

POLITECNICO DI MILANO  
Scuola di Ingegneria Industriale e dell'Informazione  
Corso di Laurea Magistrale in Ingegneria Informatica  
Dipartimento di Elettronica, Informazione, Bioingegneria



# **ANALISI DEI LIMITI DELL'IMPLEMENTAZIONE DI WI-FI DIRECT IN ANDROID PER RETI OPPORTUNISTICHE**

**RELATORE:** Prof. Luciano BARESI

**Tesi di Laurea di:**  
**Stefano CAPPA 796552**

**Anno Accademico 2014-2015**

# Obiettivi della tesi

Generali e relativi al protocollo Wi-Fi Direct:

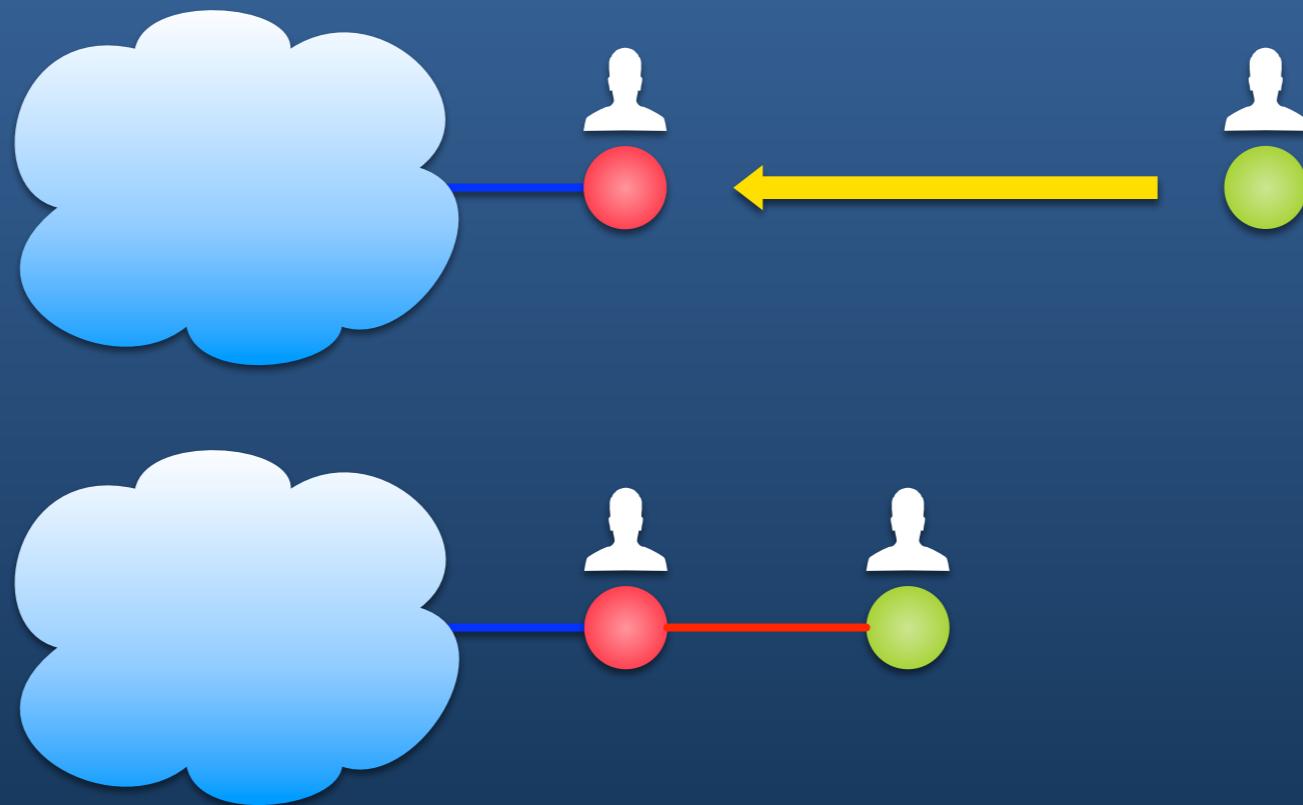
- Proporre tecniche per garantire SCALABILITÀ e DISPONIBILITÀ

Specifici per Android:

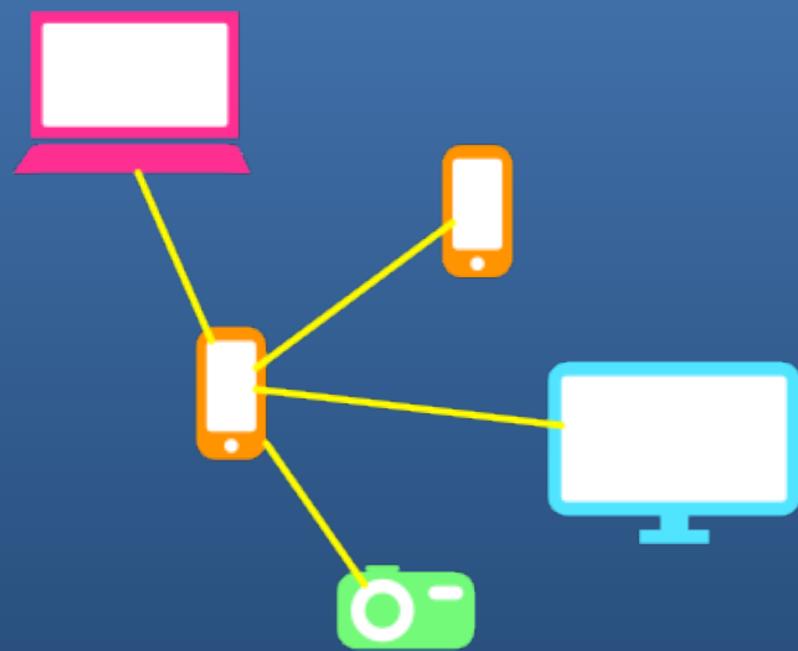
- Analizzare i limiti di Wi-Fi Direct in Android
- Verificare l'estensibilità del protocollo per supportare le RETI OPPORTUNISTICHE

# Reti Opportunistiche

- Non è necessaria una infrastruttura di rete preesistente
- Topologia di rete dinamica
- Quando i dispositivi si incontrano si connettono.

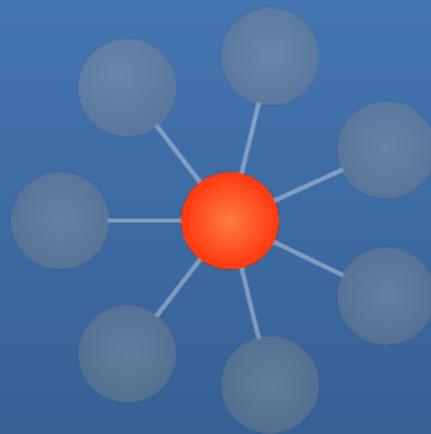


# Wi-Fi Direct (1)

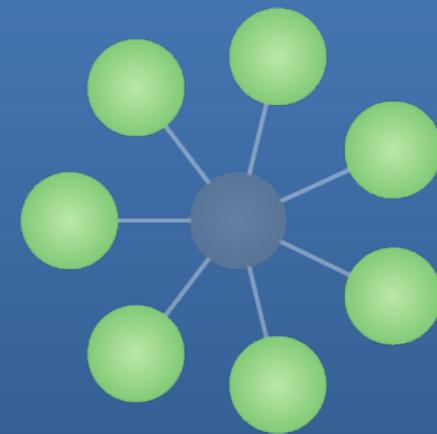


- Protocollo standard per la comunicazione P2P tra dispositivi
- Non necessita di una infrastruttura di rete preesistente.
- Implementato interamente via software

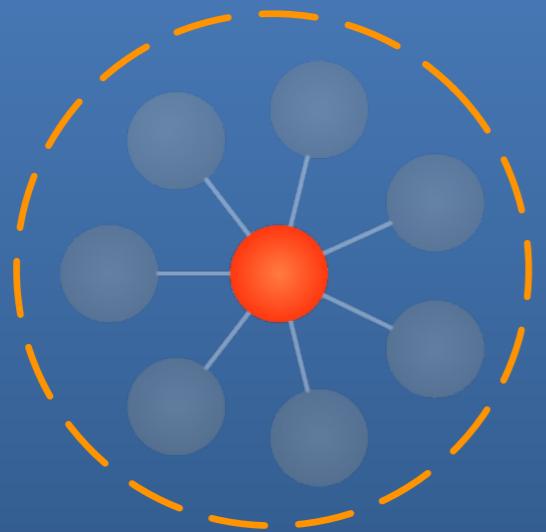
# Wi-Fi Direct (2)



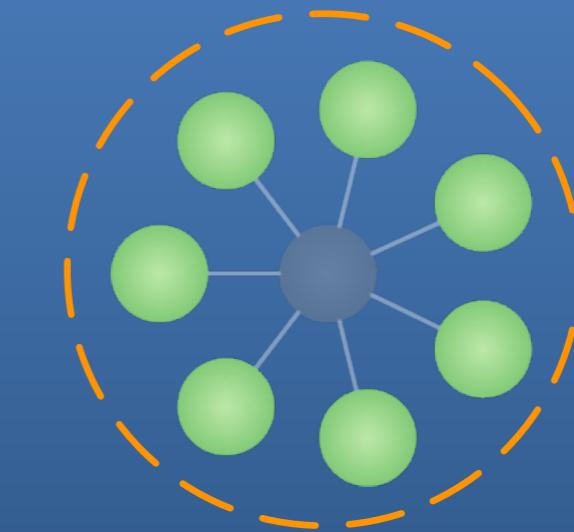
- Group Owner
- Client



# Wi-Fi Direct (2)

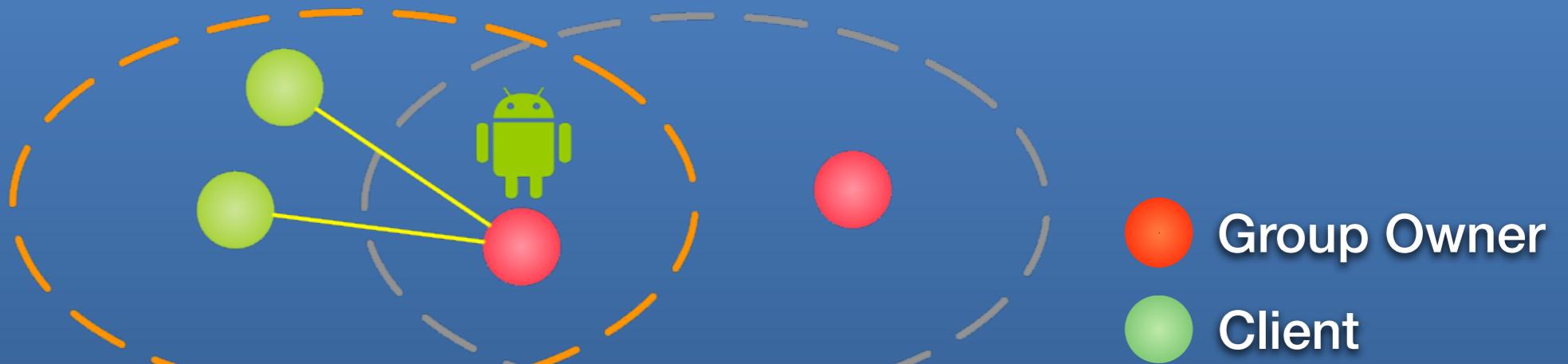


- Gruppo
- Group Owner
- Client



Le specifiche NON vietano ad un dispositivo di partecipare a più gruppi contemporaneamente.

