

# Loosely Coupled, Domain-Driven Design Architecture

An opinionated graphic Copyright © 2020 Charles L Flatt

## User Interface

How most users interact with the system

Web Browser  
Desktop Application

## Local Application Service Layer

The way specific applications see the business. Aggregated models for easy, stateless data passing.

Note: Javascript frameworks use WebApi. Adapters use Domain Service.

Angular Controller/Service [Angular service calls] getBookEdit(code) [consumes and maps] <b>BookWebApi</b> .GetBookEdit(code)	Desktop Application Adapters [Adapter call] public BookEdit GetBookEdit(code) [consumes and maps] <b>BookDomainService</b> .GetBookEdit(code)	Application Models (View Models) Usually Property bags because serialized/deserialized  BookEdit BookAggregateSales
--	--	---

## Remote Application Service Layer

The way specific applications see the business. Aggregated models for easy, stateless data passing.

Can also be an API for 3rd party consumers.

Application Services/Interfaces WebApi, SOAP, MVC Controllers App-centric, not general APIs  [service/controller call] public BookEdit GetBookEdit(code) [consumes and maps] BookDomainService.GetBook(code)	Application Models (View Models) Usually Property bags because serialized/deserialized  BookEdit AuthorEdit
---	---

## Domain Layer

The way the organization thinks of itself. The domain language. Not normally used by user interfaces.

Not good for stateless data passing. Injectable into application services for unit testing.

MAPS THE DATA MODELS TO THE DOMAIN MODELS

### Domain Services/Interfaces

Coordinator, general APIs  
Map Data Models to Domain Models  
Enforce domain-level business rules  
and validation

[service call]  
public Book GetBook( code)  
[consumes and maps]  
dbEnterprise.Book\_Main.Where(...)  
mongoClient.Authors.Where(...)  
upsClient.GetShipments(...)

### Domain Models

RICH ENTITY OBJECTS  
Identity  
Equality by reference  
containing  
Properties,  
Behaviors,  
Business Rules,  
Validation  
  
Book  
Author

VALUE OBJECTS  
No identity  
Equality by state  
Cheap to pass around  
  
Pages  
Price

### 3rd Party Services

Services the domain may depend on  
to return its models. Injectable into domain service  
for unit testing

UpsRest Service

## Data Layer

Not used by application services. The abstract way of interacting with a single  
storage resource. Injectable into domain services for unit testing.

### Data Services/Interfaces

Entity Framework,  
MongoClient,  
ADO.Net with Repository Pattern  
etc.

### Data Models

Property bags (DTOs)  
No (or limited) behavior  
  
Book\_Main  
PubSource

## Persistence

The physical sources for data. Potentially accessible in multiple ways

Persistent Storage  
Oracle, MongoDB, Files, etc.