

І. В. ЗАМРІЙ^{1,2}, д-р техн. наук, професор;
ORCID: 0000-0001-5681-1871

А. О. ЛИТВИНЕНКО², аспірант,
ORCID 0009-0001-2001-0148

¹Київський національний університет імені Тараса Шевченка

²Державний університет інформаційно-комунікаційних технологій, Київ

ВИКОРИСТАННЯ ВЕБСЕРВІСНОЇ АРХІТЕКТУРИ ДЛЯ ІНФОРМАЦІЙНИХ СИСТЕМ З ПРОГРАМНО-ТЕХНІЧНИМИ КОМПЛЕКСАМИ, ЗДАТНИМИ ДО САМОНАЛАШТУВАННЯ

У статті розглянуто особливості використання вебсервісної архітектури для побудови інформаційних систем спеціального призначення, що базуються на програмно-технічних комплексах, здатних до самоналаштування. Проаналізовано еволюцію підходів до побудови інформаційних систем у контексті розвитку сучасних військових концепцій, зокрема Network-Centric Warfare, що реалізується в країнах НАТО. Визначено основні відмінності між сервісно-орієнтованою архітектурою та вебсервісною архітектурою, зокрема перехід до використання стандартизованих вебпротоколів, відкритих прикладних програмних інтерфейсів та слабко зв'язаних сервісів. Показано, що вебсервісна архітектура забезпечує більш високий рівень інтероперабельності, масштабованості та гнучкості інформаційних систем. Запропоновано підхід до побудови еталонної архітектури програмно-технічних комплексів, що включає комунікаційний, інформаційний, прикладний та презентаційний рівні. Архітектура дозволила уніфікувати технології, які використовуються при створенні програмно-технічних комплексів, які враховують необхідність використання самокерованих адаптованих мереж та платформу інформаційного обміну. Обґрунтовано необхідність створення самоналаштовуваних програмно-технічних комплексів, здатних адаптуватися до змін умов функціонування та ролей користувачів. Такі комплекси допомагають автоматично змінювати конфігурацію залежно від умов експлуатації та забезпечувати стійкість до змін структури мережі та складу системи. Такий підхід дозволяє розширити функціональні можливості типових програмно-технічних комплексів, що дозволяє формувати різні автоматизовані робочі місця, які самоналаштовуються під конкретну роль користувача. Показано, що використання типових програмно-технічних комплексів дозволяє реалізувати принцип уніфікації та забезпечити динамічне формування автоматизованих робочих місць у межах єдиного інформаційного простору.

Ключові слова: вебсервісна архітектура, самоналаштування, програмно-технічний комплекс, мережецентричні системи, мікросервіси, інформаційні системи спеціального призначення.

Вступ

Застосування інформаційних технологій спонукало військово-політичне керівництво країн НАТО, насамперед США, до трансформації теоретичних і практичних засад розвитку збройних сил, а також до оцінювання нових тенденцій у сфері управління та організації взаємодії багатонаціональних угруповань під час проведення об'єднаних і коаліційних операцій.

У сучасних умовах саме процеси управління виходять на передній план, оскільки вони забезпечують інтеграцію військово-політичних, економічних, науково-технічних, військових і кризових факторів в єдину систему прийняття рішень. Особливого значення це набуває в умовах розширення НАТО та набуття його діяльністю глобального характеру.

Показовими у цьому контексті є військово-політичні події та операції в Югославії, Афганістані, Іраку, Сирії, Україні та Ірані, які демонструють зростаючу роль інтегрованих систем управління, інформаційної переваги та міждержавної координації дій.

Постановка проблеми

Для визначення передових технологій в НАТО існує план розвитку, в якому виділено чотири етапи розвитку інформаційних технологій. У цьому плані до 2025 року вирішувалась задача повної сумісності міжвідомчих інформаційних систем. Основними технологіями цього етапу є: архітектура інформаційних систем має бути сервісно-орієнтованою, програмне забезпечення повинно розроблятися як сервіси, а мережа має бути програмно-конфігурованою. Після 2025-го року архітектура інформаційних систем має стати вебсервісною, програмне забезпечення повинно мати властивості до самоналаштування, а мережі мають стати самокерованими та адаптованими до поточних умов експлуатації.

Мета цієї статті - провести дослідження щодо визначення основних відмінностей між сервісно-орієнтованою та вебсервісною архітектурою інформаційних систем спеціального призначення, а також, які задачі мають бути вирішені при розробці програмно-технічних комплексів, які мають властивості до самоналаштування.

