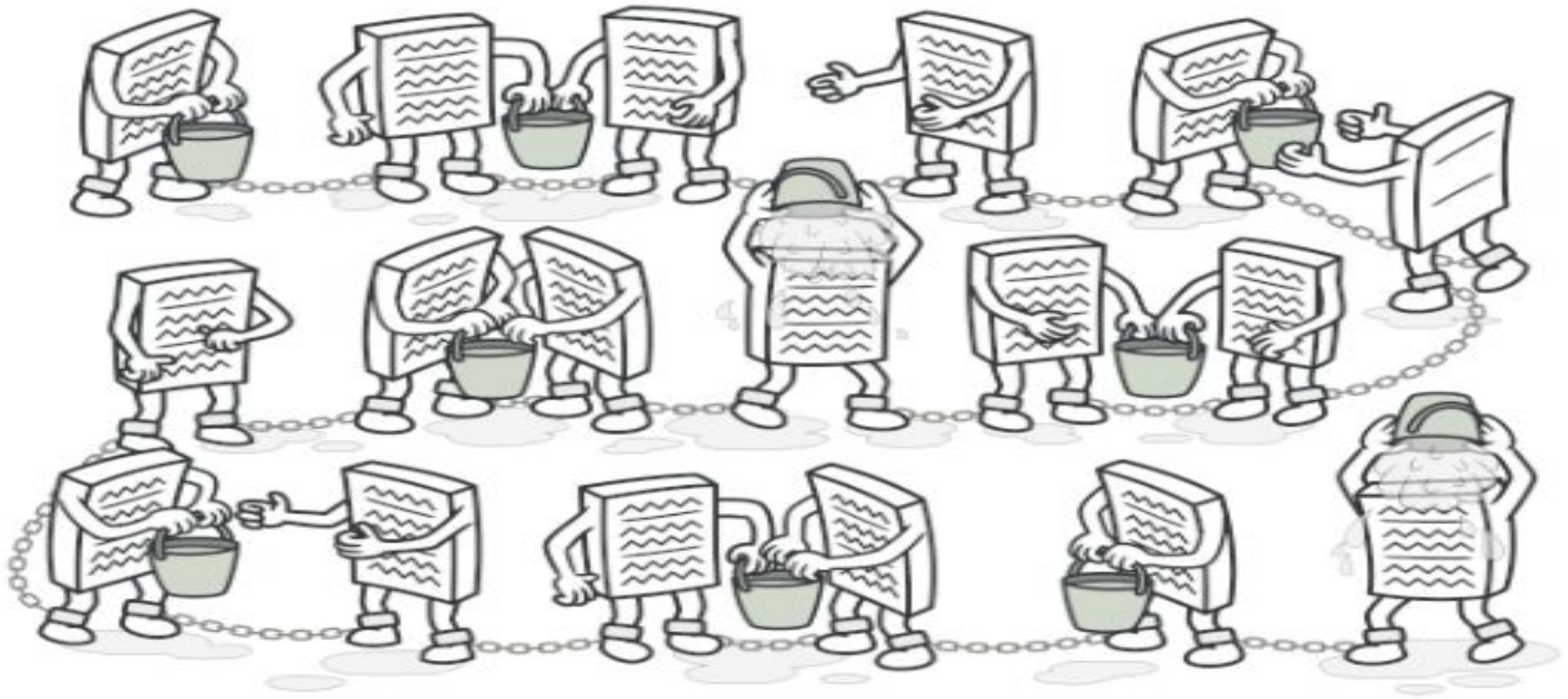
# Chain Of Responsability

Alunos: John Victor Farias de Omena e Willieny Barbosa de Magalhães

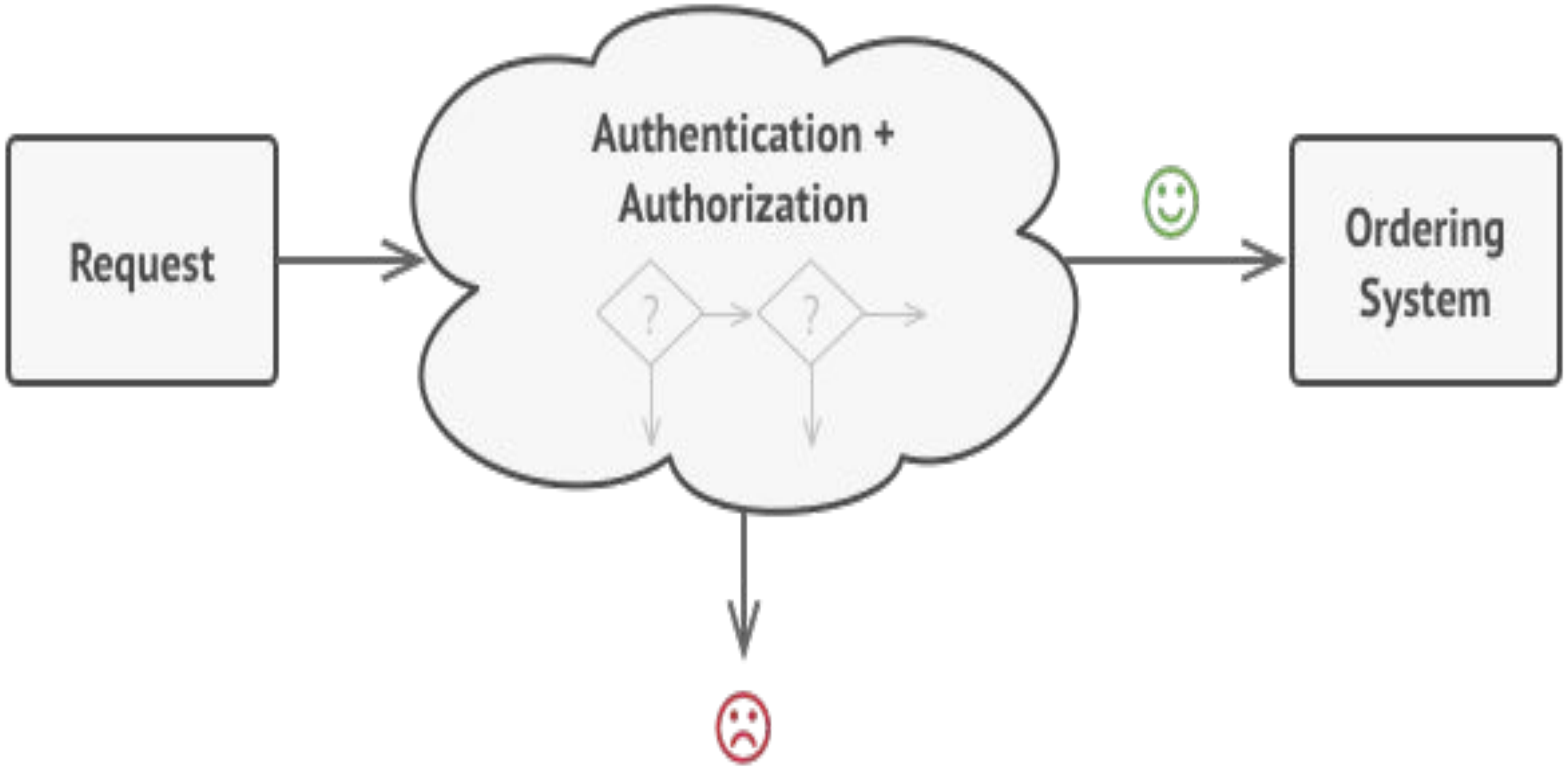
## Intent

**Chain of Responsibility** is a behavioral design pattern that lets you pass requests along a chain of handlers. Upon receiving a request, each handler decides either to process the request or to pass it to the next handler in the chain.



