

Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”  
Завідувач кафедри кібербезпеки  
та програмного забезпечення  
д.т.н., професор  
\_\_\_\_\_ Олексій СМІРНОВ  
“ \_\_\_\_ ” \_\_\_\_\_ 2024 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за другим (магістерським) рівнем вищої освіти**  
на тему  
**“Дослідження та програмна реалізація системи інтеграції**  
**мережних сервісів на основі технології АСІ”**

КБПЗ - 2024

Виконав здобувач вищої освіти  
II курсу, групи КІ-23М  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Скібінський Я.О.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Дреєв О.М.  
« \_\_\_\_ » \_\_\_\_\_ 2024 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Рівень вищої освіти магістр  
Галузь знань 12 “Інформаційні технології”  
Спеціальність 123 “Комп’ютерна інженерія”  
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ  
Завідувач кафедри  
д.т.н., проф.  
Олексій СМІРНОВ  
« 6 » вересня 2024 року

## ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

*Скібінському Ярославу Олександровичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ*

2. Керівник роботи *Дреєв Олександр Миколайович, канд. техн. наук, доцент*  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 19-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту *2.12.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

- |  |  |
|--|--|
| <i>1. Призначення та область використання.</i>           | <i>6. Наукова новизна.</i>                                     |
| <i>2. Перегляд аналогічних існуючих систем.</i>          | <i>7. Маркетингове та економічне обґрунтування ІТ-проєкту.</i> |
| <i>3. Опис і обґрунтування проектних рішень.</i>         | <i>8. Заходи з охорони праці та техніки безпеки.</i>           |
| <i>4. Етапи програмування системи.</i>                   | <i>9. Висновки.</i>  |
| <i>5. Впровадження системи в промислову експлуатацію</i> |  |

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

- |  |                 |
|--|-----------------|
| <i>Наукова новизна</i>                     | <i>1 аркуш</i>  |
| <i>Структурна схема системи</i>            | <i>1 аркуш</i>  |
| <i>Функціональна схема системи</i>         | <i>1 аркуш</i>  |
| <i>Діаграма процесів</i>                   | <i>1 аркуш</i>  |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> |
| <i>Показники економічної ефективності</i>  | <i>1 аркуш</i>  |

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання  
« 6 » вересня 2024 р.

Підпис керівника

\_\_\_\_\_  
(прізвище та ініціали)Завдання прийнято до виконання  
« 6 » вересня 2024 р.

Підпис здобувача

\_\_\_\_\_  
(прізвище та ініціали)

## АНОТАЦІЯ

**Скібінський Я.О. Дослідження та програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2024.**

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтеграції мережних сервісів на основі технології АСІ.

Метою розробки є дослідження та програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ.

Об'єктом дослідження є процес інтеграції мережних сервісів на основі технології АСІ.

Предметом дослідження є методи інтеграції мережних сервісів на основі технології АСІ.

Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

**Ключові слова:** комп'ютерна інженерія, інтеграція мережних сервісів, АСІ

## ABSTRACT

**Skibinsky Ja.O. Research and software implementation of network service integration system based on ACI technology. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.**

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the system of integration of network services based on ACI technology.

The goal of the development is the research and software implementation of the network service integration system based on ACI technology.

The object of research is the process of integration of network services based on ACI technology.

The subject of research is the methods of integration of network services based on ACI technology.

The research methods are based on the methods of the theory of building computer networks, the methods of mathematical statistics, and the methods of software development.

The result of the work is the software implementation of the network services integration system based on ACI technology.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Python environment.

**Keywords:** computer engineering, integration of network services, ACI

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	10
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	10
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	13
2.3 Розгорнута постановка завдання .....	14
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	15
3.1 Опис функціонування системи .....	15
3.2 Розробка структурної схеми.....	21
3.3 Розробка функціональної схеми .....	26
3.4 Розробка діаграми процесів.....	40
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	42
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	42
4.2 Захист розробленого програмного забезпечення.....	55
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	58
6 НАУКОВА НОВИЗНА .....	63

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>			
<b>Вим.</b>	<b>Арк.</b>	<b>№ докум.</b>	<b>Підп.</b>	<b>Дата</b>				
<b>Розроб.</b>	Скібінський Я.О.				Дослідження та програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ	<b>Літ.</b>	<b>Аркуш</b>	<b>Аркушів</b>
<b>Перев.</b>	Дресв О.М.					М	1	91
<b>Н.контр.</b>	Коваленко А.С.				ЦНТУ КІ-23М			
<b>Затв.</b>	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ .....	64
7.1	Визначення цільової аудиторії кінцевого готового продукту .....	64
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	65
7.3	Вибір методу оцінки вартості ПЗ .....	68
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	69
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ .....	69
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ .....	71
7.7	Визначення ключових факторів успіху конкретного проєкту.....	73
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ .....	74
8.1	Вступ.....	74
8.2	Аналіз умов праці на робочому місці ІТ-фахівця .....	75
8.3	Пропозиції щодо підвищення працездатності ІТ-фахівців.....	77
8.4	Розрахунок системи загального штучного освітлення виробничого приміщення де працюють ІТ-фахівці .....	79
9	ОСНОВНІ ВИСНОВКИ.....	83
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	85

КБПЗ-2024

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>2</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

БД	–	база даних
УЛММР	–	узагальнена лінійна модель множинної регресії
МНК	–	метод найменших квадратів
СУБД	–	система управління базою даних
УМНК	–	узагальнений метод найменших квадратів
ОПР	–	особа яка приймає рішення

КБПЗ – 2024

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

**Актуальність теми.** Cisco ACI – це рішення SDN, яке визначає мережеву інфраструктуру на основі мережевих політик. Щоб зробити це можливим, Cisco створила ОС ACI Fabric, яка працює на всіх системах у мережі ACI. Ця спільна ОС дає змогу різним комутаторам у мережі ACI перетворювати політики в проекти інфраструктури. Центр обробки даних ACI – інфраструктура, орієнтована на застосування. ACI складається з ключових компонентів, які працюють разом, щоб забезпечити його функціональність, що дозволяє адміністраторам легко керувати своєю мережевою інфраструктурою.

– Контролер інфраструктури політики застосування (APIC) діє як централізований контролер керування та політики для всієї вашої IT-інфраструктури.

– Комутатори Spine встановлюють швидке з'єднання між кінцевими комутаторами, які відповідають за підключення серверів і кінцевих точок до структури ACI.

– Профілі мережі додатків (ANP) містять параметри підключення до мережі, політики безпеки та якості обслуговування (QoS) для певних програм або груп.

– Кінцеві точки включають фізичні або віртуальні сервери, віртуальні машини або пристрої, підключені до мережі ACI.

**Мета й завдання дослідження.** Метою роботи є дослідження та програмна реалізація системи інтеграції мережних сервісів на основі технології ACI.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем інтеграції мережних сервісів на основі технології ACI.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

– Дослідження системи інтеграції мережних сервісів на основі технології АСІ.

– Програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ.

*Об'єктом дослідження є процес інтеграції мережних сервісів на основі технології АСІ.*

*Предметом дослідження є методи інтеграції мережних сервісів на основі технології АСІ.*

*Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.*

**Наукова новизна отриманих результатів.** У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод інтеграції мережних сервісів на основі технології АСІ.

– Розроблено вітчизняний продукт інтеграції мережних сервісів на основі технології АСІ, який має більш широкі можливості, на відміну від існуючих аналогів.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі інтеграції мережних сервісів на основі технології АСІ.

**Достовірність наукових результатів** підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ\_2024

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Середовище Cisco ACI складається з двох основних компонентів:

- Контролер інфраструктури політики застосування Cisco (APIC): APIC – це контролер SDN для Cisco ACI. Він створює політики, які визначають мережеву інфраструктуру центру обробки даних.
- Комутатори Nexus 9000: комутатори Nexus 9000 використовують ОС ACI Fabric для зв'язку з APIC і створення інфраструктури на основі політик. Вони можуть бути перемикачами Spine (розподіл) або Leaf (доступ).

Усі кінцеві точки, включаючи APIC, підключаються до мережі через комутатори Leaf. Ці перемикачі Leaf з'єднані разом за допомогою перемикачів Spine у серверній частині.

Використовуючи ці компоненти, ACI можна розгорнути в різних моделях. Це включає підтримку локальних, хмарних (включно з загальнодоступними, приватними та гібридними хмарами) і крайових середовищ SD-WAN. Це дозволяє організаціям використовувати мережеве керування на основі політик у своїх корпоративних глобальних мережах.

### Основні функції та переваги ACI

Cisco ACI дозволяє організаціям легко створити програмно-визначений центр обробки даних, який забезпечує кілька переваг, зокрема:

- Гнучкість. З таким рішенням SDN, як Cisco ACI, уся мережева інфраструктура організації реалізована як код. Це дозволяє легко оновлювати конфігурації відповідно до нових потреб бізнесу.
- Послідовна інфраструктура: Cisco ACI абстрагується від деталей базової інфраструктури. Це полегшує проектування та налаштування мережевих середовищ.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

– Автоматизація та оркестровка: Cisco ACI активно використовує автоматизацію для розробки мережевої інфраструктури на основі мережевих політик. Це полегшує внесення змін і підвищує масштабованість.

– Підтримка гібридних середовищ: Cisco ACI підтримує як локальну, так і хмарну інфраструктуру, що дає змогу розгорнути середовища ACI у багатьох різних середовищах.

Важливим кроком стала дезагрегація комутаторів – відділення апаратної частини мережних пристроїв від програмної. Цей процес стимулювали великі оператори ЦОДів, які почали прямо закуповувати «голі» апаратні платформи bare metal (без ОС) безпосередньо в їхніх виробників і – завдяки наявності великої бази власних розроблювачів – писати програмні стеки для цих пристроїв. У результаті їм удалося значно скоротити капітальні витрати.

У свою чергу, зусилля виробників мікросхем (компаній Broadcom, Marvell, Intel і ін.), привели до відділення ASIC від інших апаратних компонентів. Вони почали пропонувати так звані комерційні набори мікросхем (merchant silicon), які, на відміну від спеціалізованих мікросхем ASIC, доступні на ринку в якості окремого готового продукту, а не як компоненти іншого виробу.

## 1.2 Область застосування

Оператори ЦОДів стали ініціаторами таких проектів, як Open Compute Networking Project і Open Network Linux (ONL), діяльність яких дозволила поширити такий підхід на інших замовників. У відповідь на виниклий попит з'явилися незалежні компанії, що спеціалізуються на розробці мережних операційних систем, які можна встановлювати на різні комутаційні апаратні платформи. Подібний поділ устаткування й ПЗ дає замовникам більшу волю вибору різних платформ, підвищує рівень конкуренції між постачальниками, сприяє зниженню капітальних витрат і, нарешті, стимулює інновації, оскільки

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

кожний рівень такого IT-стека можна розвивати й удосконалювати незалежно від інших рівнів.

Паралельно із процесами дезагрегації апаратного й програмного забезпечення мережних пристроїв ішов процес розвитку програмувальних мереж SDN. Концепція SDN припускає поділ функцій по безпосередній передачі трафіку й функцій контролю з виносом останніх з мережних пристроїв і централізацією в контролері SDN. Оскільки значну частину функцій, які раніше повинен був виконувати традиційний комутатор, тепер реалізує зовнішній контролер, мережна операційна система комутатора може бути значно полегшена до рівня агента OpenFlow. Цей стандартизований протокол використовується для взаємодії між контролерами й комутаторами в мережі SDN.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи інтеграції мережних сервісів на основі технології ACI, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

**2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти**

### **Logicalis покращує мережу APDL за допомогою Cisco ACI**

Клієнт: Адміністрація портів Дору, Лейшойнс і Віана-ду-Каштелу (APDL) у Португалії є державною компанією, метою якої є управління портами в тій самій географічній зоні, здійснюючи їх економічну експлуатацію, збереження та розвиток, включаючи повноваження, покладені на управління порту.

Маючи 5 км причалів, 55 га набережних і 120 га водно-болотних угідь, порт Leixões оснащений найсучаснішими системами управління та безпеки для портового руху. Надані послуги охоплюють усі аспекти доставки та працюють 24 години на добу, 365 днів на рік.

За словами керівництва APDL, впровадження новітніх технологій стало одним із критичних факторів успіху та розвитку портового бізнесу.

APDL шукала рішення для комутації центрів обробки даних (з резервуванням, надійністю, безпекою та веб-інтерфейсом), яке дозволило б розширити центр обробки даних і зробити мікросегментацію, сприяючи гнучкому управлінню додатками, високій продуктивності та автоматизації.

### **Рішення**

Враховуючи потреби APDL, Logicalis реалізував рішення Cisco Application Centric Infrastructure (ACI), оскільки, окрім того, що воно є провідним у галузі SDN (Software Defined Networking), воно автоматизує мережу та безпеку у веб-інтерфейсі для робочих навантажень на фізичних серверах, віртуальних машинах, контейнери або публічна хмара, що забезпечує:

– Автоматизація з'єднання кількох локальних і хмарних центрів обробки даних.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

– Універсальна політика, яка забезпечує стабільну безпеку та забезпечує безперервність роботи, прискорюючи будь-які аварійні відновлення.

– Централізоване керування: єдина панель для керування, видимості, моніторингу та усунення несправностей у багатошарних середовищах.

– Розширення мереж Lan і Wan за допомогою контролера інфраструктури політики застосування (APIC) для уніфікації мережевого керування та автоматизації в центрі обробки даних і корпоративній мережі.

Рішення ACI допомогло APDL оптимізувати керування своєю роботою та ефективніше вирішувати будь-які проблеми з мережею.

Сама організація змінила своє мислення, коли справа доходить до конфігурації, відходячи від протоколів до нової парадигми.

Це також дозволило їм почати думати про функціональні можливості та поєднати низку технологій.

Він також надав можливість бачити всю мережу як єдине ціле та систематизувати спосіб конфігурації додатків APDL (зменшуючи час і складність конфігурації та прискорюючи їх впровадження в дротову та бездротову мережеву інфраструктуру).

### **Посилення безпеки Cisco ACI за допомогою Check Point**

Cisco ACI надає низку вбудованих рішень безпеки. Співпраця з Check Point покращує ці засоби захисту, щоб забезпечити розширене запобігання загрозам для середовищ ACI.

### **Як Cisco ACI інтегрується з іншими продуктами**

Cisco ACI створено з використанням відкритої екосистеми Cisco ACI. Ця відкрита екосистема розроблена, щоб надати низку різних варіантів підключення інструментів сторонніх виробників до Cisco ACI, зокрема:

– Відкриті API: API Cisco ACI є відкритими, що дозволяє іншим продуктам підключатися та взаємодіяти з середовищем Cisco ACI.

– Спільно сертифіковані рішення: Cisco співпрацює з понад 65 постачальниками технологій для створення екосистеми SDN. Ці спільні рішення сертифіковані кожною організацією, і обидві сторони надають інформацію для забезпечення сумісності попередньо створених рішень.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

– Сервіс Chaining: рішення Cisco ACI підтримують сервіс-ланцюжок, дозволяючи організаціям створювати рішення, які їм потрібні, щоб відповідати вимогам мережі та безпеки.

### **Check Point CloudGuard і Cisco ACI**

Check Point CloudGuard Network Security забезпечує узгоджене керування політикою та застосування розширених засобів захисту, автоматично розгортається та динамічно оркеструється в середовищі центрів обробки даних, визначених програмним забезпеченням. CloudGuard для Cisco ACI забезпечує провідну в галузі безпеку для середовищ ACI. CloudGuard надає такі можливості для покращення безпеки клієнтів Cisco ACI:

– Видимість і візуалізація хмарної мережі: CloudGuard реалізує мікросегментацію для середовищ ACI, забезпечуючи глибоке розуміння потоків трафіку як з півночі на південь, так і зі сходу на захід. Ця детальна видимість допомагає зрозуміти потоки даних у корпоративній мережі та забезпечити дотримання корпоративної політики безпеки.

– Розширене запобігання загрозам: розширені можливості запобігання загрозам CloudGuard поєднують повний стек безпеки, включаючи брандмауер, систему запобігання вторгненням (IPS), захист від вірусів і ботів, із безпечним віддаленим доступом, вилученням загроз і емуляцією загроз на основі пісочниці.

– Автоматизація та оркестровка: Cisco ACI дозволяє визначати мережеву інфраструктуру на основі мережевих політик. Інтеграція між Check Point CloudGuard і Cisco ACI означає, що організація може автоматично вставляти та надавати шлюзи безпеки CloudGuard у середовищах ACI для застосування політики безпеки.

– Застосування політики та відповідності: CloudGuard отримує контекст від APIC Cisco, що дозволяє використовувати інформацію про політику, визначену в середовищі ACI, для швидкого визначення політик безпеки. Потім ці політики безпеки можна легко застосувати в ACI за допомогою шлюзів CloudGuard.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

– Захист даних: інтеграція CloudGuard з АСІ дозволяє застосовувати захист від втрати даних (DLP) до середовищ АСІ. Це допомагає захистити конфіденційні дані організації від втрати чи викрадення.

– Централізоване керування безпекою: використання CloudGuard з АСІ дає змогу контролювати безпеку екосистеми АСІ та керувати нею з тієї ж консолі, що й інша мережева інфраструктура організації. Це полегшує аналітикам безпеки виявлення й реагування на потенційні загрози в публічних, приватних і локальних мережах.

Інтегроване рішення Check Point і Cisco дозволяє організаціям легко створювати та захищати програмно-визначений центр обробки даних і покращує безпеку Cisco API.

## **2.2 Обґрунтування вибору засобів для побудови системи та мови програмування**

Python – це потужна мова програмування, яка проста у вивченні. Він має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python разом з його інтерпретованим характером роблять його ідеальною мовою для створення сценаріїв і швидкої розробки додатків у багатьох сферах на більшості платформ.

Інтерпретатор Python і обширна стандартна бібліотека доступні у вихідному або двійковому вигляді для всіх основних платформ на веб-сайті Python <https://www.python.org/> і можуть вільно поширюватися. Цей же сайт також містить дистрибутиви та вказівники на багато безкоштовних сторонніх модулів Python, програм і інструментів, а також додаткову документацію.

Інтерпретатор Python легко розширюється за допомогою нових функцій і типів даних, реалізованих у С або С++ (або інших мовах, які можна викликати з С). Python також підходить як мова розширення для налаштовуваних програм.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

## 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи інтеграції мережних сервісів на основі технології АСІ.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

#### Мережа розроблювачів Cisco для АСІ

Спрощення швидкої інтеграції й настроювання систем для розгортання мережних сервісів, спостереження, керування й координації.

#### АСІ: відповідь на мінливу ситуацію

Зміни в галузі викликають зміна підходу до ІТ на всіх рівнях. Моделі використання власних ІТ-ресурсів замінюються моделями використання хмарних сервісів. Концепція «ІТ як послуга» (ITaaS) витісняється концепцією «додатка як послуга». Замість роздільної розробки й експлуатації популярність набирає інтегрована розробка й експлуатація (DevOps). Моделі керування, орієнтовані на пристрої, замінюються моделями керування, орієнтованими на додатки.

Для адаптивності бізнесу потрібна швидка адаптація додатків, тому ІТ-фахівці повинні надавати доступ до додатків за кілька годин, а не місяців. Збільшення або зменшення обсягу використовуваних ресурсів необхідно виконувати протягом декількох хвилин, а не годин.

Традиційні підходи розглядають розрізнені аспекти експлуатації й не передбачають активної взаємодії між відділами, відповідальними за додатки, мережі, безпеку й хмарні сервіси. Загальна операційна модель забезпечує швидкість адаптації додатків, спрощення експлуатації, високу продуктивність і можливість масштабування.

