



Group 16

**JUZCARE**

Software Architecture and Design  
01076024



# GROUP 16



1.นายจิรายุส	เสนาโนฤทธิ	รหัสนักศึกษา 63010151
2.นายธีรพัฒน์	เกตุสิงห์น้อย	รหัสนักศึกษา 63010467
3.นายนาวิธาน	เต๊ะหมาน	รหัสนักศึกษา 63010523
4.นายปิยชัย	แก้วชุ่ม	รหัสนักศึกษา 63010604
5.นายพรหมพิริยะ	เจริญพานทองดี	รหัสนักศึกษา 63010648
6.นางสาวพิชชาภา	เวียงทอง	รหัสนักศึกษา 63010679
7.นางสาวภัทราณิษฐ์	เทศเจริญ	รหัสนักศึกษา 63010727



# JUZCARE


คืออะไร?





# JUZCARE

Web Application ที่สามารถแจ้งปัญหาที่เกิดขึ้นภายในสถาบันได้  
เพื่อแสดงให้เห็นถึงปัญหาที่จำเป็นต่อการแก้ไขแก่ทางสถาบัน  
โดยบุคคลากรภายในหรือบริเวณใกล้เคียงกับสถาบันสามารถแจ้ง  
ปัญหา และแนบรูปปัญหาต่าง ๆ พร้อมสามารถโหวตปัญหาที่  
ต้องการได้รับการแก้ไข เพื่อให้แอดมินคัดกรองและส่งเรื่องไปยัง  
หน่วยงานที่เกี่ยวข้องให้แก้ไขปัญหา





ความสามารถของ **JUZCARE**



# ฟังก็ชื่นทั่วไป

- ดูกระทู้รายงานปัญหาปัจจุบันของผู้ใช้งานคนอื่น
- ดูเทรนด์กระทู้
- คัดกรองการมองเห็นกระทู้ผ่านแท็ก
- ค้นหากระทู้



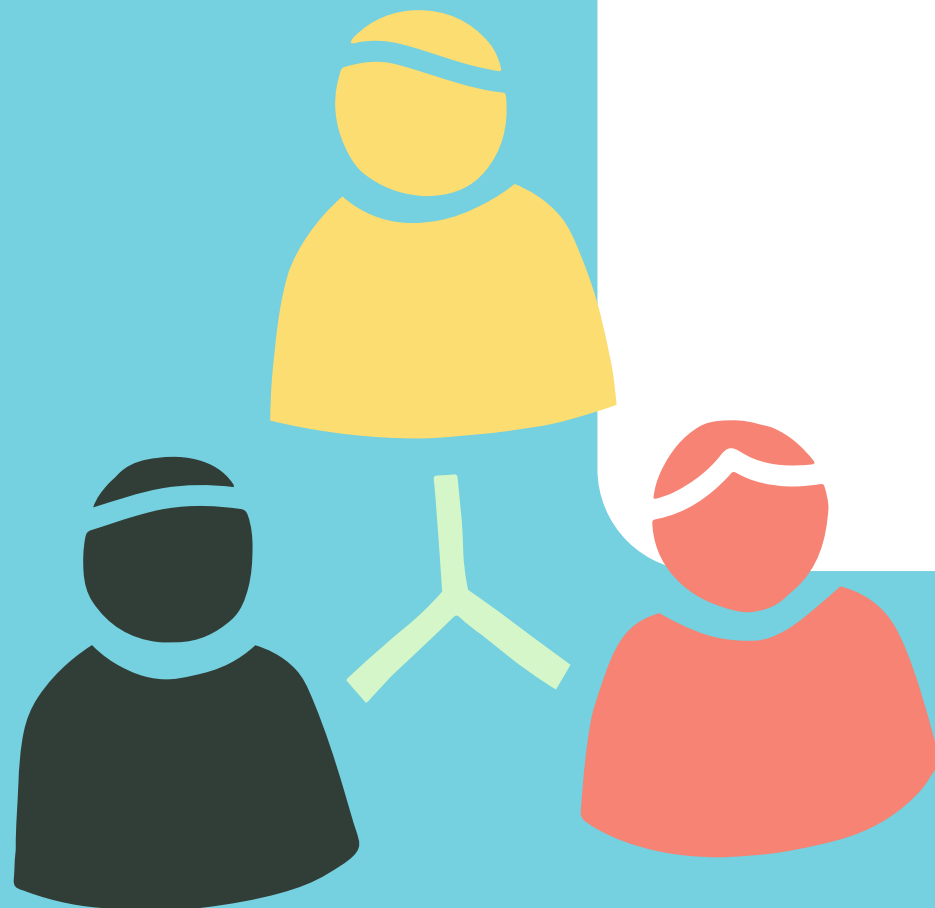


## ฟังก์ชันสำหรับผู้ใช้งาน

- สร้างกระทู้เพื่อแจ้งปัญหาต่าง ๆ ภายในสถาบัน
- แนบรูปและประวัติโอของปัญหา
- รีวิวความพอใจหลังแก้ไขปัญหา
- โหวตเรื่องร้องเรียน
- คอมเมนต์ใต้กระทู้
- Notification
- อัปเดตกระทู้
- ดูประวัติการตั้งกระทู้
- แก้ไขโปรไฟล์ได้

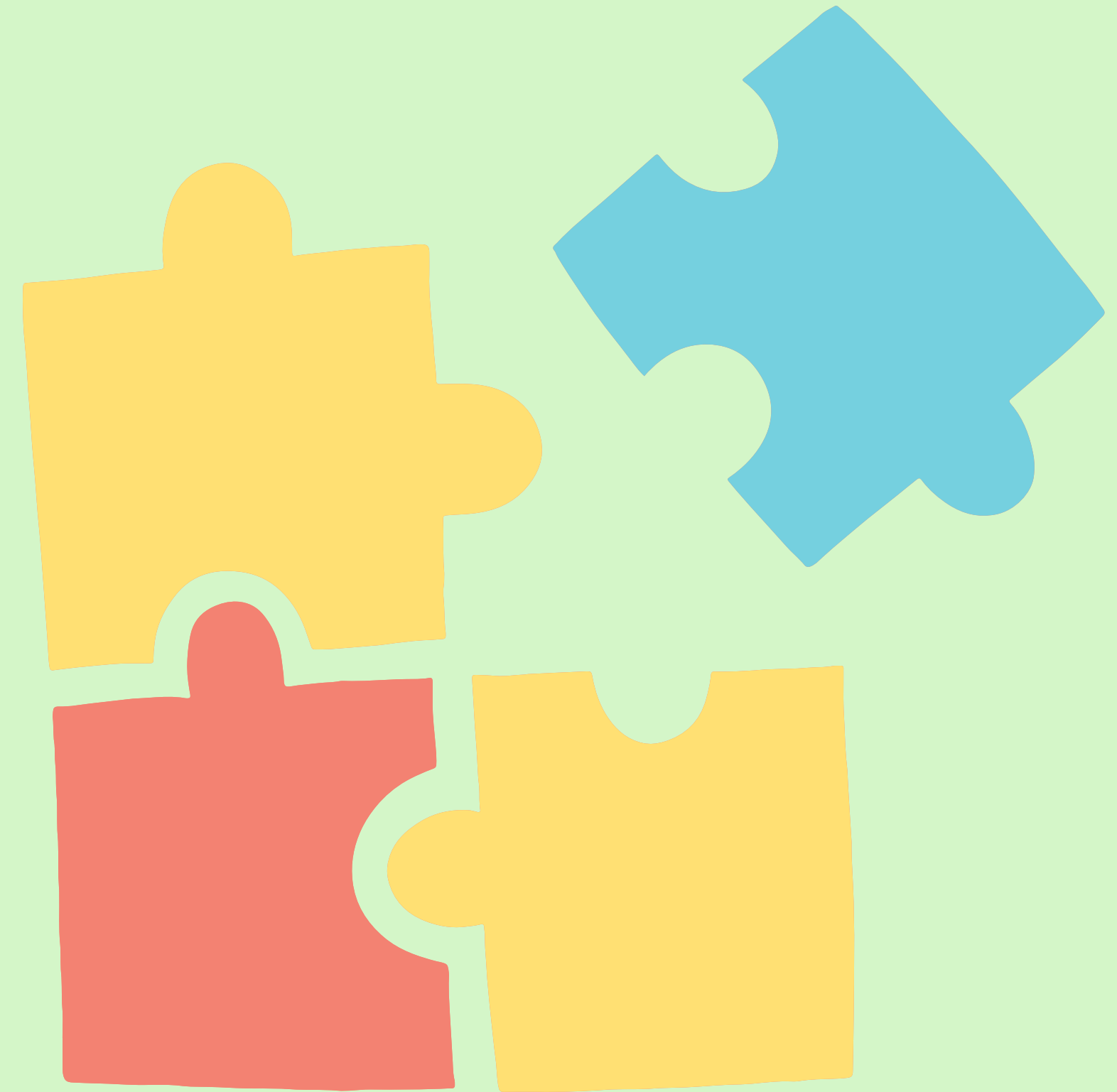
## ฟังก์ชันสำหรับแอดมิน

- ดูปัญหาในแต่ละพื้นที่
- คัดกรองปัญหา เพื่อให้หน่วยงานที่เกี่ยวข้องแก้ไขปัญหา
- เพิ่มแท็กที่คัดกรองกระทู้
- ดูรีวิวของผู้ที่แจ้งปัญหา
- ban บัญชีผู้ใช้งาน



