Demo: "DappTweet" A Solution to Use Twitter/X from MetaMask Wallet

Seyed Ahmadreza Abtahi[1], Reza Abtahi[1], Bruno Rodrigues[2], Burkhard Stiller[1]
[1]Communication Systems Group CSG, Department of Informatics IfI, University of Zurich UZH, CH–8050 Zurich, Switzerland
[abtahi, rabtahi, stiller]@ifi.uzh.ch
[2]Institute of Computer Science in Vorarlberg ICV, University of St.Gallen HSG, CH-9000 St. Gallen, Switzerland,
bruno.rodrigues@unisg.ch

***Abstract*— This demo paper presents DappTweet, a Web3 app that lets any blockchain address send posts and direct messages on Twitter/X via a MetaMask wallet using an address-proven posting workflow. A relay account publishes on the user's behalf and embeds the sender's address and the transaction hash for public verification. The prototype implements three flows: public post, private message, and verification.**

Keywords*— *DappTweet, Address-Proven, Blockchain, Twitter, X, MetaMask, Social media.

I. INTRODUCTION

Numerous decentralized applications and solutions exist to enable communication between various blockchain addresses, e.g., Coinbase Wallet Messaging [6] and Wallet-to-Wallet Blockscan [4]. However, these existing solutions still lack the functionality in which messages can be sent from blockchain addresses to social media platforms.

The newly developed Web application, DappTweet, which is presented in this paper, aims to fill this gap. It is a privacy-preserving option for crypto and Non-fungible Token (NFT) holders to reach out to social media audiences for sending public or private messages whenever required. Various use cases of such a solution/application are discussed in [8]. Sending social media posts from a blockchain address could reduce fraud. Moreover, it can be used for verifying real identities and thereby enhancing the reliability of public posts and direct messages (DM).

We term our approach address-proven posting, i.e., public-key–authenticated social messages that prove control of a blockchain address. By posting via a relay, whether publicly or privately (DM), rather than through the user's personal handle, our design mitigates the direct linkage between a user's social identity and their on-chain address, thereby reducing immediate deanonymization risk [1]. By "public-key–authenticated social messages" we mean posts or DMs whose authorship is verifiable from control of a blockchain address.

DappTweet leverages blockchain technology to publish verifiable messages through an intermediary account. This system enables "Web 3" pioneers to communicate with "Web 2" audiences. Within this prototype, DappTweet is primarily designed for an underlying Ethereum blockchain, MetaMask and Twitter/X (the most popular platforms in their respective categories). However, this application can be adapted to other blockchains, wallets, and social media platforms based on market demand and user requests.

The high-level architecture of the system as designed is depicted in Fig. 1.

Fig. 1. Overall Architecture of DappTweet

Related studies/solutions presented in [2] proposes a bidirectional binding between social media profile of a user on the one hand, and the blockchain address of the user on the other hand. This solution targets application fields similar to DappTweet. However it requires the "Web 3" users to create an account in the "Web 2" platforms (Twitter/X in this case). This is while DappTweet seeks to connect "Web 3" users to their "Web 2" audience without forcing them to create an account in the traditional social media platforms.

Reference [3] proposes a solution called "Connect2NFT". This solutions aims to prove true ownership of the NFT profile pictures in Twitter/X. DappTweet can achieve the same results. It just requires the NFT holder to publish a Tweet from the corresponding blockchain address and confirm their Twitter/X profile.

This demonstration paper is organized as follows. Section II provides relevant details on the workflow and the user interface of the DappTweet. Section III provides preview of the implementation, and Section IV provides a summary.

II. TOOL FEATURES

Three workflows are presented to explain DappTweet's functioning and features. Users can select a workflow based on their intentions, designed to be clear and straightforward without any distractions or complexities.

