UniMe

The Technical Whitepaper Powered by UniLab - Current Version 1.0

R&D Log:
15th of June, 2020 - v0.5
30th of June, 2020 - v1.0

Prelude

To those owning UniCash (UNW).

Our Blockchain (UniChain Mainnet) is ready. You can now send and receive UNW and experience lightning fast transaction speed.

Please test it out and send some UNW to experience the difference first hand. We are keen on thoroughly testing, updating, and perfecting the system with constant upgrades.

UniChain is designed to be especially useful for everyday payments and purchases and we are well on our way of making UNW a desirable option for vendors and merchants as well as listing it on exchanges.

When crypto currency got big with a market cap in the billions, transactions started to take hours unless the sender was willing to pay exorbitant fees. The same happened to ETH. This is not a way to revolutionize payments. Real revolution does not happen in tiny steps and improvements either - it's when something is done completely differently.

This is why Bitcoin was successful in the first place and it is time for the next phase. Truly scalable systems. Billions of transactions happening simultaneously. A crypto currency that increases in value as it becomes more useful, not because of people buying it for speculation with the sole purpose being selling it more expensive later. This is the corruption that has ruined BTC and it's so-called "BTC maximalist" community.

Value through usefulness, utility, scalability, and simplicity. This is the vision and purpose of UniChain. Authentic 'internet money' your grandma can use at the store to pay for groceries. To pay for a delivery, your rent, or someone's paycheck. No long waiting times, no high fees, no congestions, and last but not least no huge mining farms destroying our planet with the wasteful proof-of-work (PoW) concept.

Abstract

UniMe is a highly secure multi-function chat app with AI and Blockchain powered features. In addition to chat, voice, and video calls - wallets, payments, and chatbots can be made easily and without any coding knowledge. UniMe is one of various applications powered by the UniWorld ecosystem - enterprise grade AI and Blockchain technology built for corporate and everyday use.

UniMe can be used like any regular messenger app with two distinct differences. Namely very high security and additional functions:

1.) all data (chat messages, images, audio, and video) are encrypted on an end-to-end basis meaning that only the participants of the conversation are able to read and understand the contents,

2.) users can create chatbots with sophisticated AI systems behind them as well as manage their blockchain/crypto wallets, make transactions with them, and craft smart contracts and decentralized applications. The former is easily doable without any tech knowledge.

I. Introduction

Today, messenger applications are used increasingly more (2). Billions of Users send hundreds of billions of messages every day using their smartphones, PCs, laptops, and other smart devices.

Unfortunately, most chat applications are far from secure nor do they protect and respect their user’s data privacy. It is widely known that both companies and governmental bodies collect and misuse private data (3, 4).

In the most innocent scenario, this data is then analyzed and fed to machine learning algorithms to create customized advertisements and user experience. This per se would not be intrinsically bad would it not be for the leaks, breaches, hacks, and sell-offs of sensible data (5).

While there are secure alternatives to popular messaging apps, most are rather limited in their capabilities and features. UniWorld.io proposes UniMe. A state-of-the-art easy-to-use and highly secure way to chat, call, send money, and even build AI chatbots easily - no coding required.

In other words, UniMe users can send messages to their friends, family, or colleagues including audio and video calls without worrying about any sensitive information being leaked.

Using end-to-end encryption, only participants can read the contents. As mentioned, they can also create their own customized multi-purpose chatbots, manage any major crypto currency and craft smart contracts or explore Blockchain networks all within a single app and only the latter of which needs any technical knowledge.

For a better scale, UniMe follows a microservice architecture. Illustrated in detail in the picture below.

