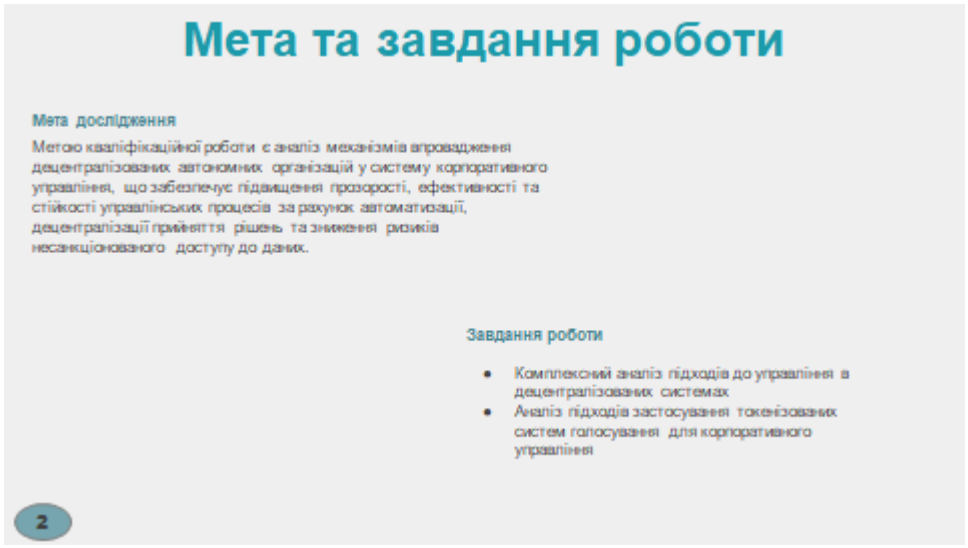
ДОДАТОК А

Графічний матеріал кваліфікаційної роботи


ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ
ФАКУЛЬТЕТ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА УПРАВЛІННЯ
Кафедра електронних обчислювальних машин
NURE
Кваліфікаційна робота на тему:
«Метод розробки децентралізованих автономних організацій у корпоративному управлінні»
другого(магістерського) рівня
Виконав:
здобувач гр. СПм-23-5
Хромов Нікіта Олексійович
Керівник:
доц. кафедри ЕОМ
Ткачов Віталій Миколайович


Мета та завдання роботи
Мета дослідження
Метою кваліфікаційної роботи є аналіз механізмів впровадження децентралізованих автономних організацій у систему корпоративного управління, що забезпечує підвищення прозорості, ефективності та стійкості управлінських процесів за рахунок автоматизації, децентралізації прийняття рішень та зниження ризиків несанкціонованого доступу до даних.
Завдання роботи
Комплексний аналіз підходів до управління в децентралізованих системах
Аналіз підходів застосування токенізованих систем голосування для корпоративного управління
2

Актуальність роботи

Сучасне корпоративне управління стикається з викликами: централізованість, низька прозорість, людський фактор.

Децентралізовані мережі усувають єдину точку відмови та забезпечують довіру без посередників.

Смарт-контракти автоматизують виконання рішень, мінімізуючи витрати та виключаючи суб'єктивні впливи.

Децентралізовані автономні організації — нова модель самоврядних цифрових структур, де управління закодоване.

Ці технології відкривають шлях до більш ефективних, прозорих і захищених бізнес-систем нового покоління.

3

Приклади застосування ДАО

Соціальні ініціативи та благодійність

Венчурне фінансування

Фінанси та інвестиції (DeFi)

Ігрові екосистеми (GameFi)

