LE PATRON OBSERVATEUR

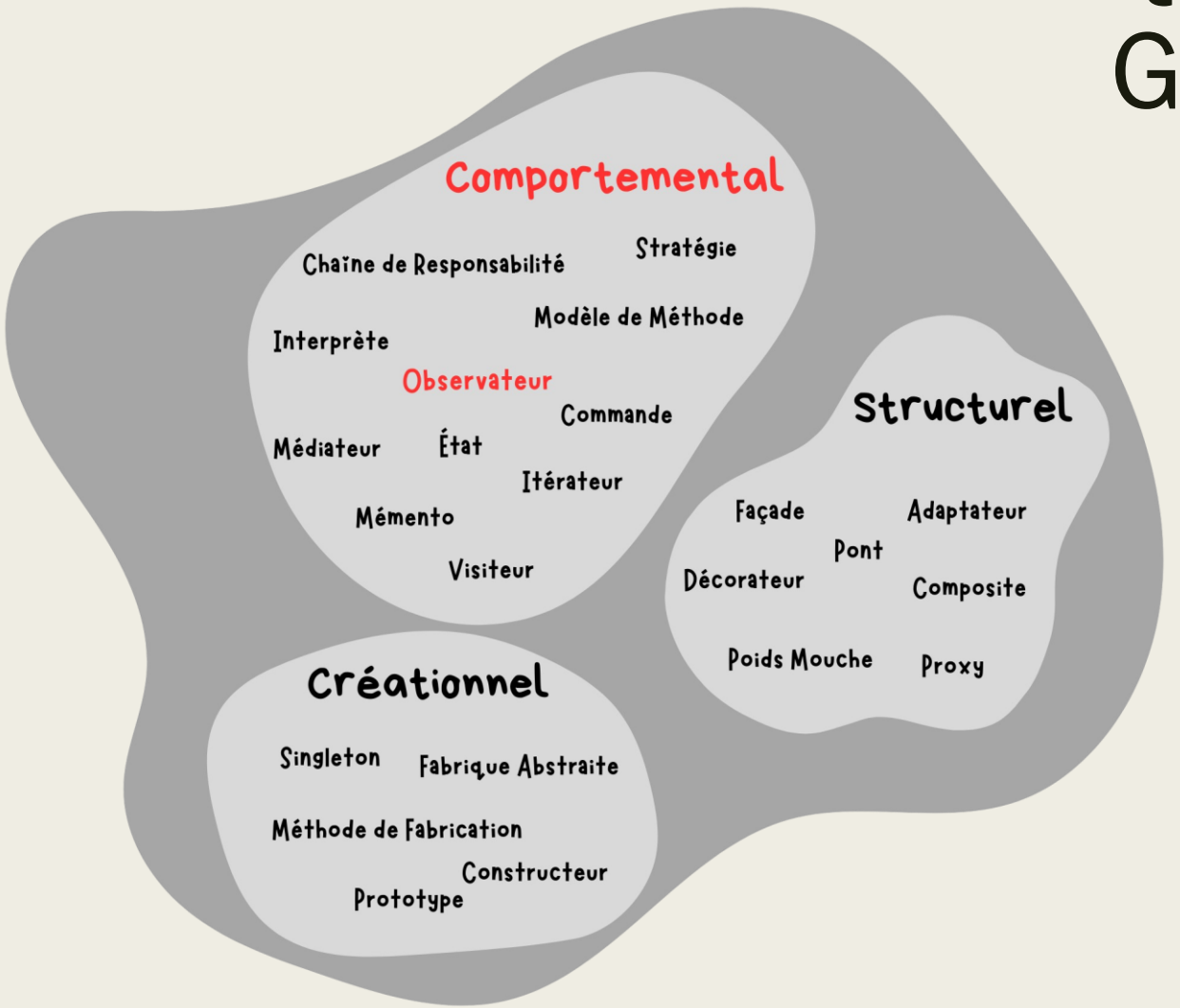
Issue de la famille des patrons comportementaux
du « Gang of Four »

Présenté par Florian CLAUX, Maxence VINCENT, Joachim FEVRE, Gaëtan CHARPENTIER

Sommaire

- Qu’est que le Gang of Four ?
- Mise en situation
  - *Enoncé de l’exemple*
  - *Modélisation et identification des problèmes*
  - *Modélisation avec le pattern Observateur*
- Structure Générale du Pattern Observateur
- Lien avec SOLID
- Le rôle du pattern dans l’architecture MVC
- Live coding d’une station météo
- Exemple d'implémentation alternative : Modèle PULL
- Conclusion
- Bibliographie/webographie
- QCM

