

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр
Освітній рівень

Програмно-технічний засіб реалізації хмарного сховища на базі Linux з
технологією Plex
Назва теми

КВРКІ 200024.21.01.01 ПЗ
Шифр

Галузь знань 12 «Інформаційні технології»
Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»
Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»
Назва

Виконав: студент IV курсу, група КІІн-21-1  Роні АБУ АЛЕЗ
Ініціали, прізвище

Керівник  Сергій ЛИСЕНКО
Ініціали, прізвище

Нормоконтролер  Тетяна КИСІЛЬ
Ініціали, прізвище

До захисту допускаю:
зав. кафедри комп'ютерної
інженерії та інформаційних
систем

 Ольга ПАВЛОВА
Ініціали, прізвище

«05» червня 2025 р.

Хмельницький 2025

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ

Зав. кафедри Ольга ПАВЛОВА

“ 10 ” 01 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Роні АБУ АЛЕЗ

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Програмно-технічний засіб реалізації хмарного сховища на базі Linux з технологією Plex

Керівник проекту (роботи) Сергій ЛИСЕНКО, д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 07.02.2025 р. № 23

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2025 р.

3. Вихідні дані до проекту (роботи) Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Аналіз програмно-технічних засобів

Проектування програмно-технічного засіб реалізації хмарного сховища на базі linux з

технологією Plex

Реалізації хмарного сховища на базі linux з технологією Plex

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Схеми підсистем Plex Media Server

Архітектура Plex Media Server

Алгоритми роботи Plex Media Server

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Тетяна КИСІЛЬ, доцент кафедри КПС		
Антиплагиат	Андрій ПИЧЕНЮРУК, доцент кафедри КПС		

7. Дата видачі завдання « 10 » 01 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2025	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2025	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2025	виконано
4	Робота над розділом 2 – вибір компонентів для проектування системи адаптивного застосування моніторингових елементів розвідувального БПЛА	01.04.2025	виконано
5	Робота над розділом 3 – проектування системи адаптивного застосування моніторингових елементів розвідувального БПЛА	29.04.2025	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2025	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2025 року	

Студент

Керівник роботи

Підпис

Роні АБУ АЛЕЗ
Ініціали, прізвище

Сергій ЛИСЕНКО
Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Програмно-технічний засіб реалізації хмарного сховища на базі Linux з технологією Plex».

Автор роботи: Роні АБУ АЛЕЗ.

Керівник роботи: Сергій ЛИСЕНКО

Пояснювальна записка: 66 с., 12 рис., 3 дод., 54 джерела.

Графічна частина: 3 креслення.

ПРОГРАМНО-ТЕХНІЧНІ ЗАСОБИ, КОМП'ЮТЕРНА СИСТЕМА, ХМАРНЕ СХОВИЩЕ, LINUX, ТЕХНОЛОГІЯ PLEX.

Метою дипломної роботи є проєктування та реалізація програмно-технічних засобів реалізації хмарного сховища на базі Linux з технологією Plex.

Об'єктом дослідження є програмно-технічні засоби реалізації хмарного сховища.

Предметом дослідження є програмно-технічний засіб реалізації хмарного сховища на базі Linux з технологією Plex.

Під час проведення даного дослідження був використаний метод систематичного огляду літератури для вивчення і аналізу предметної області даного дослідження з текстових джерел інформації.







Підпис студента

30.05.2025

Дата

№ р я д к а	Ф о р м а т	Позначення	Найменування	К і л л и с т і в	№ е к з	П р и м і т к а
			<u>Текстові документи</u>			
1		КВРКІ 200024.21.01.01 ПЗ	Пояснювальна записка	66		
			<u>Графічні матеріали</u>			
2		КВРКІ 200024.21.01.01 Е8	Схеми підсистем Plex Media Server	1		
3		КВРКІ 200024.21.01.01 Е8	Архітектура Plex Media Server и	1		
4		КВРКІ 200024.21.01.01 Е8	Алгоритми роботи Plex Media Server	1		

КВРКІ 200024.21.01.01 ВП					
Зм	Арх	№ докум	Підпис	Дата	
Розробив		Абу Алез			
Перевір.		Лисенко			
Н. контр.		Кисіль		01.06.25	
Затв.		Павлова		05.06.25	
Відомість проекту			Літера	Аркуш	Аркушів
			У	1	1
			ХНУ, Клін-21-1		

ЗМІСТ

ВСТУП.....	4
1 АНАЛІЗ ПРОГРАМНО-ТЕХНІЧНІ ЗАСОБІВ.....	5
1.1 Актуальність застосування хмарних середовищ	5
1.2 Переваги застосування хмарних середовищ	6
1.3 Актуальність застосування технології Plex.....	8
1.4 Можливості використання операційної системи для побудови хмарного сховища на базі Linux з технологією Plex	9
1.5 Розгортання автоматичної каталогізації контенту з метаданими	12
1.6 Реалізація підтримки стрімінгу в реальному часі.....	13
1.7 Реалізація підтримки віддаленого доступу з можливістю захищеного з'єднання	14
1.8 Імплементация оптимізації відео під клієнтські пристрої.....	16
1.9 Використанням docker для ізольованого керування контейнером plex	19
1.10 Висновки до першого розділу.....	21
2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСІБУ РЕАЛІЗАЦІЇ ХМАРНОГО СХОВИЩА НА БАЗІ LINUX З ТЕХНОЛОГІЄЮ PLEX.....	22
2.1 Огляд обраної технології хмарного сховища.....	22
2.2 Вибір типу хмар.....	24
2.3 Розроблення архітектури Plex Media Server.....	26
2.4 Вибір протоколу обміну даними	40
2.4.1 DLNA.....	40
2.4.2 HTTPS.....	40
2.4.3 Власні API Plex.....	41
2.5 Вибір операційної системи.....	42
2.6 Огляд застосовуваної технології хмарного сховища.....	44
2.10 Висновки до другого розділу	46

КВРКІ. 200024.21.01.01 ПЗ

Зм.	Арк.	№ док.ум.	Підпис	Дата	Літера	Аркуш	Аркушів
Виконав	Абу Алез Роні						
Перевір.	Сергій ЛИСЕНКО						
Н.КОНТР.	Тетяна КИСІДЬ			04.06.21			
Затвер.	Ольга ПАВЛОВА			15.06.21			

Програмно-технічний засіб реалізації хмарного сховища на базі Linux з технологією Plex.
Пояснювальна записка

ХНУ КІІн-21-1

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ХМАРНОГО СХОВИЩА НА БАЗІ LINUX З ТЕХНОЛОГІЄЮ PLEX.....	47
3.1 Архітектура програмно-технічного засобу. Схема взаємодії компонентів: сервер, Plex, файлове сховище, користувач.....	47
3.2 Розроблення системного програмного забезпечення.....	50
3.3 Алгоритм функціонування системи з використанням Plex, Rclone і Cloud Drives.....	52
3.4 Інсталяція системного програмного забезпечення.....	54
3.5 Застосування розробленої програмно-апаратна реалізації хмарного сховища на базі linux з технологією Plex.....	58
3.6 Налатування конфігурації з одним сховищем	61
3.7 Налатування конфігурації з кількома хмарними сховищами.....	63
3.8 Автоматизація використання одного або декількох облікових записів Google Drive як сховища для сервера Plex	65
3.9 Висновки до третього розділу.....	69
ВИСНОВКИ	70
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	71
ДОДАТОК А.....	77
ДОДАТОК Б	78
ДОДАТОК В	79

ВСТУП

Актуальність застосування технології Plex зумовлена стрімким зростанням потреби в ефективному управлінні, організації та потоковій трансляції мультимедійного контенту в умовах цифрової трансформації.

Сучасні користувачі прагнуть мати зручний доступ до власних медіабібліотек з різних пристроїв, у тому числі в умовах віддаленого доступу. Plex забезпечує високий рівень організації контенту, автоматичну каталогізацію, підтримку різних форматів медіа та широкі можливості персоналізації, що робить його ефективним інструментом для побудови універсального мультимедійного середовища.

Крім того, зростання популярності домашніх серверів та тенденція до децентралізації зберігання даних підсилюють інтерес до рішень на зразок Plex, які дозволяють зберігати контроль над інформацією, забезпечуючи водночас гнучкість і масштабованість.

Усе це обумовлює доцільність дослідження можливостей і переваг використання цієї технології в контексті сучасних ІТ-систем.

					КВРКІ. 200024.21.01.01 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ПРОГРАМНО-ТЕХНІЧНІ ЗАСОБІВ

1.1 Актуальність застосування хмарних середовищ

У сучасних умовах стрімкого розвитку цифрових технологій і зростання обсягів даних особливої актуальності набувають ефективні, масштабовані та безпечні засоби зберігання й обробки інформації. Одним із ключових рішень у цьому напрямі є хмарні сховища, які стали невід’ємною частиною інформаційної інфраструктури як у бізнесі, так і в освітніх, наукових та державних структурах.

Хмарні технології забезпечують гнучкий доступ до даних у будь-який час і з будь-якого пристрою, що особливо важливо в умовах дистанційної роботи, мобільності користувачів і розподілених команд. Завдяки цьому зберігання даних у хмарі є не лише зручним, але й економічно ефективним рішенням порівняно з традиційними локальними серверами чи фізичними носіями інформації [1].

Крім того, хмарні сервіси активно впроваджують механізми резервного копіювання, масштабованості, автоматичного оновлення та інтеграції з іншими сервісами, що дозволяє оптимізувати бізнес-процеси й уникнути втрат даних у разі збоїв, кібератак чи технічних несправностей. Це робить хмарні сховища привабливими не лише для великих корпорацій, але й для малих підприємств, стартапів, навчальних закладів та індивідуальних користувачів [2].

Особливої актуальності використання хмарних сховищ набуває в контексті забезпечення безпеки інформації. Сучасні хмарні платформи пропонують комплексні рішення з аутентифікації, шифрування, контролю доступу та моніторингу активності, що дозволяє ефективно захищати конфіденційні дані та відповідати сучасним стандартам кібербезпеки [3].

Також не варто нехтувати і екологічним аспектом. Централізоване хмарне зберігання даних сприяє енергоефективності ІТ-інфраструктур, зменшуючи потребу у великій кількості фізичних серверів на підприємствах та знижуючи вуглецевий слід [4].

					КВРКІ. 200024.21.01.01 ПЗ	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

Застосування хмарних сховищ є надзвичайно актуальним у сучасному інформаційному середовищі. Воно відкриває нові можливості для ефективного управління даними, забезпечення їхньої безпеки, масштабування інформаційних систем та підвищення загальної цифрової готовності організацій. У зв'язку з цим дослідження хмарних технологій і розробка рекомендацій щодо їх ефективного використання є важливим і своєчасним завданням.

1.2 Переваги застосування хмарних середовищ

Переваги застосування хмарних середовищ у порівнянні з локальним збереженням даних полягають у принциповій зміні підходу до зберігання, обробки та управління інформацією. Насамперед, хмарні технології забезпечують значно вищу гнучкість і доступність. Якщо локальне збереження передбачає фізичну прив'язку до конкретного пристрою або внутрішньої мережі організації, то хмарне середовище дозволяє отримати доступ до даних з будь-якої точки світу, де є підключення до Інтернету. Це особливо важливо для мобільних користувачів, віддалених команд, а також у випадках, коли необхідно оперативно ділитися інформацією з великою кількістю користувачів або пристроїв [5-9].

Ще однією ключовою перевагою є масштабованість. У випадку з локальним збереженням збільшення обсягу даних часто потребує додаткових апаратних витрат, модернізації інфраструктури або навіть повної заміни обладнання. Натомість хмарні сервіси дозволяють швидко адаптувати обсяг сховища відповідно до поточних потреб, фактично «на льоту», без простоїв і без потреби в технічному обслуговуванні з боку користувача.

Значну роль відіграє і автоматизація процесів, яку пропонують хмарні платформи. Наприклад, резервне копіювання, синхронізація, оновлення програмного забезпечення та управління правами доступу можуть виконуватися автоматично, без активного втручання користувача. Це знижує ймовірність

					КВРКІ. 200024.21.01.01 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

людських помилок, мінімізує ризики втрати даних і дозволяє зосередитися на основній діяльності, а не на обслуговуванні ІТ-інфраструктури [10-12].

Безпека також є важливим чинником на користь хмари. Хоча існує хибне уявлення про більший ризик для даних у хмарі, насправді провайдери хмарних послуг реалізують розвинені механізми захисту: шифрування даних, багатфакторну автентифікацію, контроль доступу на основі ролей, аудит операцій, а також фізичний захист дата-центрів. Для більшості користувачів це забезпечує вищий рівень безпеки, ніж те, що вони можуть реалізувати локально, особливо з огляду на обмежені ресурси.

У контексті вартості хмарні рішення зазвичай виявляються вигіднішими, особливо для малих і середніх підприємств, освітніх закладів або індивідуальних користувачів.

Замість значних початкових інвестицій у сервери, жорсткі диски, джерела безперебійного живлення та системи охолодження, хмарні послуги оплачуються за моделлю підписки або за обсягом використання, що дозволяє краще прогнозувати витрати і зменшити фінансове навантаження.

Крім того, хмарні середовища сприяють більш ефективній співпраці між користувачами. Дані, збережені в хмарі, можна редагувати спільно в реальному часі, а всі зміни зберігаються автоматично [13].

Це значно підвищує ефективність командної роботи, порівняно з локальними файлами, які потребують постійного копіювання, пересилання або синхронізації вручну.

Хмарні технології пропонують принципово новий рівень зручності, ефективності та надійності в управлінні інформацією, що робить їх логічною еволюцією систем збереження даних у цифрову епоху.

У порівнянні з традиційним локальним збереженням, хмарні середовища забезпечують вищий рівень адаптивності до вимог сучасного світу.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 7
Зм.	Арк.	№ докум.	Підпис	Дата		

1.3 Актуальність застосування технології Plex

У сучасних умовах цифровізації підприємств та активного впровадження концепції Industry 4.0 зростає потреба у використанні гнучких, масштабованих та інтегрованих інформаційних систем, які дозволяють ефективно управляти виробничими процесами, логістикою, постачанням та аналітикою в реальному часі. Однією з таких систем є Plex Manufacturing Cloud (або просто Plex) - хмарна ERP/MES-платформа, яка надає комплексне рішення для управління виробничими підприємствами [14].

Розглянемо основні переваги застосування технології.

Основним аспектом застосування технології Plex є підвищення ефективності виробництва. Plex забезпечує повну прозорість виробничого процесу, дозволяє оперативно реагувати на зміни в технологічному циклі, мінімізує простой обладнання та втрати ресурсів. Завдяки вбудованим інструментам моніторингу, платформа підтримує прийняття рішень на основі фактичних даних (data-driven decision making), що є критично важливим у сучасних умовах конкуренції.

Наступним фактором використання технології Plex є можливість інтеграція в рамках цифрових трансформацій. Plex є однією з платформ, яка органічно вписується в парадигму цифрового підприємства. Вона інтегрується з IoT-пристроями, аналітичними системами та іншими бізнес-рішеннями, формуючи єдину цифрову екосистему. Це забезпечує актуальність дослідження для завдань цифрової трансформації виробничих процесів [15-18].

Важливою можливістю технології є побудова хмарної архітектури. Перехід від локальних систем до хмарних рішень є загальносвітовим трендом. Plex, як хмарна платформа, забезпечує доступ до даних у режимі 24/7 з будь-якої точки світу, високу масштабованість та оновлення без участі користувача. Це дозволяє підприємствам зменшити витрати на IT-інфраструктуру, підвищити гнучкість та оперативність в управлінні.

					КВРКІ. 200024.21.01.01 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

На відміну від складних ERP-систем, Plex є доступнішим для середнього бізнесу, забезпечуючи швидке розгортання та простоту користування. Це робить її релевантною для дослідження в контексті впровадження інновацій на рівні регіональних виробничих компаній [19-21].

Plex підтримує роботу в реальному часі з виробничими даними, що дозволяє реалізовувати концепції Smart Manufacturing та Lean Production. Вивчення можливостей технології Plex у цьому контексті дозволяє дослідити потенціал впровадження таких підходів на практиці.

Застосування Plex у бакалаврській роботі дозволяє сформувати компетентності, затребувані на ринку праці, зокрема в галузі цифрового виробництва, управління даними, хмарних технологій та аналітики. Це підвищує практичну значущість обраної теми та її інноваційність.

1.4 Можливості використання операційної системи для побудови хмарного сховища на базі Linux з технологією Plex

У сучасних умовах активного використання мультимедійного контенту, виникає потреба в ефективних, безпечних та доступних рішеннях для зберігання та трансляції медіаданих. Одним з таких рішень є побудова власного хмарного медіасервера на базі операційної системи Linux з використанням технології Plex Media Server. Таке рішення дозволяє створити персоналізовану, масштабовану та контрольовану платформу для зберігання, організації та трансляції цифрового контенту [22-25].

Операційна система Linux є однією з найбільш стабільних, безпечних і налаштованих платформ для серверного використання. Її відкритий код дозволяє: повністю контролювати інфраструктуру - від налаштування файлових систем до систем безпеки та доступу, оптимізувати ресурси - Linux може працювати навіть на малопотужному апаратному забезпеченні, що робить її ідеальною для домашніх

NAS-рішень, а також автоматизувати обслуговування — завдяки системним скриптам, cron-завданням та іншим інструментам.

Plex Media Server - це програмна платформа для трансляції медіаконтенту (відео, аудіо, фото) через локальну мережу або Інтернет.

Платформа надає:

- автоматичну каталогізацію контенту з метаданими (постери, актори, опис);
- підтримку стрімінгу в реальному часі на різні пристрої (телевізори, смартфони, планшети);
- мультиплатформеність - доступна на linux, windows, macos, а також має мобільні та веб-додатки;
- підтримку віддаленого доступу з можливістю захищеного з'єднання з будь-якої точки світу;
- оптимізацію відео під клієнтські пристрої (транскодування в реальному часі);
- інтеграцію з plex pass - для додаткових функцій: dvr, синхронізація, live tv тощо.

На базі Linux можна розгорнути Plex як частину більших рішень:

- з використанням docker - для ізольованого та зручного керування контейнером plex;
- разом із samba/nfs - для організації мережевого доступу до файлів;
- через reverse proxy (наприклад, nginx) - для безпечного доступу через https;
- моніторинг за допомогою інструментів типу grafana та prometheus - для відстеження продуктивності;
- захист даних за допомогою брандмауерів, fail2ban та ssh.

Побудова власного хмарного сховища на базі Linux дозволяє:

- використовувати вже наявне апаратне забезпечення (наприклад, старий ПК, raspberry pi або nas);

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		

- відмовитися від сторонніх платних сервісів (google drive, icloud, netflix), зберігаючи приватність;
- знизити витрати на електроенергію та охолодження завдяки оптимізації під linux.

Використання Linux як платформи для побудови хмарного сховища з технологією Plex поєднує надійність, масштабованість та безпеку відкритої операційної системи з потужним функціоналом для організації та поширення мультимедійного контенту. Це рішення є актуальним для особистого використання, малих офісів, а також у рамках освітніх і дослідницьких проєктів [26-29].

Застосування технології Plex Media Server має широке практичне значення в контексті створення персонального хмарного медіасховища, особливо при використанні її на базі операційної системи Linux [30-34]. Plex дозволяє організувати зберігання, структурування та трансляцію мультимедійного контенту, зокрема відео, аудіо та фотофайлів, забезпечуючи зручний доступ до нього з будь-якої точки світу. Основна функціональність цієї технології полягає в перетворенні локального серверного середовища на повноцінний медіацентр, здатний транслювати контент у реальному часі на різноманітні клієнтські пристрої - від телевізорів і комп'ютерів до мобільних додатків та веббраузерів [35-39].

Користувач може розміщувати свій контент у визначених каталогах, після чого Plex автоматично здійснює його індексацію, додає метадані, такі як обкладинки, описи, актори, жанри, та створює єдину медіатеку з привабливим інтерфейсом. Однією з ключових особливостей є підтримка трансляції з автоматичним перекодуванням (транскодуванням) - це дозволяє адаптувати відео або аудіо до формату, оптимального для конкретного пристрою або швидкості з'єднання, не вимагаючи від користувача змін у первісному файлі [40-43].

Крім базової функціональності, Plex надає можливість створювати облікові записи для кількох користувачів, обмежувати їхні права доступу, а також організовувати спільне користування контентом у безпечному середовищі.

Додатково технологія підтримує віддалений доступ, що означає, що сервер, розгорнутий у домашній мережі, може використовуватись як персональна альтернатива комерційним стрімінговим платформам. Також Plex має інтеграцію з такими сервісами, як TIDAL або доступ до телепередач у реальному часі, що розширює його функціональність за межі локального контенту [44-47].

При використанні Plex на Linux-сервері відкриваються додаткові можливості з автоматизації роботи, резервного копіювання, моніторингу стану системи та оптимізації використання ресурсів. Завдяки гнучкості Linux можна адаптувати роботу сервера до будь-яких технічних умов — від малопотужного мінікомп'ютера Raspberry Pi до повноцінного серверного середовища. Це робить Plex універсальним інструментом як для домашнього користувача, так і для освітніх або дослідницьких цілей, зокрема в рамках реалізації рішень приватного хмарного зберігання мультимедійних даних.

1.5 Розгортання автоматичної каталогізації контенту з метаданими

Розгортання автоматичної каталогізації контенту з метаданими є однією з ключових переваг використання Plex Media Server у побудові сучасного хмарного сховища. Після додавання медіафайлів у відповідні каталоги на сервері, Plex автоматично виконує сканування вмісту, аналізуючи тип файлів, їхні назви та структуру. На основі отриманих даних система звертається до відкритих джерел метайнформації, таких як The Movie Database (TMDb), TheTVDB, MusicBrainz та інші, і завантажує пов'язану з кожним медіафайлом інформацію [48].

Цей процес включає отримання назв фільмів чи серіалів, описів, постерів, фонового зображення, даних про акторський склад, рейтинг, дату виходу, тривалість, жанр тощо. Для музичних файлів система витягує інформацію про виконавця, альбом, жанр, рік випуску, обкладинку альбому та навіть тексти пісень. Усе це дозволяє створити інтуїтивно зрозумілу, візуально привабливу та зручно

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

структуровану бібліотеку, яка забезпечує комфортне навігаційне середовище для користувача.

Завдяки такій каталогізації контент стає не просто набором файлів, а повноцінним цифровим архівом, що може фільтруватися, групуватися або шукатися за будь-якими характеристиками. Ця функціональність також забезпечує високий рівень персоналізації: Plex може автоматично створювати тематичні добірки, рекомендувати контент на основі історії переглядів або вподобань користувача. При цьому всі оновлення каталогу виконуються автоматично після додавання нових файлів або зміни структури директорій, що дозволяє зменшити участь користувача до мінімуму [49].

У контексті побудови хмарного сховища на базі Linux, автоматична каталогізація забезпечує інтеграцію з файловою системою та службами індексації, а також дозволяє адміністратору здійснювати централізоване управління метаданими та забезпечити їхню цілісність і актуальність. Таким чином, Plex перетворює необроблений масив цифрового контенту в добре організовану, логічну і привабливу мультимедійну базу, що відповідає вимогам сучасних систем зберігання й обробки інформації.

1.6 Реалізація підтримки стрімінгу в реальному часі

Реалізація підтримки стрімінгу в реальному часі на різні пристрої є однією з визначальних функцій Plex Media Server, яка забезпечує універсальний доступ до мультимедійного контенту незалежно від типу клієнтського пристрою. Завдяки вбудованому механізму потокової передачі даних, Plex дозволяє транслювати відео, аудіо та фото в реальному часі на широкий спектр пристроїв, включаючи смарт-телевізори, ПК, ноутбуки, смартфони, планшети, медіаплеєри (наприклад, Roku, Apple TV, Chromecast) та навіть ігрові консолі.

Ключовим елементом цієї функціональності є транскодування контенту, тобто його динамічне перекодування під технічні характеристики пристрою, що

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

приймає потік. Наприклад, якщо оригінальний відеофайл має формат або роздільну здатність, яка не підтримується певним пристроєм, Plex автоматично адаптує цей файл «на льоту» - змінюючи кодек, бітрейт, роздільну здатність чи контейнер для забезпечення сумісності та стабільної передачі. Це особливо важливо в умовах обмеженої пропускної здатності мережі або при використанні мобільного інтернету.

Серверна частина Plex відповідає за обробку запитів, а клієнтські додатки - за відображення інтерфейсу користувача та відтворення контенту. За наявності якісного мережевого з'єднання відтворення починається практично миттєво, а за умов нестабільного зв'язку Plex автоматично оптимізує потік відповідно до змін у швидкості. Крім того, система підтримує адаптивне стрімінгове передавання (adaptive bitrate streaming), яке дозволяє змінювати якість відео без зупинки відтворення [50].

Усі ці можливості реалізуються без потреби у складному налаштуванні — Plex автоматично визначає оптимальні параметри стріму відповідно до клієнта, типу мережі та характеристик сервера. За потреби користувач або адміністратор може вручну вказати пріоритети якості, максимальний бітрейт чи спосіб відтворення (пряме відтворення, транскодування, оптимізовані копії).

Plex забезпечує гнучке, ефективне й масштабоване рішення для потокової передачі даних, що перетворює сервер з локального сховища у справжню багатоплатформену хмарну медіастанцію з миттєвим доступом до контенту з будь-якого куточка світу.

1.7 Реалізація підтримки віддаленого доступу з можливістю захищеного з'єднання

Реалізація підтримки віддаленого доступу з можливістю захищеного з'єднання є ще однією важливою функцією Plex Media Server, що дозволяє користувачам безпечно отримувати доступ до свого медіаконтенту з будь-якого

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

місця, де є підключення до Інтернету. Після налаштування віддаленого доступу, користувач може під'єднуватися до свого сервера не лише в локальній мережі, а й ззовні - через веб-інтерфейс, мобільні додатки або інші сумісні пристрої.

Для забезпечення безпеки Plex використовує шифрування з'єднання на основі HTTPS (SSL/TLS), що гарантує конфіденційність передаваних даних, захист від перехоплення трафіку та зловмисного втручання. Весь трафік між клієнтом і сервером шифрується, включно з метаданими та потоковим контентом. Plex автоматично керує видачею сертифікатів та обслуговуванням доменного імені через власну систему маршрутизації запитів, що спрощує процедуру налаштування навіть для недосвідчених користувачів.

Віддалений доступ реалізується через зв'язок із обліковим записом Plex, що дозволяє аутентифікувати користувача, а також централізовано зберігати налаштування, список доступних серверів і персональні уподобання. У разі використання кількох пристроїв або спільного користування медіатекою з родиною чи друзями, адміністратор сервера може створювати додаткові облікові записи або надавати обмежений доступ до частин бібліотеки, зберігаючи при цьому повний контроль над правами доступу.

Ще однією важливою перевагою є підтримка автоматичного пробросу портів (NAT-PMP / UPnP), яка дозволяє Plex самостійно налаштовувати маршрутизатор для забезпечення зовнішнього доступу без ручного втручання. У випадках, коли автоматичне налаштування неможливе (через політику безпеки мережі або обладнання), сервер підтримує ручну конфігурацію з докладною інструкцією.

Реалізація віддаленого доступу в Plex Media Server забезпечує не лише зручність використання мультимедійного сховища поза межами локальної мережі, але й гарантує високий рівень безпеки, стабільності та керованості підключень, що особливо важливо в умовах персонального або корпоративного використання хмарних мультимедійних сервісів.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 15
Зм.	Арк.	№ докум.	Підпис	Дата		

1.8 Імплементация оптимізації відео під клієнтські пристрої

Імплементация оптимізації відео під клієнтські пристрої, що включає транскодування в реальному часі, є однією з основних можливостей Plex Media Server. Ця функція дозволяє здійснювати динамічну адаптацію відео під технічні характеристики конкретного пристрою, що приймає потік.

Транскодування - це процес перетворення відеофайлів із одного формату або якості в інший, з урахуванням вимог конкретного пристрою, швидкості мережі або користувацьких налаштувань.

Коли користувач запускає відео, Plex автоматично визначає, чи сумісний формат відео з пристроєм, що запитує відтворення. Якщо пристрій не підтримує цей формат або роздільну здатність, Plex виконує транскодування на льоту. Це може включати:

1. Зміну формату файлу, наприклад, якщо відео в форматі MKV, а пристрій підтримує тільки MP4, Plex конвертує відео в цей формат.

2. Зміну бітрейту у ситуації, коли мережа користувача має обмежену пропускну здатність, Plex знижує бітрейт для того, щоб уникнути буферизації і забезпечити плавне відтворення.

3. Зміну роздільної здатності, наприклад, якщо відео у форматі 4K, але пристрій підтримує лише 1080p або менше, Plex знижує роздільну здатність, що дозволяє зберегти оптимальну якість при обмежених ресурсах.

Процес транскодування в реальному часі у Plex є автоматичним і прозорим для користувача. Веб-сервер Plex працює як посередник між медіафайлом і клієнтським пристроєм, виконуючи перекодування і доставку відео через мережу. Це забезпечує гнучкість та сумісність з різними типами пристроїв, від смартфонів і планшетів до телевізорів, комп'ютерів та інших медіаплеєрів.

Інтерфейс Plex також дозволяє користувачам налаштовувати параметри трансляції та транскодування в залежності від їхніх потреб. Наприклад, користувач може вказати максимальний бітрейт для мобільного пристрою або вибрати якість

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

відео при обмеженому інтернет-з'єднанні. Це дозволяє досягти балансу між якістю відтворення і швидкістю передавання даних.

Ключовою перевагою такого підходу є те, що Plex підтримує транскодування на ходу без необхідності попередньої підготовки медіафайлів, що значно економить час і дозволяє уникнути необхідності зберігати кілька версій одного і того ж відео для різних пристроїв.

Транскодування в реальному часі є важливим компонентом, який дозволяє Plex забезпечити високу сумісність і зручність використання мультимедійного контенту, автоматично адаптуючи відео до характеристик кожного пристрою та забезпечуючи оптимальне відтворення за будь-яких умов.

Реалізація інтеграції з Plex Pass відкриває додаткові можливості для користувачів Plex Media Server, що забезпечують доступ до розширених функцій і підвищеного рівня персоналізації. Plex Pass є платною підпискою, яка надає цілу низку інструментів та функцій, що покращують роботу з мультимедійним контентом і розширюють можливості для відтворення і управління.

Одна з основних переваг Plex Pass - це необмежене та більш швидке транскодування. Користувачі без підписки можуть мати обмеження на кількість одночасних сеансів транскодування, а також обмеження на бітрейт при потоковому передаванні контенту.

Plex Pass дозволяє зняти ці обмеження, забезпечуючи більш високу якість і стабільність трансляції на всіх пристроях.

Це особливо важливо для користувачів, які планують використовувати Plex в якості хмарного сервісу для зберігання та перегляду великого обсягу контенту з різних пристроїв.

Plex Pass також включає інтелектуальні функції організації контенту, такі як автоматичне створення метаданих, фільмів, телевізійних шоу та музичних бібліотек.

Наприклад, він включає автоматичне сканування та категоризацію медіафайлів, а також можливість створення обкладинок та банерів для фільмів і

альбомів. Ці функції значно покращують досвід користувача, забезпечуючи зручне та приємне відображення контенту в Plex.

Інтеграція Plex Pass дозволяє користувачам завантажувати медіа для офлайн-перегляду на мобільних пристроях. Це дає можливість переглядати відео та слухати музику без підключення до Інтернету, що особливо корисно під час подорожей або в умовах обмеженого доступу до мережі.

Plex Pass користувачі отримують доступ до експериментальних функцій, які ще не стали доступними для широкого загалу. Це включає нові можливості в UI (користувацький інтерфейс), покращене транскодування, а також інтеграцію з новими типами пристроїв і сервісів. Такі функції дозволяють користувачам тестувати найсучасніші інновації в Plex.

Plex Pass також забезпечує додаткові можливості для безпечного доступу, зокрема, можливість налаштування двухфакторної аутентифікації (2FA) для захисту акаунтів. Це додатковий рівень безпеки, що дозволяє забезпечити доступ лише авторизованим користувачам. Також забезпечується пріоритетна підтримка клієнтів для користувачів Plex Pass, що дозволяє швидше отримувати допомогу та рішення для будь-яких технічних проблем.

Plex Pass дає можливість підключатися до сервісів стрімінгу музики, таких як TIDAL, що дозволяє користувачам створювати свої власні музичні бібліотеки і включати у їхній контент як локально збережену музику, так і музику з онлайн-сервісів.

Також з'являється підтримка подкастів і прямого ефіру, що дає ще більше варіантів для користувачів, які використовують Plex для прослуховування та організації аудіоконтенту.

Однією з найбільш інноваційних можливостей для користувачів Plex Pass є можливість отримувати доступ до прямого ефіру через Live TV і використовувати сервер Plex як цифровий відеореєстратор (DVR) для запису телевізійних програм і серіалів.

Це дозволяє користувачам створювати персоналізовану телевізійну бібліотеку, що зберігається на сервері, а також дивитися улюблені шоу на вимогу.

Plex Pass також дає можливість інтеграції з іншими хмарними платформами для зберігання контенту, такими як Google Drive, Dropbox або OneDrive, що дозволяє користувачам створювати велику мультимедійну бібліотеку, яка зберігається не тільки локально, а й у хмарі, збільшуючи доступність і надійність зберігання даних.

Інтеграція Plex Pass у Plex Media Server значно розширює функціональність сервісу, дозволяючи користувачам отримати доступ до безлічі додаткових інструментів і покращених можливостей. Вона забезпечує безперебійну трансляцію медіаконтенту, підвищену безпеку, додаткові опції синхронізації та гнучкість управління бібліотеками, що робить користувацький досвід більш комфортним та налаштованим під індивідуальні потреби [51].

1.9 Використанням docker для ізолюваного керування контейнером plex

Використання Docker для ізолюваного керування контейнером Plex Media Server є ефективним і зручним методом розгортання Plex у сучасних середовищах, оскільки дозволяє ізолювати сервер Plex від основної операційної системи, забезпечуючи високий рівень портативності, безпеки та простоти налаштування.

Docker - це платформа для автоматизації розгортання, масштабування і керування додатками у контейнерах, що дозволяє створювати середовище для Plex з мінімальними витратами часу та ресурсів.

Docker дозволяє ізолювати Plex Media Server в окремому контейнері, що має низку переваг:

- Ізоляція середовища в Plex працює у контейнері, що знижує ймовірність конфліктів з іншими програмами та залежностями на хост-системі.
- Портативність контейнера Plex можна легко перенести між різними серверами або хостами без необхідності перевстановлення чи налаштування.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

– Оновлення Plex у контейнері можна здійснити без зміни основної операційної системи та її залежностей, просто за допомогою перезавантаження контейнера з новою версією.

– Docker дозволяє запускати кілька інстансів Plex на різних серверах або у різних контейнерах для масштабування, що зручно при великому обсязі даних і високому навантаженні.

Щоб створити контейнер для Plex за допомогою Docker, потрібно виконати кілька кроків. Перш за все, потрібно встановити Docker на сервері. Для цього можна використати стандартні пакунки для Linux (наприклад, через `apt` або `yum`), або завантажити інсталятор з офіційного сайту Docker.

```
bash
```

```
sudo apt-get update
```

```
sudo apt-get install docker-ce
```

Plex Media Server можна запустити у контейнері Docker за допомогою стандартного образу Plex, який доступний на Docker Hub. Для цього необхідно створити контейнер, виконуючи команду, яка завантажить відповідний образ і запустить Plex з вказанням на публічний порт, через який можна підключитися до Plex, вказанням директорії на сервері для збереження конфігурацій Plex, директорії для медіаконтенту, яку Plex буде використовувати для зберігання файлів.

Після того, як контейнер буде запущений, Plex Media Server буде доступний через веб-інтерфейс за визначеною адресою [52].

У інтерфейсі користувач може налаштувати сервер, додавати бібліотеки медіафайлів, налаштовувати доступ до контенту та управління користувачами.

Docker дозволяє дуже легко масштабувати і оновлювати Plex. Наприклад, щоб оновити Plex до нової версії, можна просто зупинити контейнер, завантажити нову версію образу та перезапустити контейнер.

Використання Docker для керування Plex також має кілька переваг у контексті безпеки:

Plex працює в окремому контейнері, що мінімізує ризик взаємодії з іншими додатками на хост-системі.

Оскільки Plex працює в контейнері, можна швидко оновлювати образ контейнера без необхідності оновлювати основну операційну систему [53].

Оскільки конфігурація Plex і медіафайли мапуються на зовнішні томи, їх легко можна резервувати за допомогою стандартних інструментів Docker для збереження даних контейнера. Це забезпечує безпеку даних, навіть якщо контейнер буде видалений чи переписаний.

Використання Docker для керування Plex Media Server надає багато переваг, таких як ізоляція додатку, портативність, простота в оновленні та масштабуванні, а також підвищена безпека. Docker робить налаштування та управління Plex простішим і зручнішим, дозволяючи користувачам швидко розгорнути, масштабувати і оновлювати свій сервер, зберігаючи контроль над ресурсами та безпекою.

1.10 Висновки до першого розділу

У першому розділі проведено детальний аналіз програмно-технічних засобів, які дозволяють ефективно реалізувати хмарне сховище. Встановлено, що використання хмарних середовищ є сучасною та економічно доцільною практикою, яка забезпечує гнучкий доступ до даних, масштабованість, підвищений рівень безпеки та зниження витрат на інфраструктуру. Обґрунтовано доцільність вибору технології Plex як базової платформи для управління мультимедійним контентом - завдяки її здатності до каталогізації, потокового відтворення, захищеного доступу та інтеграції з хмарними сервісами. Також розглянуто переваги застосування операційної системи Linux, зокрема стабільність, відкритість коду та підтримку контейнеризації через Docker, що забезпечує ізоляцію процесів, простоту розгортання та безпеку роботи сервера.

