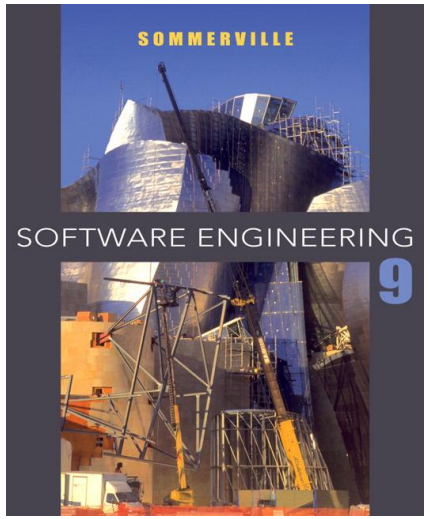
Architectural Design



Chapter 6.0, 6.1, 6.3, 6.4

What is architectural design?

- What is software architecture?
  - Organization, structure

- What is the output of architectural design?
  - Components
  - The interaction among components

Why should we do architectural design?

- We need it even in agile process
  - Can refactoring solve all bad design?

- Non-functional requirements can be affected by architectural design
  - What type of design can improve maintainability?
  - What … can improve availability?
  - What … can improve performance?

Why should we do architectural design?

- We need it even in agile process
  - Refactoring does not help/change architectural design

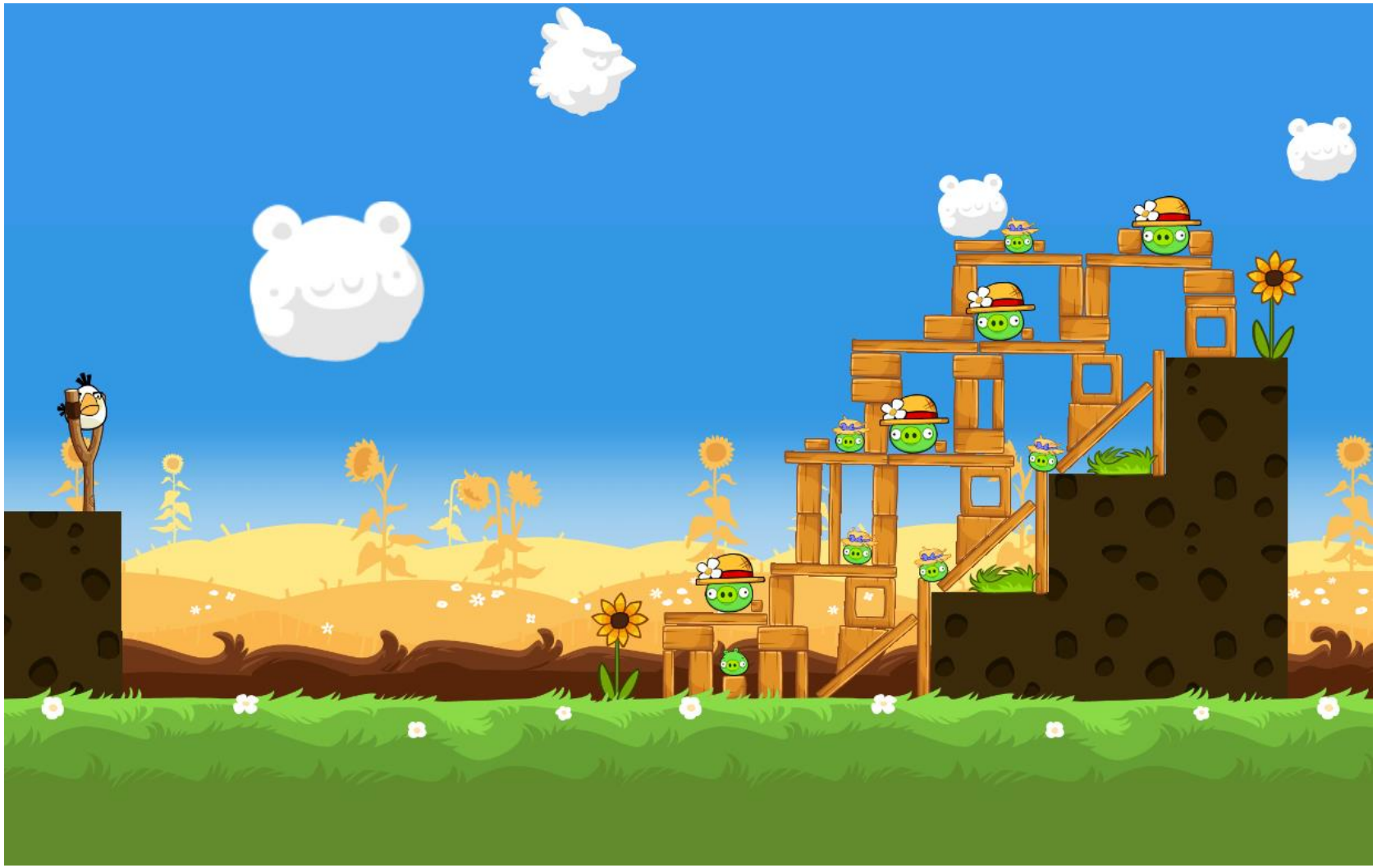
- Non-functional requirements can be affected by architectural design
  - Maintainability
    - The less interaction among components, the better
  - Availability
    - Redundant component can be used to improve availability
  - Performance
    - Be careful about communication overhead across components

How to design?

