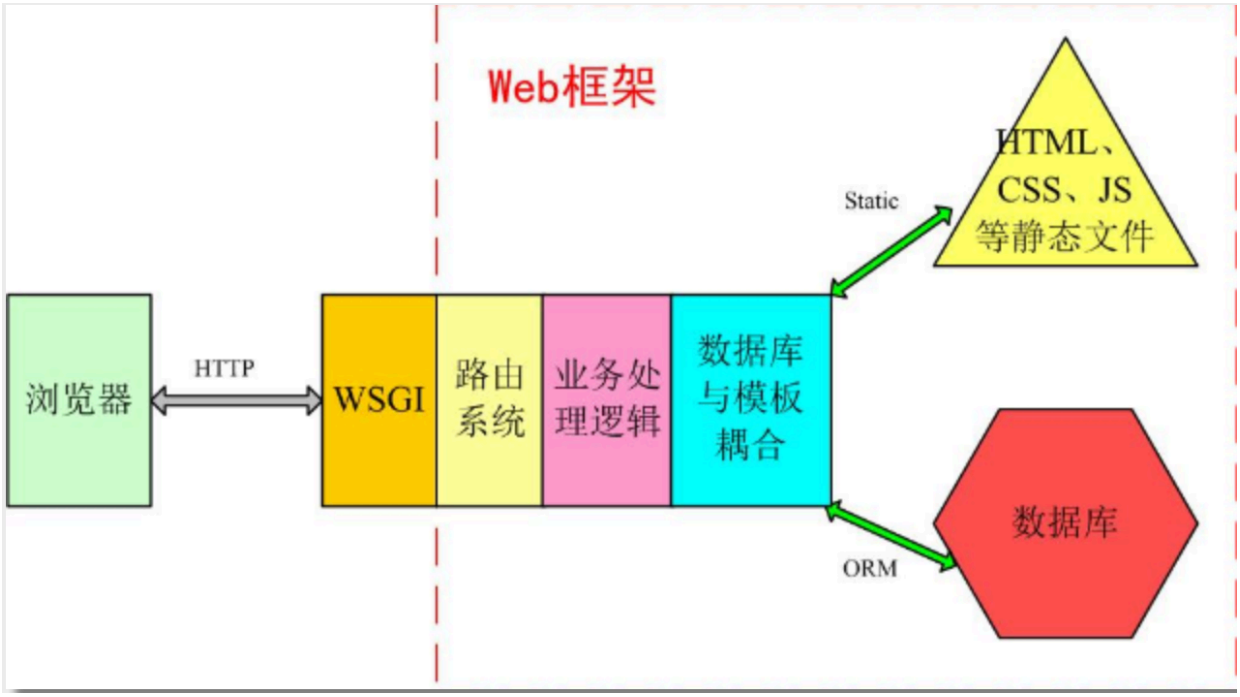
什么是Django

web开发框架

Web框架致力于解决一些共同的问题，为Web应用提供通用的架构，让用户专注于网站应用业务逻辑的开发，而无须处理网络应用底层的协议、线程、进程等方面的问题。这样能大大提高开发者的效率和Web应用程序的质量。

一般Web框架的架构是这样的：



大多数基于Python的web框架，如Django、tornado、flask、webpy都是在这个范围内进行增删裁剪的。例如Tornado用的是自己的异步非阻塞“WSGI”网关接口，Flask则只提供了最精简和基本的框架，Django则是直接使用了现成的WSGI，并实现了大部分功能，提供了大量的应用工具。

Django

Django是一个由Python编写的具有完整架站能力的开源Web框架。使用Django，只要很少的代码，Python的程序开发人员就可以轻松地完成一个正式网站所需要的大部分内容，并进一步开发出全功能的Web服务。

