E2EE Payer Notes: Private memos for Lightning Payments

Jin
Date: 26.11.06

btc++
privacy edition
Seoul

Jin

@Jin_1337



Opensource Developer

cashubtc/COCO

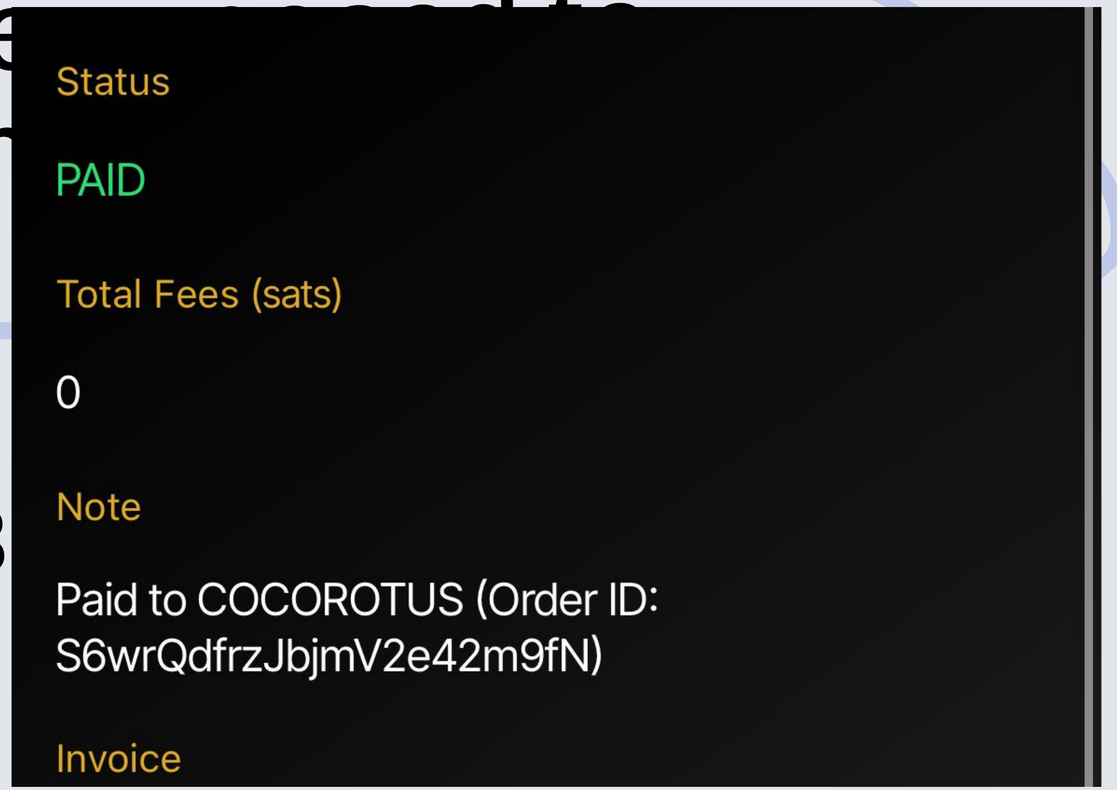
Bitcoin in pusan - casual meetups, workshops

