НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

Автоматизованих систем обробки інформації і управління
(повна назва кафедри)

«На правах рукопису»
УДК 004.021

До захисту допущено:
В.о. завідувача кафедри
_____ Олександр ПАВЛОВ
«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-науковою програмою «Інженерія програмного забезпечення комп'ютеризованих систем»

зі спеціальності 121 «Інженерія програмного забезпечення»

на тему: «Математичне та програмне забезпечення організації аукціонів продажу рекламних місць у реальному часі»

Виконав:
студент VI курсу, групи ІП-82мпв
Котляр Ілля Сергійович _____

Керівник:
Доцент каф. АСОІУ, к.т.н. Попенко В. Д. _____

Рецензент:
Доцент каф. АУТС, к.т.н. Рєпнікова Н. Б. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Автоматизованих систем обробки інформації і управління

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-наукова програма - «Інженерія програмного забезпечення комп'ютеризованих систем»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Олександр ПАВЛОВ
«___»_____20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Котляр Ілля Сергійович

1. Тема дисертації «Математичне та програмне забезпечення організації аукціонів продажу рекламних місць у реальному часі», науковий керівник дисертації доцент каф. АСОІУ, к. т. н. Попенко Володимир Дмитрович, затверджені наказом по університету від «24» березня 2020 р. № 910-с

2. Термін подання студентом дисертації 13 травня 2020

3. Об'єкт дослідження *технологія торгів у реальному часі (Real-Time bidding).*

4. Предмет дослідження *алгоритми проведення аукціонів у реальному часі для відбору рекламних оголошень.*

5. Перелік завдань, які потрібно *розробити проаналізувати ринок реклами в Інтернеті, дослідити систему RTB її задачі, переваги з традиційною моделлю, перспективи розвитку, проаналізувати існуючі підходи до обрання потрібного оголошення в системі торгів у реальному часі, розробити метод проведення торгів на основі стохастичного підходу та перевірити ефективність запропонованого методу.*

6. Орієнтовний перелік графічного (ілюстративного) матеріалу

Взаємодія компонентів в системі Real-Time Bidding

Серверна архітектура програмного забезпечення

7. Орієнтовний перелік публікацій *Стохастичний відбір рекламних оголошень в аукціоні реального часу, Технологія торгів у реальному часі в інтернет-маркетингу.*

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання *13 січня 2020 р.*

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	*Проаналізувати ринок реклами в Інтернеті*	*05.02.2020*	
2	*Дослідити систему RTB її задачі, переваги з традиційною моделлю, перспективи розвитку*	*15.03.2020*	
3	*Проаналізувати існуючі підходи до обрання потрібного оголошення в системі торгів у реальному часі*	*03.04.2020*	
4	*Розробити метод проведення торгів на основі стохастичного підходу та перевірити ефективність запропонованого методу*	*15.04.2020*	
5	*Проведення експериментальних досліджень*	*25.04.2020*	
6	*Оформлення документації*	*03.05.2020*	
7	*Подання роботи на попередній захист*	*05.05.2020*	
8	*Подання роботи на основний захист*	*20.05.2020*	

Студент Ілля КОТЛЯР

Науковий керівник Володимир ПОПЕНКО

РЕФЕРАТ

Магістерська дисертація: 126 с., 20 рис., 34 табл., 56 джерел, 2 додатка.

Актуальність. Інтернет сьогодні є невід'ємною частиною життя всіх людей на планеті. З кожним роком з'являються нові можливості, гаджети, технології та інструменти їх використання, в тому числі і для бізнесу. Природньо, що з поширенням Інтернету повсюдно, створюються і активно застосовуються різні методи просування товарів і послуг в мережі. Бізнес не стоїть на місці, і зараз кожна поважаюча себе компанія має свій сайт або, як мінімум, свою сторінку в соціальних мережах, іншими словами, так чи інакше, створює своє відображення в Інтернеті. Але мало створити сайт, потрібно ще зробити так, щоб цільова аудиторія приходила на нього і здійснювала необхідні для бізнесу дії. Цими діями можуть бути покупки (якщо це сайт інтернет - магазину), лайки, перегляди, репости і коментарі (якщо це сайт інформаційного ресурсу), перехід за реферальним посиланням (якщо цей сайт використовується як рекламний майданчик) тощо.

Щоб зробити це можливим існують системи, які дозволяють створити потік цільової аудиторії на сайт компанії. З іншого боку існують компанії, основний прибуток яких є дохід від реклами, яку вони публікують. Таких компаній стає все більше, адже навіть деякі розробки ігор користуються такою бізнес-моделлю. Таким чином, виникає попит поєднати ці дві сторони та автоматизувати процес створення реклами та визначення найкращого місця (сайту, ігри, соціальної мережі тощо) для показу цієї реклами, а також покращити процес визначення такого місця та користувача, який переглядає рекламу, для підвищення ефективності рекламної компанії.

Мета дослідження. Метою дослідження є підвищення ефективності механізмів фінансування всесвітньої мережі шляхом вдосконалення аукціонів по продажу рекламних місць у реальному часі.

Для досягнення мети необхідно виконати наступні **завдання**:

– проаналізувати ринок реклами в Інтернеті;

– дослідити систему RTB її задачі, переваги з традиційною моделлю, перспективи розвитку;

– проаналізувати існуючі підходи до обрання потрібного оголошення в системі торгів у реальному часі;

– розробити метод проведення торгів на основі стохастичного підходу та перевірити ефективність запропонованого методу.

Об'єкт дослідження – технологія торгів у реальному часі (Real-Time bidding).

Предмет дослідження – алгоритми проведення аукціонів у реальному часі для відбору рекламних оголошень.

Наукова новизна отриманих результатів полягає у розробці підходу, який дозволяє запобігати монополізації аукціонів в системі Real Time Binding. Обґрунтована ефективність стохастичного методу визначення переможця в порівнянні з розповсюдженим аукціоном найбільшої ціни обирання переможця на платформі DSP.

Практичне значення роботи полягає у застосування розробленого підходу щодо удосконалення проведення аукціону на реальній платформі DSP для покращення фінансових показників та просування на шляху до автоматизації роботи системи.

Апробація: результати викладалися на предзахисті роботи на кафедрі АСОІУ.

Публікації. Матеріали роботи опубліковані в статті «Стохастичний відбір рекламних оголошень в аукціоні реального часу» фахового журналу «Інтернаука» №5 – DOI 10.25313/2520-2057-2020-5-5825. Та тези доповіді на тему «Технологія торгів у реальному часі в Інтерне-маркетингу» у

IV Всеукраїнській науково-практична конференції молодих вчених та студентів «Інформаційні системи та технології управління».

ABSTRACT

Master's thesis: 126 pp., 20 images, 34 tables, 56 sources, 2 supplements.

Topicality. The Internet today is an integral part of the lives of all people on the planet. Every year there are new opportunities, gadgets, technologies and tools for their use, including for business. Naturally, with the spread of the Internet everywhere, various methods of promoting goods and services on the Internet are created and actively used. Business does not stand still, and now every self-respecting company has its own website or at least its own page on social networks, in other words, one way or another, creates its own reflection on the Internet. But it is not enough to create a site, you still need to make sure that the target audience came to it and carried out the necessary actions for business. These actions can be purchases (if it is an online store site), likes, views, reposts and comments (if it is an information resource site), clicking on a referral link (if this site is used as an advertising platform), etc.

In order to make it possible there are systems that allow you to create a traffic of target audience to the company's website. On the other hand, there are companies whose get income from the advertising they publish. There are more and more such companies, because even some game developers use such a business model. Thus, there is a demand to combine these two sides and automate the process of creating ads and determining the best place (site, game, social network, etc.) to display this ad, as well as improve the process of determining such place and user viewing ads to improve advertising effectiveness.

The purpose of the study is increasing the efficiency of the mechanisms of financing the world wide web by improving auctions for the sale of advertising space in real time.

To achieve the goal, the following **tasks** should be done:

– to analyze the Internet advertising market;

– snvestigate the RTB system, its tasks, advantages with the traditional model, prospects for development;

– to analyze existing approaches to selecting the right ad in the bidding system in real time;

– develop a method of bidding based on a stochastic approach and test the effectiveness of the proposed method.

The object of the research is Real-Time bidding.

The subject of the research is real-time bidding algorithms for selecting advertisements.

The scientific novelty of the obtained results is to develop an approach that prevents the monopolization of auctions in the Real Time Binding system. The efficiency of the stochastic method of determining the winner in comparison with the widespread auction of the highest price of choosing the winner on the DSP platform is substantiated.

The practical significance of the work is to apply the developed approach to improve the auction on a real DSP platform to improve financial performance and move towards automation of the system.

Testing: the results were presented at the pre-defence work at the Department of ASOIU.

Publications. Materials of the work are published in the article of the international scientific journal "Internauka" №5 - DOI 10.25313 / 2520-2057-2020-5-5825. And the thesis of the report on "Real-time bidding technology in Internet marketing" in

IV All-Ukrainian scientific-practical conference of young scientists and students "Information systems and management technologies".

ЗМІСТ

ВСТУП....12
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ....13
1.1 Real Time Bidding....13
1.2 Виникнення протоколу OpenRTB....14
1.3 Учасники RTB....15
1.3.1 Двосторонні ринки....15
1.3.2 Мережевий ефект....18
1.3.3 Demand-Side Platform....19
1.3.4 Supply-Side Platform....20
1.3.5 Data Management Platform....20
1.4 Послідовність процесу RTB....20
1.4.1 Алгоритм роботи RTB....23
1.5 Основні принципи OpenRTB....24
1.6 Архітектура RTB....25
1.6.1 Рівні протоколу RTB....25
1.6.2 Рівень-1: Транспортний....26
1.6.3 Рівень-2: Формат....27
1.6.4 Рівень-3: Транзакції....28
1.6.5 Рівень-4: Домен....29
1.7 Advertising Common Object Model....30
1.7.1 Media....31
1.7.2 Placement....34

1.7.3 Context....36
1.8 OpenRTB....39
1.9 Інші відповідності значень....41
Висновки до розділу 1....45
2 МАТЕМАТИЧНА МОДЕЛЬ....46
2.1 Аукціон другої ціни....46
2.2 Механізм Вікрі в інтернет-рекламі....49
2.3 Трагедія спільного в RTB....52
2.4 Покращення алгоритму аукціону другої ціни при виборі рекламних місць..54
2.5 Порівняння математичного сподівання стохастичного та класичного аукціонів....57
Висновки до розділу 2....59
3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ....60
3.1 Засоби розробки....60
3.2 Вимоги до технічного забезпечення....72
3.3 Архітектура програмного забезпечення....74
3.4 Функціональність системи....83
Висновки до розділу 3....90
4 ОПИС СТАРТАП ПРОЕКТУ....91
4.1 Опис ідеї проекту....91
4.2 Технологічний аудит ідеї проекту....95
4.3 Аналіз ринкових можливостей запуску стартап-проекту....96
4.4 Розроблення ринкової стратегії проекту....103

4.5 Розроблення маркетингової програми стартап-проекту 107
Висновки до розділу 4 111
ВИСНОВКИ 112
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 113
ДОДАТОК А 118
ДОДАТОК Б 119

ВСТУП

Традиційні медіа ресурси змушені проводити спеціальні дослідження, спрямовані на вивчення власної аудиторії. Користувачі Інтернету самі виставляють інформацію про себе у відкритий доступ, але ці дані необхідно збирати, аналізувати та зберігати. Упорядкувавши ці дані, з'являється можливість орієнтуватися на користувачів з певними інтересами, рівнем доходу та освіти, розрізняти користувачів за віковими та статевими групами. Такий підхід дозволяє показувати рекламу певній аудиторії, що значно може покращити результат на відміну від «гарантованих контрактів» [27] з майданчиками, які відвідують і люди, не зацікавлені в пропонованому товарі чи послузі. Рішенням такої проблеми стала поява алгоритмічних закупок рекламних оголошень. Такий підхід називають програматік (programmatic). Це сукупність методів закупівлі рекламних оголошень в Інтернеті з використанням автоматизованих систем і інструментів для прийняття рішень про угоду на основі даних про користувачів. Програматік дозволяє купувати не розміщення на сайті, а контакт з конкретним користувачем, який входить в цільову аудиторію рекламодавця [28]. Найбільшу популярність придбала технологія продажу і покупки реклами на основі аукціону в реальному часі – Real-time Bidding (RTB)[5, 23]. Оскільки інші технології не користувалися попитом, зараз єдиною технологією залишається Real-Time Bidding і тому поняття програматік і RTB є майже тотожними.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Real Time Bidding

Real Time Bidding (RTB) – технологія продажу і покупки показів реклами в Інтернеті на основі автоматизованого аукціону, що дозволяє рекламодавцеві розміщувати рекламу не на конкретному майданчику, а демонструвати її на різних майданчиках тільки для спеціально відібраної по певним фільтрам (таргетингу) аудиторії. Ціна рекламного контракту в ході аукціону встановлюється в автоматичному режимі, що дозволяє встановити оптимальний баланс між інтересами власників сайтів, які бажають отримати максимальний дохід від показу реклами, і рекламодавців, зацікавлених у максимальному ефекті від рекламної кампанії [4].

На відміну від традиційної схеми продажу «гарантованих контрактів», коли рекламодавець отримує певну кількість показів за фіксованою вартістю в необхідний період з гарантованим позиціонуванням і форматами розміщення на медійному ресурсі, при використанні RTB купується кожен конкретний показ, не прив'язаний до місця розміщення.

RTB – метод закупівлі реклами в інтернеті з використанням автоматизованих систем (роботів) і алгоритмів для прийняття рішень про угоду без участі людини на основі соціально-демографічних і поведінкових даних про користувачів, що є в розпорядженні як майданчика, так і рекламодавця. Такий підхід дозволяє поліпшити таргетинг і фактично перейти від закупівлі місць і показів до закупівлі цільової аудиторії [2].

1.2 Виникнення протоколу OpenRTB

Передумовою створення технології RTB стала поява перших рекламних мереж (Ad Networks) в кінці 1990-х рр. Рекламні мережі агрегує рекламні інвентарі (місця для розміщення) безлічі інтернет-майданчиків, тим самим спрощуючи роботу рекламодавцю, позбавляючи його від необхідності підписувати договір з кожним інтернет-майданчиком окремо. Одними з найбільш відомих рекламних мереж є DoubleClick, AdWords, Yahoo, Publisher Network.

Наступним етапом розвитку рекламних мереж стали рекламні біржі (Ad exchanges), на які майданчики виставляли доступний для продажу інвентар. Рішення про покупку реклами приймалося автоматично в режимі реального часу на підставі інформації про моделі поведінки конкретного користувача, часу доби, пристрої, з якого відбувався вихід в мережу, і позиції рекламного оголошення на сайті. Така система відрізнялася від рекламних мереж набагато більшою прозорістю: рекламодавець точно знав, за що і скільки платить [24].

Відповіддю на постійно зростаючу кількість рекламних бірж і обсягів рекламних місць на майданчиках стали системи DSP – платформи попиту. Вони допомагали рекламодавцям вибирати найбільш привабливі пропозиції. Але такі платформи задовольняли потреби тільки з боку рекламодавців. Трохи пізніше виникла аналогічна система – SSP або платформа пропозиції - для майданчиків. Вона дозволила в автоматичному режимі продавати місця під рекламу за максимальною ціною. За допомогою SSP майданчики виходили на безліч рекламодавців.

Незважаючи на автоматизацію процесу пошуку і покупки рекламних місць, покупка інвентарю на різних біржах була витратним і неефективним процесом: не було стандартизованого інтерфейсу налаштування рекламної кампанії, для кожної платформи її доводилося налаштовувати «з нуля»; реклама демонструвалася одному і тому ж цільовому користувачеві кілька разів через різні канали і плата

також стягувалася багаторазово; такий стан справ ускладнював ведення загальної аналітики.

Для вирішення цих проблем в 2010 р компанія IAB (Interactive Advertising Bureau) розробила протокол OpenRTB. Над пілотним проектом працювали представники трьох DSP (DataXu, MediaMath, Turn) і трьох SSP (Admeld, PubMatic, Rubicon Project), його метою стало стандартизувати взаємодія між усіма учасниками ланцюжка. Новий протокол об'єднав не лише біржі, DSP і SSP, але також додав до них систему обробки масивів даних DMP, інструменти аналітики і спеціальні інтерфейси для простого налаштування рекламних кампаній Trading Desk3. Технологія була зустрінута позитивно, і OpenRTB отримав подальший розвиток. Остання версія протоколу 3.0 вийшла в кінці 2018 р [1].

Таким чином, протокол OpenRTB дозволяє стандартизувати зв’язок між постачальником (SSP) та покупцем (DSP). Метою якого є полегшення інтеграції між сторонами.

1.3 Учасники RTB

Supply-Side Platform (SSP) – рекламна мережа, що надає в аукціон місця для показів реклами (Продавець).

Demand-Side Platform (DSP) – рекламна мережа, що надає в аукціон рекламу - відео, банери тощо. (Покупець).

Data Management Platform (DMP) – постачальник даних про користувачів.

1.3.1 Двосторонні ринки

SSP та DSP – це класичні представники двосторонніх ринків. Двосторонні ринки (двосторонні мережі) – мережеві ринки, які мають дві групи користувачів з виникненням мережевих ефектів між ними. В двосторонньої мережі присутні дві категорії користувачів, для яких цілі користування мережею і їх ролі чітко

розрізняються. Представники різних груп пред'являють різні вимоги до функціональності двосторонньої мережі. Але при цьому вони є взаємозалежними один від одного, і саме їх взаємодія і визначає цінність використання ними загальної мережі.

Взаємодія категорій користувачів здійснюється через єдину загальну платформу, яка і забезпечує можливість такої взаємодії. Платформа – це ті продукти і послуги, які дозволяють об'єднати дві групи користувачів. Платформа, як правило, складається з інфраструктури (апаратне і програмне забезпечення), стандартів, які гарантують сумісність між елементами інфраструктури, і правил, які визначають умови здійснення транзакцій, права і обов'язки її користувачів. Присутність платформи покращує ефективність і результативність взаємодії двох сторін в порівнянні з прямою взаємодією без її участі.

В двосторонньої мережі користувачі однієї категорії змінюють ставлення до користування мережею в залежності від числа користувачів іншої категорії. Це явище називається перехресним мережевим ефектом. Кожен член групи користувачів також змінює ставлення до її використання в залежності від числа користувачів тієї ж категорії. Це явище називається одностороннім мережевим ефектом. Мережевий ефект є позитивним, якщо зростання числа користувачів підвищує цінність користування мережею, і негативним в зворотному випадку. Перехресні мережеві ефекти у двосторонній мережі зазвичай позитивні, але можуть бути і негативними. Односторонні мережеві ефекти можуть бути як позитивними, так і негативними[4].

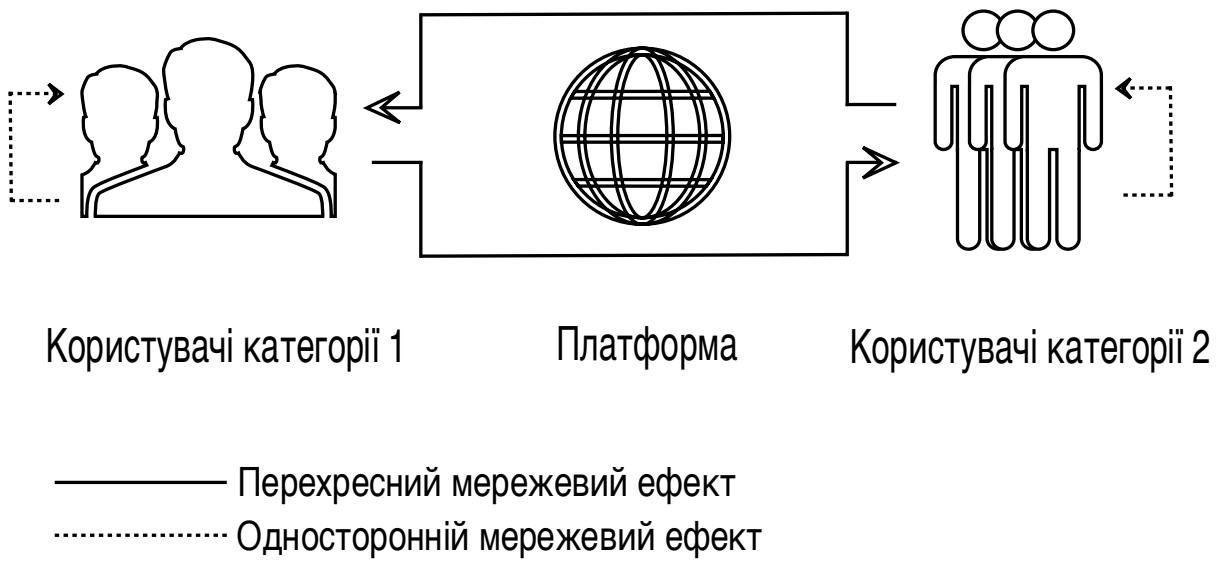


Рисунок 1.1 – Мережеві ефекти двосторонніх мереж

Платформна економіка надає комплексні типові рішення для взаємодії між користувачами, включаючи комерційні транзакції. Як правило, платформи забезпечують можливість використовувати специфічні рішення, пов'язані з ними послуги, рекламну підтримку тощо, без необхідності самостійної розробки або придбання відповідних патентів. Поняття «платформи» в економічних роботах з'явилося в 1990-х роках. Першою науковою публікацією, яка розглядала платформну бізнес-модель, була стаття Jean-Charles Rochet and Jean Tirole Platform Competition in Two-Sided Markets. Спочатку поняття «платформа» відносилося головним чином до онлайн-сервісів, що об'єднував постачальників послуг з клієнтами, таким як Uber, Airbnb і ін., Але згодом отримало більш широкий зміст [4].

Платформа – підприємство, що забезпечує взаємодію між сторонніми виробниками та споживачами. Вона надає відкриту інфраструктуру для учасників

і встановлює правила. Основна задача платформи – створити зв'язок між користувачами та створити обмін товарами або соціальною валютою, тим самим створити цінність усіх учасників [46].

Умови розвитку платформ

1) Висока частка технології в вартості товару по відношенню до інших складових, наприклад, сировину, енергія, транспорт. Основна мета платформ – поширення технологій. Зниження пов’язаних з цим витрат – драйвер розвитку платформних рішень.

2) Високий рівень конкуренції змушує компанії знижувати показник time to market – час виведення нового продукту на ринок. Висока вартість тимчасового ресурсу. Платформа забезпечує можливість швидкого виведення на ринок нових продуктів і послуг, досягти високої адаптивності до потреб клієнтів.

3) Мережевий ефект двостороннього ринку, який виникає в ситуації, коли дві групи користувачів (зазвичай творці продукту і споживачі) створюють додану вартість друг для друга, отримуючи взаємовигідні переваги. Високий розвиток мереж дає можливість залучити велику кількість різних ринкових гравців (користувачі, виробники, сервіси та ін.), тим самим примножуючи можливості створення цінності.

4) Високий ступінь розвиненості інфраструктури (транспорт, комунікація, енергія) дає можливість платформі бути повсюдною, екстериторіальною [46].

1.3.2 Мережевий ефект

Мережевий ефект – ефект в економіці і бізнесі, який користувач товару або послуги додає до цінності цього продукту або послуги для інших користувачів. Класичним прикладом прояву мережевого ефекту є телефонізація. Чим більше

користувачів володіють телефонами, тим ціннішим є участь в мережі для кожного її користувача. Цінність зростає при зростанні кількості абонентів, так як підключення кожного додаткового абонента означає можливість зв'язатися з великою кількістю людей. Це приклад позитивної екстерналії (вплив ринкової транзакції на третіх осіб, не опосередкований ринком). Характер позитивного мережевого ефекту виражається законом Меткалфа, який стверджує, що цінність мережі зростає пропорційно квадрату числа її користувачів. Вираз «мережевий ефект» часто застосовується в позитивному сенсі. Однак мережеві екстерналії можуть бути і негативними, наприклад, викликаючи уповільнення роботи комп'ютерної мережі або пробки на дорогах [46].

Мережевий ефект стає значним після того, як досягається певне число користувачів продукту або послуги, зване критичною масою. У цій точці цінність, що отримується від використання продукту або послуги, більше або дорівнює заплаченої ціною. Оскільки цінність визначається числом користувачів, після досягнення критичної маси залучення клієнтів стає набагато більш простим завданням. Нові користувачі будуть сприймати цінність послуги або товару більшої, ніж його ціна, через позитивних мережевих екстерналій. Ключовим завданням для бізнесу, що працює на ринку з наявністю мережевих ефектів, є залучення користувачів для досягнення їх критичної маси [53].

1.3.3 Demand-Side Platform

За допомогою DSP рекламодавець покупає цільову аудиторію, а не конкретні рекламні місця. У той момент, коли користувач завантажує сторінку сайту, RTB-система проводить аукціон на показ реклами саме для цього користувача. Увесь процес від першого запиту до відображення реклами займає менше секунди (у більшості випадків). За цей час DSP оброблює запит (чи підходять параметри для

показу реклами цьому користувачу) та, проводячи аукціон, видає найкращий результат.

1.3.4 Supply-Side Platform

SSP дозволяє власникам сайтів монетизувати їх. Основною задачею є збір даних про користувачів, які відвідують сайт, показувати найбільш підходящу рекламу для них. Збирати данні про користувачів для підвищення ефективності аналізу користувачів.

1.3.5 Data Management Platform

DMP – платформа для керування даними про користувачів. Виділяють три види отриманих даних: first-party – дані, які отримані за допомогою власних ресурсів рекламодавця, second-party – інформація, яка зібрана завдяки попереднім рекламним компаніям, third-party – дані, які були надані іншими платформами.

1.4 Послідовність процесу RTB

Торги відбуваються наступним чином. У момент, коли користувач заходить на сайт, підключений до RTB, сайт за допомогою SSP миттєво повідомляє системі про готовність показати рекламу. Також він передає свої технічні характеристики (формат доступного рекламного місця, адреса сторінки і т. п.) і анонімні дані про користувача, які дозволяють віднести користувача до певної соціально-демографічні групи, визначити його інтереси і навіть рівень доходу [25]. Таким чином, рекламодавцю пропонується не покупка рекламного місця, а показ оголошення конкретним користувачам в виділеної цільової аудиторії.