**Рішення: підхід до керування інфраструктурою, орієнтований на додатки**

Інфраструктура, орієнтована на додатки (АСІ), надає цілісну архітектуру із централізованою автоматизацією й профілями додатків на основі політик. АСІ

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

забезпечує гнучкість програмного забезпечення й масштабованість апаратної продуктивності.

Компоненти Cisco ACI:

- нові комутатори Cisco Nexus серії 9000;
- централізоване керування політиками й контролер Cisco APIC (Application Policy Infrastructure Controller);
- віртуальний комутатор додатків Cisco AVS для периметра віртуальної мережі;
- інноваційні програмні й апаратні технології;
- інтегрована фізична й віртуальна інфраструктура;
- відкрита система постачальників рішень для організації мережі, зберігання, керування й координації.

Основні характеристики ACI:

- спрощення автоматизації за рахунок моделі застосування політик на основі додатка;
- централізований контроль зі спостереженням за працездатністю додатків у режимі реального часу;
- відкрите програмне забезпечення для гнучкої інтеграції із групами DevOps і партнерами по системі;
- масштабування продуктивності й багатокористувальницький режим устаткування.

Майбутнє мереж ACI полягає в реалізації такого підходу до розгортання мережі, спостереженню за мережею й керуванню мережею, який би підтримував DevOps і швидко зміну додатків. Для досягнення цих цілей ACI зменшує складність і використовує загальну структуру політик, що дозволяє автоматизувати виділення ресурсів і керування ними.

ACI є інноваційною, безпечною архітектурою, що заснована на цілому ряді різних елементів. Нижче наведено короткий опис цих елементів.

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

## **Матриця комутації, орієнтована на додатки**

Об'єднання встаткування, програмного забезпечення й нових технологій ASIC для реалізації інтегрованого системного підходу, що дозволяє задіяти стандартні протоколи Ethernet і IP.

Розширювана, об'єктно-орієнтована ОС пропонує великі можливості програмування.

Оцінки стану, лічильники й політики на рівні додатків (як фізичних, так і віртуальних) спрощують усунення неполадок і забезпечують можливість контролю.

## **Централізоване керування, автоматизація й координація**

Загальна структура керування для відділів керування мережами, додатками, безпекою й віртуалізацією підвищує адаптивність IT-системи.

АСІ автоматизує всі завдання керування матрицею комутації, включаючи керування образами, настроювання й погоджування з іншими мережними сервісами інфраструктури, такими як міжмережні екрани й сервіси балансування навантаження.

## **Оптимізація змішаного робочого навантаження й міграції для будь-якого додатка, у будь-якій точці, у будь-який час**

Спрощення надання доступу до додатків при одночасному посиленні кореляції продуктивності й контролю між логічними й фізичними мережами.

Упорядкування переносу фізичних і віртуальних серверів між різними гіпервізорами й фізичними серверами для спрощення міграції з віртуального рівня на фізичний або з одного віртуального рівня на інший.

## **Безпечне багатокористувальницьке середовище**

Надійна ізоляція, що попереджає, і дотримання угод про рівень обслуговування для різних користувачів при одночасному послідовному застосуванні політик безпеки у фізичних і віртуальних додатках.

Співробітники центра обробки даних задають політики безпеки й мережних політиків, використовуючи єдину абстрактну мову. Завдяки цьому відділи

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

забезпечення безпеки можуть виконувати надійне завдання політик незалежно від топології мережі.

### **Можливості розширення й відкритість**

АСІ використовує відкриті АРІ-інтерфейси, відкритий вихідний код і відкриті стандарти, підтримуючи стратегію Cisco Open Network Environment (ONE). Це забезпечує найширший вибір елементів керування центрами обробки даних і інфраструктури.

Набір добре документованих АРІ-інтерфейсів до вищестоящих і нижчестоящих систем, надаваний через мережу розроблювачів Cisco, спрощує швидку інтеграцію системи й підвищує гнучкість наступних елементів:

- сервіси рівнів 4-7;
- інфраструктура віртуальної мережі;
- моніторинг;
- керування;
- послуги координації.

### **Захист капіталовкладень – людські ресурси й інфраструктура**

Задійте існуючу інфраструктуру й навички вже притягнутих ІТ-фахівців для зниження сукупної вартості володіння.

Інноваційний оптичний модуль двонаправленої передачі зі швидкістю 40 Гбіт/с дозволяє використовувати існуючу проводку 10 Гбіт/с для скорочення дорогої модернізації волоконно-оптичної мережі.

Замовники можуть розгорнути комутатори Cisco Nexus серії 9000 в автономному режимі або перейти на матриці комутації стандарту АСІ, виконавши відновлення програмного забезпечення. Такий підхід у сполученні з відкритими інтерфейсами для підтримки існуючої інфраструктури забезпечує захист капіталовкладень для всього сімейства продуктів.

АСІ забезпечує адаптивне розгортання додатків на підприємствах і в організаціях постачальників послуг. Лежача в її основі матриця комутації забезпечує дистанційний контроль, ефективність, масштабованість і облік

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

особливостей додатків, що приводить до зниження сукупної вартості володіння. Центри обробки даних наступного покоління буде будуватися на принципах ACI, які розраховані на вимоги поточних і майбутніх додатків.

### **Продукти**

**Комутатори Cisco Nexus серії 9000** містять інноваційні технології для забезпечення кращої в галузі продуктивності, потужності, щільності портів і можливостей програмування. Сімейство включає наступні пристрої:

- Cisco Nexus 9396PX;
- Cisco Nexus 93128TX;
- Cisco Nexus 9508.

**Віртуальний комутатор додатків Cisco AVS** надає погоджену інфраструктуру віртуального комутатора між матрицями комутації ACI і віртуальним комутатором Cisco Nexus 1000V для існуючих матриць комутації ЦОД.

**Контролер Cisco APIC (Application Policy Infrastructure Controller)** забезпечує програмну автоматизацію виділення мережних ресурсів і керування на основі вимог додатків і політик.

**Система захисту інфраструктури Cisco, орієнтованої на додатки**, для центрів обробки даних спрощує середовище замовника. Система захисту ACI розглядає міжмережні екрани як єдиний пул ресурсів і забезпечує їхнє інтелектуальне об'єднання відповідно до мережних політиків додатка. Система захисту ACI забезпечує динамічне прискорення встаткування й інтегрується безпосередньо в ACI.

Послуги, надавані компанією Cisco і нашими сертифікованими партнерами, приносять замовникам відчутну вигоду, таку як прискорення окупності на 15-20 %, зниження витрат на інфраструктуру на 30 %, прискорення аварійного відновлення на 50 % і скорочення часу розгортання на 90 %.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

## **Послуги Cisco для ACI**

**Послуги розробки бізнес-стратегії Cisco** включають підготовку економічного обґрунтування й генерального плану для впровадження інфраструктури, орієнтованої на додатки.

**Послуги підготовки й планування Cisco** забезпечують перетворення центрів обробки даних з використанням інфраструктури, орієнтованої на додатки.

**Послуга швидкого розгортання Cisco для Nexus 9000** включає консультації по розгортанню.

**Послуги прискореного розгортання Cisco для Nexus 9000** допомагають виконати швидкий перехід до архітектури, орієнтованої на додатки.

**Послуги підготовки до експлуатації Cisco для центрів обробки даних** забезпечують підготовку середовища до впровадження ACI, розглядаючи всі етапи її життєвого циклу.

**Підтримка продуктів Cisco** для програмних і апаратних рішень, пов'язаних з Cisco Nexus 9000, надається по усьому світі цілодобово й без вихідних. Розширена підтримка Cisco також включає Cisco SMARTnet або Cisco Smart Net Total Care\*.

## **Послуги Cisco для захисту інфраструктури центра обробки даних**

**Послуга оцінки архітектури безпеки центра обробки даних Cisco** допомагає проаналізувати інфраструктуру безпеки і її відповідність політиці безпеки.

**Послуга з переносу пристроїв Cisco ASA для захисту центра обробки даних** допомагає виконати перенос сторонньої платформи або багатофункціонального пристрою захисту Cisco ASA, включаючи конфігурації й правила міжмережних екранів, у віртуалізоване середовище.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

### 3.2 Розробка структурної схеми

Апаратна платформа «відкритого комутатора» може будуватися як на типових наборах мікросхем таких компаній, як Broadcom, Intel, Marvell і Mellanox, так і на стандартній архітектурі x86. Сервери x86, як правило, використовуються для підтримки віртуальних комутаторів, які споконвічно створювалися для комутації трафіку віртуальних машин усередині сервера й обміну трафіком із зовнішньою мережею. Найвідоміший віртуальний комутатор з відкритим вихідним кодом – Open vSwitch, він може функціонувати як автономно, у режимі традиційного комутатора (з формуванням таблиць комутації шляхом вивчення MAC-Адрес), так і під керуванням зовнішнього контролера SDN. Крім Open vSwitch, існують і інші рішення, наприклад Lagopus, а також комерційні віртуальні комутатори від Cisco, VMware і ін.

Останнім часом платформа x86 усе ширше використовується для побудови фізичних комутаторів, націлених на комутацію великої кількості зовнішніх потоків. Для цього застосовуються традиційні сервери Intel з більшим числом мережних інтерфейсів разом із уже згаданим Open vSwitch або, наприклад, Lagopus. При цьому вживають спеціальні міри для підвищення продуктивності таких комутаторів. Наприклад, Центр прикладних досліджень комп'ютерних мереж (ЦПДКС) разом з Intel розробили рішення, що до 10 разів підвищує продуктивність комутатора Open vSwitch. Воно реалізовано за допомогою набору бібліотек і драйверів Intel Data Plane Development Kit (DPDK), що забезпечує швидку обробку пакетів на платформах Intel. Зокрема, це рішення не вимагає переривань процесора для копіювання пакетів в оперативну пам'ять.

Як нас повідомили в ЦПДКС, комутатори на базі серверів x86 українського складання, у тому числі з підтримкою до 32 портів 10 Gb, уже тестуються українськими замовниками. Правда, рішення на 32 порту 10 Gb поки не є повністю заблокуєним. При цьому спеціалізовані рішення, наприклад на основі наборів мікросхем Broadcom і Mellanox, уже забезпечують заблокуєму

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

роботу комутатора з 32 портами 100 Gb. Так що в плані продуктивності, а також щільності портів такі платформи істотно випереджають комутатори на базі процесорів x86. Проте в замовника є вибір типу платформи для комутатора, що є важливу ознаку відкритих мереж.

Наступний ключовий елемент для вибору – мережна ОС. Як уже говорилося, раніше мережні ОС були жорстко прив'язані до «заліза». Тепер, якщо ваш постачальник комутатора прихильний принципам відкритості, у вас є вибір ОС. За останні кілька років з'явилося чимало розроблювачів незалежних мережних ОС: Cumulus Networks, Big Switch, Pica8, IP Infusion, Pluribus Networks і ряд інших компаній. Відповідно, чимало й апаратних платформ, що дозволяють інсталиювати ту мережну ОС, що більше підходить (подобається) конкретному замовникові.

У кожній мережній ОС є свої особливості й кращі області застосування. Наприклад, система Big Switch має свої переваги при створенні моніторингових комутаторів і побудові мережних фабрик у ЦОДах; рішення IP Infusion – при організації територіально розподілених мереж, у тому числі об'єднуючих віддалені ЦОДи; Pluribus Networks – при побудові розподілених фабрик ЦОДів, рішенні завдань аналітики й безпеки. Важливим плюсом системи Cumulus Linux є те, що замовники можуть використовувати уніфіковані засоби для керування мережею й серверами, які базуються на Linux. Комутатори Dell для відкритих систем (у їхній назві присутнє аббревіатура ON – від Open Networks) підтримують названі мережні ОС, а також операційну систему самої Dell, що, природно, фахівці компанії найчастіше рекомендують замовникам.

Що необхідно для того, щоб мережна ОС одного постачальника могла працювати на комутаторі іншого? Безумовно, треба гарантувати властиво установку ОС. При цьому бажано уникнути необхідності створення окремого образу під конкретний комутатор і максимально спростити установку. Для цього було розроблене середовище Open Network Installation Environment (ONIE). По суті, підтримка ONIE означає наявність передвстановленого ядра Linux

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

(микролинукс), що при першому завантаженні шукає джерело для установки повноцінної мережний ОС.

Середовище ONIE є одним із проектів Open Compute Project – співтовариства, заснованого Facebook (а згодом підтриманого такими компаніями, як Microsoft і Intel) з метою розробки оптимальної ІТ-інфраструктури для ЦОДів, включаючи сервери, системи зберігання, мережа й навіть стійки для розміщення встаткування. На відміну від більшості ініціатив, що стосуються «відкритих мереж», ОСР приділяє основну увагу не софту, а залозу. Одна з головних цілей співтовариства ОСР – просування енергоефективних і заснованих на типових компонентах технічних рішень, які б дозволили знизити капітальні й операційні витрати власників великих ЦОДів. Однак розробки ОСР, мабуть, можуть виявитися надзвичайно корисними й для інших категорій замовників.

Інша важлива розробка Open Compute Project – інтерфейс Switch Abstraction Interface (SAI), наявність якого дозволить перенести ОС на різні апаратні платформи. По суті SAI специфікує відкритий інтерфейс API для програмування мікросхем ASIC, що забезпечують високошвидкісну комутацію трафіку. Одним з основних ініціаторів розробки SAI була Microsoft, що вже розкрила деталі своєї мережної операційної системи Azure Cloud Switch (ACS). Взагалі, відкриті мережі творять чудеса. Ця ОС від Microsoft реалізована на базі Linux. Ще раз повторю: Microsoft... на базі Linux!

Слідом за поділом комутаторів на встаткування й ПЗ галузь пішла далі по шляху дезагрегації властиво мережних ОС і самих апаратних платформ. Уже згадана система Azure Cloud Switch по суті є модульною (див. мал. 3). Ще один приклад модульної мережний ОС – новітня система OS10 компанії Dell. Ця ОС включає базовий модуль (на основі немодифікованого дистрибутива Linux з відкритим вихідним кодом) і різні модулі додатків. Це можуть бути додатка самої Dell і сторонніх розроблювачів, а також «рідні» додатки Linux з відкритим вихідним кодом, включаючи програми для організації мережної фабрики,

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

маршрутизації, безпеки, керування й автоматизації. Такі інтернет-гіганти, як Google, Facebook і LinkedIn, розробили власні дезагреговані комутатори й, судячи із вступників від них інформації, цілком ними задоволені. Видимо, побоюючись, що слідом за ними на шлях створення власних комутаторів устануть і менші великі замовники, багато хто з постачальників традиційних мережних рішень також анонсували свої стратегії в області «відкритих мереж» і представили відповідні продукти. Це, зокрема, уже згадані в статті Dell і Mellanox, а також HP Enterprise і Juniper. І схоже, що найближчим часом до них приєднаються й інші гравці мережного ринку.

А що ж багаторічний і безперечний лідер ринку? А він тим часом пішов «проти тренда», запропонувавши нове покоління комутаторів Nexus на основі... власного набору мікросхем. При цьому, як затверджують представники компанії, своя елементна база – важлива конкурентна перевага Cisco, що забезпечує її комутаторам кращі характеристики. Правда, Cisco розвиває й рішення на базі комерційних чипів, але для флагманських ЦОДівських моделей робить ставку на власні.

Компанія не приєдналася до тяги, що охопила галузь, до дезагрегації комутаторів і підтримки мережних ОС сторонніх розроблювачів. Однак у вересні 2016 року запропонувала набір засобів по розширенню функцій мережний ОС NX-OS для комутаторів сімейств Nexus 3000 і 9000. Він одержав назву Open NX-OS і забезпечує багато чого з того, що активно просувають постачальники дезагрегованих комутаторів. Зокрема, підтримуються добре знайомі багатьом фахівцям по ІТ інструменти Linux ( NX-OS заснований на Linux), засобу для керування конфігураціями Puppet, Chef та ін. Більше того, можливо навіть використання засобів маршрутизації сторонніх постачальників, а новий SDK дозволяє розроблювачам створювати свої додатки з можливістю їхнього виконання в захищеному контейнері.

Важливим напрямком розвитку «відкритих мереж» для Cisco залишається SDN, у першу чергу рішення Application Centric Infrastructure (ACI). Відкриті

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

програмні інтерфейси API вхідного в це рішення контролера Application Policy Infrastructure Controller (APIC) відкривають широкі можливості для розробки й інтеграції з ACI сторонніх мережних сервісів. В систему партнерів Cisco по ACI уже входять більше 50 компаній, а найближчим часом список поповниться й українським розроблювачем засобів безпеки.

У роботі запропонований наступний підхід до дезагрегації програмного забезпечення й апаратури комутаторів, який представлений на рис. 3.1.

При необхідності завжди можна модифікувати або замінити компоненти цього комутатора. Наприклад, замість мікросервера, побудованого на процесорах Intel, установити мікросервер на процесорах ARM. Або, скажемо, розмістити всю електронну начинку в іншому корпусі, якщо існують особливі вимоги до захисту від зовнішніх впливів.

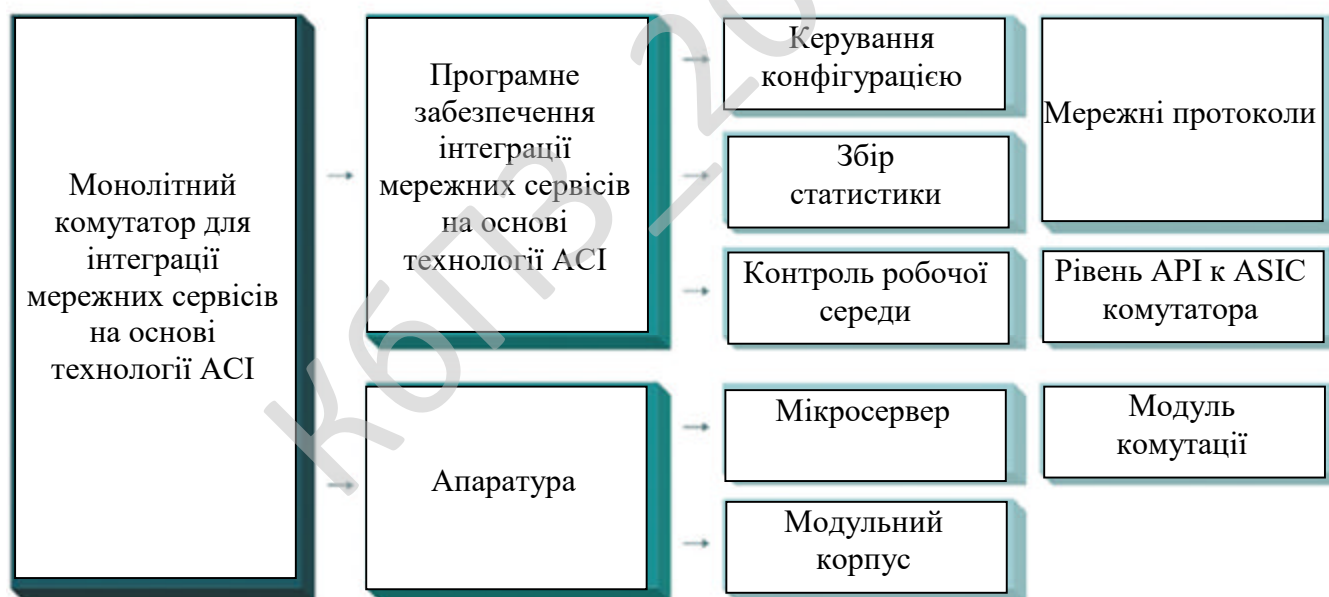


Рисунок 3.1 – Структурна схема системи

Відкритість інтерфейсів на рівні контролера SDN – річ, можна сказати, природна, споконвічно закладена в ідеологію програмувальних мереж. Забезпечення відкритості на рівні компонентів комутатора – для Cisco питання

неоднозначний і куди більше складний. Так що одна з інтриг полягає в тому, чи піде лідер ринку на дезагрегацію комутаторів і підтримку сторонніх мережних ОС.

