

01

# ICN WITH EDGE FOR 5G

Exploiting in-network  
caching in ICN-based edge  
computing for 5G networks

---

**Luca Saverio Esposito**

0334321

---

**Mobile System and Applications**

# Indice

**Suggerimento:** clicca sul nome di una pagina per raggiungerla

## INTRODUZIONE

[Introduzione](#)

[ICN](#)

[Edge computing](#)

[Problema](#)

[Obiettivi e sfide](#)

## ARCHITETTURA

[Architettura](#)

[Caching gerarchico](#)

[D2D communication](#)

[ICN implementation](#)

[Content prefetching](#)

[Prefetching naming scheme](#)

[Prefetching strategies](#)

## PERFORMANCE

[Ambiente simulato](#)

[Criteri di valutazione](#)

[Risultati](#)

## ANALISI

[Adaptation managment](#)

[Adaptation implementation](#)

[Aspetti funzionali](#)

[Data staging](#)

[Prefetching](#)

[Surrogate provisioning](#)

[Surrogate discovery](#)

[Aspetti non funzionali](#)

## CONCLUSIONI

[Possibili miglioramenti](#)

[Conclusioni](#)

03

# Introduzione

Combinare i paradigmi

- **ICN (Information-centric Networking)**
- **Edge computing**

Per sfruttare

- **Meccanismi di caching**
- **Data locality**

Col fine di migliorare le prestazioni del 5G riducendo gli accessi al core network.

[Torna all'indice](#)

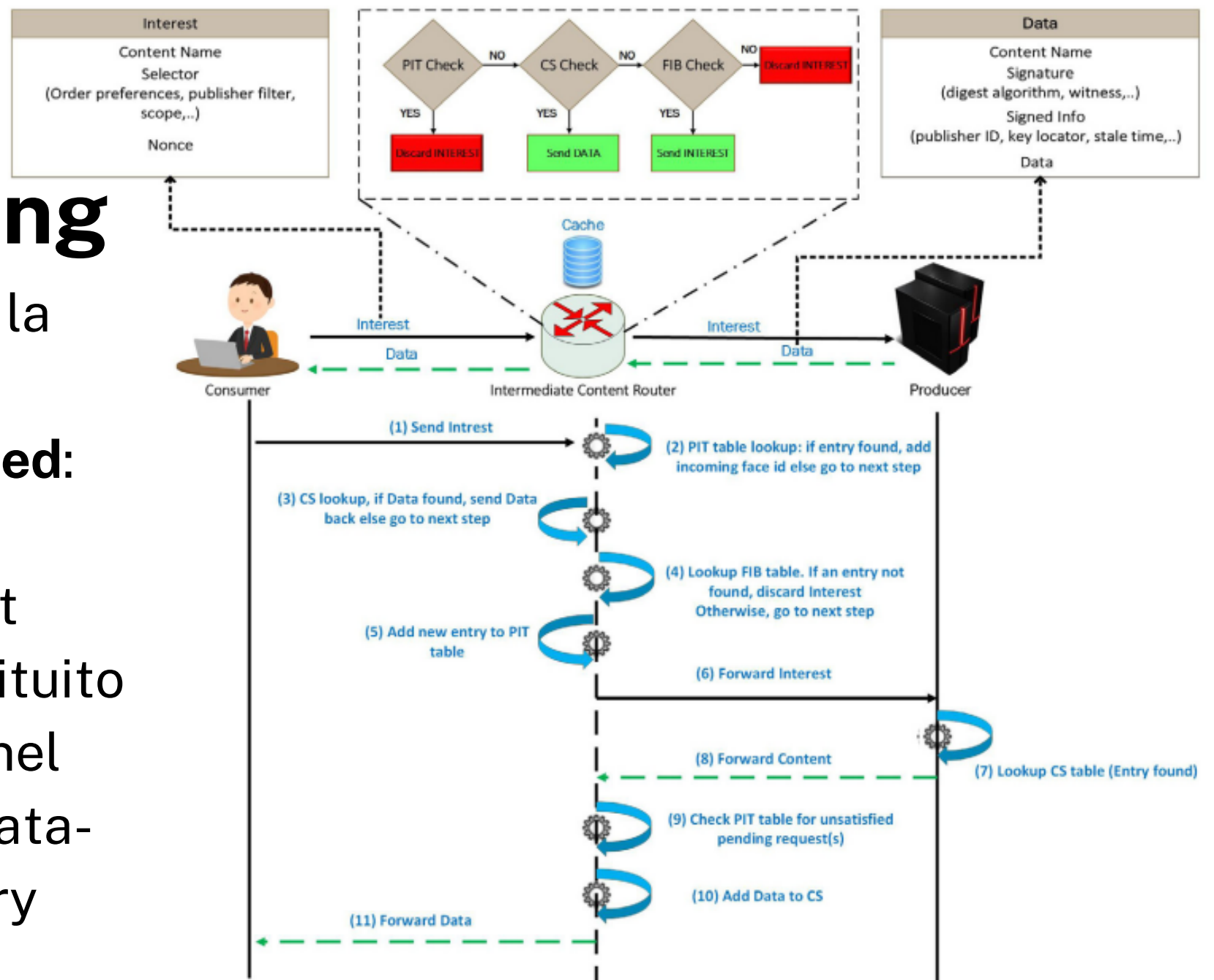
# ICN

## Information-centric networking

Un nome per ogni dato/contenuto, senza sapere la locazione fisica del provider.

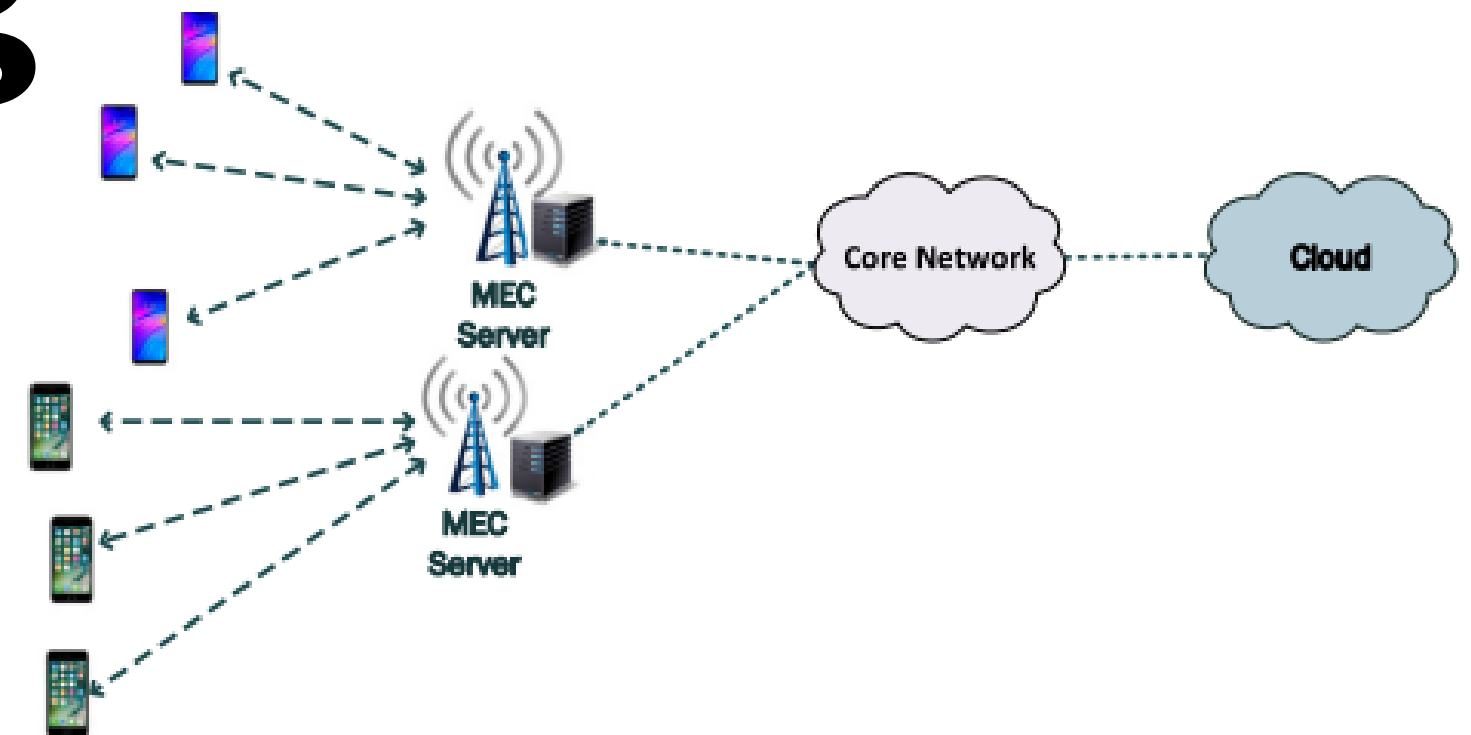
**NDN (Name Data Networking) modello pull-based:**

- Il consumer invia un Interest packet
- Se il dato è all'interno del PIT, scarta interest
- Altrimenti data-check su CS, in tal caso restituito
- Oppure inoltrata tramite interfacce salvate nel FIB, raggiunge data producer restituisce il data-packet, si salva nel CS e si aggiunge PIT entry



# Edge computing

- Spostare i servizi cloud vicino agli end-user, per cercare di ridurre accessi al core-network
- **MEC** (Multi-access edge computing) è l'architettura di tipo edge utilizzata. Utilizza **meccanismi di caching** nei MEC server per ridurre tempo di accesso alle risorse



