Chat SDK on Harmony

Phase 1

Scope:
A SDK that Harmony user A can send $ONE along with a text message (max 140 bytes) to another Harmoney user B. User B needs to confirm the recipient by replying a text message to A. If User B doesn't confirm within the given time out period, (e.g. 7 days), the $ONE deposited by A will be returned to A.

Paid messaging over Harmony

On a web interface, user can login with OneWallet, and then

- Choose a handle, e.g. @satoshi
- Set a minimum price to receive a message. It can be 0. Let's say it is set to 10

Another user login with her OneWallet (e.g. @satoshi-admirer), and decides to send a message to @satoshi

- She can write the message on the web UI
- She has to put 10 coins (as set by @satoshi) along with the message
- She can then send the message to @satoshi (she has to pay the gas fee too)

@satoshi gets an email that there is a message from @satoshi-admirer with 10 coins in it.

- If satoshi replies to the message, even just with one word OK, the 10 coins will be transferred to satoshi's wallet. (probably we can take 1% cut)
- If satoshi doesn't reply, after a certain period, e.g. 7 days, the 10 coins goes back to the sender.

Other properties

- The message content is encrypted by sender's private key and receiver's public key. Thus only the receiver can decrypt. Also the sender's identity is verified.

What's interesting

- Users can choose the price to get her reply to a message, depending on how much she thinks her time is worth. E.g. Celebrities can set a high price to filter out most spam.
- If the receiver doesn't read the message or decides to not reply, the sender gets back the coin.
- This may work as something like a red-envelope
- Messages are stored on the Harmony chain

Collectively, more than 5.5 billion users send over 8 trillion chat messages each year. In effect, chat is the glue that binds many economic activities both online and offline. Yet, most of the chat services are neither safe nor secure. Users are plagued by security breaches, spam, data leaks, spoofing... Let's build a more secure and safe communication environment on-chain.

To start, let's build a programmable chat SDK native to Blockchain. So we can make it faster & easier to add chat services for all Web 3 applications.Following are potential use cases:

- Make it open source, so all specifications are transparent
- Make it interoperable, so it is easy to communicate with other Blockchains
- Make it decentralized, so anyone can host and control their own server and data
- Make it fast, so we can communicate in real-time

Messaging Essentials:

- **Private 1-to-1chat**, Start with text based messaging services (ming's examples ☝)
- **Public channel partitioning** for large-scale community discussions. Chat is an indispensable part of live streaming experiences. It allows users to not just interact with fellow members of the community, but build connections with streamers and brands as well.

Business critical communication

- Enable chat as the foundational layer for CRM by offering SMBs to communicate with their customers, streamline conversations, and automate workflows. Successful chat services such as WhatsApp Business are working with marketplaces to enable seamless transactions at scale.
- DAO- Governed. transparent governance needs communication, all communication are authenticated & recorded on-chain

Purpose-built for confidentiality

- Secure and anonymous drop-offs: media organizations may be able to utilize a whistleblower submission process to securely receive and communicate with anonymous sources of information. Information is completely encrypted and does not include any personally identifying information.
- End-to-end encrypted messaging app, which allows use cases such as journalists to communicate directly with publication outlets. No metadata is

retained, configured to delete messages automatically at a designated time interval.

- Integrated PGP encryption plugins such as Mailvelope or Enigmail. This encrypts the contents of the message but not the subject line or the name of the sender.