Why loose coupling?

Designing and Maintaining Software (DAMS)

Louis Rose

Habitable Software

Leaner

Less **Complex**

Loosely **Coupled**

More **Cohesive**

Avoids **Duplication**

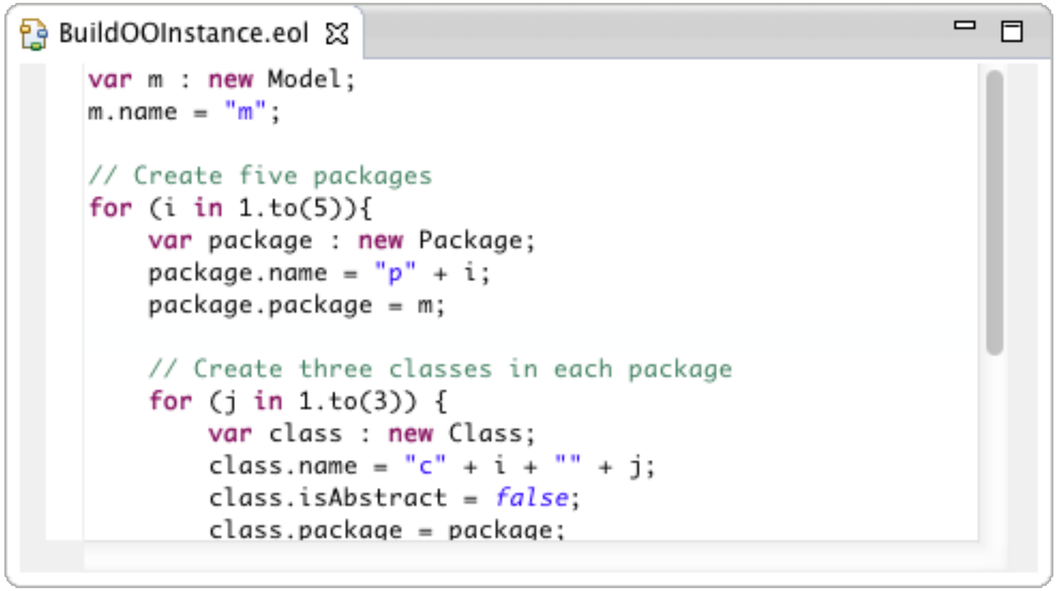
Clearer

More **Extensible**

???