Every notes are exposed to service provider

..If you are using 3rd parties Infrastructure

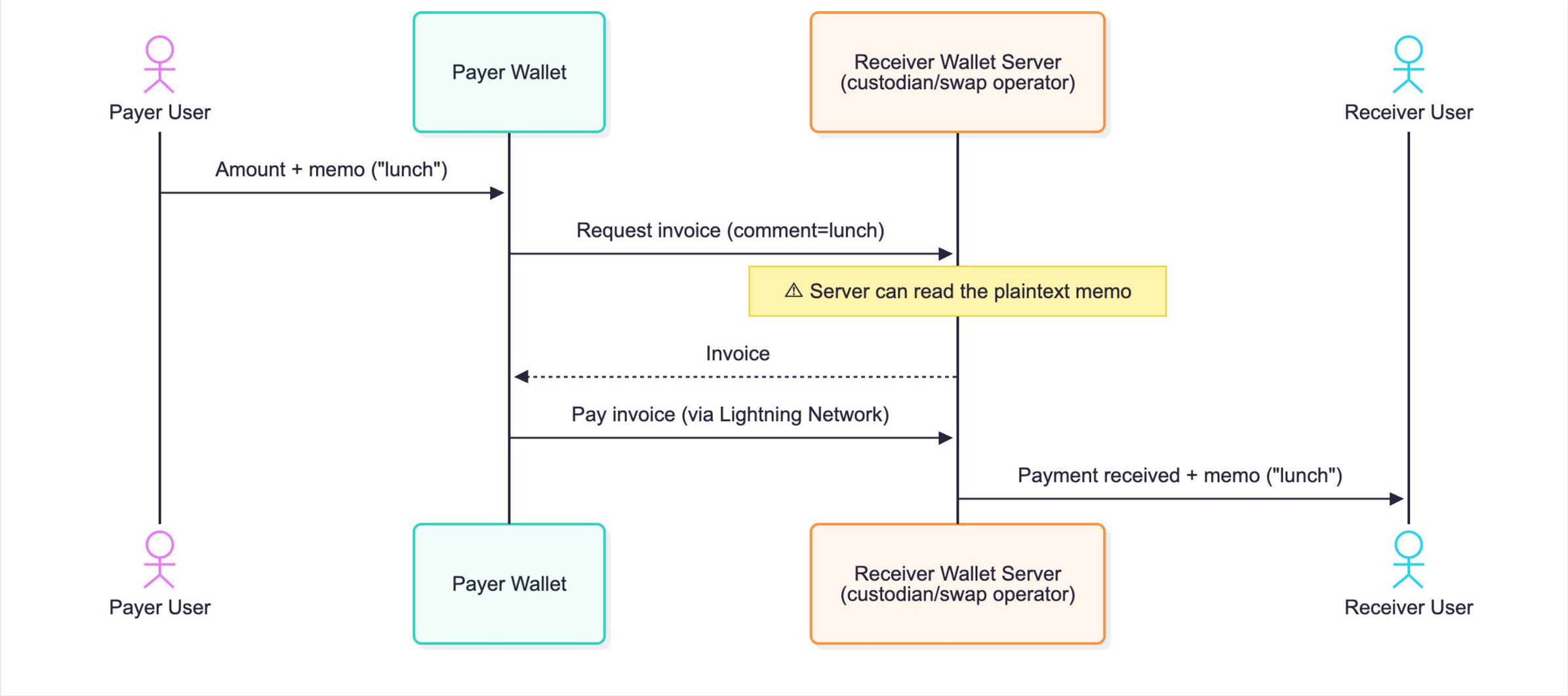
eg (Ark, Cashu, Custody Wallets, LN swaps/bridge)

Every notes are
service pr

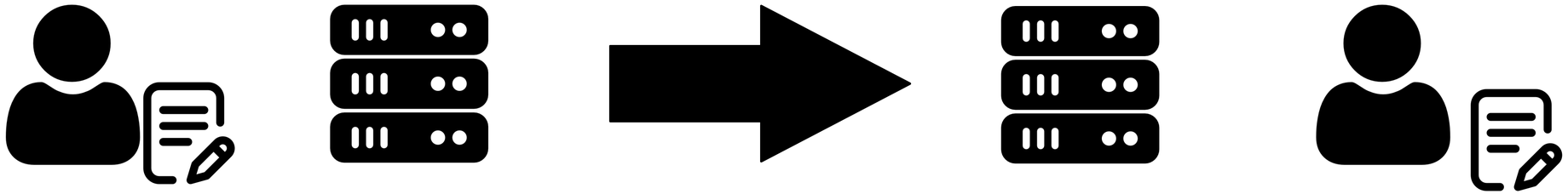


..If you are using 3
Infrastructure

eg (Ark, Cashu, Custody Wallets, LN swaps/bridge)


Payer User
Payer Wallet
Receiver Wallet Server
(custodian/swap operator)
Receiver User
Amount + memo ("lunch")
Request invoice (comment=lunch)
⚠ Server can read the plaintext memo
Invoice
Pay invoice (via Lightning Network)
Payment received + memo ("lunch")
Payer User
Payer Wallet
Receiver Wallet Server
(custodian/swap operator)
Receiver User

Plain text stays Their Infra



Affected protocol

BOLT11
Description(d)

LNURL-pay
comment

BOLT12
invreq_payer_note

Custody & Swap/Bridge service affected.

Onion Routing is only meaningful when you have your own node.

Treat Scenario

- Custody wallet
- Swap/bridge service

Payment Detail & orderID leak

bLIP Proposal

Adding Cyphertext into existing field

1. No wire format changing
2. Using already verified primitive NaCI Crypto_box
3. Optional implementation

Spec summary

Like PGP wise, encrypted using the recipient's public key and send.