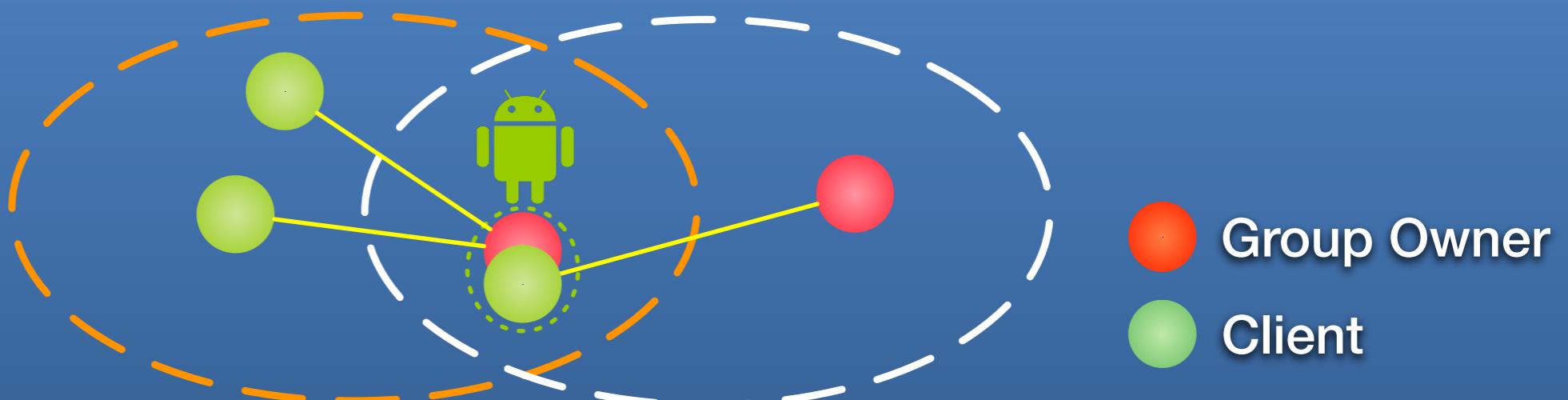
# Wi-Fi Direct in Android



In Android:

- un dispositivo NON può far parte di due o più gruppi.
- la fase di Discovery (ricerca dei dispositivi) deve essere riavviata prima della connessione.

# Estensione di Wi-Fi Direct in Android



Requisito: far parte di più gruppi contemporaneamente

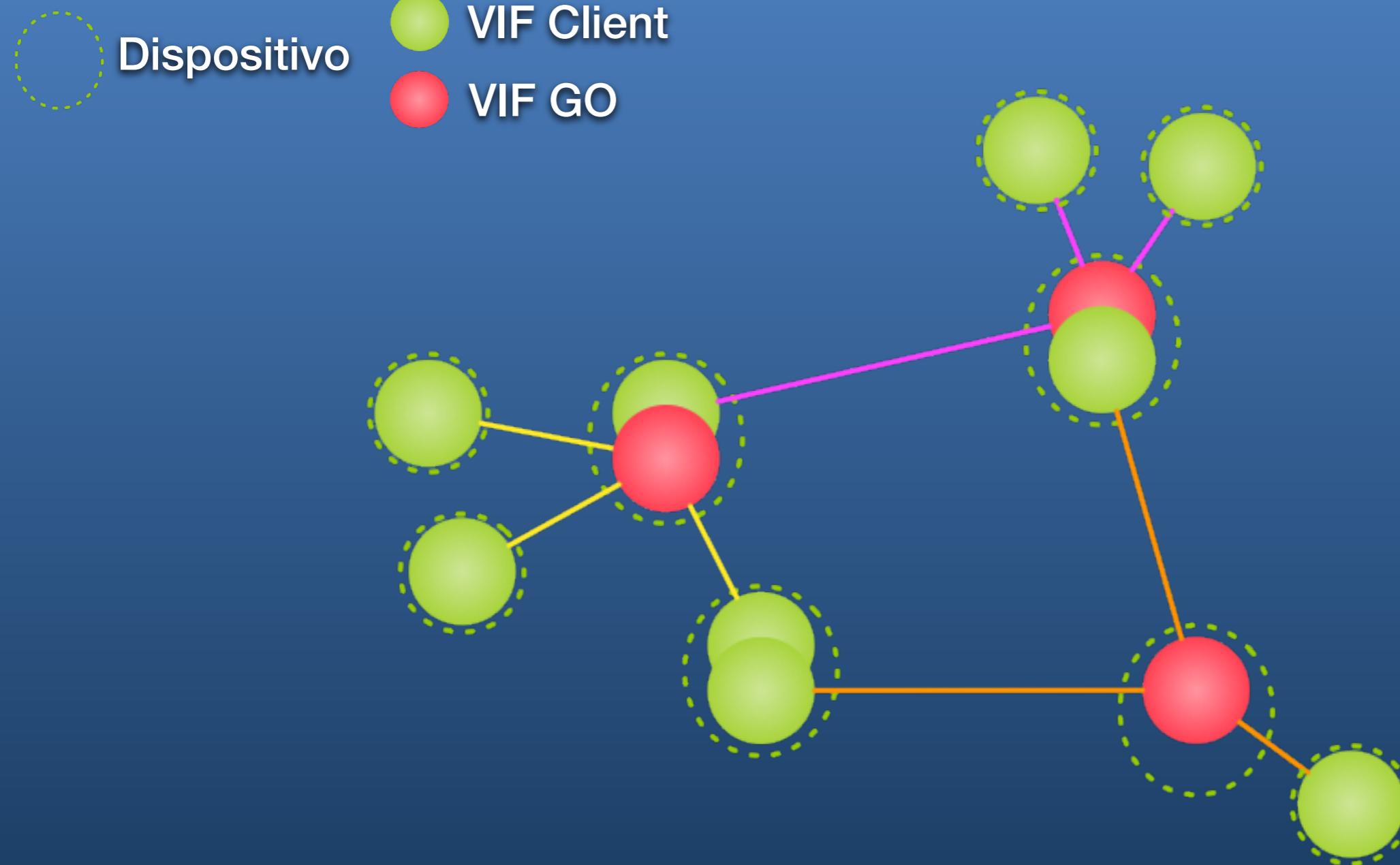


Sfruttare il concetto di interfacce di rete virtuali (VIF)



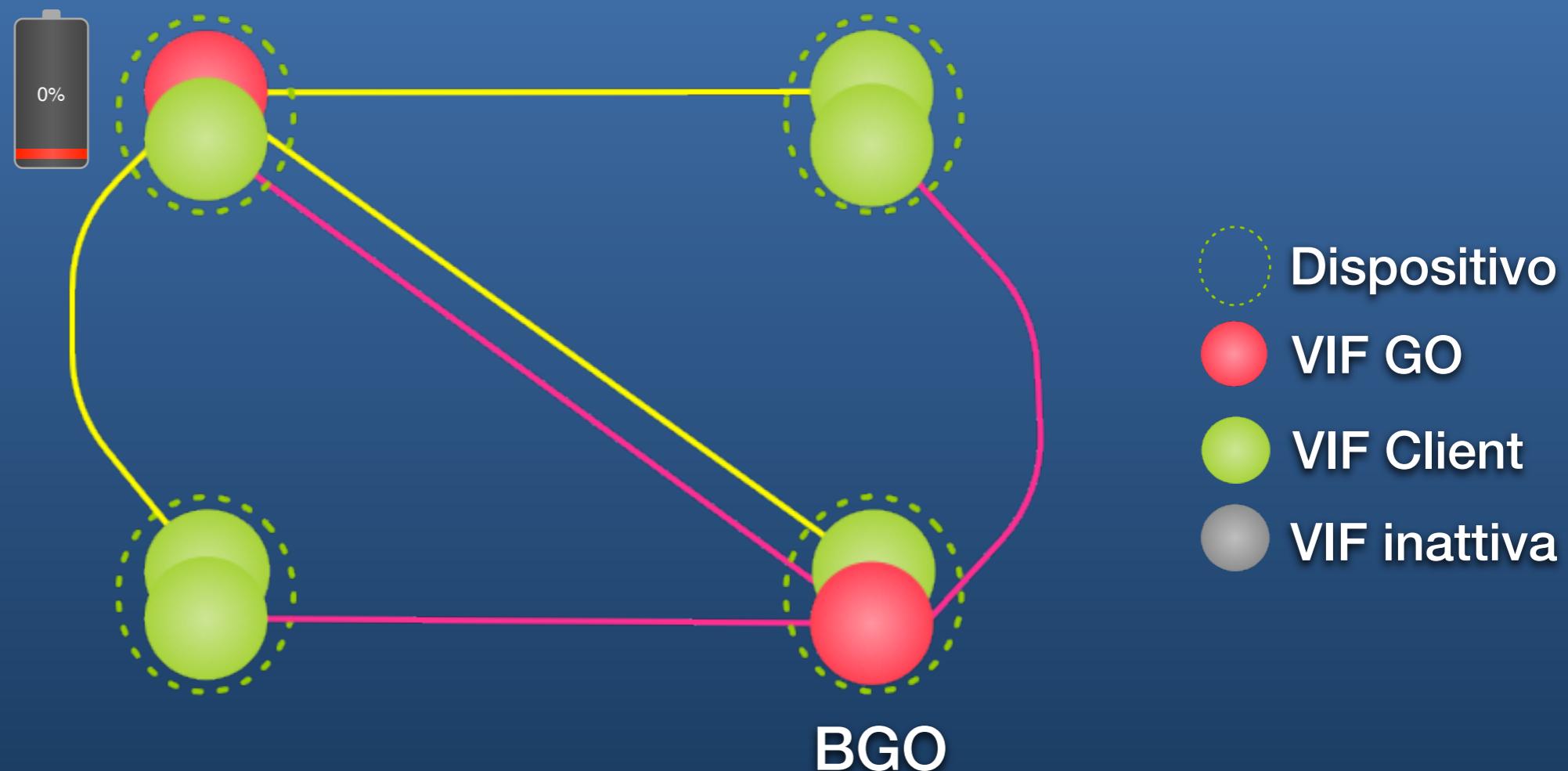
Astrazioni software di  
schede di rete fisiche

# Scalabilità (1)



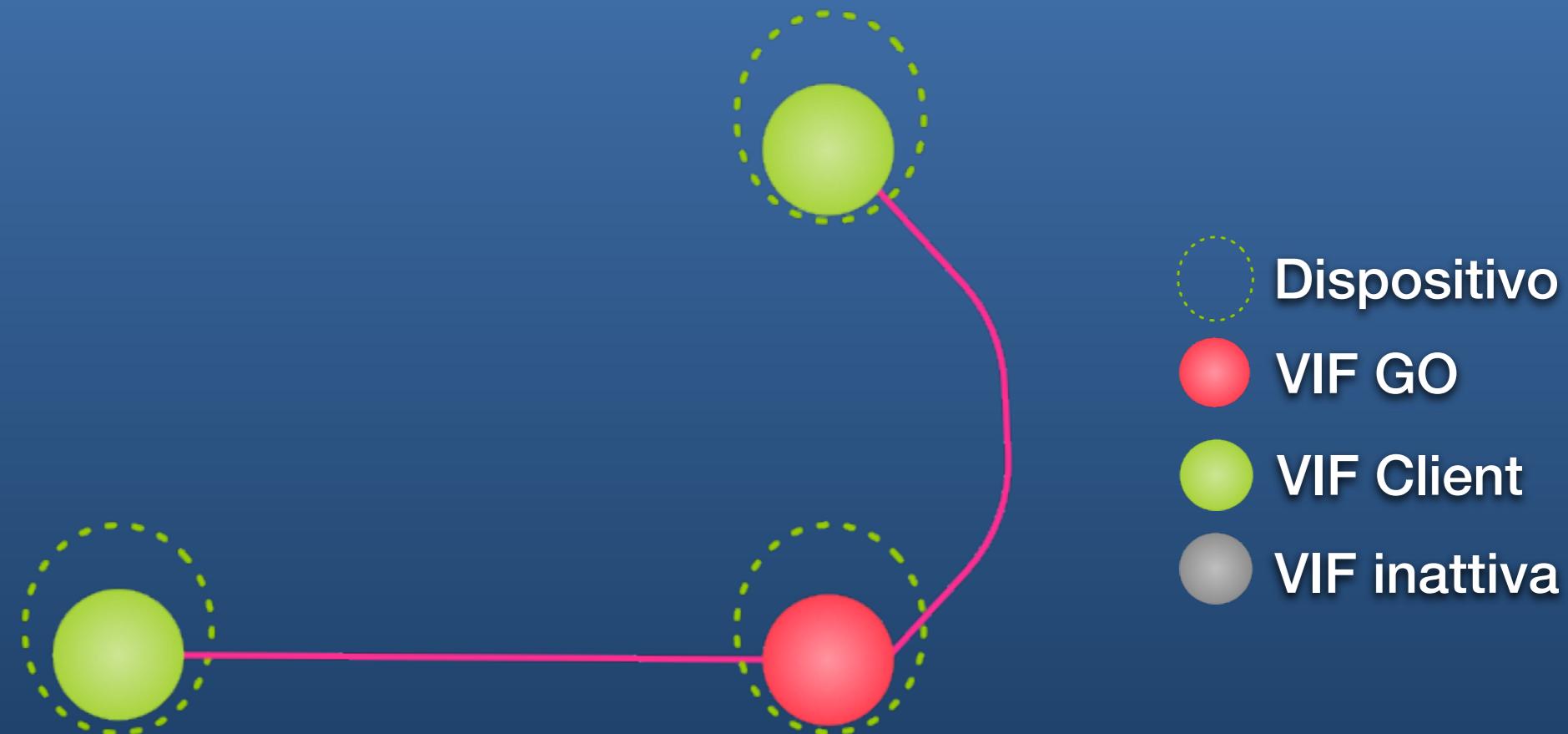
# Disponibilità (1)

