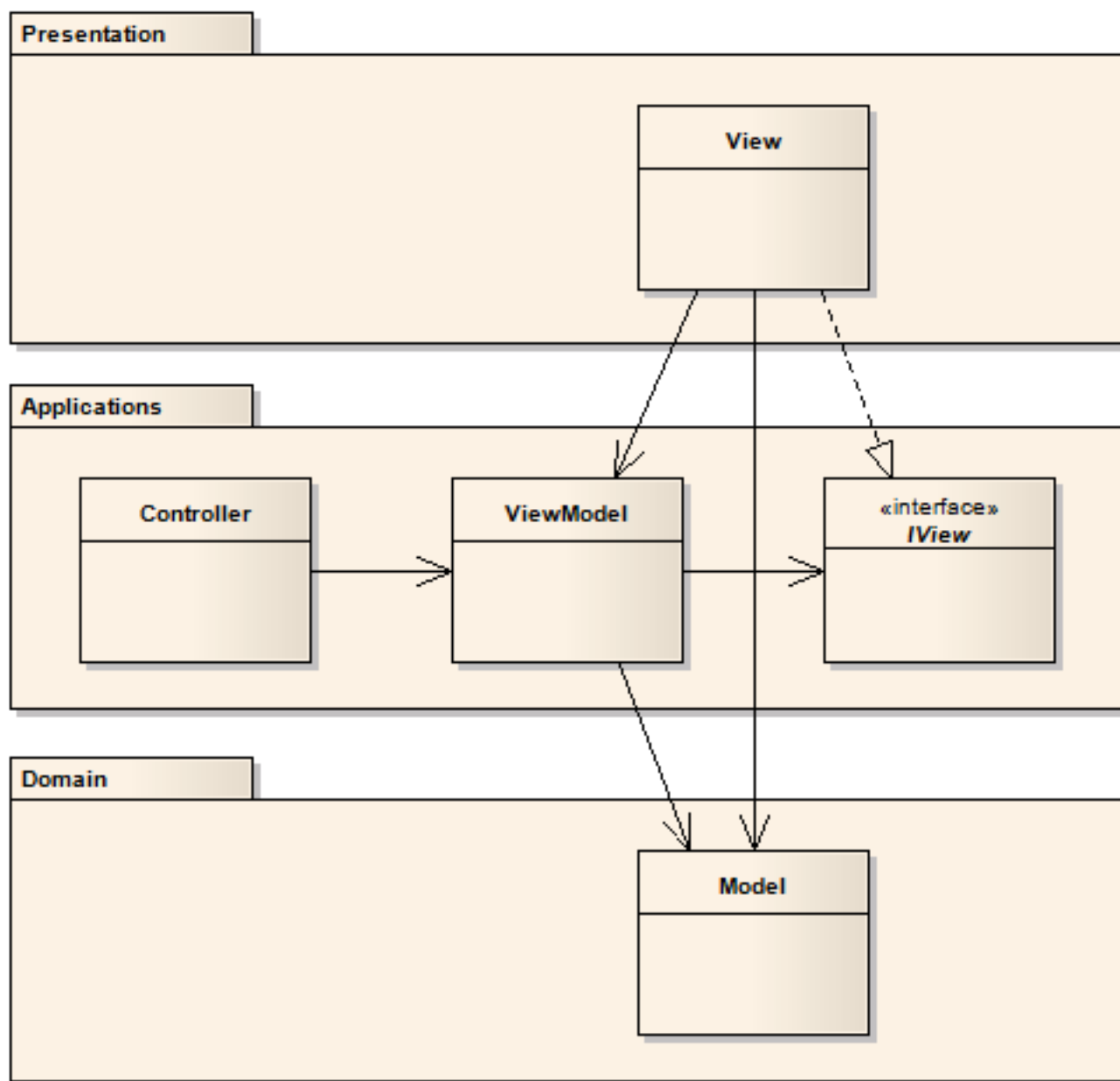


دز طراحی پروژه‌های مقیاس بزرگ و البته به صورت ماژولار همیشه ساختار پروژه اهمیت به سزایی دارد. متأسفانه این مورد خیلی در طراحی پروژه‌ها در نظر گرفته نمی‌شود و اغلب اوقات شاهد آن هستیم که یک پروژه بسیار بزرگ دقیقاً به همان صورت پروژه‌های کوچک و کم اهمیت‌تر مدیریت و پیاده سازی می‌شود که این مورد هم مربوط به پروژه‌های تحت وب و هم پروژه‌های تحت ویندوز و WPF است. برای مدیریت پروژه‌های WPF و Silverlight در این [پست](#) به اختصار درباره PRISM بحث شد. مزایا و معایب آن بررسی و در طی این پست‌ها ([^](#) و [^](#)) مثال هایی را پیاده سازی کردیم. اما در این پست مفتخرم شما را با یکی دیگر از کتابخانه‌های مربوط به پیاده سازی مدل MVVM آشنا کنم. کتابخانه ای متن باز، بسیار سبک با کارایی بالا. اما نکته ای که ذکر آن خالی از لطف نیست این است که قبلاً از این کتابخانه در یک پروژه بزرگ و ماژولار WPF استفاده کردم و نتیجه مطلوب نیز حاصل شد.

معرفی:

WPF Application Framework یا به اختصار WAF کتابخانه کم حجم سبک و البته با کارایی عالی برای طراحی پروژه‌های ماژولار WPF در مقیاس بزرگ طراحی شده است که مدل پیاده سازی آن بر مبنای مدل MVVM و MVC است. شاید برایتان جالب باشد که این کتابخانه دقیقاً مدل MVC را با مدل MVVM ترکیب کرده در نتیجه مفاهیم آن بسیار شبیه به پروژه‌های تحت وب MVC است. همانطور که از نام آن پیداست این کتابخانه صرفاً برای پروژه‌های WPF طراحی شده، در نتیجه در پروژه‌های Silverlight نمی‌توان از آن استفاده کرد.

ساختار کلی آن به شکل زیر می‌باشد:



همانطور که مشاهده می‌کنید پروژه‌های مبتنی بر این کتابخانه همانند سایر کتابخانه‌های MVVM از سه بخش تشکیل شده‌اند. بخش اول با عنوان Shell یا Presentation معرف فایل‌های Xaml پروژه است، بخش دوم یا Application معرف ViewModel و Controller و البته IView می‌باشد. بخش Domain نیز در برگیرنده مدل‌های برنامه است.

معرفی برخی مفاهیم:

«Shell»: این کلاس معادل یک فایل Xaml است که حتما باید یک اینترفیس IView را پیاده سازی نماید.

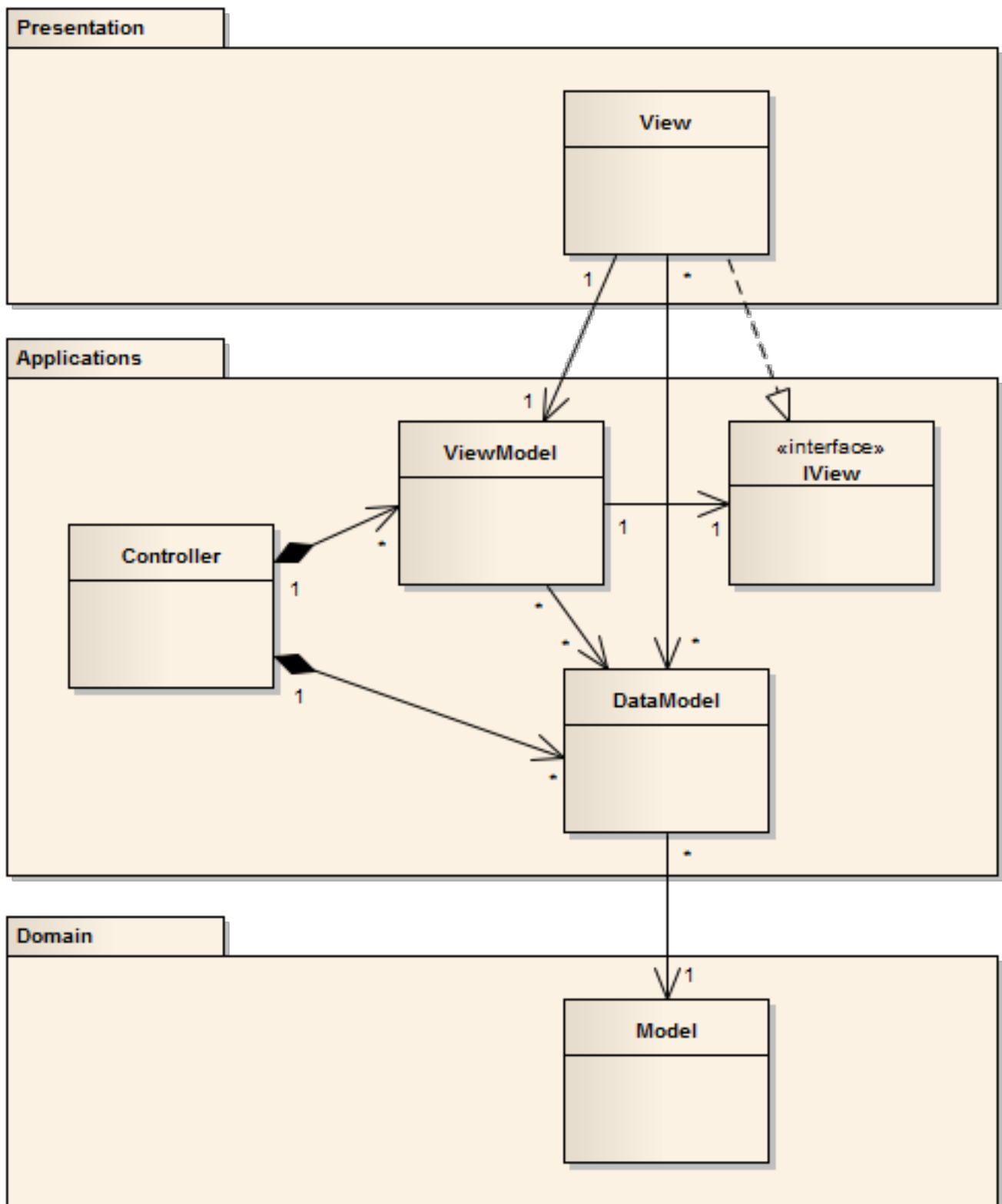
«IView»: معرف یک اینترفیس جهت برقراری ارتباط بین ViewModel و Shell

