



# **Interoperable privacy preserving user identity and discovery for E2EE messaging**

Draft proposal for Mimi  
Giles Hogben, Femi Olumofin

# Functional Requirements

For a given messaging service identity handle (Phone number or alphanumeric UserID):

1. Retrieve key material and message delivery endpoint
2. Return optional default receiver service ID user preference

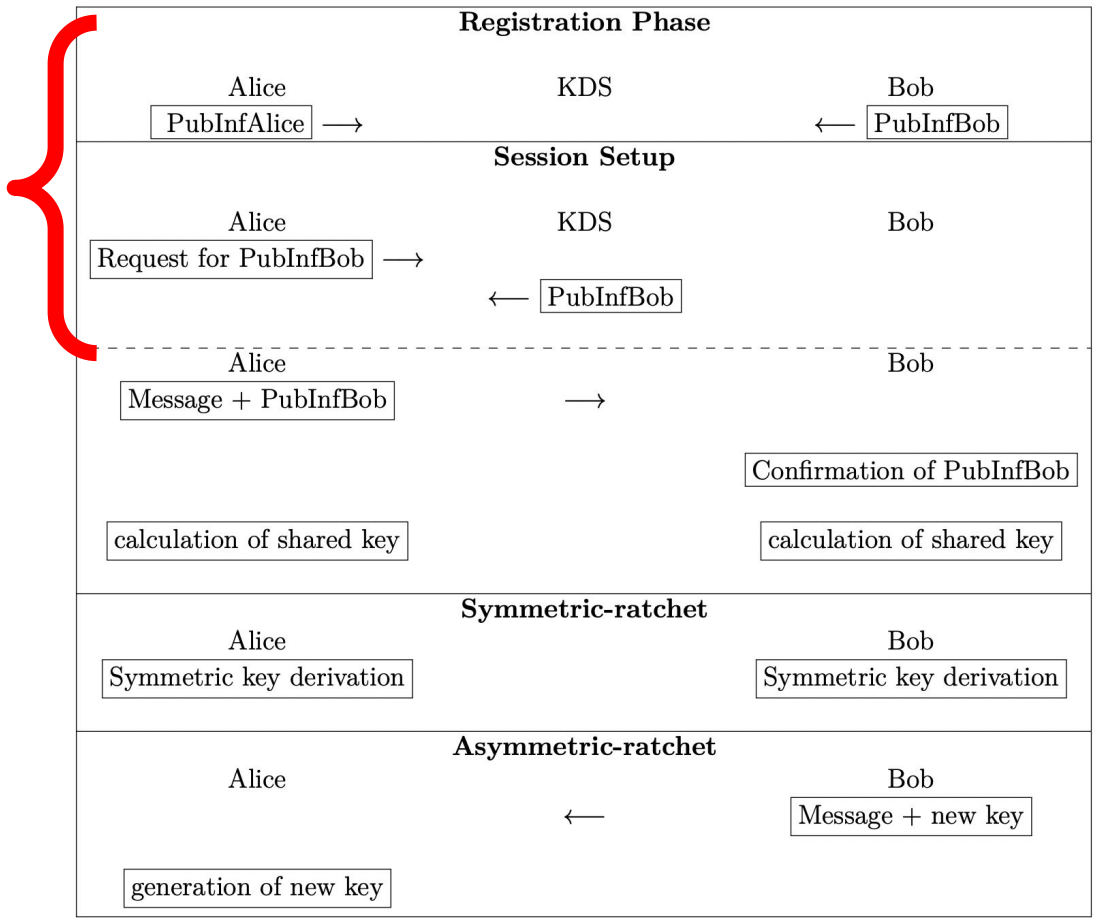
# Privacy Requirements

- Resolver service should not learn the UserID a client is querying for
- Resolver service should not learn the public identity of the querying client (i.e. who is sending a message to who)
- Resolver service should not learn the exact timing of when a message is sent

