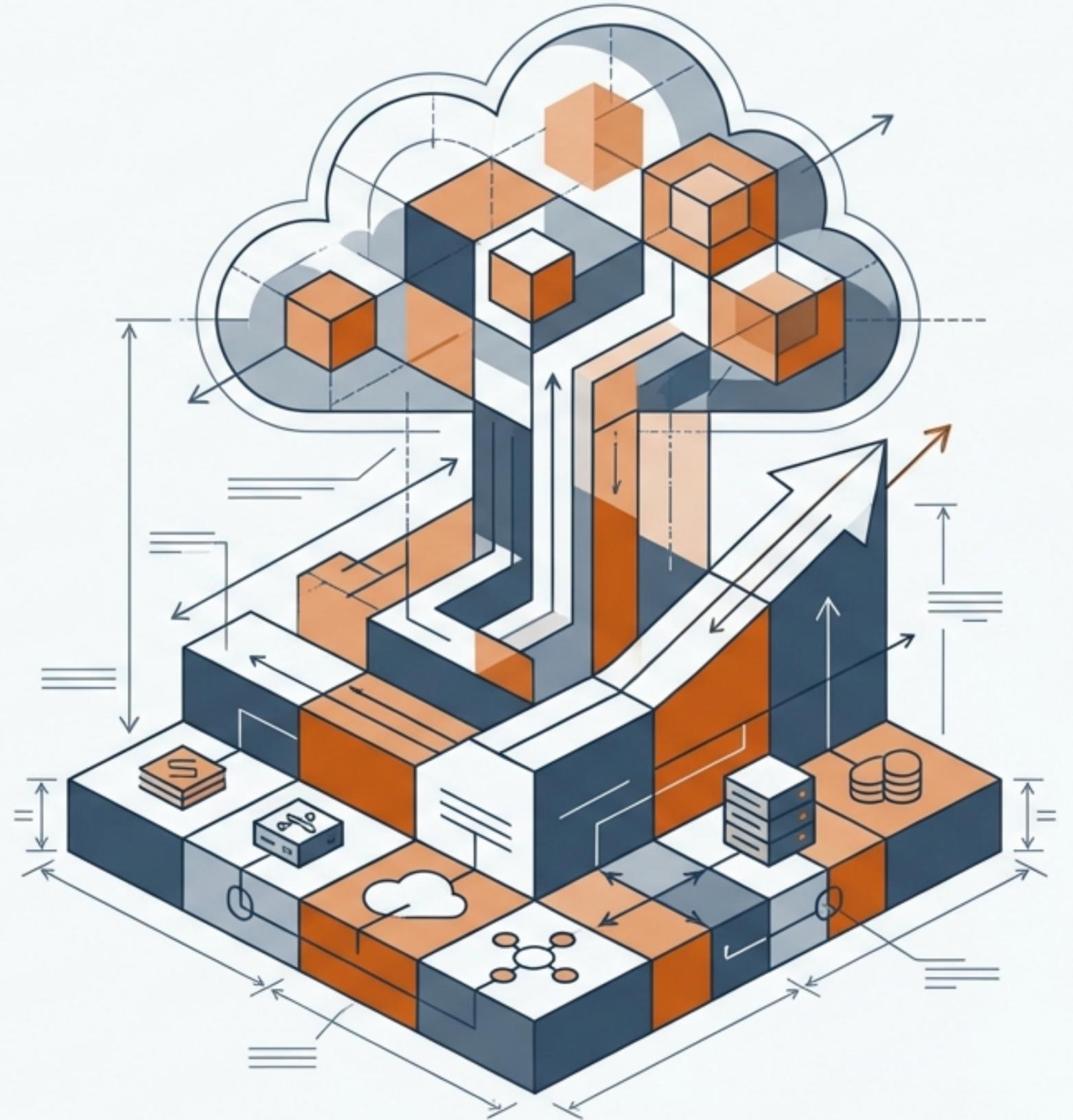


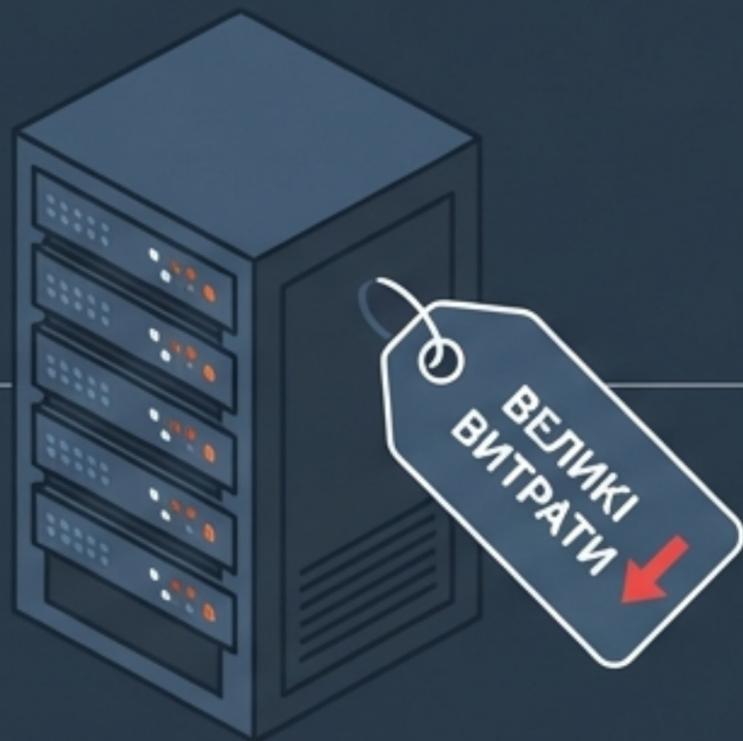
Моделі хмарних сервісів: Фундамент цифрової трансформації

Економічна ефективність,
масштабування та стратегія
вибору для сучасного бізнесу



Економіка ХааS: Від володіння до доступу

CapEx
(Капітальні витрати)



Купівля серверів,
охолодження, амортизація

OpEx
(Операційні витрати)



Оплата за використання,
масштабування, фокус на цінності

«Як послуга» — це передача технічної важкої роботи постачальнику,
щоб ви могли зосередитися на бізнесі.

Аналогія піци: Розподіл відповідальності



**On-Premises
(Домашнє)**

Тісто

Піч

Стіл для піци



**IaaS
(Спільна кухня)**

Інгредієнти

Стіл для піци

Піч/Газ

Обладнання



**PaaS
(Їжа на винос)**

Стіл для піци

Приготування

Піч/Газ

Обладнання



**SaaS
(Вечірка)**

Все включено

Приготування

Піч/Газ

Обладнання

Ви керуєте (Slate Blue)

Постачальник керує (Deep Charcoal)

Infrastructure as a Service (IaaS): Цифровий Цифровий будівельний майданчик

Визначення

Доступ до віртуалізованого обладнання. Клієнт орендує "залізо" у хмарі.

Аудиторія

Для системних адміністраторів та архітекторів.

Приклади

AWS, Microsoft Azure, Google Compute Engine.



Економічна вигода: Контроль над потужністю без інвестицій у дата-центр.

Platform as a Service (PaaS): Прискорювач розробки

Визначення

Готове середовище для коду. ОС та інструменти вже налаштовані.

Аудиторія

Для розробників програмного забезпечення.

Приклади

Heroku, Google App Engine, Azure App Service.



Економічна вигода: Швидкий вихід на ринок (Time-to-Market). Фокус лише на коді.

