КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (31), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.31.1027
УДК 004.7+621.39

Міхав Володимир Володимирович
доктор філософії з комп'ютерної інженерії,
викладач кафедри інформаційних технологій
Приватна установа «Університет науки, підприємництва та технологій», м. Київ, Україна
ORCID: 0000-0003-4816-4680
mihaw.wolodymyr@gmail.com

Мелешко Єлизавета Владиславівна
доктор технічних наук, професор,
доцент кафедри кібербезпеки та програмного забезпечення
Центральноукраїнський національний технічний університет, м. Кропивницький, Україна
ORCID: 0000-0001-8791-0063
elismeleshko@gmail.com

Якименко Микола Сергійович
кандидат фізико-математичних наук, доцент,
завідувач кафедри вищої математики та фізики
Центральноукраїнський національний технічний університет, м. Кропивницький, Україна
ORCID: 0000-0003-3290-6088
m.yakymenko@gmail.com

Босько Віктор Васильович
кандидат технічних наук, доцент,
доцент кафедри автоматизації виробничих процесів
Центральноукраїнський національний технічний університет, м. Кропивницький, Україна
ORCID: 0000-0002-4933-9676
Victorvv2@ukr.net

Лисенко Ірина Анатоліївна
кандидат технічних наук,
старший викладач кафедри кібербезпеки та програмного забезпечення
Центральноукраїнський національний технічний університет, м. Кропивницький, Україна
ORCID: 0000-0003-4394-4960
min_max@i.ua

МЕТОДИ ВЕБ-РОЗРОБКИ ТА ТЕСТУВАННЯ ТОКЕНІЗАЦІЇ ДАНИХ ТА ОБ'ЄКТІВ ДЛЯ РЕАЛІЗАЦІЇ ДОВІРЧИХ СОЦІАЛЬНИХ СЕРВІСІВ У РОЗПОДІЛЕНИХ КОМП'ЮТЕРНИХ МЕРЕЖАХ

Анотація. У статті проведено комплексне дослідження методів та інструментів веб-розробки для реалізації токенізації даних та об'єктів у середовищі Web 3.0 з акцентом на бекенд-рівень. Проаналізовано сучасні стандарти смарт-контрактів ERC-20, ERC-721/1155, ERC-1400/3643, SPL, FA2, Cadence, підходи до зберігання метаданих on-chain, IPFS, Arweave, індексаційні механізми The Graph та архітектури масштабування L1/L2, zk-rollup-рішення, альтернативні L1. Сформовано класифікацію типів токенів за призначенням і методів їх реалізації, а також класифікацію веб-інструментів за функціональними рівнями, що дозволило виявити особливості та відмінності у вимогах до продуктивності, інтероперабельності, комплаєнсу та інформаційної безпеки. Порівняльний аналіз показав, що реалізації на основі EVM-стандартів є найбільш усталеними та оптимальними для утилітарних токенів, тоді як NFT-системи мають підвищені вимоги до безпечного позаланцюгового зберігання, а security- та RWA-токени забезпечують регульованість і контроль доступу за рахунок зниження ступеня децентралізації. Показано на основі огляду проектів, що технології другого рівня, зокрема Polygon zkEVM та Immutable X, забезпечують зменшення витрат газу та підвищення пропускної здатності транзакцій у

© В.В. Міхав, Є.В. Мелешко, М.С. Якименко, В.В. Босько, І.А. Лисенко, 2025

порівнянні з базовими мережами. Також у роботі запропоновано уніфіковану формальну методику тестування процесів токенізації, яка охоплює перевірку коректності операцій, інваріантів стану, політик доступу, криптографічних гарантій та ефективності виконання транзакцій. Методика інтегрує формальні моделі поведінки токенів із критеріями gas-ефективності реалізації та дозволяє здійснювати комплексне оцінювання надійності токенізаційних систем незалежно від стандарту або технологічного стеку. Результати роботи можуть бути використані для проєктування бекенд-архітектур Web 3.0, розроблення довірчих соціальних сервісів у розподілених комп'ютерних мережах, а також для створення методологічної основи автоматизованого тестування токенів у мультичейн-екосистемах.

Ключові слова: токенізація; комп'ютерні мережі; розподілені мережі; веб-інструменти; метрики довіри; соціальні сервіси; інформаційна безпека; цифрові активи.

ВСТУП

Стрімке розгортання екосистеми Web 3.0 формує нові правила створення й експлуатації веб-сервісів, у яких бекенд перестає бути лише серверною частиною застосунку й перетворюється на вузол взаємодії з децентралізованою інфраструктурою. Якщо у Web 2.0 основним ресурсом були дані та централізовані платформи, то у Web 3.0 ключовою цінністю стають цифрові права й активи, що існують у публічних мережах і керуються смарт-контрактами. Відповідно, перед розробниками бекенду постає потреба не тільки інтегрувати блокчейн-компоненти, а й проєктувати сервіси так, щоб вони коректно працювали з токенізованими об'єктами в різних доменах.

Web 3.0 як нове покоління Інтернету ґрунтується на децентралізації, криптографічній довірі та механізмах консенсусу, що дозволяють зберігати стан і виконувати логіку без єдиного центру контролю. Однією з фундаментальних можливостей цієї парадигми є токенізація – процес цифрового представлення цінностей (активів, прав, ресурсів) у вигляді токенів у блокчейні. Токен у такій системі виступає універсальним контейнером значення: він може означати валюту, частку власності, ліцензійне право, членство спільноти, доступ до сервісу, цифрову ідентичність або відповідність фізичному об'єкту. Токенізація надає активам програмованості – можливості автоматизувати правила користування, обміну, нарахування винагород, контролю доступу та виконання угод без посередників.

Практичне значення токенізації виходить далеко за межі криптовалют. Сьогодні вона використовується у фінансових системах (DeFi-протоколи, стейблкоїни, деривативи), індустрії цифрових прав і контенту (NFT, ліцензування, роялті), сфері реальних активів (RWA – токенізація нерухомості, цінних паперів, товарів), логістиці та виробництві (простежуваність походження), а також у цифровій ідентифікації (DID-моделі, verifiable credentials). Для бекенду це означає необхідність підтримувати складні сценарії життєвого циклу токена, зокрема, випуск, верифікацію, зберігання метаданих, передачу між користувачами, взаємодію з зовнішніми протоколами й містками, а в окремих випадках – комплаєнс-обмеження або інтеграцію з традиційними реєстрами.

Водночас Web 3.0 – це не одна технологія, а ціла сукупність мереж, стандартів і інструментів, які еволюціонують надзвичайно швидко. Розробник бекенду має обирати між різними типами блокчейнів (EVM-сумісні та альтернативні L1/L2, модульні архітектури, zk-ролапи), стратегіями зберігання метаданих (повністю on-chain, децентралізовані сховища на кшталт IPFS/Arweave, гібридні моделі), способами доступу до даних (індексатори, субграфи, власні ноди), механізмами керування

ключами (кастодіальні/некостодіальні гаманці, MPC/TSS, KMS) і практиками забезпечення якості смартконтрактів (формальна верифікація, статичний аналіз, fuzzing, runtime-моніторинг). Кожен із цих виборів впливає на безпеку, продуктивність, масштабованість і економіку проєкту.

Отже, токенізація активів у Web 3.0 виступає одночасно технологічним драйвером і методично складною інженерною задачею. Її успішна реалізація потребує науково обґрунтованих підходів до проєктування бекенду, щоб цифрові активи були керованими, безпечними, сумісними з різними мережами та здатними підтримувати реальні бізнес-сценарії. Саме тому актуальним є системне дослідження інструментів і архітектур токенізації, з фокусом на бекенд-рівень і відтворювані критерії оцінювання рішень.