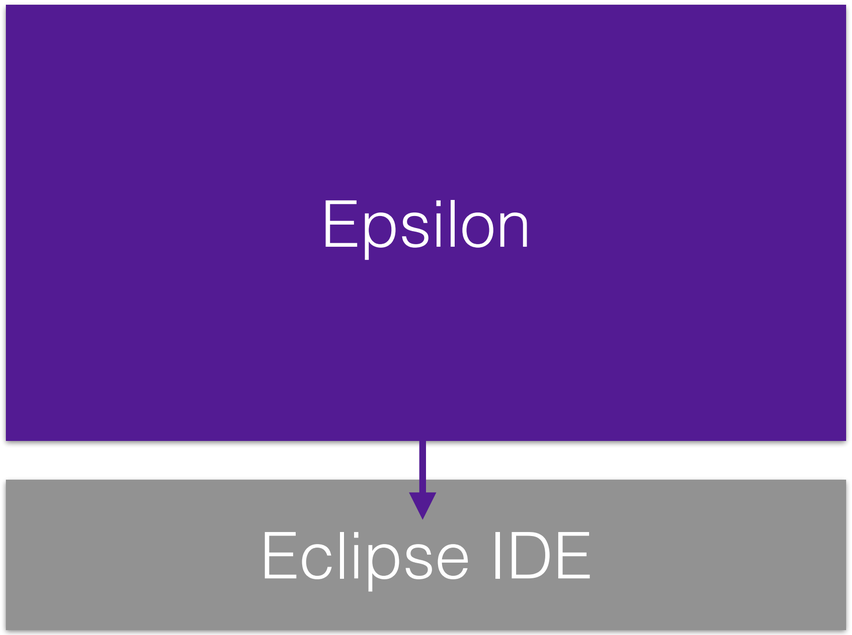
Epsilon

A family of programming languages & developer tools



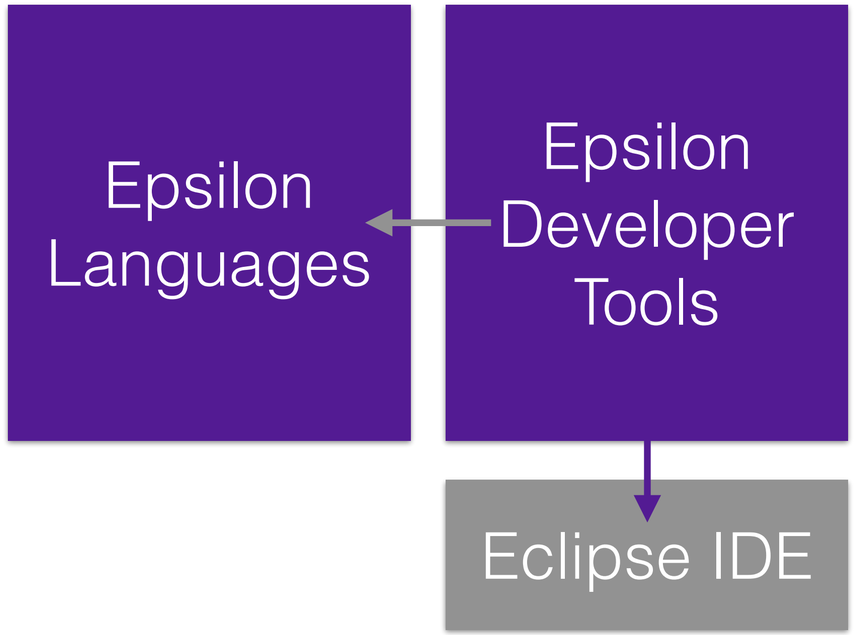
Epsilon

A family of programming languages & developer tools



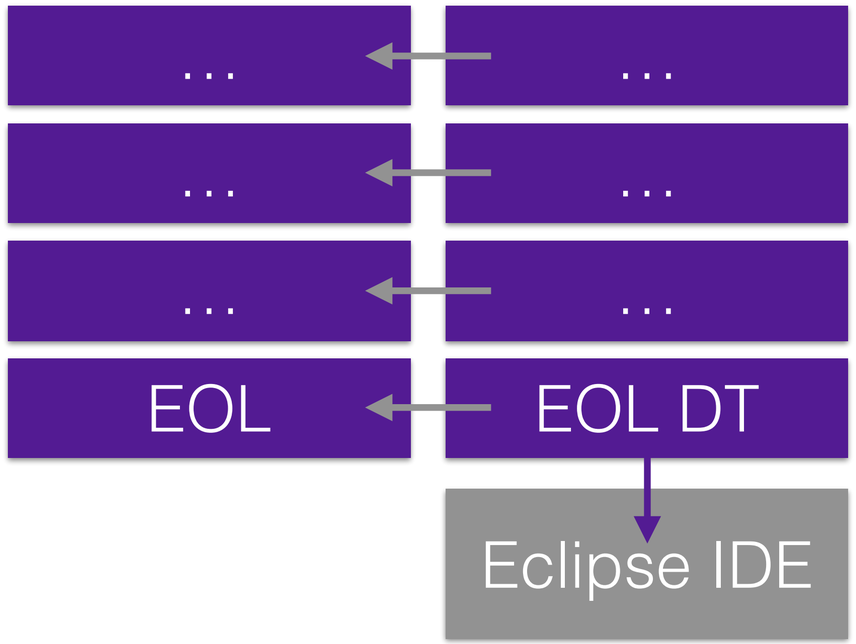
Epsilon

A family of programming languages & developer tools



Epsilon

A family of programming languages & developer tools



Power and Peril

“Because well designed objects have a single responsibility, their very nature requires that they collaborate to accomplish complex tasks. This collaboration is powerful and perilous. To collaborate, an object must know something know about others. Knowing creates a dependency. If not managed carefully, these dependencies will strangle your application.”

- Sandi Metz

http://www.informit.com/articles/article.aspx?p=1946176

Coupling

“The measure of the strength of association established by a connection from one module to another.”

- Stevens, Myers and Constantine
Structured Design
IBM Systems Journal 13:2, 1974

Coupling

*"The measure of the strength of association established by **a connection from one module to another**."*