Аналіз останніх досліджень і публікацій

Стрімкий розвиток інформаційних технологій, який зумовив формування нових тенденцій у сфері управління збройними силами, а також апробація новітніх видів озброєння, засобів розвідки, радіоелектронної боротьби, автоматизованих систем управління та зв'язку в умовах реальних бойових дій, обумовили необхідність переосмислення підходів до будівництва збройних сил. У таких умовах ефективність застосування військ визначається не лише характеристиками окремих зразків озброєння, а й рівнем інтеграції програмно-апаратних комплексів (ПАК) у єдиний інформаційно-управлінський простір [4] – [8], [10] – [12], [15], [17], [18].

Для вирішення вище зазначених задач в НАТО прийнята модель мережецентричного ведення бойових дій, що суттєво підвищило вимоги до інформаційно-комунікаційних систем [1], [3], [13], [14]. Значна увага приділяється стандартизації обміну даними, зокрема у межах стандартів НАТО, а також використанню сервісно-орієнтованої архітектури [2], [9], [10].

В останні роки архітектура інформаційних систем стає вебсервісною [16], а програмне забезпечення має властивості до самоналаштування [19].

Результати дослідження

При побудові сервісно-орієнтованої архітектури (Service-Oriented Architecture (SOA)) інформаційних систем спеціального призначення зазвичай використовують чотири рівні (рис. 1):

- 1) **Клієнта.** Рівень відповідає за встановлення інтерфейсу для користувача.
- 2) **Подання.** До цього рівня належать компоненти інтерфейсу користувача та процесів інтерфейсу користувача, які стосуються насамперед створення елементів інтерфейсу користувача та містять логіку, що забезпечує навігацію користувача в інтерфейсі.
- 3) **Бізнес-рівень.** Містить такі типи компонентів, як бізнес-потік (Business Workflow), бізнес-компонент (Business Component), бізнес-сутність (Business Entity) і службовий інтерфейс (Service Interface), до яких входять бізнес-процеси та правила, що утворюють ядро прикладної програми.
- 4) **Даних.** Містить компоненти логіки доступу до даних (Data Access logic) та службові агенти (Service Agent), до яких відносяться засоби, що відповідають за доступ до даних для спеціальних джерел даних та вебслужб.

У плані розвитку технологій НАТО після 2025-го року архітектура інформаційних систем спеціального призначення має стати вебсервісною.

Якщо перейти від класичної сервісно-орієнтованої архітектури до вебсервісної архітектури, то стоїть задача реалізувати SOA через вебтехнології - REST/SOAP.

Перехід від сервісно-орієнтованої архітектури до вебсервісної

У випадку вебсервісної архітектури отримуємо трирівневу модель (рис. 2), тому що клієнтський рівень переходить до рівня подання, який у вебсервісній архітектурі стає максимально відокремленим від логіки.

Наповнення рівнів стає більш «мережевим» і стандартизованим, при цьому її компоненти трансформуються з урахуванням використання вебсервісів, стандартів (HTTP, REST, SOAP, XML/JSON) і розподіленої взаємодії.

Розглянемо, як виглядатиме кожен рівень:

1) Рівень подання

У вебсервісній архітектурі цей рівень включає наступні клієнтські компоненти: веб-браузер (SPA, web apps); мобільні додатки; логіку інтерфейсу користувача.

Взаємодія з сервером відбувається через REST API (JSON) та SOAP (XML).

Основними ознаками рівня подання є наступне:

- інтерфейс користувача плюс логіка навігації (як і в SOA);
- не містить бізнес-логіки;
- використовує HTTP/HTTPS для виклику вебсервісів;
- може включати фронтенд-фреймворк (React, Angular тощо) та API Gateway (іноді як проміжний шар).