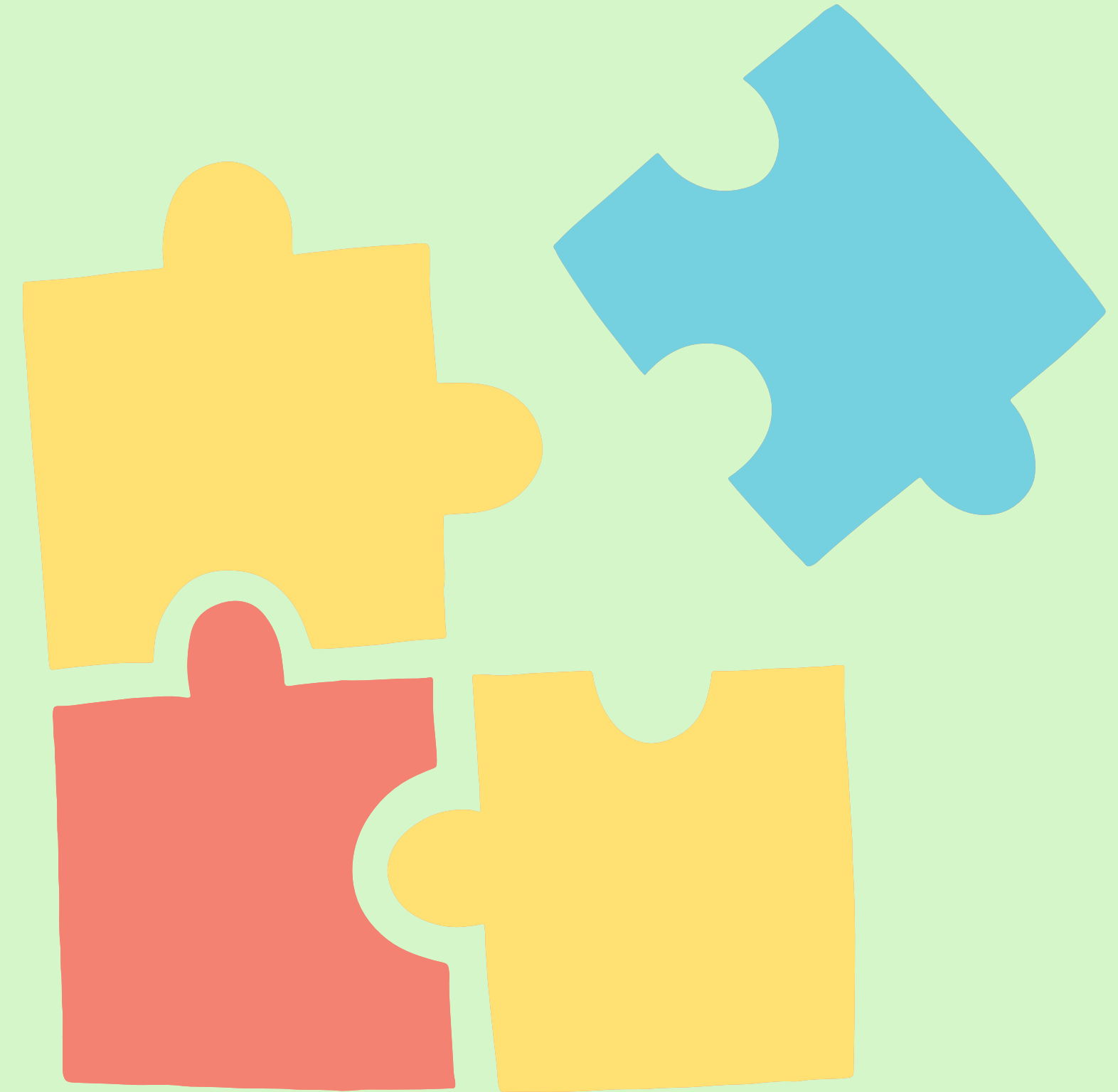


# SOFTWARE ARCHITECTURE

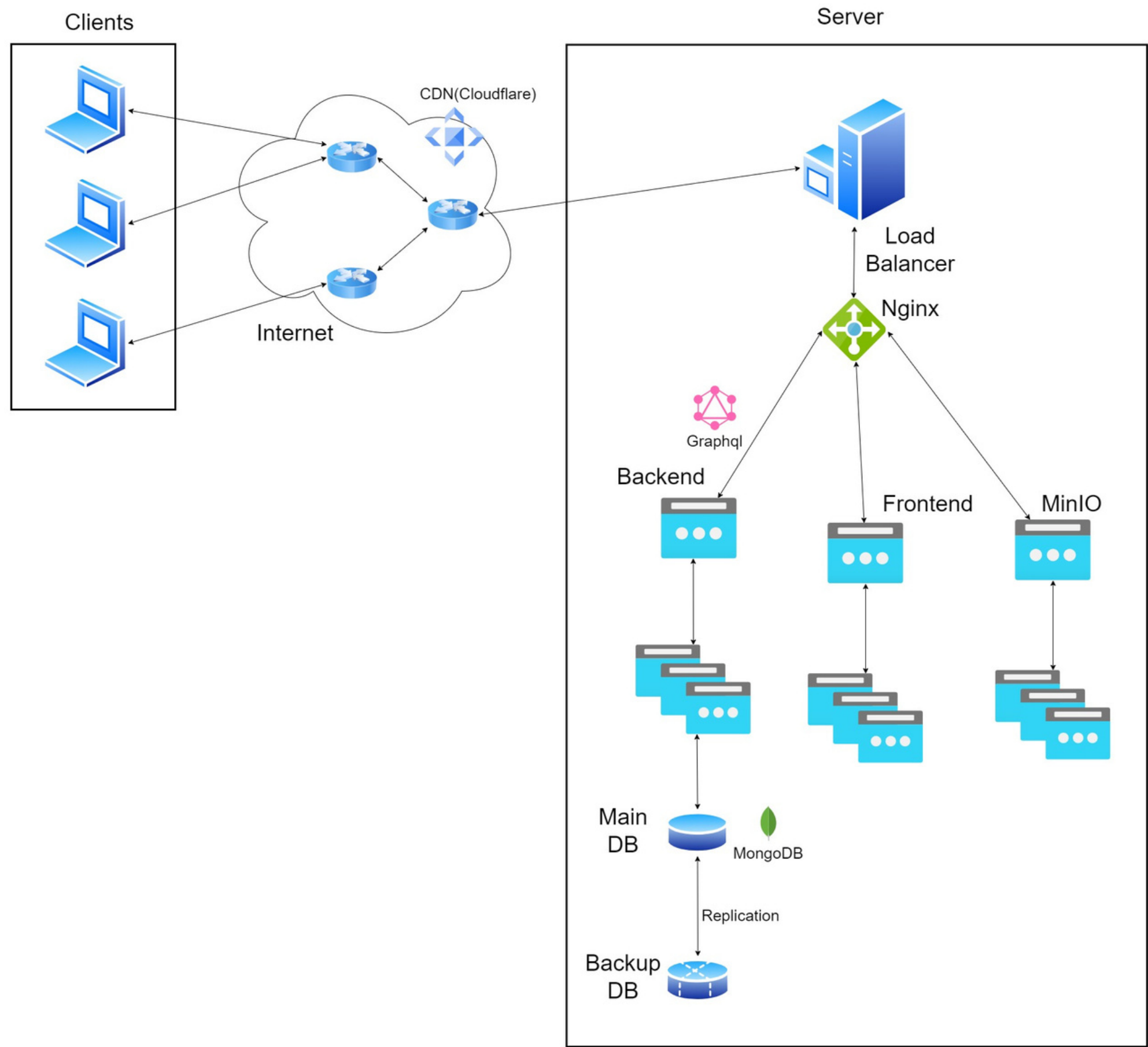


# **SOFTWARE ARCHITECTURE**

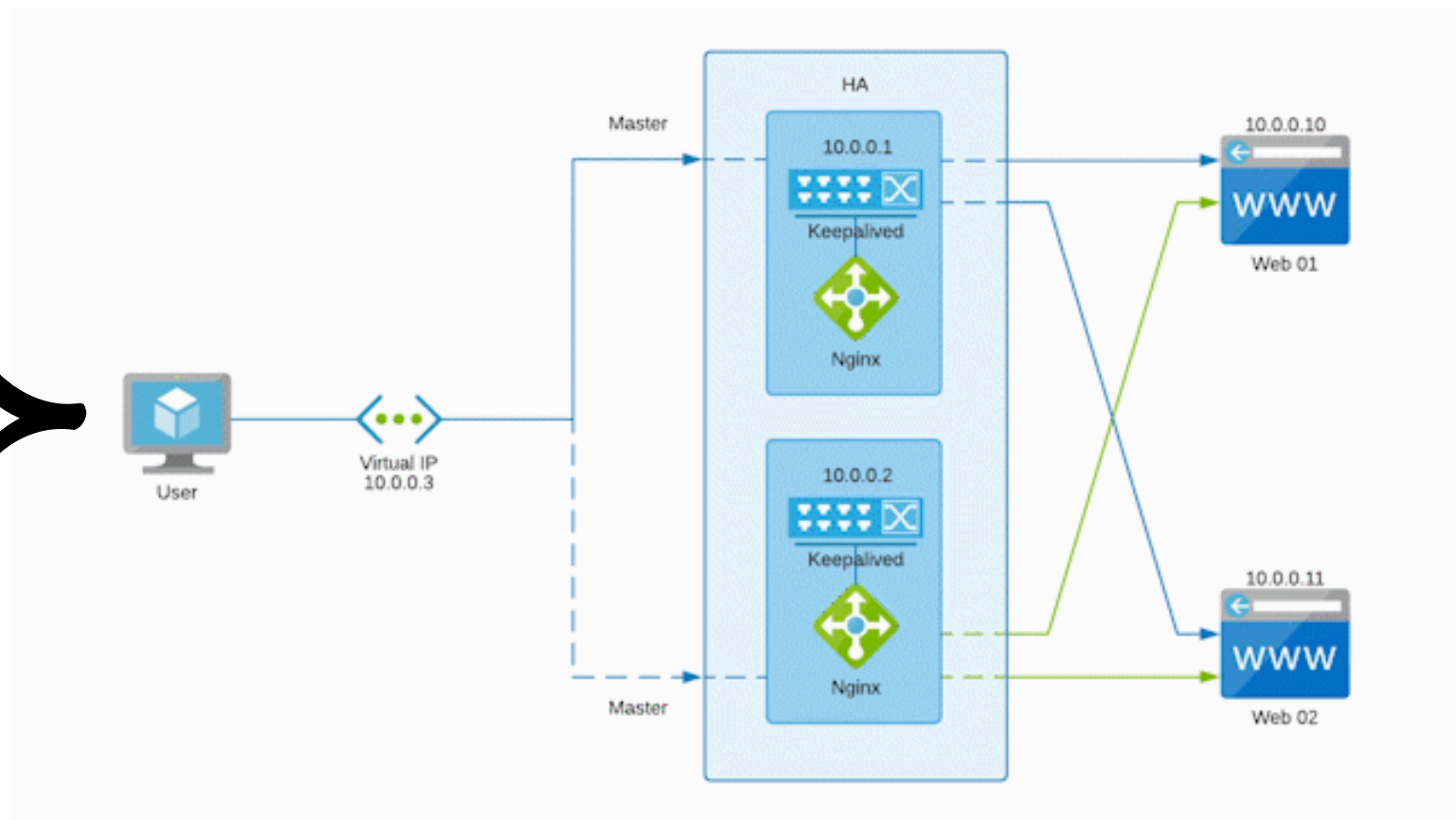
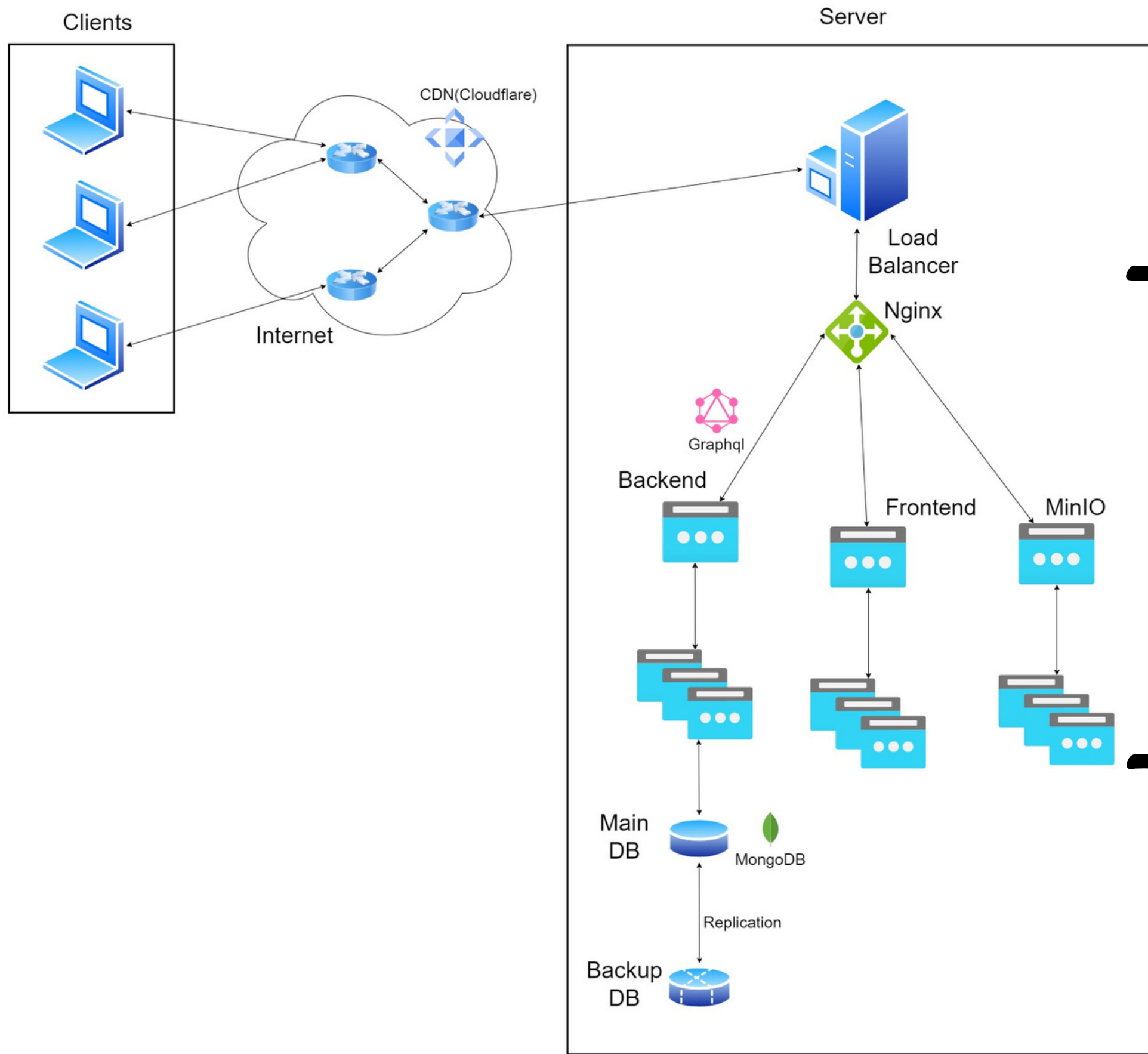
**CLIENT SERVER**



