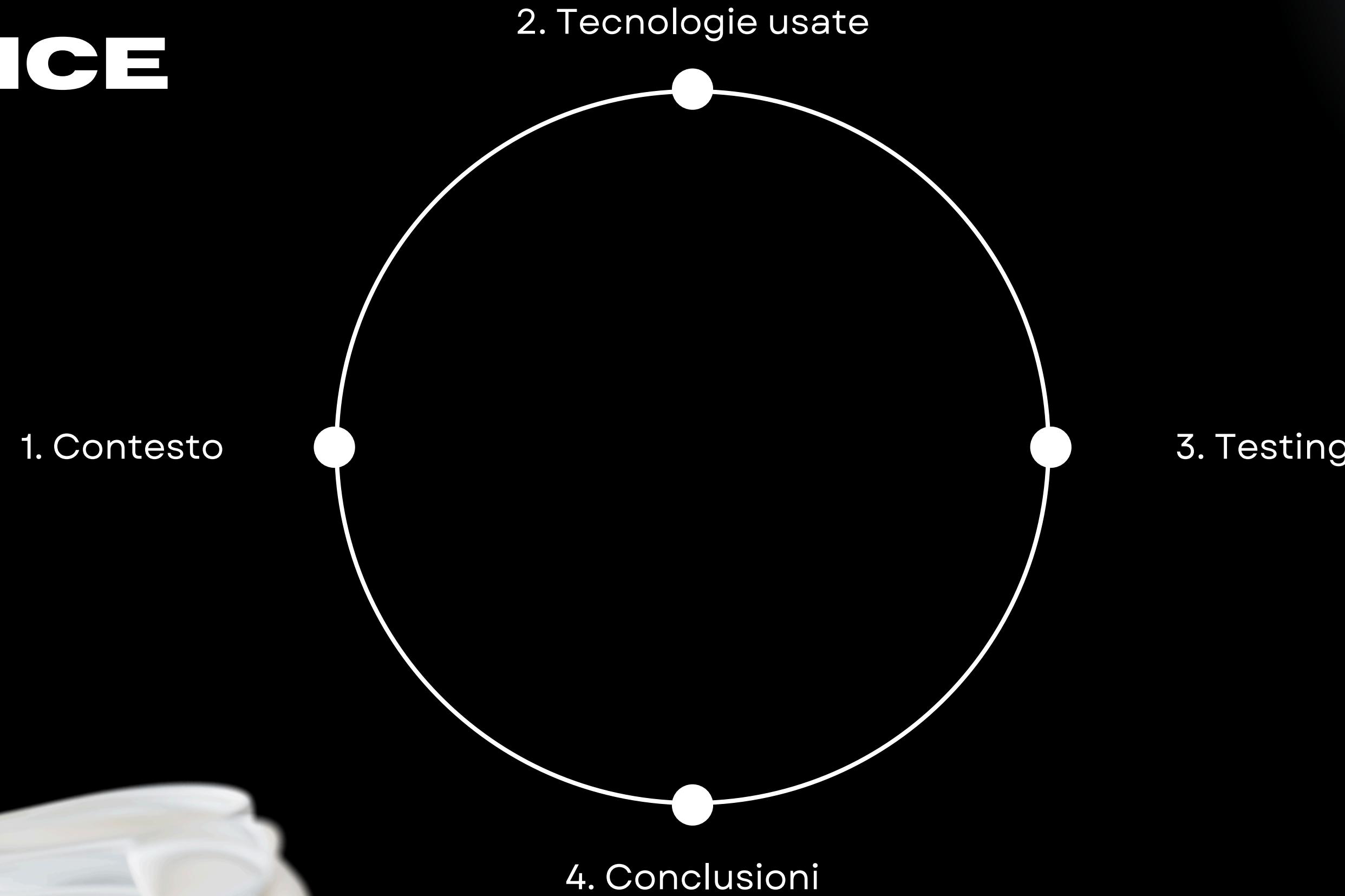


PROCESS MANAGER

PROGETTO DI TESI

Mattia Bonetti - Matricola 218448
Alessio Morelli - Matricola 211709

INDICE



REQUISITI

FUNZIONALI

- Gestione annunci
- Gestione ordini
- Gestione modelli
- Gestione processi
- Gestione macchinari
- Gestione clienti

SICUREZZA

- Accesso solo tramite autenticazione
- Password hashing
- Gestione ruoli

USABILITÀ

- Semplicità e intuitività
- Formati familiari
- Approccio WYSIWYG

RUOLI



Manager

Ruolo con la totalità dei privilegi, in quanto responsabile della gestione globale.



