

Принципы проектирования и дизайна ПО

Лекция №7

Агошков Илья 2016

Single Responsibility

Инкапсуляция

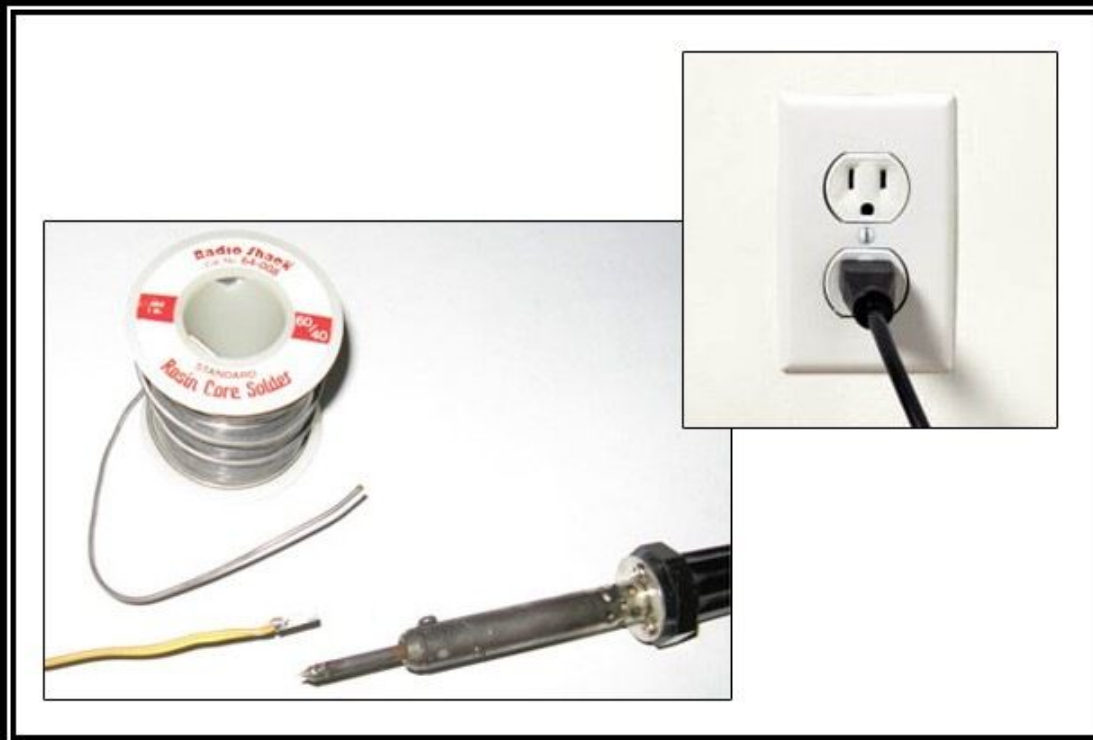
Tell don't ask

Dependency inversion principle

A. HIGH LEVEL MODULES SHOULD NOT DEPEND UPON LOW LEVEL MODULES. BOTH SHOULD DEPEND UPON ABSTRACTIONS.

B. ABSTRACTIONS SHOULD NOT DEPEND UPON DETAILS. DETAILS SHOULD DEPEND UPON ABSTRACTIONS.

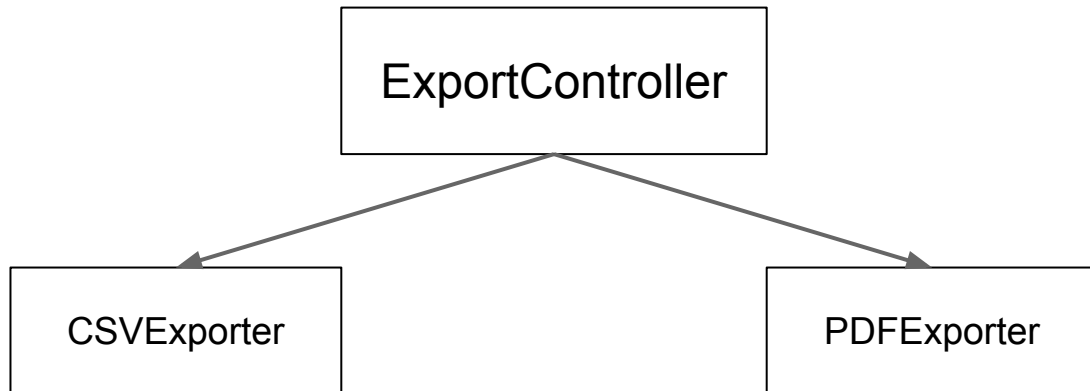
Robert C. Martin “Uncle Bob”



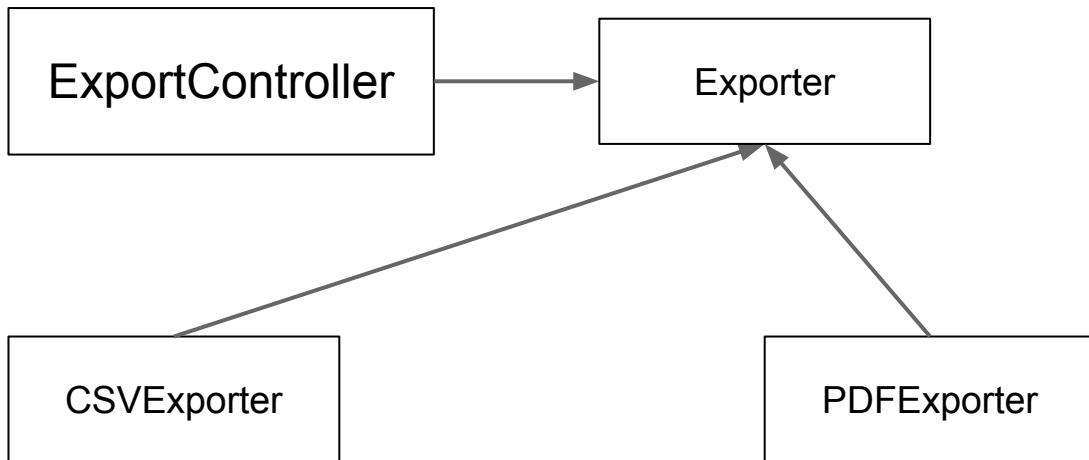
DEPENDENCY INVERSION PRINCIPLE

Would You Solder A Lamp Directly To The Electrical Wiring In A Wall?