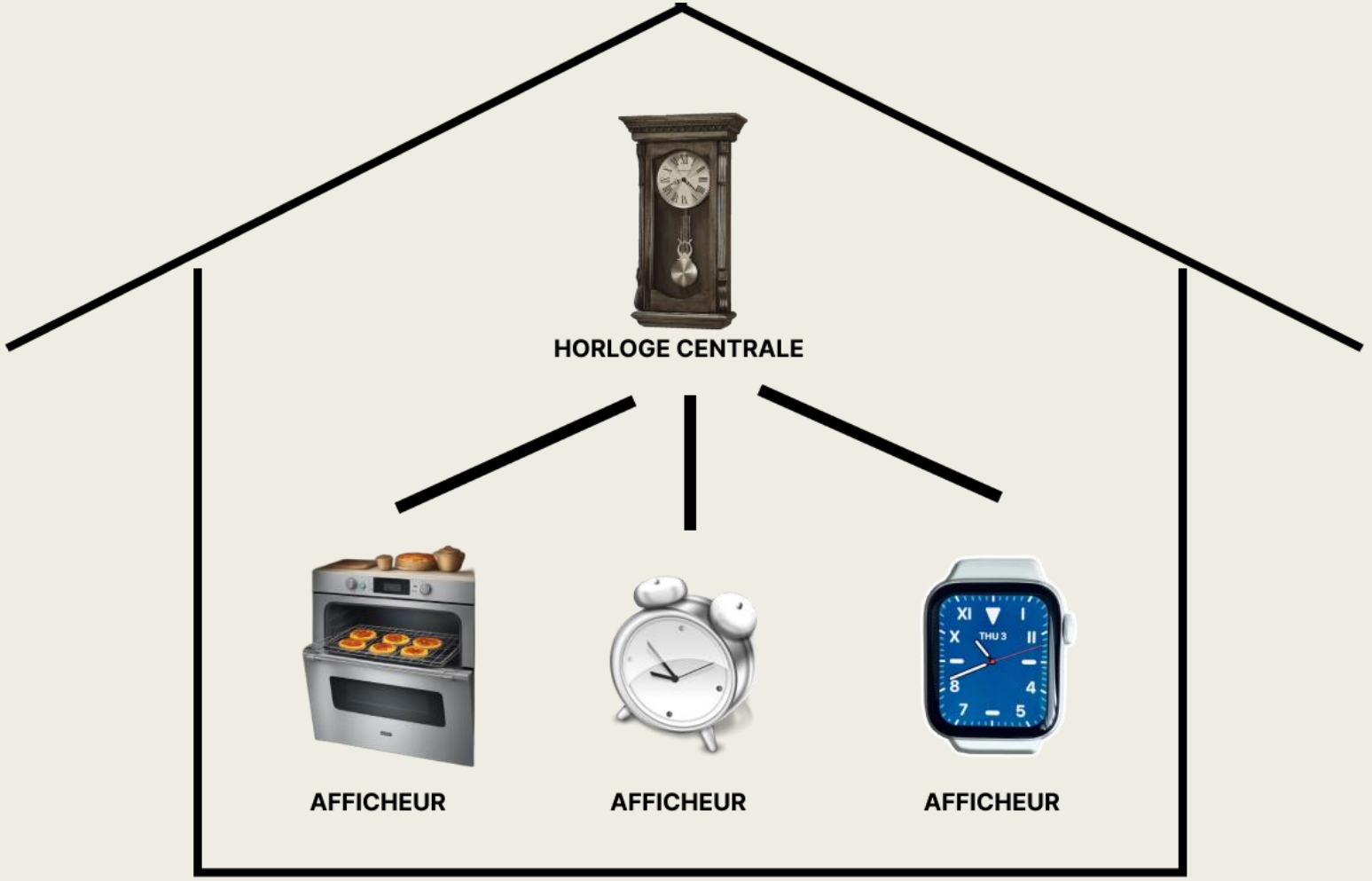
Qu’est-ce que le Gang of Four ?



- **23** patterns
- Écrit par Erich Gamma, Richard Helm, Ralph Johnson et John Vlissides en **1994**.
- Les patrons **comportementaux** définissent la manière dont les objets **interagissent** et **communiquent** pour améliorer la flexibilité et l'organisation du système.

