

## ХМАРНІ ТЕХНОЛОГІЇ У ІНДУСТРІЇ ВІДЕОІГОР

М.С. Нападайло, Л.В. Кабак, О.А. Сподинець  
(Україна, Дніпро, Національний ТУ «Дніпровська політехніка»)

**Постановка проблеми:** Десятиліттями у відеоіграх є ціла череда проблем, які розробники не раз намагались вирішити самими різними способами та засобами, але їм це не як не вдавалося, серед них такі як:

- “піратство” (Незаконне використання програмних засобів без купівлі ліцензії);
- “читери” (Незаконна зміна ігрового ПЗ/додавання до нього нових компонентів, які дають гравцю переваги над іншими гравцями);
- розповсюдження та встановлення користувацьких модифікацій без дозволу власників ПЗ;
- мультиплатформені збереження ігрового процесу (дозволяють на іншому девайсі продовжити гру, де зупинився користувач на попередньому пристрої).
- мультиплатформена розробка відеоігор.

Для захисту від “піратства” створювались цілі технології захисту від несанкціонованого злому (Наприклад Denuvo Anti-Tamper), деякі проекти вимагали навіть для однокористувацької гри постійного підключення до інтернету, щоб гра у “real time” робила перевірку на ліцензування гри. Видавництва відеоігор та студії розробників погрожували подати позови до суду, а у деяких випадках і подавали їх для того, щоб заборонити деяким “аматорським” студіям випускати свої модифікації для їх ігор. Різні студії створювали свою хмарні сервіси для реалізації мультиплатформових збережень, але вони зазвичай просто зберігали рівень, на якому зупинився гравець та параметри його героя/аккаунту. Ще у далекі часи компанія Nintendo намагалася впровадити подібне рішення для своїх ігрових консолей і випускала окремий аксесуар Nintendo Transfer Pak для передачі збережень між консолями Nintendo 64 та Nintendo GameBoy. Але цей аксесуар не знайшов тоді популярності і його підтримує лише 16 ігор[1]. Для спрощення розробки ігор на різні пристрої, навіть консолі восьмого покоління перевели на архітектуру X64 (хоча всі попередні консолі мали архітектуру, яка різнилася від архітектури звичайних ПК, і навіть ще у сьомого покоління були свої архітектури - PowerPC та Cell (повна назва Cell Broadband Engine Architecture)[3][4]).[2]

Але вже сьогодні з’явилися інструменти які в змозі вирішити ці проблеми, і це саме Хмарні технології.

**Основна частина:** Як взагалі працює хмарний геймінг?

Користувач використовує на своєму комп’ютері пристрій введення інформації, будь то клавіатура з маніпулятором миші, чи геймпад, ігровий руль, джойстик тощо. Після цього введена інформація передається через мережу на

віддалені сервери, на яких і розташована запущена гра. Після цього цей сервіс віддає відеопотік, який транслюється нам на екран.

Тобто простими словами, це можна порівняти з трансляцією відео з відеохостингу, але за умови того, що ми контролюємо події цього відео, тим самим від нас залежить, яка інформація наступною буде транслюватися на наш екран (рис. 1).

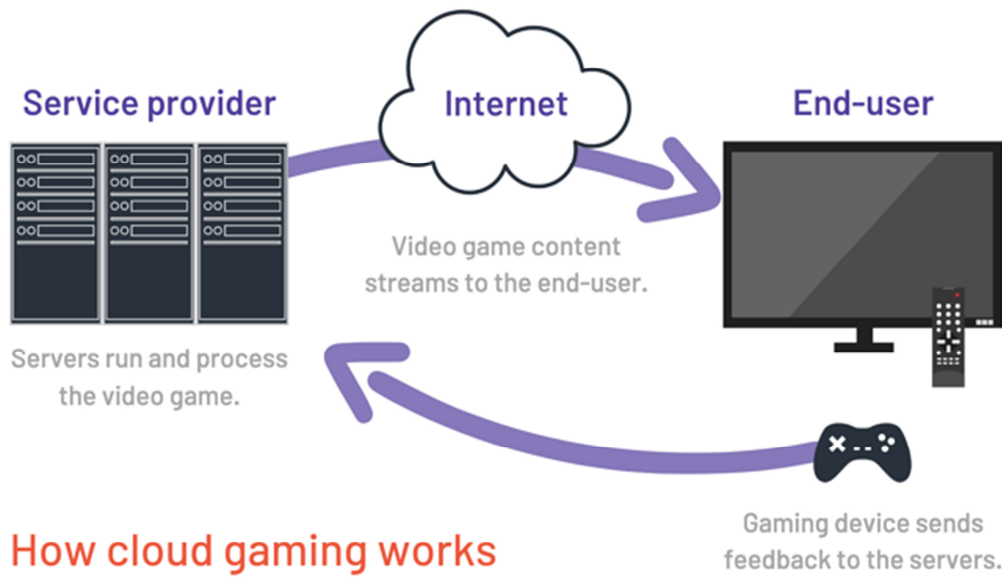


Рис. 1. Схема зв'язку

Сервери зазвичай встановлює та налаштовує та ж сама компанія, що і створює клієнт-серверний додаток та і в загалом розробляє всю систему. Зазвичай такі компанії вже мають свої сервери, які використовують у інших цілях.