Sample application



Sample application



Coupling / cohesion

couple (verb)

to join (something) to something else

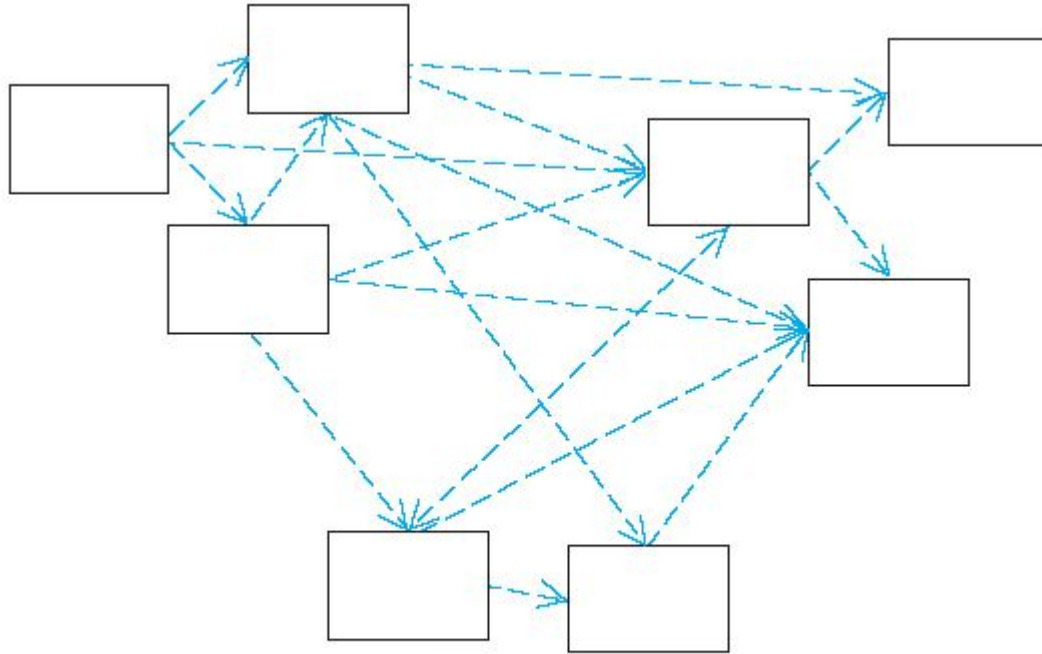
In software

In software development, coupling or dependency is the degree to which each program module relies on each one of the other modules

Highly coupled

These types of systems have interconnections, with program units dependent on each other.

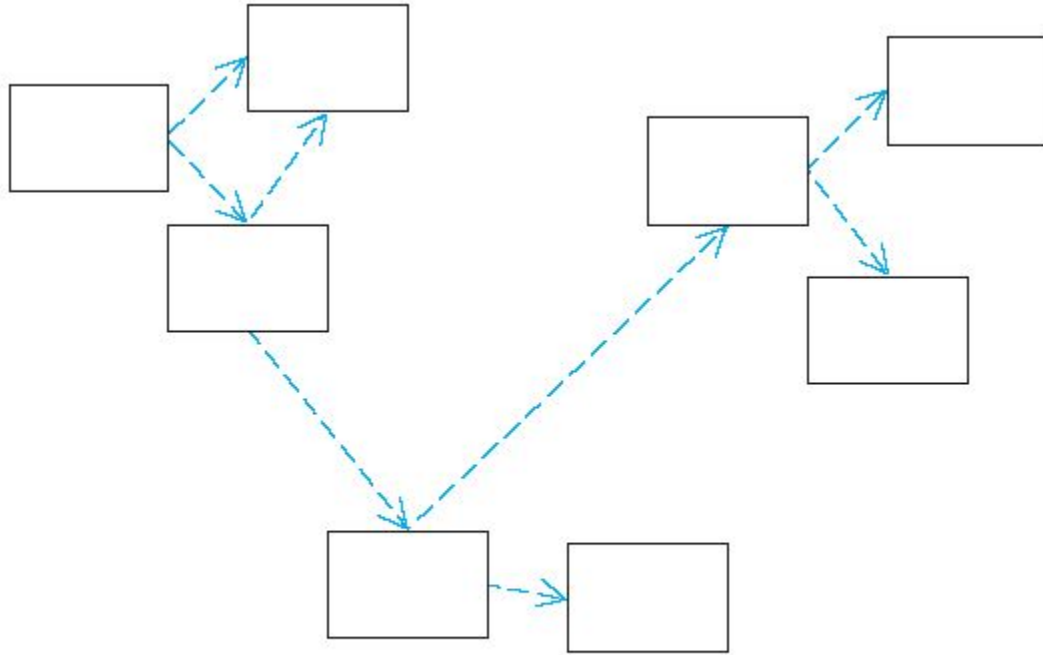
Highly coupled



Loosely coupled

Loosely coupled systems are made up of components which are independent or almost independent.

