



**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Інформаційні системи

Викладач: к.т.н., доц. Саяпіна Інна Олександрівна

План заняття:

- ▶ Case#2
- ▶ Case#3



Case #2



The background features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. These shapes are primarily located on the right side of the image, creating a modern, tech-oriented aesthetic.

Smartcar.auto

The Real Autonomous Car

Smartcar.auto

- Виробляє автономні системи для транспортних засобів
- Наразі має понад 10 000 автомобілів на дорогах
- До кінця року очікується понад 200 000 автомобілів
- Необхідно надійно отримувати телеметрію від автомобілів і відображати дані про них



Вимоги

Функціональні

Що система повинна
робити

1. Web орієнтованість
2. Отримувати телеметрію від автомобілів (місцезнаходження, швидкість, поломки тощо)
3. Зберігати телеметрію в постійному сховищі
4. Відображати інформаційні панелі із узагальненням даних
5. Виконувати аналіз даних

Нефункціональні

Як гарно вона це повинна
робити

НФВ – Що нам відомо

1. Система інтенсивного використання даних
2. Небагато користувачів
3. Багато даних
4. Важлива продуктивність

NFR – What We Ask

1. *“Скільки одночасних користувачів очікується?”* 10
2. *“Кількість телеметричних повідомлень у секунду?”* 7,000
3. *“Середній розмір повідомлення?”* 1KB
4. *“Чи має повідомлення структуру?”* Так

NFR – What We Ask

5. *“Чи можна допустити втрату деяких
повідомлень?”*

Напевно так...

6. *“Бажаний SLA?”*

Найвищий
можливий

Обсяг даних

- 1 повідомлення = 1KB
- 7,000 повідомлень / с = 7MB / с

=> ~25GB / год.

=> ~605GB / день

=> ~221TB / рік ← Це багато!

Період зберігання

Визначає, як довго записи зберігаються в базі даних

Що з ними станеться після періоду зберігання?

- Видаляться
- Перемістяться у архівне сховище даних

Smartcar.auto

Період зберігання

Призначення:

- Запобігає перевищення обсягу бази даних
- Покращує продуктивність запитів

AWS Config adds the ability to specify a data retention policy for your configuration items

Posted On: Aug 7, 2018

Smartcar потребує два типи даних:

- Оперативний, в реальному часі (місцезнаходження, швидкість, і т.д.)
- Накопичені та готові для аналізу (BI – Business Intelligence, бізнес-аналітика)

Період зберігання

Data Type	Used for...	Retention Period
оперативна	Відстежує дані з автомобілів у реальному часі. Продуктивність має вирішальне значення	1 тиждень
Накопичена	Звіти, BI. Гне у реальному часі, може бути повільніше.	Весь час

Обсяг даних

- 1 повідомлення = 1KB
- 7,000 повідомлень / с = 7MB / с

=> ~25GB / год.

=> ~605GB / день

=> ~221TB / рік

Обсяг даних

- 1 повідомлення = 1KB
- 7,000 повідомлень / с = 7MB / с

=> ~25GB / год.

=> ~605GB / день

=> ~4TB / тиждень

Вимоги

Функціональні

Що повинна робити

система

1. Веб-інтерфейс
2. Отримувати телеметрію від автомобілів (місцезнаходження, швидкість, поломки, тощо)
3. Зберігати телеметрію в постійному сховищі
4. Відображати інформаційні панелі із узагальненням даних
5. Виконувати аналіз даних

Нефункціональні

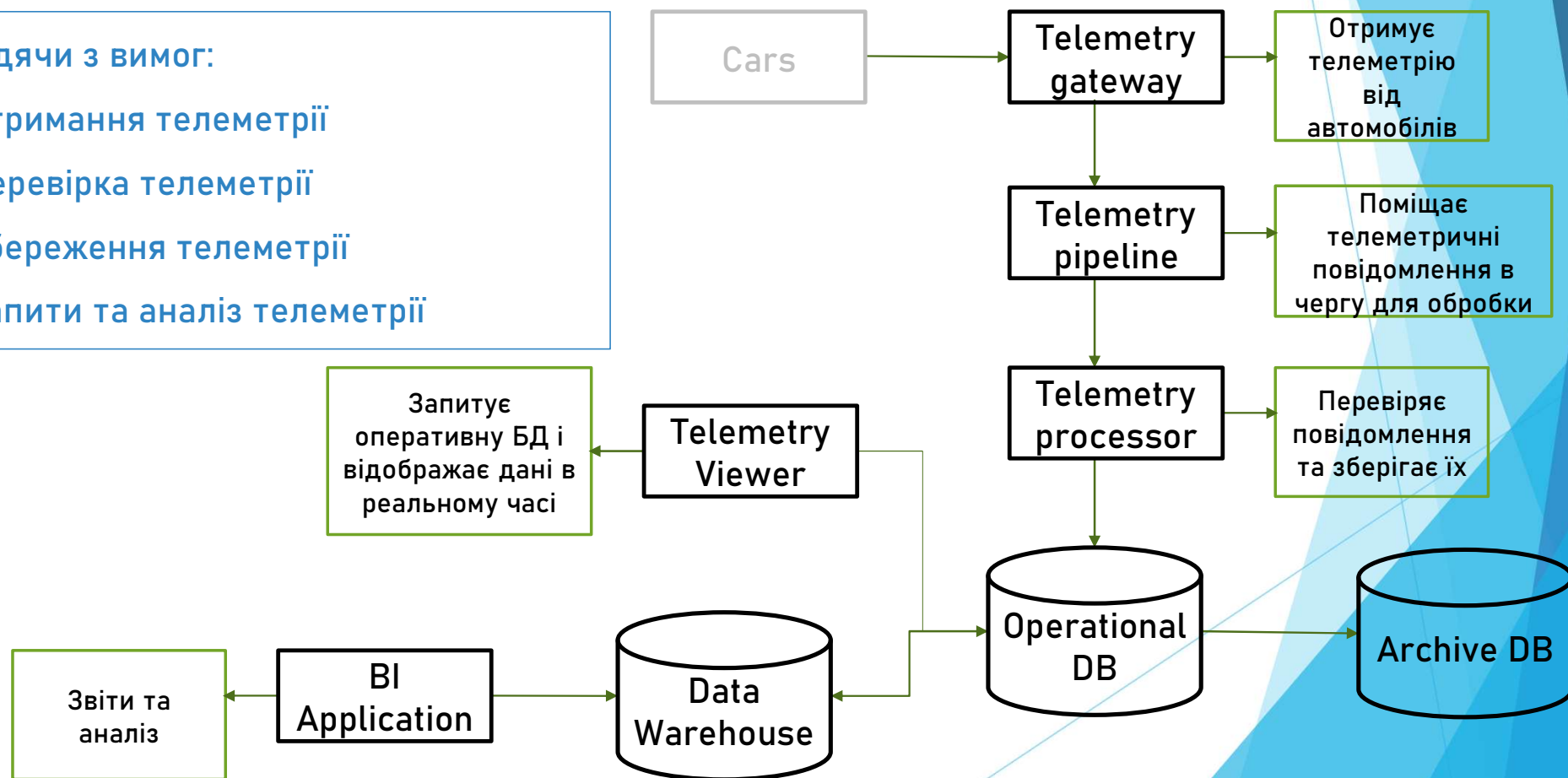
як гарно вона повинна це
робити

1. 10 одночасних користувачів
2. 7000 повідомлень/сек
3. Максимальний обсяг даних в оперативній БД: 4 ТБ
4. Критична серйозність
5. Продуктивність має вирішальне значення

Компоненти

Виходячи з вимог:

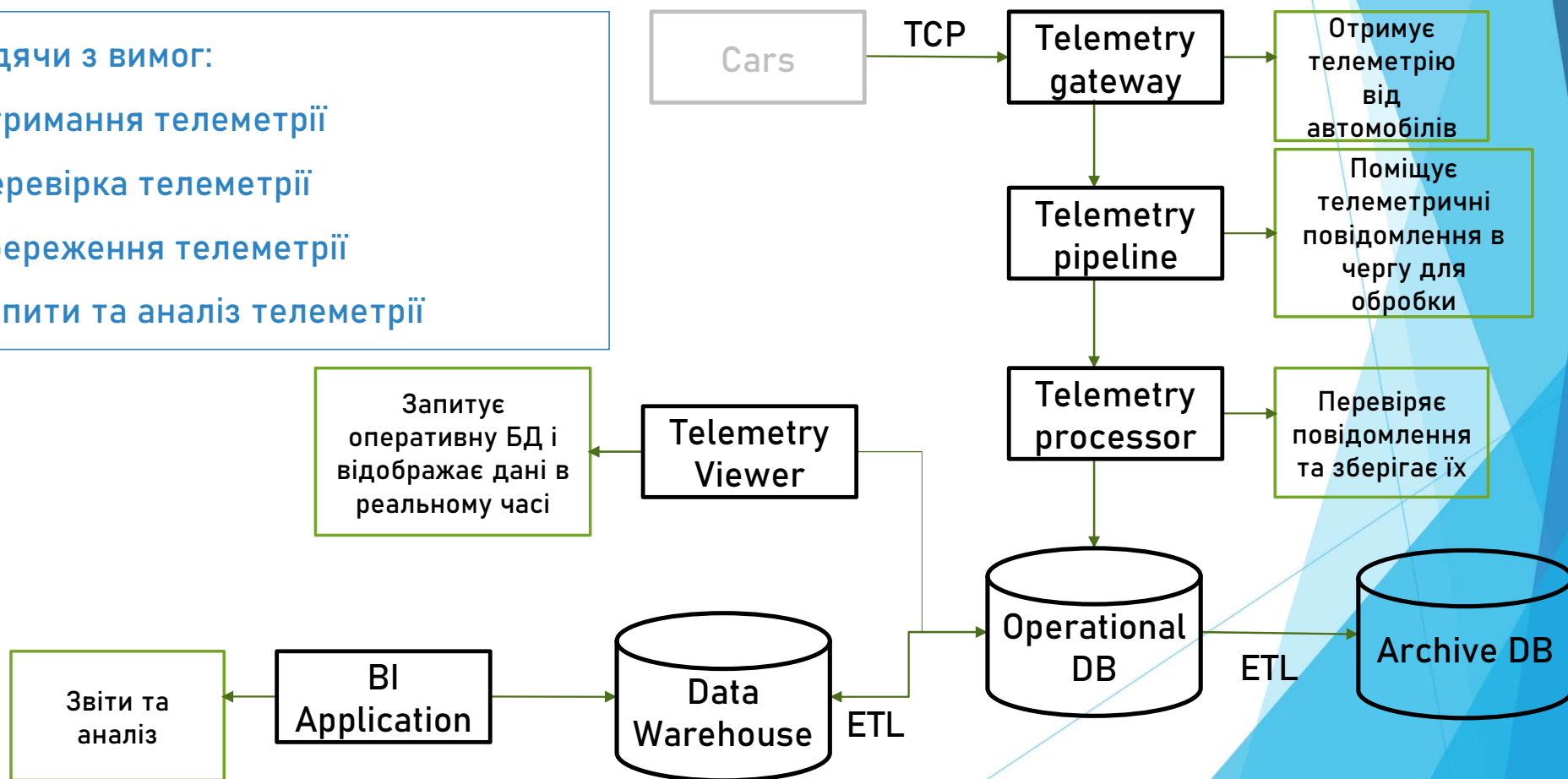
1. Отримання телеметрії
2. Перевірка телеметрії
3. Збереження телеметрії
4. Запити та аналіз телеметрії