2 ПРОЄКТУВАННЯ ПРОГРАМНО-ТЕХНІЧНОГО ЗАСІБУ РЕАЛІЗАЦІЇ ХМАРНОГО СХОВИЩА НА БАЗІ LINUX З ТЕХНОЛОГІЄЮ PLEX

2.1 Огляд обраної технології хмарного сховища

Порівнюючи такі рішення як Nextcloud, Seafile, Plex, Syncthing та власні NAS-рішення, доцільно розглядати їх не лише з точки зору функціональності, а й з огляду на цільове призначення, продуктивність, гнучкість конфігурації та рівень технічної складності впровадження.

Nextcloud є одним із найпоширеніших open-source рішень для побудови особистого хмарного середовища. Його функціональність орієнтована на спільну роботу з документами, керування файлами, календарями, контактами, відеоконференціями тощо.

Це комплексна платформа, яка може замінити Google Workspace або Microsoft 365 у межах невеликого офісу або домашньої мережі.

Проте, коли мова йде про медіа-контент, зокрема потокову трансляцію відео або музики, Nextcloud потребує додаткових модулів і не забезпечує високої якості відтворення мультимедіа "з коробки". Транскодування відео, інтеграція з телевізорами, підтримка Smart TV чи мобільних додатків для медіаплеєрів — усе це у Nextcloud реалізується через сторонні плагіни або зовнішні сервіси, що ускладнює налаштування.

Seafile, хоча має спільне коріння з Nextcloud, більше орієнтується на ефективну синхронізацію та спільний доступ до великих обсягів файлів. Його архітектура побудована на розділенні метаданих та вмісту файлів, що забезпечує кращу продуктивність, особливо у великих інсталяціях.

Однак, Seafile не фокусується на медіа-потоках чи розширених функціях мультимедіа.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

Це рішення є вдалим вибором для резервного копіювання або централізованого зберігання файлів, але не придатне для домашнього кінотеатру чи мультимедійного центру.

Plex, на відміну від попередніх двох систем, є вузькоспеціалізованим рішенням саме для роботи з мультимедіа. Його головною перевагою є високий рівень інтеграції з клієнтськими пристроями: телевізорами, смартфонами, ігровими приставками.

Plex забезпечує автоматичну організацію бібліотек, завантаження метаданих, обкладинок, описів та рейтингів. Крім цього, він підтримує транскодування відео «на льоту», що дає змогу відтворювати навіть ті формати, які не підтримуються цільовим пристроєм.

Plex надає чудовий користувацький досвід і не потребує значної технічної підготовки для початку роботи, але має певну залежність від закритих компонентів, а деякі функції (як-от доступ з мобільних додатків без обмежень) вимагають платної підписки.

Syncthing є принципово іншим підходом до зберігання та синхронізації даних. Це peer-to-peer система, яка не потребує централізованого сервера: кожен пристрій синхронізується з іншими напряму.

Вона надзвичайно безпечна, шифрує весь трафік і не зберігає дані на сторонніх серверах. Проте її функціональність обмежена синхронізацією файлів - відсутня підтримка потокового медіа, немає веб-інтерфейсу для перегляду вмісту, і в цілому вона не придатна для організації мультимедійного сховища, орієнтованого на зручність перегляду відео або прослуховування музики.

Власні NAS-рішення – це широке поняття, яке включає як програмні комплекси (наприклад, OpenMediaVault, TrueNAS), так і самостійно побудовані рішення на базі Linux або BSD. Вони можуть бути гнучко налаштовані під конкретні потреби: створення RAID-масивів, налаштування файлових протоколів (SMB, NFS, FTP), реалізація резервного копіювання, підключення медіа-серверів (у тому числі Plex або Jellyfin).

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 23
Зм.	Арк.	№ докум.	Підпис	Дата		

Такий підхід дає максимальну контрольованість, проте вимагає високого рівня технічної обізнаності, особливо при розгортанні в умовах з обмеженим апаратним ресурсом чи потребою у безперебійному доступі.

У підсумку, Plex виділяється як найбільш зручне та спеціалізоване рішення для побудови хмарного мультимедійного сховища.

Його функціональність у сфері потокової трансляції, багатоплатформенна підтримка та простота інтеграції роблять його доцільним вибором для реалізації персонального домашнього медіацентру, особливо у поєднанні з виділеним Linux-сервером.

2.2 Вибір типу хмар

У сучасному інформаційному середовищі хмарні технології поділяються на кілька типів за способом доступу, управлінням ресурсами та рівнем приватності даних. Найбільш поширеними хмарами є:

- приватна;
- публічна;
- гібридна.

Кожна з них має свої особливості, переваги, обмеження та сфери застосування, що безпосередньо впливають на архітектуру та стратегію впровадження хмарного сховища, зокрема при реалізації мультимедійних рішень на базі Plex.

Приватна хмара (англ. Private Cloud) - це інфраструктура, яка повністю контролюється однією організацією або приватною особою.

Усі ресурси - обчислювальні потужності, сховище, мережеві служби - фізично або віртуально належать конкретному користувачеві.

Такий підхід дозволяє максимально контролювати рівень безпеки, обмежувати зовнішній доступ, налаштовувати сервіси під специфічні вимоги користувача.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

Приватна хмара може бути розгорнута як у локальній мережі (on-premises), так і на сторонньому обладнанні (наприклад, у дата-центрі з віддаленим управлінням).

У контексті проекту з Plex, приватна хмара - це варіант, коли медіасервер встановлюється на власному сервері або NAS у домашній мережі, з обмеженим доступом через VPN або з налаштуванням шифрування та фаєрволу.

Основною перевагою тут є безпека і контроль над даними, а недоліком - необхідність самостійно підтримувати інфраструктуру, оновлення, резервне копіювання і моніторинг.

Публічна хмара (англ. Public Cloud) - це модель, у якій ресурси надаються великими постачальниками хмарних сервісів, такими як Amazon Web Services, Google Cloud Platform чи Microsoft Azure.

Ці провайдери надають віртуальні сервери, сховища, платформи для розробки та обробки даних через мережу Інтернет на умовах підписки або оплати за використання.

Дані та обчислювальні ресурси фізично можуть бути розміщені будь-де у світі, а клієнт не контролює фізичну інфраструктуру.

У випадку з Plex, можливо реалізувати медіасервер на базі віртуального VPS-сервера в публічній хмарі, проте це пов'язане з обмеженнями щодо трансляції великого обсягу відеоданих (особливо HD/4K), високим трафіком та витратами на обчислювальні ресурси.

Публічні хмари вигідні для швидкого масштабування, проте поступаються приватним у питаннях приватності та довгостроковій економічності при великому навантаженні.

Гібридна хмара (англ. Hybrid Cloud) поєднує переваги обох моделей - приватної та публічної.

Вона дозволяє частину інфраструктури тримати локально (наприклад, медіасервер Plex та основне сховище файлів), а інші компоненти — резервне копіювання, авторизацію, синхронізацію чи стримінг до віддалених пристроїв —

реалізовувати через публічні сервіси. Такий підхід забезпечує баланс між безпекою та масштабованістю. Гібридна модель може включати, наприклад, резервне копіювання медіафайлів на хмарне сховище Google Drive або Amazon S3, тоді як основний медіаконтент транслюється локально. Вона також зручна для мобільного доступу: мобільні пристрої підключаються до хмари, тоді як основне навантаження на трансляцію виконується домашнім сервером.

Проте гібридна архітектура складна в реалізації, оскільки вона потребує продуманого маршрутизаційного трафіку, політик безпеки, синхронізації даних і дублювання критичних сервісів.

Таким чином, при виборі типу хмари було враховано не лише технічні ресурси, а й сценарії використання.

У випадку створення програмно-технічного засобу для домашнього хмарного сховища з Plex, оптимальним рішенням є приватна або гібридна хмара, які забезпечують належну продуктивність, зручність використання та високий рівень конфіденційності при зберіганні мультимедійного контенту.

2.3 Розроблення архітектури Plex Media Server

Plex Media Server є програмною платформою, яка забезпечує зручну організацію, трансляцію та доступ до мультимедійного контенту на різних пристроях через єдину централізовану систему.

Його функціональність охоплює весь життєвий цикл взаємодії з медіафайлами - від автоматичної індексації до високоякісної потокової передачі з урахуванням характеристик клієнтського пристрою.

Основна особливість Plex полягає в тому, що користувач має повний контроль над власним медіаконтентом, на відміну від комерційних стрімінгових сервісів, де контент ліцензований і обмежений правами доступу.

З функціонального боку Plex дозволяє користувачу створити персоналізовану медіабібліотеку з фільмами, серіалами, музикою, фото та

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

відеоархівами. Після додавання файлів у відповідні каталоги система автоматично сканує вміст, розпізнає типи медіа, отримує метадані з онлайн-джерел (наприклад, IMDb чи TheTVDB), завантажує постери, описи, трейлери, субтитри та рейтинги визначеного контенту.

Завдяки цьому бібліотека виглядає як професійний каталог, а користувачеві не потрібно перейматися організацією вмісту вручну.

Окремо варто відзначити інтелектуальний механізм пошуку та фільтрації, що дозволяє швидко знаходити потрібний контент за жанром, акторами, роком випуску або ключовими словами.

Ще однією важливою функціональною складовою є підтримка потокової трансляції у реальному часі.

Plex автоматично визначає технічні можливості клієнтського пристрою - тип екрана, роздільну здатність, доступну смугу пропускання, підтримувані кодеки - і за необхідності виконує транскодування медіафайлів «на льоту».

Це означає, що навіть якщо відео збережене у форматі, який не підтримується конкретним телевізором або смартфоном, Plex здатен змінити формат, бітрейт і роздільну здатність під час трансляції, забезпечуючи безперебійну та якісну роботу сервісу без втручання користувача.

З архітектурної точки зору система на Plex реалізована за клієнт-серверною моделлю.

У центрі знаходиться Plex Media Server, який виконує роль ядра всієї системи. Він інсталується на серверну операційну систему - зазвичай це Linux, Windows або macOS, хоча можливі варіанти з розгортанням у контейнерах, на NAS або на віртуальних машинах.

Сервер відповідає за зберігання бібліотеки, індексацію вмісту, транскодування, управління доступом і логікою взаємодії.

Клієнтські додатки, які існують для більшості сучасних платформ (Android, iOS, Smart TV, Xbox, Playstation, веб-браузери), підключаються до сервера через

					КВРКІ. 200024.21.01.01 ПЗ	Арк.
						27
Зм.	Арк.	№ докум.	Підпис	Дата		

внутрішню або зовнішню мережу, отримують метадані та здійснюють запити на перегляд вмісту.

Передача даних між клієнтом і сервером реалізується через протокол HTTPS, що забезпечує шифрування трафіку.

У разі, якщо сервер перебуває за NAT або фаєрволом, Plex використовує спеціальні механізми маршрутизації через власну хмарну інфраструктуру для пробивання мережевих бар'єрів.

Усе це робить сервіс дуже зручним для віддаленого доступу, навіть якщо сервер розгорнутий у домашній мережі без статичної IP-адреси.

Варто звернути увагу й на механізми багатокористувацького доступу.

Plex дозволяє створювати окремі профілі для різних членів родини, обмежувати доступ до певного контенту, зберігати історію перегляду та впроваджувати батьківський контроль.

Платформа також підтримує інтеграцію з Plex Pass - преміальною підпискою, яка відкриває додаткові функції

- мобільне кешування медіа для офлайн-перегляду;
- автоматичне завантаження субтитрів;
- розширені звіти;
- апаратне транскодування на підтримуваному залізі.

Plex Media Server є потужним прикладом сучасної гібридної мультимедійної платформи, яка поєднує локальне зберігання з хмарними технологіями маршрутизації, забезпечує високий рівень автоматизації, зручність для кінцевого користувача та гнучкість конфігурації.

Його архітектура дозволяє реалізувати як локальний домашній медіацентр, так і повноцінну систему віддаленого доступу до особистого відеоархіву з будь-якого пристрою у світі.

Схема архітектури програмно-технічного засібу реалізації хмарного сховища на базі linux з технологією Plex подана на рисунку 2.1

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 28
Зм.	Арк.	№ докум.	Підпис	Дата		

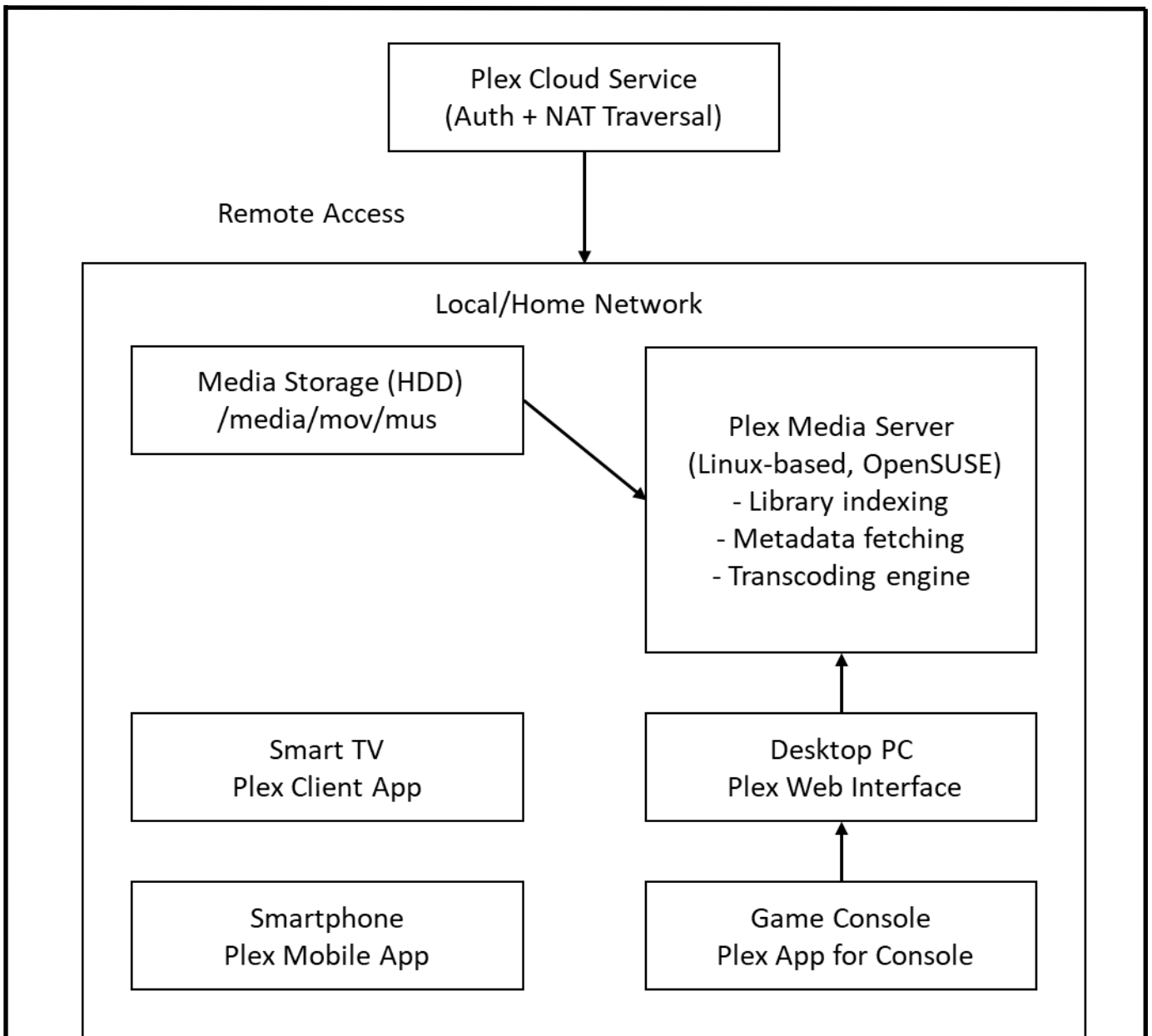


Рисунок 2.1 – Схема архітектури програмно-технічного засібів реалізації хмарного сховища на базі linux з технологією Plex

Рисунок 2.1 описує базову структуру, яку можна адаптувати під конкретні потреби користувача або розширити для гібридного сценарію з резервуванням у хмарі.

Plex Media Server розгорнуто локально на Linux-машині, яка має доступ до файлової системи з медіа.

Медіасховище - це окремий або вбудований диск із каталогами, які містять відео, музику, фотографії.

Клієнтські пристрої - телевізори, смартфони, ПК, консолі, які підключаються до сервера по локальній мережі або через Інтернет.

Plex Cloud допомагає з віддаленим доступом: пробивання NAT, авторизація, перенаправлення трафіку.

Транскодування виконується на сервері, якщо клієнт не підтримує певний формат медіа.

Схема архітектури Plex Cloud Service у контексті його функцій підтримки Plex Media сервера для забезпечення віддаленого доступу пробивання NAT, автентифікації користувачів і маршрутизації трафіку подано на рисунку 2.2.

Ця архітектура дозволяє користувачам Plex отримувати доступ до свого приватного медіасerverа з будь-якого місця, не змінюючи складних мережевих налаштувань.

Plex Cloud Core - це набір серверів у хмарі, які координують доступ між Plex-клієнтами та медіасerverами.

Вони не зберігають контент, але допомагають знайти сервер і забезпечити безпечне з'єднання.

Account DB – це база даних Plex, яка зберігає облікові записи користувачів, список авторизованих клієнтів і серверів.

NAT Traversal – це механізм, який дозволяє клієнтам знаходити сервери навіть за динамічними IP або за NAT, використовуючи проксі або пряме з'єднання.

Relay/Proху Connection використовується, якщо пряме з'єднання між клієнтом і сервером неможливе (наприклад, подвійний NAT).

Схема архітектури підсистеми Media Storage, яка є частиною загальної інфраструктури Plex-сервера подано на рисунку 2.3. Вона відповідає за зберігання, структурування, резервування й доступ до медіафайлів

Ця схема ілюструє, як побудована надійна система зберігання медіаконтенту, яка дозволяє Plex Media Server працювати ефективно й без втрат.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 30
Зм.	Арк.	№ докум.	Підпис	Дата		

Media Management відповідає за структурування папок, стандарти іменування, супровідні файли субтитрів, обкладинки тощо. Це спрощує автоматичне розпізнавання Plex Media сервером.

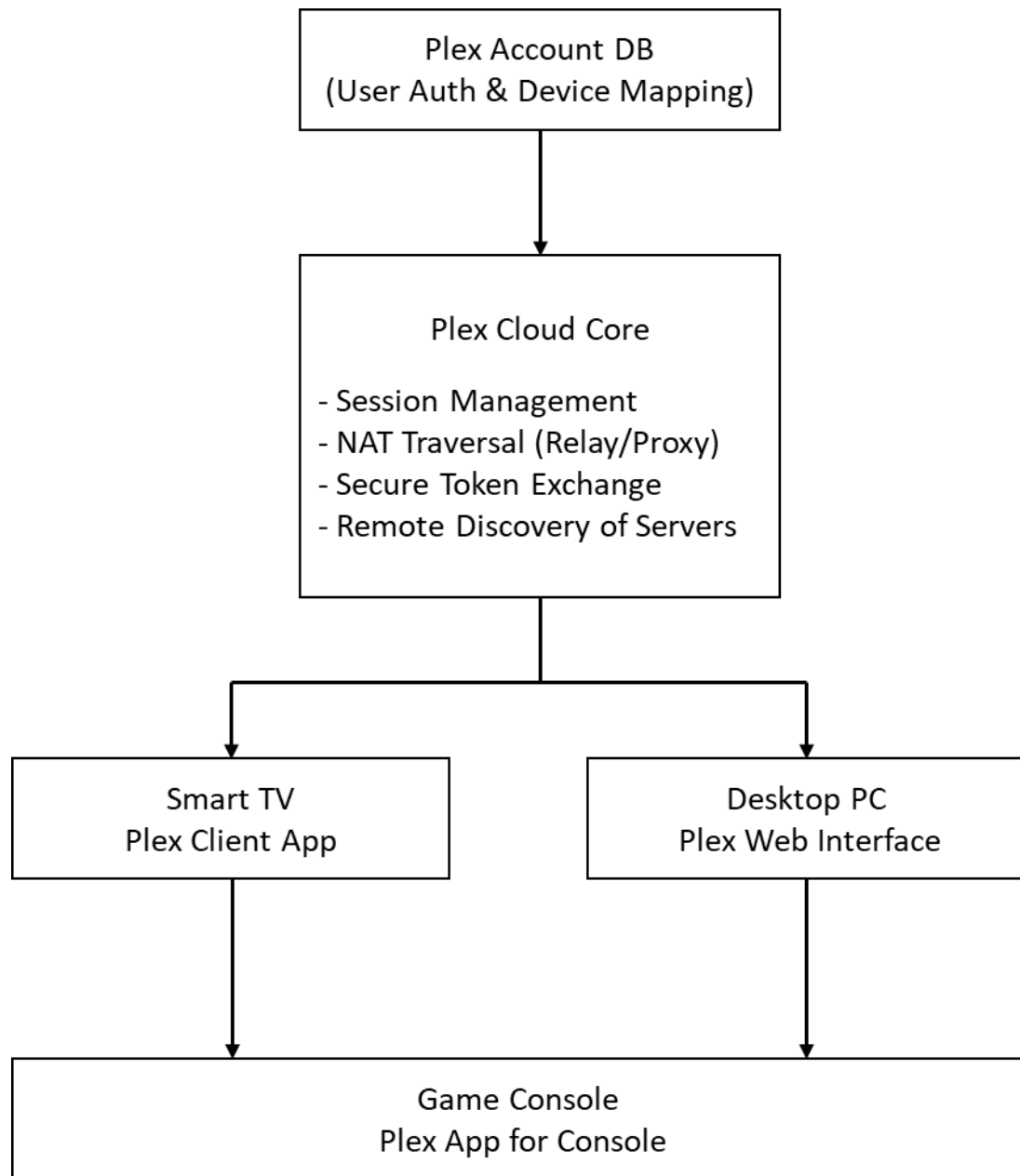


Рисунок 2.2 – Схема архітектури Plex Cloud Service у контексті його функцій підтримки Plex Media сервера

File System Layer забезпечує логічний доступ до збережених файлів, файлової організації, підтримку прав доступу та індексацію.

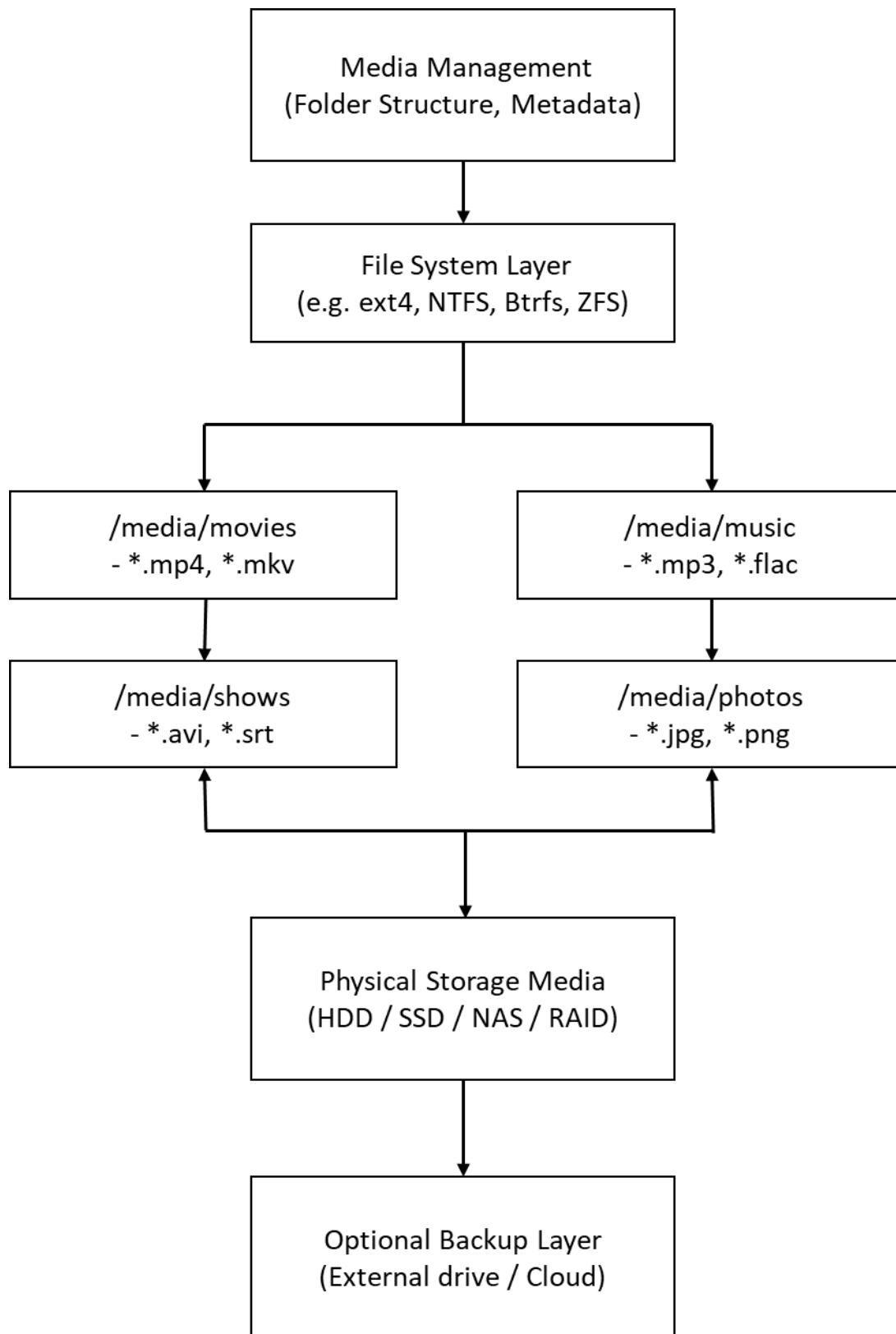


Рисунок 2.3 – Схема архітектури підсистеми Media Storage

Каталоги контенту організовані за типами мультимедіа (фільми, музика, серіали, фото), і можуть мати вкладені папки (наприклад, сезон/серія).

Фізичне сховище - це диски або RAID-масиви, що безпосередньо зберігають дані. Може бути підключене локально або через мережу (NAS).

Резервне копіювання забезпечує факультативний рівень захисту, реалізується через зовнішні диски або хмарні сервіси (наприклад, rsync на Google Drive або Amazon S3).

Схема архітектури Plex Media Server, яка є центральним компонентом, що виконує обробку, індексацію, трансляцію та керування доступом до медіаконтенту користувача, подано на рисунку 2.4.

Схема показує взаємодію компонентів Plex Media Server у повному циклі від зберігання до відтворення на користувацьких пристроях.

Plex Media Server - це ядро, яке забезпечує всі обчислювальні та координаційні процеси.

Metadata Fetcher - це модуль, що отримує інформацію з джерел типу IMDb, TheTVDB.

Media Library Index - це база, що зберігає структуру контенту, метадані, мітки, перегляди.

Media Transcoder – це ключовий компонент, який адаптує медіа під формат клієнта.

File System + Storage - це доступ до локальних або мережових медіа через файлову систему.

Plex Cloud Services забезпечує авторизацію, маршрутизацію, пробивання NAT.

Клієнти підключаються напряму або через хмару, отримують метадані, контент.

Схема архітектури Plex Cloud Core центральної хмарної служби, яка відповідає за з'єднання Plex Media Server з підключеними клієнтами, автентифікацію, маршрутизацію, NAT traversal та інші хмарні функції, подано на рисунку 2.5.

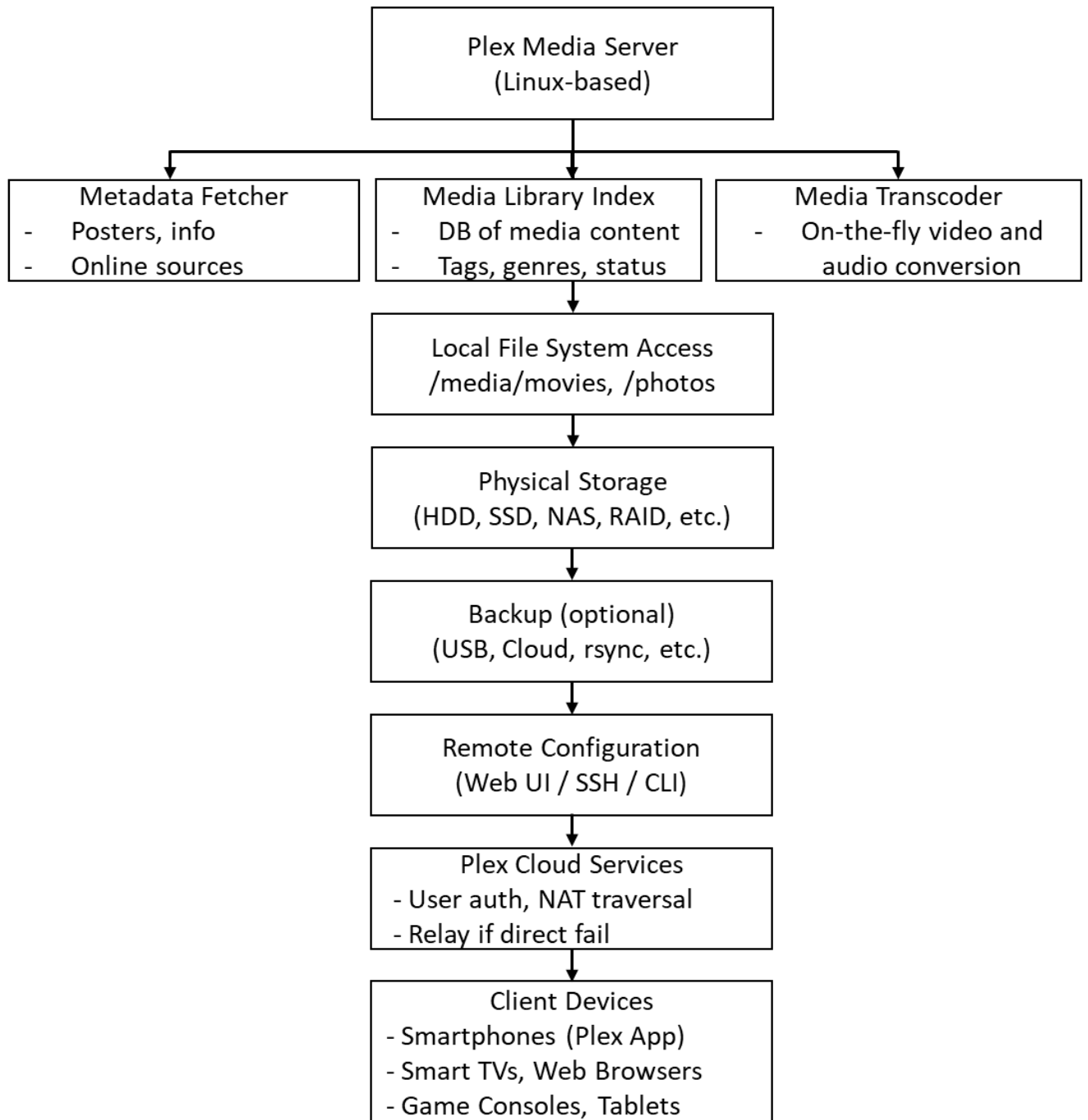


Рисунок 2.4 – Схема архітектури Plex Media Server

Архітектура дозволяє Plex забезпечити "прозорий" доступ до домашнього медіа навіть для некваліфікованого користувача без налаштування системних портів чи VPN.

Plex Cloud Core є логічним ядром хмарної платформи Plex, що діє як посередник між клієнтом і медіасервером.

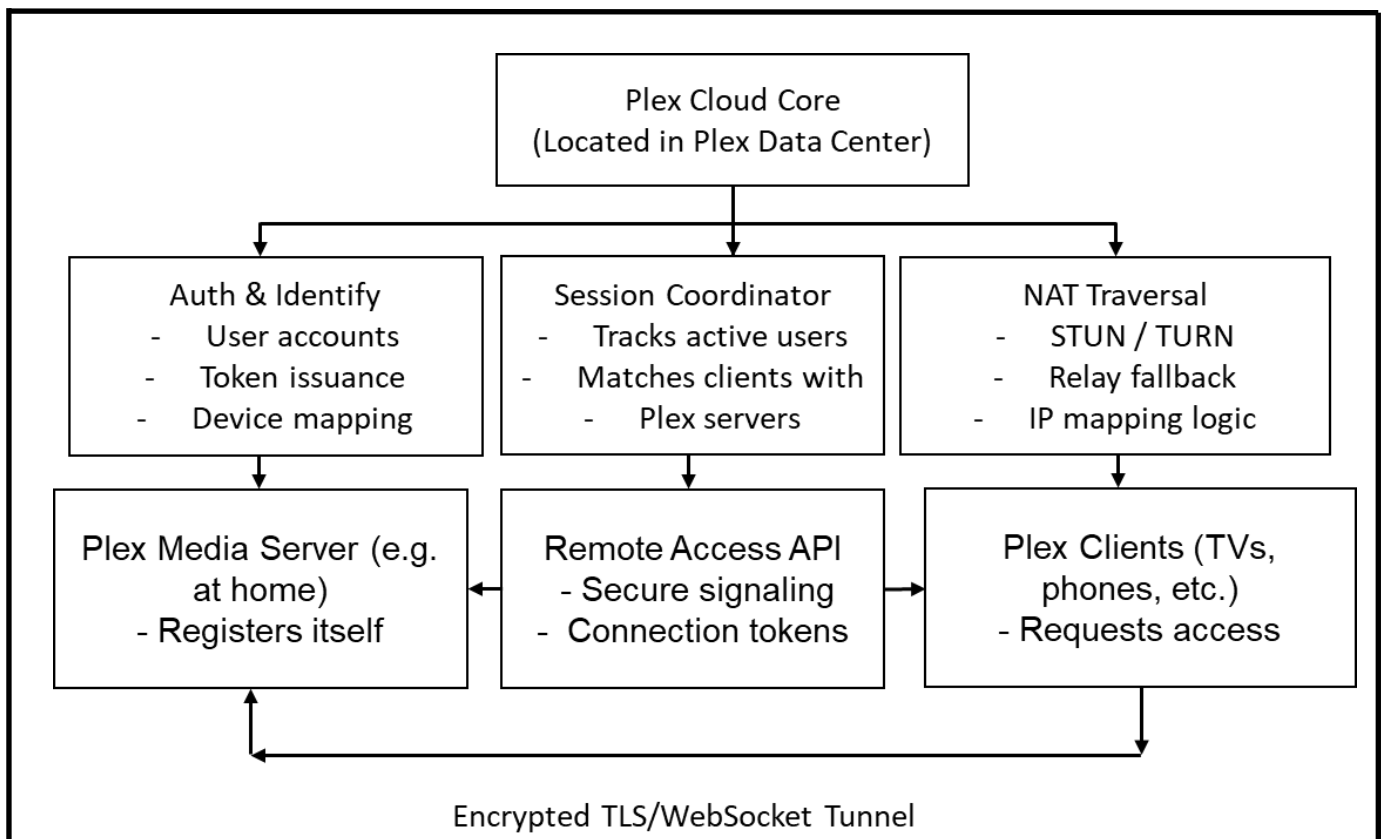


Рисунок 2.5 – Схема архітектури Plex Cloud Core

Модуль Auth & Identity відповідає за збереження облікових записів, видачу токенів доступу, а також прив'язку пристроїв до обліковки.

Session Coordinator координує активні сесії між Plex Media Server та клієнтами, підтримує актуальну інформацію про доступність сервера.

NAT Traversal забезпечує пробивання NAT через STUN/UDP або TURN/Relay, якщо пряме з'єднання неможливе.

Remote Access API використовується обома сторонами (сервером і клієнтом) для ініціалізації захищеного з'єднання.

TLS/WebSocket Tunnel є основним каналом для шифрованого управління й потоку даних, коли пряме з'єднання неможливе.

Схема архітектури взаємодії Plex Media Server (Home) з Plex Client (Remote) через Plex Cloud, з урахуванням маршруту, автентифікації, NAT, проксі/прямого з'єднання подано на рисунку 2.6.

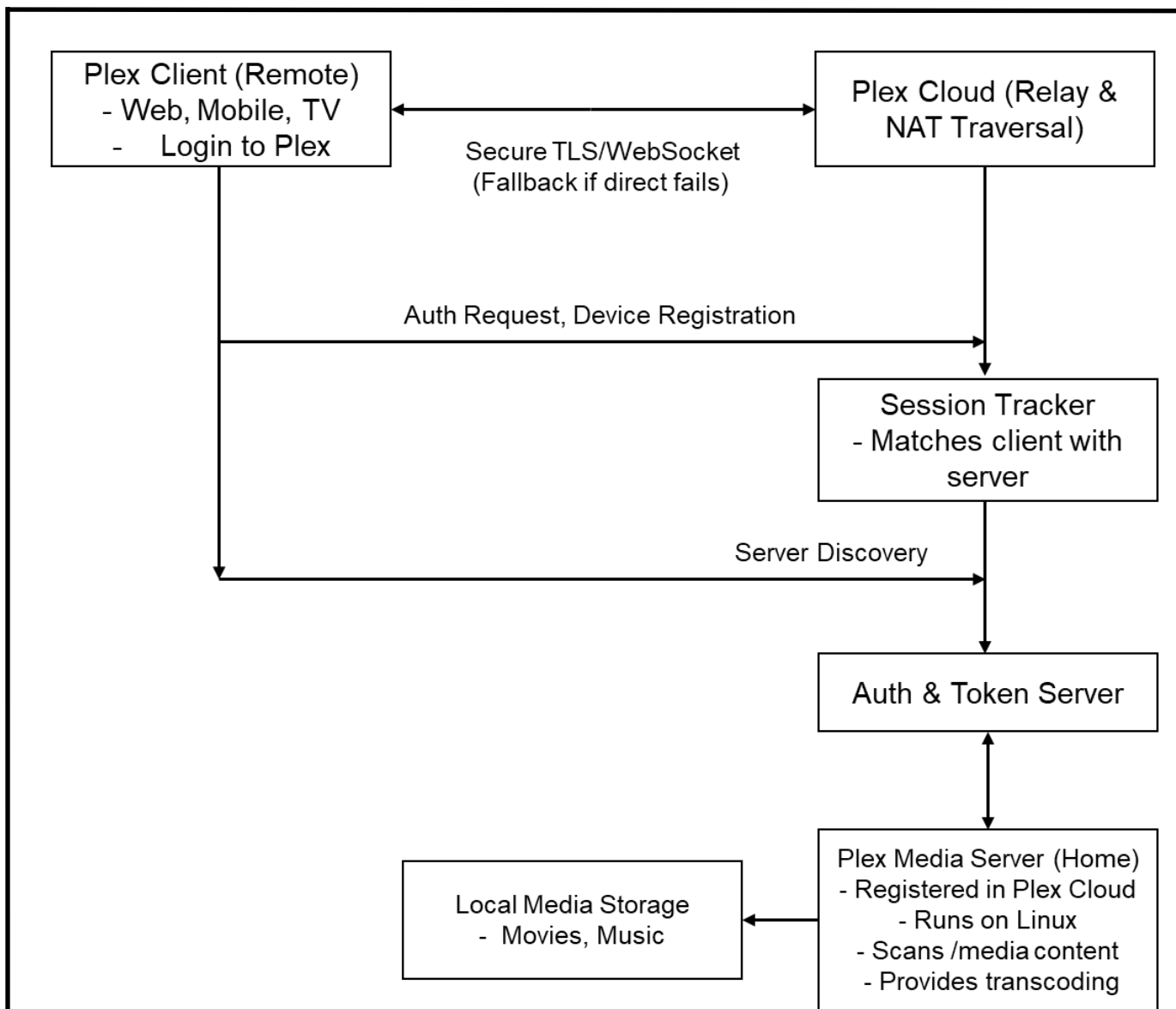


Рисунок 2.6 – Схема архітектури взаємодії Plex Media Server (Home) з Plex Client (Remote) через Plex Cloud

Схема дозволяє віддаленому клієнту без зайвих налаштувань отримати прозорий доступ до домашньої мультимедійної бібліотеки.