Loosely coupled



Loose coupling is good

cohere (verb)

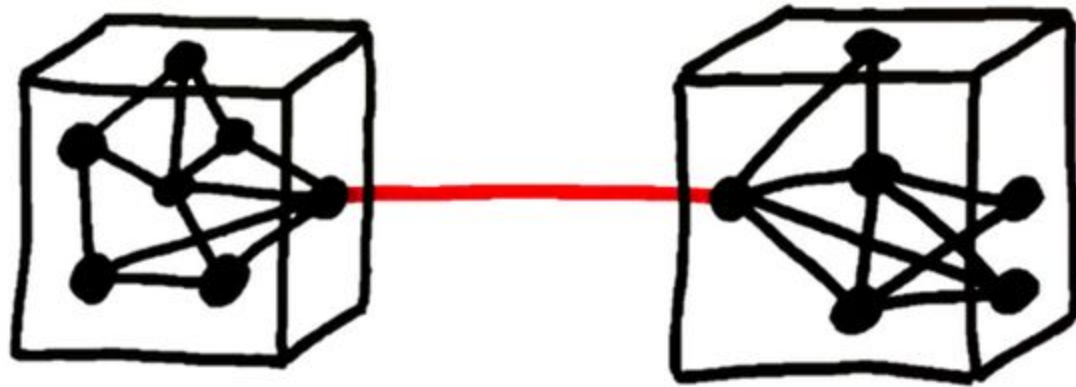
- to be combined or united in a logical and effective way
- to hold together firmly as a part of the same mass

In software

In computer programming, cohesion refers to the degree to which the elements of a module belong together. Thus, it is a measure of how strongly related each piece of functionality expressed by the source code of software module is.

High cohesion is good

High cohesion - loose coupling



Практика (10-15 минут)

Проанализировать
каждый из SOLID
принципов на предмет
того, как изменяется
cohesion/coupling при
применении принципа.

Single Responsibility
Open Closed
Liskov substitution
Interface seggregation
Dependency Inversion

TradesJob

TradesJob

downloadFileFromFtp

parseFile

saveTrades

Разбор домашнего задания

Что все еще осталось плохо?
Какие из SOLID принципов можно
применить?

Multiple responsibilities

TradesJob

downloadFileFromFtp

input host/login/password/filename

output: local filename

parseFile

input: filename

output: Collection<CSVRecord>

saveTrades

input: Collection<CSVRecord>

output: nothing if all is ok, exception if error

SRP?

TradesJob

downloadFileFromFtp

parseFile

saveTrades

SRP violation

TradesJob

downloadFileFromFtp

parseFile

saveTrades

This class methods have
low cohesion and high
coupling.

SRP violation

TradesJob

downloadFileFromFtp

parseFile

saveTrades

This class methods have
low cohesion and **high coupling**.

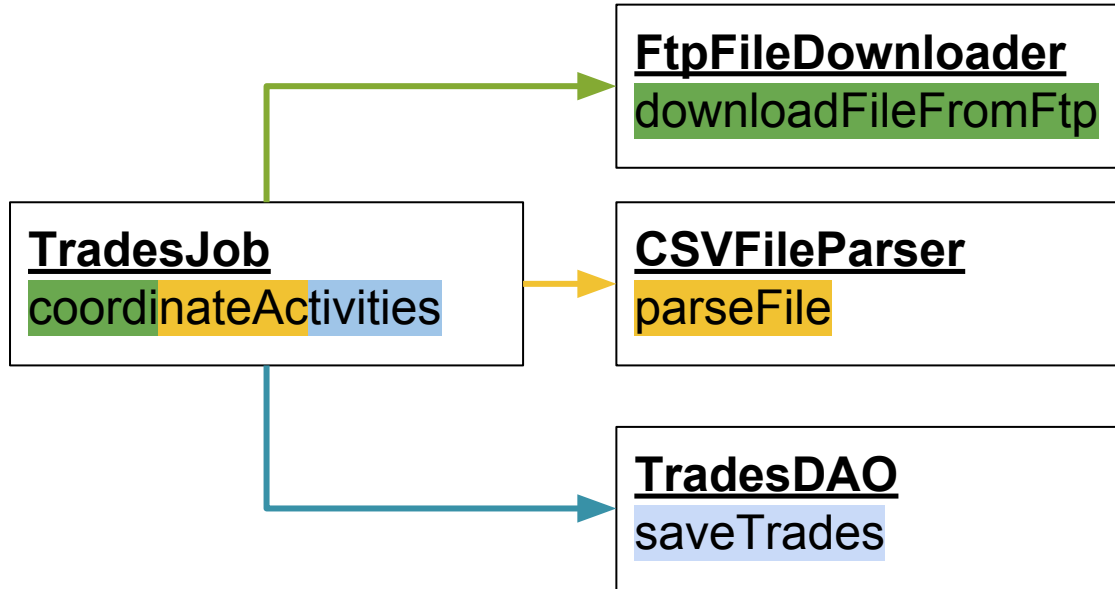
Low cohesion = doing unrelated stuff

High coupling = knowing too much about each other,
e.g.