Отримавши ці дані, RTB-система передає їх учасникам аукціону. DSP-системи, проаналізувавши отриману інформацію, визначають претендентів на показ, спираючись на вимоги рекламодавців, зазначені при розміщенні реклами, і

дані про користувача. Після того, як ставки зроблені, RTB-система вибирає переможця, який отримує право показати свою рекламу. За показ переможець платить не свою ціну, а ставку найближчого суперника плюс мінімальний крок торгів [3].

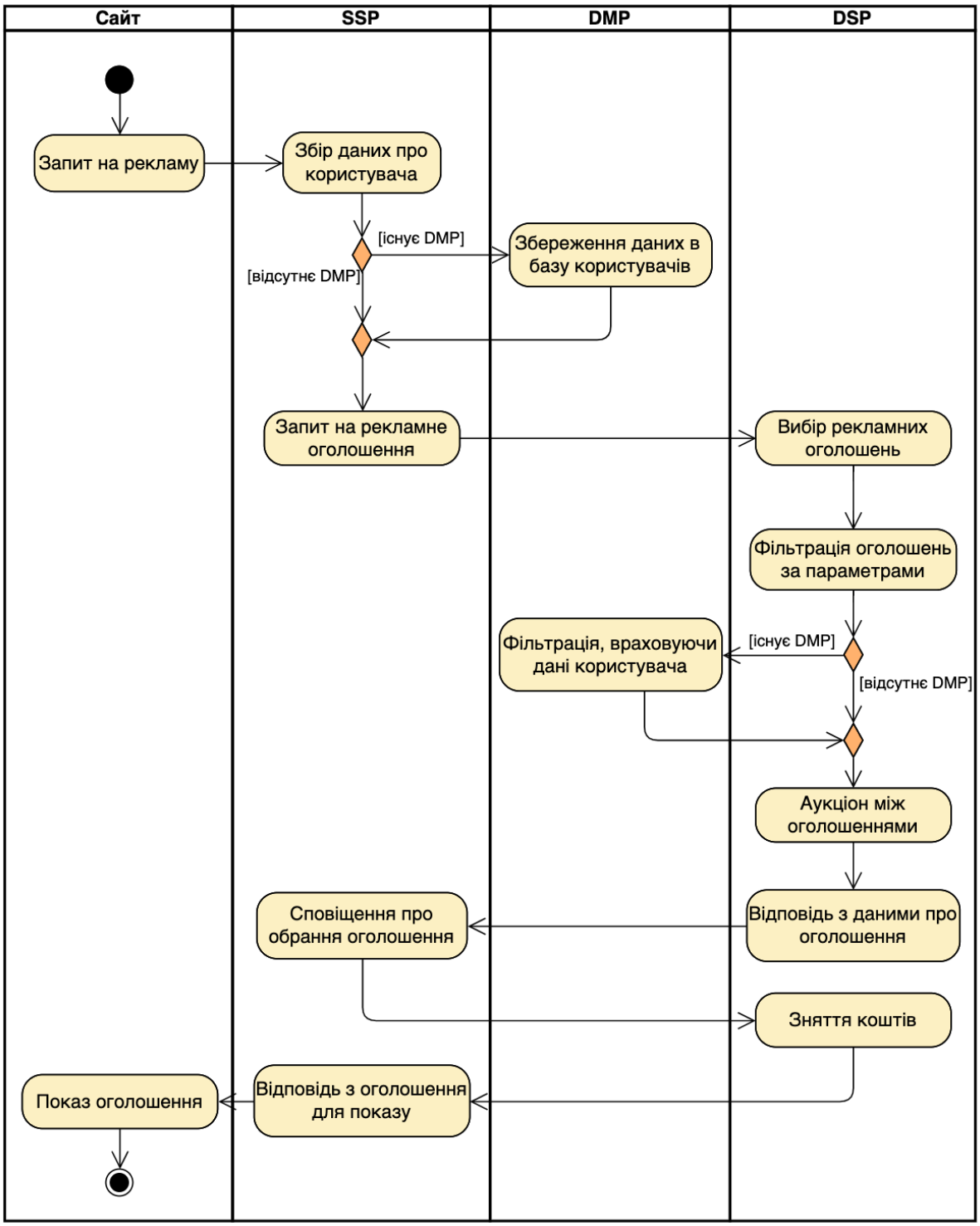


Рисунок 1.2 – Діаграма діяльності

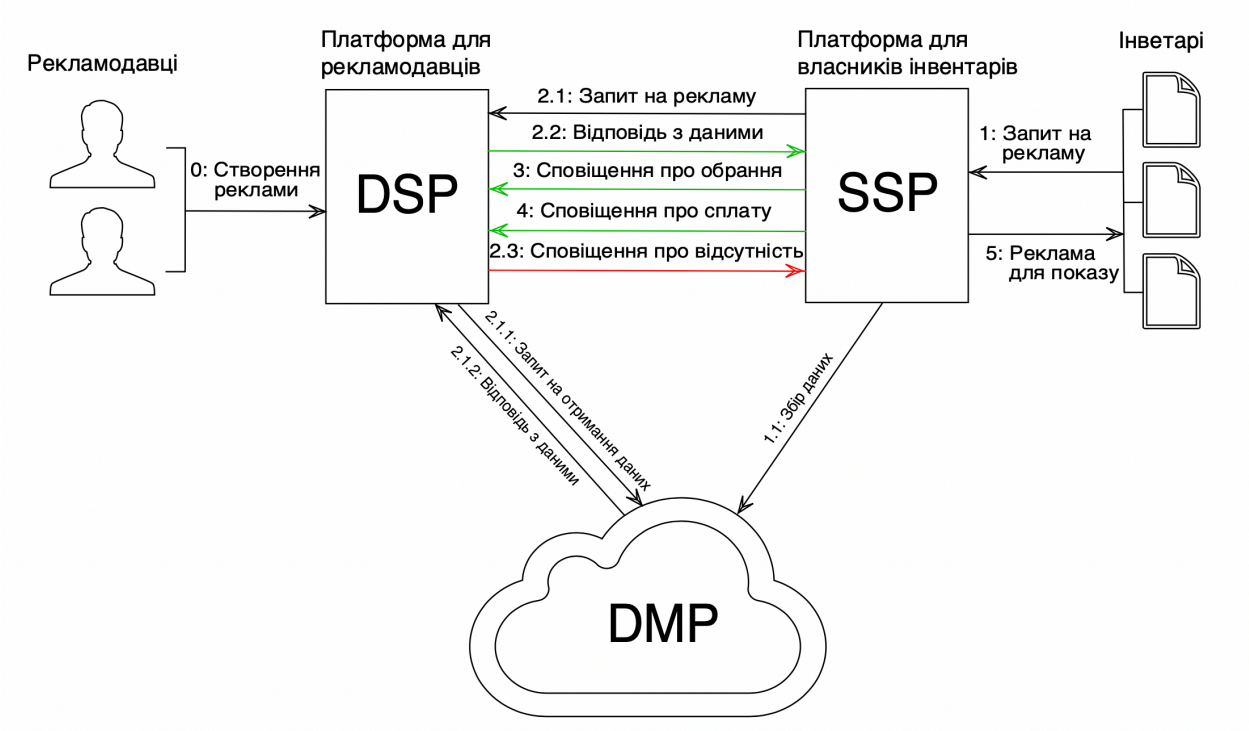


Рисунок 1.3 – Послідовність процесу RTB

1.4.1 Алгоритм роботи RTB

1) Рекламодавець створює на платформі DSP рекламу для відображення.

2) Користувач вводить в рядок браузера назву сайту, підключеного до рекламної RTB мережі. Сайт перед завантаженням сторінки відправляє мережі, до якої він підключений, запит на показ банера, а також дані про користувача.

2.1) Якщо мережа збирає дані про користувача, завантажує їх в DMP.

3) SSP виставляє торг - інформацію про сайт і користувача системам рекламодавців, відправляючи запити на всі партнерів DSP.

3.1) SSP відправляє запит на отримання реклами (Bid Request).

3.1.1) DSP, отримавши запит, обробляють його, та фільтруючи усю рекламу, яку мають. За бажанням, якщо є така

можливість, додатково відправляє запит у DMP, для більш точного таргетингу.

3.1.2) Якщо є DMP, то DSP отримує додаткову інформацію про користувача для якіснішої фільтрації.

3.2) SSP отримує відповідь (Bid Response) та обробляє інформацію від всіх рекламодавців, які відповіли на запит.

3.3) У разі відсутності підходящої реклами на показ за протоколом відповідь DSP буде пустим.

4) Вибирає найкращий варіант та сповіщає про це DSP (Win Notice).

5) Також окремим запитом SSP може сповістити DSP про успішність відображення реклами та зняття коштів.

6) Завантажується сторінка з отриманим оголошенням.

1.5 Основні принципи OpenRTB

Зазвичай OpenRTB розглядається як взаємодія між SSP (Supply side platform, продавець (постачальник) рекламних місць) та DSP (Demand Side Platform, покупець рекламних місць). OpenRTB дозволяє стандартизувати відповіді (bid responses) DSP на запити (bid requests) SSP. У результаті роботи SSP отримає декілька відповідей і визначить переможця на основі певних правил аукціону. Система OpenRTB дозволяє повідомляти усім покупцям про те, чи буде показана їх реклама та за якою ціною або сповістити про те, що на даний запит їх реклама не буде показана.

Однак сьогодні екосистема стала набагато складнішою. Власники майданчиків навчилися контролювати процес показу реклами, обираючи найбільш вигідні для себе. Також стало звичним мати більше одного посередника в ланцюжку взаємодії між рекламодавцями і майданчиками [1].

1.6 Архітектура RTB

1.6.1 Рівні протоколу RTB

Для сприяння повторному використанню об'єктів у різних специфікаціях та для того, щоб різні частини специфікації розвивалися незалежно один від одного існує такий розподіл на рівні: Рівень-1 переміщує байти між сторонами, Рівень-2 виражає мову цих байтів, Рівень-3 вказує транзакцію за допомогою цієї мови, а Рівень-4 описує товари, які використовуються в транзакції. Рівні протоколу зображені на рисунку 1.4.

Таблиця 1.1 – Рівні протоколу RTB

Рівень-1: Домен	Об’єкт, який описує предмет транзакції
Рівень-2: Транзакції	Об’єкт, подія чи макрос, які виконують функції з об’єктом домену
Рівень-3: Формат	Представлення. Метод та шифрування даних
Рівень-4: Транспорт	Протокол повідомлень для комунікації між об’єктами

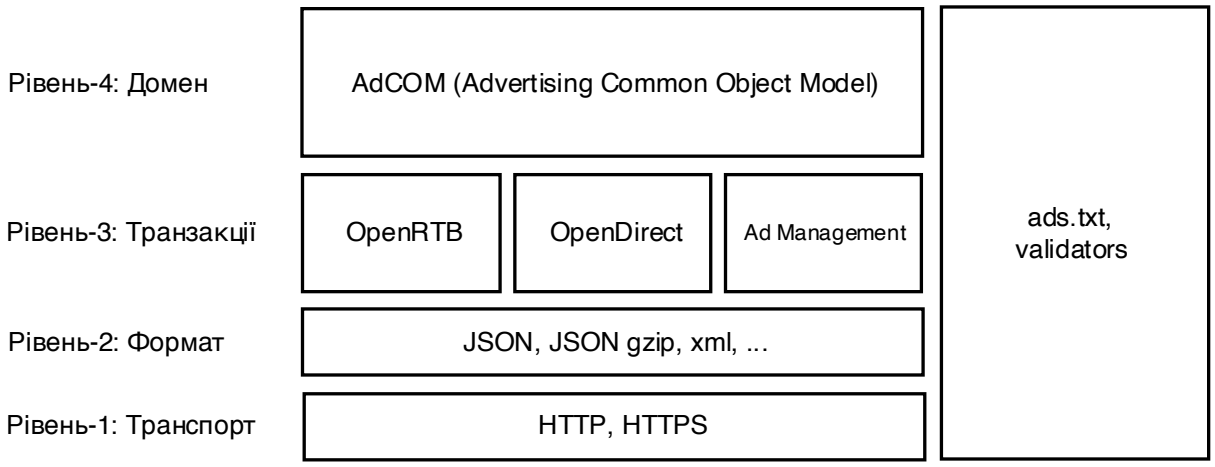


Рисунок 1.4 – Рівні протоколу RTB

1.6.2 Рівень-1: Транспортний

Комунікації

Базовим протоколом взаємодії між SSP та DSP є протокол HTTP. Зокрема, HTTP POST використовується між обміном запитів (bid request, bid response), оскільки за допомогою даного типу передачі даних можна передати набагато більший об’єм даних, ніж HTTP GET. Повідомлення про події можуть бути POST або GET на розсуд платформ.

Відповідь від сервера, що повертає вміст (пропоновану рекламу на показ), повинен повертати код HTTP 200. Якщо DSP не може запропонувати реклами на певний запит, то відповідь має бути пустою з кодом HTTP 204. Відповідь на недійсний запит (наприклад, запит, що містить неправильні або пошкоджені дані) повинен повертати HTTP 400 без вмісту [1].

Також платформа SSP може висувати додаткові вимоги щодо швидкості відповіді DSP. Так стандартним значенням очікування відповіді є 300ms, але SSP може знизити це значення через відповідний параметр у запиті [27, 33].

Безпека

Згідно з протоколу OpenRTB v3.0, безпека транспортного рівня (TLS) є обов’язковою, і тому всі з'єднання, над якими працює протокол OpenRTB, повинні бути HTTPS.

DSP має відповісти SSP на запит з одним із перелічених в таблиці 1.2 статусів.

Таблиця 1.2 – Статуси відповідей

Код	**Назва**	**Опис**
200	OK	Запит був успішним
400	Поганий запит	Запит не може бути розпізнаний

Продовження таблиці 1.2

401	Неавторизований	Запит не містить коректної інформації про авторизацію
404	Не знайдено	Ресурс не знайдено
429	Забагато запитів	DSP має багато (перевищує встановлене значення) запитів для обробки
500	Помилка внутрішньої служби	DSP зіткнувся з технічними труднощами

1.6.3 Рівень-2: Формат

Представлення

JSON (JavaScript Object Notation) – це формат за замовчуванням для bid request та bid response. JSON був обраний за його поєднання простого розуміння змісту та відносної компактності.

Необов'язково, обмін може також пропонувати двійкові представлення (наприклад, стиснуті JSON, ProtoBuf, Avro тощо), які можуть бути більш ефективними щодо часу передачі та пропускної здатності.

Bid request визначає представлення у вигляді mime типу, використовуючи HTTP-заголовок Content-Type. Mime тип для стандартного представлення JSON - це application / json. Формати bid request та bid response мають співпадати. (Content-Type: application/json).

Кодування

Переважно у bid request та bid response використовують стиснення даних при передачі. Стиснення значно зменшує розмір переданих даних і тим самим економить пропускну здатність мережі.

Стиснення можна ввімкнути у bid response за допомогою стандартних механізмів HTTP 1.1. Більшість веб-серверів вже підтримують gzip стиснення

вмісту відповідей і такий варіант на даний момент є найкращим. Якщо SSP планує приймати відповіді у стислому варіанті, то у запиті необхідно встановити стандартний заголовок HTTP 1.1 Accept-Encoding. Використовуване значення кодування має бути gzip. (Accept-Encoding: gzip).

Якщо сервер DSP правильно налаштований на обробку запитів з кодуванням, то він автоматично відповість bid response у закодованому вигляді gzip. Це буде показано за допомогою стандартного заголовка контенту HTTP 1.1. (Content-Encoding: gzip) [44].

Щоб увімкнути компресію на запит заявки, слід спочатку узгодити SSP та DSP, що це підтримується. Це аналогічно, коли використовується спеціальний формат даних, оскільки обмін повинен знати який формат, так і кодування, перш ніж надсилати запити. Якщо джерело попиту підтримує його, біржа повинна вказувати, що вона надсилає запит на стиснення gzip, встановивши заголовок HTTP 1.1 Content-Encoding. Використовуване значення кодування має бути gzip.

Якщо цей заголовок не встановлений, передбачається, що вміст запиту не закодовано. У HTTP 1.1 заголовок Content-Encoding зазвичай використовується лише для контенту відповіді. Однак, використовуючи цей заголовок і для вмісту запиту, ми можемо вказати, що запит стискається незалежно від використовуваного формату даних. Це корисно, оскільки навіть формати бінарних даних можуть скористатися стисненням [44].

1.6.4 Рівень-3: Транзакції

Рівень транзакцій є серцем протоколу торгів у режимі реального часу, що включає OpenRTB. Він визначає протокол комерції між біржою та її учасниками, або, загалом, посередником ланцюжка поставок (SSP) та його інтегрованими джерелами попиту (DSP) [44].

1.6.5 Рівень-4: Домен

Доменний рівень визначає об'єкти, над якими працює операційний рівень. У звичайному рекламному аукціоні запит на торги містив би доменні об’єкти у двох місцях. Перше, загальний запит міститиме об’єкти домену, які описують контекст продажу, такий як сайт або додаток, пристрій та користувача. По-друге, кожен предмет, що пропонується до продажу, міститиме об'єкти домену, які визначають товар, такі як дані про покази та розміщення, технічні характеристики та обмеження. Кожний bid response міститиме об’єкти домену, який визначає медійну інформацію яка буде доставлена користувачу, якщо буде виграно аукціон.

Оскільки версія специфікації рівня домену може змінюватися незалежно від рівня транзакцій, кореневий об'єкт на рівні транзакцій включає специфікацію домену та інформацію про версію. Це також важливо, оскільки підтримка різних версій буде змінюватися з часом залежно від SSP та DSP [45].

AdCOM – колекція класів з різними відносинами між ними. Ці класи належать трьом основним групам: Media, Placement, Context.

Media – група класів, які визначаються параметри самої реклами (зміст, метадані для коректного відображення, обмеження, відстеження подій, аудит якості).

Placement – група класів, які містять параметри місця, де буде розміщена реклама (розміри місця, які mime типи підтримуються, позиція на сайті та різні обмеження для реклами).

Context – група класів, які представляють поняття, які взаємодіють, представляють, обмежують та інші, які належать до параметрів impressions (показів). Включають параметри користувача та його пристрою, локації, місце показу реклами (сайт, додаток), категорія вмісту (контент).

Взаємодію Media з Placement та Context зображено на рисунку 1.5.

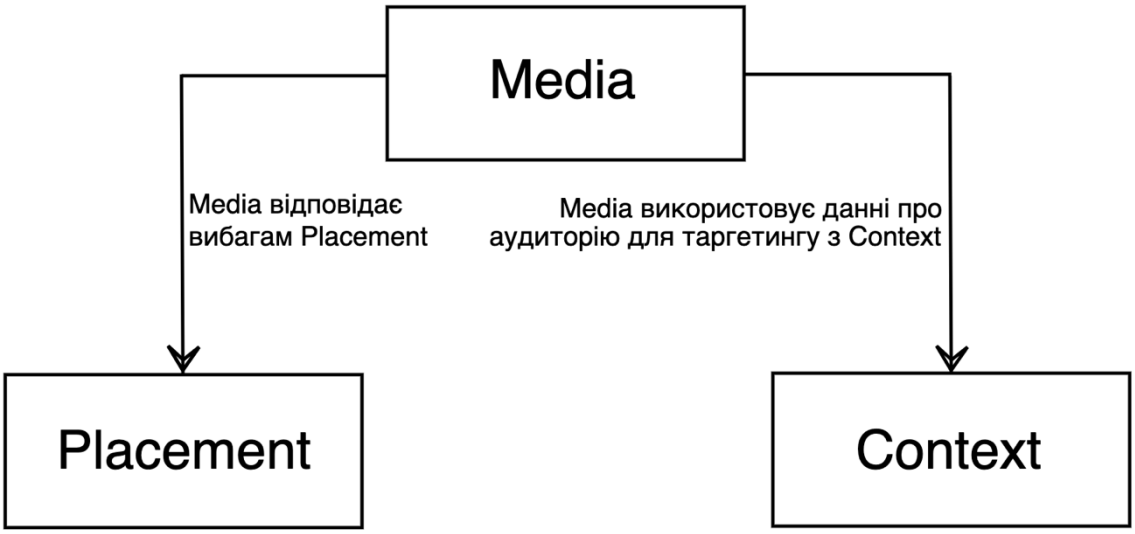


Рисунок 1.5 – Взаємодія Media з Placement та Context

1.7 Advertising Common Object Model

Стандартизація взаємодій платформ призвела до виникнення об'єктної моделі Advertising Common Object Model (AdCOM), яка описує структуру Media, Placement та Context та яке місце вони займають у OpenRTB. OpenRTB більше зосередилось на фактичній транзакції з медіа-комерцією (параметри передачі даних, deals – певні домовленості між платформами), тоді як спільні для усіх дані (оголошення, місця, користувачі, пристрої, сайти) перетворюються у власну специфікацію для багаторазового використання.

Окрім надання модульності для вигідних специфікацій на додаток до OpenRTB, AdCOM прагне вирішити інші бізнес-проблеми реклами в інтернеті. Наприклад, власники сайтів на даний момент мають обмежену можливість контролювати типи оголошень, які вони розміщують, для підтримки стандартизації. Також, це дозволяє контролювати появу небажаної реклами (має велике навантаження, зловмисне програмне забезпечення тощо). Ці та інші

проблеми призводять до поганого та потенційно згубного досвіду користувачів, зменшення довіри, встановлення блокаторів оголошень (Ad Blocks), погіршення монетизації власників сайтів та додатків, посилення конкуренції з боку рекламодавців для досягнення бажаної аудиторії.

1.7.1 MEDIA

Media визначає зовнішній вигляд рекламного оголошення: його тип, наповнення. Основі типи реклами: banner, video, native. Вони застосовуються частіше за все. Також існують event та audio. Діаграма з’язків в групі Media представлена на рисунку 1.6.

Ad – включає загальні метадані реклами та специфічні дані для кожного типу реклами. Атрибути: id, adomain, bundle, iurl, cat, cattax, lang, attr, mrating, init, lastmod, display, video, audio, audit, ext.

Display – об’єкт, який містить додаткову інформацію для відображення реклами. Декілька атрибутів відповідають за визначення параметрів реклами: banner – просте зображення, native – нативна реклама, adm – реклама у вигляді HTML, curl – посилання на рекламу. Для запобігання конфлікту одночасно має використовуватися тільки один параметр. Атрибути: mime, api, ctype, w, h, wratio, hratio, priv, adm, curl, banner, native, event, ext.

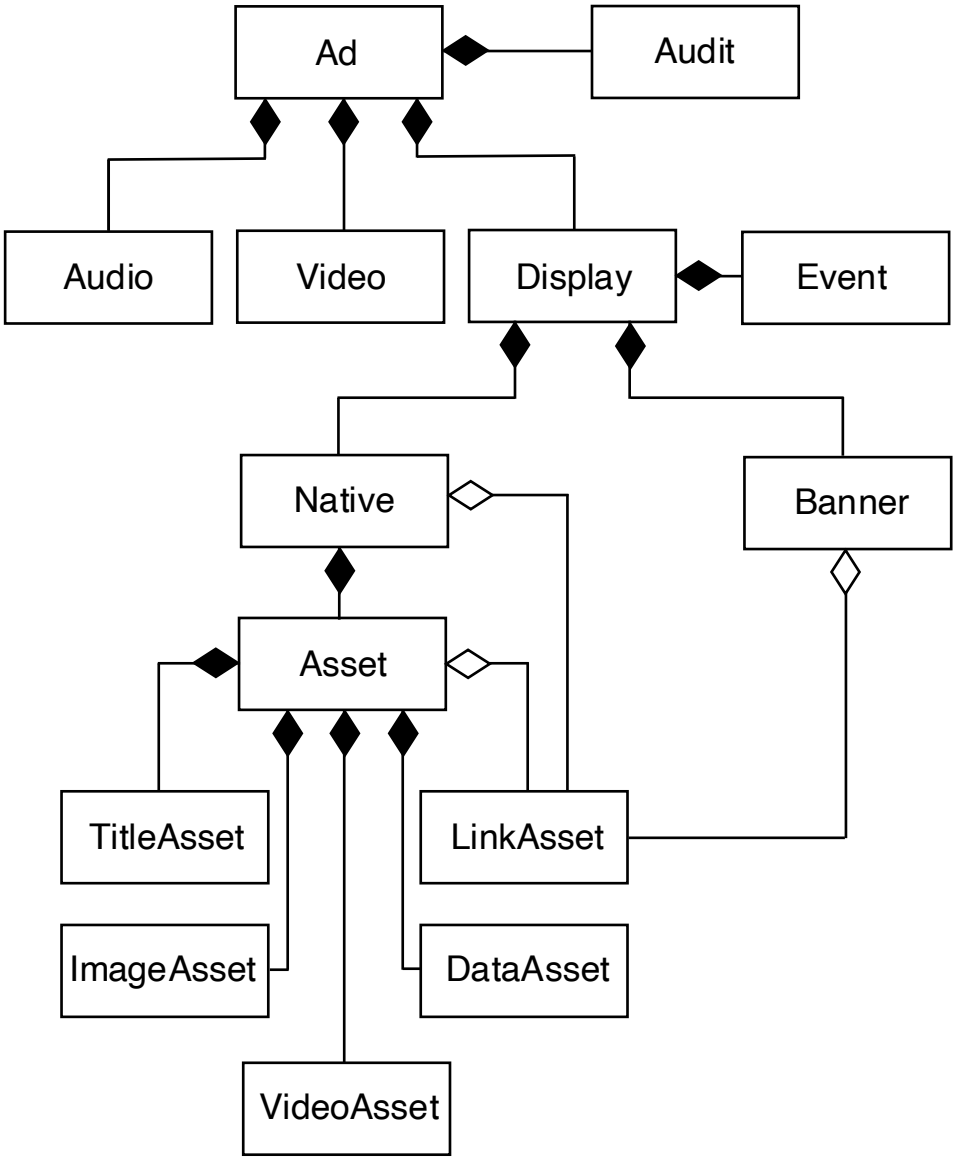


Рисунок 1.6 – Діаграма класів групи Media

Banner – об’єкт, який описує банерний тип реклами. Атрибути: img, link, ext.