Plex Client (Remote) звертається до Plex Cloud, проходить автентифікацію та отримує маршрутизаційну інформацію.

Plex Cloud посередник, що:

- здійснює пробивання NAT;
- створює сесію між клієнтом і сервером;
- надає fallback-проксі канал (TLS/WebSocket), якщо пряме з'єднання

неможливе.

Plex Media Server (Home) знаходиться за NAT або динамічним IP, але підтримує з'єднання з Plex Cloud, повідомляючи про свою доступність.

Storage сервер має прямий доступ до фізичного або мережевого сховища з мультимедіа.

Схема архітектури Secure TLS Connection via Relay, яка використовується Plex, коли пряме з'єднання між клієнтом і сервером (через NAT) неможливе, подано на рисунку 2.7. У такому випадку весь зашифрований трафік передається через хмарний релейний сервер.

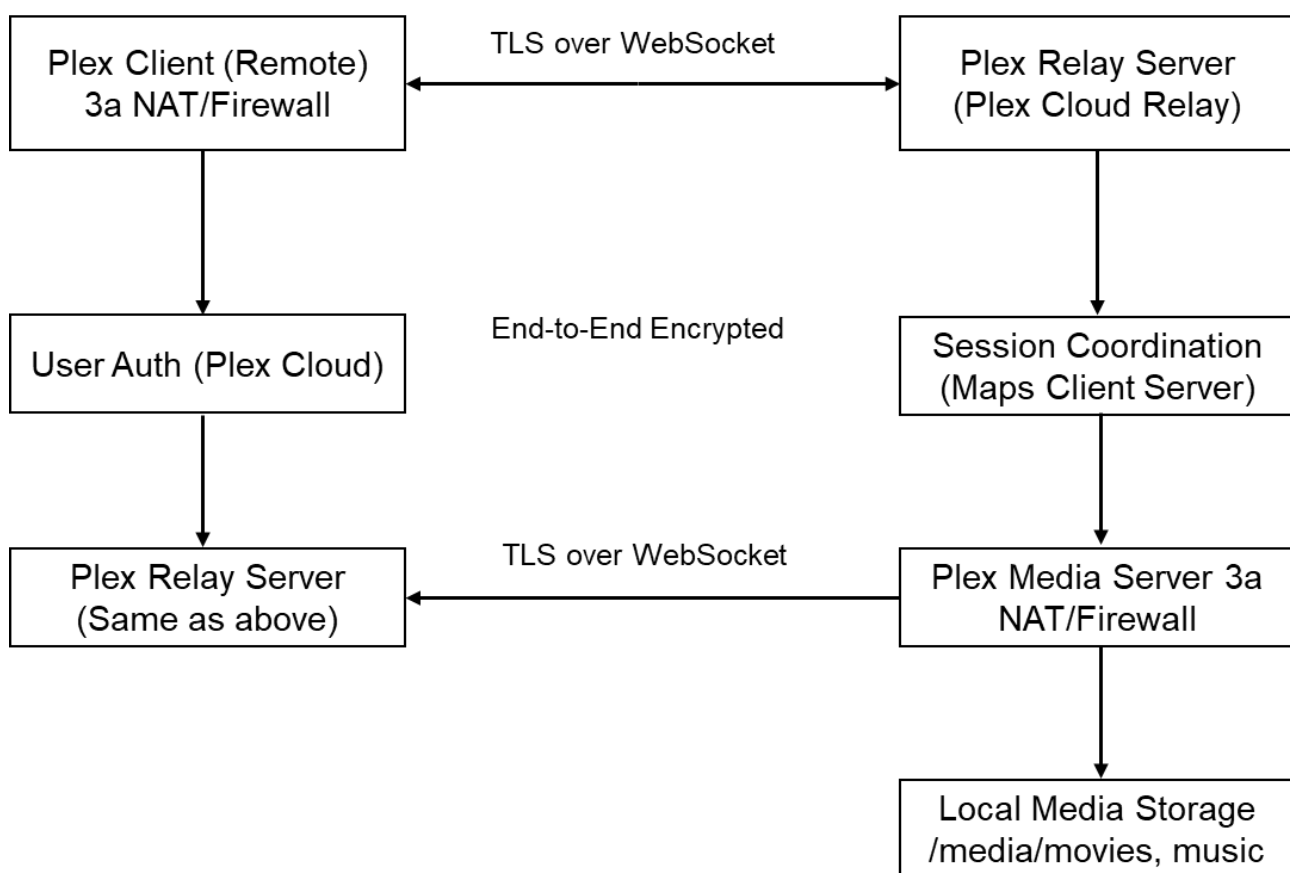


Рисунок 2.7 – Схема архітектури Secure TLS Connection via Relay

Це надійна, безпечна, але трохи повільніша альтернатива прямому з'єднанню, яку зручно застосовувати для звичайних користувачів без технічної підготовки.

Суть архітектури полягає в тому, що Plex Client встановлює TLS-з'єднання з Relay Server (у хмарі), оскільки не може напряму досягти з'єднання з домашнім сервером.

Plex Media Server, який також не має публічної IP-адреси або перебуває за подвійним NAT, сам ініціює вихідне з'єднання до того ж Relay Server.

Relay Server виступає в ролі двостороннього проксі, передаючи зашифрований трафік між клієнтом і сервером, не маючи доступу до контенту (бо трафік шифрується end-to-end).

Тоді Plex гарантує, що відео/аудіо потік буде доставлено навіть у складних мережеских умовах, без відкриття портів чи VPN.

Схема підсистеми Media Management у Plex Media Server подано на рисунку 2.8.

Вона відповідає за організацію, аналіз, індексацію, метадані та структурування мультимедійного контенту.

Ця архітектура забезпечує автоматизоване керування великою медіатекою з мінімальним втручанням користувача, а також гарантує узгодженість структури бібліотеки на всіх клієнтських пристроях.

Media Management Core є головним координатором, який ініціює всі процеси сканування та оновлення.

Metadata Agent підключається до зовнішніх баз даних, щоб автоматично отримати постери, рецензії, жанри, тривалість, акторський склад тощо.

Library Scanner аналізує файлову систему, сканує папки /media, ідентифікує нові або змінені файли.

Media Matcher на основі імен файлів і їх розташування визначає, чи це фільм, серіал, музика тощо.

Path Resolver нормалізує шляхи до медіа, забезпечує підтримку для багатокористувацьких конфігурацій і розподілених NAS.

Metadata DB зберігає усі витягнуті або згенеровані дані (локально, у вигляді SQLite або аналогічної БД).

Thumbnail Generator генерує візуальні попередні перегляди (постери, кадри), які відображаються в клієнтських інтерфейсах.

User Preferences DB враховує персоналізовані налаштування, які впливають на сортування, мову метаданих, відображення.

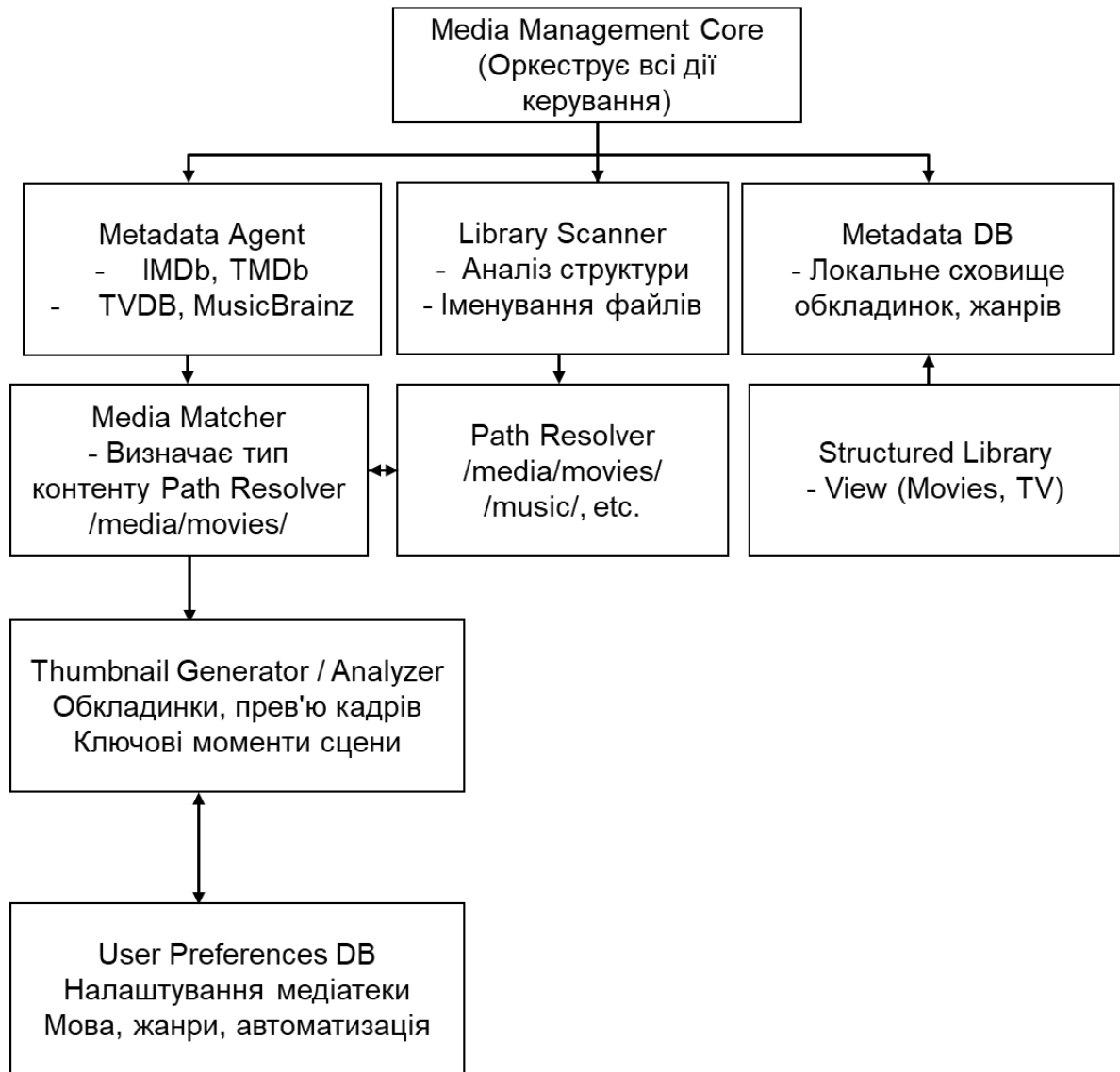


Рисунок 2.8 – Схема підсистеми Media Management у Plex Media Server

2.4 Вибір протоколу обміну даними

Обмін даними між компонентами системи мультимедійного сховища та кінцевими клієнтами базується на декількох ключових протоколах, кожен з яких має свою сферу застосування, переваги та обмеження.

В контексті реалізації Plex Media Server, як одного з варіантів побудови хмарного медіасховища, важливо розглянути три основні протокольні підходи: DLNA, HTTPS та власні API Plex.

2.4.1 DLNA

DLNA (Digital Living Network Alliance) - це галузевий стандарт для потокового передавання мультимедіа між пристроями в локальній мережі.

Він працює на основі набору технологій, включаючи UPnP (Universal Plug and Play), HTTP для передачі контенту, та SOAP для обміну службовими повідомленнями.

Plex Media Server має вбудовану підтримку DLNA, що дозволяє обслуговувати пристрої, які не мають рідного Plex-клієнта, але підтримують DLNA (наприклад, Smart TV, ігрові консолі, Blu-ray плеєри).

DLNA-підхід не вимагає встановлення додаткового ПЗ на стороні клієнта, однак є обмеженим у гнучкості доступу до розширених функцій Plex — таких як синхронізація перегляду, динамічне перекодування або доступ через Інтернет. Крім того, DLNA працює виключно в локальній мережі та не підтримує сучасні механізми автентифікації.

2.4.2 HTTPS

Для передачі даних між Plex Media Server та клієнтами, які працюють через офіційні додатки або веб-інтерфейс, використовується протокол HTTPS.

					КВРКІ. 200024.21.01.01 ПЗ	Арк.
						40
Зм.	Арк.	№ докум.	Підпис	Дата		

Це розширення стандартного HTTP, яке працює поверх TLS (Transport Layer Security), забезпечуючи шифрування трафіку та захист від перехоплення даних.

Через HTTPS реалізується:

- - передача метаданих (інформація про фільми, серіали, музику);
- - автентифікація користувачів через Plex Cloud;
- - управління сесіями перегляду;
- - передача команд (наприклад, "відтворити", "поставити на паузу");
- - передача потокового відео (у випадках, коли використовується Web-клієнт або мобільний додаток).

У випадку недоступності прямого з'єднання Plex автоматично тунелює HTTPS-трафік через власні Relay-сервери. При цьому зберігається шифрування та end-to-end захист даних між сервером та клієнтом.

2.4.3 Власні API Plex

Важливу роль у побудові взаємодії між компонентами системи Plex відіграє власний Plex API, який реалізований у вигляді набору RESTful-інтерфейсів, доступних через HTTP(S).

Plex API не є офіційно задокументованим у повному обсязі, проте активна спільнота дослідників змогла частково зворотно інженерувати функціональність цих API.

Через API реалізується:

- доступ до структури бібліотеки: фільми, серіали, сезони, треки;
- запити на сканування нових медіафайлів;
- отримання списку поточних відтворень і користувачів онлайн;
- віддалене керування сервером (перезапуск, оновлення бібліотеки, синхронізація тощо);
- доступ до інформації про пристрої, сесії, перегляди.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

API може використовуватись для розробки власних клієнтських інтерфейсів, скриптів автоматизації або інтеграції з іншими системами, наприклад, системами "розумного дому".

Більшість API вимагає автентифікації через токен користувача або OAuth-механізм Plex, а також працює лише при активному з'єднанні з Plex Cloud або локальній мережі з увімкненим дозволом.

Plex реалізує багаторівневу систему протоколів обміну даними.

Для внутрішньомережевої взаємодії підходить DLNA, що забезпечує сумісність зі старими пристроями.

HTTPS і власні API дають змогу забезпечити сучасну, гнучку, захищену та дистанційно керовану систему мультимедійного хмарного сховища.

Знання та розуміння цих протоколів є ключовим для побудови надійної архітектури програмно-технічного засобу на базі Linux і Plex.

2.5 Вибір операційної системи

Вибір операційної системи - один із фундаментальних етапів при розробці програмно-технічного засобу для хмарного сховища, особливо якщо йдеться про платформу, що має забезпечити стабільність, безпеку, простоту обслуговування та сумісність із мультимедійними сервісами, зокрема Plex.

У цьому контексті було прийнято рішення використовувати openSUSE Linux як базову операційну систему, що поєднує в собі гнучкість дистрибутивів сімейства Linux з інструментами корпоративного рівня.

Однією з ключових переваг openSUSE є її орієнтація як на серверне, так і на десктопне використання. Вона підтримує широке коло серверних компонентів "із коробки", включаючи systemd, Zypper для керування пакетами, систему автоматичного оновлення та інструментальну підтримку контейнеризації.

Це дозволяє створити стабільне середовище для роботи Plex Media Server, яке легко обслуговувати навіть у напівавтоматичному режимі.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 42
Зм.	Арк.	№ докум.	Підпис	Дата		

Завдяки наявності графічного інтерфейсу адміністрування YaST (Yet another Setup Tool), openSUSE надає додаткову зручність для налаштування мережі, сховищ, користувачів і служб, що особливо актуально при роботі з NAS-системами або домашніми серверами, де адміністратор може не мати глибокої експертизи в системному адмініструванні з командного рядка.

YaST також спрощує розгортання служб Samba, NFS або брандмауера, необхідних для безпечного доступу до медіаконтенту.

З точки зору сумісності з Plex, openSUSE повністю відповідає вимогам до системи: вона підтримує всі необхідні бібліотеки, має активну спільноту, і Plex Media Server офіційно надає збірки для дистрибутивів на базі `rpm`.

Це дозволяє легко встановити та оновлювати Plex через стандартні інструменти Zypper або вручну через `rpm`, не порушуючи структуру системи.

Особливу увагу вартує надати моделі оновлення openSUSE. У рамках двох гілок Leap та Tumbleweed можна обрати стабільну (Leap) для продакшн-рішень або ролінгову (Tumbleweed) для тестування та отримання новітніх функцій.

У випадку побудови хмарного сховища з очікуваним тривалим життєвим циклом, перевага надається openSUSE Leap як більш стабільній і прогнозованій платформі з довготривалим супроводом.

Завдяки високому рівню безпеки, регулярним патчам ядра, інтегрованому механізму AppArmor для ізоляції процесів та можливості шифрування сховищ безпосередньо на рівні ОС, openSUSE створює надійне середовище для зберігання та потокової трансляції медіаконтенту.

Поряд із Plex, який потребує ефективного управління ресурсами, така ОС дозволяє балансувати між продуктивністю, безпекою та масштабованістю.

Отже, вибір openSUSE як базової операційної системи обумовлений її технічною зрілістю, зручністю адміністрування, сумісністю з Plex, стабільністю у роботі та можливістю гнучко конфігурувати систему з урахуванням як серверного навантаження, так і мультимедійних потреб.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 43
Зм.	Арк.	№ докум.	Підпис	Дата		

Усі ці фактори роблять її придатною для побудови надійного хмарного сховища на базі Linux.

2.6 Огляд застосовуваної технології хмарного сховища

Вимоги до серверного обладнання є критично важливими для розуміння технічної основи, необхідної для реалізації надійного, масштабованого та продуктивного хмарного сховища на базі Plex Media Server.

Правильно підібране апаратне забезпечення визначає, наскільки ефективно сервер справлятиметься із завданнями трансляції, перекодування, індексації, зберігання та роздачі медіа-контенту.

У центрі будь-якого мультимедійного сервера стоїть центральний процесор (CPU). Для Plex це особливо важливо, оскільки саме процесор відповідає за перекодування відеопотоків у режимі реального часу, якщо пристрій клієнта не підтримує оригінальний формат або якщо потрібно зменшити бітрейт для повільнішого з'єднання.

Мінімально прийнятним вважається 4-ядерний процесор з підтримкою апаратного перекодування (наприклад, Intel Quick Sync або AMD VCE/VCN), однак для стабільної роботи з кількома одночасними потоками краще використовувати сучасні 6–8-ядерні процесори.

Важливим фактором є підтримка інструкцій AVX або AVX2, які прискорюють обробку мультимедіа та шифрування.

Наступним ключовим компонентом є оперативна пам'ять (RAM). Хоча Plex як сервіс не є надто вимогливим до об'єму пам'яті, у практичному застосуванні RAM активно використовується для кешування метаданих, попередньої буферизації потоків, роботи фонових процесів (наприклад, індексації чи генерації мініатюр).

Для побудови хмарного сховища з Plex на базі Linux базовий об'єм має становити не менше 8 ГБ.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

При додаванні додаткових сервісів, таких як системи моніторингу, доступ через SAMBA/NFS, індексація великих медіатек або використання Docker-контейнерів, доцільно закладати 16 ГБ і більше, залежно від навантаження.

Дискова підсистема виконує роль як сховища медіа, так і простору для системних та тимчасових файлів.

Надійність, продуктивність і масштабованість цієї підсистеми безпосередньо впливають на швидкість завантаження відео, час відповіді серверу, можливість паралельної роботи з великим об'ємом даних.

Основним медіа-сховищем зазвичай є жорсткі диски (HDD) великої ємності (4–10 ТБ і більше), об'єднані у масив (наприклад, RAID 5 або ZFS-пул).

Для системного розділу доцільно використовувати SSD-накопичувачі, які забезпечують високу швидкодію при завантаженні ОС та обробці баз даних, які використовує Plex.

Також важливою є організація резервного копіювання — навіть у домашніх умовах доцільно передбачити дублювання даних чи періодичну реплікацію на зовнішній диск або інше сховище.

Не менш важливою складовою є мережева підсистема. Для внутрішнього передавання потоків у локальній мережі достатньо 1 Gbps Ethernet, але для масштабованих рішень або роботи з кількома HD/UHD потоками одночасно рекомендується 2.5 Gbps або 10 Gbps інтерфейс.

Особливо критичним є параметр затримки (latency) при трансляції через Інтернет: Plex використовує HTTPS або Relay-сервери, тому стабільне підключення з мінімальними втратами пакетів суттєво покращує якість потокового відтворення. Варто також звернути увагу на стабільність маршрутизатора, можливість пробросу портів (port forwarding) або підтримку NAT-PMP / UPnP для забезпечення прямого доступу до сервера без посередництва хмарних Relay-серверів Plex.

Формування вимог до серверного обладнання базується не лише на теоретичних характеристиках Plex, а й на практичному розумінні його

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 45
Зм.	Арк.	№ докум.	Підпис	Дата		

навантаження: кількість користувачів, типи контенту, кількість одночасних сесій, спосіб доступу (локально чи віддалено).

Правильне компонування CPU, RAM, дисків та мережі дозволяє побудувати збалансовану, масштабовану систему хмарного мультимедійного сервера на базі Linux з технологією Plex.

2.10 Висновки до другого розділу

У другому розділі зосереджено увагу на проектуванні системи. Проведено порівняльний аналіз існуючих рішень хмарного зберігання, таких як Nextcloud, Seafile, Syncthing та Plex, що дозволило обґрунтувати вибір Plex як оптимального рішення для мультимедійного середовища.

Розроблено архітектуру Plex Media Server, визначено протоколи обміну даними (HTTPS, DLNA, власні API), що дозволяє забезпечити сумісність з клієнтськими пристроями.

Обрано операційну систему OpenSUSE на базі Linux, яка відповідає вимогам до стабільності, безпеки та відкритості. Також сформульовано вимоги до апаратного забезпечення, враховуючи типове навантаження, кількість одночасних підключень і специфіку використання. Отже, було закладено архітектурну основу для подальшої реалізації.

					КВРКІ. 200024.21.01.01 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

3 ПРОГРАМНО-АПАРАТНА РЕАЛІЗАЦІЯ ХМАРНОГО СХОВИЩА НА БАЗІ LINUX З ТЕХНОЛОГІЄЮ PLEX

3.1 Архітектура програмно-технічного засобу. Схема взаємодії компонентів: сервер, Plex, файлове сховище, користувач

Архітектура програмно-технічного засобу, який реалізує хмарне сховище на базі Linux з інтеграцією Plex, побудована за принципами модульності, розподіленої відповідальності та клієнт-серверної моделі взаємодії. Основу системи становить серверна частина, розгорнута на Linux-дистрибутиві (у цьому проєкті — openSUSE), яка поєднує в собі обчислювальні, мережеві та сховищні функції. До цієї серверної платформи підключаються користувацькі пристрої через офіційний Plex-клієнт або сумісні технології, такі як DLNA або HTTPS.

У центрі взаємодії знаходиться Plex Media Server, який виконує роль програмного медіа-контролера. Він індексує дані з локального файлового сховища, автоматично сканує структуру каталогів, генерує мініатюри, обкладинки, метадані (жанри, рік, актори, опис) і формує медіабібліотеку, доступну користувачу через інтерфейс клієнта. Plex не зберігає файли у власному форматі — замість цього він здійснює логічне представлення медіаконтенту, який фізично розміщується у файловій підсистемі, побудованій на традиційних файлових системах Linux, таких як ext4 або XFS. Ця підсистема доступна Plex через відповідні шляхи до каталогів, які вказуються під час налаштування бібліотек.

Файлове сховище реалізовано як частина локальної дискової підсистеми серверу. Воно може бути структуровано як звичайна монолітна директорія або як RAID/ZFS-масив з високим ступенем надійності. У разі потреби можлива організація спільного доступу до сховища з інших систем за допомогою Samba, NFS або WebDAV, однак для Plex основним джерелом даних залишається локально змонтована директорія, що забезпечує максимальну швидкість при потоковому читанні.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

На іншому кінці архітектури розміщено користувацькі пристрої — це можуть бути смарт-телевізори, смартфони, планшети, ПК, ігрові приставки, медіаплеєри. Кожен із них взаємодіє з Plex Media Server через спеціалізовані клієнти або браузер через HTTPS. При запиті на відтворення контенту сервер аналізує характеристики клієнтського пристрою (дозвіл, підтримувані кодеки, пропускну здатність мережі), після чого приймає рішення про необхідність перекодування. У разі необхідності перекодування, Plex запускає фоновий процес, який трансформує файл у режимі реального часу, адаптуючи його під технічні параметри клієнта. Це призводить до навантаження на CPU і, залежно від налаштувань, використання кешу в RAM або тимчасових файлів на диску.

Ключовим елементом, який забезпечує зв'язок між сервером і клієнтом через Інтернет, є хмарна служба Plex Cloud, яка виконує роль посередника для автентифікації, передачі керувальних команд.

У випадку, коли прямий доступ неможливий, то проксирування медіапотоку через захищений Relay TLS-канал.

При цьому сама передача контенту залишається end-to-end зашифрованою, а Plex Cloud не зберігає самі медіафайли.

Усі компоненти взаємодіють у єдиному логічному ланцюгу: сервер приймає запити від клієнта, перевіряє права доступу, формує або транслює контент, використовуючи ресурси файлового сховища та обчислювального ядра.

Завдяки відкритій системній архітектурі Linux, кожен з компонентів може бути детально налаштований: наприклад, серверна частина може бути розгорнута в контейнері, файлове сховище автоматично монтуватись при старті, а мережеві правила можуть забезпечувати пріоритетність потокового трафіку.

Таким чином, архітектура програмно-технічного засобу передбачає чітку сегментацію ролей:

- Файлове сховище відповідає за фізичне збереження користувацького контенту.
- Plex відповідає за логіку доступу та трансляцію.

- Linux-сервер відповідає за координацію та безпеку.
- Користувач відповідає за взаємодію з бібліотекою через клієнтський інтерфейс.

Це дозволяє створити гнучку, масштабовану та технологічно сучасну хмарну медіасистему.

Схема архітектури взаємодії компонентів: сервер, Plex Media Server, файлове сховище та користувач подано на рисунку 3.1.

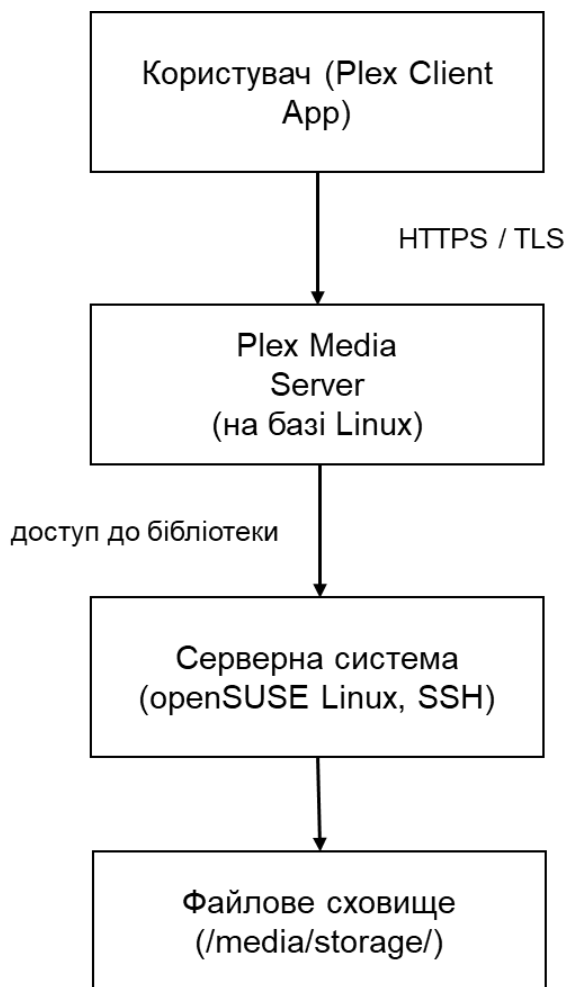


Рисунок 3.1 – Схема архітектури взаємодії компонентів: сервер, Plex Media Server, файлове сховище та користувач

Архітектура дозволяє централізовано керувати вмістом, забезпечуючи безперервний доступ користувачів з будь-якої точки світу.

Користувач підключається до Plex Media Server через клієнтський застосунок або вебінтерфейс.

Plex Media Server обробляє запит, звертається до файлового сховища, зчитує потрібні дані та, за потреби, перекодує їх.

Серверна ОС Linux (openSUSE) керує ресурсами, мережею, сховищами, а також забезпечує стабільну роботу Plex.

Файлове сховище містить мультимедійний контент, доступний Plex для трансляції.

3.2 Розроблення системного програмного забезпечення

Розглянемо системного програмного забезпечення, яке реалізує системний скрипт.

Він перевіряє, чи є завдання на сканування нових фільмів або серіалів у файлі `plex-scan`, і для кожного з них запускає Plex Media Scanner із вказаним шляхом та відповідною секцією бібліотеки.

Після обробки записів він очищає їх з файлу, щоб не сканувати повторно.

Подемо скрипт псевдокодом з кроками:

1. Ініціалізація шляху до каталогу скрипта

Встановити змінну `ple_dir` як каталог, де знаходиться поточний скрипт

Перейти до каталогу `ple_dir`

2. Зчитування конфігураційного файлу

Якщо у поточному каталозі існує файл `ple.conf`, то

Завантажити змінні з цього файлу

Інакше:

Вивести повідомлення про відсутність конфігурації

Записати помилку в лог `upload-error`

Завершити виконання скрипта

Логування старту сканування:

Додати розділювач у файл plex-scan.log

Записати дату та повідомлення "Starting plex scan" у лог-файл

3. Перевірка наявності завдань для сканування

Якщо файл plex-scan не є порожнім:

Якщо шляхи до серіалів або фільмів не змонтовані, але відповідна опція увімкнена:

Записати повідомлення про помилку у лог

Завершити скрипт

4. Читання списку завдань із файлу plex-scan

Для кожного рядка у файлі plex-scan:

Зчитати рядок як line

Розділити рядок по символу ":" на масив current

Якщо перший елемент current — це "tv":

Змінна show = другий елемент

Змінна season = третій елемент

Побудувати шлях до папки серіалу

Встановити змінну середовища LD_LIBRARY_PATH

Виконати Plex Media Scanner для сканування цієї папки серіалу

Записати завершення сканування у лог

Інакше якщо перший елемент current — це "mov":

Змінна folder = другий елемент

Побудувати шлях до папки фільму

Встановити LD_LIBRARY_PATH

Виконати Plex Media Scanner для сканування фільму

Записати завершення сканування у лог

Записати у лог, що рядок було проскановано

Збільшити лічильник n на 1

5. Очищення просканованих рядків

Видалити перші n рядків із файлу plex-scan

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 51
Зм.	Арк.	№ докум.	Підпис	Дата		

3.3 Алгоритм функціонування системи з використанням Plex, Rclone і Cloud Drives

Розглянемо алгоритм функціонування програмно-апаратна реалізація хмарного сховища на базі linux з технологією plex.

Алгоритм функціонування системи з використанням Plex, Rclone і Cloud Drives включає кроки:

1. Підготовка хмарного сховища через Rclone.
2. Завантаження мультимедійних файлів у "вхідну" директорію.
3. Автоматичне перенесення файлів у хмару
4. Запис інформації про нові файли у чергу сканування
5. Перевірка монтування хмарного сховища
6. Сканування Plex тільки вказаних директорій
7. Завершення процесу

На початковому етапі системний адміністратор або користувач налаштовує доступ до одного чи кількох хмарних сховищ (наприклад, Google Drive) через інструмент rclone.

Це дозволяє системі працювати з хмарою як з віддаленим файловим сховищем, монтуючи її як частину файлової системи.

Користувач або інша підсистема (наприклад, автоматизований завантажувач) додає нові медіафайли (фільми або серіали) до спеціальної локальної директорії завантаження.

Ця директорія не є частиною бібліотеки Plex, а використовується як буфер для подальшої обробки.

Періодично або подієво запускається скрипт (наприклад, через cron або systemd.timer), який:

- аналізує вміст вхідної директорії;
- визначає нові або змінені файли;

- копіює або переміщує ці файли до відповідної директорії у Google Drive (або іншому хмарному сховищі) за допомогою ``rclone copy`` або ``rclone move``;
- при використанні кількох акаунтів Google Drive, система може дублювати файли у два або більше сховища за принципом RAID 1, тобто для резервування.

Після успішного завантаження скрипт додає запис до файлу-черги (наприклад, ``plex-scan``) із зазначенням:

- типу медіа (``tv`` або ``mov``);
- назви шоу/фільму;
- номера сезону (для серіалів);
- запис структурується у форматі ``tv:ShowName:Season 1`` або ``mov:MovieFolder``.

Окремий скрипт перевіряє, чи хмарне сховище змонтоване у файлову систему через ``rclone mount``. Це важливо, оскільки Plex повинен мати прямий доступ до файлів для індексації.

Замість повного сканування всієї бібліотеки (що викликає перевищення API-квоти Google), скрипт:

- читає чергу ``plex-scan``;
- для кожного запису визначає точний шлях до нової директорії у хмарному сховищі (наприклад, ``/mnt/plexdrive/TV/ShowName/Season 1``);
- запускає Plex Media Scanner із параметрами: ``--scan`` та ``--refresh`` для оновлення; ``-d`` із вказанням конкретного шляху; ``-c`` із номером відповідної бібліотеки (фільми або серіали);
- після успішного сканування видаляє оброблений запис із черги, щоб уникнути повторів.

Система оновлює журнали (``plex-scan.log`` та ``upload-error``) із результатами операцій – це дозволяє вести аудит та діагностику.

Завдяки тому, що сканується лише новододана частина медіабібліотеки, а не вся колекція, значно зменшується кількість запитів до Google Drive API.

Це дозволяє уникати помилок типу "API quota exceeded", які часто виникають при великих бібліотеках у Plex.

Файли можуть одночасно копіюватися у кілька Google Drive акаунтів, що дає надійність при втраті доступу до одного облікового запису дані залишаються на іншому, а також розвантаження API шляхом сканування, яке може розподілятися по акаунтах.

Система поєднує автоматизацію завантаження, інкрементне сканування бібліотеки Plex, резервування даних та економію API-квоти, що робить її високоефективним рішенням для хмарного мультимедійного сховища на основі Plex.

Блок-схема алгоритму функціонування системи з використанням Plex, Rclone і Cloud Drives подана на рисунку 3.2.

3.4 Інсталяція системного програмного забезпечення

Розглянемо детальний опис процесу інсталяції системного програмного забезпечення в системі Linux для проєкту хмарного мультимедійного сховища з Plex. Цей опис подає інсталяцію в етапах із поясненням кожної дії.

Перелік етапів наступний:

1. Клонування репозиторію системного ПЗ.
2. Надання прав доступу.
3. Інсталяція Rclone.
4. Конфігурація облікових записів Google Drive.
5. Монтування хмарного сховища у файлову систему.
6. Визначення секцій бібліотеки Plex.

