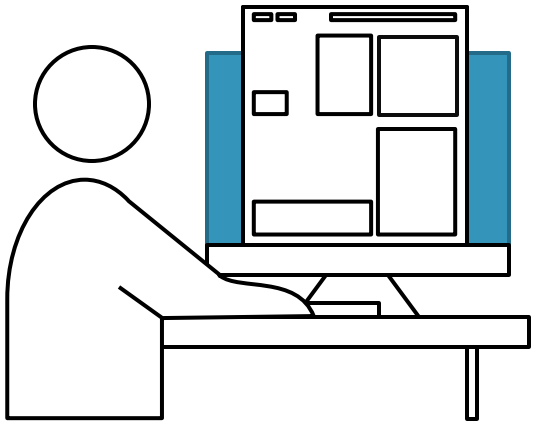
ПОШУК ОПТИМАЛЬНИХ ПАРАМЕТРІВ В АУКЦІОНІ ЗА ДОПОМОГОЮ НАВЧАННЯ З ПІДКРІПЛЕННЯМ

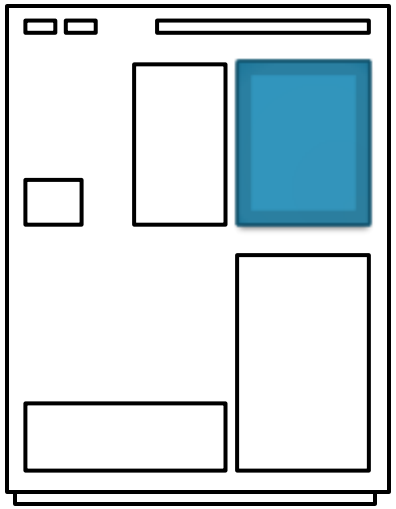
Сивохіп А.В.

Керівник: Крюкова Г. В.

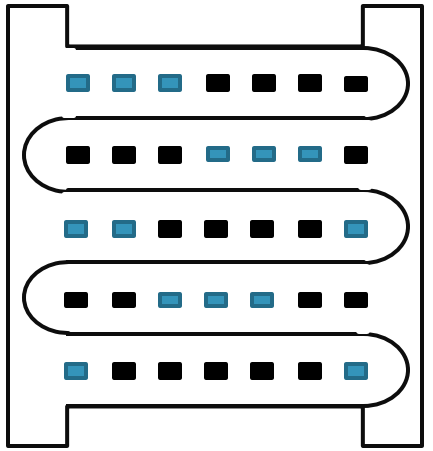
Open Real Time Bidding (oRTB)