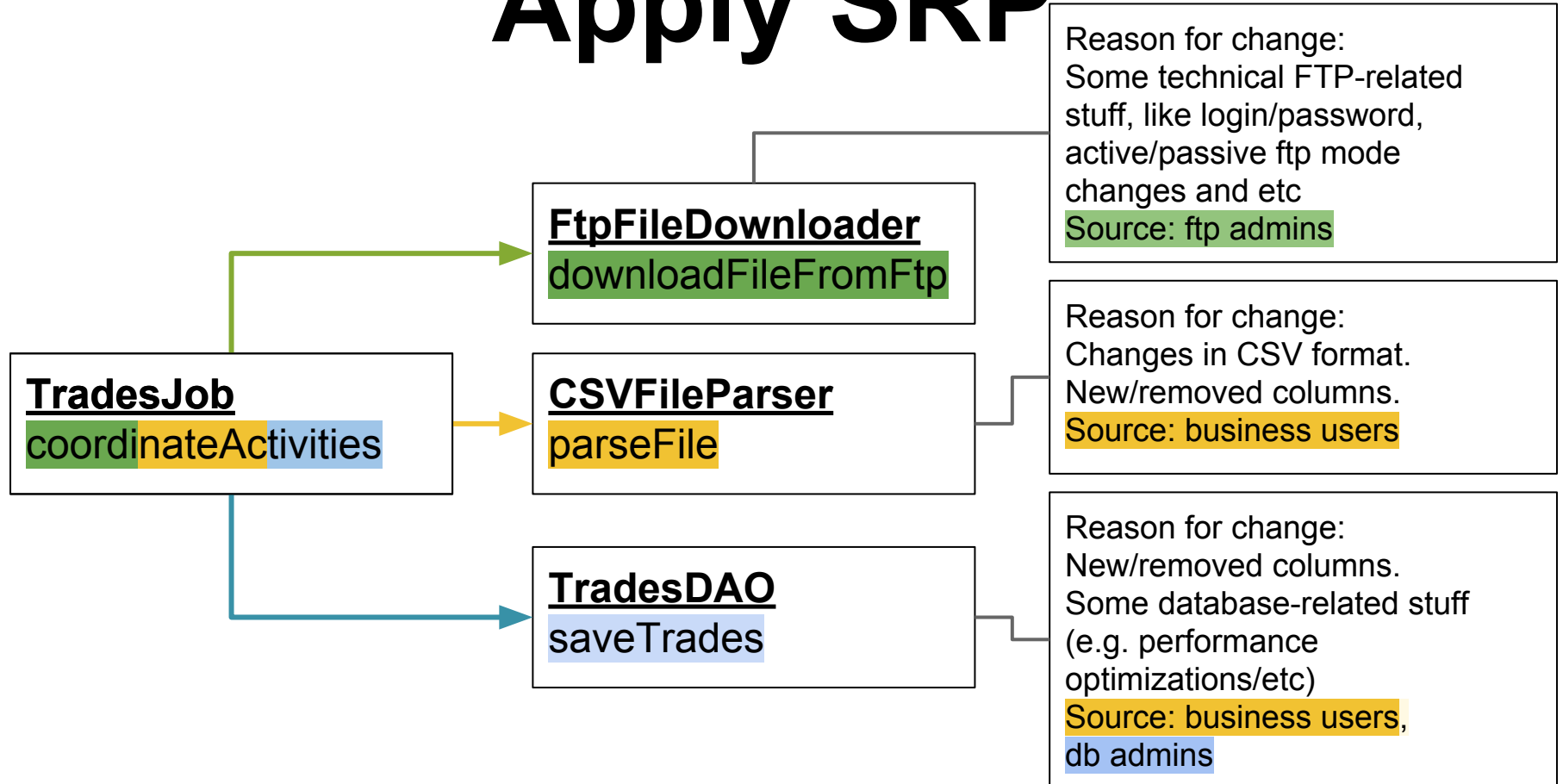
parseFile method knows requires a filename thus it
knows about the files,

saveTrades method accepts `Collection<CSVRecord>`
thus it knows about CSV.

