



# PATTERN COMMAND

---

ALEXANDRE BOURDARIE

NICOLAS PRÉVÔT

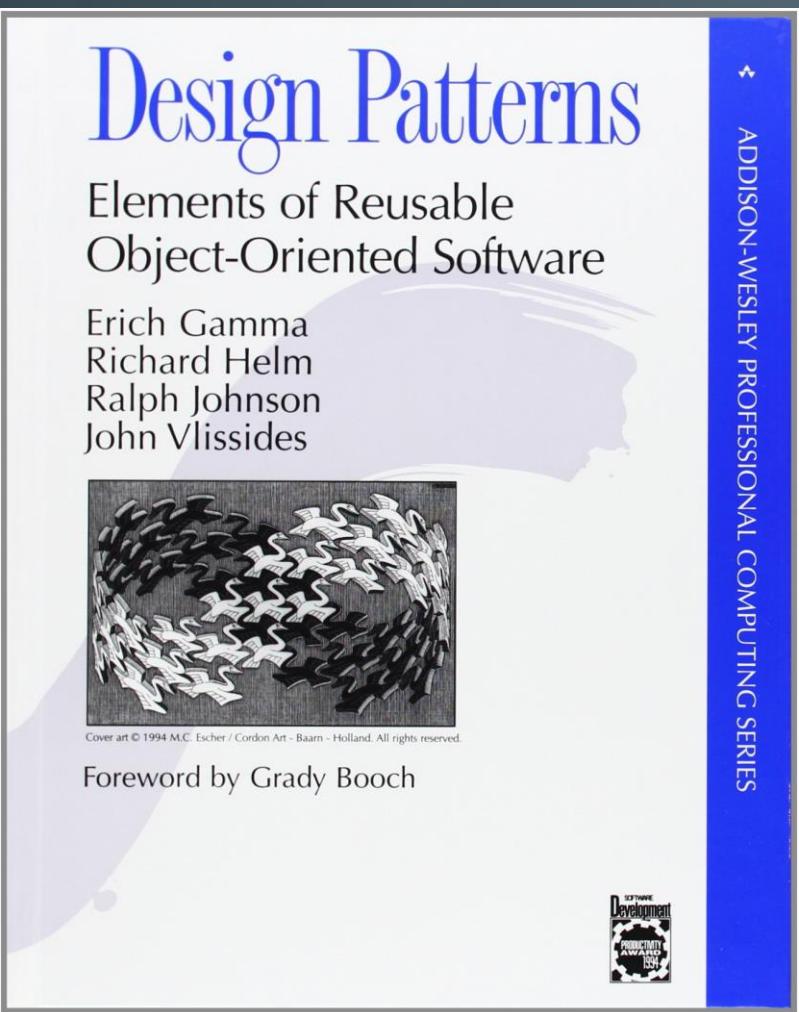
AXEL SALEM

# SOMMAIRE :

- Qu'est ce qu'un Design Pattern ?
- Le Pattern Command
  - Présentation du Pattern
  - Démonstration avec un exemple
  - Principes SOLID
  - Avantages et Inconvénients
- Live Coding
- Rapprochement avec d'autres Patterns
- QCM