Постановка проблеми. Незважаючи на наявність стандартів токенів і розвинену інфраструктуру Web 3.0, відсутня коротка, узгоджена та відтворювана методологія вибору бекенд-інструментів і архітектур для різних сценаріїв токенізації. Гетерогенність блокчейн-стеків, моделей метаданих та засобів розробки унеможливлює порівняння рішень за єдиними критеріями, що підвищує ризики вразливостей, ускладнює аудит і збільшує вартість життєвого циклу систем. Потрібні системна класифікація методів токенізації на бекенд-рівні та методика їх оцінювання за метриками безпеки, масштабованості, відповідності вимогам і сукупної вартості володіння.

Аналіз останніх досліджень і публікацій. У сучасних дослідженнях токенізацію на рівні бекенду визначають як процес формалізованого перетворення прав власності або доступу на цифрові представлення у вигляді смарт-контрактів [1-3]. У середовищі Ethereum Virtual Machine (EVM) основними стандартами інтерфейсів є ERC-20 для взаємозамінних токенів [1], ERC-721 – для невзаємозамінних [2] та ERC-1155 – для комбінованих сценаріїв, що забезпечують «напівзамінність» і пакетну передачу активів [3]. В існуючих роботах підкреслюється, що дотримання цих інтерфейсів гарантує інтероперабельність між гаманцями, біржами та бекенд-сервісами, а також зменшує ризик помилок реалізації завдяки стандартизованим подіям і методам [4-6].

Регульовані токени – security-токени та RWA (real-world assets) – потребують додаткових механізмів комплаєнсу. Для таких випадків розроблено стандарти ERC-1400/1410, які підтримують ролі операторів, часткову замінність і формалізовану логіку випуску та викупу [7], та ERC-3643 (T-REX), який інтегрує механізм on-chain ідентифікації через ONCHAINID [8-9]. Аналітичні звіти спільнот Polymath та Tokeny демонструють, що ERC-3643 дозволяє програмно імплементувати перевірки KYC/AML безпосередньо на рівні контракту, формуючи новий клас «permissioned» токенів [10].

Безпека токенів суттєво залежить від способу зберігання метаданих і стійкості каналів доступу. Звіт NIST IR 8472 від 2024 року систематизує 27 типів ризиків, що охоплюють вразливості контрактів, підміну метаданих, залежність від централізованих офчейн-сховищ та атак через неперевірені шлюзи [11]. У сучасній практиці реалізації безпечної токенізації переважає використання IPFS (структура Merkle-DAG, ідентифікація CID) [12] та Arweave (модель Blockweave, механізм Proof-of-Access) [13], які забезпечують децентралізоване зберігання контенту з криптографічною верифікацією.

Для агрегування подій та побудови API-доступу до станів блокчейну застосовують індексаційний протокол The Graph, який формує децентралізований ринок даних Web 3.0 і підтримує мову запитів GraphQL [14]. Дослідження з масштабування підтверджують, що оптимізація використання обчислювальних

ресурсів мережі, так званих «витрат газу» – плати за виконання обчислень у блокчейні, та підвищення пропускної здатності досягається шляхом винесення транзакцій типу mint (створення токенів) і transfer (передача токенів) у L2-рішення (додаткові рівні обчислень, які працюють поверх основного блокчейна). Так, у проєкті Polygon 2.0 реалізовано мережу ZK-ланцюгів із уніфікованою ліквідністю та сумісністю з EVM [15], а платформа Immutable X на основі STARK-доказів забезпечує пакетизацію NFT-транзакцій і скорочення часу фіналізації [16].

Альтернативні архітектури блокчейнів також розробили власні токен-примітиви: SPL/Token-2022 для Solana [17], FA2 для Tezos [18] і стандартні контракти мови Cadence у Flow [19]. Ці рішення розширюють інтероперабельність Web 3.0 екосистеми поза межами Ethereum.

Типовий стек бекенд-токенізації включає бібліотеки OpenZeppelin Contracts (аудитовані реалізації стандартів ERC, модулі Access Control, Ownable, Upgrades) [20], середовище розробки Hardhat 3 (тестування, трасування, емуляція вузлів) [21], а також бібліотеку ethers.js v6 для взаємодії з RPC-провайдерами й підписів транзакцій [22]. Для локального моделювання часто використовують Truffle або Ganache [23].

Хмарні вузли та провайдери, такі як Infura, Alchemy та QuickNode, забезпечують стійкий доступ до mainnet/testnet, а також шлюзи до IPFS і систем аналітики, що дозволяє реалізувати повноцінний CI/CD-конвеєр для бекенд-токенізації [20-23].

Безпека смарт-контрактів розглядається як багаторівнева система, що поєднує формальні інваріанти стану, криптографічні протоколи та інструменти аналізу коду. Наукові праці та звіти дослідницьких груп підкреслюють, що більшість критичних помилок у токенах пов’язані з некоректним керуванням пам’яттю (re-entrancy, integer overflow тощо) і порушенням інваріантів суми балансів [24-26].

Дослідження у сфері фінансових токенів і стейблкоїнів показують компроміс між децентралізацією та відповідністю нормативним вимогам, зокрема, у моделі DAI використовується децентралізована система заставних CDP [27], тоді як централізовані стейблкоїни типу USDC мають можливість блокування адрес і реверс-операцій [28]. Міжнародні регулятори (BIS, FSB) відзначають потенціал токенізації для атомарних розрахунків та підвищення прозорості фінансової інфраструктури, одночасно наголошуючи на операційних ризиках і правовій невизначеності [29-30].

У Європейському Союзі регламент MiCA (Markets in Crypto-Assets, 2024-2025 pp.) задає єдину рамку для емітентів і провайдерів послуг щодо crypto-assets, що створює нормативну основу для розвитку токенізації реальних активів (RWA) та security-токенів [31].

Попри значний прогрес у стандартизації токенів і розвитку інструментів їхньої реалізації, питання верифікації, тестування та безпеки токенізаційних систем залишаються фрагментарно дослідженими. У більшості робіт увагу зосереджено на технічних аспектах імплементації стандартів (ERC, FA, SPL) та на окремих кейсах аудиту смарт-контрактів, тоді як узгодженої методології комплексного тестування бекенд-токенізації досі не сформовано. Відсутня також уніфікована система метрик, яка б дозволяла порівнювати моделі токенів за рівнем безпеки, ефективності, витратами газу та відповідністю інваріантам стану. З огляду на це, актуальною є розробка методики формалізованого тестування процесів токенізації у Web 3.0 – з урахуванням криптографічних інваріантів, процедур статичного та динамічного аналізу, вимірювання обчислювальної складності (gas efficiency) і верифікації політик доступу. Запропонована у подальших розділах статті методологія спрямована саме на усунення цієї прогалини – формування системного підходу до оцінювання надійності та якості

реалізацій токенів у децентралізованих бекенд-середовищах.

