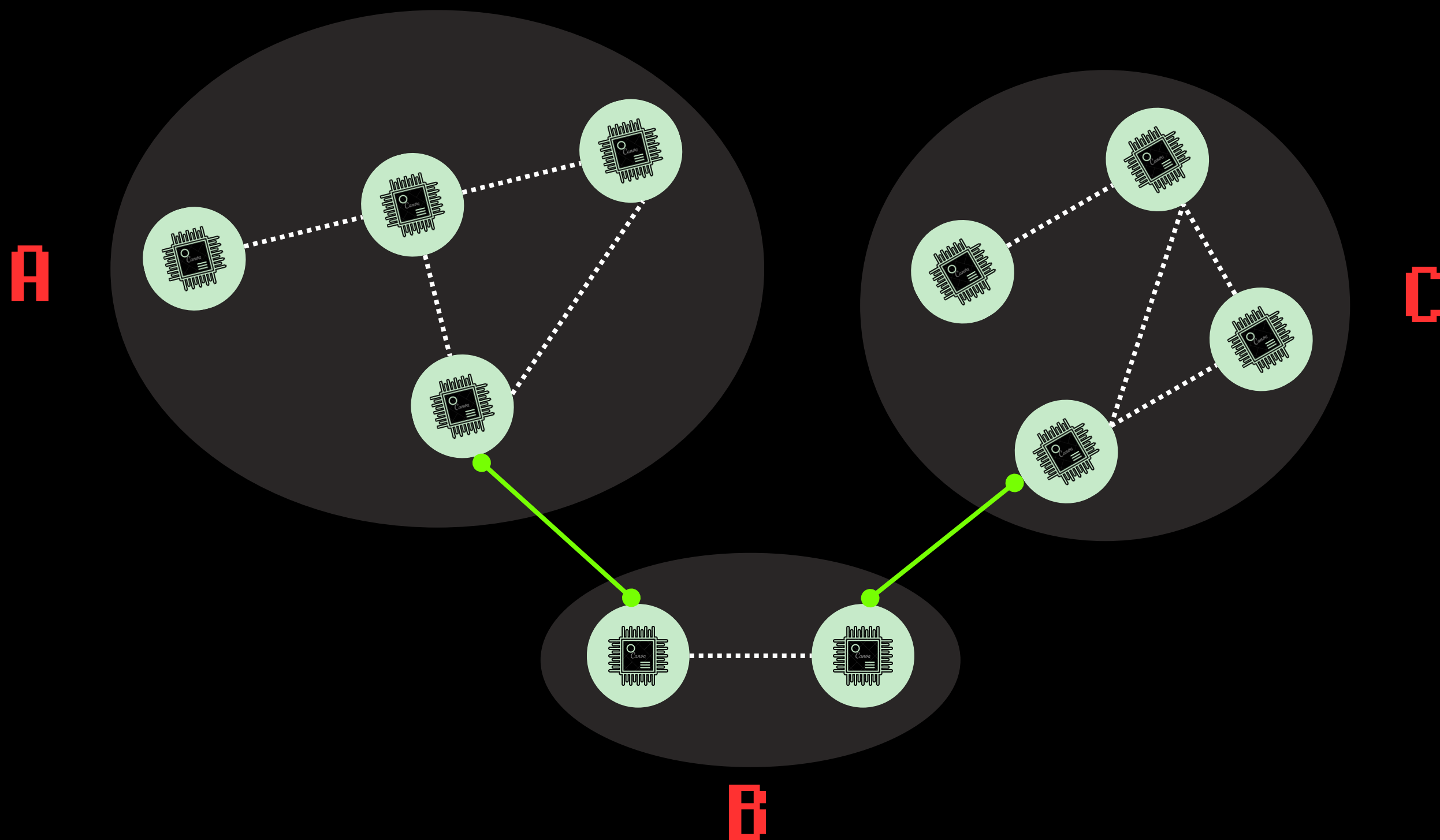
# RICERCA COMPONENTI

## CONNESSE

NEL REGISTRY

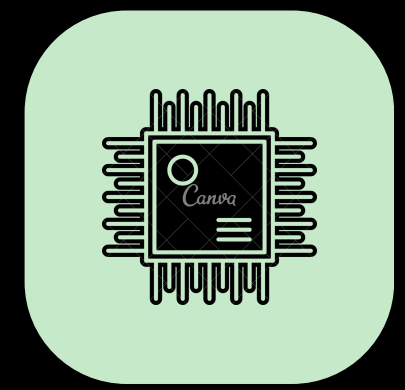
50%

Una volta ricostruito il grafo delle connessioni...



# HEARTBEAT

VERSO IL  
BALANCER

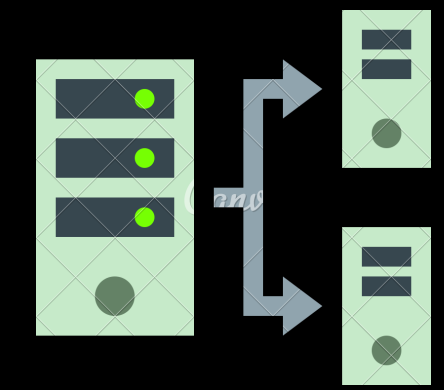


EDGE

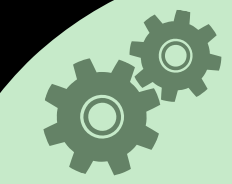
*I'm alive and i am serving X requests!*

 N SECONDS LATER...