NFT-екосистеми

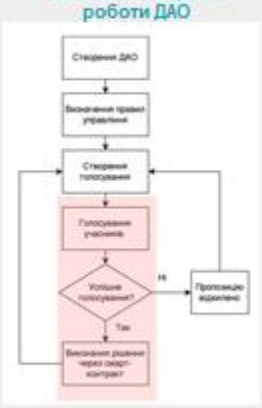
Управління інфраструктурою блокчейну

Управління цифровими спільнотами

4

Стаття дослідження

Базовий алгоритм роботи ДАО



Опублікована стаття

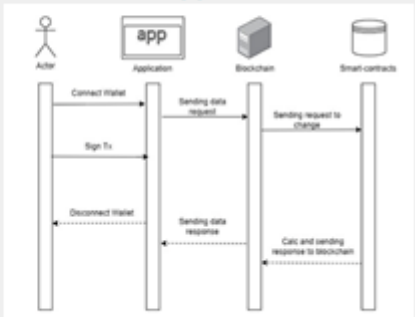
Покращений алгоритм роботи ДАО

5

Розробка вебзастосунку

Діаграма компонентів

Діаграма послідовності

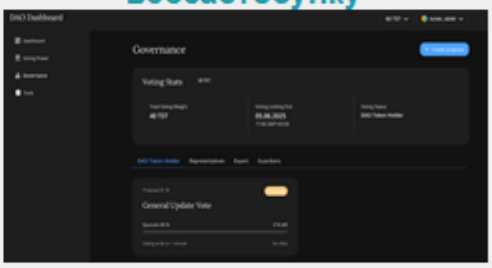


6

Інтерфейс застосунку

Карта сайту

Сторінка вебзастосунку



7

Тестування

Тестування:

- Проведено модульне тестування контрактів із використанням Hardhat + Chai.
- Інтеграційне тестування
- Зібрано соціальні метрики

Таблиця 4.2 – Метрики тестування вебзастосунку для управління ДАО

№	Метрика	Значення
1	Середній час генерації блоку, с	3.2
2	Відсоток пропущених блоків (за останній місяць)	0.15%
3	Середня затримка транзакцій, с	0.8 секунди
4	Пропускна здатність, с	1500 транзакцій
5	Коефіцієнт делегування (від загальної кількості токенів)	70%
6	Середня вартість транзакції, $	0.05
7	Активність голосування (середнє значення за 3 місяці)	45%
8	Частка успішно реалізованих пропозицій	85%
9	Кількість значних інцидентів безпеки	0
10	Середній час відновлення після збою, хв	< 10

8

Висновки

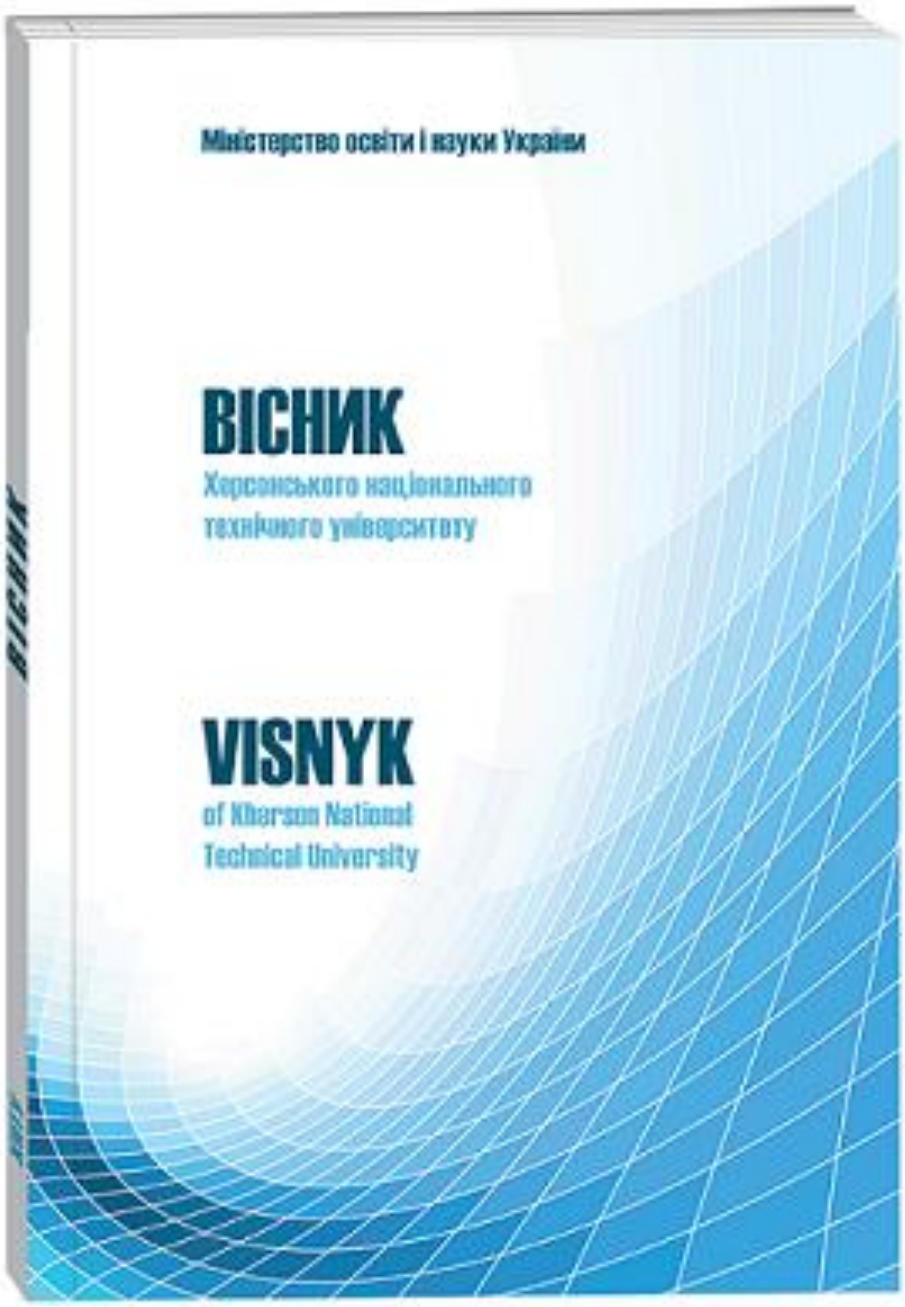
В кваліфікаційній роботі досягнуто основної мети – здійснено всебічний аналіз технологій та методів побудови та функціонування децентралізованих автономних організацій, реалізованих на основі блокчейн-архітектур.