Мета статті. Метою статті є систематизація сучасних методів та інструментів веб-розробки, що використовуються для реалізації токенізації у середовищі Web 3.0, із акцентом на бекенд-рівень – стандарти смарт-контрактів, архітектури зберігання й масштабування (L1/L2), засоби безпеки та комплаєнсу, а також інфраструктурні рішення для взаємодії з децентралізованими мережами. У роботі передбачається здійснити порівняльний аналіз технологічних підходів до створення, управління та інтеграції різних типів токенів (фунгібельних, невзаємозамінних, гібридних і токенів цінних паперів), виявити їх переваги та обмеження з погляду продуктивності, безпеки, інтероперабельності та відповідності нормативним вимогам, а також сформулювати узагальнені рекомендації щодо вибору оптимального інструментарію для побудови бекенд-систем токенізації у Web 3.0, а також забезпечення їх безпеки та тестування. Також передбачається запропонувати методологію оцінювання надійності та якості реалізацій токенів у децентралізованих бекенд-середовищах.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Токенізація є процесом перетворення права власності, доступу чи цінності на цифровий еквівалент (токен), який існує у децентралізованому середовищі (блокчейні). Кожен токен має унікальну логіку створення, обігу та знищення, визначену смарт-контрактом. На бекенд-рівні токенізація реалізується як послідовність атомарних транзакцій, що змінюють стан мережі відповідно до певних інваріантів (балансів, дозволів, політик доступу). Узагальнено, будь-який токен можна розглядати як абстрактний об’єкт із такими компонентами:

$$T = \langle ID, Type, Meta, Owner, State, Policy \rangle, \quad (1)$$

де ID – унікальний ідентифікатор токена, $Type$ – тип токена (FT, NFT, SFT, Security, RWA, SBT, LP), $Meta$ – набір метаданих (опис, URI, посилання на IPFS тощо), $Owner$ – адреса або набір адрес, що мають право володіння, $State$ – баланс, стан забезпечення, дозволи, кворум тощо, $Policy$ – набір умов (комплаєнс, KYC, часові обмеження, заморозка).

Будь-яка операція з токеном є транзакцією:

$$f: S \to S', \quad (2)$$

яка переводить систему зі стану S у новий стан S', зберігаючи визначені інваріанти:

$$\mathcal{I}(S) = \mathcal{I}(S') = \text{істинно}. \quad (3)$$

Токенізація об’єктів та даних сьогодні застосовується переважно для формалізованого представлення цифрових і фізичних активів у розподілених обчислювальних середовищах. У сфері фінансів вона забезпечує програмованість прав власності, автоматизацію обліку та можливість атомарних розрахунків. Для цифрового контенту токенізація використовується як механізм підтвердження автентичності та відстежуваності метаданих, що є критичним для авторського права, ліцензування та

соціальних платформ. У корпоративних інфраструктурах токенізація дозволяє керувати доступом, ідентичністю та репутаційними параметрами у децентралізованих сервісах. З огляду на зростання масштабів використання таких систем, ключовим завданням стає вибір технологій, які забезпечують коректність, безпеку та інтероперабельність токенів у Web 3.0. У табл. 1 наведено проведене порівняння основних технології токенізації.

Таблиця 1

Порівняльна таблиця основних технологій токенізації

№	Технологія	Тип токенів	Основна функція	Переваги	Обмеження та ризики
1.	ERC-20	Фунгібельні (FT, utility)	Стандарт передачі цінності	Простота, підтримка всіма гаманцями	Вразливість до re-entrancy, відсутність on-chain комплаєнсу
2.	ERC-721 / 1155	Невзаємозамінні (NFT) / мульти-токени	Унікальні активи, колекції, ігри	Універсальність, підтримка ринків OpenSea, Blur	Залежність від off-chain даних, дорогі операції
3.	ERC-1400 / 3643	Security / RWA-токени	Вбудований KYC/AML, регульовані операції	Відповідність закону, контроль доступу	Зниження децентралізації, складність аудиту
4.	IPFS / Arweave	Метадані, файли NFT	Децентралізоване зберігання контенту	Незмінність, доступність з будь-якої мережі	Нестабільність пінінгу, ризик втрати CID
5.	The Graph	Індексація on-chain даних	API для запитів станів і подій	Прискорення бекенду, GraphQL підтримка	Витрати на індексаторів, затримки синхронізації
6.	Polygon / zkEVM	L2 масштабування FT/NFT	Зниження витрат газу, EVM-сумісність	Висока швидкість, єдність ліквідності	Додаткова складність бриджів
7.	Immutable X / StarkNet	NFT, геймінг	ZK-rollup масштабування	Нульовий газ для користувача, миттєві операції	Обмежена інтероперабельність поза EVM
8.	Solana / Flow / Tezos	Альтернативні L1	Власні стандарти FT/NFT	Висока TPS, низькі комісії	Інша мова розробки, менша сумісність
9.	OpenZeppelin / Hardhat / ethers.js	Інструменти бекенду	Розробка та аудит контрактів	Широка спільнота, аудит, CI/CD	Потребує оновлення через зміни EVM
10.	Security Audit / Formal Verification	Усі типи токенів	Перевірка логіки контрактів	Зниження ризику експлойтів	Висока вартість та тривалість аналізу

Порівняння технологій токенізації засвідчує, що рішення для utility-токенів (EVM-стандарти, OpenZeppelin) є найзрілішими та оптимальними за співвідношенням «функціональність/ризики». NFT-системи досягають найбільшої гнучкості завдяки зовнішнім сховищам та індексації, але потребують більшого захисту метаданих. Security-токени та RWA-платформи забезпечують регульованість і прозорість, однак зменшують ступінь децентралізації через впровадження permissioned-механізмів. L2-рішення (Polygon, Immutable) знижують комісії до двох порядків порівняно з Ethereum mainnet, зберігаючи сумісність із існуючим інструментарієм бекенду.

Розглянемо класифікацію токенів за призначенням та їх методи реалізації у табл.

2.

Таблиця 2

Класифікація токенів за призначенням та їх методи реалізації

№	Тип токена / призначення	Приклади використання	Основний стандарт / модель	Метод реалізації на бекенді	Особливості безпеки та комплаєнсу
1	Фунгібельні (Utility / Payment Tokens)	Платежі, внутрішня економіка DApp (наприклад, MANA, BAT)	ERC-20, BEP-20, TRC-20	Смарт-контракт на Solidity; функції transfer, approve, mint, burn; OpenZeppelin реалізація; деплой через Hardhat	Контроль перепо-внення, аудит коду; відсутність комп-лаєнсу KYC; ризик повторних викликів
2	Стейблкоїни (Asset-backed / Algorithmic)	Платіжна стабільна валюта (USDC, DAI)	ERC-20 + модулі адміністратора або CDP-системи	Для кастодіальних – мультипідпис і freeze/blacklist; для алгоритмічних – смарт-контракти CDP, oracles, механізм ребалансу	Вразливість до втрати резервів чи помилок оракулів; централізований ризик
3	Security Tokens (Цінні папери, RWA)	Токенізовані акції, облігації, частки у фондах	ERC-1400, ERC-3643 (T-REX), ST-20	Контракти з whitelist/blacklist, ролі операторів, інтеграція з DID/KYC-сервісами (ONCHAINID)	Регуляторний комплаєнс; обмеження доступу; журналювання транзакцій
4	Governance Tokens (Токени управління)	Голосування у DAO, зміна параметрів протоколу (COMP, UNI)	ERC-20 + модуль Governor / Timelock	Контракти голосування (GovernorBravo), делегування прав, таймлок-контроль; інтеграція Snapshot	Ризик концентрації голосів; потреба у прозорому аудиті
5	NFT (Невзаємоза-мінні токени)	Цифрове мистецтво, колекції, ігрові предмети	ERC-721, ERC-1155, SPL (Solana), FA2 (Tezos)	Контракти з унікальними ID, функціями safeTransferFrom, метаданими URI; зберігання через IPFS / Arweave	Ризик втрати метаданих; необхідність пінінгу; перевірка URI
6	Soulbound Tokens (Непереда-вані, іденти-фікаційні)	Атестати, дипломи, репутація	EIP-5114 (Soulbound Token), ERC-721 з модифікова-ним transfer	Контракт з блокуванням передачі; опціональне відкликання (revocation) через довірених осіб; інтеграція з DID	Захист приватності; потреба у соціальних механізмах відновлення
7	Fractional / Derivative NFT Tokens	Часткове володіння активом, колективні інвестиції	ERC-20 поверх ERC-721 (Fractiona-lization)	Контракт-сховище NFT + випуск ERC-20 часток; аукціон викупу 100%	Ризик шахрайських викупів; перевірка згоди учасників
8	Commodity / Real-World Asset Tokens	Золото, нерухомість, енергетичні активи	ERC-3643, Polymesh стандарт, FA2	Контракти з підтвердженням резервів (proof of reserve); інтеграція з оракулами і кастодіальними сервісами	Юридична відповідність, перевірка резервів, аудит оракулів
9	DeFi-Liquidity Tokens (LP, Farming)	Депозити в пулах ліквідності (Uniswap LP, Aave aTokens)	Специфічні протокольні контракти (UniswapV2Pair, cToken)	Контракт-«rebase» або «mint on deposit»; оновлення балансу згідно з часткою у пулі	Ризик re-entrancy; залежність від ціни базового активу