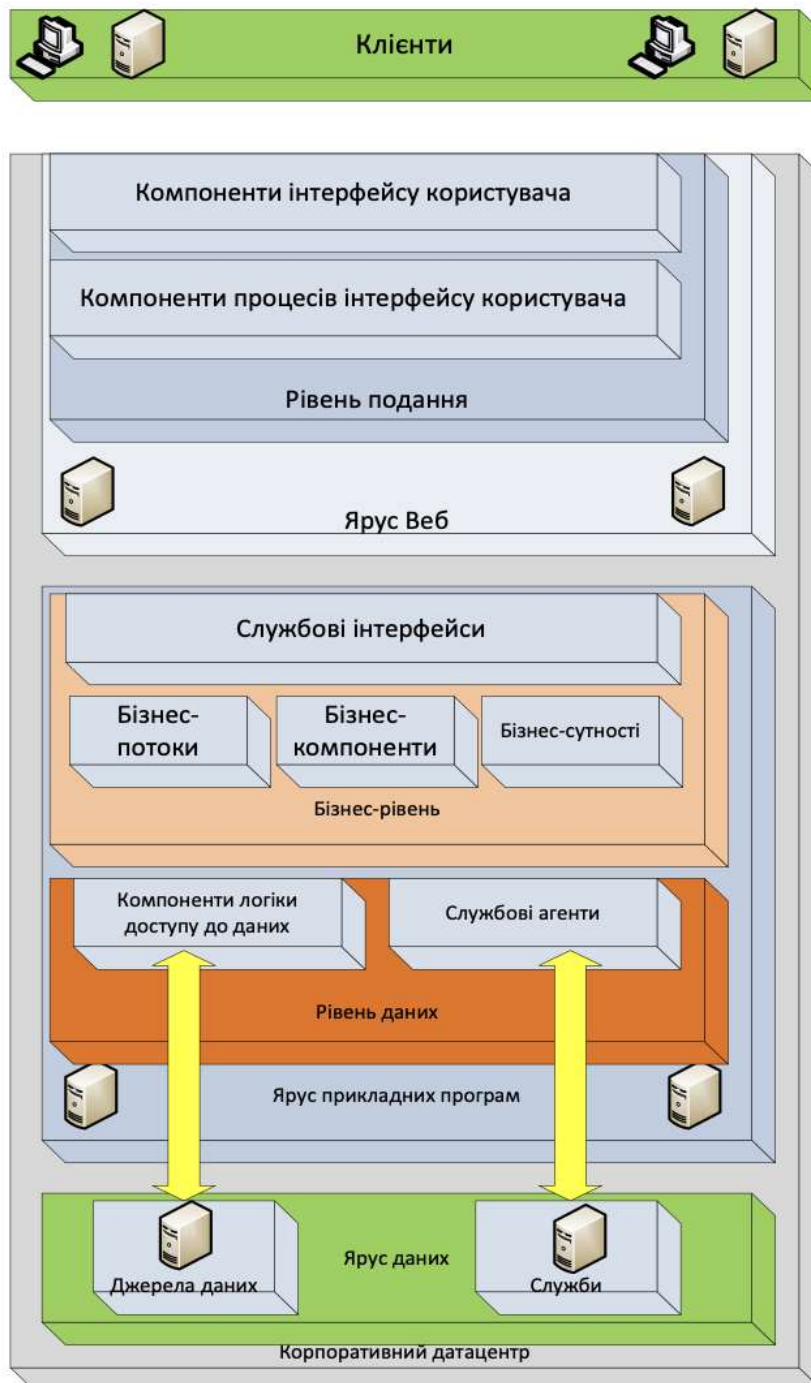


Рис. 1. Модель сервісно-орієнтованої архітектури

Під фронтенд-фреймворком розуміємо набір інструментів із попередньо написаного коду, компонентів та шаблонів дизайну, які більшість розробників використовують для створення візуальних та інтерактивних розділів програми або вебсайту. Він спрощує процес розробки та полегшує створення інтерфейсів, які є адаптивними, масштабованими та узгодженими.

Особливості **рівня подання**:

- взаємодія з системою тільки через HTTP/HTTPS;
- використання API (REST або SOAP);
- забезпечення навігації користувача в інтерфейсі;
- керування станом інтерфейсу користувача;
- підготовка запитів до сервісів.

