Dependency Injection

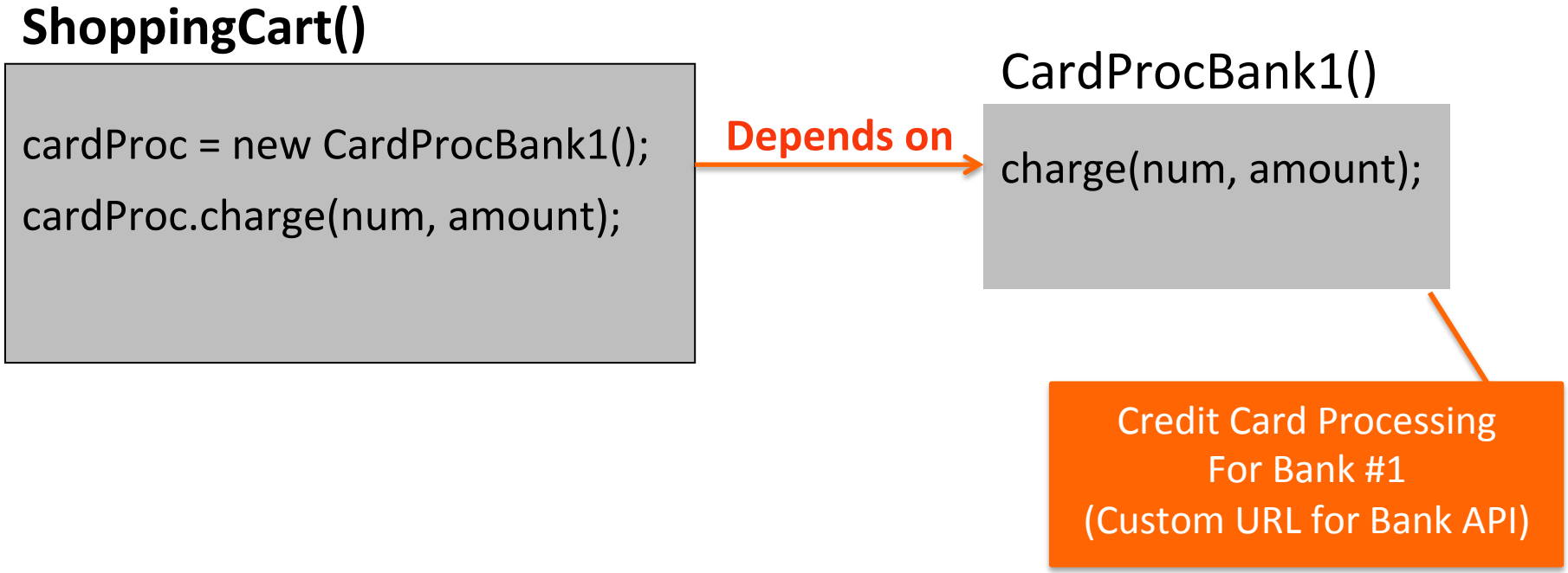
```
angular.module('myApp', [])
.controller('MyCtrl', function ($scope) {
  $scope.name = "Yaakov";
});
```

Where did $scope come from?

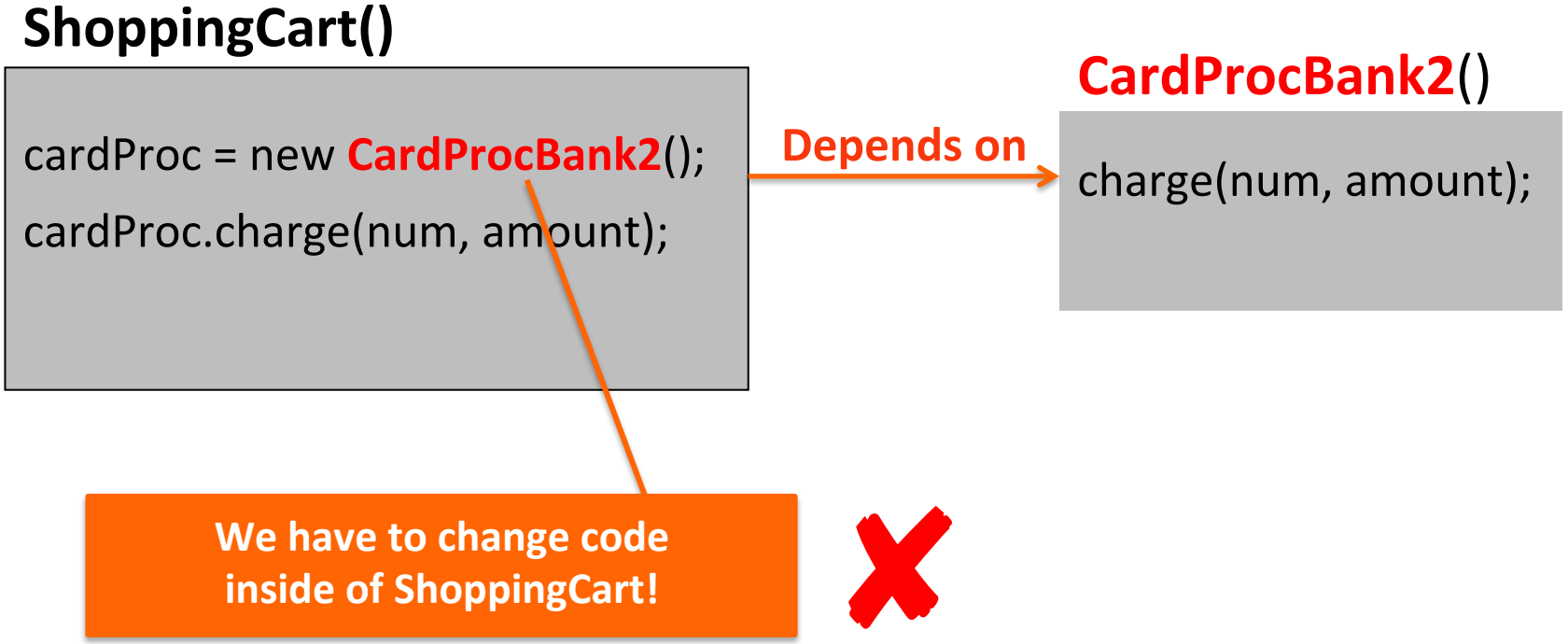
Dependency Injection

Design pattern that implements *inversion of control* for resolving dependencies

“Regular” Control



“Regular” Control



Inversion of Control (IoC)

ShoppingCart(cardProc)

```
cardProc.charge(num, amount);
```

CardProcBank1()

```
charge(num, amount);
```

CardProcBank2()

```
charge(num, amount);
```

System

```
cardProc = new CardProcBank1();
ShoppingCart(cardProc);
```

If we need a different bank
for card processing,
ShoppingCart code will not change

Summary

- Design pattern: Dependency Injection (DI)
- Implements Inversion of Control (IoC)
- Client gets called with the dependency by some system
  - In our case, the “system” is AngularJS
- Client is not responsible for instantiating the dependency