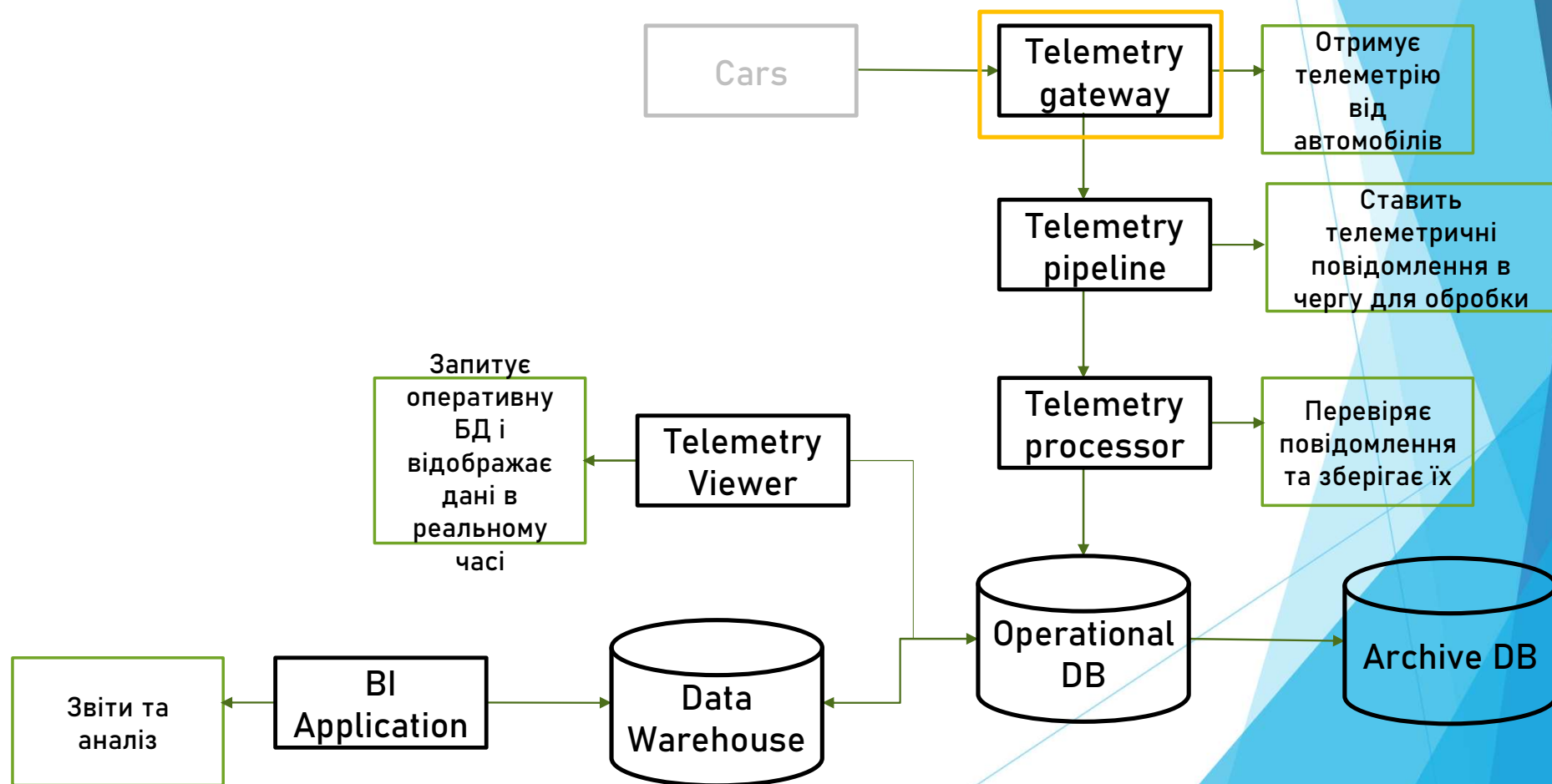
Обмін повідомленнями

Виходячи з вимог:

1. Отримання телеметрії
2. Перевірка телеметрії
3. Збереження телеметрії
4. Запити та аналіз телеметрії



Компоненти



Smartcar.auto

Телеметричний шлюз

Що він робить:

- Отримує телеметричні дані від автомобілів
за допомогою TSP
- Надсилає телеметричні дані в конвеєр

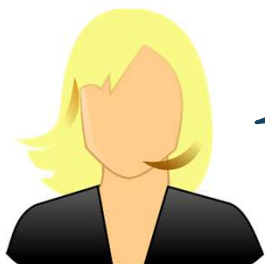
Тип програми

- Веб-додаток і веб-API ✗
- Мобільний додаток ✗
- Консоль ✓
- Сервіс ✓
- Настільний додаток ✗

Міркування:

- Навантаження (7 тис повідомлень/сек)
- Продуктивність
- Поточні знання команди
- Середовище (ОС,тощо)

Стек технологій



Наші розробники знайомі з Python і є експертами в JavaScript. Крім того, ми використовуємо лише сервери Linux.

Python не можна використовувати як шлюз

Занадто повільний

Потрібна мова з високою продуктивністю, яка працює на Linux, і використовує поточні навички (Python і JavaScript)



Smartcar.auto

Стек технологій



Чудова
продуктивність



Працює на Linux



Використовує
навички JS

Smartcar.auto

Архітектура

Традиційний:

User Interface /
Service Interface

Business Logic

Data Access

Data Store



Smartcar.auto

Архітектура

У нашому
випадку:

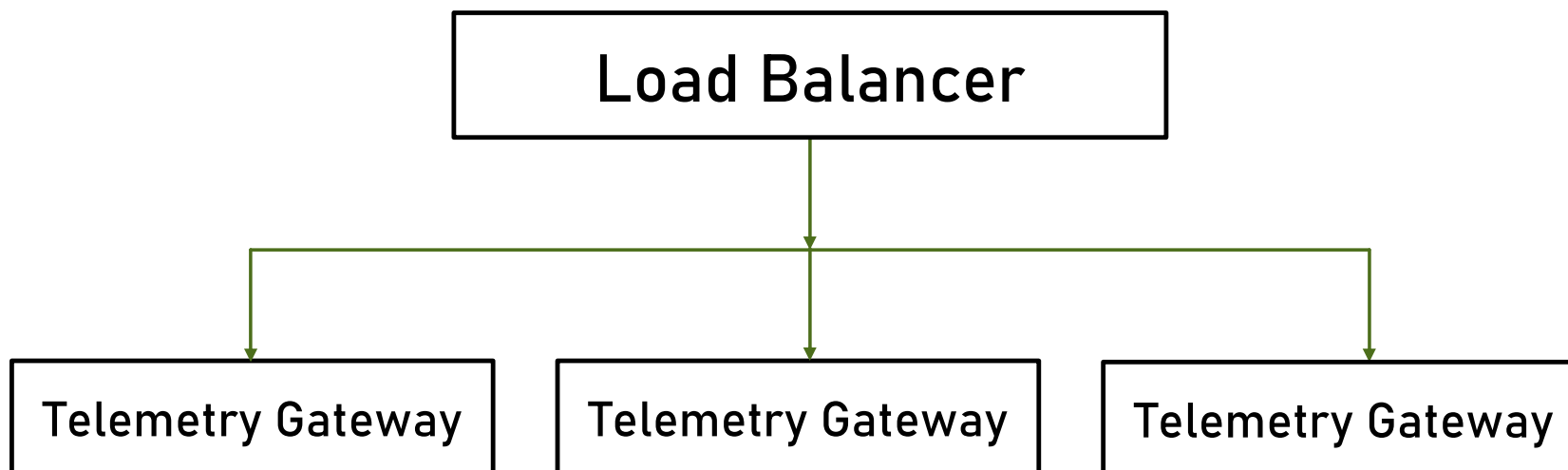
Service Interface

Pipeline

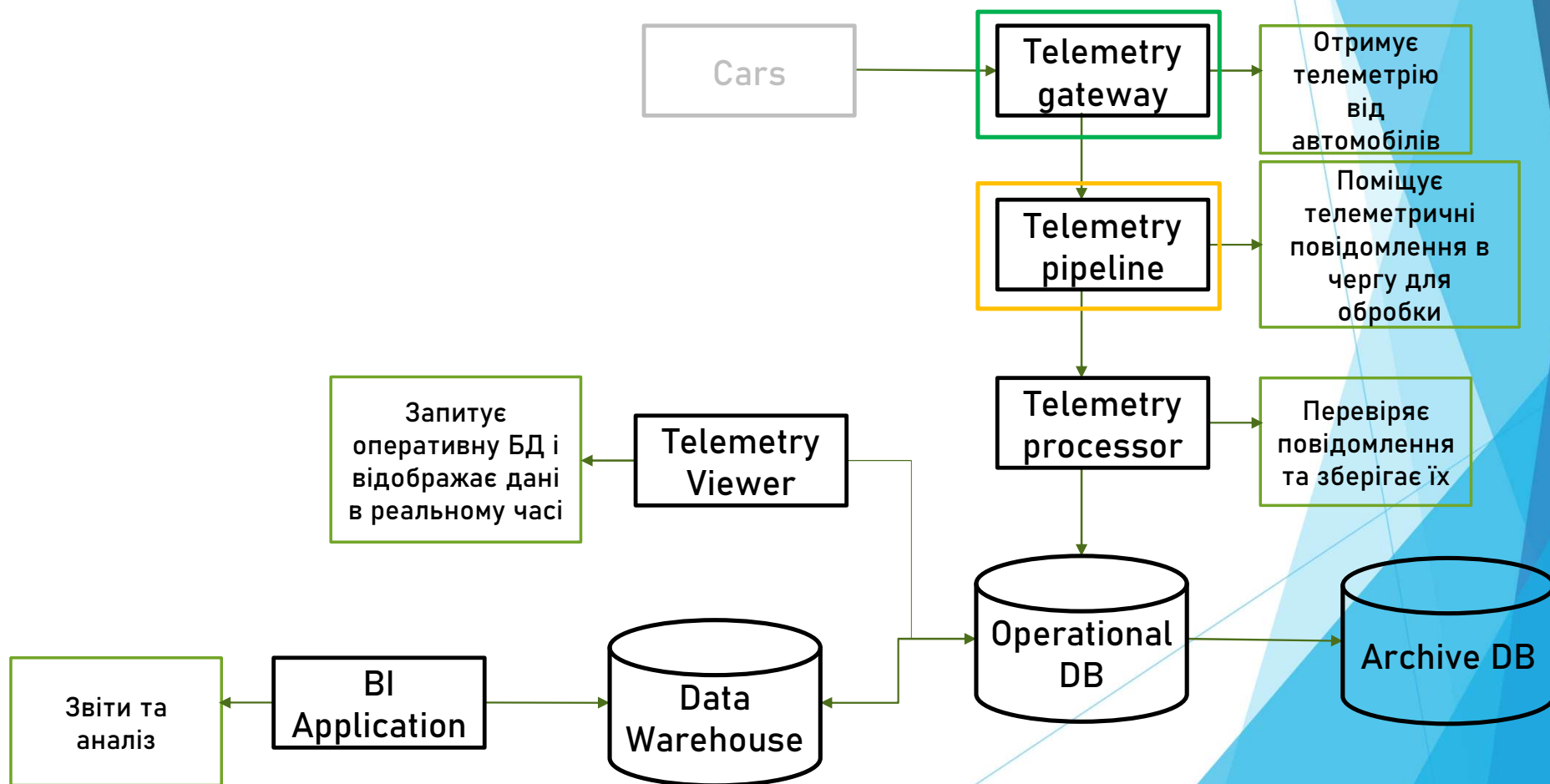


Smartcar.auto

Резервування телеметричного шлюзу



КОМПОНЕНТИ



Telemetry Pipeline

Що він робить:

- Отримує телеметричні повідомлення від шлюзу
- Поміщує телеметрію в чергу для подальшої обробки
- По суті – це черга для потокової передачі даних великого обсягу

Telemetry Pipeline
– запитання

1. Чи існує в компанії механізм черги?
2. Розробити власний або
використовувати зовнішній?

Немає

Smartcar.auto

Telemetry Pipeline
– запитання

Озирнімося навколо...



Telemetry Pipeline – Кафка



Плюси:

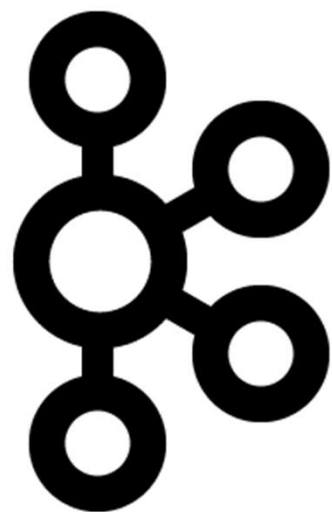
- Дуже популярний
- Може обробляти величезну кількість даних
- Підтримка високої доступності

Мінуси:

- Складне налаштування
- Складна конфігурація

Smartcar.auto

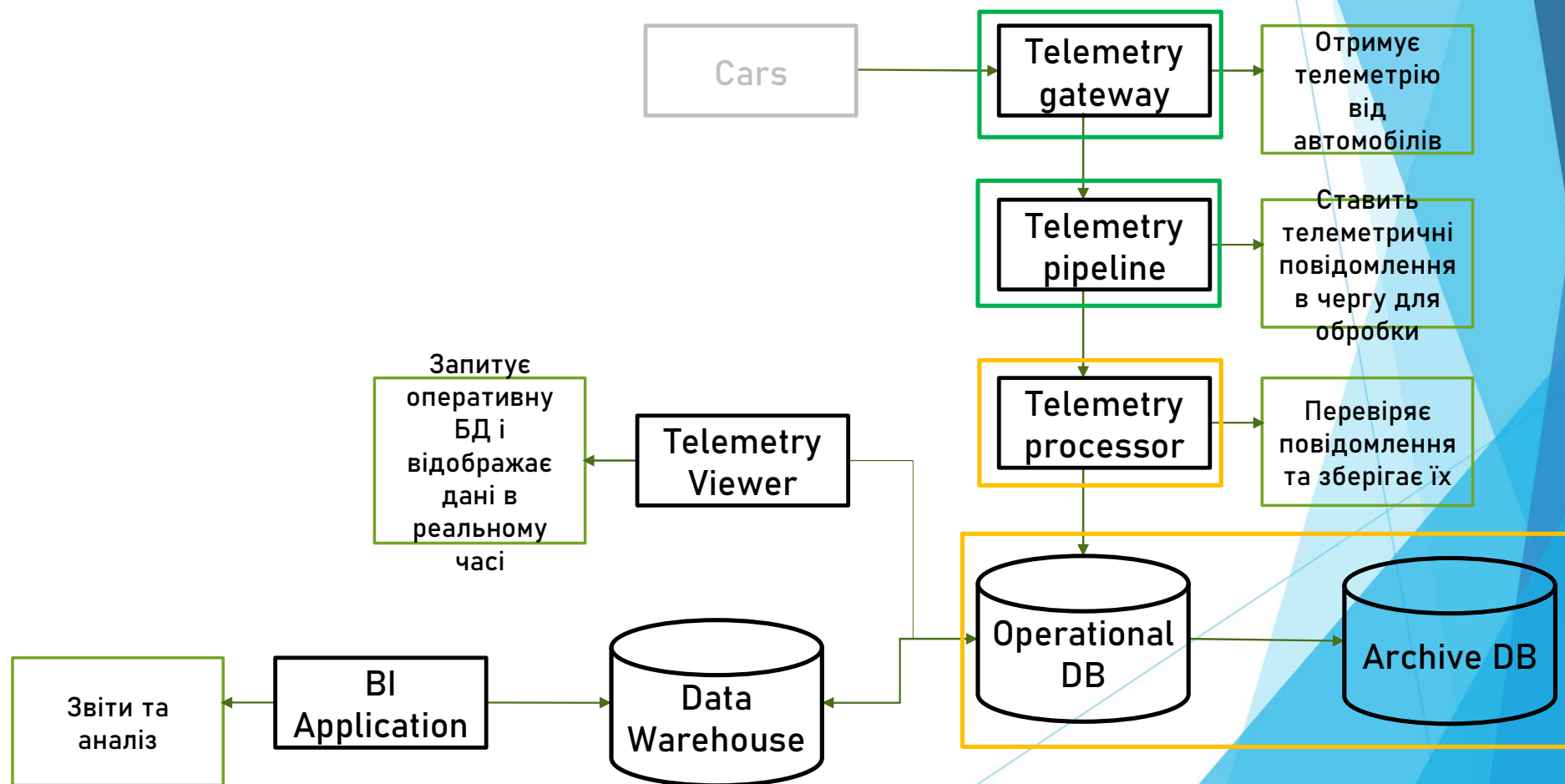
Telemetry Pipeline
– рішення



APACHE
kafka®

A distributed streaming platform

Компоненти



Telemetry Processor

Що він робить:

- Отримує повідомлення з конвеєра
- Обробляє повідомлення (переважно перевіряє)
- Зберігає повідомлення в сховищі даних

Тип програми

- Веб-додаток і веб-API ✗
- Мобільний додаток ✗
- Консоль ✓
- Сервіс ✓
- Desktop додаток ✗

Smartcar.auto

Стек технологій

для:

- Процесор
- Сховище даних

Smartcar.auto

Стек технологій

Процесор:



- Вже використовується в системі
- швидкий
- Чудова підтримка Kafka

Smartcar.auto

Стек технологій

Сховище даних – те, що ми шукаємо:

- Підтримка повідомлень без структури
- Швидкий пошук
- Жодних складних запитів



mongoDB®

Smartcar.auto

Стек технологій

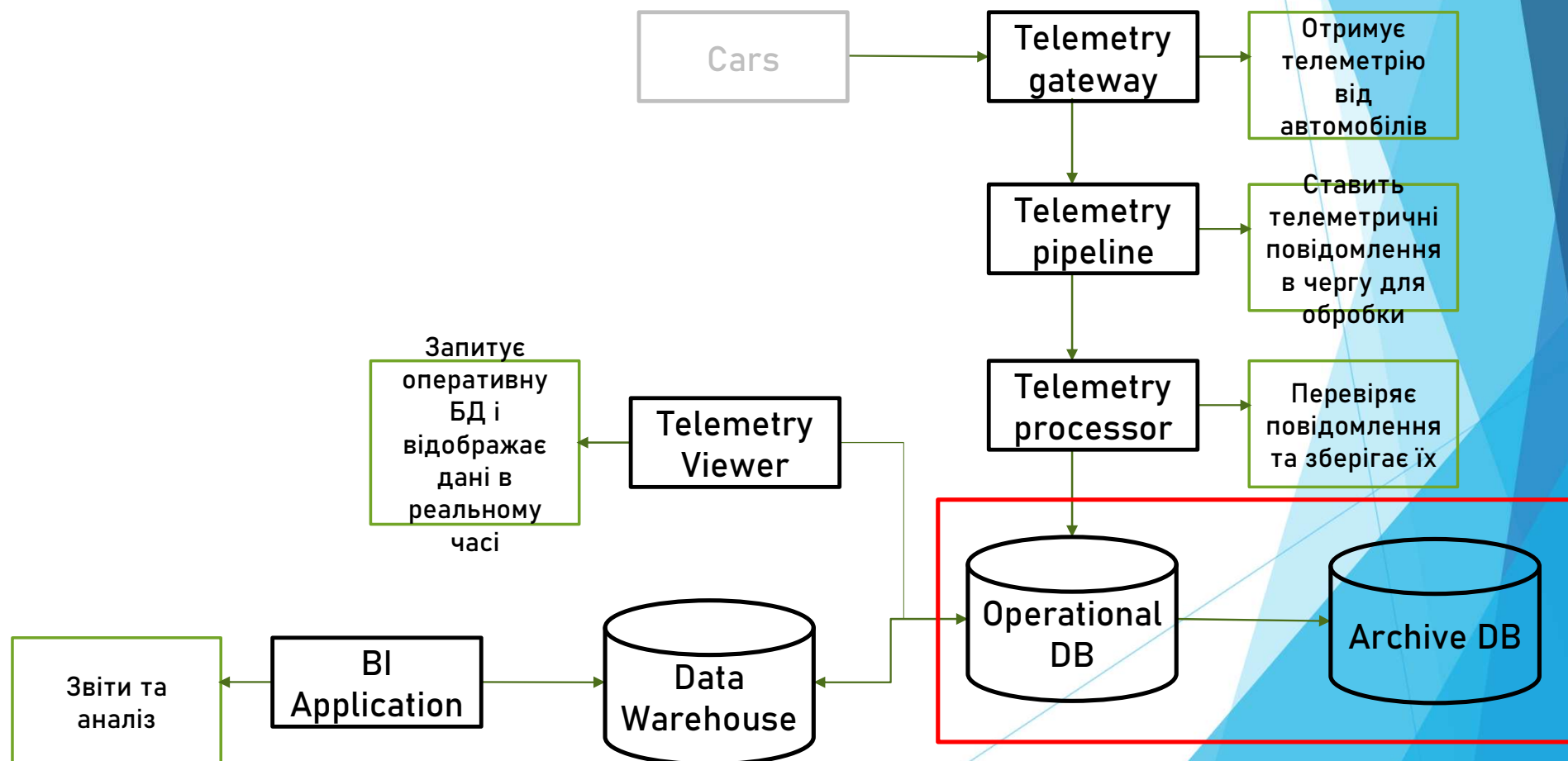


mongoDB®

- Підтримка повідомлень без структури
- Швидкий пошук
- Жодних складних запитів



Компоненти



Архів – що ми шукаємо:

- Підтримка величезної кількості даних (221 ТБ / рік)
- Доступ нечастий
- Немає необхідності швидкого пошуку
- Економія витрат



Перехід до хмари

Хмарне сховище:

- Величезні обсяги даних (221 ТБ / рік)
- Доступ нечастий
- Немає необхідності швидкого пошуку
- Економія витрат



Хмарне сховище



Microsoft Azure

	PREMIUM	HOT	COOL	ARCHIVE
First 50 terabyte (TB) / month	\$0.195 per GB	\$0.0196 per GB	\$0.01 per GB	\$0.0018 per GB
Next 450 TB / Month	\$0.195 per GB	\$0.0189 per GB	\$0.01 per GB	\$0.0018 per GB
Over 500 TB / Month	\$0.195 per GB	\$0.0181 per GB	\$0.01 per GB	\$0.0018 per GB

За 1 рік: 221 ТБ => 398
доларів

Smartcar.auto

Архітектура

User Interface /
Service Interface

Business Logic

Data Access

Data Store

```
graph TD; A[User Interface / Service Interface] --- B[Business Logic]; B --- C[Data Access]; C --- D[(Data Store)];
```

Smartcar.auto

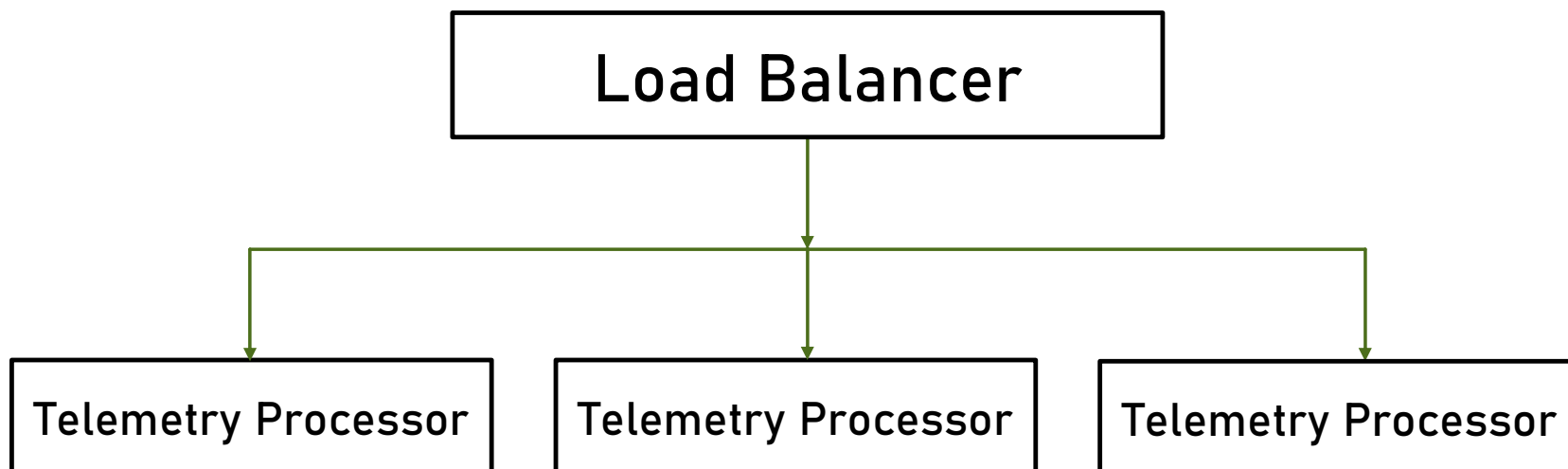
Резервування Telemetry
Processor

Load Balancer

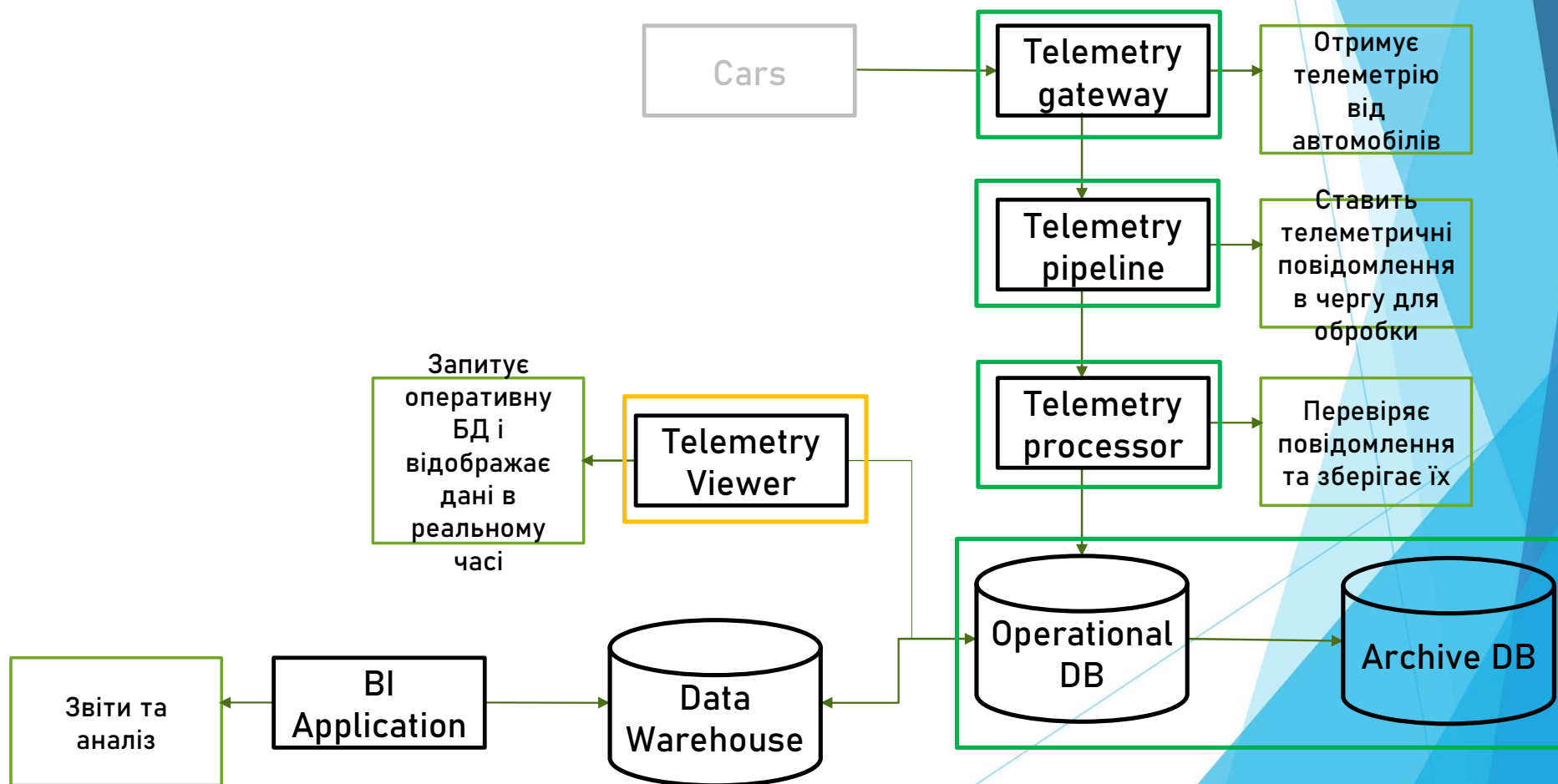
Telemetry Processor

Telemetry Processor

Telemetry Processor



Компоненти



Smartcar.auto

Telemetry Viewer

Що він робить:

- Дозволяє кінцевим користувачам запитувати дані телеметрії
- Відображає дані в реальному часі

Що він не робить:

- Не аналізує дані

Тип програми

- Веб-додаток і веб-API ✓
- Мобільний додаток ✗
- Консоль ✗
- Сервіс ✗
- Desktop додаток ✗

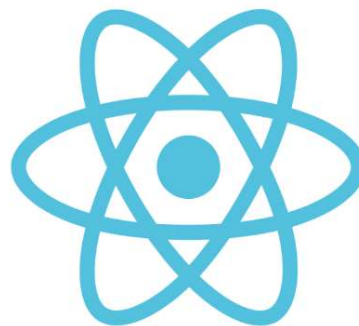
Smartcar.auto

Стек технологій

Back End



Front End



React JS

Smartcar.auto

Архітектура

Service Interface

Business Logic

Data Access

Data Store

```
graph TD; SI[Service Interface] --- BL[Business Logic]; BL --- DA[Data Access]; DA --- DS[(Data Store)];
```

The diagram illustrates the architecture of Smartcar.auto. It consists of four main components arranged vertically. At the top is a light blue rectangular box labeled 'Архітектура'. Below it are three stacked rectangular boxes: 'Service Interface', 'Business Logic', and 'Data Access'. At the bottom is a cylindrical shape labeled 'Data Store'. The entire diagram is set against a background with abstract blue geometric shapes on the right side.

Smartcar.auto

API

- Отримати останні помилки для всіх автомобілів
- Отримати останню телеметрію для конкретного автомобіля
- Отримати останні помилки для конкретного автомобіля

API

Functionality	Path	Return Codes
Отримати останні помилки для всіх автомобілів	GET /api/v1/telemetry/errors	200 OK
Отримати останню телеметрію для конкретного автомобіля	GET /api/v1/telemetry/{carId}	200 OK 404 Not Found
Отримати останні помилки для конкретного автомобіля	GET /api/v1/telemetry/errors/{carId}	200 Ok 404 Not Found

Smartcar.auto

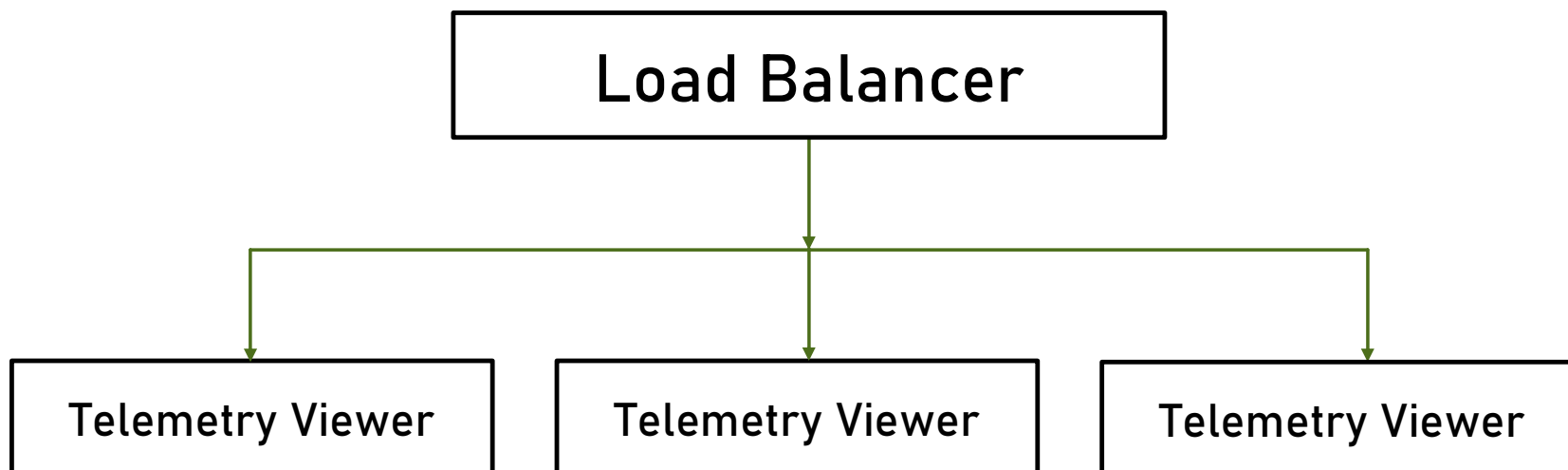
Резервування Telemetry
Viewer

Load Balancer

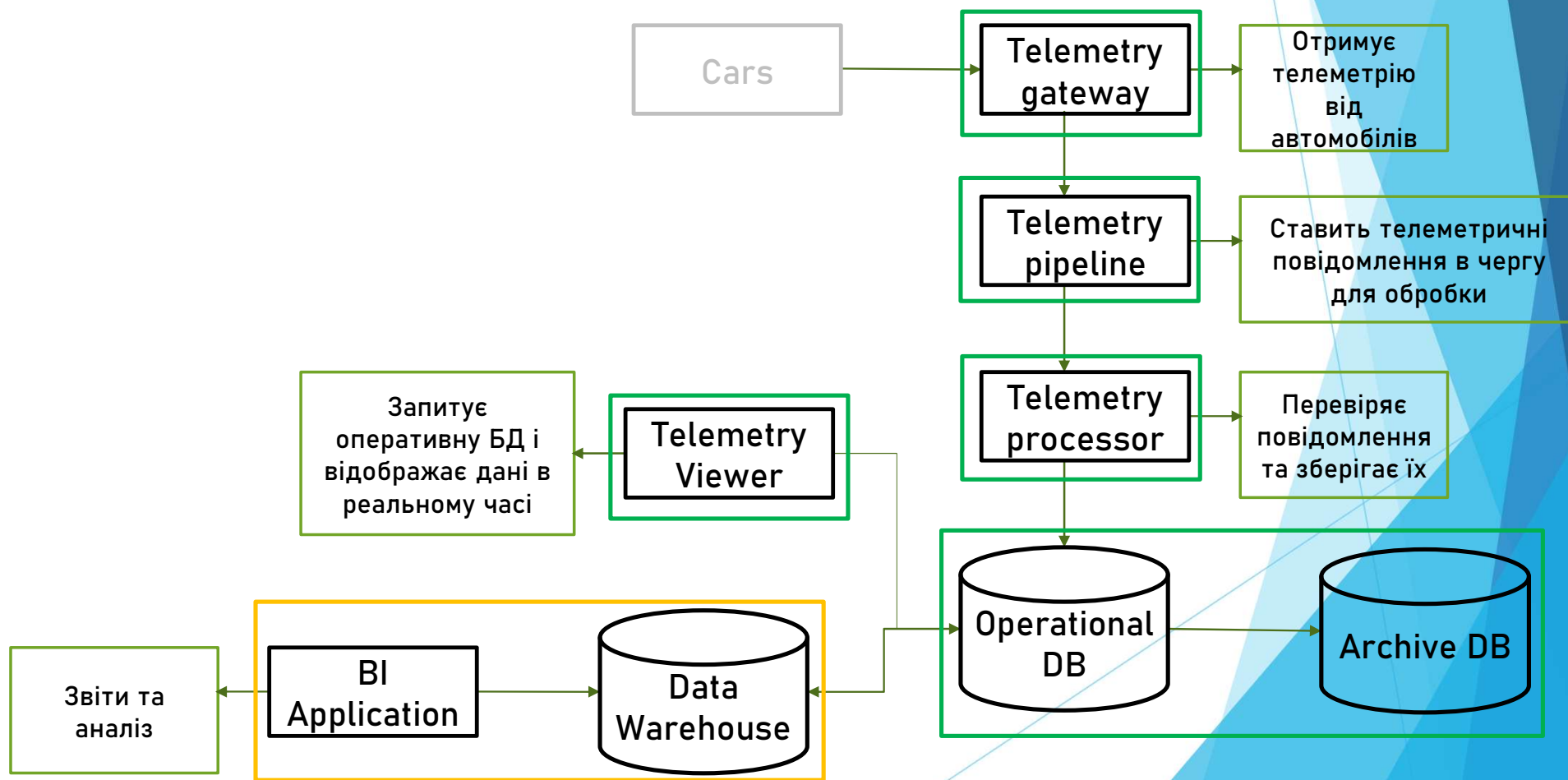
Telemetry Viewer

Telemetry Viewer

Telemetry Viewer



Компоненти



Smartcar.auto

Бізнес-
аналітика

BI Application

Що він робить:

- Аналізує дані телеметрії
- Відображає спеціальні звіти про дані, тенденції, прогнози тощо.
- Скільки машин зламалося за останній місяць?
- Яку загальну відстань проїхали автомобілі?

Smartcar.auto

Тип програми

- Не має значення
- BI Application ЗАВЖДИ базується на існуючому інструменті

Smartcar.auto

Инструменти BI



Інструменти BI

Figure 1. Magic Quadrant for Analytics and Business Intelligence Platforms



Source: Gartner (February 2019)