Employee

Ruolo prettamente operativo, con accesso limitato alle attività produttive.



Accountant

Ruolo focalizzato sulla gestione amministrativa e contabile, con accesso ai dati economici e ai clienti.

PERMESSI

Sections	Manager				Employee				Accountant			
	C	R	U	D	C	R	U	D	C	R	U	D
Notice Board	✓	✓	✗	✓	✓	✓	✗	✗	✓	✓	✗	✗
Profile	✗	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓	✓
Employee List	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
Gantt	✗	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗
Clients	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓
Orders	✓	✓	✓	✓	✗	✓	✗	✗	✓	✓	✓	✓
Models	✓	✓	✓	✓	✗	✓	✗	✗	✗	✓	✗	✗
Processes	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗
Machinery	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
Inventory	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓

TECNOLOGIE



Backend

Spring MVC and API REST

Spring Data JPA with
Hibernate

Spring Security



Frontend

JavaFX

FlexGanttFX



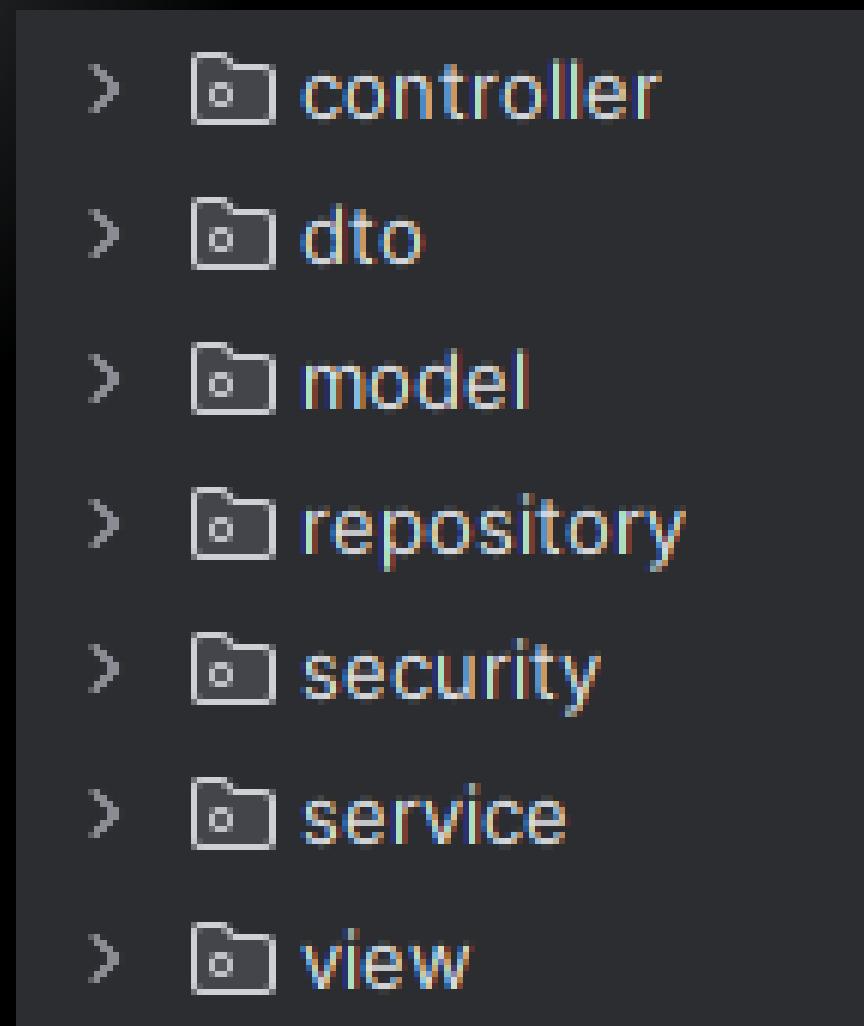
Testing

JUnit

Mockito

TestFX

PATTERN MVC E LAYERED ARCHITECTURE



BACKEND

Controller
Richieste HTTP