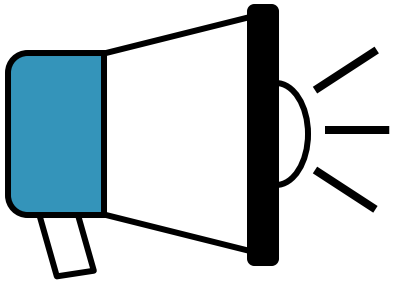
КОРИСТУВАЧ



ПАБЛІШЕР

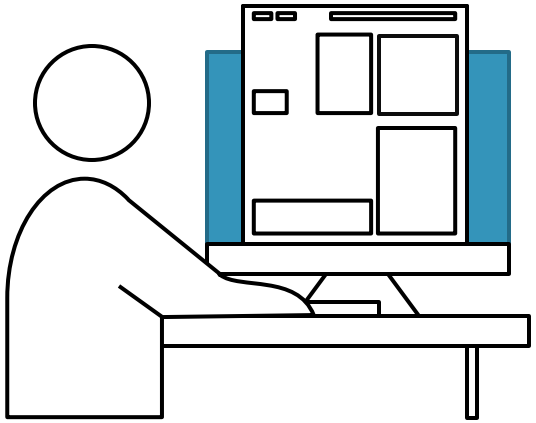


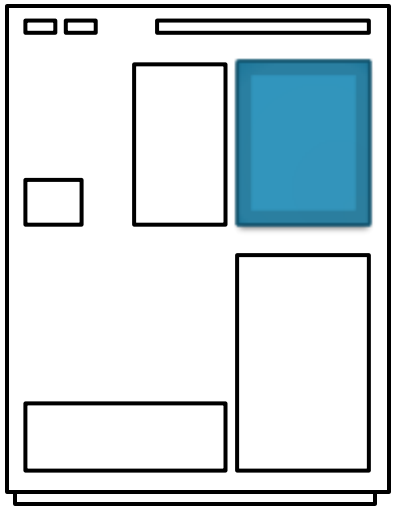
ПЛАТФОРМА
AD EXCHANGE

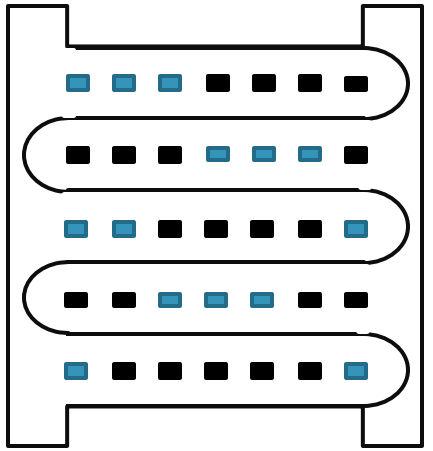


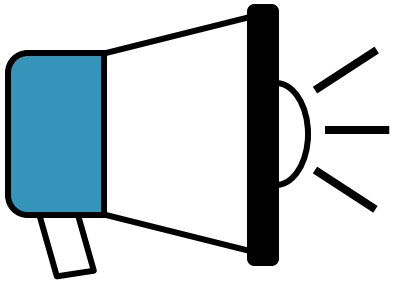
РЕКЛАМОДАВЕЦЬ

1 → 2 → 3 → 4