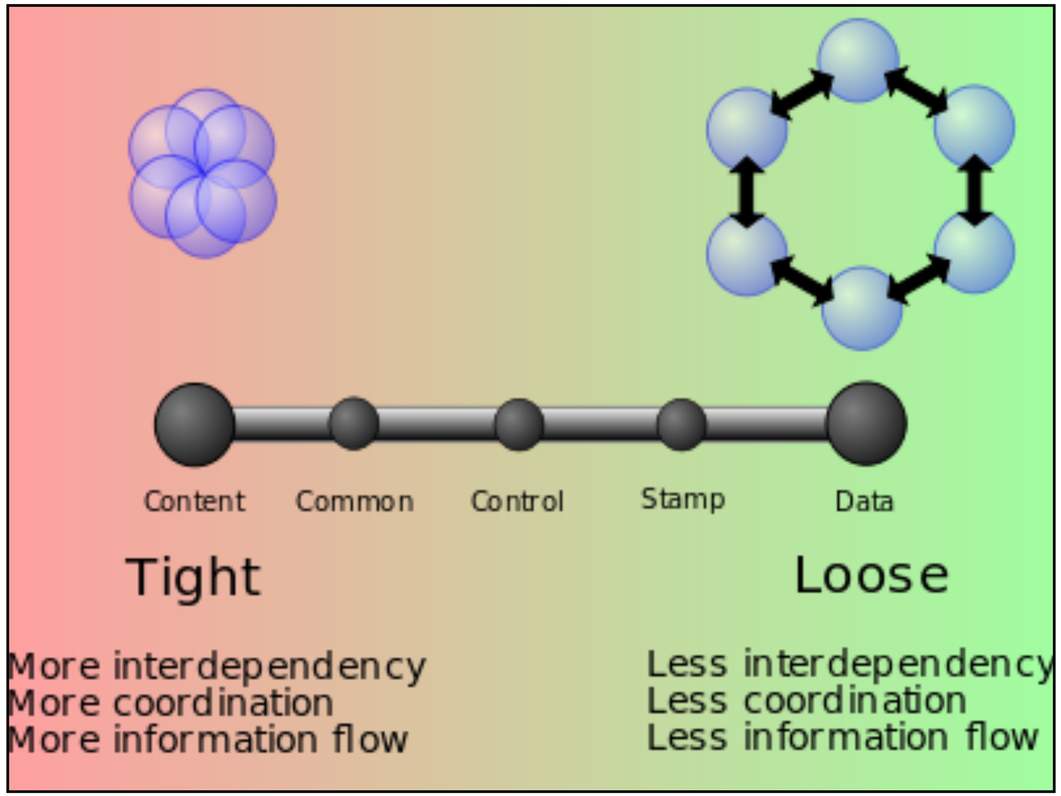
- Stevens, Myers and Constantine
Structured Design
IBM Systems Journal 13:2, 1974

Coupling

*"The measure of the **strength of association** established by a connection from one module to another."*

- Stevens, Myers and Constantine
Structured Design
IBM Systems Journal 13:2, 1974

Coupling is a Spectrum



http://www.ustudy.in/node/7980

Loosely coupled software is…

Flexible: the impact of a change is small

Reasonable: the impact of a change is localised

Mobile: the system is decomposed into reusable parts

- Bob Martin
http://www.objectmentor.com/resources/articles/dip.pdf

DIP

The argument in favour of loosely coupled software

Dependency Inversion Principle

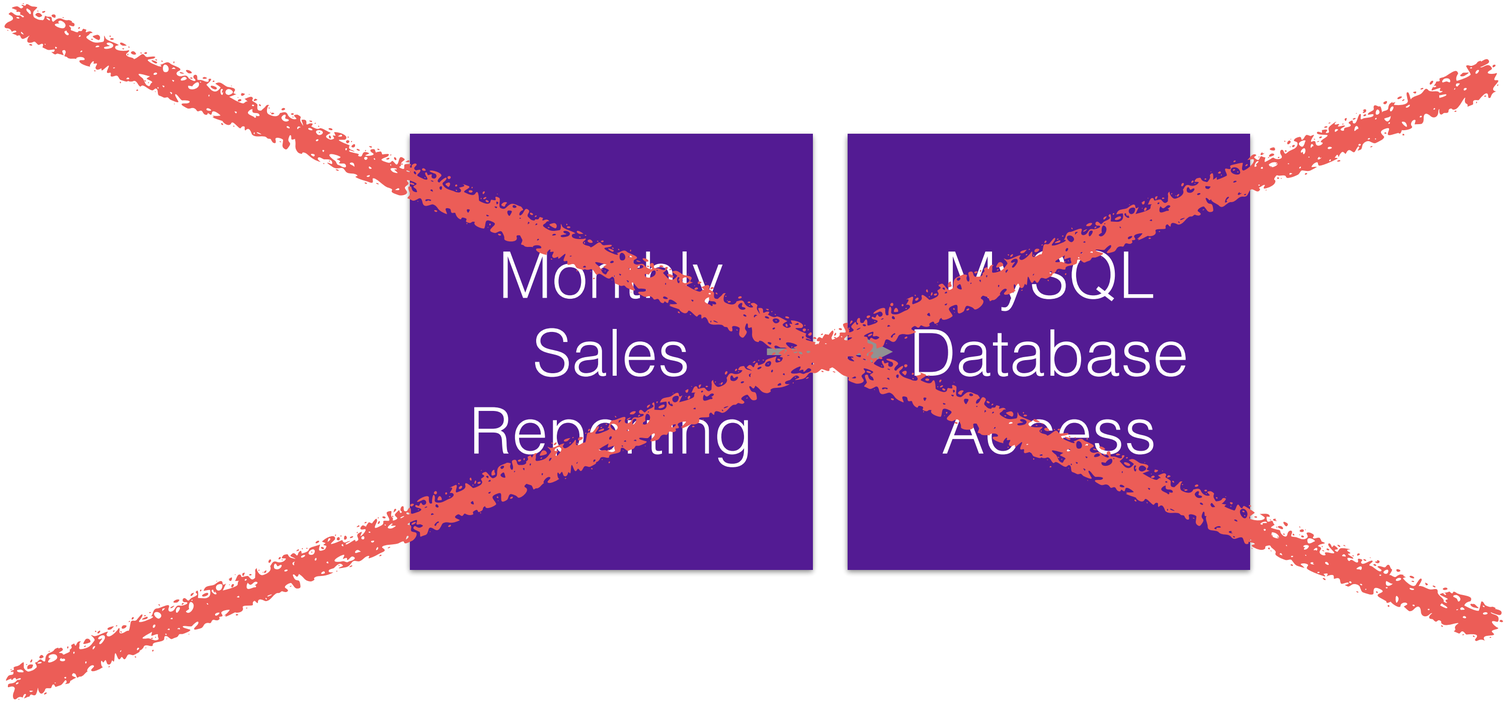
“High-level modules should not depend on low-level modules. Both should depend on abstractions.”