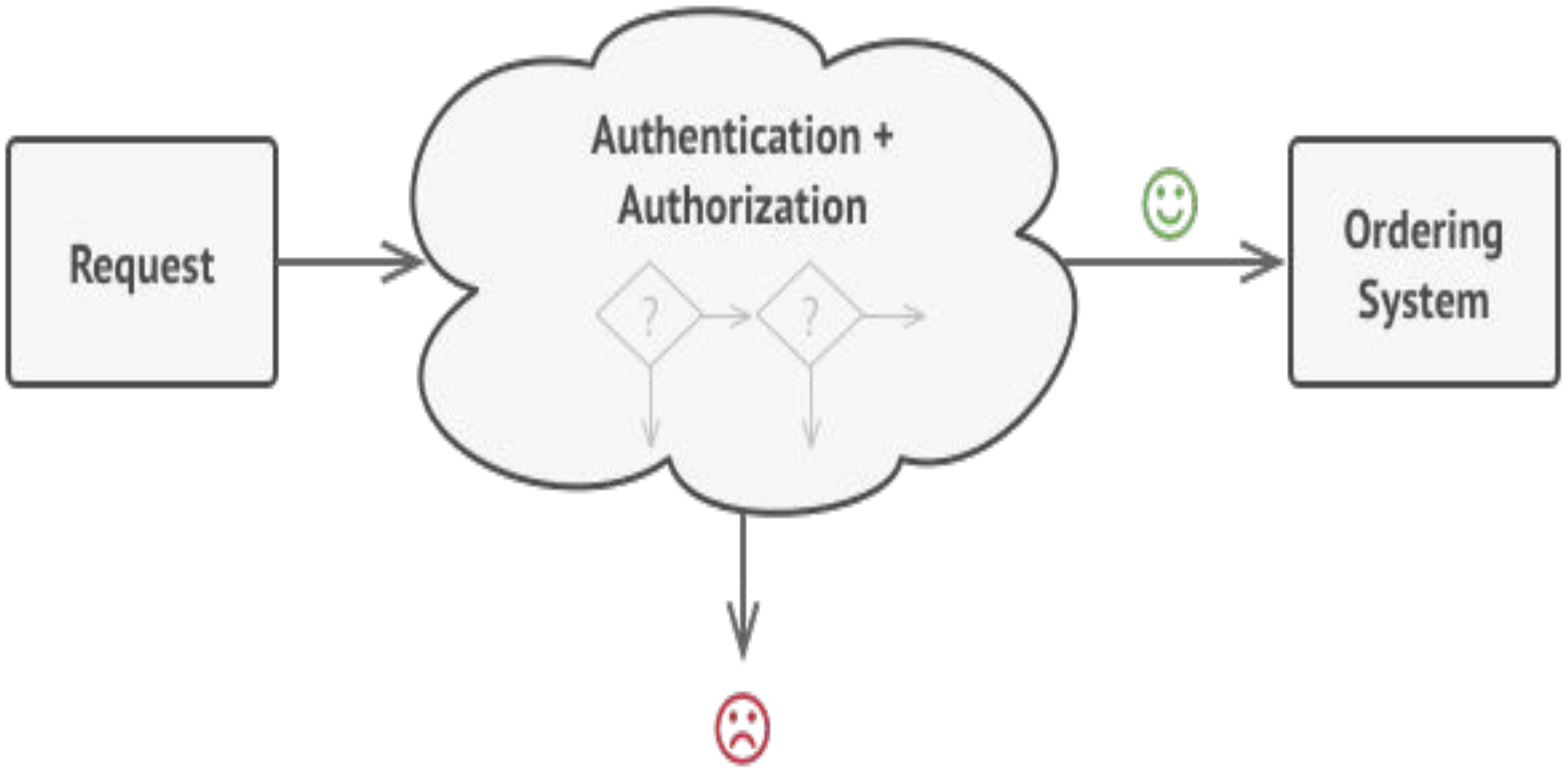
## Problem

Request on an online ordering system with restrict access and administrative permission for some users.