### 3.3 Розробка функціональної схеми

Для реалізації програмного забезпечення інтеграції мережних сервісів на основі технології АСІ, сформуємо функціональну схему. Для того, щоб було можливо інтегрувати мережні сервіси, у роботі потрібно оцінити завантаженість трафіку у мережі, тобто необхідно реалізувати прогноз завантаженості. Це пропонується зробити з використанням математичного апарату аналізу часових рядів.

Для оцінки якості прогнозу введемо допоміжні критерії для різних видів вимог до якості прогнозу виду  $V = V(X_{екз}, X'_{екз})$  []. Для зручності опису процедури оцінки якості прогнозу введемо величину  $y(t) = |x'(t) - x(t)|$ , де  $t = t_1, t_2, \dots, t_k$  – абсолютна величина відхилення прогнозного значення від спостереженого на часовому такті  $1 \leq t \leq N$  (нев'язання).

#### Крапковий прогноз

Якість побудови прогнозу з найкращим збігом значення на  $m$ -му такті будемо оцінювати критерієм:

$$V_0 = y(m), 1 < m < N. \quad (3.1)$$

#### Траєкторний прогноз

Для оцінки якості побудови прогнозу на  $\tau$  тактів ( $1 \leq \tau < N$ ) з найкращим збігом прогнозних значень на всіх тактах, використовуємо один з наступних критеріїв:

$$V_1 = \sum_{j=1}^k \alpha_{t_j} \cdot y(t_j), \quad (3.2)$$

$$V_2 = \frac{1}{k-1} \sqrt{\sum_{j=1}^k \alpha_{t_j} \cdot y(t_j)^2}, \quad (3.3)$$

$$V_3 = \max_{t \in \{t_1, \dots, t_k\}} \alpha_t \cdot y(t) \quad (3.4)$$

$$V_4 = \frac{1}{k} \sum_{j=1}^k \alpha_{t_j} \cdot \frac{y(t_j)}{x(t_j)}, \quad (3.5)$$

де  $\alpha_{t_j}$  – вагові коефіцієнти, що відбивають ступінь важливості збігу прогнозного й реального значення на такті  $t_j$ , що задаються експертним шляхом.

### Прогноз максимуму

Потрібно побудувати траєкторний прогноз на  $\tau$  тактів з найкращим збігом величини максимуму прогнозу:

$$V_5 = |\max\{x(t_1), \dots, x(t_k)\} - \max\{x'(t_1), \dots, x'(t_k)\}|, \quad (3.6)$$

і величини прогнозу значення такту  $m^*$  з максимальним прогнозним значенням:

$$V_6 = |\arg \max\{x(t_1), \dots, x(t_k)\} - \arg \max\{x'(t_1), \dots, x'(t_k)\}|. \quad (3.7)$$

Найпоширеніший критерій оцінки якості прогнозу (3.3), що складається із суми квадратів відхилень реальних даних від результатів розрахунку по прогнозній моделі, не завжди точно описує подання прогнозиста про гарний прогноз. Наприклад, даний критерій може допускати помітні відхилення прогнозу від реальних значень на окремих тактах. Ясно, що така оцінка якості прогнозу не досить точно описує подання про гарний прогноз. На наш погляд, надійний критерій якості прогнозу повинен описувати подання прогнозиста про «гарний» і «поганий» прогноз, «велике» і «мале» відхилення реального процесу від прогнозу. Із цієї причини пропонується множина критеріїв оцінки якості прогнозу й постановок задач оптимізації для опису й вибору кращого прогнозу. В основі конструкцій критеріїв і постановок задач оптимізації лежить використання багатокритеріального опису й принципи оптимальності [7].

### Рішення задачі прогнозування як задачі оптимізації

Сформулюємо задачу оцінки якості прогнозу як задачу оптимізації [8]. При постановці оптимізаційних задач можливе використання оцінок як по одному з допоміжних критеріїв (3.1)–(3.7), так і по декількох критеріях одночасно []. Другий підхід дозволяє більш гнучко підбудуватися під вимоги експерта до якості прогнозу за рахунок вибору відповідної постановки задачі.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Нехай було побудовано  $L$  різних прогнозів  $X'_1, X'_2, \dots, X'_i, \dots, X'_L, i = 1, \dots, L$ , яким відповідають  $L$  наборів оцінок  $X'_{екзi} = \{x'_i(t_1), x'_i(t_2), \dots, x'_i(t_j), \dots, x'_i(t_k)\}$  значень часового ряду на тактах екзаменаційної вибірки  $X_{екз} = \{x(t_1), x(t_2), \dots, x(t_j), \dots, x(t_k)\}$ , де  $1 \leq t_j \leq N$  з використанням різних (або по-різному настроєних) прогнозних моделей  $M_p$ . Тоді абсолютна величина відхилення значення  $i$ -го прогнозу на часовому такті  $1 \leq t \leq N$  від спостереженого значення складе  $y_i(t_j) = |\hat{x}_i(t_j) - x(t_j)|$ .

**Опишемо однокритеріальні постановки задач оптимізації (оцінки якості прогнозу).**

1. Кращим вважається прогноз с мінімальним відхиленням прогнозних значень від реальних на  $m$ -ому такті:

$$X^{i*} = \arg \min_{i \in \{1, L\}} V_0(X_{екз}, X'_i) = \arg \min_{i \in \{1, L\}} (|x'_i(m) - x(m)|). \quad (3.8)$$

2. Кращим вважається прогноз із найменшим сумарним відхиленням прогнозних значень від реальних:

$$X^{i*} = \arg \min_{i \in \{1, L\}} V_1(X_{екз}, X'_i) = \arg \min_{i \in \{1, L\}} \sum_{j=1}^k \alpha_{t_j} \cdot y_i(t_j). \quad (3.9)$$

3. Кращим вважається прогноз із найменшим середньоквадратичним відхиленням прогнозних значень від реальних:

$$X^{i*} = \arg \min_{i \in \{1, L\}} V_2(X_{екз}, X'_i) = \arg \min_{i \in \{1, L\}} \frac{1}{k-1} \sqrt{\sum_{j=1}^k \alpha_{t_j} \cdot y(t_j)^2}. \quad (3.10)$$

4. Кращим вважається прогноз із найменшим максимальним відхиленням прогнозних значень від реальних, тобто гарантується, що відхилення прогнозу будуть не більше знайденого мінімального значення на всіх  $\tau$  тактах:

$$X^{i*} = \arg \min_{i \in \{1, L\}} V_3(X_{екз}, X'_i) = \arg \min_{i \in \{1, L\}} \max_{t \in \{t_1, \dots, t_k\}} \alpha_t \cdot y(t). \quad (3.11)$$

5. Кращим вважається прогноз із мінімальною величиною середньої помилки відхилення прогнозних значень від реальних:

$$X^{i*} = \arg \min_{i \in \{1, L\}} V_4(X_{екз}, X'_i) = \arg \min_{i \in \{1, L\}} \frac{1}{k} \sum_{j=1}^k \alpha_{t_j} \cdot \frac{y(t_j)}{x(t_j)}. \quad (3.12)$$

6. Кращим вважається прогноз із найменшим відхиленням максимального значення ряду:

$$\begin{aligned} X^* &= \arg \min_{i \in \{1, L\}} V_5(X_{екз}, X'_i) = \\ &= \arg \min_{i \in \{1, L\}} (|\max\{x(t_1), \dots, x(t_k)\} - \max\{x'(t_1), \dots, x'(t_k)\}|) \end{aligned} \quad (3.13)$$

7. Кращим вважається прогноз, у якому найбільше точно обчислюється такт із максимальним значенням ряду:

$$\begin{aligned} X^* &= \arg \min_{i \in \{1, L\}} V_5(X_{екз}, X'_i) = \\ &= \arg \min_{i \in \{1, L\}} (|\arg \max\{x(t_1), \dots, x(t_k)\} - \arg \max\{x'(t_1), \dots, x'(t_k)\}|) \end{aligned} \quad (3.14)$$

### Багатокритеріальні постановки задач

8. Кращим вважається прогноз із мінімальними зваженими середньоквадратичним відхиленням, відхиленням прогнозного максимального значення від реального й найбільш точним пророкуванням такту з максимальним значенням ряду:

$$\begin{aligned} X^* &= \arg \min_{i \in \{1, L\}} (\lambda_2 \cdot V_2(X_{екз}, X'_i) + \lambda_5 \cdot V_5(X_{екз}, X'_i) + \lambda_6 \cdot V_6(X_{екз}, X'_i)) = \\ &= \arg \min_{i \in \{1, L\}} (\lambda_2 \cdot \frac{1}{k-1} \sqrt{\sum_{j=1}^k \alpha_{t_j} \cdot y(t_j)^2} + \\ &+ \lambda_5 \cdot |\max\{x(t_1), \dots, x(t_k)\} - \max\{x'(t_1), \dots, x'(t_k)\}| + \\ &+ \lambda_6 \cdot |\arg \max\{x(t_1), \dots, x(t_k)\} - \arg \max\{x'(t_1), \dots, x'(t_k)\}|) \end{aligned} \quad (3.15)$$

де вагові коефіцієнти  $\lambda_2$ ,  $\lambda_5$ ,  $\lambda_6$  визначаються прогнозістом. За рахунок застосування ваг можна мініяти вимоги до якості прогнозу.

9. Кращим вважається прогноз із мінімальними зваженими середньоквадратичним відхиленням, сумарним відхиленням і максимальним відхиленням прогнозних значень від реальних:

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

$$\begin{aligned}
X^* &= \arg \min_{i \in \{1, L\}} (\lambda_1 \cdot V_1(X_{екз}, X'_i) + \lambda_2 \cdot V_2(X_{екз}, X'_i) + \lambda_3 \cdot V_3(X_{екз}, X'_i)) = \\
&= \arg \min_{i \in \{1, L\}} (\lambda_1 \cdot \sum_{j=1}^k \alpha_{t_j} \cdot y(t_j) + \lambda_2 \cdot \frac{1}{k-1} \sqrt{\sum_{j=1}^k \alpha_{t_j} \cdot y(t_j)^2} + \\
&\quad + \lambda_3 \cdot \max_{t \in \{t_1, \dots, t_k\}} \alpha_t \cdot y(t)) \quad , \quad (3.16)
\end{aligned}$$

де вагові коефіцієнти  $\lambda_1, \lambda_2, \lambda_3$  визначаються прогнозістом. За рахунок застосування ваг можна мініяти вимоги до якості прогнозу.

Відзначимо, що, застосовуючи різні принципи оптимальності й методику їхнього використання, викладені в [7], можна розширити множина формальних постановок задач. Однак, для розглянутих нижче задач прогнозування наведеної множини постановок цілком достатньо, тому що воно покриває основні вимоги експертів до якості прогнозів.

### Алгоритм побудови множини конкуруючих прогнозів

Розглянемо задачу побудови множини прогнозів з використанням різних прогнозних моделей, параметри яких перебувають у заданих інтервалах. Рішення цієї задачі дозволить сформуванню множини конкуруючих прогнозів для подальшої оцінки їхньої якості, і вибору найкращого з них.

Особливість поставленої задачі полягає в тому, що зазначено не конкретні значення параметрів прогнозних моделей, а діапазони їхніх можливих значень. Такий спосіб завдання значень параметрів актуальний у силу того, що експерт, апріорі не знаючи, при яких значеннях параметрів прогнозної моделі буде побудований кращий прогноз, позбувається від необхідності вручну перебирати його варіанти, що цікавлять, значень параметрів прогнозних моделей.

Нехай дано  $k$  різних прогнозних моделей  $X = M_p(\Theta_p, X, \tau)$ , де  $p = 1, 2, \dots, k$ ,  $X$  – вихідний часовий ряд,  $\tau$  – обрій прогнозування,  $\Theta = (\theta_\rho, \theta^2_\rho, \dots, \theta^j_\rho, \dots, \theta^{C_p}_\rho)^T$  – вектор-стовпець, що визначає параметри  $\rho$ -ої прогнозної моделі ( $C_p$  – кількість параметрів прогнозної моделі  $M_p$ ). І нехай експертом для побудови прогнозів обрано  $n$  різних прогнозних моделей  $t_i = M_i(\Theta_i, X, \tau)$ ,  $i = 1, 2, \dots, n$ .

Для кожної прогнозної моделі  $t_i$  задамо матрицю діапазонів значень її параметрів:

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

$$R_p^i = \begin{pmatrix} \theta \min_i^1 & \theta \max_i^1 & \theta st_i^1 \\ \dots & \dots & \dots \\ \theta \min_i^j & \theta \max_i^j & \theta st_i^j \\ \dots & \dots & \dots \\ \theta \min_i^{C_p} & \theta \max_i^{C_p} & \theta st_i^{C_p} \end{pmatrix} \quad (3.17)$$

у якій укажемо мінімальні  $\theta \min_i^j$  й максимальні  $\theta \max_i^j$  значення параметрів моделі, а також і крок їхньої зміни  $\theta st_i^j > 0$  ( $\theta \min_i^j < \theta \max_i^j$ ).

Таким чином,  $j$ -а змінна в  $i$ -ій прогнозної моделі може приймати  $v_i^j = \frac{\theta \max_i^j - \theta \min_i^j}{\theta st_i^j} + 1$  різних значень.

Побудуємо морфологічний ящик [7], що дозволить сформувати всі можливі комбінації значень параметрів прогнозної моделі. Для цього виявимо повний перелік можливих значень, які можуть приймати параметри прогнозної моделі  $m$ :

$$\begin{aligned} & \theta \min_i^1, \theta \min_i^1 + \theta st_i^1, \dots, \theta \min_i^1 + (v_i^1 - 1) \cdot \theta st_i^1 \\ & \theta \min_i^2, \theta \min_i^2 + \theta st_i^2, \dots, \theta \min_i^2 + (v_i^2 - 1) \cdot \theta st_i^2 \\ & \dots \dots \dots \\ & \theta \min_i^j, \theta \min_i^j + \theta st_i^j, \dots, \theta \min_i^j + (v_i^j - 1) \cdot \theta st_i^j \\ & \dots \dots \dots \\ & \theta \min_i^{C_p}, \theta \min_i^{C_p} + \theta st_i^{C_p}, \dots, \theta \min_i^{C_p} + (v_i^{C_p} - 1) \cdot \theta st_i^{C_p} \end{aligned} \quad (3.18)$$

Відзначимо, що число значень  $v_i^j$ , прийнятих різними параметрами моделей, може бути різним.

Для побудови множини можливих параметрів прогнозної моделі виявимо всі можливі альтернативи у вигляді векторів-стовпців виду  $\Theta_i$ , одержуваних зі сполучення різних значень параметрів прогнозної моделі, беручи з кожного рядка по одному значенню.

Вирішимо цю задачу, використовуючи наступний алгоритм. Візьмемо мінімально можливі значення всіх параметрів і одержимо альтернативу  $\Theta_i^1 = (\theta \min_i^1, \theta \min_i^2, \dots, \theta \min_i^j, \dots, \theta \min_i^{C_p})$ . Послідовно задаючи значення параметра

$\theta_i^{C_p}$  зі списку можливих значень  $\theta \min_i^{C_p}, \theta \min_i^{C_p} + \theta st_i^{C_p}, \dots, \theta \min_i^{C_p} + (v_i^{C_p} - 1) \cdot \theta st_i^{C_p}$  не міняючи значень інших параметрів, одержимо наступних  $v_i^{C_p} - 1$  альтернатив  $\Theta_i^2, \Theta_i^3, \dots, \Theta_i^{v_i^{C_p}}$  де  $\Theta_i^{v_i^{C_p}} = (\theta \min_i^{C_p}, \theta \min_i^{C_p} + \theta st_i^{C_p}, \dots, \theta \min_i^{C_p} + (v_i^{C_p} - 1) \cdot \theta st_i^{C_p})$ . Надалі, послідовно перебираючи всі можливі значення параметра  $\theta_i^{C_p-1}$ , знову будемо змінювати значення параметра  $\theta_i^{C_p}$  для кожного з них. І так далі до одержання повної множини альтернатив, потужність якого складе  $L_i = \prod_{j=1}^{j \leq C_i} v_i^j$ .

Побудуємо  $L_i$  прогнозів  $X_i^l = m_i(\Theta_i^l, X, \tau)$ , послідовно використовуючи як параметри прогнозу моделі  $m$  елементи сформованої множини альтернатив  $\Theta_i^1, \Theta_i^2, \dots, \Theta_i^j, \dots, \Theta_i^{L_i} \dots$

Провівши вищеописану процедуру для кожної із прогнозних моделей  $m_1, m_2, \dots, m_n$ , одержимо множину з  $L = \sum_{i=1}^n L_i$  конкуруючих прогнозів. Структура алгоритму побудови набору конкуруючих прогнозів наведена на рисунку 3.1.

### Аддитивні моделі прогнозування

Нами розглянутий опис якості прогнозів, постановки задач прогнозування як задач оптимізації й запропоновані алгоритми пошуку параметрів моделей, що приводять до кращих прогнозів. Для рішення всіх цих задач необхідне використання інструментарію у вигляді спеціальних прогнозних моделей [13, 14, 17, 18, 28, 32, 33, 35] на основі часових рядів.

На додаток до прогнозних моделей, описаних у літературі, у даній магістерській роботі пропонуються прогнозні моделі, що мають аддитивну структуру. Особливістю даних моделей є можливість проведення покомпонентного аналізу часового ряду з виділенням його складових у вигляді тренда, сезонного й випадкового компонента.

При побудові моделі часового ряду будемо відштовхуватися від його аддитивного розкладання.

Сформулюємо задачі аналізу часового ряду для побудови моделі:

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

1. Визначити, які з не випадкових функцій  $f_{mp}(t)$ ,  $\varphi(t)$  і  $\psi(t)$  (тренд, сезонні й циклічні фактори відповідно) присутні в розкладанні, тобто визначити значення індикаторів  $\alpha$ ,  $\beta$ ,  $\gamma$ .

2. Побудувати «гарні» оцінки для тих не випадкових функцій, які присутні в розкладанні.

3. Підібрати модель, що адекватно описує поведінку «випадкових залишків»  $s(t)$ , і статистично оцінити параметри цієї моделі.

### Визначення наявності не випадкової складової в часовому ряді

У першу чергу необхідно перевірити сам факт наявності або відсутності не випадкової (і залежної від часу  $t$ ) складової у розкладанні (1.1); по суті, мова йде про статистичну перевірку гіпотези [6]:

$$H_0 : \mathbf{E}x(t) = a = const, \quad (3.19)$$

(включаючи твердження про взаємну статистичну незалежність членів аналізованого часового ряду  $X = \{x(1), x(2), \dots, x(N)\}$  при різних варіантах конкретизації альтернативних гіпотез типу [6])

$$H_1 : \mathbf{E}x(t) \neq const. \quad (3.20)$$

Для перевірки цієї гіпотези застосуємо критерій серій, заснований на медіані. Розташуємо члени аналізованого часового ряду  $X = \{x(1), x(2), \dots, x(N)\}$  у порядку зростання, тобто утворимо варіаційний ряд  $x(1), x(2), \dots, x(N)$ .

Визначимо вибіркву медіану  $x_{med}$  за формулою [5]:

$$x_{med} = \begin{cases} x\left(\frac{n+1}{2}\right), & \text{якщо } n - \text{непарне} \\ \frac{1}{2} \left[ x\left(\frac{n}{2}\right) + x\left(\frac{n}{2} + 1\right) \right], & \text{якщо } n - \text{парне} \end{cases} \quad (3.21)$$

Після цього утворимо «серії» з нулів і одиниць, для чого замість кожного члена ряду  $X(t)$  запишемо 1, якщо  $X(t) > x_{med}$  або 0, якщо  $X(t) < x_{med}$  (члени ряду, що дорівнюють  $x_{med}$ , не враховуються). На статистичному аналізі цих серій заснована процедура перевірки гіпотези (3.19).