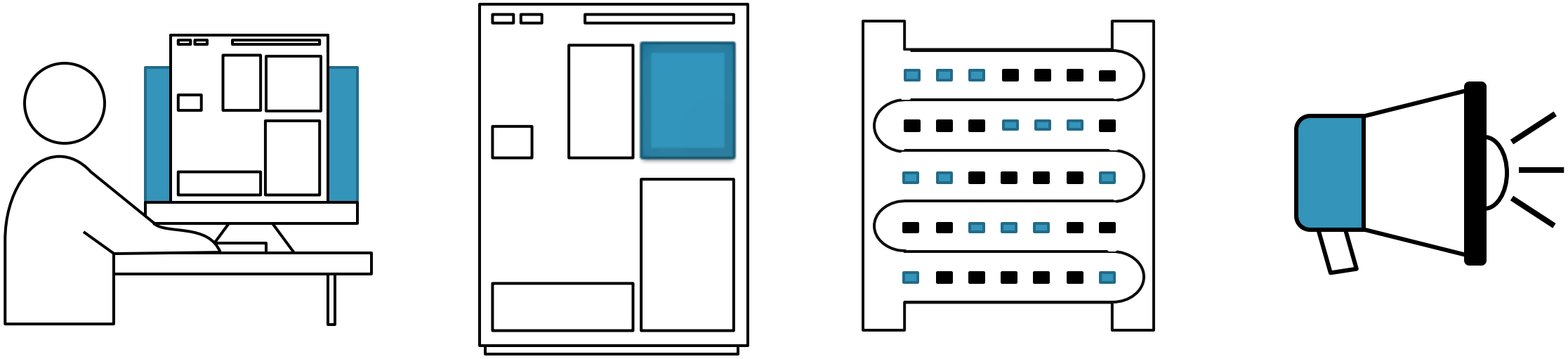
КОРИСТУВАЧ ПАБЛІШЕР ПЛАТФОРМА AD EXCHANGE РЕКЛАМОДАВЕЦЬ

7 6 5

Крок 1. Користувач

1 2 3 4

КОРИСТУВАЧ → ПАБЛІШЕР → ПЛАТФОРМА AD EXCHANGE → РЕКЛАМОДАВЕЦЬ



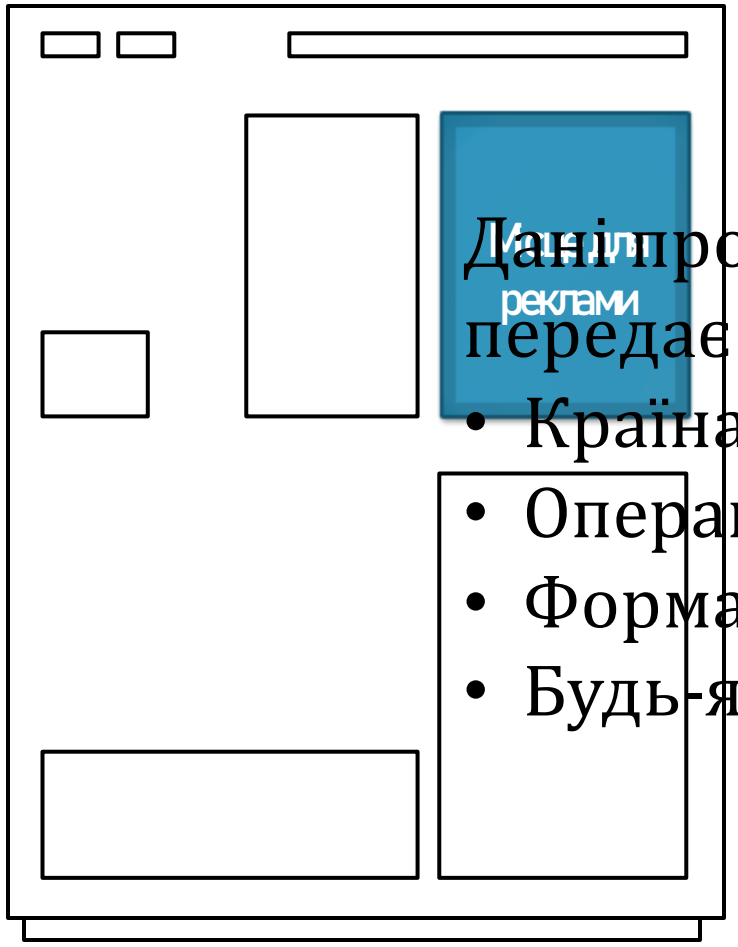
7 6 5

Користувач заходить на ресурс та ініціює oRTB процес

Крок 2. Партнер

