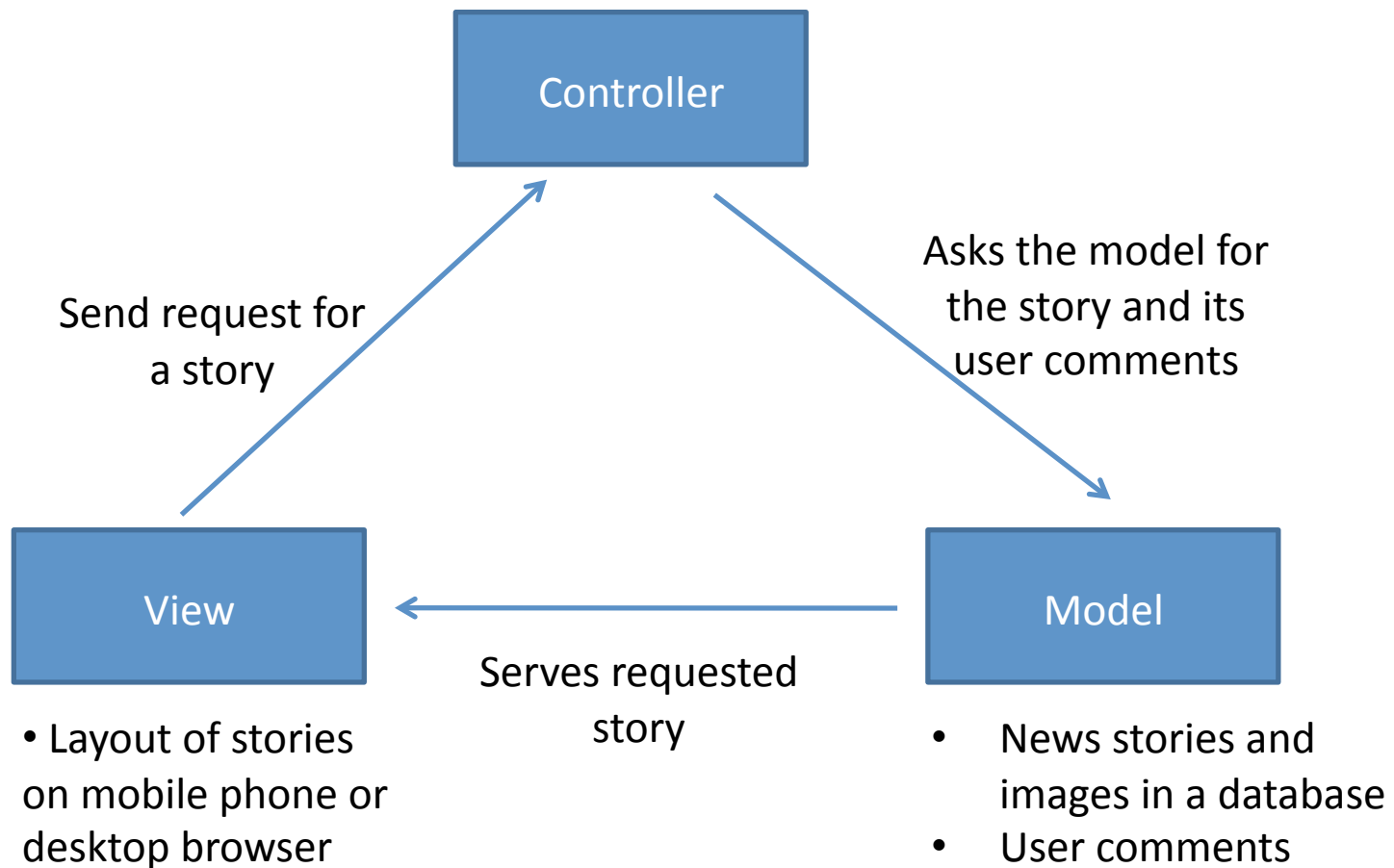


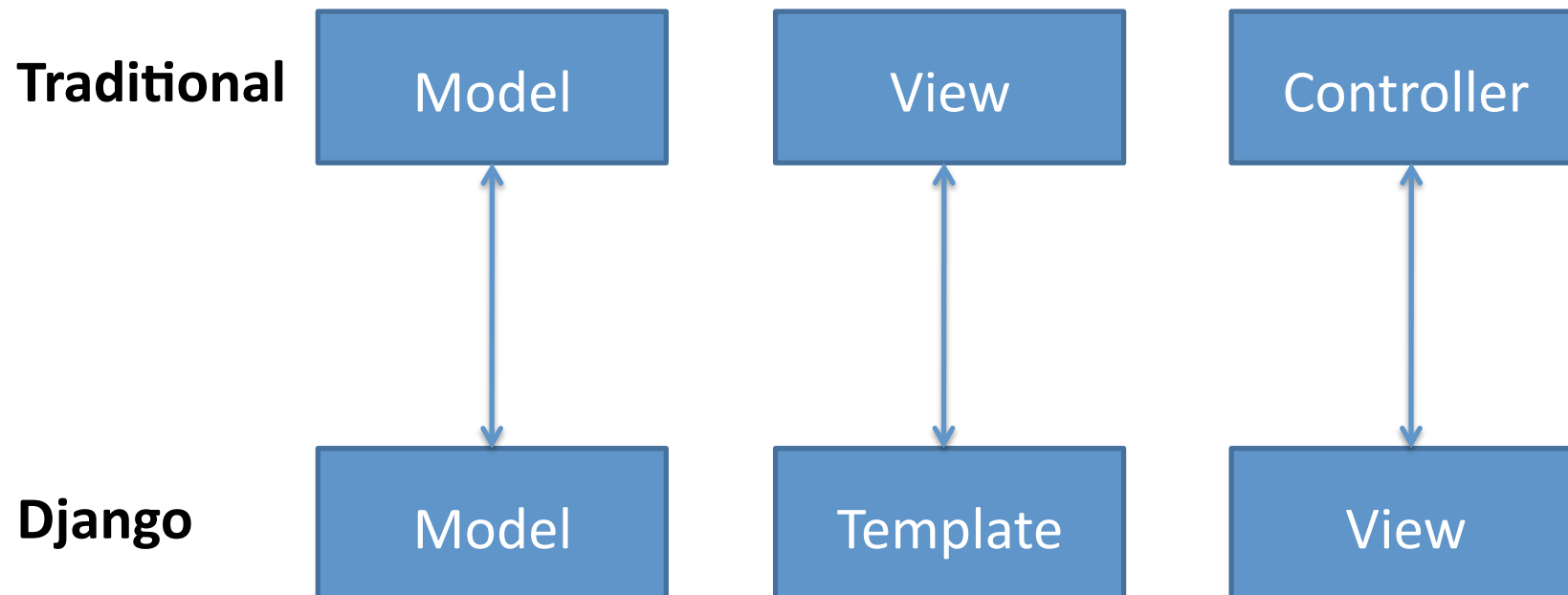


Django Lecture 4: Views

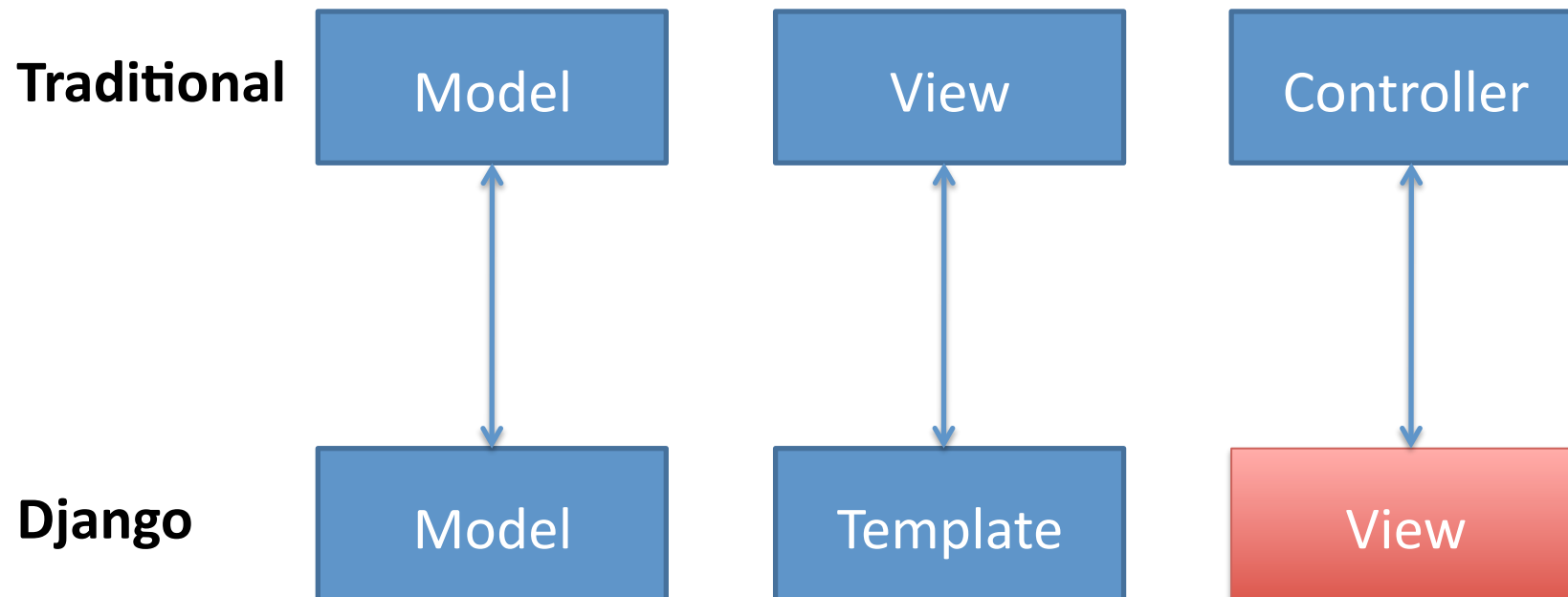
Model-View-Controller (MVC) (news site example)



Django Architecture: Model-Template-View (MTV)



Django Architecture: Model-Template-View (MTV)



Views

- Views are the logical interface between data (Models) and presentation (Templates)
- Defined in **views.py** inside the *appFolder*
- EVERY view takes a request object as first parameter
- EVERY view returns an **HttpResponse** object

```
from django.http import HttpResponse
def hello(request):
    return HttpResponse("Hello world")
```

Our `views.py`

```
from django.template import Context, loader
from django.http import HttpResponse
from models import Musician

def musicians_list(request):
    musician_list = Musician.objects.all()
    t = loader.get_template('musicians/list.html')