# QU'EST-CE QU'UN DESIGN PATTERN ?



Publié en 1994

## Gang of Four (GoF)

- Erich Gamma
- Richard Helm
- Ralph Johnson
- John Vlissides



BONNE PRATIQUE EN  
RÉPONSE À UN PROBLÈME DE  
CONCEPTION RÉCURRENT



NE CONTIENT NI CODE, NI  
ALGORITHME



# LE PATTERN COMMAND

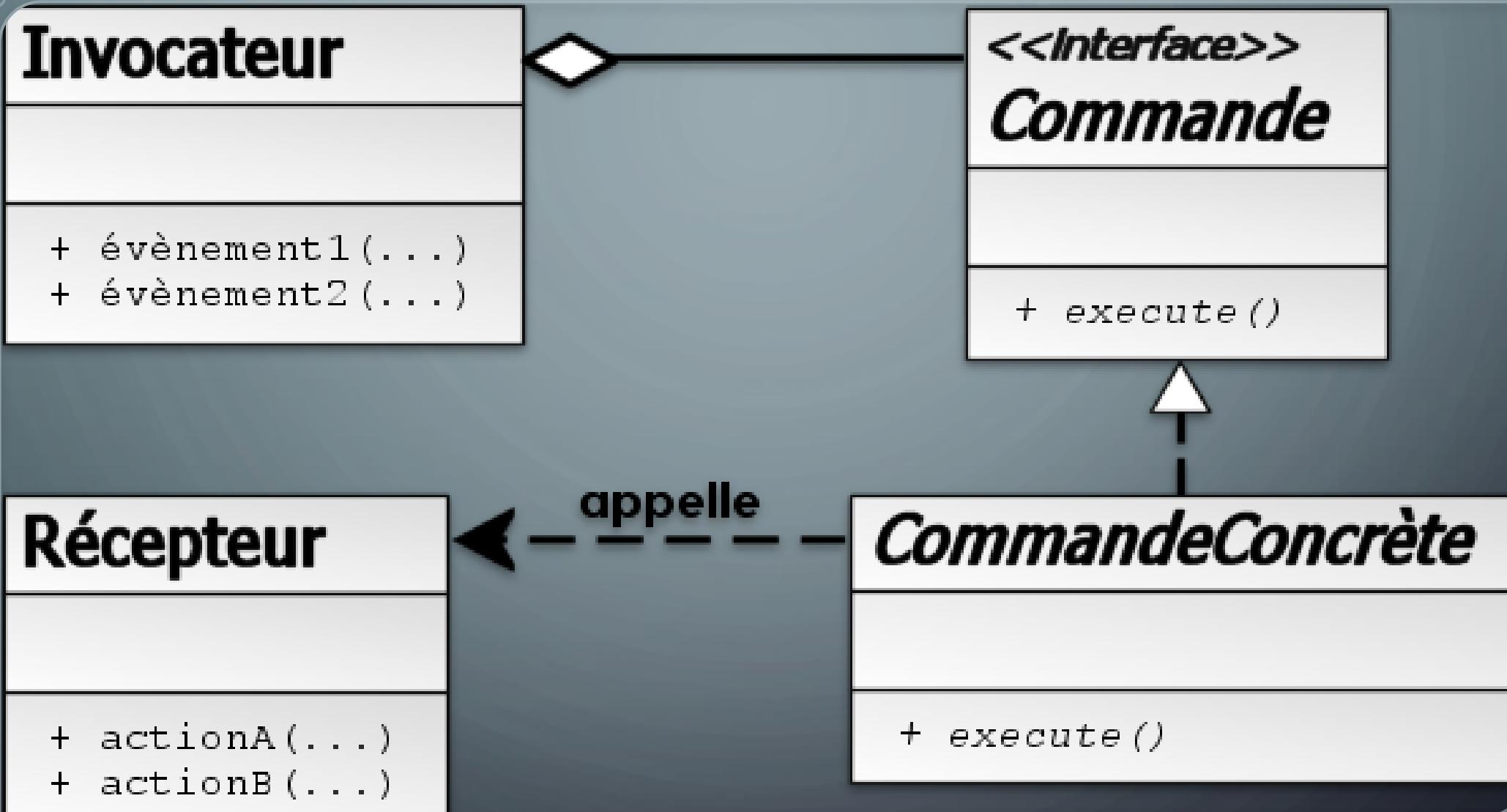


# PRÉSENTATION DU PATTERN

## L'intention du patron "Commande"

Transformer une action en un objet autonome qui contient les détails de cette action

Permet entre autres de paramétriser des méthodes et de planifier leur exécution.