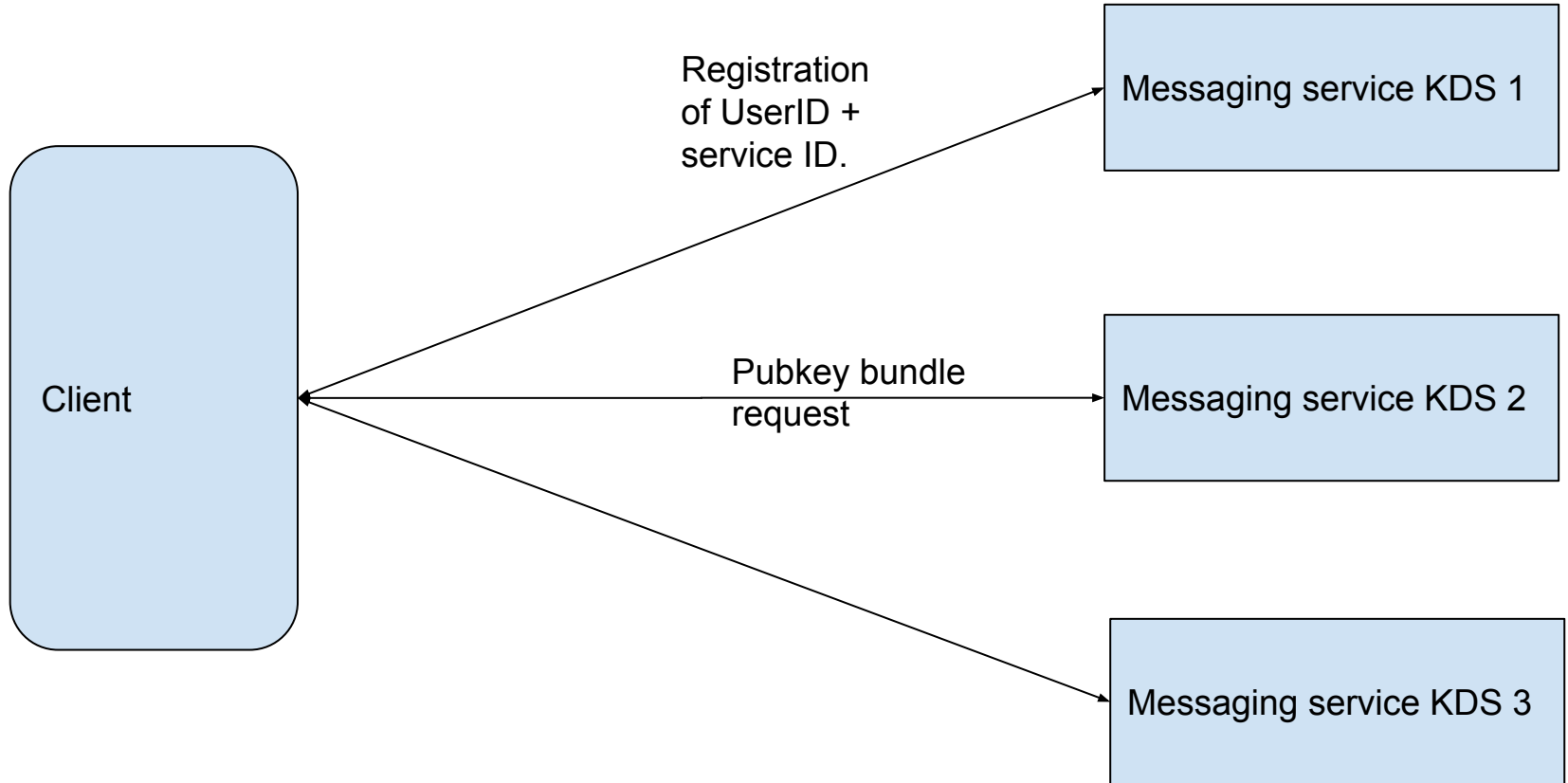
# Privacy non-requirement

## **Hiding service reachability**

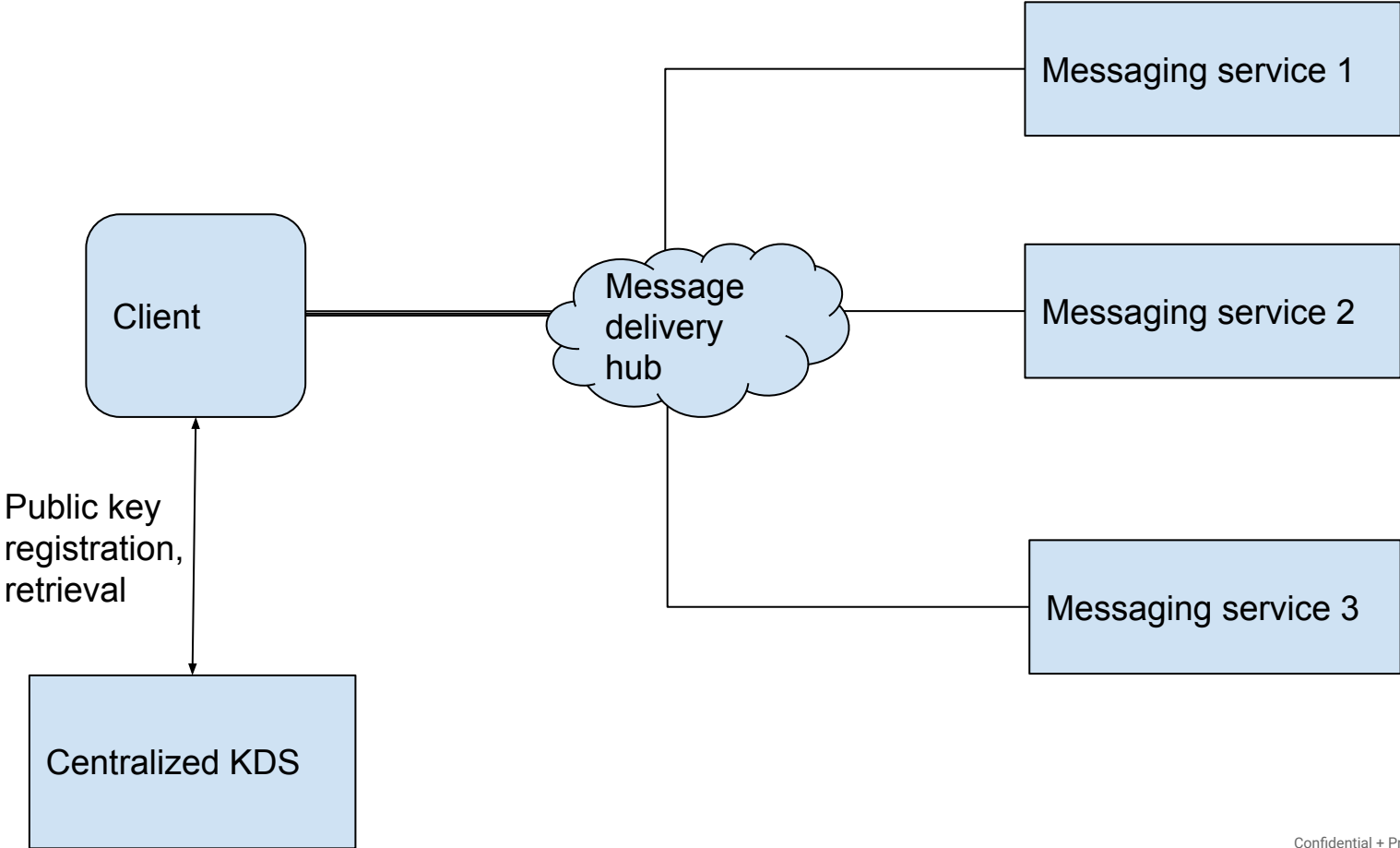
- All major E2EE messaging services already publish unACL'd