# SOFTWARE ARCHITECTURE

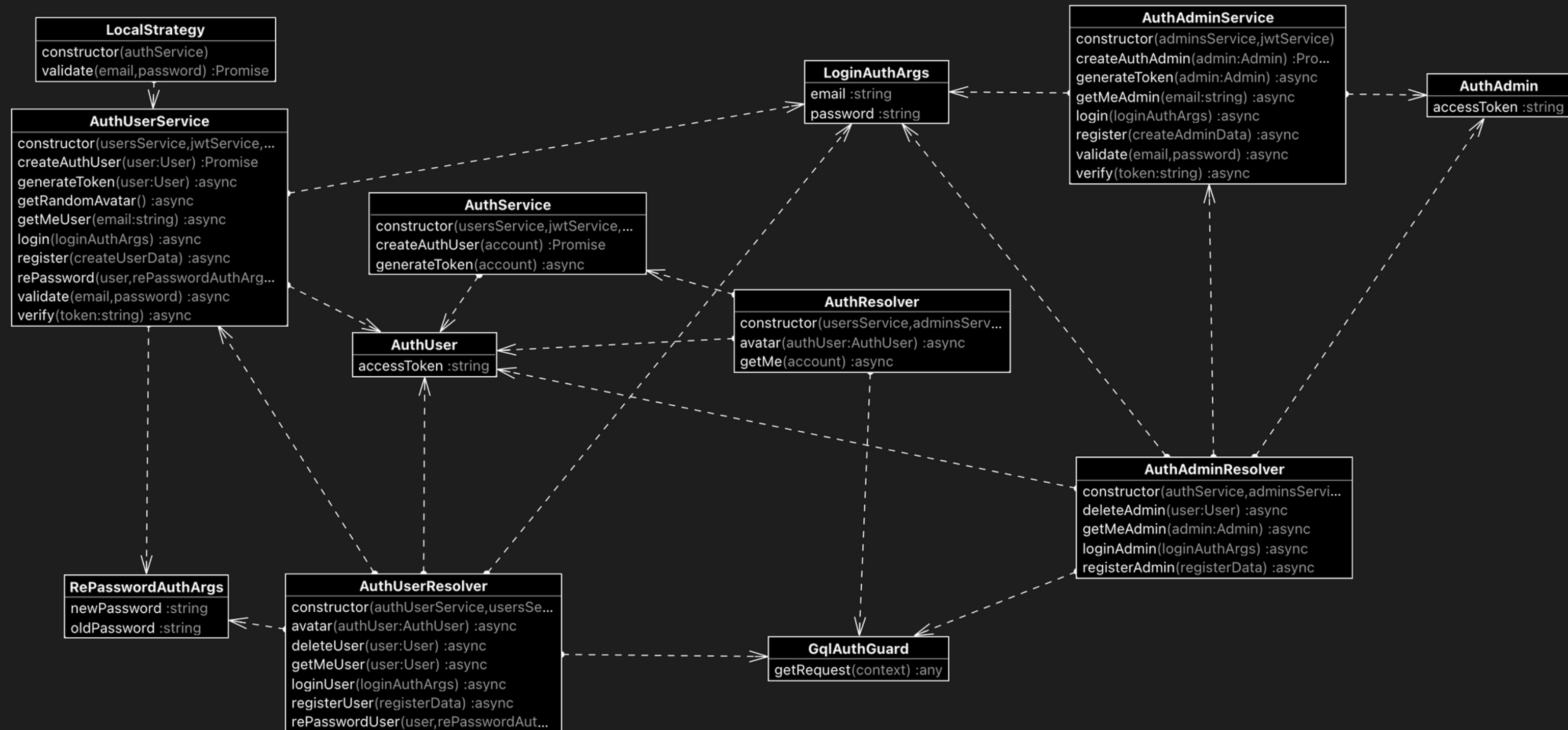


# SOFTWARE ARCHITECTURE



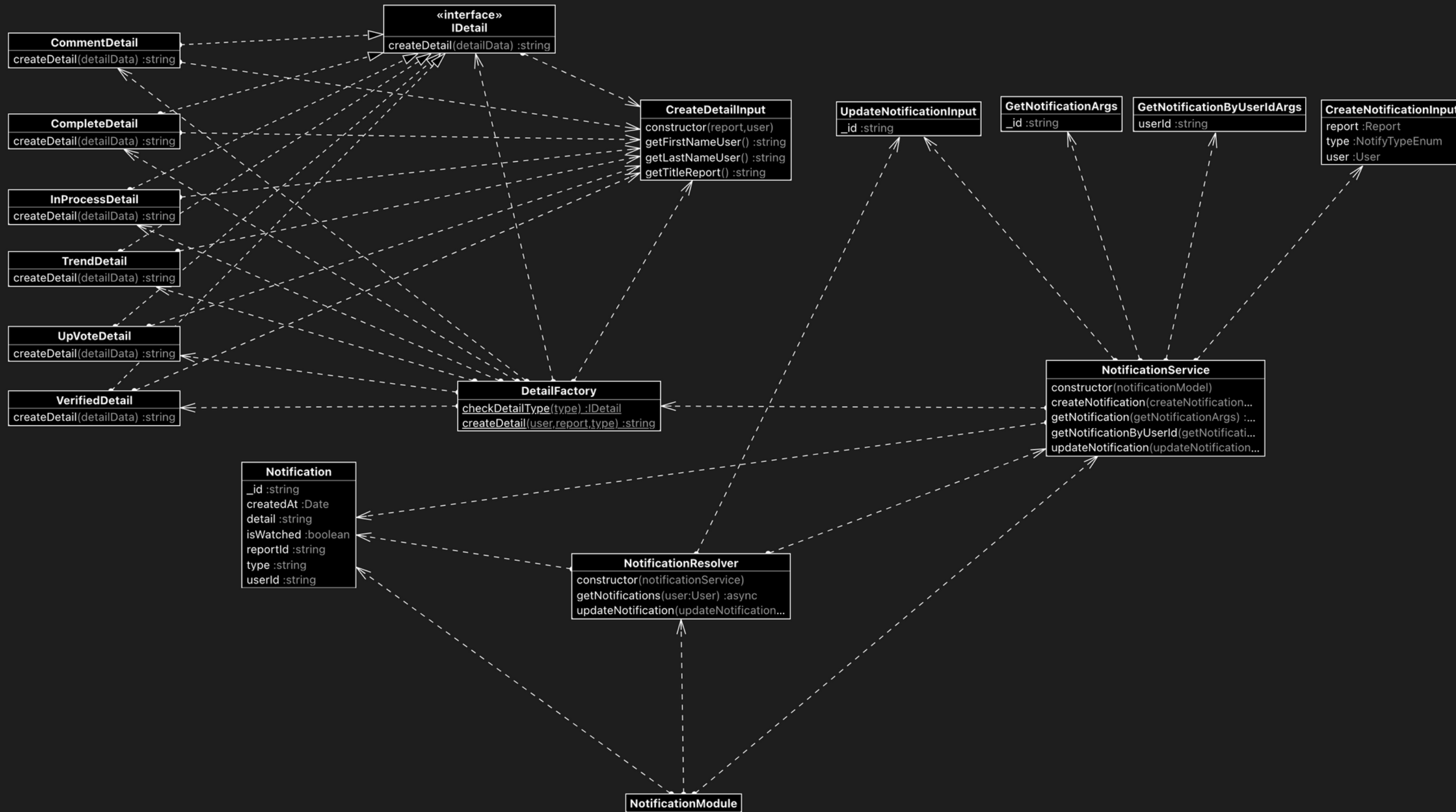
# BOUNDED CONTEXTS

## 1. Authentication



# BOUNDED CONTEXTS

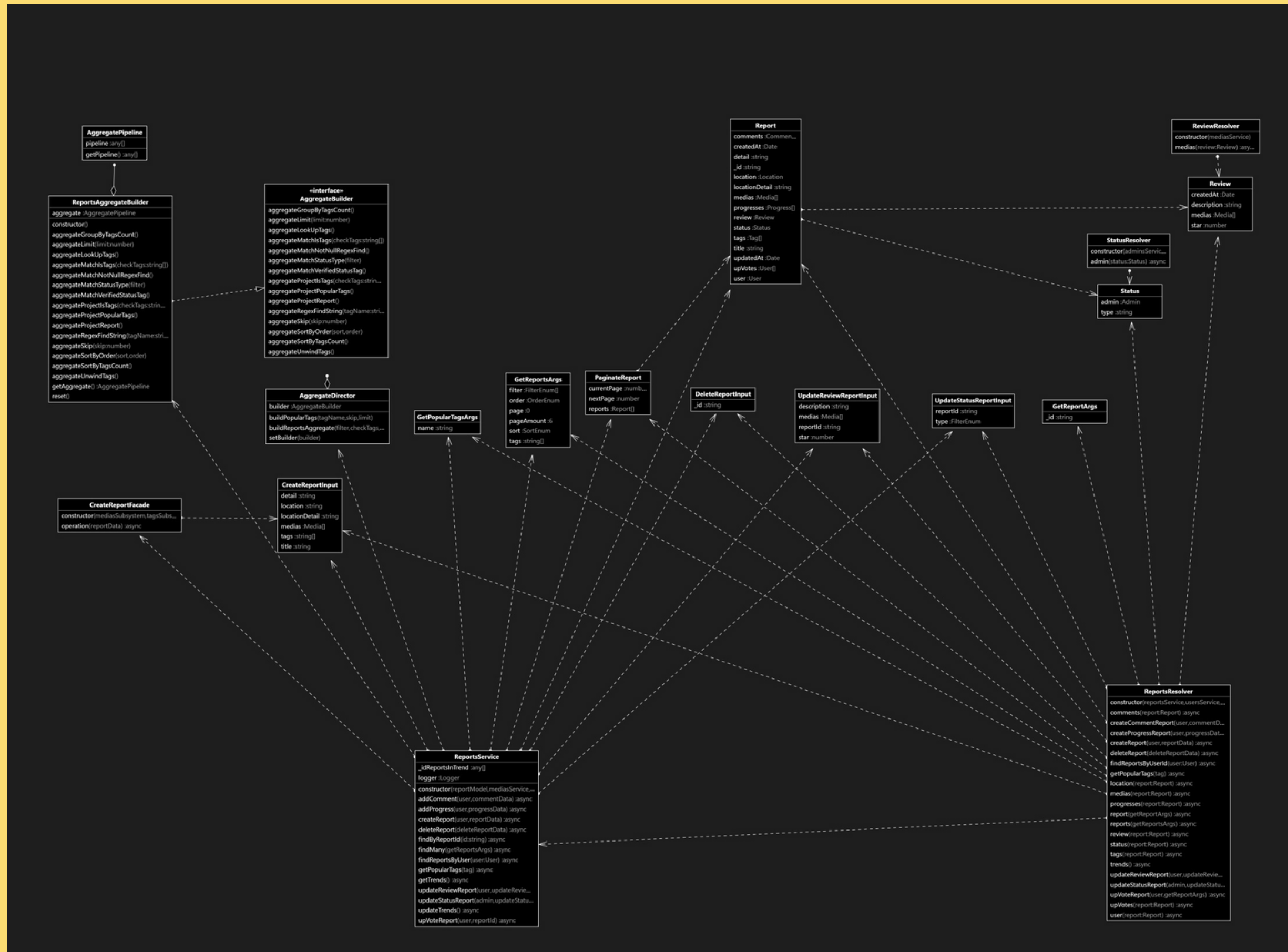