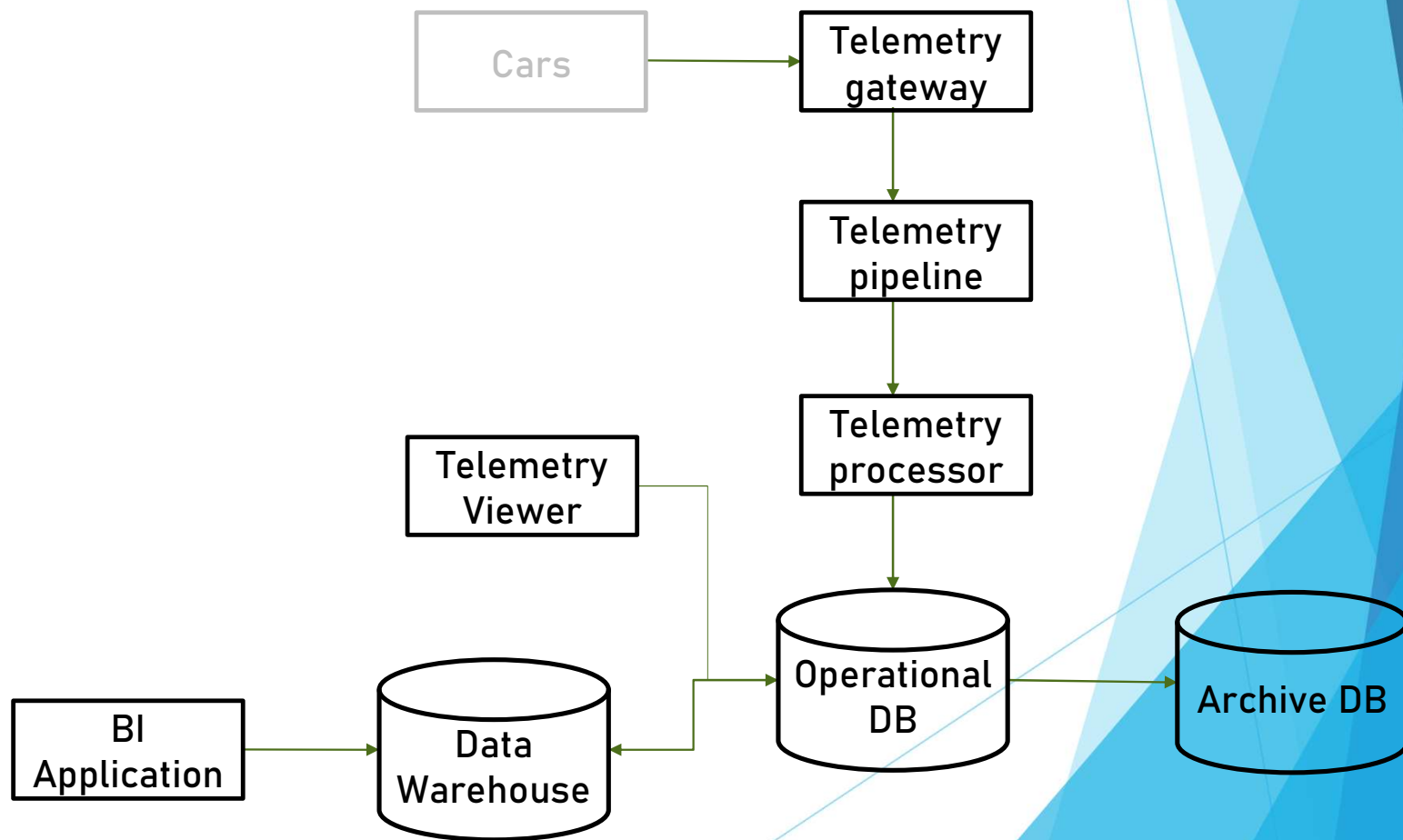
As of January 2019

© Gartner, Inc

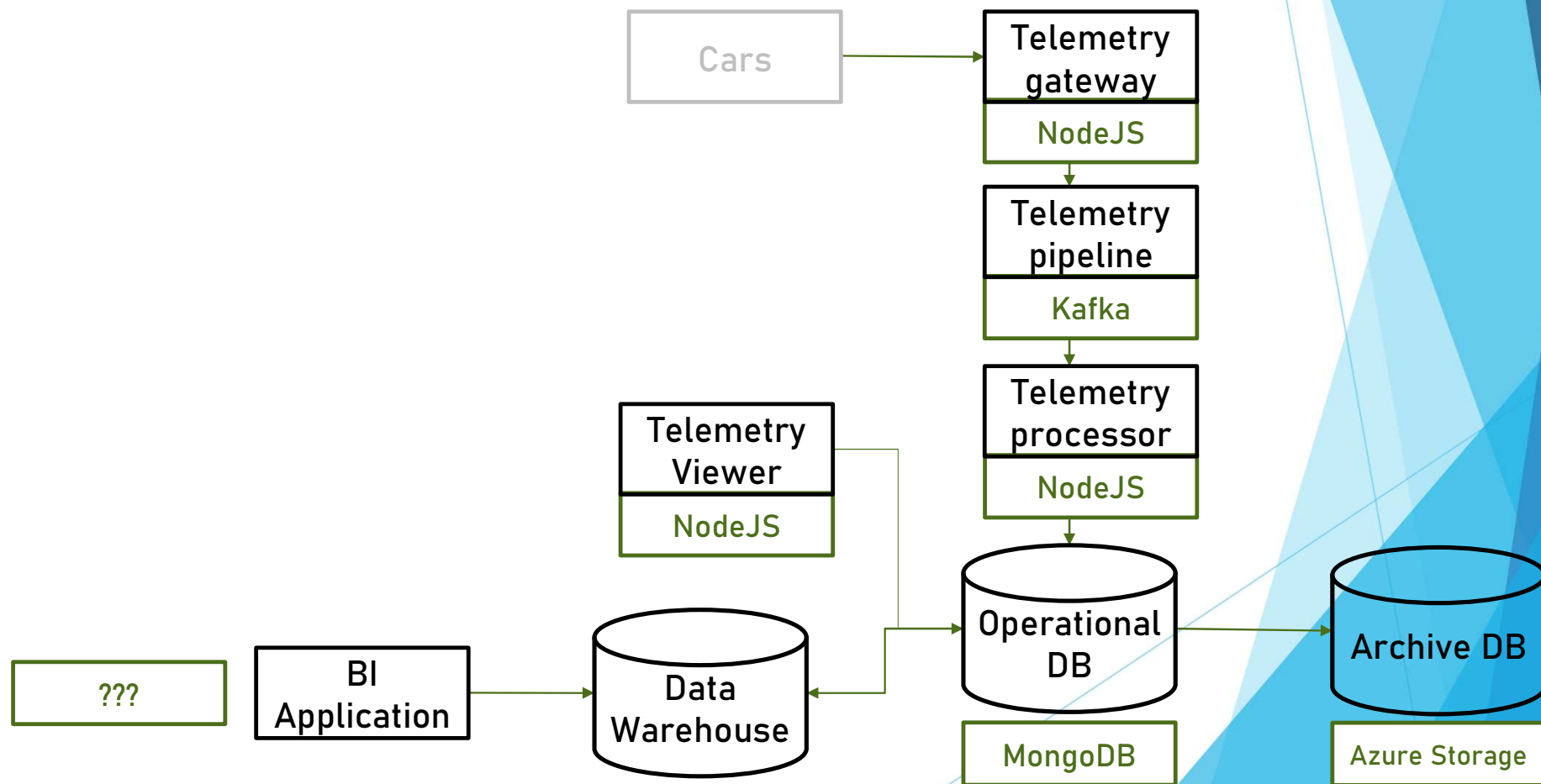
Інструменти ВІ

- Важливий урок:
 - Розробка рішення ВІ НЕ є частиною роботи архітектора
 - ЗАВЖДИ використовуйте експерта ВІ для цього завдання

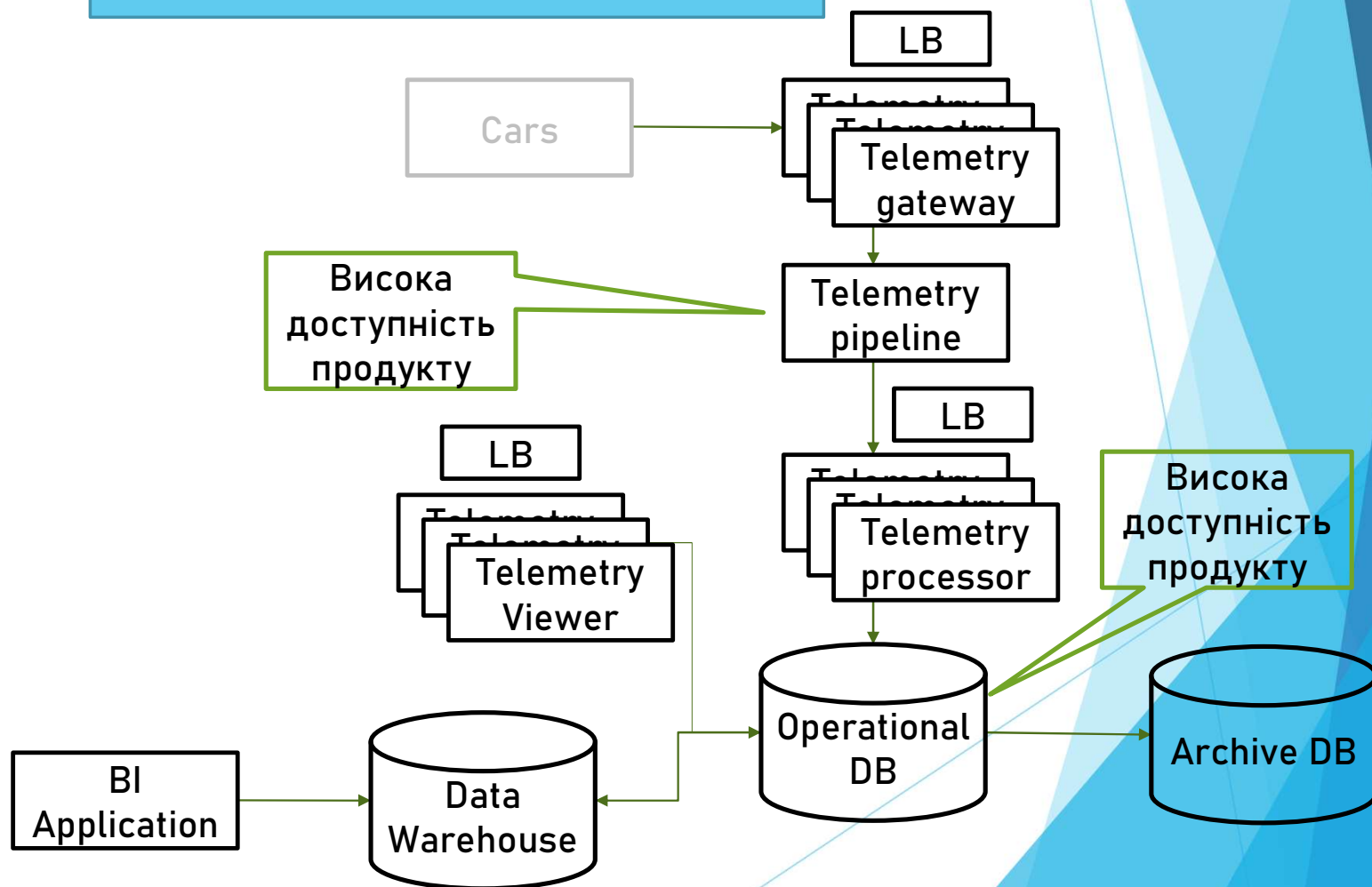
Логічна схема



Технічна схема



Фізична схема



Case #3





GROCEMALL

GROCEMALL

- Служба збору продуктів
- Дозволяє клієнтам створювати списки покупок, які збираються та доставляються працівниками GroceMall
- Доступний у всьому світі



GROCEMALL

- Співробітники мають спеціальні планшети, на яких відображається список
- Потрібно розробити частину системи для збірки замовлення
 - Клієнтська сторона вже розроблена





Вимоги

Функціональні

Що повинна робити

система

1. Веб-інтерфейс
2. Планшети отримують список для збору
3. Співробітники можуть позначати товари як зібрані або недоступні
4. Після завершення збору список має бути переданий до платіжної системи
5. Підтримка офлайн є обов'язковою

Нефункціональні

Як гарно вона повинна це робити



НФВ – Питання

- 1. «Скільки очікуваних одночасних користувачів?»*
- 2. «Скільки списків буде оброблено на день?»*
- 3. «Який середній розмір списку покупок?»*

200

10 000

500 КБ



NFR – Що ми просимо

4. *«Чи потрібна нам офлайн-підтримка?»*

Так!

5. *«Який бажаний SLA?»*

Найвищий
можливий

6. *«Як списки надходять до системи?»*

Черга



Обсяг даних

- 1 список = 500 Кб
 - 10 000 списків / день = 5 ГБ / день
- => ~2 ТБ / рік



Вимоги

Функціональні

Що повинна робити
система

1. Веб-інтерфейс
2. Планшети отримують список для збору
3. Співробітники можуть позначати товари як зібрані або недоступні
4. Після завершення збору список має бути переданий до платіжної системи
5. Підтримка офлайн є обов'язковою

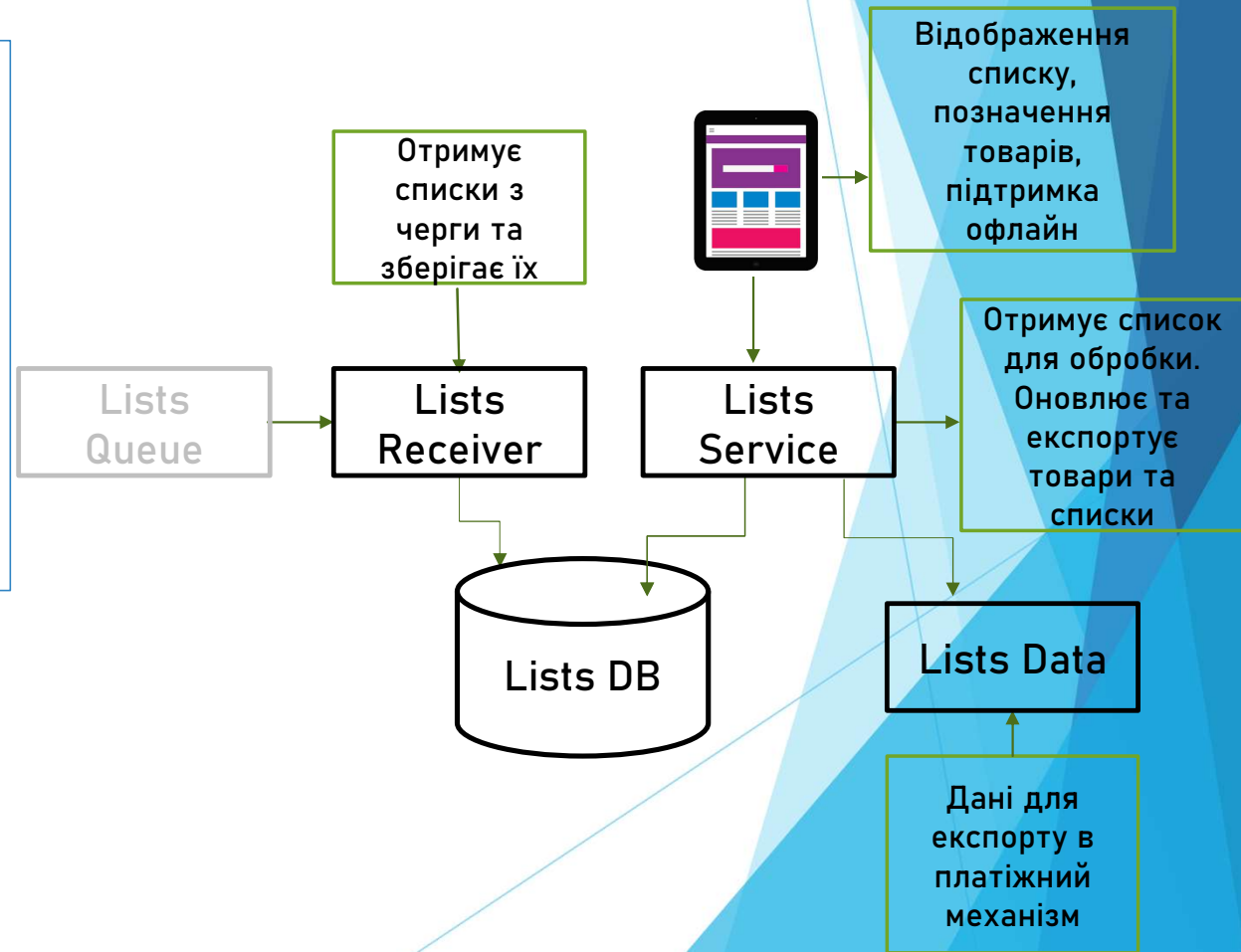
Нефункціональні

Як добре вона повинна це
робити

1. 200 одночасних користувачів
2. 10 000 списків/день
3. Річний обсяг: 2 ТБ
4. Високий SLA
5. Офлайн підтримка