*I'm alive and i am serving Y requests!*

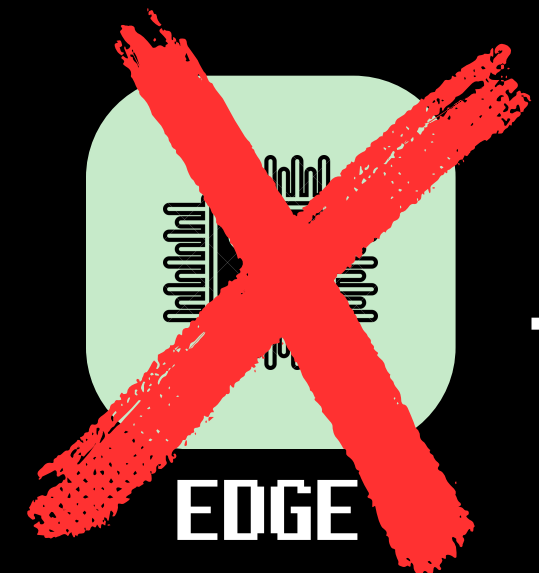


BALANCER



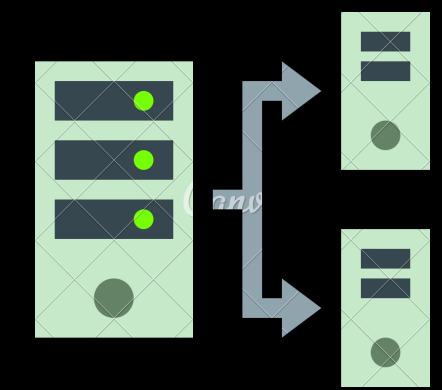
## ELABORAZIONE

Se conosco l'edge aggiorno  
il *CURRENT\_LOAD*,  
altrimenti lo aggiungo  
agli edge attivi e salvo  
il valore di carico.



EDGE

 T SECONDS LATER...



BALANCER

Not hearing from edge  
for a while...  
assuming he's dead...

# CONNESSIONE AL BALANCER

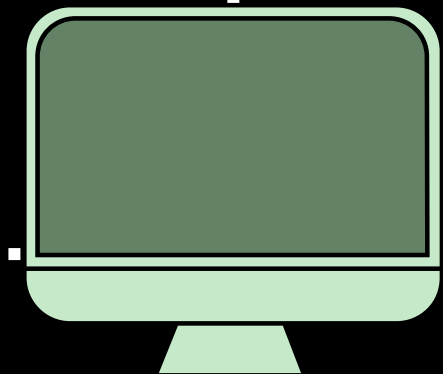
59%

*edge.DOWNLOAD/UPLOAD/DELETE "temp\_sensorK.json"*

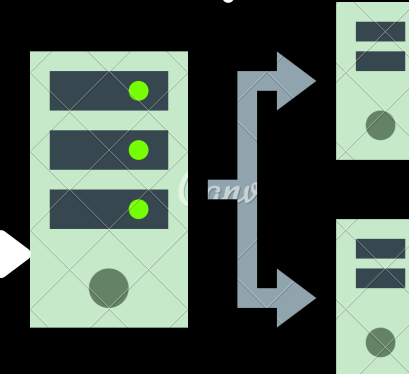
*username: sensorA  
password: s3cr3t*



*Give me an Edge to contact.*



CLIENT



BALANCER



*Allow/Deny*



REDIS DB



search  
for  
user

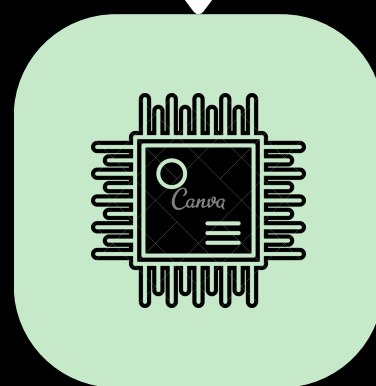
```
IF isLoggedIn(Client) {  
    edge := ChooseEdgeWithMinLoad()  
    edge.CURRENT_LOAD++  
    return edge.IPaddr  
}
```



# DOWNLOAD REQUEST

64%

DOWNLOAD  
"temp\_sensorK.json"



EDGE

## RICERCA DEL FILE



1

Ricerca del file nella cache locale

2

Ricerca del file nella rete

3

Ricerca del file su S3



# LOOKUP DEL FILE

INVIO DELLE RICHIESTE

```
IF req NOT IN ReqCache {  
    insertInReqCache(req)  
} ELSE {  
    ignoreReq(req)  
}
```

68%

LOOKUP SERVER

POSITIVE EDGE'S  
NEIGHBOURS

BLOOM FILTERS

TTL--

TTL--

TTL--

TTL--

EDGE

NEGATIVE EDGE'S  
NEIGHBOURS

TTL--

TTL--

TTL--

Not enough  
contacted?