Native – об’єкт, який описує нативний тип реклами. Атрибути: link, asset, ext.

Asset – об’єкт, який є контейнером для кожної нативної реклами. Атрибути: id, req, title, image, video, data, link, ext.

LinkAsset – об’єкт, який ідентифікує нативну рекламу за посиланням та відслідковування натисків на цю рекламу. Атрибути: url, urlfb, trkr, ext.

TitleAsset – об’єкт, який ідентифікує нативну рекламу за заголовком. Атрибути: text, len, ext.

ImageAsset – об’єкт, який ідентифікує нативну рекламу за картинкою. Атрибути: url, w, h, type, ext.

VideoAsset – об’єкт, який ідентифікує нативну рекламу за відео. У даному випадку в adm використовується розмітка VAST для передачі контенту. Атрибути: adm, curl, ext.

DataAsset – об’єкт, який ідентифікує нативну рекламу за даними. Використовується для таких елементів, як назва бренду, рейтинг, кількість відгуків, кількість завантажень, ціна, кількість. Актибути: value, len, type, ext.

Event – об’єкт, який визначає тип події, яку хочу відслідковувати рекламодавець або платформа, яка надає місця для розміщення реклами. Атрибути: type, method, api, url, cdata, ext.

Video – об’єкт, який надає додаткову інформацію про відео рекламу. Атрибути: mime, api, ctype, dur, adm, curl, ext.

Audio – об’єкт, який надає додаткову інформацію про аудіо рекламу. Атрибути: mime, api, ctype, dur, adm, curl, ext.

Audit – об’єкт, який містить інформацію про дозвіл реклами на показ. Використовується при скануванні програмою на загрози або інші перевірки на якість. Атрибути: status, feedback, init, lastmod, corr, ext. Значення, які може приймати статус: 1 – «Чекає перевірки», 2 – «Попередньо схвалено», 3 – «Схвалено», 4 – «Відхилено», 5 – «Змінено, потребує перегляду», 6 – «Термін дії закінчився».

1.7.2 PLACEMENT

Placement відповідає за визначення параметрів місця, де буде розташовано оголошення. Діаграма з’язків в групі Placement представлена на рисунку 1.7.

Placement – об’єкт, який представляє властивості місць, де буде розміщена реклама та необхідні характеристики про рекламу, для успішного показу. Містить підтипові об’єкти (Video, Audio, Display). Атрибути: tagid, ssai, sdk, skdver, reward, wlang, secure, admx, curlx, display, video, audio, ext.

DisplayPlacement – об’єкт, який містить повідомлення про те, чи може бути розміщена реклама на певному місті. Атрибути: pos, instl, topframe, ifebust, clktype, ampren, ptype, context, mime, api, ctype, w, h, unit, priv, displayfmt, nativefmt, event, ext.

DisplayFormat – об’єкт, який містить інформацію про дозволені параметри реклами. Атрибути: w, h, wratio, hratio, expdir, ext.

NativeFormat – об’єкт, який містить інформацію для розміщення нативної реклами. Атрибути: asset, ext.

AssetFormat – об’єкт, який містить інформацію про дозволені параметри реклами. Атрибути: id, req, title, img, video, data, ext.

TitleAssetFormat – об’єкт, який містить параметри заголовку нативної реклами. Атрибути: len, ext.

ImageAssetFormat – об’єкт, який містить параметри зображення нативної реклами. Атрибути: type, mime, w, h, wmin, hmin, wratio, hratio, ext.

DataAssetFormat – об’єкт, який містить параметри даних нативної реклами. Атрибути: type, len, ext.

EventSpec – об’єкт, який містить тип подія, яку треба відслідковувати та за допомогою яких методів. Атрибути: type, method, apo, jstrk, wjs, pxtrk, wpx, ext.

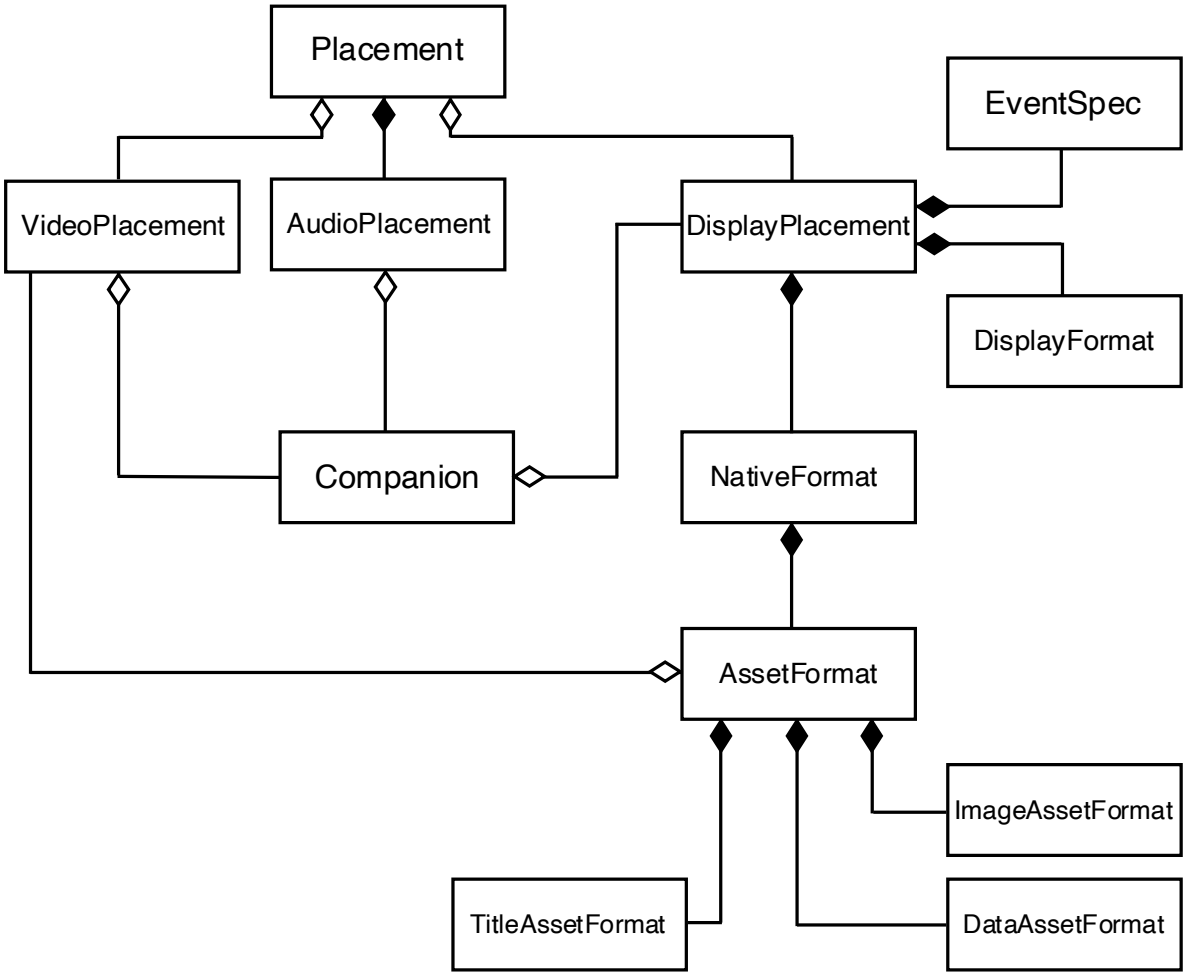


Рисунок 1.7 – Діаграма класів групи Placement

VideoPlacement – об’єкт, який містить інформацію про місце, де може бути розміщена відео реклама та додаткову інформацію про дозволені параметри реклами. Атрибути: ptype, pos, delay, skip, skipmin, skipafter, playmethod, playend, clktype, mime, api, ctype, w, h, unit, mindur, macdur, maxext, minbitr, maxbitr, delivery, maxseq, linear, boxing, comp, comptype, ext.

AudioPlacement – об’єкт, який містить інформацію про місце, де може бути розміщена аудіо реклама та додаткова інформація про дозволені параметри реклами. Атрибути: delay, skip, skipmin, skipafter, playmethod, playend, feed, nvol,

mime, api, ctype, mindur, maxdur, maxext, minbitr, maxbitr, delivery, maxseq, comp, comptype, ext.

Companion – об’єкт, який використовується в місцях, які призначені для розміщення відео та аудіо рекламу. Атрибути: id, vcm, display, ext.

1.7.3 CONTEXT

Context відповідає за визначення параметрів користувача, якому буде показано рекламне оголошення. Діаграма з’язків в групі Context представлена на рисунку 1.8.

Publisher – об'єкт, який описує видавництво, яке розміщує рекламу. Атрибути: id, name, domain, cat, cattax, ext.

Content – об’єкт, який описує заміст реклами для показу. Атрибути: id, episode, title, series, season, artist, genre, album, isrc, url, cat, cattax, prodq, context, rating, urating, mrating, keywords, live, srcrel, len, lang, embed, producer, data, ext.

Producer – об’єкт, який описує предмет, який рекламується. Атрибути: id, name, domain, cat, cattax, ext.

DistributionChannel – абстракція різних типів об’єктів або каналів, через які поставляється реклама (веб-сайти, мобільні додатки та різні системи, які використовуються для реклами на вулиці (digital out-of-home)). Атрибути: id, name, pub, content.

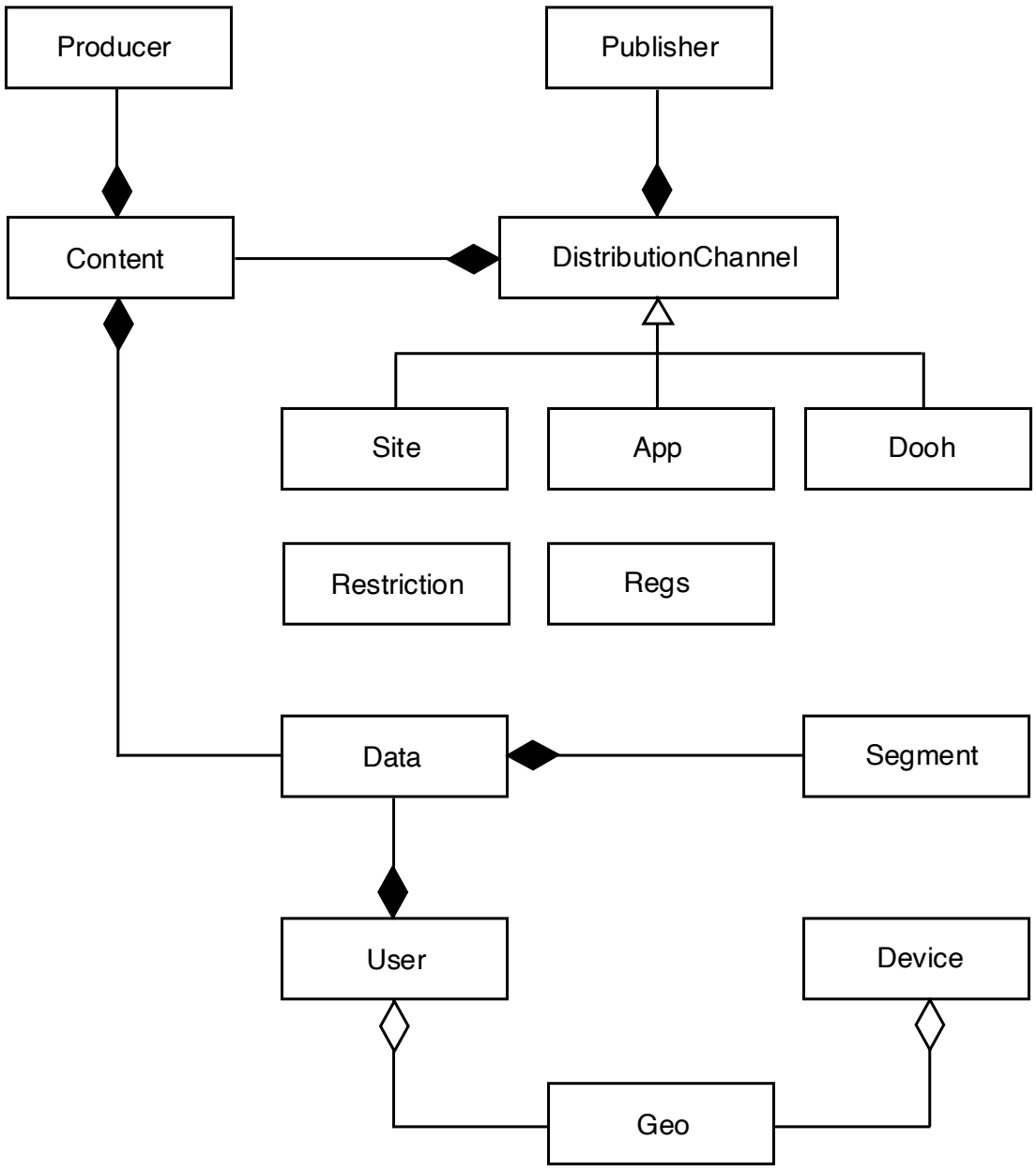


Рисунок 1.8 – Діаграма класів групи Context

Site – об’єкт, який визначає рекламу, яка підтримується веб-сайтами. Атрибути: domain, cat, sectcat, pagecat, cattax, privpolicy, keywords, page, ref, search, mobile, amp, ext.

App – об’єкт, який визначає рекламу для мобільних додатків. Атрибути: domain, cat, sectcat, pagecat, cattax, privpolicy, keywords, bundle, storeid, storeurl, ver, paid, ext.

Dooh – об'єкт, який визначає рекламу на вулиці. Атрибути: venue, fixed, etime, dpi, ext.

User – об’єкт, який містить інформацію про пристрій користувача, де буде розміщена реклама. Атрибути: id, buyeruid, yob, gender, keywords, consent, geo, data, ext.

Device – об’єкт, який містить інформацію про характеристики пристрою, який використовує користувач. Атрибути: type, ua, ifa, dnt, lmt, make, model, os, osv, hwv, h, w, ppi, pxratio, js, lang, ip, lang, ip, ipv6, xff, iptr, carrier, mccmnc, mccmncsim, contype, geofetch, geo, ext.

Geo – об’єкт, що має сукупність методів для вказання геолокації, де буде показана реклама. Атрибути: type, lat, lon, accur, lastfix, ipserv, country, region, metro, city, zip, utcoffset, ext.

Data – об'єкт, який містить додаткові дані (про користувача, про контент). Атрибути: id, name, segment, ext.

Segment – об’єкт, який містить основні значення для Data у вигляді ключ-значення. Атрибути: id, name, value, ext.

Regs – об’єкт, який містить правову інформацію. Атрибути: coppa, gdpr, ext.

Restrictions – об’єкт, який містить список обмежень для bid response. Атрибути: bcat, cattax, badv, bapp, battr, ext.

1.8 OpenRTB

Діаграма з’язків OpenRTB, якапредставлена на рисунку 1.9, описує структуру основних дій за протоколом OpenRTB: запити, відповіді, призначення ставок для публічних та приватних аукціонів.

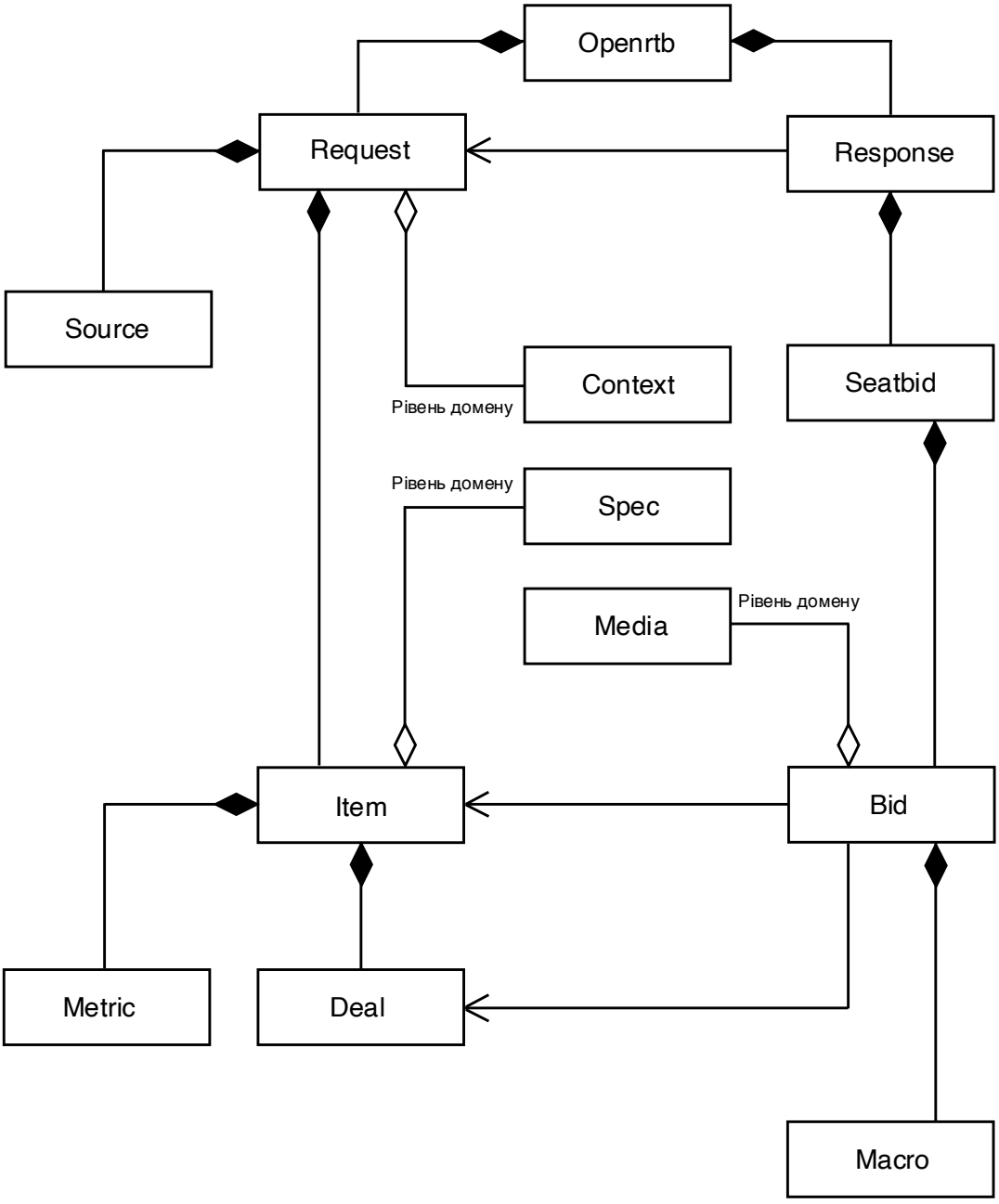


Рисунок 1.9 – Діаграма класів OpenRTB

OpenRTB – об’єкт, який є кореневим для Request (Запиту) та Response (Відповіді). Включає інформацію о версії, яка використовується та дані про 4 рівень на якому базується передача даних. Атрибути: ver, domainspec, domainver, request, response.

BidRequest

Request – об’єкт, який містить усі необхідні параметри для запиту реклами. Атрибути: id, test, tmax, at, cur, seat, wseat, cdata, source, item, package, context, ext.

Source – об’єкт, який містить інформація про джерело транзакції. Атрибути: tid, ts, ds, dsmap, cert, digest, pchain, ext.

Item – об’єкт, який містить перелік одиниць товару для продажу (реклама) як на відкритому так і на приватному ринках. Атрибути: id, qty, seq, flr, flrcur, exp, dt, dlvy, metric, deal, private, spec, ext.

Deal – об’єкт, який містить інформацію для продажу реклами на приватному ринку. Атрибути: id, flr, flrcur, at, wseat, wadomain, ext.

Metric – об'єкт, який містить додаткову інформацію для кожного товару (item). Атрибути: type, value, vendor, ext.

BidResponse

Response – об’єкт, який містить усі параметри для успішного відображення реклами на запит. Для вказання, що реклами на певний запит не існує прийнято відповідати HTTP 204 (пустим об’єктом). Атрибути: id, bidid, nbr, cur, cdata, seatbid, ext.

Seatbid – об’єкт, який містить перелік Bid (реклами, які підпадають під задані параметри у запиті). Атрибути: seat, package, bid, ext.

Bid – об’єкт, який містить інформацію про рекламу. Атрибути: id, item, price, deal, cid, tactic, purl, burl, lurl, exp, mid, macro, media, ext.

Macro – об’єкт, який містить перелік даних у вигляді ключ-значення. При обробці запиту макроси замінюються відповідними значеннями. Значення макросів наведені у таблиці 1.3. Атрибути: key, value, ext.

Таблиця 1.3 – Значення макросів

Макрос	Значення
${OPENRTB_ID}	ID запиту. Атрибут request.id
${OPENRTB_BID_ID}	ID bid. Атрибут response.bidid
${OPENRTB_ITEM_ID}	ID товару, який буде показаний. Атрибут item.id
${OPENRTB_SEAT_ID}	ID назви SSP. Атрибут seatbid.seat
${OPENRTB_MEDIA_ID}	ID заздалегідь зареєстрованого media. Атрибути bid.mid
${OPENRTB_PRICE}	Ціна у домовленій між SSP та DSP валюті
${OPENRTB_CURRENCY}	Валюта, яку використовують SSP та DSP
${OPENRTB_MBR}	MBR (Market Bid Ratio) = ціна закупки реклами / ціна продажу реклами
${OPENRTB_LOSS}	Коди причин втрати

1.9 Інші відповідності значень

Перед початком показу реклами платформи DSP, а також деякі SSP, мають затвердити оголошення рекламодавця. Оголошення може мати один зі статусів, які навереді в таблиці 1.4.

Таблиця 1.4 – Audit Status Code

Значення	Зміст
1	«Очікує розгляду». Аудит реклами не був проведений і очікує перевірки.
2	«Попередньо розглянуто». Аудит реклами не був проведений але вона рекомендована на показ, оскільки відповідає усім необхідним вимогам.
3	«Затверджено». Аудит проведено і реклама затверджена на показ.
4	«Відхилено». Аудит проведено і реклама відхилена.
5	«Змінено, потребує перегляду». Після аудиту реклама була змінена, тому потребує перегляду знову.
6	«Закінчений термін». Позначка ставиться рекламодавцем.

Запит може мати значення мережі, яку використовує користувач. Це дозволяє визначати, яким пристроєм він користується та чи достатня швидкість для завантаження оголошення. Також для більш точного таргетингу може використовуватися тип контерту сайта. Відповідності значень наведені в таблицях 1.5 та 1.6 відповідно.

Таблиця 1.5 – Типи з’єднань

Значення	Зміст
1	Ethernet
2	Wi-Fi

Продовження таблиці 1.5

3	Невідома мобільна мережа
4	Мобільна мережа – 2G
5	Мобільна мережа – 3G
6	Мобільна мережа – 4G
7	Мобільна мережа – 5G

Таблиця 1.6 – Типи контенту

Значення	Зміст
1	Відео
2	Ігра
3	Музика
4	Додаток
5	Текст
6	Інше
7	Невідоме

Для рекламодавця край необхідно знати статистику своїх оголошень. Також її використовують платформи DSP та SSP, порівнюючи власні результати, розраховуються один з одним. Тому існує два метода відслідковування подій (якщо на сайті дозволено javascript – функція, яка надсилає запит, у іншому випадку – картинка розміром в один піксель, яка, при завантаженні, відправляє такий самий запит) [26, 34]. Зокрема, подія відображення оголошення є найважливішою, оскільки саме за неї платять усі учасники.

Таблиця 1.7 – Методи відслідковування подій

Значення	Зміст
1	Image-Pixel: в рекламу вбудовується невидимий піксель, через який відбувається відслідковування необхідних подій (показ, натискання)
2	JavaScript: в рекламу вбудовується JavaScript скрипт, за допомогою якого відбувається відслідковування подій.

Для визначення місця розташування користувача за його ip-адресою може використовуватися різні бази відповідностей адрес та місць. Перелік наведений у таблиці 1.8.

Таблиця 1.8 – Засоби визначення позиції користувача по ip-адресі

Значення	Зміст
1	Ip2location
2	Neustar
3	MaxMind
4	NetAcuity (Digital Envoy)

Місце розміщення реклами для відео відрізняється від звичайних банерних оголошень. Це впливає на саме оголошення: його розмір, якість, впливає на ефективність оголошення. Перелік усіх можливих варіантів наведений у таблиці 1.9.

Таблиця 1.9 – Місце розміщення реклами для відео

Значення	Зміст
1	In-Stream: Показувати відео рекламу під час іншого відео.
2	In-Banner: Відео в банерній рекламі.
3	In-Article: Відео, розміщене в статті або іншій сторінці з великою кількістю тексту.
4	In-Feed: Відео у новосній стрічці.
5	Інше: В інших місцях.

ВИСНОВКИ ДО РОЗДІЛУ 1

У даному розділі розглянуто та описану систему торгів у реальному часі (Real-Time Bidding). У ході опрацювання даного розділу описано бізнес-процеси для оглянутої системи, її платформи-учасники та їх алгоритм взаємодії.

Описані основні рекомендації для платформ по форматам запитів та їх обробці, а також деякі архітектурні рішення від компанії-засновника системи.

Також описані відповідності значень параметрів у запиті за стандартом OpenRTB.

2 МАТЕМАТИЧНА МОДЕЛЬ

В інтернет-рекламі для вибору оголошення, яке буде показуватися на певному місці проводять аукціон. Використовують закритий аукціон двох типів: першої та другої ціни. Інформація про ставки інших учасників невідома і у кожного з них є можливість зробити ставку один раз [29]. Товар (у даному випадку рекламне місце) той, хто запропонував найбільшу ціну, але платить по різному, в залежності від типу аукціону. В аукціоні першої ціни – сплачується найбільша ставка, в аукціоні другої ціни (аукціон Вікрі) – другу найбільшу ціну [7].