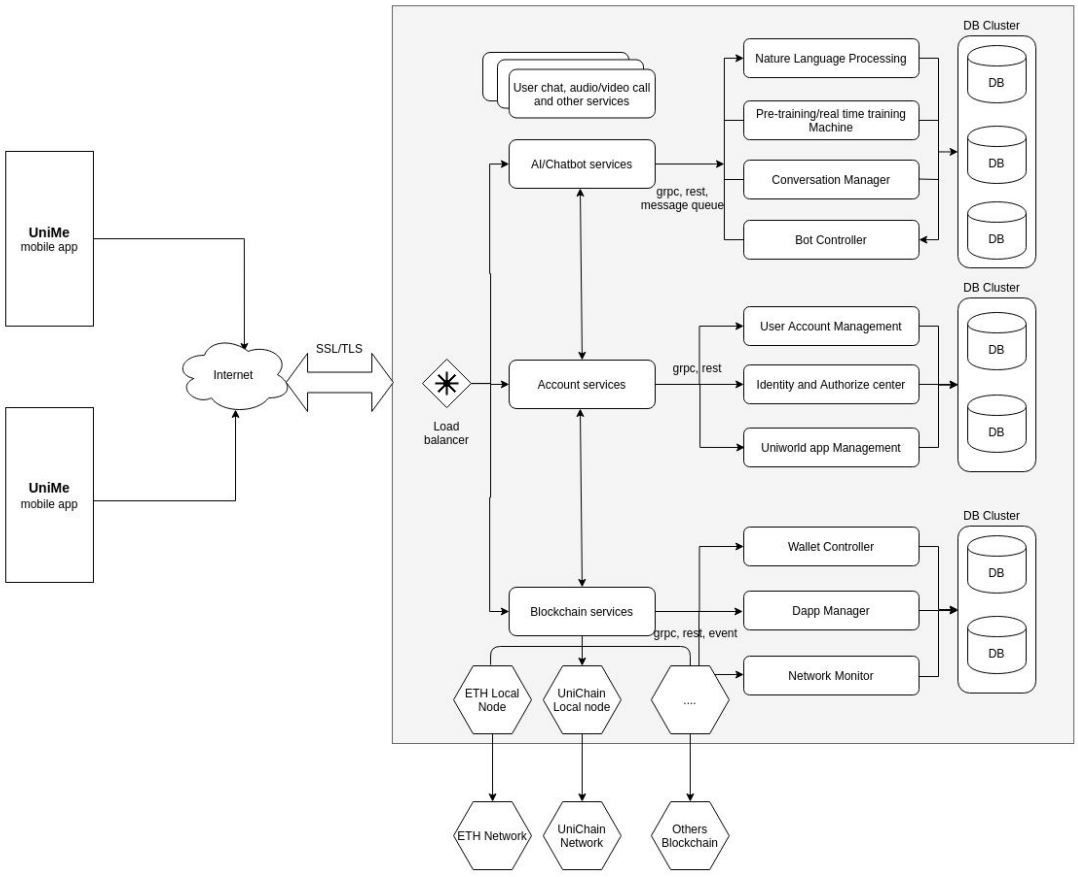


Fig. 1. UniMe application high level architecture illustrated.

II. Secured end-to-end chat app

UniMe's level of security stems from TextSecure protocols (also known as Signal protocols)[2]. The protocols can be roughly divided into three types and stages:

- the initial key exchange, or X3DH (extended triple Diffie-Hellman) protocol [3], which combines long-term, medium-term, and ephemeral Diffie-Hellman keys to produce a shared secret "root" value,
- an asymmetric ratchet stage, where users alternate in sending new ephemeral Diffie-Hellman keys in a ping-pong fashion with previously generated root keys to generate forward-secret chaining keys,
- a symmetric ratchet stage, where users take no additional entropy but instead use key derivation functions to ratchet forward chaining keys to create symmetric encryption keys.

Each message sent by a user is encrypted using a fresh message key to provide a high degree of forward secrecy. The ping-pong pattern of new ephemeral Diffie-Hellman keys injects additional entropy into the process, which is intended to continually achieve the utmost highest level of forward secrecy as well as post-compromise security.

The illustration below: the multi-stage authenticated key exchange protocol.

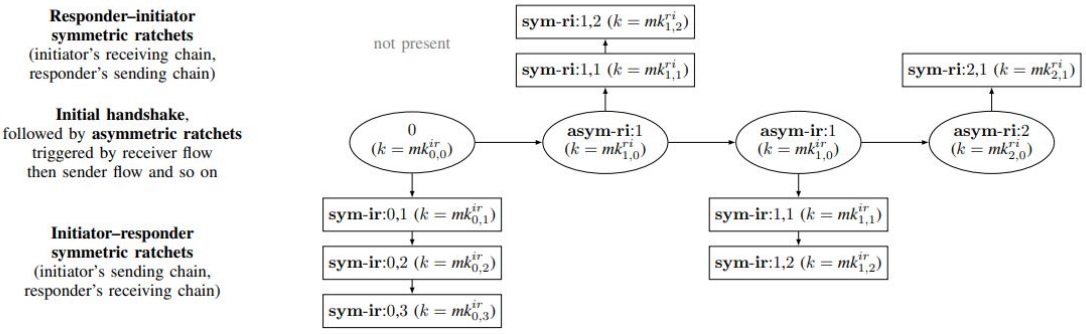


Fig. 2. An example of an execution. Multiple stages of trees are formed. The content of each node is the stage name and the session key is derived during the stage.