Власник ресурсу, на якому є можливість розміщати рекламу

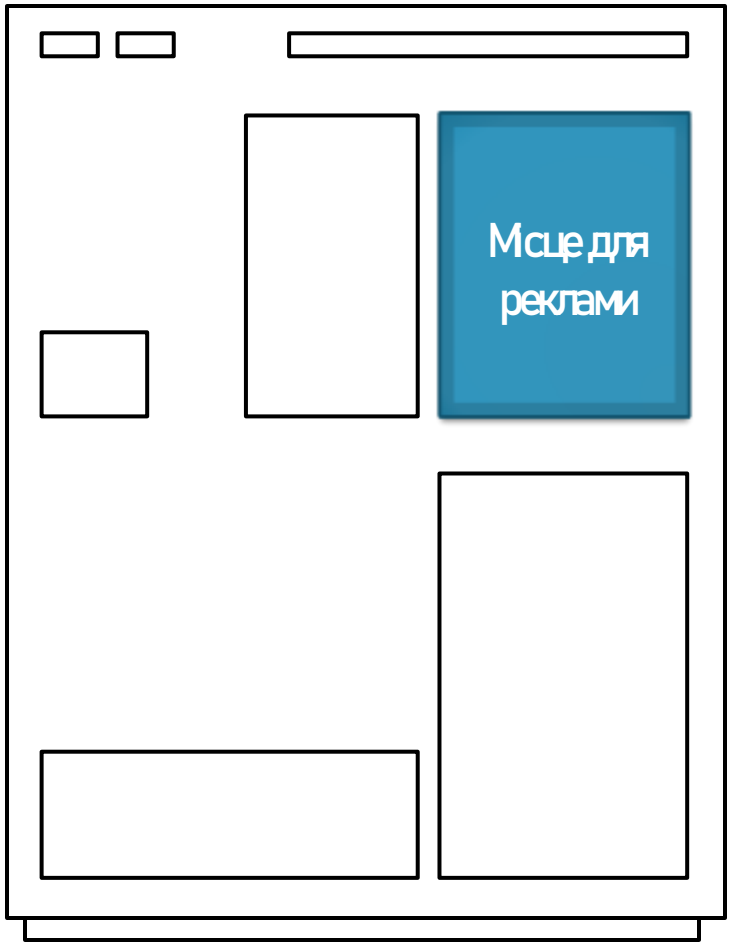
Крок 2. Паблішер

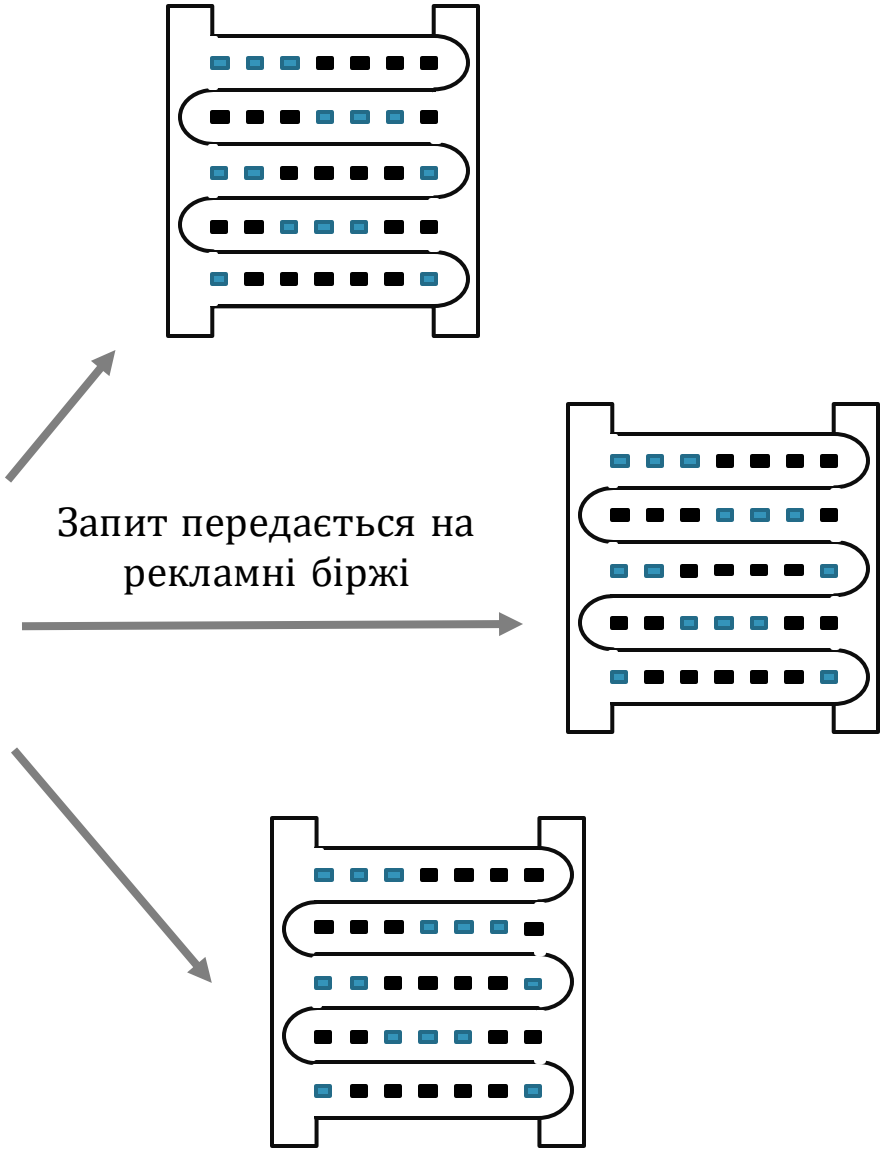


Дані про користувача, які передає паблішер у запиті:

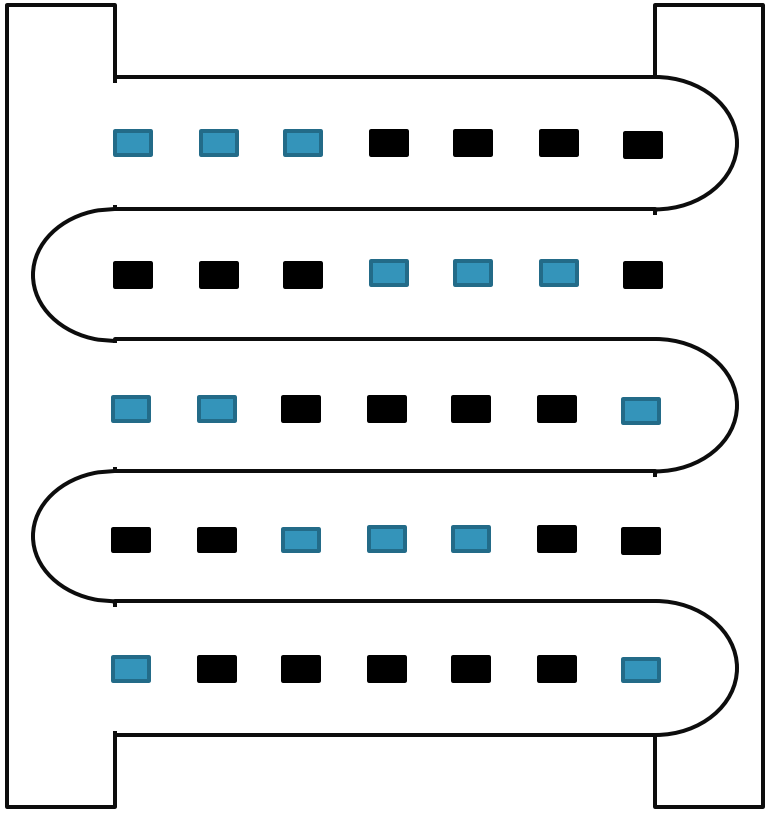
- Країна
- Операційна система
- Формат реклами
- Будь-яка інша інформація