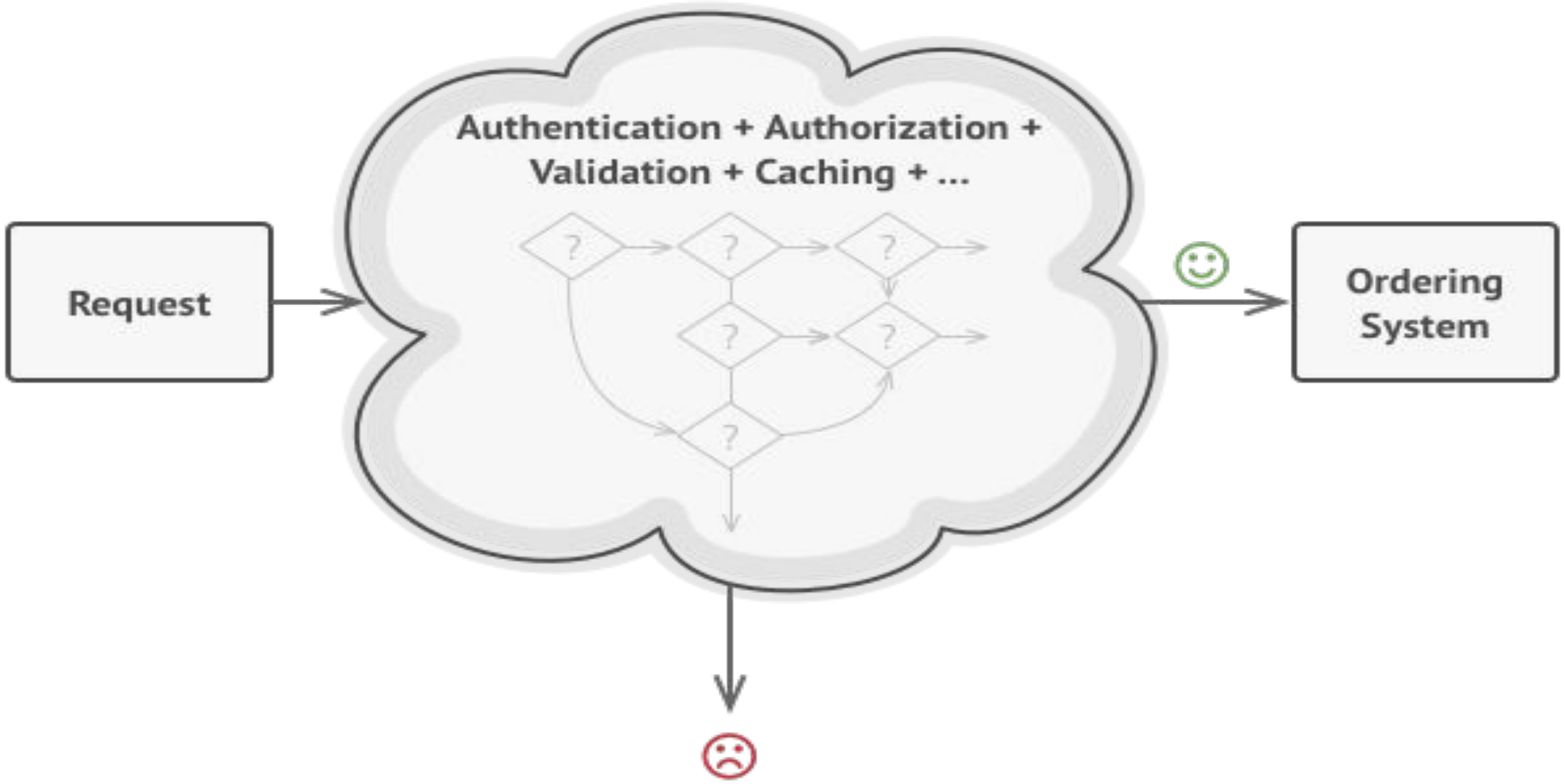
## Problem

After a bit of planning, you realized that these checks must be performed sequentially.



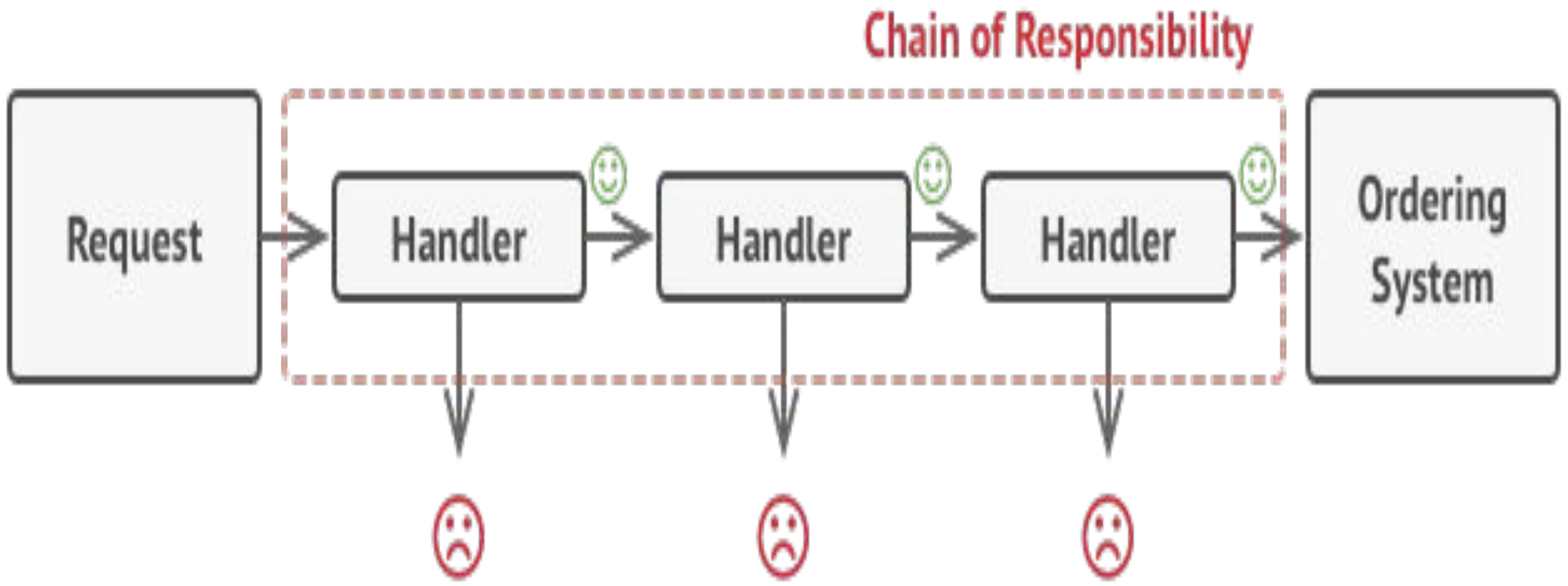
## Problem

During the next few months, you implemented several more of those sequential checks. The system became very hard to comprehend and expensive to maintain.