06

# Internet è in continua evoluzione

[Torna all'indice](#)

- 
- Quantità e diversità di device
  - Tipologia di dati trasmessi
  - Durata delle connessioni

- 
- Smartphone, tablet, dispositivi IoT
  - Multimediali
  - Brevi e frequenti
  - Circa 50 exabytes ( $10^{18}$ ) nel 2021

# Obiettivi e sfide

[Torna all'indice](#)

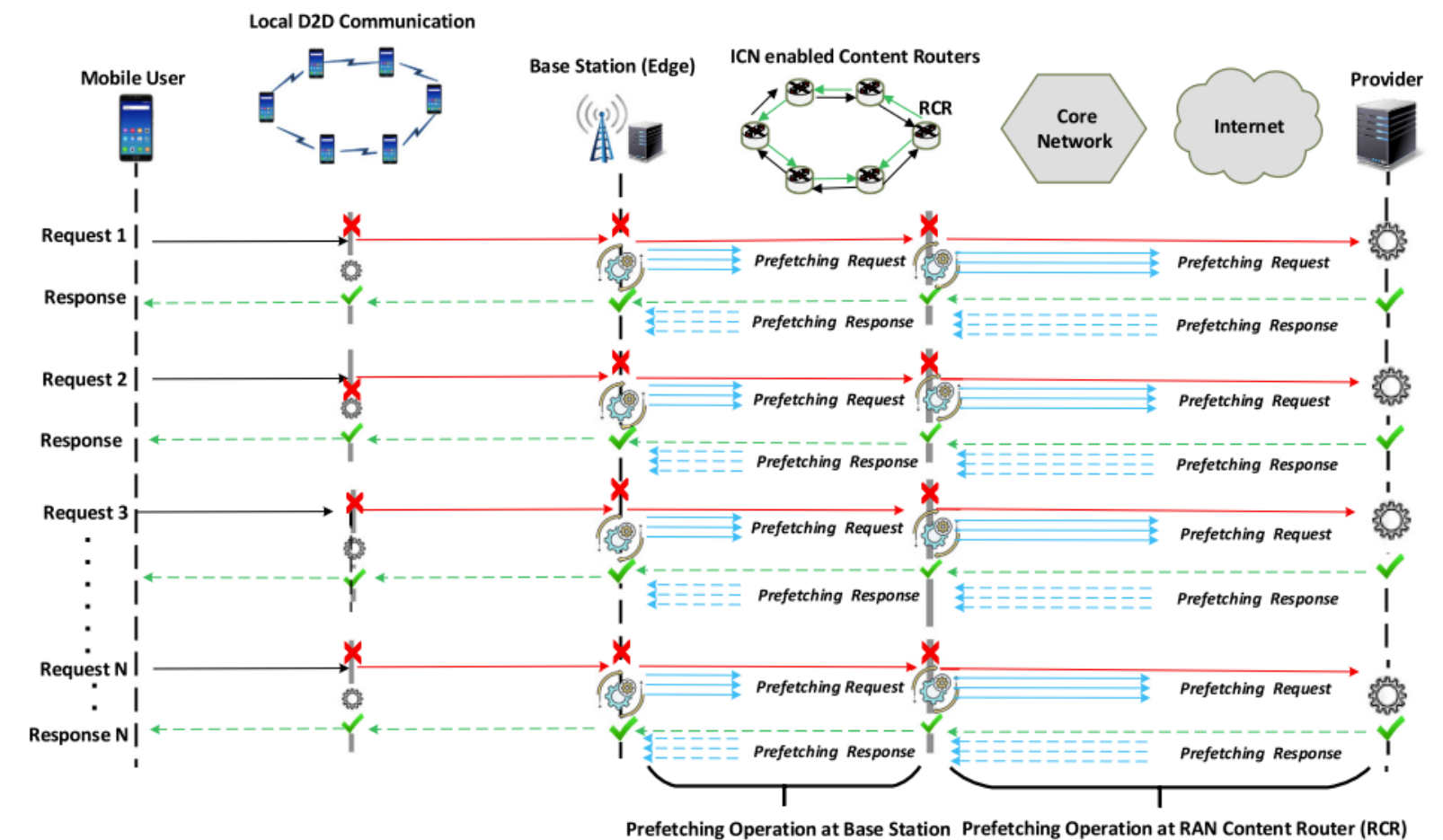
Obiettivi	Sfide
<ul style="list-style-type: none"><li>• Sfruttare la capacita di caching di ICN a livello di device e a livello di base stations</li><li>• Portate i contenuti vicino agli end-user per ridurre gli accessi al core network</li></ul>	<ul style="list-style-type: none"><li>• Compensare handoff delay prodotto da ICN quando viene abilitato a livello di device</li><li>• Gestione del contenuto dinamico, non adatto ai meccanismi di caching</li></ul>

# Architettura

Un mobile user alla ricerca di un contenuto invia un interest request. Questa si propaga tra i vari livelli dell'architettura.

- **Nearby mobile device**
- **Base Station**
- **ICN-enabled routers**
- **Cloud provider**

La richiesta si ferma ad un livello intermedio non appena c'è una hit nella cache.





# Architettura

## Caching gerarchico

[Torna all'indice](#)

### Device

Il mobile user invia l'interest packet ai device vicini, cercano nella loro cache il contenuto. In caso positivo lo inoltrano altrimenti contattano la BS.

### Base Station

La richiesta arriva alla base station, cerca nel suo CS per il contenuto. Se lo trova lo inoltra al mobile user, altrimenti, invia la richiesta agli ICN-routers.

### ICN-router

Controllano nella loro cache se è presente il contenuto. Se lo trovano lo restituiscono al mobile user altrimenti contattano il provider/cloud via core network.

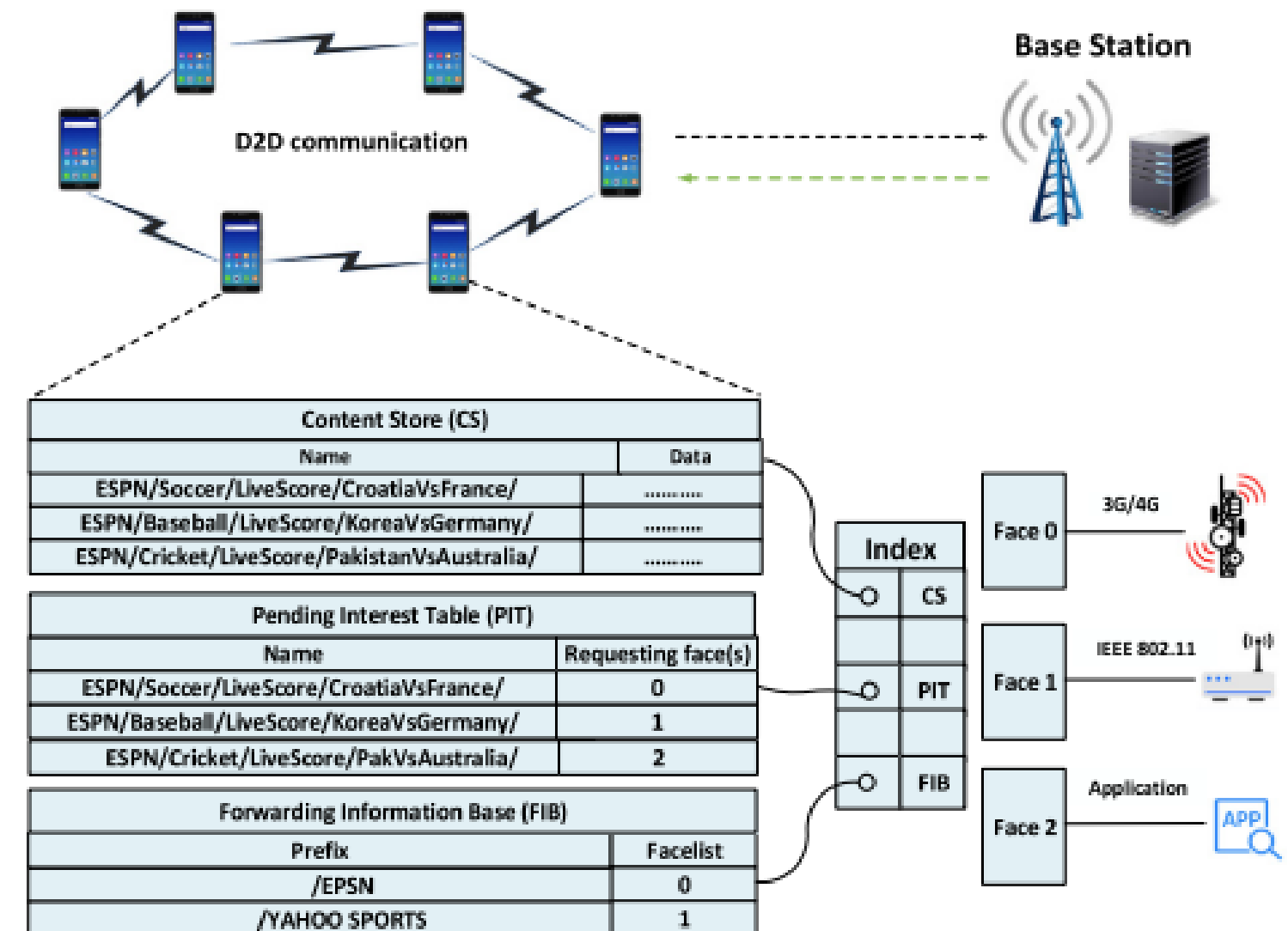
# Architettura

## D2D Communication

La comunicazione avviene sotto due assunzioni.