Extract classes

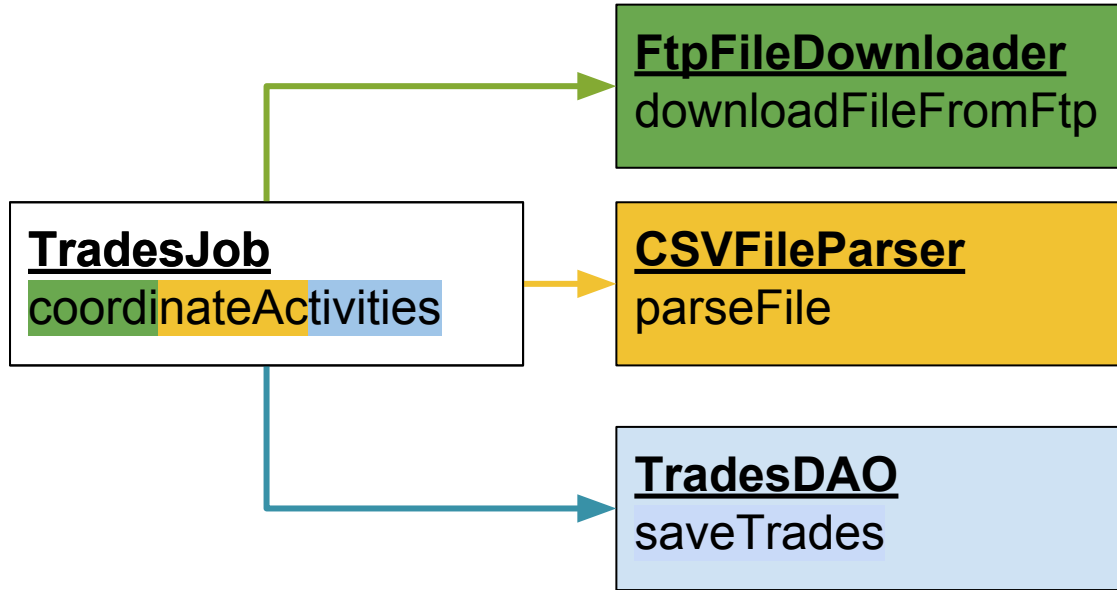


Apply SRP



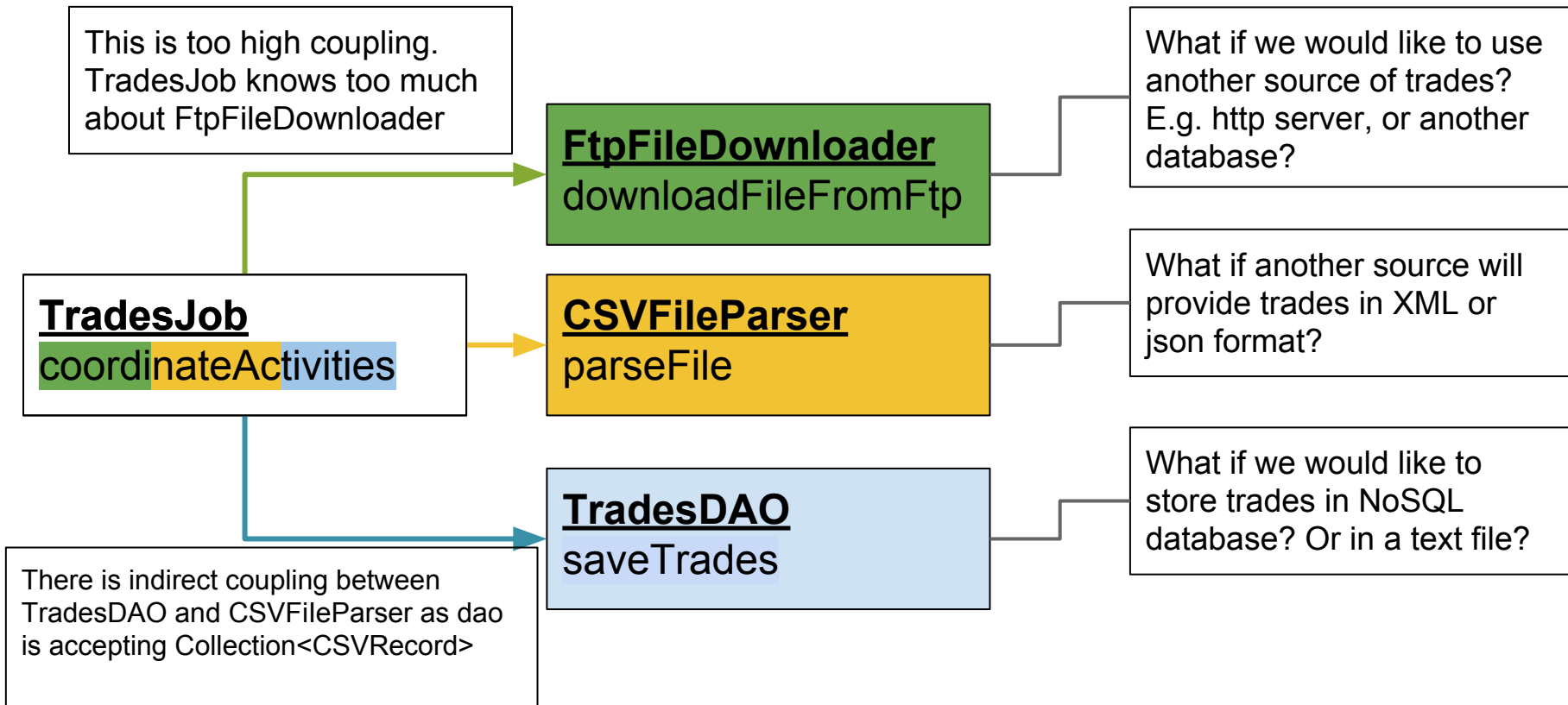
Apply SRP

 Still too high coupling

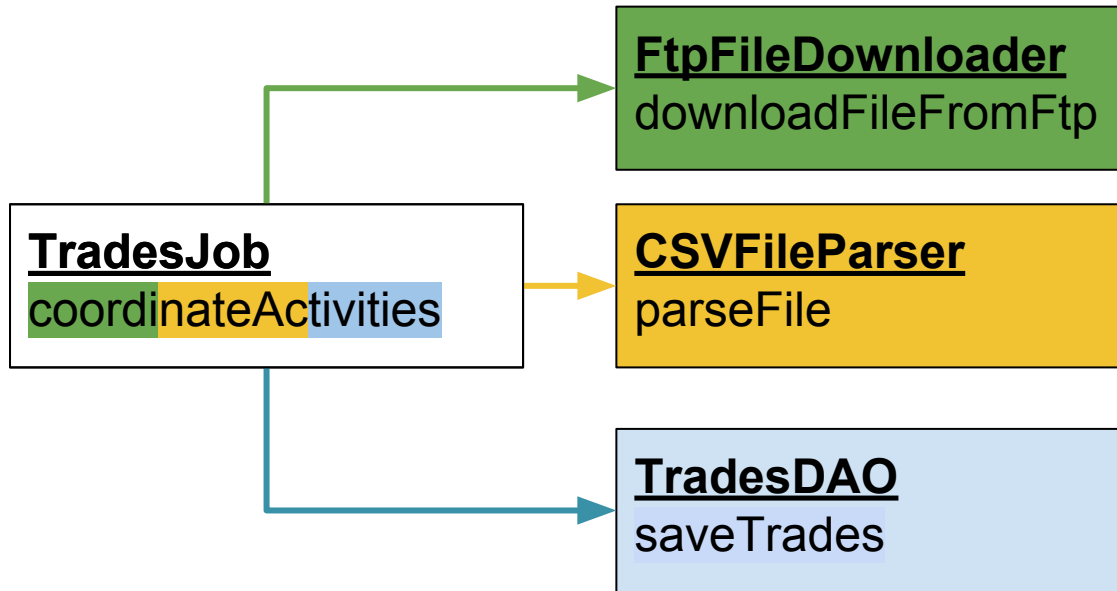


Higher degree of cohesion.
Each class performing one function all class methods are highly cohesive.