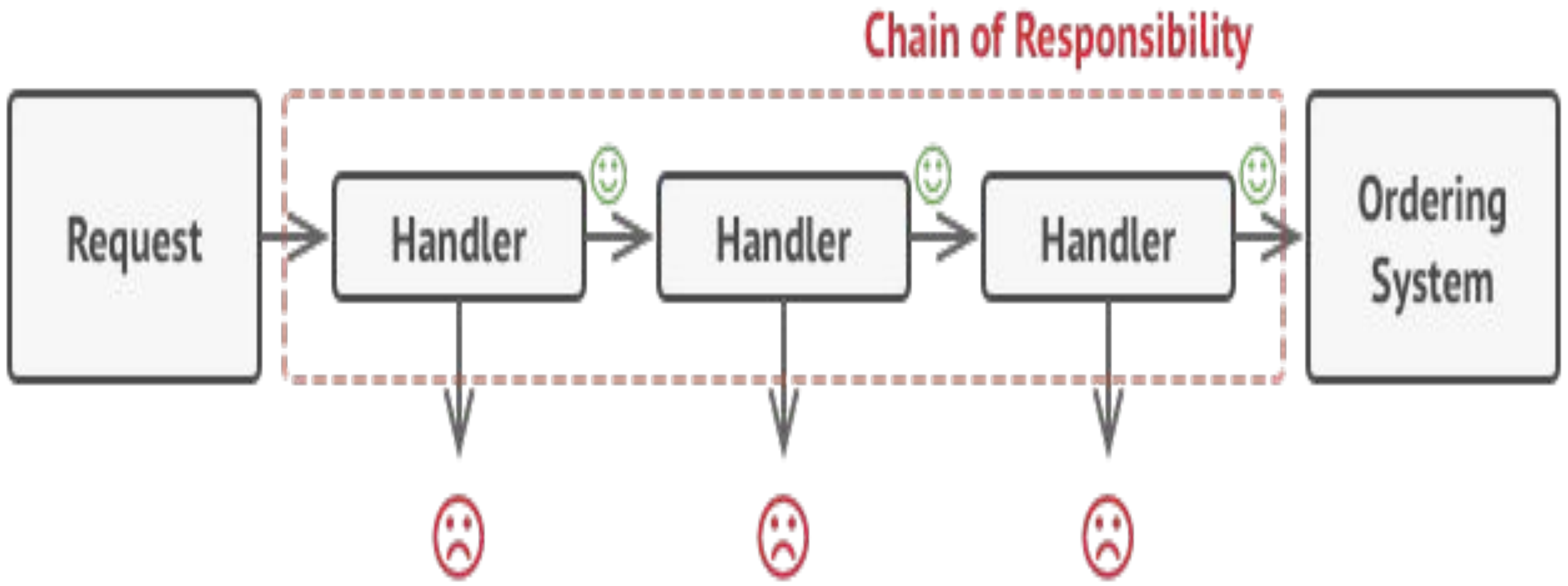
## Chain of Responsibility solution

We must transform particular behaviors into stand alone objects called handlers, each check should be extracted to its own class with a single method that performs the check. The request, along with its data, is passed to this method as an argument.



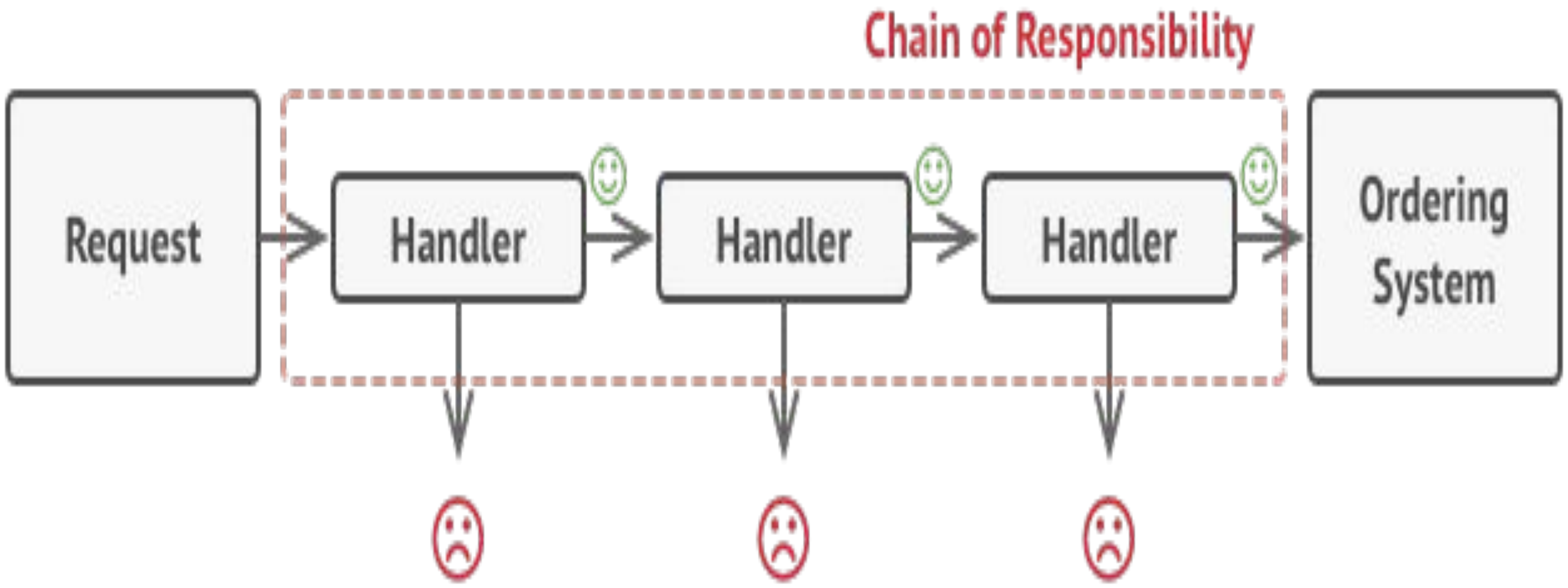
## Chain of Responsibility solution

Each linked handler has a field for storing a reference to the next handler in the chain. In addition to processing a request, handlers pass the request further along the chain.