## 2. Notification



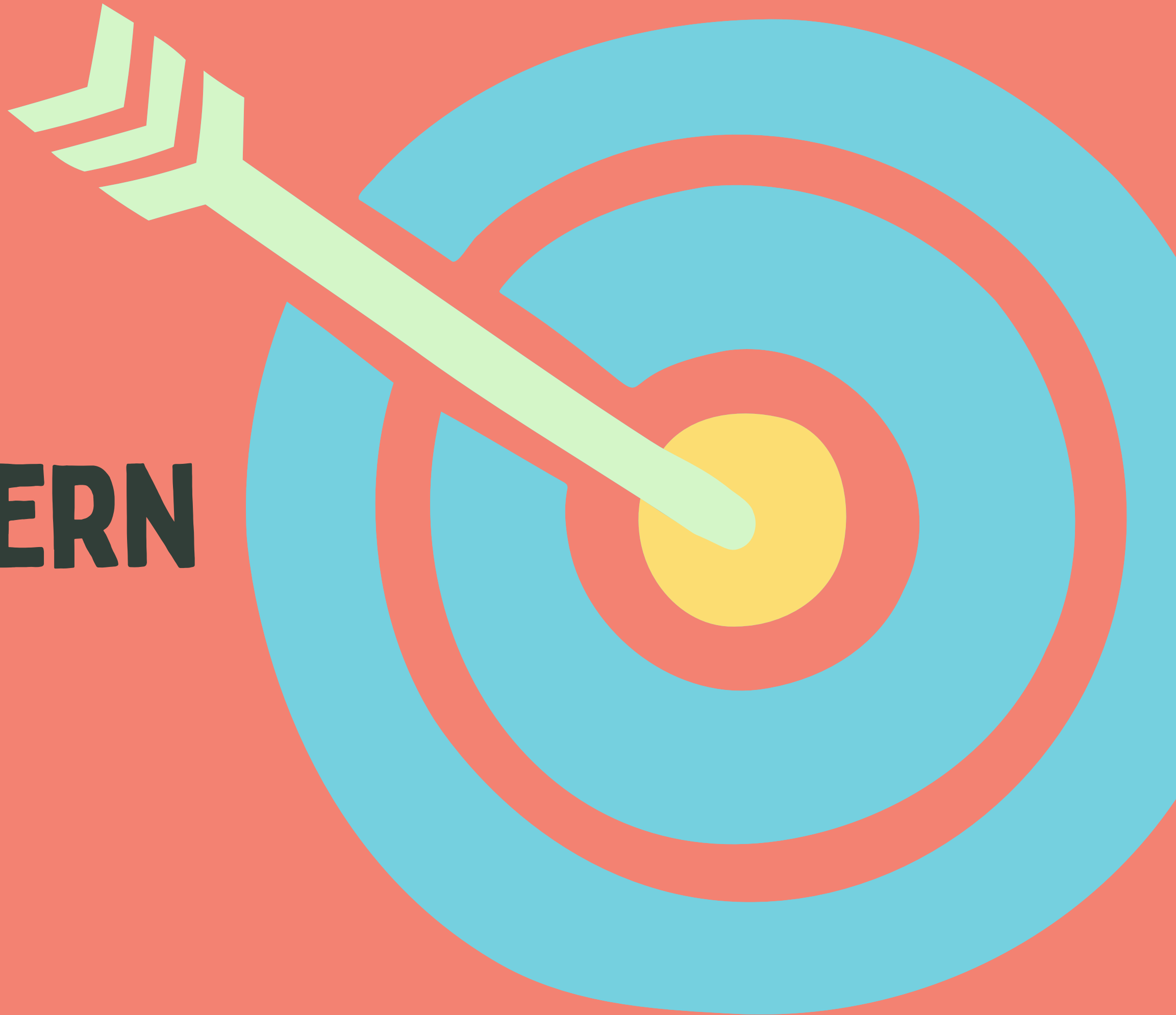


# BOUNDED CONTEXTS

## 3. Reporting

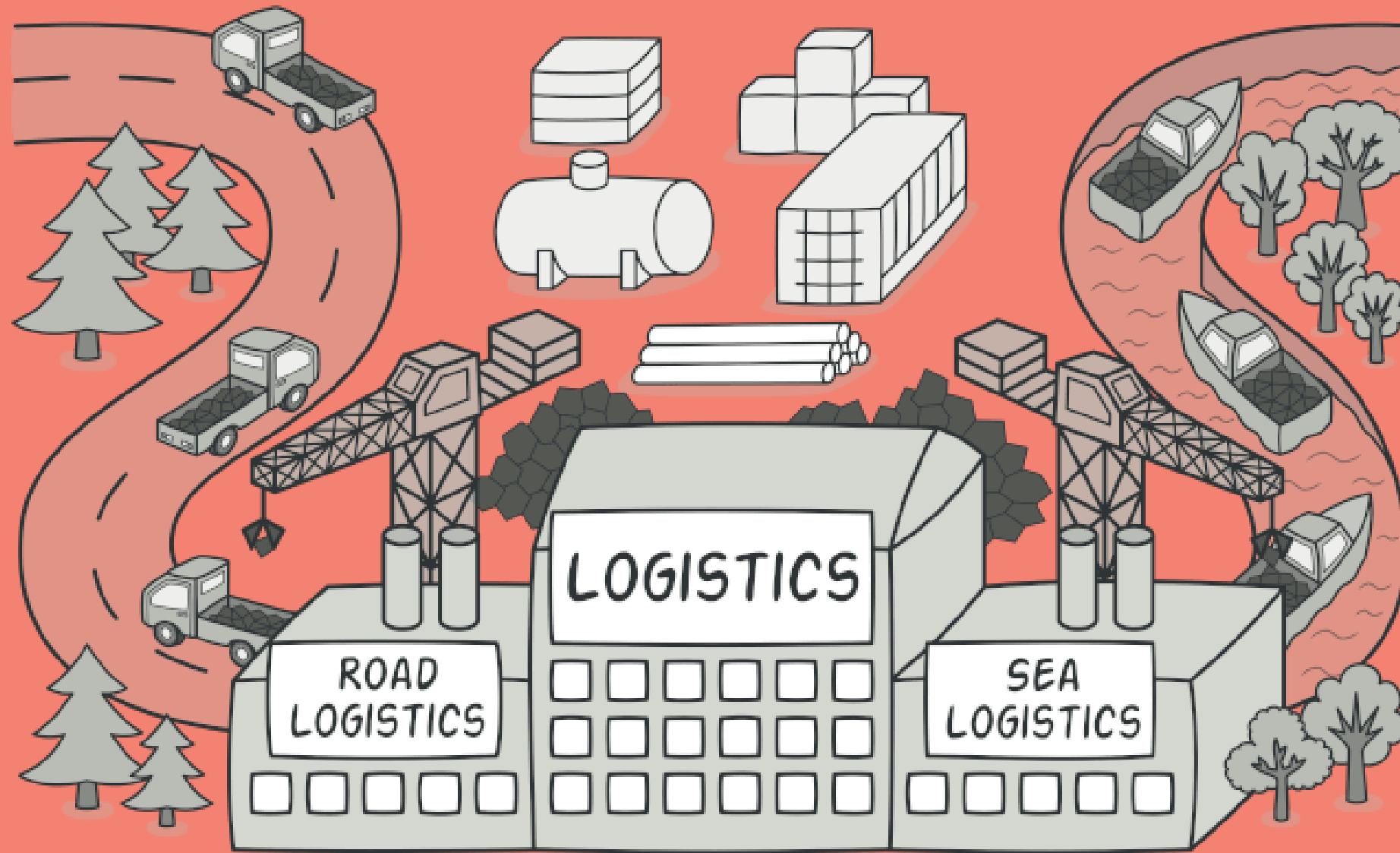


# DESIGN PATTERN





# DESIGN PATTERN



## 1. Factory method

# DESIGN PATTERN

## 1.Factory method



ทำไมถึงใช้?