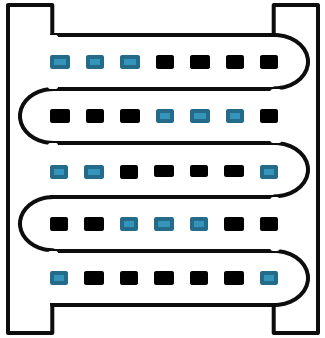
Крок 2. Паблішер





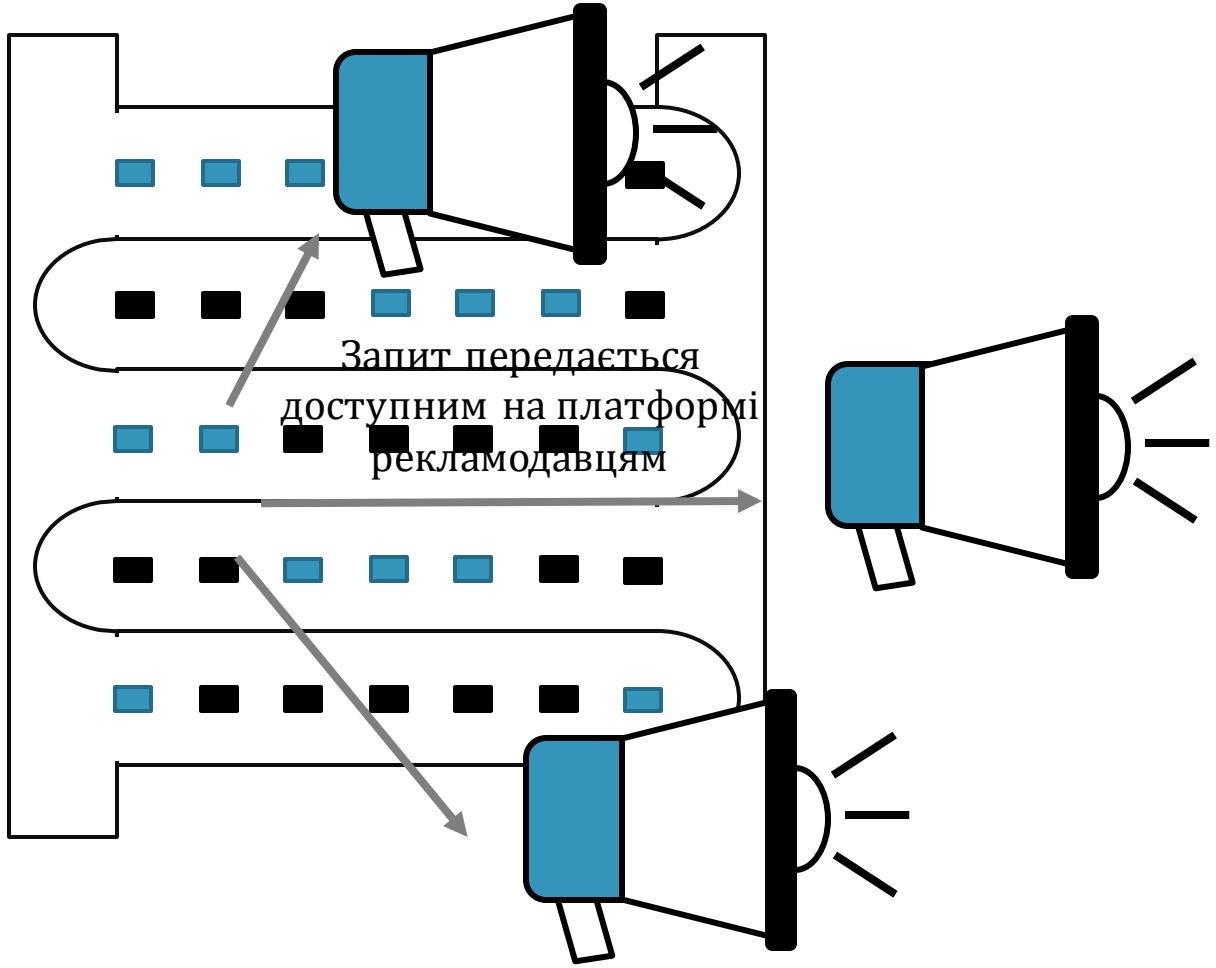
Крок 3. Рекламна біржа (Ad Exchange Platform)