Uso un Group Owner di Backup, detto BGO



# Disponibilità (2)

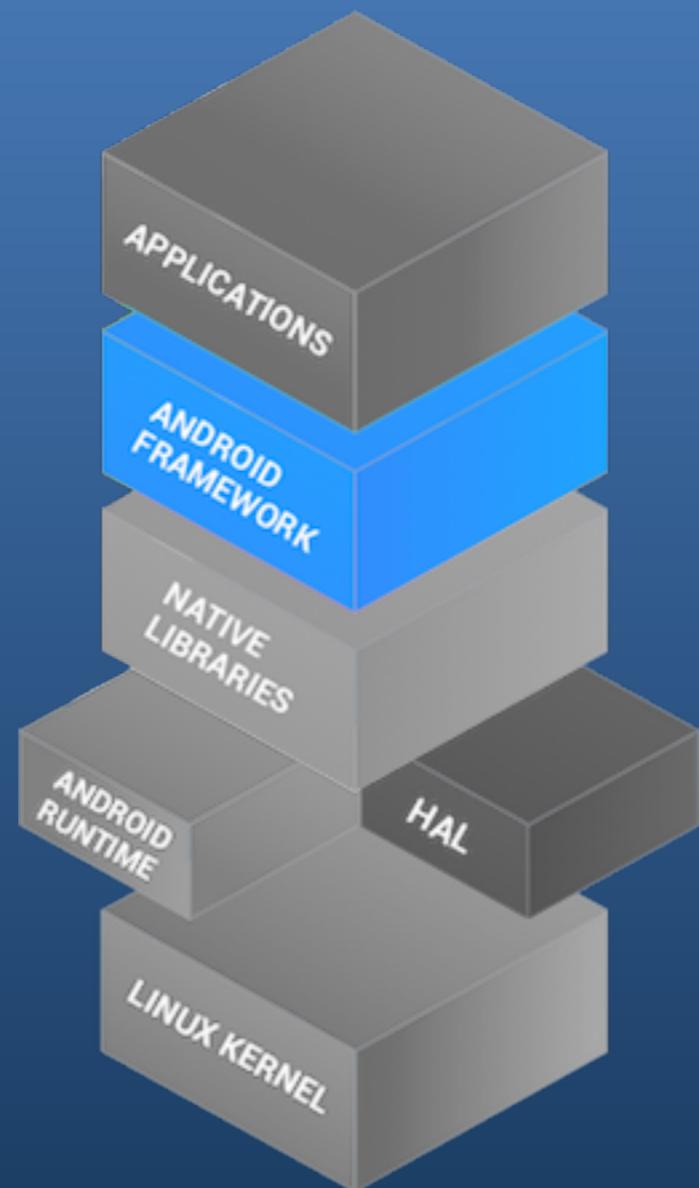
La rete secondaria del BGO non è soggetta a ritardi per la formazione, perché già creata

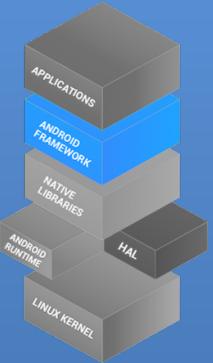


# Architettura di Android



# Architettura di Android





# Android Framework

Wi-Fi Direct è implementato in Java con forti limiti

Le interfacce di rete non sono create in Java o in C++

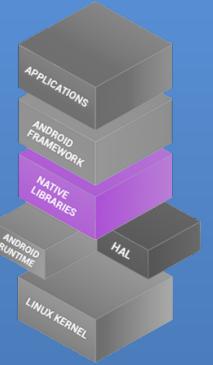


Modificare il Framework Android non risolve la situazione



Le funzionalità Wi-Fi sono gestite da `wpa_supplicant`

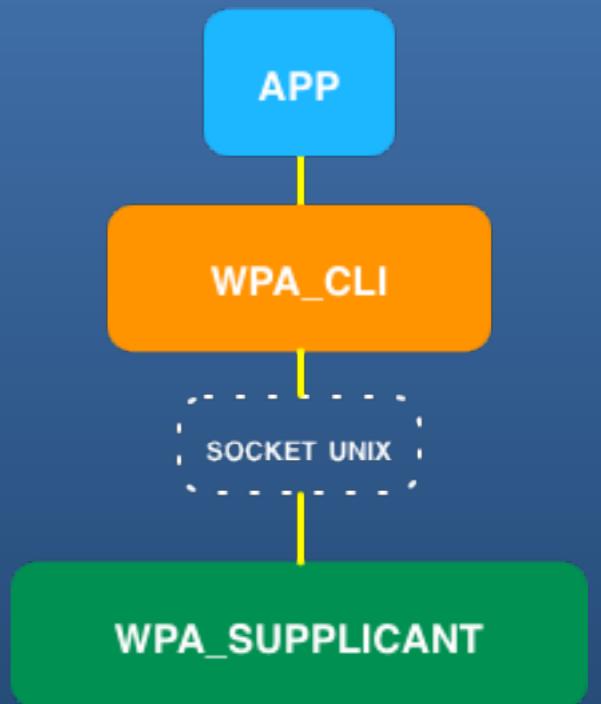
# Native libraries (1)

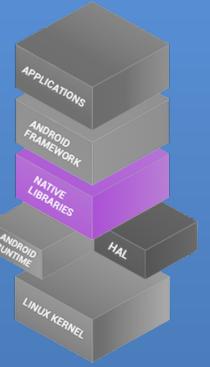


Il modo più semplice per interagirci  
è usare `wpa_cli`

`Wpa_cli`:

- non incluso in Android
- espone le funzionalità Wi-Fi ad alto livello, superando il Framework Android

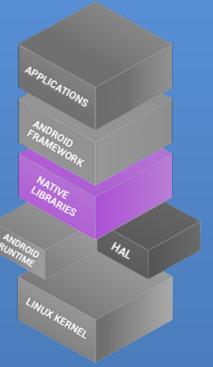




# Native libraries (2)



# Native libraries (3)





# Linux Kernel



-12 (out of memory)

+



virtual face add failed (-2)

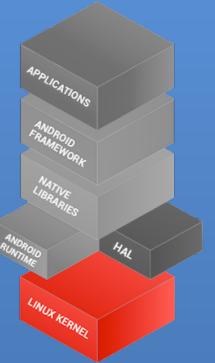


Modifica dei driver Wi-Fi

+

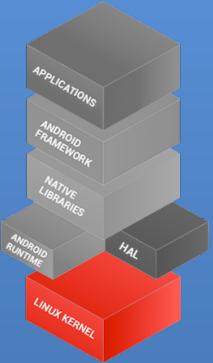
Installazione sul dispositivo

# Driver wireless (1)



wpa\_supplicant aggiunge interfacce  
sempre con lo stesso MAC address

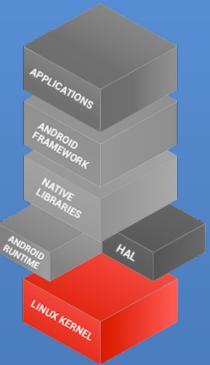
# Driver wireless (1)



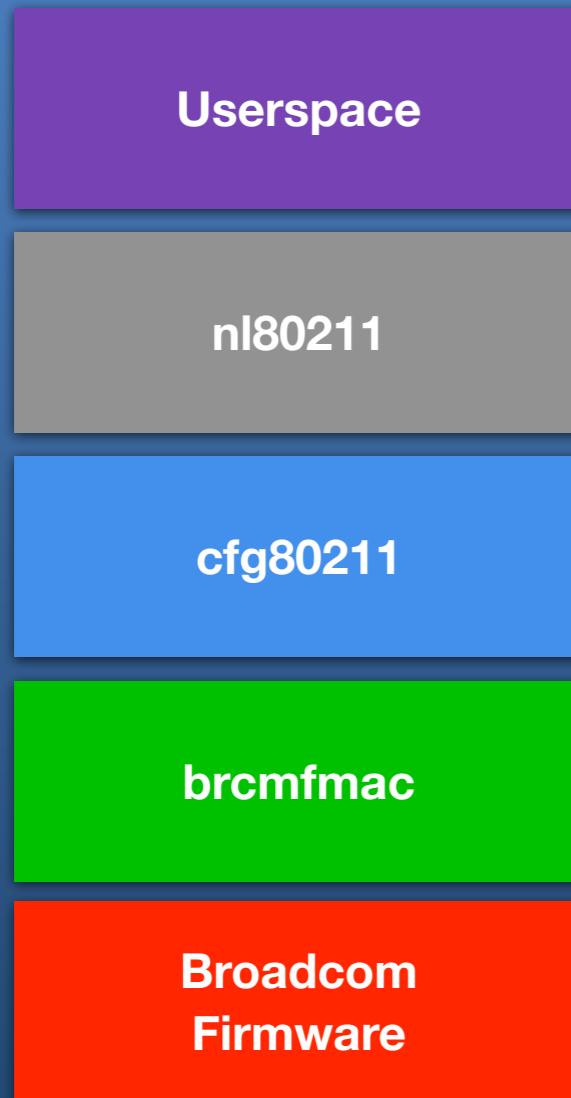
wpa\_supplicant aggiunge interfacce  
sempre con lo stesso MAC address



Il MAC Address è un identificativo univoco,  
quindi dovrebbe cambiare



# Driver wireless (2)



Il chip Wi-Fi degli Smartphone a disposizione è di tipo Full MAC



Le funzioni di rete, sono implementate nel **firmware proprietario** di Broadcom.