Проведене тестування вебзастосунку, що реалізує модель управління ДАО в екосистемі CoreChain, дозволило підтвердити ефективність децентралізованих моделей управління. Аналіз технічних метрик ефективності функціонування вебзастосунку засвідчив високий рівень стабільності, надійності, безпеки та залучення спільноти до процесів прийняття рішень.

Отримані результати підтверджують доцільність впровадження ДАО як інструменту для створення нових моделей самоврядних цифрових спільнот та організацій, здатних ефективно функціонувати на основі прозорості, з орієнтацією на соціальну відповідальність в надійних інформаційних системах.

ДОДАТОК Б

Апробація результатів кваліфікаційної роботи



УДК 004.75 DOI https://doi.org/10.35546/kntu2078-4481.2025.1.2.32

Н. О. ХРОМОВ
магістр кафедри електронних обчислювальних машин
Харківський національний університет радіоелектроніки
ORCID: 0009-0008-5954-0462

АНАЛІЗ ДЕЦЕНТРАЛІЗОВАНИХ АВТОНОМНИХ ОРГАНІЗАЦІЙ У КОРПОРАТИВНОМУ УПРАВЛІННІ

У сучасному світі корпоративне управління стикається з низкою викликів, пов'язаних із централізацією влади, непрозорістю прийняття рішень, неефективним розподілом ресурсів та корупційними ризиками. Традиційні організації значною мірою покладаються на людський фактор, що може призводити до затримок, конфлікту інтересів та зайвих витрат. У цьому контексті децентралізовані автономні організації (ДАО) пропонують альтернативний підхід, використовуючи смарт-контракти та блокчейн для автоматизованого й прозорого управління.

Одна з основних переваг ДАО полягає в тому, що вони усувають необхідність у посередниках та централізованих структурах, замінюючи їх алгоритмічними механізмами голосування та фінансового управління. Такий підхід дозволяє кожному учаснику організації мати реальний вплив на процеси та гарантує чесний розподіл ресурсів відповідно до заздалегідь визначених правил у коді смарт-контрактів. В умовах цифровізації такі механізми є дуже важливими, оскільки вони забезпечують довіру між учасниками незалежно від їхнього мислення чи статусу та прозорість виконання всіх процесів. Це досягається завдяки тому, що всі транзакції, голосування та рішення зберігаються у блокчейні, де їх не можна змінити або приховати. Такий підхід управління значно підвищує рівень довіри з боку спільноти. Він також вирішує проблему централізованого управління, коли ключові рішення ухвалюються обмеженим колом осіб, що може призводити до недосконалих стратегій розвитку або конфліктів.

Уряди багатьох країн, зокрема США та держав Європейського Союзу, активно досліджують способи інтеграції ДАО у чинні правові системи, що свідчить про зростаючу зацікавленість у використанні таких структур для реального бізнесу. Незважаючи на певні виклики у сфері юридичного визначення, цей напрямок продовжує активно розвиватися, і вже зараз є приклади успішних проектів, що працюють на основі децентралізованих механізмів управління.