# LOOKUP DEL FILE

MECCANISMO DI CALLBACK

73%

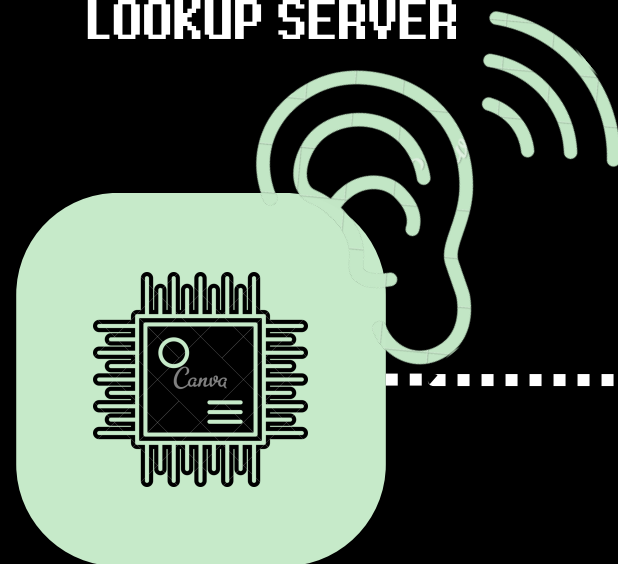
*I have it! Here is my ip...* 



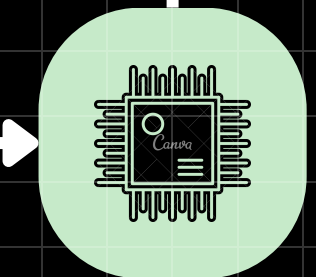
 *N SECONDS LATER...*

*Hey you! Give me the file...*