Fig. 2 illustrates the workflows of the DappTweet, tailored to the users' needs: publishing a public tweet (for blockchain address holders), sending a private message to a desired Twitter/X account (for blockchain address holders), and verifying a received message or a published tweet (for Twitter/X account holders). The intermediary account, a trusted DappTweet account on Twitter/X, is responsible for transmitting posts containing the desired content of the publisher or the sender of the private message.

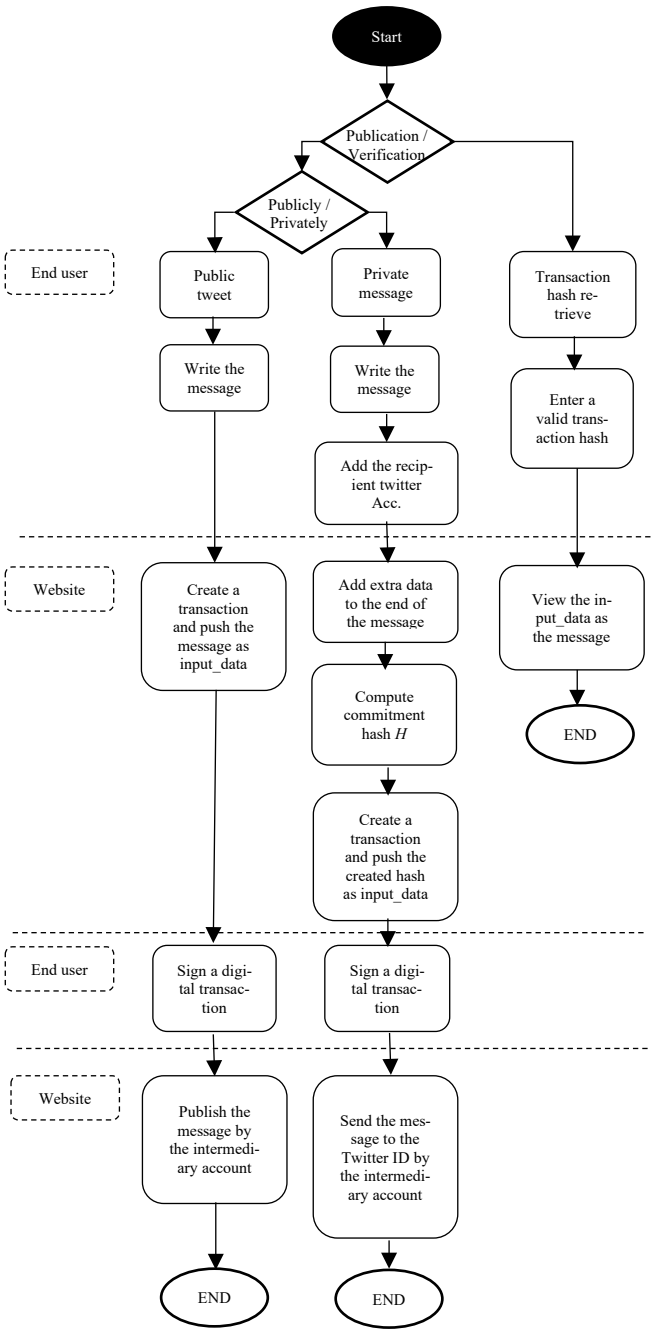

Fig. 2. Workflows of the DappTweet

This demonstration of DappTweet shows how a user can publish a tweet or send a DM from their blockchain address without using a traditional Twitter/X account. The interface visibly walks through each step: message input, MetaMask signature prompt, and final tweet/message publication by the intermediary account. A verification page is also presented, where users can paste a transaction hash and confirm message authenticity. This walkthrough emphasizes usability, privacy-preserving design, and seamless user experience. All interactions are performed on a public Ethereum testnet and an active Twitter/X account configured for the demo, while no personal credentials are required.

A preview of the proposed solution is provided in the following. Fig. 3 shows the homepage of the DappTweet. It simply provides two options for the users: “Publish Tweet” or “Send Direct Message”.