reachability information without opt-out. e.g. +16501234567, reachable on Whatsapp.



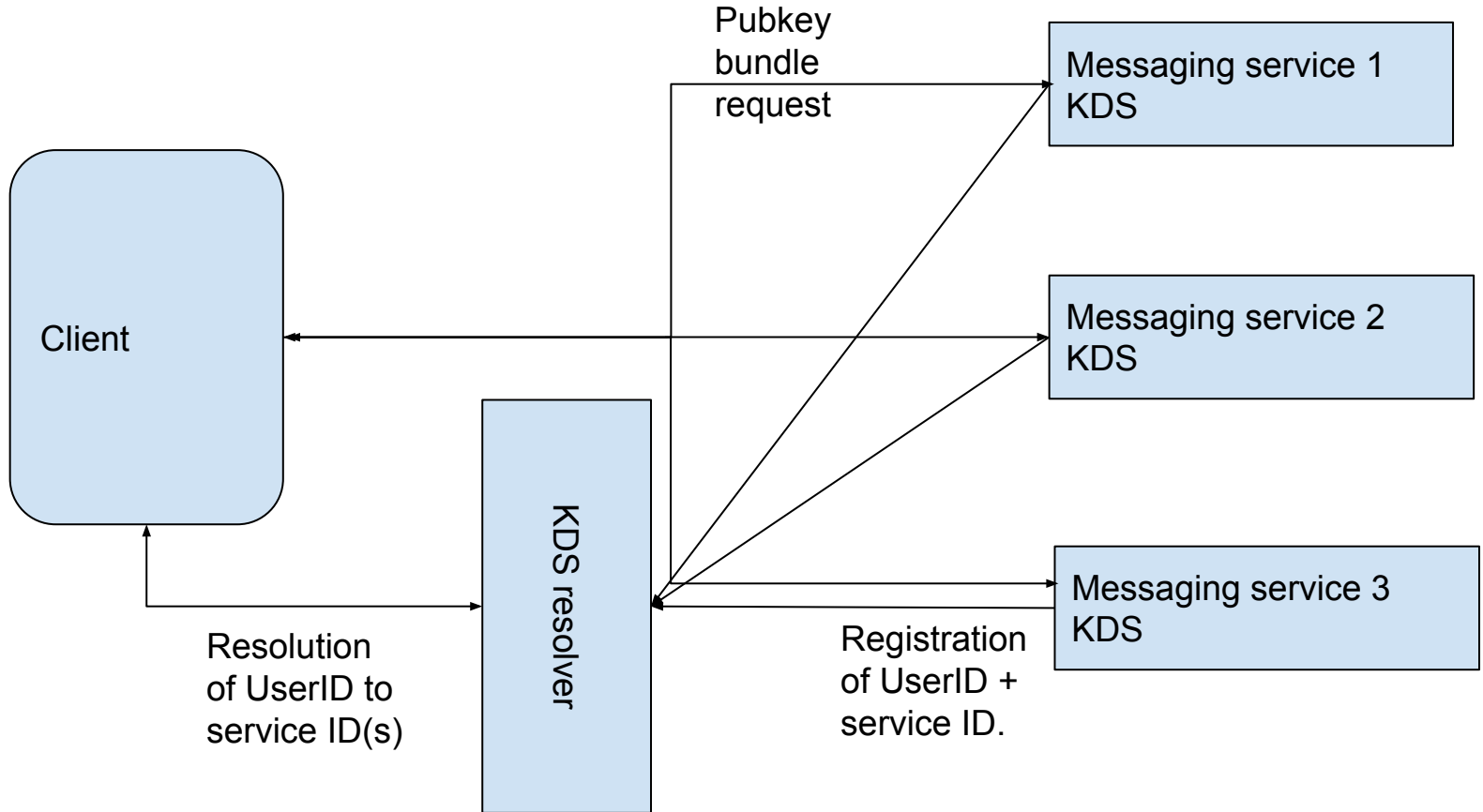
# Option 1: brute force query - too expensive and leaks private info