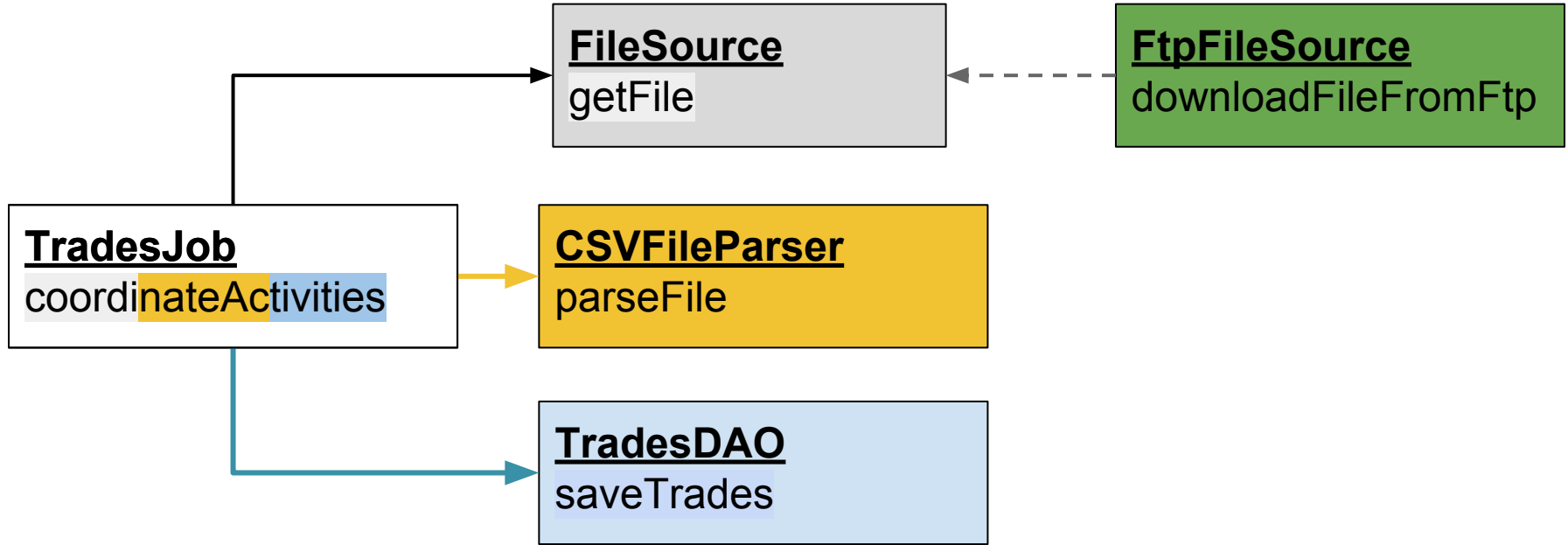
OCP violation



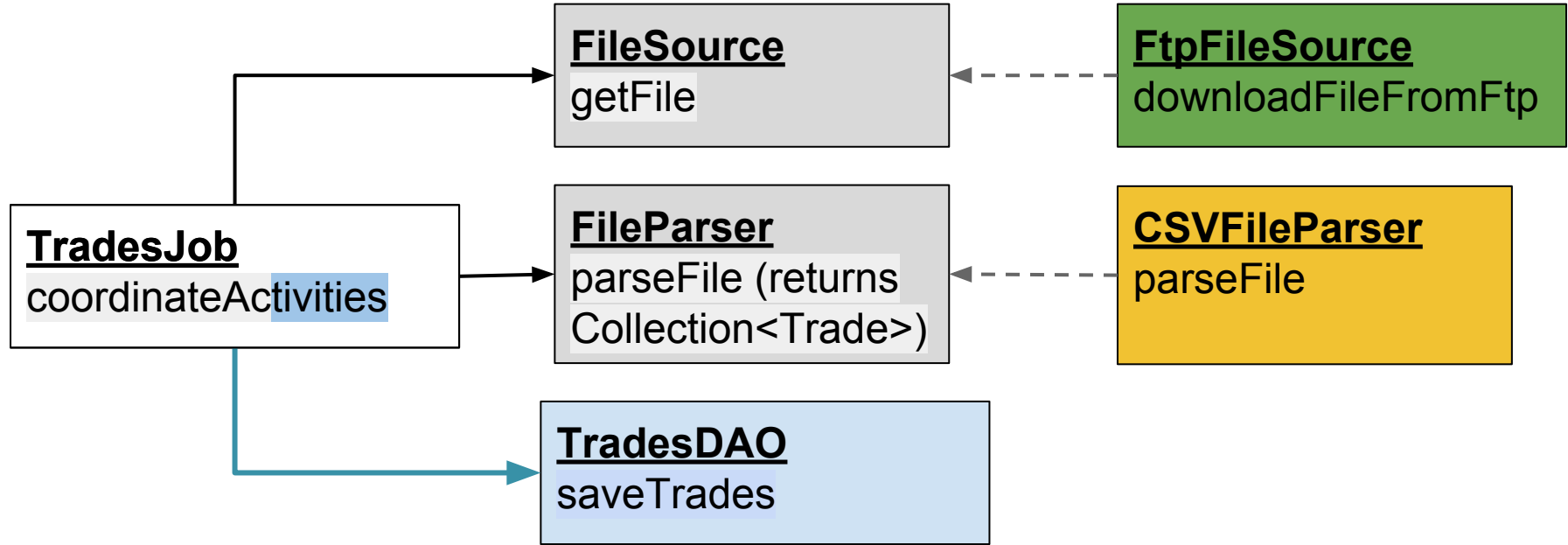
Fixing OCP and DIP



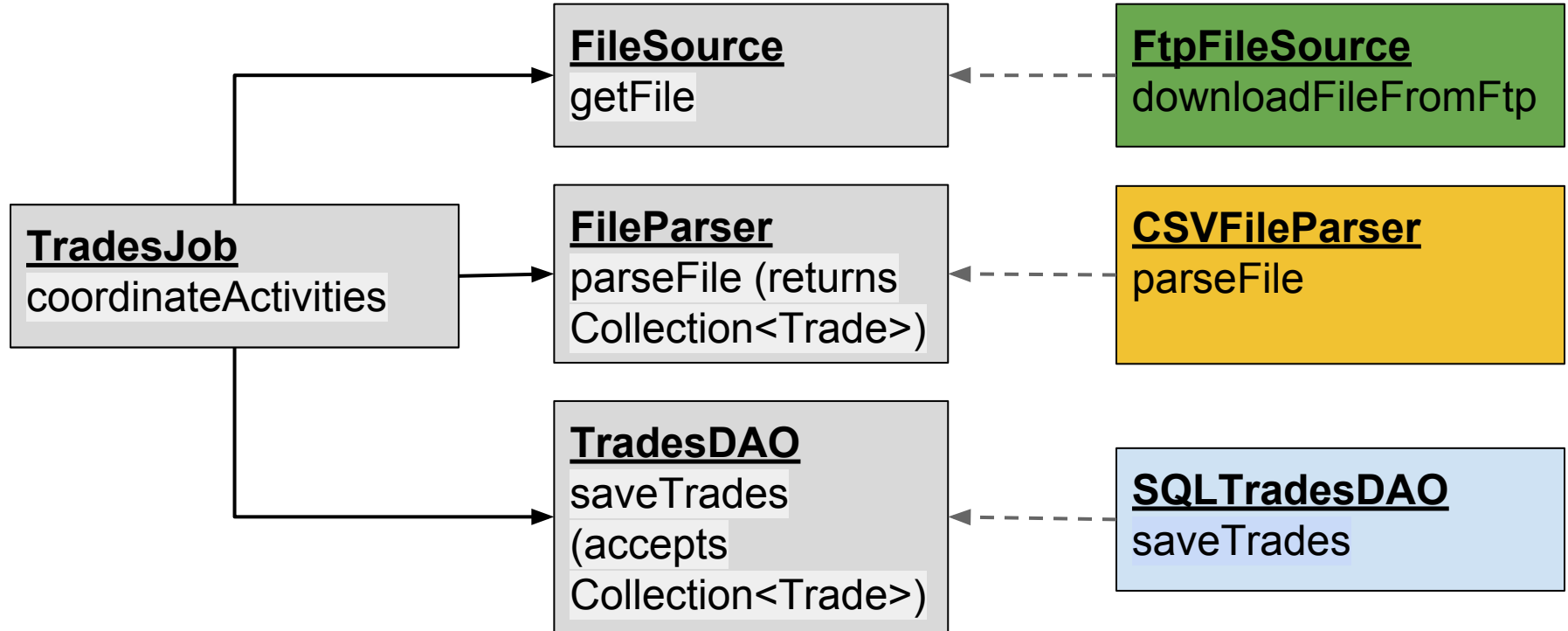
Introduce abstractions (interfaces)