Першим кроком є отримання коду проєкту, що містить сценарії керування системою. Це виконується за допомогою команди `git clone`, яка завантажує вміст репозиторію GitHub до домашньої директорії користувача.

```
```bash
git clone https://github.com/roni/PLE
```

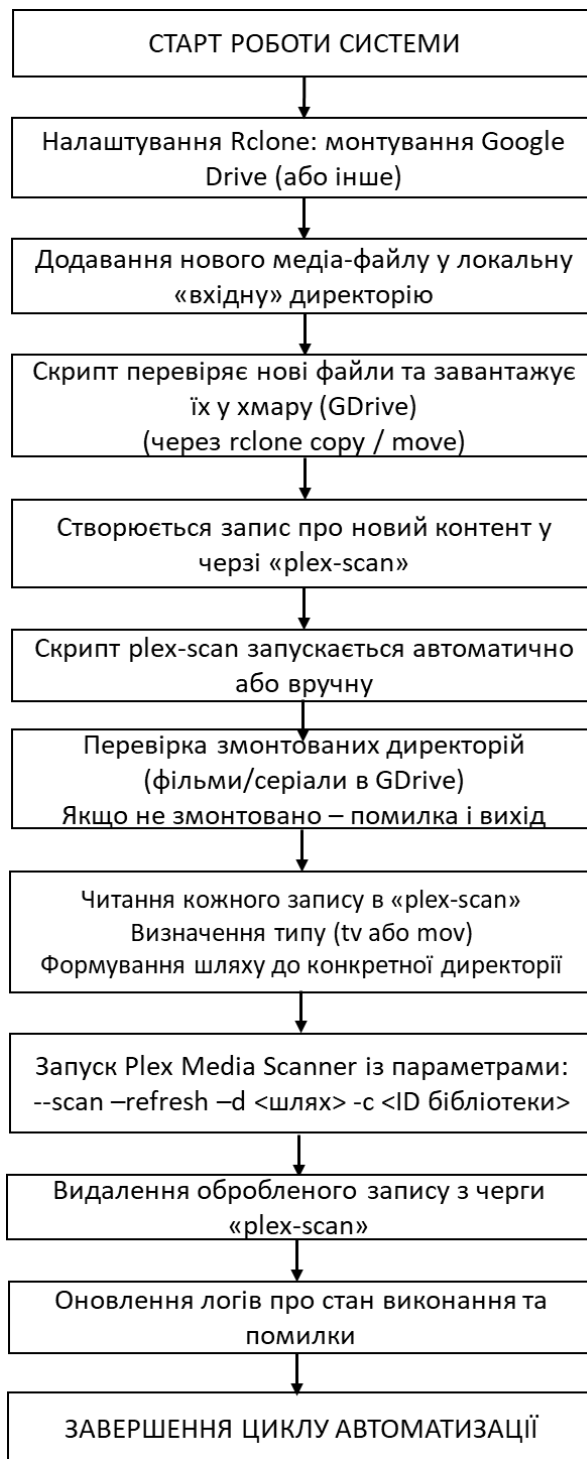


Рисунок 3.2 – Блок-схема алгоритму функціонування системи з використанням Plex, Rclone і Cloud Drives

Після виконання цієї команди створюється підкаталог 'PLE' з усіма необхідними скриптами. Це основна структура, яка керуватиме завантаженням,

монтуванням та скануванням контенту в Plex. Оскільки Plex виконується під окремим системним користувачем (`plex`), необхідно надати повний доступ до скриптів та конфігурацій репозиторію цьому користувачу.

Команда `chmod -R 777` надає повні права на читання, запис і виконання для всіх користувачів — це тимчасовий захід на етапі налаштування:

```
```bash
sudo chmod -R 777 PLE
``
```

У реальних умовах на продакшн-системах слід застосовувати більш безпечні налаштування прав.

`rclone` - це утиліта, яка забезпечує синхронізацію, копіювання та монтування хмарних сховищ, таких як Google Drive. Спочатку необхідно завантажити останню версію `rclone` з офіційного сайту (або за допомогою `curl`, `wget`, `snap` чи `apt`).

Далі виконуються такі дії:

- переміщення бінарного файлу до системної директорії, яка вже прописана у змінній середовища `\$PATH`, наприклад `/usr/local/bin/`;
- встановлення прав доступу для цього файлу.

Встановлення прав доступу передбачає виконання таких системних команд в консолі:

```
```bash
sudo mv rclone /usr/local/bin/
sudo chown root:root /usr/local/bin/rclone
sudo chmod 755 /usr/local/bin/rclone
``
```

Після цього `rclone` буде доступний з будь-якої точки системи.

`rclone config` дозволяє створити віддалені підключення (ремоути) до облікових записів Google Drive.

					КВРКІ. 200024.21.01.01 ПЗ	Арк. 56
Зм.	Арк.	№ докум.	Підпис	Дата		

У процесі конфігурації задається ім'я кожного підключення (наприклад, `gdrive-main`) і проходить авторизація через OAuth 2.0.

Результати зберігаються у `~/.config/rclone/rclone.conf`.

Для того, щоб Plex мав змогу безпосередньо читати файли з Google Drive, необхідно змонтувати ремуот за допомогою `rclone mount`.

Це забезпечує доступ до хмари як до локальної директорії:

```
```bash
sudo mkdir /mnt/gdr-main
sudo rclone mount --allow-non-empty --allow-other gdrive-
main:/ /mnt/gdr-main &
```
```

Прозглянемо використовувані параметри:

`--allow-non-empty` дозволяє монтувати директорію, навіть якщо вона не порожня;

`--allow-other` дозволяє іншим користувачам (наприклад, `plex`) мати доступ до змонтованої директорії.

Як альтернативу можна використати ple, що також забезпечує монтування, але має вбудований кеш для роботи з Plex.

У випадку зашифрованого монтування (через `rclone crypt`) важливо переконатися, що Plex сканує розшифровану директорію, а у конфігураційних файлах використовуються назви саме зашифрованих ремуотів.

Для того щоб вказати правильні бібліотеки (Movies / TV Shows), потрібно визначити внутрішні ідентифікатори цих бібліотек у Plex.

Для цього викликається вбудований сканер з параметром `--list` під користувачем `plex`:

```
```bash
sudo su -c 'export LD_LIB_PATH=/usr/lib/ple;
/usr/lib/ple/Pl\ Media\ Scan --list' ple
```
```

Вивід виглядатиме так:

...

1: Movies

2: TV Shows

...

Ці номери (наприклад, `1`, `2`) відповідають параметрам `ple\_movies\_sec\_num` та `ple\_tvs\_sec\_num`, які необхідно записати в конфігураційний файл `ple.conf`. Після завершення всіх вищезгаданих етапів система готова до автоматизованого завантаження мультимедіа в Google Drive та сканування лише нових директорій у Plex. Це дозволяє зменшити навантаження на API Google, зберегти продуктивність і оптимізувати ресурсоспоживання системи. Блок-схема алгоритму інсталяції СПЗ подано на рис. 3.3.

### 3.5 Застосування розробленої програмно-апаратна реалізації хмарного сховища на базі linux з технологією Plex

Розроблена програмно-технічна система забезпечує інтеграцію Plex Media Server із хмарними сховищами на базі rclone, використовуючи Google Drive як основний носій мультимедійних даних.

Для запуску і повсякденного використання система передбачає просту взаємодію користувача з двома основними скриптами: завантаженням мультимедійного контенту у хмару та ініціалізацією сканування бібліотеки Plex на наявність нових файлів.

Після попередньої конфігурації файлу `ple.conf`, яка включає в себе вказання шляхів до локальних директорій з медіаконтентом, налаштування віддалених Google Drive облікових записів, параметри видалення файлів після завантаження, дозволені типи файлів, а також номерів секцій бібліотеки Plex, користувач може запускати скрипт `ple.sh`.

Цей скрипт виконує автоматичне виявлення нових файлів у локальних директоріях, перевірку їхніх типів, завантаження на вказані Google Drive в одному

або кількох екземплярах подібно до RAID1 за допомогою rclone, і, за необхідності, очищення локального сховища після успішної передачі.

Після завершення завантаження даних користувач запускає інший скрипт `ples.sh`.

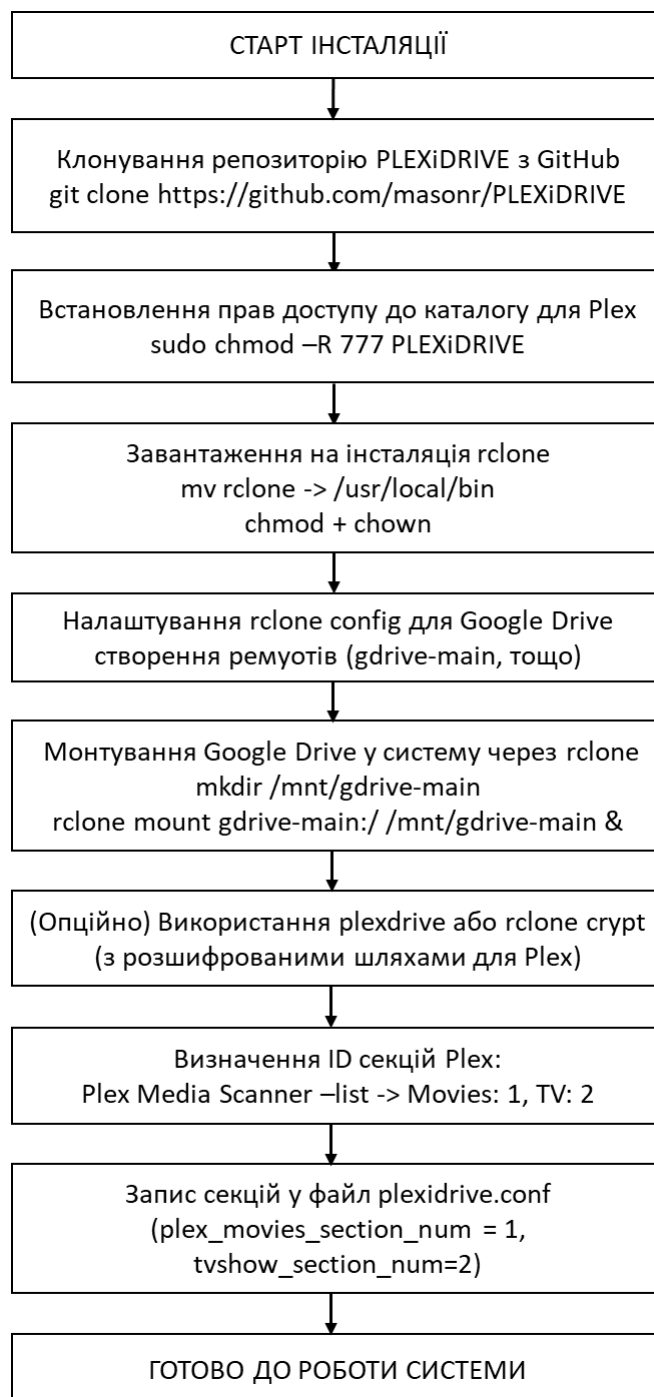


Рисунок 3.3 – Блок-схема алгоритму інсталяції системного програмного забезпечення

Цей скрипт виконується від імені системного користувача `ple`, і викликає вбудований сканер Ple, який ініціалізує оновлення конкретної секції медіатеки (наприклад, лише фільмів або лише певного сезону серіалу), замість повного сканування всієї бібліотеки.

Такий підхід дозволяє уникнути перевищення лімітів API викликів Google Drive, що є критичним при роботі з великими колекціями медіа.

Для забезпечення безперервної роботи системи передбачено можливість автоматизації через планувальник задач `cron`.

Користувач налаштовує два незалежні `cron job`:

- перший запускає `ple.sh` з інтервалом, наприклад, кожні 4 години
- другий `ples.sh` від імені користувача `ple` одразу після завершення завантаження.

Така організація гарантує, що нові медіафайли регулярно додаються до бібліотеки та одразу ж стають доступними для відтворення без ручного втручання користувача.

Таким чином, система забезпечує повний цикл роботи від розміщення медіафайлів у локальному середовищі до їх доступності у Plex, інтегрованому з хмарною інфраструктурою.

Уся логіка функціонування централізується навколо гнучкого конфігураційного файлу `ple.conf`, де користувач може:

- змінювати шляхи;
- змінювати ідентифікатори бібліотек;
- модифікувати список підтримуваних форматів;
- вказувати кількість Google Drive, що використовуються;
- задавати параметри поведінки після завантаження.

Це дозволяє адаптувати систему під індивідуальні потреби без необхідності модифікації програмного коду.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>60 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

### 3.6 Налаштування конфігурації з одним сховищем

Розглянемо опис процесу налаштування конфігурації системи з одним хмарним сховищем (на прикладі Google Drive).

Для коректної роботи розробленої системи з одним обліковим записом Google Drive необхідно виконати налаштування конфігураційного файлу ``ple.conf``, який є ключовим елементом управління усією системою.

У цьому файлі визначаються параметри підключення до хмарного сховища, специфікація обробки медіафайлів, шляхи до бібліотек Plex, локальні шляхи до медіа, а також режими активності окремих підсистем.

Першим етапом є зазначення кількості хмарних облікових записів, що будуть використовуватись для завантаження медіа.

У випадку одного сховища змінна ``num_of_gdrives`` встановлюється в значення 1.

Відповідно, в масиві ``drive_names`` вказується лише одне ім'я, яке відповідає імені ремоуту, попередньо налаштованого через ``rclone config``.

Наприклад, для основного Google Drive ім'я ремоуту може бути ``gd-main``.

Наступним блоком у конфігурації є опції, що визначають поведінку скрипта завантаження.

Зокрема, змінна ``del_a_up`` дозволяє встановити, чи потрібно видаляти локальні файли після успішного копіювання їх у хмару.

Це корисно у випадках, коли важливо зекономити дисковий простір. В даній конфігурації значення встановлено в ``true``, тобто система автоматично очищатиме локальні каталоги після завершення передачі.

Змінна ``fil_tp`` визначає перелік форматів відеофайлів, які підлягають обробці: наприклад, ``mkv``, ``avi``, ``mp4``, ``m4v``, ``mpg``, ``wmv``, ``flv``.

Це дозволяє уникнути випадкової обробки несумісних або інших службових файлів.

Параметр ``rcl_conf`` у цьому випадку залишено порожнім, тобто буде використовуватись стандартний шлях до конфігурації `rclone` (зазвичай ``~/config/rclone/rcl.conf``).

Якщо ж потрібне використання нестандартного шляху, його слід вказати в цьому місці.

Далі вказуються шляхи до директорій Plex, які є результатом монтування хмарного сховища за допомогою `rclone`.

У даному випадку змінна ``ple_t_p`` вказує на ``/mnt/mVs``, а ``ple_mp`` на ``/mnt/main/Ms``.

Це ті шляхи, які повинні бути відображені в Plex при створенні системних бібліотек.

Обов'язковою умовою є відсутність символу ``/`` наприкінці шляху, що враховується у скриптах для уникнення дублювання роздільників у кінцевих шляхах.

Змінні ``ple_ms_`` та ``plets_`` відповідають ідентифікаторам бібліотек Plex, які визначаються окремо через спеціальну команду Plex.

Окрім того, необхідно вказати шляхи до локальних каталогів, де зберігаються вихідні відеофайли перед завантаженням у хмару.

Для телесеріалів це шлях ``/home/masonr/tv-shows/``, а для фільмів ``/home/roni/vid/``.

У цьому випадку, навпаки, символ ``/`` наприкінці є обов'язковим для коректної обробки у скриптах.

У конфігурації активуються відповідні підсистеми.

Змінна ``ene_sup`` визначає, чи потрібно шукати й завантажувати телесеріали, а ``enupls`` визначає фільми.

Обидві змінні мають значення ``true``, тобто система одночасно оброблятиме обидва типи контенту.

За допомогою цього налаштування система повністю готова до автоматичної роботи з одним Google Drive, що дозволяє забезпечити безперервне завантаження

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>62 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

нового контенту у хмарне сховище, його подальше сканування Plex та доступ кінцевих користувачів до актуальної бібліотеки мультимедійних матеріалів.

### 3.7 Налаштування конфігурації з кількома хмарними сховищами

Розглянемо детальний опис процесу налаштування конфігурації з двома хмарними сховищами.

У випадку розгортання розробленої системи з двома Google Drive-сховищами, метою є не лише зберігання контенту, а й забезпечення відмовостійкості шляхом дублювання медіа на два незалежні хмарні облікові записи.

Така архітектура нагадує принцип RAID 1: кожен файл, після завантаження, буде автоматично продубльований на друге сховище, що значно підвищує надійність зберігання даних.

Весь процес керується через конфігураційний файл `ple.conf`.

Розглянемо секцію, що містить параметри GDR.

У першій частині конфігурації вказується кількість сховищ, до яких буде здійснюватись завантаження.

Змінна `ngdr` встановлюється в `2`, що означає використання двох хмарних облікових записів.

Далі в масиві `drm` задаються імена `rclone`: `gdrive-main` - є основний, та `gdrive-backup` - резервний.

Ці імена мають точно відповідати тим, що були попередньо сконфігуровані у `rclone config`.

Розглянемо секцію, що містить опції обробки файлів.

Змінна `delup` встановлена як `true`, що передбачає автоматичне видалення локальних медіафайлів після їх повного завантаження на обидва диски.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>63 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Це корисно у випадках, коли обсяг локального сховища обмежений. `file\_types` містить розширення відеофайлів, які система має виявляти і переносити у хмару.

Список форматів розширено до `mkv`, `avi`, `mp4`, `m4v`, `mpg`, `wmv`, `flv`, `mpeg`, що охоплює практично всі основні формати мультимедійного контенту користувача.

Значення змінної `rclone\_conf` явно вказує на повний шлях до користувацького конфігураційного файлу `rclone: /home/roni/.conf/rclone/rcl.conf`.

Це важливо для ситуацій, коли конфігурація була створена під окремим обліковим записом користувача, а не в системному просторі.

Розглянемо секцію, що містить параметри Plex/

Наступним етапом є прив'язка змонтованих директорій до відповідних бібліотек Plex.

У такій конфігурації шляхи для фільмів і телешоу різні: `pleoath` посилається на основний диск (`/mnt/mTS`), тоді як `pler` - на резервне хмарне сховище (`/mnt/bacMs`).

Такий підхід дає змогу розділяти контент користувача не лише логічно, але й фізично.

Plex має бути попередньо налаштований так, щоб вказані каталоги відповідали бібліотекам типу TV Shows і Movies.

Ідентифікатори бібліотек вказуються за допомогою змінних `pleo\_sm` та `plesen\_m` у цьому конфігурації це відповідно 2 та 1, що означає другий і перший розділ у Plex.

Задання локальних директорій медіа здійснюється таким чином. Система очікує, що нові відеофайли будуть завантажені до локальних директорій.

Шлях до телесеріалів встановлений у `locra` в `/home/roni/tvs/`, а шлях до фільмів - у `locr` в `/home/roni/vid/`.

Символ `^` наприкінці вказує на те, що це саме директорії, а не окремі файли чи відносні шляхи.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>64 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Сканування цих каталогів дозволяє виявляти нові файли для копіювання.

На завершення, активуються відповідні підсистеми: ``ene_shads=true`` та ``eneps=true`` означають, що система буде обробляти як телесеріали, так і фільми користувача.

Якщо будь-який з цих параметрів встановити в ``false``, відповідна категорія контенту ігноруватиметься.

Конфігурація з двома сховищами забезпечує баланс між ефективністю і надійністю.

Вся система працює прозоро для кінцевого користувача Plex, водночас автоматизуючи рутинні операції і мінімізуючи ризики втрати даних завдяки резервному дублюванню.

### 3.8 Автоматизація використання одного або декількох облікових записів Google Drive як сховища для сервера Plex

Для здійснення автоматизації використання одного або декількох облікових записів Google Drive як сховища для сервера Plex було розроблено спеціалізоване системне програмне забезпечення у вигляді скрипту ``pled.sh``.

Він є центральним елементом системи автоматизації завантаження мультимедійних файлів на хмарні сховища та подальшої інтеграції цих файлів із Plex-сервером.

Його функціонування охоплює кілька ключових етапів:

- ініціалізація;
- перевірка середовища;
- обробку конфігурації;
- запуск допоміжних сценаріїв для завантаження даних;
- ініціювання сканування Plex бібліотек.

Розглянемо процес перевірки унікальності виконання.

На першому етапі скрипт перевіряє, чи не запущений він уже в іншому процесі.

Це досягається шляхом перегляду списку всіх процесів, які виконують скрипт з іменем ``pled.sh``, і порівняння їхніх PID із поточним процесом.

Якщо виявлено, що інша копія скрипта вже працює, виконання переривається.

Такий підхід запобігає одночасному завантаженню тих самих файлів кількома процесами, що може призвести до дублювання або пошкодження використовуваних даних.

Ініціалізація середовища виконується так: визначається шлях до поточної директорії, де зберігається скрипт, і виконується перехід до неї.

Це важливо для того, щоб усі подальші відносні шляхи працювали коректно незалежно від того, з якого місця викликається скрипт.

Розглянемо процес обробки конфігураційного файлу.

Сценарій перевіряє наявність конфігураційного файлу ``pled.conf``.

Якщо файл існує, його вміст підключається до середовища виконання через команду ``source``, що дозволяє системному скрипту використовувати всі змінні, задані в ньому.

У разі відсутності конфігураційного файлу виводиться повідомлення про помилку як на екран, так і в лог-файл ``upload-error``, після чого скрипт завершує свою роботу.

Розглянемо процес підготовки службових файлів

На цьому етапі створюються порожні службові файли (``ples``, ``ple.log``, ``upload-error``), якщо вони ще не існують.

Це забезпечує стабільну роботу скриптів, які можуть записувати або зчитувати дані з цих файлів.

Розглянемо процес перевірки узгодженості параметрів

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
|     |      |          |        |      |                           | 66   |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |      |

Далі виконується критично важлива перевірка: кількість облікових записів Google Drive, зазначена в `ngd`, повинна відповідати фактичній кількості імен дисків, указаних у масиві `ds`.

У разі невідповідності скрипт зупиняється, фіксуючи помилку у файлі `upload-error`.

Це захищає систему від некоректної роботи внаслідок неповної або неправильної конфігурації.

Розглянемо процес завантаження телешоу.

Якщо параметр `enads` у конфігураційному файлі встановлений у `true`, то запускається допоміжний скрипт `uws.sh`, який відповідає за пошук нових епізодів телешоу в локальній директорії та їхнє завантаження до всіх зазначених облікових записів Google Drive.

Після завершення перевіряється код повернення цього скрипта.

Якщо завантаження завершилося з помилкою, основний сценарій припиняє роботу, виводячи відповідне повідомлення.

Розглянемо процес завантаження фільмів.

За аналогічною логікою працює блок, що обробляє фільми.

Якщо активовано `enads`, викликається скрипт `uploies.sh`.

Якщо цей скрипт повертає ненульовий код (тобто сталася помилка), основний сценарій завершує виконання, щоб уникнути подальших некоректних дій.

Розглянемо процес запуску сканування бібліотек Plex.

Код передбачає можливість автоматичного запуску скрипта `plx.sh` після завершення завантажень.

Цей скрипт виконується від імені користувача `ple` і оновлює бібліотеки Plex, використовуючи CLI-інтерфейс.

Його використання можливе лише тоді, коли параметр `plex\_scan\_after\_upload` явно активовано в конфігурації.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>67 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Скрипт `pled.sh` виконує роль керуючого механізму в розробленій системі, забезпечуючи:

- - однозначність виконання;
- - контроль відповідності конфігурації;
- - керування окремими завантаженнями контенту за категоріями;
- - гнучку інтеграцію з Plex для забезпечення актуальності медіабібліотек.

Завдяки структурованій логіці та передбаченню помилок, скрипт дозволяє автоматизувати ключові етапи роботи з мультимедійним сервером у середовищі з хмарним зберіганням.

Запуск скрипта в операційній системі OpenSuse показано на рисунку 3.4.

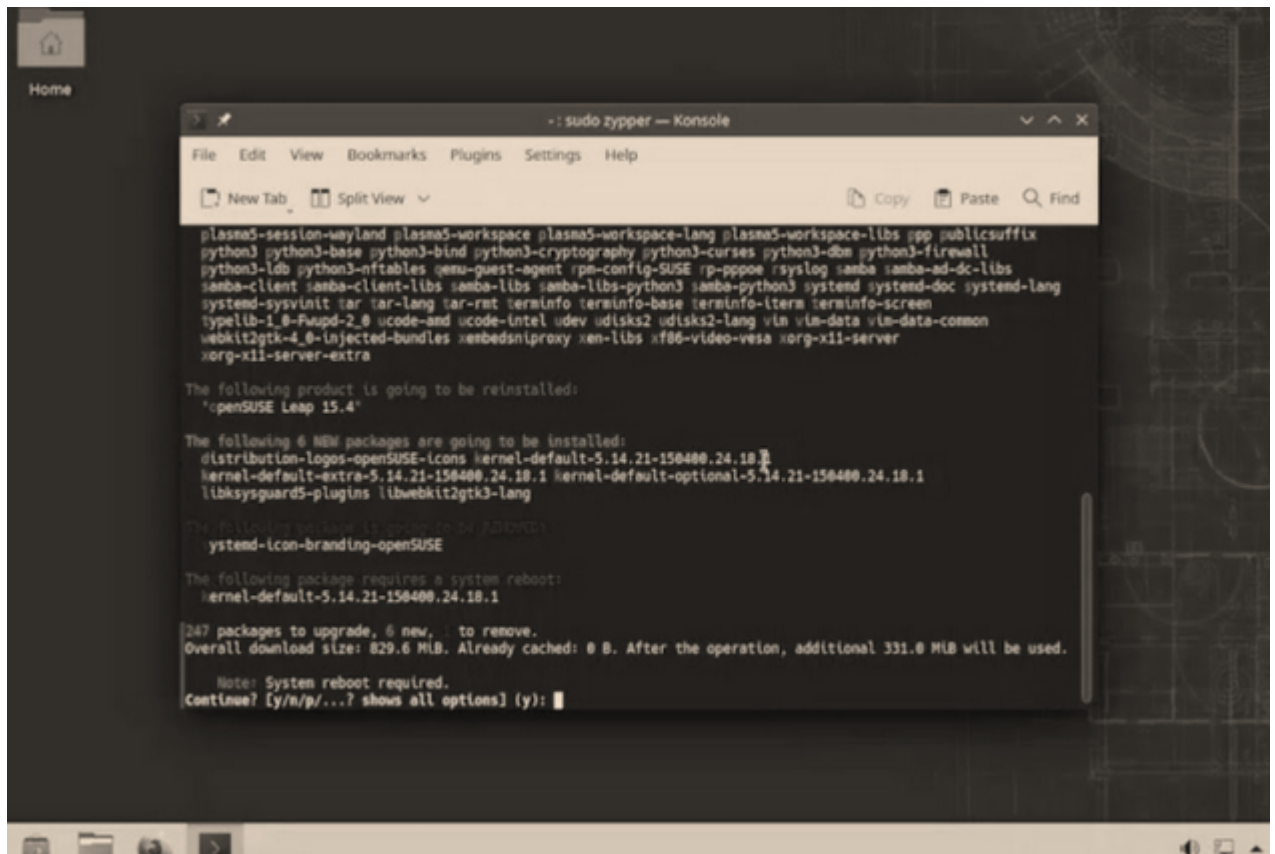


Рисунок 3.4 – Запуск скрипта

### 3.9 Висновки до третього розділу

У третьому розділі здійснено програмно-апаратну реалізацію хмарного сховища. Розроблено архітектуру взаємодії серверної частини з Plex, файловим сховищем та користувачем.

Впроваджено програмне забезпечення, налаштовано Rclone для синхронізації з хмарними сервісами Google Drive, автоматизовано інсталяцію компонентів та запуск оновлення медіабібліотек. Описано алгоритм функціонування системи, враховуючи сценарії з одним і кількома обліковими записами, а також реалізовано механізми автоматичного запуску скриптів для оновлення бібліотек Plex.

Система забезпечує високу адаптивність, простоту адміністрування та безперебійну роботу з мультимедійним контентом у хмарному середовищі.

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
|     |      |          |        |      |                           | 69   |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |      |

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було досягнуто поставлену мету - розроблено та реалізовано програмно-технічний засіб хмарного сховища на базі Linux з використанням технології Plex.

На основі глибокого теоретичного аналізу та порівняння сучасних підходів до хмарного зберігання даних, було обґрунтовано вибір відповідної технології та програмно-апаратної архітектури.

Реалізована система поєднує переваги відкритого ПЗ, контейнеризації, автоматизації та хмарної інтеграції, що дозволяє забезпечити ефективно, безпечно та масштабоване управління мультимедійним контентом.

Результати роботи можуть бути використані для впровадження у домашніх мультимедійних серверах, освітніх проєктах або малих організаціях, які потребують ефективного стрімінгу й збереження даних без залежності від комерційних провайдерів.

У першому розділі проведено детальний аналіз програмно-технічних засобів, які дозволяють ефективно реалізувати хмарне сховище. Встановлено, що використання хмарних середовищ є сучасною та економічно доцільною практикою, яка забезпечує гнучкий доступ до даних, масштабованість, підвищений рівень безпеки та зниження витрат на інфраструктуру.

У другому розділі зосереджено увагу на проєктуванні системи. Проведено порівняльний аналіз існуючих рішень хмарного зберігання, таких як Nextcloud, Seafile, Syncthing та Plex, що дозволило обґрунтувати вибір Plex як оптимального рішення для мультимедійного середовища.

У третьому розділі здійснено програмно-апаратну реалізацію хмарного сховища. Розроблено архітектуру взаємодії серверної частини з Plex, файловим сховищем та користувачем.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>70 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Hernandez S, Lazcano R, Serrano A, Powell S, Kostousov L, Mehta J, Solis LM. Challenges and opportunities for immunoprofiling using a spatial high-plex technology: the NanoString GeoMx® digital spatial profiler. *Frontiers in Oncology*. 2022. Vol. 12. pp. 890410.
2. Ramamurthy M, Kumar SM, Palaniappan S. An approach for cloud linux server security and management. *Annals of the Romanian Society for Cell Biology*. 2021. Vol. 25(3). pp. 2453–2459.
3. Oleksiuk V, Spirin O. The experience of using cloud labs in teaching Linux operating system. *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications*. 2021. pp. 281–291.
4. ThiBac D, Minh NH. Design of network security storage system based on under cloud computing technology. *Computers and Electrical Engineering*. 2022. Vol. 103. pp. 108334.
5. Wertenbroek R, Thoma Y, Dassatti A. A portable linux-based firmware for NVMe computational storage devices. *ACM Transactions on Storage*. 2025. Vol. 21(2). pp. 1–36.
6. Zhang X, Zhu F, Li S, Wang K, Xu W, Xu D. Optimizing performance for open-channel SSDs in cloud storage system. *2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 2021. pp. 902–911.
7. Singh SK. Linux yourself: concept and programming. Chapman and Hall/CRC. 2021. P. 474.
8. Ward B. How Linux works: What every superuser should know. No Starch Press. 2021. P. 421.
9. Kiran Kumar M, Kranthi Kumar S, Kalpana E, Srikanth D, Saikumar K. A novel implementation of Linux based android platform for client and server. *A Fusion of Artificial Intelligence and Internet of Things for Emerging Cyber Systems*. 2021. pp. 151–170.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>71 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

10. Fox R. Linux with operating system concepts. *Chapman and Hall/CRC*. 2021.
11. Raza A, Unger T, Boyd M, Munson EB, Sohal P, Drepper U, Krieger O. Unikernel linux (ukl). *Proceedings of the Eighteenth European Conference on Computer Systems*. 2023. pp. 590–605.
12. Ali M, Bilal K, Khan SU, Veeravalli B, Li K, Zomaya AY. DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security. *IEEE Transactions on Cloud Computing*. 2018. Vol. 6(2, SI). pp. 303–315.
13. Alkhanak EN, Lee SP, Khan SUR. Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. *Future Generation Computer Systems*. 2015. Vol. 50. pp. 3–21.
14. Alteen N, Fisher J, Gerena C, Gruver W, Jalis A, Osman H, Pagan M, Patlolla S, Roth M. AWS Certified Developer Official Study Guide: Associate (DVA-C01) Exam. John Wiley & Sons. 2019.
15. Balaji S, Krishnan MN, Vajha M, Ramkumar V, Sasidharan B, Kumar PV. Erasure coding for distributed storage: an overview. *Science China Information Sciences*. 2018. Vol. 61(10). pp. 1–45.
16. Barhate SM, Dhore M. Hybrid Cloud: A Solution to Cloud Interoperability. *Proceedings of the Second International Conference on Inventive Communication and Computational Technologies (ICICCT 2018)*. 2018. P. 1242–1247.
17. Barhate S, Dhore M. Hybrid Cloud: A Cost Optimised Solution To Cloud Interoperability. *Proceedings of the International Conference on Innovative Trends in Information Technology (ICITIIT 2020)*. 2020. pp. 1–5.
18. Barika M, Garg S, Zomaya AY, Wang L, Moorsel AV, Ranjan R. Orchestrating Big Data Analysis Workflows in the Cloud: Research Challenges, Survey, and Future Directions. *ACM Computing Surveys*. 2019. Vol. 52(5). pp. 14-21.
19. Belgacem A. Dynamic resource allocation in cloud computing: analysis and taxonomies. *Computing*. 2022. Vol. 104(3). pp. 681–710.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>72 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

20. Bessani A, Correia M, Quaresma B, André F, Sousa P. DepSky: Dependable and Secure Storage in a Cloud-of-Clouds. *ACM Transactions on Storage*. 2013. Vol. 9(4). pp. 88-98.

21. Bhamare D, Samaka M, Erbad A, Jain R, Gupta L, Chan HA. Optimal virtual network function placement in multi-cloud service function chaining architecture. *Computer Communications*. 2017. Vol. 102. pp. 1–16.

22. Bhatt R, Datta R. Cost Modelling and Studies with Different Deployment Strategies for Wireless Multimedia Sensor Network Over Flat and Elevated Terrains. *International Journal of Wireless Information Networks*. 2014. Vol. 21. pp. 15–31.

23. Blass A, Gurevich Y. Ordinary interactive small-step algorithms, I. *ACM Transactions on Computational Logic*. 2006. Vol. 7(2). pp. 363–419.

24. Bokhari SMA, Theel O. A flexible hybrid approach to data replication in distributed systems. *Proceedings of the Computing Conference (SAI 2020)*. 2020. Vol. 1228. pp. 196–207.

25. Brewer EA. Towards robust distributed systems. *Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing (PODC 2000)*. 2000. Vol. –. pp. 7.

26. Buckley S. Spread Networks tightens up latency along Chicago to New York route. *FierceTelecom*. 2023. URL: <https://www.fiercetelecom.com/telecom/spread-networks-tightens-up-latency-along-chicago-to-new-york-route>

27. Bulent E, Tarek A. Analysis Of Interoperability In Cloud Computing. *Proceedings of the 2019 5th International Conference on Computer and Technology Applications (ICCTA 2019)*. 2019. pp. 189–192.

28. Campêlo RA, Casanova MA, Guedes DO, Laender AH. A brief survey on replica consistency in cloud environments. *Journal of Internet Services and Applications*. 2020. Vol. 11(1). pp. 1–13.

29. Corodescu A.-A., Nikolov N., Khan A.Q., Soylyu A., Matskin M., Payberah A.H., Roman D. Big Data Workflows: Locality-Aware Orchestration Using Software Containers. *Sensors*. 2021. Vol. 21(24). pp. 8212.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>73 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

30. Daniel E.J., White C.M., Teague K.A. An interarrival delay jitter model using multistructure network delay characteristics for packet networks. *Proceedings of the Thrity-Seventh Asilomar Conference on Signals, Systems & Computers*. 2003. Vol. 2. pp. 1738–1742.

31. Dillon T., Wu C., Chang E. Cloud Computing: Issues and Challenges. *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010)*. 2010. pp. 27–33.

32. Dong X., Zhao L., Zhou X., Li K., Guo D., Qiu T. An Online Cost-Efficient Transmission Scheme for Information-Agnostic Traffic in Inter-Datacenter Networks. *IEEE Transactions on Cloud Computing*. 2019. Vol. 10(1). pp. 202–215.

33. Dowell S., Barreto A., Michael J.B., Shing M.-T. Cloud to cloud interoperability. *Proceedings of the 6th International Conference on System of Systems Engineering (SoSE 2011)*. 2011. pp. 258–263.

34. Edwin E.B., Umamaheswari P., Thanka M.R. An efficient and improved multi-objective optimized replication management with dynamic and cost aware strategies in cloud computing data center. *Cluster Computing*. 2019. Vol. 22(5). pp. 11119–11128.

35. Endo P.T., Rodrigues M., Gonçalves G.E., Kelner J., Sadok D.H., Curescu C. High availability in clouds: systematic review and research challenges. *Journal of Cloud Computing*. 2016. Vol. 5(1). pp. 1–15.

36. Erradi A., Mansouri Y. Online cost optimization algorithms for tiered cloud storage services. *Journal of Systems and Software*. 2020. Vol. 160. pp. 110457.

37. Georgios C., Evangelia F., Christos M., Maria N. Exploring cost-efficient bundling in a multi-cloud environment. *Simulation Modelling Practice and Theory*. 2021. Vol. 111. pp. 102338.

38. Gessert F., Wingerath W., Friedrich S., Ritter N. NoSQL Database Systems: A Survey and Decision Guidance. *Computer Science – Research and Development*. 2017. Vol. 32. pp. 353–365.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>74 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

39. Ghoreishi S.E., Karamshuk D., Friderikos V., Sastry N., Dohler M., Aghvami A.H. A Cost-Driven Approach to Caching-as-a-Service in Cloud-Based 5G Mobile Networks. *IEEE Transactions on Mobile Computing*. 2019. Vol. 19(5). pp. 997–1009.

40. Gill S.S., Buyya R. A Taxonomy and Future Directions for Sustainable Cloud Computing: 360 Degree View. *ACM Computing Surveys*. 2018. Vol. 51(5). pp. 1–33.

41. Girault A., Gössler G., Guerraoui R., Hamza J., Seredinschi D.-A. Monotonic Prefix Consistency in Distributed Systems. *Formal Techniques for Distributed Objects, Components, and Systems*. 2018. Vol. 10854. pp. 41–57.

42. Gómez D., Romero J., López P., Vázquez J., Cappo C., Pinto D., Villalba C. Cloud architecture for electronic health record systems interoperability. *Technology and Health Care*. 2022. Vol. 30(3). pp. 551–564.

43. Gopinaath C., Kiruthika C. A Server Side Encryption for Cloud Storage with Federation Sharing in Hybrid Cloud Environment. *Proceedings of the 2017 International Conference on Technical Advancements in Computers and Communications (ICTACC 2017)*. 2017. pp. 128–131.

44. Hale J.S., Li L., Richardson C.N., Wells G.N. Containers for Portable, Productive, and Performant Scientific Computing. *Computing in Science & Engineering*. 2017. Vol. 19(6). pp. 40–50.

45. Han J., Kim S., Kim T., Han D. Toward scaling hardware security module for emerging cloud services. *Proceedings of the 4th Workshop on System Software for Trusted Execution (SysTEX 2019)*. 2019. Vol. –. pp. 1–6.

46. Hayes S. Analyzing network performance management. *IEEE Communications Magazine*. 1993. Vol. 31(5). pp. 52–58.

47. Höfer C., Karagiannis G. Cloud computing services: taxonomy and comparison. *Journal of Internet Services and Applications*. 2011. Vol. 2. pp. 81–94.

48. Hong J., Dreibholz T., Schenkel J.A., Hu J.A. An overview of multi-cloud computing. *Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA 2019)*. 2019. Vol. 927. pp. 1055–1068.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>75 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

49. Hossain K., Roy S. A Data Compression and Storage Optimization Framework for IoT Sensor Data in Cloud Storage. *Proceedings of the 21st International Conference of Computer and Information Technology (ICCIT 2018)*. 2018. pp. 1–6.

50. Ilieva G., Yankova T., Hadjieva V., Doneva R., Totkov G. Cloud service selection as a fuzzy multi-criteria problem. *TEM Journal*. 2020. Vol. 9(2). pp. 484.

51. Irie R., Murata S., Hsu Y.F., Matsuoka M. A Novel Automated Tiered Storage Architecture for Achieving Both Cost Saving and QoE. *Proceedings of the 8th International Symposium on Cloud and Service Computing (SC2)*. 2018. pp. 32–40.

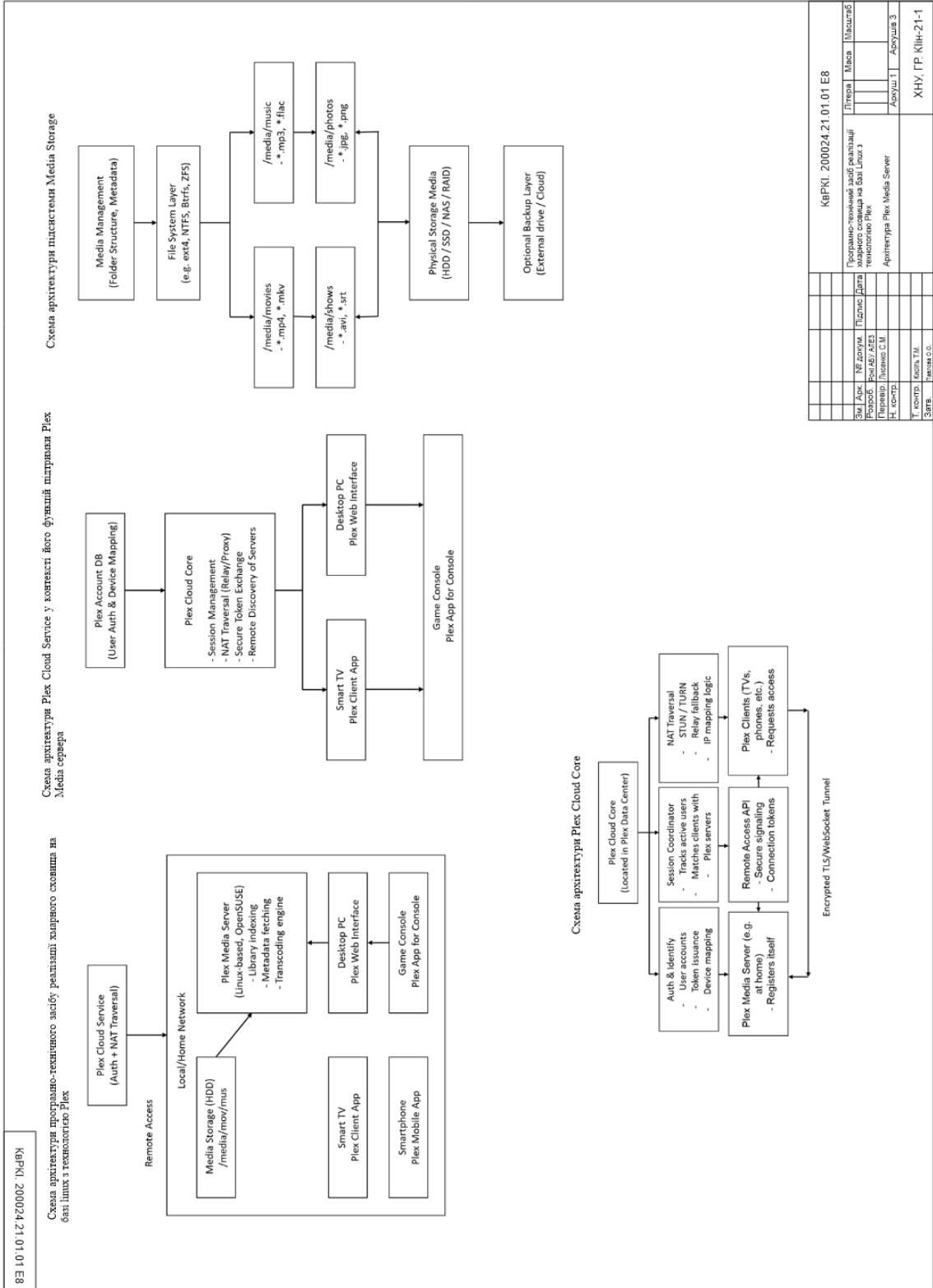
52. Jayakumar S., Prakash S., Akki C. Design of a Novel Architecture for Cost-Effective Cloud-based Content Delivery Network. *International Journal of Advanced Computer Science and Applications*. 2021. Vol. 12(10). pp. 84-98.

53. Jiang F., Cheng Y., Hui Z., Yan R. Modeling and Analyzing for Data Durability Towards Cloud Storage Services. *Proceedings of the 20th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2020)*. 2020. Vol. 12453. pp. 646–661.

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                           | 76   |

# Додаток А (обов'язковий)

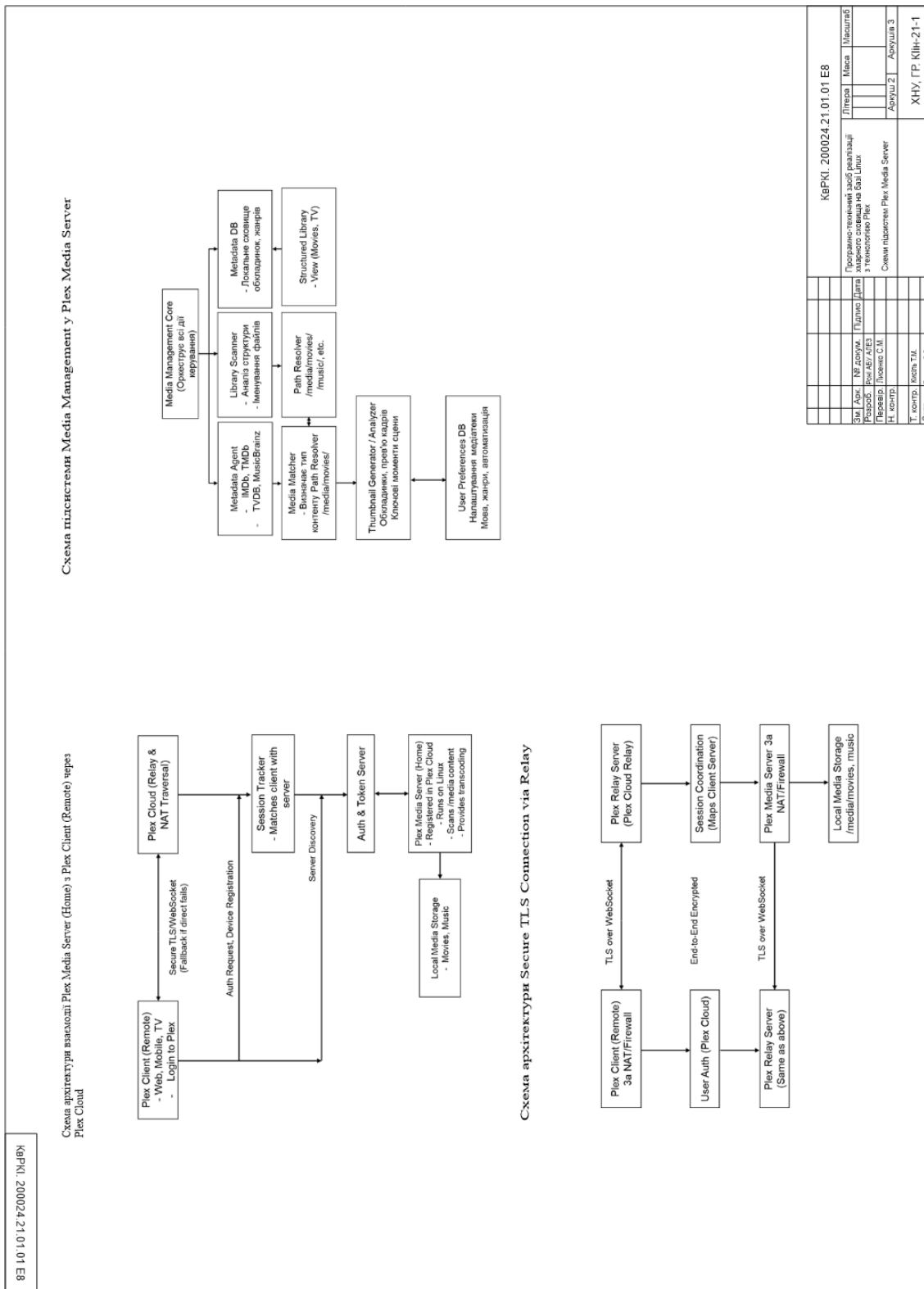
## КОПІЯ КРЕСЛЕННЯ «АРХІТЕКТУРА PLEX MEDIA SERVER»



| КВРКІ: 200024.21.01.01.E8 |                                                                                        |              |            |
|---------------------------|----------------------------------------------------------------------------------------|--------------|------------|
| Літера                    | Масштаб                                                                                |              |            |
| А                         | Програмно-технічний засіб реалізації хмарного сховища на базі Linux з технологією Plex | Підпис       | Дата       |
| Б                         | Архітектура Plex Media Server                                                          | Познач. АТЕЗ | Автомат. 1 |
| В                         |                                                                                        | Ім'я С.М     | Автомат. 3 |
| Г                         |                                                                                        | Ім'я С.М     |            |
| Д                         |                                                                                        | Ім'я С.М     |            |
| Е                         |                                                                                        | Ім'я С.М     |            |
| Ж                         |                                                                                        | Ім'я С.М     |            |
| З                         |                                                                                        | Ім'я С.М     |            |
| И                         |                                                                                        | Ім'я С.М     |            |
| Й                         |                                                                                        | Ім'я С.М     |            |
| К                         |                                                                                        | Ім'я С.М     |            |
| Л                         |                                                                                        | Ім'я С.М     |            |
| М                         |                                                                                        | Ім'я С.М     |            |
| Н                         |                                                                                        | Ім'я С.М     |            |
| О                         |                                                                                        | Ім'я С.М     |            |
| П                         |                                                                                        | Ім'я С.М     |            |
| Р                         |                                                                                        | Ім'я С.М     |            |
| С                         |                                                                                        | Ім'я С.М     |            |
| Т                         |                                                                                        | Ім'я С.М     |            |
| У                         |                                                                                        | Ім'я С.М     |            |
| Ф                         |                                                                                        | Ім'я С.М     |            |
| Х                         |                                                                                        | Ім'я С.М     |            |
| Ц                         |                                                                                        | Ім'я С.М     |            |
| Ч                         |                                                                                        | Ім'я С.М     |            |
| Ш                         |                                                                                        | Ім'я С.М     |            |
| Щ                         |                                                                                        | Ім'я С.М     |            |
| Ю                         |                                                                                        | Ім'я С.М     |            |
| Я                         |                                                                                        | Ім'я С.М     |            |
| ХНУ                       | ГР: КІІІ-21-1                                                                          |              |            |

## Додаток Б (обов'язковий)

# КОПІЯ КРЕСЛЕННЯ «СХЕМИ ПІДСИСТЕМ PLEX MEDIA SERVER»

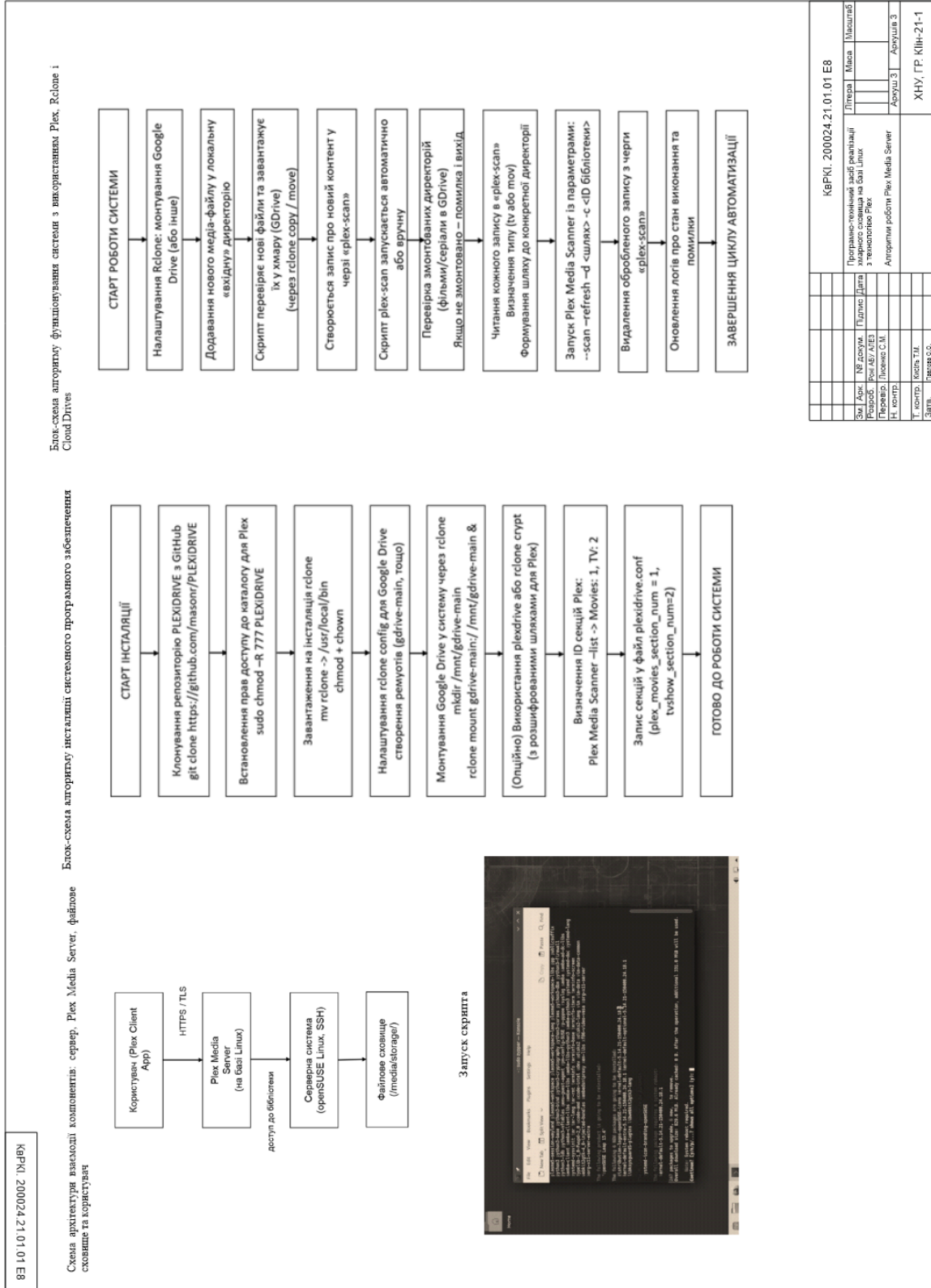


КВРКІ. 200024.21.01.01.E8

|                                                                                            |               |             |               |
|--------------------------------------------------------------------------------------------|---------------|-------------|---------------|
| КВРКІ. 200024.21.01.01.E8                                                                  |               |             |               |
| Зм. Авт.                                                                                   | № докум.      | Підпис/Дата | Підпис/Дата   |
| Розроб.                                                                                    | Розр. АВ/А/ЕЗ | Розроб.     | Розр. АВ/А/ЕЗ |
| Проектув.                                                                                  | Людмила С.М.  | Проєкт.     | Людмила С.М.  |
| Н. контр.                                                                                  | Юрій Т.М.     | Н. контр.   | Людмила С.М.  |
| Т. контр.                                                                                  | Юрій Т.М.     | Т. контр.   | Людмила С.М.  |
| Самп.                                                                                      | Юрій Т.М.     | Самп.       | Людмила С.М.  |
| Категорія: Програмне рішення для реалізації функцій Plex Media Server з використанням Plex |               |             |               |
| Специфікація: Plex Media Server                                                            |               |             |               |
| ХНУ, ГР. КІІВ-21-1                                                                         |               |             |               |

## Додаток В (обов'язковий)

# КОПІЯ КРЕСЛЕННЯ «АЛГОРИТМИ РОБОТИ PLEX MEDIA SERVER»



# Anti-Plagiarism (UA) v-15.281 Educational

**The maximum coincidence with one document 31.0%**

Dictionary check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 16%

|                                                                                                                                                                                                                               |          |         |                           |           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------|---------------------------|-----------|
| ID: 243347<br>Title: БКР Програмно-технічний засіб реалізації хмарного сховища на базі Linux з технологією Plex<br>Added in a DB: 2025-06-04<br>Authors: Роні АБУ АЛЕЗ<br>Heads: Сергій ЛИСЕНКО<br>Consultants:<br>Opponents: | Document |         | Sum coincidence on the DB |           |
|                                                                                                                                                                                                                               | Symbols  | Lexemes | Symbols                   | Lexemes   |
|                                                                                                                                                                                                                               | 98147    | 771     | 36102 (37%)               | 276 (36%) |

## Plagiarism sources

| ID     | Description                                                                                                                                                                                                         | Plagiarism presence in the document |             |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|-------------|
|        |                                                                                                                                                                                                                     | Symbols                             | Lexemes     |
| 240794 | Title: Звіт з ПДП Програмно-технічний засіб реалізації хмарного сховища на базі Linux з технологією Plex<br>Added in a DB: 2025-05-04<br>Authors: Абу Алез Роні<br>Heads: Павлова О.О<br>Consultants:<br>Opponents: | 30248 (31.0%)                       | 204 (26.0%) |

## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Роні АБУ АЛЕЗ

Співавтор:

Назва: Роні\_Програмно-технічний засіб реалізації хмарного сховища на базі Linux з технологією Plex

Експерт:

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1:9.8%

Коефіцієнт подібності 2:3.8%

Мікропробіли: 6

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-06-04 17:30:59.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВ● (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2025-06-04

Дата



Доцент Андрій Нічепорук

експерт

## РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Роні АБУ АЛЕЗ

Тема: Програмно-технічний засіб реалізації хмарного сховища на базі Linux з технологією Plex

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   66  

1. Короткий зміст роботи та прийнятих рішень: Метою кваліфікаційної роботи є розроблення та реалізація програмно-технічного засобу хмарного сховища на базі Linux з використанням технології Plex.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено детальний аналіз програмно-технічних засобів, які дозволяють ефективно реалізувати хмарне сховище. Встановлено, що використання хмарних середовищ є сучасною та економічно доцільною практикою, яка забезпечує гнучкий доступ до даних, масштабованість, підвищений рівень безпеки та зниження витрат на інфраструктуру.

У другому розділі зосереджено увагу на проектуванні системи. Проведено порівняльний аналіз існуючих рішень хмарного зберігання, таких як Nextcloud, Seafiler, Syncthing та Plex, що дозволило обґрунтувати вибір Plex як оптимального рішення для мультимедійного середовища.

У третьому розділі здійснено програмно-апаратну реалізацію хмарного сховища. Розроблено архітектуру взаємодії серверної частини з Plex, файловим сховищем та користувачем.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага налаштувань терміналу Linux.

6. Оцінка графічного оформлення та пояснювальної записки роботи:  
Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

8. Інші зауваження: \_\_\_\_\_

9. Оцінка дипломної роботи: добре

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

*Бєрєзюк А. П., д. фб-м. н., проф.,  
завзубор каф. ІПЗ*

“ ” \_\_\_\_\_ 2025 р.

 (підпис)

Завідувачу кафедри КІС  
д-р. філософії, доц. Ользі ПАВЛОВІЙ

Роні АБУ АЛЕЗ

ІІБ здобувачів вищої освіти

ФІТ, 4 курсу, групи КІін-21-1

### ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності у Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений(а). Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений(а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Strike-Plagiarism та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

5 червня 2025 року



**РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ**  
**КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ**  
**ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Програмно-технічний засіб реалізації хмарного сховища на базі Linux з технологією Plex

Автор: Роні АБУ АЛЕЗ

Спеціальність: 123- Комп'ютерна інженерія

Освітня програма: освітньо-професійна

Науковий керівник: Сергій ЛИСЕНКО, д.т.н, професор

Після аналізу звіту подібності зроблено такий висновок:

| № | Висновок                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Позначка про відповідність |
|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
| 1 | Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота <u>приймається до захисту.</u>                                                                                                                                                                                                                                                                                                                                                        | відповідає                 |
| 2 | Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання <u>письмової роботи та ідентичності друкованої та електронної версії роботи</u> |                            |
| 3 | Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і <u>успішно пройде повторну перевірку на академічний плагіат.</u>        |                            |
| 4 | Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить <u>фабрикацію або фальсифікацію даних.</u> Робота не допускається до захисту.                                                                                                                                                                                                                                  |                            |

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

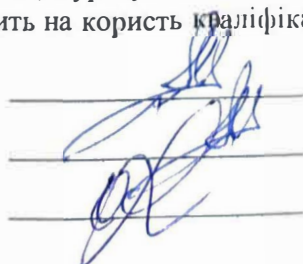
- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 4) збіги зі Звітом з ПДП Програмно-технічний засіб реалізації хмарного сховища на базі Linux з технологією Plex.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості StrikePlagiarism, складає 9.8% і адресується до 401 першоджерела; та системою Anti-Plagiarism складає 31.0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



Сергій ЛИСЕНКО

Сергій ЛИСЕНКО

Ольга ПАВЛОВА

QUALIFICATION WORK

bachelor  
Educational level

Linux- based cloud storage with Plex technology  
Topic name

КБPKI 200024.21.01.01 ПЗ  
Code

Area of knowledge 12 "Information technologies"  
Code, name

Specialty 123 "Computer Engineering"  
Code, name

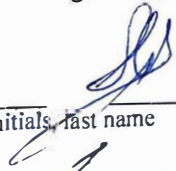
Educational program "Computer Engineering and Programming"  
Name

4th year student, group KIin-21-1

  
Signature Initials, last name

Roni ABU ALEZ

Supervisor

  
Signature, date Initials, last name

Sergii LYSENKO

Norm controller

  
Signature, date Initials, last name

Tetyana KYSI L

I allow for protection:

Head of the Department of Computer Science  
engineering and information

systems

  
Signature Initials, last name

Olga PAVLOVA

"05" June 2025

Khmelnytskyi 2025