เพราะว่าเว็บของเรามี detail ของ notification  
ที่หลากหลายแบบ เลยทำการสร้าง concrete class  
เพื่อรองรับรูปแบบการแจ้งเตือนเหล่านั้น

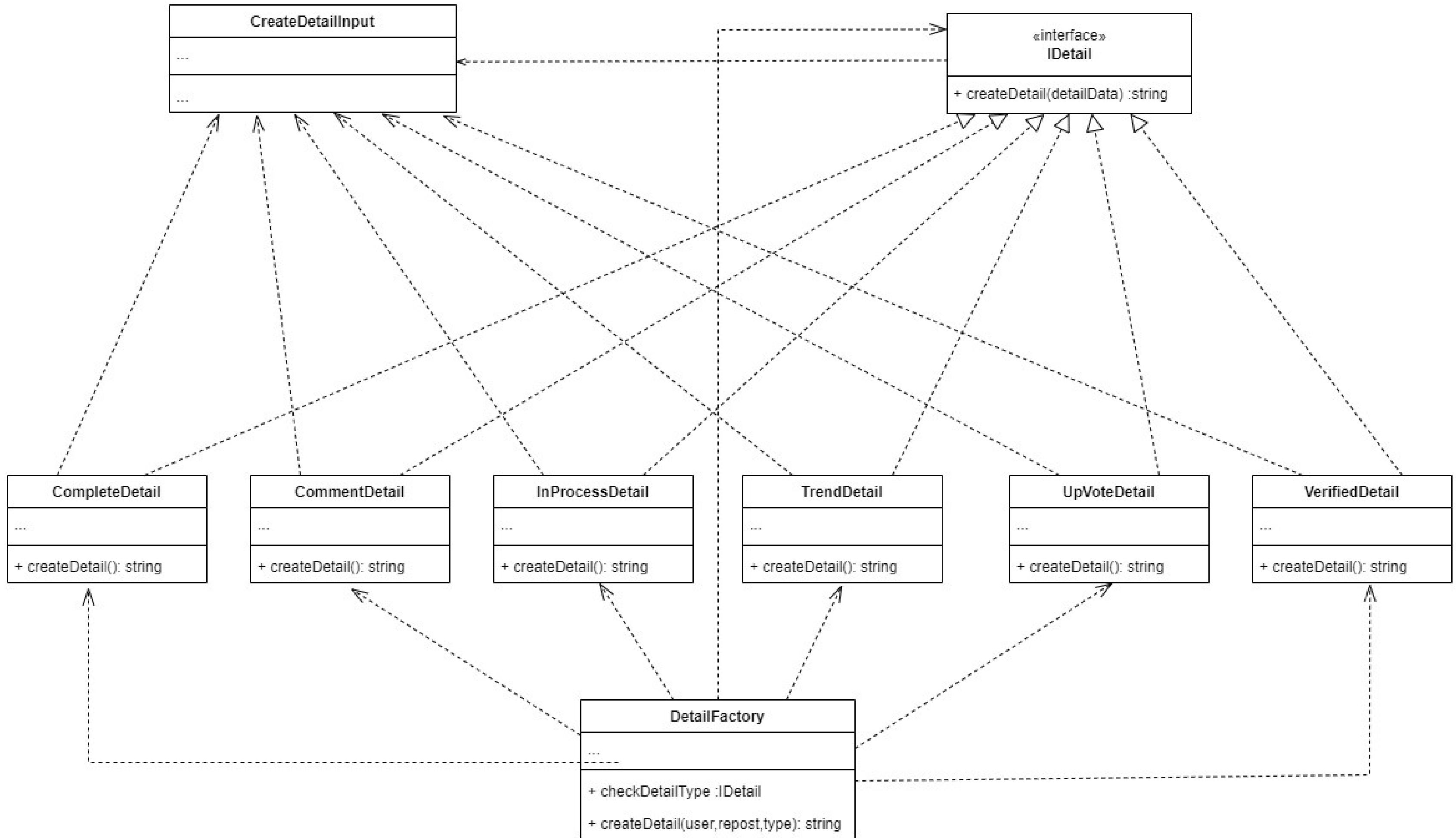
# DESIGN PATTERN

## 1.Factory method



ใช้อย่างไร?

ใช้ในเรื่อง notification ที่มีหลายแบบ เช่น  
การแจ้งเตือนสำหรับ Trend, Comment,  
InProgress, Complete, UpVote, Verified



# DESIGN PATTERN

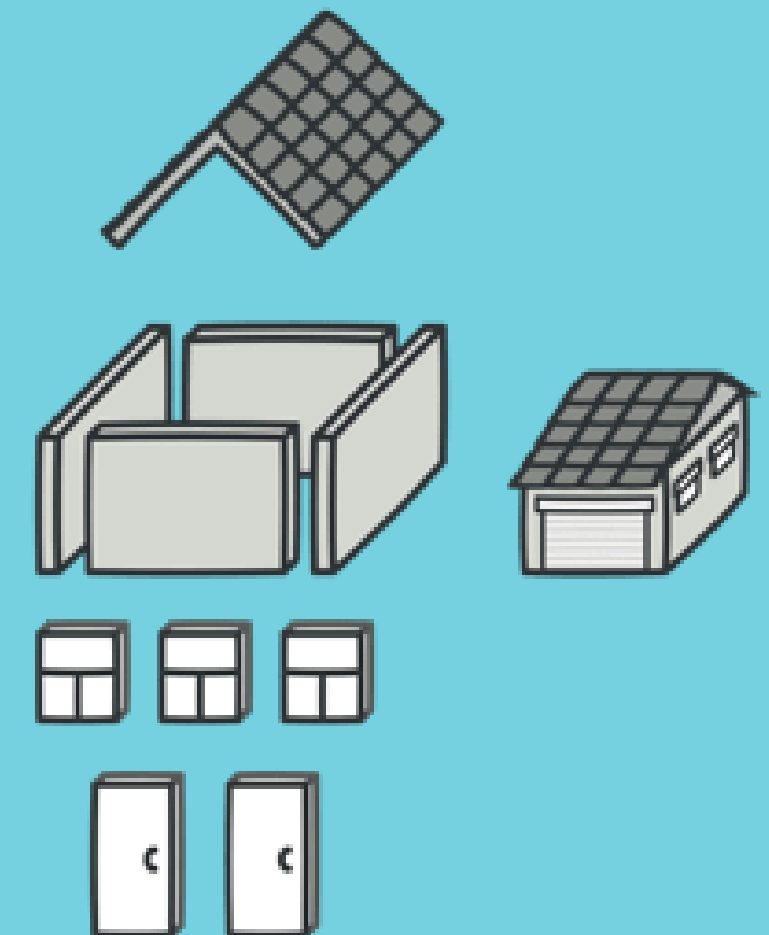
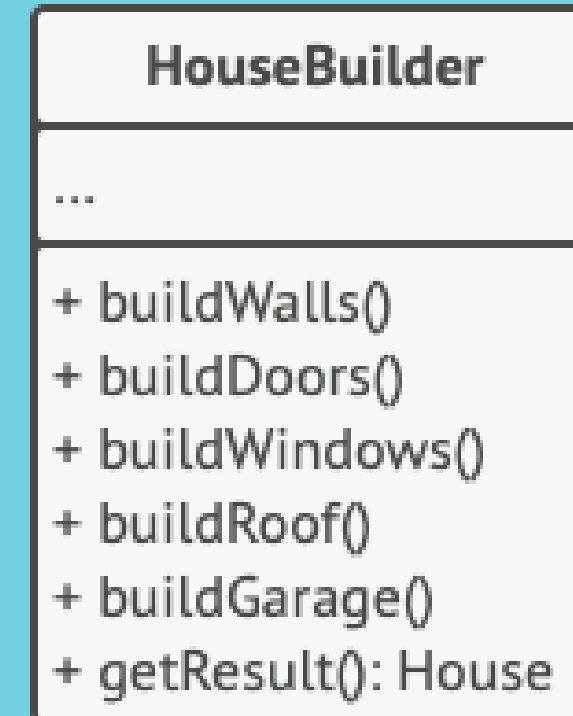
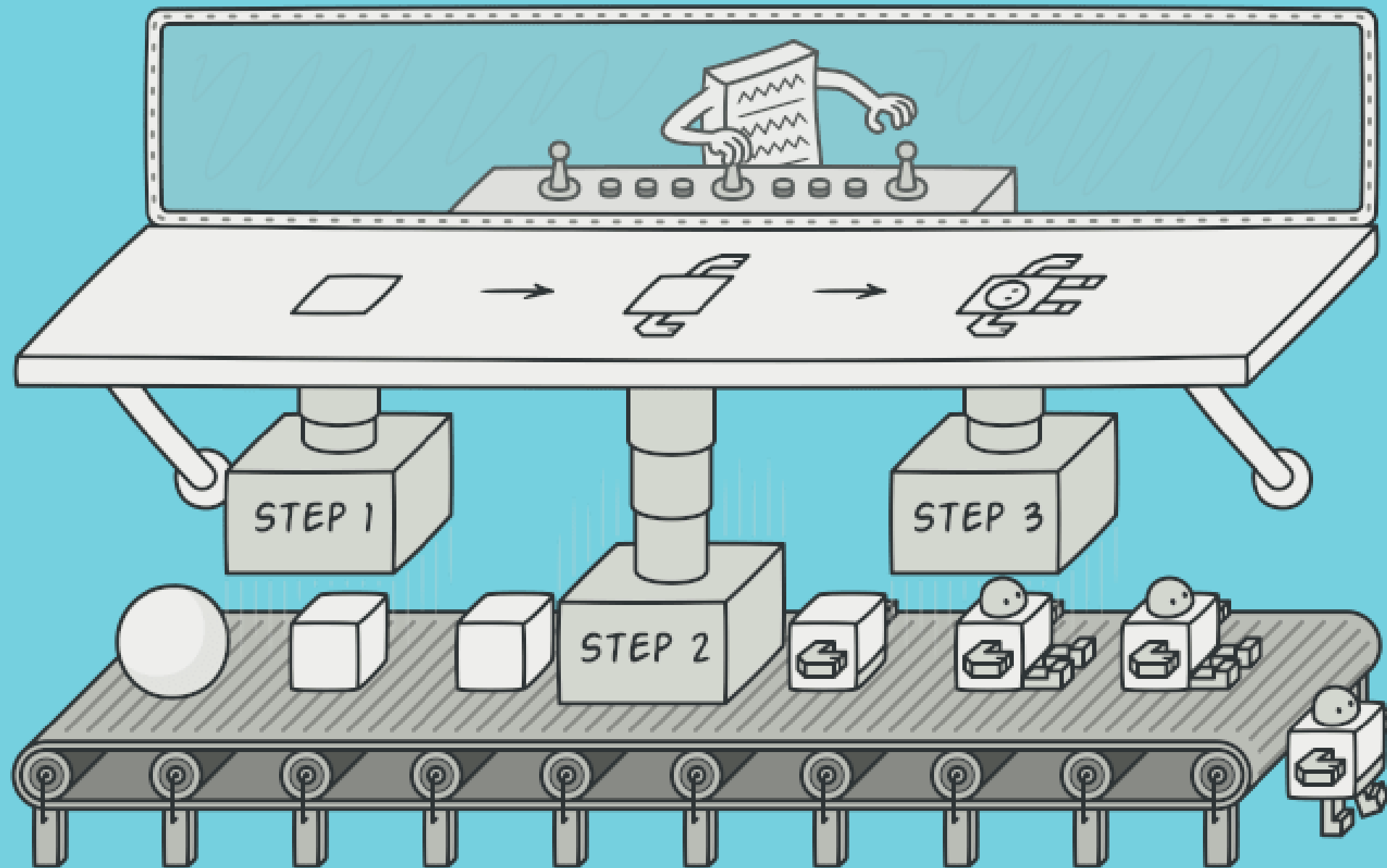
## 1.Factory method



ใช้แล้วมีผลดียังไง?

ใช้แล้วทำให้โค้ดที่เรียกใช้งานมีความยืดหยุ่น  
กระบวนการการสร้าง object กับ object  
แยกออกจากกัน

# DESIGN PATTERN



## 2. Builder

# DESIGN PATTERN

## 2. Builder



ทำไมถึงใช้?