- **No privacy-issue:** durante la trasmissione, tutti i device sono autenticati
- **Friendly environment:** risorse liberamente disponibili per gli altri device

Solo quando la risorsa non è disponibile tra i device si contatta la base station.



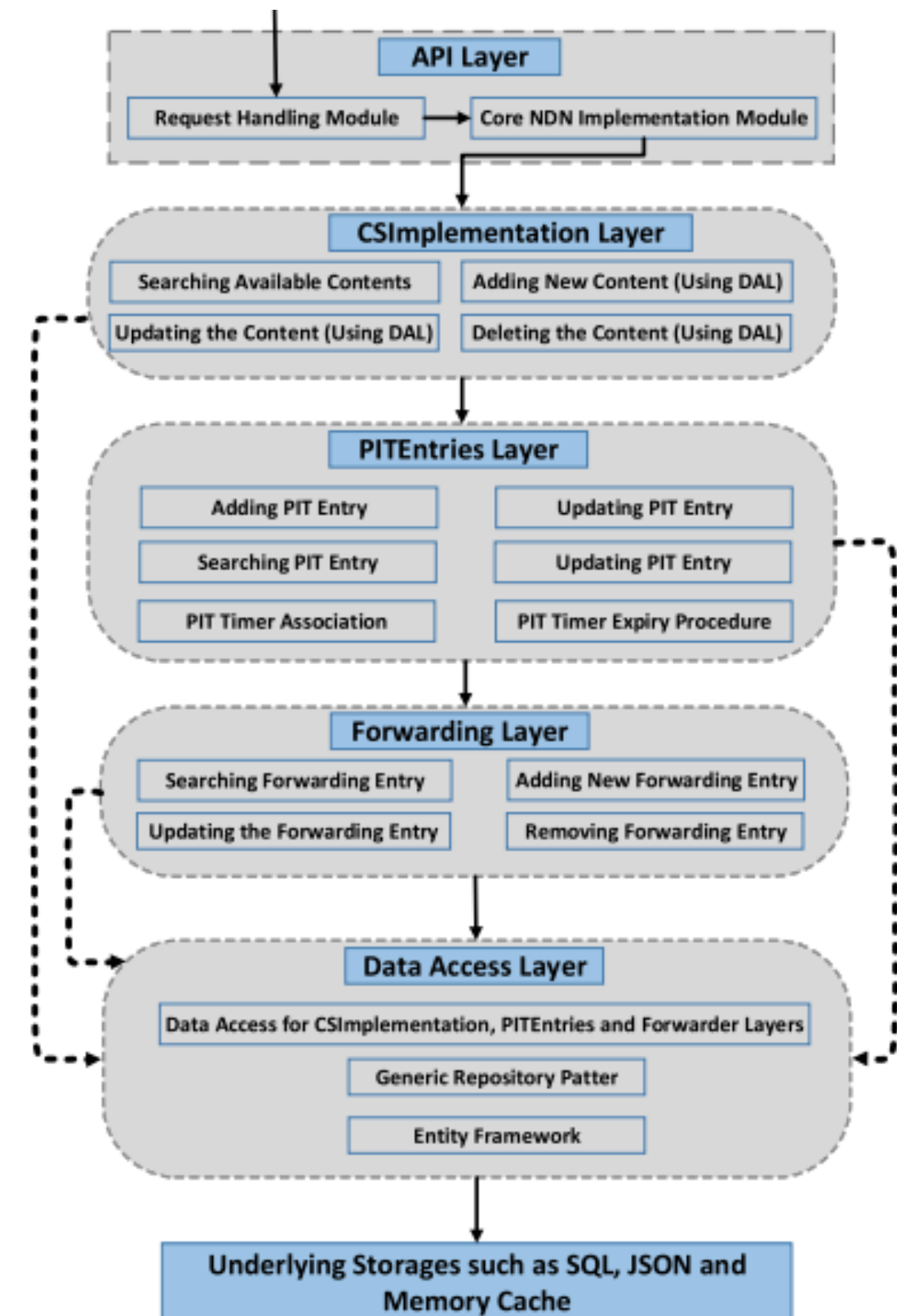
# Architettura

## ICN implementation su BSs

Per utilizzare le strutture dati di ICN sulle base stations, implementazione su application layer. Composto da vari livelli.

- API layer
- CS implementation layer
- PIT entries layer
- Forwarding layer
- DAL (Data access layer)

[Torna all'indice](#)



# Architettura

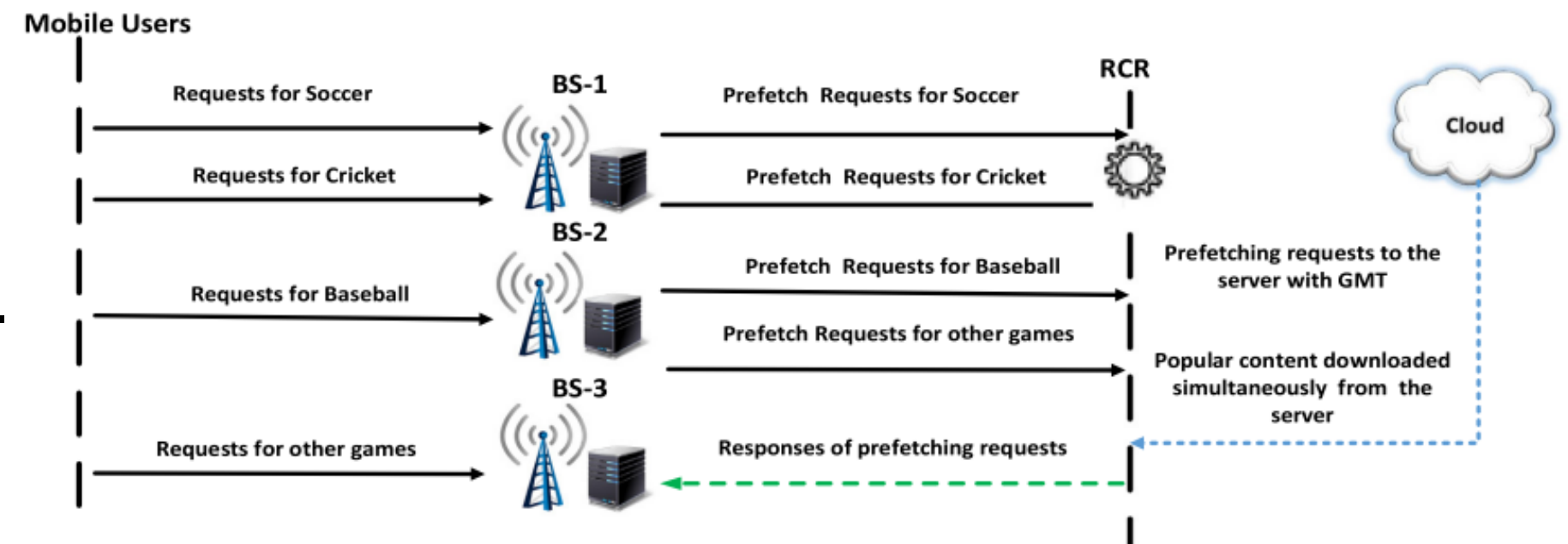
## Content prefetching

I contenuti dinamici cambiano nel tempo, utilizzare caching non è l'approccio corretto.

Strategia di **prefetching** su

- Base stations
- RCR (RAN content router) ossia ICN router connesso al core network

Basata sulla **popolarità delle richieste**.