LOOKUP SERVER



EDGE



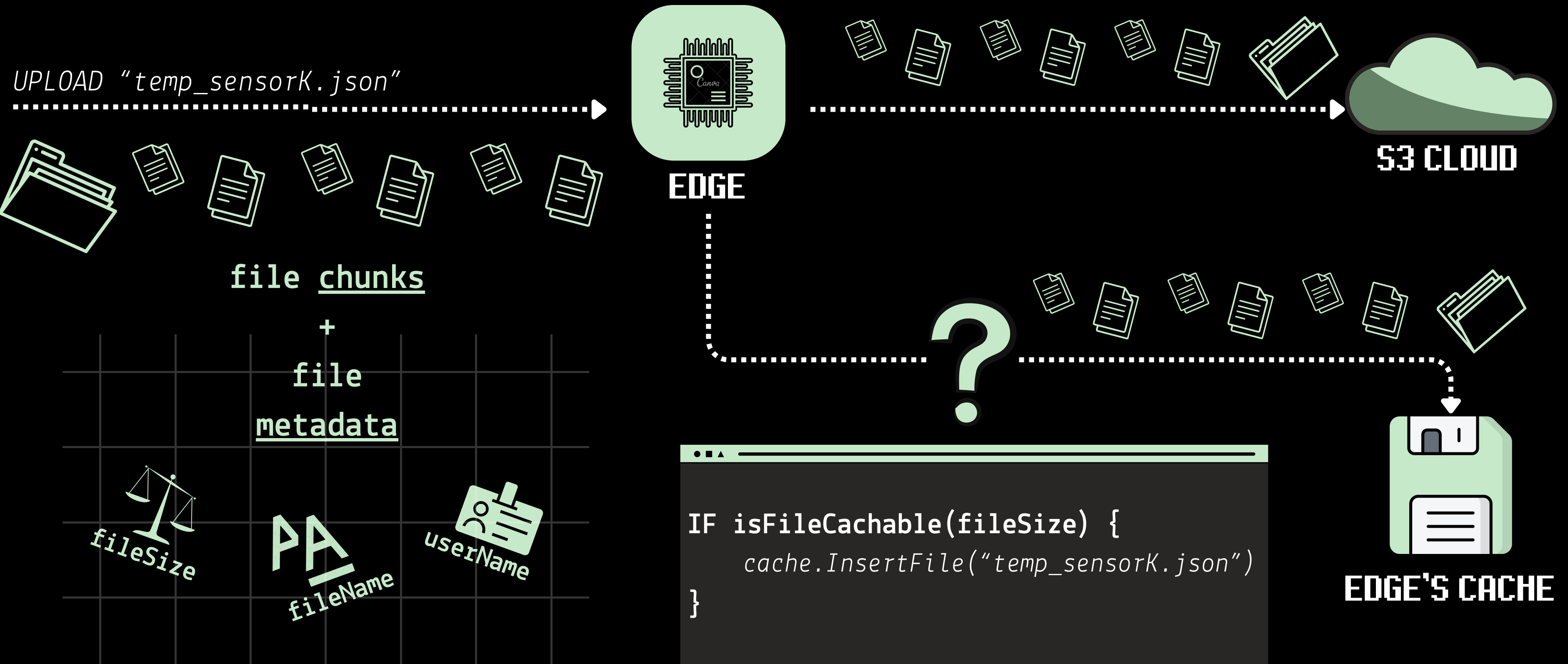
OWNER EDGE



FILE CHUNKS

# UPLOAD REQUEST

77%

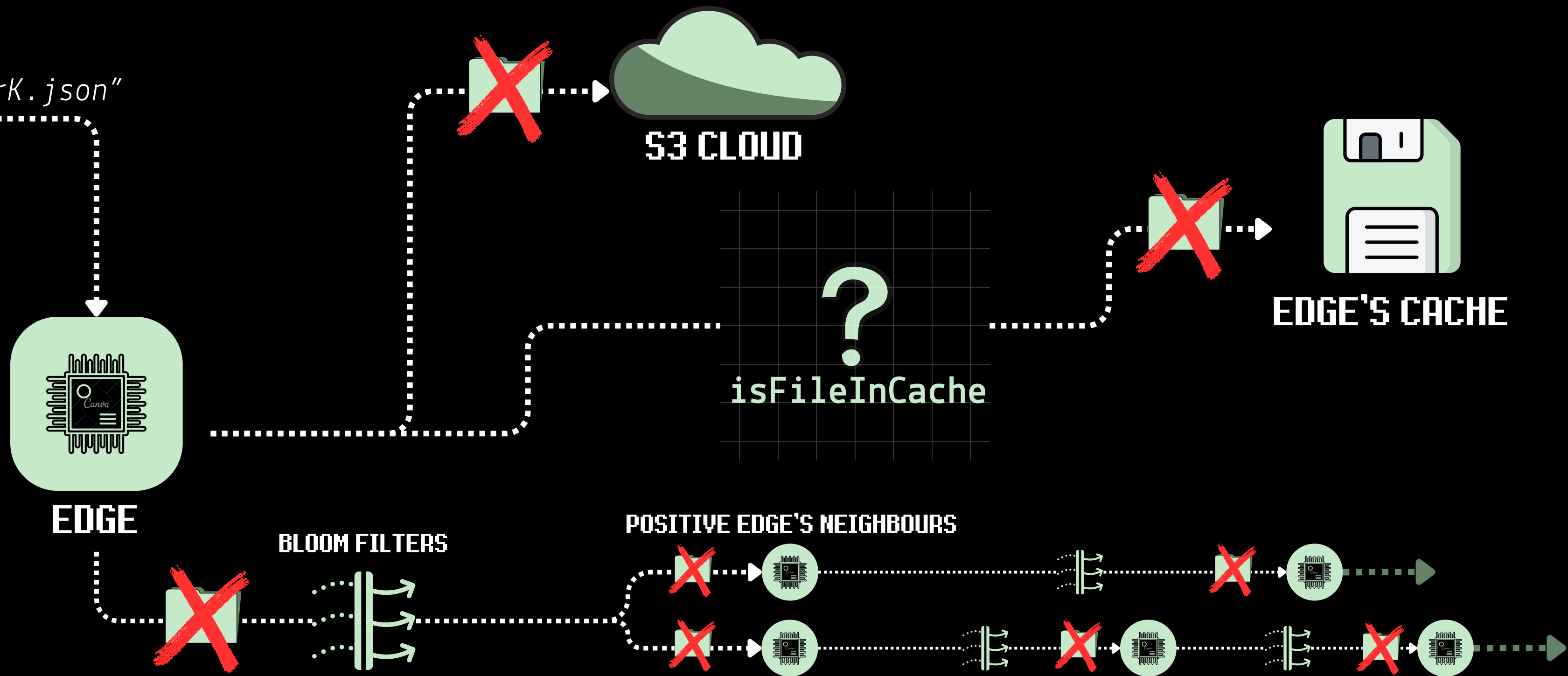


# DELETE REQUEST

82%

DELETE

"temp\_sensorK.json"