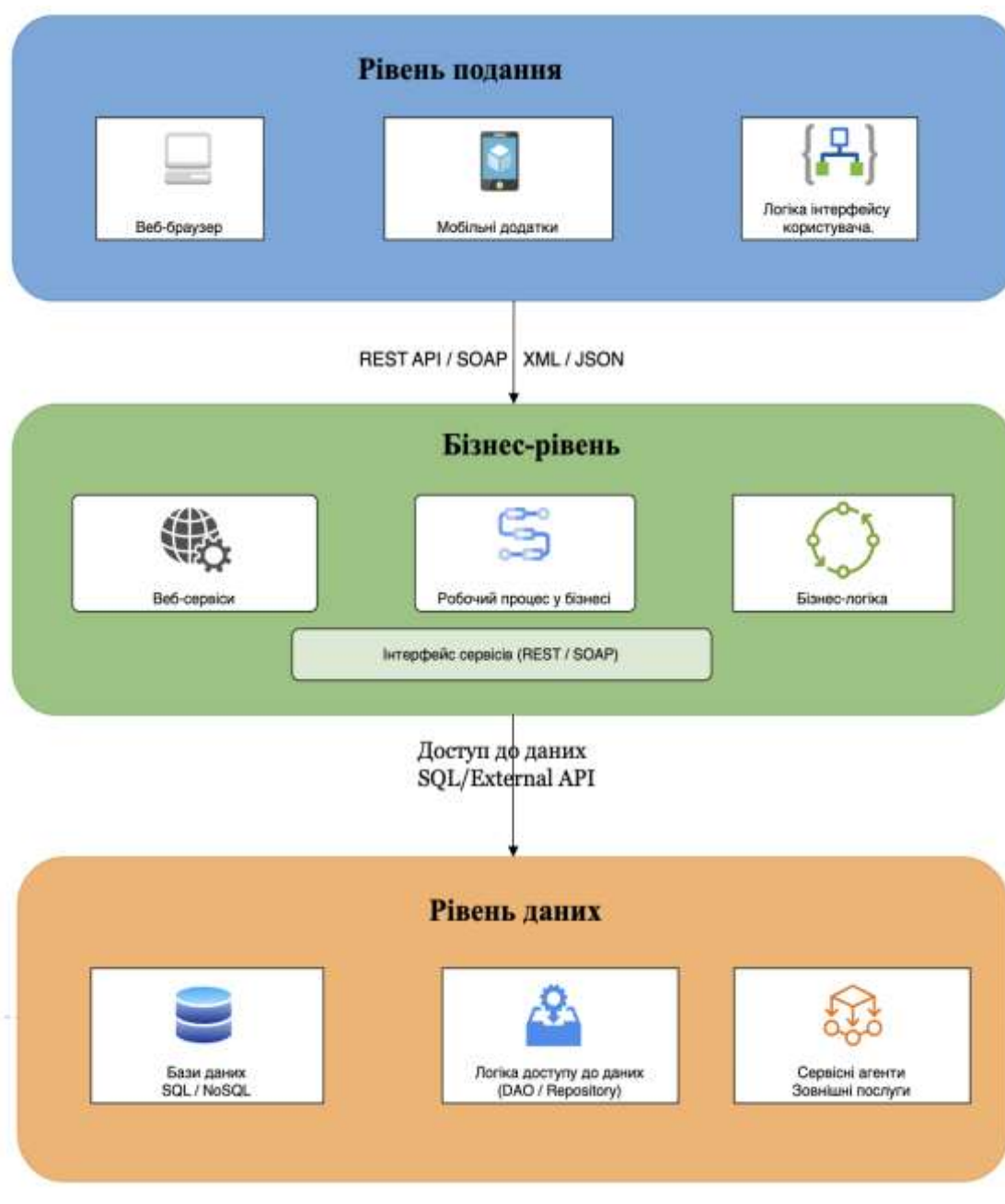


Рис. 2. Модель вебсервісної архітектури

Ключова зміна у вебсервісній архітектурі по відношенню до SOA - це те, що інтерфейс користувача більше не викликає «внутрішні компоненти», а працює через вебсервіси як зовнішні API.

2) Бізнес-рівень

У вебсервісній архітектурі цей рівень фактично реалізується як набір вебсервісів.

Бізнес-рівень включає наступні компоненти:

- **вебсервіси (REST/SOAP endpoints);**
- **робочий процес у бізнесі (Business Workflow)**, який відповідає за оркестрацію сервісів (наприклад, через BPM або оркестрація мікросервісів);
- **бізнес-логіка/компоненти відповідають за** реалізовані всередині сервіси.

Прикладний програмний інтерфейс (**API**) відповідає за чітко визначені специфікації (OpenAPI, WSDL). Бізнес-рівень містить REST API (JSON), SOAP-сервіси (XML), мікросервіси. Основними ознаками бізнес-рівня є:

- сервіси є незалежними, слабо зв'язаними, доступними через мережу;
- при цьому можливе використання мікросервісної архітектури, контейнерів;
- оркестровки.

Особливості **бізнес-рівня**:

- слабка зв'язаність;
- чіткі зв'язки (API contracts);
- можливість виклику ззовні (іншими системами).

Ключова зміна у вебсервісній архітектурі по відношенню до SOA - це те, що бізнес-логіка «виноситься» у мережеві сервіси, доступні через стандартизовані протоколи.

3) Рівень даних (Data Layer)

У вебсервісній архітектурі цей рівень часто стає більш розподіленим.

Рівень даних у вебсервісній архітектурі включає наступні компоненти:

- бази даних - SQL (PostgreSQL, MySQL); NoSQL (MongoDB, Redis);
- доступ до даних реалізується через - DAO (Data Access Objects); ORM (Hibernate, Entity Framework);

– **service Agents трансформуються у API-клієнти** до зовнішніх сервісів, інтеграційні адаптери (REST/SOAP clients);

– як виглядає рівень даних - бази даних (SQL/NoSQL); зовнішні API (third-party services); хмарні сховища.