Розробка децентралізованих автономних організацій у корпоративному управлінні є важливим і перспективним напрямком, що дозволяє значно підвищити ефективність бізнес-процесів, забезпечити чесність та прозорість у прийнятті рішень, а також адаптувати компанії до нових умов цифрової економіки. З огляду на стрімке зростання популярності Web3-технологій, DeFi та інших децентралізованих систем, можна впевнено стверджувати, що децентралізовані автономні організації стануть невід'ємною частиною майбутнього корпоративного управління.

Ключові слова: *децентралізовані автономні організації, ДАО, управління, блокчейн, смарт-контракти, прийняття рішень.*

N. O. KHROMOV
Master at the Department of Electronic Computers
Kharkiv National University of Radioelectronics
ORCID: 0009-0008-5954-0462

ANALYSIS OF DECENTRALIZED AUTONOMOUS ORGANIZATIONS IN CORPORATE GOVERNANCE

In the modern world, corporate governance faces numerous challenges, including power centralization, non-transparent decision-making, inefficient resource allocation, and corruption risks. Traditional organizations rely heavily on human factors, which can lead to delays, conflicts of interest, and excessive costs. In this context, Decentralized Autonomous Organizations (DAOs) offer an alternative approach by leveraging smart contracts and blockchain technology for automated and transparent governance.

One of the key advantages of DAOs is that they eliminate the need for intermediaries and centralized structures, replacing them with algorithmic mechanisms for voting and financial management. This approach enables each member of the organization to have a tangible influence on decision-making while ensuring a fair distribution of resources based on predefined rules embedded in smart contracts. In an era of digital transformation, such mechanisms are crucial as they establish trust among participants regardless of their background or status and ensure full transparency in all processes. This is achieved by recording all transactions, votes, and decisions on the blockchain, where they cannot be altered or concealed. Such a governance model significantly enhances community trust and addresses the issue of centralized management, where critical decisions are made by a limited group of individuals, potentially leading to flawed development strategies or conflicts.

ДОДАТОК В

Лістинг коду файлу MasterDAOFactoryInstance.ts

```
class MasterDAOFactoryInstance extends
SystemContractInstance<MasterDAOFactory> {
  constructor(signer: Signer | providers.Provider, address:
string) {
    super(signer, "MasterDAOFactory.json", address);
  }
  async deployDAO(params: DAOConstructorParametersStruct,
txOptions?: TxOptions): Promise<ContractTransaction> {
    return this.submitTransaction("deployDAO", [params],
txOptions);
  }
  async deployDAOPanel(
    dao: string,
    params: DAOPanelConstructorParametersStruct,
    txOptions?: TxOptions,
  ): Promise<ContractTransaction> {
    return this.submitTransaction("deployDAOPanel", [dao,
params], txOptions);
  }
  async addModule(
    dao: string,
    params: DAOModuleContractorParametersStruct,
    txOptions?: TxOptions,
  ): Promise<ContractTransaction> {
    return this.submitTransaction("addModule", [dao, params],
txOptions);
  }

  async configureVotingSituations(
    dao: string,
    situations: IDAOVoting.ExtendedSituationStruct[],
    txOptions?: TxOptions,
  ): Promise<ContractTransaction> {
    return this.submitTransaction("configureVotingSituations",
[dao, situations], txOptions);
  }
  async configureVetoGroups(
    dao: string,
    params: IPermissionManager.VetoGroupStruct[],
    txOptions?: TxOptions,
  ): Promise<ContractTransaction> {
    return this.submitTransaction("configureVetoGroups", [dao,
params], txOptions);
  }

  async revokeDAOCreatorRole(dao: string, txOptions?:
```

```
TxOptions): Promise<ContractTransaction> {
    return this.submitTransaction("revokeDAOCreatorRole", [dao],
txOptions);
  }

  async predictDAOAddress(salt: string): Promise<string> {
    return this.instance.predictDAOAddress(salt);
  }

  getDAORegistryAddressFromTx(deployDAOReceipt:
ContractReceipt): string {
    return
this.getTransactionEvents(deployDAOReceipt)[EventNames.DeployedD
AORegistry].args.daoRegistryProxy_;
  }
}
```