№	Тип токена / призначення	Приклади використання	Основний стандарт / модель	Метод реалізації на бекенді	Особливості безпеки та комплаєнсу
10	Hybrid / Cross-Chain Tokens	Мости між мережами, мультичейн NFT	ERC-20 / ERC-721 + Bridge Wrapper	Смарт-контракти мостів, підпис мультипідписантів, lock-mint модель	Ризики зламу мостів, подвійної емісії; необхідність доказів з іншого ланцюга

Представлена класифікація показує, що метод реалізації токенів безпосередньо залежить від їхнього призначення. Фунгібельні токени характеризуються простими контрактами з мінімальним числом подій і функцій, тоді як security та RWA-токени вимагають складної логіки з контролем доступу та ідентифікацією користувачів. NFT-технології залежать від позаланцюгового зберігання, а soulbound-токени відкривають напрямок інтеграції з децентралізованими ідентичностями (DID). Для DeFi кросчейн-токенів актуальні питання перевірки автентичності й ліквідності, що вимагають додаткових бекенд-модулів, зокрема, оракулів, смарт-аудиту, мультипідписів).

У табл. 3 наведено класифікацію веб-інструментів та методів для реалізації токенів.

Таблиця 3

Веб-інструменти реалізації токенів за функціональними рівнями

№	Рівень / Завдання	Основні веб-інструменти	Призначення	Типи токенів	Особливості
1.	Розробка смарт-контрактів	Remix IDE; Hardhat + Ignition; Truffle Suite + Ganache; Brownie / ApeWorx	Написання, компіляція та тестування контрактів у середовищах Solidity / Python; емуляція вузлів	Усі типи токенів (ERC-20, 721, 1155, 1400, 3643)	Швидке прототипування; автоматизація деплою; підтримка CI/CD
2.	Бібліотеки взаємодії (Web API)	ethers.js; web3.js; web3.py; Web3j (Java)	SDK для транзакцій, підписів, роботи з ABI та RPC-провайдерами	FT, NFT, security	Стандартний API для бекенду і DApp; інтеграція з Infura, Alchemy
3.	RPC-інфраструктура	Infura; Alchemy; QuickNode; Ankr; Moralis SDK	Хмарні вузли та API доступу до Ethereum, Polygon, BSC тощо	Усі типи токенів	Масштабований доступ без власних нод; авторизація через API-ключі
4.	Безпека та аудит	OpenZeppelin Contracts & Defender; MythX; Slither; Certora Prover	Аудитовані шаблони контрактів, формальна верифікація та моніторинг	ERC-20, 721, 1155, 1400, DeFi	CI/CD-моніторинг, контроль прав доступу, зниження ризику експлойтів
5.	Зберігання даних і метаданих	Pinata; NFT.storage; Web3.Storage; Arweave Gateway / Bundlr	IPFS-пінінг та перманентне зберігання контенту	NFT, RWA	Гарантована доступність CID; одноразова оплата за вічне зберігання
6.	Індексація та аналітика	The Graph (Subgraph Studio); Dune Analytics; Nansen; Covalent API	Створення GraphQL-API, SQL-запити до ланцюгів	NFT, DeFi, governance	Прискорення бекенду; дашборди та дослідження

№	Рівень / Завдання	Основні веб-інструменти	Призначення	Типи токенів	Особливості
7.	CI/CD та деплой у хмарі	Alchemy Deploy; Infura Deploy; GitHub Actions + Hardhat	Автоматизований деплой і тестування смарт-контрактів	Усі типи	Підтримка безперервної інтеграції та оновлення бекенду
8.	Авторизація та ідентичність	WalletConnect; Web3Auth; SpruceID / ONCHAINID	Веб-SDK для входу через гаманець і реалізації DID-ідентифікації	Security, soulbound	KYC/AML-модулі; OAuth/JWT-сумісність
9.	Інтеграція з фронтендом	Next.js + wagmi / RainbowKit; Thirdweb; Tatum API	React-та low-code-бібліотеки для підключення гаманців та маркетплейсів	NFT, RWA, utility	Генерація контрактів без коду; зручний SDK для DApp
10.	Моніторинг і логування	Tenderly; Blockscout; Etherscan API	Веб-інструменти для симуляцій і відстеження транзакцій	Усі типи токенів	Налагодження, debug-режим, аналітика в реальному часі

Зазначені інструменти дозволяють реалізувати повний цикл реалізації токенів у Web 3.0 – від написання контрактів (Remix, Hardhat, OpenZeppelin) до їх деплою та індексації (The Graph, Moralis) і веб-інтеграції (WalletConnect, Thirdweb). Це дозволяє проєктам швидко створювати децентралізовані системи токенізації з підтримкою зберігання, комплаєнсу та масштабування у мережах Ethereum, Polygon, Solana та інших.

Комплекс методів веб-розробки, що забезпечують повний цикл токенізації у середовищі Web 3.0 – від формального опису токена та розгортання смарт-контрактів до зберігання метаданих, масштабування та безпеки, виглядає наступним чином:

1. Методи смарт-контрактної токенізації (EVM-стандарти), наприклад, ERC-20, ERC-721 / ERC-1155, ERC-1400 / ERC-3643.

2. Методи децентралізованого зберігання метаданих, наприклад, протоколи IPFS та Arweave.

3. Методи індексації та доступу до даних токенів на бекенді, наприклад, інтеграції з The Graph (subgraphs, GraphQL) для побудови веб-API над подіями смарт-контрактів і доступу до станів токенів.

4. Методи масштабування операцій токенізації. Винесення обчислювально складних операцій (*mint, transfer*) на L2-ланцюги – Polygon, zkEVM, Immutable X або StarkNet. Вони зменшують витрати газу, підвищують пропускну здатність і забезпечують швидку фіналізацію транзакцій.

5. Методи міжланцюгової токенізації (альтернативні L1). Застосувуть блокчейн-платформи Solana, Flow, Tezos, які мають власні стандарти активів і забезпечують високопродуктивні або формально верифіковані сценарії токенізації.

6. Методи розробки та тестування смарт-контрактів. Потребують для реалізації наступних інструментів веб-розробника:

– OpenZeppelin – аудитовані імплементації стандартів;
– Hardhat – тестування, трасування та симуляція мережі;
– ethers.js – бекенд-взаємодія з блокчейном;
– Truffle/Ganache – локальне моделювання.

7. Методи аудиту та формальної верифікації токенів. Застосовуються для забезпечення безпеки та перевірки інваріантів токенів, коректності переходів станів:

статичний аналіз (Slither, Mythril), fuzz-тестування (Echidna, Foundry), формальна верифікація інваріантів (Scribble, Certora). Ці методи гарантують дотримання таких властивостей токенів як аутентичність, цілісність, незмінність, верифікованість та комплаєнс.