«ViewModel»: در این جا ViewModel با مفهوم ViewModel در سایر کتابخانه‌های MVVM کمی متفاوت است. در این کتابخانه ViewModel فقط شامل تعاریف است و هیچ گونه پیاده سازی در اینجا صورت نمی‌گیرد. دقیقا معادل مفهوم ViewModel در پروژه‌های MVC تحت وب.

«Controller»: پیاده سازی ViewModel و تعریف رفتارها در این قسمت انجام می‌گیرد.

اما در بسیاری از پروژه‌ها نیاز به پیاده سازی الگوی DataModel-View-ViewModel است که این کتابخانه با در اختیار داشتن برخی

کلاس‌های پایه این مهم را برایمان میسر کرده است.



همانطور که می‌بینید در این حالت بر خلاف حالت قبلی ViewModel و کنترلرهای پروژه به جای ارتباط با مدل با مفهوم DataModel تغذیه می‌شوند که یک پیاده‌سازی سفارشی از مدل‌های پروژه است. هم چنین این کتابخانه یک سری Converterهای سفارشی

جهت تبدیل Model به DataModel و برعکس را ارائه می‌دهد. سرویس‌های پیش فرض: که شامل DialogBox جهت نمایش پیام‌ها و Save|Open File Dialog سفارشی نیز می‌باشد. «برای پیاده سازی Modularity از کتابخانه MEF استفاده شده است. Command های سفارشی: پیاده سازی خاص از اینترفیس ICommand «مفاهیم مربوط به [Weak Event Pattern](#) به صورت توکار در این کتابخانه تعبیه شده است. «به صورت پیش فرض مباحث مربوط به اعتبارسنجی با استفاده از [DataAnnotation](#) و [IDataErrorInfo](#) در این کتابخانه تعبیه شده است. «ارائه Extension های مربوط به UnitTest نظیر Exceptions و CanExecuteChangedEvent و PopertyChanged جهت سهولت در تهیه unit test

دانلود و نصب

با استفاده از nuget و دستور زیر می‌توانید این کتابخانه را نصب نمایید:

```
Install-Package waf
```

هم چنین می‌توانید سورس آن به همراه فایل‌های باینری را از [اینجا](#) دریافت کنید. در پست بعدی یک نمونه از پیاده سازی مثال با این کتابخانه را بررسی خواهیم کرد.