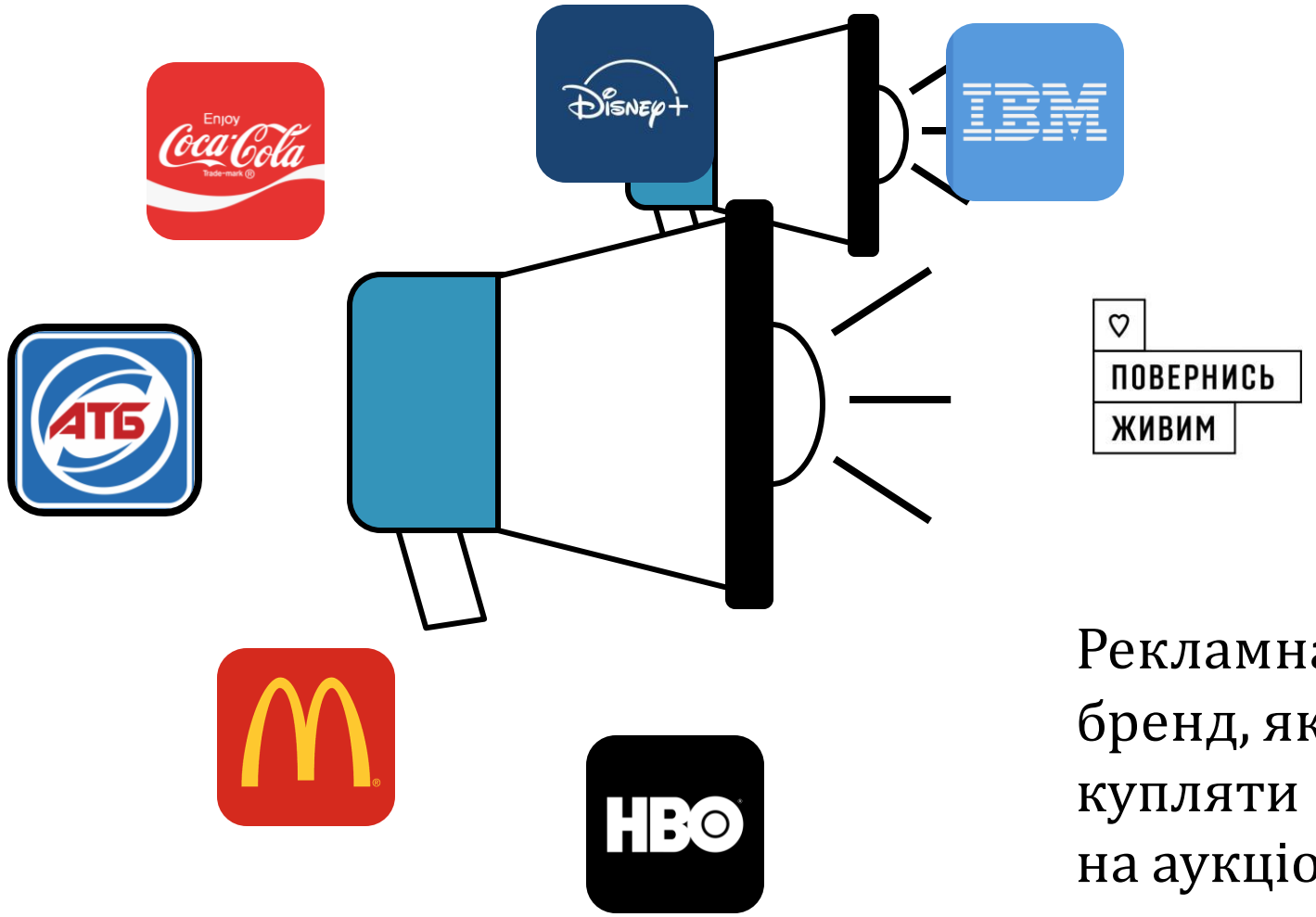


Платформа, що об’єднує паблішерів та рекламодавців

Крок 3. Рекламна біржа (Ad Exchange Platform)