Основними ознаками рівня даних є:

- кожен сервіс може мати **власну базу даних** (у мікросервісах);
- дані ізольовані між сервісами;
- інтеграція через API, а не прямий доступ до баз даних.

Особливості рівня даних полягає в тому, що доступ до даних може бути локальний або віддалений, або кожен сервіс має власну базу даних.

Ключова зміна у вебсервісній архітектурі по відношенню до SOA - це те, що дані можуть бути не тільки в базах даних, а й у інших вебсервісах.

Узагальнена відповідність компонентів класичної сервісно-орієнтованої архітектури та вебсервісної архітектури наведена у табл. 1.

Таблиця 1

Класична SOA	Вебсервісна архітектура
UI компоненти	Web / Mobile клієнти
Service Interface	REST / SOAP API
Business Components	Вебсервіси / мікросервіси
Workflow	Оркестрація сервісів
Data Access	DAO / ORM / API clients
Service Agents	Інтеграційні REST/SOAP клієнти

Підсумовуючи аналіз відмінностей між сервісно-орієнтованою та вебсервісною архітектурою інформаційних систем спеціального призначення можна виділити ключові відмінності вебсервісній архітектурі:

- 1) Всі взаємодії відбуваються через мережу;

2) Сервіси є явно опублікованими прикладними програмними інтерфейсами;

3) Інтеграція здійснюється через стандартні вебпротоколи.

Реалізація програмно-технічних комплексів, здатних до самоналаштування

У плані розвитку технологій НАТО після 2025-го року зафіксовано, що програмно-технічні комплекси (ПТК) для систем спеціального призначення повинні бути здатними до самоналаштування. Такий підхід дозволяє уніфікувати розробку ПТК і зробити більш гнучким використання автоматизованих робочих місць (АРМ) у системах спеціального призначення.

Головною відмінністю таких систем від традиційних автоматизованих систем управління військами є те, що в даному випадку реалізовано концепцію мережецентричного ведення бойових операцій (Network-Centric Warfare, (NCW)). NCW - це воєнна концепція, яка базується на поєднанні всіх учасників бойових дій (від окремого солдата та дрона до штабу командування) у єдину цифрову мережу.

Головна ідея полягає не в потужності окремої одиниці техніки (танка чи літака), а в інформаційній перевазі. Коли всі бачать спільну картину бою в реальному часі, це дозволяє діяти швидше та ефективніше за противника.

Це потребує створення єдиної інформаційної мережі, що об'єднує сенсори, органи управління та вогневі засоби в єдиний інформаційний простір, який оптимізує структури та форми подання даних, способи передачі та регламент обміну інформацією, має єдину технологію збирання, обробки та відображення інформації, єдину структуру класифікації й кодування, та єдиний адресний простір.

Еталонна архітектура ПТК, здатних до самоналаштування

При створенні комплексу основну увагу приділяли проектуванню архітектури ПТК, здатних до самоналаштування. Для уніфікації технологій, які використовують ПТК, було розроблено еталонну архітектуру ПТК, яка враховує необхідність використання самокерованих адаптованих мереж та платформу інформаційного обміну (рис. 3).

Суть архітектури полягає в тому, що для кожного рівня інформаційної системи виділяються певні технології, які використовуються при проектуванні ПТК.

В архітектурі виділяємо чотири рівні:

1) **комунікаційний рівень** – надає базові функції і сервіси зв'язку, що забезпечуються IP-центричною мережею;

2) **інформаційний рівень** - надає послуги, які підтримують управління та обмін інформації між додатками в мережі і зовнішнім середовищем;

3) **рівень додатків** - реалізує функціональну обробку спеціалізованих завдань;

4) **рівень презентації** - реалізує інтерфейси користувача.

На кожному рівні використовуються певні сервіси, які використовують різні технології:

1) **комунікаційний рівень** – надає сервіси Інтернет мережі (BGP, OSPF, IPv4, IPv6, Ipvsec) та сервіси Інтернет транспорту (TCP, UDP);

2) **інформаційний рівень** - інформаційна інфраструктура (OS Svc, Infrastructure Svc, SQLite, MongoDB) та сервіси для передачі інформації (SMTP, HTTP/S, MQTT);

3) **рівень додатків** – додатки домену (планування та управління; BMS core; GIC; DSS core; інтеграційні сервіси);

4) **рівень презентації** - додатки презентаційного рівня (C++, Python, JavaScript+CSS)) та адаптер презентаційного рівня (HTML, SSE, REST).