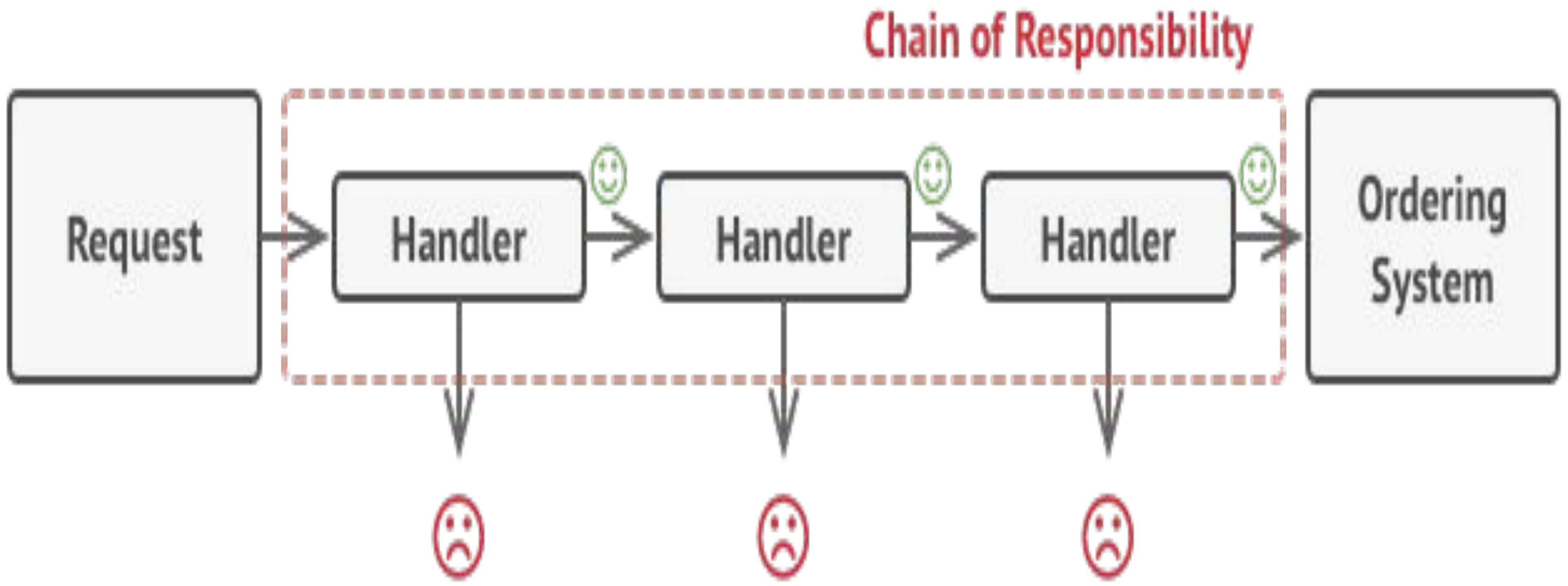
## Chain of Responsibility solution

A handler can decide not to pass the request further down the chain and effectively stop any further processing.