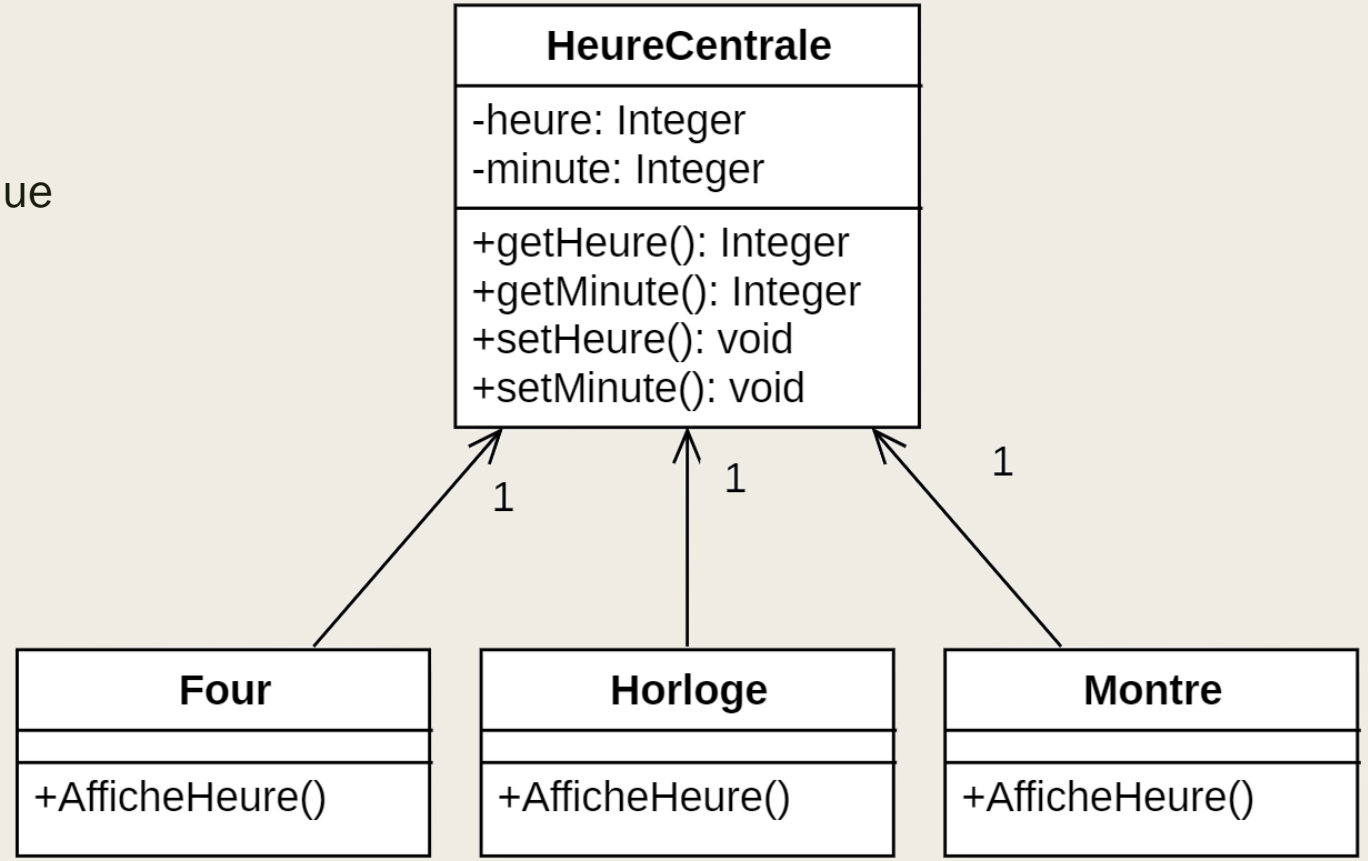
Cas pratique

- Dans ce contexte, on peut observer deux rôles principaux :
  1. *L'horloge centrale*
  2. *Les afficheurs*