“Abstractions should not depend on details. Details should depend on abstractions.”

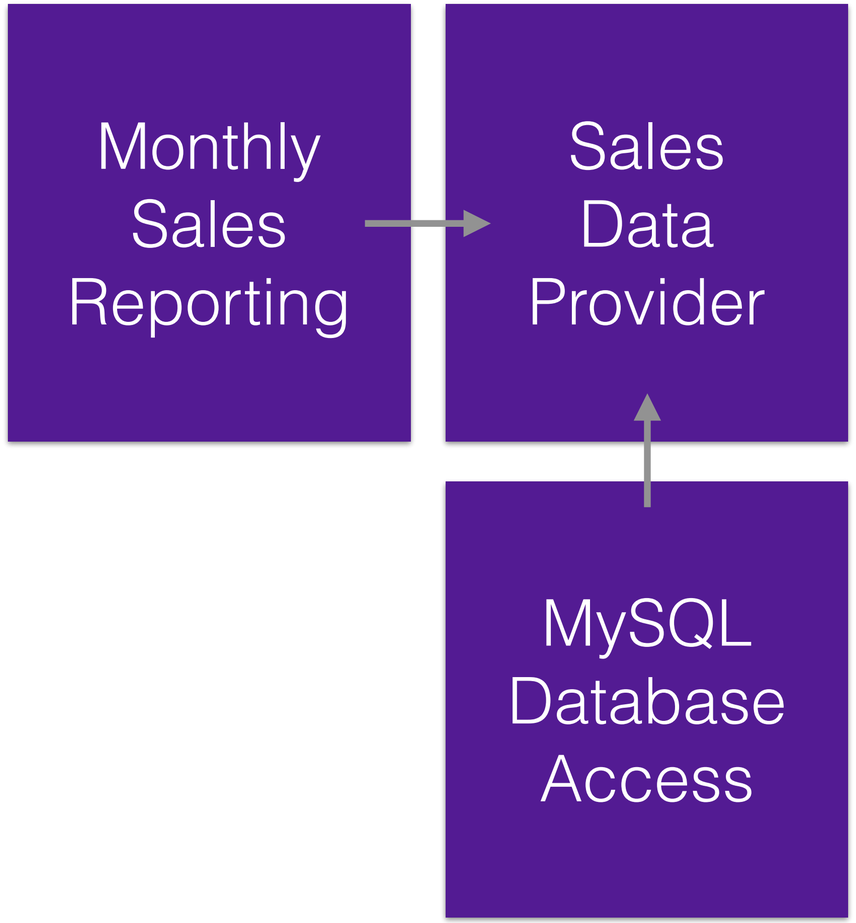
- Bob Martin

http://www.objectmentor.com/resources/articles/dip.pdf

DIP Example



DIP Example


Monthly Sales Reporting
Sales Data Provider
MySQL Database Access

Ignore DIP when…

The low-level details are highly unlikely to change.

The abstraction is the same as the low-level details.

Summary

Coupling is a measure of the extent to
which modules depend on each other

Loose coupling enables flexible,
reasonable and mobile code

DIP tells us to avoid coupling high-level and
low-level details; and to depend on abstractions