

Важливим доповненням до технічних заходів є організаційні процедури, зокрема регулярний аудит безпеки, навчання персоналу та розробка внутрішніх політик роботи з персональними даними. Детальне логування операцій без фіксації чутливого контенту та інтеграція з SIEM-системами дозволяють своєчасно виявляти аномальну активність і реагувати на потенційні інциденти безпеки.

Висновки. Проведене дослідження показало, що інтеграція Telegram Bot API у CRM-системи управління житловими комплексами супроводжується підвищеними ризиками для безпеки персональних даних мешканців. Водночас використання сторонніх API не обов'язково означає компроміс у рівні захисту за умови впровадження багаторівневої моделі безпеки, яка поєднує технічні та організаційні заходи. Запропонований підхід може бути адаптований для інших месенджерів і сторонніх сервісів, що робить результати дослідження актуальними для широкого кола організацій. Перспективи подальших досліджень пов'язані з автоматизацією аудиту безпеки API-інтеграцій та застосуванням сучасних криптографічних технологій для підвищення рівня конфіденційності.

Список використаних джерел

1. European Parliament and Council. (2016). General Data Protection Regulation (EU) 2016/679. <https://eur-lex.europa.eu/>.
2. OWASP Foundation. (2023). OWASP API Security Top 10. <https://owasp.org/>.
3. Schneier, B. (2015). Secrets and lies: Digital security in a networked world. Wiley.
4. Верховна Рада України. (2010). Закон України «Про захист персональних даних» № 2297-VI. <https://zakon.rada.gov.ua/>.

DOI 10.70286/EOSS-19.01.2026.003.84-87

СТВОРЕННЯ РОЗПОДІЛЕНИХ СИСТЕМ НА ОСНОВІ ТЕХНОЛОГІЇ REMOTE METHOD INVOCATION

Рибак Ольга

к.т.н., доцент

ORCID: 0000-0002-0250-3037

Кафедра інформаційних технологій проектування та дизайну
Національний університет «Одеська політехніка», Україна

Розподілені системи використовуються для роботи з великими обсягами даних, віддаленого доступу до ресурсів та побудови масштабованих сервісів. Такі системи дозволяють кільком комп'ютерам працювати як одна обчислювальна структура, користувачі якої залишаються пов'язаними, а процеси

взаємоузгодженими, незважаючи на те, що ресурси знаходяться у різних частинах мережі. Саме тому важливою складовою розподілених систем є технології, які приховують фізичне розташування даних та забезпечують зручний доступ до них.

Для спрощення взаємодії між вузлами розподілених систем та покращення їхньої архітектури використовується проміжне програмне забезпечення (Middleware). Його задача полягає у тому, щоб приховати гетерогенність мережі та надати зручний інтерфейс програмісту. Однією з перших і найбільш поширених технологій цього класу для платформи Java стала RMI (Remote Method Invocation).

Технологія Remote Method Invocation є потужним інструментом для організації розподілених систем, що ґрунтується на виклику методу віддаленого об'єкту. Вона дозволяє розробникам створювати застосунки, в яких об'єкти можуть взаємодіяти через мережу, навіть якщо вони знаходяться на різних комп'ютерах або є частиною різних процесів [1-3]. Основні принципи технології RMI полягають у запиті методів об'єктів на віддаленому комп'ютері, передачі параметрів та отриманні результатів виконання.

Програмний інтерфейс RMI надає користувачам засоби для забезпечення функціонування розподіленої системи, включаючи віддалений інтерфейс, віддалений об'єкт, реєстр віддалених об'єктів та підтримку процесів маршалінгу та демаршалінгу оброблюваних даних. Віддалені інтерфейси (Remote Interface) визначають набір методів, які можуть бути викликані віддалено, і повинні оголошувати виняток `java.rmi.RemoteException` як базовий клас потенційних помилок зв'язку, таких як збої з'єднання, проблеми обробки сервером та відмова виклику заданих методів [4]. Віддалений інтерфейс слід проектувати таким чином, щоб він відображав функціональні можливості, які будуть доступні віддаленим клієнтам.

Створення віддалених об'єктів (Remote Object) передбачає визначення класу, що включає декларацію методів, оголошених у віддаленому інтерфейсі, та наслідує клас `java.rmi.server.UnicastRemoteObject`. Віддалені об'єкти повинні бути зареєстровані, щоб їх можна було знайти та викликати. Для цього використовується реєстр віддалених об'єктів (Remote Object Registry) – централізоване сховище, яке дозволяє зберігати віддалені об'єкти та надавати доступ до них іншим клієнтам. Віддалені об'єкти реєструються з використанням імені або ідентифікатора, за допомогою якого їх можна відшукати. Клієнти можуть викликати методи віддалених об'єктів, передаючи параметри та очікуючи результату. Це здійснюється за допомогою створення віддаленого посилання (Remote Reference), яке включає інформацію про віддалений об'єкт та спосіб зв'язку з ним.