The multi-stage authenticated key exchange protocol consists of the following main steps:

- ***Registration***: During installation (and periodically afterwards), both Alice and Bob independently register their identity with a key distribution server and upload long-term, medium-term, and ephemeral public keys.
- ***Session setup***: Alice requests and receives a set of Bob's public keys from the key distribution server and uses them to set up a "long-lived" messaging session to establish initial symmetric encryption keys. This is called the TripleDH handshake or X3DH.
- ***Synchronous messaging:*** (a.k.a. asymmetric-ratchet updates). When Alice wants to send a message to Bob (or vice versa) and has just received a message from Bob, she exchanges Diffie–Hellman values with Bob. This generates new shared secrets and uses them to begin new chains of message keys. Each DH operation is a stage of the "asymmetric ratchet". They strictly occur in a ping-pong fashion.
- ***Asynchronous messaging:*** (a.k.a. symmetric-ratcheting). When Alice wants to send a message to Bob (or vice versa) but has yet to receive a message from Bob, she derives a new symmetric encryption key from her previous state using a PRF. Each PRF application is a stage of the "symmetric ratchet" in itself.

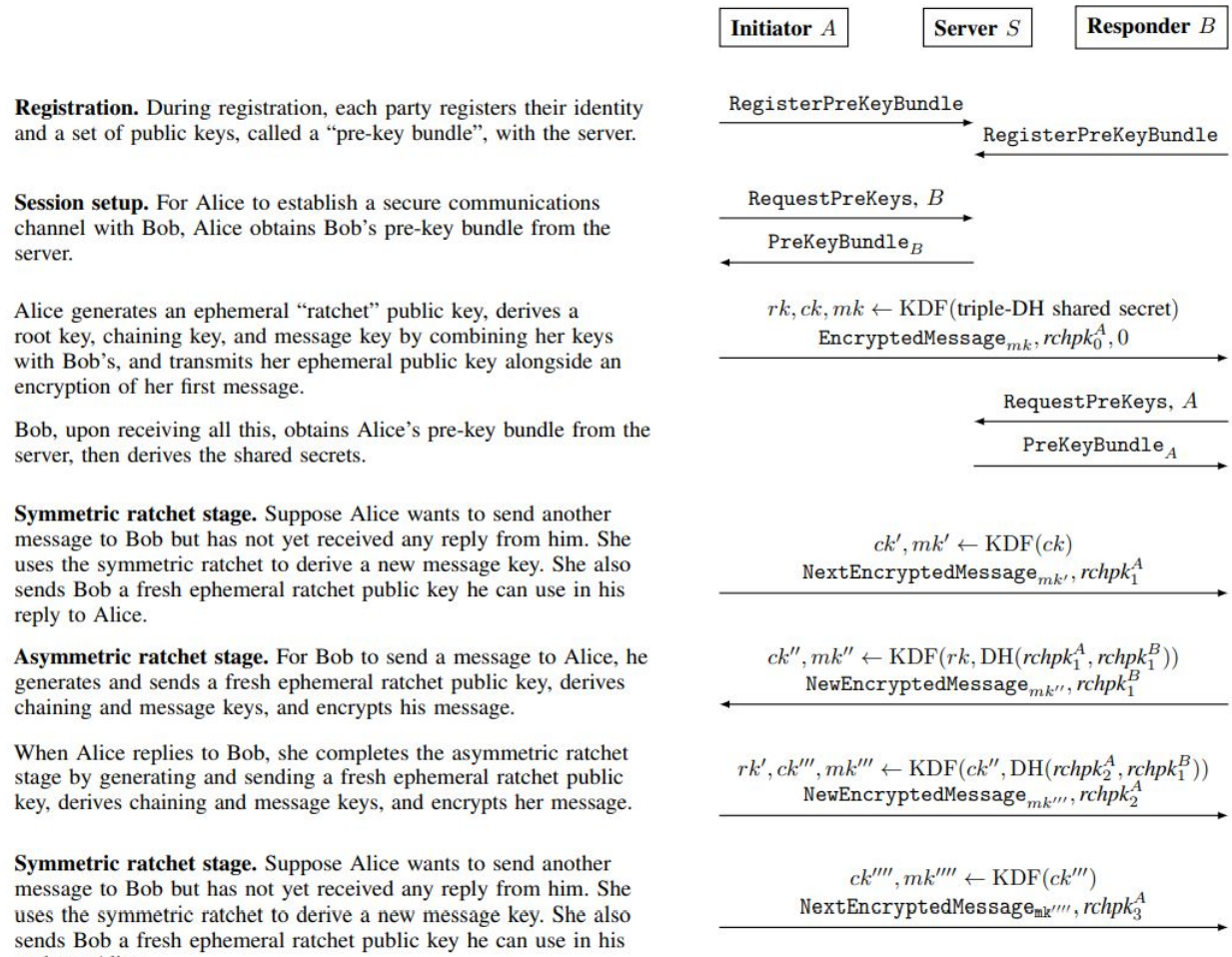


Fig. 3. Message flow of an example execution between two clients A and B via a server S. Notation and some operations have been simplified for clarity compared to later use.