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Утворена послідовність одиниць і нулів характеризується загальним числом серій  $v(n)$  і довжиною самої довгої серії  $r(n)$ . При цьому під «серією» розуміється послідовність одиниць, що йдуть підряд, або нулів, що йдуть підряд.

Для перевірки гіпотези (3.19) скористаємося наступним правилом.

Якщо система нерівностей:

$$\begin{cases} v(n) > 0,5(n + 2 - 1,96\sqrt{n-1}) \\ r(n) < 1,42 \cdot \ln(n+1) \end{cases}, \quad (3.22)$$

не має рішення, то гіпотеза (3.19) відкидається з імовірністю помилки  $0.05 < a < 0.0975$  і підтверджується наявність залежної від часу не випадкової складової в розкладанні аналізованого часового ряду [6].

За допомогою даної методики нами були перевірені всі часові ряди, що піддавалися аналізу в ході виконання роботи. У всіх цих рядах була виявлена не випадкова складова. Візуальний аналіз показав, що як не випадкову складову у всіх аналізованих часових рядах має сенс виділяти тренд і сезонний компонент. Відмова від аналізу циклічного компонента була прийнята на підставі того факту, що, по-перше, часові ряди мають довжину не більше 6 років, у той час як циклічні явища проявляються з періодичністю перевищуючу довжину цих рядів. По-друге, модель, використана при аналізі сезонного компонента, може одночасно враховувати й циклічний компонент часового ряду.

Таким чином, розкладання часового ряду приймає вид:

$$x(t) = f_{mp}(t) + \varphi(t) + \varepsilon(t). \quad (3.23)$$

### Аналіз тренда

У роботі для опису тренда розглядається модель регресії виду

$$x(t) = f_{mp}(t; \Theta) + s(t), \quad t = 1, 2, \dots, n, \quad (3.24)$$

у якій передбачається відомим загальний вид функції  $f(t; \Theta)$ , але невідомими значення параметрів  $\Theta = (\theta_1, \theta_2, \dots, \theta_m)^T$ . Оцінки  $\Theta$  параметрів  $\Theta$  будуються за допомогою методів, описаних раніше.

У роботі розглядаються два випадки загального виду функції  $f_{mp}(t; \Theta)$ :

1. Функція  $f_{mp}(t; \Theta)$  має вигляд алгебраїчного полінома ступеня  $n$ , тобто:

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

$$f_1(t; \Theta) = \theta_0 + \theta_1 t + \theta_2 t^2 + \theta_3 t^3 + \dots + \theta_n t^n. \quad (3.25)$$

2. Функція  $f(t; \Theta)$  має експонентний вигляд, тобто:

$$f_1(t; \Theta) = \theta_0 \cdot e^{\theta_1 t} \quad (3.26)$$

Для знаходження оцінок  $\Theta$  невідомих параметрів  $\Theta$  для опису трендів виду (3.25) і (3.26) моделями лінійної множинної регресії необхідна заміна змінних для лінеаризації. Після того, як оцінки  $\Theta$  визначено, буде потрібно зворотна заміна змінних для знаходження прогнозних значень при різних  $t$ .

Для випадку (3.25) використовуємо заміну  $t^i = x_i$ ,  $i = 1, 2, \dots, n$ . Тоді функція прийме вид:

$$f_1(t; \Theta) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n. \quad (3.27)$$

Для знаходження нових значень  $f_1(X; \Theta)$  будемо підставляти в неї  $x_i = t^i$ ,  $i = 1, 2, \dots, n$ .

Для випадку (3.26) після логарифмування обох частин рівняння замінимо  $\theta_0' = \ln \theta_0$ , і замість вихідних даних при знаходженні  $\Theta$  будемо використовувати їхні натуральні логарифми:

$$\ln(f_2(t; \Theta)) = \theta_0' + \theta_1 \cdot t. \quad (3.28)$$

Для знаходження нових значень  $f_2(t; \Theta)$  використовуємо зворотну заміну  $\theta_0 = \beta^{\theta_0'}$  і одержимо:

$$f_2(t; \Theta) = e^{\theta_0' + \theta_1 t}. \quad (3.29)$$

При роботі з даними часових рядів у більшості випадків неможливо прийняти гіпотезу про взаємну некоррельованість регресійних залишків  $\varepsilon(t)$ , справедливість якої можна перевірити, використовуючи, наприклад, критерій Дербина-Уотсона [9]. Тому у випадках, коли гіпотеза відкидається, замість класичної лінійної моделі множинної регресії використовуємо узагальнену лінійну модель множинної регресії з автокоррельованими залишками.

Однак, при роботі з моделлю нам апріорі залишається невідомим значення коефіцієнта кореляції  $\rho$  між сусідніми за часом регресійними залишками. У цьому випадку скористаємося ітераційною процедурою Кохрейна-оркатта знаходження значення параметра  $\rho$  [8]:

1. Обчислюються звичайні МНК-оцінки  $\Theta'_{МНК}$ .
2. Підраховуються нев'язання 1-й ітерації  $\varepsilon^{(1)} = Y - X \cdot \Theta'_{МНК}$ .
3. Перше наближення  $\rho^{(1)}$  оцінки  $\rho'$  невідомого параметра  $\rho$  визначається в якості МНК-оцінки коефіцієнта регресії  $\rho$  у моделі  $\varepsilon_i^{(1)} = \rho \varepsilon_{i-1}^{(1)} + \delta_i^{(1)}$ .
4. Обчислюються УМНК-оцінки  $\Theta'_{УМНК}(\rho^{(1)})$  з матрицею  $\Sigma_0(\rho^{(1)})$ .
5. Підраховуються нев'язання 2-й ітерації  $\varepsilon^{(2)} = Y - X \cdot \Theta'_{МНК}(\rho^{(1)})$ .
6. Друге наближення  $\rho^{(2)}$  оцінки  $\rho'$  невідомого параметра  $\rho$  визначається в якості МНК-оцінки коефіцієнта регресії  $\rho$  у моделі  $\varepsilon_i^{(2)} = \rho \varepsilon_{i-1}^{(2)} + \delta_i^{(2)}$ , де залишки  $\delta_i^{(2)}$  задовольняють умовам класичної моделі.
7. Обчислюються УМНК-оцінки  $\Theta'_{УМНК}(\rho^{(2)})$  з матрицею  $\Sigma_0(\rho^{(2)})$  – і т.д.

Процедуру закінчують при стабілізації одержуваних значень  $\rho^{(k)}$ , тобто на стадії, коли чергове наближення  $\rho$  мало відрізняється від попереднього [8].

Після знаходження оцінок  $\Theta$  невідомих параметрів  $\Theta$ , розрахуємо оцінки  $J'_{mp}(t; \Theta')$  функції  $fm(t; \Theta)$  для  $t = 1, 2, \dots, n$ .

### Аналіз сезонного компонента

Перед початком аналізу сезонного компонента необхідно виділити тренд із вихідного часового ряду (3.5):

$$x_v(t) = x(t) - \hat{f}_{mp}(t). \quad (3.30)$$

Далі, отриманий часовий ряд  $x_v(t)$  апроксимуємо рядами Фур'є.

Коефіцієнти розкладання розраховуються по формулах для розглянутого випадку у вигляді:

$$b_n = \frac{2}{T} \sum_{k=0}^{T-1} x_v(k) \cdot \cos(n \cdot \omega \cdot k), \quad (3.31)$$

$$a_n = \frac{2}{T} \sum_{k=0}^{T-1} x_v(k) \cdot \sin(n \cdot \omega \cdot k), \quad (3.32)$$

$$b_0 = \frac{2}{T} \sum_{k=0}^{T-1} x_\varphi(k), \quad (3.33)$$

де  $\omega_0 = \frac{2\pi}{T};$

$$n = 1, 2, \dots, M;$$

$$M = \left\| \text{fft}(x) \right\|;$$

$T$  – період (є параметром моделі).

Оцінки значень сезонного компонента  $x'_\varphi(t)$  одержимо за формулою:

$$\varphi'(t) = x'_\varphi(t) = b_0 + \sum_{n=1}^M a_n \cdot \sin(n \cdot \omega_0 \cdot k) + \sum_{n=1}^M b_n \cdot \cos(n \cdot \omega_0 \cdot k) \quad (3.34)$$

### Аналіз випадкового компонента

Для аналізу випадкового компонента виділимо сезонний компонент  $\varphi'(t)$  з часового ряду:

$$x_\varepsilon(t) = x_\varphi(t) - \varphi'(t). \quad (3.35)$$

Допущення: після звільнення від тренда й сезонного компонента вважаємо ряд стаціонарним, як це прийнято в [29, 36].

Оцінку випадкового компонента  $x'_\varepsilon(t)$  одержимо, використовуючи модель авторегресії порядку  $p$  ( $AR(p)$  модель). Оцінка випадкового компонента розраховується за формулою:

$$\varepsilon'(t) = x'_\varepsilon(t) = \sum_{j=1}^p \alpha_j x_\varepsilon(t-j). \quad (3.36)$$

Оцінки майбутніх значень часового ряду будемо обчислювати за формулою:

$$\varepsilon'(t) = J'_{mp}(t) + \varphi'(t) + \varepsilon'(t), \quad (3.37)$$

де  $t = N + 1, N + 2, \dots, N + \tau;$

$N$  – число елементів вихідного часового ряду;

$1 < \tau < N$  – обрій прогнозування.

Підведемо короткі підсумки розглянутих вище моделей прогнозування на основі аналізу числових рядів.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

1. Розроблено багатокритеріальний опис якості прогнозів на підставі сформованої множини критеріїв оцінки їхньої якості, що враховують переваги експерта.

2. Для різних ситуацій змістовно описані вимоги до якості прогнозів як формальні постановки задачі прогнозування.

3. Задача прогнозування трансформована в задачу оптимізації й вирішена на множині конкуруючих прогнозів, де як цільова функція використовується функція якості прогнозу, сконструйована на основі критеріїв оцінки якості прогнозування.

4. Розроблено алгоритм побудови множини конкуруючих прогнозів з використанням різних і/або по-різному настроєних прогнозних моделей.

5. Розроблено діалоговий алгоритм рішення задачі прогнозування як багатокритеріальної задачі оптимізації на основі сформульованих критеріїв оцінки якості прогнозу.

6. Описано прогнозні моделі аддитивної структури, що дозволяють проводити покомпонентний аналіз складових часового ряду: тренда, сезонного й випадкового компонента.

Принцип дії системи наступний. З ядра TCP/IP одержуємо дані про RTT-затримки для всіх поточних з'єднань (сокетів), ці дані зберігаються в буфері, з буфера послідовність даних (вектор) надходить на вхід аналізатора часового ряду, на виході ми одержуємо прогноз наступного значення RTT-затримки. На підставі прогнозу розраховується розмір наступного TCP-вікна (розмір буфера вікна перевантаження). За допомогою ядра TCP/IP відбувається керуючий вплив на пропускну здатність каналу.

За допомогою протоколу SNMP в MIB формуються якісні й кількісні дані про потік даних (трафік), через даний мережний адаптер. Аналізатор часових рядів у взаємодії зі стандартним SNMP агентом, витягає необхідні дані з бази MIB, обробляє й зберігає в БД для подальшого аналізу.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38



### 3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

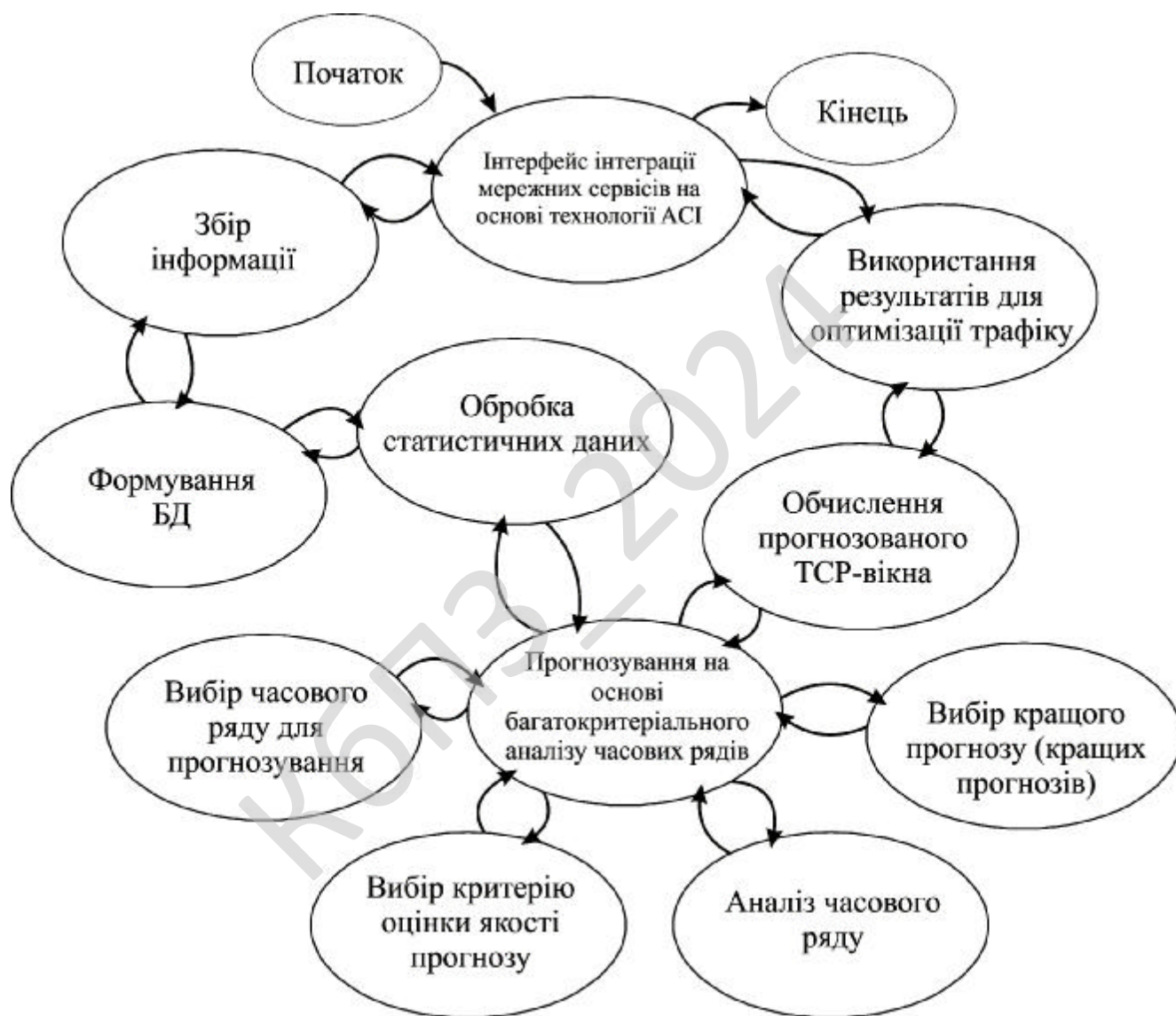


Рисунок 3.3 – Діаграма взаємодії процесів

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та

потоків даних з метою показати систему що розробляється. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

– Процеси які являють собою трансформацію даних в рамках описуваної системи.

– Сховища даних (репозиторії).

– Зовнішні по відношенню до системи сутності.

– Потоки даних між елементами трьох попередніх типів.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КБПЗ – 2024

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

## 4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

### 4.1 Блок-схеми та опис алгоритмів функціонування системи

Блок-схеми є основою ПЗ. Тому від точності і детальності проробки блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації, також те, що при розробці програми слід надати особливу увагу модулю інтеграції мережних сервісів на основі технології АСІ.

Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні блоки можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірки поточного стану та поверненням на початок схеми чи з завершенням роботи розробленого ПЗ.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