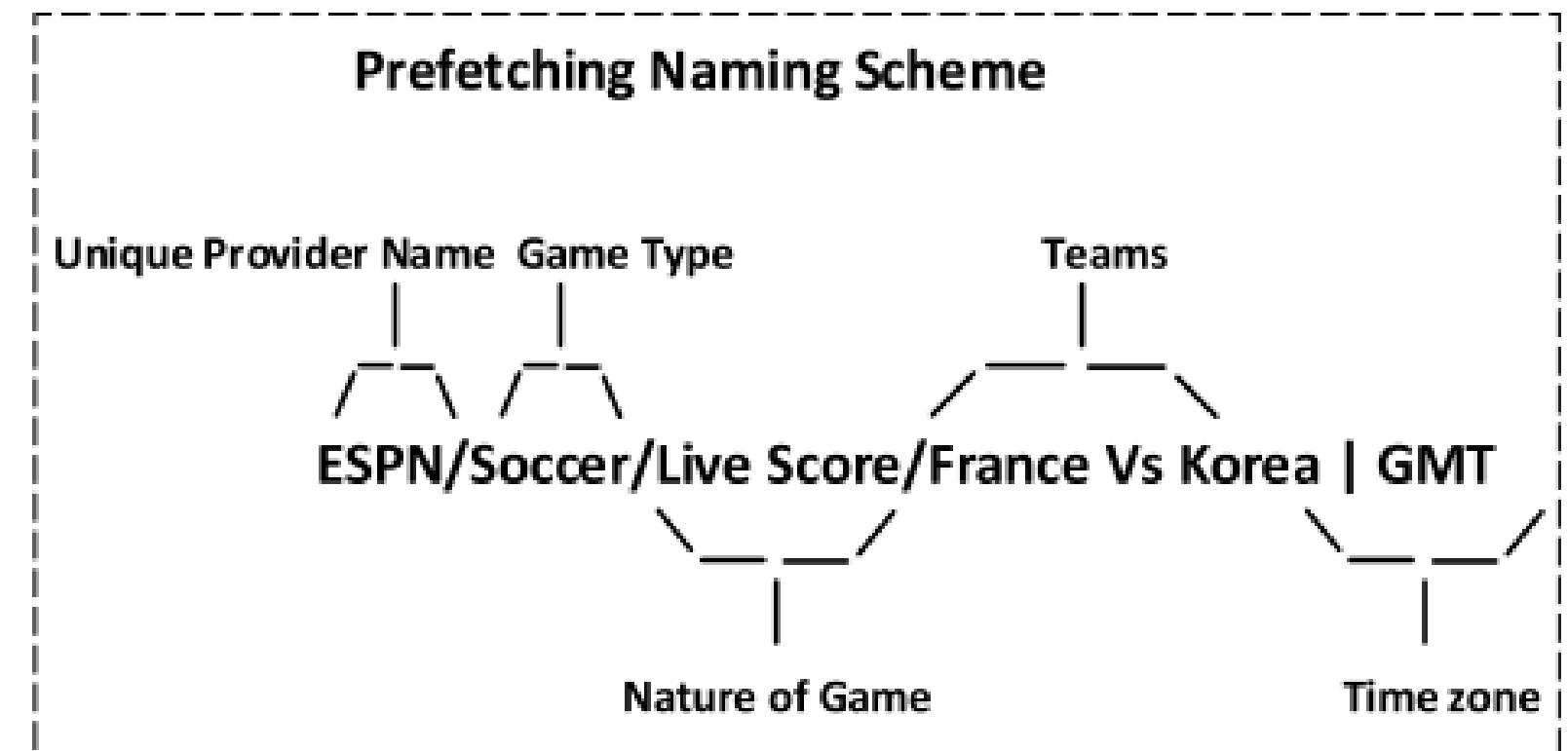


# Architettura

## Prefetching naming scheme

Lo schema comprende diverse componenti.

- Provider name (univoco)
- Nome del contenuto
- Natura dei dati
- Entità coinvolte
- Time zone



[Torna all'indice](#)

# Architettura

## Prefetching strategies

### Base station

Ogni edge node misura la frequenza delle richieste. Superata una certa threshold, esegue prefetch dell'ultima versione del contenuto. Questo viene salvato nel suo CS.

### RAN content router

Anche il nodo RCR (RAN content router) misura la frequenza delle richieste. Tuttavia provengono da differenti BSs, quindi profili d'uso differenti e maggior traffico.

Le possibili politiche di aggiornamento.

- LRU (Least recently used)
  - LFU (Least frequently used)
-

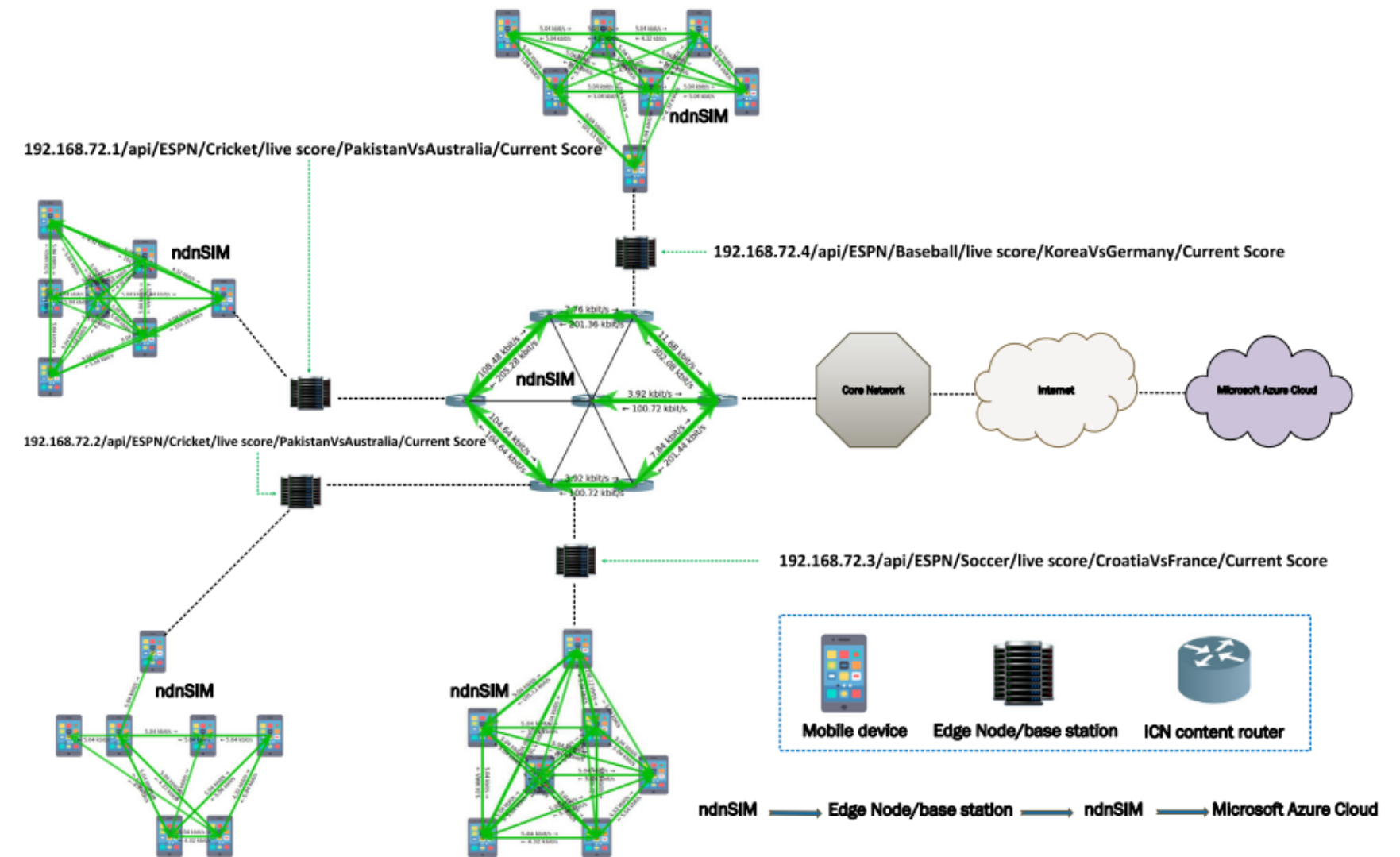
# Performance

## Ambiente simulato

L'ambiente è stato realizzato mediante

- ndnSIM per le richieste da mobile-device
- Edge node/base station
- ndnSIM per simulare ICN content router
- Microsoft Azure Cloud come provider

Utilizzati cataloghi da 1000 a 5000 dati, su diverse threshold per un tempo di 120 sec.



# Performance

## Criteri di valutazione

### Metriche

Per valutare il framework sono state utilizzate

- **Average Cache Hit Ratio (CHR)**  
misura quanti interest-packets una cache soddisfa su quanti sono inviati
- **Average latency** il tempo consumato dai vari dispositivi per adempiere ai data-exchanges

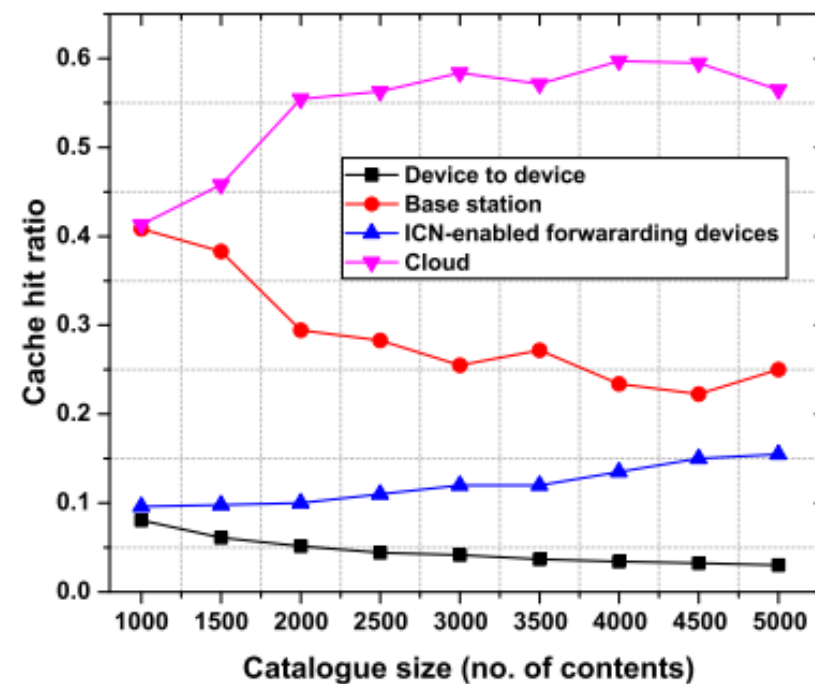
### Confronto

Sono state confrontate le metriche nelle due situazioni

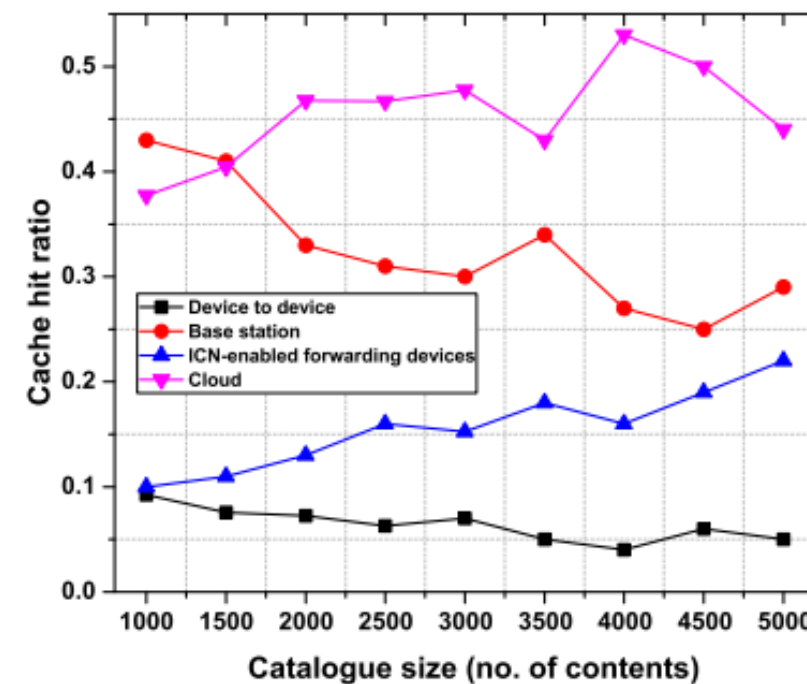
- Prefetching disabilitato
- Prefetching abilitato