# KHMELNYTSKY NATIONAL UNIVERSITY

Faculty of INFORMATION TECHNOLOGIES

Department of COMPUTER ENGINEERING AND INFORMATION SYSTEMS

Educational level BACHELOR

Area of Knowledge 12 INFORMATION TECHNOLOGY

Specialty 123 COMPUTER ENGINEERING

Educational program "COMPUTER ENGINEERING AND PROGRAMMING"

APPROVE

Head of the Department Olha PAVLOVA

"10" 01 2025



## TASK FOR BACHELOR'S QUALIFICATION WORK

Rony ABU ALEZ

Student's last name, first name, patronymic

1. Project topic (work) Software and hardware tool for implementing cloud storage based on Linux with Plex technology

Project Manager (Works) Sergii LYSENKO, Doctor of Technical Sciences, Professor

Surname, first name, patronymic, academic degree, academic title

Approved by order of the rector of the university dated 07.02.2025 No. 5

2. The deadline for submitting a project (work) to the department by the student is 01.06.2025.

3. Initial data for the project (work) Tasks for the qualification work

4. Contents of the explanatory note (list of issues to be developed) \_\_\_\_\_

Analysis of software and hardware

Design of a software and hardware tool for implementing cloud storage based on Linux with Plex technology

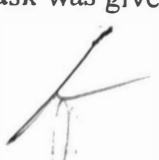



Linux-based cloud storage implementations with Plex technology

5. List of graphic material (indicating mandatory drawings) \_\_\_\_\_ Plex subsystem diagrams Media Server

Plex Architecture Media Server

Plex working algorithms Media Server

6. Consultants of diploma project sections (theses)

| Section         | Consultant's last name, initials and position        | Signature, date                                                                     |                                                                                     |
|-----------------|------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|                 |                                                      | the task was given                                                                  | accepted the task                                                                   |
| Norm control    | Tetyana KYSIL, Associate Professor of the Department |  |  |
| Anti-plagiarism | Andriy NICHEPORUK, Associate Professor of the        |  |  |

7. Date of issue of the task " 10 " 01 20 25.

CALENDAR PLAN

| No. of employees/employees | Name of stages (sections) diploma project (work)                                                                                               | Deadline project stages (work) | Note |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|------|
| 1                          | Choosing a research direction and agreeing on the topic of the qualification work with the supervisor                                          | 10.01.2025                     | done |
| 2                          | Introduction to the subject area; formulation of the goal and objectives of the research; definition of the object and subject of the research | 01.02.2025                     | done |
| 3                          | Work on section 1 – researching the subject area and setting the problem                                                                       | 01.03.2025                     | done |
| 4                          | Work on section 2 – selection of components for designing a system for adaptive application of monitoring elements of a reconnaissance UAV     | 01.04.2025                     | done |
| 5                          | Work on section 3 – designing a system for adaptive application of monitoring elements of a reconnaissance UAV                                 | 04/29/2025                     | done |
| 6                          | Preparation of an explanatory note according to the requirements                                                                               | 05/25/2025                     | done |
| 7                          | Preliminary protection of the VKR                                                                                                              | 05/26/2025                     | done |
| 8                          | Defense of the VRK at the EC meeting                                                                                                           | June 2025                      |      |

Student

Supervisor

Signature

Signature



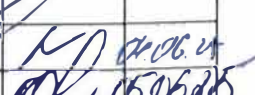
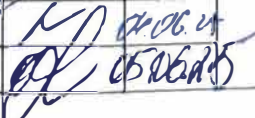
Rony ABU ALEZ

Initials, last name

Serhiy LYSENKO

Initials, last name

| Line number | format | Marking                  | Name                                    | Number | No | Note |
|-------------|--------|--------------------------|-----------------------------------------|--------|----|------|
|             |        |                          | <u>Text documents</u>                   |        |    |      |
| 1           |        | КВРКІ 200024.21.01.01 ПЗ | Explanatory note                        | 66     |    |      |
|             |        |                          | <u>Graphic materials</u>                |        |    |      |
| 2           |        | КВРКІ 200024.21.01.01 E8 | Plex subsystem diagrams<br>Media Server | 1      |    |      |
| 3           |        | КВРКІ 200024.21.01.01 E8 | Plex Architecture Media<br>Server and   | 1      |    |      |
| 4           |        | КВРКІ 200024.21.01.01 E8 | Plex working algorithms<br>Media Server | 1      |    |      |

|                          |     |              |                                                                                     |          |        |
|--------------------------|-----|--------------|-------------------------------------------------------------------------------------|----------|--------|
| КВРКІ 200024.21.01.01 ПЗ |     |              |                                                                                     |          |        |
| Change                   | Ark | Document No. | Signature                                                                           | Date     |        |
| Developed                |     | Abu Alez     |  |          |        |
| Check                    |     | Lysenko      |  |          |        |
| N. counter.              |     | Kysil        |  | 02.06.24 |        |
| Approved.                |     | Pavlova      |  | 05.06.25 |        |
| Project announcement     |     |              | Letter                                                                              | Sheet    | Sheets |
|                          |     |              | IN                                                                                  | 1        | 1      |
| KhNU, Klin-21-1          |     |              |                                                                                     |          |        |

## ANNOTATION

Topic of the qualification work: " Linux- based cloud storage with Plex technology."

Author of the work: Roni ABU ALEZ.

Supervisor: Sergii LYSENKO

Explanatory note: 66 p., 12 fig., 3 appendices, 5 4 sources.

Graphic part: 3 drawings.

SOFTWARE, COMPUTER SYSTEM, CLOUD STORAGE, LINUX, PLEX TECHNOLOGY.

The purpose of the thesis is to design and implement software and hardware tools for implementing cloud storage based on Linux with Plex technology.

The object of research is software and hardware for implementing cloud storage.

The subject of the study is a software and hardware tool for implementing cloud storage based on Linux with Plex technology.

During this study, a systematic literature review method was used to study and analyze the subject area of this study from textual sources of information.



05/30/2025

Student Signature Date

# CONTENT

|                                                                                                                                |           |
|--------------------------------------------------------------------------------------------------------------------------------|-----------|
| INTRODUCTION .....                                                                                                             | 4         |
| <b>1 ANALYSIS OF SOFTWARE AND HARDWARE</b> .....                                                                               | <b>5</b>  |
| 1.1 The relevance of cloud 5 .....                                                                                             | 6         |
| 1.2 Advantages of using cloud environments .....                                                                               | 6         |
| 1.3 The relevance of Plex .....                                                                                                | 7         |
| 1.4 Possibilities of using the operating system to build a Linux-based cloud storage with Plex .....                           | 8         |
| 1.5 Deploying automatic content cataloging with metadata .....                                                                 | 11        |
| 1.6 Implementing real-time streaming support .....                                                                             | 12        |
| 1.7 Implementing remote access support with secure connection capability ....                                                  | 13        |
| 1.8 Implementing video optimization for client devices .....                                                                   | 14        |
| 1.9 Using docker to manage a plex .....                                                                                        | 17        |
| 1.10 Conclusions to the first chapter .....                                                                                    | 19        |
| <b>2 DESIGN OF A SOFTWARE AND HARDWARE TOOL FOR IMPLEMENTING CLOUD STORAGE ON THE BASE OF LINUX WITH PLEX TECHNOLOGY</b> ..... | <b>20</b> |
| 2.1 Overview of the selected cloud storage technology .....                                                                    | 20        |
| 2.2 Choosing a cloud type .....                                                                                                | 22        |
| 2.3 Developing the Plex Media Server Architecture .....                                                                        | 24        |
| 2.4 Selecting a data exchange protocol .....                                                                                   | 36        |
| 2.4.1 DLNA .....                                                                                                               | 37        |
| 2.4.2 HTTPS .....                                                                                                              | 37        |
| 2.4.3 Plex's own APIs .....                                                                                                    | 38        |
| 2.5 Choosing an operating system .....                                                                                         | 39        |
| 2.6 Overview of the applied cloud storage technology .....                                                                     | 40        |
| 2.10 Conclusions to the second chapter .....                                                                                   | 42        |

КвРКІ. 200024.21.01.01 ПЗ

| Зм.      | Арк. | № док.ум.      | Підпис             | Дата |                                                          | Літера | Аркул | Аркулів |
|----------|------|----------------|--------------------|------|----------------------------------------------------------|--------|-------|---------|
| Виконав  |      | Абу Алез Роні  | <i>[Signature]</i> |      | Linux- based cloud storage with Plex technology.<br>Note | у      | 2     | 00      |
| Черевін. |      | Sergii LYSENKO | <i>[Signature]</i> |      |                                                          |        |       |         |
| Н.КОНТД. |      | Тетяна KYCİL   | <i>[Signature]</i> |      |                                                          |        |       |         |
| Ватвер.  |      | Olha PAVLOVA   | <i>[Signature]</i> |      |                                                          |        |       |         |

KhNU, Klu-21-1

|                                                                                                                                  |    |
|----------------------------------------------------------------------------------------------------------------------------------|----|
| <b>3 SOFTWARE AND HARDWARE IMPLEMENTATION OF LINUX-BASED CLOUD STORAGE WITH PLEX TECHNOLOGY</b> .....                            | 43 |
| 3.1 Architecture of the software and hardware tool. Diagram of interaction of components: server, Plex, file storage, user ..... | 43 |
| 3.2 System Software Development .....                                                                                            | 46 |
| 3.3 Algorithm of system operation using Plex, Rclone and Cloud Drives .....                                                      | 47 |
| 3.4 Installing system software .....                                                                                             | 49 |
| 3.5 Application of the developed software and hardware implementation of cloud storage based on Linux with Plex .....            | 53 |
| 3.6 Configuring a single-storage configuration .....                                                                             | 55 |
| 3.7 Configuring a multi-cloud storage configuration .....                                                                        | 57 |
| 3.8 Automate the use of one or more Google Drive accounts as storage for a Plex server .....                                     | 59 |
| 3.9 Conclusions to the third chapter .....                                                                                       | 62 |
| <b>CONCLUSIONS</b> .....                                                                                                         | 63 |
| <b>LIST OF REFERENCES</b> .....                                                                                                  | 64 |
| <b>APPENDIX A</b> .....                                                                                                          | 70 |
| <b>APPENDIX B</b> .....                                                                                                          | 71 |
| <b>APPENDIX B</b> .....                                                                                                          | 72 |

## INTRODUCTION

Plex technology is due to the rapid growth of the need for effective management, organization, and streaming of multimedia content in the context of digital transformation.

Modern users want to have convenient access to their media libraries from different devices, including remote access. Plex provides a high level of content organization, automatic cataloging, support for various media formats, and extensive personalization capabilities, making it an effective tool for building a universal multimedia environment.

Additionally, the rise of home servers and the trend toward decentralized storage are fueling interest in solutions like Plex, which allow you to maintain control over your information while providing flexibility and scalability.

All this makes it advisable to study the possibilities and advantages of using this technology in the context of modern IT systems.

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                           | 4    |

# 1 ANALYSIS OF SOFTWARE AND HARDWARE

## 1.1 Relevance application cloudy environments

In today's rapidly developing digital technologies and growing data volumes, efficient, scalable, and secure means of storing and processing information are becoming increasingly important. One of the key solutions in this direction is cloud storage, which has become an integral part of the information infrastructure in business, education, science, and government structures.

Cloud technologies provide flexible access to data at any time and from any device, which is especially important in the context of remote work, user mobility and distributed teams. Due to this, storing data in the cloud is not only convenient, but also a cost-effective solution compared to traditional local servers or physical media [1].

In addition, cloud services are actively implementing mechanisms for backup, scalability, automatic updates, and integration with other services, which allows you to optimize business processes and avoid data loss in the event of failures, cyberattacks, or technical malfunctions. This makes cloud storage attractive not only for large corporations, but also for small businesses, startups, educational institutions, and individual users [2].

The use of cloud storage is particularly relevant in the context of ensuring information security. Modern cloud platforms offer comprehensive solutions for authentication, encryption, access control, and activity monitoring, which allows you to effectively protect confidential data and meet modern cybersecurity standards [3].

The environmental aspect should also not be neglected. Centralized cloud data storage contributes to the energy efficiency of IT infrastructures, reducing the need for a large number of physical servers in enterprises and reducing the carbon footprint [4].

The use of cloud storage is extremely relevant in the modern information environment. It opens up new opportunities for effective data management, ensuring their security, scaling information systems and increasing the overall digital readiness of

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
|     |      |          |        |      |                           | 5    |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |      |

organizations. In this regard, the study of cloud technologies and the development of recommendations for their effective use is an important and timely task.

## 1.2 Advantages of using cloud environments

The advantages of using cloud environments compared to local data storage are a fundamental change in the approach to storing, processing and managing information. First of all, cloud technologies provide significantly higher flexibility and availability. If local storage involves physical binding to a specific device or an internal network of the organization, then the cloud environment allows you to access data from anywhere in the world where there is an Internet connection. This is especially important for mobile users, remote teams, as well as in cases where it is necessary to quickly share information with a large number of users or devices [5-9].

Another key advantage is scalability. In the case of local storage, increasing the amount of data often requires additional hardware costs, infrastructure upgrades, or even complete replacement of equipment. In contrast, cloud services allow you to quickly adapt the storage volume to current needs, virtually “on the fly”, without downtime and without the need for technical maintenance from the user.

The automation of processes offered by cloud platforms also plays a significant role. For example, backups, synchronization, software updates, and access rights management can be performed automatically, without active user intervention. This reduces the likelihood of human error, minimizes the risks of data loss, and allows you to focus on your core business, rather than on maintaining your IT infrastructure [10-12].

Security is also a major factor in favor of the cloud. While there is a misconception that data in the cloud is at greater risk, cloud service providers actually implement advanced security mechanisms: data encryption, multi-factor authentication, role-based access control, transaction auditing, and physical protection of data centers. For most users, this provides a higher level of security than what they can implement on-premises, especially given limited resources.

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
|     |      |          |        |      |                           | 6    |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |      |

In terms of cost, cloud solutions usually prove to be more beneficial, especially for small and medium-sized enterprises, educational institutions, or individual users.

Instead of significant initial investments in servers, hard drives, uninterruptible power supplies, and cooling systems, cloud services are paid for on a subscription or usage model, which allows for better cost predictability and reduced financial burden.

Additionally, cloud environments facilitate more efficient collaboration between users. Data stored in the cloud can be edited collaboratively in real time, and all changes are saved automatically [13].

This significantly increases the efficiency of teamwork, compared to local files that require constant copying, forwarding, or manual synchronization.

Cloud technologies offer a fundamentally new level of convenience, efficiency, and reliability in information management, making them a logical evolution of data storage systems in the digital age.

Compared to traditional local storage, cloud environments provide a higher level of adaptability to the demands of the modern world.

### 1.3 The relevance of Plex technology

In today's digitalization of enterprises and the active implementation of the Industry 4.0 concept, there is a growing need for flexible, scalable and integrated information systems that allow for effective management of production processes, logistics, supply and analytics in real time. One of such systems is Plex Manufacturing Cloud (or simply Plex ) is a cloud-based ERP/MES platform that provides a comprehensive solution for managing manufacturing enterprises [14].

Let's consider the main advantages of using technology.

Plex technology is to increase production efficiency. Plex provides full transparency of the production process, allows you to quickly respond to changes in the technological cycle, minimizes equipment downtime and resource loss. Thanks to built-

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
|     |      |          |        |      |                           | 7    |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |      |

in monitoring tools, the platform supports data -driven decision-making decision making ), which is critically important in today's competitive environment.

Plex technology is the possibility of integration within the framework of digital transformations. Plex is one of the platforms that organically fits into the paradigm of a digital enterprise. It integrates with IoT devices, analytical systems and other business solutions, forming a single digital ecosystem. This ensures the relevance of the research for the tasks of digital transformation of production processes [15-18].

An important opportunity of technology is building a cloud architecture. The transition from local systems to cloud solutions is a global trend. Plex, as a cloud platform, provides access to data 24/7 from anywhere in the world, high scalability and updates without user intervention. This allows enterprises to reduce costs for IT infrastructure, increase flexibility and efficiency in management.

Unlike complex ERP systems, Plex is more accessible to medium-sized businesses, providing fast deployment and ease of use. This makes it relevant for research in the context of implementing innovations at the level of regional manufacturing companies [19-21].

Plex supports real-time work with production data, allowing you to implement Smart concepts Manufacturing and Lean Production. Exploring the capabilities of Plex technology in this context allows us to explore the potential of implementing such approaches in practice.

Using Plex in your bachelor's thesis allows you to develop competencies that are in demand in the labor market, particularly in the areas of digital manufacturing, data management, cloud technologies, and analytics. This increases the practical relevance of the chosen topic and its innovativeness.

#### 1.4 Linux- based cloud storage with Plex technology

In today's conditions of active use of multimedia content, there is a need for effective, secure and affordable solutions for storing and broadcasting media data. One

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
|     |      |          |        |      |                           | 8    |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |      |

such solution is to build your own cloud media server based on the Linux operating system using Plex technology. Media Server. This solution allows you to create a personalized, scalable, and controlled platform for storing, organizing, and broadcasting digital content [22-25].

Linux operating system is one of the most stable, secure, and customizable platforms for server use. Its open source code allows you to: n Take full control of your infrastructure - from file system configuration to security and access systems, optimize resources - Linux can run even on low-power hardware, making it ideal for home NAS solutions, and automate maintenance - thanks to system scripts, cron jobs, and other tools.

Plex Media Server is a software platform for broadcasting media content (video, audio, photos) over a local network or the Internet.

The platform provides:

- automatic cataloging of content with metadata ( posters, actors, description);
- support for real-time streaming to various devices (TVs, smartphones, tablets);
- multiplatform - available on linux, windows, macos, and also has mobile and web applications;
- support for remote access with the possibility of a secure connection from anywhere in the world;
- video optimization for client devices ( real-time transcoding );
- integration with plex pass - for additional functions: dvr, synchronization, live TV, etc.

On Linux, you can deploy Plex as part of larger solutions:

- using docker - for isolated and convenient management of the plex container ;
- together with samba / nfs - for organizing network access to files;
- via reverse proxy (e.g. nginx ) - for secure access via https ;
- monitoring using tools such as grafana and prometheus - to track performance;
- data protection using firewalls, fail2ban and ssh.

Linux- based cloud storage allows you to:

- use existing hardware (e.g. old PC, raspberry pi or nas );

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
|     |      |          |        |      |                           | 9    |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |      |

- refuse third-party paid services ( google drive, icloud, netflix ), while maintaining privacy ;
- reduce power and cooling costs through optimization under Linux.

Using Linux as a platform for building cloud storage with Plex technology combines the reliability, scalability, and security of an open operating system with powerful functionality for organizing and distributing multimedia content. This solution is relevant for personal use, small offices, as well as within educational and research projects [26-29].

Plex technology Media Server has wide practical significance in the context of creating a personal cloud media storage, especially when used on the basis of the Linux operating system [30-34]. Plex allows you to organize the storage, structuring and broadcasting of multimedia content, including video, audio and photo files, providing convenient access to it from anywhere in the world. The main functionality of this technology is to transform the local server environment into a full-fledged media center, capable of broadcasting content in real time to various client devices - from TVs and computers to mobile applications and web browsers [35-39].

The user can place their content in defined directories, after which Plex automatically indexes it, adds metadata such as covers, descriptions, actors, genres, and creates a single media library with an attractive interface. One of the key features is support for streaming with automatic transcoding ( transcoding ) - this allows you to adapt video or audio to a format optimal for a specific device or connection speed, without requiring the user to make changes to the original file [40-43].

In addition to basic functionality, Plex provides the ability to create accounts for multiple users, restrict their access rights, and organize content sharing in a secure environment. Additionally, the technology supports remote access, which means that a server deployed on a home network can be used as a personal alternative to commercial streaming platforms. Plex also has integration with services such as TIDAL or access to live TV, which extends its functionality beyond local content [44-47].

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>10 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Using Plex on a Linux server opens up additional possibilities for automation, backup, system status monitoring, and resource optimization. Thanks to the flexibility of Linux, you can adapt the server to any technical conditions - from a low-power minicomputer Raspberry Pi to a full-fledged server environment. This makes Plex a versatile tool for both home users and educational or research purposes, including as part of implementing private cloud media storage solutions.

### 1.5 Deploying automatic content cataloging with metadata

Deploying automatic content cataloging with metadata is one of the key benefits of using Plex Media Server in building a modern cloud storage. After adding media files to the appropriate directories on the server, Plex automatically scans the content, analyzing the type of files, their names and structure. Based on the data obtained, the system turns to open sources of metadata, such as The Movie Database ( TMDb ), TheTVDB, MusicBrainz, and others, and downloads information associated with each media file [48].

This process includes retrieving movie or TV series titles, descriptions, posters, background images, cast information, ratings, release date, duration, genre, etc. For music files, the system extracts information about the artist, album, genre, year of release, album cover, and even lyrics. All of this allows you to create an intuitive, visually appealing, and conveniently structured library that provides a comfortable navigational environment for the user.

Thanks to this cataloging, content becomes not just a set of files, but a full-fledged digital archive that can be filtered, grouped or searched for by any characteristics. This functionality also provides a high level of personalization: Plex can automatically create thematic collections, recommend content based on the user's viewing history or preferences. At the same time, all updates to the catalog are performed automatically after adding new files or changing the directory structure, which allows reducing user participation to a minimum [49].

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>11 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Linux- based cloud storage, automatic cataloging provides integration with file system and indexing services, and also allows the administrator to centrally manage metadata and ensure its integrity and relevance. Thus, Plex transforms a raw array of digital content into a well-organized, logical and attractive multimedia database that meets the requirements of modern information storage and processing systems.

## 1.6 Implementing real-time streaming support

Implementing support for real-time streaming to multiple devices is one of Plex's defining features. Media Server, which provides universal access to multimedia content regardless of the type of client device. With a built-in streaming engine, Plex allows you to broadcast video, audio and photos in real time to a wide range of devices, including smart TVs, PCs, laptops, smartphones, tablets, media players (e.g. Roku, Apple TV, Chromecast ) and even game consoles.

A key element of this functionality is content transcoding, i.e. its dynamic recoding to the technical specifications of the device receiving the stream. For example, if the original video file has a format or resolution that is not supported by a particular device, Plex automatically adapts this file "on the fly" - changing the codec, bitrate, resolution or container to ensure compatibility and stable transmission. This is especially important in conditions of limited network bandwidth or when using mobile Internet.

The Plex server is responsible for processing requests, and the client applications are responsible for displaying the user interface and playing content. With a high-quality network connection, playback starts almost instantly, and in unstable connection conditions, Plex automatically optimizes the stream according to changes in speed. In addition, the system supports adaptive streaming ( adaptive bitrate streaming ), which allows you to change the video quality without stopping playback [50].

All of these features are implemented without the need for complex configuration - Plex automatically determines the optimal stream parameters based on the client, network type, and server characteristics. If necessary, the user or administrator can

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>12 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

manually specify quality priorities, maximum bitrate, or playback method (direct playback, transcoding, optimized copies).

Plex provides a flexible, efficient, and scalable streaming solution that transforms a server from local storage into a true multi-platform cloud media center with instant access to content from anywhere in the world.

### 1.7 Implementing remote access support with secure connection capability

Implementing remote access support with the ability to secure connections is another important feature of Plex. Media Server, which allows users to securely access their media content from anywhere with an Internet connection. After configuring remote access, the user can connect to their server not only on the local network, but also from outside - via a web interface, mobile applications or other compatible devices.

For security, Plex uses HTTPS (SSL/TLS) encryption to ensure the confidentiality of transmitted data, protection against traffic interception and malicious interference. All traffic between the client and the server is encrypted, including metadata and streaming content. Plex automatically manages certificate issuance and domain name maintenance through its own request routing system, which simplifies the setup procedure even for inexperienced users.

Plex account, which allows you to authenticate the user and centrally store settings, a list of available servers, and personal preferences. If you use multiple devices or share your media library with family or friends, the server administrator can create additional accounts or grant limited access to parts of the library, while maintaining full control over access rights.

Another important advantage is the support for automatic port forwarding (NAT-PMP / UPnP ), which allows Plex to independently configure the router to provide external access without manual intervention. In cases where automatic configuration is not possible (due to network security policies or hardware), the server supports manual configuration with detailed instructions.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>13 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Implementing remote access in Plex Media Server not only provides the convenience of using multimedia storage outside the local network, but also guarantees a high level of security, stability, and manageability of connections, which is especially important in the context of personal or corporate use of cloud multimedia services.

### 1.8 Implementation optimization video under client devices

Implementing video optimization for client devices, including real-time transcoding, is one of Plex's core capabilities. Media Server. This feature allows you to dynamically adapt video to the technical characteristics of a specific device receiving the stream.

Transcoding is the process of converting video files from one format or quality to another, taking into account the requirements of a specific device, network speed, or user settings.

When a user launches a video, Plex automatically determines whether the video format is compatible with the device requesting playback. If the device does not support that format or resolution, Plex performs transcoding on the fly. This may include:

1. Changing the file format, for example, if the video is in MKV format and the device only supports MP4, Plex will convert the video to this format.
2. Changing the bitrate in a situation where the user's network has limited bandwidth, Plex lowers the bitrate to avoid buffering and ensure smooth playback.
3. Changing the resolution, for example, if the video is in 4K format but the device only supports 1080p or less, Plex lowers the resolution, allowing you to maintain optimal quality with limited resources.

Plex 's real-time transcoding process is automatic and transparent to the user. The Plex web server acts as an intermediary between the media file and the client device, performing the transcoding and delivering the video over the network. This provides flexibility and compatibility with a variety of device types, from smartphones and tablets to TVs, computers, and other media players.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>14 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Plex interface also allows users to customize streaming and transcoding settings to suit their needs. For example, users can specify a maximum bitrate for a mobile device or select video quality for limited internet connections. This allows for a balance between playback quality and data transfer speed.

The key advantage of this approach is that Plex supports on-the-fly transcoding without the need for pre-preparation of media files, which saves a lot of time and eliminates the need to store multiple versions of the same video for different devices.

transcoding is an essential component that allows Plex to provide high compatibility and usability of multimedia content, automatically adapting video to the characteristics of each device and ensuring optimal playback under all conditions.

Implementing integration with Plex Pass opens up additional features for Plex users Media Servers that provide access to advanced features and a higher level of personalization. Plex Pass is a paid subscription that provides a range of tools and features that enhance your multimedia experience and expand your playback and management capabilities.

One of the main advantages of Plex Pass is unlimited and faster transcoding. Non-subscribers may have limits on the number of simultaneous transcoding sessions, as well as restrictions on bitrate when streaming content.

Plex Pass allows you to remove these restrictions, ensuring higher quality and stability of the broadcast on all devices.

This is especially important for users who plan to use Plex as a cloud service for storing and viewing a large amount of content from different devices.

Plex Pass also includes intelligent content organization features, such as automatic creation of metadata, movies, TV shows, and music libraries.

For example, it includes automatic scanning and categorization of media files, as well as the ability to create covers and banners for movies and albums. These features greatly improve the user experience, ensuring a convenient and enjoyable display of content in Plex.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>15 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Plex integration Pass allows users to download media for offline viewing on mobile devices. This makes it possible to watch videos and listen to music without an internet connection, which is especially useful when traveling or in environments with limited network access.

Plex Pass users get access to experimental features that are not yet available to the general public. This includes new UI features, improved transcoding, and integration with new types of devices and services. These features allow users to test the latest innovations in Plex.

Plex Pass also provides additional features for secure access, including the ability to set up two-factor authentication (2FA) to protect accounts. This is an extra layer of security that ensures only authorized users have access. It also provides priority customer support for Plex users. Pass, which allows you to get faster help and solutions for any technical problems.

Plex Pass provides the ability to connect to music streaming services such as TIDAL, allowing users to create their own music libraries and include both locally stored music and music from online services in their content.

Support for podcasts and live streaming is also coming, giving even more options for users who use Plex to listen to and organize audio content.

One of the most innovative features for Plex users Pass is the ability to access live broadcasts via Live TV and use the Plex server as a digital video recorder (DVR) to record television programs and series.

This allows users to create a personalized TV library stored on the server, as well as watch their favorite shows on demand.

Plex Pass also enables integration with other cloud content storage platforms, such as Google Drive, Dropbox or OneDrive, allowing users to create a large multimedia library that is stored not only locally but also in the cloud, increasing the availability and reliability of data storage.

Plex integration Pass in Plex Media Server significantly expands the functionality of the service, allowing users to access a variety of additional tools and enhanced features.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>16 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

It provides seamless streaming of media content, enhanced security, additional synchronization options, and flexible library management, making the user experience more comfortable and customized to individual needs [51].

### 1.9 Using docker for isolated plex container management

Using Docker to manage a Plex container in isolation Media Server is an efficient and convenient method of deploying Plex in modern environments because it allows you to isolate the Plex server from the main operating system, providing a high level of portability, security, and ease of setup.

Docker is a platform for automating the deployment, scaling, and management of containerized applications, allowing you to create an environment for Plex with minimal time and resource consumption.

Docker allows you to isolate Plex Media Server in a separate container, which has a number of advantages:

- Plex environment isolation works in a container, which reduces the chance of conflicts with other applications and dependencies on the host system.
- Plex container can be easily moved between different servers or hosts without the need for reinstallation or configuration.
- Upgrading Plex in a container can be done without changing the underlying operating system and its dependencies, simply by rebooting the container with the new version.
- Docker allows you to run multiple instances Plex on different servers or in different containers for scaling, which is convenient for large data volumes and high load.

To create a Plex container using Docker, there are a few steps you need to take. First, you need to install Docker on your server. You can do this using standard Linux packages (e.g., via ` apt ` or ` yum `), or you can download the installer from the official Docker website.