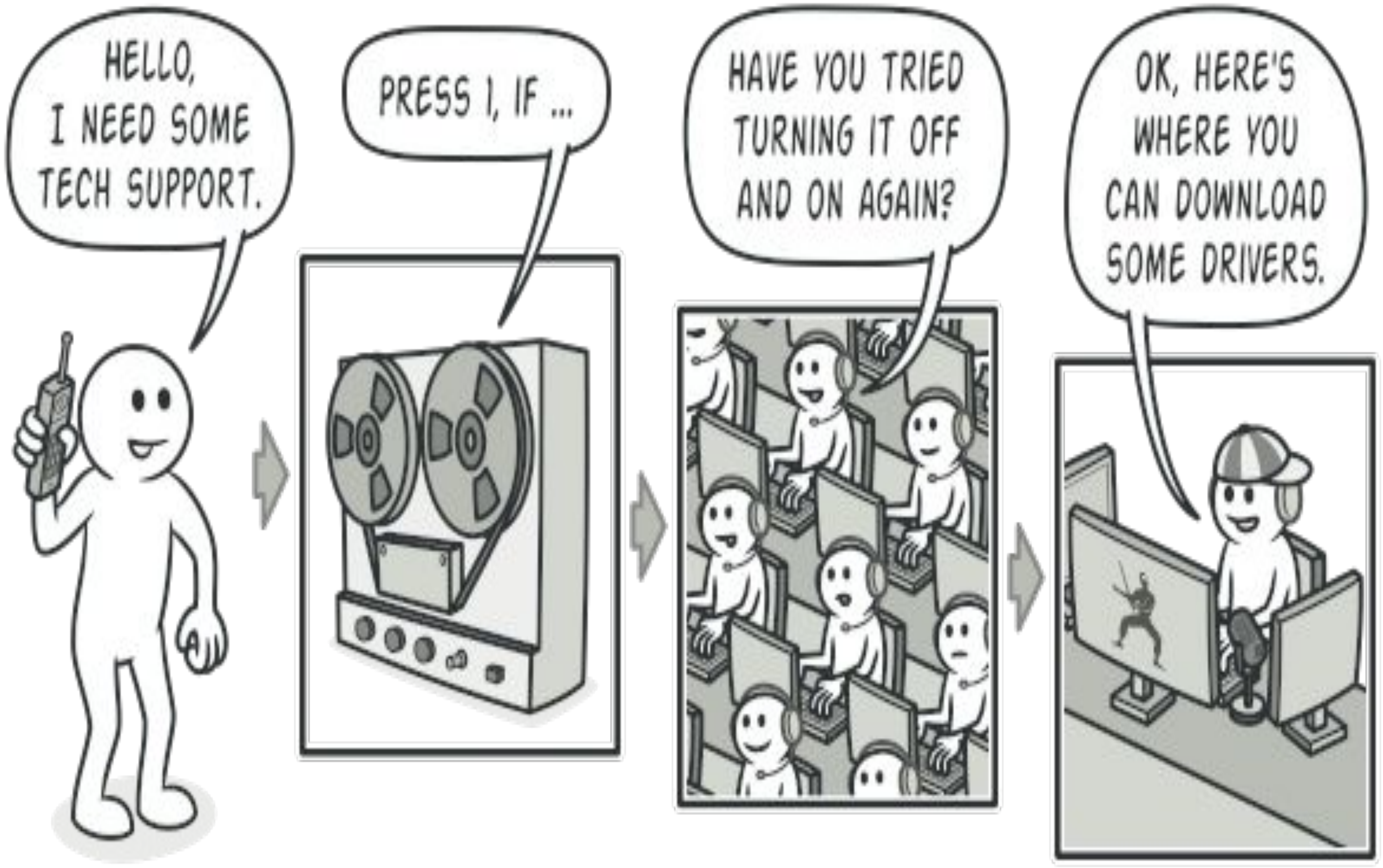


## Chain of Responsibility solution

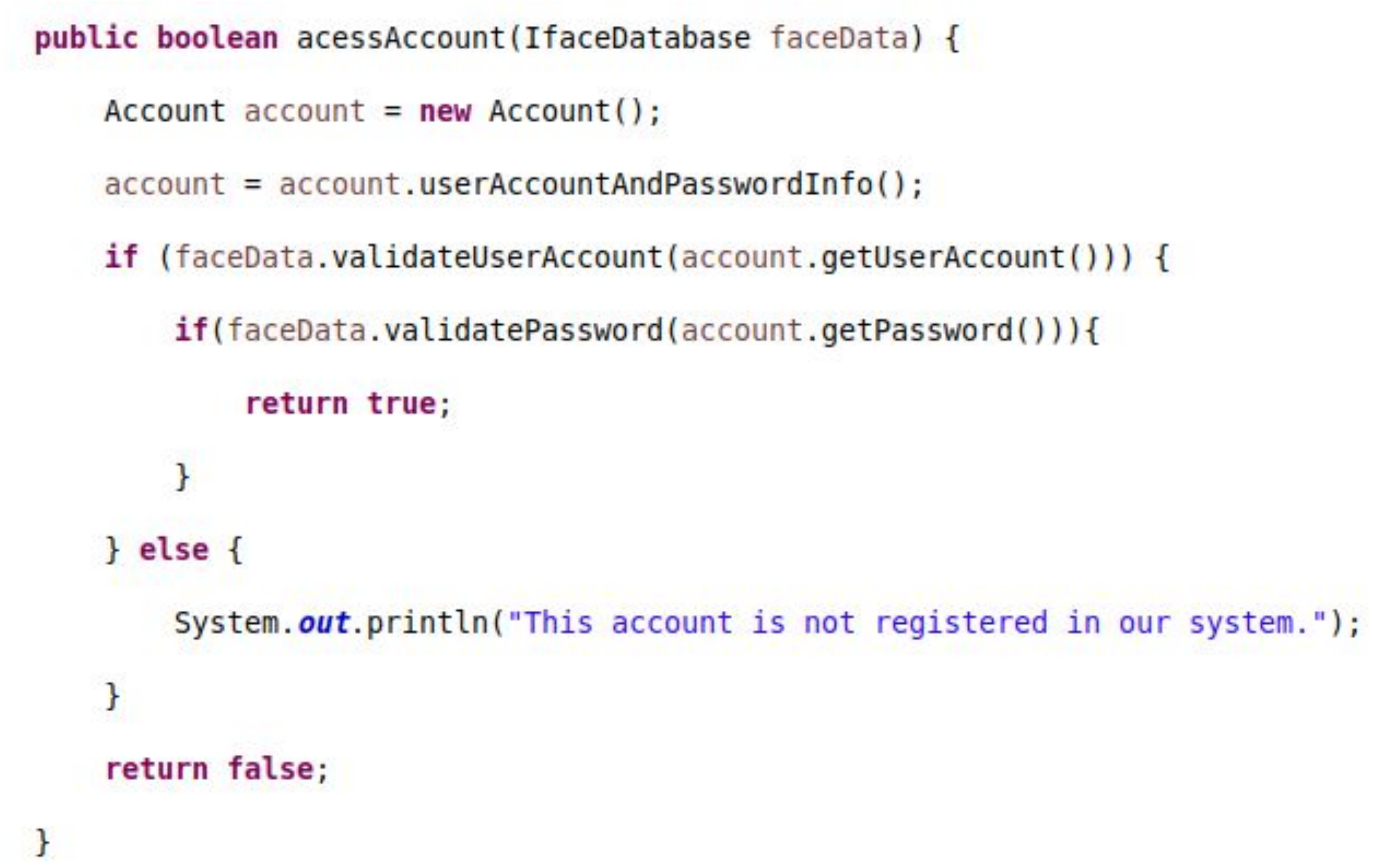
A handler can decide not to pass the request further down the chain and effectively stop any further processing.



## Real-World Analogy

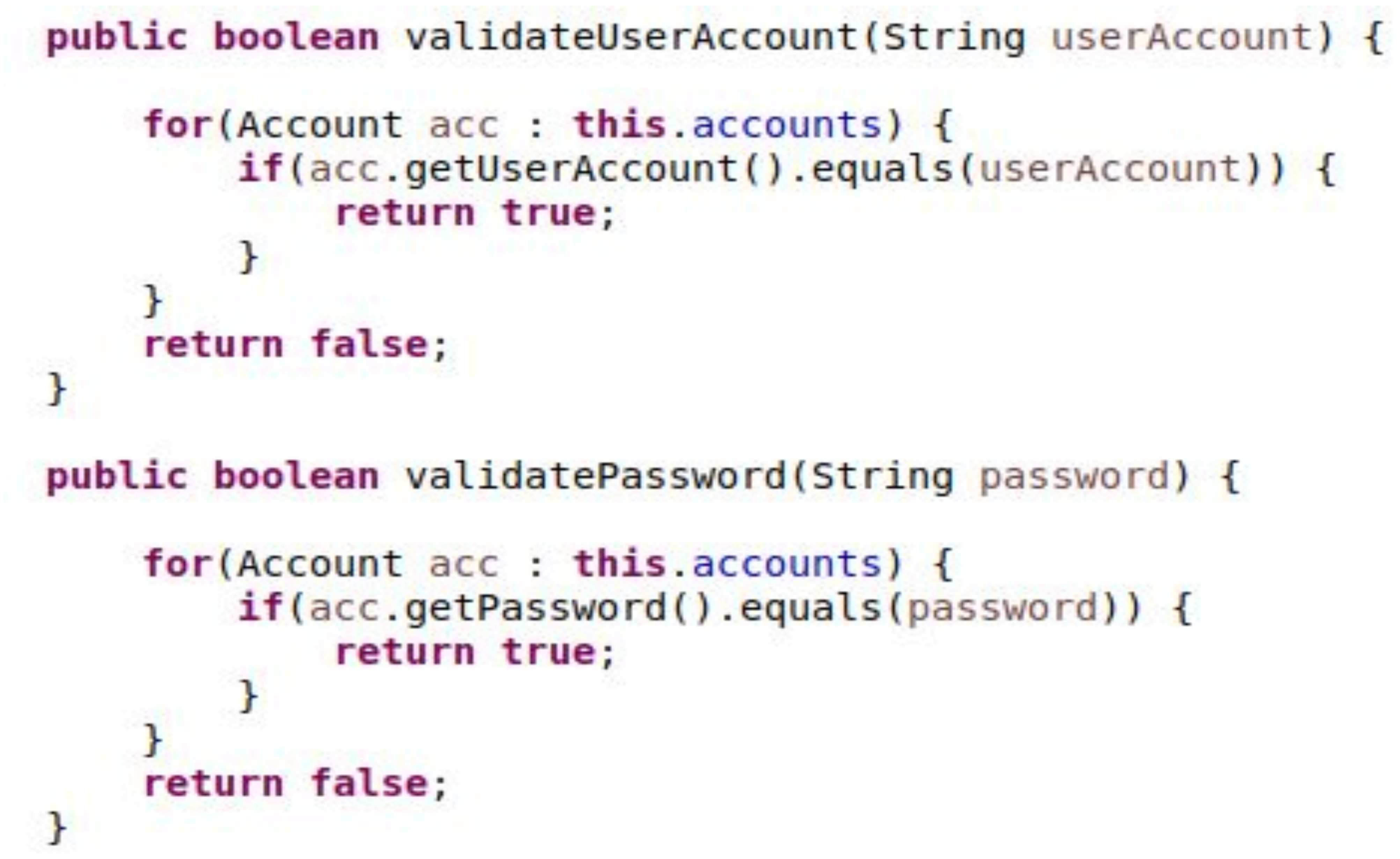

HELLO, I NEED SOME TECH SUPPORT.
PRESS 1, IF ...
HAVE YOU TRIED TURNING IT OFF AND ON AGAIN?
OK, HERE'S WHERE YOU CAN DOWNLOAD SOME DRIVERS.