Яким чином хмарні технології вирішують вищепоставлені проблеми?

1. Гра знаходиться на віддалених серверах і тому її модифікація неможлива. На сервери неможливо встановити неліцензійне ПЗ, яке зламає програмний ключ гри, або підмінить його, неможливо також встановити ПЗ, яке буде давати перевагу над іншими гравцями. Крім цього в такому випадку власники гри отримують можливість контролювати усі користувацькі модифікації для своєї гри, можна буде створювати окремі системи, в які користувачи зможуть викладати свої модифікації і тільки після проходження модерації від розробників, ці модифікації можуть ставати доступними для завантаження іншими гравцями.

2. Є можливість створити лаунчери для різних платформ (windows, linux, android, ios, webOS, Tizen OS), що дозволить одну і ту ж саму гру запускати на різних пристроях. Так як гра знаходиться на віддалених серверах і весь процес у реальному часі зберігається там, то проблематика “мультиплатформених збережень прогресу” також вирішена.

Чи є подібні системи, які вже реалізовані, так на даний момент вже є ряд систем, які вже працюють, проходять бетатестування чи тільки анонсовані:

- Playstation Now;
- Geforce Now;
- Google Stadia;
- Microsoft Project xCloud (робоча назва);
- Drove;
- Shadow;
- Vortex.

#### **ПЕРЕЛІК ПОСИЛАНЬ:**

1. Nintendo Famdom, Transfer Pak //URL: [https://nintendo.fandom.com/wiki/Transfer\\_Pak](https://nintendo.fandom.com/wiki/Transfer_Pak).
2. Wikipedia, Сьоме покоління ігрових систем //URL: [https://uk.wikipedia.org/wiki/Сьоме\\_покоління\\_ігрових\\_систем](https://uk.wikipedia.org/wiki/Сьоме_покоління_ігрових_систем) (або [https://uk.wikipedia.org/wiki/%D0%A1%D1%8C%D0%BE%D0%BC%D0%B5\\_%D0%BF%D0%BE%D0%BA%D0%BE%D0%BB%D1%96%D0%BD%D0%BD%D1%8F\\_%D1%96%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D1%85\\_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC](https://uk.wikipedia.org/wiki/%D0%A1%D1%8C%D0%BE%D0%BC%D0%B5_%D0%BF%D0%BE%D0%BA%D0%BE%D0%BB%D1%96%D0%BD%D0%BD%D1%8F_%D1%96%D0%B3%D1%80%D0%BE%D0%B2%D0%B8%D1%85_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC)).
3. Wikipedia, PowerPC //URL: <https://uk.wikipedia.org/wiki/PowerPC>
4. Wikipedia, Cell //URL: [https://uk.wikipedia.org/wiki/Cell\\_\(процесор\)](https://uk.wikipedia.org/wiki/Cell_(процесор)) (або [https://uk.wikipedia.org/wiki/Cell\\_\(%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80\)](https://uk.wikipedia.org/wiki/Cell_(%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80)))

УДК 004.3

## **DEPENDENCY INJECTION КОНТЕЙНЕР ТА ВПРОВАДЖЕННЯ ЙОГО В СУЧАСНІ WEB-ДОДАТКИ**

К.Ю. Островська, О.В. Захарченко  
(Україна, Дніпро, Національна металургійна академія України)

**Постановка проблеми.** Впровадження залежності (Dependency injection, DI) - процес надання зовнішньої залежності програмного компоненту. Є специфічною формою «інверсії управління» (Inversion of control, IoC), коли вона застосовується до управління залежностями. У повній відповідності з принципом єдиних обов'язків об'єкт віддає піклуватися про побудову необхідних йому залежностей зовнішньому, спеціально призначеному для цього спільного механізму.

При цьому впровадження залежностей об'єкт пасивний і не вживає взагалі ніяких кроків для з'ясування залежностей, а надає для цього сеттери і / або приймає своїм конструктором аргументи, за допомогою яких впроваджуються залежності.