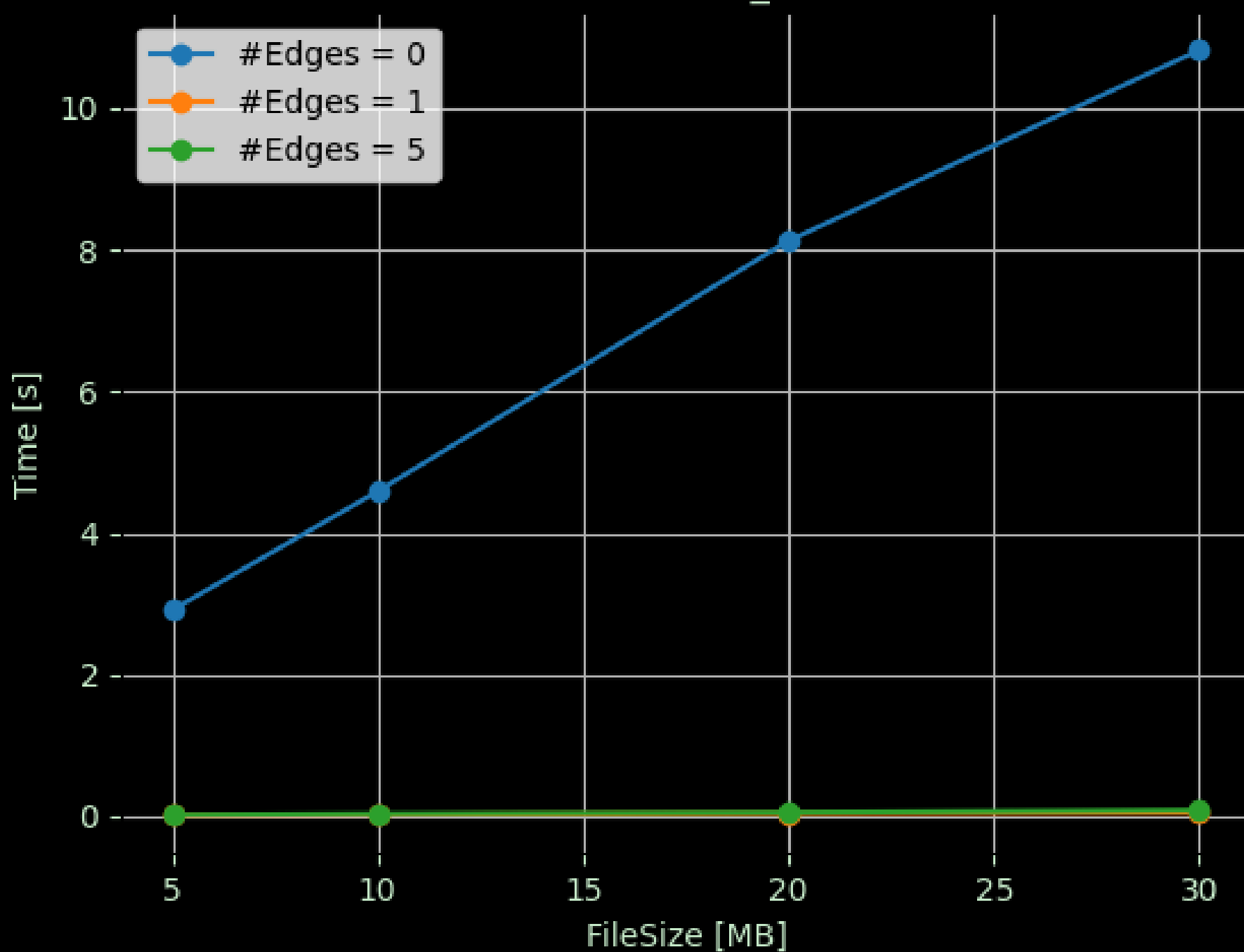


# RISULTATI

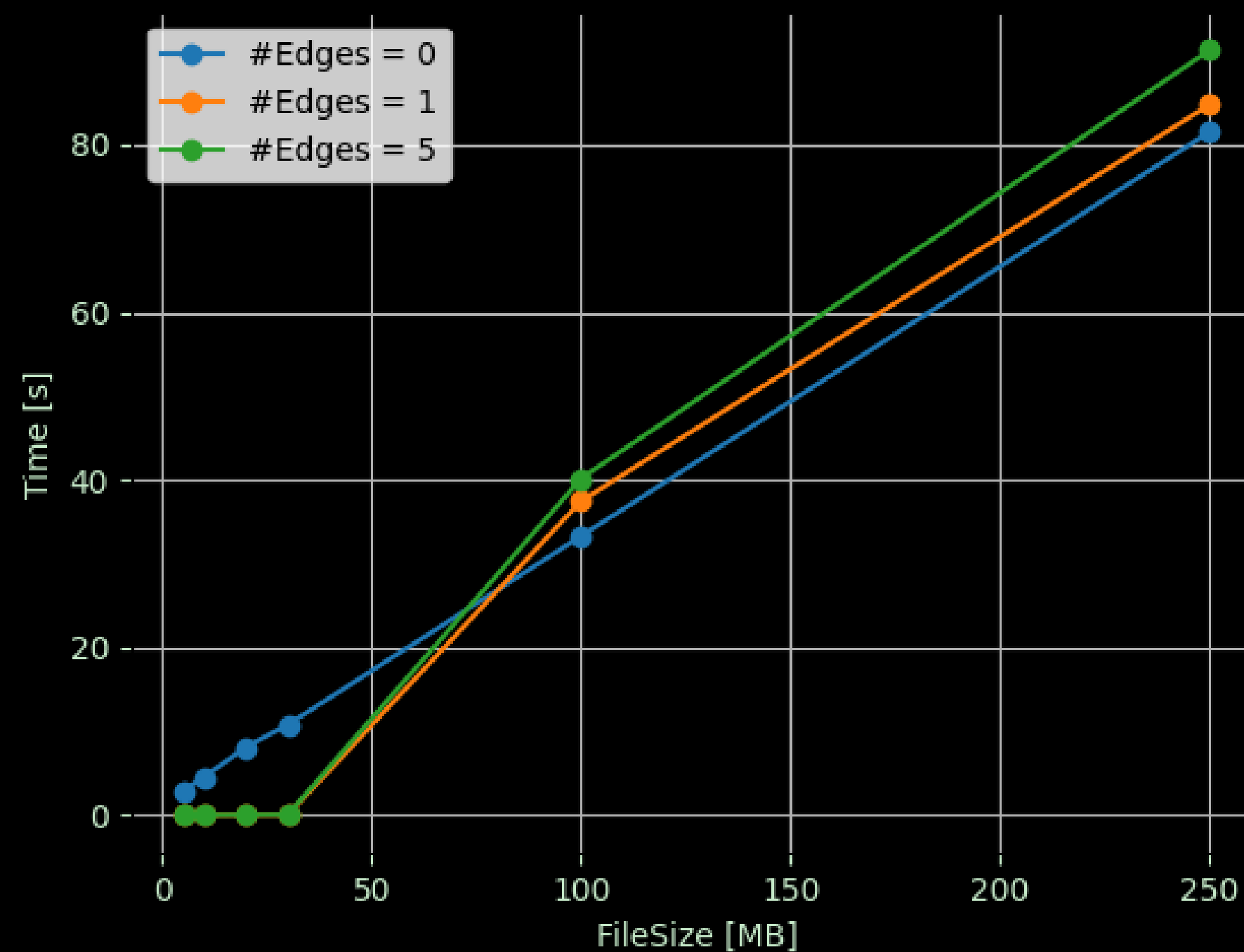
TEST SEQUENZIALI

86%

DOWNLOAD\_CUT



DOWNLOAD

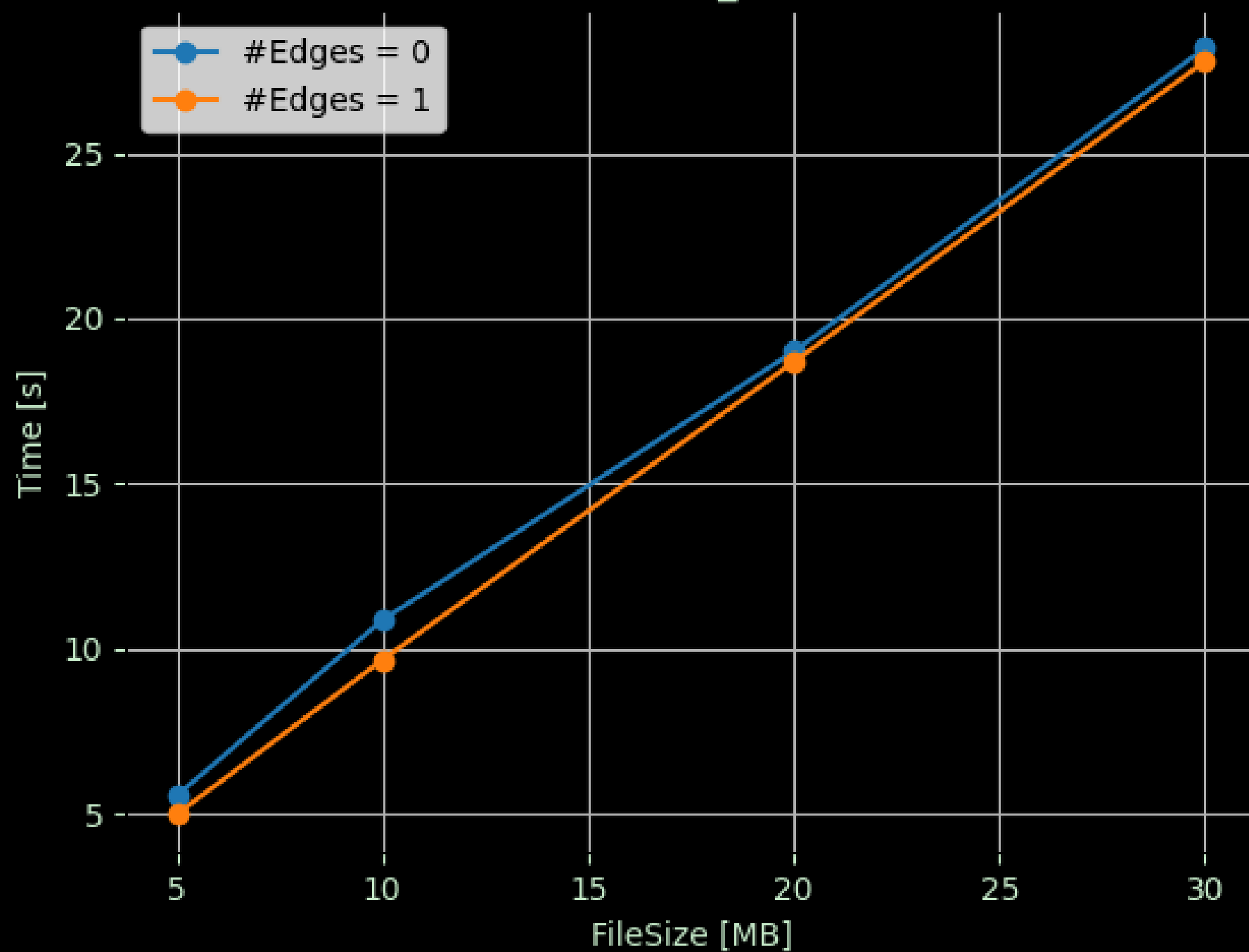


# RISULTATI

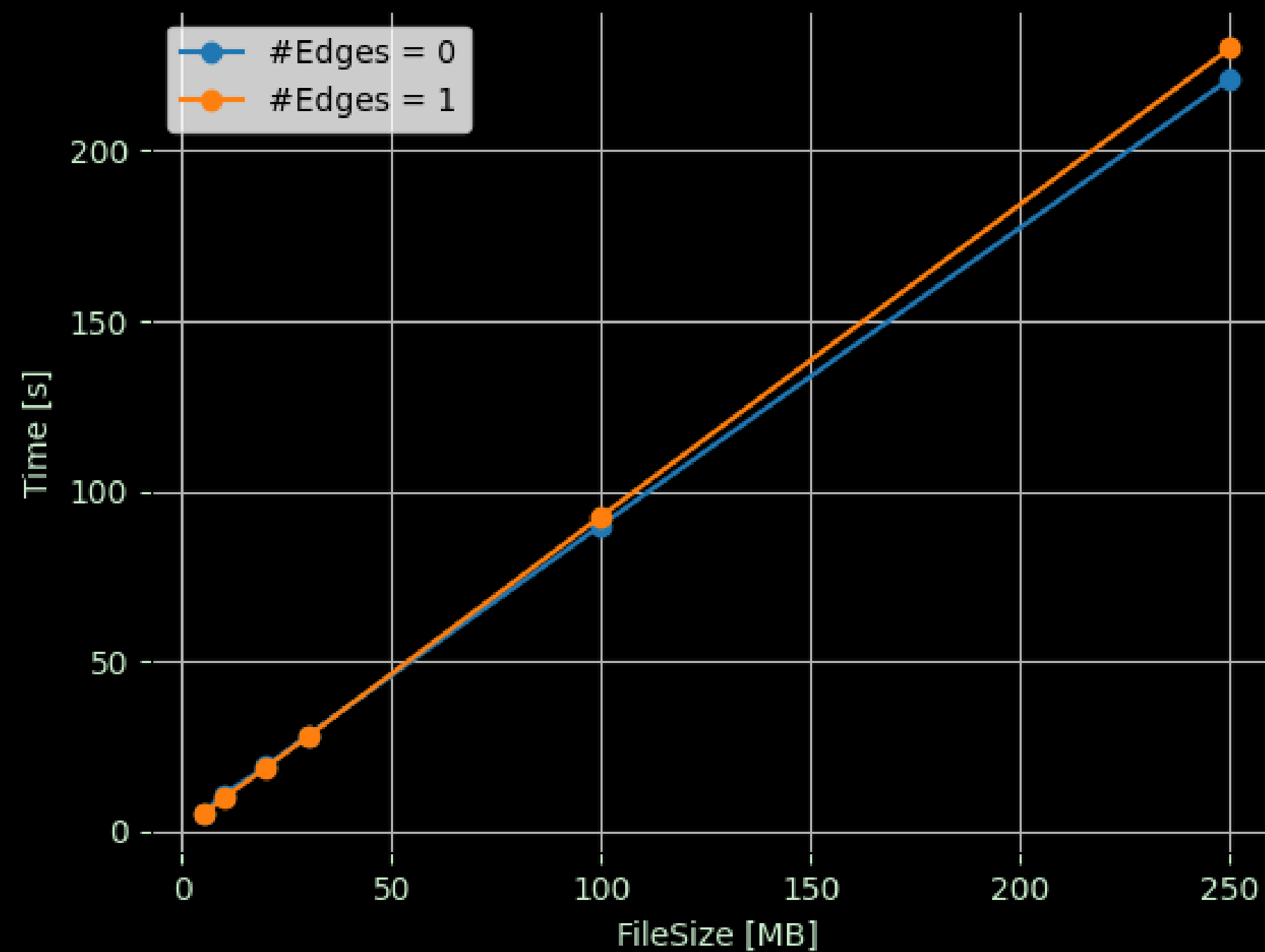
TEST SEQUENZIALI

91%