```
bash
```

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>17 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

```
sudo apt-get update
```

```
sudo apt-get install docker-ce
```

Plex Media Server can be run in a Docker container using the standard Plex image, which is available on Docker. Hub. To do this, you need to create a container by running a command that will download the appropriate image and start Plex, specifying a public port through which you can connect to Plex, specifying a directory on the server to save Plex configurations, and a directory for media content that Plex will use to store files.

Once the container is started, Plex The Media Server will be accessible via a web interface at a specified address [52].

In the interface, the user can configure the server, add media libraries, configure content access, and manage users.

Docker makes it very easy to scale and upgrade Plex. For example, to upgrade Plex to a new version, you can simply stop the container, download the new version of the image, and restart the container.

Using Docker to manage Plex also has several advantages in the context of security:

Plex runs in a separate container, which minimizes the risk of interaction with other applications on the host system.

Because Plex runs in a container, you can quickly update the container image without having to update the underlying operating system [53].

Because Plex configuration and media files mapped to external volumes, they can be easily backed up using standard Docker tools to preserve container data. This ensures data security even if the container is deleted or overwritten.

Using Docker to manage Plex Media Server provides many benefits, such as application isolation, portability, ease of updating and scaling, and increased security. Docker makes Plex setup and management simpler and more convenient, allowing users to quickly deploy, scale, and update their server while maintaining control over resources and security.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>18 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

## 1.10 Conclusions to the first section

The first section provides a detailed analysis of software and hardware tools that allow for the effective implementation of cloud storage. It is established that the use of cloud environments is a modern and cost-effective practice that provides flexible access to data, scalability, increased security, and reduced infrastructure costs. The feasibility of choosing Plex technology as the basic platform for managing multimedia content is justified - due to its ability to catalog, stream, secure access, and integrate with cloud services. The advantages of using the Linux operating system are also considered, in particular, stability, open source, and support for containerization via Docker, which provides process isolation, ease of deployment, and server security.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>19 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

## 2 DESIGN OF A SOFTWARE AND HARDWARE TOOL FOR IMPLEMENTING CLOUD STORAGE ON THE BASE OF LINUX WITH PLEX TECHNOLOGY

### 2.1 Overview of the selected cloud storage technology

When comparing solutions such as Nextcloud, Seafile, Plex, Syncthing, and your own NAS solutions, it is advisable to consider them not only in terms of functionality, but also in terms of intended purpose, performance, configuration flexibility, and the level of technical complexity of implementation.

Nextcloud is one of the most popular open-source solutions for building a personal cloud environment. Its functionality is focused on document collaboration, file management, calendars, contacts, video conferencing, etc.

This is a comprehensive platform that can replace Google Workspace or Microsoft 365 within a small office or home network.

However, when it comes to media content, such as streaming video or music, Nextcloud requires additional modules and does not provide high-quality multimedia playback out of the box. Video transcoding, integration with TVs, support for Smart TV or mobile applications for media players - all of this in Nextcloud is implemented through third-party plugins or external services, which makes configuration difficult.

Seafile, while sharing roots with Nextcloud, is more focused on efficient synchronization and sharing of large files. Its architecture is built on the separation of metadata and file content, which provides better performance, especially in large installations.

However, Seafile does not focus on media streaming or advanced multimedia features.

This solution is a good choice for backup or centralized file storage, but is not suitable for a home theater or multimedia center.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>20 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Plex, unlike the previous two systems, is a highly specialized solution specifically for working with multimedia. Its main advantage is a high level of integration with client devices: TVs, smartphones, game consoles.

Plex automatically organizes libraries, downloads metadata, covers, descriptions, and ratings, and supports on-the-fly video transcoding, allowing you to play even formats that aren't supported by the target device.

Plex provides a great user experience and doesn't require much technical training to get started, but it has some reliance on proprietary components, and some features (like unrestricted mobile app access) require a paid subscription.

Syncthing is a fundamentally different approach to data storage and synchronization. It is a peer-to-peer system that does not require a centralized server: each device synchronizes with the others directly.

It is extremely secure, encrypts all traffic and does not store data on third-party servers. However, its functionality is limited to file synchronization - there is no support for streaming media, there is no web interface for viewing content, and in general it is not suitable for organizing multimedia storage focused on the convenience of watching videos or listening to music.

Custom NAS solutions are a broad term that includes both software packages (e.g. OpenMediaVault, TrueNAS ) and custom solutions based on Linux or BSD. They can be flexibly configured to meet specific needs: creating RAID arrays, configuring file protocols (SMB, NFS, FTP), implementing backups, connecting media servers (including Plex or Jellyfin ).

This approach provides maximum control, but requires a high level of technical awareness, especially when deployed in environments with limited hardware resources or the need for uninterrupted access.

In conclusion, Plex stands out as the most convenient and specialized solution for building cloud multimedia storage.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>21 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Its streaming functionality, multi-platform support, and ease of integration make it a viable choice for implementing a personal home media center, especially when combined with a dedicated Linux server.

## 2.2 Choosing a cloud type

In the modern information environment, cloud technologies are divided into several types based on access methods, resource management, and data privacy levels. The most common clouds are:

- private;
- public;
- hybrid.

Each of them has its own characteristics, advantages, limitations, and areas of application that directly affect the architecture and strategy for implementing cloud storage, in particular when implementing multimedia solutions based on Plex.

Private cloud (Cloud ) is an infrastructure that is fully controlled by a single organization or private individual.

All resources - computing power, storage, network services - physically or virtually belong to a specific user.

This approach allows you to maximize control over the level of security, limit external access, and customize services to meet specific user requirements.

A private cloud can be deployed both on-premises and on third-party equipment (for example, in a remotely managed data center).

In the context of a Plex project, a private cloud is an option where the media server is installed on your own server or NAS on your home network, with limited access via VPN or with encryption and firewall setup.

The main advantage here is security and control over data, and the disadvantage is the need to independently maintain the infrastructure, updates, backups, and monitoring.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>22 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Public cloud (Cloud ) is a model in which resources are provided by large cloud service providers such as Amazon Web Services, Google Cloud Platform or Microsoft Azure.

These providers provide virtual servers, storage, and platforms for development and data processing over the Internet on a subscription or pay-per-use basis.

Data and computing resources can be physically located anywhere in the world, and the client does not control the physical infrastructure.

In the case of Plex, it is possible to implement a media server based on a virtual VPS server in the public cloud, but this is associated with limitations on streaming large amounts of video data (especially HD/4K), high traffic, and costs for computing resources.

Public clouds are advantageous for rapid scaling, but they are inferior to private clouds in terms of privacy and long-term cost-effectiveness under heavy load.

Hybrid cloud (Cloud ) combines the advantages of both models - private and public.

It allows you to keep part of the infrastructure locally (for example, a media server Plex and the main file storage), and other components — backup, authorization, synchronization or streaming to remote devices — are implemented through public services. This approach provides a balance between security and scalability. A hybrid model could include, for example, backing up media files to Google cloud storage Drive or Amazon S3, while the main media content is streamed locally. It is also convenient for mobile access: mobile devices connect to the cloud, while the main streaming load is performed by the home server.

However, a hybrid architecture is difficult to implement because it requires thoughtful traffic routing, security policies, data synchronization, and duplication of critical services.

Thus, when choosing a cloud type, not only technical resources but also usage scenarios were taken into account.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>23 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

In the case of creating a software and hardware tool for home cloud storage with Plex, the optimal solution is a private or hybrid cloud, which provides proper performance, ease of use, and a high level of privacy when storing multimedia content.

### 2.3 Developing the Plex Architecture Media Server

Plex Media Server is a software platform that provides convenient organization, streaming, and access to multimedia content on various devices through a single centralized system.

Its functionality covers the entire lifecycle of interaction with media files - from automatic indexing to high-quality streaming taking into account the characteristics of the client device.

The main feature of Plex is that the user has complete control over their own media content, unlike commercial streaming services, where content is licensed and restricted by access rights.

On the functional side, Plex allows the user to create a personalized media library with movies, TV shows, music, photos, and video archives. After adding files to the appropriate directories, the system automatically scans the content, recognizes media types, obtains metadata from online sources (such as IMDb or TheTVDB ), and downloads posters, descriptions, trailers, subtitles, and ratings for the specified content.

This makes the library look like a professional catalog, and the user doesn't have to worry about organizing the content manually.

It is worth noting the intelligent search and filtering mechanism, which allows you to quickly find the desired content by genre, actors, release year or keywords.

Another important functional component is support for real-time streaming.

Plex automatically detects the technical capabilities of the client device - screen type, resolution, available bandwidth, supported codecs - and performs transcoding if necessary media files "on the fly".

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>24 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

This means that even if the video is saved in a format that is not supported by a specific TV or smartphone, Plex is able to change the format, bitrate, and resolution during streaming, ensuring smooth and high-quality operation of the service without user intervention.

From an architectural point of view, the Plex system is implemented using a client-server model.

At the center is Plex Media Server, which acts as the core of the entire system. It is installed on a server operating system - usually Linux, Windows or macOS, although variants with deployment in containers, on NAS or on virtual machines are possible.

The server is responsible for storing the library, indexing content, transcoding, access control, and interaction logic.

Client applications, which exist for most modern platforms ( Android, iOS, Smart TV, Xbox, Playstation, web browsers), connect to the server via an internal or external network, receive metadata, and make requests to view content.

Data transfer between the client and the server is implemented via the HTTPS protocol, which provides traffic encryption.

If the server is behind a NAT or firewall, Plex uses special routing mechanisms through its own cloud infrastructure to break through network barriers.

All this makes the service very convenient for remote access, even if the server is deployed on a home network without a static IP address.

It is also worth paying attention to multi-user access mechanisms.

Plex allows you to create separate profiles for different family members, restrict access to certain content, save viewing history, and implement parental controls.

The platform also supports integration with Plex Pass - a premium subscription that unlocks additional features

- mobile media caching for offline viewing;
- automatic subtitle download;
- advanced reports;
- hardware transcoding on supported hardware.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>25 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

Plex Media Server is a powerful example of a modern hybrid multimedia platform that combines local storage with cloud routing technologies, providing a high level of automation, end-user convenience, and configuration flexibility.

Its architecture allows you to implement both a local home media center and a full-fledged system for remote access to your personal video archive from any device in the world.

Architecture diagram of a software and hardware tool for implementing cloud storage based on Linux with Plex technology field in Figure 2.1.

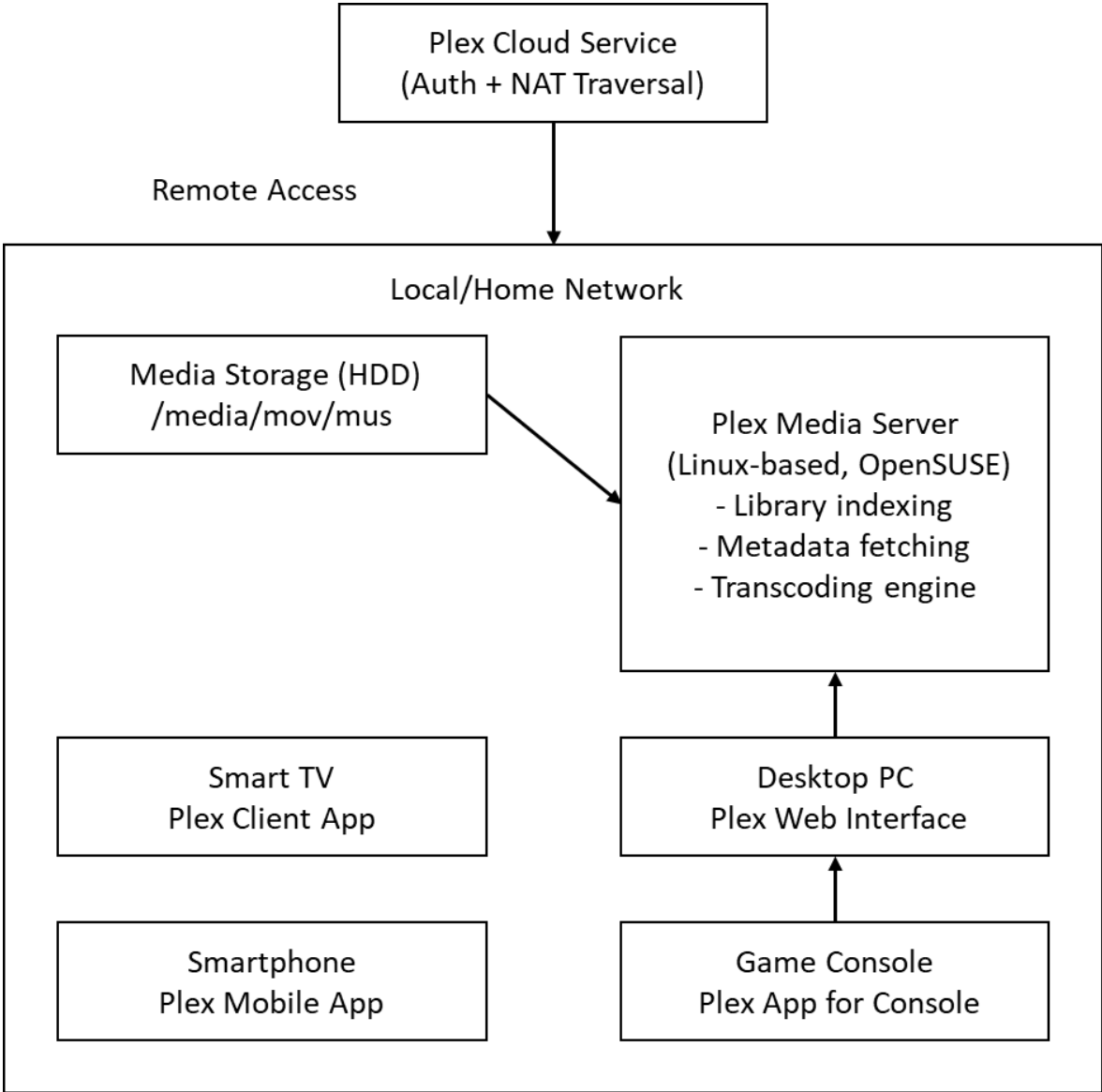


Figure 2.1 – Architecture diagram of a software and hardware tool for implementing cloud storage based on Linux with Plex technology

|     |      |          |        |      |
|-----|------|----------|--------|------|
|     |      |          |        |      |
| Зм. | Арк. | № докум. | Підпис | Дата |

Figure 2.1 describes a basic structure that can be adapted to specific user needs or extended for a hybrid scenario with cloud backup.

Plex Media Server is deployed locally on a Linux machine that has access to the media file system.

Media storage is a separate or built-in drive with directories containing videos, music, and photos.

Client devices are TVs, smartphones, PCs, consoles that connect to the server via a local network or the Internet.

Plex Cloud helps with remote access: NAT traversal, authorization, traffic redirection.

Transcoding is performed on the server if the client does not support a certain media format.

Plex architecture diagram Cloud Service in the context of its Plex support features Media servers for providing remote access, NAT traversal, user authentication, and traffic routing are shown in Figure 2.2.

This architecture allows Plex users to access their private media server from anywhere without having to change complex network settings.

Plex Cloud Core is a set of servers in the cloud that coordinate access between Plex clients and media servers.

They do not store content, but help find the server and ensure a secure connection.

Account DB is a Plex database that stores user accounts, a list of authorized clients, and servers.

NAT Traversal is a mechanism that allows clients to find servers even with dynamic IPs or behind NAT, using a proxy or direct connection.

Relay /Proxy Connection is used if a direct connection between the client and the server is not possible (for example, double NAT).

Media subsystem architecture diagram Storage, which is part of the overall Plex server infrastructure, is shown in Figure 2.3. It is responsible for storing, structuring, backing up, and accessing media files.

This diagram illustrates how a reliable media storage system is built that allows Plex Media Server to work efficiently and without loss.

Media Management is responsible for folder structuring, naming standards, accompanying subtitle files, cover art, etc. This simplifies Plex's automatic recognition Media server.

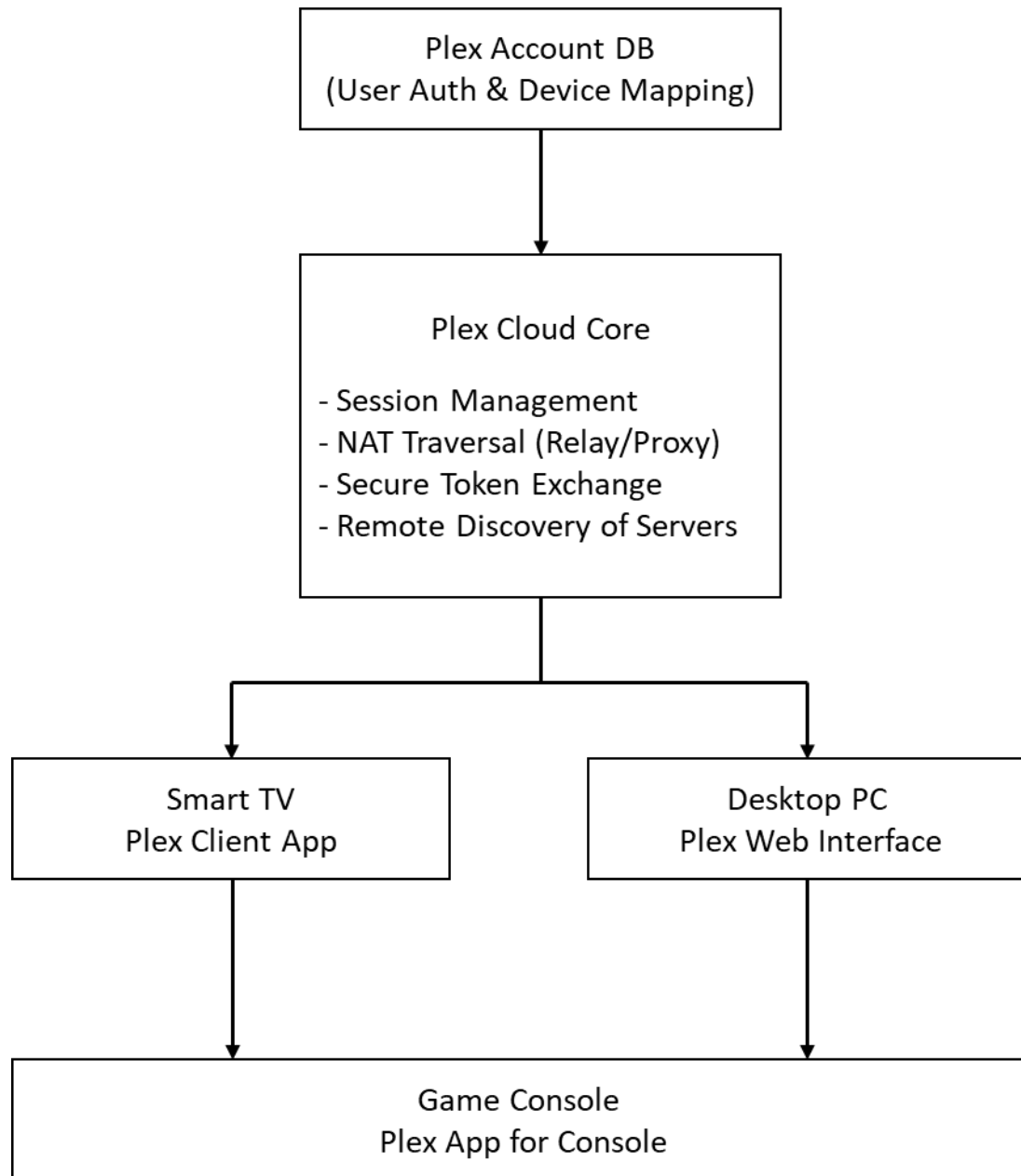


Figure 2.2 – Plex architecture diagram Cloud Service in the context of its Plex support features Media servers

File System Layer provides logical access to stored files, file organization, access rights support, and indexing.

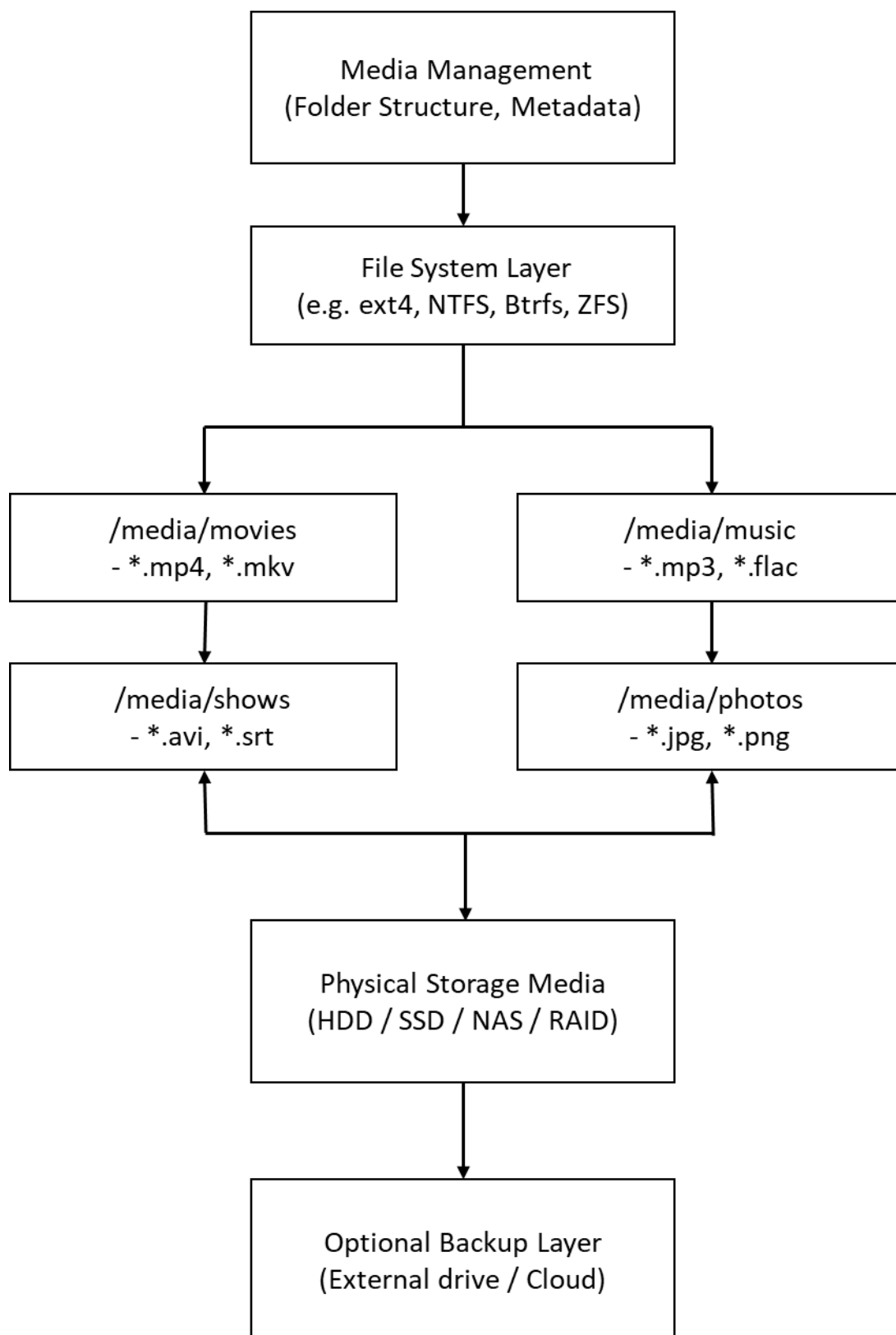


Figure 2.3 – Architecture diagram of the Media subsystem Storage

Content directories are organized by media type (movies, music, TV series, photos), and may have subfolders (e.g., season/series).

Physical storage is the disks or RAID arrays that directly store data. It can be connected locally or via a network (NAS).

Backup provides an optional level of protection, implemented via external drives or cloud services (e.g. rsync on Google Drive or Amazon S3).

Plex architecture diagram The Media Server, which is the central component that processes, indexes, broadcasts, and controls access to user media content, is shown in Figure 2.4.

Plex components interact. Media Server in the full cycle from storage to playback on user devices.

Plex Media Server is the core that provides all computing and coordination processes.

Metadata Fetcher is a module that retrieves information from sources such as IMDb, TheTVDB.

Media Library Index is a database that stores content structure, metadata, tags, and views.

Media Transcoder is a key component that adapts media to the client's format.

File System + Storage is access to local or network media through the file system.

Plex Cloud Services provides authorization, routing, and NAT traversal.

Clients connect directly or via the cloud, receive metadata and content.

Plex architecture diagram Cloud Core is the central cloud service responsible for Plex connections Media Server with connected clients, authentication, routing, NAT traversal and other cloud functions is shown in Figure 2.5.

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
|     |      |          |        |      |                           | 30   |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |      |

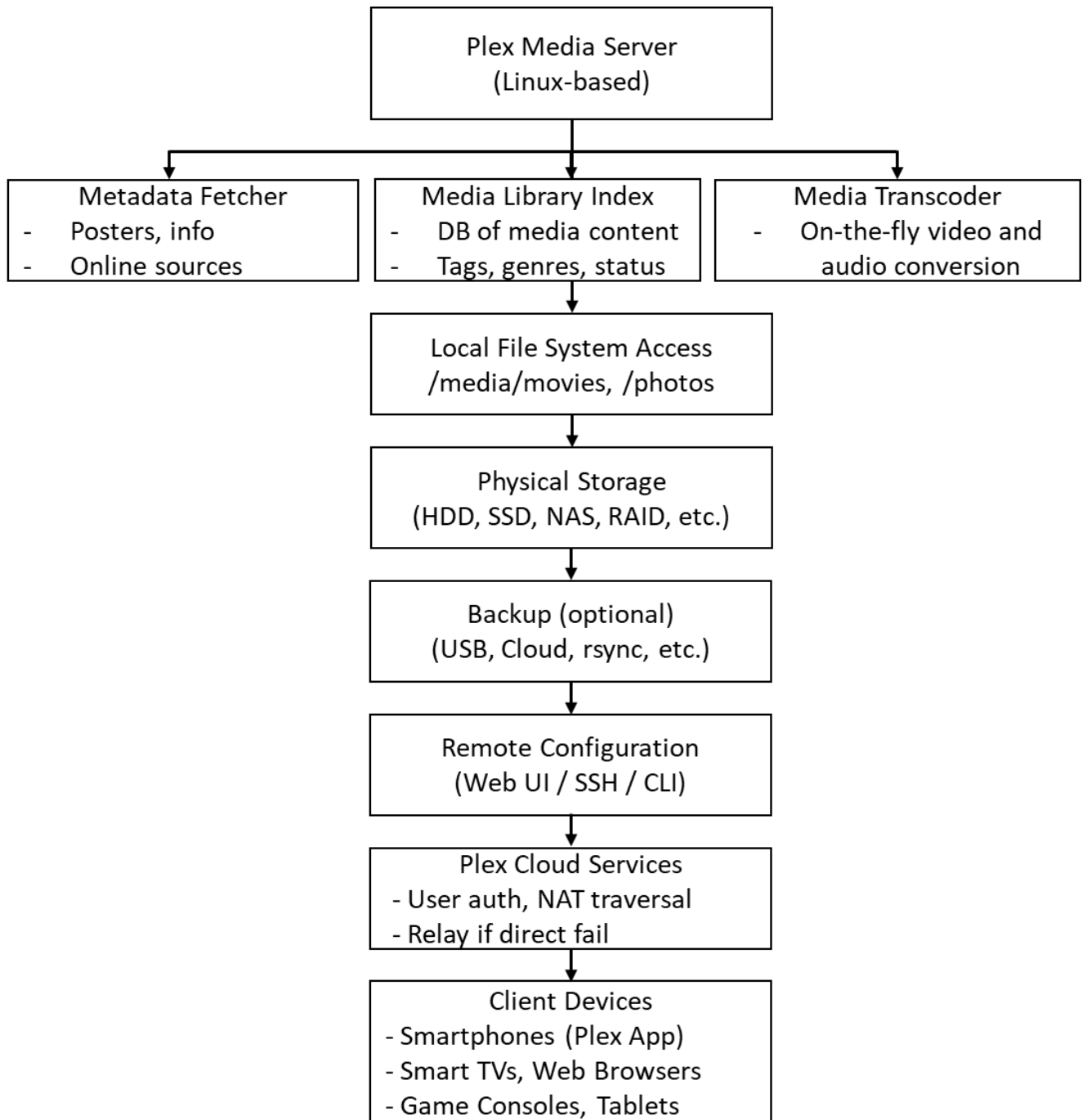


Figure 2.4 – Plex architecture diagram Media Server

The architecture allows Plex to provide "transparent" access to home media even for an unskilled user without configuring system ports or VPNs.

Plex Cloud Core is the logical core of the Plex cloud platform, acting as an intermediary between the client and the media server.

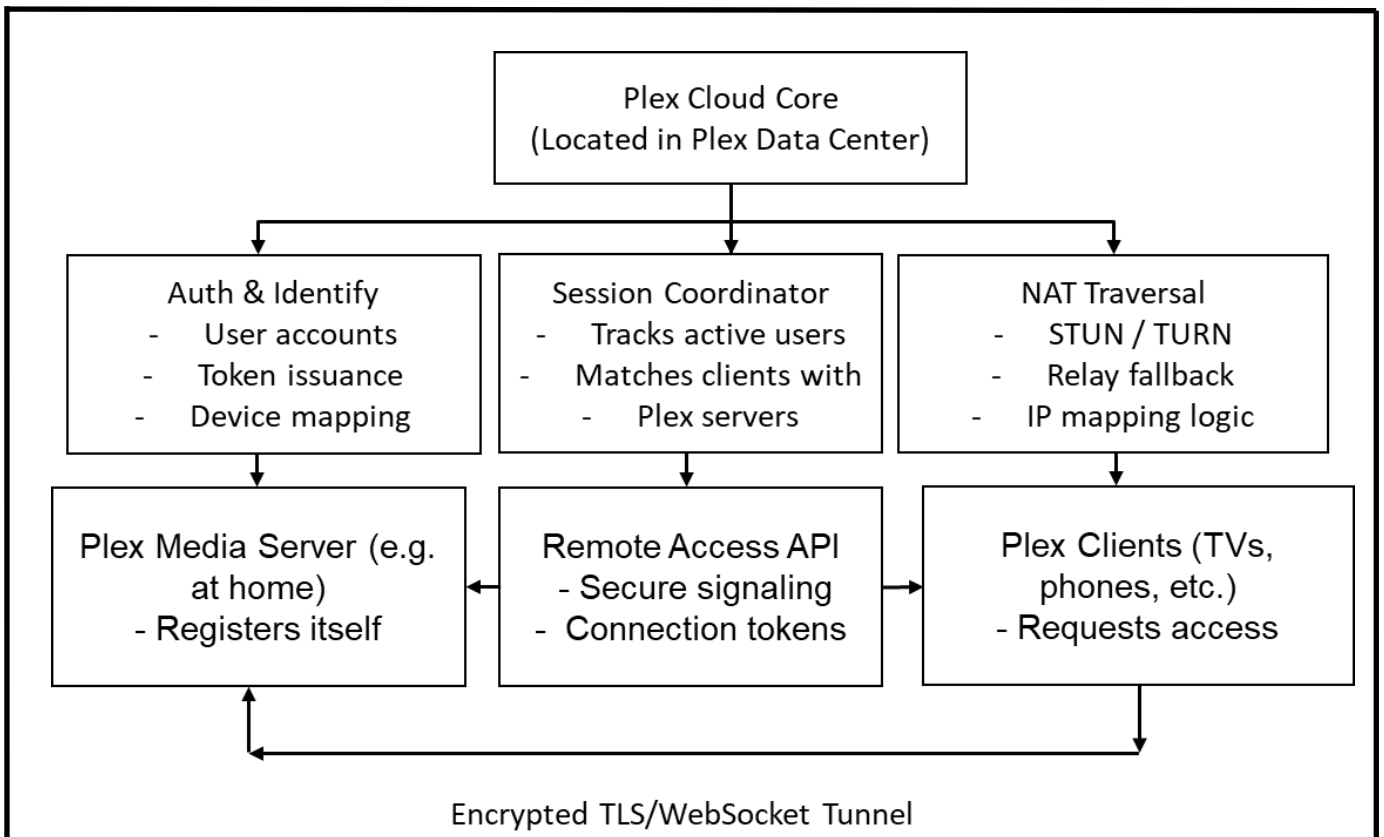


Figure 2.5 – Plex architecture diagram Cloud Core

Auth & Identity module is responsible for storing accounts, issuing access tokens, and binding devices to accounts.

Session Coordinator coordinates active sessions between Plex Media Server and clients, maintains up-to-date information about server availability.

NAT Traversal provides NAT traversal via STUN/UDP or TURN/ Relay if a direct connection is not possible.

The Remote Access API is used by both parties (server and client) to initiate a secure connection.

TLS/ WebSocket Tunnel is the primary channel for encrypted control and data flow when direct connection is not possible.

Plex Interaction Architecture Diagram Media Server ( Home ) with Plex Client ( Remote ) via Plex Cloud, taking into account route, authentication, NAT, proxy/direct connection is shown in Figure 2.6.

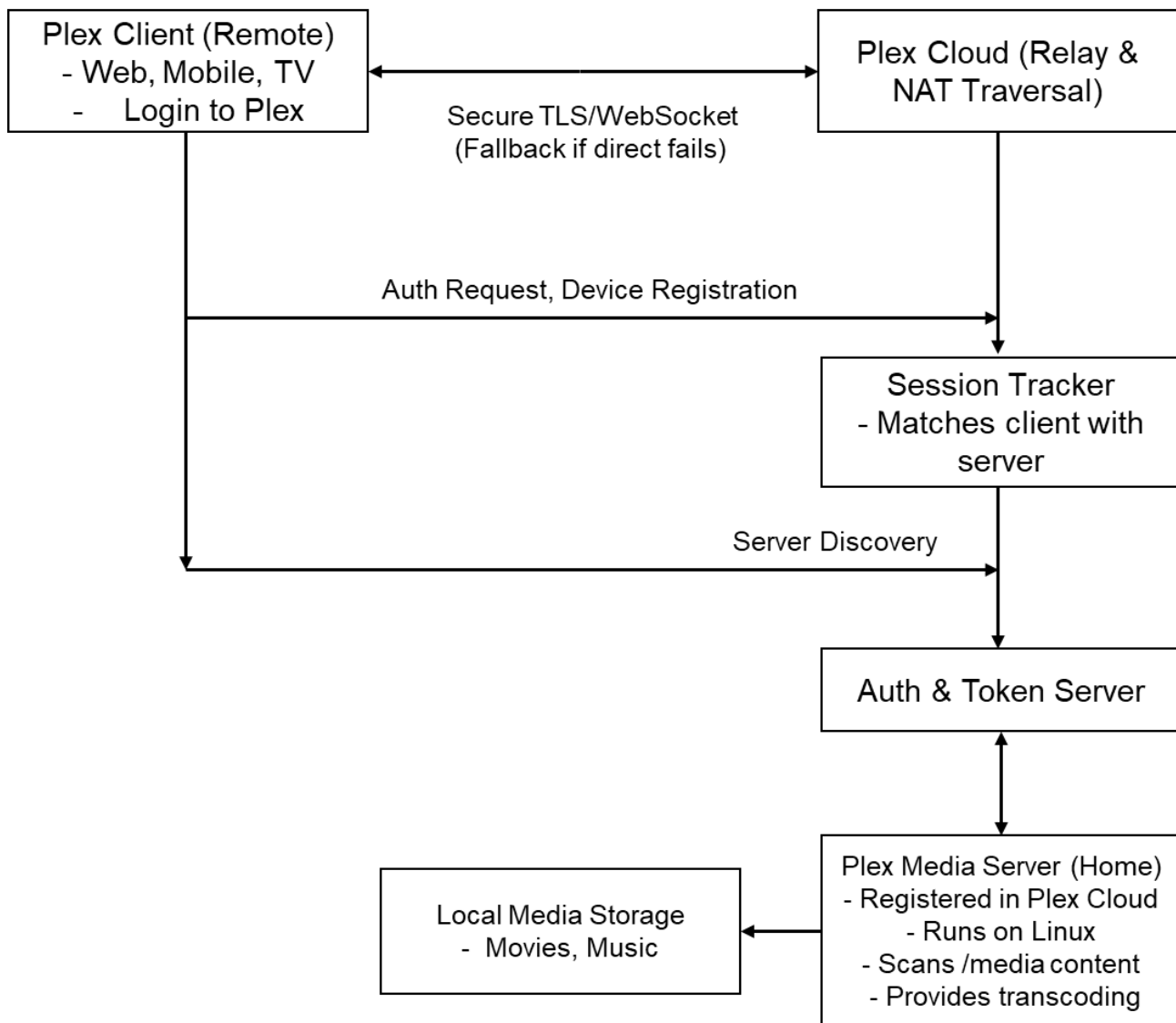


Figure 2.6 – Plex interaction architecture diagram Media Server ( Home ) with Plex Client ( Remote ) via Plex Cloud

The scheme allows a remote client to gain transparent access to the home multimedia library without any additional settings.

Plex Client ( Remote ) accesses Plex Cloud, authenticates and receives routing information.

Plex Cloud intermediary that:

- performs NAT traversal;
- creates a session between the client and the server;
- provides a fallback proxy channel (TLS/ WebSocket ) if a direct connection is not possible.

Plex Media Server ( Home ) is behind NAT or dynamic IP, but supports connection to Plex Cloud, announcing its availability.

The storage server has direct access to physical or network storage with multimedia.

Secure TLS Connection architecture diagram via Relay, which Plex uses when a direct connection between the client and the server (via NAT) is not possible, is shown in Figure 2.7. In this case, all encrypted traffic is transmitted through a cloud relay server.

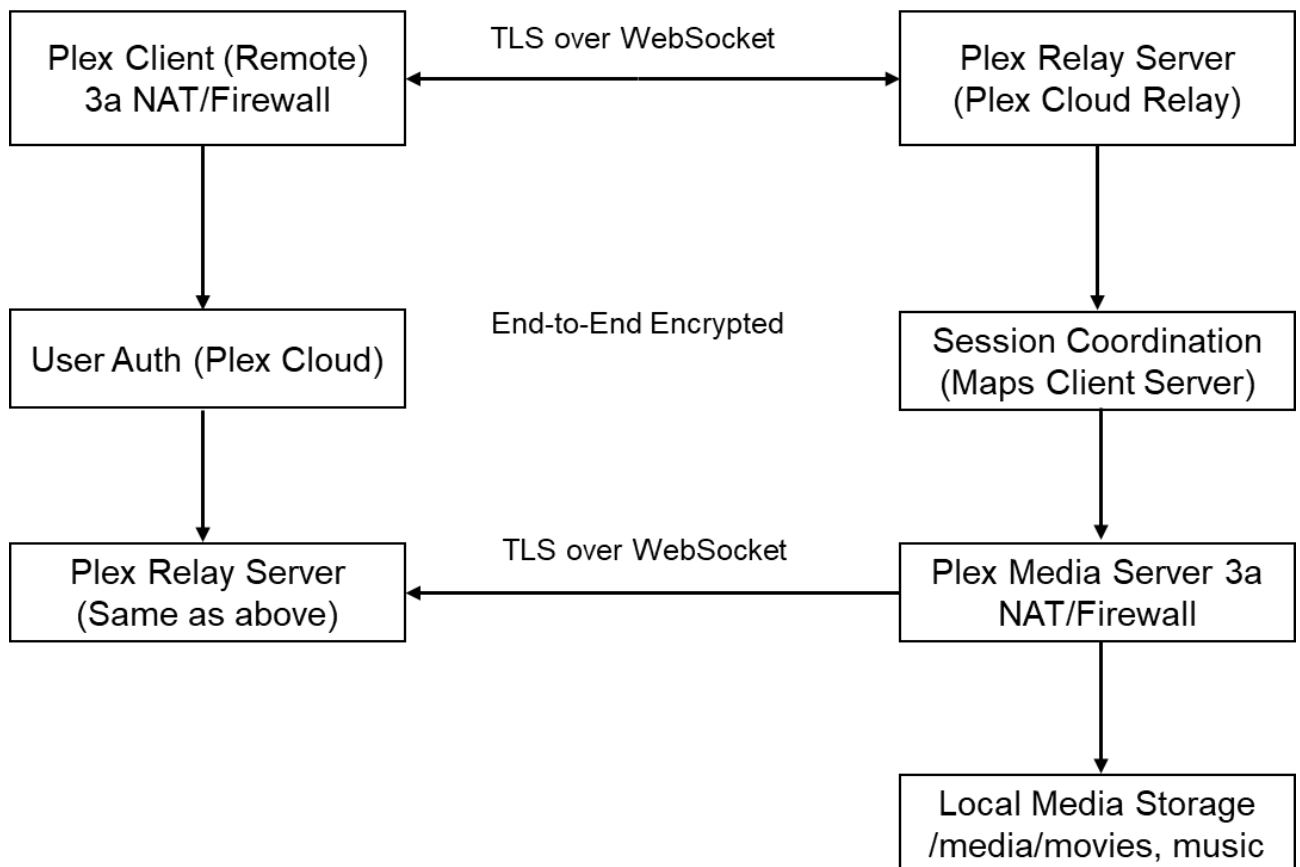


Figure 2.7 – Secure TLS Connection architecture diagram via Relay

This is a reliable, secure, but slightly slower alternative to a direct connection, which is convenient for ordinary users without technical training.

The essence of the architecture is that Plex The Client establishes a TLS connection to the Relay Server (in the cloud) because it cannot directly reach the home server.

Plex The Media Server, which also does not have a public IP address or is behind double NAT, itself initiates an outbound connection to the same Relay Server.

Relay Server acts as a two-way proxy, transmitting encrypted traffic between the client and the server without having access to the content (because the traffic is encrypted end-to-end ).

Plex then ensures that the video/audio stream will be delivered even in difficult network conditions, without opening ports or VPNs.

Media subsystem diagram Management in Plex Media Server is provided Figure 2.8.

It is responsible for the organization, analysis, indexing, metadata, and structuring of multimedia content.

This architecture provides automated management of a large media library with minimal user intervention, and also ensures that the library structure is consistent across all client devices.

Media Management Core is the main coordinator that initiates all scanning and updating processes.

Metadata Agent connects to external databases to automatically retrieve posters, reviews, genres, duration, cast, etc.

Library Scanner analyzes the file system, scans the / media folders, identifies new or changed files.

Media Matcher determines whether it is a movie, TV series, music, etc. based on file names and their location.

Path Resolver normalizes media paths, provides support for multi-user configurations and distributed NAS.

Metadata DB stores all extracted or generated data (locally, in the form of SQLite or a similar DB).

Thumbnail Generator generates visual previews ( posters, frames) that are displayed in client interfaces.

User Preferences DB takes into account personalized settings that affect sorting, metadata language, and display.

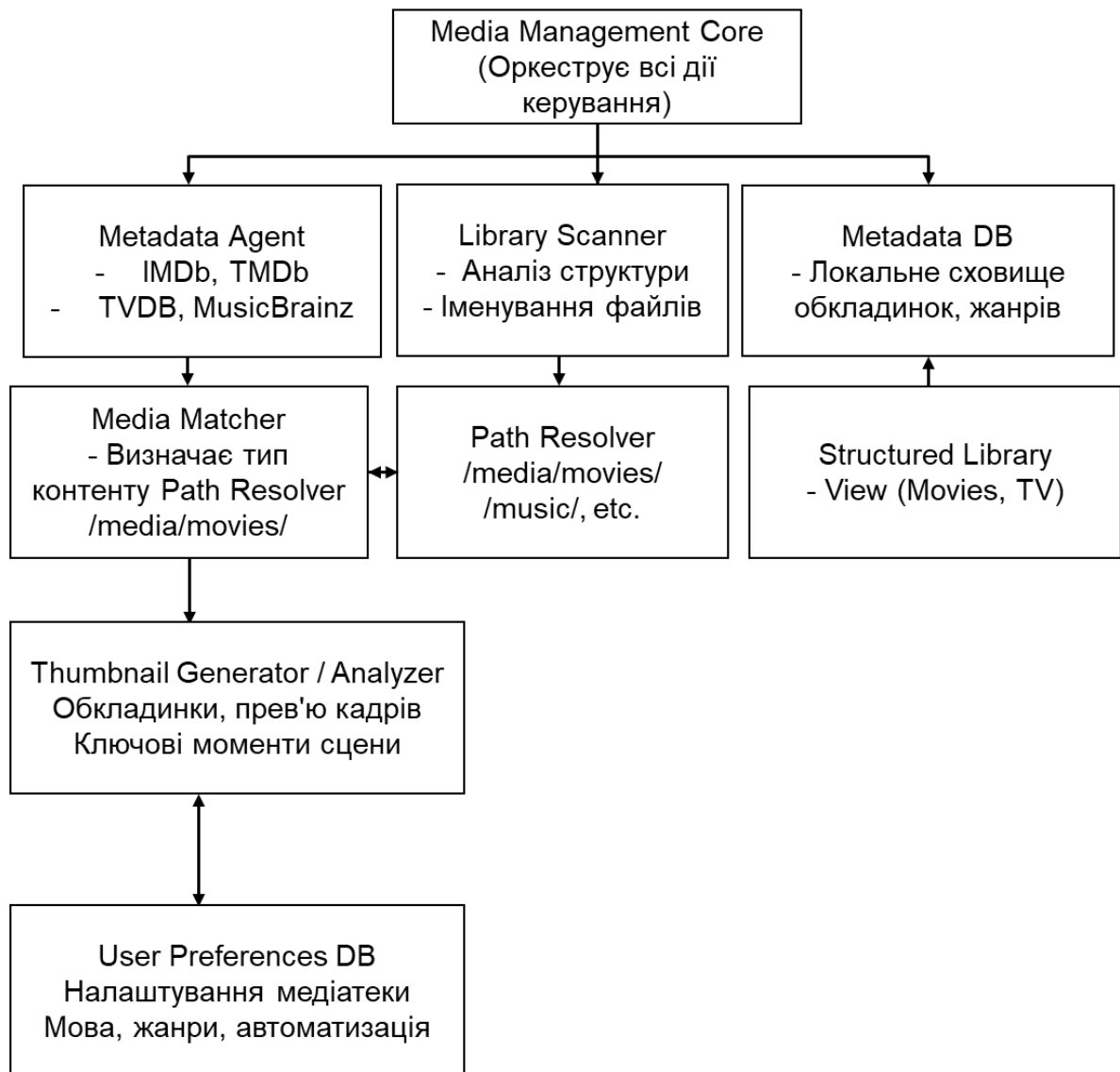


Figure 2.8 – Diagram of the Media subsystem Management in Plex Media Server

## 2. 4 Selecting a data exchange protocol

Data exchange between multimedia storage system components and end clients is based on several key protocols, each of which has its own scope, advantages, and limitations.

In the context of Plex implementation Media Server, as one of the options for building cloud media storage, it is important to consider three main protocol approaches: DLNA, HTTPS, and Plex's own API.

#### 2.4.1 DLNA

DLNA ( Digital Living Network Alliance ) is an industry standard for streaming multimedia between devices on a local area network.

It works on a set of technologies, including UPnP ( Universal Plug and Play ), HTTP for content transfer, and SOAP for service message exchange.

Plex Media Server has built-in DLNA support, allowing you to serve devices that do not have a native Plex client but support DLNA (e.g. Smart TVs, game consoles, Blu-ray players).

The DLNA approach does not require the installation of additional software on the client side, but is limited in the flexibility of accessing advanced Plex features - such as viewing synchronization, dynamic transcoding or access over the Internet. In addition, DLNA works exclusively on a local network and does not support modern authentication mechanisms.

#### 2.4.2 HTTPS

To transfer data between Plex Media Server and clients working through official applications or the web interface use the HTTPS protocol.

This is an extension of standard HTTP that works on top of TLS ( Transport Layer Security ). Layer Security ), providing traffic encryption and protection against data interception. HTTPS implements:

- transfer of metadata (information about movies, TV series, music);
- user authentication via Plex Cloud ;
- managing viewing sessions;
- transmitting commands (e.g., "play", "pause");

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>37 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

- streaming video (in cases where a Web client or mobile application is used).

If a direct connection is unavailable, Plex automatically tunnels HTTPS traffic through its own Relay servers, maintaining encryption and end-to-end data protection between the server and the client.

### 2.4.3 Plex's own APIs

An important role in building interaction between Plex system components is played by the proprietary Plex API, which is implemented as a set of RESTful interfaces accessible via HTTP(S).

The Plex API is not fully documented, but an active community of researchers has been able to partially reverse engineer the functionality of these APIs.

Through the API, the following is implemented:

- access to the library structure: movies, series, seasons, tracks;
- requests to scan new media files ;
- getting a list of current plays and online users;
- remote server management (restart, library update, synchronization, etc.);
- access to information about devices, sessions, views.

The API can be used to develop your own client interfaces, automation scripts, or integration with other systems, such as smart home systems.

Most APIs require authentication via a user token or Plex's OAuth mechanism, and only work when connected to Plex. Cloud or local network with permission enabled.

Plex implements a multi-layered system of data exchange protocols for intranet interaction, ensuring compatibility with older devices.

HTTPS and native APIs enable a modern, flexible, secure, and remotely managed multimedia cloud storage system.

Knowledge and understanding of these protocols is key to building a robust software and hardware architecture based on Linux and Plex.

## 2.5 Choosing an operating system

Choosing an operating system is one of the fundamental stages in developing a software and hardware tool for cloud storage, especially when it comes to a platform that should provide stability, security, ease of maintenance, and compatibility with multimedia services, including Plex.

In this context, the decision was made to use openSUSE Linux as a base operating system, combining the flexibility of Linux distributions with enterprise-level tools.

One of the key advantages of openSUSE is its focus on both server and desktop use. It supports a wide range of server components out of the box, including systemd, Zypper for package management, an automatic update system, and tool support for containerization.

This allows you to create a stable environment for Plex to work in. Media Server that is easy to maintain even in semi-automatic mode.

Thanks to the presence of the YaST ( Yet another Setup Tool ), openSUSE provides additional convenience for configuring the network, storage, users, and services, which is especially important when working with NAS systems or home servers, where the administrator may not have deep expertise in command-line system administration.

YaST also simplifies the deployment of Samba, NFS, or firewall services required for secure access to media content.

In terms of compatibility with Plex, openSUSE fully meets the system requirements: it supports all the necessary libraries, has an active community, and Plex Media Server officially provides builds for `rpm` based distributions.

This allows you to easily install and update Plex via the standard Zypper tools or manually via `rpm` without breaking the system structure.

Of particular note is the openSUSE update model. Within the two branches Leap and Tumbleweed, you can choose between stable ( Leap ) for production solutions or rolling ( Tumbleweed ) for testing and getting the latest features.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>39 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

When building a cloud storage system with an expected long lifecycle, openSUSE is preferred. Leap as a more stable and predictable platform with long-term support.

With a high level of security, regular kernel patches, integrated AppArmor process isolation mechanism, and the ability to encrypt storage directly at the OS level, openSUSE creates a reliable environment for storing and streaming media content.

Along with Plex, which requires efficient resource management, such an OS allows for a balance between performance, security, and scalability.

Therefore, the choice of openSUSE as the base operating system is due to its technical maturity, ease of administration, compatibility with Plex, stability in operation, and the ability to flexibly configure the system taking into account both server load and multimedia needs.

### 2.6 Overview of the cloud storage technology used

Server hardware requirements are critical to understanding the technical foundation needed to implement reliable, scalable, and performant cloud storage based on Plex Media Server.

Properly selected hardware determines how effectively the server will cope with the tasks of broadcasting, transcoding, indexing, storing, and distributing media content.

At the heart of any media server is the central processing unit (CPU). This is especially important for Plex, as the CPU is responsible for transcoding video streams in real time if the client device doesn't support the original format or if the bitrate needs to be reduced for a slower connection.

A 4-core processor with hardware transcoding support (e.g. Intel) is considered the minimum acceptable. Quick Sync or AMD VCE/VCN), however, for stable operation with multiple simultaneous threads, it is better to use modern 6–8-core processors.

An important factor is support for AVX or AVX2 instructions, which accelerate multimedia processing and encryption.

The next key component is RAM. While Plex as a service is not particularly memory-intensive, in practical applications RAM is heavily used for caching metadata, pre- buffering streams, and running background processes (such as indexing or thumbnail generation).

To build cloud storage with Plex on Linux, the base volume must be at least 8 GB.

When adding additional services, such as monitoring systems, access via SAMBA/NFS, indexing large media libraries, or using Docker containers, it is advisable to allocate 16 GB or more, depending on the load.

The disk subsystem acts as both media storage and space for system and temporary files.

The reliability, performance, and scalability of this subsystem directly affect video download speed, server response time, and the ability to work in parallel with large amounts of data.

The primary media storage is usually large-capacity hard drives (HDDs) (4–10 TB or more) combined into an array (e.g., RAID 5 or ZFS pool).

For the system partition, it is advisable to use SSD drives, which provide high performance when loading the OS and processing databases used by Plex.

It is also important to organize backups - even at home, it is advisable to provide for data duplication or periodic replication to an external drive or other storage.

An equally important component is the network subsystem. For internal transmission of streams in a local network, 1 Gbps is sufficient. Ethernet, but for scalable solutions or working with multiple HD/UHD streams simultaneously, a 2.5 Gbps or 10 Gbps interface is recommended.

Latency is especially critical when streaming over the Internet: Plex uses HTTPS or Relay servers, so a stable connection with minimal packet loss significantly improves the quality of streaming. It is also worth paying attention to the stability of the router, the ability to forward ports ( port forwarding ) or NAT-PMP / UPnP support to provide direct access to the server without the mediation of Plex cloud Relay servers.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>41 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

The formation of requirements for server equipment is based not only on the theoretical characteristics of Plex, but also on a practical understanding of its load: number of users, types of content, number of simultaneous sessions, method of access (locally or remotely).

## 2.10 Conclusions to the second chapter

The second section focuses on system design. A comparative analysis of existing cloud storage solutions, such as Nextcloud, Seafiler, Syncthing, and Plex, was conducted, which justified the choice of Plex as the optimal solution for a multimedia environment.

Developed Plex architecture Media Server, data exchange protocols (HTTPS, DLNA, proprietary APIs) have been defined, which allows for compatibility with client devices.

OpenSUSE operating system based on Linux was selected, which meets the requirements for stability, security and openness. Hardware requirements were also formulated, taking into account the typical load, the number of simultaneous connections and the specifics of use. Thus, the architectural foundation for further implementation was laid.

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                           | 42   |

### 3 SOFTWARE AND HARDWARE IMPLEMENTATION OF LINUX-BASED CLOUD STORAGE WITH PLEX TECHNOLOGY

3.1 Architecture of the software and hardware tool. Diagram of interaction of components: server, Plex, file storage, user

The architecture of the software and hardware tool that implements cloud storage based on Linux with Plex integration is built on the principles of modularity, distributed responsibility, and a client-server interaction model. The basis of the system is the server part, deployed on a Linux distribution (in this project, openSUSE ), which combines computing, networking, and storage functions. User devices connect to this server platform via the official Plex client or compatible technologies such as DLNA or HTTPS.

At the center of interaction is Plex Media Server, which acts as a software media controller. It indexes data from local file storage, automatically scans the directory structure, generates thumbnails, covers, metadata (genres, year, actors, description) and forms a media library, accessible to the user through the client interface. Plex does not store files in its own format - instead, it provides a logical representation of media content, which is physically located in a file subsystem built on traditional Linux file systems, such as ext 4 or XFS. This subsystem is accessible to Plex through the corresponding directory paths specified when configuring libraries.

The file storage is implemented as part of the server's local disk subsystem. It can be structured as a regular monolithic directory or as a RAID / ZFS array with a high degree of reliability. If necessary, it is possible to organize shared access to the storage from other systems using Samba, NFS or WebDAV, however, for Plex, the main data source remains a locally mounted directory, which provides maximum performance during streaming reading.

At the other end of the architecture are user devices—smart TVs, smartphones, tablets, PCs, game consoles, media players —each of which interacts with Plex. Media Server via specialized clients or browser via HTTPS. When a request is made to play content, the server analyzes the characteristics of the client device (resolution, supported

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>43 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

codecs, network bandwidth), and then decides whether to transcode. If transcoding is necessary, Plex launches a background process that transforms the file in real time, adapting it to the technical parameters of the client. This leads to a load on the CPU and, depending on the settings, the use of a cache in RAM or temporary files on disk.

The key element that provides communication between the server and the client over the Internet is the Plex cloud service. Cloud, which acts as an intermediary for authentication and transmission of control commands.

In the case where direct access is not possible, proxying media stream via secure Relay TLS channel.

At the same time, the content transmission itself remains end - to - end encrypted, and Plex Cloud does not store the media files themselves.

All components interact in a single logical chain: the server accepts requests from the client, checks access rights, generates or broadcasts content, using the resources of the file storage and computing core.

Linux system architecture, each of the components can be configured in detail: for example, the server part can be deployed in a container, file storage can be automatically mounted at startup, and network rules can ensure priority of streaming traffic.

Thus, the architecture of the software and hardware tool provides for a clear segmentation of roles:

- File storage is responsible for the physical storage of user content.
- Plex is responsible for access logic and streaming.
- Linux server is responsible for coordination and security.
- The user is responsible for interacting with the library through the client interface.

This allows you to create a flexible, scalable and technologically advanced cloud media system.

Component interaction architecture diagram: server, Plex The Media Server, file storage, and user are shown in Figure 3.1.

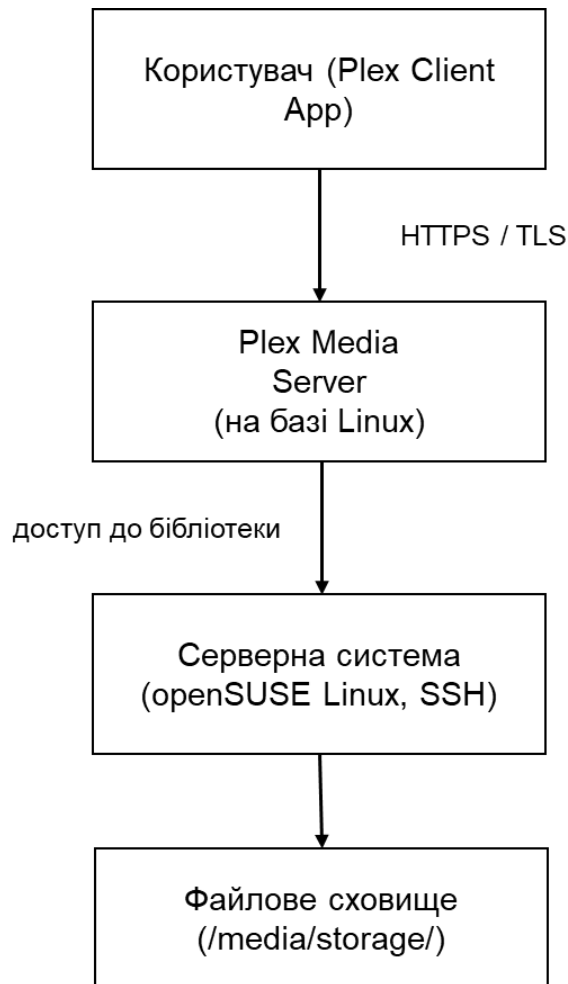


Figure 3.1 – Diagram of the architecture of component interaction: server, Plex Media Server, file storage and user

The architecture allows for centralized content management, ensuring continuous user access from anywhere in the world.

User connects to Plex Media Server via client application or web interface.

Plex The Media Server processes the request, accesses the file storage, reads the required data, and, if necessary, transcodes it.

Linux server OS ( openSUSE ) manages resources, network, storage, and also ensures the stable operation of Plex.

File storage contains multimedia content available to Plex for streaming.

## 3.2 System software development

Consider the system software that implements system encryption.

It checks if there are any tasks to scan for new movies or TV shows in the plex-scan file, and for each one, it launches Plex Media Scanner with the specified path and corresponding library section.

After processing the records, it clears them from the file so as not to rescan.

Let's present the script in pseudocode with steps:

- the script directory path;
- set the ple\_dir variable to the directory where the current script is located;
- go to the ple\_dir directory.

2. Reading the configuration file. ple.conf file exists in the current directory, then load variables from this file

Otherwise:

Display a message about missing configuration

Write an error to the log upload-error

script execution

Logging the start of scanning:

Add a separator to the plex-scan.log file

Record the date and message " Starting plex scan " to the log file

3. Check for scan jobs

If the plex-scan file is not empty:

If the paths to TV series or movies are not mounted, but the corresponding option is enabled:

Write error message to log

End script

4. Reading the task list from the plex-scan file

For each line in the plex-scan file :

Read a string as line

Split a string by the ":" character into the array current

If the first element of current is " tv ":

Variable show = second element

Variable season = third element

Build the path to the series folder

Set the LD\_LIBRARY\_PATH environment variable

Run Plex Media Scanner to scan this series folder

Record scan completion in the log

Otherwise, if the first element of current is " mov ":

Variable folder = second element

Build the path to the movie folder

Set LD\_LIBRARY\_PATH

Run Plex Media Scanner for scanning film

Record scan completion in the log

Log that the line was scanned

Increase counter n by 1

#### 5. Cleaning scanned lines

Remove the first n lines from the plex-scan file

### 3. 3 Algorithm of system operation using Plex, Rclone and Cloud Drives

Let's consider the algorithm of functioning of the software and hardware implementation of cloud storage based on Linux with Plex technology.

Algorithm of system operation using Plex, Rclone and Cloud Drives includes steps:

1. Preparing cloud storage via Rclone.
2. Uploading multimedia files to the "input" directory.
3. Automatically transfer files to the cloud.
4. Writing information about new files to the scan queue.
5. Verifying the cloud storage mount.

6. Plex scans only specified directories.

7. Completion of the process.

At the initial stage, the system administrator or user configures access to one or more cloud storages (for example, Google Drive ) via the rclone tool.

This allows the system to work with the cloud as a remote file storage, mounting it as part of the file system.

The user or another subsystem (for example, an automated downloader) adds new media files (movies or TV series) to a special local download directory.

This directory is not part of the Plex library, but is used as a buffer for further processing.

Periodically or event-based (for example, via cron or systemd.timer ) that:

- analyzes the contents of the input directory;
- identifies new or changed files;
- copies or moves these files to the appropriate directory on Google Drive (or other cloud storage) using `rclone copy` or `rclone move`;
- when using multiple Google accounts Drive, the system can duplicate files to two or more storage devices using the RAID 1 principle, i.e. for redundancy.

After a successful upload, the script adds an entry to a queue file (e.g. `plex-scan`) stating:

- media type (`tv` or `mov`);
- show/movie titles;
- season numbers (for TV series);
- the entry is structured in the format `tv:ShowName:Season 1` or `mov:MovieFolder`.

A separate script checks if the cloud storage is mounted on the filesystem via `rclone mount`. This is important because Plex needs to have direct access to the files for indexing.

Google's API quota to be exceeded ), the script:

- reads the `plex-scan` queue ;

- - for each entry, it defines the exact path to the new directory in the cloud storage (for example, `/mnt/plexdrive/TV/ShowName/Season 1`);
- - launches Plex Media Scanner with parameters: `-- scan` and `-- refresh` for updating; `-d` with specifying a specific path; `-c` with the number of the corresponding library (movies or TV series);
- - after a successful scan, removes the processed record from the queue to avoid repetitions.

The system updates the logs (`plex-scan.log` and `upload-error`) with the results of operations – this allows for auditing and diagnostics.

By scanning only the newly added portion of your media library, rather than the entire collection, the number of queries to Google is significantly reduced. Drive API.

This allows you to avoid errors like "API quota exceeded", which often occur with large libraries in Plex.

Files can be copied to multiple Google accounts at the same time. Drive accounts, which provides reliability when losing access to one account, the data remains on the other, as well as offloading the API by scanning, which can be distributed across accounts.

The system combines upload automation, incremental scanning of the Plex library, data backup, and API quota savings, making it a highly efficient solution for Plex-based cloud media storage.

Flowchart of the system operation algorithm using Plex, Rclone and Cloud Drives is shown in Figure 3.2.

### 3.4 Installing system software

Let's take a detailed look at the process of installing system software on a Linux system for a Plex cloud media storage project. This walkthrough provides a step-by-step installation with an explanation of each step.

The list of stages is as follows:

1. Cloning the system software repository.
2. Granting access rights.

### 3. Installing Rclone.

Google Account Configuration Drive.

5. Mounting cloud storage in the file system.

6. Defining Plex Library Sections.

The first step is to get the project code that contains the system management scripts. This is done using the `git clone`, which downloads the contents of a GitHub repository to the user's home directory.