No prefetching



Con prefetching



17

# Risultati

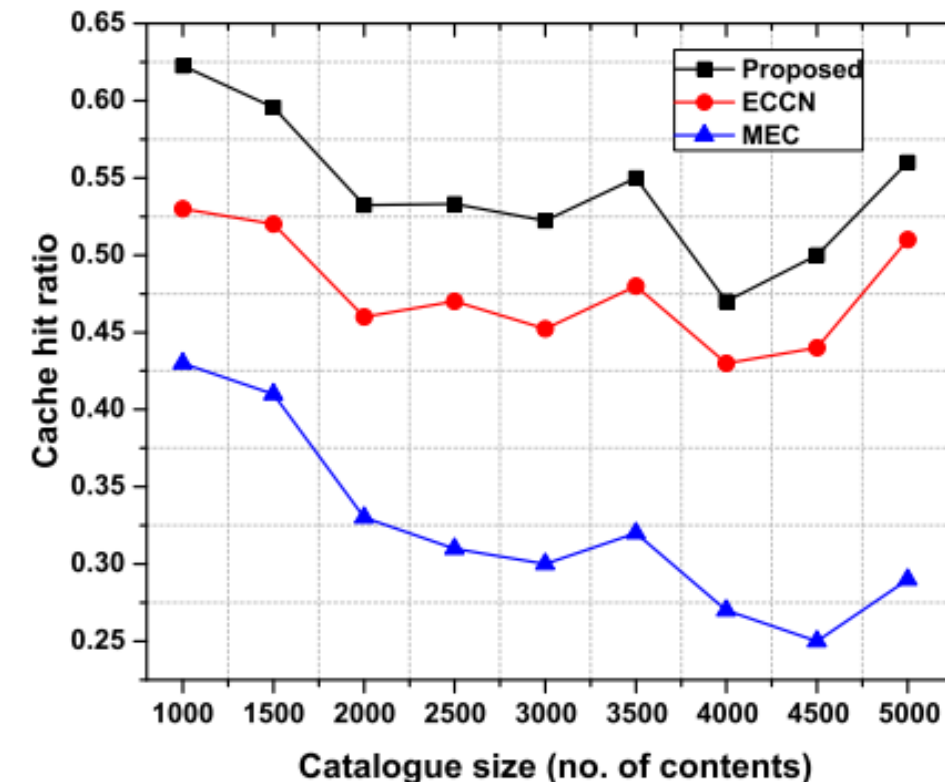
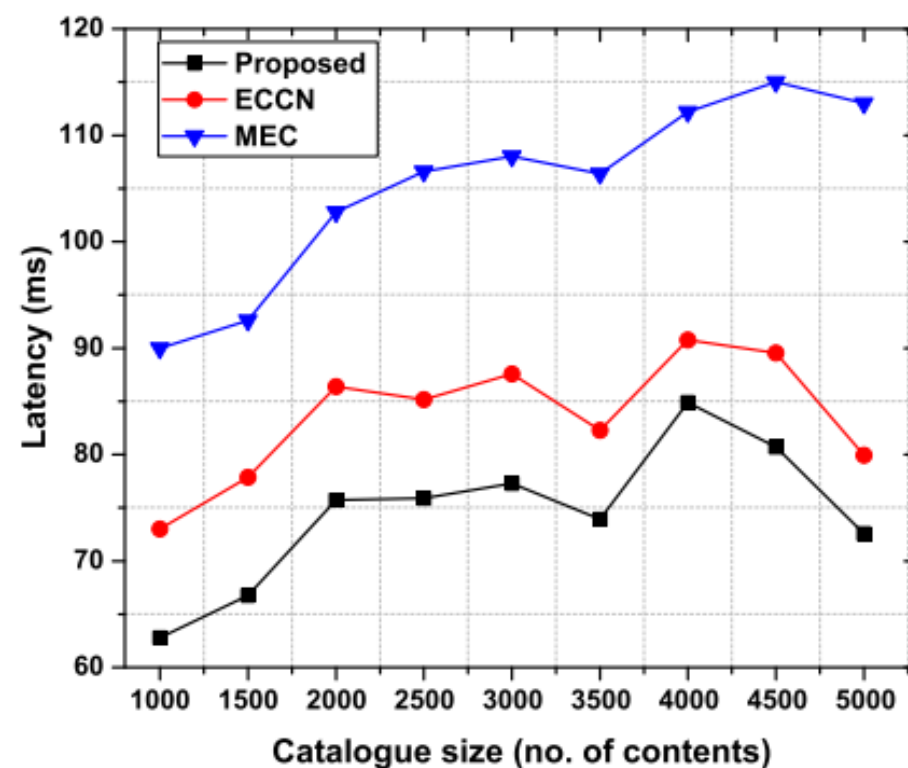
I risultati della simulazione mostrano una riduzione del traffico nel core network, fino ad un 26% con prefetching attivo. Infatti, aumenta la CHR sia sulle BSs sia sugli ICN-router, mentre diminuisce nel Cloud.

Sono state confrontate anche le metriche ottenute su due sistemi esistenti

- ECCN
- MEC

I risultati ottenuti dalla soluzione proposta sono migliori rispetto ad essi.

Si evince un'inversa proporzionalità tra cache hit ratio e latenza.



[Torna all'indice](#)

# Analisi

## Adaptation management

### Adaptation management (“intelligence”)

- ❑ Category: Locus of responsibility
  - (from the application level viewpoint)

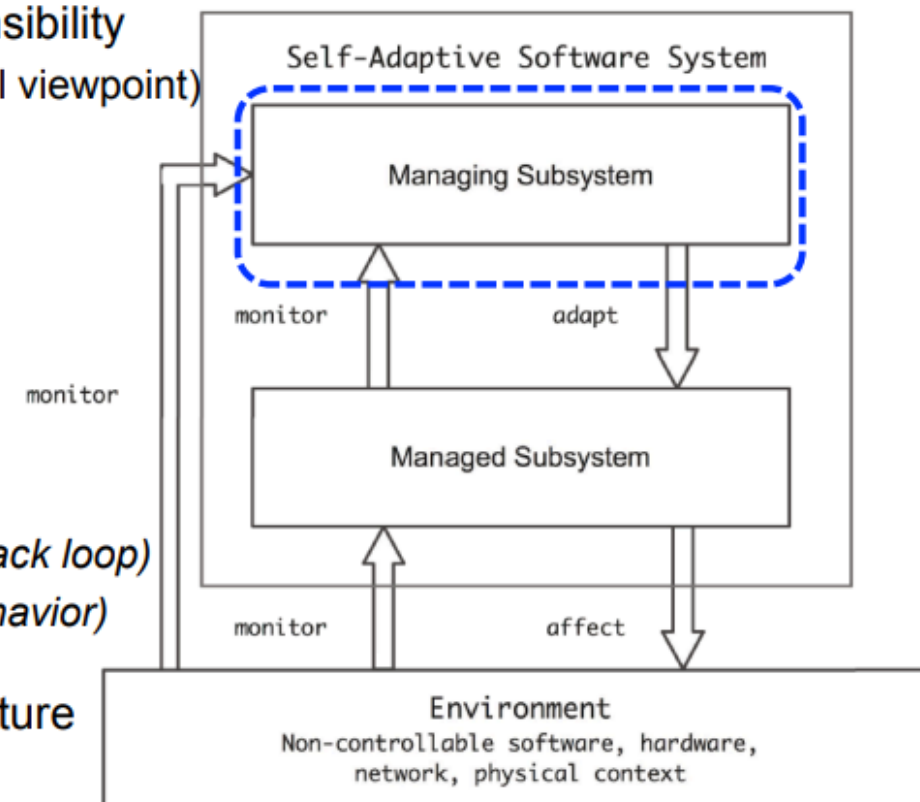
- ❑ Tactics :
  - *total transparency*
  - *total responsibility*
  - *application-aware*

- ❑ Category: Type of control

- ❑ Tactics :
  - *Top-down (explicit feedback loop)*
  - *Bottom-up (emergent behavior)*

- ❑ Category: Control architecture

- ❑ Tactics :
  - *decentralized control*
  - *hierarchical control*



- **locus of responsibility** → application-aware

Sia l'applicazione che il sistema sottostante collaborano per garantire l'adattamento. Es. ICN application layer su BSs, mentre nativo su ICN routers.