В інформаційних системах спеціального призначення типові рішення визначаються щодо: функціональних і базових сервісів, які відповідають Таксономії СЗ НАТО (Consultation, Command and Control) [8]; складу технологічних засобів; типових (незалежно від об'єкту автоматизації) прикладних застосувань сервісів; типових комплексів засобів автоматизації КЗА (інтерфейсу користувачів); типових форм електронних документів; єдиної бази даних.

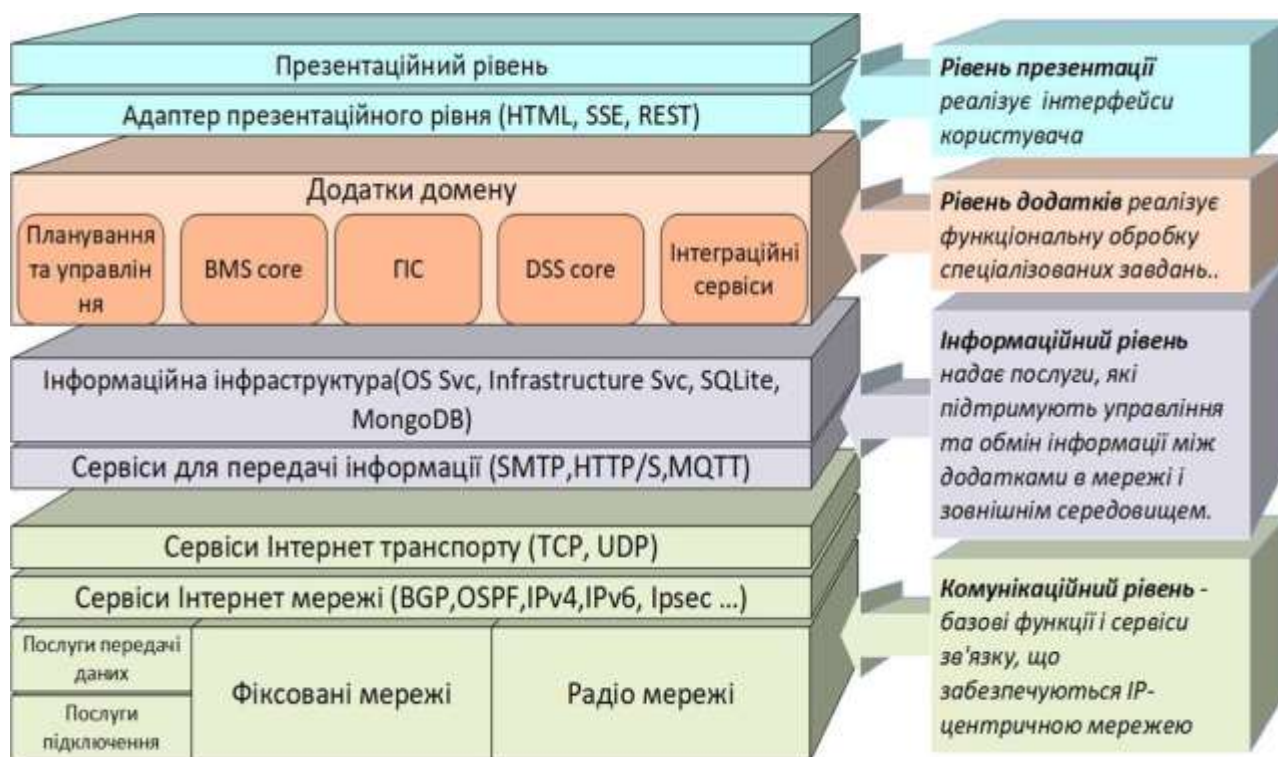


Рис. 3. Еталонна архітектура ПТК

Інформаційні системи спеціального призначення функціонально складаються з типових програмно-технічних комплексів (ПТК), що забезпечує самоналаштування ПЗ. Залежно від специфіки застосування ПТК можуть бути різних видів, причому їхній інтерфейс принципово відрізняється один від одного. Так ПТК, призначений для роботи в штабі чи на пункті управління, реалізовано так, щоб можна було використовувати можливості офісної роботи та здійснювати заздалегідь невідомі управлінські процеси. ПТК, призначений для роботи в польових умовах, використовує лише формалізовані управлінські процеси, а оператор має можливість швидко реагувати завдяки формалізованим повідомленням. ПТК, призначений для роботи на передовій, має специфічні особливості й передбачає захист від можливої втрати.