Розглянемо детальніше інформаційну безпеку токенів. Безпека токенів забезпечується поєднанням математично верифікованих інваріантів стану та криптографічних механізмів, які гарантують автентичність, цілісність і незмінність транзакцій у децентралізованому середовищі. Принципи інформаційної безпеки токенізації наведені у табл. 4.

Таблиця 4

Принципи безпеки токенізації

№	Рівень безпеки	Математичний принцип	Криптографічний механізм
1	Аутентичність	Операція вважається дійсною лише тоді, коли цифровий підпис успішно верифікується відкритим ключем відправника над хешем транзакції.	ECDSA або EdDSA
2	Цілісність	Цілісність забезпечується перевіркою послідовного хеш-ланцюга кожного запису, де кожен новий елемент пов’язаний з попереднім.	SHA-256 або Keccak-256
3	Незмінність станів	Кожна транзакція перевіряється на збереження наперед заданих інваріантів стану системи, тобто дозволені операції не порушують ключові обмеження.	Консенсус алгоритм (PoW, PoS, PBFT), який фіксує стан у блокчейні
4	Верифікованість	Для кожного токена існує криптографічний доказ (Меркл-доказ), який дає змогу перевірити, що він входить до глобального дерева станів.	Merkle Tree
5	Контроль доступу та комплаєнс	Операція дозволена лише за умови, що вона відповідає формалізованій політиці доступу та вимогам комплаєнсу.	Рольові модулі (Ownable, AccessControl), permissioned-стандарти (ERC-1400, ERC-3643)
6	Конфіденційність (опціонально)	Використовується механізм селективного розкриття – користувач розкриває лише необхідну частину даних, не відкриваючи повний стан.	zk-SNARK або zk-STARK

Аутентичність. Система повинна забезпечувати, що кожна транзакція походить від власника приватного ключа, пов’язаного з Owner у моделі (1).

Цілісність. Метадані Meta, стан State та інші складники токена не повинні бути змінені непомітно.

Незмінність станів. Жодна операція f: $S \to S'$ не може порушити інваріанти токена.

Верифікованість. Кожен елемент токена може бути перевірений незалежно від централізованої сторони.

Контроль доступу та комплаєнс. Параметр Policy з моделі (1) описує правила KYC/AML, часові обмеження, freeze/mint/revoke-права.

Конфіденційність (необов’язково, її необхідність залежить від цілей застосування токенізації). У деяких сценаріях токенізація вимагає селективного розкриття інформації – наприклад, перевірки права доступу без розкриття повних метаданих.

Таким чином безпека токенів базується на поєднанні математичних інваріантів стану та криптографічних доказів коректності. Еліптична криптографія гарантує автентичність підписів, хеш-ланцюги – незмінність історії, а консенсусні протоколи – узгодженість стану між вузлами.

Методика тестування методів токенізації

Тестування методів токенізації має на меті перевірку коректності виконання операцій смарт-контрактів, дотримання інваріантів стану та оцінку ефективності реалізації з погляду витрат обчислювальних ресурсів (*газу*). Під *газом* у середовищі Ethereum розуміють абстрактну одиницю обчислень, яка визначає кількість операцій, виконаних віртуальною машиною EVM під час транзакції. Загальна вартість транзакції визначається виразом:

$$Fee = GasUsed \times GasPrice, \quad (4)$$

де $GasUsed$ – кількість газових одиниць, витрачених під час виконання операції, а $GasPrice$ – ціна однієї одиниці газу в ETH.

Отже, чим більша обчислювальна складність функції, тим більше газу вона споживає, що безпосередньо впливає на економічну ефективність реалізації токена.

У даній роботі було розроблено методику формалізованого тестування процесів токенізації. Метою методики є перевірка коректності, безпеки, узгодженості та відповідності політик усіх операцій токенізації, незалежно від конкретної технологічної реалізації. Розробка методики спирається на:

– формальні описи стандартів токенів і смарт-контрактів (ERC-20/721/1155/1400/3643, SPL, FA2, Cadence) та їхніх офіційних специфікацій [1-3, 7-8, 17-19];

– практики безпечної розробки та аудиту, описані у відкритих матеріалах OpenZeppelin, ConsenSys Diligence, Trail of Bits, а також у роботах з формального аналізу Solidity-контрактів (Slither, статичний аналіз, security reviews) [6, 20, 24-26];

– дослідження ризиків та інфраструктури NFT і токенізаційних систем (зокрема, NIST IR 8472, документація IPFS, Arweave, The Graph) [11-14];

– регуляторні та концептуальні документи, присвячені токенізації активів, стейблкоїнам і ринковій інфраструктурі (BIS, FSB, MiCA, MakerDAO, Circle) [27-31].

Синтез цих джерел дозволив перейти від фрагментарних «best practices» та інструментальних рекомендацій до уніфікованої формальної методології, в якій токен розглядається як об’єкт із чітко заданою структурою, множиною станів, інваріантів, політик та критеріїв ефективності. Це створює основу для подальшої автоматизації тестування токенізації у Web 3.0 та для побудови формально обґрунтованих довірчих соціальних сервісів у розподілених комп’ютерних мережах.

Розглянемо запропоновану методику.

Методика передбачає попередню формалізацію множини допустимих операцій $F = \{f_1, \dots, f_n\}$, які охоплюють усі базові дії токенізації – емісію, передачу, знищення, делегування, заморозку, зміну політик тощо. Для кожної операції обчислюється новий стан $S' = f(S)$, після чого здійснюється наступна перевірка відповідності інваріантам:

– збереження сумарної пропозиції токена для взаємозамінних моделей:

$$\sum_{u \in A} bal(u) = totalSupply, \quad (5)$$

де A – множина всіх адрес/акаунтів користувачів, яким можуть належати токени; $u \in A$ – окремий користувач (адреса гаманця); $bal(u)$ – баланс токена на акаунті користувача u; $totalSupply$ – загальна пропозиція (загальна кількість випущених

одиниць) даного токена;

– унікальність власника у невзаємозамінних моделях:

$$\forall t \in ID\colon\ \exists!\, u \in A\colon\ owner(t) = u, \tag{6}$$

де ID – множина всіх ідентифікаторів токенів (наприклад, tokenId для NFT); $t \in ID$ – окремий токен (його унікальний ідентифікатор); A – множина всіх адрес/акаунтів користувачів; $u \in A$ – власник токена; $\mathrm{owner}(t)$ – функція, що повертає власника токена t; $\exists!\, u$ – існує єдиний u (унікальність власника).

– коректність виконання політик доступу й комплаєнсу:

$$\mathrm{Allow}(f) \iff f \vDash Policy, \tag{7}$$

де $\mathrm{Allow}(f)$ – предикат «операція f дозволена»; f – конкретна операція токенізації (з множини F); $Policy$ – формальна специфікація політик доступу та комплаєнсу (KYC/AML, ролі, часові обмеження, стани freeze/revoke тощо); $f \vDash \mathrm{Policy}$ – операція f задовольняє всім умовам, заданим у Policy (у логічному/формальному сенсі).

Перевірка коректності кожної операції потребує встановлення автентичності ініціатора транзакції. Це досягається шляхом валідації цифрового підпису, який повинен відповідати відкритому ключу власника, визначеному у компоненті Owner:

$$\sigma = \mathrm{Sign}_{priv}(H(\tau)), \tag{8}$$

де σ – цифровий підпис транзакції; $\mathrm{Sign}_{priv}(\cdot)$ – операція підписання за допомогою приватного (секретного) ключа власника; $H(\cdot)$ – криптографічна хеш-функція (наприклад, Keccak-256); τ – тіло транзакції, яке підписується (дані операції: відправник, одержувач, сума, параметри функції тощо); $H(\tau)$ – хеш від вмісту транзакції τ.