## Code



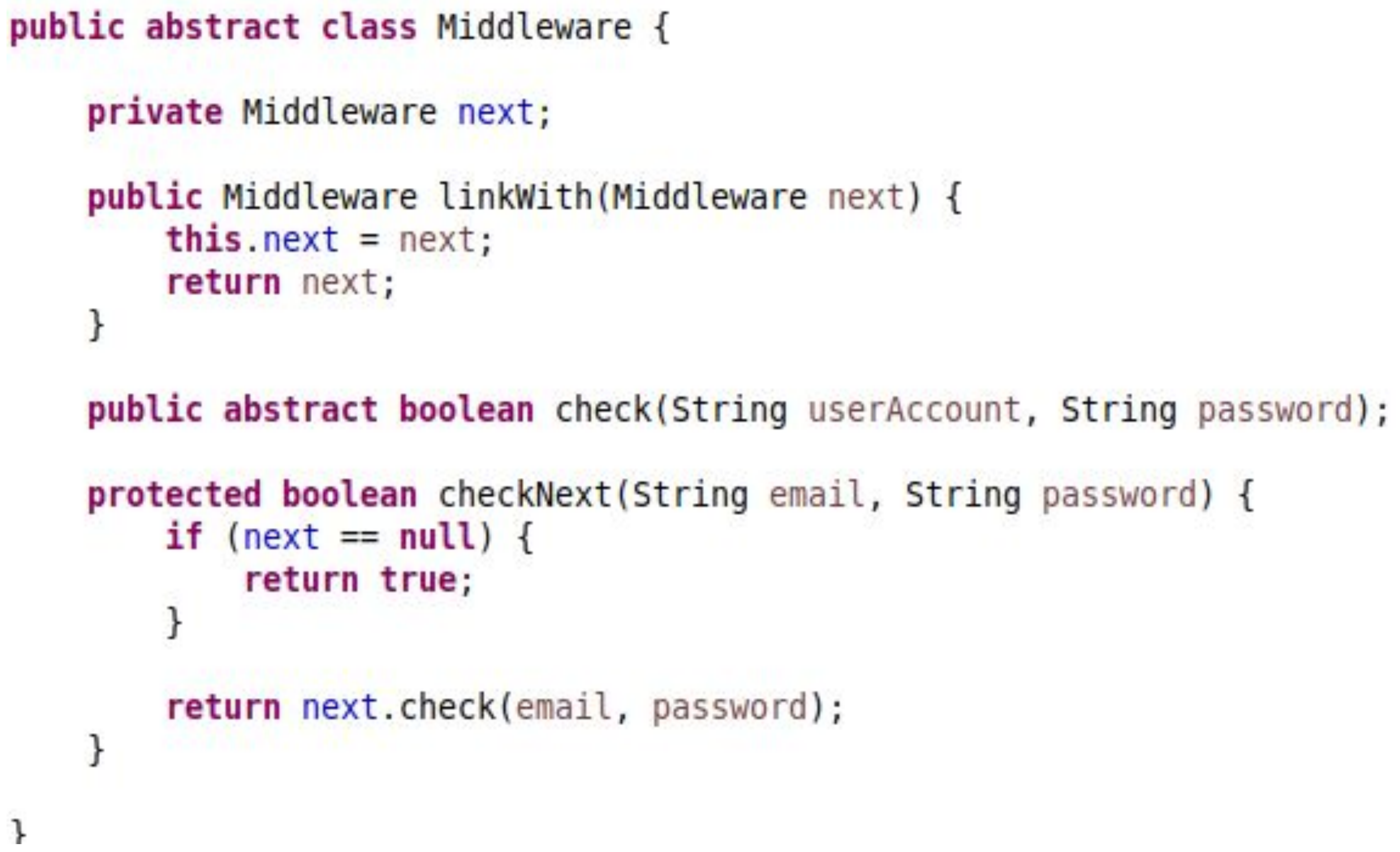
```
public boolean acessAccount(IfaceDatabase faceData) {
    Account account = new Account();
    account = account.userAccountAndPasswordInfo();
    if (faceData.validateUserAccount(account.getUserAccount())) {
        if(faceData.validatePassword(account.getPassword())){
            return true;
        }
    } else {
        System.out.println("This account is not registered in our system.");
    }
    return false;
}
```

## Code



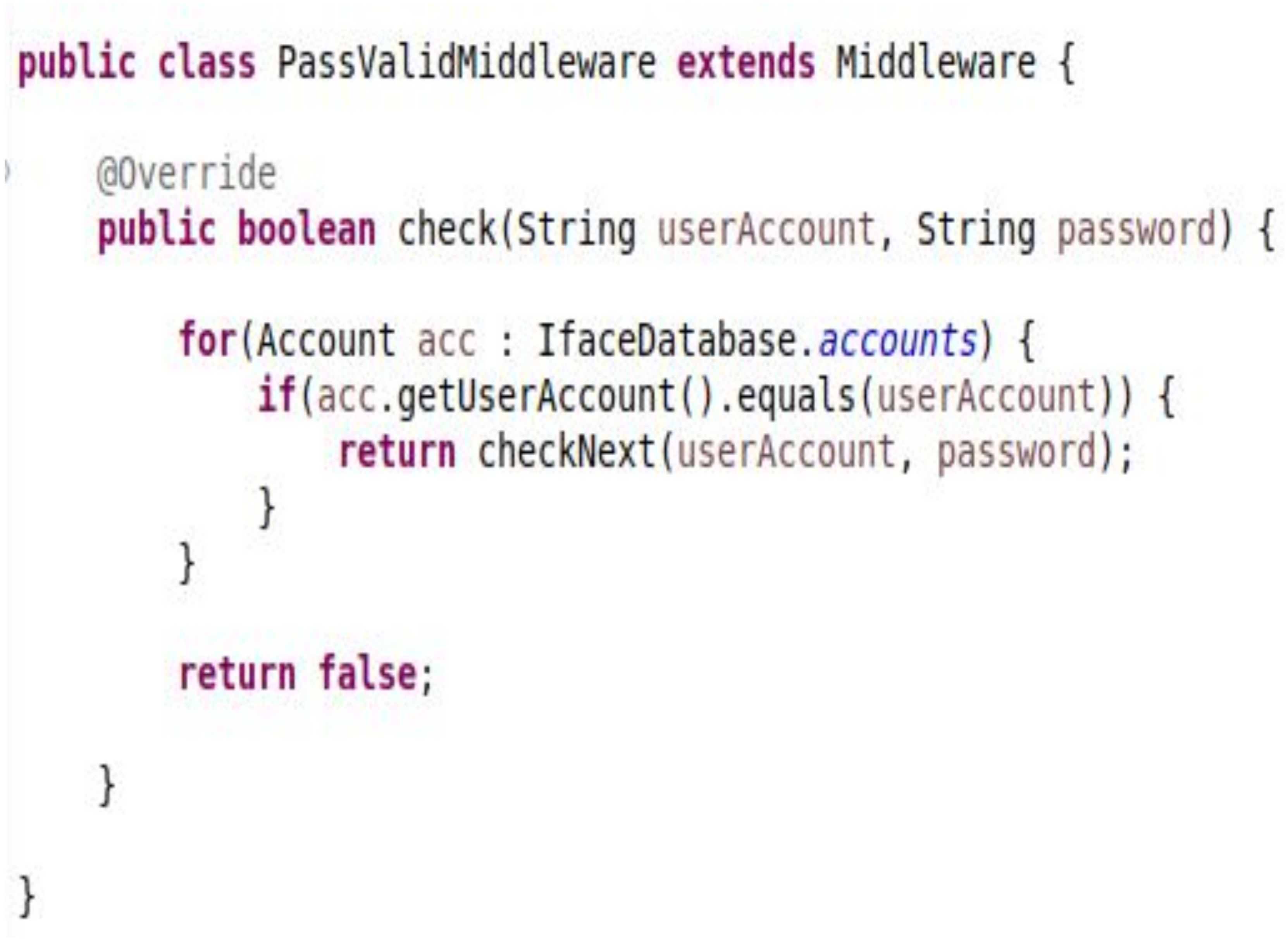
```java
public boolean validateUserAccount(String userAccount) {

    for(Account acc : this.accounts) {
        if(acc.getUserAccount().equals(userAccount)) {
            return true;
        }
    }
    return false;
}

public boolean validatePassword(String password) {

    for(Account acc : this.accounts) {
        if(acc.getPassword().equals(password)) {
            return true;
        }
    }
    return false;
}
```