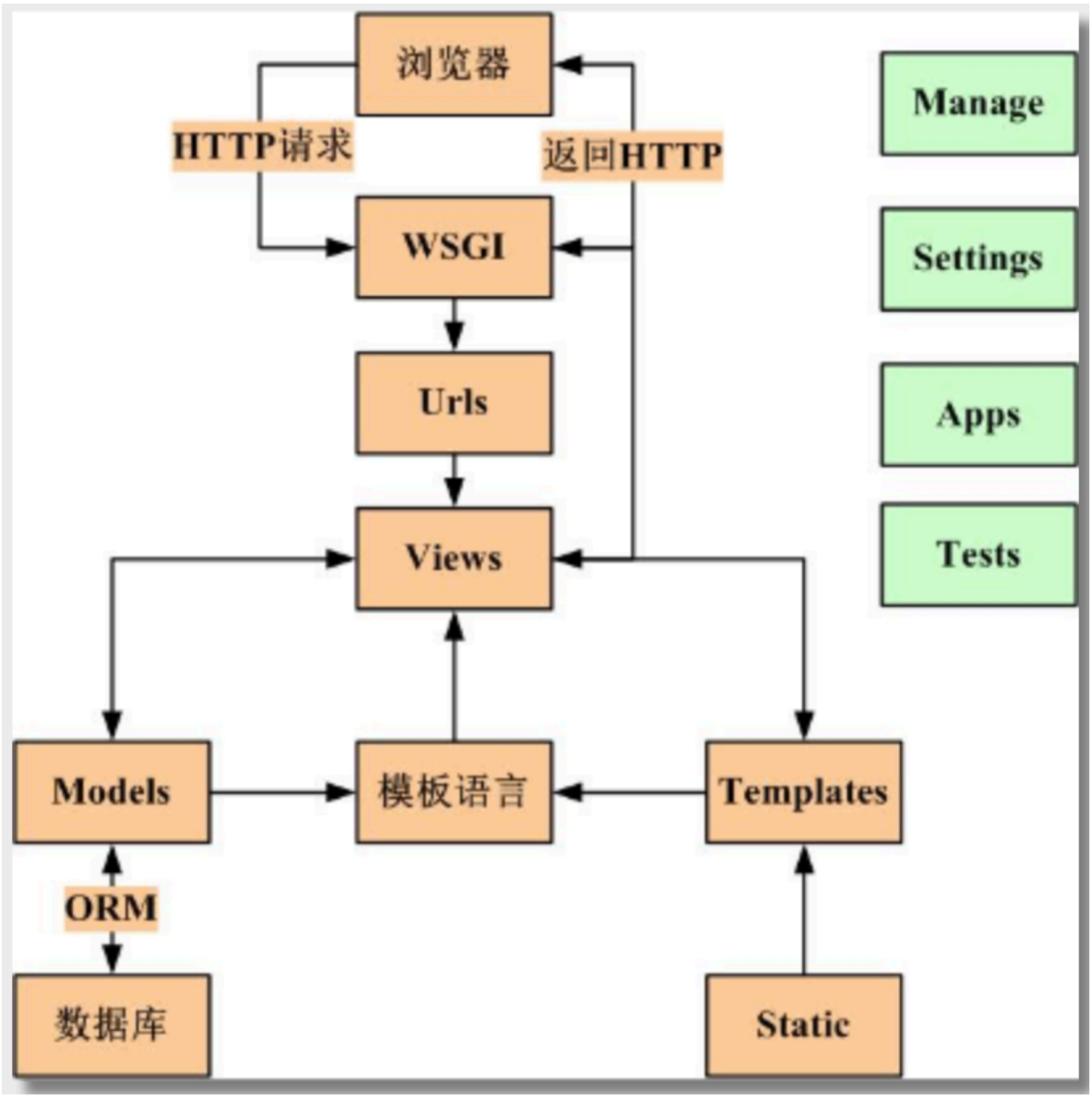
Django本身基于MVC模型，即Model（模型）+View（视图）+ Controller（控制器）设计模式，因此天然具有MVC的出色基因：开发快捷、部署方便、可重用性高、维护成本低等。Python加Django是快速开发、设计、部署网站的最佳组合。

Django对传统的MVC设计模式进行了修改，将视图分成View模块和Template模块两部分，将动态的逻辑处理与静态的页面展现分离开。而Model采用了ORM技术，将关系型数据库表抽象成面向对象的Python类，将表操作转换成类操作，避免了复杂的SQL语句编写。MTV和MVC本质上是一样的。

模型(Model)：和MVC中的定义一样

模板(Template)：将数据与HTML语言结合起来的引擎

视图(View)：负责实际的业务逻辑实现



新建项目

进入你指定的项目保存目录，然后运行下面的命令：

```
$ django-admin startproject mysite
```

这将在目录下生成一个mysite目录，也就是你的这个Django项目的根目录。它包含了一系列自动生成的目录和文件，具备各自专有的用途。

一个新建立的项目结构大概如下：

```
mysite/
    manage.py
    mysite/
        __init__.py
        settings.py
        urls.py
        wsgi.py
```

各文件和目录解释：

- 外层的`mysite/`目录与Django无关，只是你项目的容器，可以任意命名。
- `manage.py`：一个命令行工具，用于与Django进行不同方式的交互脚本，非常重要！
- 内层的`mysite/`目录是真正的项目文件包裹目录，它的名字是你引用内部文件的包名，例如：`mysite.urls`。
- `mysite/__init__.py`:一个定义包的空文件。
- `mysite/settings.py`:项目的主配置文件，非常重要！
- `mysite/urls.py`:路由文件，所有的任务都是从这里开始分配，相当于Django驱动站点的内容表格，非常重要！
- `mysite/wsgi.py`:一个基于WSGI的web服务器进入点，提供底层的网络通信功能，通常不用关心。