Key separation (MUST)

Dedicated X25519 Keypair app sk/app_pk,(different from node_pubkey)

Pubkey Discovery (MUST)

Out-of-band: LNURL-pay metadata (eg.encryptionPubkey)
BOLT12 cutom TLV

Payload struct

Version Prefix	payer_app_pk 32B	nonce 24B	cipher text msg + MAC 16B

Encode/Decode

Payer App

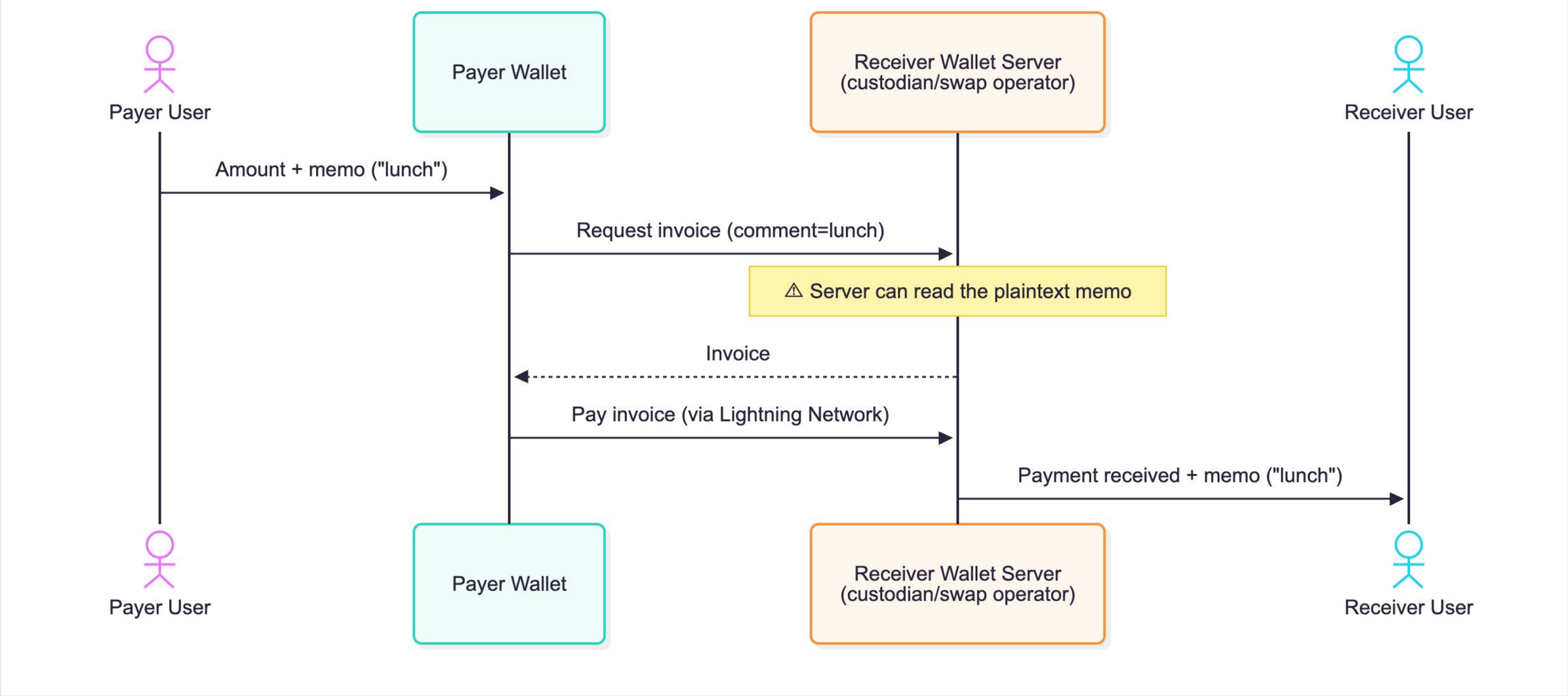
1. nonce = random(24B), one time use only.
2. ciphertext = crypto_box (msg, nonce, receiver_app_pk, payer_app_sk)
3. Payload = 0x01 || payer_app_sk || nonce || ciphertext. encoding with base64