**CODICE SORGENTE NON  
MODIFICABILE**

# Applications (1)



## Pigeon Messenger



- Gestisce chat testuali
- Accoda i messaggi quando la connessione non è disponibile
- Gestisce le riconnessioni automatiche



Convince l'utente di poter partecipare  
a più gruppi contemporaneamente

Demo  
per motivi di spazio non posso  
integrarlo nella presentazione,  
quindi l'ho caricato su YouTube

[https://www.youtube.com/watch?  
v=qcfsWBDRhto](https://www.youtube.com/watch?v=qcfsWBDRhto)

# Applications (2)



## Pigeon Messenger



- Open Source \*
- Scalabile al crescere dei dispositivi (testata con 6 dispositivi)
- Rende trasparente all'utente il concetto di gruppo ed i suoi limiti

\*<https://github.com/deib-polimi/PigeonMessenger>

# Sviluppi futuri

- Verificare se i dispositivi più recenti (con Kernel Linux 3.10) supportano interfacce di rete Wi-Fi Direct multiple
- Trovare nuovi scenari di utilizzo per creare app basate su Wi-Fi Direct
- Aggiungere funzionalità a Pigeon Messenger

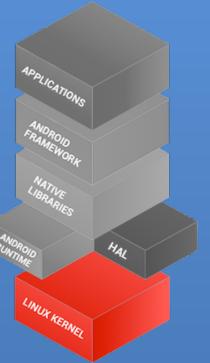
# Conclusioni

Ho mostrato:

- la reale causa dei limiti di Wi-Fi Direct in Android
  - driver proprietari limitati e **NON MODIFICABILI**
- i limiti e come scavalcare il Framework Android
  - comunicando con `wpa_supplicant` tramite `wpa_cli`
- che esistono scenari di utilizzo in cui è possibile creare soluzioni funzionanti
  - App di messaggistica chiamata Pigeon Messenger

# Grazie per l'attenzione

# Grazie per l'attenzione

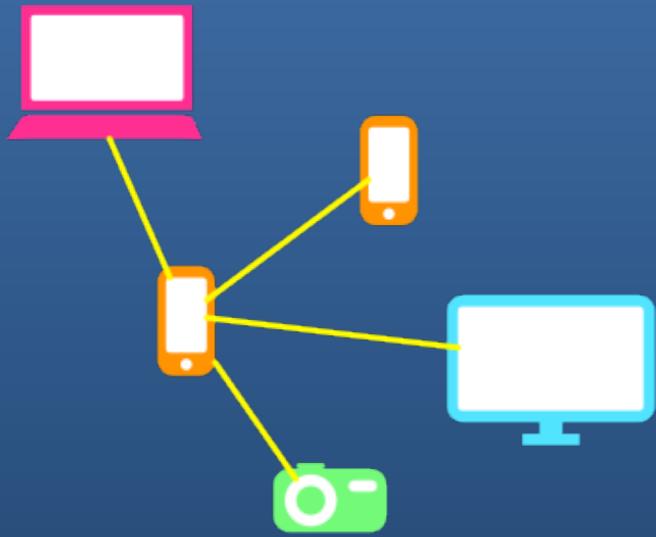


# Driver wireless

## Dispositivi Soft MAC



# Wi-Fi Direct



Evoluzione delle reti ad-hoc per avere:

- maggiore velocità di trasferimento dati
- minori consumi energetici usando protocolli creati apposta per Wi-Fi Direct
- retrocompatibilità con Wi-Fi classico (Client Legacy vedono i GO come AP)
- implementato interamente via software

# Wi-Fi Direct

Diverse fasi sequenziali:

1. Discovery → Ricerca dispositivi
2. Negoziazione GO → Scelta del GO
3. WPS → Autenticazione
4. DHCP → Assegnamento IP



# Wi-Fi Direct

**Device Address:**

- indirizzo univoco usato per identificare il dispositivo;
- dovrebbe coincidere col MAC Address.

**P2P Interface Address (IA):**

- non deve essere globalmente univoco, ma è sufficiente che lo sia localmente;
- può coincidere con il Device Address;
- deve essere assegnato alla creazione del gruppo e durare per tutta la vita del gruppo;
- se si usano più interfacce si devono usare IA diversi.

# Wi-Fi Direct

## FASE 1 - P2P Discovery

Costituita da:

- **Device Discovery:** per trovare dispositivi, diviso in 2 fasi:
  - SCAN: per trovare il miglior Operating Channel
  - FIND: iniziano 2 fasi alternate SEARCH e LISTEN, sui Social Channel (1,6,11) per periodi di tempo casuali perché arrivino sullo stesso canale di comunicazione. La casualità garantisce l'arrivo sul canale comune.
- Group Formation
- P2P Invitation (usato per gruppi Persistent)
- Service Discovery (opzionale)

# Wi-Fi Direct

## FASE 2 - GO Negotiation

Three Way Frame Exchange per mettersi d'accordo su chi sarà il GO. Si scambia il go\_intent.

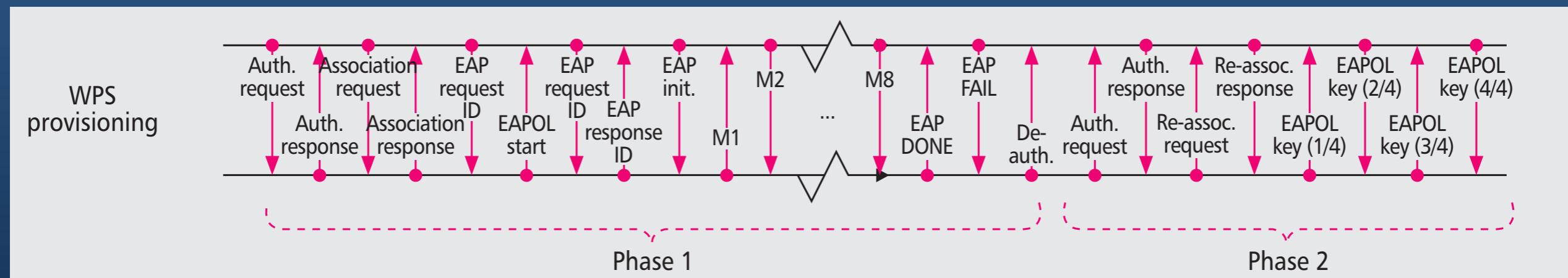
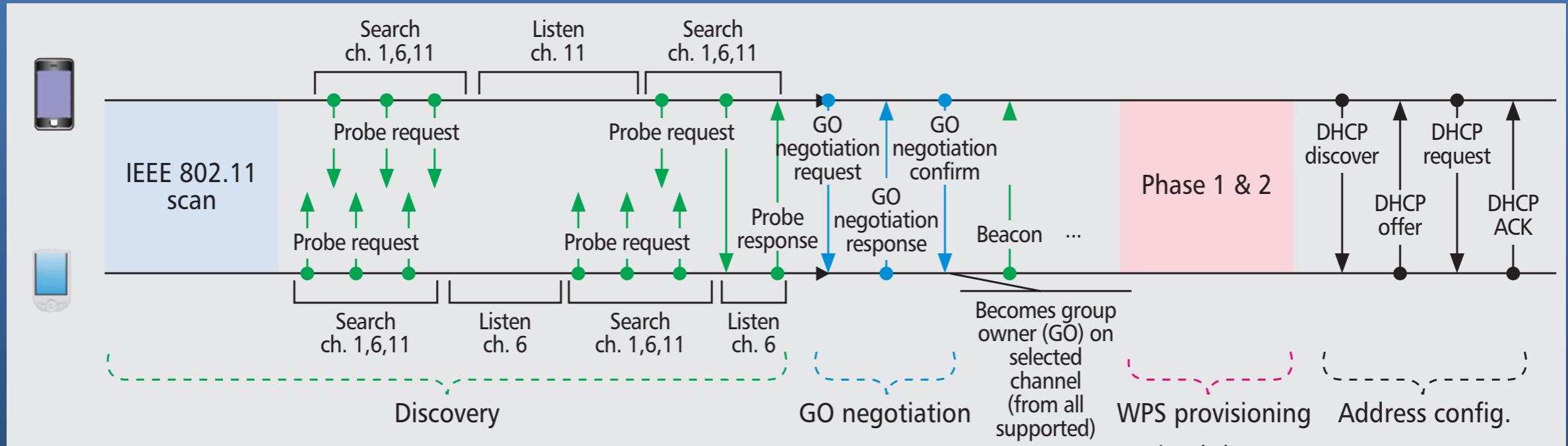
Chi ha questo valore più alto diventa il GO.

In caso di parità si usa “Tie Breaker bit” scelto casualmente con stessa probabilità tra 0 e 1.

Chi manda il bit a 1, in caso di parità, diventa il GO.

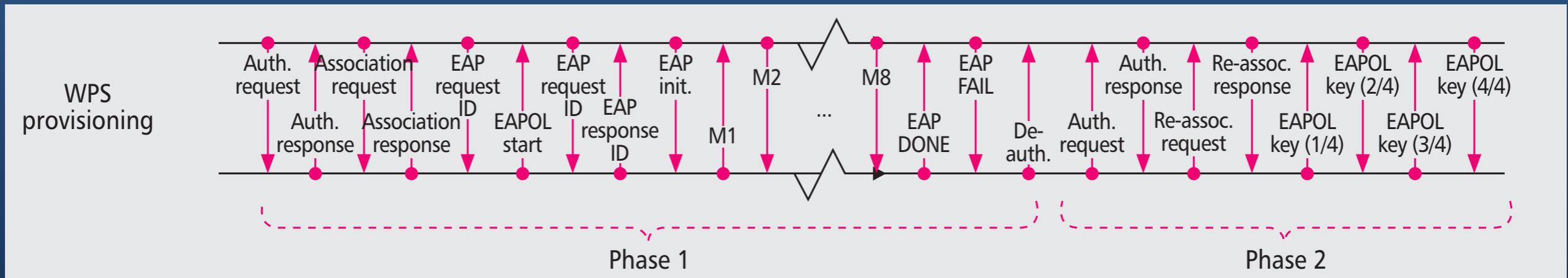
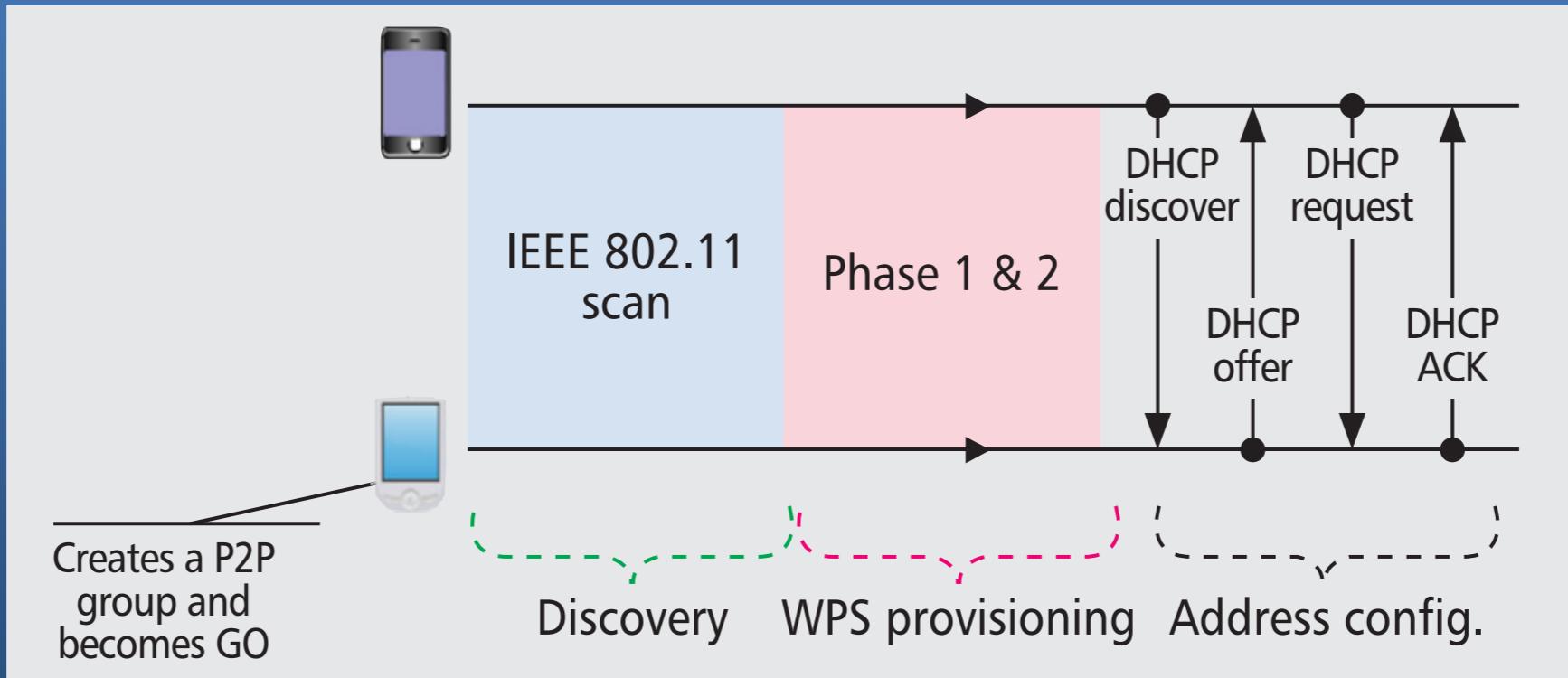
# Wi-Fi Direct

