

Nico Schottelius, 2012-06-19

## Onion Routing

Anonymity is an important feature for freedom of speech.

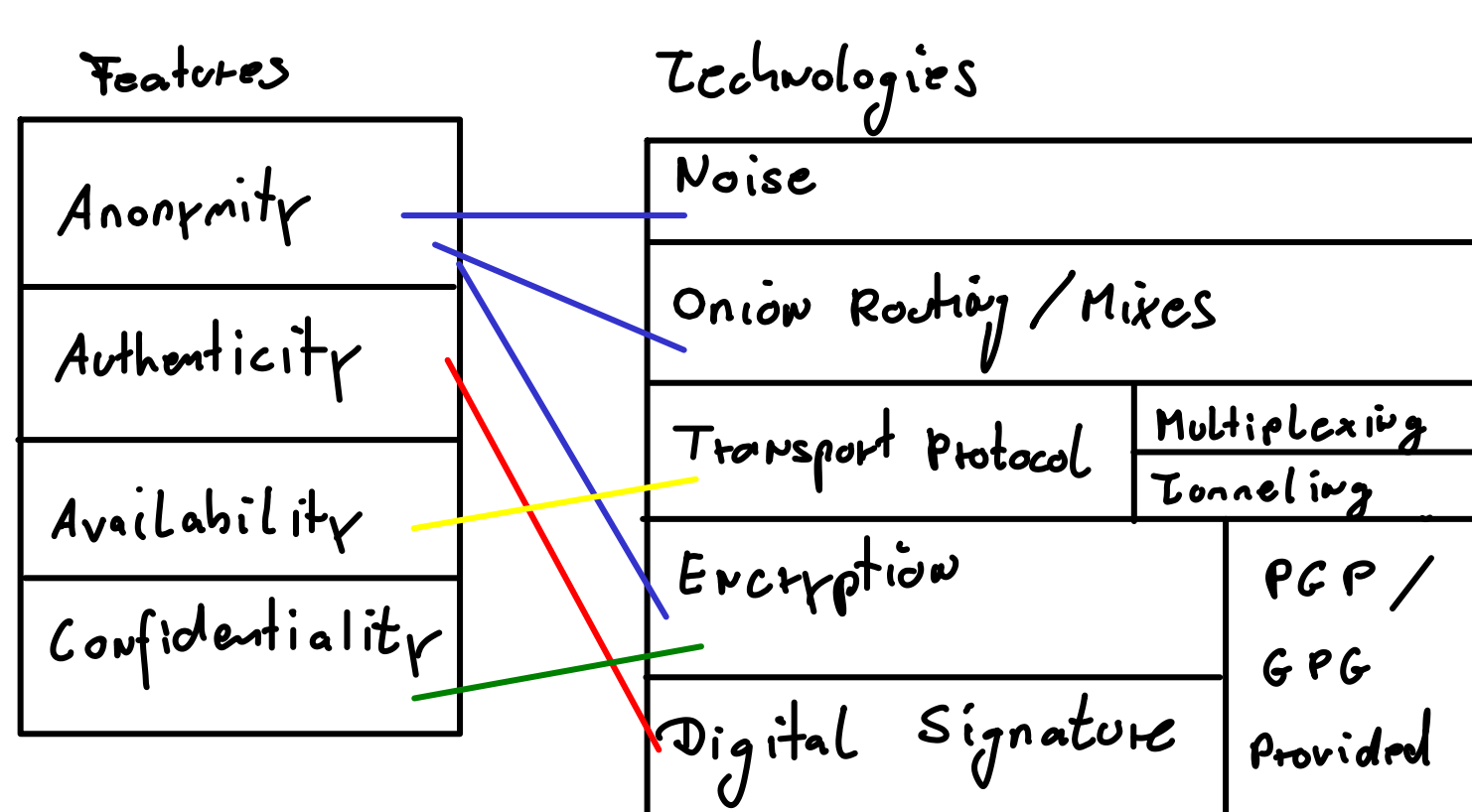
Anonymity systems are often heavy weighted or do not provide end-to-end anonymisation.

Current chat systems are not designed to provide anonymity.

Even worse, some are designed to hide from the user what they are doing.

This thesis presents a secure, peer-to-peer, decentralised anonymous chat system.