```
``` bash
git clone https://github.com/roni/PLE
```
```

This command creates a `PLE` subdirectory with all the necessary scripts. This is the main structure that will control the loading, mounting, and scanning of content in Plex. Since Plex runs under a separate system user (`plex`), it is necessary to grant full access to the scripts and repository configurations to this user.

The `chmod -R 777` command grants full read, write, and execute permissions for all users — this is a temporary measure during the setup phase:

```
``` bash
sudo chmod -R 777 PLE
```
```

In real-world conditions, more secure permissions settings should be used on production systems.

`rclone` is a utility that provides synchronization, copying, and mounting of cloud storages such as Google Drive. First, you need to download the latest version of `rclone` from the official website (or using `curl`, `wget`, `snap` or `apt`).

The following actions are then performed:

- moving the binary file to a system directory that is already specified in the `$PATH` environment variable, for example `/usr/local/bin/`;
- setting access rights for this file.

Setting access rights involves executing the following system commands in the console:

```
``` bash
sudo mv rclone /usr/local/bin/
sudo chown root:root /usr/local/bin/rclone
sudo chmod 755 /usr/local/bin/rclone
```
```

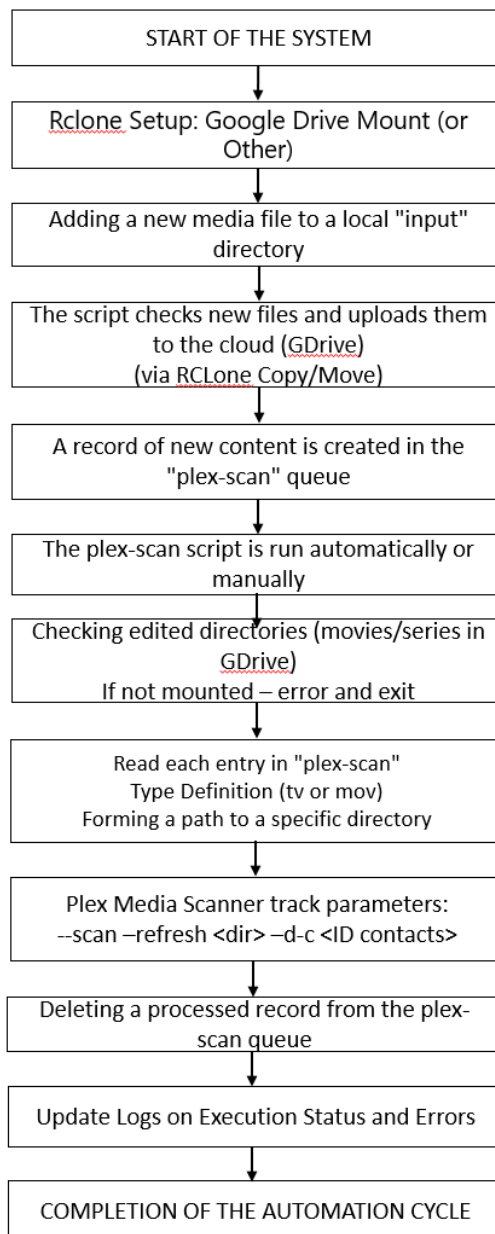


Figure 3.2 – Flowchart of the system operation algorithm using Plex, Rclone and Cloud Drives

After that `rclone` will be accessible from anywhere on the system.

`rclone config` allows you to create remote connections (remotes) to Google accounts Drive.

During the configuration process, the name of each connection is set (for example, `gdrive-main`) and authorization is performed via OAuth 2.0.

The results are stored in `~/.config/rclone/rclone.conf`.

In order for Plex to be able to read files directly from Google Drive, you need to mount the remote using `rclone mount`.

This provides access to the cloud as a local directory:

```
``` bash
sudo mkdir /mnt/gdr-main
sudo rclone mount --allow-non-empty --allow-other gdrive-
main :/ /mnt/gdr-main &
```
```

Let's look at the parameters used:

`--allow-non-empty` allows mounting a directory even if it is not empty;

`--allow-other` allows other users (e.g. `plex`) to access the mounted directory.

Alternatively, you can use ple, which also provides mounting but has a built-in cache for working with Plex.

In case of encrypted mount (via `rclone crypt`) it is important to make sure that Plex scans the decrypted directory, and the names of the encrypted remotes are used in the configuration files.

In order to specify the correct libraries (Movies / TV Shows), you need to determine the internal identifiers of these libraries in Plex.

To do this, the built-in scanner is called with the `--list` option under the `plex` user:

```
``` bash
sudo su -c 'export LD_LIBRARY_PATH=/usr/lib/ple;
/usr/lib/ple/Pl\Media\Scan --list' ple
```

```

The output will look like this:

```

1: Movies

2: TV Shows

```

These numbers (e.g. `1`, `2`) correspond to the `ple\_movies\_sec\_num` and `ple\_tvshows\_sec\_num` parameters, which must be written to the `ple.conf` configuration file. After completing all the above steps, the system is ready for automated media upload to Google Drive and scanning only new directories in Plex. This allows you to reduce the load on the Google API, maintain performance, and optimize system resource consumption. The flowchart of the SSP installation algorithm is shown in Fig. 3.3.

### 3.5 Application developed software and hardware implementation of cloud storage based on Linux with Plex technology

Plex integration Media Server with cloud storage based on rclone, using Google Drive as the main media for multimedia data.

For launch and everyday use, the system involves simple user interaction with two main scripts: uploading multimedia content to the cloud and initializing a scan of the Plex library for new files.

After the preliminary configuration of the `ple.conf` file, which includes specifying paths to local directories with media content, setting up remote Google Drive accounts, options for deleting files after downloading, allowed file types, and Plex library section numbers, the user can run the `ple.sh` script.

This script automatically detects new files in local directories, checks their types, and uploads them to the specified Google Drive in one or more instances similar to RAID1 using rclone, and, if necessary, clearing local storage after successful transfer.

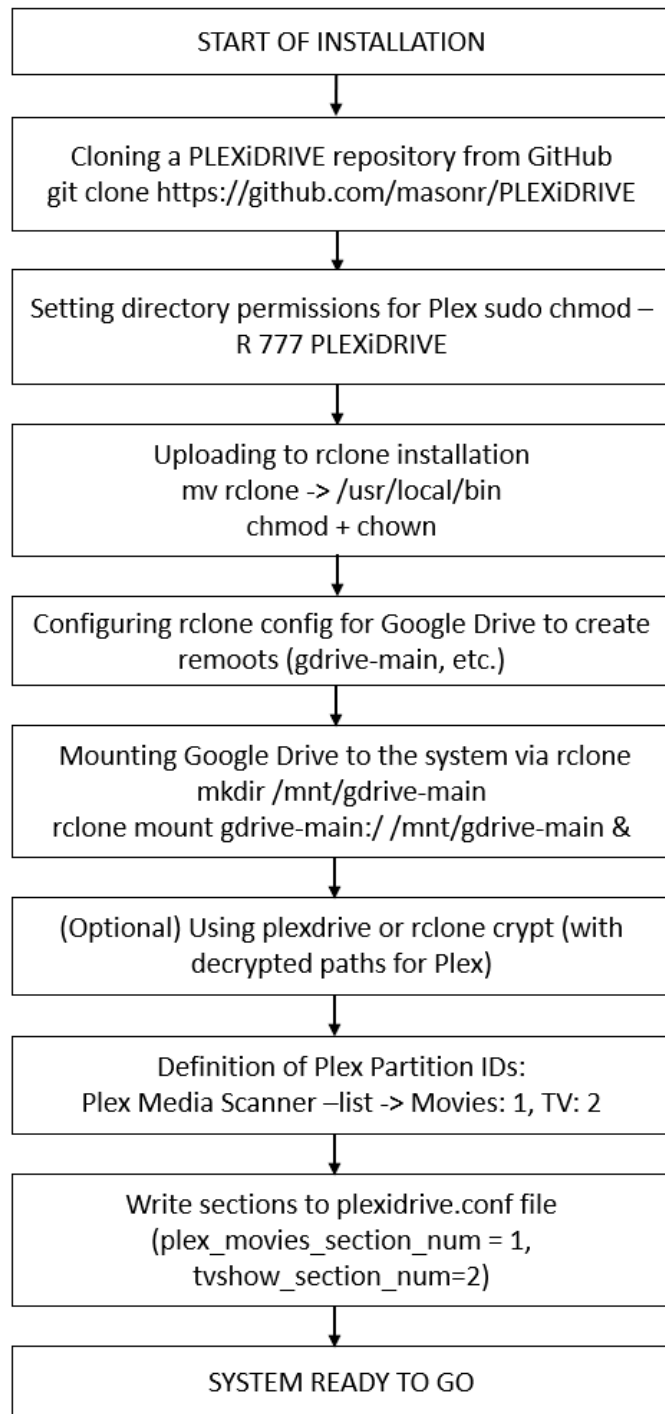


Figure 3.3 – Flowchart of the system software installation algorithm

After the data download is complete, the user runs another `ples.sh` script.

This script is run as the system user `ple`, and calls the built-in Plex scanner, which initiates an update of a specific section of the media library (for example, only movies or only a specific season of a TV series), instead of a full scan of the entire library.

This approach avoids exceeding Google API call limits. Drive, which is critical when working with large media collections.

To ensure continuous operation of the system, automation is possible via the `cron` task scheduler.

The user configures two independent `cron job`:

- the first one runs `ple.sh` at an interval, for example, every 4 hours
- second `ples.sh` from user `ple` immediately after the download is complete.

This organization ensures that new media files are regularly added to the library and immediately become available for playback without manual user intervention.

Thus, the system provides a full work cycle from placing media files in a local environment to their availability in Plex, integrated with the cloud infrastructure.

All the logic of operation is centralized around a flexible configuration file `ple.conf`, where the user can:

- change paths;
- change library identifiers;
- modify the list of supported formats;
- specify the number of Google Drive used;
- set behavior parameters after loading.

This allows you to adapt the system to individual needs without the need to modify the program code.

### 3.6 Configuring a single-storage configuration

Let's consider a description of the process of configuring a system with one cloud storage (using Google as an example). Drive ).

Google account Drive needs to configure the `ple.conf` configuration file, which is a key element for controlling the entire system.

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                           | 55   |

This file defines cloud storage connection parameters, media file processing specifications, Plex library paths, local media paths, and activity modes of individual subsystems.

The first step is to specify the number of cloud accounts that will be used for media uploads.

In the case of a single storage, the `num_of_gdrives` variable is set to 1.

Accordingly, in the `drive_names` array, only one name is specified, which corresponds to the name of the remote previously configured via `rclone config`.

For example, for the main Google The remote drive name could be `gd-main`.

The next block in the configuration are the options that define the behavior of the boot script.

In particular, the `del_a_up` variable allows you to set whether to delete local files after successfully copying them to the cloud.

This is useful in cases where saving disk space is important. In this configuration, the value is set to `true`, meaning the system will automatically clean up local directories after the transfer is complete.

`fil_tp` variable defines a list of video file formats to be processed: for example, `mkv`, `avi`, `mp4`, `m4v`, `mpg`, `wmv`, `flv`.

This avoids accidentally processing incompatible or other service files.

The `rcl_conf` parameter is left empty in this case, meaning the standard path to the rclone configuration will be used (usually `~/config/rclone/rcl.conf`).

If you need to use a non-standard path, you should specify it here.

The following are the paths to the Plex directories, which are the result of mounting the cloud storage using rclone.

In this case, the variable `ple_t_p` points to `/mnt/mVs`, and `ple_mp` points to `/mnt/main/Ms`.

These are the paths that should be mapped to Plex when creating system libraries.

A mandatory condition is the absence of the `/` character at the end of the path, which is taken into account in scripts to avoid duplication of separators in final paths.

The variables `ple\_ms` and `plets` correspond to the identifiers of the Plex libraries, which are defined separately via the special Plex command.

In addition, you must specify the paths to local directories where the source video files are stored before uploading to the cloud.

For TV series, this path is `/home/masonr/tv-shows/`, and for movies, `/home/roni/vid/`.

In this case, on the contrary, the `/` character at the end is required for correct processing in scripts.

The corresponding subsystems are activated in the configuration.

The `ene\_sup` variable determines whether to search for and download TV series, and `enupls` determines movies.

Both variables have the value `true`, meaning the system will process both content types simultaneously.

With this setup, the system is fully ready to work automatically with one Google Drive, which allows for continuous uploading of new content to cloud storage, its subsequent scanning by Plex, and end-user access to an up-to-date library of multimedia materials.

### 3.7 Configuring a multi-cloud storage configuration

Let's consider a detailed description of the process of setting up a configuration with two cloud storages.

In the case of deploying the developed system with two Google Drive storage, the goal is not only to store content, but also to ensure fault tolerance by duplicating media to two independent cloud accounts.

This architecture resembles the RAID 1 principle: each file, after being downloaded, will be automatically duplicated on the second storage, which significantly increases the reliability of data storage.

The entire process is controlled through the `ple.conf` configuration file.

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
|     |      |          |        |      |                           | 57   |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |      |

Consider the section containing the GD R parameters.

The first part of the configuration specifies the number of repositories to which the upload will be performed.

The ``ngdr`` variable is set to ``2``, which means using two cloud accounts.

Next, in the ``drm`` array, the ``rclone`` names are specified: ``gdrive-main`` is the main one, and ``gdrive-backup`` is the backup one.

These names must exactly match those previously configured in ``rclone config``.

Let's consider the section containing file processing options.

The ``delup`` variable is set to ``true``, which means that local media files will be automatically deleted after they are fully downloaded to both disks.

This is useful in cases where local storage is limited. ``file_types`` contains the extensions of video files that the system should detect and transfer to the cloud.

The list of formats has been expanded to ``mkv``, ``avi``, ``mp4``, ``m4v``, ``mpg``, ``wmv``, ``flv``, ``mpeg``, covering almost all major formats of user multimedia content.

The value of the ``rclone_conf`` variable explicitly points to the full path to the user rclone configuration file: ``~/home/roni/.conf/rclone/rcl.conf``.

This is important for situations where the configuration was created under a separate user account and not in system space.

Let's look at the section containing Plex settings /

The next step is to link the mounted directories to the corresponding Plex libraries.

In this configuration, the paths for movies and TV shows are different: ``pleoath`` refers to the primary drive (``/mnt/mTS``), while ``plep`` refers to the backup cloud storage (``/mnt/bacMs``).

This approach allows you to separate user content not only logically, but also physically.

Plex should be pre-configured so that the specified directories correspond to libraries like TV Shows and Movies.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>58 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

The library identifiers are specified using the variables `pleo\_sm` and `plesen\_m` in this configuration they are 2 and 1 respectively, which means the second and first section in Plex.

Setting local media directories is done as follows. The system expects new video files to be uploaded to local directories.

The path to TV series is set to `locpa` in `/home/roni/tvs/`, and the path to movies is set to `locp` in `/home/roni/vid/`.

The `/` symbol at the end indicates that these are directories, not individual files or relative paths.

Scanning these directories allows you to detect new files to copy.

Finally, the corresponding subsystems are activated: `ene\_shads = true` and `eneps = true` mean that the system will process both the user's TV series and movies.

If any of these options are set to `false`, the corresponding content category will be ignored.

The dual-storage configuration provides a balance between efficiency and reliability.

The entire system works transparently for the Plex end user, while automating routine operations and minimizing the risk of data loss through backup.

### 3.8 Google accounts Drive as storage for Plex server

Google accounts Drive as a storage for the Plex server, specialized system software was developed in the form of the `pled.sh` script.

It is the central element of the system for automating the upload of multimedia files to cloud storage and the subsequent integration of these files with the Plex server.

Its functioning covers several key stages:

- initialization;
- environment check;
- configuration processing;

- running auxiliary scripts for data loading;
- Initiating a scan of Plex libraries.

Let's consider the process of checking the uniqueness of an execution.

In the first stage, the script checks whether it is already running in another process.

This is achieved by looking at the list of all processes running a script named `pled.sh` and comparing their PIDs to the current process.

If it is detected that another copy of the script is already running, execution is interrupted.

This approach prevents multiple processes from loading the same files at the same time, which could lead to duplication or corruption of the data being used.

The environment is initialized as follows: the path to the current directory where the script is stored is determined, and a transition is made to it.

This is important so that all subsequent relative paths work correctly regardless of where the script is called from.

Let's consider the process of processing a configuration file.

The script checks for the existence of the `pled.conf` configuration file.

If the file exists, its contents are connected to the runtime environment via the `source` command, allowing the system script to use all variables specified in it.

If there is no configuration file, an error message is displayed both on the screen and in the `upload-error` log file, after which the script terminates.

Let's consider the process of preparing service files

At this stage, empty service files (`ples`, `ple.log`, `up-er`) are created if they do not already exist.

This ensures stable operation of scripts that can write or read data from these files.

Let's consider the process of checking the consistency of parameters

Next, a critical check is performed: the number of Google accounts Drive specified in `ngd` must match the actual number of drive names specified in the `ds` array.

In case of a mismatch, the script stops, recording an error in the `upload-error` file.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>60 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

This protects the system from incorrect operation due to incomplete or incorrect configuration.

Let's look at the process of downloading a TV show.

If the ``enads`` parameter in the configuration file is set to ``true``, then a helper script ``uws.sh`` is run, which is responsible for searching for new episodes of a TV show in the local directory and downloading them to all specified Google accounts. Drive.

Upon completion, the return code of this script is checked.

If the download fails, the main script stops running and displays an appropriate message.

Let's look at the process of downloading movies.

The block that processes films works according to a similar logic.

If ``enads`` is enabled, the ``uploies.sh`` script is called.

If this script returns a non-zero code (i.e. an error occurred), the main script terminates execution to avoid further incorrect actions.

Plex library scan.

The code provides the ability to automatically run the ``pln.sh`` script after downloads are complete.

This script runs as the ``ple`` user and updates Plex libraries using the CLI interface.

Its use is only possible when the ``plex_scan_after_upload`` parameter is explicitly enabled in the configuration.

The ``pled.sh`` script acts as a control mechanism in the developed system, providing:

- - unambiguous execution;
- - configuration compliance control;
- - manage individual content downloads by category;
- - flexible integration with Plex to ensure your media libraries are up to date.

Thanks to structured logic and error prediction, the script allows you to automate key stages of working with a multimedia server in a cloud storage environment.

Running the script in the OpenSuse operating system is shown in Figure 3.4.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>61 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |



## CONCLUSIONS

Linux-based cloud storage software and hardware tool was developed and implemented using Plex technology.

Based on an in-depth theoretical analysis and comparison of modern approaches to cloud data storage, the choice of appropriate technology and software and hardware architecture was justified.

The implemented system combines the advantages of open source software, containerization, automation, and cloud integration, which allows for efficient, secure, and scalable multimedia content management.

The results of the work can be used for implementation in home multimedia servers, educational projects, or small organizations that need efficient streaming and data storage without dependence on commercial providers.

The first section provides a detailed analysis of software and hardware tools that allow for the effective implementation of cloud storage. It is established that the use of cloud environments is a modern and cost-effective practice that provides flexible data access, scalability, increased security, and reduced infrastructure costs.

The second section focuses on system design. A comparative analysis of existing cloud storage solutions, such as Nextcloud, Seafiler, Syncthing, and Plex, was conducted, which justified the choice of Plex as the optimal solution for a multimedia environment.

In the third section, the hardware and software implementation of the cloud storage is carried out. The architecture of the server part interaction with Plex, the file storage and the user is developed.

|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
|     |      |          |        |      |                           | 63   |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |      |

## LIST OF SOURCES OF LINKS

1. Hernandez S, Lazcano R, Serrano A, Powell S, Kostousov L, Mehta J, Solis LM. Challenges and opportunities for immunoprofiling using a spatial high-plex technology: the NanoString GeoMx ® digital spatial profiler. *Frontiers in Oncology*. 2022. Vol. 12.pp. 890410.
2. Ramamurthy M, Kumar SM, Palaniappan S. An approach for cloud Linux server security and management. *Annals of the Romanian Society for Cell Biology*. 2021. Vol. 25(3). pp. 2453–2459.
3. Oleksiuk V, Spirin O. The experience of using cloud labs in teaching Linux operating system. *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications*. 2021. pp. 281–291.
4. ThiBac D, Minh NH. Design of network security storage system based on under cloud computing technology. *Computers and Electrical Engineering*. 2022. Vol. 103.pp.108334.
5. Wertenbroek R, Thoma Y, Dassatti A. A portable linux-based firmware for NVMe computational storage devices. *ACM Transactions on Storage*. 2025. Vol. 21(2). pp. 1–36.
6. Zhang X, Zhu F, Li S, Wang K, Xu W, Xu D. Optimizing performance for open-channel SSDs in cloud storage system. *2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 2021. pp. 902–911.
7. Singh SK. Linux yourself : concept and programming. *Chapman and Hall /CRC*. 2021.
8. Ward B. How Linux works : What every superuser should know. *No Starch Press*. 2021.
9. Kiran Kumar M, Kranthi Kumar S, Kalpana E, Srikanth D, Saikumar K. A novel implementation of Linux based android platform for client and server. *A Fusion of Artificial Intelligence and Internet of Things for Emerging Cyber Systems*. 2021. pp. 151–170.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВПКІ. 200024.21.01.01 ПЗ | Арк.<br>64 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

10. Fox R. Linux with operating system concepts. *Chapman and Hall /CRC*. 2021.
11. Raza A, Unger T, Boyd M, Munson EB, Sohal P, Drepper U, Krieger O. Unikernel linux (ukl). *Proceedings of the Eighteenth European Conference on Computer Systems*. 2023. pp. 590–605.
12. Ali M, Bilal K, Khan SU, Veeravalli B, Li K, Zomaya AY. DROPS: Division and Replication of Data in Cloud for Optimal Performance and Security. *IEEE Transactions on Cloud Computing*. 2018. Vol. 6(2, SI). pp. 303–315.
13. Alkhanak EN, Lee SP, Khan SUR. Cost-aware challenges for workflow scheduling approaches in cloud computing environments : Taxonomy and opportunities. *Future Generation Computer Systems*. 2015. Vol. 50. pp.3–21.
14. Alteen N, Fisher J, Gerena C, Gruver W, Jalis A, Osman H, Pagan M, Patlolla S, Roth M. AWS Certified Developer Official Study Guide : Associate (DVA-C01) Exam. *John Wiley & Sons*. 2019.
15. Balaji S, Krishnan MN, Vajha M, Ramkumar V, Sasidharan B, Kumar PV. Erasure coding for distributed storage : an overview. *Science China Information Sciences*. 2018. Vol. 61(10). pp. 1–45.
16. Barhate SM, Dhore M. Hybrid Cloud : A Solution this Cloud Interoperability. *Proceedings of the Second International Conference on Inventive Communication and Computational Technologies (ICICCT 2018)*. 2018. pp. 1242–1247.
17. Barhate S, Dhore M. Hybrid Cloud : A Cost Optimized Solution That Cloud Interoperability. *Proceedings of the International Conference on Innovative Trends in Information Technology (ICITIIT 2020)*. 2020. pp. 1–5.
18. Barika M, Garg S, Zomaya AY, Wang L, Moorsel AV, Ranjan R. Orchestrating Big Data Analysis Workflows in the Cloud : Research Challenges, Surveys, and Future Directions. *ACM Computing Surveys*. 2019. Vol. 52(5). pp. 14-21.
19. Belgacem A. Dynamic resource allocation in cloud computing : analysis and taxonomies. *Computing*. 2022. Vol. 104(3). pp. 681–710.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>65 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

20. Bessani A, Correia M, Quaresma B, André F, Sousa P. DepSky : Dependable and Secure Storage in a Cloud-of-Clouds. *ACM Transactions on Storage*. 2013. Vol. 9(4). pp. 88-98.

21. Bhamare D, Samaka M, Erbad A, Jain R, Gupta L, Chan HA. Optimal virtual network function placement in multi-cloud service function chaining architecture. *Computer Communications*. 2017. Vol. 102. pp.1–16.

22. Bhatt R, Datta R. Cost Modeling and Studies with Different Deployment Strategies for Wireless Multimedia Sensor Network Over Flat and Elevated Terrains. *International Journal of Wireless Information Networks*. 2014. Vol. 21. pp. 15–31.

23. Blass A, Gurevich Y. Ordinary interactive small step algorithms, I. *ACM Transactions on Computational Logic*. 2006. Vol. 7(2). pp. 363–419.

24. Bokhari SMA, Theel O. A flexible hybrid approach this data replication in distributed systems. *Proceedings of the Computing Conference (SAI 2020)*. 2020. Vol. 1228. pp. 196–207.

25. Brewer EA. Towards robust distributed systems. *Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing (PODC 2000)*. 2000. Vol. -. pp. 7.

26. Buckley S. Spread Networks tightens up latency along Chicago this New York route. *FierceTelecom*. 2023. URL: <https://www.fiercetelecom.com/telecom/spread-networks-tightens-up-latency-along-chicago-to-new-york-route>.

27. Bulent E, Tarek A. Analysis Of Interoperability In Cloud Computing. *Proceedings of the 2019 5th International Conference on Computer and Technology Applications (ICCTA 2019)*. 2019. pp. 189–192.

28. Campêlo RA, Casanova MA, Guedes DO, Laender AH. A brief survey on replica consistency in cloud environments. *Journal of Internet Services and Applications*. 2020. Vol. 11(1). pp. 1–13.

29. Corodescu A.-A., Nikolov N., Khan AQ, Soyly A., Matskin M., Payberah AH, Roman D. Big Data Workflows : Locality-Aware Orchestration Using Software Containers. *Sensors*. 2021. Vol. 21(24). pp. 8212.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВПКІ. 200024.21.01.01 ПЗ | Арк.<br>66 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

30. Daniel EJ, White CM, Teague KA An interarrival delay jitter model using multistructure network delay characteristics for packet networks. *Proceedings of the Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*. 2003. Vol. 2.pp.1738–1742.

31. Dillon T., Wu C., Chang E. Cloud Computing : Issues and Challenges. *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA 2010)*. 2010. pp. 27–33.

32. Dong X., Zhao L., Zhou X., Li K., Guo D., Qiu T. An Online Cost-Efficient Transmission Scheme for Information-Agnostic Traffic in Inter-Datacenter Networks. *IEEE Transactions on Cloud Computing*. 2019. Vol. 10(1). pp. 202–215.

33. Dowell S, Barreto A, Michael JB, Shing M-T. Cloud this cloud interoperability. *Proceedings of the 6th International Conference on System of Systems Engineering ( SoSE 2011)*. 2011. Pp. 258–263.

34. Edwin EB, Umamaheswari P., Thanka MR An efficient and improved multi-objective optimized replication management with dynamic and cost aware strategies in cloud computing data center. *Cluster Computing*. 2019. Vol. 22(5). pp. 11119–11128.

35. Endo PT, Rodrigues M, Gonçalves GE, Kelner J, Sadok DH, Curescu C. High availability in clouds : systematic review and research challenges. *Journal of Cloud Computing*. 2016. Vol. 5(1). pp. 1–15.

36. Erradi A., Mansouri Y. Online cost optimization algorithms for tiered cloud storage services. *Journal of Systems and Software*. 2020. Vol. 160.pp.110457.

37. Georgios C., Evangelia F., Christos M., Maria N. Exploring cost-effective bundling in a multi-cloud environment. *Simulation Modeling Practice and Theory*. 2021. Vol. 111. pp. 102338.

38. Gessert F., Wingerath W., Friedrich S., Ritter N. NoSQL Database Systems : A Survey and Decision Guidance. *Computer Science – Research and Development*. 2017. Vol. 32. pp. 353–365.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>67 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

39. Ghoreishi SE, Karamshuk D., Friderikos V., Sastry N., Dohler M., Aghvami AH A Cost-Driven Approach this Caching - as -a- Service in Cloud-Based 5G Mobile Networks. *IEEE Transactions on Mobile Computing*. 2019. Vol. 19(5). pp. 997–1009.

40. Gill SS, Buyya R. A Taxonomy and Future Directions for Sustainable Cloud Computing : 360 Degree View. *ACM Computing Surveys*. 2018. Vol. 51(5). pp. 1–33.

41. Girault A., Gössler G., Guerraoui R., Hamza J., Serebinschi D.-A. Monotonic Prefix Consistency in Distributed Systems. *Formal Techniques for Distributed Objects, Components, and Systems*. 2018. Vol. 10854. pp. 41–57.

42. Gómez D., Romero J., López P., Vázquez J., Cappo C., Pinto D., Villalba C. Cloud architecture for electronic health record systems interoperability. *Technology and Health Care*. 2022. Vol. 30(3). pp. 551–564.

43. Gopinath C., Kiruthika C. A Server Side Encryption for Cloud Storage with Federation Sharing in Hybrid Cloud Environment. *Proceedings of the 2017 International Conference on Technical Advancements in Computers and Communications (ICTACC 2017)*. 2017. pp. 128–131.

44. Hale JS, Li L, Richardson CN, Wells GN Containers for Portable, Productive, and Performative Scientific Computing. *Computing in Science & Engineering*. 2017. Vol. 19(6). pp. 40–50.

45. Han J, Kim S, Kim T, Han D. Toward scaling hardware security module for emerging cloud services. *Proceedings of the 4th Workshop on System Software for Trusted Execution ( SysTEX 2019)*. 2019. Vol. -. pp. 1–6.

46. Hayes S. Analyzing network performance management. *IEEE Communications Magazine*. 1993. Vol. 31(5). pp. 52–58.

47. Höfer C., Karagiannis G. Cloud computing services : taxonomy and comparison. *Journal of Internet Services and Applications*. 2011. Vol. 2. pp.81–94.

48. Hong J, Dreibholz T, Schenkel JA, Hu JA An overview of multi-cloud computing. *Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA 2019)*. 2019. Vol. 927. pp. 1055–1068.

|     |      |          |        |      |                           |            |
|-----|------|----------|--------|------|---------------------------|------------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк.<br>68 |
| Зм. | Арк. | № докум. | Підпис | Дата |                           |            |

49. Hossain K., Roy S. A Data Compression and Storage Optimization Framework for IoT Sensor Data in Cloud Storage. *Proceedings of the 21st International Conference of Computer and Information Technology (ICCIT 2018)*. 2018. pp. 1–6.

50. Ilieva G., Yankova T., Hadjieva V., Doneva R., Totkov G. Cloud service selection as a fuzzy multi-criteria problem. *TEM Journal*. 2020. Vol. 9(2). pp. 484.

51. Irie R., Murata S., Hsu YF, Matsuoka M. A Novel Automated Tiered Storage Architecture for Achieving Both Cost Saving and QoE. *Proceedings of the 8th International Symposium on Cloud and Service Computing (SC2)*. 2018. pp. 32–40.

52. Jayakumar S., Prakash S., Akki C. Design of a Novel Architecture for Cost - Effectiveness Cloud-based Content Delivery Network. *International Journal of Advanced Computer Science and Applications*. 2021. Vol. 12(10). pp. 84-98.

53. Jiang F., Cheng Y., Hui Z., Yan R. Modeling and Analyzing for Data Durability Towards Cloud Storage Services. *Proceedings of the 20th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2020)*. 2020. Vol. 12453. pp. 646–661.

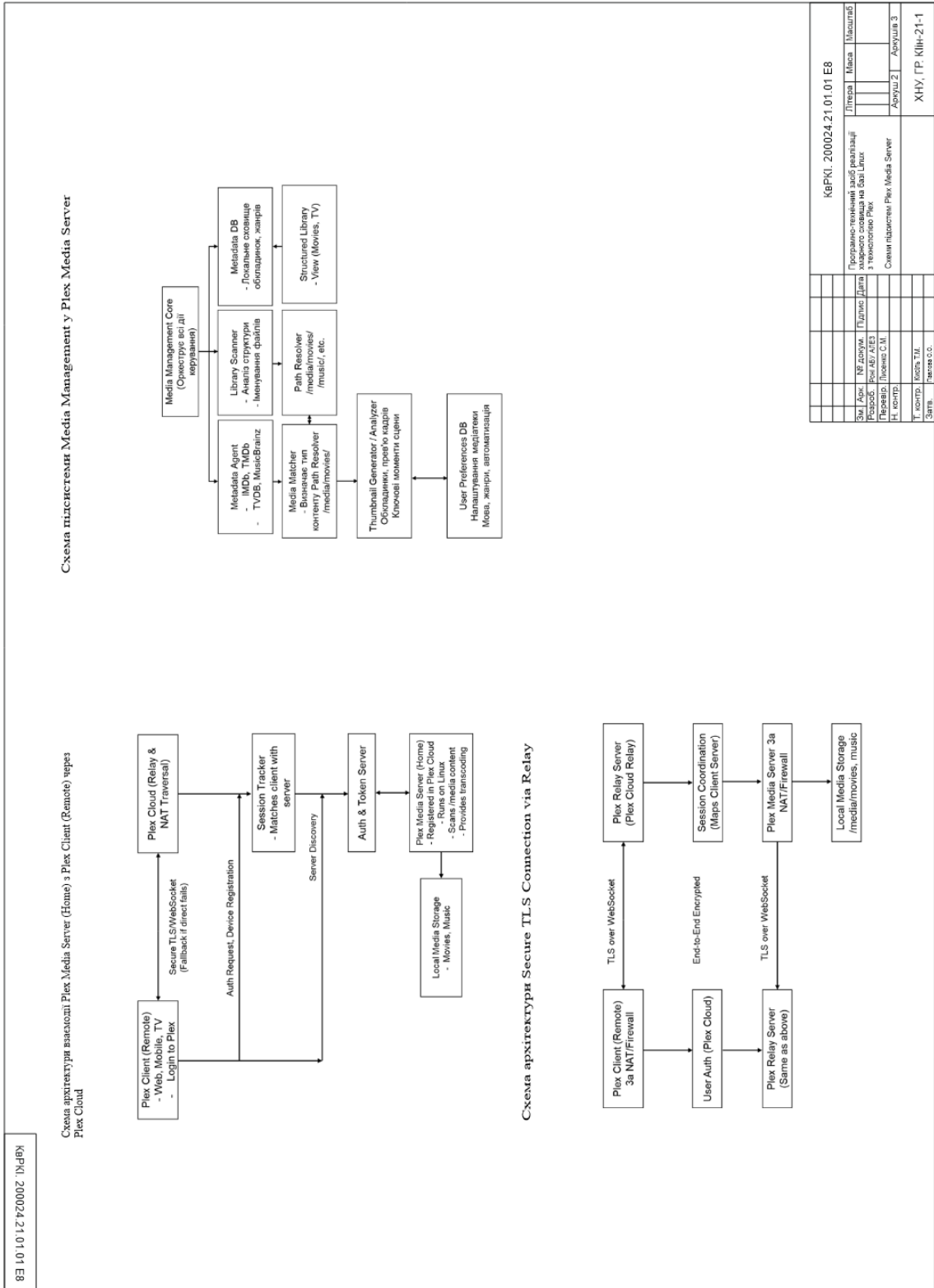
|     |      |          |        |      |                           |      |
|-----|------|----------|--------|------|---------------------------|------|
|     |      |          |        |      | КВРКІ. 200024.21.01.01 ПЗ | Арк. |
| Зм. | Арк. | № докум. | Підпис | Дата |                           | 69   |



# Appendix B

( required )

## COPY OF THE DRAWING "PLEX MEDIA SERVER SUBSYSTEM DIAGRAM"



# Appendix C

( required )

## COPY OF THE DRAWING " PLEX MEDIA SERVER WORKING ALGORITHMS "