Таким чином, лише валідно автентифікований суб’єкт може ініціювати зміну стану, що унеможливлює несанкціонований доступ.

Далі забезпечується перевірка цілісності токена шляхом аналізу хеш-залежностей між поточними та попередніми станами. Для цього використовується узагальнений хеш-ланцюг, що гарантує неможливість непомітної модифікації будь-якої частини метаданих або станів токена:

$$H_i = H(T_i, H_{i-1}), \tag{9}$$

де H_i – хеш i-го елемента в хеш-ланцюгу станів (наприклад, поточного стану токена або блоку метаданих); T_i – дані i-го стану токена (або відповідного запису: метадані, балансні зміни, політики тощо); $H()$ – криптографічна хеш-функція, що приймає на вхід поточні дані й попередній хеш та повертає новий хеш.

Додатковим механізмом верифікованості є можливість побудови Меркл-доказу, який формально підтверджує належність конкретного токена до узгодженого глобального дерева станів, незалежно від будь-якого централізованого джерела:

$$\pi_t = \text{MerkleProof}(T_t), \quad (10)$$

де π_t – Меркл-доказ для токена t; $\text{MerkleProof}(\cdot)$ – алгоритм побудови доказу включення вузла в Меркл-дерево; T_t– вузол (лист) Меркл-дерева, який відповідає даним токена t (його стану чи метаданим).

В усіх випадках зміна стану токена повинна задовольняти обмеженням політик доступу. У формальному вигляді це представлено як логічний предикат, що гарантує дотримання ролей, KYC/AML-вимог, часових обмежень, freeze- або revoke-станів:

$$\text{Permit}(f, S) = \text{істинно}, \quad (11)$$

де $\text{Permit}(f, S)$ – логічний предикат, який набуває значення «істина», якщо операція f, виконувана в стані S, дозволена згідно з політиками доступу та комплаєнсу; f– операція токенізації (емісія, трансфер, заморозка тощо); S – поточний стан системи (баланси, ролі, KYC-статуси, стани freeze/revoke).

Таким чином, політики стають невід'ємною частиною інваріантної моделі та мають таку ж вагу, як і балансні чи структурні інваріанти.

Окремим елементом методики є формалізація критеріїв ефективності, відповідно до яких операція вважається прийнятною, якщо її обчислювальна вартість не перевищує референтного значення:

$$Gas(f) \leq \theta_{ref}, \quad (12)$$

а розмір байткоду реалізації – допустимої межі:

$$| bytecode | \leq \beta_{max}. \quad (13)$$

де $\text{Gas}(f)$ – функція, що повертає кількість одиниць газу, які споживає операція f при виконанні (її обчислювальна вартість); θ_{ref} – референтне (граничне) значення споживання газу, вище якого реалізація вважається неефективною або неприйнятною. $|\text{bytecode}|$ – довжина байткоду реалізації смарт-контракту (кількість байтів машинного коду, який буде розгорнуто в мережі); β_{max} – максимально допустиме значення довжини байткоду згідно з обмеженнями платформи або проєктними вимогами (наприклад, для зменшення вартості розгортання та складності аудиту).

На завершальному етапі результати формальної перевірки інваріантів та вимірювання витрат газу слід доповнювати статичним аналізом байткоду та вихідного коду смарт-контракту (Slither тощо) з метою виявлення відомих класів вразливостей.

Це дозволяє оцінювати оптимальність реалізацій токенів не лише з погляду безпеки, але й із погляду продуктивності та архітектурної ефективності.

Стислий алгоритм застосування запропонованої методики тестування токенізації:

Крок 1. Задати модель токена. Формалізувати стани, операції F та інваріанти (збереження пропозиції, унікальність власника, політики доступу) – формули (5)-(7).

Крок 2. Визначити політики та пороги ефективності. Описати Policy, предикати Allow/Permit та встановити граничні значення для газу й розміру байткоду – формули (7), (12), (13), з урахуванням вартості транзакції – (4).

Крок 3. Сформувати тестові сценарії для всіх операцій. Для кожної $f \in F$ задати коректні, граничні та некоректні (атаки, порушення політик) сценарії виконання.

Крок 4. Згенерувати й підписати транзакції. Побудувати транзакції для всіх сценаріїв, підписати їх і перевірити автентичність ініціатора за цифровим підписом – (8).

Крок 5. Виконати операції й зафіксувати результати. Запустити тести в контрольованому середовищі, одержати нові стани, спожитий газ і оновлені хеш-структури – формули (4), (9).

Крок 6. Перевірити інваріанти, політики й цілісність даних. Для кожної операції перевірити інваріанти стану (балансові, власності), виконання політик (Allow/Permit), цілісність хеш-ланцюга та наявність коректних Меркл-доказів – (5)-(7), (9), (10), (11).

Крок 7. Оцінити ефективність реалізації. Порівняти споживання газу та розмір байткоду з референтними порогами – (12), (13), за потреби позначити місця для оптимізації.

Крок 8. Провести статичний аналіз байткоду та вихідного коду смарт-контракту з метою виявлення відомих класів вразливостей.

Крок 9. Зробити висновок про коректність токена. Вважати реалізацію коректною, якщо для всіх операцій одночасно виконуються інваріанти, політики, вимоги до підписів, хеш-структур, Меркл-доказів та критерії ефективності – узагальнення умов (5)-(13), відсутні відомі класи вразливостей.

Узагальнюючи, методика встановлює, що реалізація токена є коректною тоді й тільки тоді, коли для всіх $f \in F$ виконується збереження інваріантів, валідність підпису, відповідність політикам, цілісність хеш-структур, наявність верифікованого Меркл-доказу, а також дотримання формальних критеріїв ефективності. Такий підхід дозволяє уніфіковано тестувати токени всіх типів – від ERC-20 до ERC-3643 і FA2 – на рівні логічних, криптографічних та системних властивостей, забезпечуючи повну формальну перевірку коректності токенізації у Web 3.0.

Це універсальна система, яка може бути застосована до будь-якого стандарту токенів (ERC-20/721/1155/1400/3643, SPL, FA2, Cadence).

Таким чином, удосконалено методику тестування моделей токенізації шляхом розроблення токен-центричної процедури, незалежної від конкретних стандартів, яка розглядає токен як абстрактний об’єкт зі множиною станів та операцій і задає єдину формальну умову коректності, що інтегрує перевірку інваріантів, політик доступу, цілісності хеш-структур та показників ефективності для всіх допустимих операцій.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У роботі здійснено комплексне дослідження методів і інструментів веб-розробки токенізації даних та об’єктів у середовищі Web 3.0 з акцентом на бекенд-рівень. Проведено систематизацію сучасних стандартів смарт-контрактів, моделей зберігання метаданих, інфраструктурних сервісів, архітектур масштабування та засобів безпеки, що використовуються у процесах токенізації. Сформовано класифікацію токенів за призначенням та класифікацію веб-інструментів за функціональними рівнями, що дозволило виявити відмінності між утилітарними, невзаємозамінними, фінансовими, репутаційними та регульованими активами з точки зору їх бекенд-реалізації, вимог до зберігання, індексації та комплаєнсу. На основі проведеного порівняльного аналізу показано, що найзрілішими та найефективнішими з погляду продуктивності та

простоти інтеграції залишаються стандарти EVM-екосистеми (ERC-20, ERC-721/1155), тоді як NFT-системи характеризуються високою гнучкістю, але підвищеною залежністю від безпечного позаланцюгового зберігання, а security-та RWA-токени забезпечують високий рівень регульованості ціною зниження децентралізації та збільшення складності аудиту. Також у роботі запропоновано уніфіковану формальну методику тестування процесів токенізації, що охоплює валідацію коректності операцій, перевірку виконання інваріантів стану, оцінювання політик доступу і комплаєнсу, аналіз криптографічних гарантій, а також вимірювання ефективності реалізації за витратами обчислювальних ресурсів і характеристиками байткоду. Такий підхід дозволяє комплексно оцінити якість, безпеку та надійність токенів незалежно від їх типу та архітектури реалізації, усуваючи методичну фрагментарність, властиву наявним підходам.