## Anonymisation Techniques



## Packet Types

of messages - plain text

onions - multiply times encrypted

postcards - onions + transport protocol frame

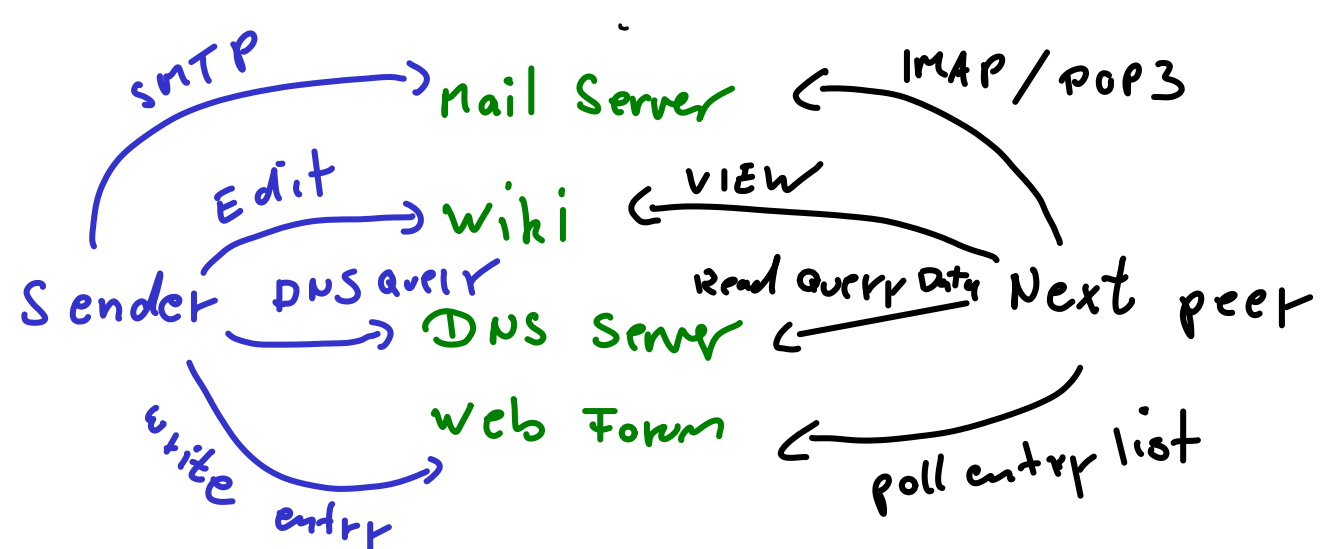
End messages

Command	Drop/Forward	Message/Voice
3000	Drop	Voice
3001	Forward	Voice
3002	Drop	Message
3003	Forward	Message
3004	A	C