- **type of control** → top-down

Feedback control loop parzialmente-decentralizzato

- **control architecture** → hierarchical control pattern

Approccio gerarchico, loop a basso livello operano su una scala limitata di richieste rispetto ai layer superiori.

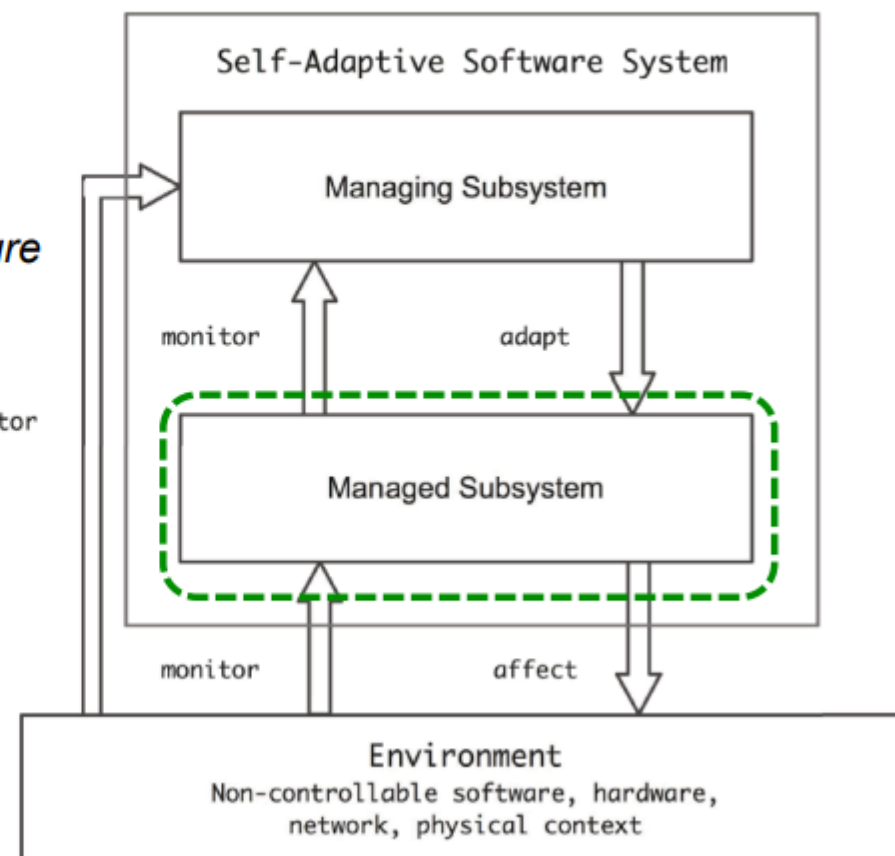
[Torna all'indice](#)

# Analisi

## Adaptation implementation

### Adaptation implementation ("toolset")

- ❑ Category: Malleability
- ❑ Tactics :
  - *variable data fidelity*
  - *loosely coupled architecture*
    - *loose connectors*
    - *loose components*
    - *loose deployment*
- ❑ Category: Cyber foraging
- ❑ Tactics :
  - *computation offload*
  - *data staging*
  - ...



- **malleability** → loosely coupled architecture

L'approccio è di tipo loose connectors.

Request-Response è la soluzione utilizzata.

- **cyber foraging** → funzionali

- Data staging
- Surrogate provisioning
- Surrogate discovery

- **cyber foraging** → non funzionali

Alcuni di questi aspetti sono trattati come future challenges da realizzare.

[Torna all'indice](#)

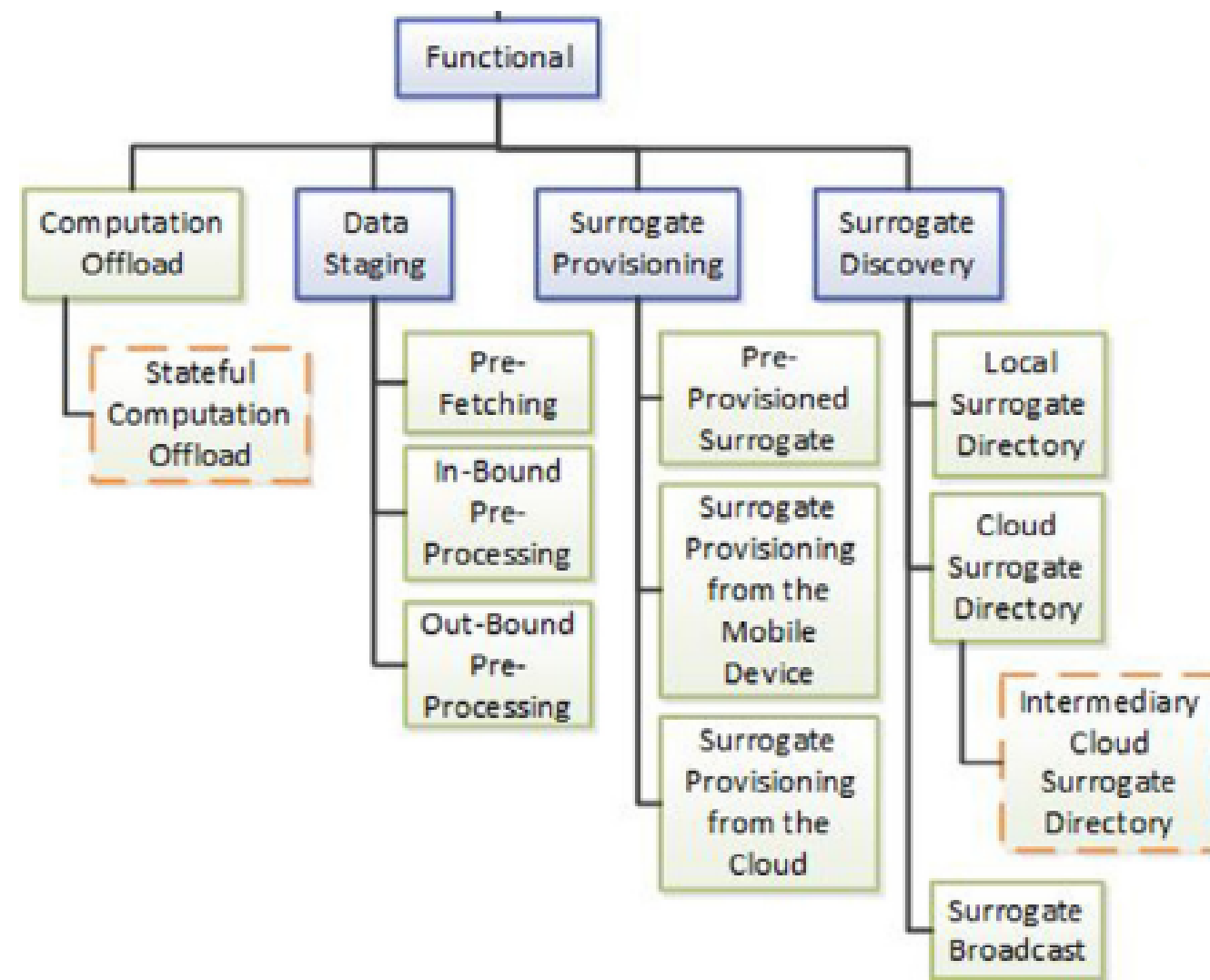
# Analisi

## Aspetti funzionali

E' possibile identificare le soluzioni utilizzate per

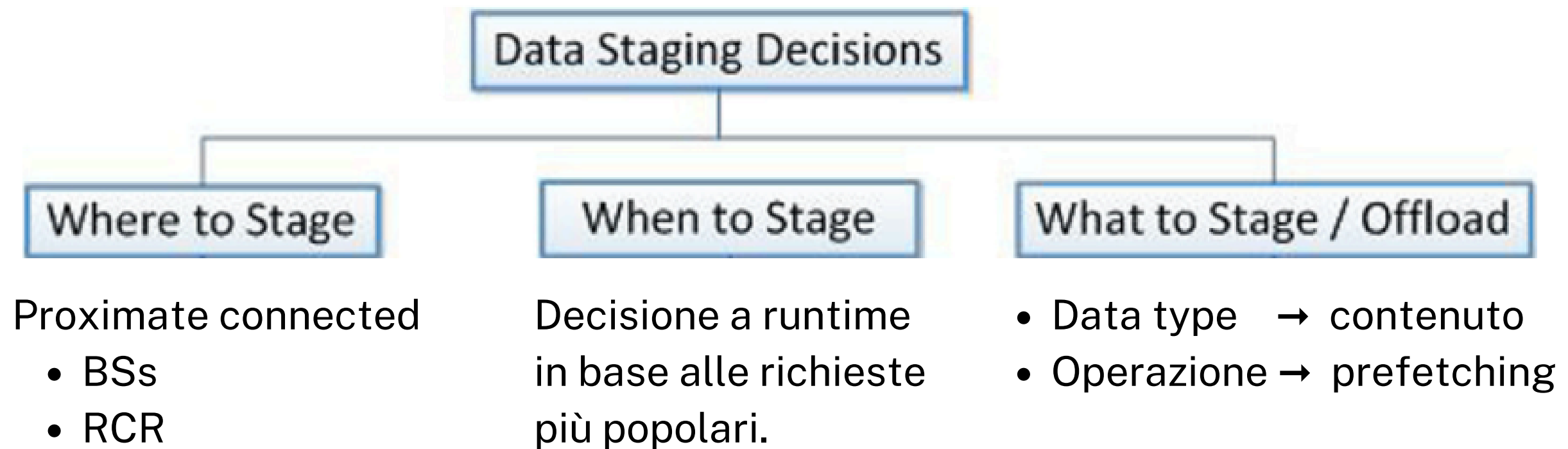
- **Data staging**
- **Surrogate provisioning**
- **Surrogate discovery**

Mentre, fa parte degli aspetti lasciati in sospeso la possibilità di eseguire **computation offload**.