Більшість платформ стягують комісію. Як правило, цю комісію перераховує покупець при здійсненні операції, її розмір обчислюється як певний відсоток від ціни угоди (ціна, зазначена в ставці, збільшена на вартість комісії).

Найпопулярнішим серед цих аукціонів в інтернет-рекламі є аукціон другої ціни [31, 52].

2.1 Аукціон другої ціни

Розглянемо аукціон, де продається набір однакових товарів. Учасники торгів можуть взяти участь в аукціоні, оголосивши максимальну ціну, яку вони готові платити за отримання N товарів. Кожен покупець оцінює граничну вартість товару для себе та робить пропозицію. Учасники торгів не можуть бачити ставки інших людей (видно лише в аукціонній системі). Щойно всі торги зроблені, аукціон закривається.

Після цього система аукціону розглядає всі пропозицій і обирається найкраща. Учасник, який зробив успішну заявку, отримує товар. Однак ціна, яку він платить, не є тією сумою, яку він запропонував спочатку, а тією, якою була друга найвища ціна. [13]

Товар дістається учаснику, для якого цінність товару є найвища.

Визначення

Для будь-якого набору товарів на аукціоні $M = \{t_1, \dots, t_m\}$ та набору учасників $N = \{b_1, \dots, b_n\}$, нехай V_N^M – це «суспільна вигода» (суспільство – усі учасники аукціону) від результатів аукціону при певному наборі ставок. Для учасника b_i та товару t_j ставка учасника буде $v_i(t_j)$ [41].

Переможець

Учасник b_i, ставка якого для товару t_j, а саме $v_i(t_j)$ є найбільшою серед інших гравців, виграє аукціон, але платить $V_{N/\{b_i\}}^M - V_{N/\{b_i\}}^{M/\{t_j\}}$, що дорівнює неотриманому доходу залишившихся учасників від його виграшу (виграш визначається іншими учасниками).

Пояснення

Множина учасників-конкурентів для b_i визначається наступним чином: $N/\{b_i\}$. Коли товар t_j доступний для конкурентів, вони можуть досягти капіталу $V_{N/\{b_i\}}^M$. Виграш предмету b_i зменшує набір доступних предметів до $M/\{t_j\}$ і капітал, який можна досягти стає $V_{N/\{b_i\}}^{M/\{t_j\}}$. Таким чином, різниця між двома рівнями капіталу і є втраті упущеною вигодою інших гравців при умові, що гравець b_i отримає товар t_j. Ця величина залежить від заявок учасників-конкурентів і не змінна для учасника b_i [43].

Виграш учасника аукціону, ставкою якого є істина цінність A товару t_j, $v_i(t_j) = A$, отримує найбільший прибуток $A - (V_{N/\{b_i\}}^M - V_{N/\{b_i\}}^{M/\{t_j\}})$.

Доведення

Стратегія аукціону другої ціни полягає в тому, щоб кожен учасник зробив ставку відповідну до цінності, яку принесе йому товар.

Нехай v_i є цінність учасника товару i та b_i – його ставка. Тоді прибуток буде

$$\begin{cases} v_i - \max\limits_{j\,!=i} b_j, & b_i > \max\limits_{j\,!=i} b_i \\ 0, & \text{у всіх інших випадках} \end{cases}$$

Правильною стратегією є робити ставку еквівалентну цінності товару ($b_i = v_i$).

Випадок, коли $b_i > v_i$:

1) Якщо $\max\limits_{j\,!=i} b_j < v_i$, то учасник все одно отримає товар та ціна, яку він заплатить, залишиться незмінною, оскільки він заплатить другу найвищу ставку.

2) Якщо $\max\limits_{j\,!=i} b_j > b_i$, то учасник у будь-якому випадку не отримає товар.

3) Якщо $v_i < \max\limits_{j\,!=i} b_j < b_i$, то учасник отримає товар, але його прибуток буде від'ємним. [7]

Випадок, коли $b_i < v_i$:

1) Якщо $\max\limits_{j\,!=i} b_j > v_i$, то учасник у будь-якому випадку не отримає товар.

2) Якщо $\max\limits_{j\,!=i} b_j < b_i$, то учасник отримає товар у будь-якому випадку і заплатить однакову ціну.

3) Якщо $b_i < \max\limits_{j\,!=i} b_j < v_i$, то учасник втратить можливість отримати товар.

Таким чином, найкращою стратегією є робити ставку, еквівалентну цінності товару. [7]

2.2 Механізм Вікрі в інтернет-рекламі

Аукціон другої ціни (аукціон Вікрі) використовується для продажу рекламних місць на інтернет-платформах. Зокрема, цю модель аукціону використовують Facebook та Google.

Існує декілька моделей продажу рекламних місць: CPM (ціна за 1000 показів), CPC (ціна за клік), CPA (ціна за конверсію). У класичній моделі, плата береться за CPM і на основі цієї вартості вираховується CPC, як витрачена сума на кількість отриманих кліків та CPA, як витрачена сума на кількість отриманих конверсій [30].

Нехай в рекламному блоці K місць (інвентарів). За ці місця конкурують кілька рекламних оголошень. У моделі, коли оплата здійснюється за кліки, важливими параметрами конкуруючих оголошень є їхні ставки, які дорівнюють b_i і ймовірності кліка (CTR) p_i.

Цінність кандидата в цій моделі задається функцією $V(b,p) = b * p$. Найкращі за цінністю K оголошень ідуть на показ. Для i-го гравця маємо $V_i = V(b_i, p_i) = b_i * p_i$.

Правило аукціону для заданої функції цінності $V(b,p)$ та K місць в рекламному блоці звучить так: треба вибрати K оголошень з найбільшим значенням V та з i-го гравця взяти за клік стільки грошей c_i, що цінність $V(\mathrm{c}_i, p_i)$ менша за цінність $V(b_i, p_i)$ його первісної ставки рівно на стільки, на скільки впала би сумарна цінність показаних гравців, якби гравець i не приймав участь в аукціоні [40].

Розглянемо випадок, коли всі позиції однаково хороші, тобто ймовірності кліка оголошень не залежать від позиції.

Для випадку $K = 3$ для обчислення вартості кліка першого оголошення необхідно вирішити наступне рівняння:

$$V(\mathrm{c}_1, p_1) + V(b_2, p_2) + V(b_3, p_3) = V(b_2, p_2) + V(b_3, p_3) + V(b_4, p_4)$$

$$V(\mathrm{c}_1, p_1) = V(b_4, p_4)$$

Тобто для обчислення ціни кліка першого оголошення потрібно зменшити його ставку настільки, щоб його цінність зменшилася до цінності першого непоказаного гравця (в даному випадку - 4-го оголошення). [16]

$$c_1 = b_4 * p_4/p_1$$

Аналогічно це твердження є правдивим і для 2-го, і для 3-го гравців:

$$c_2 = b_4 * p_4/p_2$$

$$c_3 = b_4 * p_4/p_3$$

Таким чином, якщо ймовірності кліка, які беруть участь в аукціоні оголошень рівні (оцінки CTR збігаються), і їх ставки рівні 10, 7, 5, 2, то в показ підуть перші три, і всі вони будуть платити 2 - ціну 4-го оголошення [40].

Властивість правдивості призначення ставок (thruthfulness) аукціону Вікрі у галузі інтернет-реклами означає наступне: для вирішення завдання максимізації свого прибутку рекламодавцю потрібно ставити таку ставку, що в разі, якби ціна, яка списується, дорівнювала в точності виставленій, рекламодавець отримав би нульовий прибуток від кліка в середньому [41]. Для випадку, коли рекламодавець хоче отримувати прибуток з ROI вище деякого заданого значення, йому потрібно ставити мінімальну ставку, при якій досягається необхідний йому ROI [36]. Як з обмеженням так і без обмеження на ROI оптимальна ставка не залежить від ставок інших гравців [37].

Коли у рекламодавця крім обмеження на ROI є фіксований бюджет на рекламу в одиницю часу і це обмеження не фіктивне, а регулярно досягається, то

його алгоритм виставлення оптимальної ставки (який максимізує його прибуток) в аукціоні Вікрі вже не має простого опису [56].

Алгоритм обчислення оптимальної ставки також складний і залежить від ставок конкурентів, коли максимізується не прибуток, а якась комбінація обігу і прибутку.

Випадок, коли клікабельність оголошення залежить від місця розташування інвентаря

Нехай для оголошення i вірогідність кліка на місцях 1, 2, 3 дорівнює відповідно $x_1 * p_i$, $x_2 * p_i$, $x_3 * p_i$, тобто є множина $\{x_1, x_2, x_3\}$ менше 1, яка визначає мультиплікативні поправки до початкової вірогідності кліка (клікабельність місця). Найпоширенішим випадком є розташування місць в порядку зменшення клікабельності ($x_1 > x_2 > x_3$). [9]

Рівняння, яке визначає вартість кліка c_1 певного оголошення має наступний вигляд:

$$x_1 V(c_1, p_1) + x_1 V(b_2, p_2) + x_1 V(b_3, p_3) = x_1 V(b_2, p_2) + x_1 V(b_3, p_3) + x_1 V(b_4, p_4)$$

Підставимо $V_i = V(b_i, p_i) = b_i p_i$:

$$c_1 p_1 = ((x_1 - x_2)V_2 + (x_2 - x_3)V_3 + x_3 V_4)/x_1$$

$$c_1 = ((x_1 - x_2)b_2 p_2 + (x_2 - x_3)b_3 p_3 + x_3 b_4 p_4)/(x_1 p_1)$$

Таким чином, ставка 1-го зменшується рівно на стільки, щоб його цінність $V(c_1, p_1)$ стала дорівнювати середньо-зваженому цінностей оголошень, показаних нижче та одного невидимого оголошення. Ваги в цьому усередненні визначаються клікабельністю позицій [10].

2.3 Трагедія спільного в RTB

Кожен рекламодавець хоче досягти певних результатів (кількості кліків, прибутку) не виходячи за рамки запланованого бюджету. Якщо n рекламодавців в аукціоні і середня ймовірність виграшу кожного з них дорівнює $\frac{1}{n}$, рекламодавці спробують витратити весь бюджет на отримання найкращого результату показників. Однак, така рівновага не є ефективною. Насправді, це призводить до ситуації дуже низького соціального забезпечення серед рекламодавців: вони вичерпують весь свій бюджет, щоб завоювати ту саму корисність, тобто $\frac{1}{n}$ показів і кліків. Набагато краще буде ситуація, якщо рекламодавці витратять $\frac{B}{k}$ бюджету і отримають бажані результати. Особливо, коли $k \to +\infty$, соціальне забезпечення (дохід мінус витрачена ціна) серед рекламодавців максимізується. В такому випадку кожен учасник буде робити ставку, яка дорівнює нулю і переможець, який буде обиратися випадково (оскільки всі ставки однакові) буде платити нуль. Проте, така ситуація ніколи не трапиться через те, що учасники аукціону завжди змагаються між собою, а не домовляються, в противному випадку, такий вид торгівлі не може називатися аукціоном [20]. В такій нестабільній ситуації кожен рекламодавець буде підвищувати ставку, щоб отримати більше корисності з огляду на поточну ринкову ситуацію. Це і називається трагедією спільного [21], що є відображенням багатокористувацьких випадків дилеми в'язня [22] в конкурентній боротьбі рекламних компаній з обмеженим бюджетом в системі RTB.

Рекламодавці витрачають більше грошей на отримання тих же результатів, які вони могли отримати за значно менші кошти, якби домовилися між собою, проте правильне використання механізмів RTB дозволяє підвищити ефективність використання бюджету.

Приклад трагедії спільного в системі RTB наведений в таблиці 2.1.

Таблиця 2.1 – Прилад трагедії спільного в RTB

	Всі інші ставки нижче	Всі інші ставки вище
Вища ставка	$(v - b_{low})/n$	0
Нища ставка	$v - b_{low}$	$(v - b_{high})/n_{high}$

Нехай в аукціоні n учасників і для них доступні дві ставки – b_{low} та b_{high} і найбільша вартість, яку може заплатити рекламодавець – v. $b_{low} < b_{high} < v$. У лівому стовпчику описані дії для рекламодавця. Перший ряд описує можливі дії інших рекламодавців. Значення в таблиці – прибуток, який може отримати рекламодавець у кожній ситуації.

В ситуації, коли усі рекламодавці роблять ставку b_{low}, то ймовірність виграшу становить $\frac{1}{n}$ і прибуток буде $v - b_{low}$. Якщо рекламодавець зробить ставку b_{high}, у той час, коли всі інші зроблять ставку b_{low}, то він точно виграє аукціон і його дохід буде $v - b_{low}$.

В ситуації, коли усі рекламодавці роблять ставку b_{high} і останній робить ставку b_{low}, то він точно програє і не отримає ніякого прибутку. Якщо він зробить ставку b_{high}, то його ймовірність виграшу складе $\frac{1}{n_{high}}$, де n_{high} - кількість учасників, які зробити ставку b_{high}, та отримає прибуток $v - b_{high}$ у разі виграшу. Очевидно, що стратегія високих ставок домінує над стратегією низьких ставок. У випадку, коли усі рекламодавці оберуть стратегію з максимальною ставкою, тобто $n_{high} = n$, що призведе до $(r - b_{high}) / n_{high} < (r - b_{low}) / n$, система досягне рівноваги низького рівня соціального забезпечення [47].

У такому випадку рішенням проблеми може бути встановлення ціни відповідно до витрат та встановлення верхньої границі ставок. Проте, для того щоб

досягти кращих результатів, все одно треба співпрацювати усім рекламодавцям так само, як співпрацюють ув'язнені для досягнення найбільшої нагороди для кожного з них.

2.4 Покращення алгоритму аукціону другої ціни при виборі рекламних місць

На перший погляд використання класичного алгоритму аукціону другої ціни є логічним та очевидним, однак, у розрізі торгів у реальному часі (Real Time Bidding), де за одну секунду можуть проводитися тисячі аукціонів [39], стає питання про монополію одного користувача [35]. Коли один користувач установить ціну, яка значно перевищує ціну інших користувачів, при необмеженому на певному проміжку часу бюджеті, отримаємо такий випадок, що один користувач буде скуповувати всі рекламні місця, у той час як реклама інших користувачів не буде показана. Це може призвести до втрати усіх користувачів, оскільки усі користувачі просто відмовляться брати участь у таких аукціонах, а той, який вказував найбільшу ціну, просто почне платити мінімальну ціну [38]. Таким чином, необхідно зробити так, щоб усі користувачі, незалежно від ціни, яку вони вказали, мали можливість купувати рекламні місця. Рішенням такої проблеми може бути залежність вірогідності [11] виграшу від ціни, яку вказує користувач, але необхідно враховувати, що при такому підході платформа може отримати менше прибутку.

Для порівняння стратегії аукціону з випадковістю виграшу та стратегії, де завжди виграє той, у кого найбільша ціна, скористаємось формулою підрахунку надходжень грошових платежів через однаковий проміжок часу (ануїтет). [12]

Вартість ануїтету з постійними платіжками обчислюється за формулою:

$$P = R * \frac{1 - (1 + d)^{-n}}{i}$$

де

d – процентна ставка для певного періоду,

n – кількість періодів,

R – періодична виплата.

Нехай

R_{max}– сума доходу за місяць від аукціону по найбільшій ціні;

$$R_{max} = \sum_{j=1}^{N} b_j^{max}$$

$$b_j^{max} = \max_i b_i * k$$

Де b_i – ставка кожного гравця,

k – коефіцієнт, який визначає платформа.

Тоді формула підрахунку вартості ануїтету буде мати наступний вигляд:

$$P_{max} = R_{max} \frac{1 - (1+i)^{-n}}{i}$$

Нехай

R_{rand} – сума доходу від аукціону по випадковій ціні.

$$R_{rand} = \sum_{j=1}^{N} b_j^{rand}, N \gg n$$

$$b_j^{rand} = b_i * p_i * k$$

$$p_i = \frac{b_i}{\sum_{i=1}^{K} b_i}$$

$$P = \sum_{i=1}^{K} p_i = 1$$

Де K – кількість учасників аукціону,

k – коефіцієнт, який визначає платформа,

p_i – ймовірність виграшу певного гравця.

Для випадку з необмеженим часом аукціону скористаємося формулою безкінечної геометричної прогресії:

$$P = \frac{R}{1 - r}$$

де

$$r = \frac{1}{1 + d}$$

Отримаємо:

$$P_{rand} = R_{rand} * \frac{1 + d}{d}$$

Для системи RTB якщо «максимальний» аукціон монополізується швидко, то вигіднішим є «випадковий» аукціон, а якщо «максимальний» аукціон протримається достатньо довго без монополізації, то вигіднішим є він.

Скільки періодів має протриматись аукціон з найбільшою ціною, щоб бути вигіднішим ніж аукціон з випадковою ціною, без монополізації процесів покупцем з найбільшою ставкою, який, протягом достатньо великого часу, або таємно домовиться з іншими покупцями, або витіснить їх, купуючи одноосібно всі рекламні місця, щоб бути вигіднішим ніж аукціон з випадковою ціною? Для

відповіді на це запитання запишемо умову більшої вигідності «випадкового» аукціону в формі нерівності:

$$R_{rand}\frac{(1+d)}{d} > R_{max}\frac{1-\ (1+d)^{-n}}{d}$$

Перетворимо цей вираз так, щоб отримати n:

$$\frac{(1+d)R_{rand}}{R_{max}} >\ 1-(1+d)^{-n}$$

$$1-\ \frac{(1+d)R_{rand}}{R_{max}} < (1+d)^{-n}$$

$$\frac{R_{max}}{R_{max}-\ (1+d)R_{rand}} > (1+d)^{n}$$

$$\ln\left(\frac{R_{max}}{R_{max}-\ (1+d)R_{rand}}\right) > n\ ln(1+d)$$

$$n < \frac{\ln\left(\frac{R_{max}}{R_{max}-\ (1+d)R_{rand}}\right)}{\ln(1+d)}$$

Таким чином, було отримано формулу, яка дозволяє визначити, не більше якої кількості періодів часу може використовуватись аукціон з найбільшою ціною, аби аукціон з випадковою ціною був вигіднішим.

2.5 Порівняння математичного сподівання стохастичного та класичного аукціонів

За наявними даними, підрахуємо математичне очікування прибутку у випадку з випадковим обиранням переможця. Для аналізу було обрано один з найпоширеніших розмірів оголошення – 300x250. У таблиці 2.2 наведені данні

аналізу, де b – ціна за 1000 показів оголошень (CPM) та p – ймовірність виграшу в аукціоні.

Таблиця 2.2 – Дані з платформи по цінам на рекламні оголошення

i	1	2	3	4	5	6	7
b_i	1.55	2.00	0.35	0.5	0.35	1.55	1.5
p_i	0.1987	0.2565	0.0449	0.064	0.0449	0.1987	0.1923

За місяць при такій ціні CPM зроблено приблизно 10 000 000 показів. Оскільки CPM – це ціна за 1000 показів і платформа у даному випадку мала 10 000 000, при підрахунку прибутку необхідно помножити його на 10 000.

Математичне очікування для аукціону з ймовірністю перемоги має наступний вигляд:

$$M(x) = \sum_{i=1}^{n} x_i p_i$$

$$M(x) = (1.55 * 0.1987 + 2 * 0.2565 + 0.35 * 0.0449 + 0.5 * 0.064 + 0.35 * 0.0449 + 1.55 * 0.1987 + 1.5 * 0.1923) * 10000 = 14808{,}5$$

$$R_{rand} = 14808{,}5 * 25\% \approx 3702$$

$$R_{max} = 2 * 10000 * 25\% = 5000$$

У даному випадку 25% – комісія платформи, яку ми намагаємось максимізувати. Підставимо дані в отриману нерівність (при місячній процентній ставці для дисконтування $d = 2\%$):

$$\frac{\ln\left(\frac{5000}{5000 - (1 + 0.02) * 3702}\right)}{\ln(1 + 0.02)} = \frac{\ln\left(\frac{5000}{1223.96}\right)}{\ln(1.02)} = \frac{1.4073}{0.0198} = 71 \text{ міс.} \approx 6 \text{ р.}$$

Таким чином, аукціону за найбільшою ціною необхідно протриматись близько 6 років, щоб бути вигіднішим за аукціон з випадковою ціною. Якщо він монополізується раніше, то з самого початку перевагу треба надавати «випадковому» аукціону.

Висновки до розділу 2

У даному розділі було описано класичний аукціон другої ціни, його властивості, обирання найкращої стратегії та її доведення. Такий аукціон використовується в системі RTB. Описана проблема обирання оголошення на платформі DSP та існуючи варіанти вирішення з використанням додаткових параметрів.

Запропоновано стохастичний підхід до аукціону Вікрі та доведення його ефективності, базуючись на даних платформи A4G DSP.

3 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Платформа DSP має дві складові – сервер, який оброблює потік запитів та Trading Desk (графічний інтерфейс для управління рекламними оголошеннями). Вони кардинально відрізняються один від одного, адже виконують різні задачі, тому і архітектура, і засоби розробки різняться.x

Для серверної частини необхідно враховувати обробку великої кількості запитів на секунду, і тому має відповідати характеристикам високонавантаженої системи. Так, серверна мова програмування NodeJS є одним з найкращих виборів на сьогодні. Мова підтримується професіональною командою розробників та стрімко розвивається. Також, вона дуже гарно взаємодіє з об’єктами у форматі JSON і це є необхідним критерієм, оскільки саме такий формат використовується у запитах в RTB. За допомогою NodeJS можна швидко та якісно реалізувати принципи специфікації OpenRTB. Для забезпечення обробки великої кількості паралельних запитів у Node.js використовується асинхронна модель запуску коду, заснована на обробці подій в неблокуючому режимі та визначенні обробників зворотніх викликів (callback). Також мова NodeJS дозволяє реалізовувати принципи RESTful API, що відповідає специфікації OpenRTB.

REST

REST – архітектурний стиль взаємодії компонентів розподіленого додатка у мережах. Являє собою узгоджений набір обмежень, що враховуються при проектуванні розподіленої гіпермедіа-системи. [17]

Властивості архітектури, які залежать від обмежень, накладених на REST-системи:

– продуктивність – взаємодія компонентів системи може бути домінуючим фактором продуктивності і ефективності мережі з точки зору користувача;

– масштабованість для забезпечення великої кількості компонентів і взаємодій компонентів.

Рой Філдінг – один з головних авторів специфікації протоколу HTTP, описує вплив архітектури REST на масштабованість наступним чином:

– простота уніфікованого інтерфейсу;

– відкритість компонентів до можливих змін для задоволення мінливих потреб (навіть при працюючому додатку);

– прозорість зав’язків між компонентами системи для сервісних служб;

– переносимість компонентів системи шляхом переміщення програмного коду разом з даними;

– надійність, що виражається в стійкості до відмов на рівні системи при наявності відмов окремих компонентів, з'єднань або даних.

REST, як і кожен архітектурний стиль відповідає ряду архітектурних обмежень. Це гібридний стиль який успадковує обмеження з інших архітектурних стилів [18].

Перша архітектура від якої він успадковує обмеження – це клієнт-серверна архітектура. Її обмеження вимагає розділення відповідальності між компонентами, які займаються зберіганням та оновленням даних (сервером), і тими компонентами, які займаються відображенням даних на інтерфейсі користувача та реагування на дії з цим інтерфейсом (клієнтом). Таке розділення дозволяє компонентам еволюціонувати незалежно [53].

Наступним обмеженням є те, що взаємодії між сервером та клієнтом не мають стану, тобто кожен запит містить всю необхідну інформацію для його обробки, і не покладається на те, що сервер знає щось з попереднього запиту. Обмеження "відсутність стану" не означає, що архітектура REST дозволяє

будувати лише системи, в яких немає стану. Відсутність стану означає, що сервер не знає про стан клієнта і не повинен запам'ятовувати послідовність здійснених до нього запитів, тому що кожен з них є незалежним. Коли клієнт, наприклад, запитує головну сторінку сайту, сервер відповідає на запитання і забуває про клієнта. Клієнт може залишити сторінку відкритою протягом кількох років, перш ніж натиснути посилання, і тоді сервер відповість на інший запит. Тим часом сервер може відповідати на запити інших клієнтів, або нічого не робити — для клієнта це не має значення. Таким чином, наприклад дані про стан сесії (користувача, який автентифікувався) зберігаються на клієнті і передаються з кожним запитом. Це покращує масштабованість, бо сервер після закінчення обробки запиту може звільнити всі ресурси, задіяні для цієї операції, без жодного ризику втратити цінну інформацію. Також спрощується моніторинг і зневадження, бо для того аби розібратись у тому, що відбувається в певному запиті, досить подивитись лише на цей запит. Збільшується надійність, бо помилка в одному запиті не зачіпає інші. Мінусом цього обмеження є те, що знижується продуктивність через те, що в кожен запит тепер доводиться додавати дані сесії з клієнта. Також збереження стану на різних клієнтах важче підтримувати, бо реалізації клієнтів можуть відрізнятись, тоді як середовище сервера повністю під контролем розробника [42, 48].

Додатковим обмеженням стилю REST є те, що системи, написані в цьому стилі, повинні підтримувати кешування, тобто дані, які передаються сервером, повинні містити інформацію про те, чи можна їх кешувати, і якщо можна, то як довго. Це дозволяє збільшувати продуктивність, уникаючи зайвих запитів, але також зменшує надійність системи через те, що дані в кеші можуть застаріти [48].

Елементи даних

Компоненти REST системи спілкуються, передаючи один одному представлення ресурсу в форматі, що обирається з оновлюваного набору

стандартних форматів даних. Формат обирається динамічно відповідно до бажань компонента-клієнта і можливостей сервера. Чи представлення має той самий формат, що й сам ресурс, чи є результатом якогось перетворення — це деталь реалізації, яка ховається за інтерфейсом [19].