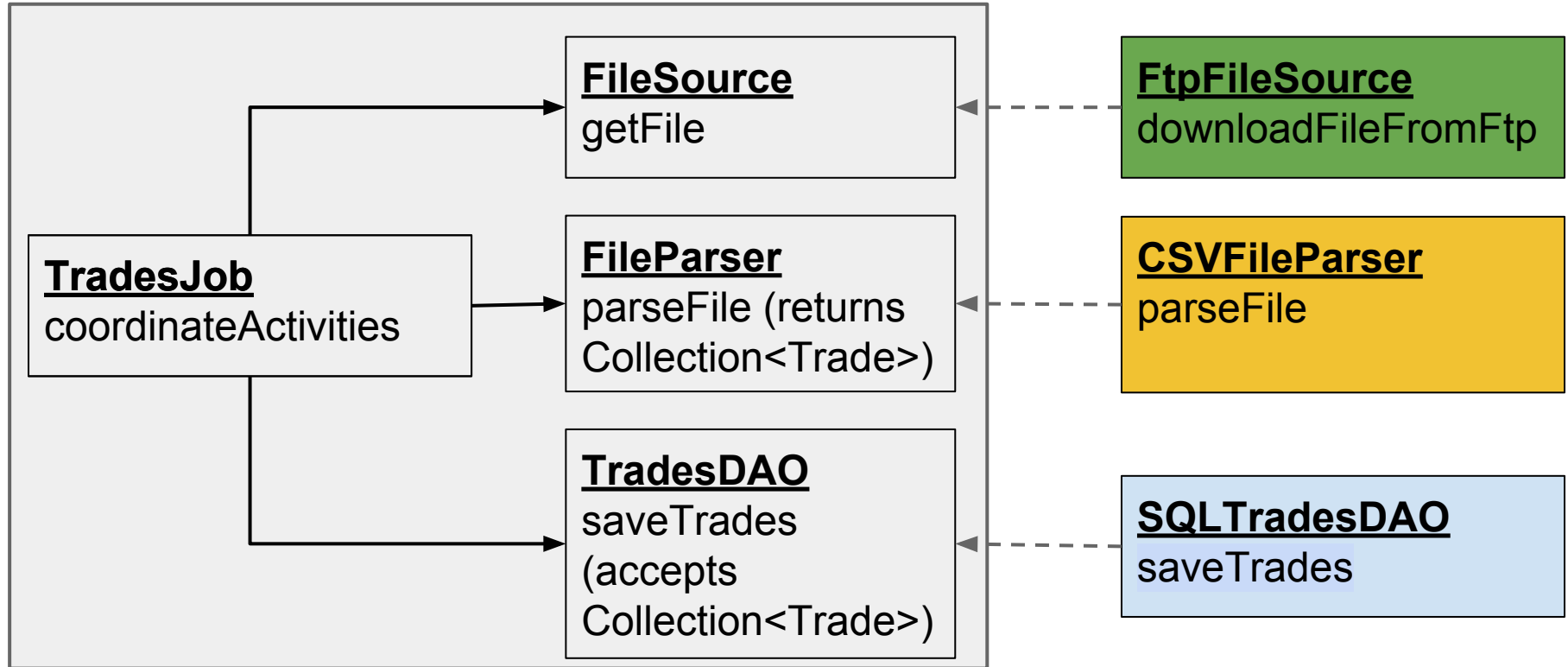
Make more detailed thing depend on more abstract