เพราะช่วยแก้ปัญหาเวลาที่มีการสร้าง object  
หลาย ๆ อย่าง ที่มีขั้นตอนคล้าย ๆ กัน  
และช่วยแก้ปัญหาเวลาที่มีการสร้าง class ที่ซับซ้อน

# DESIGN PATTERN

## 2. Builder



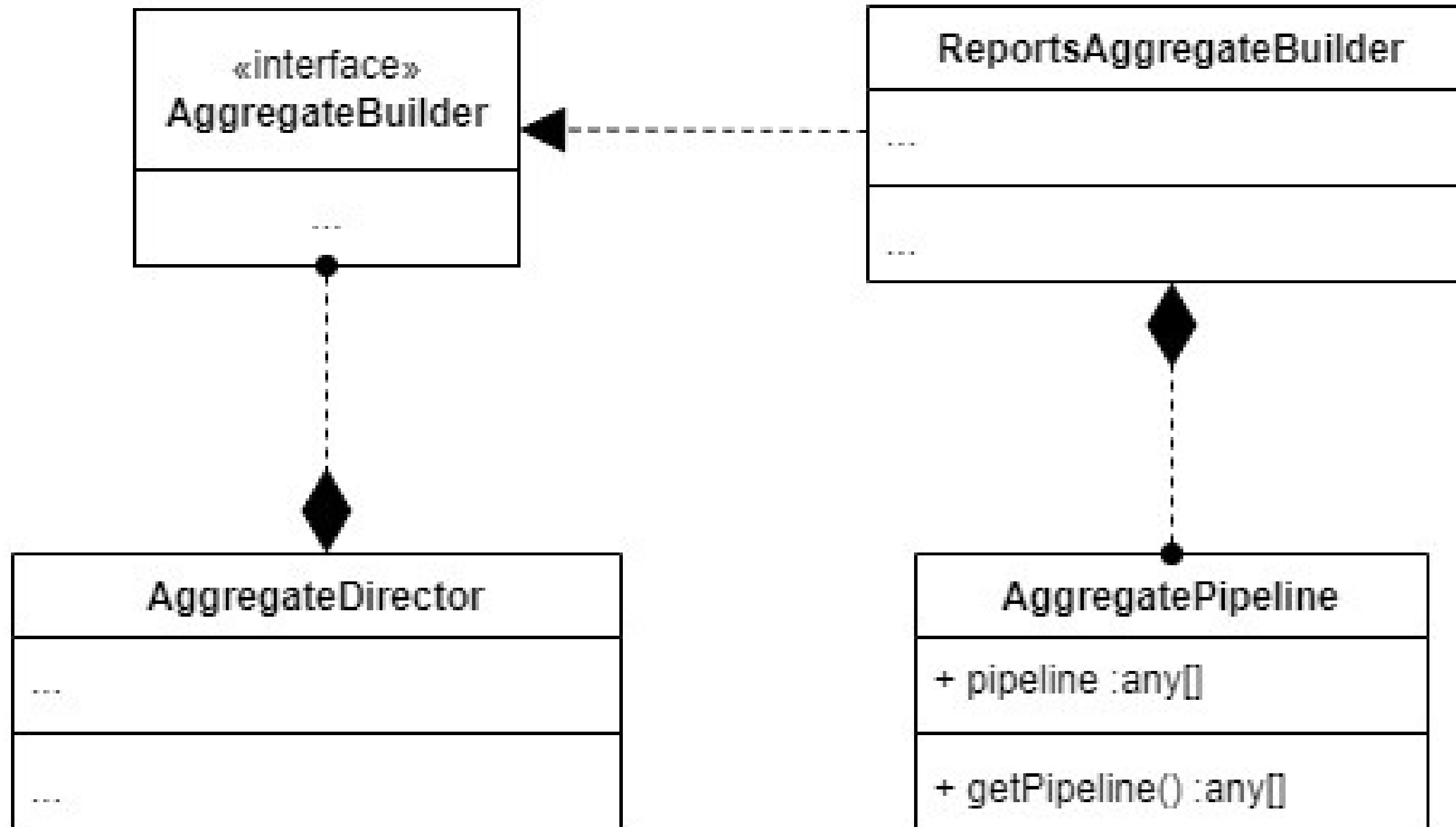
ใช้อย่างไร?

เช่น การสร้างรูปแบบการ Query ถ้ารูปแบบการ Query ใหม่ มีบางส่วนที่เหมือนกับรูปแบบเดิม ก็จะนำ ส่วนของการ Query แบบ เก่า มาต่อด้วยรูปแบบการ Query ใหม่ที่ต่างไปจากเดิม โดยที่ไม่ต้องเขียนรูปแบบการ Query เก่าขึ้นมาใหม่อีกครั้ง



# DESIGN PATTERN

## 2. Builder



# DESIGN PATTERN

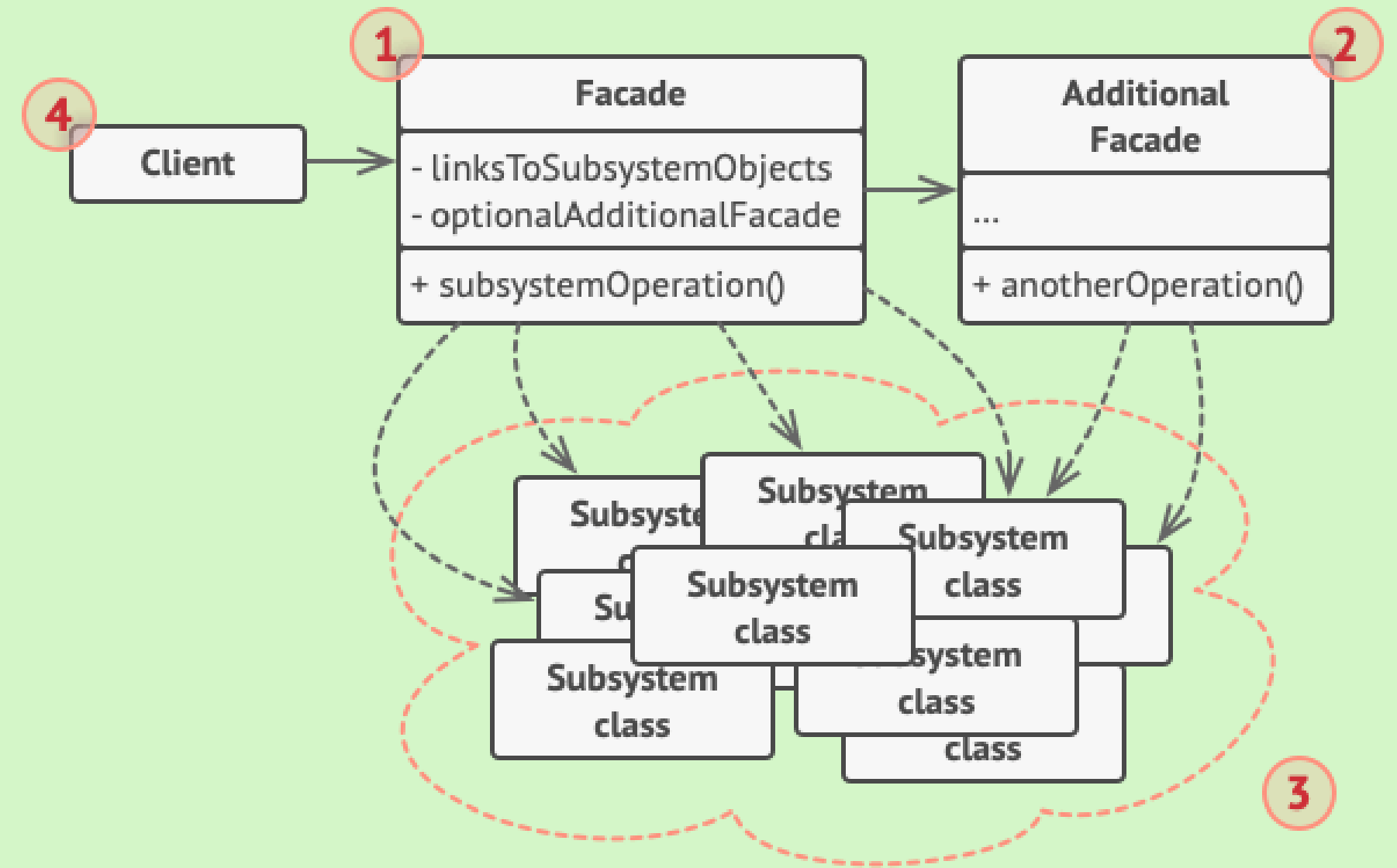
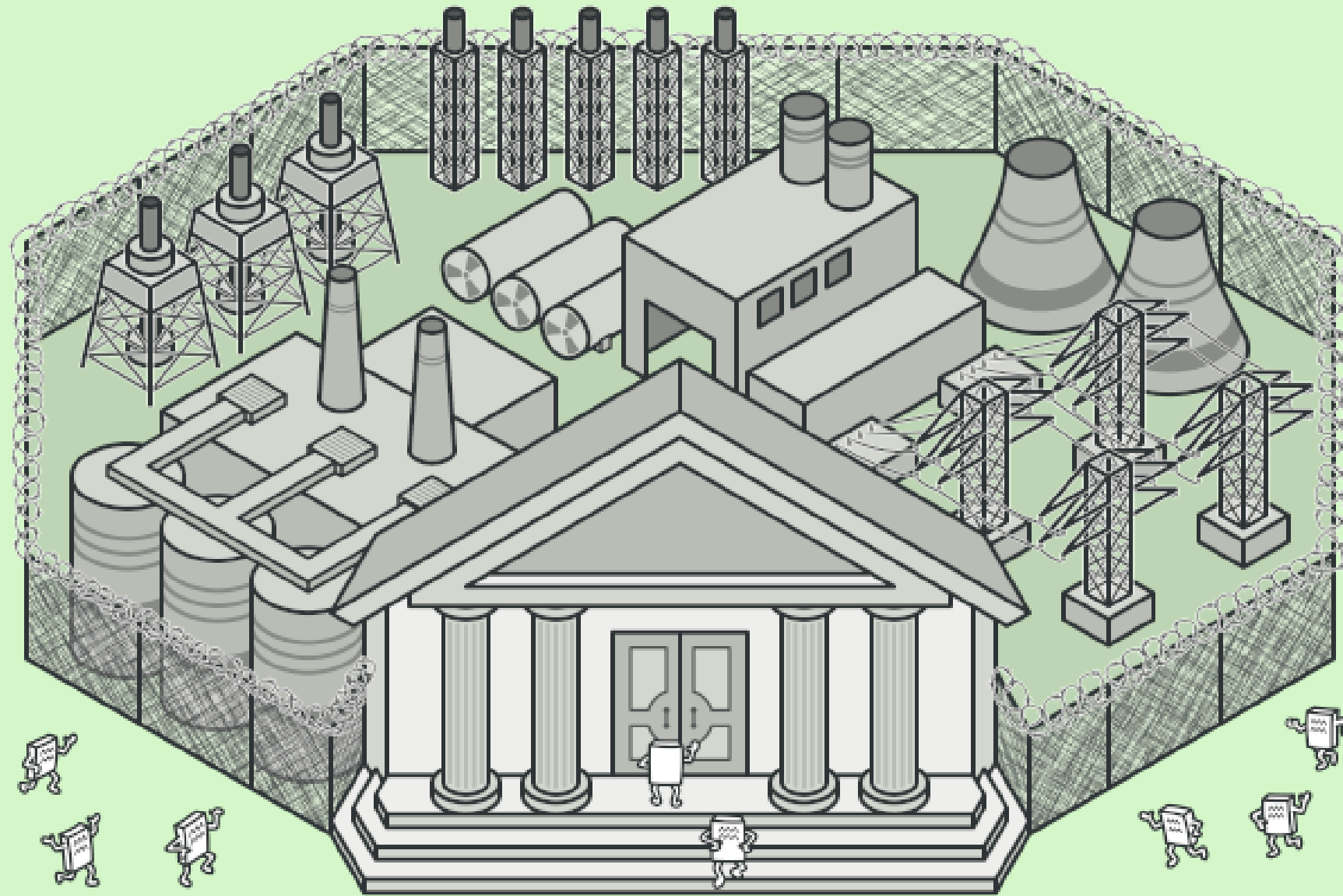
## 2. Builder



ใช้แล้วมีผลดียังไง?