ДОДАТОК Г

Лістинг коду файлу DAOInstance.ts

```
class DAOInstance {
  public DAORegistryInstance: DAORegistryInstance;
  constructor(
    public signer: Signer | providers.Provider,
    public DAORegistry: string,
  ) {
    this.DAORegistryInstance = new
DAORegistryInstance(this.signer, this.DAORegistry);
  }
  public async getConstitutionHash(): Promise<string> {
    const storage = await
this.getConfParameterStorageInstance(DAO_RESERVED_NAME);
    const hash = (await
storage.getDAOParameter(CONSTITUTION_HASH_ID)).value;
    return hash.toString();
  }
  public async getPanelVotingTokenAddress(panelName: string):
Promise<string> {
    const votingInstance = await
this.getDAOVotingInstance(panelName);
    return votingInstance.instance.votingToken();
  }
  public async getPermissionManagerInstance():
Promise<PermissionManagerInstance> {
    const permissionManagerAddress = await
this.DAORegistryInstance.instance.getPermissionManager();
    return new PermissionManagerInstance(this.signer,
permissionManagerAddress);
  }
  public async getVaultInstance(): Promise<DAOVaultInstance> {
    const vaultAddress = await
this.DAORegistryInstance.instance.getDAOVault();
    return new DAOVaultInstance(this.signer, vaultAddress);
  }
  public async getDAOVotingInstance(panelName: string):
Promise<DAOVotingInstance> {
    const voting =
      panelName === DAO_RESERVED_NAME
        ? await
this.DAORegistryInstance.instance.getGeneralDAOVoting(panelName)
        : await
this.DAORegistryInstance.instance.getExpertsDAOVoting(panelName)
;
    return new DAOVotingInstance(this.signer, voting);
  }
  public async getExpertsVotingInstance(panelName: string):
```

```
Promise<DAOVotingInstance> {
    const voting = await
this.DAORegistryInstance.instance.getExpertsDAOVoting(panelName)
;
    return new DAOVotingInstance(this.signer, voting);
  }
  public async getGeneralVotingInstance(panelName: string):
Promise<DAOVotingInstance> {
    const voting = await
this.DAORegistryInstance.instance.getGeneralDAOVoting(panelName)
;
    return new DAOVotingInstance(this.signer, voting);
  }

  public async getMemberStorageInstance(panelName: string):
Promise<DAOMemberStorageInstance> {
    const memberStorage = await
this.DAORegistryInstance.instance.getDAOMemberStorage(panelName)
;
    return new DAOMemberStorageInstance(this.signer,
memberStorage);
  }

  public async getConfParameterStorageInstance(panelName:
string): Promise<DAOParameterStorageInstance> {
    const parameterStorage = await
this.DAORegistryInstance.instance.getConfDAOParameterStorage(pan
elName);
    return new DAOParameterStorageInstance(this.signer,
parameterStorage);
  }

  public async getRegParameterStorageInstance(panelName:
string): Promise<DAOParameterStorageInstance> {
    const parameterStorage = await
this.DAORegistryInstance.instance.getRegDAOParameterStorage(pane
lName);
    return new DAOParameterStorageInstance(this.signer,
parameterStorage);
  }

  public async createVoting(
    votingInstance: DAOVotingInstance,
    params: CreateVotingParameters,
  ): Promise<ContractTransaction> {
    return votingInstance.createProposal(params.situation,
params.remark, params.callData);
  }
  public async getAggregatedParameters(panelName: string):
Promise<ParameterStructOutput[]> {
    let regParams: ParameterStructOutput[] = [];

    if (panelName !== DAO_RESERVED_NAME) {
```

```
      const parameterStorage = await
this.getRegParameterStorageInstance(panelName);
      regParams = await
parameterStorage.instance.getDAOParameters();
    }

    const confParameterStorage = await
this.getConfParameterStorageInstance(panelName);
    const confParams = await
confParameterStorage.instance.getDAOParameters();

    return regParams.concat(confParams);
  }
  public async getProposalTotalParticipate(panelName: string,
proposalId: BigNumberish): Promise<BigNumberish> {
    const votingInstance = await
this.getDAOVotingInstance(panelName);
    const proposalInfo = await
votingInstance.getProposal(proposalId);

    if (panelName !== DAO_RESERVED_NAME &&
proposalInfo.params.votingType === VotingType.Restricted) {
      const memberStorageInstance = await
this.getMemberStorageInstance(panelName);
      return await
memberStorageInstance.instance.getMembersCount();
    }
    const daoVaultInstance = await this.getVaultInstance();
    const token = await votingInstance.instance.votingToken();

    return daoVaultInstance.instance.getTokenSupply(token);
  }
  public async doesDaoSupportExternalLinks(): Promise<boolean> {
    try {
      const votingInstance = await
this.getGeneralVotingInstance(DAO_RESERVED_NAME);
      await votingInstance.instance.proposalSituationLink(0);

      return true;
    } catch (error) {
      console.error(`Error checking external link support in
DAO: ${error.message}`);

      return false;
    }
  }
}
```