## Tipi di gruppo: STANDARD



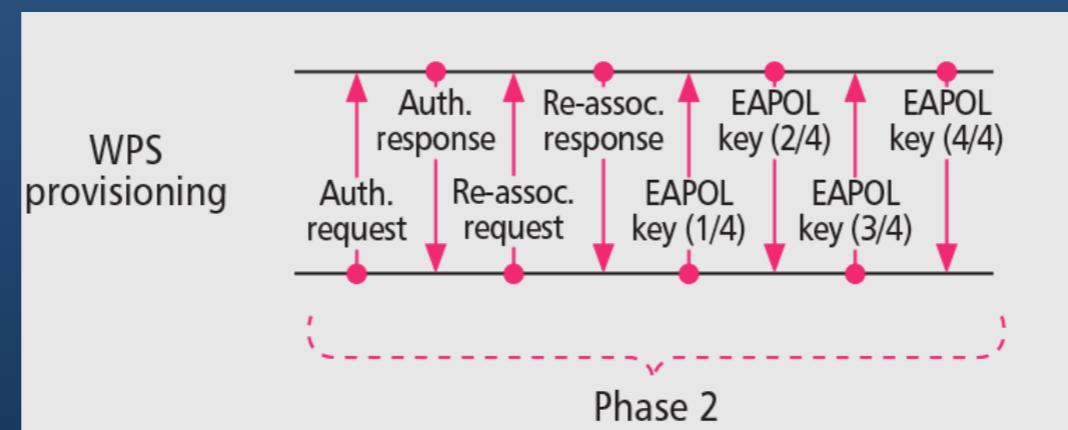
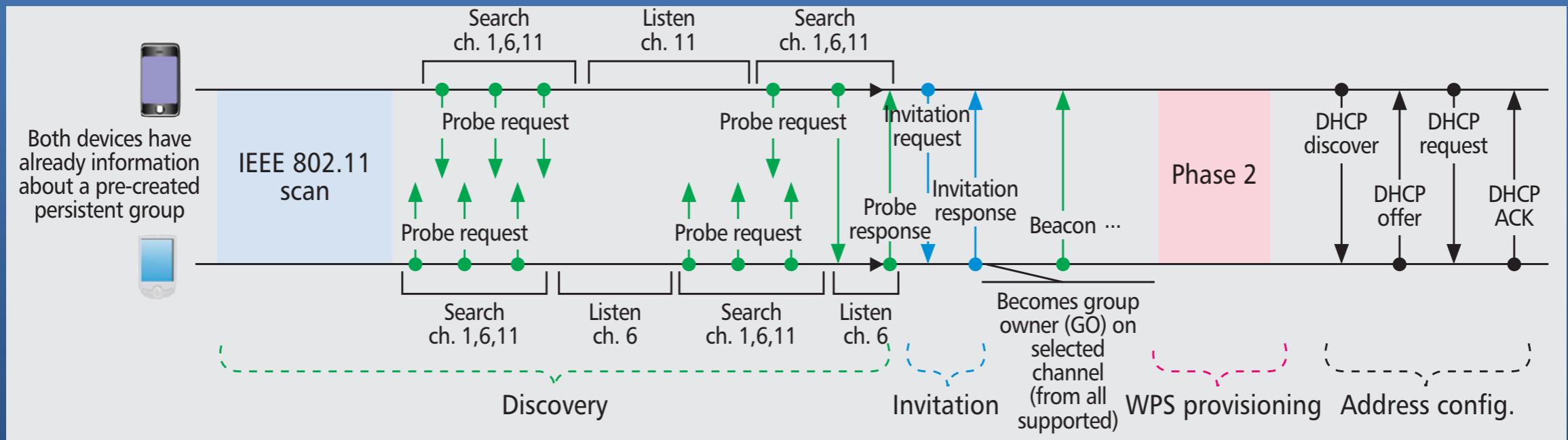
# Wi-Fi Direct

## Tipi di gruppo: AUTONOMOUS



# Wi-Fi Direct

## Tipi di gruppo: PERSISTENT



# Wi-Fi Direct

## FASE 3 - WPS Provisioning

Permette di creare connessione sicura con pulsante o PIN.  
GO implementa una interval REGISTRAR, il Client l'ENROLLEE.

Fase1: Interval registrar genera le credenziali. Usa WPA-2  
(AES-CCMP come cypher e PSK per autenticazione)

Fase2: Enrollee si scollega e si ri-associa usando le nuove  
credenziali di autenticazione

Se le credenziali sono già salvate, la fase 1 non serve più

# Wi-Fi Direct

## Power Saving:

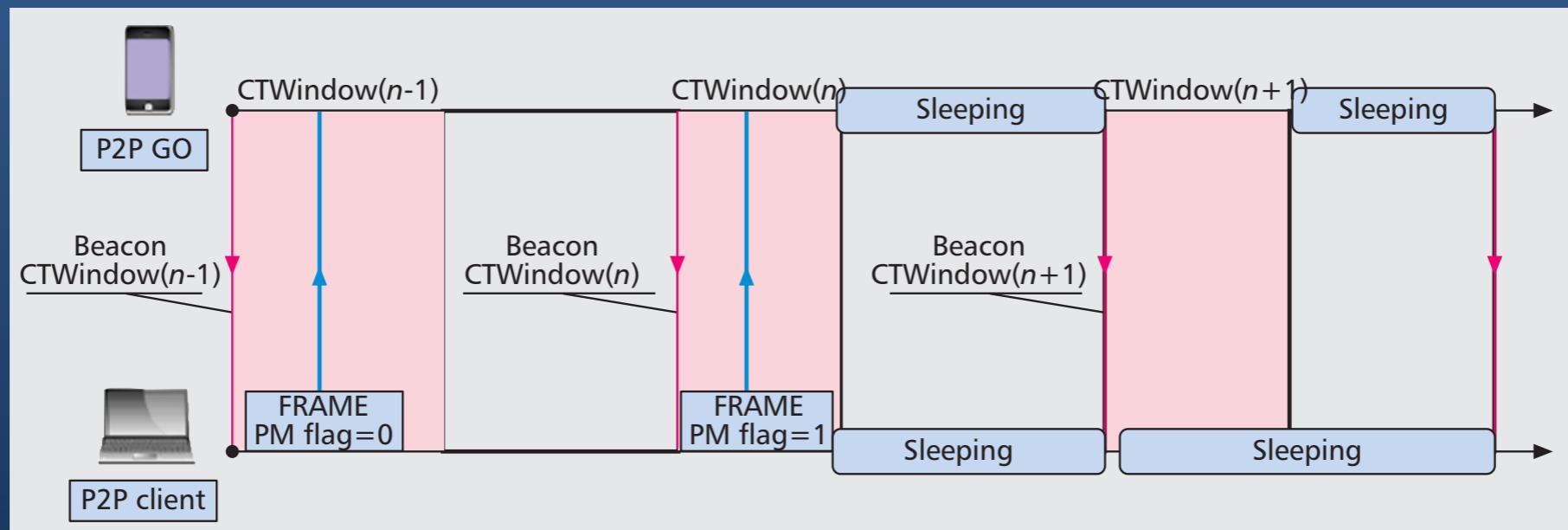
- Opportunistic Power Save (OPS)
- Notice of Absence (NoA)

# Wi-Fi Direct

## Opportunistic Power Save (OPS):

Il GO usa una finestra temporale (CTWindow) che specifica il tempo in cui sarà sveglio, quindi i Client possono inviare i Frame. Il GO non ha la decisione sull'ingresso in “sleep”, perché dipende dall'attività dei Client.

Per dare più controllo al GO si usa anche il NoA

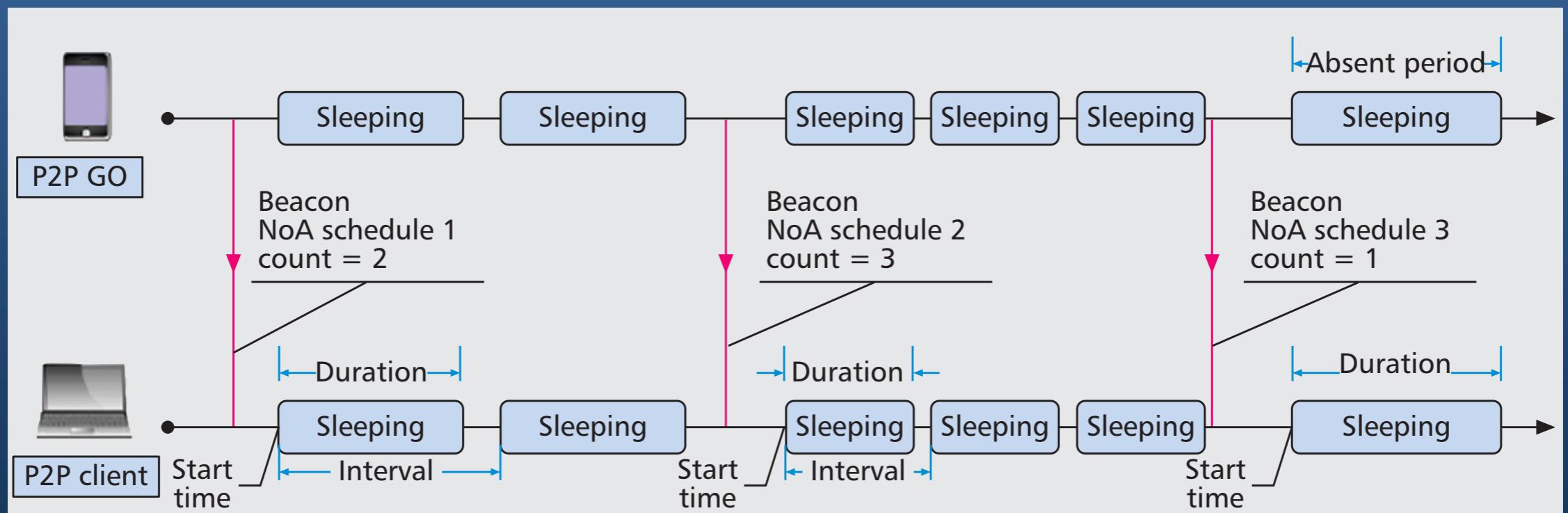


# Wi-Fi Direct

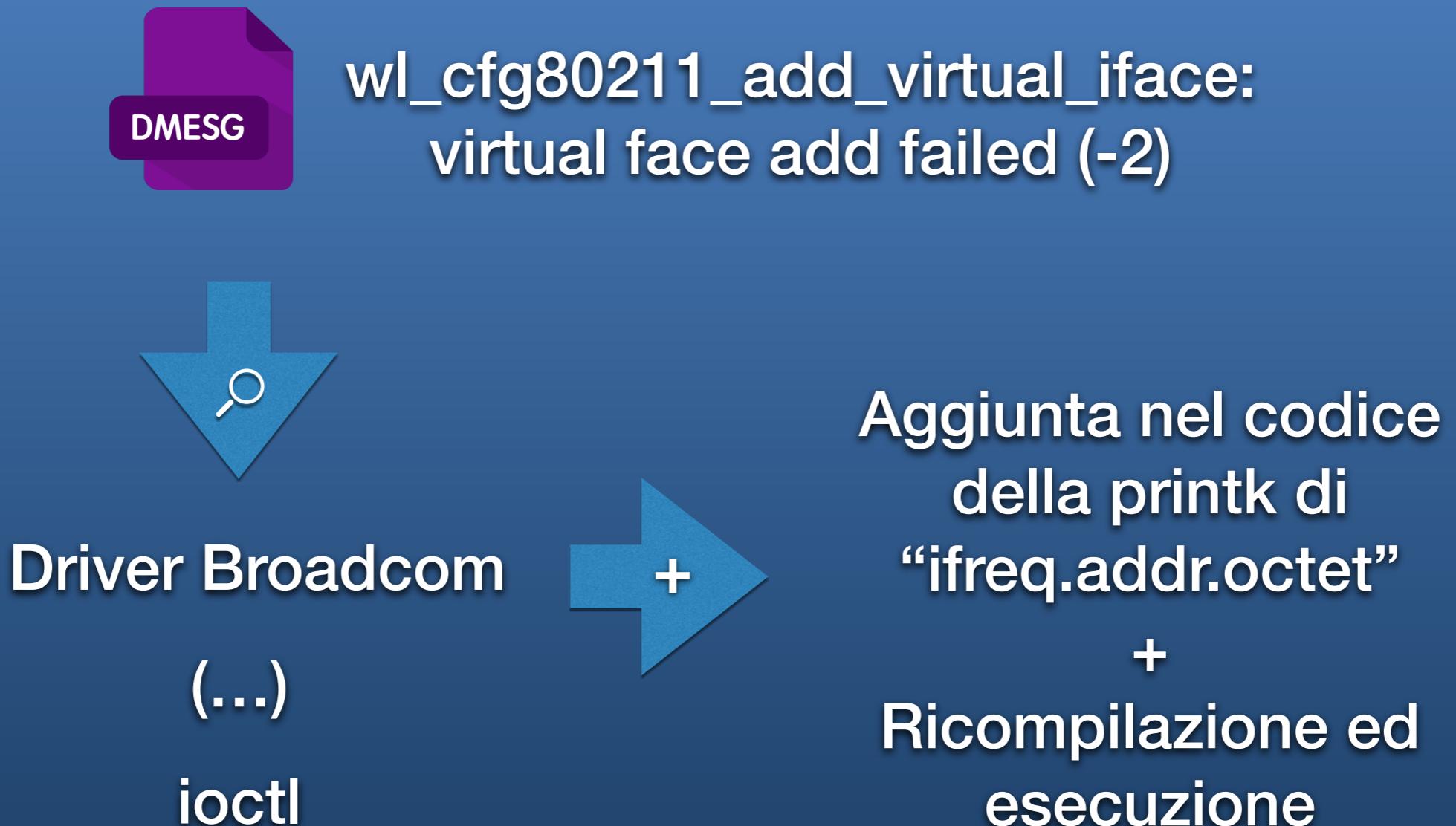
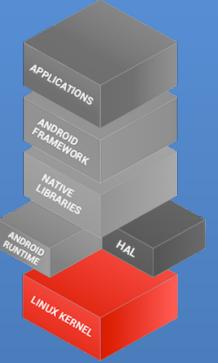
## Notice of Absence (NoA):