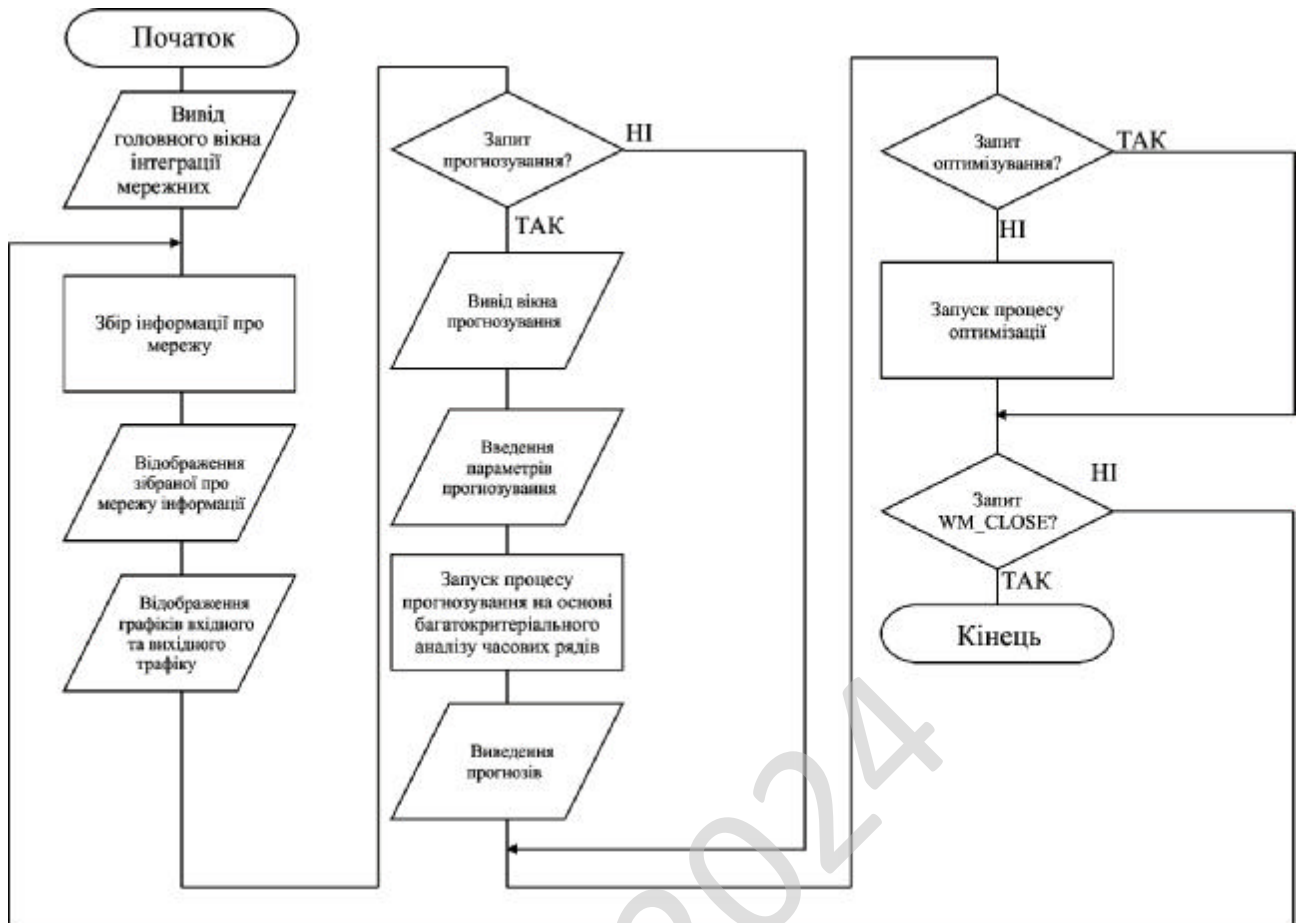


Рисунок 4.1 – Блок-схема основної програми

При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

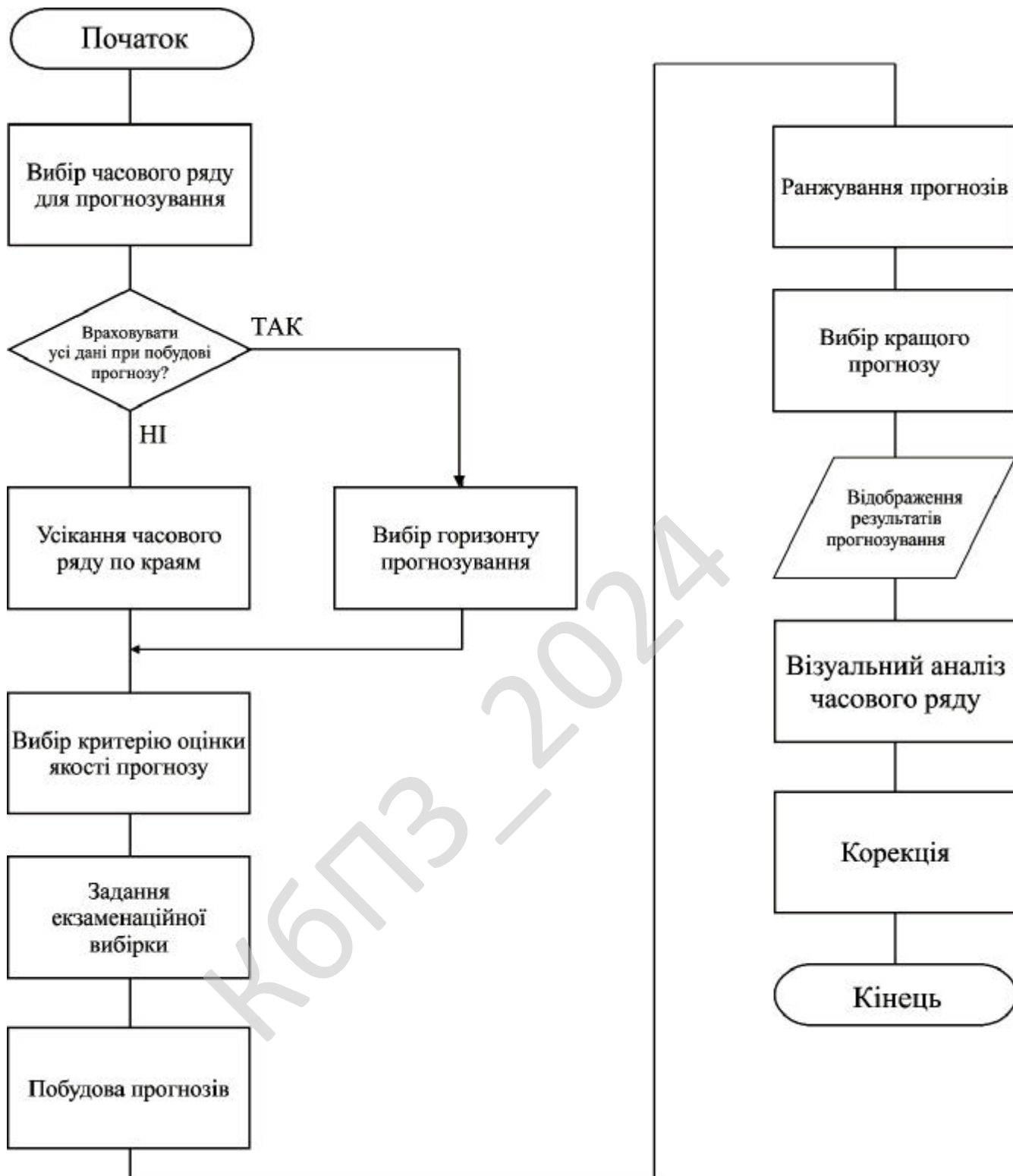


Рисунок 4.2 – Блок-схема роботи підпрограми

Розглянемо використані технології та їх основні компоненти що підтверджують правильність використаних проектних рішень.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі. Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

– рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;

– прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;

– рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

– модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

– модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML. Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузером.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі.

Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення – звичайний браузер. Це позбавляє його необхідності завантажувати та інсталювати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які

забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Python.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

Для того, щоб оцінити трафік мережі та його зміну з плином часу, тобто побудувати залежність завантаженості трафіку від часу, й отримати часовий ряд, треба взяти дані про рух у мережі з таблиць маршрутизації відповідних маршрутизаторів. Розглянемо вихідний код.

```
# Імпорт необхідних модулів для роботи з АСІ та мережними сервісами
import acitoolkit.acitoolkit as aci
import json
import requests
# Основний клас для інтеграції з АСІ
```

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

```

class ACINetworkIntegration:
    def __init__(self, url, username, password):
# Створення сесії з ACI API
        self.session = aci.Session(url, username, password)
        self.session.login()

    def create_tenant(self, tenant_name):
# Створення нового тенанта в ACI
        tenant = aci.Tenant(tenant_name)
        return tenant

    def create_application_profile(self, tenant, app_name):
# Створення нового профілю застосунку в тенанті
        app_profile = aci.AppProfile(app_name, tenant)
        return app_profile

    def create_epg(self, app_profile, epg_name):
# Створення End Point Group (EPG)
        epg = aci.EPG(epg_name, app_profile)
        return epg

    def create_contract(self, tenant, contract_name):
# Створення нового контракту для взаємодії між EPG
        contract = aci.Contract(contract_name, tenant)
        return contract

    def apply_contract_to_epg(self, epg, contract):
# Застосування контракту до EPG
        epg.provide(contract)
        epg.consume(contract)

# Функція для підрахунку використання ресурсів в мережі на основі отриманих даних
def calculate_resource_utilization(network_data):
    # Підрахунок використаних ресурсів (швидкість передавання, затримка)
    total_bandwidth = sum([data['bandwidth'] for data in network_data])
    total_latency = sum([data['latency'] for data in network_data])
    return total_bandwidth, total_latency

# Основний скрипт інтеграції з ACI
if __name__ == "__main__":
    # Параметри для підключення до ACI
    aci_url = "https://apic.example.com"

```

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>49</b>

```

aci_username = "admin"
aci_password = "password"

# Створення об'єкта для інтеграції з ACI
aci_integration = ACINetworkIntegration(aci_url, aci_username, aci_password)

# Створення нового тенанта та профілю застосунку
tenant = aci_integration.create_tenant("MyTenant")
app_profile = aci_integration.create_application_profile(tenant, "MyApp")

# Створення EPG для бази даних та веб-застосунку
db_epg = aci_integration.create_epg(app_profile, "DB_EPG")
web_epg = aci_integration.create_epg(app_profile, "Web_EPG")

# Створення контракту між EPG
contract = aci_integration.create_contract(tenant, "Web-DB-Contract")
aci_integration.apply_contract_to_epg(db_epg, contract)
aci_integration.apply_contract_to_epg(web_epg, contract)

# Збір даних про мережу для подальших розрахунків
network_data = [
    {"bandwidth": 1000, "latency": 10},
    {"bandwidth": 500, "latency": 20},
    {"bandwidth": 2000, "latency": 5}
]

# Виконання підрахунків використання ресурсів
total_bandwidth, total_latency = calculate_resource_utilization(network_data)

# Виведення результатів підрахунків
print(f"Загальна пропускна здатність мережі: {total_bandwidth} Mbps")
print(f"Загальна затримка: {total_latency} ms")

```

Ця система забезпечує базову архітектуру для інтеграції мережних сервісів на основі технології Cisco ACI. Вона включає створення тенантів, профілів застосунків, End Point Group (EPG), і контрактів для їх взаємодії. Для підтвердження правильності проектних рішень додано функцію розрахунку використання мережних ресурсів (пропускна здатність, затримка).

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

## Об'єктна модель системи прогнозування

При побудові інформаційної системи прогнозування завантаженості трафіку на основі багатокритеріального аналізу часових рядів (надалі підсистеми «Прогноз») застосовувався об'єктно-орієнтований підхід, у рамках якого й розглянемо особливості реалізації підсистеми «Прогноз».

У підсистемі «Прогноз» можна виділити наступні логічні групи об'єктів:

- об'єкти, що утворюють ядро системи прогнозування;
- об'єкти, що реалізують різні постановки задач прогнозування;
- об'єкти, що реалізують прогнозні моделі;
- інтерфейсні об'єкти.

**Управління вимогами** це процес запису, аналізу, трасування, пріоритетизації і узгодження вимог та контролю змін і доведення до їх зацікавлених сторін. Це безперервний процес протягом всього життя проекту. Вимога – якість, якій мають відповідати результати проекту (продукту або послуги).

Мета управління вимогами полягає в тому, щоб переконатися, що організація відповідає потребам і очікуванням своїх клієнтів, внутрішніх або зовнішніх зацікавлених сторін. Управління вимогами починається з аналізу і виявлення цілей і обмежень організації. Управління вимогами додатково включає в себе підтримку планування вимог, інтеграції вимог і організації роботи з ними (атрибути для вимог).

Управління вимогами передбачає спілкування між членами проектної групи і зацікавленими сторонами, і адаптацію до змін у вимогах протягом всього проекту. Щоб запобігти перетину поля одного класу вимог з іншим, постійні зв'язки між членами команди розробників є критичними. Наприклад, при розробці програмного забезпечення для внутрішнього використання у бізнесу можуть бути настільки сильні потреби, що він може проігнорувати вимоги користувачів, або вважати, що створені сценарії використання покривають також і користувальницькі вимоги.

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Відстеження вимоги фактично означає документування всього життєвого циклу вимоги. Часто необхідно дізнатися першоджерело кожної вимоги. Для цього всі зміни вимог повинні бути задокументовані, щоб досягти стану повного відстеження. Відстежувати треба бути навіть використання реалізованих вимог.

Вимоги мають різні джерела, такі як ділова людина, що замовляє продукт, менеджер зі збуту і фактичний користувач. У всіх цих людей є різні вимоги до продукту. Використовуючи відстеження вимог, реалізована в системі функція може бути простежена назад до людини або групі, яка замовляла її під час збору вимог. Ця особливість може, наприклад, використовуватися в процесі розробки для пріоритезації вимог, визначаючи, наскільки цінною є дана вимога для певного користувача.

Відстеження може також використовуватися після розгортання продукту. Наприклад, коли вивчення використання системи показує, що якась функція не використовується, можна визначити навіщо вона була потрібна спочатку.

### **Завдання управління вимогами**

На кожному етапі процесу розробки існують ключові методи і задачі пов'язані з управлінням вимогами. Для ілюстрації, розглянемо наприклад стандартний процес розробки з п'ятьма фазами: дослідженням, аналізом здійсненності, дизайном, розробкою та тестуванням і випуском.

Дослідження. Під час фази дослідження збираються перші три класи вимог від користувачів, бізнесу і команди розробників. У кожній області задають однакові питання: які цілі, які обмеження, які використовуються процеси та інструменти і так далі. Тільки коли ці вимоги добре зрозумілі, можна приступати до розробки функціональних вимог.

Тут необхідне застереження: незалежно від того, як сильно група намагається це зробити, вимоги не можуть бути повністю визначені на початку проекту. Деякі вимоги змінюються, або тому що вони просто не були знайдені спочатку, або тому що внутрішні чи зовнішні сили торкаються проекту в середині

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

циклу. Таким чином, учасники групи повинні спочатку погодитися, що головна умова успіху – гнучкість у мисленні та діях.

Результатом стадії дослідження є документ – специфікація вимог, схвалений усіма членами проекту. Пізніше, в процесі розробки, цей документ буде важливий для запобігання розповзанню меж проекту або непотрібних змін. Оскільки система розвивається, кожна нова функція відкриває світ нових можливостей, таким чином специфікація вимог прив'язує команду до оригінального бачення системи і дозволяє контрольоване обговорення змін.

У той час як багато організацій все ще використовують звичайні документи для керування вимогами, інші управляють своїми базовими вимогами, використовуючи програмні інструменти.

Ці інструменти керують вимогами використовуючи базу даних, і зазвичай мають функції автоматизації відстеження (наприклад, дозволяючи створювати зв'язки між батьківськими і дочірніми вимогами, або між тестами і вимогами), управління версіями, і управління змінами. Зазвичай такі інструментальні засоби містять функцію експорту, яка дозволяє створювати звичайний документ, експортуючи дані вимог.

### **Аналіз здійсненності**

На стадії аналізу здійсненності визначається вартість вимог. Для користувальницьких вимог поточна вартість роботи порівнюється з майбутньою вартістю встановленої системи. Задаються питання такі як: «Скільки нам зараз варті помилки введення даних?» Або, «Яка вартість втрати даних через помилки оператора пов'язаної з використанням інтерфейсом?». Фактично, потреба в новому інструменті часто розпізнається, коли подібні питання потрапляють до уваги людей, що займаються в організації фінансами.

Ділова вартість включає відповіді на такі питання як: «У якого відділу є бюджет на це?» «Який рівень повернення коштів від нового продукту на ринку?» «Який рівень скорочення внутрішніх витрат на навчання і підтримку, якщо ми зробимо нову, більш просту в використанні систему?»

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>53</b>

Технічна вартість пов'язана з вартістю розробки програмного забезпечення та апаратною вартістю. «Чи є у нас потрібні люди, щоб створити інструмент?» «Чи потребуємо ми нове устаткування для підтримки нової системи?»

Подібні питання дуже важливі. Група повинна з'ясувати, чи буде новий автоматизований інструмент мати достатню ефективність аби перенести частину тягаря користувачів на систему і зекономити час людей.

Ці питання також вказують на основну суть управління вимогами. Людина і інструмент формують систему, і це розуміння особливо важливе, якщо інструмент – комп'ютер або новий додаток на комп'ютері. Людський розум вкрай ефективний у паралельній обробці та інтерпретації тенденцій з недостатніми даними. Комп'ютерний процесор ефективний у послідовній обробці і точному математичному обчисленні. Основна мета управління вимогами для програмного проекту полягала б у тому, щоб гарантувати, що автоматизована робота призначена «правильному» процесору.

Наприклад, «не змушуйте людину пам'ятати, де вона знаходиться в системі. Примусьте інтерфейс завжди повідомляти про місцезнаходження людини в системі». Або «не змушуйте людини вводити ті ж самі дані в два екрани. Примусьте систему зберігати дані і заповнювати їх де необхідно автоматично». Результатом стадії аналізу здійсненності є бюджет і графік проекту.

Дизайн. Припускаючи, що вартість точно визначена і переваги, які будуть отримані, є досить великими, проект може перейти до стадії проектування.

На стадії дизайну основна діяльність управління вимогами полягає в тому, щоб перевіряти чи відповідають результати дизайну документу вимог, щоб упевнитися, що робота залишається в межах проекту.

І знову, гнучкість є ключем до успіху. Ось класичний приклад змін проекту, які відмінно працювали. Проектувальники Форда на початку 1980-х очікували, що ціни на бензин піднімуться до 3,18 дол за галон до кінця десятиліття. На середині процесу дизайну автомобіля Ford Taurus, ціни встановилися приблизно на рівні 1,50 дол за галон. Колектив дизайнерів вирішив,

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

що вони могли б створити більший, більш зручний, і більш потужний автомобіль, якщо б ціни на бензин залишилися низькими. Таким чином, вони перепроєктувати автомобіль. Коли новий автомобіль вийшов, він встановив загальнонаціональні рекорди продажів.

У більшості випадків, однак, відступ від оригінальних вимог до такої міри не працює. Таким чином документ вимог стає ключовим інструментом, який допомагає команді приймати рішення про зміни дизайну.

Розробка та тестування. На стадії розробки і тестування, основна діяльність управління вимогами – це гарантувати, що робота і ціна залишаються в межах графіка і бюджету, і що створюваний інструмент дійсно відповідає вимогам. Основним інструментом, використовуваним на цій стадії, є створення прототипу і ітераційне тестування. Для програмного додатка користувацький інтерфейс може бути створений на папері і перевірений з потенційними користувачами, в той час як створюється основа програми. Результати цих тестів записуються в керівництві по дизайну користувацького інтерфейсу і передаються колективу дизайнерів. Це економить їх час і робить їх завдання набагато простіше.

#### 4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм LOKI\_91. Механізм алгоритму LOKI\_91 подібний DES (рисунок 4.3). Блок даних розщеплюється на ліву й праву половини й проходить 16 раундів, що досить нагадує DES. У кожному раунді права половина спочатку піддається операції XOR із частиною ключа, а потім розширювальній перестановці (таблиця 4.1).

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

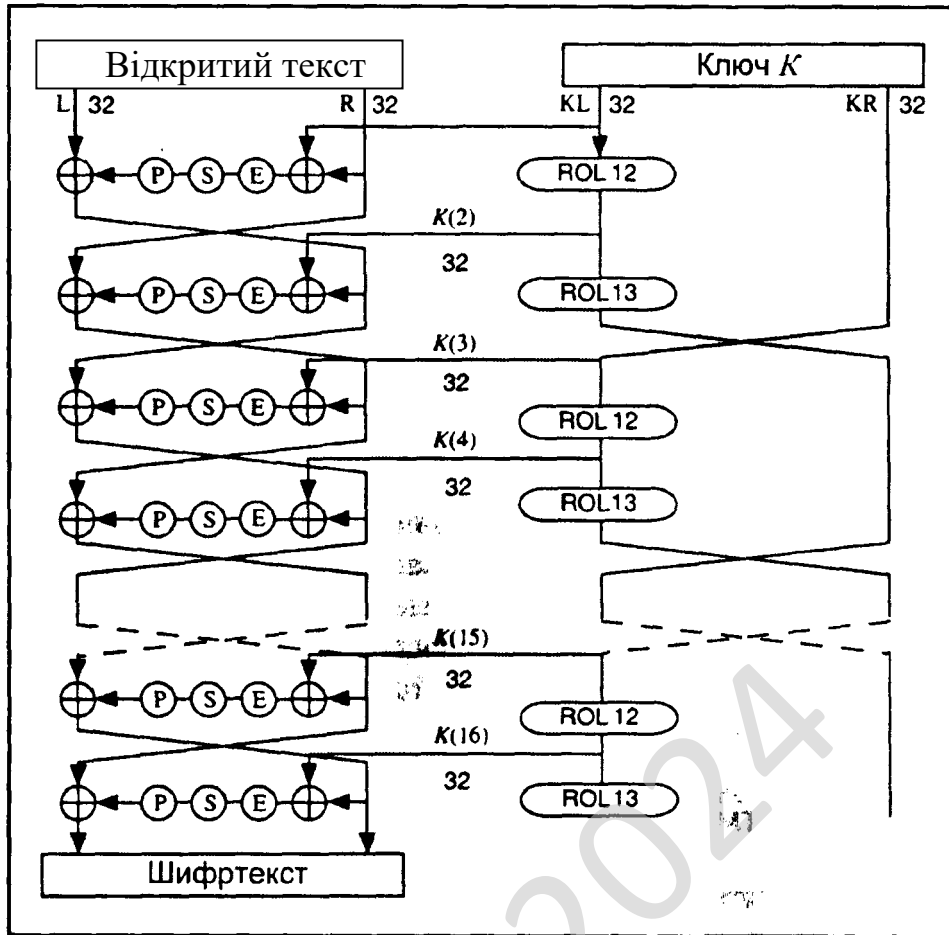


Рисунок 4.3 – Алгоритм LOKI91

Таблиця 4.1 – Перестановка з розширенням

4	3	2	1	32	31	30	29	28	27	26	25
28	27	26	25	24	23	22	21	20	19	18	17
20	19	18	17	16	15	14	13	12	11	10	9
12	11	10	9	8	7	6	5	4	3	2	1

48-бітовий вихід розділяється на чотири 12-бітових блоки. У кожному блоці виконується така підстановка з використанням S-блоку: береться кожний 12-бітовий вхід, 2 старших і 2 молодших біти використовуються для утворення номера r, а вісім внутрішніх біт утворюють номер c. Вихід S-блоку, O, має наступне значення:  $O(r,c) = (c + ((r*17) \oplus 0xff) \& 0xff)^{31} \bmod P_r$ .



## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ інтеграції мережних сервісів на основі технології ACI яке зображено на рисунку 5.1. Розрахунки виконуються через командний рядок з наступною передачею результатів у графічний інтерфейс. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи:

- Розділу вкладок з налаштуванням системи.
- Розділу обрання об'єкту.
- Вікно виведення результату роботи системи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ.

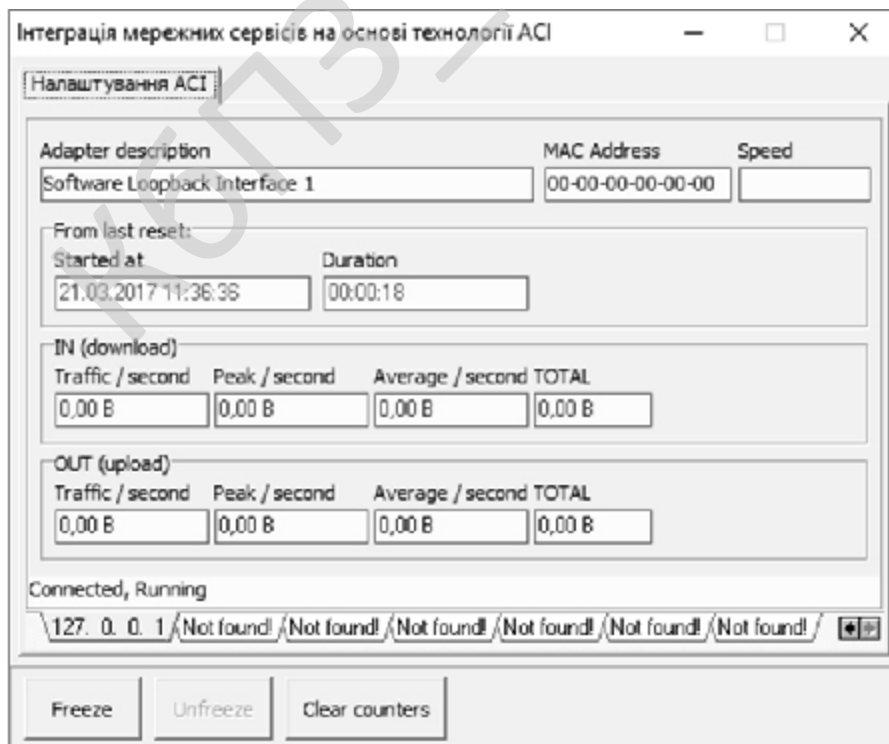


Рисунок 5.1 – Головне вікно ПЗ

Розроблена програма має дуже простий і інтуїтивно зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий.

Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

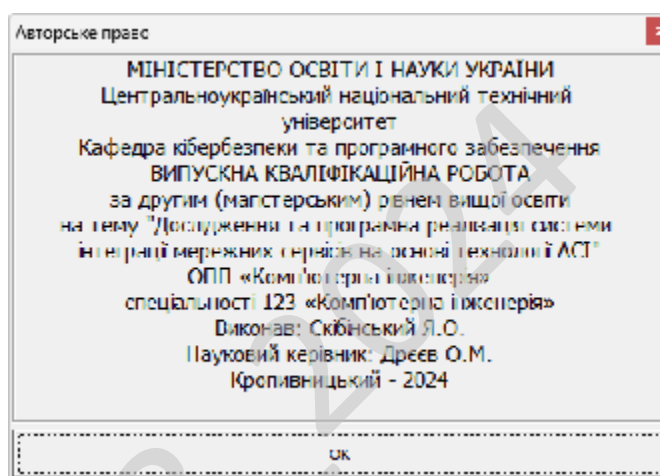


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити.

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Обновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>60</b>

Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів. Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів. Проводилось тестування форматом білої скриньки. Засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління. У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

– Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.

– Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.

– При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Обрано умови розповсюдження – Freeware.

Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів. Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення. Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються. Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

## 6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи інтеграції мережних сервісів на основі технології АСІ.

*Метою розробки є дослідження та програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ.*

*Об'єктом дослідження є процес інтеграції мережних сервісів на основі технології АСІ.*

*Предметом дослідження є методи інтеграції мережних сервісів на основі технології АСІ.*

*Методи дослідження базуються на методах теорії побудови комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.*

**Наукова новизна отриманих результатів.** У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод інтеграції мережних сервісів на основі технології АСІ.
- Розроблено вітчизняний продукт інтеграції мережних сервісів на основі технології АСІ, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

## 7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

### 7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи інтеграції мережних сервісів на основі технології ACI (Application Centric Infrastructure) можуть зацікавити наступні категорії, представлені на рисунку 7.1.

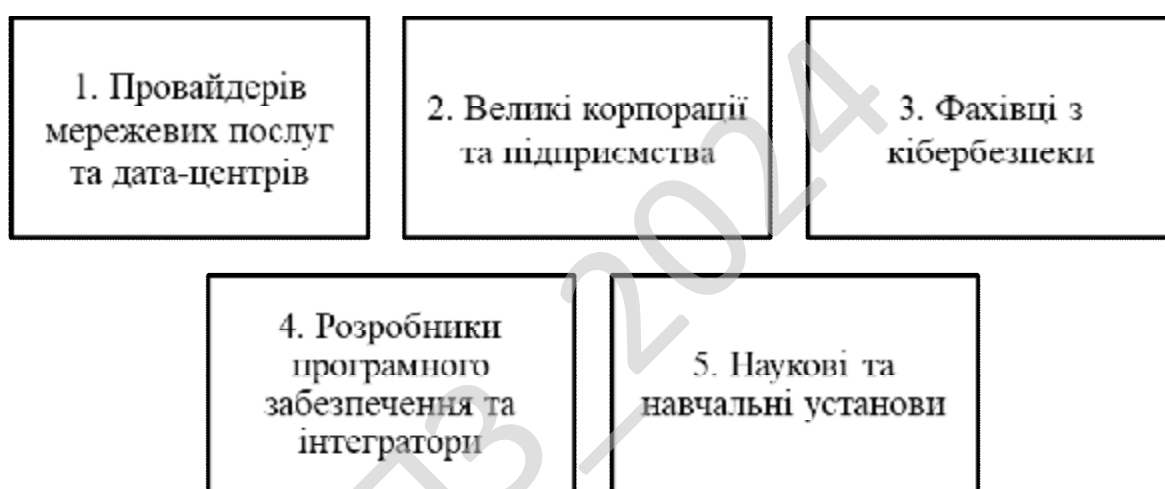


Рисунок 7.1 – Цільова аудиторія

Провайдерів мережних послуг та дата-центрів – це мережеві інженери та архітектори (результати можуть допомогти їм краще розуміти та інтегрувати ACI для оптимізації інфраструктури та забезпечення високої продуктивності мережі), адміністратори дата-центрів (ACI може спростити управління та масштабування їх інфраструктури, забезпечуючи автоматизацію та централізоване керування мережами).

Великі корпорації та підприємства зацікавлені в підвищенні ефективності роботи мережевої інфраструктури, що дозволить їм забезпечити більшу надійність і безпеку даних і полегшить роботу їх ІТ-департаментів та системних

адміністраторів. Керівниками IT-проектів інформація про можливості АСІ може бути корисною для обґрунтування рішень щодо інвестування в нову технологію.

Фахівці з кібербезпеки – аналітики та архітектори безпеки (АСІ надає можливості для підвищення рівня безпеки за допомогою сегментації мережі та контролю доступу, що важливо для безпеки інформації в сучасних мережах).

Розробники програмного забезпечення та інтегратори зацікавлені теж в результатах дослідження системи інтеграції мережних сервісів на основі технології АСІ. Адже DevOps-спеціалістам та фахівцям з автоматизації АСІ дозволяє автоматизувати управління мережами, що є корисним для реалізації CI/CD практик в організаціях.: Розробка інтеграцій на основі АСІ відкриває інтеграторам мережеских рішень нові можливості для впровадження рішень у різних галузях.

Для дослідників наукових та навчальних установ результати дослідження та програмної реалізації системи інтеграції мережеских сервісів на основі технології АСІ можуть бути корисними для подальших наукових досліджень в галузі оптимізації мережеских інфраструктур. Здобувачам та викладачам IT-дисциплін це дає можливість отримати практичний досвід та знання щодо використання сучасних технологій інтеграції мережеских сервісів.

Таким чином, проєкт має значний потенціал для привернення уваги широкого кола спеціалістів у сфері IT, що прагнуть оптимізувати мережеву інфраструктуру і забезпечити її відповідність сучасним вимогам.

## **7.2 Оцінка привабливості шляхом застосування методів експертних оцінок**

Для оцінки привабливості програмної реалізації системи інтеграції мережеских сервісів на основі технології АСІ варто скористатися методом експертних оцінок.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

Щоб експертна оцінка була об'єктивною, спочатку необхідно визначити ключові критерії оцінювання. Критерії для інтеграції мережевих сервісів на основі АСІ представлено на рисунку 7.2.

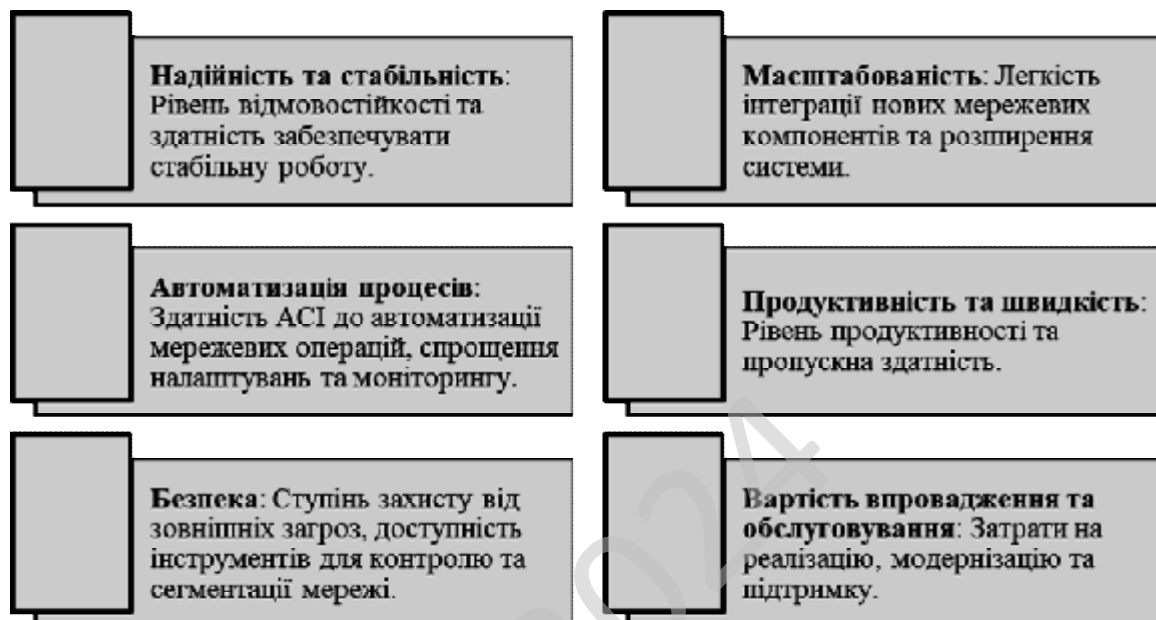


Рисунок 7.2 – Критерії експертної оцінки

Далі важливо зібрати команду з різнопрофільних фахівців, які можуть об'єктивно оцінити систему. Експерти мають включати: мережевих інженерів, фахівців з кібербезпеки, ІТ-архітекторів та адміністраторів, керівників проєктів у сфері ІТ.

За допомогою анкети або опитування кожен експерт оцінює зазначені критерії за шкалою від 1 до 5, де: 1 – дуже погано, 2 – погано, 3 – задовільно, 4 – добре, 5 – відмінно.

Результати оцінок від трьох експертів зведено в таблицю 7.1.

Таблиця 7.1 – Зведені результати експертних оцінок

Критерій	Експерт 1	Експерт 2	Експерт 3	Середня оцінка
Надійність	5	4	4	4.33
Масштабованість	4	5	4	4.33
Автоматизація	5	5	5	5.00
Продуктивність	4	4	4	4.00
Безпека	5	4	5	4.67

Для обчислення загальної привабливості системи кожному критерію присвоюються вагові коефіцієнти, що відображають їхню важливість: надійність: 20%, масштабованість: 20%, автоматизація: 15%, продуктивність: 15%, безпека: 20%, вартість: 10%.

Загальна оцінка розраховується за формулою зваженого середнього:  $(0.20 \cdot 4.33) + (0.20 \cdot 4.33) + (0.15 \cdot 5.00) + (0.15 \cdot 4.00) + (0.20 \cdot 4.67) + (0.10 \cdot 3.33)$ . Загальна оцінка привабливості для проєкту інтеграції мережевих сервісів на основі технології АСІ становить 4.35. Це свідчить про високий рівень привабливості проєкту, враховуючи ключові критерії, такі як надійність, автоматизація та безпека.

Отримані результати дають змогу визначити загальну привабливість проєкту для впровадження. Якщо загальна оцінка є високою, це сигналізує про доцільність реалізації проєкту, оскільки він відповідає ключовим вимогам, таким як надійність, автоматизація та безпека. Цей підхід дозволяє створити об'єктивну оцінку на основі знань та досвіду експертів, що допоможе ухвалити обґрунтоване рішення щодо впровадження технології АСІ.



ТСО також сприяє ухваленню обґрунтованих рішень, даючи змогу зіставити потенційні витрати з вигодами, які технологія АСІ може забезпечити для компанії, наприклад, підвищену безпеку та зручність управління мережею.

#### **7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості**

Економічна ефективність від впровадження системи інтеграції мережевих сервісів на основі технології АСІ може бути оцінена через різні показники, зокрема, за рахунок оптимізації витрат, збільшення продуктивності мережі та зменшення простоїв. Зведені дані наведемо в таблиці 7.2.

Економічна ефективність від впровадження системи інтеграції мережевих сервісів на основі технології АСІ проявляється у чистій економії \$50,000 за 5 років та коефіцієнті окупності 33.33%. Це показує, що інвестиція є вигідною у довгостроковій перспективі та дозволяє компанії значно зменшити витрати на мережеве управління, мінімізувати простой та підвищити продуктивність.

#### **7.5 Пропозиція алгоритму просування проєкту розробки ПЗ**

Для просування проєкту програмної реалізації системи інтеграції мережевих сервісів на основі технології АСІ можемо запропонувати алгоритм, який орієнтований на привернення уваги цільової аудиторії, демонстрацію переваг і формування попиту (рисунок 7.4).

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

Таблиця 7.2 – Основні показники впровадження проєкту

Вхідні дані:

**Початкові витрати на впровадження АСІ – \$150,000.**

**Зменшення операційних витрат завдяки автоматизації (20% від поточних витрат на управління мережею):** до впровадження витрати складала \$50,000 на рік.

**Скорочення простоїв** завдяки стабільності та оптимізації мережі (на 25%). До впровадження щорічні втрати від простоїв складала \$40,000.

**Підвищення продуктивності персоналу:** скорочення ручної роботи на 30%, економія \$20,000 на рік.

**Окупність проєкту (ROI)** розраховується на основі економії за 5 років.

**Економія на операційних витратах:**  $\$50,000 * 20\% = \$10,000$

**Зниження витрат від простоїв:**  $\$40,000 * 25\% = \$10,000$

**Економія завдяки підвищенню продуктивності персоналу:** \$20,000

Загальна економія на рік:

Економія на рік =  $10,000 + 10,000 + 20,000 = 40,000$

Оцінка окупності та економічної ефективності:

**Загальна економія за 5 років:** Економія за 5 років =  $40,000 \times 5 = 200,000$

**Чиста економія (враховуючи початкові витрати):**

Чиста економія =  $200,000 - 150,000 = 50,000$

**Коефіцієнт окупності (ROI):**

$ROI = (\text{Чиста економія} / \text{Початкові витрати}) \times 100\% = (50,000 / 150,000) \times 100\% = 33.33\%$

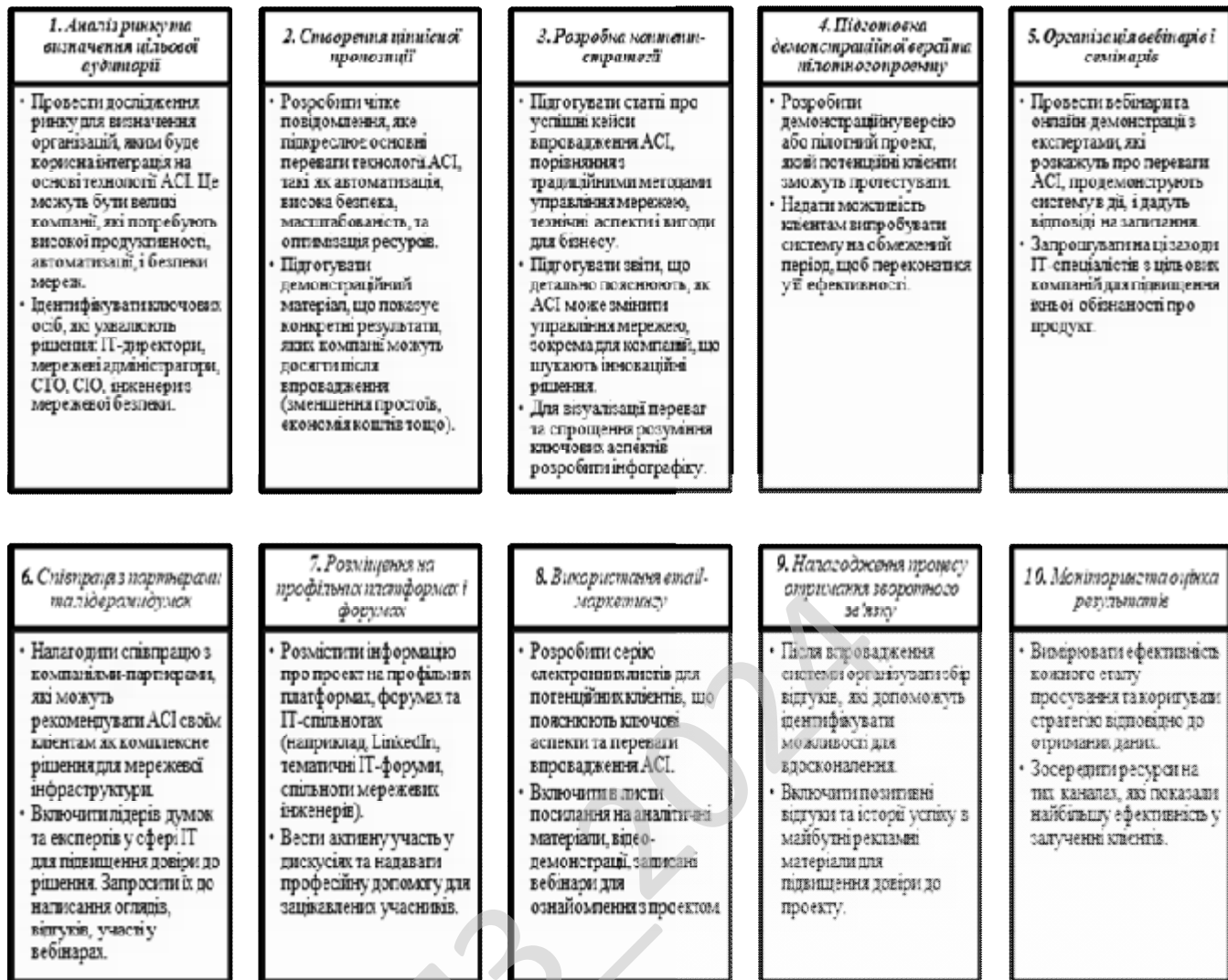


Рисунок 7.4 – Алгоритм просування проєкту

Цей алгоритм дозволить комплексно підійти до просування проєкту на ринку, забезпечити ефективне інформування цільової аудиторії та формування попиту на систему інтеграції мережевих сервісів на основі технології АСІ.