Model
Entità JPA

Service
Logica di business

Repository
Query Hibernate

FRONTEND

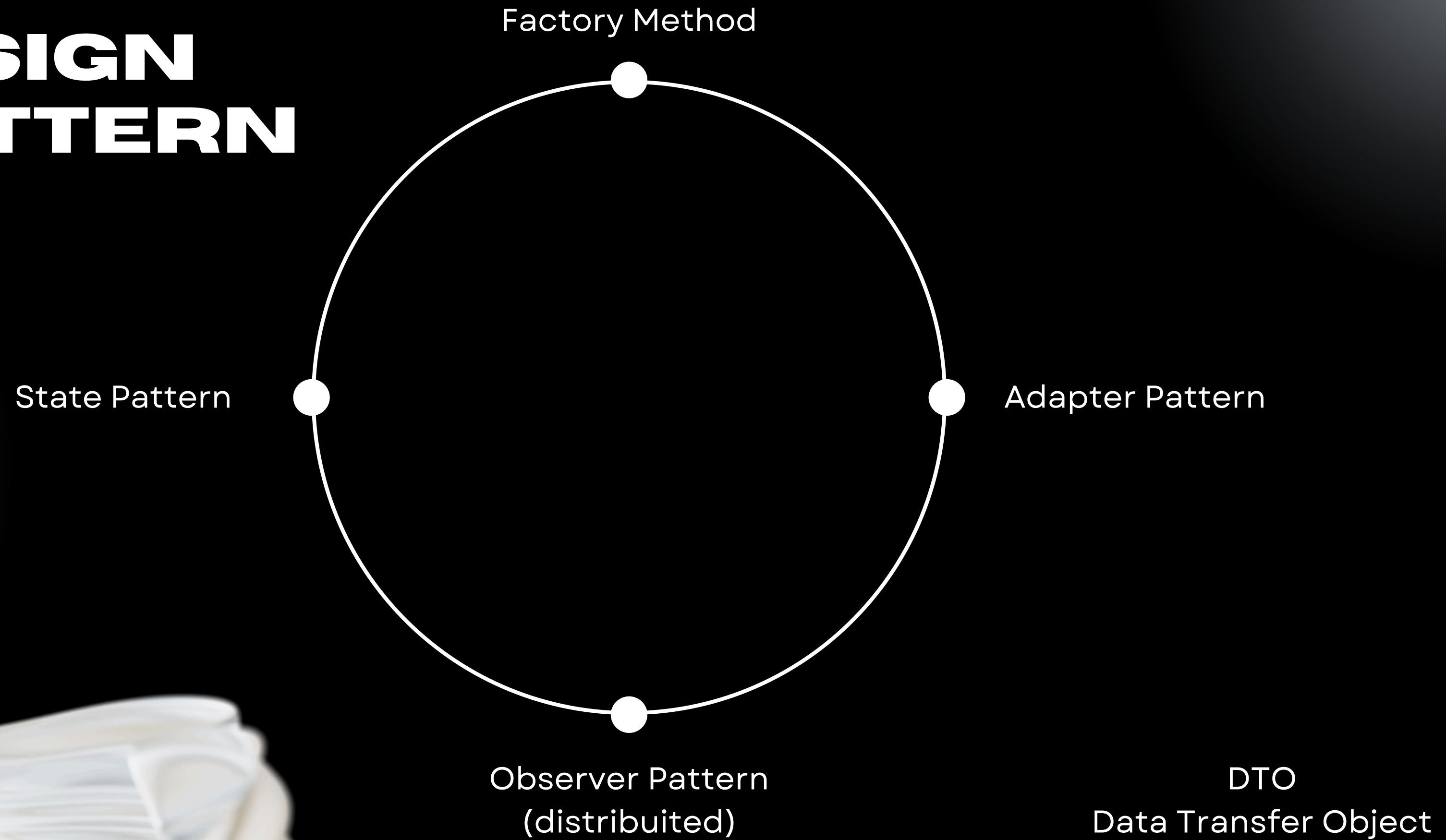
Controller
Gestione view

Model
Entità dominio

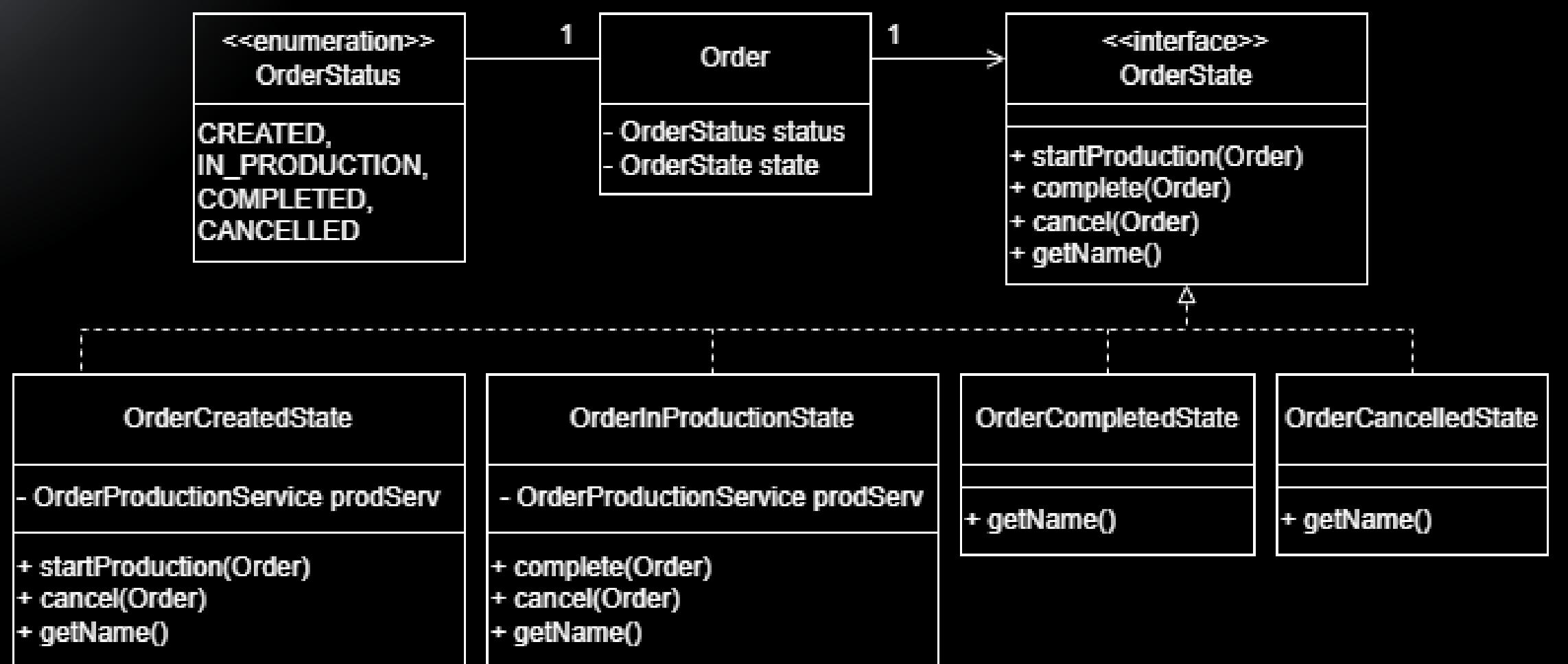
Service
Richieste HTTP

View
View statiche JavaFX

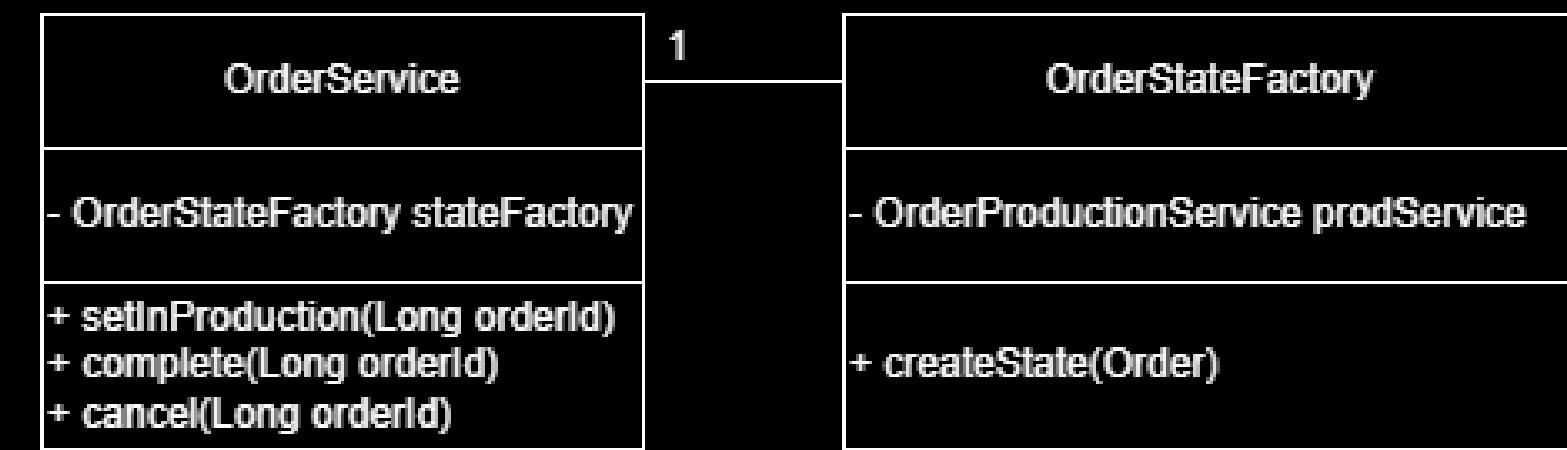
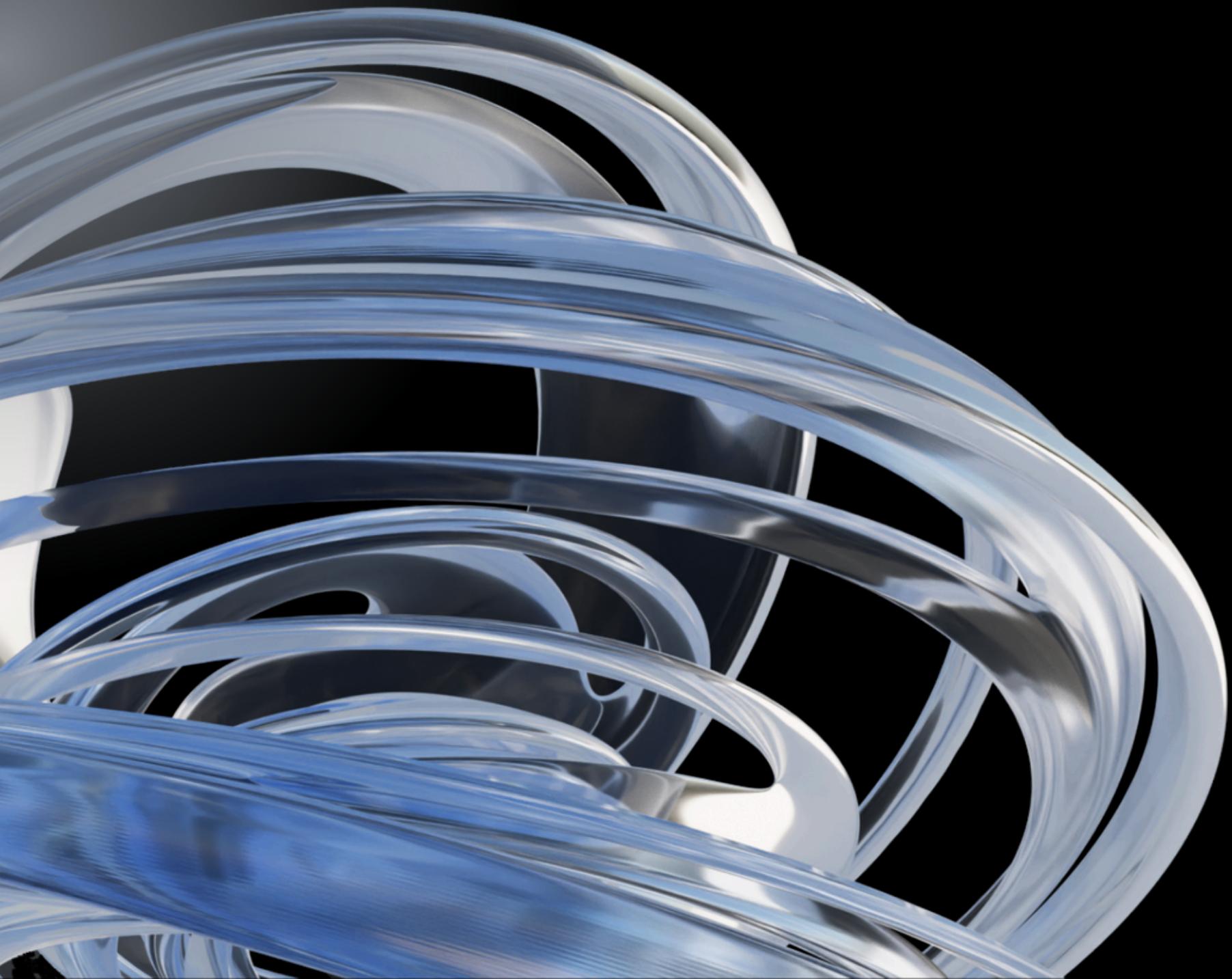
DESIGN PATTERN