# Analisi

## Data staging decisions



[Torna all'indice](#)



# Analisi

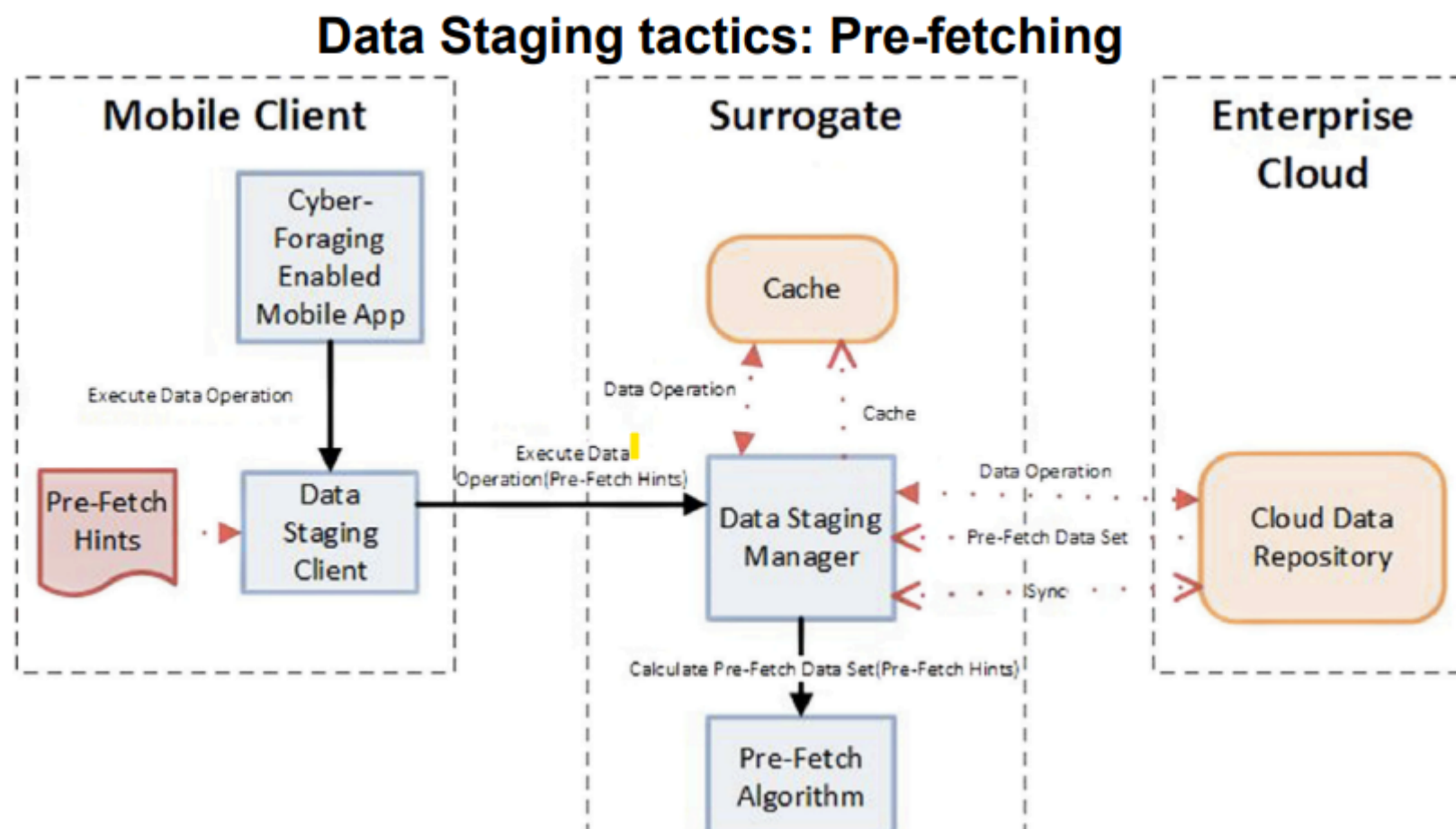
## Data staging

L'obiettivo cardine del sistema è proprio quello di ridurre gli accessi al core network, utilizzando meccanismi di caching e **prefetching**.

I dati dinamici vengono pre-caricati sia sui BSs che sul RAN content router, in modo tale da limitare il numero di richieste verso il cloud-provider.

Questo implica la presenza di

- **surrogate provisioning**
- **surrogate discovery**



[Torna all'indice](#)

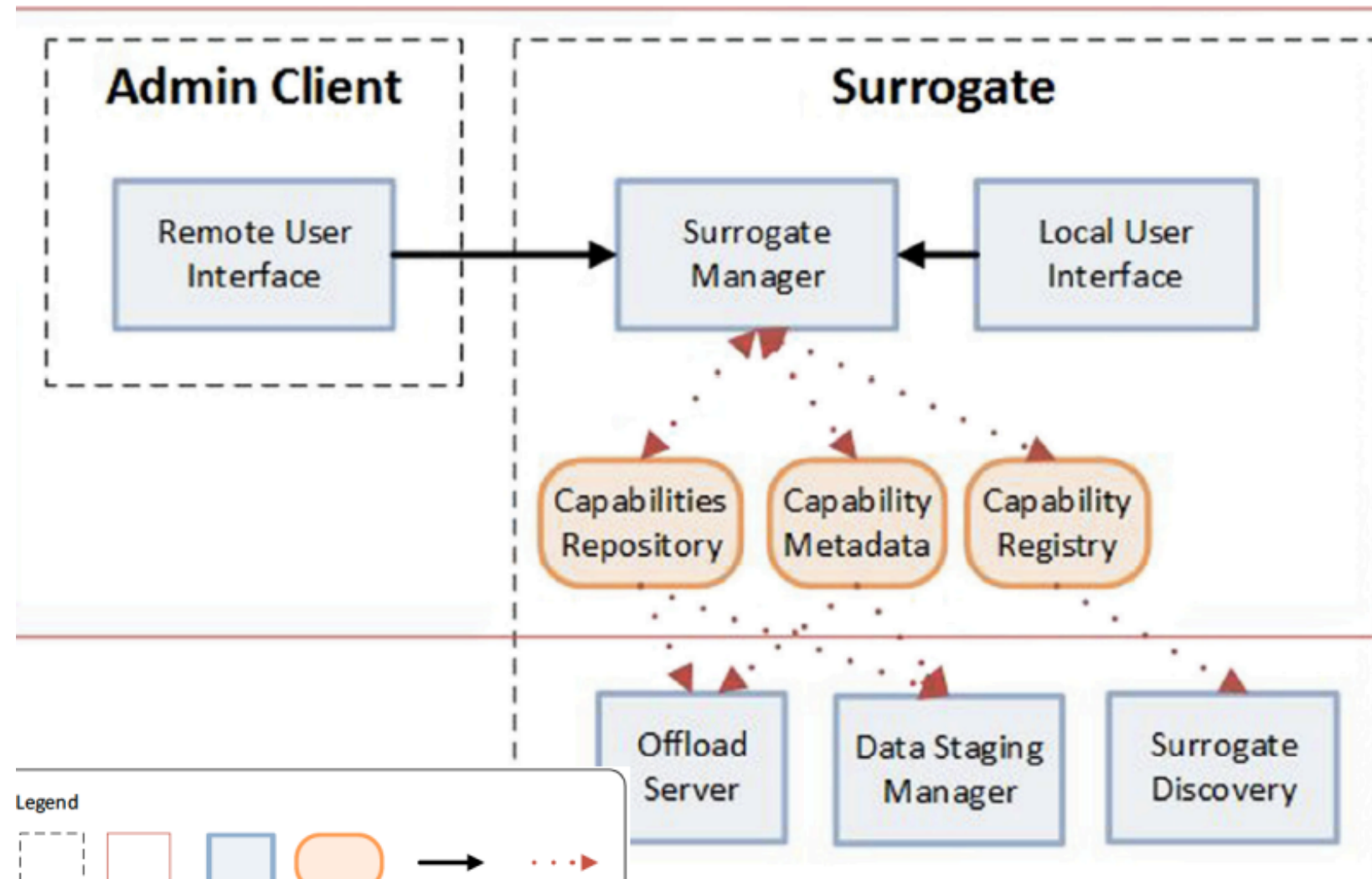
23

# Analisi

## Surrogate provisioning

La struttura del sistema è ben definita. BSs e RCR sono predisposti in anticipo e sono in costante ascolto delle richieste per i vari contenuti. Stesso vale per il cloud provider. Per questo si identifica una soluzione **pre-provisioned surrogate**.