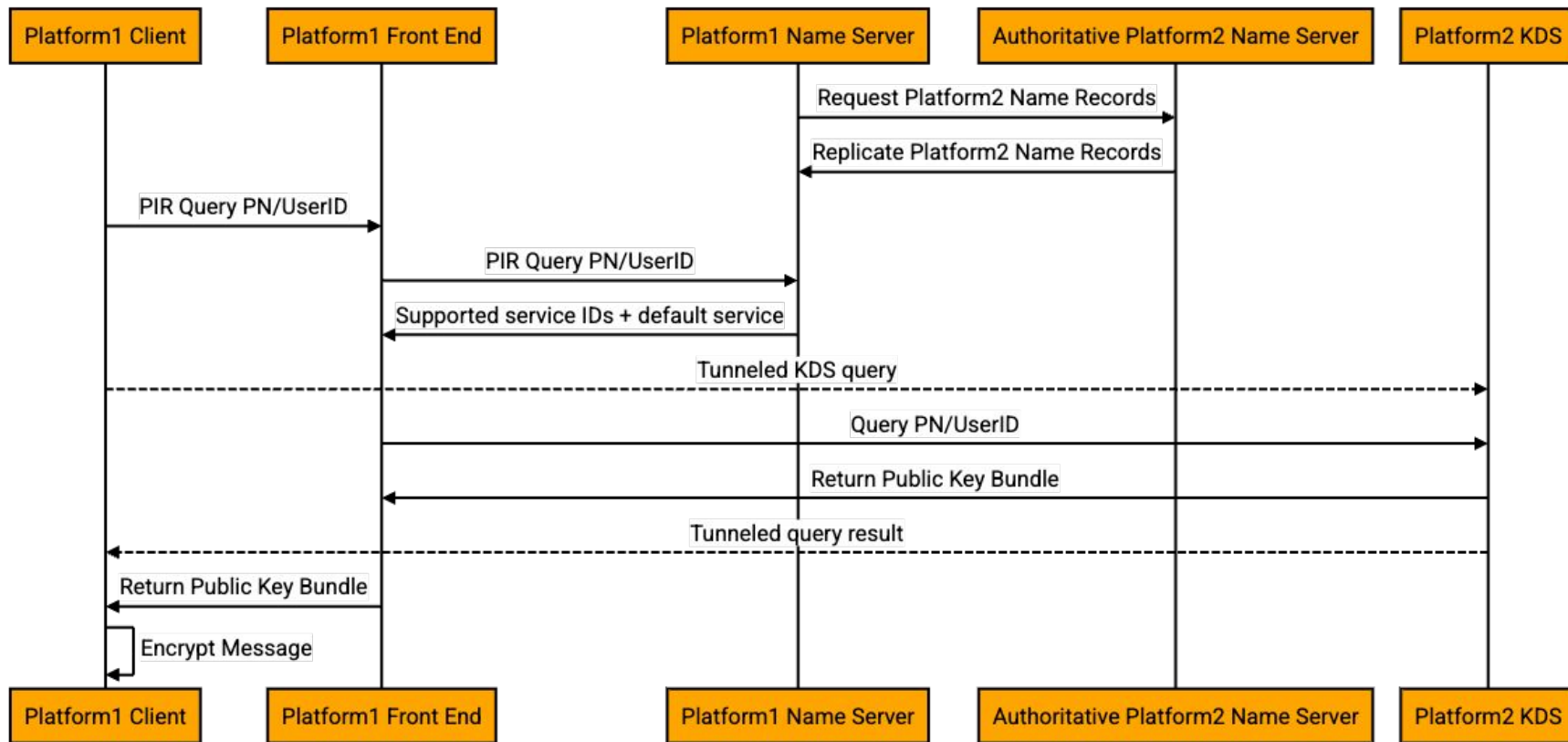
# Option 2: Centralized hub - expensive and organizationally complex