UPLOAD\_CUT



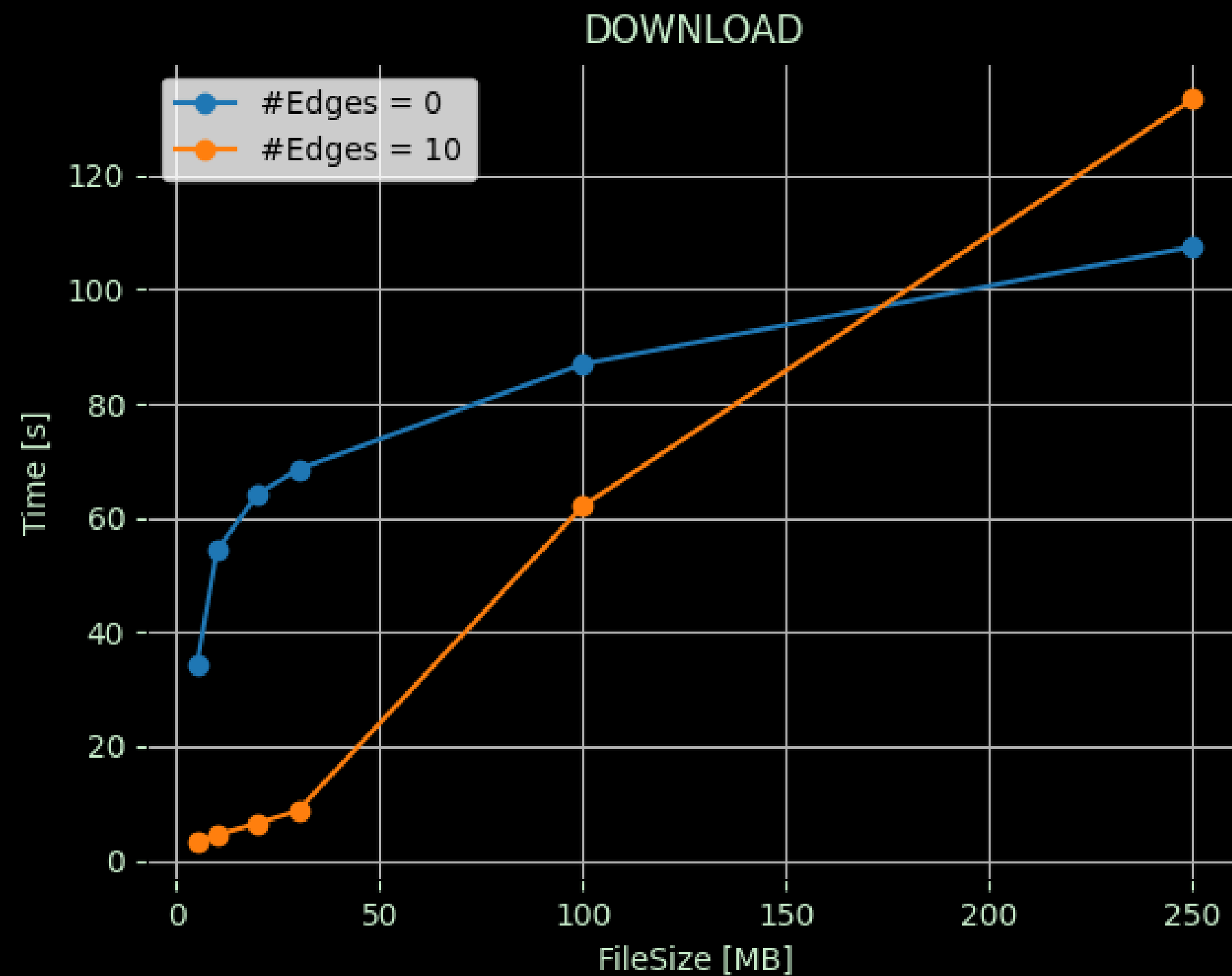
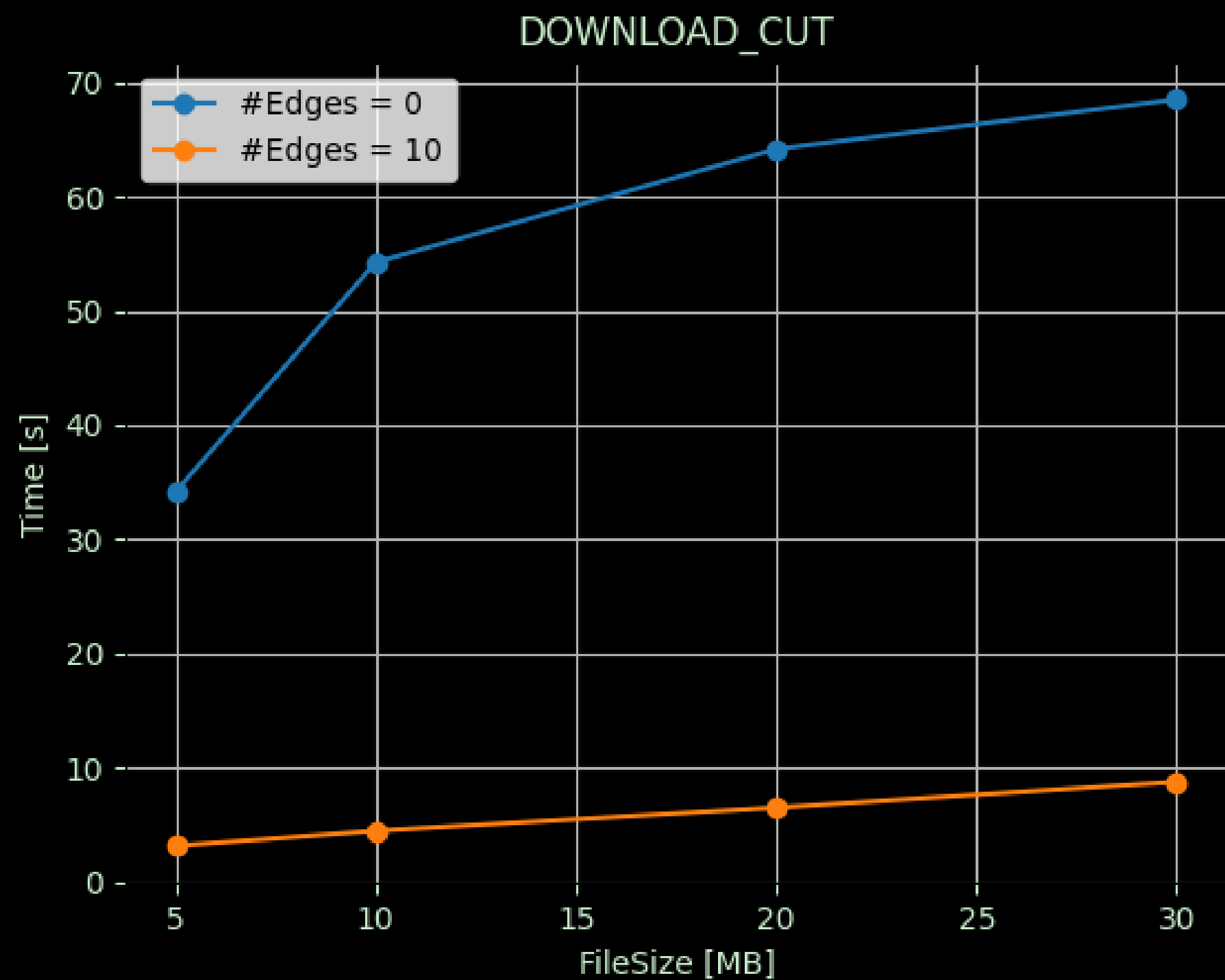
UPLOAD



# RISULTATI

TEST DI CARICO

95%



# SVILUPPI FUTURI

100%

- Supporto per il *versioning* sui file nel sistema.
- Miglioramento del meccanismo di *recovery* sui file scaricati da *Owner Edges* in maniera da supportare la ripresa dello scaricamento dal punto di interruzione.
- Ulteriori test per trovare la configurazione ottimale dell'insieme di parametri configurabili per il sistema *SAE Storage System*.

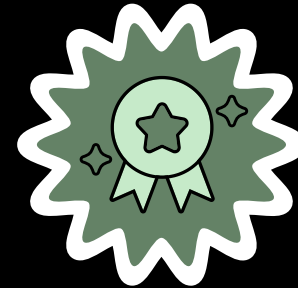


Manenti Edoardo

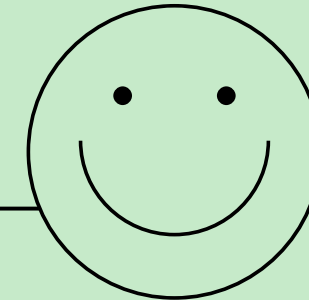
Simone Nicosanti

Andrea De Filippis

**GAME COMPLETED**



A.A. 2022/2023



**GRAZIE DELL'ATTENZIONE**



Presentazione del progetto  
*"SAE storage service"*



hope you liked it!