Отже, у роботі виконано систематизацію технологічних рішень, сформовано класифікації токенів та інструментів, проведено порівняльний аналіз основних моделей токенізації та розроблено формальну методику тестування, що може бути використана як основа для побудови довірчих соціальних сервісів та удосконалення бекенд-архітектур у Web 3.0. Перспективним напрямом подальших досліджень є поширення методики на мультичейн-інфраструктури, а також формалізація спеціалізованих токенів для соціальних, репутаційних і прикладних сервісів у розподілених комп'ютерних мережах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ethereum Foundation. (2015). *ERC-20: Token standard (EIP-20).* Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-20
2. Ethereum Foundation. (2018). *ERC-721: Non-fungible token standard (EIP-721).* Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-721
3. Ethereum Foundation. (2019). *ERC-1155: Multi token standard (EIP-1155).* Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-1155
4. Buterin, V. (2014). *Ethereum: A next-generation smart contract and decentralized application platform (White paper).* Ethereum Foundation. https://ethereum.org/en/whitepaper/
5. Wood, G. (2019). *Ethereum: A secure decentralised generalised transaction ledger (Yellow Paper, Berlin version).* Ethereum Foundation. https://ethereum.github.io/yellowpaper/paper.pdf
6. ConsenSys. (2023). *Ethereum smart contract best practices.* ConsenSys Diligence. https://consensysdiligence.github.io/smart-contract-best-practices/
7. Dossa, A., Ruiz, P., Vogelsteller, F., & Gosselin, S. (2018). *ERC-1400: Security token standard (EIP-1400, draft).* Ethereum Improvement Proposals. https://github.com/ethereum/EIPs/issues/1411
8. Tokeny Solutions. (2023). *ERC3643: The T-REX protocol – The token standard for real-world asset tokenization* (Whitepaper v4.0). https://tokeny.com/wp-content/uploads/2023/05/ERC3643-Whitepaper-T-REX-v4.pdf
9. ONCHAINID. (n.d.). Developers, welcome (ONCHAINID technical documentation). *ONCHAINID Docs.* https://docs.onchainid.com/docs/developers/intro/ (accessed: 20.11.2025)
10. Tokeny Solutions. (2023). *ERC3643: The token standard for real-world assets (RWAs).* Tokeny Resources. https://tokeny.com/erc3643-the-token-standard-for-real-world-assets-rwas/
11. Mell, P., & Yaga, D. (2024). *Non-fungible token security (NIST Interagency/Internal Report 8472).* National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8472
12. Benet, J. (2014). *IPFS – Content addressed, versioned, P2P file system.* arXiv. https://doi.org/10.48550/arXiv.1407.3561
13. Williams, S., Diordiiev, V., Berman, L., Raybould, I., & Uemlianin, I. (2019). *Arweave: A protocol for economically sustainable information permanence.* Arweave. https://www.arweave.org/yellow-paper.pdf
14. The Graph Foundation. (n.d.). *About The Graph*. https://thegraph.com/docs/en/about/ (accessed: 20.11.2025)
15. Polygon Labs. (2023). *Introducing Polygon 2.0: The value layer of the internet.* Polygon Technology

Blog. https://polygon.technology/blog/introducing-polygon-2-0-the-value-layer-of-the-internet
16. Immutable X. (2023). *Immutable X Whitepaper. V1.2.* https://www.immutable.com/
17. Solana Labs. (2023). *SPL Token-2022 standard documentation.* https://spl.solana.com/token-2022
18. Tezos Foundation. (2022). *TZIP-12: FA2 multi-asset interface standard.* Tezos Improvement Proposals. https://tzip.tezosagora.org/proposal/tzip-12/
19. Flow Developers. (2025). *Token development and registration: Fungible and non-fungible token standards.* Flow Developer Docs. https://developers.flow.com/blockchain-development-tutorials/tokens
20. OpenZeppelin. (2024). *OpenZeppelin Contracts v5.0: Audited Solidity implementations.* GitHub repository. https://github.com/OpenZeppelin/openzeppelin-contracts
21. Nomic Foundation. (n.d.). *Hardhat 3: Rust-powered Solidity tests Ethereum development environment for professionals.* Hardhat documentation. https://hardhat.org/ (accessed: 20.11.2025)
22. Ethers Project. (2023). *ethers.js v6 API documentation.* https://docs.ethers.org/
23. ATNO For Blockchain Developer. (2024). *How to Use Truffle and Ganache for Blockchain Development*. Medium. https://medium.com/@atnoforblockchain/how-to-use-truffle-and-ganache-for-blockchain-development-712322b9408a
24. Trail of Bits. (2019). *246 findings from our smart contract audits: An executive summary.* Trail of Bits Blog. https://blog.trailofbits.com/2019/08/08/246-findings-from-our-smart-contract-audits-an-executive-summary/
25. Diligence. (2021). *Umbra Smart Contracts.* https://diligence.security/audits/2021/03/umbra-smart-contracts/
26. Feist, J., Grieco, G., & Groce, A. (2019). *Slither: A static analysis framework for smart contracts.* In Proceedings of the 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB). P. 8-15. https://doi.org/10.1109/WETSEB.2019.00008
27. MakerDAO. (2024). *The Sky Protocol: Sky's Multi-Collateral Dai (MCD) System.* https://makerdao.com/whitepaper/
28. TRM Labs. (2025). *Seize, burn, block, reissue: Understanding the legal tools behind crypto asset recovery*. TRM Blog. https://www.trmlabs.com/resources/blog/
29. Bank for International Settlements. (2024). *Tokenisation in the context of money and other assets: Concepts and implications for central banks.* https://www.bis.org/cpmi/publ/d225.pdf
30. Financial Stability Board. (2024). *The financial stability implications of tokenisation.* https://www.fsb.org/2024/10/the-financial-stability-implications-of-tokenisation/
31. EUR-Lex (2023). *Regulation (EU) 2023/1114 of the European Parliament and of the Council*. Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2023/1114/oj/eng

.