1. Працівники мають планшети
2. Офлайн підтримка
3. Отримання списків
4. Позначення товарів
5. Експортування списку до платіжної системи

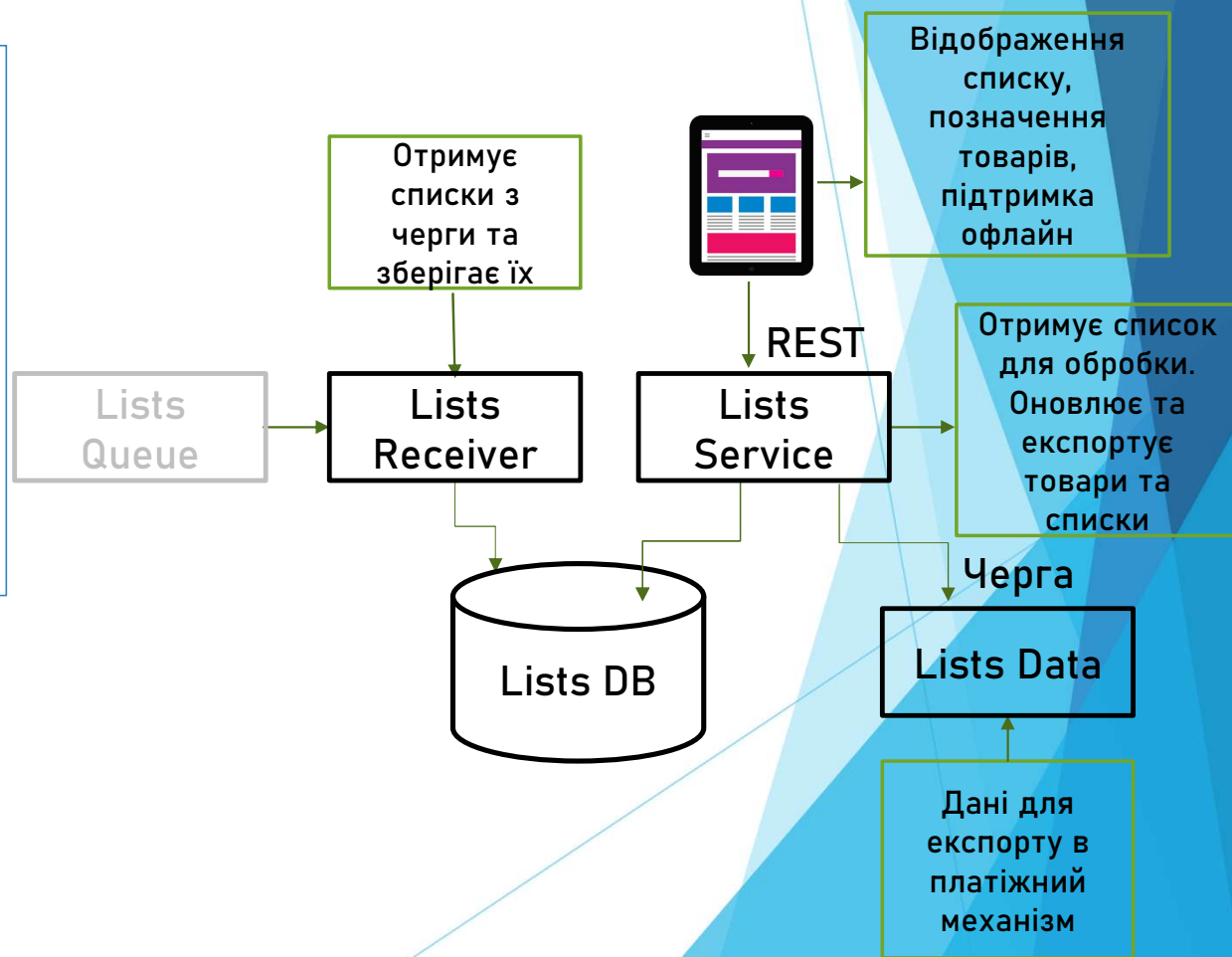


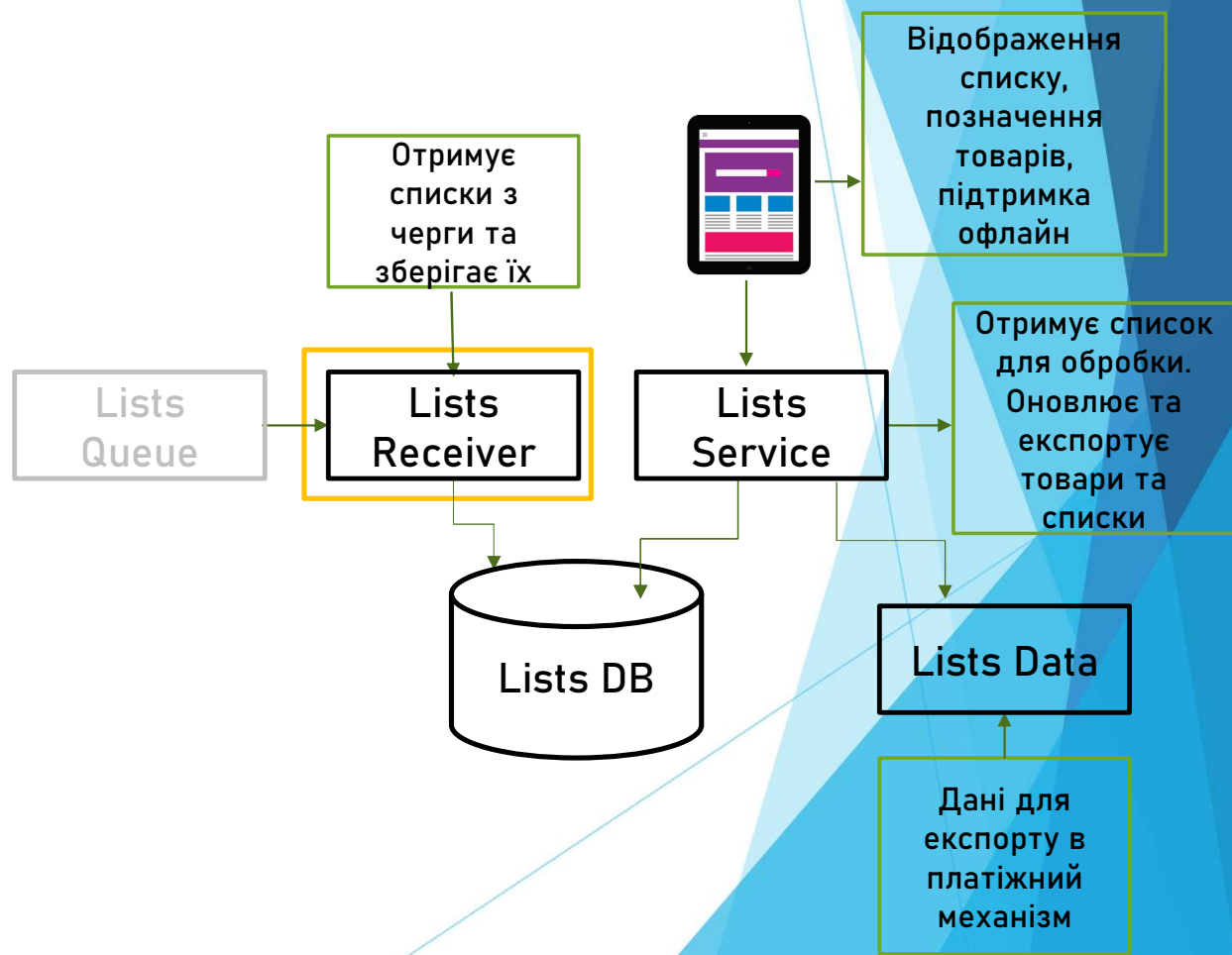


Обмін повідомленнями

Виходячи з вимог:

1. Працівники мають планшети
2. Офлайн підтримка
3. Отримання списків
4. Позначення товарів
5. Експортування списку до платіжної системи







Lists Receiver

Що він робить:

- Отримує списки покупок для обробки з черги
- Зберігає списки в сховищі даних



Тип програми

- Веб-додаток і веб-API ❌
- Мобільний додаток ❌
- Консоль ✔️
- Сервіс ✔️
- Настільний додаток ❌



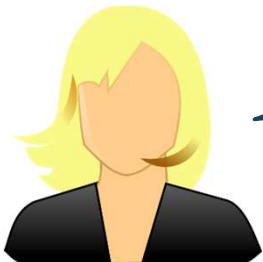
Стек технологій

Міркування:

- Має бути можливість підключитися до черги



Стек технологій



У нас магазин на основі Java, і
наша база даних MySQL.

Java ідеально підходить для цього
завдання, тому ми оберемо її.



Стек технологій



У нас магазин на основі Java, і наша база даних MySQL.

А як щодо бази даних?

- Наші дані є реляційними, а MySQL є реляційною БД
- Очікуваний обсяг становить 2 ТБ/рік, що дуже багато
 - Але можна використовувати сегментування
 - Так...



Стек технологій





Архітектура

Queue Receiver

Business Logic

Data Access

Data Store





Резервування Lists Receiver

Consumer Group

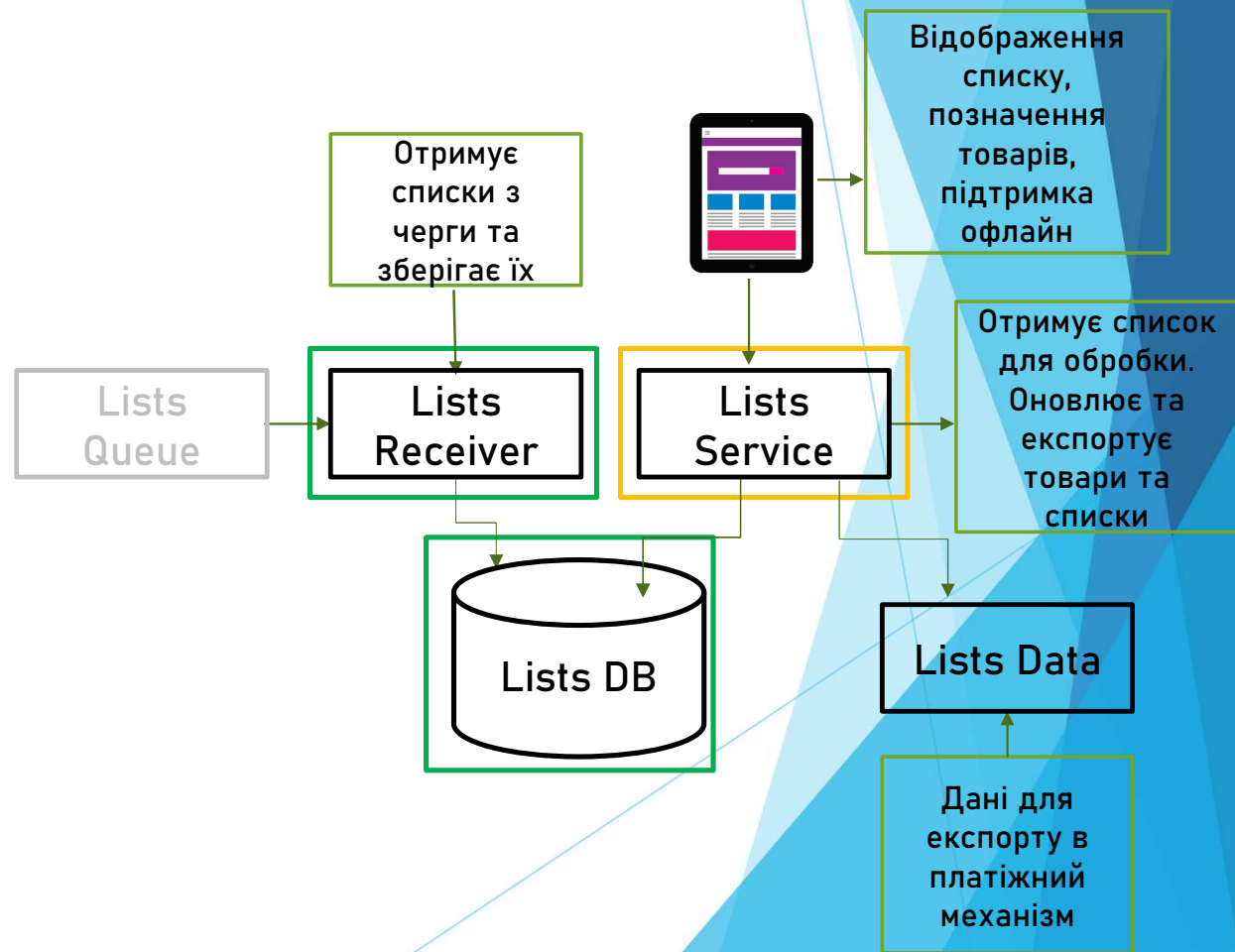
Lists Receiver

Lists Receiver

Lists Receiver



КОМПОНЕНТИ





Lists Service

Що він робить:

- Дозволяє співробітникам запитувати списки
- Позначає елементи в списку
- Експортує платіжні дані



Тип програми

- Веб-додаток і веб-API ✓
- Мобільний додаток ✗
- Консоль ✗
- Сервіс ✗
- Desktop додаток ✗



Стек технологій





Архітектура

Service Interface

Business Logic

Data Access

Data Store





API

- Отримати наступний список для обробки (за місцезнаходженням)
- Позначити товар як зібраний/недоступний
- Експорт платіжних даних зі списку

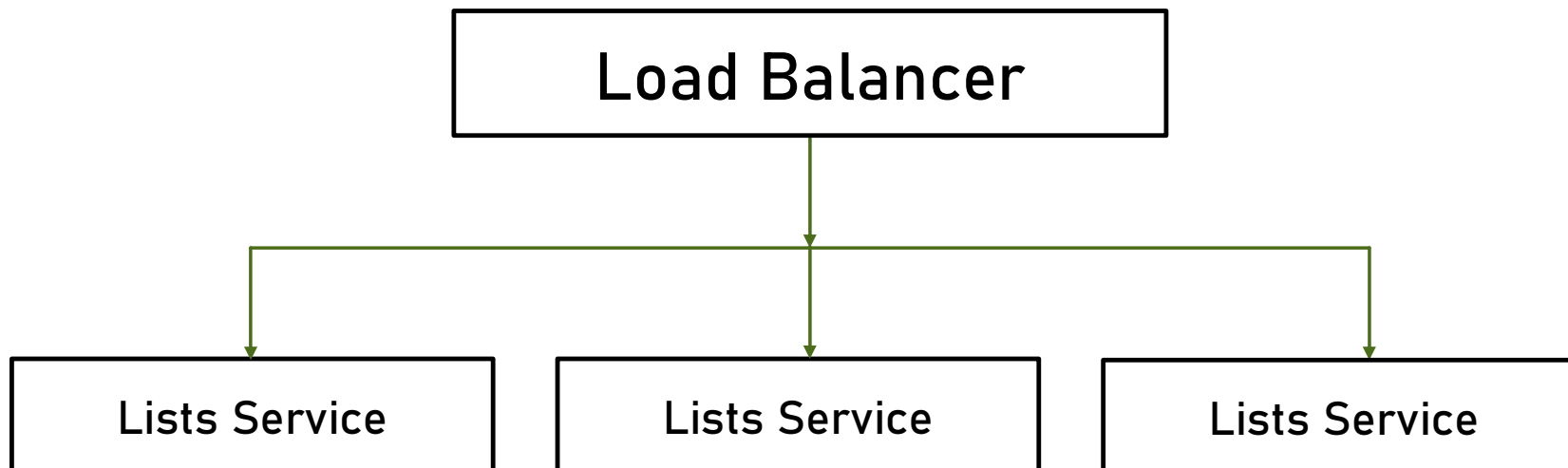


API

Functionality	Path	Return Codes
Отримати наступний список для обробки	GET /api/v1/lists/next?location=...	200 OK 400 Bad Request
Позначити товар як зібраний/недоступний	PUT /api/v1/list/{listId}/item/{itemId}	200 OK 404 Not Found
Експорт платіжних даних зі списку	POST /api/v1/list/{listId}/export	200 Ok 404 Not Found

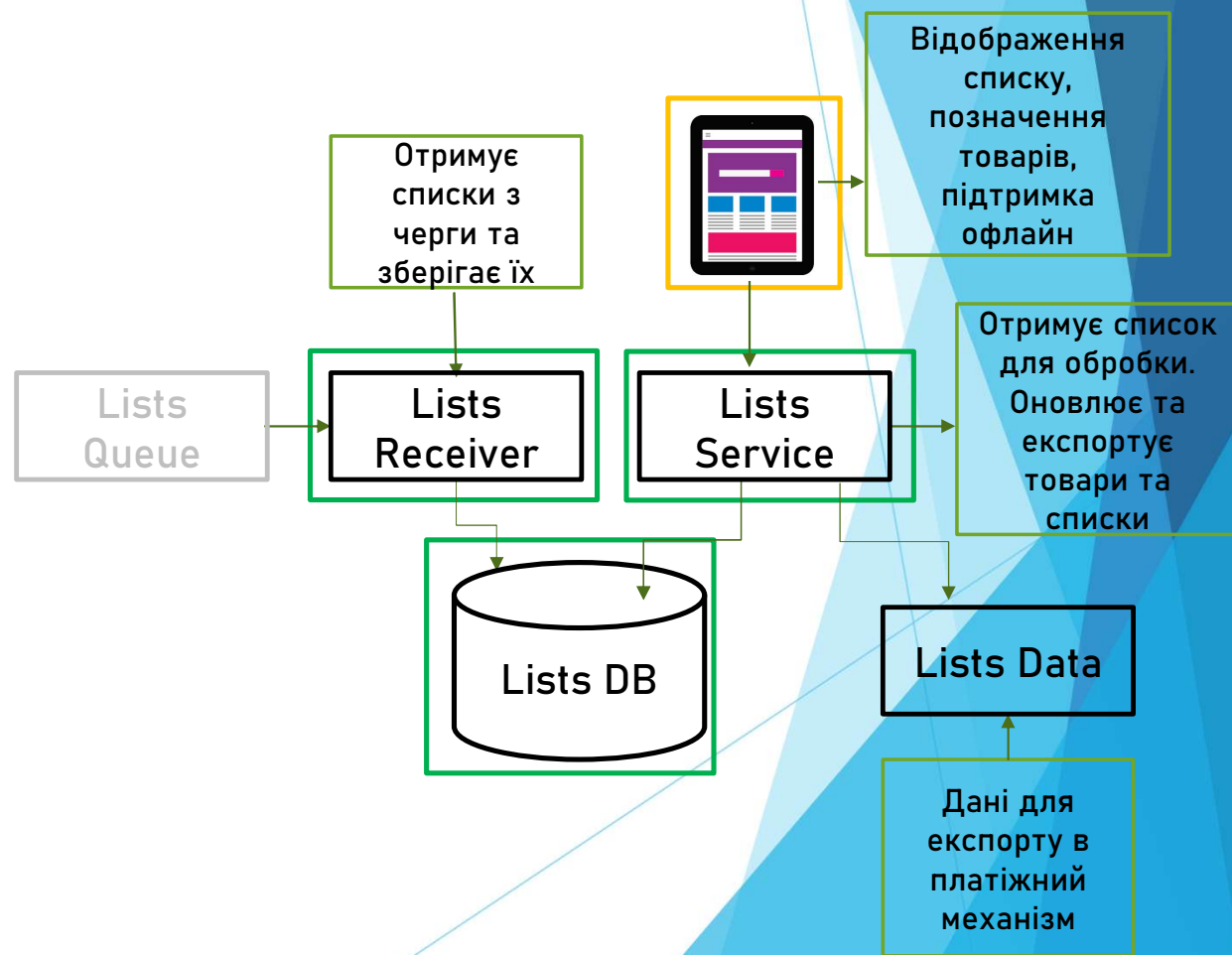


Резервування Lists Service





Компоненти





Front End

Що він робить:

- Відображає список покупок
- Позначає елементи як недоступні/зібрані
- Відправляє список до платіжної системи
- Підтримує автономний режим



Тип програми

- Веб-додаток і веб-API ❌
- Мобільний додаток ✔
- Консоль ❌
- Сервіс ❌
- Desktop додаток ✔



Стек технологій

Потрібно вибрати між:

Desktop, Windows (WPF)

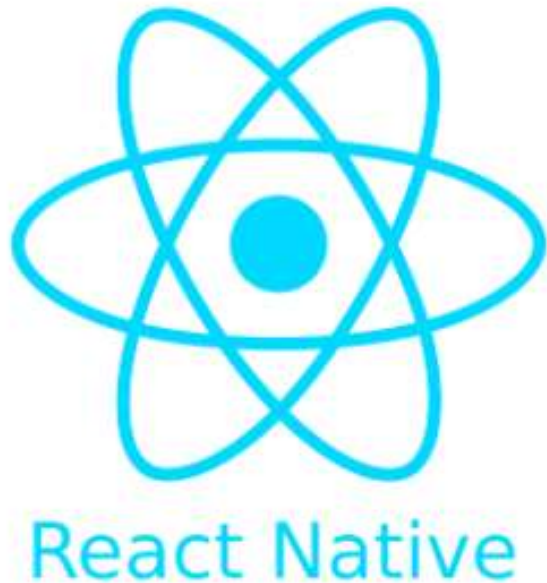
- Підтримує всі функції ОС
- Використовує інші програми на машині (наприклад, DB)
- Потрібне налаштування, Windows

Веб-інтерфейс (Electron, React Native)

- Обмежена функціональність
- Не можна використовувати інші програми
- Повністю сумісний з іншими кінцевими приладами (телефонами тощо)
- Налаштування не потрібно
- Дешевше обладнання



Стек технологій



Веб-інтерфейс (Electron, React Native)

- Обмежена функціональність
- Не можна використовувати інші програми
- Повністю сумісний з іншими кінцевими приладами (телефони тощо)
- Налаштування не потрібно
- Дешевше обладнання

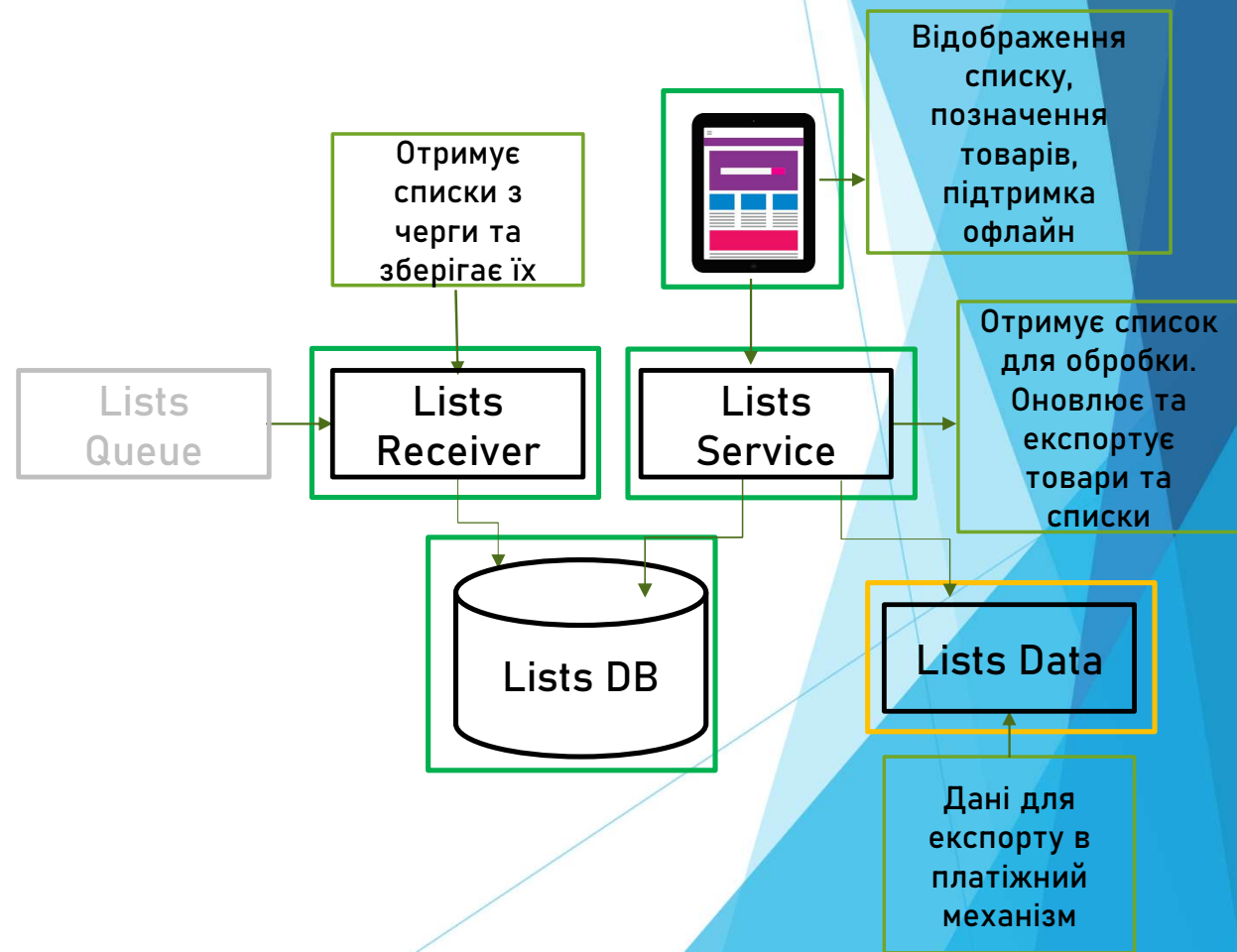


Резервування Front End

Не
актуально...