III. AI and Chatbots

UniMe uses many Deep Learning algorithms such as Bi-directional LSTM, Machine Reading Comprehension, Transfer Learning, Sequence to Sequence Modeling with multi-headed attention mechanisms, and Generative Adversarial Networks to build conversational AI chatbots. Users can create and chat with their custom chatbots. The below picture illustrates the conceptual map of UniMe chatbots and how they use Deep Learning.

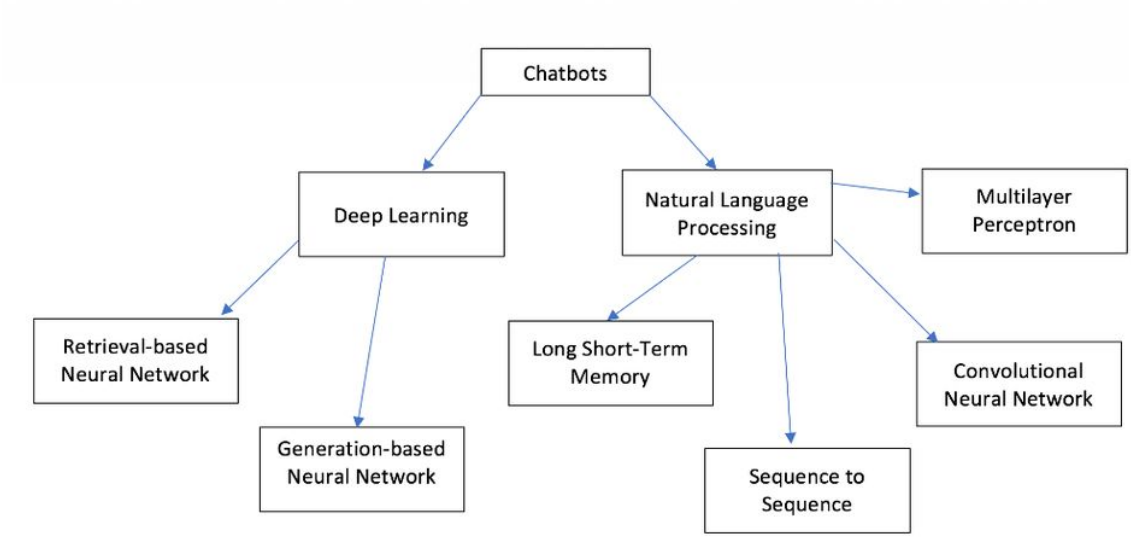


Fig. 4. Conceptual map of a UniMe chatbot using Deep learning.

The chatbot needs to be able to understand the intentions of the sender's message, determine what type of response message is appropriate - e.g a follow-up question, direct response, etc. - and follow correct grammatical, contextual and lexical rules when responding.

The chatbot's thought process can be divided into three main parts as illustrated below.