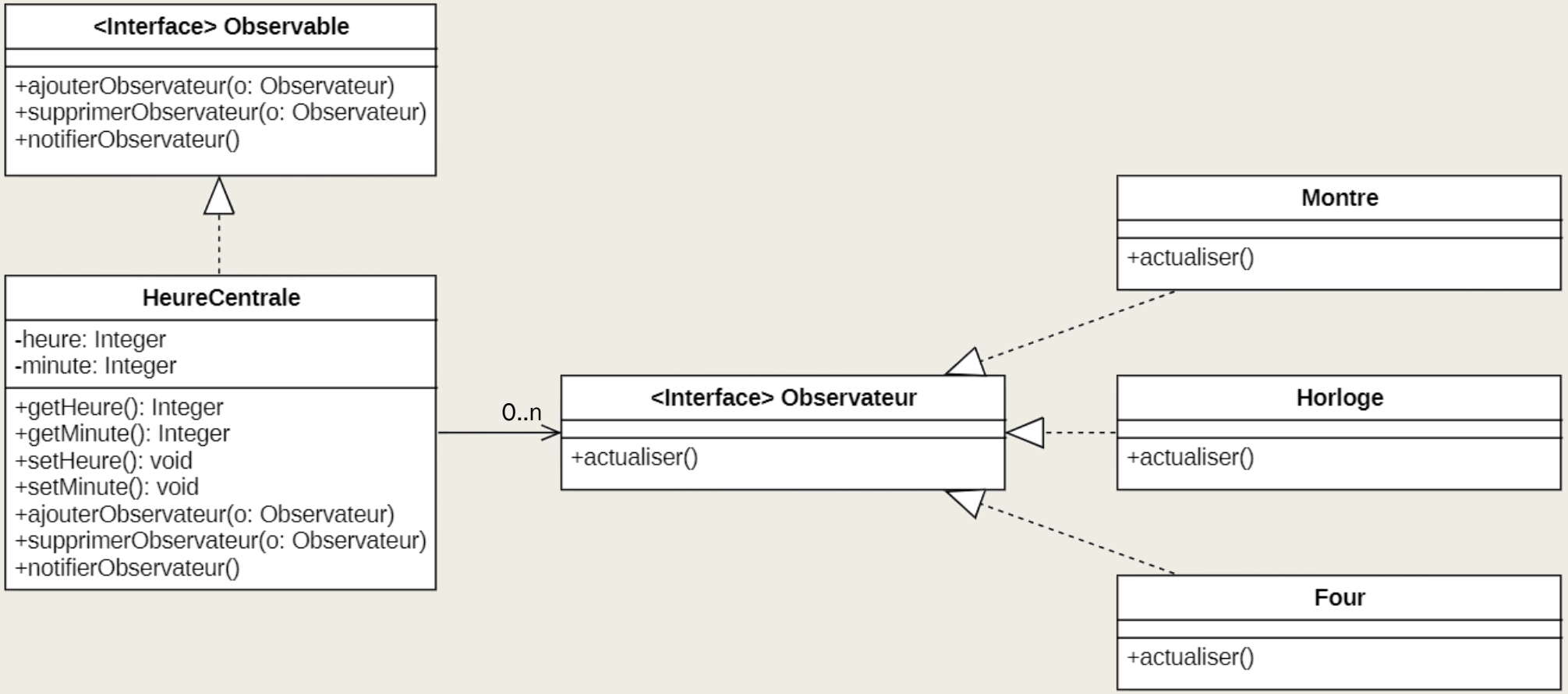


Modélisation de l’exemple et identification des problèmes

- Absence de Notification Automatique
- Couplage fort
- Problème de Scalabilité
- Inefficacité