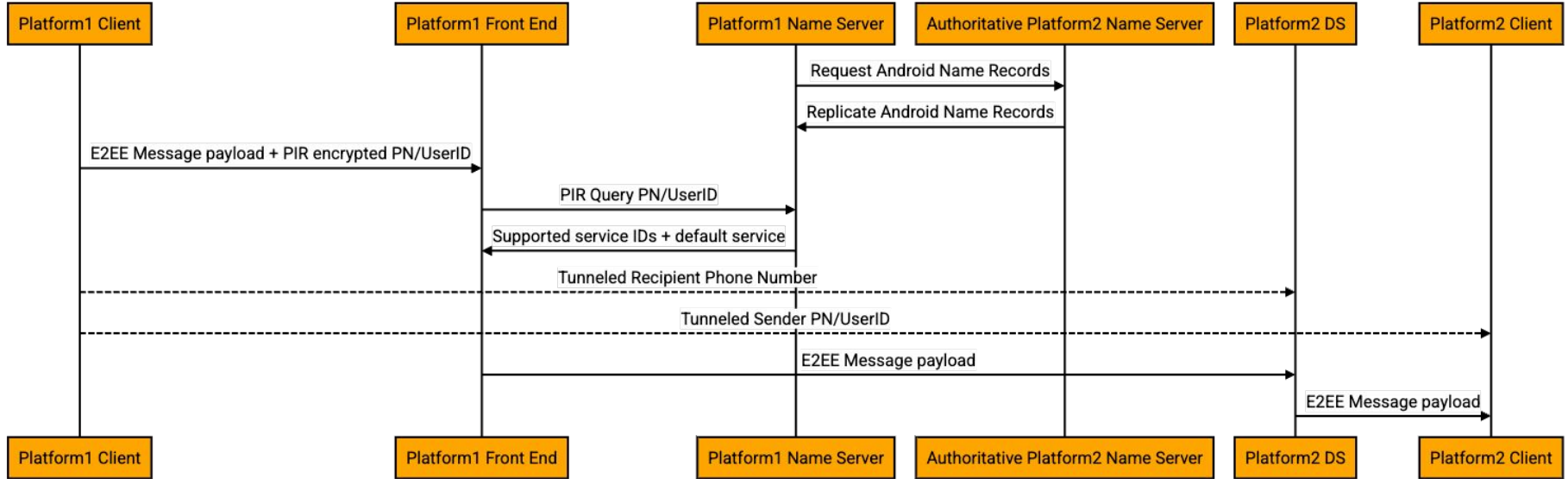
# Option 3 (Preferred): Federated with KDS resolver service