→ cmd id addr group msg

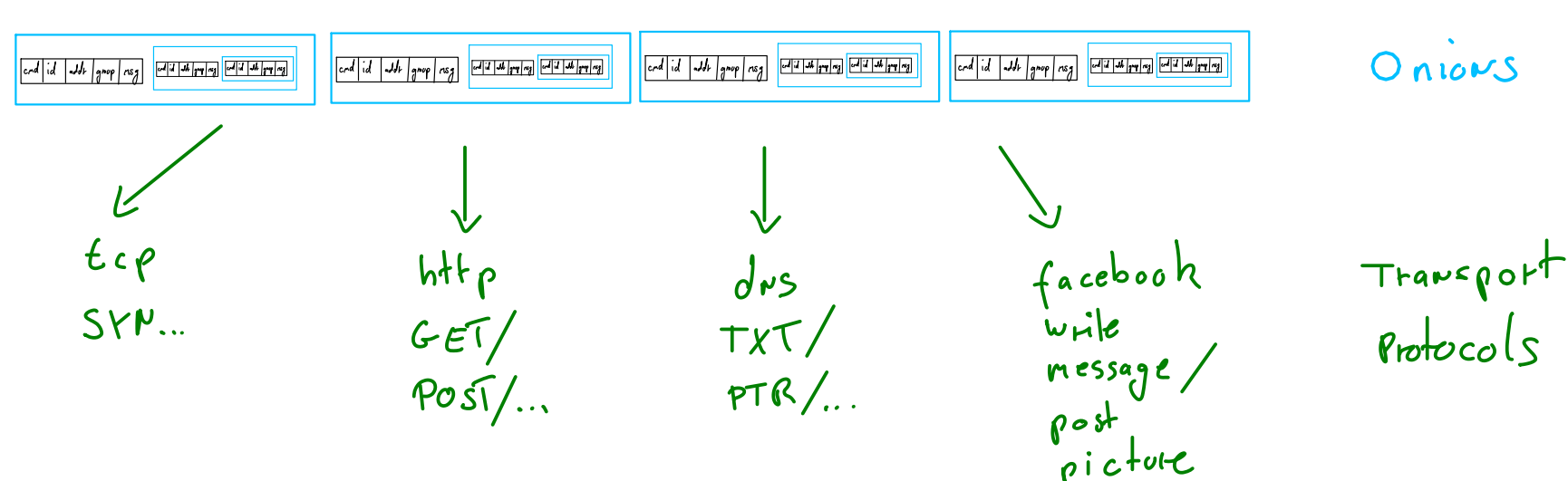
## Transport Protocols

The chat system is transport protocol agnostic: It can use arbitrary protocols for transport.



## Transport Protocol Tunneling

To make life harder for an attacker, onions can be tunneled via existing protocols like HTTP, SMTP, DNS or Facebook.