STATE PATTERN



FACTORY METHOD

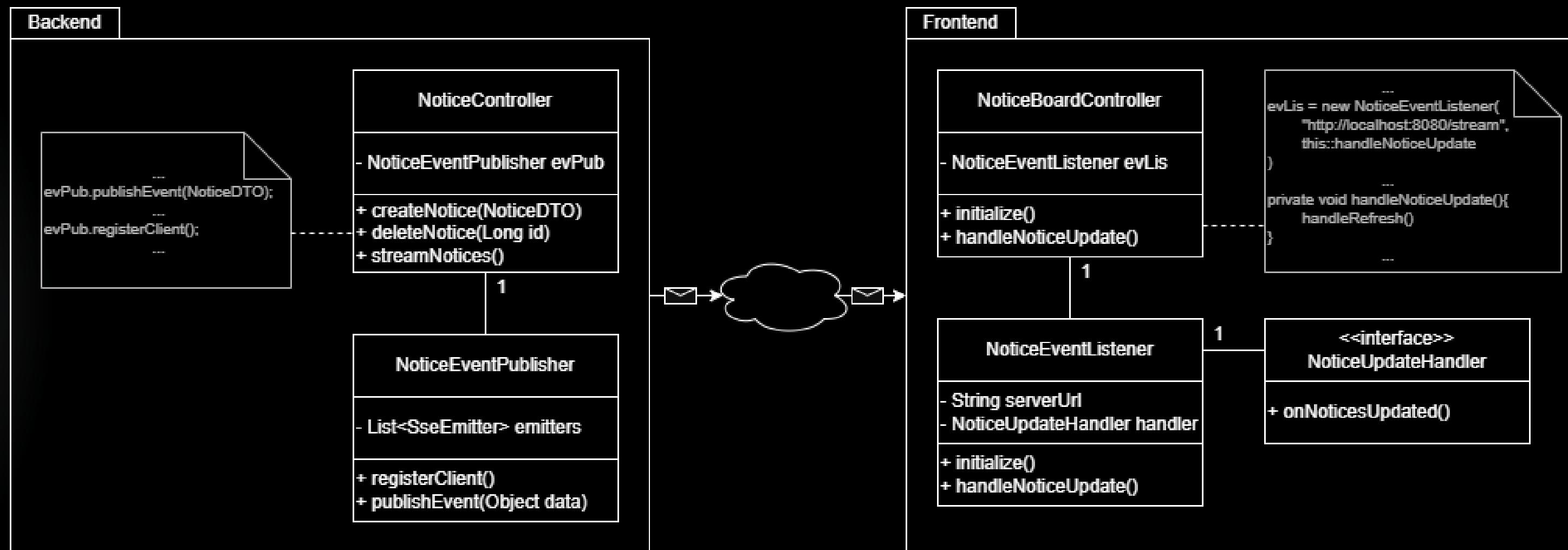


```
return switch(order.getStatus()){  
    case CREATED ->  
        new OrderCreatedState(prodService);  
    case IN_PRODUCTION ->  
        new OrderInProductionState(prodService);  
    case COMPLETED ->  
        new OrderCompletedState();  
    case CANCELLED ->  
        new OrderCancelledState();  
};
```