- ช่วยให้ประหยัดเวลาเพราะไม่ต้องเขียนโปรแกรมหรือรูปแบบการ Query เดิมซ้ำอีกครั้ง
- ช่วยทำให้สะดวกในการสร้าง class จาก builder ที่เรากำหนด โดยกำหนดคุณสมบัติจากการเรียกใช้ method ต่างๆ

# DESIGN PATTERN



## 3. Facade

# DESIGN PATTERN

## 3. Facade



ทำไมถึงใช้?

แก้ไขปัญหาในเรื่องการที่ Client จะต้องการเรียกใช้ Subsystem หลายๆตัว จึงแก้ไขปัญหาดังกล่าวโดยการสร้าง Facade Class ที่ใช้ Method มาเรียกใช้ Subsystem นั้นๆ และให้ Client เรียกใช้ Facade Class นั้นแทนการเรียก Subsystem หลายๆตัว

# DESIGN PATTERN

## 3. Facade

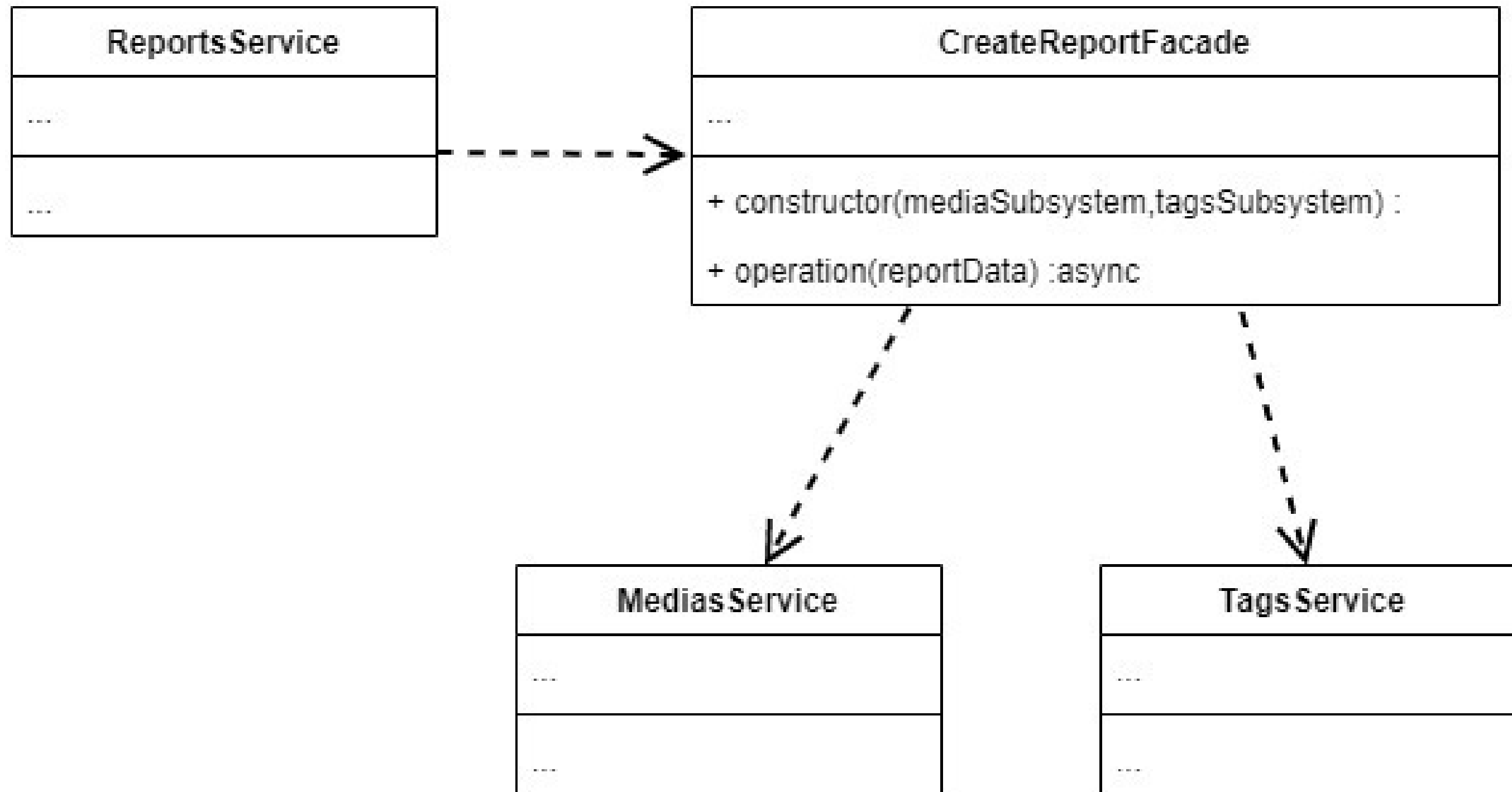


ใช้อย่างไร?

เช่น Client สามารถดึงข้อมูลการสร้างกระตุ้มมาได้เลยว่า ใครเป็นคนสร้าง หัวข้ออะไร รูปคืออะไร ใช้แท็กอะไร เวลาที่สร้างเมื่อไร โดยไม่ต้องลงไปยุ่งกับ Subsystem

# DESIGN PATTERN

## 3. Facade



# DESIGN PATTERN

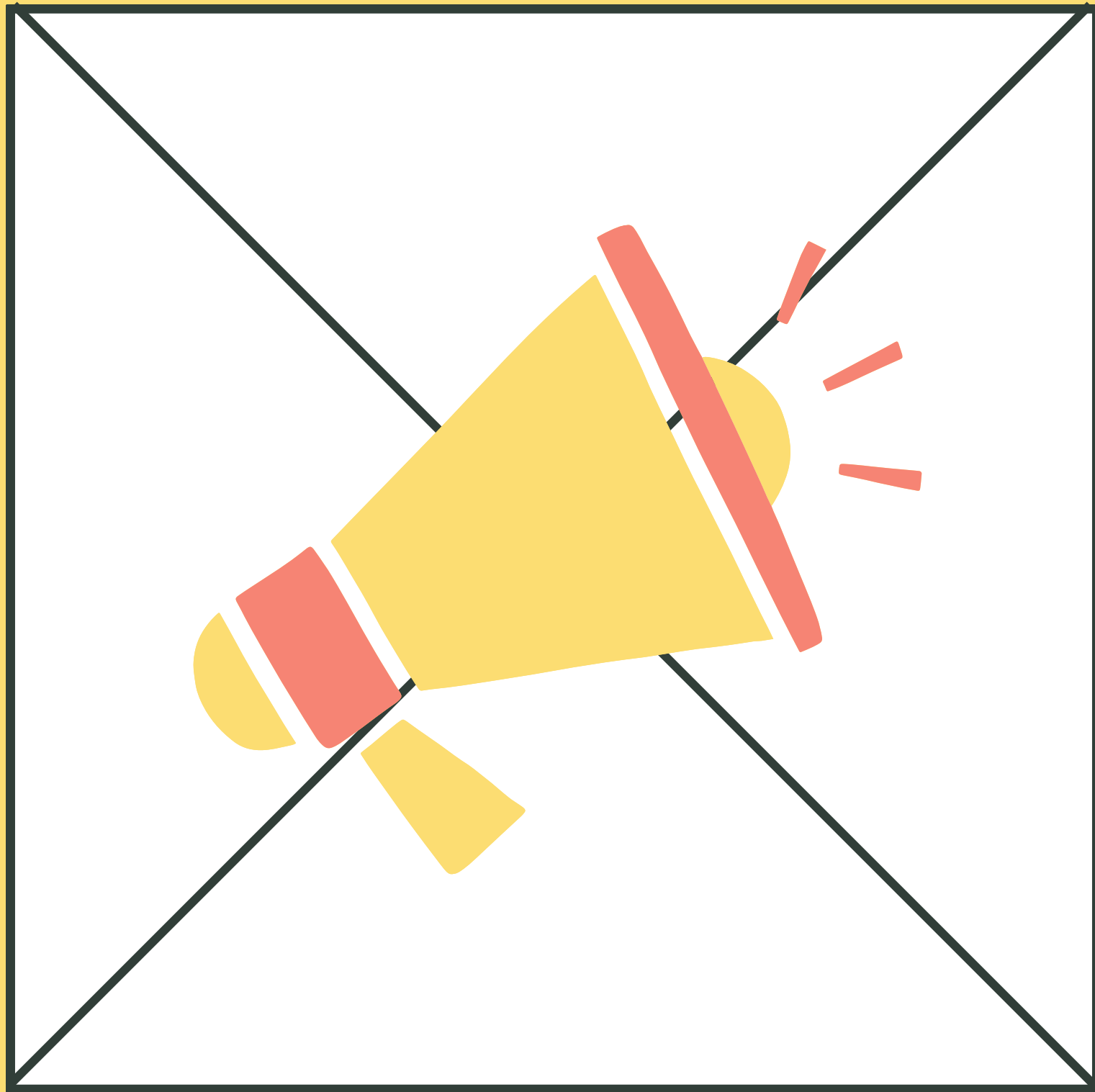
## 3. Facade



ใช้แล้วมีผลดียังไง?

ช่วยให้สามารถประยุกต์ใช้งานกับไลบรารีอื่น ๆ ได้ง่ายขึ้น  
แล้วสามารถเพิ่มหรือแก้ไขฟังก์ชันการทำงานได้ง่ายขึ้น  
มี unified interface เพื่อเชื่อมต่อกับ subsystem ใน Facade  
จะกำหนด interface ในระดับสูง ที่ทำให้ subsystem ง่ายต่อ  
การใช้งาน

# QUALITY ATTRIBUTE



- Availability
- Performance
- Modifiability
- Security
- Usability



# 1. Availability : Load Balancer, MongoDB Cluster



**Load Balancer**

- > load balancer จะมีการ helth check เพื่อตรวจสอบและหยุดส่ง traffic ของ user ไปยัง application ที่อยู่บน server นั้นทันที และส่งต่อไปยัง application บน server ที่พร้อมให้บริการ ซึ่งทำให้ application สามารถให้บริการและใช้งานได้อย่างต่อเนื่อง

**MongoDB Cluster**



- > scale-out ด้วยการทำ load balancing ในกรณีที่ต้องการรองรับการใช้งานของ user ที่มีจำนวนมากขึ้น
- > มีการใช้ MongoDB Cluster (Replication) คือการกระจายข้อมูลท่ามกลาง MongoDB servers หลายๆ node โดย MongoDB สามารถกระจายข้อมูลไปยัง 1 หรือมากกว่านั้นและข้อมูลจะ sync กันตลอดเวลา

A square box with a black border and a large 'X' inside. In the center is a red rounded rectangle containing the text 'Load Balancer' in white.