Software as a Service (SaaS): Готове бізнес-рішення

Визначення

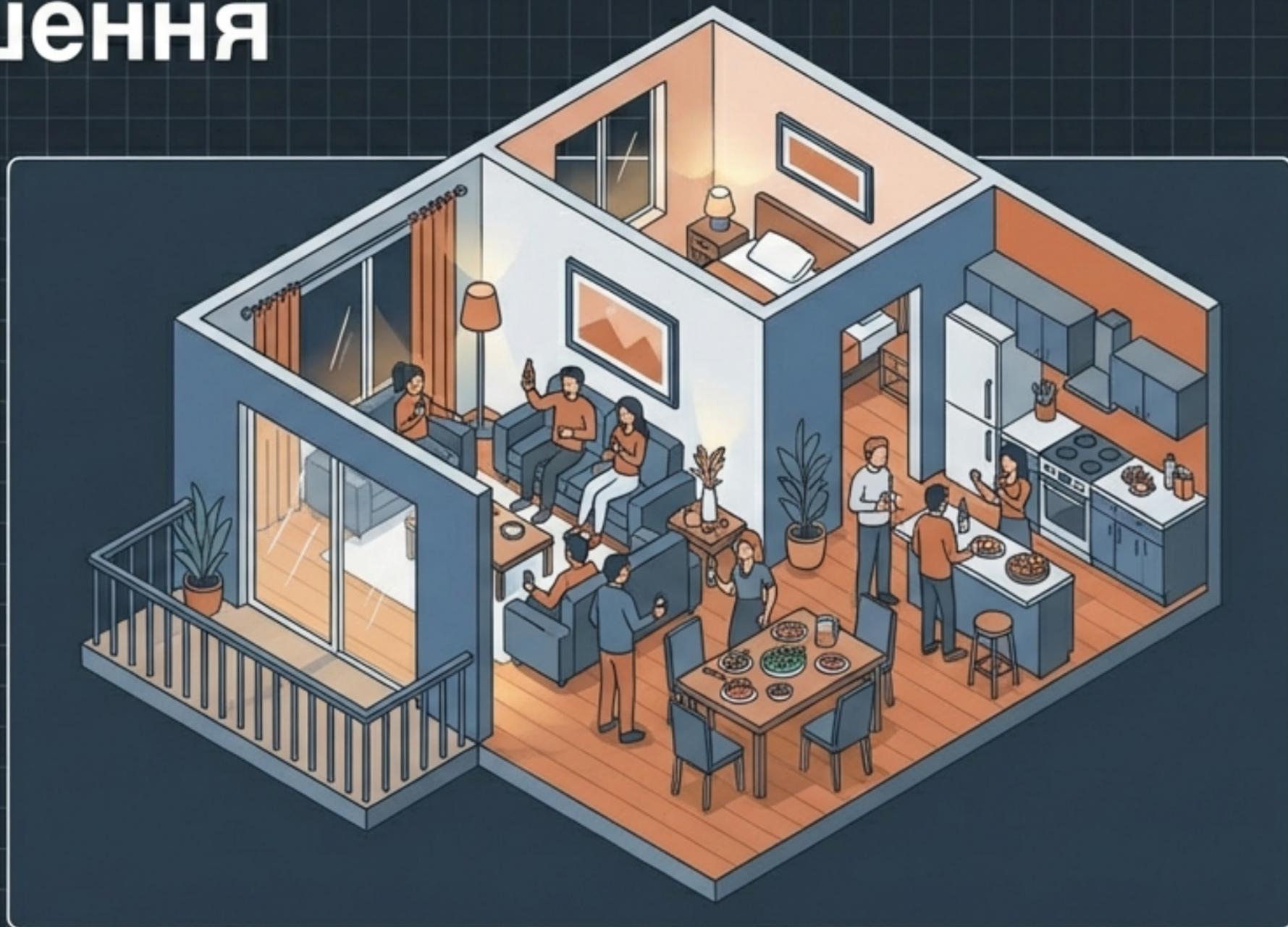
ПЗ за підпискою. Миттєвий доступ через браузер.

Аудиторія

Для кінцевих користувачів та бізнесу.

Приклади

Salesforce, Google Workspace, Shopify, Dropbox.



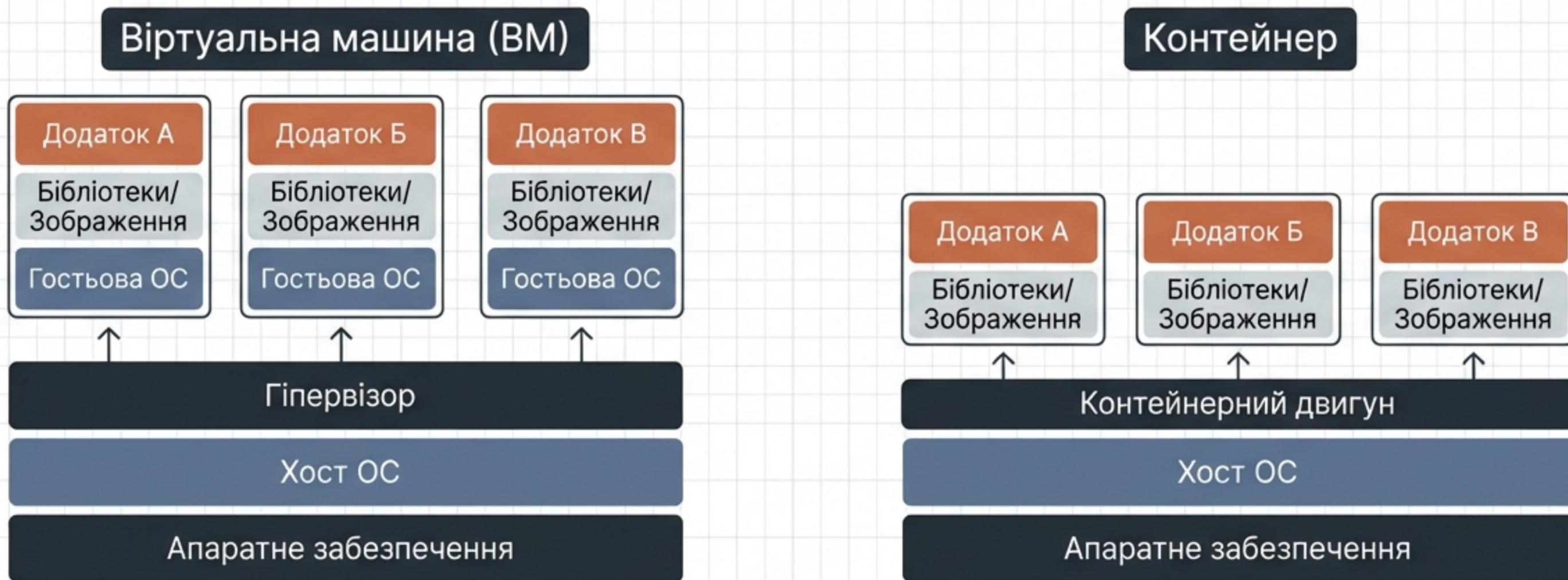
Економічна вигода: Нульові витрати на підтримку. Купуй, не будуй.