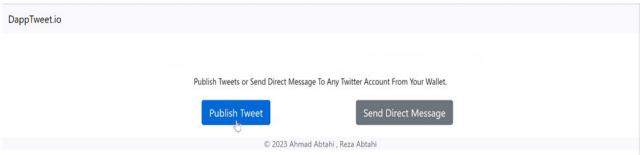

Fig. 3. DappTweet Homepage

Upon pressing the “Publish Tweet” button, the user is directed to the corresponding page as shown in Fig. 4. The user writes his intended message in the provided message box and presses the “Publish Tweet” button. This will create a transaction which contains the intended message in its *input_data* section. This section is an additional data included for the transactions and commonly used as part of contract interaction or as a message sent to the recipient [5]. This transaction is then passed to the user in a pop up message to be signed by them.

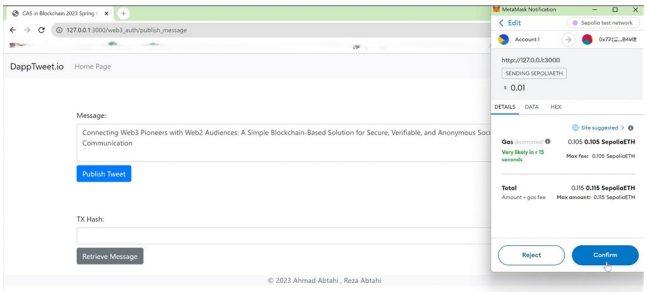

Fig. 4. Public Tweet Section

In the final step, upon submitting the signature, an intermediary Twitter/X account publishes a tweet. This tweet contains the sender's blockchain address, the content of the message box, and the corresponding transaction hash. This transaction hash enables any interested parties to independently verify the authenticity of the published tweet. An example of such a tweet is displayed in Fig. 5.

Fig. 5. Sample Tweet

In the second thread of the workflow, in a similar manner, users can send DMs to their desired Twitter/X accounts. The main difference is that in this case, a hashed version of the message is stored on the blockchain, rather than the full message. This design ensures that private messages remain confidential, visible only to the intended recipient, who can then verify the authenticity of the message.

To enable reproducible verification of private messages, DappTweet exposes a public, hash-based commitment scheme. Let X denote the sender's blockchain address and Y the DM payload, which is visible only to the recipient within the Twitter/X private messaging interface. We compute the following commitment:

$$H = \text{SHA-256}(X \parallel Y),$$

Here, $\parallel$ denotes byte-wise concatenation. The value H is recorded in the Ethereum transaction's input_data field, forming a permanent on-chain commitment to the message. This design yields three practical properties for the solution: (1) DappTweet publishes only the commitment H on-chain and does not disclose the DM plaintext; (2) the sender can rely on the fact that the DM content is not made publicly available by the protocol; and (3) the recipient (or any third party with access to Y) can verify authorship by recomputing H, matching it against the on-chain value, either manually or via the "Transaction Details" view on the DappTweet website.

Fig. 6 shows the webpage where the blockchain address holder can add the intended message and the recipient's Twitter/X ID.

Fig. 6. Private Tweet Section

According to the last thread of the workflows, Twitter/X users can also retrieve their desired input data (plain public Tweet or the H which is related to private message section) by submitting the received transaction hash, as illustrated at the bottom of Fig. 4. This retrieval can also be performed directly by using any other blockchain explorer as well. This feature is critical in a trustless solution. An example of the retrieved transaction details is shown in Fig. 7.

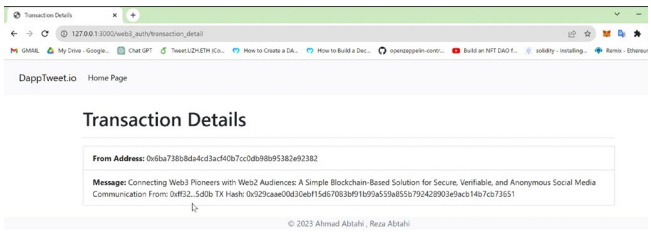

Fig. 7. Transaction Details Section

III. Tool Maturity

The backend and frontend of the proposed web application are developed using Django [9], a high-level Python framework that manages the interface logic and coordinates interactions with the Ethereum blockchain, the Twitter/X API, and the MetaMask wallet. Blockchain operations are handled using Web3.py [10], a Python library for Ethereum-based transactions. Twitter integration is implemented via the Tweepy library [11], which enables authenticated interactions with Twitter/X services.

