

Unit of Work Pattern

A pattern for data access



Why?

Unit of Work Pattern

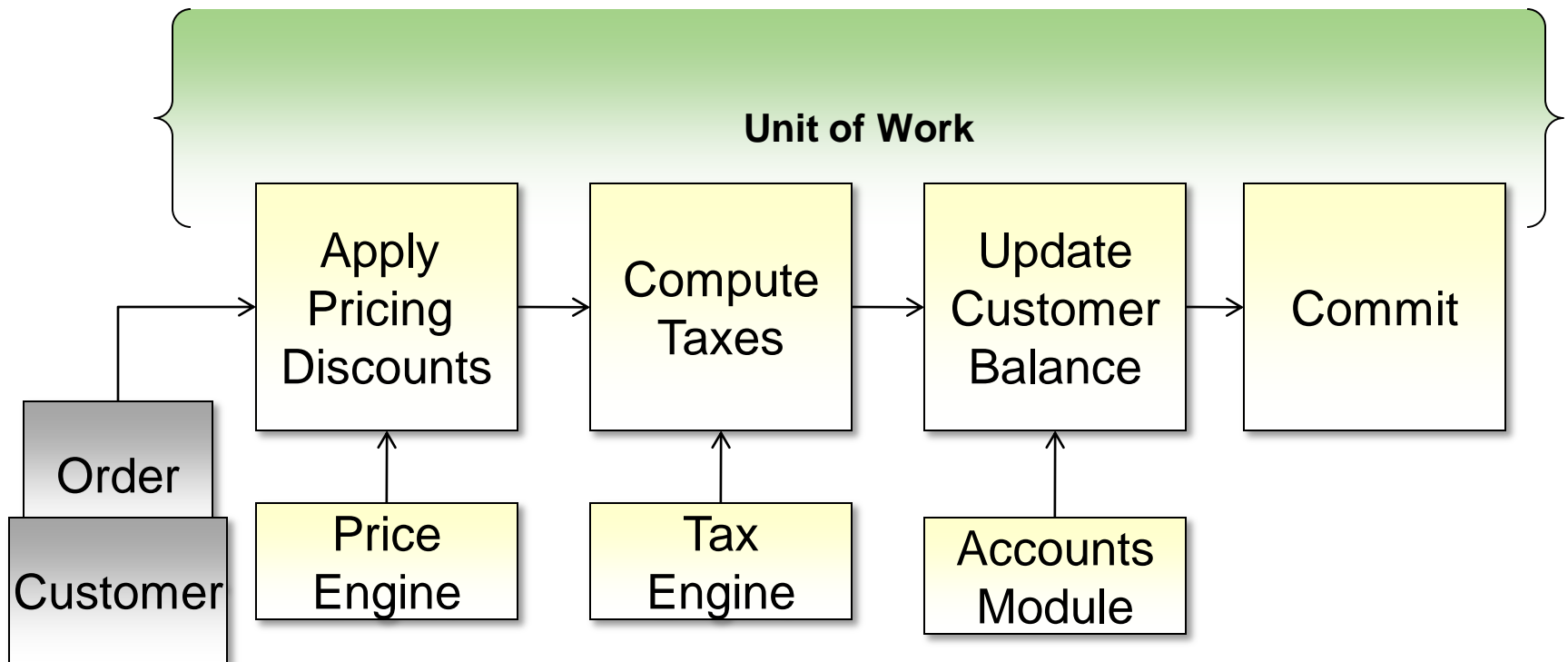
- **Track changes in persistent objects**
 - Efficient data access
 - Manage concurrency problems
 - Manage transactions

```
var employee = repository.FindEmployeeById(id);  
employee.Name = newName;  
unitOfWork.Commit();
```

Intent

**Unit of Work
Pattern**

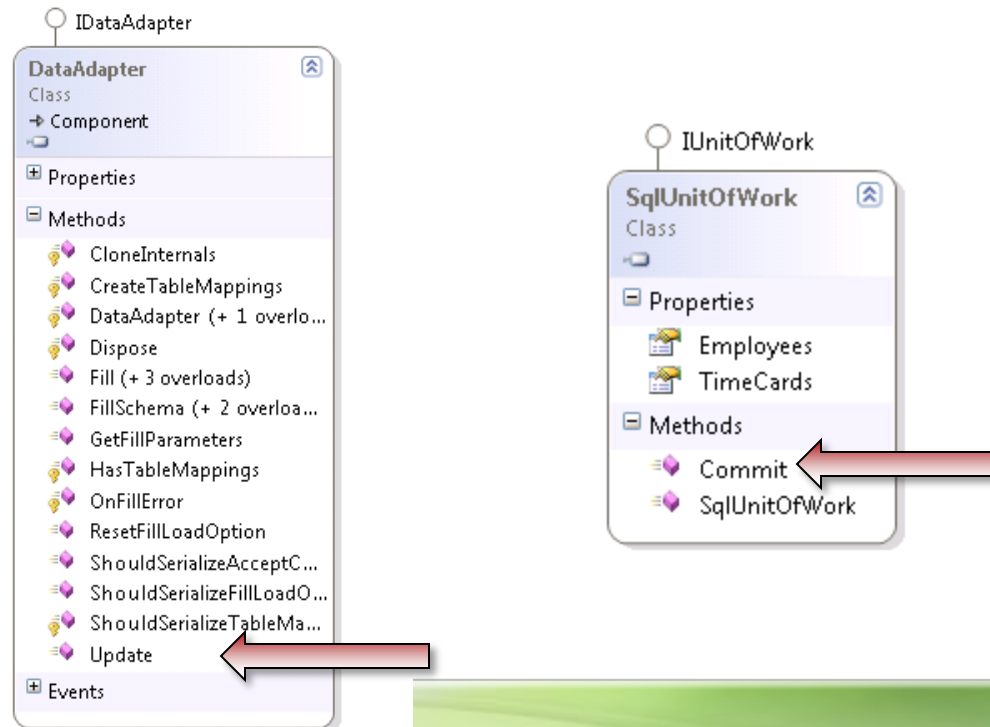
- **Logical transactions**
 - Multiple components contribute work



Applicability

Unit of Work Pattern

- Anytime you need data persistence
- Commonly implemented by persistence frameworks
 - ORMs (NHibernate, Entity Framework)
 - ADO.NET (DataSets & DataAdapters)



Consequences

**Unit of Work
Pattern**

- **Beware of lifetime issues**
 - Singletons are deadly!

Related Patterns

**Unit of Work
Pattern**

- **Repository**
- **Identity Map**

Employees	
ID	Name
1	Scott
2	Paul
3	Poonam

Summary

Unit of Work Pattern

- **Keep business logic free of data access code**
- **Keep business logic free from tracking changes**
- **Allow business logic to work with logical transactions**