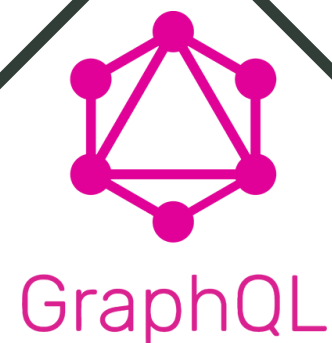
**Load Balancer**

## 2. Performance : Load Balancer, GraphQL

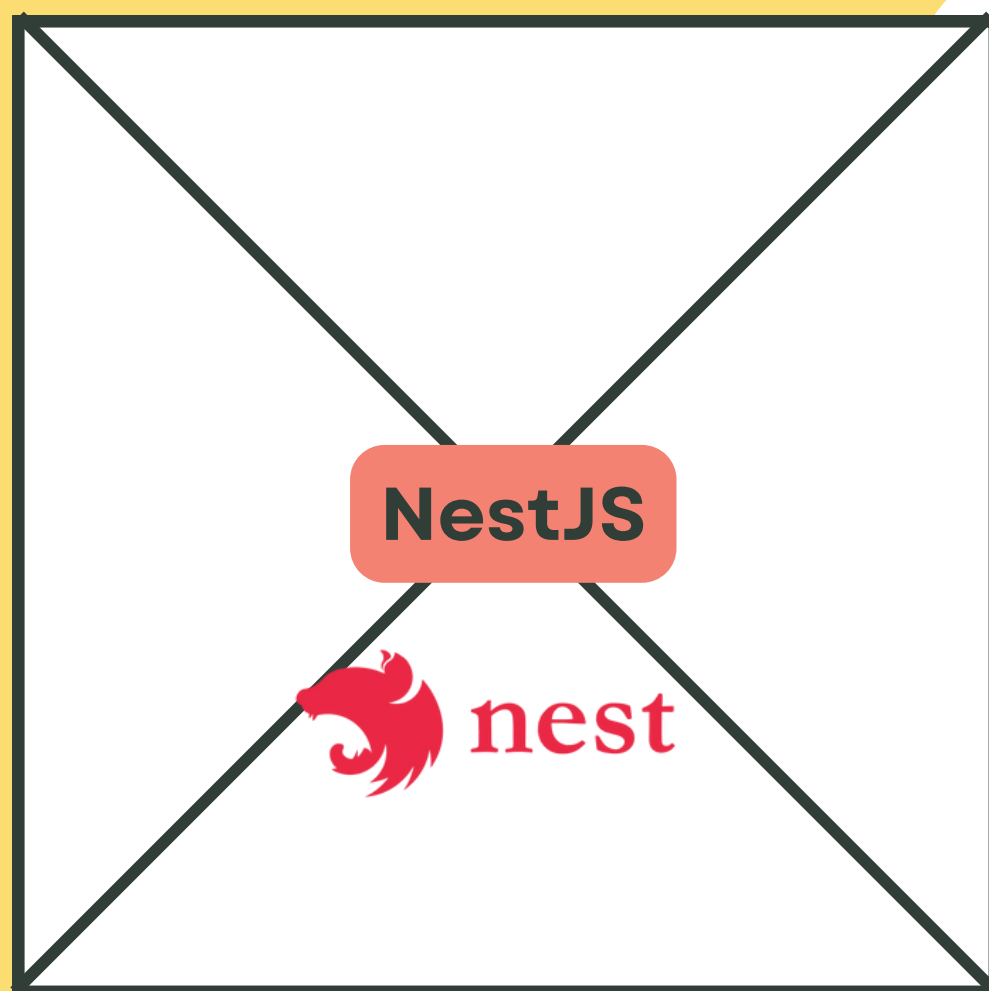
> ช่วยเพิ่มความเร็วของ application หรือ service เนื่องจาก load balancer ทำการกระจาย traffic ของ user ไปยัง application ที่ทำงานอยู่บน server หลายๆ เครื่อง ทำให้การใช้งานของ user ไม่ต้องรอคิว

A square box with a black border and a large 'X' inside. In the center is a red rounded rectangle containing the text 'GraphQL' in white.

**GraphQL**



> ใช้ GraphQL ในเรื่องของ caching เพื่อให้การเรียกใช้ข้อมูลในครั้งต่อไปรวดเร็วมากยิ่งขึ้น และเพื่อให้เราสามารถดึงข้อมูลได้ตรงกับความต้องการ ลดความซับซ้อนในการเขียนโค้ดเพื่อดึงข้อมูล

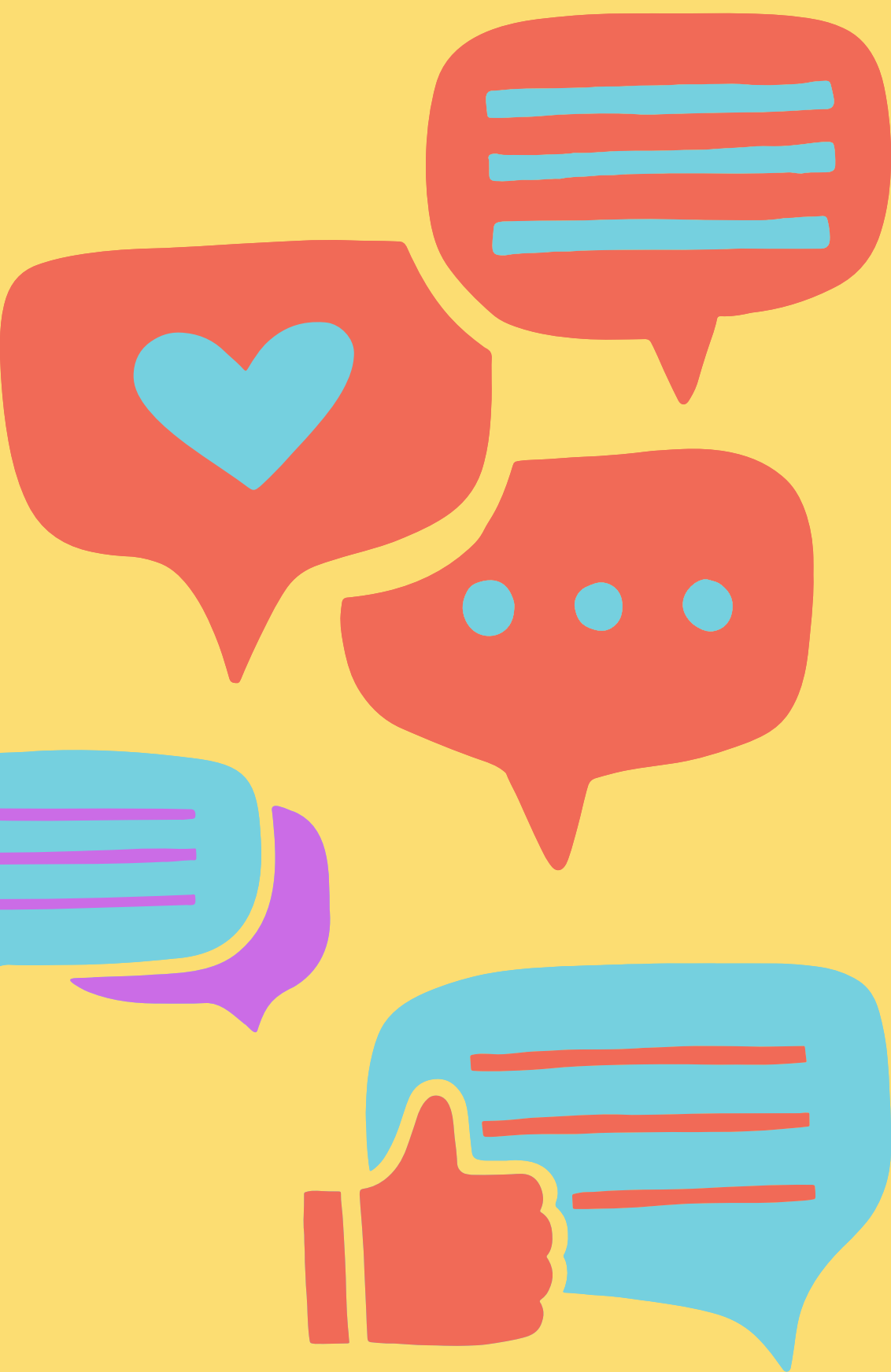


### 3. Modifiability : NestJS

> NestJS คือ Node Framework ใช้สำหรับเขียน Backend โดยมีข้อดีคือ การที่โครงสร้างของมันเหมาะกับการ Scale และยังมี Dependency Injection หรือที่เรียกกันแบบว่า DI เป็นเทคนิคการเขียนโค้ดที่จะช่วยให้โค้ดมีความยืดหยุ่นมากขึ้น ไม่ผูกกับคลาสด้วยกันจนเกินไป

## 4. Security : Authentication

- › มีการจำกัดการเข้าถึงข้อมูลต่างๆ เช่น ฝั่งผู้ใช้งาน ฝั่งแอดมิน
- › การระบุตัวตนและยืนยันตัวตน ใช้ในการตรวจสอบสิทธิ์เข้าใช้บริการ
- › มีการใช้ Password encryption



## 5. Usability : UxUi

> มีการทำ Ux/Ui Design เพื่อให้เว็บไซต์มีการใช้งานที่ง่ายขึ้น  
และมีความสวยงาม

# EXAMPLE QUALITY ATTRIBUTE SCENARIO (1)

A  
V  
A  
I  
L  
A  
B  
I  
L  
I  
T  
Y

Portion of Scenario	Possible values
Source	ผู้ใช้งาน
Stimulus	ต้องการกดไลก์
Artifact	ไลก์ของกระทู้
Environment	Normal mode
Response	ไลก์ถูกอัปเดตลงฐานข้อมูลทันที
Response Measure	กระทู้มีการจัดเรียงใหม่ในเทรนด์กระทู้ทันที

## EXAMPLE QUALITY ATTRIBUTE SCENARIO (2)

P  
E  
R  
F  
O  
R  
M  
A  
N  
C  
E

Portion of Scenario	Possible values
Source	ผู้ใช้งาน
Stimulus	ต้องการตั้งกระดุมโดยแบบรูปหรือวิดีโอที่มีขนาดความ จุมากหรือมีจำนวนที่เยอะ
Artifact	ระบบ
Environment	Normal mode
Response	ระบบทำการสร้างกระดุม
Response Measure	ระยะเวลาในการอัปโหลดกระดุม

# EXAMPLE QUALITY ATTRIBUTE SCENARIO (3)

M  
O  
D  
I  
F  
I  
A  
B  
I  
L  
I  
T  
Y

Portion of Scenario	Possible values
Source	ผู้พัฒนา
Stimulus	ต้องการเปลี่ยน UI
Artifact	UI
Environment	runtime, compile time
Response	เปลี่ยนโดยไม่มีผลกระทบกับฟังก์ชันอื่นๆ, ตรวจสอบการเปลี่ยน, การนำไปใช้งาน
Response Measure	ระยะเวลาในการเปลี่ยน



**THANK YOU  
FOR ATTENTION**