## 7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту та шляхів реалізації проєкту системи інтеграції мережевих сервісів на основі технології АСІ важливо вибудувати стратегічний підхід до вибору та розвитку каналів, щоб максимально охопити цільову аудиторію та спростити процес впровадження (рисунок 7.5).

- Прямий продаж:** Впровадження прямих продажів для великих корпоративних клієнтів, що потребують індивідуальних консультацій, адаптації та підтримки.
- Партнерські канали:** Співпраця з IT-консультантами, провайдерами керування послуг (MSP) та реселлерами, які можуть пропонувати рішення на базі АСІ своїм клієнтам як частину комплексних послуг. Це допоможе залучити менші компанії та середній бізнес.
- Онлайн-продаж:** Розробка порталу або сайту з можливістю онлайн-консультацій, ознайомлення з демонстраційними матеріалами та поданням заявки на впровадження.
- Створення програми для партнерів:** Запропонуйте партнерам програми сертифікації з навчанням особливостей впровадження АСІ, підтримки та технічного обслуговування.
- Стимулювання партнерів:** Використовуйте спеціальні знижки, бонуси за кількість залучених клієнтів, та підтримку у вигляді рекламних матеріалів та маркетингової підтримки.
- Інформаційна підтримка:** Регулярно організовуйте вебінари та консультації для партнерів, щоб вони мали актуальну інформацію про продукт та змогли краще пояснити його переваги клієнтам.
- Маркетинговий веб-сайт:** Оптимізуйте сайт з високим рівнем навчання АСІ, де буде представлено усі переваги, специфікації, приклади успішних впроваджень, та пропозиції щодо початкових консультацій.
- SEO та контент-маркетинг:** Створіть оптимізований контент (статті, дослідження, аналітику) для залучення органічного трафіку. Наприклад, кейси ефективності АСІ у певних галузях, як-от фінансовий сектор, де потрібна висока безпека і масштабованість.
- Реклама в соціальних мережах:** Використовуйте таргетовану рекламу в LinkedIn, Facebook, та інших платформах для досягнення IT-спеціалістів та керівників, які відповідальні за мережеву інфраструктуру.
- Сегментація ринку:** Визначте ключові галузі, такі як фінансові установи, медицина, освіта та телекомунікації, де потрібна висока продуктивність та безпека.
- Адаптація пропозицій для галузей:** Розробіть маркетингові та технічні матеріали, що підкреслюють переваги АСІ саме для цих секторів (наприклад, захист персональних даних в медичних установах).
- Сервісна підтримка:** Забезпечте безперервну технічну підтримку для нових клієнтів, щоб спростити впровадження та зменшити перешкоди для користувачів.
- Підготовка матеріалів та навчання:** Створіть документацію, відеоінструкції та курси для технічного персоналу клієнта, що сприятиме швидшому освоєнню системи.
- Моніторинг ефективності каналів:** Відстежуйте ефективність кожного каналу збуту, зокрема кількість нових клієнтів, витрати на залучення та задоволеність клієнтів.
- Оцінка та коригування стратегій:** Проводьте регулярні аналізи, щоб визначити найефективніші канали та спрямовувати ресурси на їхній розвиток.
- Демонстраційний доступ:** Запропонуйте компаніям доступ до демо-версії або короткотермінову пілотну програму, щоб вони могли переконатися в перевагах рішення до повного впровадження.
- Відгуки та кейс-стаді:** Використовуйте успішні кейси як маркетинговий інструмент, публікуйте їх на сайті та розповсюджуйте через партнерські канали для залучення нових клієнтів.

Рисунок 7.5 – Ключові фактори успіху проекту

Оптимізація збутових каналів для проєкту на базі АСІ передбачає багаторівневий підхід, який включає партнерські програми, таргетовану рекламу та підтримку клієнтів на кожному етапі, від перших консультацій до технічної підтримки. Це дозволить підвищити рівень довіри до продукту, забезпечити стабільний потік клієнтів та ефективно просувати систему на ринку.

### 7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху для проєкту програмної реалізації системи інтеграції мережних сервісів на основі технології АСІ (представлено на рисунку 7.6).

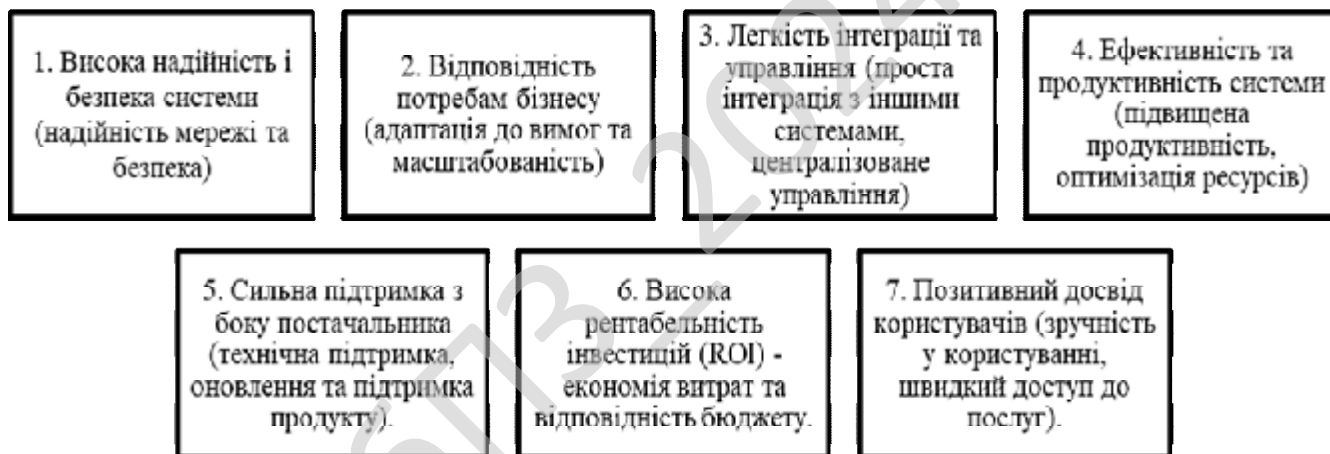


Рисунок 7.6 – Ключові фактори успіху проєкту

Ці фактори забезпечують успіх проєкту інтеграції мережних сервісів на базі технології АСІ, підвищуючи його привабливість, економічну ефективність та конкурентоспроможність на ринку, а також знижуючи загрози та ризики, пов'язані з управлінням мережею.

## 8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

### 8.1 Вступ

Аналізуючи умови працівників ІТ-сфери, на перший погляд, може здатися, що працівники сфери інформаційних технологій не схильні до ризиків на виробництві, та якщо більш глибоко розглянути умови і специфіку праці фахівців сфері ІТ-індустрії, можна виявити ряд факторів які будуть мати негативний вплив на стан охорони праці, та на самого ІТ-фахівця зокрема. Сюди можна віднести як невідповідність освітлення, так і високий рівень шуму, що негативно позначатимуться як на емоційному так і на фізичному стані фахівця, призводитимуть до зниження ефективності праці та виробничих травм. Також, важливим моментом охорони праці ІТ-фахівця є врахування його психологічних можливостей (швидкість реакції, особливості пам'яті та уваги, емоційний стан, тощо). Для того, щоб забезпечити ефективну роботу ІТ-фахівця, потрібно враховувати та максимально компенсувати такі негативні фактори як: надмірне нервово-емоційне навантаження, довготривалі статичні перевантаження, обмежена рухова активність. Всі ці чинники призводять до різноманітних відхилень у стані здоров'я, зокрема до перевтоми, зниження фізичної та розумової працездатності, неврозів, захворювань серцево-судинної системи тощо. Метою даного розділу є огляд конкретних умов праці спеціаліста у сфері ІТ-індустрії. Завданнями для даного розділу є: аналіз умов праці на робочому місці фахівця ІТ-індустрії, розробка конкретних рекомендацій щодо покращення умов праці фахівців ІТ-індустрії, огляд пожежної безпеки на ІТ-підприємстві та розрахунок системи загального штучного освітлення виробничого приміщення де працюють ІТ – фахівці.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

## 8.2 Аналіз умов праці на робочому місці ІТ-фахівця

На робочому місці ІТ-фахівця (або програміста) виникають небезпечні та шкідливі для безпечної життєдіяльності фактори:

- підвищений рівень шуму;
- недостатній рівень освітленості;
- шкідливі речовини;
- підвищений рівень електромагнітних випромінювань радіочастот;
- висока напруга електричної мережі;
- статична електрика та інші.

Робота продовжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, вивчення умов праці на робочому місці програміста є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів. Робоче місце, добре пристосоване до трудової діяльності інженера, правильно і доцільно організоване, щодо простору, форми, розміру забезпечує йому зручне положення при роботі і високу продуктивність праці при найменшому фізичному і психічному напруженні. Нормування параметрів проводиться в залежності від періоду року та категорії важкості виконуваних робіт. Для постійних робочих місць, якими є робочі місця ІТ-фахівців, встановлені оптимальні параметри мікроклімату, а за неможливості їх дотримання використовують допустимі параметри. Робота ІТ-фахівця за важкістю відноситься до Іа (роботи, що виконуються сидячи і не потребують фізичного напруження) та Іб (роботи, що виконуються сидячи, стоячи або пов'язані з ходінням та супроводжуються деяким фізичним напруженням) категорій. В таблиці 8.1. наведені оптимальні параметри мікроклімату в приміщеннях.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75



Створення сприятливих умов праці і правильне естетичне оформлення робочих місць на виробництві має велике значення як для полегшення праці, так і для підвищення його привабливості, позитивно впливає на продуктивність праці. Забарвлення приміщень і меблів повинні сприяти створенню сприятливих умов для зорового сприйняття, гарного настрою. У службових приміщеннях, у яких виконується одноманітна розумова робота, що вимагає значної нервової напруги і великого зосередження, забарвлення повинно бути спокійних тонів – малонасичені відтінки холодного зеленого або блакитного кольорів.

При розробці оптимальних умов праці програміста необхідно враховувати освітленість. Раціональне освітлення робочого місця є одним з найважливіших факторів, що впливають на ефективність трудової діяльності людини, що попереджають травматизм і професійні захворювання. Правильно організоване освітлення створює сприятливі умови праці, підвищує працездатність і продуктивність праці. Освітлення на робочому місці програміста повинно бути таким, щоб працівник міг без напруги зору виконувати свою роботу. Стомлюваність органів зору залежить від ряду причин: недостатність освітленості; надмірна освітленість; неправильний напрям світла. Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань. [2]

### **8.3 Пропозиції щодо підвищення працездатності ІТ – фахівців**

Поява та впровадження нових інформаційно-комунікаційних технологій зумовлює необхідність подальшого вдосконалення охорони праці фахівців ІТ-індустрії. Все це потребує розробки нових нормативно-правових актів з регламентації праці та відпочинку фахівців ІТ-індустрії і стандартів підприємств,

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

центрів комп'ютерної техніки, центрів інформаційних технологій, сучасних комп'ютерних класів. Для підвищення розумової працездатності то зорової роботи повинна здійснюватися ергономічна оптимізація в рамках системи «оператор-термінал», яка сприятиме результативній фізичній та інтелектуальній працездатності і відновленню психосоматичного здоров'я фахівців ІТ-індустрії. Всі наведені заходи щодо вдосконалення охорони праці фахівців ІТ-індустрії повинні контролюватися службою охорони праці та комісією з охорони праці підприємства. Особливе значення у соціальному захисті цієї категорії працівників належить прийняття комплексного договору, який може забезпечити фахівців додатковими пільгами та компенсаціями.

Для більшого розуміння, пропозиції щодо підвищення працездатності ІТ-фахівців, розіб'ємо на декілька категорій:

1 Середовище і розпорядок праці. Для мінімізації негативних ефектів, що пов'язані з перевтомленням ІТ-фахівців, потрібно чітко прописати і реалізувати графік періодів праці-відпочинку, щоб фахівець міг можливість переключити увагу, дати можливість відпочити очам, мозку, елементарно, встати розім'яти ноги. Також потрібно зробити максимально комфортними умови мікроклімату у офісному приміщенні, де працюють ІТ-фахівці. Мається на увазі встановлення і експлуатація, коли виникає необхідність, кондиціонерів, опалення, та системи вентиляції, задля попередження перегрівання, переохолодження ІТ-фахівців, і подальшої неможливості ними виконувати свої функції. Також, за можливості, нами пропонується введення практики віддаленої праці ІТ-фахівцями, якщо роботодавець не може забезпечити оптимальні і безпечні умови в офісному приміщенні, або якщо фахівця вони не влаштовують із певних причин.

2 Фізичні і психоемоційні чинники. Першим і найважливішим чинником, що впливає на працездатність ІТ-фахівців є робоче місце, і саме тому, роботодавець має забезпечити максимальний його комфорт і безпеку. Гарантією цих факторів може слугувати сертифікація меблів, що використовуються на підприємстві ІТ-галузі. Тому нами пропонується закупівля тільки меблів , які

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

пошли сертифікацію на відповідність. Під психоемоційними чинниками ми розуміємо гарне самопочуття фахівців, позитивний настрій, гарний психологічний клімат у колективі, тощо. Задля того, щоб психоемоційні чинники мали максимально позитивний ефект, керівництву слід поводити заходи, які сприятимуть укріпленню і покращенню міжособистісних стосунків у колективі, таких як психологічні тренінги, тимбілдінг, спортивні змагання і естафети. Також, сюди можна віднести розробку і впровадження системи мотивації працівників, як фінансової, так моральної і адміністративної.

#### **8.4 Розрахунок системи загального штучного освітлення виробничого приміщення де працюють IT-фахівці**

Приміщення з ПК повинні мати природне і штучне освітлення, яке відповідало б вимогам ДБН В.2.5-28-2006 «Природне і штучне освітлення», ДСАНПН 3.3.2.007-98 «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин». Приміщення для роботи із ПЕОМ повинні мати природне й штучне освітлення. Віконні прорізи повинні бути орієнтовані на північ або на північний схід, забезпечувати коефіцієнт природної освітленості (К.П.О.) не менш 1,5% і мати жалюзі або штори. Віконні прорізи повинні мати регульовані пристрої для відкривання, а також жалюзі, завіски, зовнішні козирки тощо. Приміщення із ПЕОМ повинні бути обладнані системою загального рівномірного освітлення. У виробничих і адміністративно-суспільних 130 приміщеннях, де переважно ведеться робота з документами, допускається комбінована система штучного освітлення. Штучне освітлення має здійснюватися системою загального рівномірного освітлення, яка включає суцільні або такі, що перериваються лінії світильників, розташованих збоку робочих місць (переважно ліворуч), паралельно лінії зору користувачів ПК. Світильники повинні мати розсіювачі світла. У світильниках місцевого освітлення можна використовувати лампи накаливання. При розміщенні ПК по

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79



системи відбитого освітлення не повинна перевищувати 200 кд/м<sup>2</sup>. Величина коефіцієнта пульсації освітленості не повинна перевищувати 5%. Що стосується розподілу яскравості в полі зору працюючих за дисплеями ПК, то відношення значень яскравості робочих поверхонь не повинно перевищувати 3:1

Проведемо розрахунок штучного освітлення за методом коефіцієнта використання світлового потоку для приміщення ширина якого складає 6 м, довжина – 7 м, висота – 2,9 м. У зазначеному приміщенні працює 7 людей.

Для того, щоб визначити потрібну кількість світильників, які повинні забезпечити нормований рівень освітленості, визначимо світловий потік, що падає на робочу поверхню за формулою [1]:

$$F = E \cdot S \cdot K \cdot Z / n,$$

де:  $F$  – світловий потік, що розраховується, Лм;

$E$  – нормована мінімальна освітленість, Лк;  $E = 300$  Лк;

$S$  – площа освітлюваного приміщення (у нашому випадку  $S = 6 \times 7 = 42$  м<sup>2</sup>);

$K$  – коефіцієнт запасу, що враховує зменшення світлового потоку лампи в результаті забруднення світильників в процесі експлуатації (його значення залежить від типу приміщення і характеру робіт, що проводяться в ньому, в нашому випадку  $K = 1,5$ );

$Z$  – відношення середньої освітленості до мінімальної (зазвичай приймається рівним 1.1... 1.2, в нашому випадку  $Z = 1,1$ );

$n$  – коефіцієнт використання світлового потоку, (відношення світлового потоку, що падає на розрахункову поверхню, до сумарного потоку всіх ламп і обчислюється в долях одиниці [8]; залежить від характеристик світильника, розмірів приміщення, забарвлення стін і стелі, що характеризуються коефіцієнтами відбиття від стін ( $\rho_{\text{стін}}$ ) і стелі ( $\rho_{\text{стелі}}$ ), значення коефіцієнтів дорівнюють  $\rho_{\text{стін}} = 50\%$  і  $\rho_{\text{стелі}} = 50\%$ .

Обчислимо індекс приміщення за формулою:

$$i = S / (h \cdot (A + B)),$$

де:  $S$  – площа приміщення,  $S = 42$  м<sup>2</sup>;

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

$h$  – розрахункова висота підвісу,  $h = 2,9$  м (співпадає з висотою стелі, т.я. лампи освітлення закріплюються на стелі);

$A$  – ширина приміщення,  $A = 6$  м;

$B$  – довжина приміщення,  $B = 7$  м.

Підставимо всі значення у формулу та визначимо індекса приміщення:  
 $i=1,4$ .

Знаючи індекс приміщення, за знаходимо  $n = 0,29$  (з табличних даних коефіцієнтів використання світлового потоку ( $n$ ) світильників з відповідним типом ламп) [8]. Підставимо всі значення у формулу, визначимо світловий потік:  $F=71689$  Лм.

Для розрахунку дудемо використовувати світлодіодні стельові панелі Delux LED Panel 41 44Вт., світловий потік яких  $F_n = 3600$  Лм.

Число ламп визначається по формулі:

$$N=F/F_n$$

де:  $F$  – світловий потік,

$F_n$  – світловий потік однієї лампи.

Підставимо всі значення у формулу та визначимо індекса приміщення:

$$N= 71689 / 3600=19,9 \text{ шт.}$$

Приймаємо необхідну кількість світлодіодних світильників 20 шт.

### **Висновки до розділу**

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва вцілому.

З цих міркувань було здійснено аналіз умов праці на робочому місці ІТ-фахівця, призначеного для праці програмістів, проведено розгляд небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи. Виконано розрахунок штучного освітлення, як одного з ключових факторів впливу на працездатність та здоров'я програміста. Розроблено заходи з умов поліпшення охорони праці.

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

## 9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи інтеграції мережних сервісів на основі технології АСІ.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів інтеграції мережних сервісів на основі технології АСІ.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем інтеграції мережних сервісів на основі технології АСІ.

– Досліджена система інтеграції мережних сервісів на основі технології АСІ.

– На основі отриманих результатів досліджень створена програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання інтеграції мережних сервісів на основі технології АСІ.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>83</b>

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм LOKI\_91.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>84</b>

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Скiбiнський Я.О. Дослiдження та програмна реалiзацiя системи iнтеграцiї мережних сервiсiв на основi технологiї АСI // Збiрник праць молодих науковцiв ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
3. Ramon Nastase «Computer Networking: The Beginner’s guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
4. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
5. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.
6. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
7. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
8. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.
9. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

10. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». *Communications in Computer and Information Science*, 2021, vol 1486. Springer, Cham. pp 169-184.

11. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». *CEUR Workshop Proceedings*. Volume 2740, 2020, Pages 102-114.

12. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346.

13. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

14. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587.

15. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

16. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 366-379.

17. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated

with the Markov chains», *CEUR Workshop Proceedings* Volume 2608, 2020, Pages 633-645.

18. Smirnov O., Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». *International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019*; Odessa; Ukraine; 9-13 September 2019. P.22-28.

19. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

20. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

21. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyzy, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

22. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019.

23. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019*, P. 395-399.

24. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in

Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.

25. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

26. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

27. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», *Telecommunications and Radio Engineering*. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

28. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

29. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

30. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

31. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного

захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

32. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки»*, № 2 (307). С. 46-52. 2022.

33. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 1(67). С. 84-89.

34. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи*. 2021. Т. 5, № 4. С. 79-95

35. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки*. №4. С. 103-110. 2020.

36. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», *Кібербезпека: освіта, наука, техніка*. № 3(7). С. 43-62. 2020.

37. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

					ВКРМ-123.24.0041.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

38. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у *Кібербезпека та інформаційні технології: монографія*. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

39. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». *Центральноукраїнський науковий вісник. Технічні науки*. № 2(33). с. 161-172, 2019.

40. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

41. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

42. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

43. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. *Центральноукраїнський науковий вісник. Технічні науки*. № 1(32). с. 173-183, 2019.

44. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. *Центральноукраїнський науковий вісник. Технічні науки*. № 1(32). с. 184-194, 2019.

45. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

46. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

47. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

48. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

49. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". – Випуск 3 (140). – Х.: ХУПС – 2016. – С. 36-39.

50. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

51. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

52. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

					<b>ВКРМ-123.24.0041.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

Додаток А  
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					<b>ВКРМ-123.24.0041.00.00.ТЗ</b>			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Скібінський Я.О.				<i>Дослідження та програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ</i>	Літ.	Аркуш	Аркушів
Перевірів	Дресв О.М.					М	1	6
Н. Контр.	Коваленко А.С.				<b>ЦНТУ КІ-23М</b>			
Затв.	Смірнов О.А.							

# 1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи інтеграції мережних сервісів на основі технології АСІ.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 19-13 від 07.08.2024 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи інтеграції мережних сервісів на основі технології АСІ.

## 4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.24.0041.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи інтеграції мережних сервісів на основі технології АСІ;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

## 5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

## 5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					<b>ВКРМ-123.24.0041.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

## 5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

## 5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

### 5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

### 5.8.2 Мова програмування

Середовище Python.

					ВКРМ-123.24.0041.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

## 7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

## 8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинен бути розглянутий аналіз умов праці на робочому місці ІТ-фахівця.

					ВКРМ-123.24.0041.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

## 9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 91 аркуш.

## 10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 16.12.2024 р.

					<b>ВКРМ-123.24.0041.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

**ЗАТВЕРДЖУЮ**

Керівник випускної кваліфікаційної роботи за  
другим (магістерським) рівнем вищої освіти

\_\_\_\_\_ Дреєв О.М.

*Дослідження та програмна реалізація  
системи інтеграції мережних сервісів на основі технології АСІ*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 19

Літера: РП

Кропивницький – 2024 року

**Основна програма ACI\_Network\_Project**

```
import acitoolkit.acitoolkit as aci
import json
import requests
import time

class ACINetworkIntegration:
    def __init__(self, url, username, password):
        self.session = aci.Session(url, username, password)
        self.session.login()

    def create_tenant(self, tenant_name):
        tenant = aci.Tenant(tenant_name)
        return tenant

    def create_application_profile(self, tenant, app_name):
        app_profile = aci.AppProfile(app_name, tenant)
        return app_profile

    def create_epg(self, app_profile, epg_name):
        epg = aci.EPG(epg_name, app_profile)
        return epg

    def create_contract(self, tenant, contract_name):
        contract = aci.Contract(contract_name, tenant)
        return contract

    def apply_contract_to_epg(self, epg, contract):
        epg.provide(contract)
        epg.consume(contract)

    def create_filter(self, tenant, filter_name):
        aci_filter = aci.Filter(filter_name, tenant)
        return aci_filter

    def add_filter_entry(self, aci_filter, entry_name, protocol, port_start,
port_end):
        filter_entry = aci.FilterEntry(entry_name,
                                        aci_filter,
                                        protocol=protocol,
                                        dFromPort=port_start,
                                        dToPort=port_end)

    def configure_contract_with_filter(self, contract, aci_filter):
        contract_subject = aci.ContractSubject('allow-http', contract)
        contract_subject.add_filter(aci_filter)

    def create_bridge_domain(self, tenant, bd_name):
        bd = aci.BridgeDomain(bd_name, tenant)
        return bd
```

```

def create_subnet(self, bd, subnet_ip):
    subnet = aci.Subnet('subnet1', bd)
    subnet.addr = subnet_ip
    return subnet

def create_vrf(self, tenant, vrf_name):
    vrf = aci.Context(vrf_name, tenant)
    return vrf

def bind_bd_to_vrf(self, bd, vrf):
    bd.add_context(vrf)

def deploy_configuration(self, tenant):
    self.session.push_to_apic(tenant.get_url(), data=tenant.get_json())

def calculate_network_performance(network_data):
    total_bandwidth = sum([data['bandwidth'] for data in network_data])
    total_latency = sum([data['latency'] for data in network_data])
    total_throughput = total_bandwidth - total_latency
    return total_bandwidth, total_latency, total_throughput

def optimize_network_configuration(bandwidth, latency):
    if bandwidth > 1000 and latency < 50:
        return "Optimized"
    elif bandwidth < 1000 and latency > 100:
        return "Need Optimization"
    else:
        return "Moderate Performance"

def generate_network_topology():
    topology = {
        "nodes": [
            {"id": "1", "name": "Router1", "type": "router"},
            {"id": "2", "name": "Switch1", "type": "switch"},
            {"id": "3", "name": "Firewall", "type": "firewall"}
        ],
        "links": [
            {"source": "1", "target": "2", "bandwidth": 1000},
            {"source": "2", "target": "3", "bandwidth": 500}
        ]
    }
    return topology

def monitor_traffic(endpoint, interval=5):
    while True:
        try:
            response = requests.get(f"http://{endpoint}/traffic")
            if response.status_code == 200:
                data = response.json()
                print(f"Traffic data: {data}")
                time.sleep(interval)
        except requests.ConnectionError:

```

```

        print("Unable to connect to traffic endpoint")

def calculate_resource_allocation():
    resources = {
        "CPU": 80, # %
        "Memory": 75, # %
        "Disk": 60 # %
    }
    total_allocation = sum(resources.values())
    return resources, total_allocation

if __name__ == "__main__":
    aci_url = "https://apic.example.com"
    aci_username = "admin"
    aci_password = "password"

    aci_integration = ACINetworkIntegration(aci_url, aci_username, aci_password)

    tenant = aci_integration.create_tenant("EnterpriseNetwork")
    app_profile = aci_integration.create_application_profile(tenant,
"EnterpriseApp")

    web_epg = aci_integration.create_epg(app_profile, "Web_EPG")
    db_epg = aci_integration.create_epg(app_profile, "DB_EPG")

    contract = aci_integration.create_contract(tenant, "WebToDBContract")
    aci_integration.apply_contract_to_epg(web_epg, contract)
    aci_integration.apply_contract_to_epg(db_epg, contract)

    aci_filter = aci_integration.create_filter(tenant, "WebToDBFilter")
    aci_integration.add_filter_entry(aci_filter, "http-entry", "tcp", "80",
"80")
    aci_integration.configure_contract_with_filter(contract, aci_filter)

    bd = aci_integration.create_bridge_domain(tenant, "BD1")
    vrf = aci_integration.create_vrf(tenant, "VRF1")
    aci_integration.bind_bd_to_vrf(bd, vrf)
    subnet = aci_integration.create_subnet(bd, "192.168.1.1/24")

    aci_integration.deploy_configuration(tenant)

    network_data = [
        {"bandwidth": 2000, "latency": 30},
        {"bandwidth": 1500, "latency": 20},
        {"bandwidth": 1000, "latency": 50}
    ]

    total_bandwidth, total_latency, total_throughput =
calculate_network_performance(network_data)
    performance = optimize_network_configuration(total_bandwidth, total_latency)

    topology = generate_network_topology()

```

```
resources, total_allocation = calculate_resource_allocation()
```

```
monitor_traffic("192.168.1.100")
```

К6П3\_2024

## Файл aci\_integration.py

```
import acitoolkit.acitoolkit as aci

class ACINetworkIntegration:
    def __init__(self, url, username, password):
        self.session = aci.Session(url, username, password)
        self.session.login()

    def create_tenant(self, tenant_name):
        tenant = aci.Tenant(tenant_name)
        return tenant

    def create_application_profile(self, tenant, app_name):
        app_profile = aci.AppProfile(app_name, tenant)
        return app_profile

    def create_epg(self, app_profile, epg_name):
        epg = aci.EPG(epg_name, app_profile)
        return epg

    def create_contract(self, tenant, contract_name):
        contract = aci.Contract(contract_name, tenant)
        return contract

    def deploy_configuration(self, tenant):
        self.session.push_to_apic(tenant.get_url(), data=tenant.get_json())
```

```
import requests
import logging
from smtplib import SMTP

# Ініціалізація журналювання
logging.basicConfig(filename='network_monitor.log', level=logging.INFO,
format='% (asctime)s - %(message)s')

class NetworkMonitoring:
    def __init__(self, nodes):
        self.nodes = nodes

    def check_node_status(self):
        for node in self.nodes:
            response = requests.get(f"http://{node}/status")
            if response.status_code == 200:
                logging.info(f"Node {node} is up")
            else:
                logging.warning(f"Node {node} is down")
                self.send_alert(f"Node {node} is down")

    def send_alert(self, message):
        with SMTP("smtp.mailtrap.io", 2525) as server:
            server.login("username", "password")
            server.sendmail("from@example.com", "to@example.com", message)
```

```
class NetworkScaling:
    def __init__(self, initial_resources):
        self.resources = initial_resources

    def scale_up(self, additional_resources):
        self.resources += additional_resources
        print(f"Resources scaled up by {additional_resources}")

    def scale_down(self, removed_resources):
        self.resources -= removed_resources
        print(f"Resources scaled down by {removed_resources}")

    def auto_scale(self, current_load):
        if current_load > 80:
            self.scale_up(20)
        elif current_load < 20:
            self.scale_down(20)
```

КБПЗ\_2024

```
class NetworkStatistics:
    def __init__(self):
        self.stats = []

    def collect_traffic_stats(self, node, traffic_data):
        self.stats.append({"node": node, "traffic": traffic_data})

    def generate_report(self):
        for stat in self.stats:
            print(f"Node: {stat['node']}, Traffic: {stat['traffic']}")
```

КБПЗ\_2024

## Файл gui.py

```

from flask import Flask, jsonify, render_template_string

app = Flask(__name__)

@app.route("/network_status", methods=["GET"])
def network_status():
    status = {
        "nodes": ["Router1", "Switch1", "Firewall"],
        "status": "Operational"
    }
    return jsonify(status)

# Додавання маршруту для перегляду ліцензії
@app.route("/license", methods=["GET"])
def license():
    license_text = """
    <h1>Ліцензія Freeware</h1>
    <p>
    Назва проекту: ACI Network Project<br>
    Ліцензія: Freeware License<br><br>
    Copyright (C)<br><br>

    Це програмне забезпечення надається "як є", без будь-якої гарантії, явно
    вираженої чи такої, що мається на увазі.
    У жодному випадку автори не несуть відповідальності за будь-які збитки, що
    виникають в результаті використання цього програмного забезпечення.<br><br>
    Дозвіл надається будь-кому використовувати це програмне забезпечення для будь-
    яких цілей, включаючи комерційне використання, а також змінювати та поширювати
    його вільно,
    з дотриманням таких обмежень:<br><br>
    1. Походження цього програмного забезпечення не повинно бути спотворено; ви
    не повинні стверджувати, що ви написали оригінальне програмне забезпечення.
    Якщо ви використовуєте це програмне забезпечення у продукті, буде
    доречним вказати походження в документації продукту, але це не є
    обов'язковим.<br><br>
    2. Змінені версії джерельного коду повинні бути чітко позначені як такі, і
    не повинні бути представлені як оригінальне програмне забезпечення.<br><br>
    3. Це повідомлення не може бути видалене або змінене з будь-якої частини
    вихідного коду.<br><br>
    Це програмне забезпечення ліцензується за ліцензією Freeware, яка дозволяє
    вільне використання, модифікацію та поширення, але без будь-яких гарантій.<br>
    </p>
    """
    return render_template_string(license_text)

if __name__ == "__main__":
    app.run(port=5000)

```

## Файл optimization.py

```
def optimize_network_configuration(bandwidth, latency):  
    if bandwidth > 1000 and latency < 50:  
        return "Optimized"  
    elif bandwidth < 1000 and latency > 100:  
        return "Need Optimization"  
    else:  
        return "Moderate Performance"  
  
def predict_network_load(current_load):  
    predicted_load = current_load * 1.1 # Спрощений алгоритм прогнозування  
    return predicted_load
```

КБПЗ\_2024

## Файл backup.py

```
import json
import os

class BackupManager:
    def __init__(self, backup_dir="backups"):
        self.backup_dir = backup_dir
        if not os.path.exists(self.backup_dir):
            os.makedirs(self.backup_dir)

    def create_backup(self, config_data, backup_name="backup"):
        backup_file = os.path.join(self.backup_dir, f"{backup_name}.json")
        with open(backup_file, 'w') as file:
            json.dump(config_data, file)
        print(f"Backup created: {backup_file}")

    def restore_backup(self, backup_name="backup"):
        backup_file = os.path.join(self.backup_dir, f"{backup_name}.json")
        if os.path.exists(backup_file):
            with open(backup_file, 'r') as file:
                config_data = json.load(file)
            print(f"Backup restored: {backup_file}")
            return config_data
        else:
            print(f"No backup found: {backup_file}")
            return None
```

```
import boto3

class CloudIntegration:
    def __init__(self, aws_access_key, aws_secret_key, region):
        self.client = boto3.client(
            'ec2',
            aws_access_key_id=aws_access_key,
            aws_secret_access_key=aws_secret_key,
            region_name=region
        )

    def list_instances(self):
        instances = self.client.describe_instances()
        for reservation in instances['Reservations']:
            for instance in reservation['Instances']:
                print(f"Instance ID: {instance['InstanceId']}, State: {instance['State']['Name']}")

    def start_instance(self, instance_id):
        self.client.start_instances(InstanceIds=[instance_id])
        print(f"Instance {instance_id} started")

    def stop_instance(self, instance_id):
        self.client.stop_instances(InstanceIds=[instance_id])
        print(f"Instance {instance_id} stopped")
```

## Файл load\_balancer.py

```
class LoadBalancer:
    def __init__(self):
        self.servers = []

    def add_server(self, server):
        self.servers.append(server)

    def remove_server(self, server):
        self.servers.remove(server)

    def distribute_traffic(self, traffic):
        if not self.servers:
            print("No servers available")
            return

        # Простий алгоритм Round Robin
        server_index = 0
        for _ in range(traffic):
            print(f"Sending traffic to server: {self.servers[server_index]}")
            server_index = (server_index + 1) % len(self.servers)
```

## Файл error\_handling.py

```
import logging

logging.basicConfig(filename='errors.log', level=logging.ERROR)

class ErrorHandling:
    @staticmethod
    def handle_error(function):
        def wrapper(*args, **kwargs):
            try:
                return function(*args, **kwargs)
            except Exception as e:
                logging.error(f"Error occurred in {function.__name__}:
{str(e)}")
                print(f"Error: {str(e)}")
        return wrapper
```

КБПЗ\_2024

Файл `advanced_monitoring.py`

```
from prometheus_client import start_http_server, Gauge
import random
import time

class AdvancedMonitoring:
    def __init__(self):
        self.cpu_usage = Gauge('cpu_usage', 'CPU Usage of the network devices')
        self.memory_usage = Gauge('memory_usage', 'Memory Usage of the network
devices')

    def update_metrics(self):
        while True:
            cpu_value = random.uniform(0, 100)
            memory_value = random.uniform(0, 100)
            self.cpu_usage.set(cpu_value)
            self.memory_usage.set(memory_value)
            print(f"CPU: {cpu_value}%, Memory: {memory_value}%")
            time.sleep(5)

    def start_monitoring(self):
        start_http_server(8000)
        print("Prometheus monitoring started on port 8000")
        self.update_metrics()
```

КБПЗ\_2024

```
from aci_integration import ACINetworkIntegration
from monitoring import NetworkMonitoring
from security import SecurityPolicy
from scaling import NetworkScaling
from stats import NetworkStatistics
from optimization import optimize_network_configuration, predict_network_load
from backup import BackupManager
from cloud_integration import CloudIntegration
from load_balancer import LoadBalancer
from error_handling import ErrorHandling
from advanced_monitoring import AdvancedMonitoring
import threading

@ErrorHandling.handle_error
def main():
    # Інтеграція з ACI
    aci_integration = ACINetworkIntegration("https://apic.example.com", "admin",
"password")
    tenant = aci_integration.create_tenant("EnterpriseNetwork")
    aci_integration.deploy_configuration(tenant)

    # Запуск моніторингу
    monitoring = NetworkMonitoring(["192.168.1.1", "192.168.1.2"])
    monitoring_thread = threading.Thread(target=monitoring.check_node_status)
    monitoring_thread.start()

    # Масштабування мережі
    scaling = NetworkScaling(initial_resources=100)
    scaling.auto_scale(current_load=85)

    # Збір статистики
    stats = NetworkStatistics()
    stats.collect_traffic_stats("Router1", {"in": 1000, "out": 900})
    stats.generate_report()

    # Оптимізація конфігурації
    performance = optimize_network_configuration(1500, 30)
    predicted_load = predict_network_load(70)
    print(f"Network performance: {performance}, Predicted load:
{predicted_load}")

    # Резервне копіювання конфігурацій
    backup_manager = BackupManager()
    config_data = {"tenant": "EnterpriseNetwork", "profile": "EnterpriseApp"}
    backup_manager.create_backup(config_data, "network_backup")

    # Інтеграція з AWS
    cloud = CloudIntegration("aws_access_key", "aws_secret_key", "us-west-2")
    cloud.list_instances()
```

```
# Балансування навантаження
load_balancer = LoadBalancer()
load_balancer.add_server("Server1")
load_balancer.add_server("Server2")
load_balancer.distribute_traffic(10)

# Запуск Prometheus моніторингу
advanced_monitoring = AdvancedMonitoring()
monitoring_thread =
threading.Thread(target=advanced_monitoring.start_monitoring)
monitoring_thread.start()

if __name__ == "__main__":
    main()
```

КБПЗ\_2024

```
import sqlite3

class DatabaseManager:
    def __init__(self, db_name="network.db"):
        self.conn = sqlite3.connect(db_name)
        self.create_table()

    def create_table(self):
        with self.conn:
            self.conn.execute("""
                CREATE TABLE IF NOT EXISTS network_logs (
                    id INTEGER PRIMARY KEY AUTOINCREMENT,
                    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
                    node_name TEXT,
                    status TEXT
                )
            """)

    def log_network_event(self, node_name, status):
        with self.conn:
            self.conn.execute("INSERT INTO network_logs (node_name, status)
VALUES (?, ?)", (node_name, status))

    def get_logs(self):
        with self.conn:
            return self.conn.execute("SELECT * FROM network_logs").fetchall()
```