Функціональні можливості типових ПТК забезпечують формування різних АРМ військовослужбовців. Тому для забезпечення кількох сотень АРМ для військової частини можна використовувати 5 типових ПТК, що дозволяє у кожному сегменті мати АРМ, які самоналаштовуються під конкретну роль користувача.

Висновки та перспективи подальших досліджень

У результаті проведеного дослідження встановлено, що для інформаційних систем спеціального призначення перехід від сервісно-орієнтованої архітектури до вебсервісної є закономірним етапом розвитку інформаційних систем в умовах зростання вимог до інтеграції, масштабованості та адаптивності.

Визначено, що ключовими відмінностями вебсервісної архітектури є:

- використання стандартизованих протоколів взаємодії;
- реалізація сервісів у вигляді відкритих API;
- забезпечення слабкої зв'язаності компонентів;
- повна мережева взаємодія між елементами системи.

Обґрунтовано, що сучасні інформаційні системи спеціального призначення, побудовані відповідно до концепції Network-Centric Warfare, повинні забезпечувати інтеграцію різномірних компонентів у єдину інформаційну мережу, що створює передумови для досягнення інформаційної переваги.

Показано, що впровадження вебсервісної архітектури створює технологічну основу для реалізації програмно-технічних комплексів, здатних до самоналаштування.

Такі комплекси дозволяють:

- адаптувати функціональність системи до ролі користувача;
- автоматично змінювати конфігурацію залежно від умов експлуатації;
- забезпечувати стійкість до змін структури мережі та складу системи.

Було розроблено еталонну архітектуру програмно-технічних комплексів, яка забезпечує уніфікацію технологічних рішень та дозволяє формувати широкий спектр автоматизованих робочих місць на основі обмеженої кількості типових компонентів.

Використання еталонної архітектури при побудові інформаційних систем спеціального призначення дозволило ефективно використовувати програмно-технічні комплекси, які за рахунок функції самоналаштування дозволяють ефективно задіювати автоматизовані робочі місця в залежності від ситуації, яка виникає в певний період використання завдання.

Отримані результати можуть бути використані при розробці перспективних інформаційних систем спеціального призначення та модернізації існуючих автоматизованих систем управління.

Внесок авторів

Ірина ЗАМРІЙ – концептуалізація, архітектури інформаційних систем; Аліна ЛИТВИНЕНКО – проектування програмно-технічних комплексів, здатних до самоналаштування, еталонна архітектура програмно-технічних комплексів.

Декларація про штучний інтелект

Штучний інтелект не застосовувався.

Конфлікт інтересів

Автори заявляють про відсутність конфлікту інтересів та підтверджують, що під час підготовки цієї роботи не існувало жодних комерційних, фінансових чи інших взаємовідносин, які могли б бути розцінені як такі, що здатні вплинути на результати дослідження або їх інтерпретацію. Робота виконана відповідно до принципів академічної доброчесності, етичних норм проведення наукових досліджень та вимог редакційної політики щодо запобігання конфлікту інтересів.

Список використаної літератури

1. *Alberts, D. S. (2019). The Future of Network-Centric Warfare: From Information Superiority to Decision Advantage. 2019 IEEE Military Communications Conference (MILCOM), 12-18. <https://doi.org/10.1109/MILCOM47813.2019.9020910>*
2. *Bass, L., Clements, P., & Kazman, R. (2013). Software Architecture in Practice (3rd ed.). Addison-Wesley.*
3. *Department of Defense. (2003). DoD Net-Centric Data Strategy. Washington: DoD. <https://apps.dtic.mil/sti/tr/pdf/ADA482100.pdf>*
4. *Dovgiy, S., & Kopyika, O. (2023). Standard Model of System Architecture of Enterprise IT Infrastructure. Lecture Notes in Networks and Systems, 548, 181–201. https://doi.org/10.1007/978-3-031-16368-5_9*
5. *Dovgiy, S., Kopyika, O., Kalinin, V., & Kozlov, O. (2025). Інтеграція інформаційно-комунікаційних систем з використанням інтелектуальної платформи. НЦ «МАН України». <https://doi.org/10.1016/j.matdes.2025.114301>*
6. *Dovgiy, S., Kopyika, O., Kozlov, O., & Lytvynenko, A. (2023). Features of Message Transport Service in Automated Special-Purpose Systems. Lecture Notes in Networks and Systems, 809, 119–132. https://doi.org/10.1007/978-3-031-46880-3_8*
7. *Довгий, С. О., & Копійка, О. В. (2023). IT-інфраструктура як базова складова цифрової трансформації. Юстон.*