Ресурс — це ключовий елемент даних в REST. Ресурсом може бути що завгодно що можна назвати: якийсь документ (наприклад зображення), динамічне значення (наприклад погода у Львові), щось з реального світу (наприклад працівник компанії). Але якщо точніше, то ресурс R – це функція приналежності $M_R(t)$, що відображає моменти в часі на множину однотипних сутностей чи значень [19]. Множина може бути порожньою, тобто REST дозволяє посилання на якийсь об'єкт якого ще не існує [48].

Для того, щоб посилатись на ресурси, використовуються ідентифікатори ресурсів. Компонент, який надав ресурсу ідентифікатор і дозволяє звертатись до нього за цим ідентифікатором, відповідає за збереження функції приналежності незмінною. Якість ідентифікатора залежить від якості компонента, який цей ідентифікатор надає, тому деякі ідентифікатори стають «мертвими посиланнями», коли інформацію переміщують або знищують.

Представлення – це послідовність байтів та метадані представлення, для того щоб описати ці байти. Часто, представлення називають документом, файлом, повідомленням HTTP тощо.

Конектори

Конектори надають інтерфейс для комунікації компонентів, приховуючи реалізації ресурсів та механізм комунікації. Конектори подібні на віддалений виклик процедур, але з певними нюансами щодо передачі параметрів та результату виклику. Параметри складаються з ідентифікатора ресурсу, контрольних даних та необов'язково, представлення. Результат – з контрольних даних відповіді і

представлення. Можна абстрагуватись і вважати такий виклик синхронним, але насправді передача даних відбувається потоково, тому обробку даних можна починати ще до того як отримані всі дані, таким чином зменшуючи латентність.

Двома найважливішими типами конекторів є клієнт і сервер. Відмінністю між ними є те що клієнт ініціює запит, в той час як сервер очікує запитів і відповідає на них даючи доступ до своїх сервісів. Компонент може містити одночасно як серверні так і клієнтські конектори [19].

Додатковим типом конектора є кеш. Кеш може бути як клієнтським, для уникнення зайвих запитів, так і серверним — для уникнення зайвого обчислення відповіді на запит. Тому що інтерфейс однорідний, кеш легко може дізнатись чи запит можна кешувати. За замовчуванням, відповідь на запит отримання ресурсу можна кешувати, а запити зміни ресурсу — ні. Проте ці замовчування можна перевантажити контрольними даними.

Резолвер – це ще один тип конектора, який перетворює ідентифікатори ресурсів в інформацію про мережеві адреси необхідну компонентам щоб отримати цей ресурс. Наприклад в URI міститься доменне ім'я, і для доступу до відповідного домену, потрібно дізнатись в DNS-сервера адресу. Тому в цьому випадку система DNS грає роль резолвера. Використання одного чи кількох резолверів може збільшити життєздатність ідентифікатора ресурсу, через те що він не вказує напряму на фізичне розташування ресурсу яке може змінитись. Останньою формою конектора є тунель, який просто проводить запити через межу системи, наприклад, через фаєрвол. Причиною, через яку тунелі включено до архітектури REST, а не закрито абстракцією мережі, є те, що певні компоненти можуть перетворюватись на тунелі за запитом. Наприклад, тунель HTTP активується при отриманні запиту з методом CONNECT [42, 49].

REST використовує протокол HTTP. Стандарт HTTP визначає 8 типів повідомлень: GET – отримати представлення ресурсу, DELETE – знищити ресурс,

POST – створити новий ресурс на місці даного використавши передане представлення, PUT – замінити стан поточного ресурсу станом, що описується переданим представленням, HEAD – отримати заголовки, які б відсилались разом з представленням цього ресурсу, але не саме представлення, OPTIONS – визначити список методів, на які цей ресурс відповідає. Ще два методи описуються в пропозиції до стандарту «Internet-Draft „snell-link-method“»: LINK — прив'язати певний ресурс, UNLINK — відв'язати ресурс [42, 49].

Загалом в RTB використовується 2 типу повідомлень: POST та GET. Статус коди відповідей описані в розділі 1.6.

Trading Desk

Trading Desk – це веб-застосунок і у більшості випадків використовується десктопними пристроями, оскільки обмеженість розмірів мобільних пристроїв не дозволяє умістити усю функціональність. Також, перевагою веб-застосунків є те, що вони доступні з будь-якого місця та не навантажують пристрій, оскільки всі обчислення виконуються на сервері.

Для Trading Desk обрано мову програмування PHP, оскільки вона не вимагає додаткового використання серверу для взаємодії з базою даних, так як це вимагають, наприклад, популярні JavaScript бібліотеки VueJS, AngularJS та ReactJS. Загалом, PHP – скриптова мова загального призначення, інтенсивно застосовується для розробки веб-додатків. На даний момент підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів.

Зберігання даних

Для зберігання даних використовується декілька баз даних, в залежності від їх функціональності. Так, для зберігання основних даних про користувачів,

налаштування рекламних оголошень, інформація про завантажені файли, заблоковані рекламні місця, ip-адреси, загальна інформація про статистику використовується MySql. Кеш та тимчасово не збереженні дані зберігаються в базі даних Aerospike. Великий об’єм даним – детальна статистика для кожного користувача зберігається в ClickHouse.

MySQL

MySQL – це вільна реляційна система управління базами даних. Характеризується високою швидкістю, стійкістю і простотою використання. MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

– простота у встановленні та використанні;

– підтримується необмежена кількість користувачів, що одночасно працюють із БД;

– кількість рядків у таблицях може досягати 50 млн;

– висока швидкість виконання команд;

– наявність простої і ефективної системи безпеки.

Aerospike

Aerospike – це розподілена, масштабована база даних класу NoSQL. Архітектура має три ключові завдання:

– створіть гнучку, масштабовану платформу для веб-програм;

– забезпечити надійність та надійність (як у ACID), що очікуються від традиційних баз даних;

– забезпечити експлуатаційну ефективність при мінімальному залученні вручну. [51]

За заявою творців, на виконання 99% запитів витрачається менше 1 мс, а 99.9% запитів виконуються не довше 5 мс. Використання спеціально оптимізованого для SSD-накопичувачів режиму роботи сховища у поєднанні з розміщенням індексів в оперативній пам'яті і забезпеченням паралельної обробки даних на різних ядрах CPU, дозволило добитися в Aerospike в 5-10 разів вищої продуктивності, у порівнянні з іншими NoSQL БД.

На базі Aerospike можливе створення кластерних конфігурацій з реплікацією даних на декілька вузлів, які забезпечують близьке до лінійного зростання продуктивності при додаванні нових вузлів у кластер. Розмір сховища може досягати десятка терабайт і обслуговувати розміщення більше сотні мільярдів об'єктів. Підтримується кілька схем реплікації, націлених на забезпечення відмовостійкості, продуктивності та територіального рознесення бази за різними центрам обробки даних.

Система надійно захищена від втрати даних в результаті збою — застосовується блокування на рівні рядків і миттєва фіксація транзакцій. При використанні синхронної реплікації кластер зберігання Aerospike відповідає вимогам ACID (атомарность, узгодженість, ізольованість, надійність). У разі виходу вузла з ладу, автоматично виконується ребалансування даних і продовження роботи без збійного вузла. Оновлення програмного забезпечення і резервне копіювання виробляються без призупинення роботи і без здійснення впливу на продуктивність.

Дані зберігаються в рядках у формі записів ключ/значення. Записи можуть групуватися в набори і таблиці (роздільний простір імен). Для обробки даних на стороні сервера підтримується написання визначених користувачем функцій (UDF) мовою Lua. Можливе виконання складних аналітичних запитів (map-reduce) і

використання великих типів даних, при цьому виконання запиту можна розпаралелювати на кілька вузлів кластера, кожен з яких виконує свою частину роботи з обробки запиту. [50]

ClickHouse

ClickHouse – це колоночна аналітична СУБД з відкритим кодом, що дозволяє виконувати аналітичні запити в режимі реального часу на структурованих великих даних. Використовує власний діалект SQL близький до стандартного, але містить різні розширення: масиви та вкладені структури даних, функції вищого порядку, імовірнісні структури, функції для роботи з URI, можливість для роботи з зовнішніми key-value сховищами («словниками»), спеціалізовані агрегатні функції, функціональності для семплірованія, приблизних обчислень, можливість створення збережених уявлень з агрегацією, наповнення таблиці з потоку повідомлень Apache Kafka і т. д.

Однак є і обмеження – відсутність транзакцій, відсутність точкових UPDATE / DELETE (пакетний UPDATE / DELETE був введений в червні 2018 роки), обмежена підтримка синтаксису JOIN, строгі типи з необхідністю явного приведення, для деяких операцій проміжні дані повинні міститися в оперативну пам'ять , відсутність віконних функцій, відсутність повноцінного оптимізатора запитів, точкового читання, присутність обмежень в реалізації деяких функцій, пов'язаних зі специфікою використання ClickHouse.

Система оптимізована для зберігання даних на жорстких дисках (використовуються переваги лінійного читання, стиснення даних). Для забезпечення відмовостійкості і масштабованості ClickHouse може бути розгорнутий на кластері (для координації процесу реплікації використовується Apache ZooKeeper). Для роботи з базою даних існує консольний клієнт, веб-клієнт,

HTTP інтерфейс, ODBC і JDBC-драйвери, а також готові бібліотеки для інтеграції з багатьма популярними мовами програмування і бібліотеками.

Для передачі даних в ClickHouse використовується Kafka.

Kafka

Apache Kafka – проект з відкритим вихідним кодом, що розробляється в рамках фонду Apache. Спроектовано як розподілена, горизонтально масштабована система, що забезпечує нарощування пропускної здатності як при зростанні числа і навантаження з боку джерел, так і кількості систем-передплатників. Підписники можуть бути об'єднані в групи. Підтримується можливість тимчасового зберігання даних для подальшої пакетної обробки. Однією з особливостей реалізації інструменту є застосування техніки, схожої з журналами транзакцій, використовуваними в системах управління базами даних.

Розподіленої називається така система, яка в сегментованому вигляді працює відразу на безлічі машин, що утворюють цілісний кластер; тому для кінцевого користувача вони виглядають як єдиний вузол. Розподіленість Kafka полягає в тому, що зберігання, отримання та розсилка повідомлень у нього організовано на різних вузлах (так званих «брокерів»). Найважливіші плюси такого підходу - високодоступність і відмовостійкість.

Горизонтальне масштабування

Ватро спочатку визначимося з тим, що таке вертикальна масштабованість. Припустимо, у нас є традиційний сервер бази даних, і він поступово перестає справлятися з наростаючою навантаженням. Щоб впоратися з цією проблемою, можна просто наростити ресурси (CPU, RAM, SSD) на сервері. Це і є вертикальне масштабування – на машину навішуються додаткові ресурси. При такому «масштабування вгору» виникає два серйозні недоліки:

– є певні межі, пов'язані з можливостями обладнання. Нескінченно нарощуватися не можна;

– така робота зазвичай пов'язана з простоями, а великі компанії не можуть дозволити собі простоїв.

Горизонтальна масштабованість вирішує рівно ту ж проблему, просто ми підключаємо до справи все більше машин. При додаванні нової машини ніяких простоїв не відбувається, при цьому, кількість машин, які можна додати в кластер, нічим не обмежена. Заковика в тому, що не у всіх системах підтримується горизонтальна масштабованість, багато системи не розраховані на роботу з кластерами, а ті, що розраховані - зазвичай дуже складні в експлуатації.

Відмовостійкість

Для нерозподілених систем характерна наявність так званої єдиної точки відмови. Якщо єдиний сервер вашої бази даних з якоїсь причини відмовить - ви потрапили.

Розподілені системи проектуються таким чином, щоб їх конфігурацію можна було коригувати, підлаштовуючись під відмови. Кластер Kafka з п'яти вузлів залишається працездатним, навіть якщо два вузла ляжуть. Необхідно відзначити, що для забезпечення відмовостійкості обов'язково доводиться частково жертвувати продуктивністю, оскільки чим краще система переносить відмови, тим нижче її продуктивність.

Журнал коммітів

Журнал коммітів (журнал транзакцій) – це довгострокова упорядкована структура даних, дані в таку структуру можна тільки додавати. Записи з цього журналу можна ні змінювати, ні видаляти. Інформація зчитується зліва направо; таким чином гарантується правильний порядок елементів.

Робота Kafka

Додатки (генератори) посилають повідомлення (записи) на вузол Kafka (брокер), і повідомлення про прийняття обробляються іншими додатками, так званими споживачами. Зазначені повідомлення зберігаються в темі, а споживачі підписуються на тему для отримання нових повідомлень.

Теми можуть розростатися, тому великі теми поділяються на більш дрібні секції для поліпшення продуктивності і масштабованості. (Приклад: припустимо, ви зберігали призначені для користувача запити на вхід в систему; в такому випадку можна розподілити їх по першому символу в імені користувача).

Kafka гарантує, що всі повідомлення в межах секції будуть впорядковані саме в тій послідовності, в якій надійшли. Конкретне повідомлення можна знайти за його зміщення, яке можна вважати звичайним індексом в масиві, порядковим номером, який збільшується на одиницю для кожного нового повідомлення в даній секції.

У Kafka дотримується принцип «тупий брокер - розумний споживач». Таким чином, Kafka не відслідковує, які записи зчитуються споживачем і після цього видаляються, а просто зберігає їх протягом заданого періоду часу (наприклад, доби), або до тих пір, поки не буде досягнутий певний поріг. Споживачі самі опитують Kafka, чи не з'явилося у нього нових повідомлень, і вказують, які записи їм потрібно прочитати. Таким чином, вони можуть збільшувати або зменшувати зміщення, переходячи до потрібного запису; при цьому події можуть повторно оброблятися.

Слід зазначити, що насправді мова йде не про поодинокі споживачах, а про групи, в кожній з яких - один або більше процесів-споживачів. Щоб не допустити ситуації, коли два процеси могли б двічі прочитати одне й те саме повідомлення,

кожна секція прив'язується лише до одного процесу-споживачеві в межах групи [15].

Zookeeper

Zookeeper - це розподілене сховище ключів і значень. Воно сильно оптимізовано для зчитування, але записи в ньому відбуваються повільніше. Найчастіше Zookeeper застосовується для зберігання метаданих та обробки механізмів кластеризації (пульс, розподілені операції оновлення / конфігурації, тощо).

Таким чином, клієнти цього сервісу (брокери Kafka) можуть на нього підписуватись - і будуть отримувати інформацію про будь-які зміни, які можуть статися. Саме так брокери дізнаються, коли ведучий в секції змінюється.

3.2 Вимоги до технічного забезпечення

Для коректної роботи Trading Desk має бути комп’ютер з наступними характеристиками:

- процесор з тактовою частотою не нижче 1.1 ГГц;
- об’єм оперативної пам’яті RAM не менше 2048 МБ;
- ємність дисків HDD або SSD не нижча за 20 ГБ;
- постійний доступ до мережі Інтернет.

Для роботи клієнтського застосування повинні використовуватись наступні периферійні пристрої, під’єднані до комп’ютера:

- монітор;
- клавіатура дротова або бездротова;
- мишка або трекпад.

Також варто зазначити, що запуск застосування можливий також на планшетах та смартфонах, які теж відповідають мінімальним технічним вимогам сьогодення і мають доступ до мережі Інтернет.

Окрім цього, для запуску веб-застосування повинно бути встановлене додаткове програмне забезпечення, а саме – один з перелічених веб-браузерів:

– Google Chrome версії 45 або більш новий;
– Microsoft Edge версії 20 або більш новий;
– Mozilla Firefox версії 43 або більш новий;
– Opera версії 30 або більш новий;
– Safari версії 12 або більш новий.

У таких браузерах, як, наприклад, Internet Explorer та Opera Mini, можливі збої у роботі програми. Не рекомендується використовувати дані браузери для запуска застосування.

Для розгортання серверної частини використовується два сервери (US та EU Balancer) для розподілення запитів, одинадцять серверів для обробки запитів (endpoint), три для кешування даних (Aerospike) і по одному для баз даних MySQL та ClickHouse.

Таблиця 3.1 – Технічні характеристики серверів

Назва серверу	**RAM**	**SSD**	**Кількість ядр**
Balancer	32 ГБ	100 ГБ	32
Endpoint	64 ГБ	100 ГБ	32
Aerospike	64 ГБ	1 ТБ	32
MySQL	128 ГБ	1 ТБ	32

3.3 Архітектура програмного забезпечення

Запити від SSP приходять на один з серверів Balancer, в залежності від регіону (Європа чи США). Далі запити розподіляються між endpoint'ами. При обробці запиту та успішному аукціоні дані про переможця зберігаються в базі даних Aerospike. В цій базі тимчасово (2 дні) зберігається інформація про аукціони, кількість показів, кліків та конверсій усіх оголошень за цей час. Також, дані записуються в kafka для більш детальної статистики. База даних ClickHouse налаштована так, щоб зчитувати дані з kafka.

Паралельно процесу торгів працює Script – це сервіс, який збирає накопичені дані з Aerospike, агрегує їх та завантажує інформацію в базу даних MySQL.

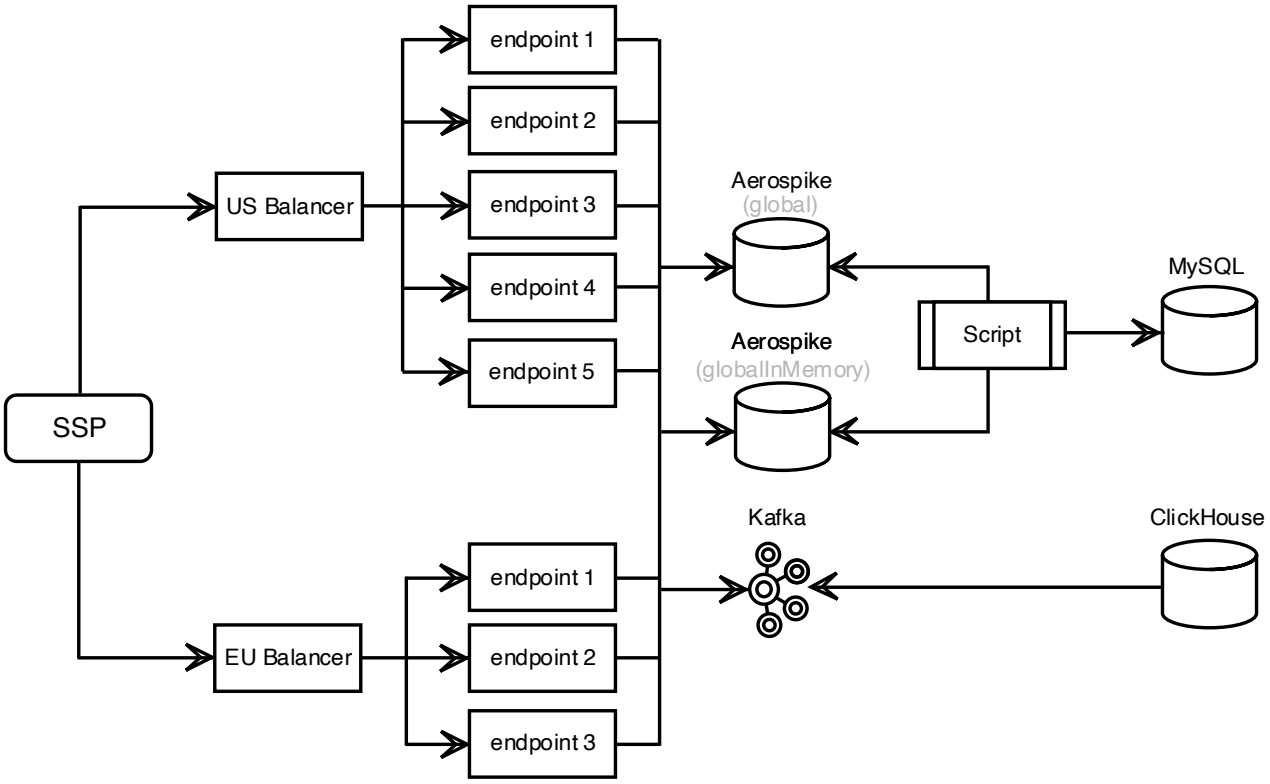


Рисунок 3.1 – Схема серверів

Balancer – сервер, який приймає запити та розподіляє їх між основними серверами для обробки, в залежності від їх завантаженості.

Endpoint – основний сервер, який містить програму з обробки запиту.

Bidder – програма, яка встановлена на кожному endpoint. Проводить фільтрацію запитів та аукціон між оголошеннями.

Aerospike – швидка база даних для кешування даних на сервері.

ClickHouse – база даних для зберігання великого об’єму даних.

Script – окрема програма, яка збирає тимчасові дані, які стосуються статистики, з Aerospike та записує її в ClickHouse.

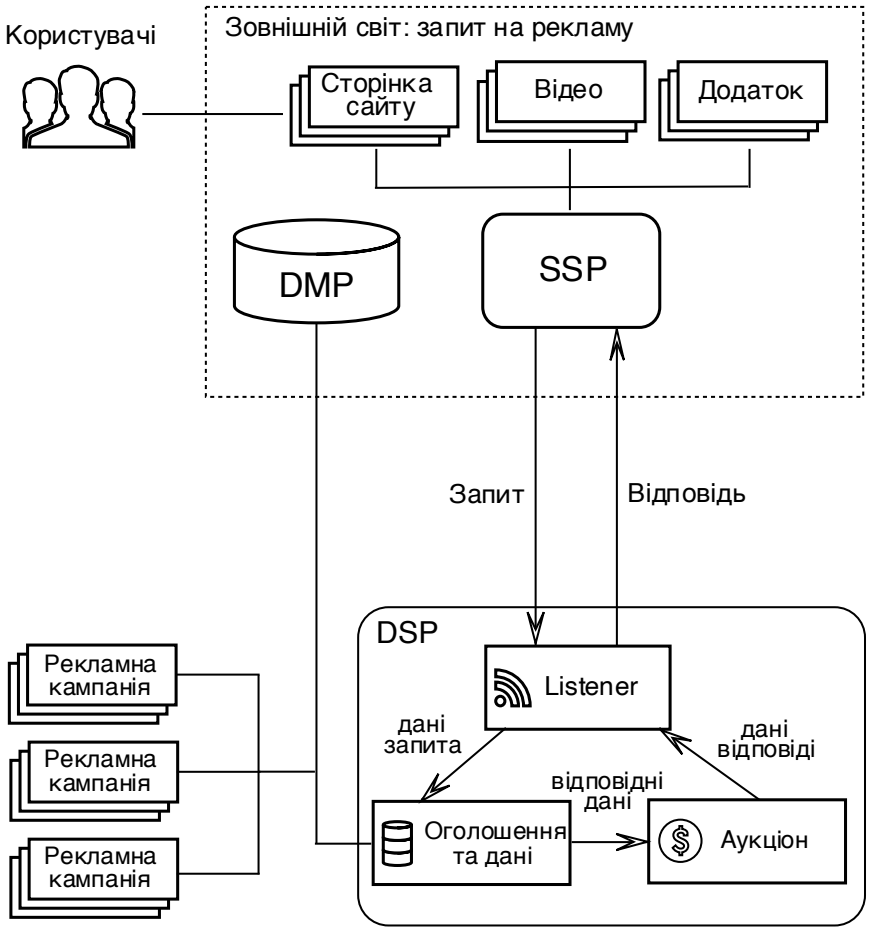


Рисунок 3.2 – Схема взаємодії компонентів RTB

Основними діями DSP є: прийняти запит, обробити параметри та обрати рекламні оголошення, які підпадають під вимоги, провести серед них аукціон та надати відповідь з медіаданими оголошення. Схема взаємодії компонентів RTB та біль детальний опис компонентів DSP зображено на рисунку 3.2.

Послідовність виконання функцій в роботі DSP:

1) Balancer отримує запит від SSP. Перевіряючи доступні сервери та їх завантаженість, відправляє запит на менш завантажений.

2) Сервер, отримавши запит, перевіряє його. Запит має відповідати певним критеріям. Бути надісланим методом POST. Мати GET параметри для аутентифікації SSP. Та мати тіло запиту у вигляді JSON об’єкту з певними параметрами.

2.1) getPartners(): аутентифікація SSP, від якої прийшов запит.

2.2) parseBodySSP():Перевірка тіла функції на пустоту та аналіз параметрів тіла запиту.

3) countIncomeQpsCore(): Перевірка QPS (кількість запитів, які надсилає певна SSP за одну секунду). Чи не перевищин ліміт, якщо такий встановлений.

4) getTypeRequst(): Визначається тип реклами, на яку прийшов запит (banner, video, native).

5) validationValuesRequest(): Перевірка наявності та коректності необхідних параметрів у запиті для певного типу реклами.

6) За допомогою сервіса DigitalEnvoy визначається місце розташування користувача, якому буде показана реклама.

7) getDeviceCharacteristics(): Перевірка інформації про пристрій користувача на якому буде відображатись реклама.

8) checkIsPmp(): Перевірка чи є запит приватним.

9) checkIsSecure(): Перевірка чи захищений протокол передачі даних (чи використовується протокол HTTPS).

10) getTypeDevice(): Визначення типу пристрою (desktop, mobile, application), на якому буде показуватися реклама.

11) getCategoryIAB(): Перевірка на категорію сайту чи додатку, де буде показуватися реклама.

12) checkGlobalBanList(): Перевірка глобального списку блокування запитів за різними параметрами.

13) getCreatives(): Завантаження усіх рекламних одиниць з кешу в оперативну пам’ять.

14) asyncGeneralFilter(): Фільтрація списку рекламних запитів за параметрами, які були у запиті.

15) filterImpressFrequenceCap(): Перевірка отриманих рекламних запитів після фільтрації на обмеження кількості показів за секунду.

16) auction(): аукціон між відібраними оголошеннями реклами для відповіді на запит.

17) getBidResponse(): Формування та відправлення відповіді.

Як ми бачимо з роботи DSP та схеми, яка зображена на рисунку 3.1, аукціон відбувається в останню чергу, тому більшість оголошень буде відфільтровано, проте, як правило усі користувачі мають оголошення кожного розміру, а тому велика ймовірність великої конкуренції. Тому важливо задовільнити усіх користувачів і надати їм можливість досягти бажаних результатів рекламних компаній. Тому стохастичний підхід, який описано у розділах 2.4, 2.5, може задовільнити таку потребу.