Il GO annuncia ai Client i periodi di tempo in cui sarà assente e i Client non potranno accedere al canale, sia se attivi, sia se in power save. Quindi, i Client possono spegnere la “radio”.

Il GO definisce: la durata del periodo, intervallo (tempo tra periodi consecutivi), start time, count (num. periodi)



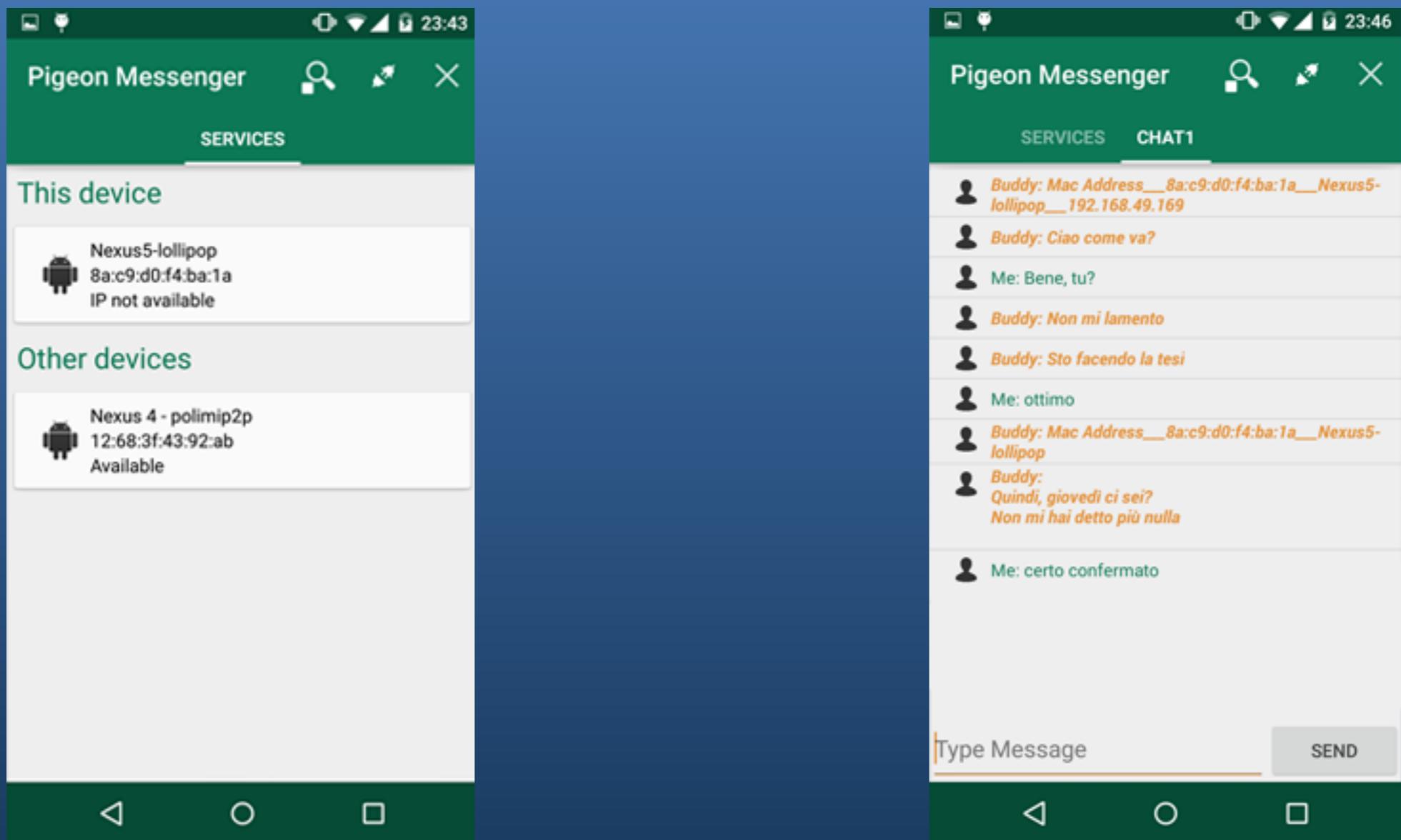
# Linux Kernel





# Applications

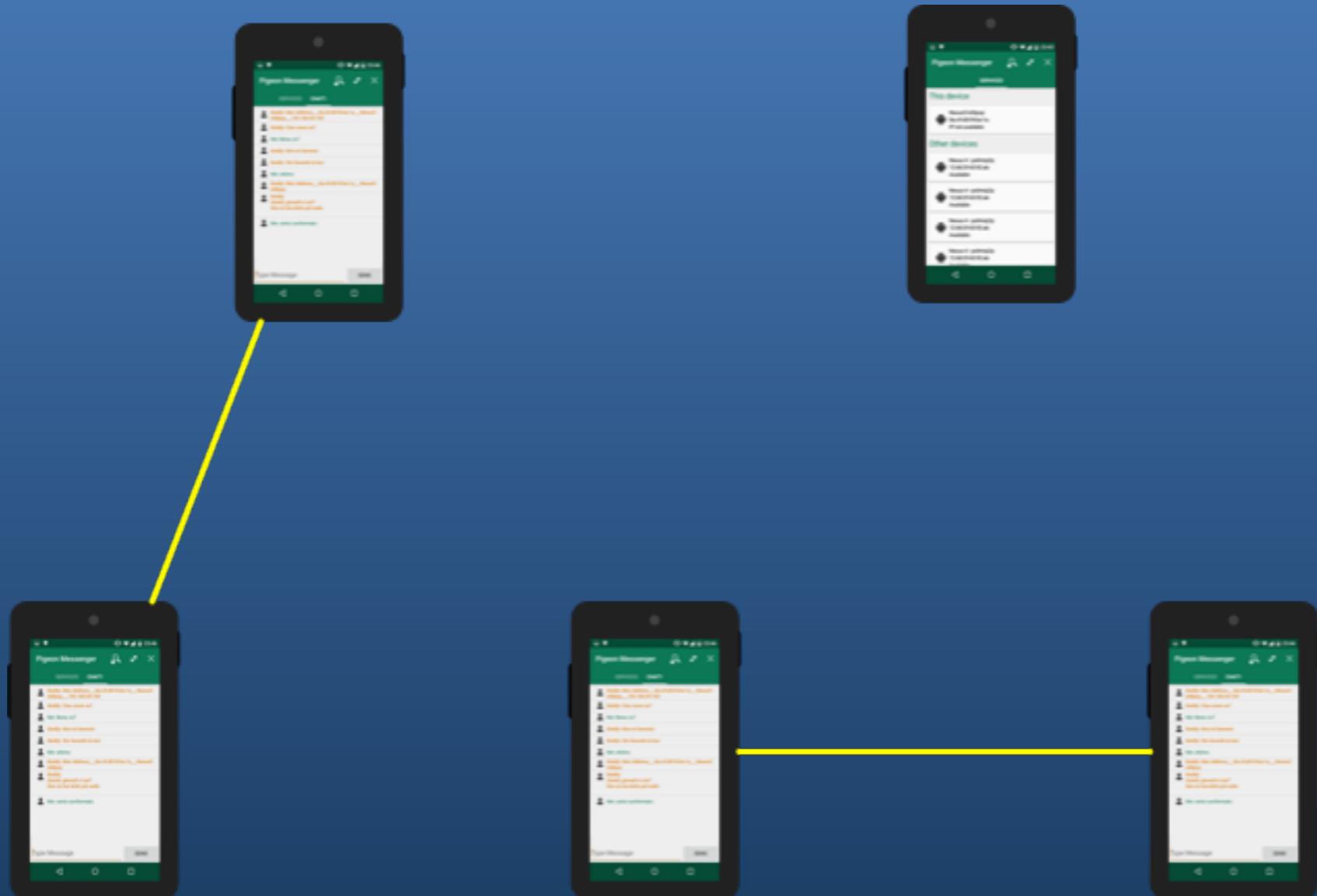
## Pigeon Messenger



# Applications



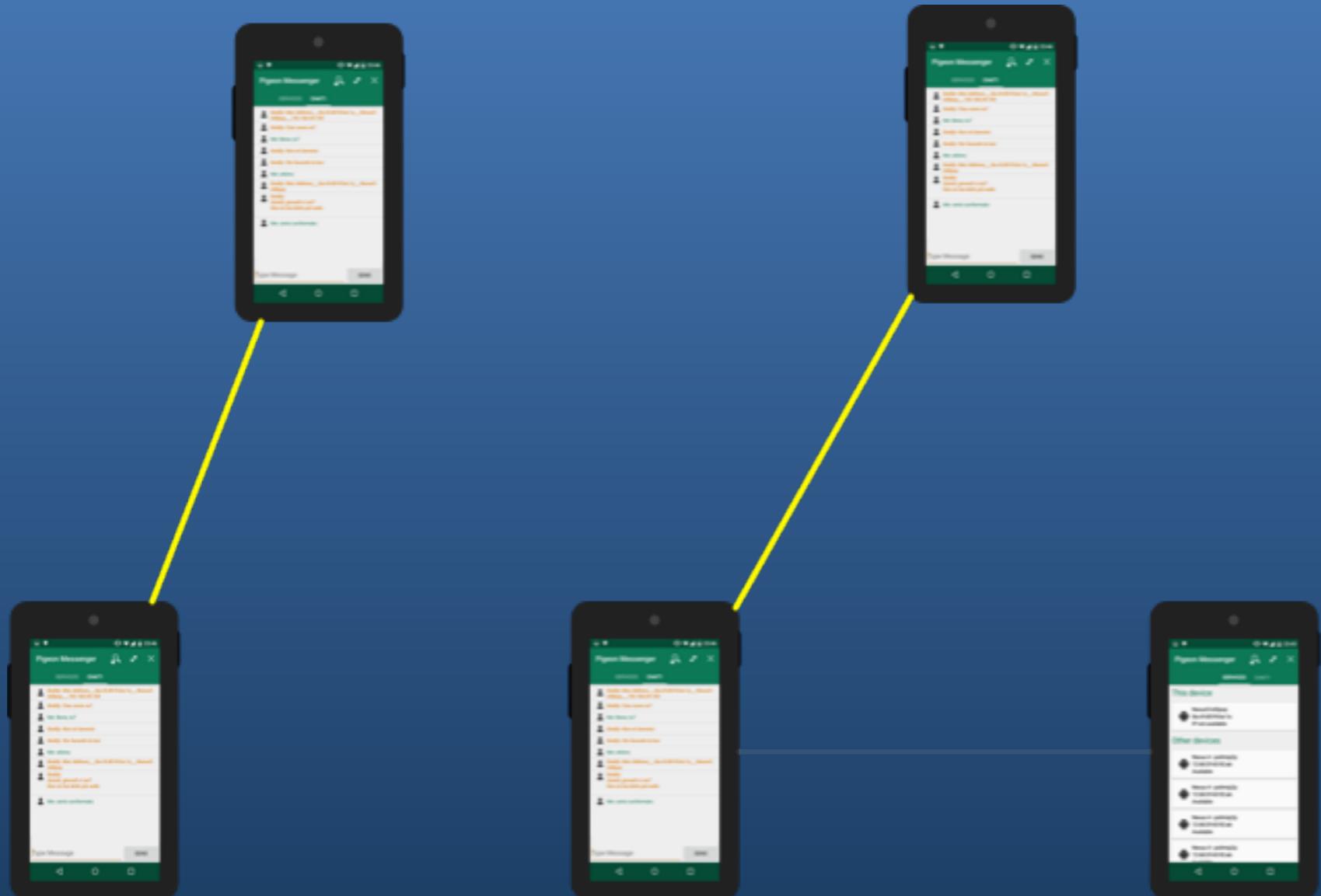