Modélisation avec le Pattern Observateur



Structure Générale du Pattern Observateur

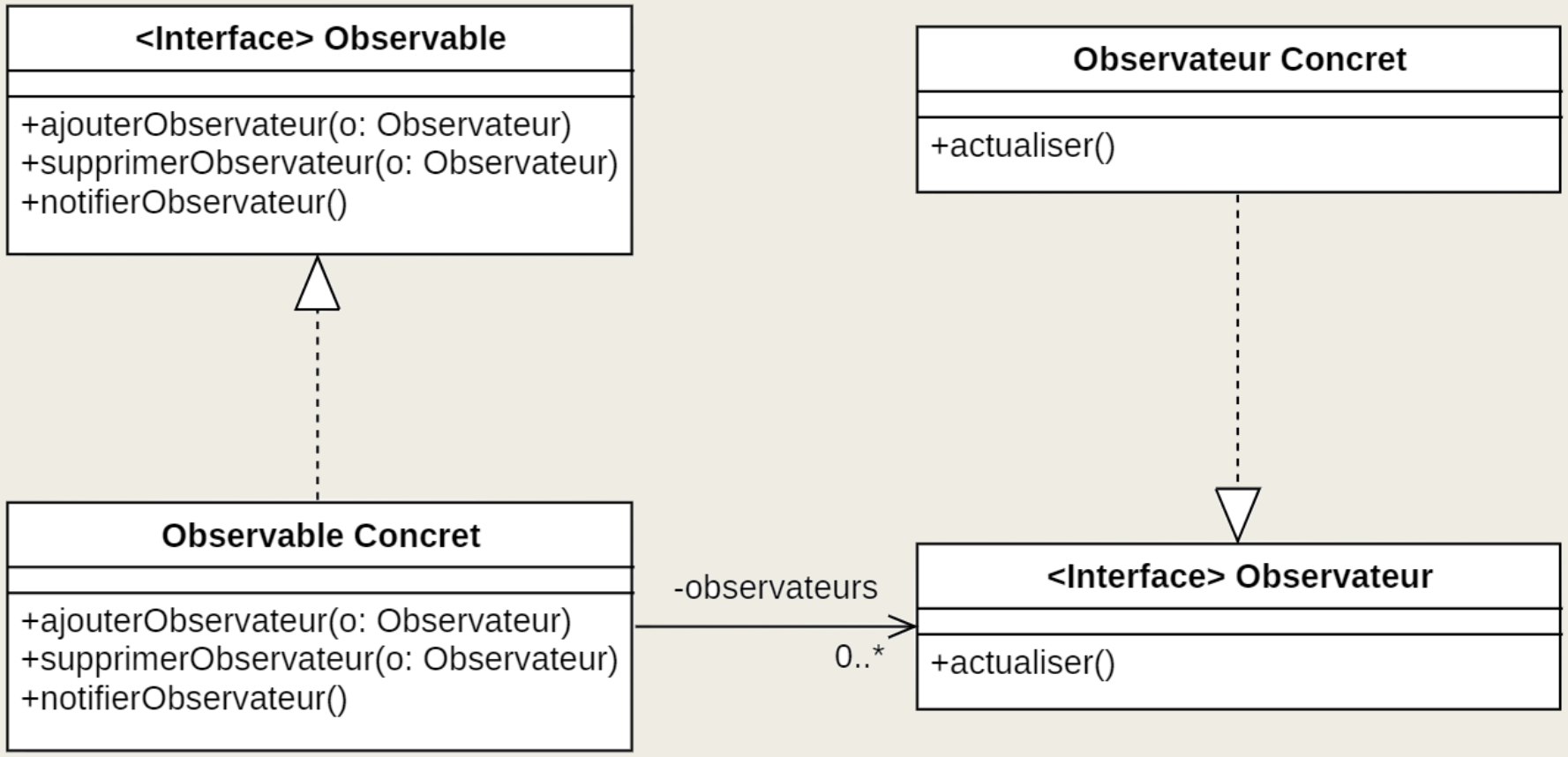
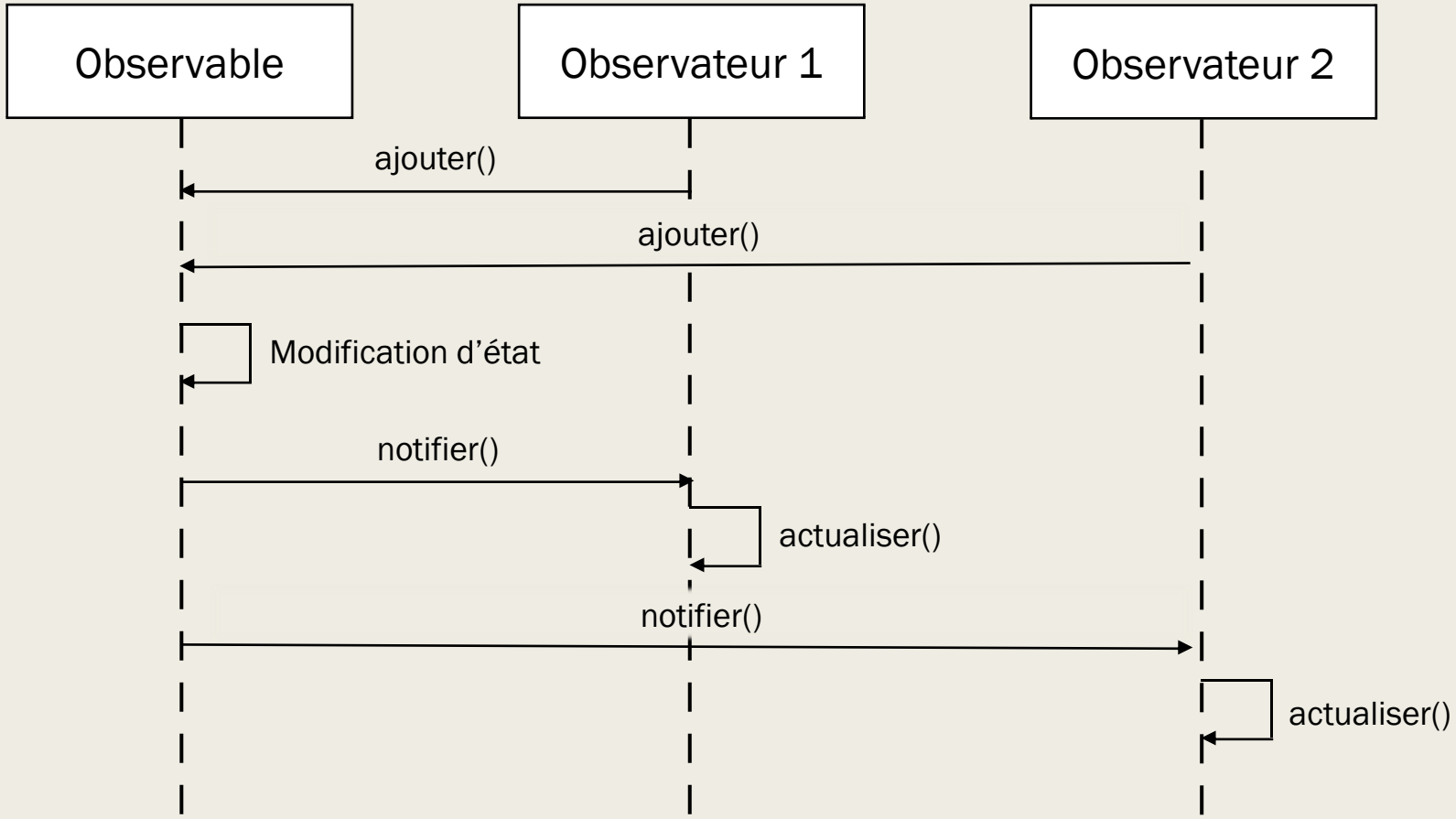