### Surrogate provisioning tactics: Pre-provisioned Surrogate

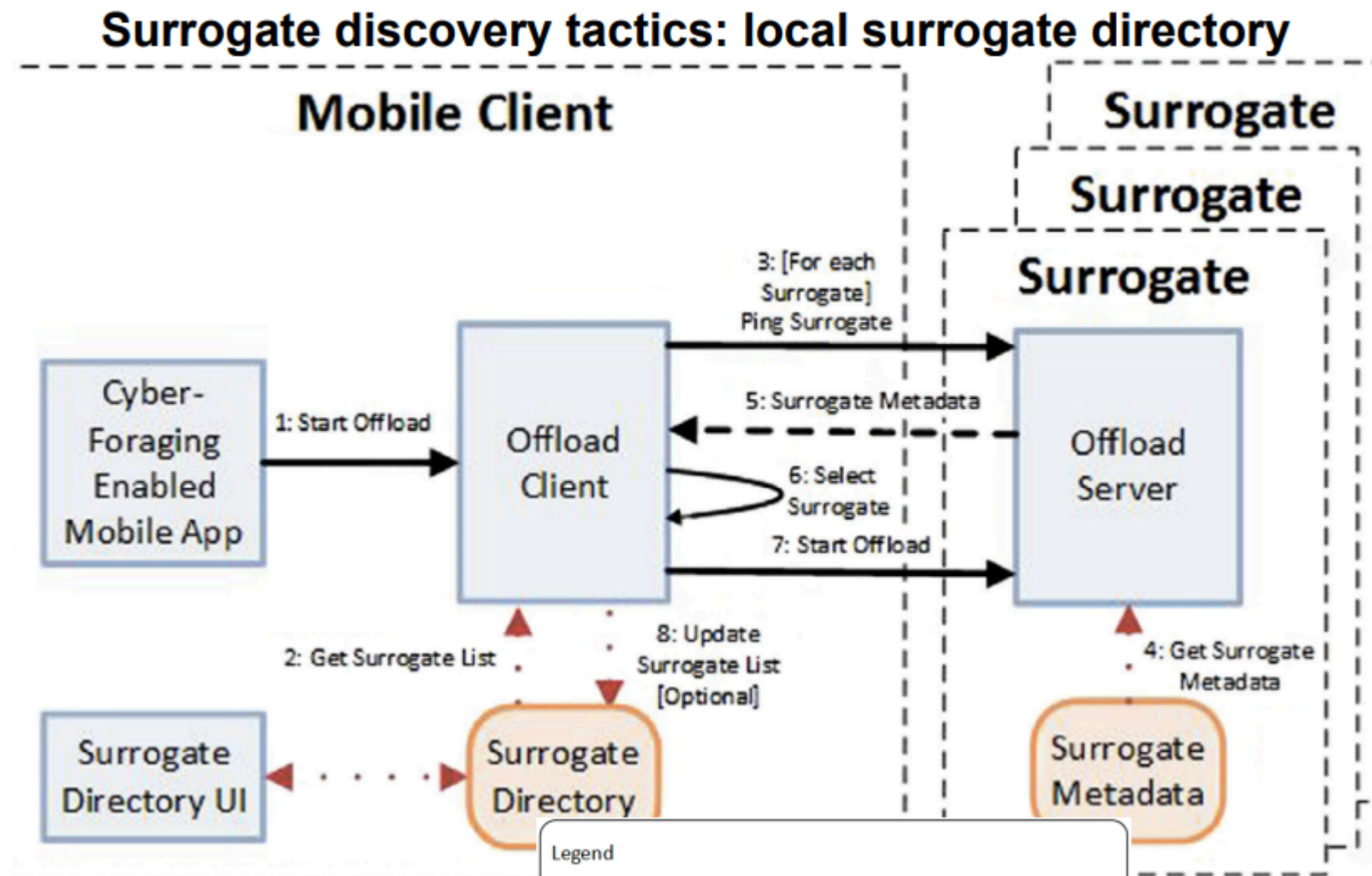


[Torna all'indice](#)

# Analisi

## Surrogate discovery

Nell'ambiente di setup, per semplicità, viene utilizzato il paradigma **local surrogate directory**. Ogni dispositivo conosce "staticamente" i device di "livello superiore", ossia i surrogati da contattare. Tra le challenge aperte, identificare quale sia la soluzione più efficiente da utilizzare.



[Torna all'indice](#)



# Analisi

## Aspetti non funzionali

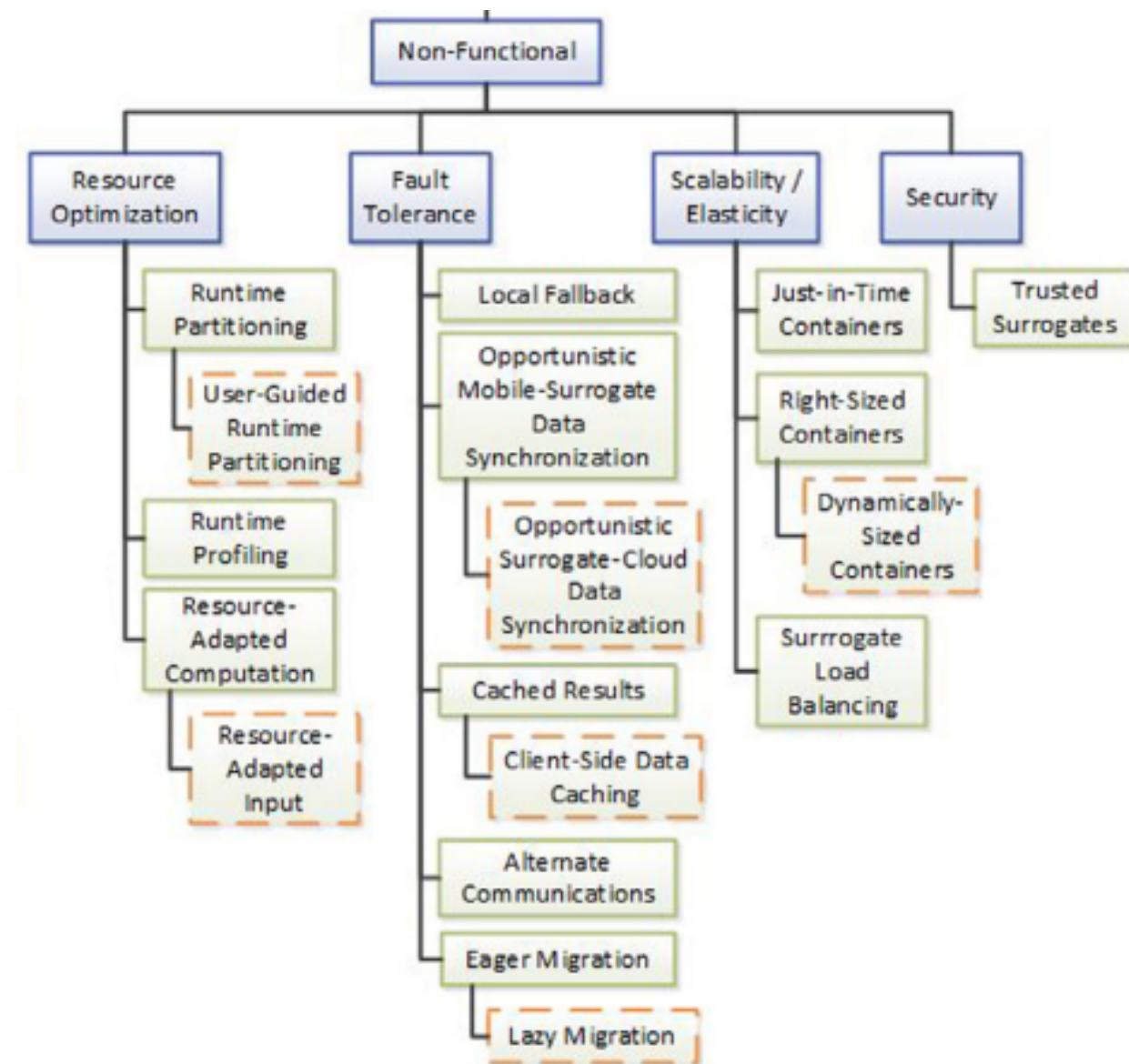
Riguardo **Fault tolerance** si utilizzano meccanismi di caching su ogni layer della struttura.

Invece, gli aspetti lasciati in sospeso sono

- **Resource optimization**
- **Scalability/Elasticity**
- **Security**

Questi infatti sono trattati nel paragrafo finale del paper come possibili sviluppi della ricerca.

Non è possibile individuare una tattica ad essi associata.



# Possibili miglioramenti

## Security: trusted surrogates

In un contesto realistico è impossibile assumere che tutti dispositivi siano trusted, è necessaria l'autenticazione.

- Realizzare **mutual authentication** tra i vari device
- Oppure, usare un **certification authority**

Inoltre, alcuni dati potrebbero esser sensibili, salvarli nelle cache potrebbe esporli a pericoli.

- Utilizzo di **tecniche di cifratura** sui dati nella cache

Questi miglioramenti aggiungerebbero complessità al sistema, quindi un calo di prestazione da quantificare.

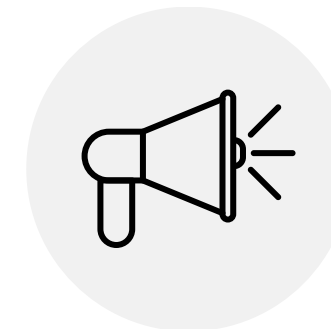
# Conclusioni



La soluzione proposta risulta esser efficace , al passo con il paradigma attuale di continuum computing.



Alcuni degli aspetti tralasciati andrebbero maggiormente approfonditi per validare ulteriormente i risultati ottenuti.



ICN e Edge computing si sono rivelate tecniche promettenti per ridurre latenza e migliorare prestazioni nelle reti 5G.

28

# Grazie!

---

Ci sono domande?