# POURQUOI UTILISER CE PATRON ?

Le paramétrage

L'ordonnancement

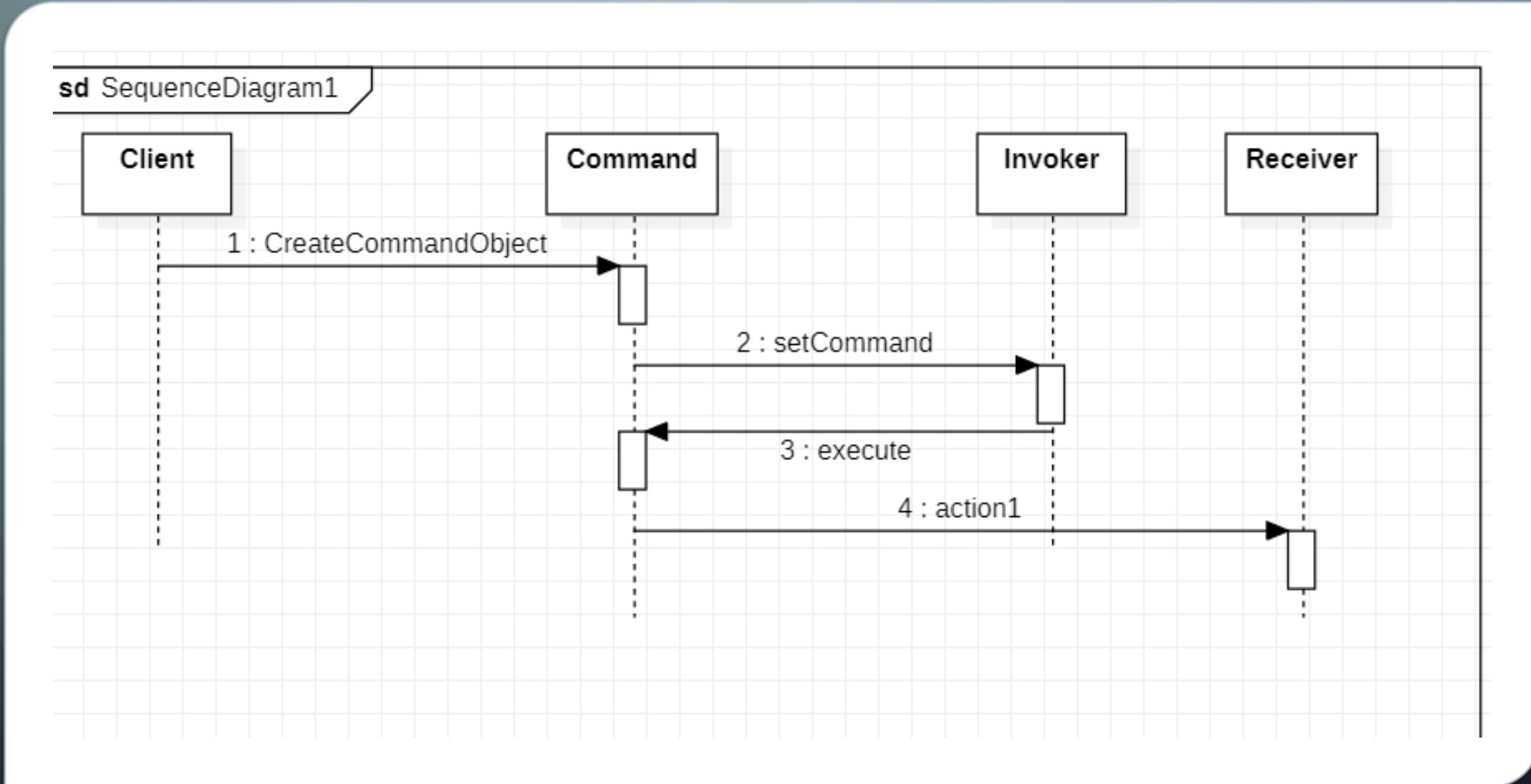
Organisation

# DÉMONSTRATION AVEC UN EXEMPLE



# IDENTIFIONS LES CLASSES QUI PARTICIPENT...

Serveuse	Préparer la commande	Commande x prête	Commande	Clients	Prendre la commande
Command	execute()	Action n°x	invoker	Receiver	SetCommand()