Також існують певні функції для зарахування відповідних статистичних значень. Функції маршрутизації, які викликаються через GET запити на сервер з відповідним маршрутом:

setImpression() – фіксує показ оголошення.

setWinNotification() – фіксує сповіщення про обрання оголошення на стороні SSP.

setClick() – фіксує натискання на оголошення.

setClickRedirect() – фіксує натискання на оголошення та переадресує за вказаним посиланням.

setPostback() – фіксує конверсію.

showSSPTrafficStatistics() – збирає запити на певну компанію чи оголошення.

getBlockedBannerStatistics() – збирає відфільтровані запити для кожної компанії чи оголошення.

tackVastEvents() – фіксує подію відео оголошення (початок перегляду, 25%, 50%, 75%, 100%).

getVideoAdm() – отримання тіла відео оголошення.

Коли сервер отримує одне з повідомлень, створюється об’єкт з даними, де додається відповідний показник і записується в Aerospike, де дані зберігаються до двох днів. За цей час Script, який запущено через cron, збирає нові дані за період з минулого запиту і зберігає їх в MySQL. Також паралельно запису даних в Aerospike, дані також записуються і в Kafka. Сервер ClickHouse налаштований так, щоб, як і Script, читати дані з Kafka та зберігати у відповідні таблиці (для кожного ключового параметра окрема таблиця), які об’єднуються в єдину з усіма даними.

Надалі, коли користувач переглядає дані статистики, вони завантажуються або з ClickHouse для користувацького звіту або з MySQL для головної сторінки. Також загальні данні по всій платформі зберігаються в MySQL. Послідовність цих процесів зображена на рисунку 3.3.

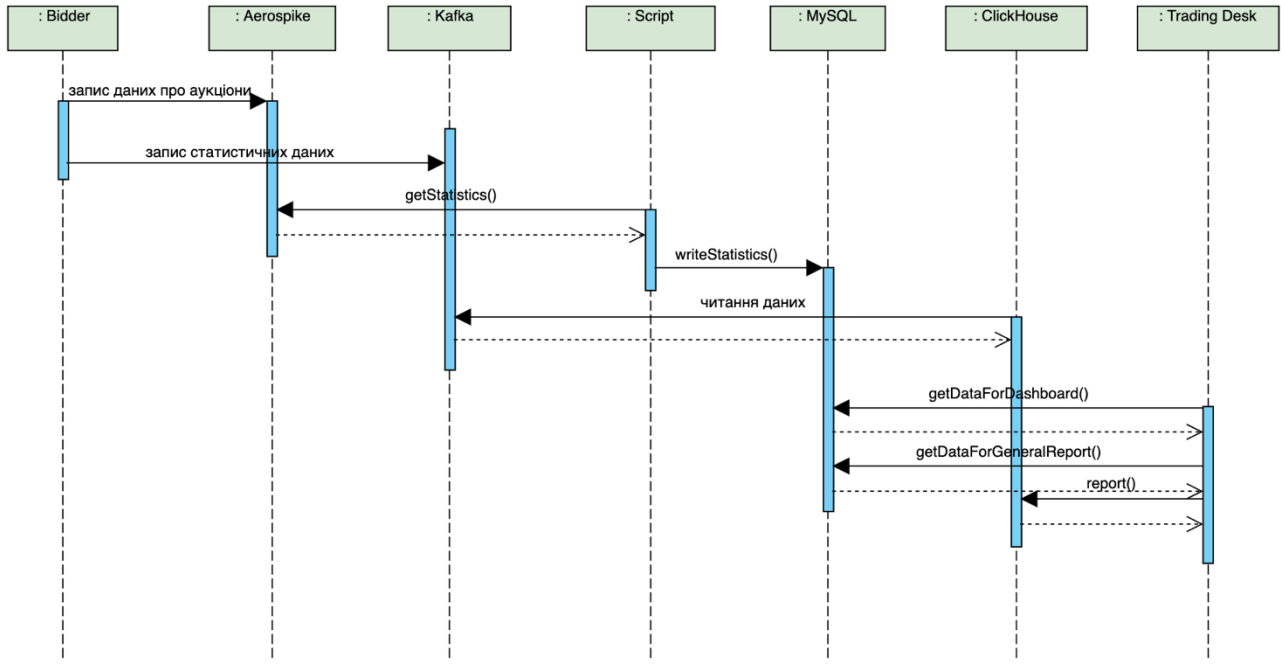


Рисунок 3.3 – Діаграма послідовності

Trading Desk

Trading Desk побудований за класичним патерном MVC (рисунок 3.4). *Модель* надає дані і методи роботи з ними: запити до бази даних, перевірка на коректність. Модель не залежить від уявлення (не знає як дані візуалізувати) і контролера (не має точок взаємодії з користувачем), просто надаючи доступ до даних і управління ними. Модель будується таким чином, щоб відповідати на запити, змінюючи свій стан, при цьому може бути вбудовано повідомлення «спостерігачів». Модель, за рахунок незалежності від візуального представлення, може мати кілька різних представлень для однієї «моделі».

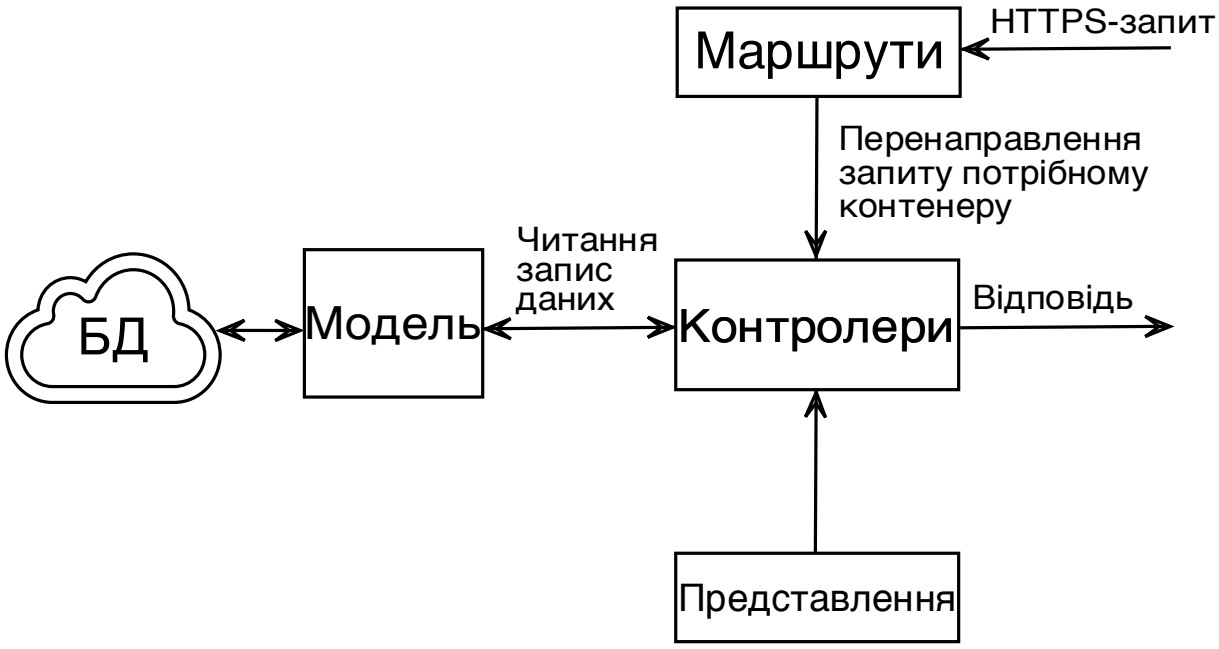


Рисунок 3.4 – Схема патерну MVC

Представлення відповідає за отримання необхідних даних з моделі і відправляє їх користувачеві. Подання і не виконує жодних введені дані користувача.

Контролер забезпечує «зв'язок» між користувачем і системою. Контролює і направляє дані від користувача до системи і навпаки. Використовує модель і уявлення для реалізації необхідного дії.

Основна мета застосування цієї концепції полягає в відділенні бізнес-логіки (моделі) від її візуалізації (представлення). За рахунок такого поділу підвищується можливість повторного використання коду. Найбільш корисне застосування даної концепції в тих випадках, коли користувач повинен бачити ті ж самі дані одночасно в різних контекстах або з різних точок зору [55]. Патерн виконує наступні задачі:

– до однієї моделі можна приєднати кілька представлень (view), при цьому не зачіпаючи реалізацію моделі. Наприклад, деякі дані можуть бути одночасно представлені у вигляді електронної таблиці, гістограми і кругової діаграми;

– не торкаючись реалізації представлень, можна змінити реакції на дії користувача (натискання мишею на кнопці, введення даних) - для цього досить використовувати інший контролер;

– ряд розробників спеціалізується тільки в одній з областей: або розробляють графічний інтерфейс, або розробляють бізнес-логіку. Тому можливо добитися того, що програмісти, які займаються розробкою бізнес-логіки (моделі), взагалі не будуть обізнані про те, яке уявлення буде використовуватися.

Оскільки MVC не має суворої реалізації, то реалізований він може бути по-різному. Немає загальноприйнятого визначення, де повинна розташовуватися бізнес-логіка. Вона може знаходитися як в контролері, так і в моделі. В останньому випадку, модель буде містити всі бізнес-об'єкти з усіма даними і функціями. Саме такий підхід рекомендований PHP-фреймворком Laravel, який був використаний для реалізації Trading Desk.

Варіанти використання Trading Desk для авторизованого користувача та для адміністратора зображено на рисунках 3.5 та 3.6 відповідно.

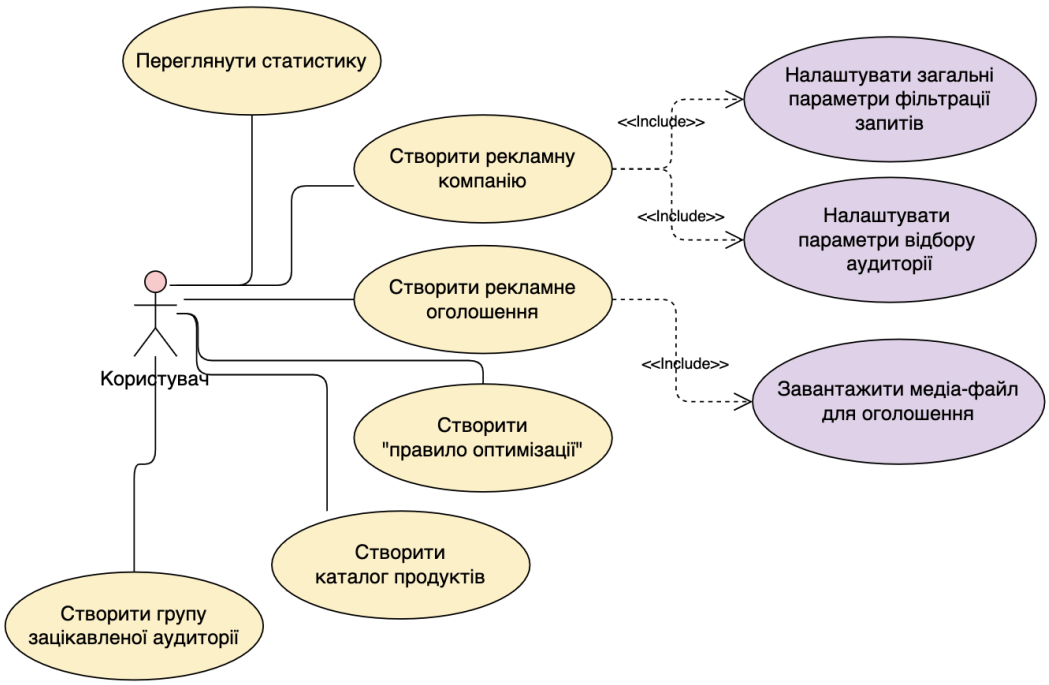


Рисунок 3.5 – Варіанти використання для авторизованого користувача

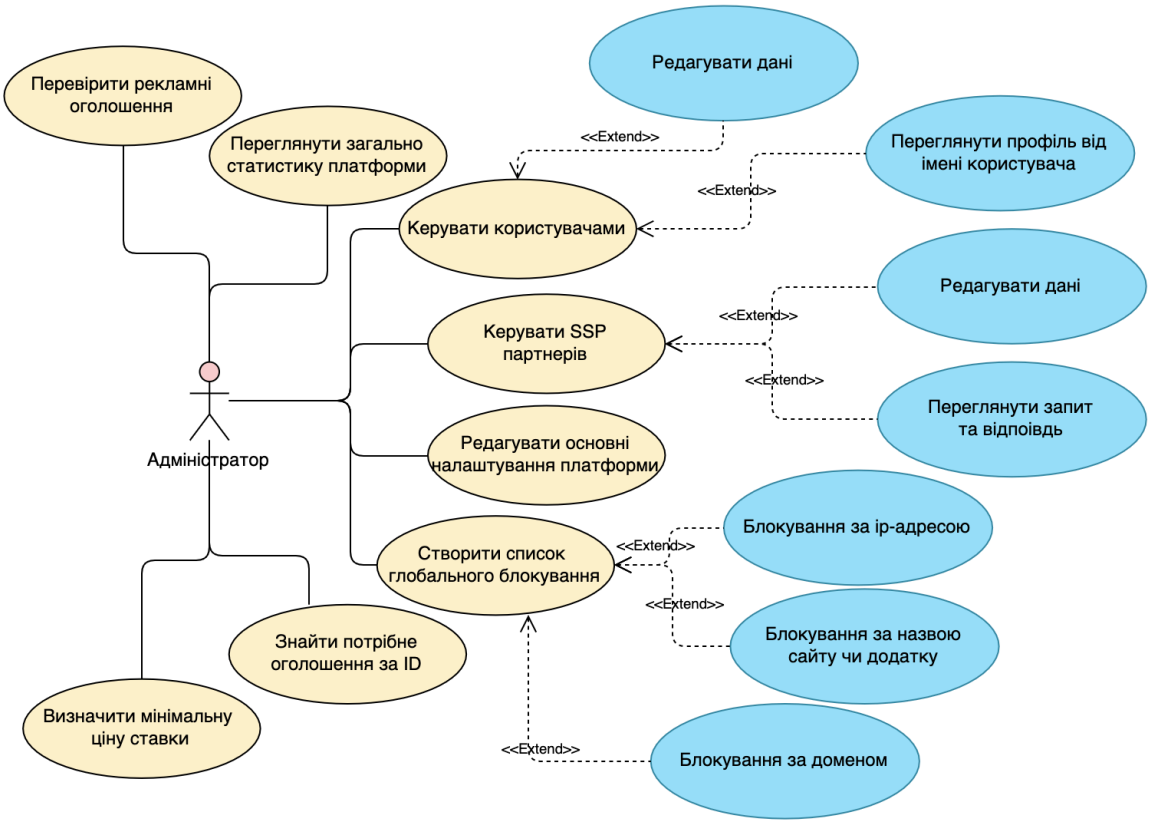


Рисунок 3.6 – Варіанти використання для адміністратора

3.4 Функціональність системи

Користувач, після входу, потрапляє на головну сторінку, де відображаються показники з показу реклами та витрачений бюджет. На скріншоті головної сторінки (рисунок 3.7) бюджет відображається від'ємний, то що у даному випадку користувач має кредит.

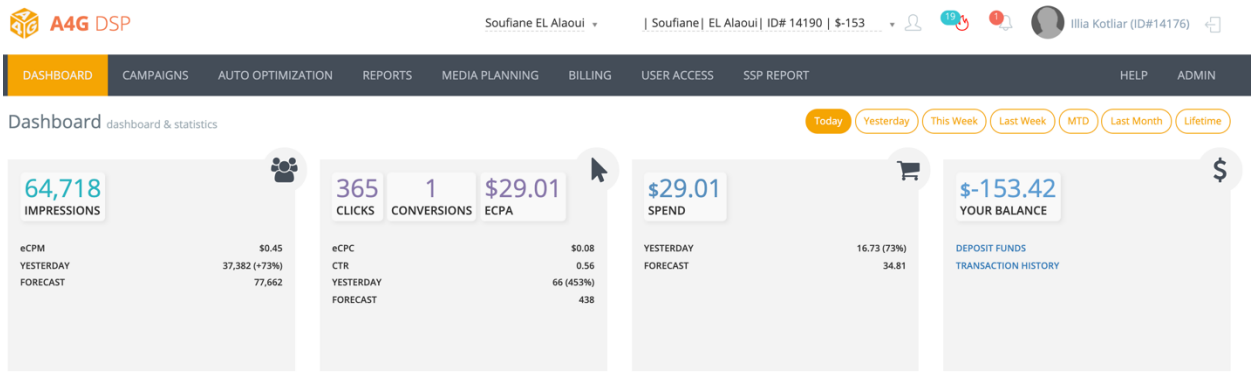


Рисунок 3.7 – Головна сторінка додатку

На сторінці звіту (reports) є досить багато параметрів, які допомагають аналізувати ефективність рекламних компаній (рисунок 3.8).

Рисунок 3.8 – Параметри на сторінці звіту

Impressions – кількість показів рекламних оголошень.

Bids – кількість виграних аукціонів.

Spend – загальна витрачена вартість за покази оголошень.

Clicks – кількість кліків на оголошення.

CTR (Click-Through Rate) – Відношення кількості кліків до кількості показів. Вираховується як відношення clicks до imressions.

CPM – ціна за 1000 показів оголошення. Встановлюється користувачем.

eCPM – Реальна ціна за показ 1000 оголошень. Вираховується як відношення spend до impressions помножений на 1000 [32].

Conversions – Кількість конверсій.

eCPA – ціна за одну конверсію. Вираховується як conversions до sprend.

Також існують додаткові параметри, які відображаються в окремій табличці та тій же сторінці звіту. Перелік додаткових параметрів зображено на рисунок 3.9.

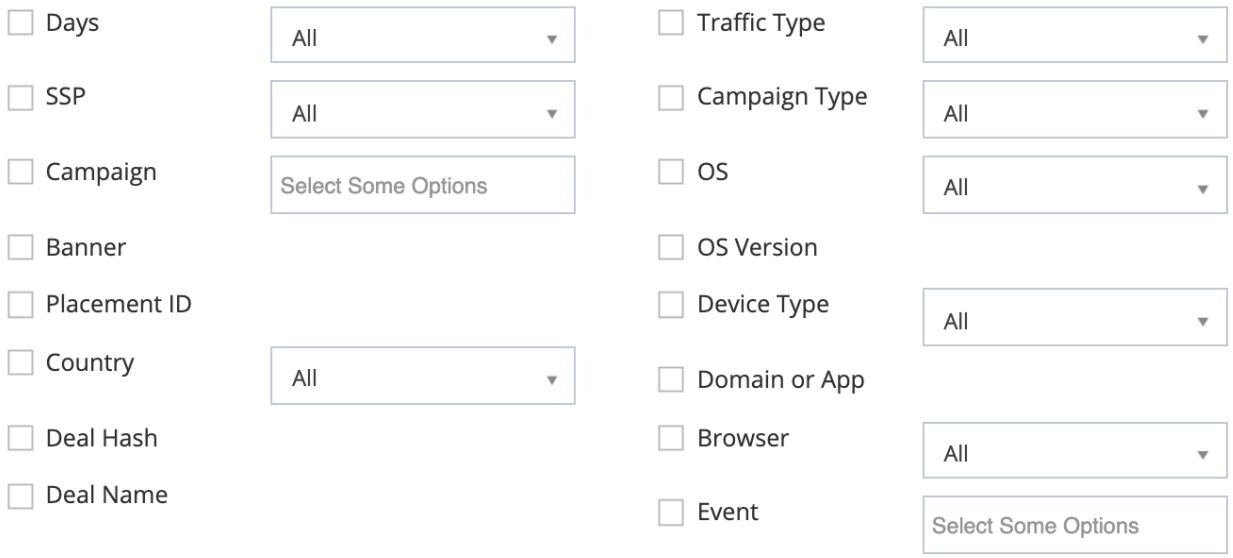


Рисунок 3.9 – Додаткових параметри

Додатковий звіт можна згенерувати, обравши наступні параметри:

Days – звіт на певний день.

SSP – за певним партнером Supply-side Platform.

Campaigns – за певними компаніями.

Banner – включити тільки банерні оголошення у звіт.

Country – за певною країною.

Placement Id – ідентифікатор рекламного місця.

Deal Hash та Deal Name – хеш та назва договору для приватного ринку.

Traffic Type – вид трафіку (сайт чи додаток).

Campaign Type – вид рекламної компанії.

OS та OS Version – операційна система та її версія.

Device Type – тип пристрою (комп’ютер, мобільний, планшет).

Domain or App – назва сайту чи додатку.

Browser – назва браузеру.

Event – події для конверсій.

Створення рекламного оголошення складається з двох етапів. На першому – користувачу необхідно створити компанію з параметрами, по яким буде відбуватися відбір аудиторії, на другому – безпосередньо саме оголошення (його вигляд, тип, ціну тощо).

Налаштування компанії поділені на дві складові: загальні налаштування (рисунок 3.10) та налаштування таргетингу (рисунок 3.11).

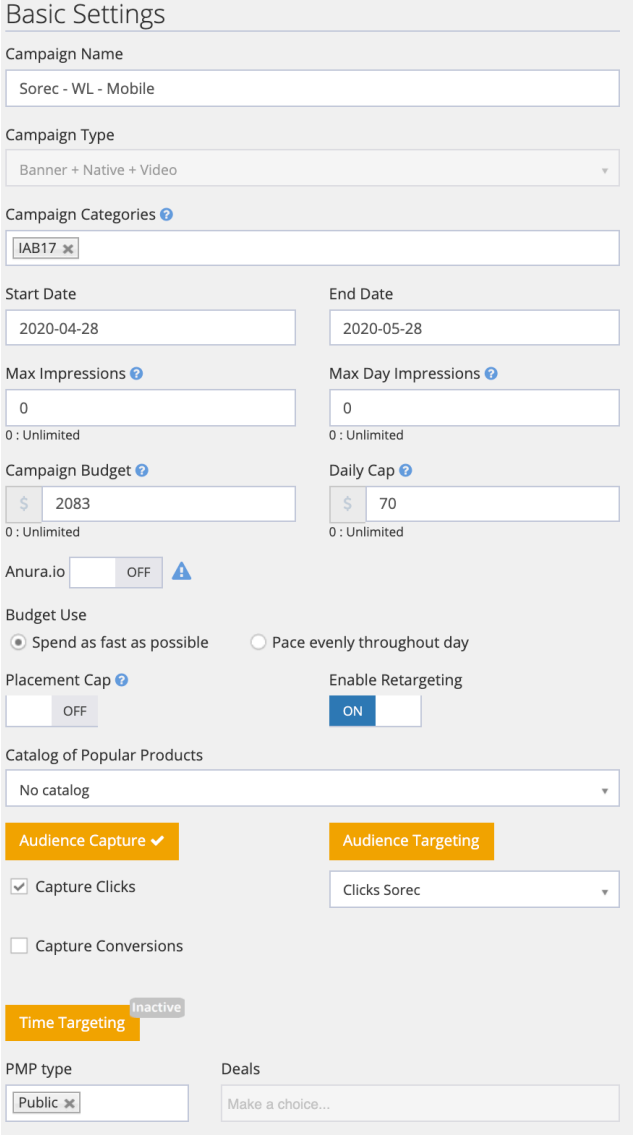


Рисунок 3.10 – Загальні налаштування компанії

Campaign Name – назва компанії.

Campaign Type – тип компанії (можна встановити тільки один раз при створенні компанії).

Campaign Categories – категорія, до якої відноситься компанія за стандартними категоріями IAB.

Start / End Date – дата початку та кінець роботи компанії.

Max (Day) Impressions – обмеження на максимальну кількість показів оголошень загально та на кожен день.

Campaign Budget – обмеження на загальний бюджет компанії.

Daily Cap – обмеження на денний бюджет компанії.

Anura.io – аналізатор трафіку.

Budget Use – як буде використовуватися бюджет: якомога швидше або протягом усього дня.

Placement Cap – додає обмеження на бюджет для кожного місця для оголошень.

Enable Retargeting – дозволяє збирати данні о користувачах за кліками або конверсіями.

Time Targeting – дозволяє встановити час, коли мають показуватися оголошення.

PMP (Private Marketplace) – приватний аукціон.

PMP Type – тип приватного аукціону.

Deals – номера договорів для приватних аукціонів.