## Chain of Responsability



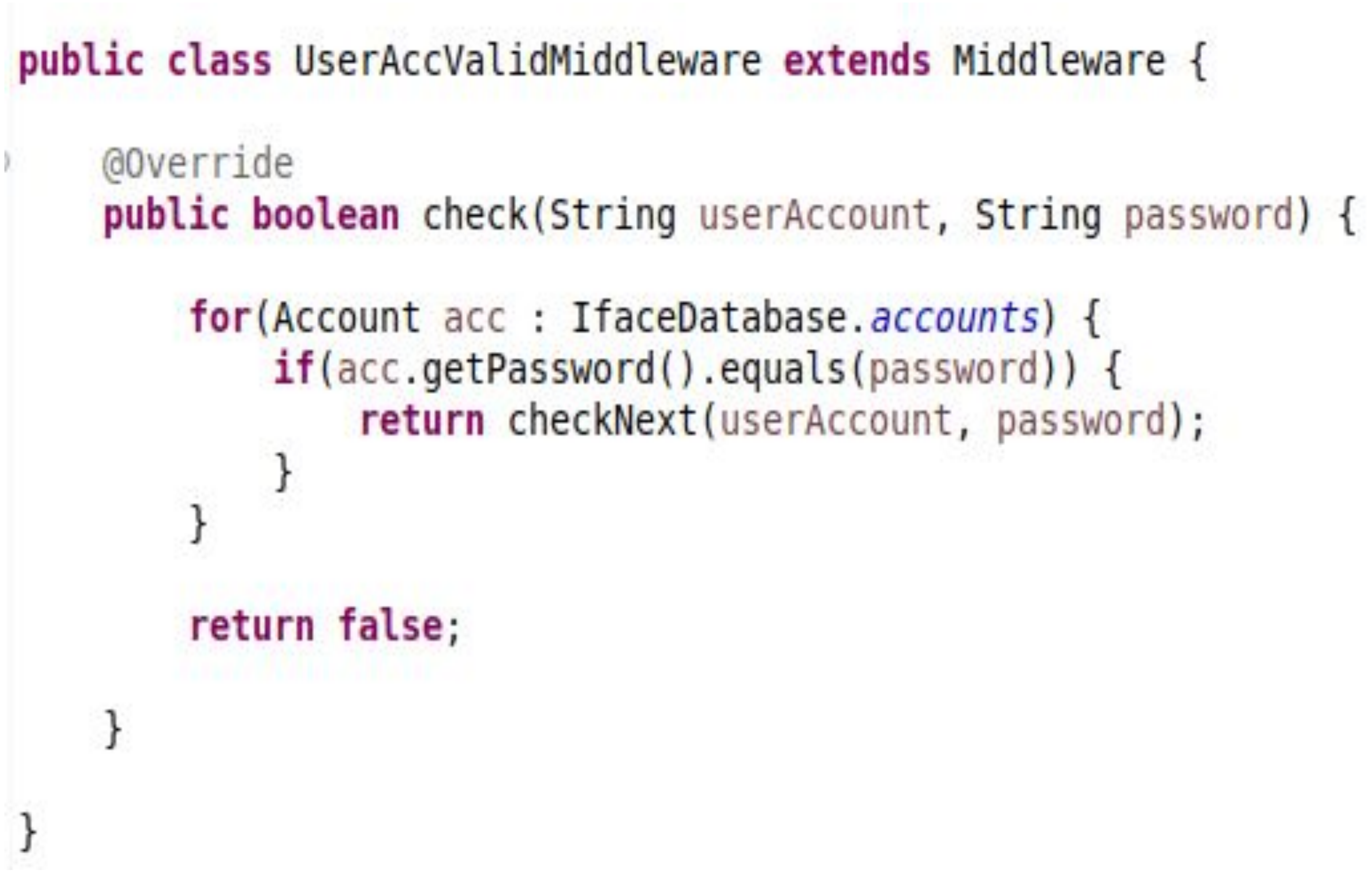
```java
public abstract class Middleware {

    private Middleware next;

    public Middleware linkWith(Middleware next) {
        this.next = next;
        return next;
    }

    public abstract boolean check(String userAccount, String password);

    protected boolean checkNext(String email, String password) {
        if (next == null) {
            return true;
        }

        return next.check(email, password);
    }

}
```

## Chain of Responsability



```java
public class PassValidMiddleware extends Middleware {

    @Override
    public boolean check(String userAccount, String password) {

        for(Account acc : IfaceDatabase.accounts) {
            if(acc.getUserAccount().equals(userAccount)) {
                return checkNext(userAccount, password);
            }
        }

        return false;

    }

}
```

## Chain of Responsability



```java
public class UserAccValidMiddleware extends Middleware {

    @Override
    public boolean check(String userAccount, String password) {

        for(Account acc : IfaceDatabase.accounts) {
            if(acc.getPassword().equals(password)) {
                return checkNext(userAccount, password);
            }
        }

        return false;

    }

}
```

## Chain of Responsability

```
Middleware middleware = new PassValidMiddleware().linkWith(new UserAccValidMiddleware());


if(middleware.check(userAccount, passWord)) {

    ProfileScreen profileScreen = new ProfileScreen();
    faceData = profileScreen.profileScreen(faceData, account);

}
```

## Reference

Dive Into
DESIGN PATTERNS
Alexander Shvets