# PRINCIPES SOLID ET PATTERN COMMAND

---

# PRINCIPES SOLID

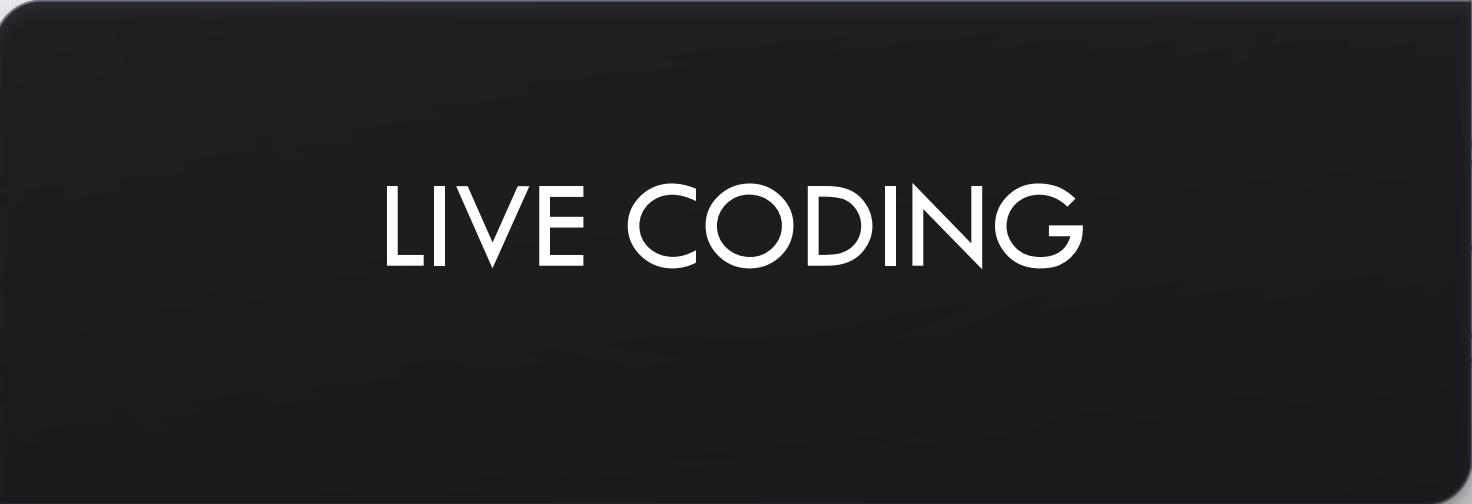
- SRP (Single Responsibility Principle) : Une classe doit avoir une et une seule responsabilité
- OCP (Open-Closed Principle) : Une classe doit être ouverte aux extensions, mais fermée aux modifications.
- LSP (Liskov's Substitution Principle) :
  - Les sous-types doivent pouvoir être substitués à leurs types de base.
- ISP (Interface Segregation Principle) :
  - Un client ne devrait jamais être forcé de dépendre d'une interface qu'il n'utilise pas.
- DIP (Dependency Inversion Principle) :
  - 1. Les modules de haut niveau ne doivent pas dépendre de modules de plus bas niveau. Les deux doivent dépendre d'abstractions.
  - 2. Les abstractions ne doivent pas dépendre des détails. Les détails doivent dépendre des abstractions.