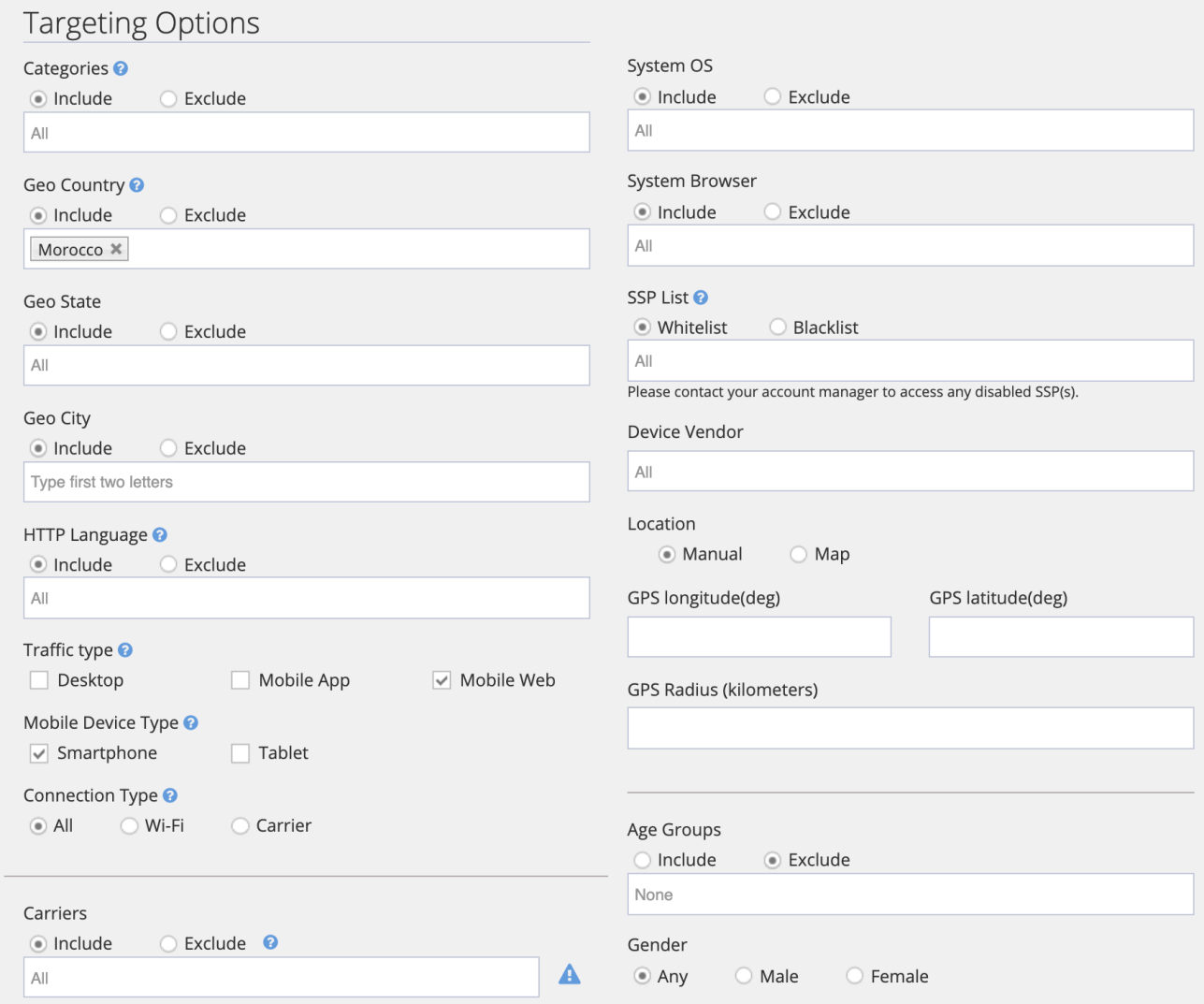


Рисунок. 3.11 – Налаштування таргетингу

Categories – категорії сайтів чи додатків, на яких можна публікувати оголошення.

Geo Country – країна, де має показуватися оголошення.

Geo State – область чи штат, де має показуватися оголошення.

Geo City – місто, де має показуватися оголошення.

HTTP Language – мова сайту чи додатку.

Traffic type – пристрої для показу оголошень.

Connection Type – типи зв’язку.

Carriers – назва провайдера.

System OS – операційні системи.

System Browser – браузер.

SSP List – перелік партнерів, які можуть пропонувати місця.

Device Vendor – виробник пристрою.

Location – завдання координат з радіусом або вибір території на карті, де мають показуватися оголошення.

Age Groups – вікові категорії аудиторії.

Gender – стать, якій показувати оголошення.

Наступним етапом іде створення безпосередньо самого оголошення, його вигляду. На рисунку 3.12 зображено основні налаштування оголошення.

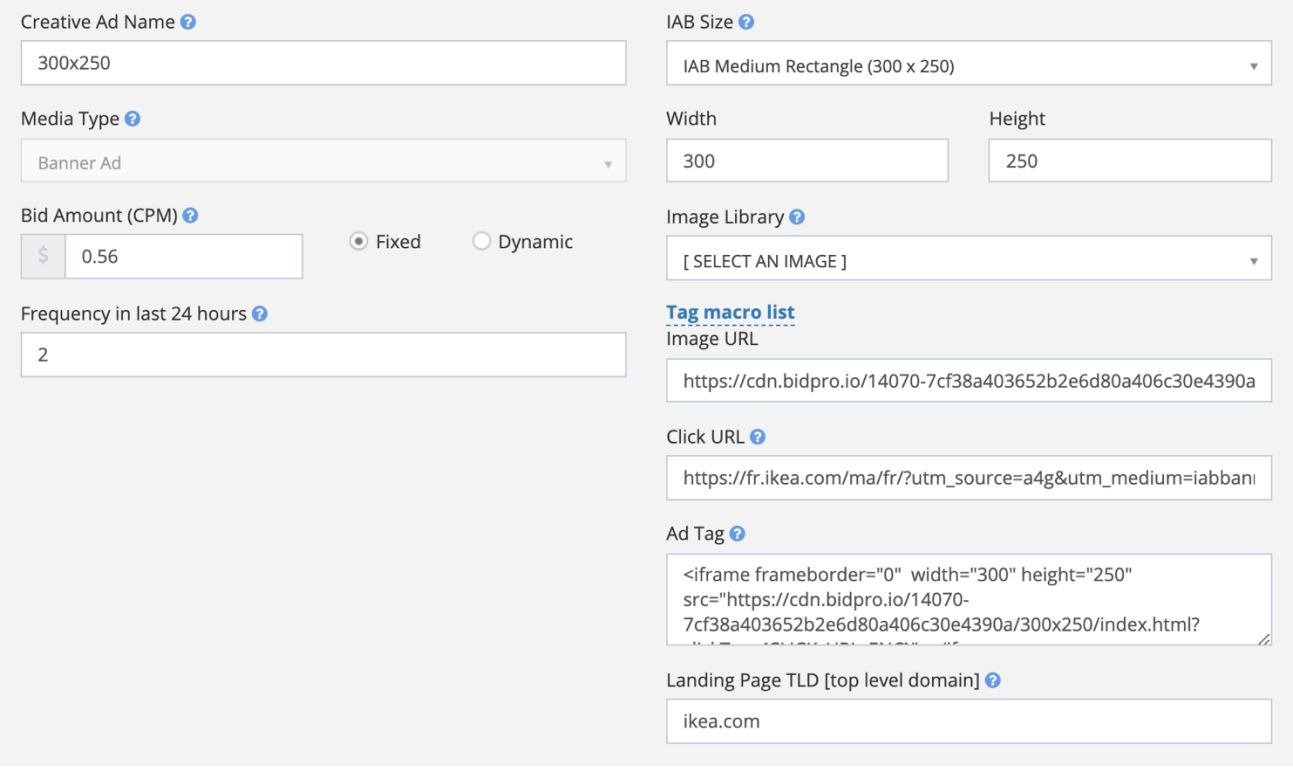


Рисунок 3.12 – Основні налаштування оголошення

За RTB специфікацією такі оголошення називають креативом.

Creative Ad Name – назва оголошення. Media Type – тип оголошення (банер, відео чи нативне).

Bid Amount (CPM) – ціна за 1000 показів.

Fixed / Dynamic – тип аукціону (за першою чи другою ціною). Доступно тільки адмінам.

Frequency in last 24 hours – дозволена кількість показів оголошення на одному ip-адресу за 24 години.

IAB Size – розмір рекламного оголошення за стандартом IAB.

Width/Height – можна змінити висоту та ширину власноруч (але це може вплинути на кількість трафіку).

Image Library – обрати картинку з попередньо завантажених.

Image URL – додати картинку через посилання.

Click URL – посилання, куди буде потрапляти користувач при натисканні на рекламне оголошення.

Ad Tag – те, у якому вигляді буде відповідь на запит. Доступно тільки адмінам.

Landing Page TLD – домен верхнього рівня для click url.

Окрім того, користувач з правами адміністратора має доступ до додаткових функцій, які дозволяють керувати платформою. В лівій частині меню з’являється розділ Admin, який включає:

– pending approval: Список оголошень, які необхідно перевірити та підтвердити їх можливість на показ;

– general report: загальна статистика по всіх користувачах на платформі.

– manage users: керування користувачами;

– manage transactions: керування транзакціями;

– manage SSP endpoints: керування підключенням SSP партнерів;

– settings: загальні налаштування платформи;

– global blacklist: глобальне блокування трафіку з певними параметрами;

– supply settings: керування трафіком, який отримують користувачі;

– search: пошук потрібної компанії чи оголошення за ID.

ВИСНОВКИ ДО РОЗДІЛУ 3

У даному розділі в процесі роботи було розглянуто важливі питання, які стосуються архітектури програмного продукту.

У ході роботи було описано засоби розробки, бібліотеки та сервіси для веб-застосунку і серверної частини продукту. В розділі також було наведено керівництво для користувача, яке описує основні функції та надає рекомендації щодо роботи із веб-застосуванням.

4 ОПИС СТАРТАП ПРОЕКТУ

4.1 Опис ідеї проекту

Ідея проекту полягає у створенні платформи для рекламодавців, яка відповідає вимогам специфікації OpenRTB. Такою платформа в системі Real-Time Bidding є DSP (Demand Side Platform). Завдяки DSP, рекламодавці можуть використовувати більш широкий спектр потенційних рекламних матеріалів і мати доступні налаштування для точного таргетингу. Мета DSP – якомога дешевше купити покази аудиторії, максимально відповідної запитам рекламодавця. DSP дозволяє рекламодавцям купувати аудиторію, а не конкретні місця для розміщення реклами.

Як і в контекстній рекламі, система на основі ключових показників ефективності допомагає оптимізувати такі показники як eCPC (effective Cost per Click) і eCPA (effective Cost per Action). Проте DSP має унікальні властивості:

– безпрецедентна швидкість роботи – DSP дозволяє агентствам і рекламодавцям здійснювати операції в реальному часі;

– широкі можливості таргетування аудиторії: прямий таргетинг – за соціально-демографічними, географічними, тимчасовим, поведінковими характеристиками, за інтересами тощо і латеральний таргетинг, який виділяє цільові групи під конкретну задачу з динамічно змінюваними критеріями в ході рекламної кампанії;

– формування детальних звітів по кожному показу, що відображають всі задіяні майданчики і результати їх роботи.

Зміст ідеї проекту викладений в таблиці 4.1.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигода для користувача
DSP (Demand Side Platform) – платформа, яка дає можливість рекламодавцям показувати свої оголошення певним людям за інтересами та іншими параметрами. Платформа універсальна – приймає рекламу, як сайтів так і мобільних додатків. Географічно реклама буде доступна для розміщення в Сполучених Штатах Америки.	1. Розміщення реклами	Потік цільової аудиторії на власний сайт чи додаток
	2. Відслідковування конверсій	Отримання статистичних даних по власному сайту чи додатку
	3. Ефективне використання рекламного бюджету компанії та залучення нових користувачів на сайт чи додаток	Отримання додаткового прибутку з конверсій

Основним функціоналом платформи є можливість показувати оголошення на всіх платформах (комп'ютери, мобільні пристрої, планшети) як на сайтах так і в додатках, наявність DMP та Cookie Sync, що дозволяє працювати з аудиторією (збирати про неї інформацію та аналізувати перегляди оголошень, покращується таргетинг), можливість створювати динамічні оголошення (це дозволяє рекламувати товари інтернет-магазинів поштучно), захист від шахрайства (основною проблемою є неякісний трафік або склікування бюджету, тому платформа оснащена захисною системою Anura, яку можуть використовувати користувачі, проте за це може здійматися додаткова плата) та «автооптимізація цін», яка дозволяє користувачу налаштувати параметри, які будуть впливати на ціну за яку вони будуть купувати рекламні місця або взагалі виключити деякі місця з показу.

Порівняємо провідні платформи у галузі: SmartyAds (компанія-розробник, продає платформи іншим, тому задає певну тенденцію на ринку), DoubleClick by Google (провідна платформа, оскільки її власником є Google, займає більше 70% всього ринку) та AppNexus (платформа, яка має унікальну систему підтвердження оголошень та рекламних місць, щоб дозволити їм брати участь в рекламному процесі, що дозволяє їй мати дуже якісний трафік). Порівняння характеристик платформи з конкурентами наведено в таблиці 4.2.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№	Техніко-економічні характеристики ідеї	(потенційні) концепції конкурентів				W	N	S
		A4G	SmartyAds	DoubleClick	AppNexus			
1	Доступні усі пристрої (як мобільні так і десктоп)	+	+	+	-			+
2	Наявність DMP	+	-	+	-			+
3	Наявність Cookie Sync	+	+	+	+		+	
4	Динамічна реклама по елементам	+	-	+	+			+
5	Збір даних про аудиторію	+	+	+	+		+	
6	Захист від шахрайства	+	-	+	+			+
7	Автооптимізація ціни	+	-	+	-			+

Продовження таблиці 4.2

8	Якісний трафік	+	-/+	+	+			+

Таким чином, можна побачити, що проект є конкурентоспроможним, тому що має усю необхідну функціональність та може скласти навіть провідним компаніям у галузі.

4.2 Технологічний аудит ідеї проекту

На розгляд виберемо три технології: Ad Management API, OpenRTB та розробка власної. Їх порівняння наведено в таблиці 4.3.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технологія реалізація	Наявність технологій	Доступність технологій
1	Торгівля рекламними оголошеннями в інтернеті.	Самостійна	-	-
2		Ad Management API	+	-
3		OpenRTB 2.5	+	+
Обрана технологія реалізації ідеї проекту: OpenRTB 2.5				

Технологія OpenRTB є доступною та досить розповсюдженою. OpenRTB є стандартом в галузі інтернет-реклами, також є повний опис специфікації. Запити мають вигляд JSON об’єкта з певними параметрами.

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначимо можливості ринку та загрози, які можуть перешкодити успішності проекту. Також, враховуючи стан ринка, потреби користувачів та пропозиції конкурентів, можна спланувати подальший розвиток платформи. Попередня оцінка ринку маркетингу в Інтернеті наведена в таблиці 4.4.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

№	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	>15
2	Загальний обсяг продаж, ум.од	$ 50 млдр./рік
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу	Не має
5	Специфічні вимоги до стандартизації та сертифікації	Підтримка специфікації OpenRTB v2.5
6	Середня норма рентабельності в галузі, %	72%

За попередніми оцінками, ринок є дуже привабливим.

Визначимо потенційні групи клієнтів, їх характеристики та сформуємо орієнтовний перелік вимог для кожної групи (таблиця 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних груп клієнтів	Вимоги споживачів
1	Реклама власного бренду, сайту, додатку, товарів для продажу	Власники компаній	Персональний менеджер для клієнта. Ефективність рекламної компанії.	Персональний підхід.
2		Відділи маркетингу у крупних компаніях	Помірне розподілення рекламного бюджету	Проста реєстрація.

Проведемо аналіз ринкового середовища: складемо таблиці факторів, що сприяють впровадженню та розвитку проекту (таблиця 4.7), та факторів, що перешкоджають цьому (таблиця 4.6).

Таблиця 4.6 – Фактори загроз

№	Фактор	Змість загрози	Можливі реакції компанії
1	Велика конкуренція	На ринку більше 15 гравців	Впровадження додаткової функціональності, підвищення впізнаваємості бренду

Таблиця 4.7 – Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Обмежена кількість платформ з достатнім функціоналом	Клієнти не задоволені функціональністю більшості платформ	Розвиток платформи, додавання нового функціоналу, в залежності від потреб користувачів.
2	Зростання попиту в галузі	Поява місць для нових гравців	Отримання певної долі ринку

Визначимо загальні риси конкуренції на ринку у таблиці 4.8.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
Тип конкуренції: Олігополія	Існує два основних гравця на ринку: DoubleClick та AppNexus	Дотримання стандартів якості, які пропонують ведучі компанії. Зниження ціни на розміщення оголошень. Володіння унікальною базою користувачів.

Продовження таблиці 4.8

За рівнем конкурентної боротьби: Національний	Ринок, на який націлена компанія, є світовим.	Запропоновувати ціну на показ оголошень, меншу ніж у конкурентів
За галузевою ознакою: Внутрішньогалузева	Компанія спеціалізується на певній галузі – реклама в інтернеті	Поява певної асоціації з брендом у користувачів. Фокус на певній спеціалізації дозволяє якісніше розробляти продукт.
Конкуренція за видами товарів: товарно-видова	Пропонується оголошення банерного, нативного та відео типів	Збільшення кількості користувачів
Характер конкурентних переваг	Ціна, відносно конкурентів, є меншою, проте якість залишається на рівні	Збільшення кількості користувачів
Інтенсивність марочна	Платформа має свій бренд	Впізнаваємість компанії та підвищення репутації

Проведемо більш детальний аналіз конкуренції в галузі за моделлю 5 сил М. Портера (таблиця 4.9).

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	DoubleClick, AppNexus	SmartyAds	Постачальники досить різноманітні і тому сильного впливу не несуть	Споживачі несуть більш впливове значення ніж постачальники	Фактор-загрози з боку є зменшення ціни на показ оголошень
Висновки:	Оскільки ринок є зростаючим, сильної боротьби між прямими конкурентами не існує.	Є можливість появи нових конкурентів у зв'язку зі зростанням ринку	Постачальники висувають вимогу щодо прозорості ведення транзакцій.	Клієнти можуть висувати вимоги щодо функціональності платформи.	Зменшення долі ринку

Перелік факторів конкурентоспроможності наведені в таблиці 4.10 та, на основі цих факторів, проведено аналіз сильних на слабких сторін стартап-проекту.

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Підтримка усіх видів пристроїв	Багато платформ спеціалізуються тільна на певному виді пристроїв: або мобільні, або комп’ютери
2	Автооптимізація ціни	Дозволяє ефективніше використовувати бюджет користувача
3	Динамічна реклама	Розширює охват ринку

Порівняння платформи A4G DSP з найблищим конкурентом на ринку США є компанія SmartyAds наведено у таблиці 4.11.

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін «A4G DSP»

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з SmartyAds						
			-3	-2	-1	0	1	2	3
1	Підтримка усіх видів пристроїв	5					+		
2	Автооптимізація ціни	20							+
3	Динамічна реклама	15					+		

Фінальним етапом ринкового аналізу можливостей провадження стартап-проекту є SWOT-аналіз (матриця аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) на основі виділених ринкових загроз та можливостей, та сильних і слабких сторін). SWOT-аналіз наведений у таблиці 4.12.

Таблиця 4.12 – SWOT-аналіз стартап-проекту

Сильні сторони: - Стабільна робота платформи; - Високий рівень сервісу; - Персональний підхід до клієнта;	**Слабкі сторони:** - Сильна конкуренція та популярність існуючих гравців;
Можливості: - Отримання певного відсодку ринку; - Розширення географії продажів;	**Загрози:** - Втрата або не можливість залучати нових користувачів; - Складність отримати домовленості з провідними SSP; - Зниження доходів клієнтів;

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Альтернативи впровадження проекту описано у таблиці 4.13.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Підключення провідних SSP	0.7	До 1 року
2	Залучення крупних клієнтів	0.5	До 2 років
3	Розширення географії продажів	0.8	До 2 років

4.4 Розроблення ринкової стратегії проекту

Визначимо стратегію охоплення ринку: опис цільових груп потенційних споживачів у таблиці 4.14.

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Власники компаній	Готові	Високий	Середня	Середня складність
2	Маркетинговий відділ компаній	Готові	Високий	Середня	Середня складність

Продовження таблиці 4.14

Які цільові групи обрано: Вибрано усі дві запропоновані групи, оскільки клієнтів в кожній групі досить складно знайти та час конверсій дуже тривалий, тому обираються усі споживачі сегменту.

Враховуючи результати аналізу потенційних груп споживачів, товар буде пропонуватися для обох груп, оскільки підхід до клієнтів у цих групах має багато спільного, а, також, вони складають значну частину ринку. Таким чином, буде використано стратегію диференційного маркетингу.

Визначимо базову стратегію розвитку проекту (таблиця 4.15).

Таблиця 4.15 – Визначення базової стратегії розвитку

№	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Розвиток додаткової функціональності	Стратегія диференційованого маркетингу	Особистий підхід до клієнтів	Залучення нових користувачів на ринку

Проведемо аналіз для вибору стратегії конкурентної поведінки у таблиці 4.16.

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Проект не є «першопрохідцем»	Компанія проводить активну маркетингову компанію, тому будуть залучатись як нові користувачі, так і ті, які вже знайомі з галуззю.	Деякі характеристики є обов’язковими для вдалої конкуренції: наявність показувати оголошення аудиторії за інтересами, тому саме на цьому буде зосереджена увага.	Маркетингова компанія для залучення нових користувачів.

Враховуючи попередній аналіз, розробимо стратегію позиціонування (таблиця 4.17). Сформуємо ринкову пропозицію (комплексну асоціацію), за якою споживачі мають ідентифікувати торгівельну марку.

Таблиця 4.17 – Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Платформа має підтримувати функцію збереження аудиторії за інтересами.	Персональна домавленість з власниками компаній та маркетинговими відділами.	Маркетинг спрямований на новостворені компанії.	- Доступна динамічна реклама; - Реклама поставляється на всі пристрої; - Можливість динамічно контролювати та корегувати процес показу оголошень.

Результатом є узгоджена система рішень щодо ринкової поведінки стартап-компанії, яка визначатиме напрями роботи стартап-компанії на ринку. Отже, робота стартап-компанії на ринку повинна бути спланована орієнтовано таким чином: за стратегією диференціації виконаний і буде поширюватись товар відмінний за властивостями від своїх аналогів, дотримуючись у конкурентній

поведінці стратегії «виклику лідера», тобто випускається один товар для усіх можливих споживачів.

4.5 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Сформуємо маркетингову концепцію товару, який отримає споживач (таблиця 4.18).

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Ефективне використання бюджету рекламних компаній	Процес динамічного контролю процесу показу оголошень	Корегування ціни для кожного рекламного місця або повністю його блокування, враховуючи показники статистики

Розроблена трирівнева маркетингова модель товару: уточняються ідея продукту, його фізичні складові, особливості процесу його надання (таблиця 4.19).

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Реклама сайту чи додатку для залучення нових користувачів		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Вид оголошень (banner, video, native);	М	Тл

Продовження таблиці 4.19

	2. Динамічне / не динамічне;	М	Тх
	3. Тип трафіку, який підтримується (mobile, desktop, tablet)	М	Тл
	Якість: стандарт OpenRTB v2.5, трафік та оголошення мають буди якісними (не містити шахрайського та обмежувального контенту).		
	Пакування відсутне.		
	Марка: A4G DSP		
III. Товар із підкріпленням	Товару до продажу не передбачається.		
	Товар після продажу може бути додаткове рекламне оголошення іншого типу.		
За рахунок чого потенційний товар буде захищено від копіювання: Унікальність бренду, база підключених партнерів SSP.			

Визначимо цінові межи, якими необхідно куруватися при встановлені ціни на потенційний товар.

Насправді ціна на показ оголошення залежить від багатьох факторів: конкуренція галузі оголошення, конкуренція між подібними оголошеннями на самій платформі, мінімальна ціна, яку приймає SSP партнер, тощо.

Визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари-аналоги, а також аналіз рівня доходів цільової групи споживачів описано в таблиці 4.20.

Таблиця 4.20 – Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
$0.05 - $5	$0.05 - $5	$1000 - $10000 / міс.	$0.05 - необмежено

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (таблиця 5.21): проводити збут власними силами або залучати сторонніх посередників, вибір та обґрунтування оптимальної глибини каналу збуту, вибір та обґрунтування виду посередників. Специфікація RTB вже передбачає певну систему продажу рекламних оголошень та покупку рекламних місць методом аукціону другої ціни.

Таблиця 4.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Клієнти хочуть рекламуватись на якісних сайтах та додатках з найбільшою ефективністю	Постачальник має пропонувати якісний трафік	Глибина може коливатися від 1 до нескінченності (теоретично, може бути нескінченна кількість посередників)	Оптимальною системою є постачальник (SSP) та наша платформа (DSP) без посередників (Ad Exchange)

Опишемо концепцію маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування та визначену поведінку користувачів у таблиці 4.22.

Таблиця 4.22 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуютьс я цільові клієнти	Ключові позиції, обрані для позиціонуванн я	Завдання рекламного повідомлення	Концепція рекламного звернення
Клієнти шукають як вкласти гроші в рекламу з найбільшою ефективніст ю	Пошукові системи, соціальні мережі	Пропозиція якісного трафіку для клієнтів та перевірка клієнтів на	Унікальна функціональність , ефективність реклами користувачів	Донести до користувача суть продукту, його якість, та залучити якомога більше зацікавлени х клієнтів

Таким чином, було розроблено маркетингову програму, що включає в себе концепцію товару, його збуту, просування та попередній аналіз можливостей ціноутворення, спираючись на цінності та потреби клієнтів, конкурентні переваги платформи, стан та динаміку ринкового середовища, та відповідну обрану альтернативу ринкової поведінки.

Висновки до розділу 4

Враховуючи проведений аналіз, можна зробити висновок, що проект має можливість доброго входження на ринок, оскільки наявний великий попит та динаміка зростання ринку, обрано стратегію диференційованого маркетингу для охоплення більшої кількості потенційних клієнтів. Варто враховувати велику конкуренцію, проте платформа має значну перевагу, що спрощує залучання нових клієнтів. Є доцільним, також, подальше розширення функціоналу платформи.

ВИСНОВКИ

Для досягнення мети було виконано наступні завдання:

a) Проаналізовано ринок реклами в Інтернеті, описано бізнес-процеси для технології продажу і покупки показів реклами в Інтернеті на основі автоматизованого аукціону, її платформи-учасники, їх стандарти та процес взаємодії.

b) Описано розвиток RTB, переваги над традиційною системою, основні принципи, архітектура, а також задачі, які вона вирішує.

c) Розглянуто питання застосування аукціону другої ціни в RTB, формулу визначення ціни за клік та ймовірність натискання користувачем на рекламне оголошення в залежності від розташування оголошення на сайті. Описано випадок виникнення «Трагедії спільного» в RTB.

d) Проаналізовано підходи до обрання потрібного оголошення в системі торгів у реальному часі на платформі DSP та запропоновано стохастичний підхід до обрання оголошення. Було підраховано математичне сподівання стохастичного та класичного підходів.