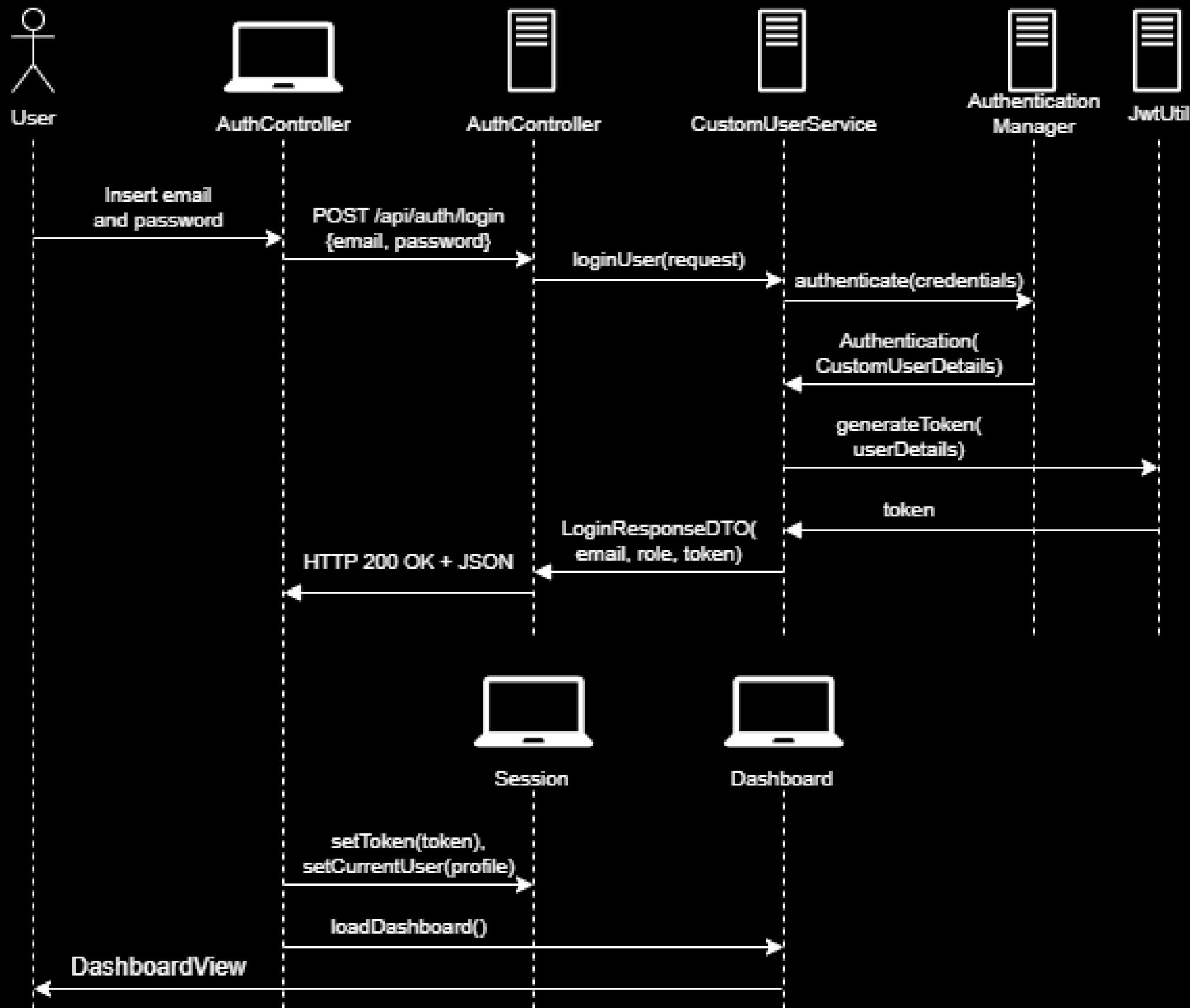
ADAPTER PATTERN



OBSERVER PATTERN

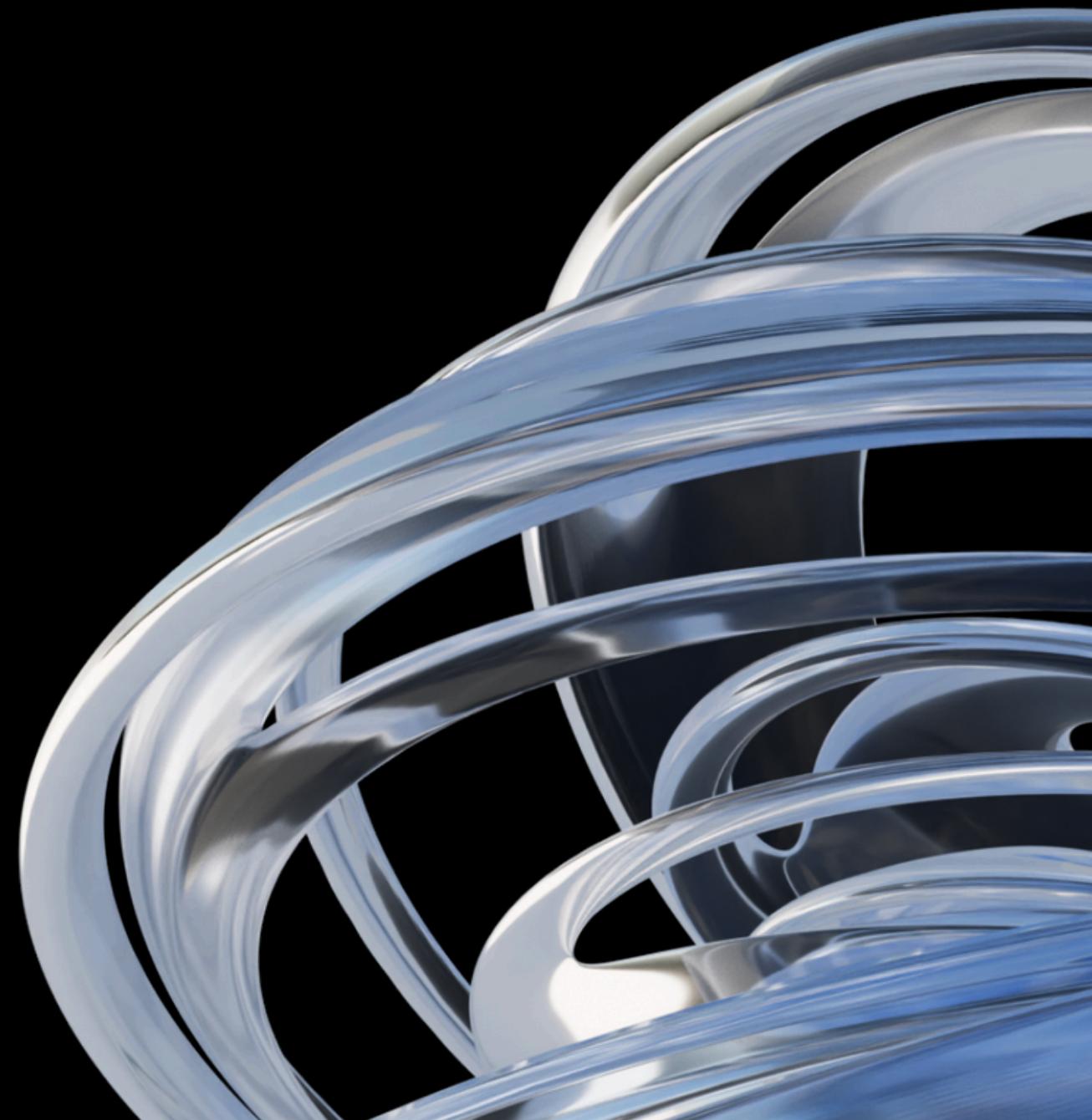


SPRING SECURITY



MULTITHREADING

```
61     void loadModelList() {
62         new Thread(() -> {
63             try {
64                 List<Model> models = modelService.loadAllModels();
65                 Platform.runLater(() -> {
66                     modelTable.getItems().setAll(models);
67                     statusLabel.setText("Loaded " + models.size() + " models");
68                     statusLabel.setStyle("-fx-text-fill: green;");
69                 });
70             } catch (Exception e) {
71                 e.printStackTrace();
72                 Platform.runLater(() -> {
73                     statusLabel.setText("Error while loading");
74                     statusLabel.setStyle("-fx-text-fill: red;");
75                 });
76             }
77         }).start();
78     }
```



Project Coverage Report					
Element ^	Class, %	Method, %	Line, %	Branch, %	