## Transport Protocol Multiplexing

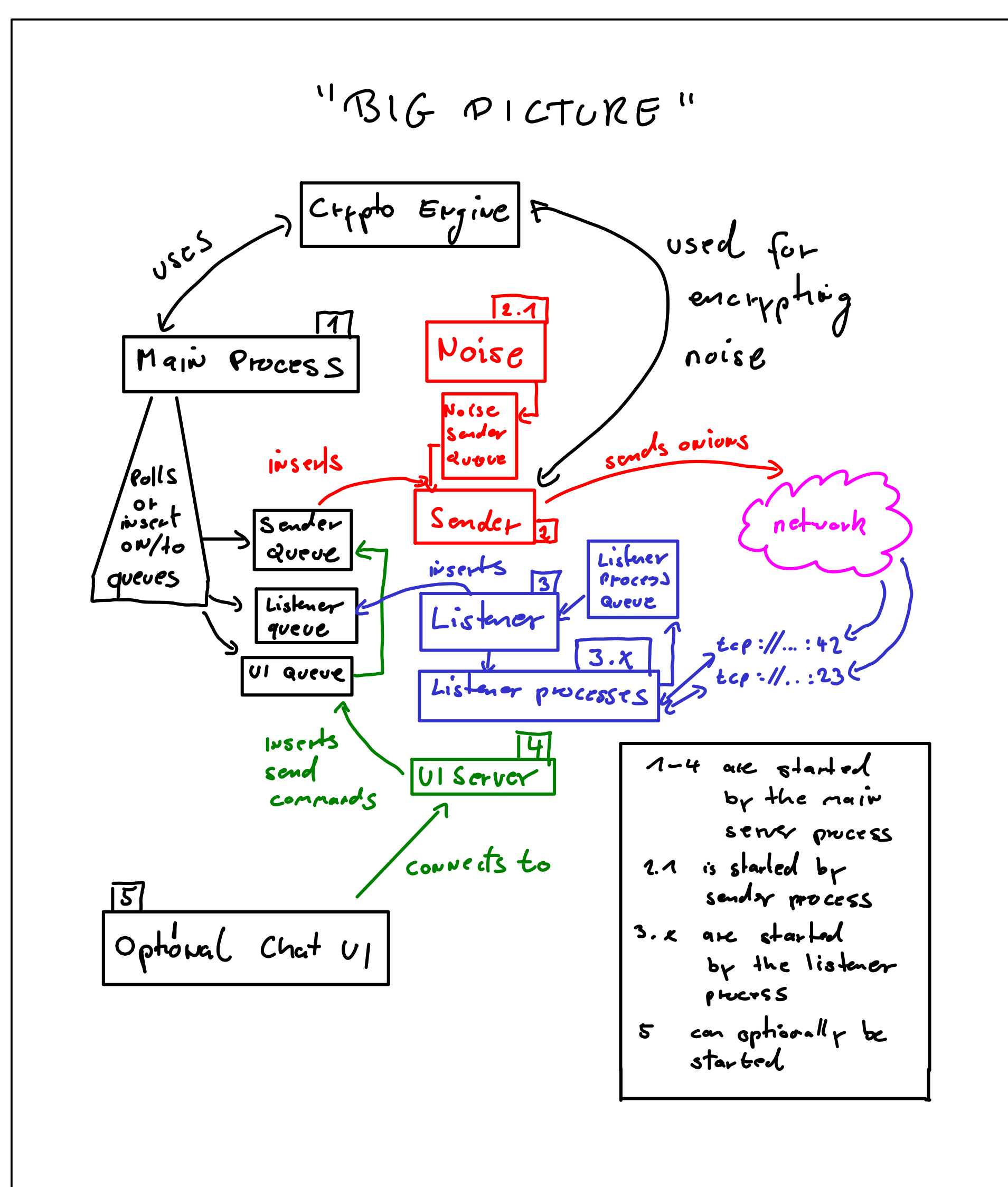
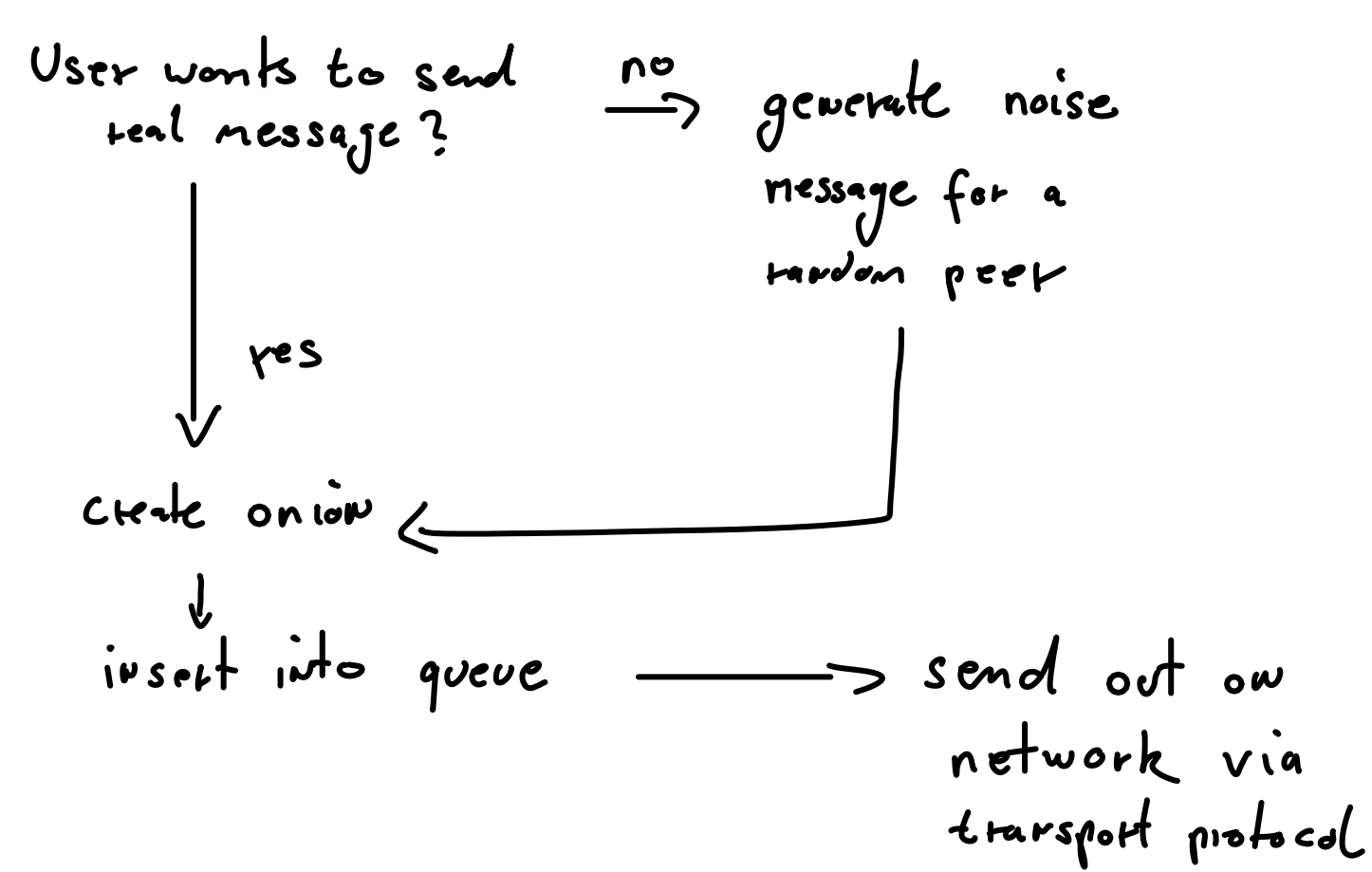
To enhance availability every peer can listen to a variety of different addresses. Every address could be supported by a different transport protocol.