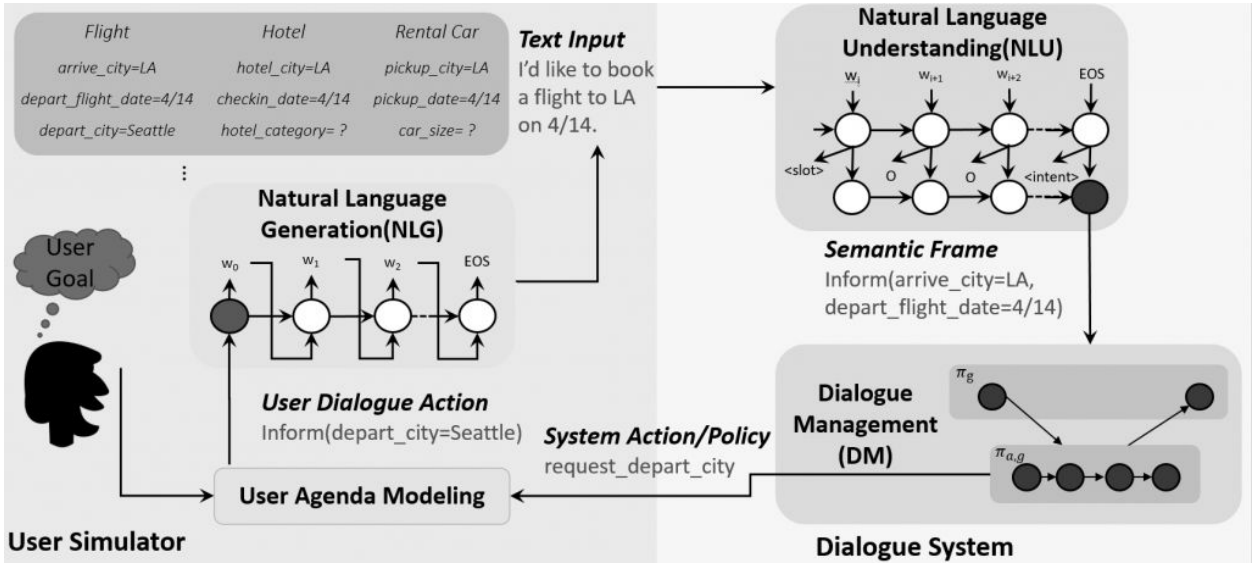


Fig. 5. Chatbot based on Deep Learning.

- ***Nature Language Understanding (NLU)***

The NLU unit is responsible for transforming the user utterance to a predefined semantic frame according to the system's conventions. In other words to a format understandable for the system.

This includes slot filling and intent detection. These tasks are seen as sequence tagging problems. For this reason, the NLU component implemented is LSTM-based [4]. The recurrent neural network uses a Conditional Random Field (CRF) layer on top of it. The model is sequence-to-sequence using a bidirectional LSTM network, which fills the slots and predicts the intent at the same time. The model is doing the same using an attention-based RNN [5]. See the illustration below.

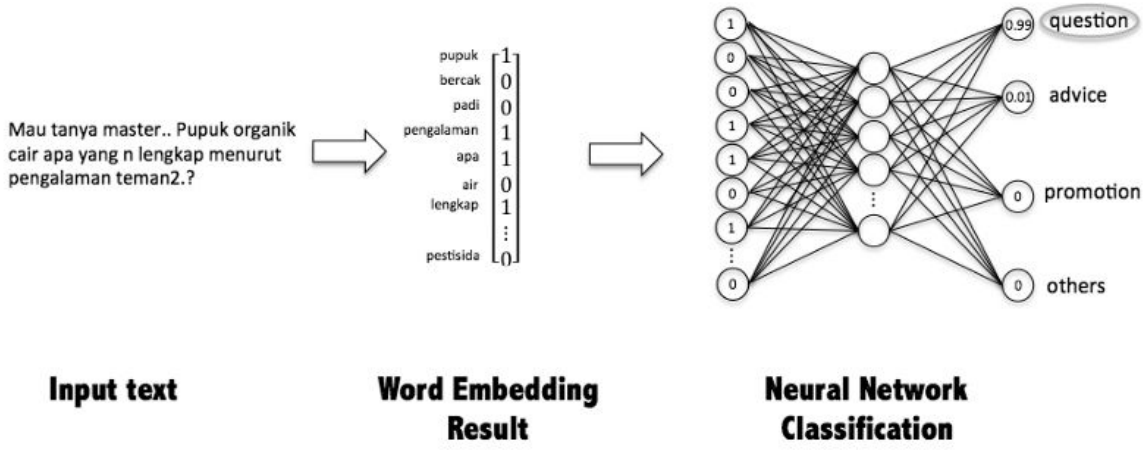


Fig. 6. Process for intent classification using Neural Network.

- ***Nature Language Generation (NLG)***

The NLG is the process of generating text from the meaning. Can be seen as the reverse of Natural Language Understanding. NLG systems provide a critical role for text summarization, machine translation, and dialog systems. In NLG, the system responds as a semantic frame. It maps back to a Natural Language Sentence (NLS) understandable for the end user. The NLG component can be rule-based or

model-based. In some scenarios it can be a hybrid model, i.e. a combination of both.

Trainable NLG systems can produce various utterances candidates — e.g. scholastically or rule base — and use a statistical model to rank them. On the other hand, NLG based semantically controlled Long Short-Term Memory (LSTM) recurrent networks can learn from unaligned data by jointly optimising sentence planning and surface realisation components. This happens using a simple cross entropy training criterion without heuristics.