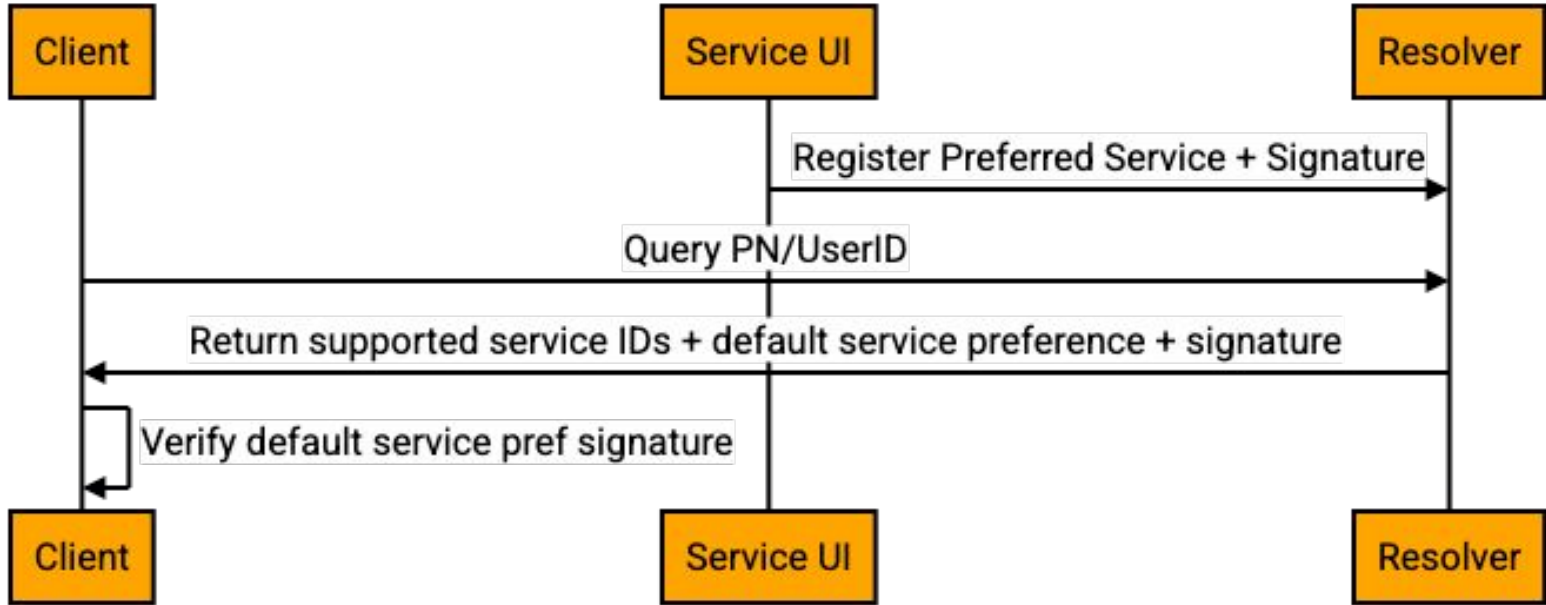
# Key distribution



# Message delivery (similar to key distribution)



# Preferred service integrity



# Privacy of resolver queries

- Goal: prevent leakage of the user's social graph to resolvers and other parties
- Setting: User may query a PN/userID in an ad hoc manner or in a batch (e.g., key bundle download for all of a user's address book contacts)
- Our proposal: Private Information Retrieval (PIR)
  - Google's PIR framework to transform any standard lattice-based homomorphic PIR scheme into efficient keyword PIR
  - Approach is feasible with privacy - cost tradeoff that we consider as reasonable

# Homomorphic Encryption

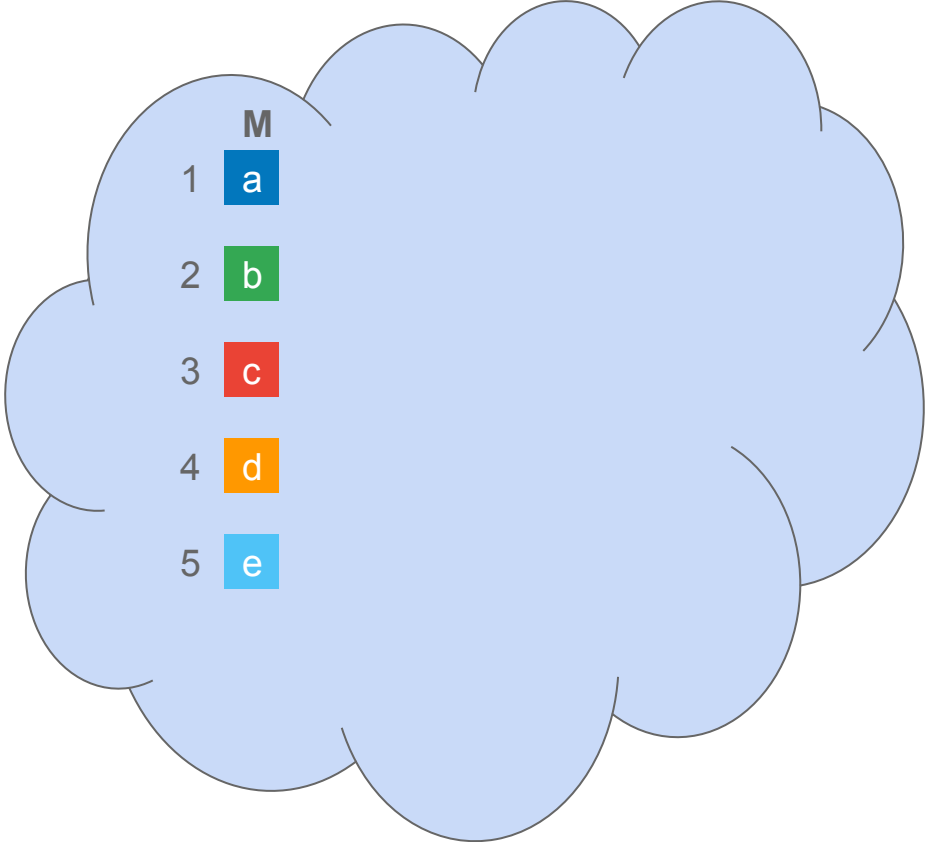
$$\text{[Red box with lock icon and 'a']} + \text{[Yellow box with lock icon and 'b']} = \text{[Orange box with lock icon and 'a+b']}$$

$$\text{[Blue box with 'a']} \times \text{[Yellow box with lock icon and 'b']} = \text{[Green box with lock icon and 'a*b']}$$