    c = Context({'musician_list': musician_list,})
    return HttpResponse(t.render(c))
```

Use `urls.py` to direct HTTP requests to views

```
# urls.py in musicSite
from django.conf.urls.defaults import *
from django.contrib import admin

urlpatterns = patterns('',
    url(r'^admin/', 'admin.site.urls'),
    url(r'^musicians/', 'musicSite.musicians.urls'))
```

Use `urls.py` to direct HTTP requests to views

```
# urls.py in musicians
from django.conf.urls.defaults import *

urlpatterns = patterns('',
    url(r'^list/', 'musicians.views.musician_list'),
    url(r'^detail/(?P<id>\d+)/$',
        'musicians.views.musician_detail'))
```


Dynamic URLs

```
...
urlpatterns = patterns('',
    ...
    url(r'^detail/(?P<id>\d+)/$',
        'musicians.views.musician_detail'),
    url(r'^albumlist/(?P<year>\d{4})/$',
        'musicians.views.album_list'))
```

- URLs can contain regular expressions
- Parameters from URLs can be captured and used as arguments in views

Dynamic URLs: `view.py`

...

```
def musician_detail(request, id):  
    m = Musician.objects.get(pk=id)  
    t = loader.get_template('musician/detail.html')  
    c = Context({'musician' : m})  
    return HttpResponse(t.render(c))
```

Parameters from passed from URL can be used by the view for dynamic page generation

Request Life Cycle

1. User requests to view URL
2. Django determines the root URLconf by looking at the `ROOT_URLCONF` setting.
3. Django looks at all of the URLpatterns in the URLconf for the *first* one that matches the URL
4. If it finds a match, it calls the associated view function.
 - Repeats Steps 3-4 if redirected to another `urls.py`
5. The view function returns an `HttpResponse`.
6. Django converts the `HttpResponse` to the proper HTTP response, which results in a Web page.