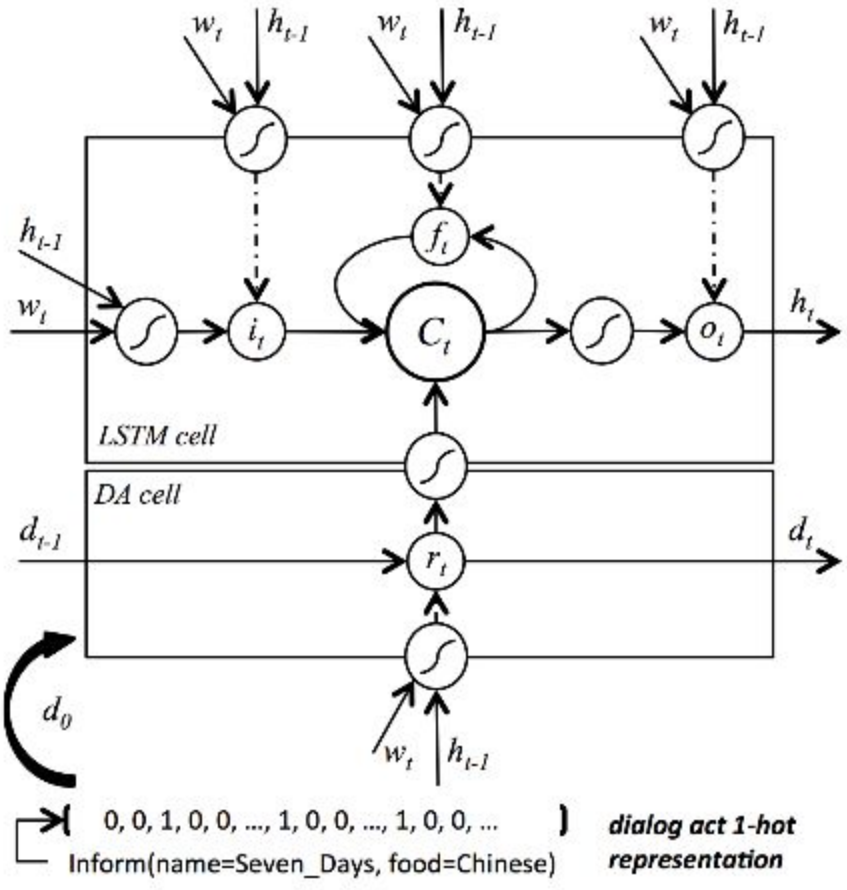


Fig. 7. Semantic controlled LSTM cell.

- ***Dialog Management (DM)***

DM can be connected to an external Knowledge Base (KB) or DataBase (DB) to produce more meaningful answers. The Dialogue Manager consists of the following two components: The Dialogue State Tracker (DST) and Policy Learning which

refers to the Reinforcement Learning (RL) agent. The Dialogue State Tracker is a complex and essential component that should correctly infer the belief about the state of the dialogue according to all history up to that point. Policy Learning is responsible for selecting the best action.

IV. Blockchain Hub

There are hundreds - and increasingly more - blockchain platforms and applications available on the market. Virtually each of them requires end-users to install at least one application in order to work or communicate with it. Blockchain is still a very new technology and often interchangeably used as a term to refer to Bitcoin (crypto currency) and vice-versa. Yet, Bitcoin is merely one of many use cases for the ledger technology that is Blockchain.

End-users need applications with more functionality beyond single crypto transactions. Mainstream adoption stems from a certain degree of utility. This could comprise an app that can manage any major crypto token/coin, deploy smart contracts, act as a market browser, and utilize/scan codes including QR to track on the Blockchain registered items. That's the purpose of UniMe and why it was born.

1. Master key

Each UniMe account has only one private (“master") key. It’s the root key of all blockchain wallets. The master key is generated randomly from the client side using the user’s data seed.

$$masterKey = random(timestamp, \ user's\ metadata, \ user's\ input\ string)$$

For convenience, the mnemonic phrases are also generated following the BIP39 [6] standards.

$$mnemonicPhrase = BIP39(seed \mid masterKey)$$

The master key is encrypted by the user's password. Only encrypted data is sent to the server. The master key can be decrypted exclusively with the user's password from the clients side. It’s the users responsibility to keep their password, master key, and recovery (“mnemonic”) phrases secure and safe.

Are both the key and phrase lost, the account and wallet cannot be restored and access to a wallet and all of the funds contained within is lost indefinitely.

UniMe has two types of wallets. Standalone - and HD wallets. For standalone wallets, the private key is the master key. For HD wallets, UniMe uses the derived key from the master key for each account and each type of wallet. HD wallets follow the BIP32[7] standards.