To provide initial performance insights, we report key implementation-aligned metrics. For transaction cost, data-carrying transactions incur minimal on-chain computation. Therefore, gas usage is primarily determined by the 21,000 intrinsic cost and the calldata size. For a representative payload of 111 UTF-8 bytes (the average length of our test set message), the estimated gas usage is approximately 23,188. Latency is measured from the moment of wallet-signature confirmation to the visibility of the post on Twitter/X, including blockchain confirmation and API propagation. The observed median latency is approximately 14.2 seconds. Due to the constraints of the free-tier Twitter/X API (e.g., monthly post limits), the number of executions was restricted. Future work will extend the evaluation with additional metrics, including a detailed success/failure taxonomy, rate-limit recovery behavior, and gas cost variations across payload sizes and platforms.

IV. Summary

This paper presented a demo of DappTweet, a novel developed web application. DappTweet offers three distinct functions through a straightforward process flow:

1. Publishing a public tweet from the MetaMask Wallet
2. Sending a DM to a desired Twitter/X account from a MetaMask Wallet
3. Verifying a received message or a published tweet on Twitter/X.

A video demo of the UI is available in [7]. Thus, this demonstration shows that posting to Twitter/X from a MetaMask wallet is technically feasible, implementable with standard technologies, and consistent with address-proven posting, giving individuals control over what they disclose to prove ownership of their blockchain address.

References

[1] Chen, S., & Norman, S. M. U. (2022). Social networks are divulging your identity behind crypto addresses. *arXiv preprint arXiv:2211.09656*.

[2] Liu, Y., Lu, Q., Paik, H. Y., & Xu, X. (2020, July). Design Patterns for Blockchain-Based Self-Sovereign Identity. In *Proceedings of the European Conference on Pattern Languages of Programs 2020* (pp. 1-14).

[3] Bellagarda, J., & Abu-Mahfouz, A. M. (2022). Connect2NFT: A Web-Based, Blockchain Enabled NFT Application with the Aim of Reducing Fraud and Ensuring Authenticated Social, Non-Human Verified Digital Identity. *Mathematics*, *10*(21), 3934.

[4] Blockscan. (2023). Blockscan Chat - Wallet to Wallet Messaging for Web3. Retrieved May 21, 2025, from https://chat.blockscan.com

[5] Etherscan. (n.d.). Retrieved May 15, 2025, from https://etherscan.io/

[6] Coinbase. (n.d.). Coinbase Wallet messaging. Retrieved May 1, 2024, from https://help.coinbase.com/en/wallet/messaging/info

[7] DappTweet. (2024). DappTweet Application Demo [Video]. Retrieved October 25, 2025, from https://youtu.be/G_1sHkWaBZU

[8] Abtahi, A., Abtahi, R., Rodrigues, B., & Stiller, B. (2023). Sending Social Media Posts from Crypto Wallets: A Simple Blockchain-Based Solution for Connecting Web 3 Pioneers to Web 2 Audience. In *ChainScience 2023, Conference Proceedings*. Boston, MA. arXiv. https://arxiv.org/html/2307.03277v2/

[9] Django Software Foundation. (2023). *Django* (Version 4.2.23) [Computer software]. https://www.djangoproject.com/

[10] Ethereum Foundation Python Team (Snake Charmers). (2023). *Web3.py* (Version 6.11.2) [Computer software]. https://web3py.readthedocs.io

[11] Tweepy contributors. (2023). *Tweepy* (Version 4.14.0) [Computer software]. https://docs.tweepy.org