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (31), 2025
ISSN 2663 - 4023

Volodymyr V. Mikhav
Doctor of Philosophy in Computer Engineering, Lecturer, Department of Information Technologies
Private Institution "University of Science, Entrepreneurship and Technology", Kyiv, Ukraine
ORCID: 0000-0003-4816-4680
mihaw.wolodymyr@gmail.com

Yelyzaveta V. Meleshko
Doctor of Technical Sciences, Professor, Associate Professor, Department of Cybersecurity and Software
Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine
ORCID: 0000-0001-8791-0063
elismeleshko@gmail.com

Mykola S. Yakymenko
Candidate of Physical and Mathematical Sciences, Associate Professor, Head of the Department of Higher Mathematics and Physics
Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine
ORCID: 0000-0003-3290-6088
m.yakymenko@gmail.com

Viktor V. Bosko
Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Automation of Production Processes
Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine
ORCID: 0000-0002-4933-9676
Victorvv2@ukr.net

Iryna A. Lysenko
Candidate of Technical Sciences, Senior Lecturer of the Department of Cybersecurity and Software
Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine
ORCID: 0000-0003-4394-4960
min_max@i.ua

METHODS OF WEB DEVELOPMENT AND TESTING OF DATA AND OBJECT TOKENIZATION FOR TRUSTED SOCIAL SERVICES IN DISTRIBUTED COMPUTER NETWORKS

Abstract. The article presents a comprehensive study of web development methods and tools for implementing tokenization of data and objects in the Web 3.0 environment, with a focus on the backend level. Modern smart-contract standards ERC-20, ERC-721/1155, ERC-1400/3643, SPL, FA2, and Cadence are analyzed, along with metadata storage approaches (on-chain, IPFS, Arweave), indexing mechanisms such as The Graph, and scaling architectures including L1/L2, zk-rollup solutions, and alternative L1 networks. A classification of token types by purpose and of their implementation methods is developed, as well as a classification of web tools by functional layers, which made it possible to identify specific features and differences in requirements for performance, interoperability, compliance, and information security. The comparative analysis shows that EVM-based implementations are the most mature and optimal for utility tokens, whereas NFT systems impose higher demands on secure off-chain storage, and security and RWA tokens provide regulatory controls and access management at the cost of a lower degree of decentralization. Based on a review of projects, it is shown that second-layer technologies, in particular Polygon zkEVM and Immutable X, reduce gas costs and increase transaction throughput compared to base networks. The paper also proposes a unified formal methodology for testing tokenization processes, which covers verification of operation correctness, state invariants, access policies, cryptographic guarantees, and transaction execution efficiency. The methodology integrates formal models of token behavior with gas-efficiency criteria and enables comprehensive assessment of the reliability of tokenization systems независимо of the underlying standard or technological stack. The results can be used for designing Web 3.0 backend architectures, developing trust-based social services in distributed computer networks, and creating a methodological foundation for automated token testing in multichain ecosystems.

Keywords: tokenization; computer networks; distributed networks; web tools; trust metrics; social services; information security; digital assets.

REFERENCES

1. Ethereum Foundation. (2015). *ERC-20: Token standard (EIP-20).* Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-20
2. Ethereum Foundation. (2018). *ERC-721: Non-fungible token standard (EIP-721).* Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-721
3. Ethereum Foundation. (2019). *ERC-1155: Multi token standard (EIP-1155).* Ethereum Improvement Proposals. https://eips.ethereum.org/EIPS/eip-1155
4. Buterin, V. (2014). *Ethereum: A next-generation smart contract and decentralized application platform (White paper).* Ethereum Foundation. https://ethereum.org/en/whitepaper/
5. Wood, G. (2019). *Ethereum: A secure decentralised generalised transaction ledger (Yellow Paper, Berlin version).* Ethereum Foundation. https://ethereum.github.io/yellowpaper/paper.pdf
6. ConsenSys. (2023). *Ethereum smart contract best practices.* ConsenSys Diligence. https://consensysdiligence.github.io/smart-contract-best-practices/
7. Dossa, A., Ruiz, P., Vogelsteller, F., & Gosselin, S. (2018). *ERC-1400: Security token standard (EIP-1400, draft).* Ethereum Improvement Proposals. https://github.com/ethereum/EIPs/issues/1411
8. Tokeny Solutions. (2023). *ERC3643: The T-REX protocol – The token standard for real-world asset tokenization* (Whitepaper v4.0). https://tokeny.com/wp-content/uploads/2023/05/ERC3643-Whitepaper-T-REX-v4.pdf
9. ONCHAINID. (n.d.). Developers, welcome (ONCHAINID technical documentation). *ONCHAINID Docs.* https://docs.onchainid.com/docs/developers/intro/ (accessed: 20.11.2025)
10. Tokeny Solutions. (2023). *ERC3643: The token standard for real-world assets (RWAs).* Tokeny Resources. https://tokeny.com/erc3643-the-token-standard-for-real-world-assets-rwas/
11. Mell, P., & Yaga, D. (2024). *Non-fungible token security (NIST Interagency/Internal Report 8472).* National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8472
12. Benet, J. (2014). *IPFS – Content addressed, versioned, P2P file system.* arXiv. https://doi.org/10.48550/arXiv.1407.3561
13. Williams, S., Diordiiev, V., Berman, L., Raybould, I., & Uemlianin, I. (2019). *Arweave: A protocol for economically sustainable information permanence.* Arweave. https://www.arweave.org/yellow-paper.pdf
14. The Graph Foundation. (n.d.). *About The Graph*. https://thegraph.com/docs/en/about/ (accessed: 20.11.2025)
15. Polygon Labs. (2023). *Introducing Polygon 2.0: The value layer of the internet.* Polygon Technology Blog. https://polygon.technology/blog/introducing-polygon-2-0-the-value-layer-of-the-internet
16. Immutable X. (2023). *Immutable X Whitepaper. V1.2.* https://www.immutable.com/
17. Solana Labs. (2023). *SPL Token-2022 standard documentation.* https://spl.solana.com/token-2022
18. Tezos Foundation. (2022). *TZIP-12: FA2 multi-asset interface standard.* Tezos Improvement Proposals. https://tzip.tezosagora.org/proposal/tzip-12/
19. Flow Developers. (2025). *Token development and registration: Fungible and non-fungible token standards.* Flow Developer Docs. https://developers.flow.com/blockchain-development-tutorials/tokens
20. OpenZeppelin. (2024). *OpenZeppelin Contracts v5.0: Audited Solidity implementations.* GitHub repository. https://github.com/OpenZeppelin/openzeppelin-contracts
21. Nomic Foundation. (n.d.). *Hardhat 3: Rust-powered Solidity tests Ethereum development environment for professionals.* Hardhat documentation. https://hardhat.org/ (accessed: 20.11.2025)
22. Ethers Project. (2023). *ethers.js v6 API documentation.* https://docs.ethers.org/
23. ATNO For Blockchain Developer. (2024). *How to Use Truffle and Ganache for Blockchain Development*. Medium. https://medium.com/@atnoforblockchain/how-to-use-truffle-and-ganache-for-blockchain-development-712322b9408a
24. Trail of Bits. (2019). *246 findings from our smart contract audits: An executive summary.* Trail of Bits Blog. https://blog.trailofbits.com/2019/08/08/246-findings-from-our-smart-contract-audits-an-executive-summary/
25. Diligence. (2021). *Umbra Smart Contracts.* https://diligence.security/audits/2021/03/umbra-smart-contracts/
26. Feist, J., Grieco, G., & Groce, A. (2019). *Slither: A static analysis framework for smart contracts.* In Proceedings of the 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software

Engineering for Blockchain (WETSEB). P. 8-15. https://doi.org/10.1109/WETSEB.2019.00008
27. MakerDAO. (2024). *The Sky Protocol: Sky's Multi-Collateral Dai (MCD) System.* https://makerdao.com/whitepaper/
28. TRM Labs. (2025). *Seize, burn, block, reissue: Understanding the legal tools behind crypto asset recovery*. TRM Blog. https://www.trmlabs.com/resources/blog/
29. Bank for International Settlements. (2024). *Tokenisation in the context of money and other assets: Concepts and implications for central banks.* https://www.bis.org/cpmi/publ/d225.pdf
30. Financial Stability Board. (2024). *The financial stability implications of tokenisation.* https://www.fsb.org/2024/10/the-financial-stability-implications-of-tokenisation/
31. EUR-Lex (2023). *Regulation (EU) 2023/1114 of the European Parliament and of the Council*. Official Journal of the European Union. https://eur-lex.europa.eu/eli/reg/2023/1114/oj/eng

(cc) BY-NC-SA
This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.