Diagramme de séquence



Lien avec SOLID

Principes respecter :

- SRP :
  - l'observable gère son état et notifie les changements, tandis que chaque observateur réagit aux notifications
- OCP :
  - Possibilité d'ajouter des observateurs sans modifier le code.
- ISP :
  - Interfaces simples, dédiées à une seule tâche.

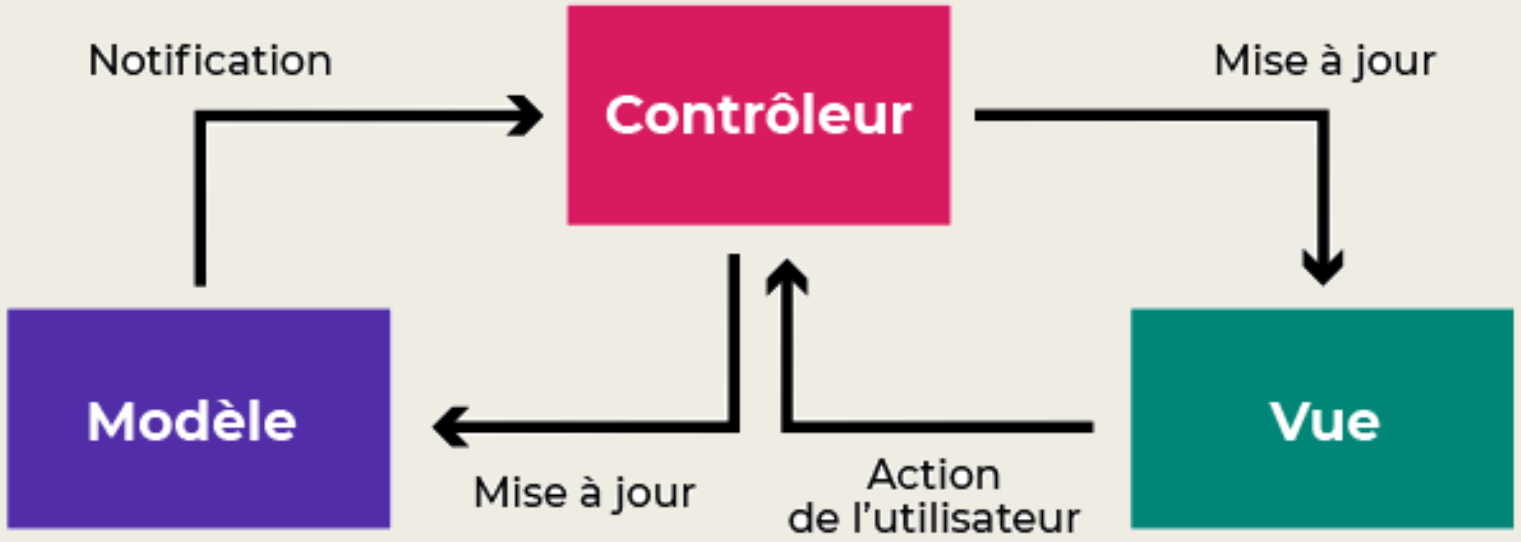
Principes non respecter :