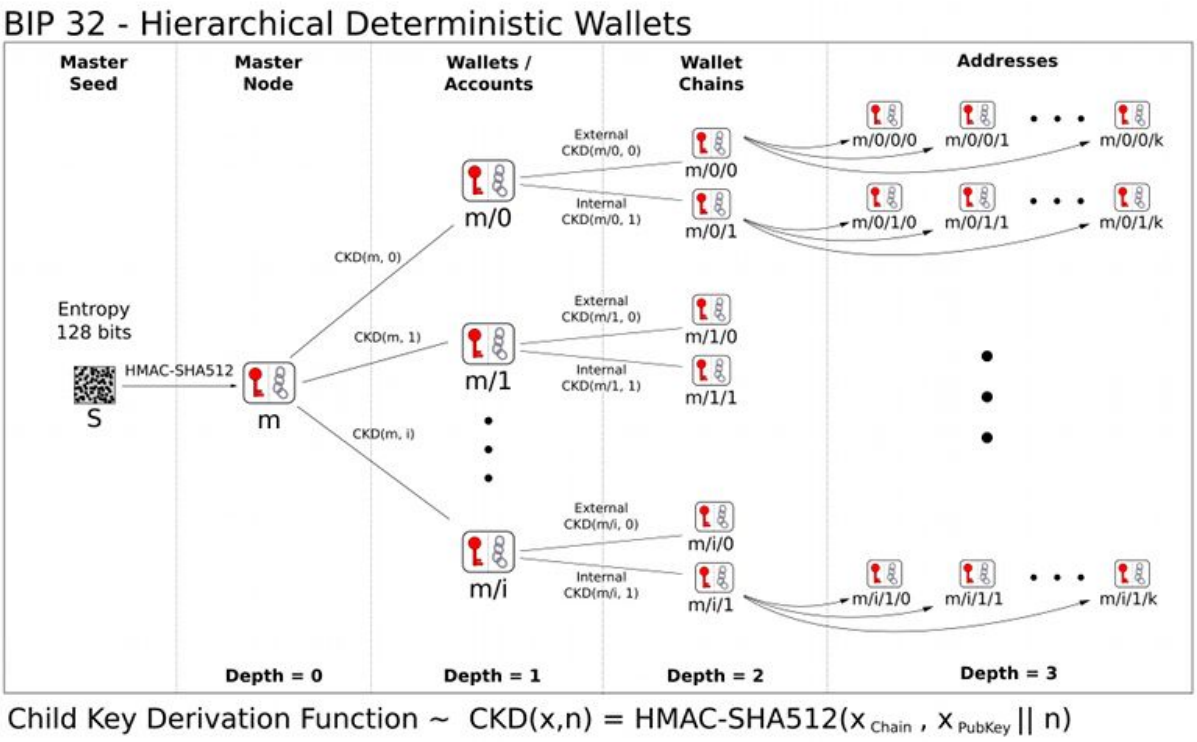


Fig. 8. Key derivation from BIP32.

2. *Generating wallet addresses*

Each blockchain platform has a different way of generating their wallet addresses. A key difference lies in cryptography functions ‘ei’. Hash functions and address are encoded but they are following the same principles:

Generate a public key from a private key -> generate an address from the public key -> encode address.

The example below described how a Bitcoin address is generated step by step:

```
(1)privateKey = generatePrivateKey() -> see master key step
(2)ellipticPublicKey = EDCSA(privateKey)
(3)publicKey = 0x04 + ellipticPublicKey
(4)encryptedPublicKey = RIPEMD-160(SHA-256(publicKey))
(5)mainnetEncryptedPublicKey = 0x00 + encryptedPublicKey
(6)checkSum =
   first4Bytes(SHA-256(SHA-256(mainnetEncryptedPublicKey)))
(7)Address = base58(mainnetEncryptedPublicKey + checkSum)
```

3. Signing transactions

When creating a transaction, UniMe needs information from the user such as the transaction type - e.g. coin transfer, smart contract execution, master/witness node votes, and so forth - the type of blockchain (Bitcoin, Etherium, UniChain, etc.), the corresponding information.

Transactions will be created using transaction builders. The result is the hex-string-unsigned encoded transaction. The final step before broadcasting a transaction on the blockchain is the signing procedure. To sign a transaction, users use their password to decrypt the private keys. Then, using the private keys, the cryptography signature function is performed.

After the transaction is signed, a signed hex string transaction is also returned. Now, transactions can be broadcasted onto the blockchain network. Note that all calculations happen on the client’s side meaning the server has no way of getting the user's private key.

4. Restoring wallets

When users forget their account password, they can restore wallets based with their private key or mnemonic phrase.