Receiver App

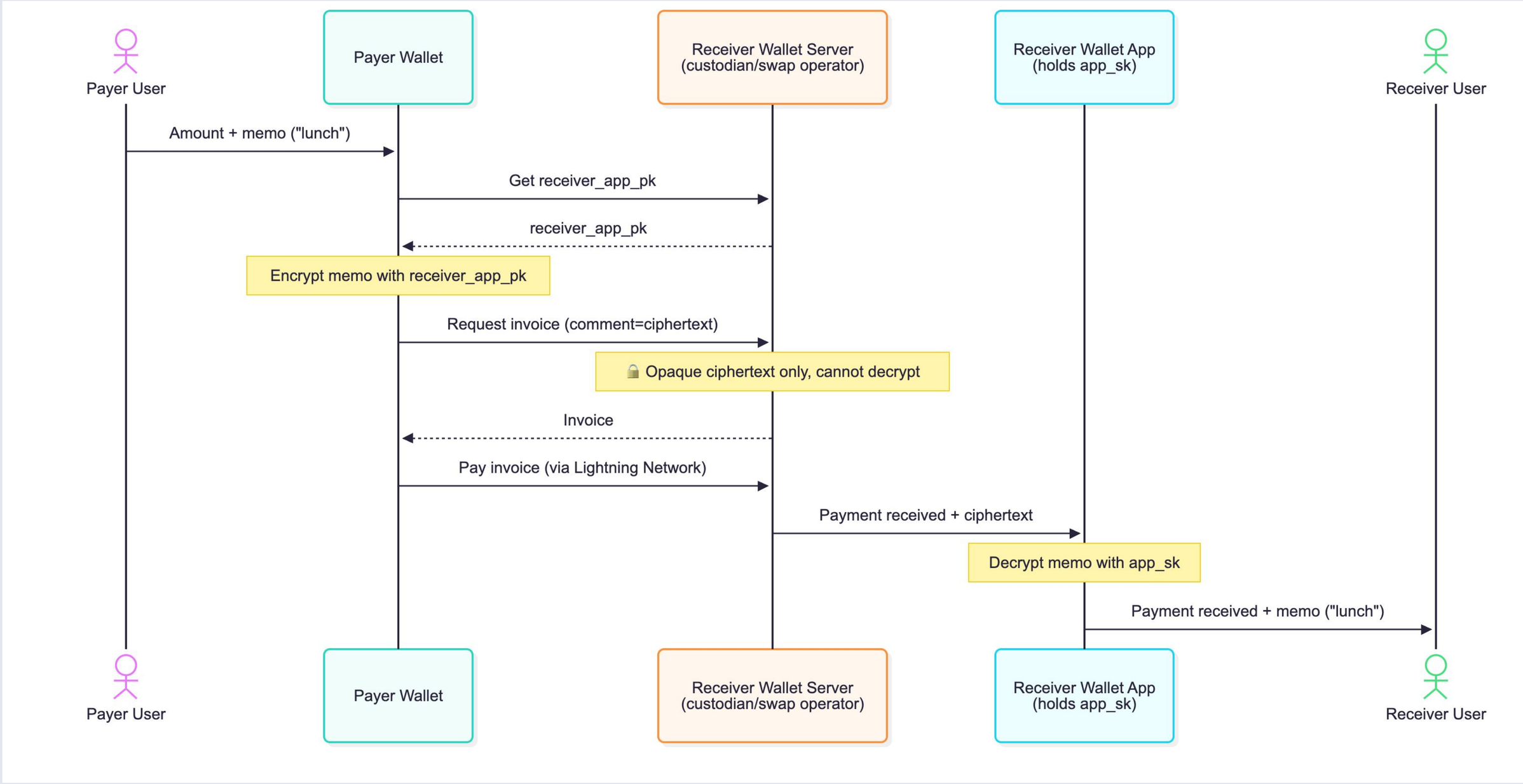
1. app receives event from server
2. detect prefix, crypto_box_open(ciphertext, nonce, receiver_app_sk)
3. treat as plain text if no prefix.

E2EE Note
Applicable Sections

Section	Key discovery	Note input	Applicable?
LNURL-pay	API respond field	comment	✅
BOLT12	custom TLV in offer	invreq_payer_note	✅
BOLT11	None	None	❌


Payer User
Payer Wallet
Receiver Wallet Server
(custodian/swap operator)
Receiver User
Amount + memo ("lunch")
Request invoice (comment=lunch)
⚠ Server can read the plaintext memo
Invoice
Pay invoice (via Lightning Network)
Payment received + memo ("lunch")
Payer User
Payer Wallet
Receiver Wallet Server
(custodian/swap operator)
Receiver User

After encryption



What's Changing?

1. App layer

A new key pair (app_sk / app_pk) is created in the user's app. This is separate from the node_id.

2. LNURL-pay

A public key field (public_key) needs to be added to the endpoint.

3. BOLT12

A custom TLV needs to be added to include the public key in the offer.

What It Solved, And It’s Limits

Solves

- Note contents are no longer exposed to infrastructure and service providers





Limits

- A server can forge the receiver's public key. If the payer stores the key, later changes are detectable
- Blinding the receiving node is still impossible with BOLT11 (possible with BOLT12)
- Unusable if commentAllowed is shorter than the payload

E2EE Payer Notes: Private memos for Lightning Payments

Jin
Date: 26.11.06

btc++
privacy edition
Seoul