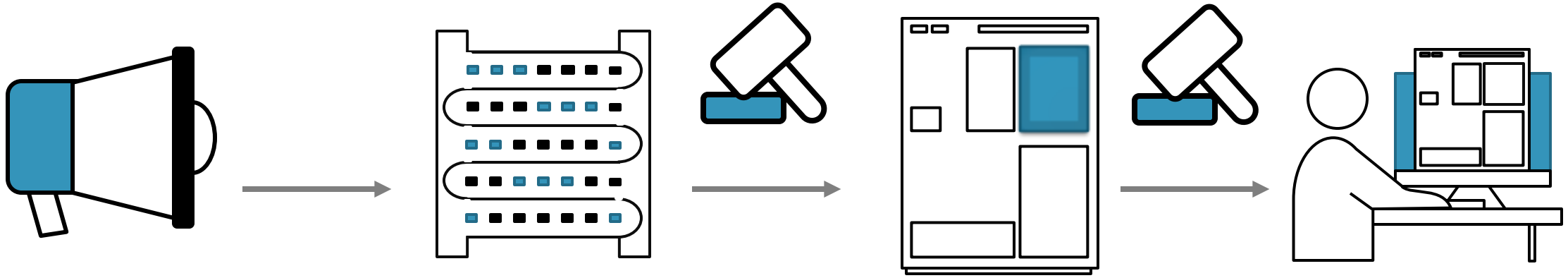


Крок 3. Рекламна біржа (Ad Exchange Platform)



Рекламна агенція або бренд, який має бажання купляти рекламні місця на аукціоні

Запит проходить зворотній шлях



5

Ставка рекламодавця надсилається на платформу

6

Ставка рекламодавця, що виграла аукціон на біржі, надсилається паблішеру

7

Рекламодавець, що надіслав найвищу ставку, виграє можливість показати рекламу користувачу

Основні моменти та актуальні проблеми

- У аукціоні oRTB виграє найвища ставка
- Платформа Ad Exchange є посередником між паблішером та рекламодавцем
- Платформа додає свою націнку до запиту
- **Якщо націнка формує зависоку ставку – рекламодавець не відповість на запит**
- **При мінімальній націнці втрачається можливість отримати вищий прибуток**

Мета задачі:

Автоматизувати та оптимізувати процес виставлення націнки для кожного запиту на основі наявної інформації з метою максимізації заробітку платформи

Метод вирішення проблеми:

- **Q-навчання** — алгоритм безмодельного навчання з підкріпленням.
- Ітераційне вивчення оптимальної функції Q-значення за допомогою рівняння Белмана:

$$V(s) = \max_a (R(s, a + \gamma \sum_{s'} P(a, s, s')V(s'))$$

- Таблиця Q-значень, яка буде оновлюватись після кожного кола алгоритму Q-навчання.

Компоненти Q-навчання

- Множини станів середовища S
- Множини дій A
- Функція винагороди
- Процес оновлення Q-значень

Стани S

Стан S на момент часу t складається з усієї доступної інформації про поточний запит на аукціон.

Визначимо стан s як сукупність (m, c, os, f), де

- m — мінімальна ставка
- c — країна користувача,
- os — операційна система користувача,
- f — формат оголошення.

Дії A

Дії A відповідатимуть набору всіх можливих націнок, які платформа може застосувати до ціни пропозиції, отриманої від рекламодавця.

$$A = \{a_1, a_2, \ldots, a_n\}$$

a_i представляє i-й можливий відсоток націнки, який платформа може застосувати до мінімальної ставки.