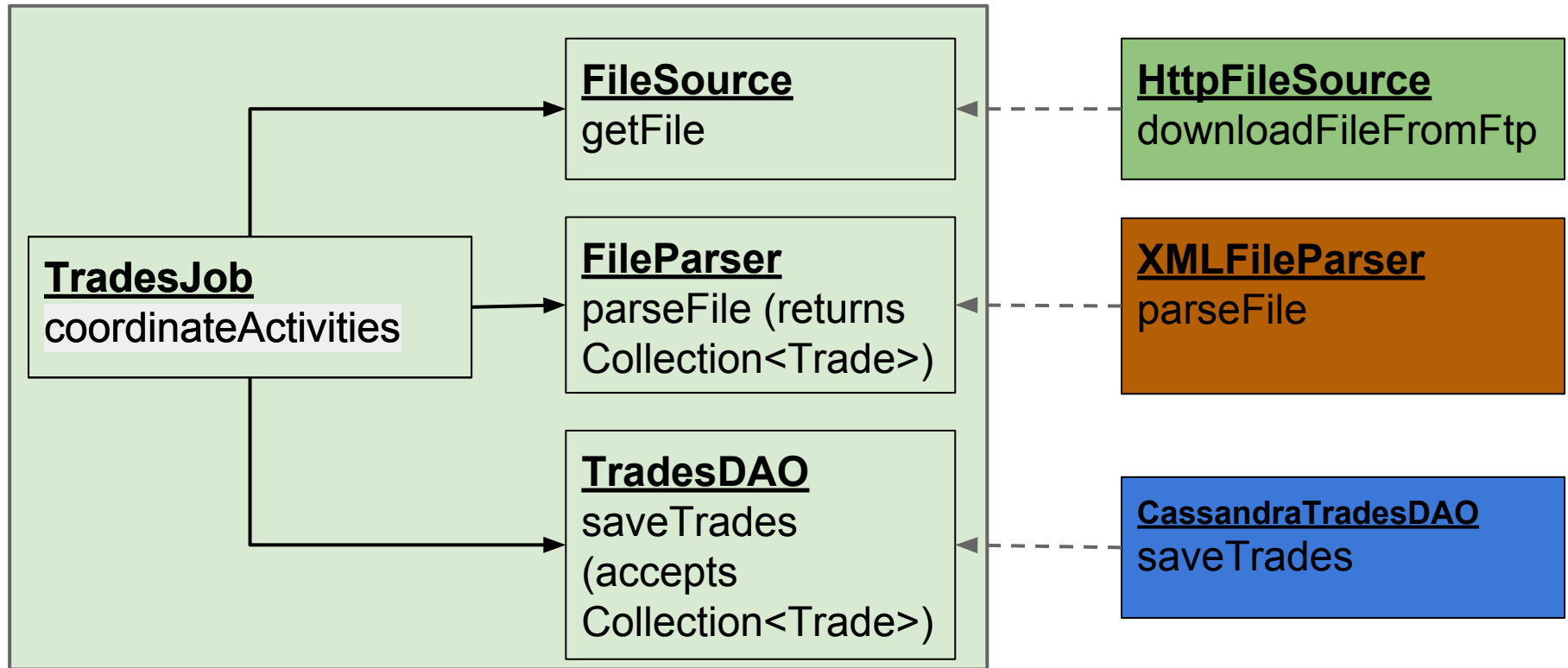
Get loose coupling



Good design is easier to understand



Good design allows change



HIGH COHESION AND LOOSE COUPLING!



YAGNI

You Ain't Gonna Need It

KISS

Keep It Simple, Stupid!

DRY

Don't Repeat Yourself

Additional reading

PodCast with Robert C. Martin on SOLID principles

<http://www.hanselminutes.com/145/solid-principles-with-uncle-bob-robert-c-martin>

SOLID papers by Robert C. Martin

<http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>

Robert C. Martin blog

<http://blog.8thlight.com/uncle-bob/archive.html>

Martin Fowler's website and blog

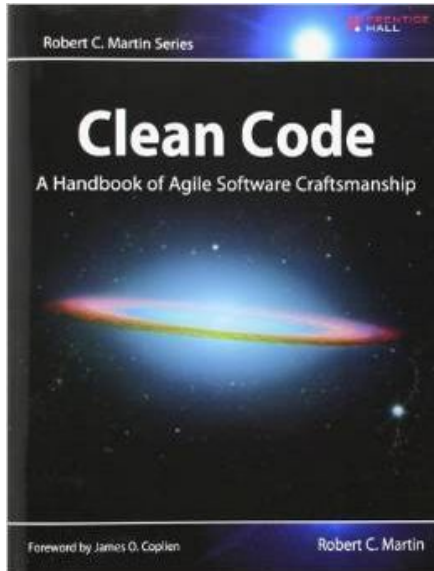
<http://martinfowler.com/>

Lostechies

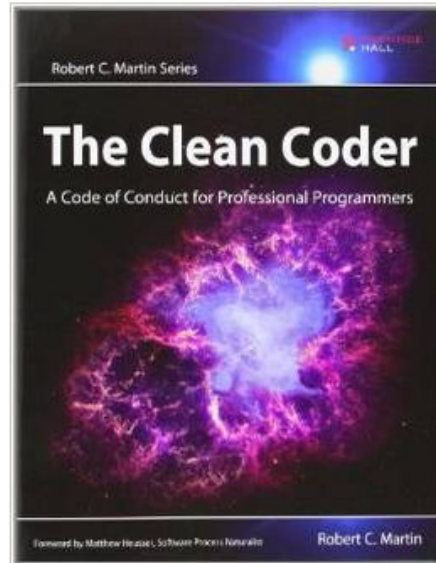
<https://lostechies.com/gabrielschenker/2009/01/21/real-swiss-don-t-need-srp-do-they>

Additional reading

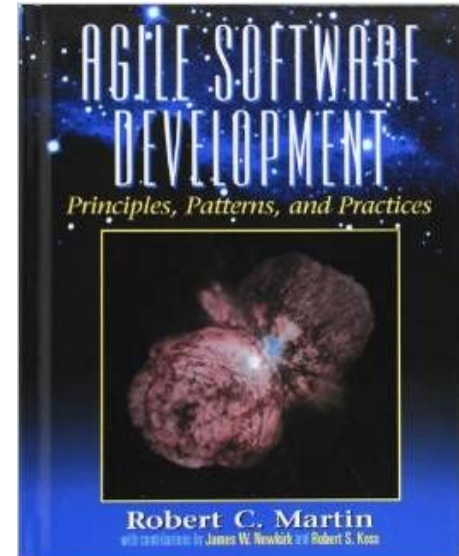
Clean Code
by Robert C. Martin



Clean Coder
by Robert C. Martin



Agile Software Development
by Robert C. Martin



Additional reading

<http://c2.com/cgi/wiki?CouplingAndCohesion>

<http://c2.com/cgi/wiki?YouArentGonnaNeedIt>

http://en.wikipedia.org/wiki/You_aren't_gonna_need_it

http://en.wikipedia.org/wiki/KISS_principle

http://en.wikipedia.org/wiki/Don%27t_repeat_yourself

Sample

git clone <https://agoshkoviv@bitbucket.org/agoshkoviv/solid-example.git>

Homework

git clone <https://agoshkoviv@bitbucket.org/agoshkoviv/solid-homework.git>

Potential changes:

New sources of salary data (other than SQL databases)

New kinds of reports (“min salary report”, “avg salary report”)

New types of report format (pdf/excel/etc)

Publish reports to ftp/local folder (instead of sending via e-mail)

Git for windows

<http://git-scm.com/download/win>