8. Довгий, С. О., Копійка, О. В., Козлов, О. С., & Литвиненко, А. О. (2023). Інформаційні технології IT-сервісів архітектури додатків. *Екологічна безпека*, 3(47), 75-93. <https://doi.org/10.32347/2411-4049.2023.3.75-93>
9. Erl, T., & Chou, D. (2010). *SOA with .NET and Windows Azure: Realizing Service-Oriented Architecture with the Microsoft Platform*. Prentice Hall PTR.
10. Ford, N., Richards, M., Sadalage, P., & Dehghani, Z. (2020). *Fundamentals of Software Architecture: An Engineering Approach*. O'Reilly Media. <http://fundamentalsofsoftwarearchitecture.com>
11. Kopyika, O., Kalinin, V., & Lytvynenko, A. (2025). Use of Heterogeneous Special Purpose Telecommunication Networks for Provision of Convergent Services. *Lecture Notes in Networks and Systems*, 34-47. https://doi.org/10.1007/978-3-031-89296-7_3
12. Kopyika, O., & Skladannyi, P. (2021). Use of service-oriented information technology to solve problems of sustainable environmental management. *Information Technology and Mathematical Modeling for Environmental Safety*, 3021, с. 66–75.
13. Кучук, Г. А., Можаяев, О. О., & Лаврут, О. О. (2016). *Мережецентричні системи управління військами*. ХНУПС.
14. Лисенко, О. І. (2020). Інформаційні системи спеціального призначення: принципи побудови та інтеграції. *Системи управління, навігації та зв'язку*, 4(60), с. 15–21.
15. Литвиненко, А. О. (2026). Інфраструктура інтеграції програмно-апаратних комплексів інформаційних систем спеціального призначення. *Проблеми інформатизації та управління*, 1(85), с. 69-71. <https://doi.org/10.18372/2073-4751.85.21095>
16. Mittal, S., & Ranganathan, A. (2021). Evolution of Microservices in Military Information Systems. *2021 IEEE Military Communications Conference (MILCOM)*, с. 310-316. <https://doi.org/10.1109/MILCOM52596.2021.9652431>
17. The Open Group. (2021). *SOA Source Book - What Is SOA?*. <https://collaboration.opengroup.org/projects/soa-book/pages.php?action=show&ggid=1314>
18. Tarasenko, I., Kisselevskiy, A., Karichenskiy, A., & Valiulin, T. (2007). Softline applies TMF standards as a guide when building Resource Inventory solution for nation-wide carrier Ukraine Telecom. *TM Forum Case Study Handbook*, 3, 27.
19. Weyns, D. (2020). *An Introduction to Self-Adaptive Systems: A Guide for Complex Software Systems*. Wiley.

I. Zamrii, A. Lytvynenko

USING WEB SERVICES ARCHITECTURE FOR INFORMATION SYSTEMS WITH SELF-CONFIGURATING SOFTWARE AND HARDWARE COMPLEXES

This paper investigates the application of web services architecture for building special-purpose information systems based on self-adaptive hardware and software complexes. The study is motivated by the rapid evolution of information technologies and their critical role in modern military operations, particularly within the framework of Network-Centric Warfare adopted by NATO. The paper analyzes the transformation of architectural approaches from Service-Oriented Architecture (SOA) to web services architecture. The key differences between these paradigms are identified, including the transition to standardized web protocols, the use of open application programming interfaces (APIs), and the implementation of loosely coupled distributed services. It is shown that web services architecture provides enhanced interoperability, scalability, and flexibility compared to traditional SOA-based systems. A reference architecture for self-adaptive hardware and software complexes is proposed, consisting of four layers: communication, information, application, and presentation. The architecture allowed for the unification of technologies used in the creation of software and hardware complexes that take into account the usage of self-managed adapted networks and an information

exchange platform. The necessity of developing self-configuring software and hardware complexes capable of adapting to changes in operating conditions and user roles has been substantiated. Such complexes facilitate automatic configuration changes based on operating conditions and ensure resilience to shifts in network structure and system composition. This approach enables the expansion of functional capabilities of standard hardware and software systems, allowing for the creation of various automated workstations that self-configure for specific user roles. It is demonstrated that the use of standardized and modular hardware and software complexes enables the dynamic formation of automated workstations within a unified information environment.

Keywords: web services architecture, service-oriented architecture, self-adaptive systems, hardware and software complex, network-centric warfare, microservices, distributed systems.

Надійшла до редакції: 19.06.2026

Прийнята до друку: 03.07.2026

Опубліковано: 17.08.2026

© 2026 Замрій І. В., Литвиненко А. О.

Цей матеріал ліцензовано за умовами CC BY 4.0. <https://creativecommons.org/licenses/by/4.0/>