## Pigeon Messenger



# Applications



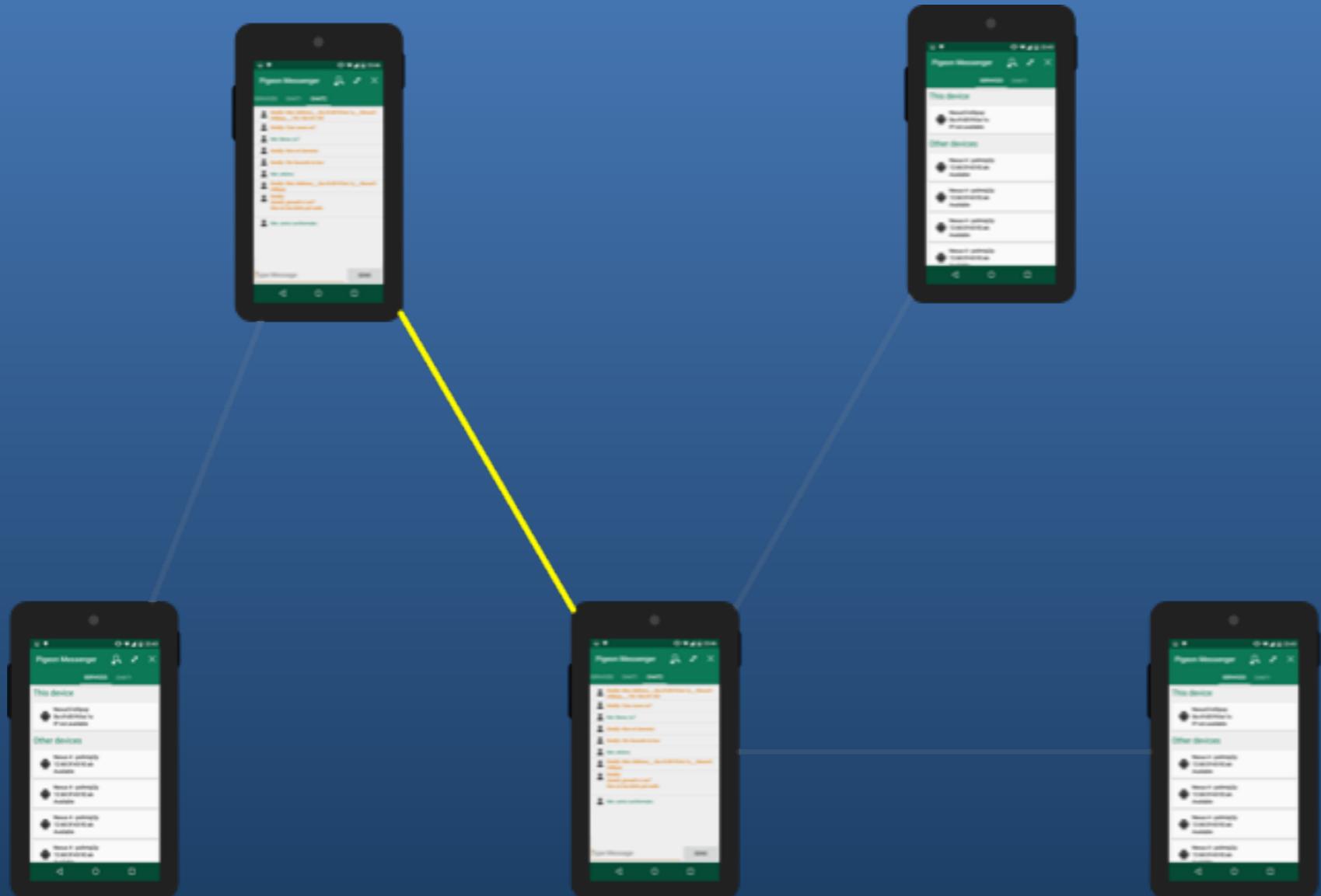
## Pigeon Messenger



# Applications



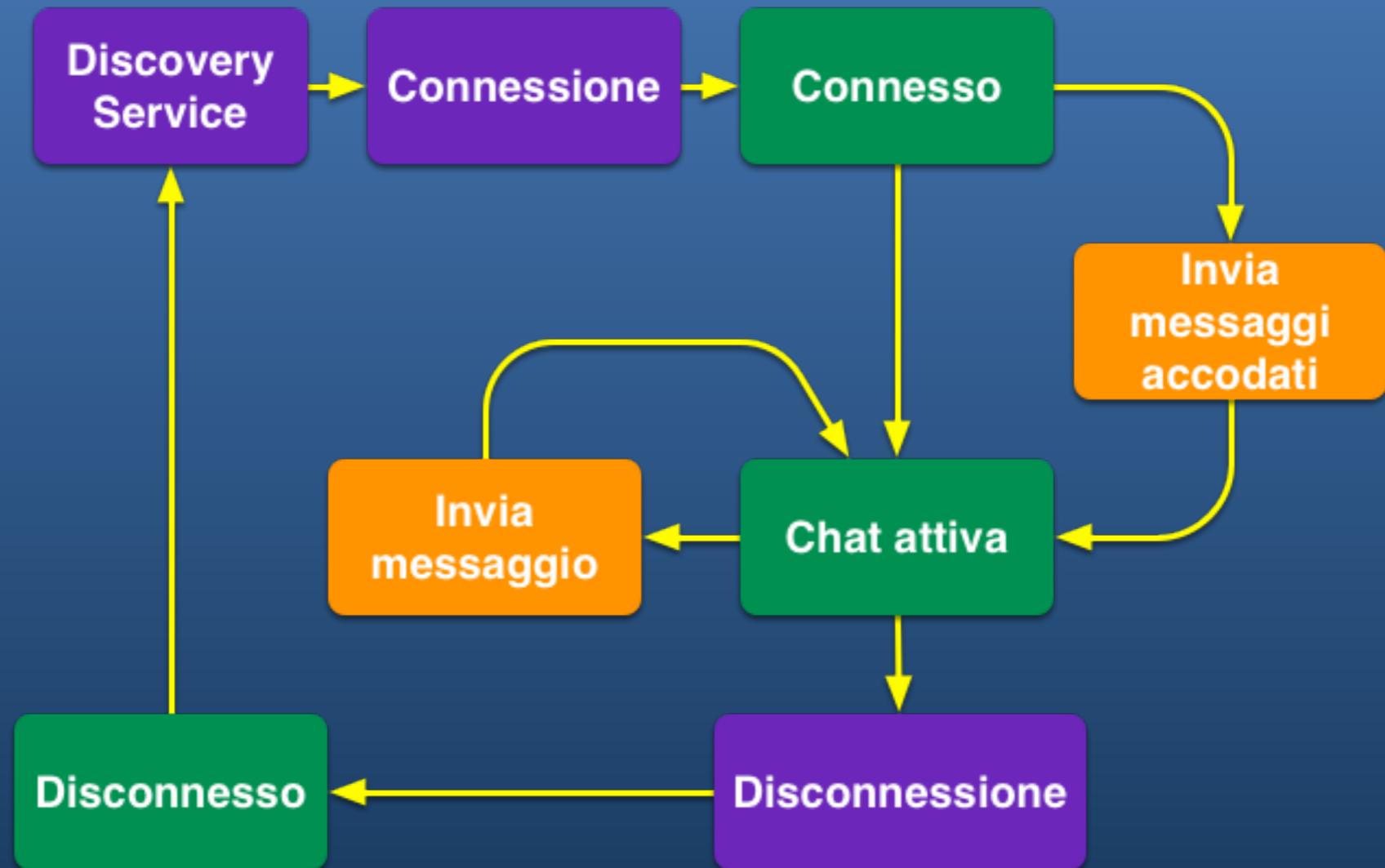
## Pigeon Messenger





# Applications

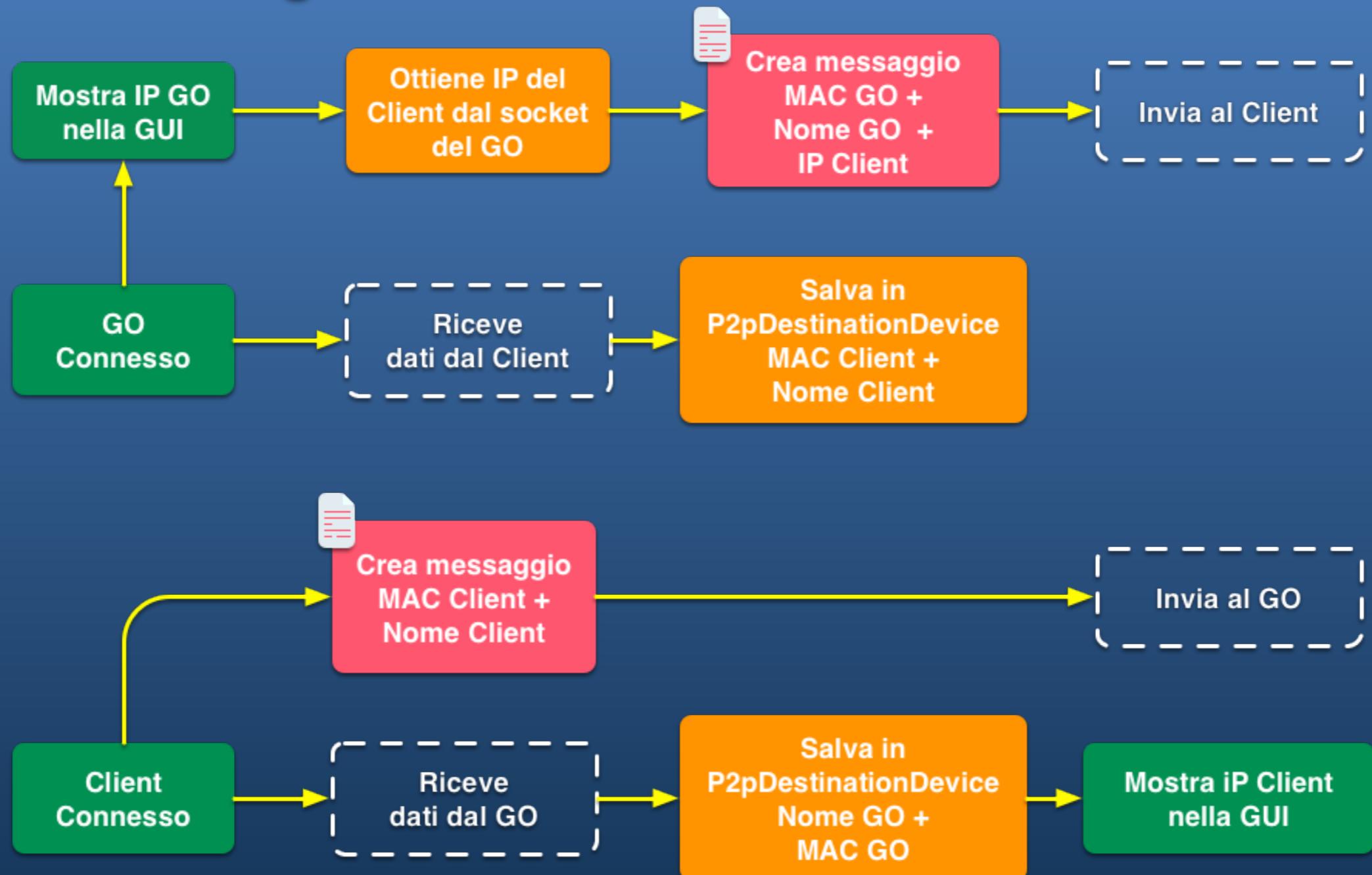
## Pigeon Messenger





# Applications

## Pigeon Messenger





# Applications

PingPong



- Un dispositivo “salta” da un gruppo all’altro
- Esegue il trasferimento di un file
- Gestisce le riconnessioni automatiche