Noise is used to prevent an attacker to do statistical analysis on the network traffic.

Every peer sends a postcard every 250 ms or 4 postcards per second.

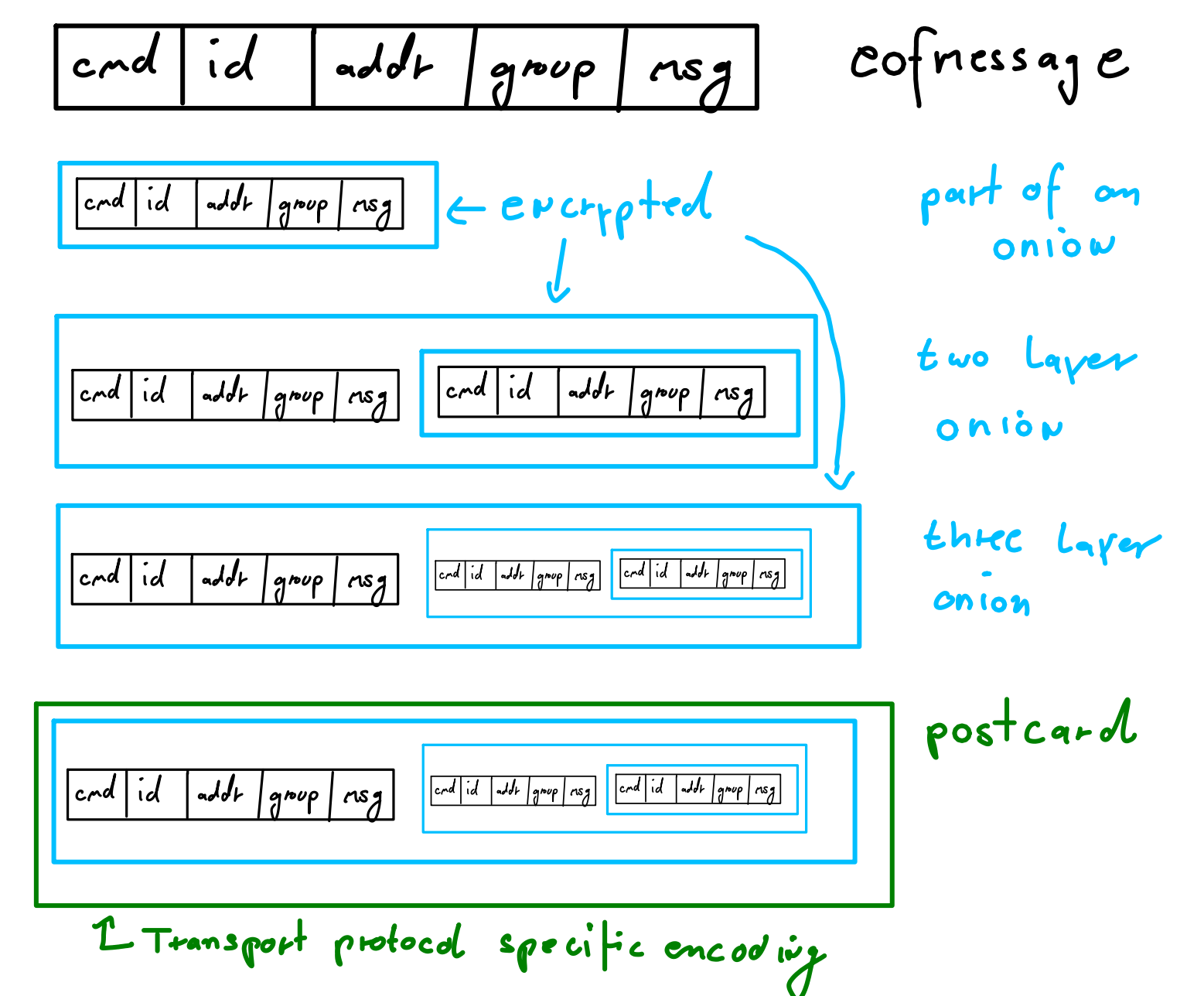
with an average postcard size of 6 KiB, this results in a continuous network flow of 24 KiB/s or 192 Kbit/s.