Матриця рішень: IaaS, PaaS чи SaaS?



Економічна вигода: Чіткий шлях до оптимізації ресурсів та швидкості впровадження.

Контейнеризація: Стандартизація логістики коду



Docker дозволяє упакувати код і залежності в один легкий стандартний блок.

Оркестрація: Керування флотом контейнерів



Kubernetes (K8s) —
автоматизація
розгортання.



Scaling: Автоматичне
масштабування під
навантаженням.



Self-healing:
Перезапуск контейнерів
при збоях.

Serverless: Архітектура без серверів

Core Concept: Виконуйте код, не думаючи про сервери.

Економічна революція:

- 💰 Pay-per-execution (Оплата за мілісекунди).
- 🚫 Zero Idle Cost (0\$ коли немає трафіку).
- 📈 Нескінченне масштабування.

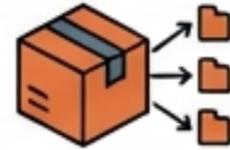


Методологія 12-Factor App: Стандарт якості



Кодова база
(Codebase)

One codebase tracked in revision control, many deploys.



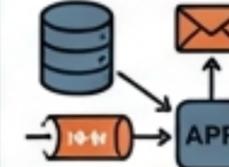
Залежності
(Dependencies)

Explicitly declare and isolate dependencies.



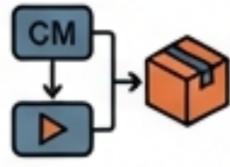
Конфігурація
(Config)

Store config in the environment.



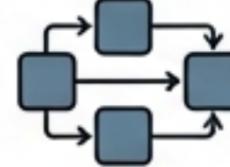
Служби
(Backing services)

Treat backing services as attached resources.



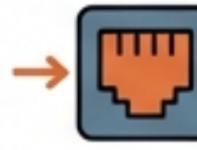
Збірка/Випуск/
Запуск
(Build, release, run)

Strictly separate build and run stages.



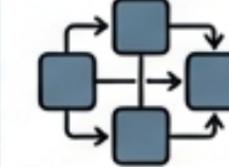
Процеси
(Processes)

Execute the app as one or more stateless processes.



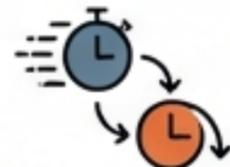
Прив'язка
портів
(Port binding)

Export services via port binding.



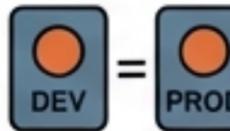
Паралелізм
(Concurrency)

Scale out via the process model.



Одноразовість
(Disposability)

Maximize robustness with fast startup and graceful shutdown.



Паритет
(Dev/prod parity)

Keep development, staging, and production as similar as possible.



Журнали
(Logs)

Treat logs as event streams.



Адмін-процеси
(Admin processes)

Run admin/management tasks as one-off processes.



Фундамент для масштабованих та надійних систем.

Висновок: Хмара як бізнес-стратегія



Найуспішніші компанії цифрової економіки керують **не серверами, а сервісами.**