Для перетворення параметрів методів та результатів з формату Java в формат, який може бути переданий через мережу, в RMI передбачено реалізацію процесів маршалінгу та демаршалінгу (Marshaling/Unmarshaling), причому

маршалінг виконується на стороні відправника, а демаршалінг – на стороні отримувача [5]. При використанні RMI важливо правильно обробляти винятки, що можуть з'явитись під час виклику віддаленого методу. Для цього у віддаленому інтерфейсі має бути оголошено RemoteException, а віддалений об'єкт повинен передбачати обробку цих винятків та повертати результати клієнту. Крім того, безпека роботи системи також є важливим аспектом розподілених систем, і RMI надає можливості для забезпечення аутентифікації, авторизації та шифрування даних.

Виклик віддаленого методу у розподіленій системі на основі RMI відбувається за схемою, зображеною на рис.1.



Рисунок 1. Процес виклику віддаленого методу

Спочатку сервер створює екземпляр віддаленого об'єкта і реєструє його в службі імен. Для цього використовується операція bind або rebind, яка пов'язує унікальне ім'я об'єкта з посиланням на нього. Коли віддалений клієнт звертається до реєстру із запитом lookup, вказуючи ім'я потрібного сервісу, відбувається його пошук. Якщо потрібний об'єкт знайдено, реєстр повертає клієнту об'єкт-заглушку Stub – посередника, що представлятиме віддалений сервіс на стороні клієнта. Якщо запитуваний об'єкт у службі імен відсутній, RMI дозволяє динамічно завантажити необхідний файл із зовнішнього веб-сервера. Клієнт виконує запит get і отримує необхідний код. Безпосередній виклик віддаленого методу здійснюється на основі серії параметрів при передачі їх через мережу до віддаленого об'єкту. При цьому відбувається обробка запиту і повернення результатів виконання назад клієнту.

З огляду на особливості функціонування, технологія RMI виявляється досить гнучкою та масштабованою. Вона забезпечує надійну передачу даних через мережу та дозволяє зручно керувати віддаленими викликами. Прикладами

застосування технології RMI є клієнт-серверні системи, веб-сервіси, розподілені бази даних, мережеві ігри та додатки, а також створення систем Internet of Things, де різні пристрої здатні взаємодіяти та обмінюватися даними через мережу [6].

Список використаних джерел

1. Graba J. (2013). Remote Method Invocation (RMI). In: An Introduction to Network Programming with Java. Springer, London, – pp. 129-149. Doi: https://doi.org/10.1007/978-1-4471-5254-5_5
2. Рибак О.В. (2024) Застосування технології Java Message Service у розподілених системах. Матеріали тез доповідей XIV Міжнародної науково-практичної конференції у 2 т. / НУ «Чернігівська політехніка», 2024. – Т. 2. – с. 301-303. URI: <http://ir.stu.cn.ua/handle/123456789/32159>
3. Müller-Hofmann F., Hiller M., Wanner G. (2015). Remote Method Invocation. In: Programmierung von verteilten Systemen und Webanwendungen mit Java EE. Springer Vieweg, Wiesbaden. – pp. 57–80. Doi: https://doi.org/10.1007/978-3-658-10512-9_3
4. Le M., Clyde S., Kwon Y. (2019). Enabling multi-hop remote method invocation in device-to-device networks. Human-centric Computing and Information Sciences. Vol. 9 (20) – pp. 22. Doi: <https://doi.org/10.1186/s13673-019-0182-9>
5. Беседіна С.В. (2020) Особливості використання технології RMI, DCOM та CORBA для побудови розподілених систем. Jurnal-AITU, Т. 2 (№ 51). – р. 13.
6. Кожевніков Г.К., Маркевич М.І., Матяш О.Ю. (2024). Технологія взаємодії між вузлами граничного шару інтернету речей. Системи управління, навігації та зв'язку. Збірник наукових праць, 4(78). – 92-95. Doi: 10.26906/SUNZ.2024.4.092

ОПТИМІЗАЦІЯ ЛЕГКОВАГОВИХ НЕЙРОМЕРЕЖ ДЛЯ ГОЛОСОВОЇ ВЕРИФІКАЦІЇ НА МОБІЛЬНИХ ПРИСТРОЯХ

Руда Христина Степанівна

д-р філос., доцент

Кос Ігор Іванович

здобувач вищої освіти

Кафедра захисту інформації

Національний університет «Львівська політехніка», Україна

Голосова верифікація є одним з багатьох ефективних біометричних механізмів автентифікації, які використовуються в сучасному світі. Завдяки природності взаємодії та відсутності потреби в додатковому обладнанні цей спосіб є особливо привабливим для користувачів та компаній. Перехід до верифікації на кінцевому пристрої дозволяє зменшити затримку, підвищити надійність роботи без мережі та покращити конфіденційність, однак в той же час