Винагорода $\boldsymbol{R}$

Винагорода $\boldsymbol{R(s, a)}$ у момент часу $\boldsymbol{t}$ — це прибуток, отриманий платформою за запит на аукціон, враховуючи відсоток націнки $\boldsymbol{a}$.

$$R(s,a) = \begin{cases} b * \dfrac{a}{100}, \text{якщо } b \text{ є виграшною ставкою} \\ 0, \quad \text{якщо рекламодавець не відповів або } b \text{ не є виграшною ставкою} \end{cases}$$

Оновлення Q-значень

$$\boldsymbol{Q(s,a) \leftarrow Q(s,a) + \alpha * [r + \gamma * max_{a'} Q(s',a') - Q(s,a)]}$$

- $\boldsymbol{Q(s,a)}$ - Q-значення пари стан-дія $\boldsymbol{(s,a)}$,
- $\boldsymbol{\alpha}$ - швидкість навчання,
- $\boldsymbol{r}$ - миттєва винагорода, отримана при виборі дії $\boldsymbol{a}$ у стані $\boldsymbol{s}$,
- $\boldsymbol{\gamma}$ - дисконтний коефіцієнт або фактор знецінювання
- $\boldsymbol{max_{a'} Q(s',a')}$ - максимальне Q-значення серед всіх можливих дій $\boldsymbol{a'}$ у наступному стані $\boldsymbol{s'}$.

Кроки застосування методу

1. Ініціалізація
2. Дослідження чи використання (Exploration or Exploitation)
3. Вибір дії та перехід до нового стану
4. Обробка запиту
5. Розрахунок винагороди та оновлення Q-значень
6. Повторення кроків 3-5

Приклад реалізації методу Q-навчання

Спочатку ми імпортуємо необхідні бібліотеки

```
import math
import numpy as np
import itertools as it
import math as m
import random
import rtb  # imaginary module to work with
```

Приклад реалізації методу Q-навчання

Визначення дискретних значень дій та кроку

```
action_step = 5
actions = np.arange(0, 100, action_step)[1:]
number_of_actions = len(actions)
```

Приклад реалізації методу Q-навчання

Визначення дискретних значень мінімальної ставки, країн, типів реклами та операційних систем

```
bid_floor_step = 0.5
bid_floors = np.arange(0, 50, bid_floor_step)
countries = np.array(["Ukraine", "USA"])
type_of_ads = np.array(["banner", "video"])
platforms = np.array(["IOS", "Android", "Windows"])
```

Приклад реалізації методу Q-навчання

Визначення станів

```
states = [
    (b, c, t, p)
    for b in bid_floors
    for c in countries
    for t in type_of_ads
    for p in platforms
]
number_of_states = len(states)
states_map = {k: v for (v, k) in enumerate(states)}
```

Приклад реалізації методу Q-навчання

Створення Q-таблиці

```
Q = np.zeros((number_of_states, number_of_actions))
```

Приклад реалізації методу Q-навчання

Визначення параметрів

```
eps = 0.1
lr = 0.5
```

Приклад реалізації методу Q-навчання

Початок навчання

```
while True:
    req = (
        rtb.get_next_request()
    )  # get next request with all info needed for algorithm to work
```

Приклад реалізації методу Q-навчання

Вибір дії

```
state_index = states_map[
        req.state(bid_floors)
    ]   # state is a convinient method to get all needed states in
tuple, including finding of corresponding bit floor bucket


    if random.uniform(0, 1) < eps:
        action_index = random.randint(0, number_of_actions)
    else:
        action_index = np.argmax(Q[state_index, :])
```

Приклад реалізації методу Q-навчання

Обробка дії та отримання винагороди

```
 res = req.proceed(
         markup=actions[action_index]
    )   # method on request to proceed with selected markup and get results
    reward = res.income / req.bid_floor  # calculated normalized income
```

Приклад реалізації методу Q-навчання

Оновлення Q-значень

```
Q[state_index, action_index] = Q[state_index, action_index] + lr * (
        reward - Q[state_index, action_index]
    )
```

Процес повторюється безкінечно, додаючи і оновлюючи Q-значення в таблиці.

Результати та висновки

- Створений алгоритм Q-навчання
- Можливе реальне застосування у індустрії
- Покращення точності алгоритму при більшій кількості станів:
  - зменшення кроку ставки
  - додаткові дані запиту
- Ефективними також можуть бути методи SARSA та Policy gradient methods
- Загалом, використання методу Q-навчання є перспективним інструментом для оптимізації націнки в аукціоні oRTB

ДЯКУЮ ЗА УВАГУ!