Steps to restore an account/wallet:

(1) Users clicks forgot password/restore account
(2) Fills in email address and checks email for verification
(3) Enters private key/master key or mnemonic phrases
(4) *Enters new password*

From the application side, the system will check and verify if the account restoration request is valid or not. It then compares if the address generated comes from the user's private key or mnemonic phrase or not. If it does, the account is recovered and the user encrypts their private keys using a new password. As mentioned above, all calculations are performed on the client's side.

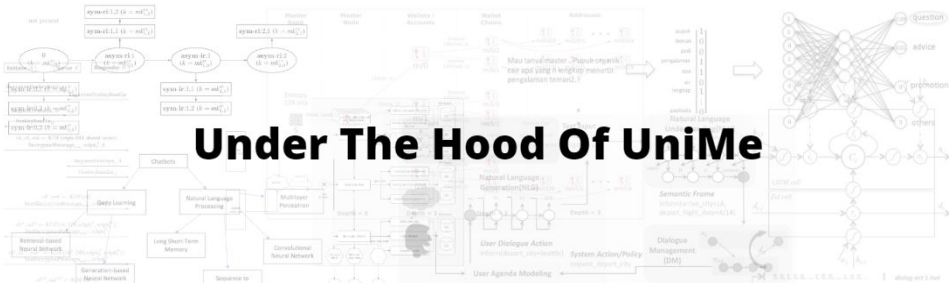
V. Summary - The Swiss Army Knife Chat App: UniMe

Consists of an easy-to-use and highly secure way to chat, call, send money, build bots, and deploy smart contracts - no coding required save the last part.

Another core product of UniWorld.io, this App utilizes the entire tool belt of the Uni ecosystem featuring:

- ZERO ads
- Highest privacy standards
- Quadruple secured chats, voice & video calls
- No interference from third parties whatsoever
- Blockchain based payment system (you can use all major token/coins
- Customizable chatbots that actually work and use AI to improve automatically.

Uni's interface is built customer friendly, but in the background the App is loaded with tech.



Comparison:

	Facebook Messenger	iMessage	Telegram	Whatsapp	Wire	UniMe
Provides transparency reports	✓	✓	✗	✓	✓	✓
Doesn't collect user data	✗	✗	✗	✗	✓	✓
Encryption by default	✗		✗	✓	✓	✓
Metadata is encrypted	✗	✗	✗	✗	✗	✓
Doesn't store timestamps and IP addresses	✗	✗	✗	✗	✗	✓
Connect to AI chatbot	✓	✗	✗	✗	✗	✓
Connect to blockchain network and wallet	✗	✗	✗	✗	✗	✓

Table 1. Compare UniMe and other popular chat apps.

VI. Conclusion

Chat apps have become an essential part of our everyday lives. Modern technology is built upon- and around communication and anybody with access to it, uses it. It is, in fact, essential to any developed society. There is, in fact, no ground for them not to stay a critical and central part of any future society concept including Society 5.0.

Disclaimer: UniMe is one of many applications powered by UniWorld.io to ð among other goals ð actualize Society 5.0 or The Smart Society 5.0.

References/Links

1. Uniword ecosystem (uniworld.io) including UniBot (unibot.org), UniLab (Coming Soon) UniGreen (Coming Soon), UniChain (UniChain.world)and Mia (mia.world)
2. Global Messenger Usage, Penetration and Statistics https://www.messengerpeople.com/global-messenger-usage-statistics/#:~:text=Facebook%20is%20the%20measure%20of,WeChat%20with%201.1%20billion%20users.
3. The Guardian on Edward Snowden https://www.theguardian.com/us-news/ng-interactive/2019/sep/13/edward-snowden-interview-whistleblowing-russia-ai-permanent-record
4. Decades of Facebook data scandals https://www.techrepublic.com/article/facebook-data-privacy-scandal-a-cheat-sheet/
5. Frederik Bussler’s “The Death of Privacy is An Abusive Government's Wet Dream” https://hackernoon.com/the-death-of-privacy-is-an-abusive-governments-wet-dream-675sj46n7
6. Katriel Cohn-Gordon , Cas Cremers† , Benjamin Dowling , Luke Garratt , Douglas Stebila - A Formal Security Analysis of the Signal Messaging Protocol. July 2019
7. Moxie Marlinspike Trevor Perrin. The X3DH Key Agreement Protocol. April, 2016

8. **Sepp Hochreiter**, Long Short-term Memory, December 1997
9. **Alex Sherstinsky**, Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network, August 2018
10. Bitcoin Improvement Protocol - 39. Github: https://github.com/bitcoin/bips
11. Bitcoin Improvement Protocol - 32. Github: https://github.com/bitcoin/bips