(Not encrypted)

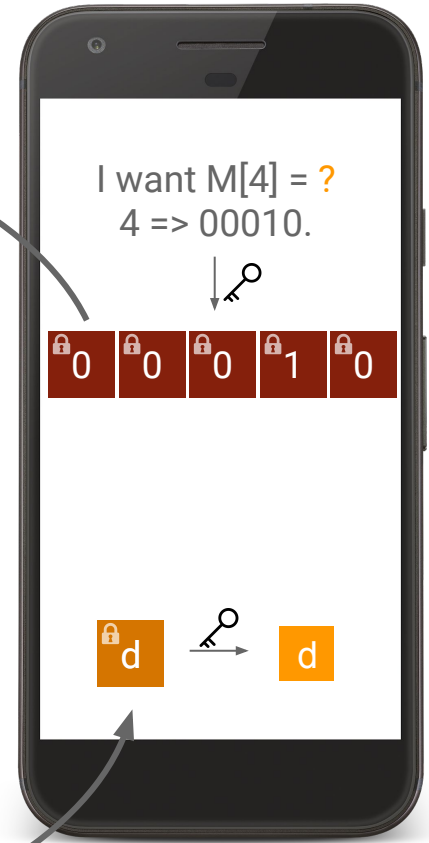
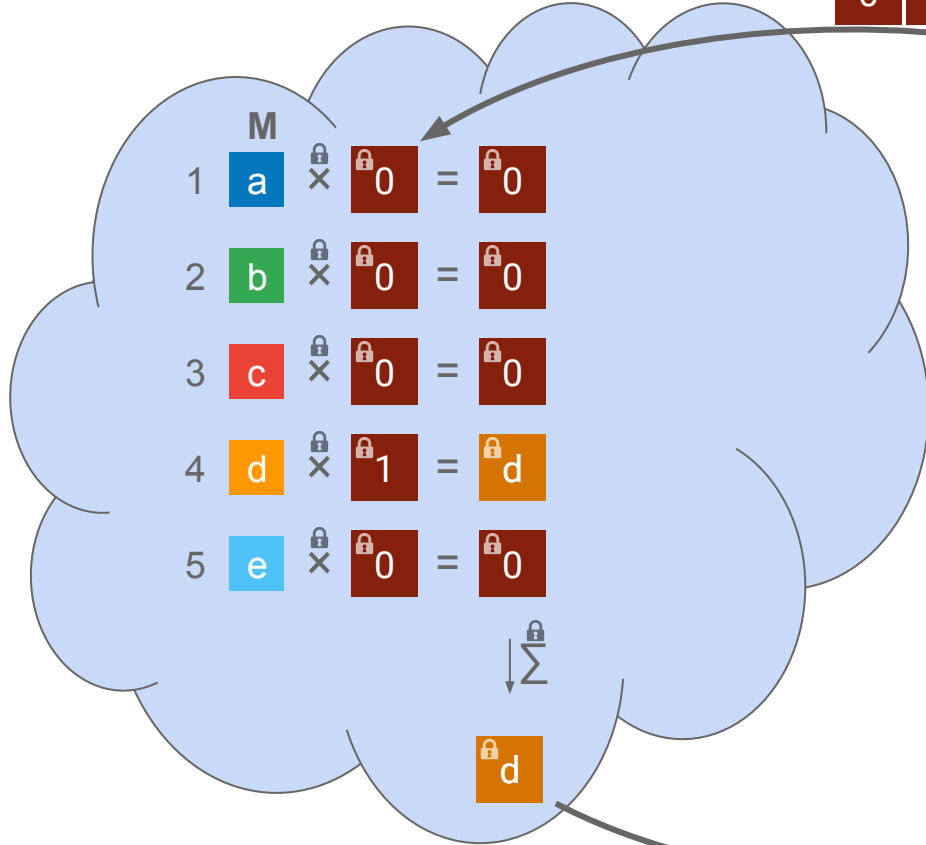
$$\text{[Dark blue box with lock icon and 'a']} \times \text{[Yellow box with lock icon and 'b']} = \text{[Green box with lock icon and 'a*b']}$$