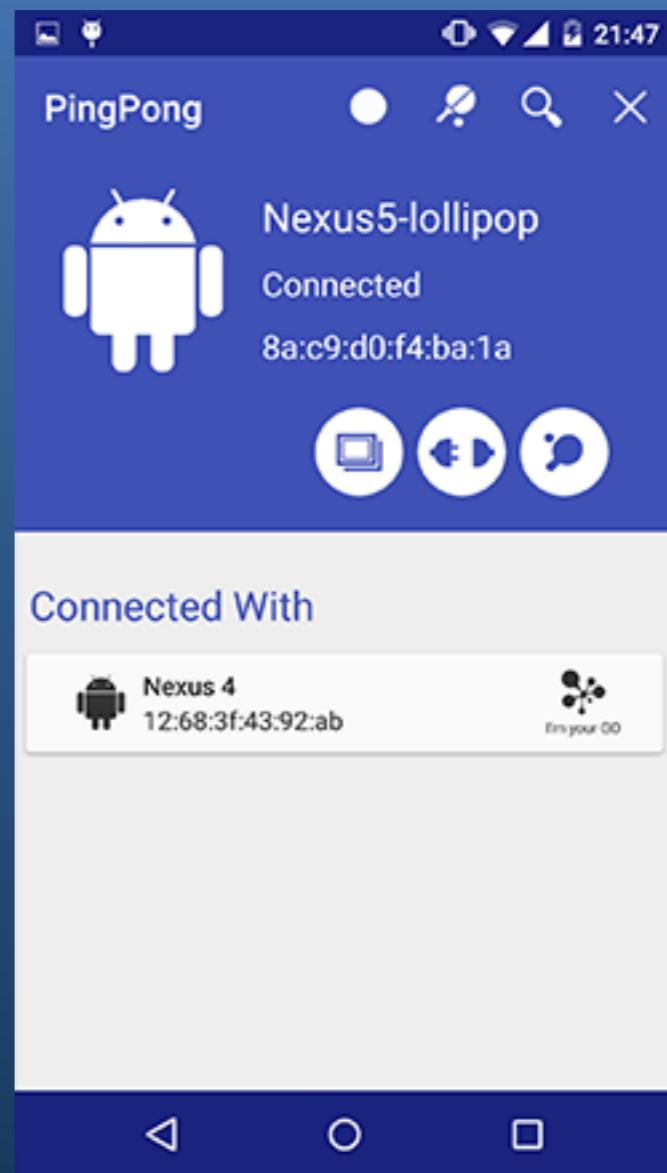


Convincere l’utente di poter partecipare  
a più gruppi contemporaneamente

# Applications



## PingPong





# Applications

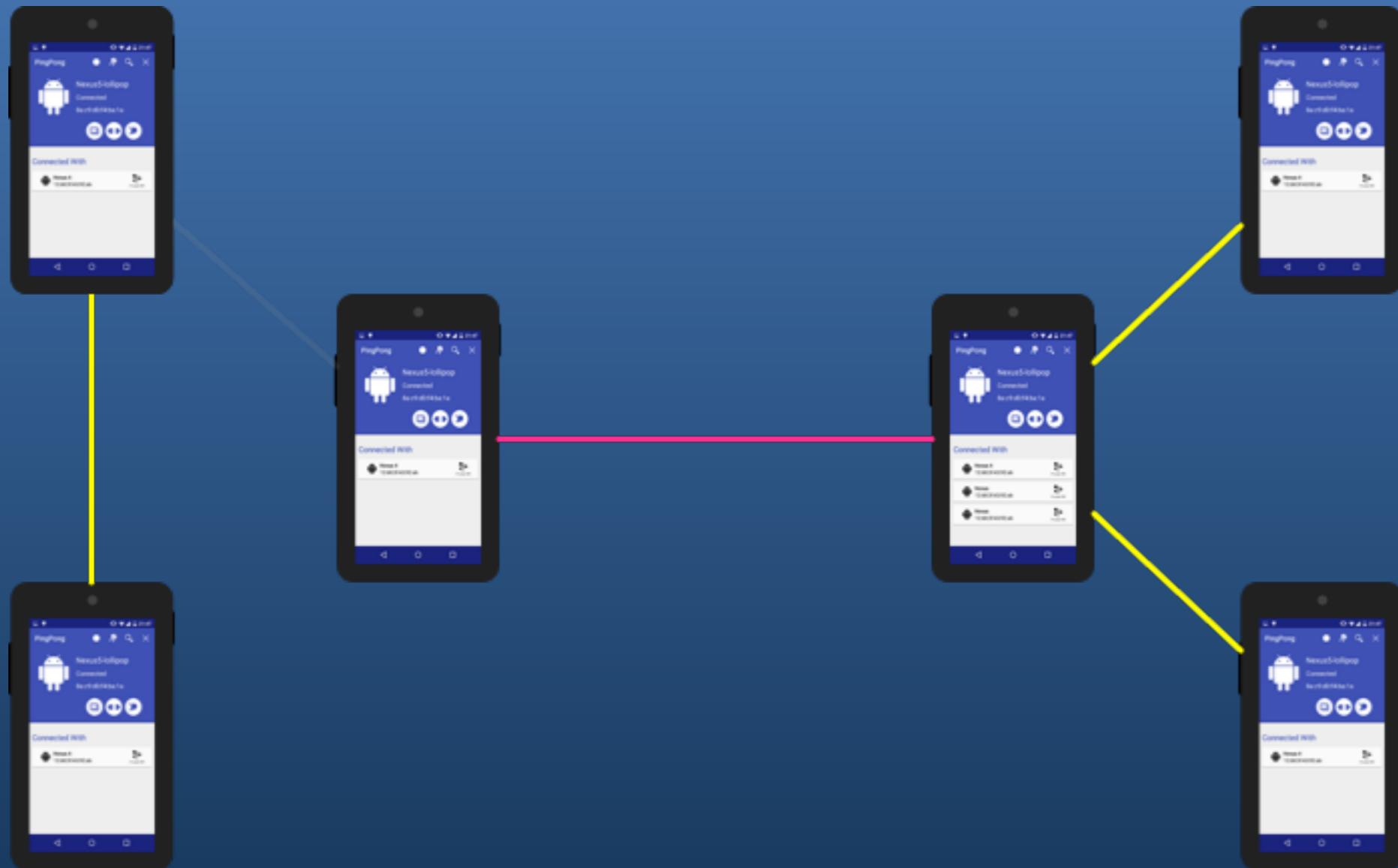
## PingPong



# Applications



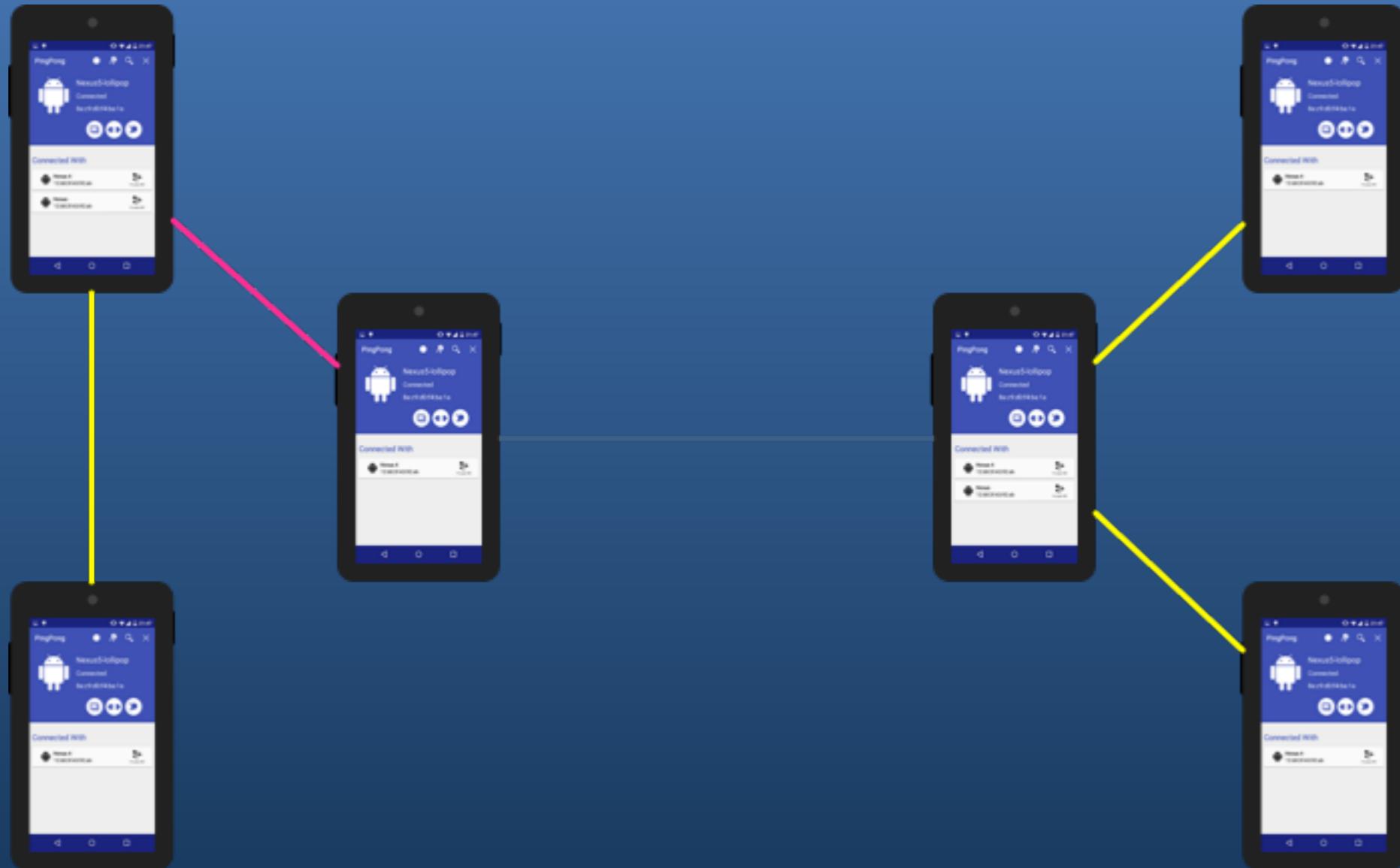
## PingPong

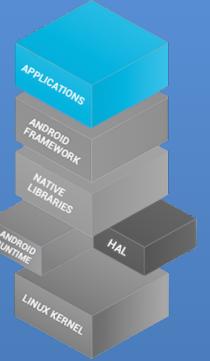


# Applications



## PingPong





# Applications

## PingPong



- Idea realmente attuabile con le API di Android
- La fase di Discovery è troppo lenta
- Richiede la sincronizzazione tra i dispositivi



PingPong funziona in un ambiente controllato,  
ma non è utilizzabile in uno scenario reale



# Applications

## PingPong



### Ciclo PingPong lento a causa:

- delle basse performance della fase di Discovery
- della difficoltà di sincronizzazione dovuta da Android