Подальші напрямки дослідження мають бути направлення на удосконалення визначення ціни під час аукціону для кожного оголошення, з урахуванням показників якості обраних оголошень, параметрів трафіку та потреб користувача платформи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) AdCOM Specification. – [Електронний ресурс]. – Режим доступу: https://github.com/InteractiveAdvertisingBureau/AdCOM/blob/master/AdCOM%20v1.0%20FINAL.md

2) Основні принципи OpenRTB. – [Електронний ресурс]. – Режим доступу: https://iabtechlab.com/standards/openrtb/

3) Основні поняття RTB. – [Електронний ресурс]. – Режим доступу: http://konverta.ru/how

4) Parks Associates: RTB will change online display advertising RTB. – [Електронний ресурс]. – Режим доступу: https://www.dmnews.com/customer-experience/news/13059564/parks-associates-rtb-will-change-online-display-advertising

5) Sell ads without damaging privacy. – [Електронний ресурс]. – Режим доступу: https://www.nytimes.com/2019/06/19/opinion/facebook-google-privacy.html

6) Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz: «Internet Advertising and the Generalized Second-Price Auction: Selling Billions of Dollars Worth of Keywords». American Economic Review 97(1), 2007. DOI 10.1257/aer.97.1.242.

7) А.Савватеев. Аукцион второй цены: доказательство теоремы Викри. Московский физико-технический институт. – [Електронний ресурс]. – Режим доступу: https://ru.coursera.org/lecture/gametheory/auktsion-vtoroi-tsieny-dokazatiel-stvo-tieoriemy-vikri-PKVzZ.

8) Основи RTB від компанії between exchange. – [Електронний ресурс]. – Режим доступу: http://rtb-media.ru/wiki-auction/.

9) Основні поняття та опис загальноприйнятої архітектури RTB. – [Електронний ресурс]. – Режим доступу: https://habr.com/ru/post/169267/.

10) Загальний опис принципів відбору рекламних оголошень в Яндекс. – [Електронний ресурс]. – Режим доступу: https://habr.com/ru/company/yandex/blog/246399/.

11) Г.С.Фінін. Методи розв'язання ігрових задач. (Рекомендовано МОН як підручник для студентів ВНЗ). – К.: КНУБА – 2014 – с. 36 – 188 с.

12) Банковское дело: Учебник для вузов. // Под ред. Г. Белоглазовой, Л. Кроливецкой. – 2-е изд. – СПб.: Питер. – 2010. – с. 240. – 400 с.

13) Vickrey, William. Counterspeculation, Auctions, and Competitive Sealed Tenders // The Journal of Finance – 1961 – Vol. 16, № 1. – P. 8 – 37. – DOI 10.1111/j.1540-6261.1961.tb02789.x.

14) Офіційна документація OpenRTB від IAB. – [Електронний ресурс]. – Режим доступу: https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/OpenRTB v3.0 FINAL.md.

15) Neha Narkhede. Kafka: The Definitive Guide: Real-Time Data and Stream Processing at Scale, 2017.

16) Vazirani, Vijay V. Algorithmic Game Theory. Cambridge, UK: Cambridge University Press, 2007, ISBN 0-521-87282-0.

17) Arthur P. L. Hypermedia History: Changing Technologies of Representation for Recording and Portraying the Past, 2006.

18) Roy Fielding. Representational State Transfer (REST), 2000.

19) Roy Fielding. Architectural Styles and the Design of Network-based Software Architectures, 2000.

20) Weinan Zhang. Optimal Real-Time Bidding for Display Advertising, 2016.

21) Steven Kuhn. Prisoner’s dilemma. In Edward N. Zalta, editor, The Stanford Encyclopedia of Philosophy, 2014.

22) Anatol Rapoport and Albert M Chammah. Prisoner's dilemma: A study in conflict and cooperation, volume 165. University of Michigan press, 1965.

23) Google. The arrival of real-time bidding, 2011.

24) Daniel C Fain and Jan O Pedersen. Sponsored search: A brief history. Bulletin of the American Society for Information Science and Technology, 32(2):12–13, 2006.

25) Benjamin Edelman, Michael Ostrovsky, and Michael Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. Technical report, National Bureau of Economic Research, 2005.

26) Avi Goldfarb and Catherine Tucker. Online display advertising: Targeting and obtrusiveness. Marketing Science, 2011.

27) Shuai Yuan, Jun Wang, and Xiaoxue Zhao. Real-time bidding for online advertising: measurement and analysis. In Proceedings of the Seventh International Workshop on Data Mining for Online Advertising, page 3. ACM, 2013.

28) Romer Rosales, Haibin Cheng, and Eren Manavoglu. Post-click conversion ´ modeling and analysis for non-guaranteed delivery display advertising. In Proceedings of the fifth ACM international conference on Web search and data mining, 2012.

29) Arpita Ghosh and Aaron Roth. Selling privacy at auction. Games and Economic Behavior, 2013.

30) Lampros C Stavrogiannis, Enrico H Gerding, and Maria Polukarov. Auction mechanisms for demand-side intermediaries in online advertising exchanges, 2014.

31) Ye Chen, Pavel Berkhin, Bo Anderson, and Nikhil R Devanur. Real-time bidding algorithms for performance-based display ad allocation, 2011.

32) Ying Cui, Ruofei Zhang, Wei Li, and Jianchang Mao. Bid landscape forecasting in online ad exchange marketplace, 2011.

33) Michael Durbin. All about high-frequency trading. McGraw Hill Professional, 2010.

34) Bowei Chen, Shuai Yuan, and Jun Wang. A dynamic pricing model for unifying programmatic guarantee and real-time bidding in display advertising, 2014.

35) Michel Crouhy, Dan Galai, and Robert Mark. The essentials of risk management, volume 1, 2006.

36) Benjamin Edelman. The design of online advertising markets, 2010.

37) Jun Wang and Bowei Chen. Selling futures online advertising slots via option contracts, 2012.

38) John Hull. Risk Management and Financial Institutions, volume 733. Wiley, 2012.

39) Steve Hogan, Robert Jarrow, Melvyn Teo, and Mitch Warachka. Testing market efficiency using statistical arbitrage with applications to momentum and value strategies, 2004.

40) Ruggiero Cavallo, R Preston Mcafee, and Sergei Vassilvitskii. Display advertising auctions with arbitrage, 2015.

41) Vijay Krishna. Auction theory. Academic press, 2009.

42) REST – [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/REST

43) Аукцион Викри – [Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Аукцион_Викри

44) OpenRTB specification – [Електронний ресурс]. – Режим доступу: https://github.com/InteractiveAdvertisingBureau/openrtb/blob/master/OpenRTB%20v3.0%20FINAL.md#appendixd_versioning

45) Ad Management API v1.0 – [Електронний ресурс]. – Режим доступу: https://github.com/InteractiveAdvertisingBureau/AdManagementAPI/blob/master/Ad%20Management%20API%201.0%20FINAL.md

46) Сангит Пол Чаудари, Маршалл ван Альстайн и Джеффри Паркер. Революция платформ. Как сетевые рынки меняют экономику – и как заставить их работать на вас. – 2017 – с. 338.

47) Weinan Zhang. Optimal Real-Time Bidding for Display Advertising, 2016.

48) REST API – [Електронний ресурс]. – Режим доступу: wiki.cuspu.edu.ua/index.php/REST_API

49) RESTful – [Електронний ресурс]. – Режим доступу: uk.m.wikipedia.org/wiki/RESTful

50) База даних Aerospike. – [Електронний ресурс]. – Режим доступу: uk.m.wikipedia.org/wiki/Aerospike

51) Aerospike for NodeJS. – [Електронний ресурс]. – Режим доступу: aerospike.com/docs/client/nodejs/usage/

52) Budget-Constrained Real-Time Bidding Optimization – [Електронний ресурс]. – Режим доступу: dl.acm.org/doi/abs/10.1145/3375393

53) Мережевий ефект – [Електронний ресурс]. – Режим доступу: uk.wikipedia.org/wiki/Мережевий_ефект

54) Тестування REST-систем – [Електронний ресурс]. – Режим доступу: github.com/s0nerik/Graduation/blob/master/sections/3.tex

55) Паттерн MVC – [Електронний ресурс]. – Режим доступу: uk.wikipedia.org/wiki/Модель-вид-контролер

56) Real-Time Bidding with Multi-Agent Reinforcement Learning. – [Електронний ресурс]. – Режим доступу: arxiv.org/pdf/1802.09756.pdf

ДОДАТОК А
ГРАФІЧНІ ЗОБРАЖЕННЯ

ДОДАТОК Б
ПРОГРАМНИЙ КОД ОБРОБКИ ЗАПИТУ

```
let contentObjBody = HttpFunctions.parseBodySSP(req.url, bodyString, isOpenRtb);
if (contentObjBody === false) {
    logger.warn('Error parseBodySSP():', partner.seat);
    HttpFunctions.echoBadEnd(res, 'Error parse body request');
    return false;
}

// Check type of request -> banner, native, video, exadsBanner, exadsPop, popunder
let requestType = HttpFunctions.getTypeRequest(isOpenRtb, contentObjBody, partner,
isPushAd);
if (requestType === false) {
    logger.warn('Error getTypeRequest():', partner.seat);
    HttpFunctions.echoBadEnd(res, 'Error not found request type');
    return false;
}

if (requestType === 'popunder') {
    let resultPreparePopunder = HttpFunctions.preparePopunder(contentObjBody);
    if (resultPreparePopunder === false) {
        HttpFunctions.echoBadEnd(res, 'Invalid popunder request');
        return false;
    }
    contentObjBody = resultPreparePopunder; //bode for popunder
    resultPreparePopunder = null;
}

if (requestType === 'push') {
    let resultPreparePush = HttpFunctions.preparePush(contentObjBody);
    if (resultPreparePush === false) {
        HttpFunctions.echoBadEnd(res, 'Invalid push request');
        return false;
    }
    contentObjBody = resultPreparePush;
    resultPreparePush = null;
}

//Check body for SSP request
let checkResult = HttpFunctions.checkRequiredFields(contentObjBody, requestType);
if (checkResult !== true) {
    HttpFunctions.echoBadEnd(res, checkResult);
    return false;
}
checkResult = null;

if (requestType === 'exadsBanner' || requestType === 'exadsPop') {
    contentObjBody = HttpFunctions.prepareExAds(contentObjBody);
}

// Object parameters for transmission
let options = {
    'content': contentObjBody,
    'requestType': requestType,
    'partner': partner,
    'isOpenRtb': isOpenRtb,
```

```
    'res': res,
    tlsRequestData: new Map(),
    tlsCache: new Map(),
    debugMode: debugMode,
    debugOptions: {
        'ssp': partner['seat'],
        'creatives': []
    },
    creativesDebugCheckCounter: 0,
    wasSend: false
};

if (requestType === 'jsonPop') {
    options.requestType = 'popunder';
    options.popuderJsonFormat = true;
}

// clear variables
contentObjBody = null;
requestType = null;
partner = null;

//Clear request content
if (HttpFunctions.clearValuesRequest(options) === false) {
    noBid('clearValues');
    HttpFunctions.echoNoBidEnd(res);
    return false;
}

if (!HttpFunctions.setTrustData(options)) {
    //todo change no bid reason
    noBid('noGeo');
    HttpFunctions.echoNoBidEnd(res);
    return false;
}

// Get User Agent object
options['objUaParser'] = HttpFunctions.parseUserAgent(options.content['device']['ua']);
if (options['objUaParser'] === false) {
    logger.warn('Error parseUserAgent():', options.partner['seat']);
    HttpFunctions.echoBadEnd(res, 'Error parse User Agent');
    return false;
}

options["pmpType"] = HttpFunctions.getPmpType(options);
options["pmpDealsHashInReq"] = HttpFunctions.getPmpDealsHashArray(options);
options["pmpDealsHash"] = new Map();

options["pmp"] = false; //options.content["imp"][0].hasOwnProperty('pmp');

options['isSecureReq'] = HttpFunctions.checkIsSecure(options);

//Add status of the type of traffic (Device type definition)
options['isDesktopApp'] = 0;
options['isDesktopWeb'] = 0;
options['isMobileApp'] = 0;
options['isMobileWeb'] = 0;
options['isTablet'] = 0;
options['isSmartphone'] = 0;

extracted(options);
```

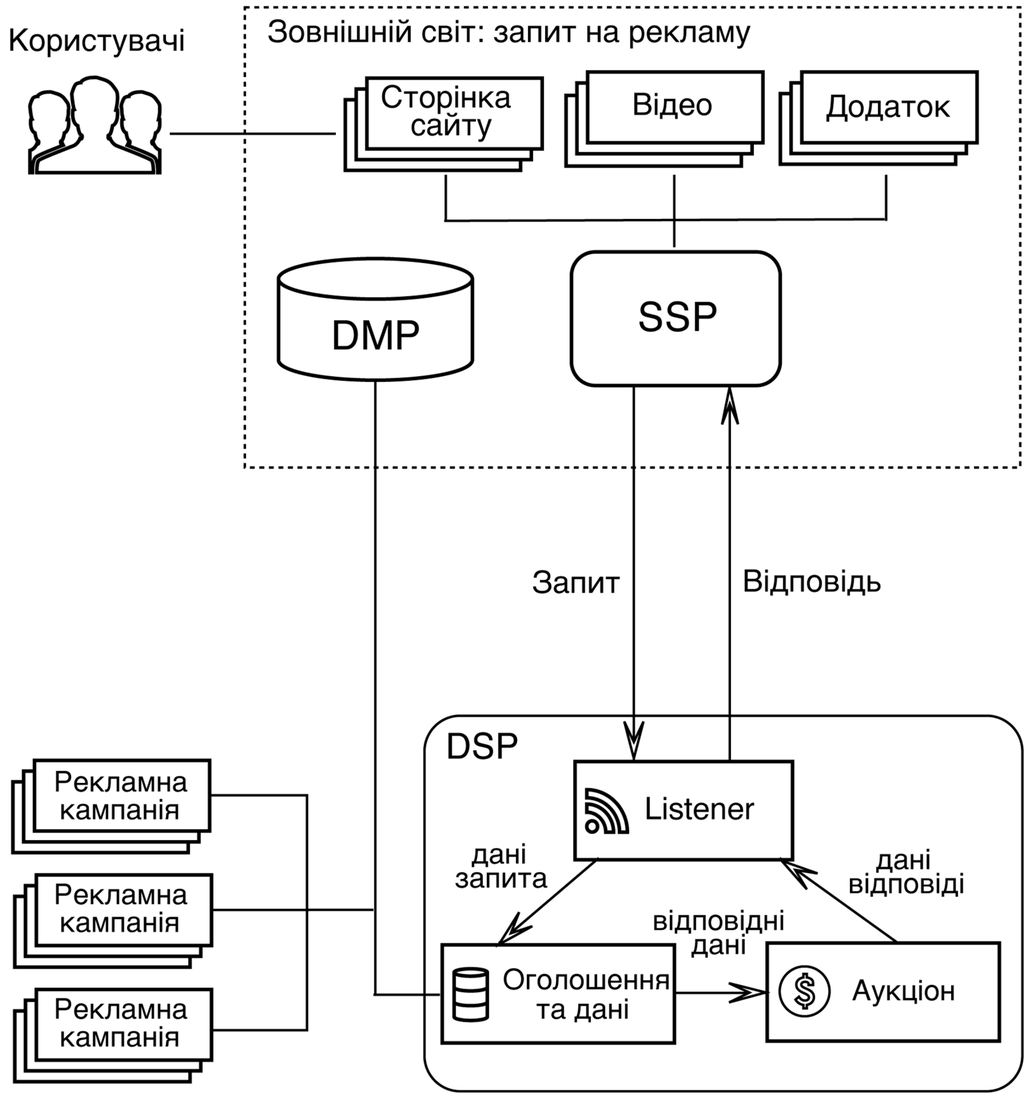
```
let isMobile = HttpFunctions.checkIsMobile(options);
let isApp = HttpFunctions.checkIsApplication(options);
let isDesktop = (isMobile === 1) ? 0 : 1;

// Add dop params
options['isDesktop'] = isDesktop;
options['isMobile'] = isMobile;
options['isApp'] = isApp;

// Desktop
if (isDesktop === 1) {
    //Desktop App
    if (isApp === 1) {
        options['isDesktopApp'] = 1;
    } else {
        //Desktop Web
        options['isDesktopWeb'] = 1;
    }
} else if (isMobile === 1) {
    // Mobile App
    if (isApp === 1) {
        options['isMobileApp'] = 1;
    } else {
        //Mobile Web
        options['isMobileWeb'] = 1;
    }

    //Tablet or Smartphone
    options['isTablet'] = HttpFunctions.checkIsTablet(options);
    options['isSmartphone'] = (options['isTablet'] === 1) ? 0 : 1;
}
isMobile = null;
isApp = null;
isDesktop = null;

options.auUserId = getUserFirstId(options);

// Checking Global Ban list
let resultGlobalList = HttpFunctions.checkGlobalBanList(options);

if (resultGlobalList === false) {
    noBid('globalBan');
    HttpFunctions.echoNoBidEnd(res);
    return false;
}
resultGlobalList = null;

(new Promise(resolve => {
    kafka.push(options, resolve);
}))
    .then(f => f)
    .catch(err => logger.error('Kafka:', err));

// Get Banners
if (!getBanners(options, res)) { return false; }

//save creatives for debugging
if (debugMode) {
    for (let i = 0; i < options.arrListKeysCreatives.length; ++i) {
        options.debugOptions.creatives.push({
```

```
            'creativeId': options.arrListKeysCreatives[i],
            'filters': [],
            'success bid': false
        })
    }
}

let generalFlow = [
    (callback) => {
        const userAudId = getUserFirstId(options);
        fillUserInfo(userAudId, (userInfo) => {
            if (!options.content.user) {
                options.content.user = {};
            }
            if (userInfo && userInfo.gender) {
                options.content.user.gender = userInfo.gender;
            }
            if (userInfo && userInfo.yob !== 0) {
                options.content.user.yob = userInfo.yob;
            }
            return callback(null, options);
        });
    },
    (options, callback) => {

        options.userInfo = null;
        options.retargetingFilterCache = new Map();

        const user = options.content.user;
        const ssp = options.partner;

        const userAudId = getUserFirstId(options);
        getUserAudInfo(userAudId, (audiences) => {
            if (typeof (user) === 'object' && ssp.useUserMatching) {

                const idFieldName = ssp.umIdFieldName || 'id';
                const userId = user[idFieldName] || null;

                getUserInfo(userId, audiences, (userInfo) => {
                    options.userInfo = userInfo;
                    return callback(null, options);
                });
            } else {
                options.userInfo = {audienceIDs: audiences};
                return callback(null, options);
            }
        });

    },
    // Filtering for Banners
    (options, callback) => {
        options.deviceCharacteristics = HttpFunctions.getDeviceCharacteristics(options);
        options.reqAdditionalOps = HttpFunctions.setAdditionalValues(options);
        if (HttpFunctions.checkGlobalPlacementId(options.reqAdditionalOps.placementId)) {
            noBid('globalPlacementIdBan');
            HttpFunctions.echoNoBidEnd(res);
            return false;
        }

        options.readyDataBanners = {}; //ready data after prepare data
        HttpFunctions.asyncGeneralFilter(options, callback);
```

```
        return false;
    },

    // Filtering for Frequency Cap Impressions
    (options, callback) => {
        HttpFunctions.filterImpressFrequencyCap(options, callback);
        return false;
    },

    // Filtering for Frequency Cap Bid Request
    (options, callback) => {
        return callback(null, options); //next callback
    },

    ////////////////////// Filtering for Type Request
    (options, callback) => {

        if (options.requestType === 'banner') {
            HttpFunctions.asyncBannerFilter(options, callback);
        } else if (options.requestType === 'native') {
            HttpFunctions.asyncNativeFilter(options, callback);
        } else if (options.requestType === 'video') {
            HttpFunctions.asyncVideoFilter(options, callback);
        } else {
            callback(null, options);
        }
        return false;
    },

    auction()
];

//Get answer for server
Async.waterfall(generalFlow, (err) => {

    generalFlow = null; //clear

    if (!checkError(err, res)) {
        return false
    }

    if (!options.bidResponseBanner) {
        noBid('noBanner');
        HttpFunctions.echoNoBidEnd(res);
        return false;
    }

    //Bid Response
    let resultBidRequest = HttpFunctions.getBidResponse(options);

    //good Bid Response
    if (resultBidRequest !== false) {
        if (!HttpFunctions.checkBidFloorRemoteCreative(options, resultBidRequest)) {
            return false;
        }
        const bidRequestResponseLogger =
Container.get(constants.container.SSP_REQUESTS_LOGGER);
        bidRequestResponseLogger.log(options, bodyString, resultBidRequest);
        //// requestType ////
        //1)banner      - JSON
        //2)native      - JSON
```

```
        //3)video       - JSON
        //4)exadsBanner - JSON
        //5)exadsPop    - JSON
        //6)popunder    - XML

        (new Promise(resolve => {
            kafka.pushBid(options, resolve);
        }))
            .then(f => f)
            .catch(err => logger.error('Kafka Bid:', err));

        if (debugMode) {
            if (!options.wasSend) {
                HttpFunctions.echoGoodEnd(res, JSON.stringify(options.debugOptions));
            }
            return false;
        }

        const ContentType = (options.requestType === 'popunder')
            ? options.popuderJsonFormat
                ? 'json'
                : 'xml'
            : 'json';

        //Gzip response
        if (req.headers['accept-encoding'] && req.headers['accept-
encoding'].match(/\bgzip\b/i)) {
            HttpFunctions.echoGzipResponse(res, resultBidRequest, ContentType);
            return false;
        }

        //XML headers
        if (ContentType === 'xml') {
            HttpFunctions.echoXmlEnd(res, resultBidRequest);
            return false;
        }

        // JSON headers
        HttpFunctions.echoGoodEnd(res, resultBidRequest);
        return false;
    }

    resultBidRequest = null;

    // Echo Bad End
    HttpFunctions.echoBadEnd(res);
    return false;
});
```

ВЗАЄМОДІЯ КОНМПОНЕНТІВ В СИСТЕМІ REAL-TIME BIDDING



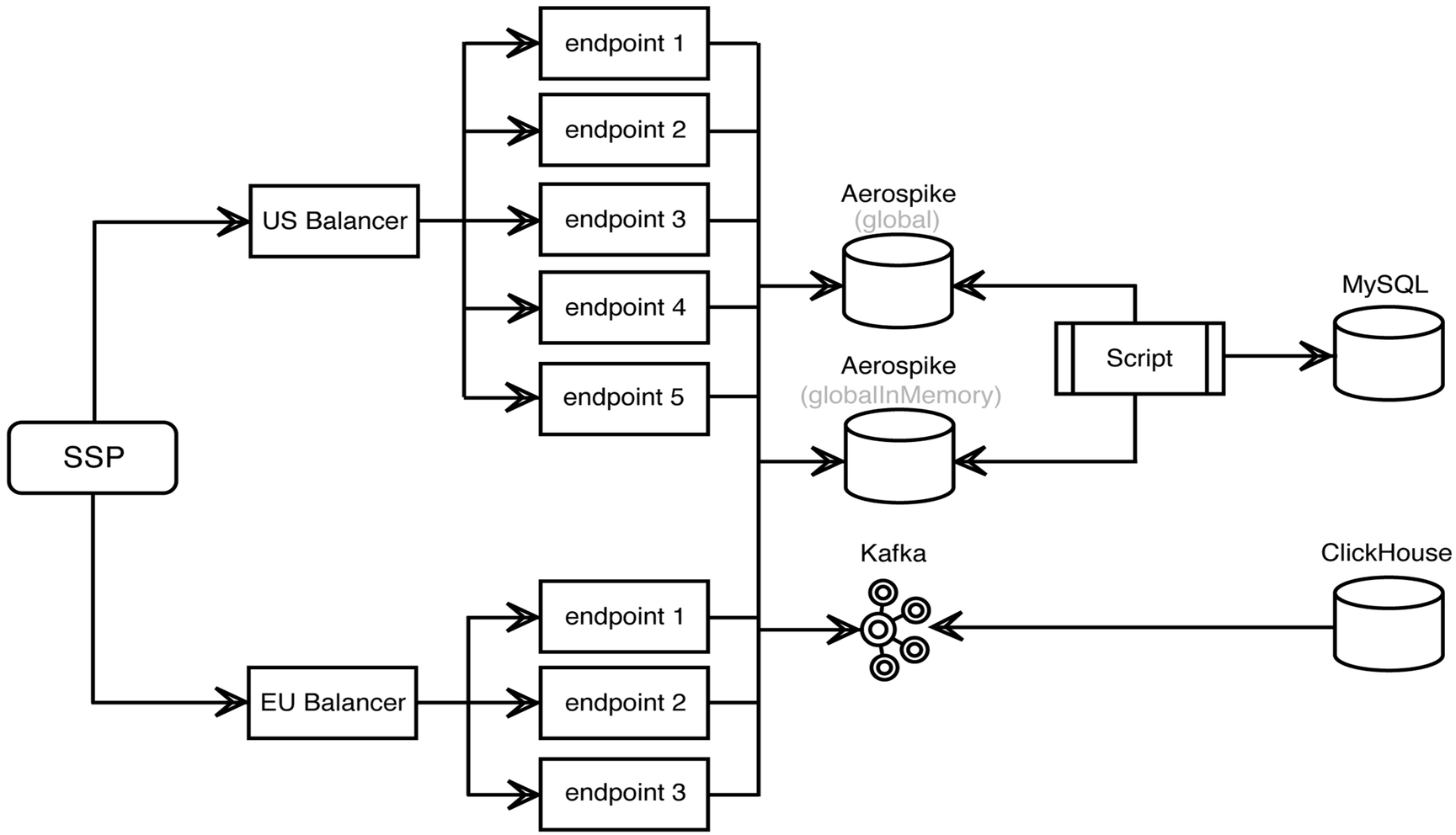
Демонстаціний плакат до магістерської диссертації

Математичне та програмне забезпечення організації аукціонів процесу продажу рекламних місць в реальному часі

Виконав Котляр І. С.

Керівник Попенко В. Д.

СЕРВЕРНА АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Демонстаціний плакат до магістерської диссертації

Математичне та програмне забезпечення організації аукціонів процесу продажу рекламних місць в реальному часі

Виконав Котляр І. С.

Керівник Попенко В. Д.