# Private Information Retrieval



# Private Information Retrieval

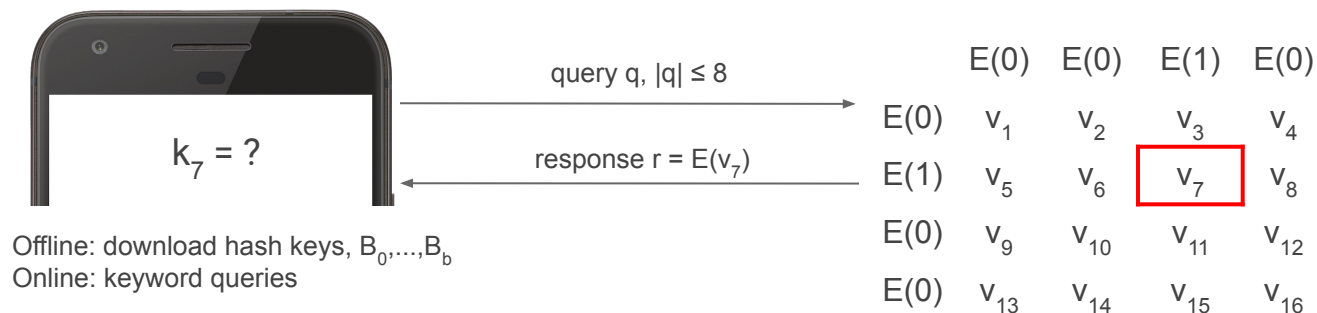
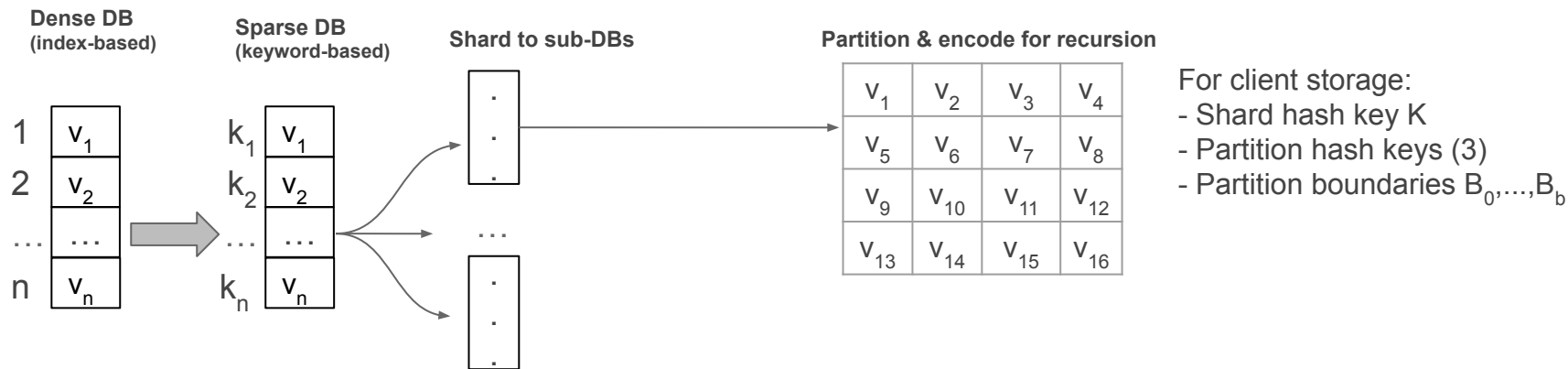
$\begin{matrix} \text{0} & \text{0} & \text{0} & \text{1} & \text{0} \\ \text{0} & \text{0} & \text{0} & \text{1} & \text{0} \end{matrix} ?$



# Google Keyword PIR framework

- Framework transforms standard lattice-based PIR schemes into keyword PIR
  - User has a query PN/userID, not the index of the DB record
- Encodes a sparse DB as linear combination of its records
  - DB size reduction using small additional client storage
  - Compatible with recursion
  - Ensures minimal noise growth for fully homomorphic encryption
- Performance
  - 2x reduction in response size
  - 2x reduction in response overhead for batch PIR

# Google Keyword PIR framework



# Cost estimates

## Assumptions:

- 10BN records
- Size 1.28 TB
- 10k shards -> 1M records each

Parameter	Cost estimate
PIR Public Key Size Per Device (storage required)	14 MB
Upload Bandwidth Per Query	14 KB
Download Bandwidth Per Query	21 KB
Client Time Per Query	0.1s
Server Time Per Query (Single Thread)	0.8-1s

# Questions

# Cross-service identity spoofing

- Alice messages Bob at Bob's preferred service (bob@Threema)
- Eve messages Alice impersonating Bob using bob@FooService
- Alice needs some indicator or UI to know that bob@Threema isn't bob@FooService and that when bob@FooService messages, it should not be assumed that bob@FooService is bob@Threema.