Компоненти





Lists Data

Що він робить:

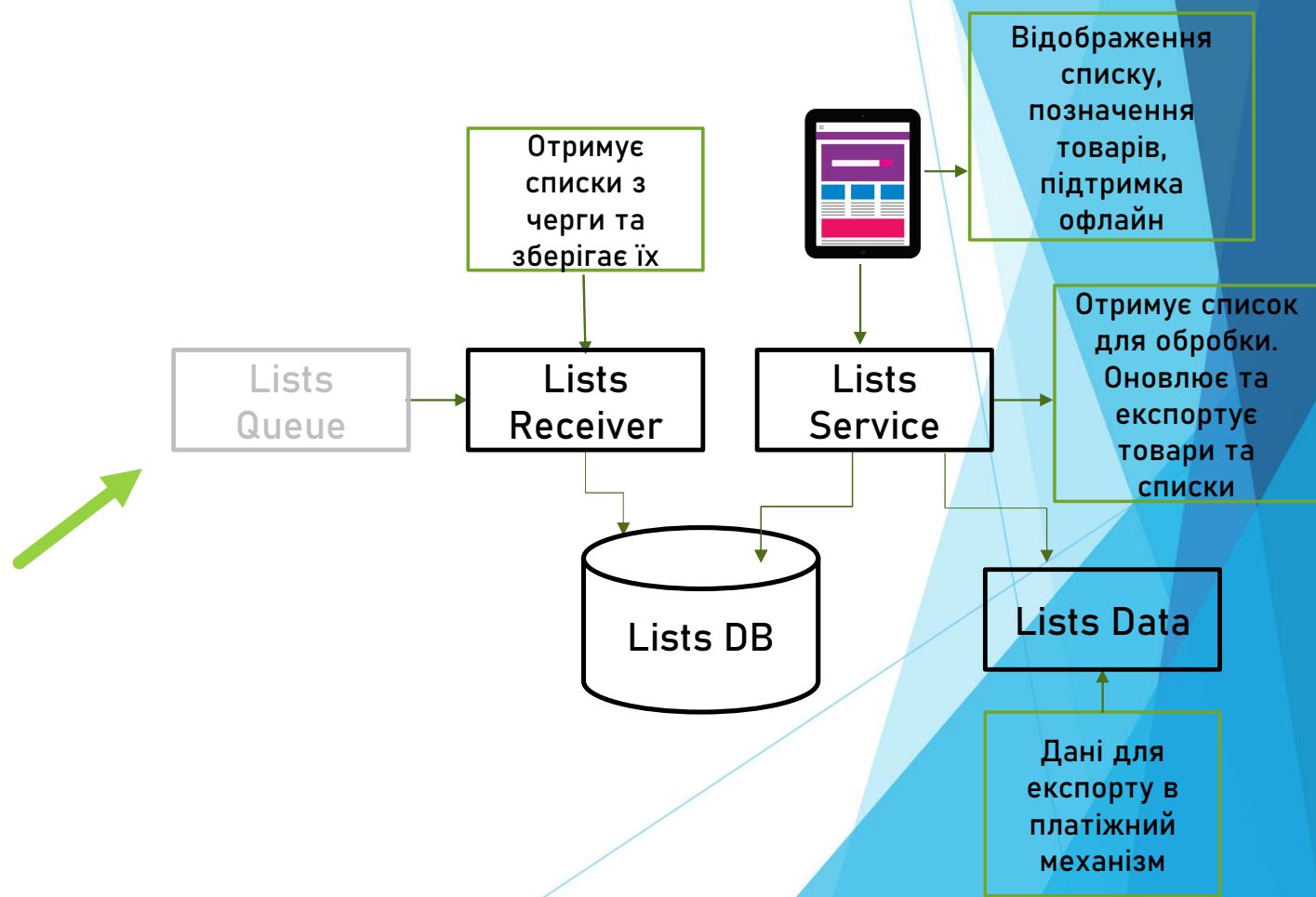
- Використовується для надсилання даних списків покупок до платіжної системи
- В основі – черга

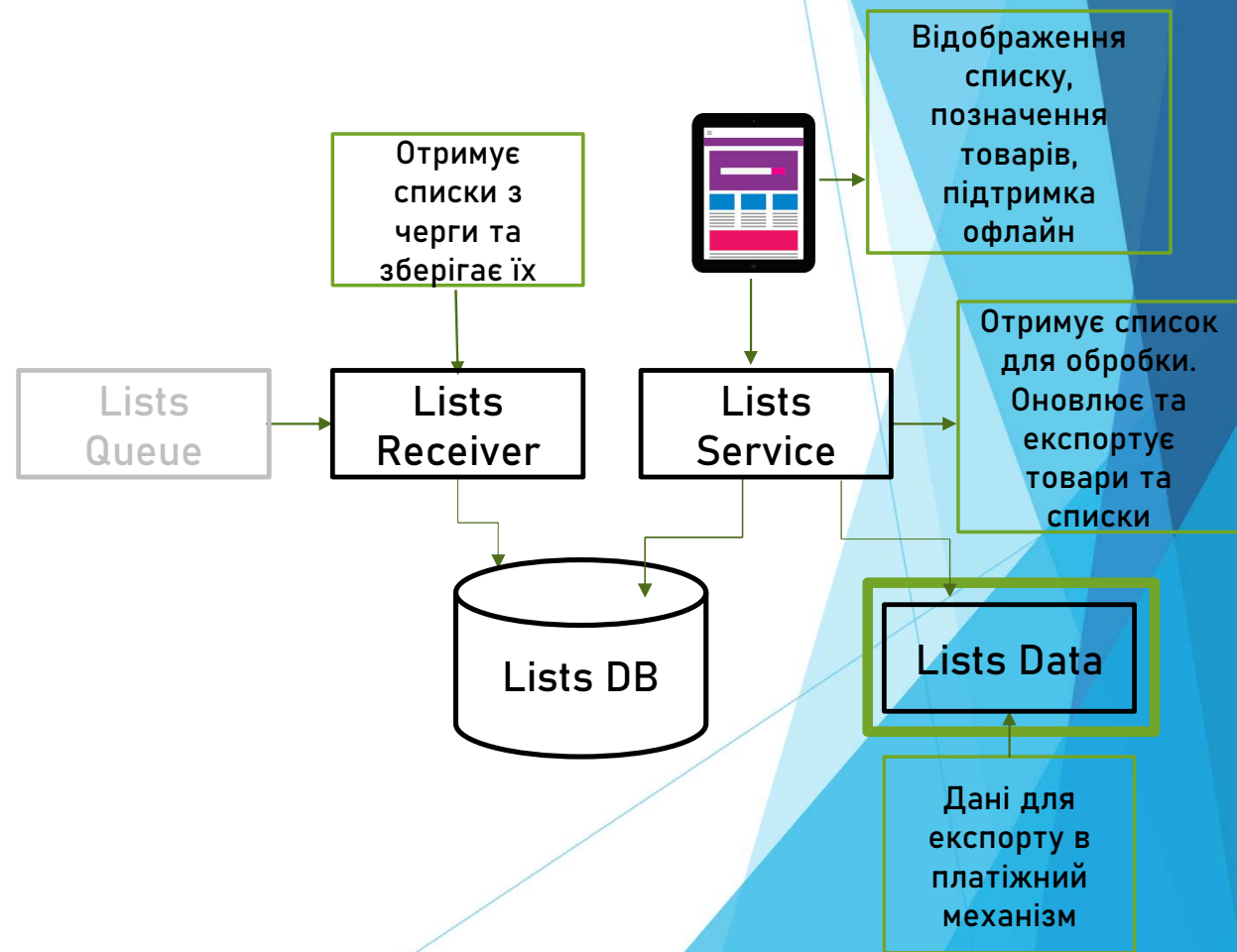


Lists Data – Питання

1. Чи існує в компанії механізм черги?
2. Розробити власний або
використовувати сторонній?

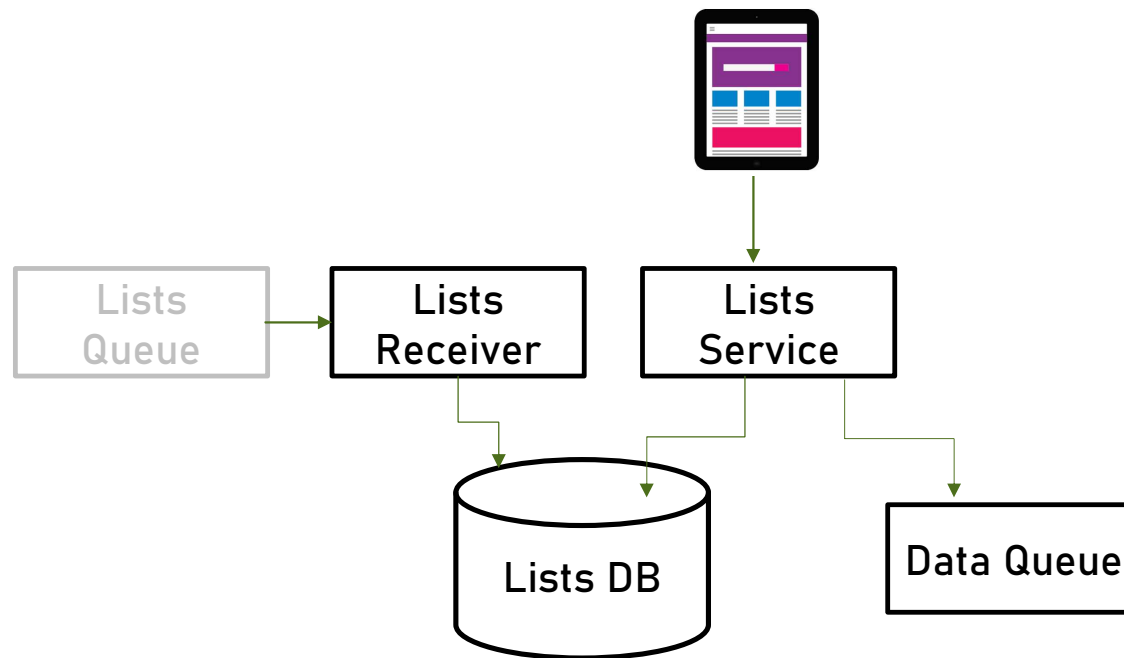
Так





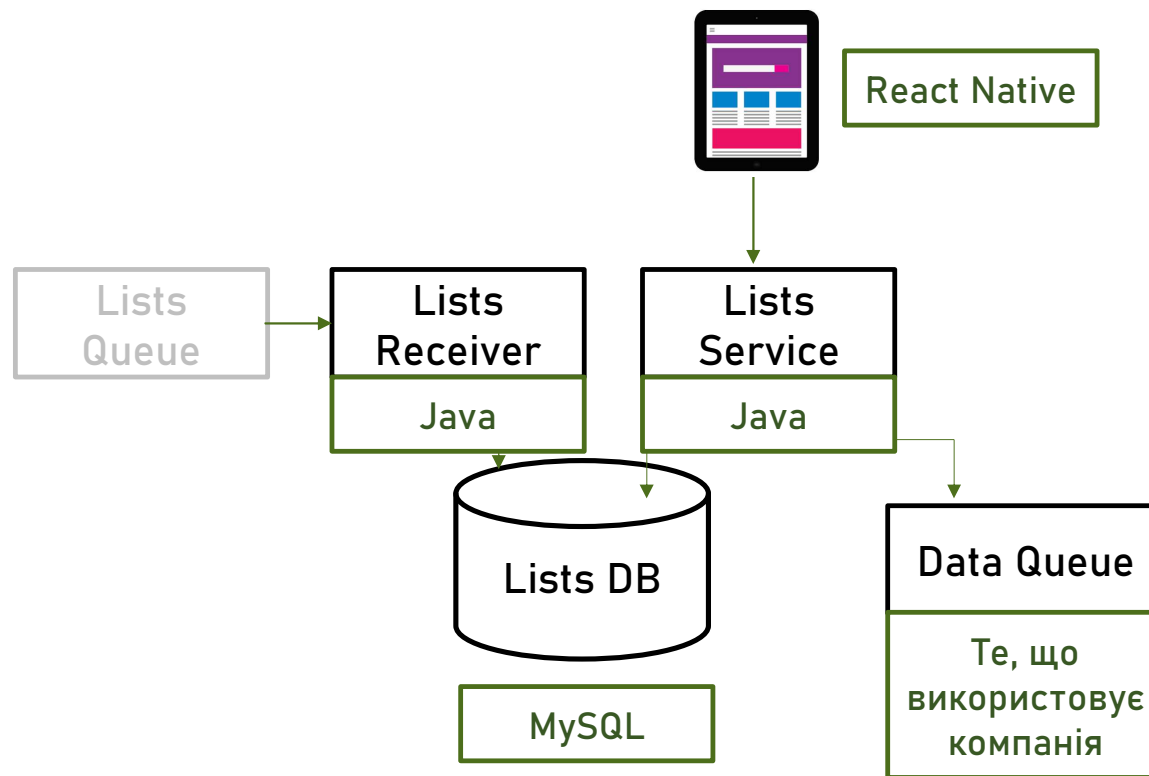


Логічна схема





Технічна схема





Фізична схема