## Noise workflow



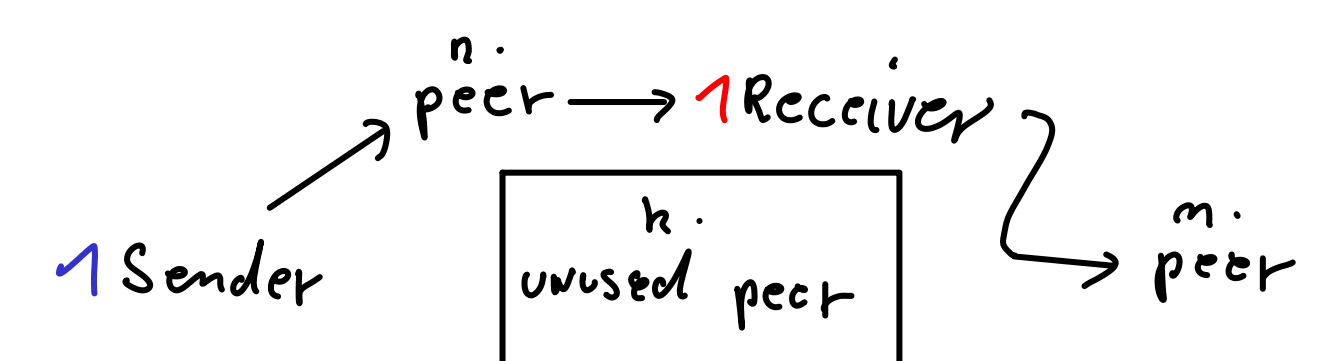
An Onion is a multiply encrypted message. Every recipient can read one part of an onion and knows only about its predecessor and successor.

The readable part is called `eofmessage` and always consists of the same fields.



To hide the real recipient of a message, the recipient is inserted at a random position of the onion.

$$\begin{aligned} n &\geq 0 & \text{proxp-peers} &= n + m \\ m &\geq 0 & \text{route-peers} &= \text{proxp-peers} + 1 \\ k &\geq 0 & \text{network-peers} &= \text{route-peers} + k \end{aligned}$$



De-Anonymise :

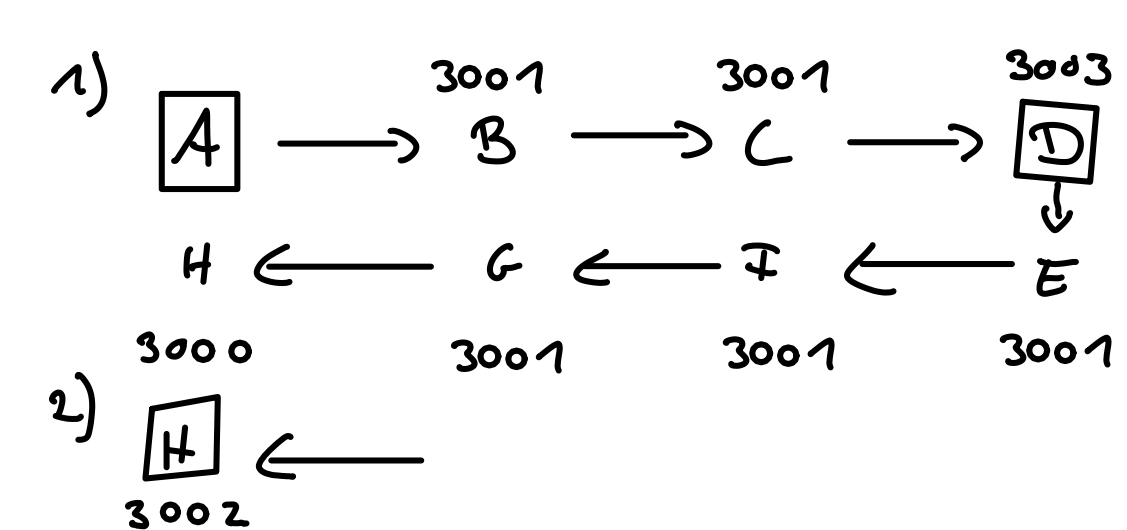
- Compromise  $\uparrow$  Sender
- Compromise proxy-peers
- Compromise  $\uparrow$  Receiver

## How to create an onion

Sender

- 1) Generate route  
Receiver is at random position  
Use random address of  
each peer
- 2) Create Onion  
Encrypt for last peer first  
Re-encrypt until first peer

### Example Onion Route



## Conclusions

- The proposed chat system works - proven by tests of the prototype.
- For real world usage, many participants are required.
- Usability should be improved.