BACKEND	server_group	85% (49/57)	79% (308/387)	83% (710/855)	73% (89/121)
	controller	100% (12/12)	96% (60/62)	97% (191/196)	100% (36/36)
	dto	100% (11/11)	69% (76/110)	68% (106/155)	44% (4/9)
	model	70% (12/17)	73% (101/138)	65% (106/162)	0% (0/4)
	repository	100% (0/0)	100% (0/0)	100% (0/0)	100% (0/0)
	security	50% (2/4)	58% (7/12)	39% (20/51)	0% (0/10)
	service	100% (12/12)	100% (64/64)	98% (287/290)	79% (49/62)
	Server	0% (0/1)	0% (0/1)	0% (0/1)	100% (0/0)
Coverage Summary					
Element ^	Class, %	Method, %	Line, %	Branch, %	
FRONTEND	client_group	86% (50/58)	76% (415/540)	73% (2022/2738)	49% (236/478)
	controller	94% (17/18)	64% (170/264)	64% (1154/1789)	34% (98/284)
	dto	87% (7/8)	93% (80/86)	93% (92/98)	100% (0/0)
	model	71% (10/14)	79% (87/109)	70% (107/151)	20% (4/20)
	service	94% (16/17)	98% (78/79)	96% (669/693)	77% (134/174)
	Main	0% (0/1)	0% (0/2)	0% (0/7)	100% (0/0)
	Media Pesata: 75.38%				



THANK YOU

Mattia Bonetti
Alessio Morelli