- LSP :
  - difficile à assurer pour les observateurs substituables.
- DIP :
  - Partiellement respecté, car dans certains cas, les observateurs dépendent directement de l'observable

Limites du pattern Observateur

- Les souscripteurs sont **avertis** dans un **ordre aléatoire** 
- Risques de **Fuites Mémoire** 
- Propagation de **Mises à Jour** Indésirables 
- Difficulté de **Débogage** 

Rôle du pattern dans l'architecture MVC

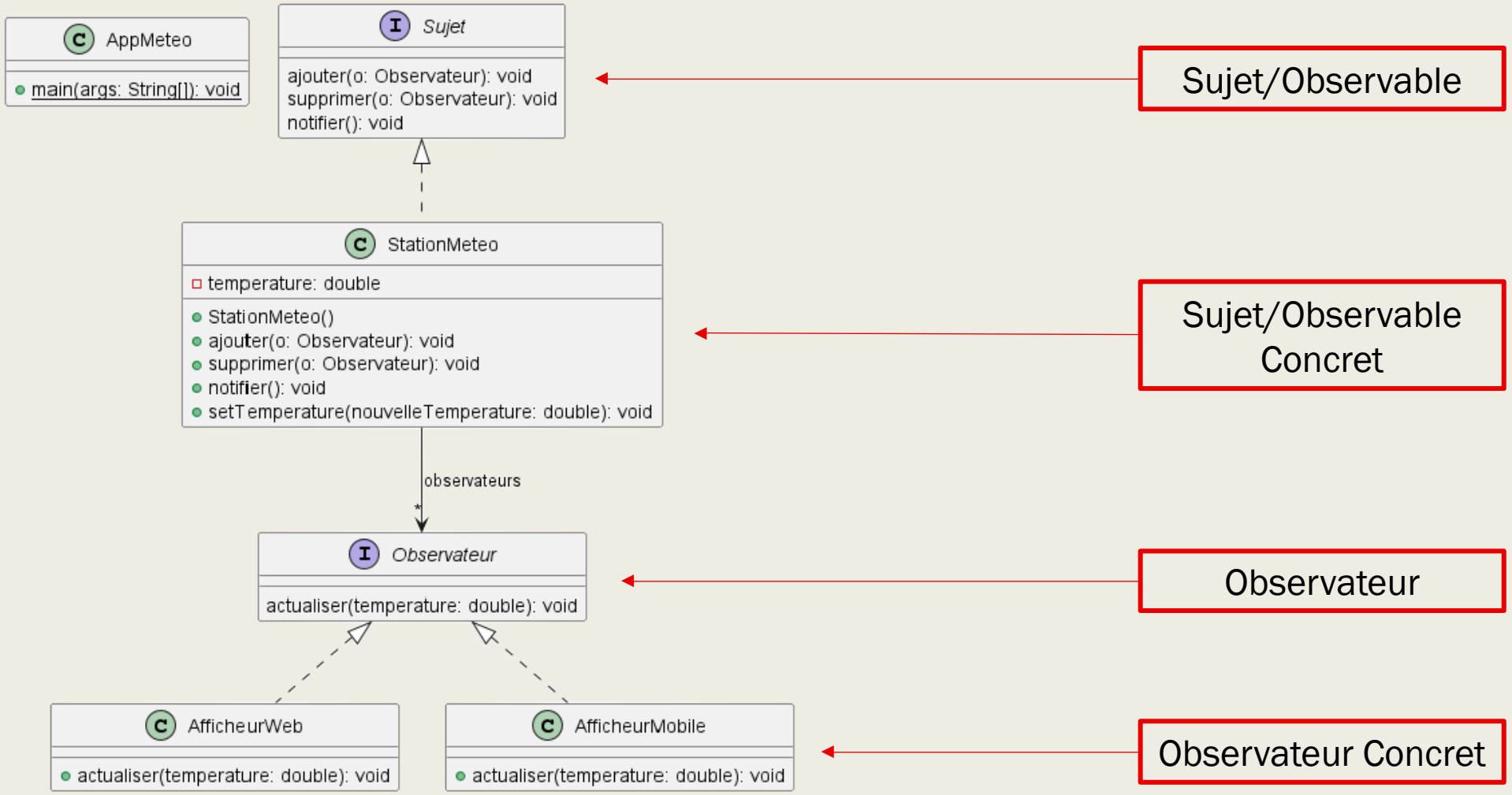


Live coding d’une station météo

- https://www.youtube.com/watch?v=6wAqwOar7Z4

- Dans cette session, nous allons explorer l'implémentation du pattern Observateur à travers une **station météo**. Ce pattern permet à la station de notifier automatiquement plusieurs affichages (comme un **afficheur web** et un **afficheur mobile**) lorsqu’une nouvelle température est enregistrée. Nous verrons comment la **StationMeteo** agit comme un "**sujet**" qui gère une **liste d’observateurs**, leur envoie des mises à jour, et assure ainsi une synchronisation efficace des données en temps réel.

Diagramme de l'application Météo



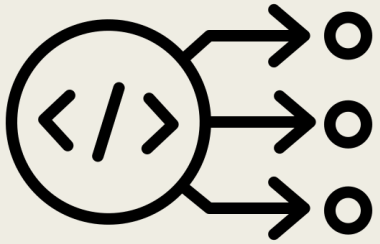
Exemple d'implémentation alternative : Modèle PULL

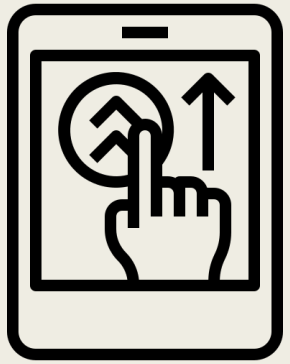
```java
public class StationMeteo implements Sujet {

    private List<Observateur> observateurs;
    private double temperature;

    @Override
    public void ajouterObservateur(Observateur o) {
        observateurs.add(o);
        o.setSujet(this);
    }
    @Override