# AVANTAGES ET INCONVÉNIENTS DU PATTERN COMMAND

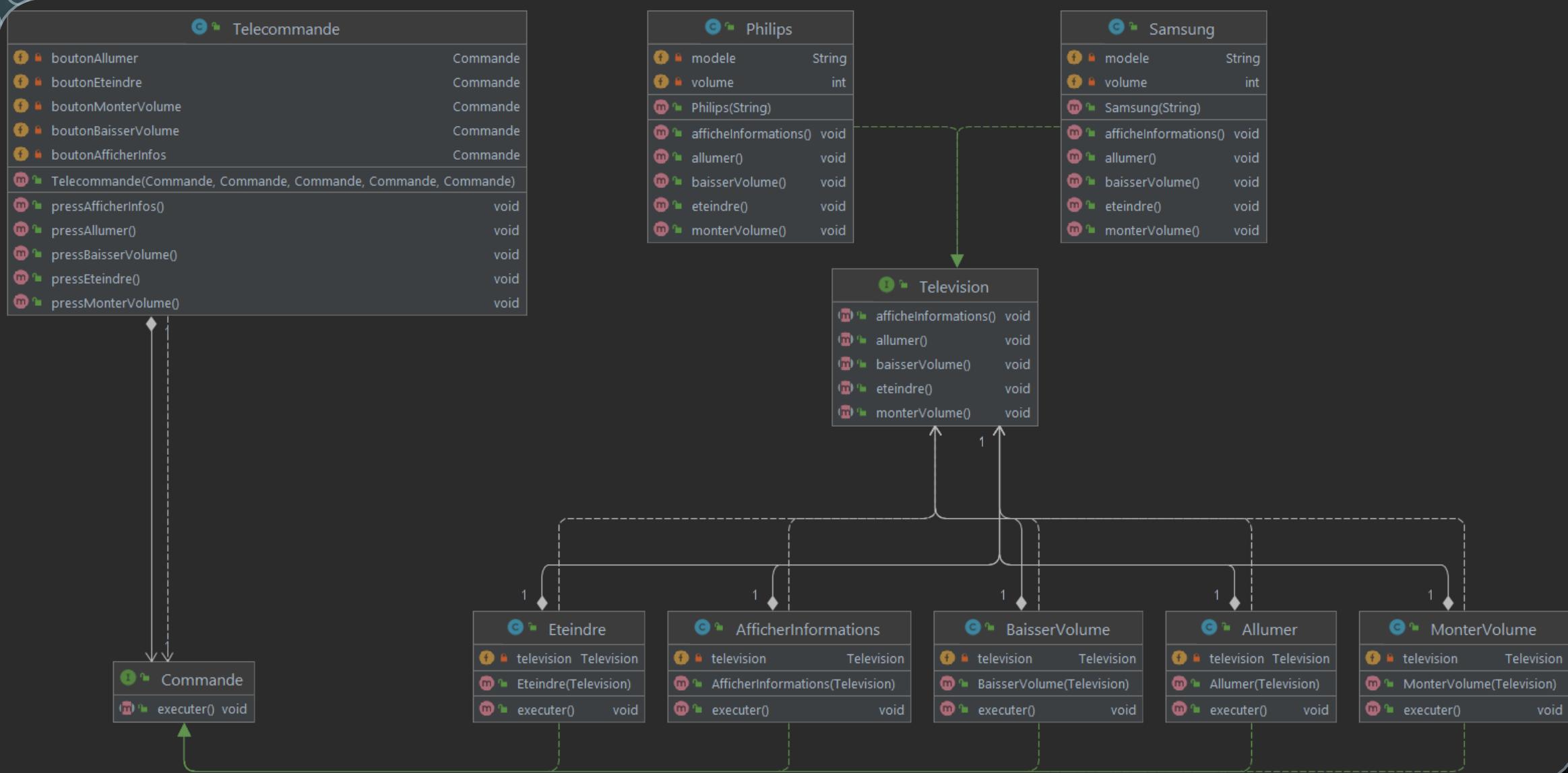
---

- + Assembler des commandes simples en une plus complexe
- + Différer l'exécution des traitements
- Code plus complexe, car on crée une nouvelle couche entre demandeurs et récepteurs



LIVE CODING

# DIAGRAMME DE CLASSE





# LA VIDÉO

[HTTPS://YOUTU.BE/WDZPRQTDG4Q](https://youtu.be/wdzprqtdg4q)



## RAPPROCHEMENT AVEC D'AUTRES PATTERN

**Observateur**

**Médiateur**

**Visiteur**



QCM

<https://tech.io/playgrounds/28bb22905614e2d8d6047ac525136f8d63495/quiz-sur-le-pattern-command>