    public void notifierObservateurs() {
    for (Observateur observateur : observateurs) {
            observateur.actualiser();
        }
    }
    public void setTemperature(double nouvelleTemperature) {
        this.temperature = nouvelleTemperature;
        notifierObservateurs();
    }
    public double getTemperature() {
        return this.temperature;
    }
}
```

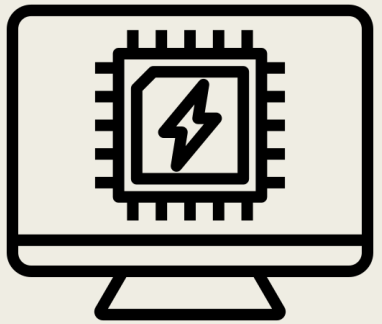
```java
public class AfficheurWeb implements Observateur {

    private StationMeteo stationMeteo;

    public void setSujet(StationMeteo stationMeteo) {
        this.stationMeteo=stationMeteo;
    }
    public void actualiser() {
        System.out.println("App web : " + stationMeteo.getTemperature() + "°C");
    }
}
```

Conclusion

- Couplage faible
- Application dynamique
- Gourmand en ressources







Bibliographie/webographie

- https://www.bob-le-developpeur.com/notions/pattern-observer
- https://openclassrooms.com/fr/courses/7133336-utilisez-des-design-patterns-en-javascript/7478472-ecoutez-vos-objets-avec-l-observer-pattern
- https://stackoverflow.com/questions/50682618/what-solid-principles-does-the-observer-pattern-follow-violate
- https://design-patterns.fr/observateur
- https://www.lipn.univ-paris13.fr/~santini/Patrons_conception/seance4/seance4_cours.pdf
- http://www.goprod.bouhours.net/?page=pattern&pat_id=16
- https://refactoring.guru/fr/design-patterns/observer

QCM (Kahoot)

- https://play.kahoot.it/v2/?quizId=ea7127b5-a4dd-4a28-a4dc-efd91b3c1241