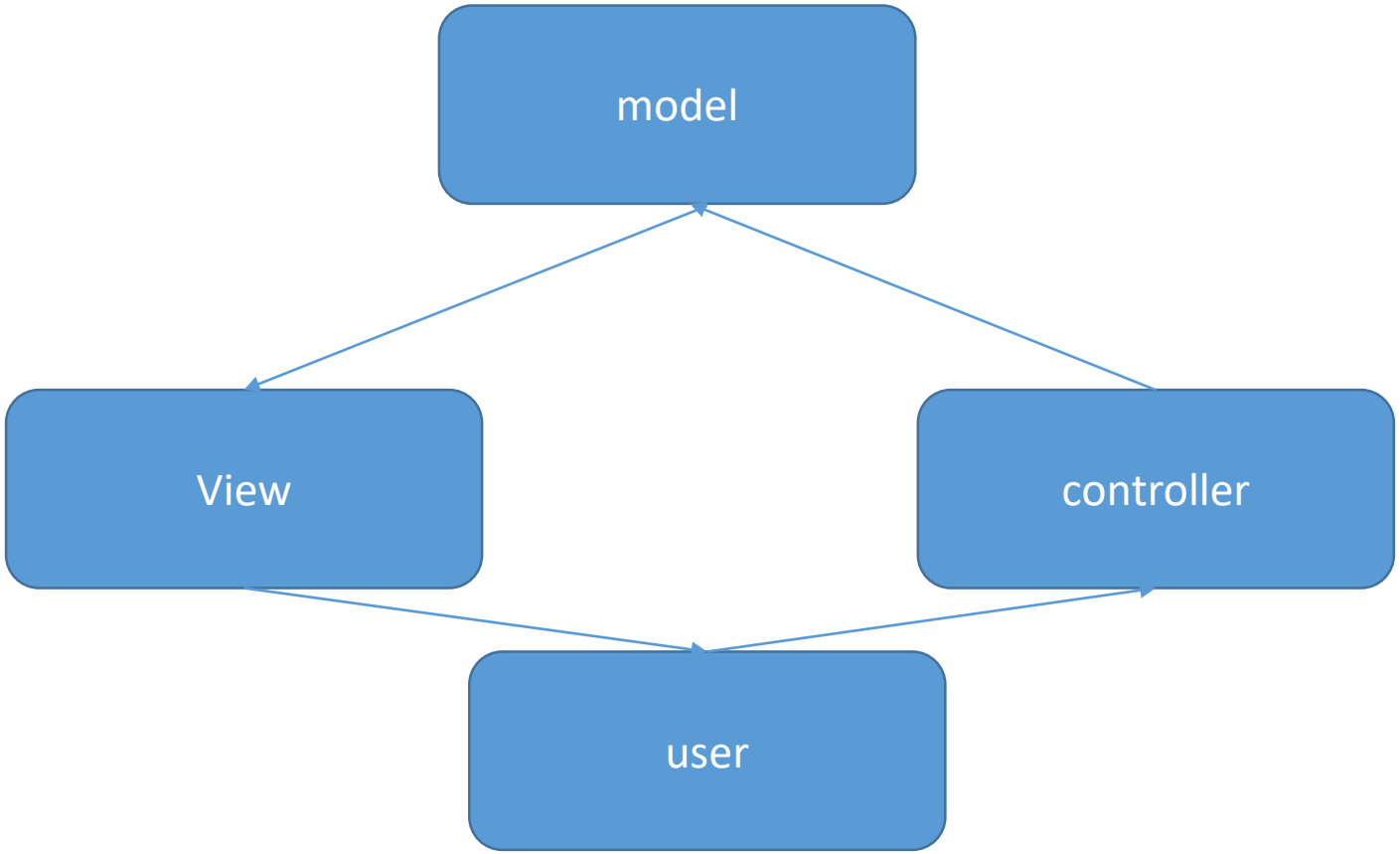
- Follow the requirement and modeling
  - Architectural design can be conducted right after use-case study

- Architecture styles/patterns

What is the architecture of …



MVC pattern



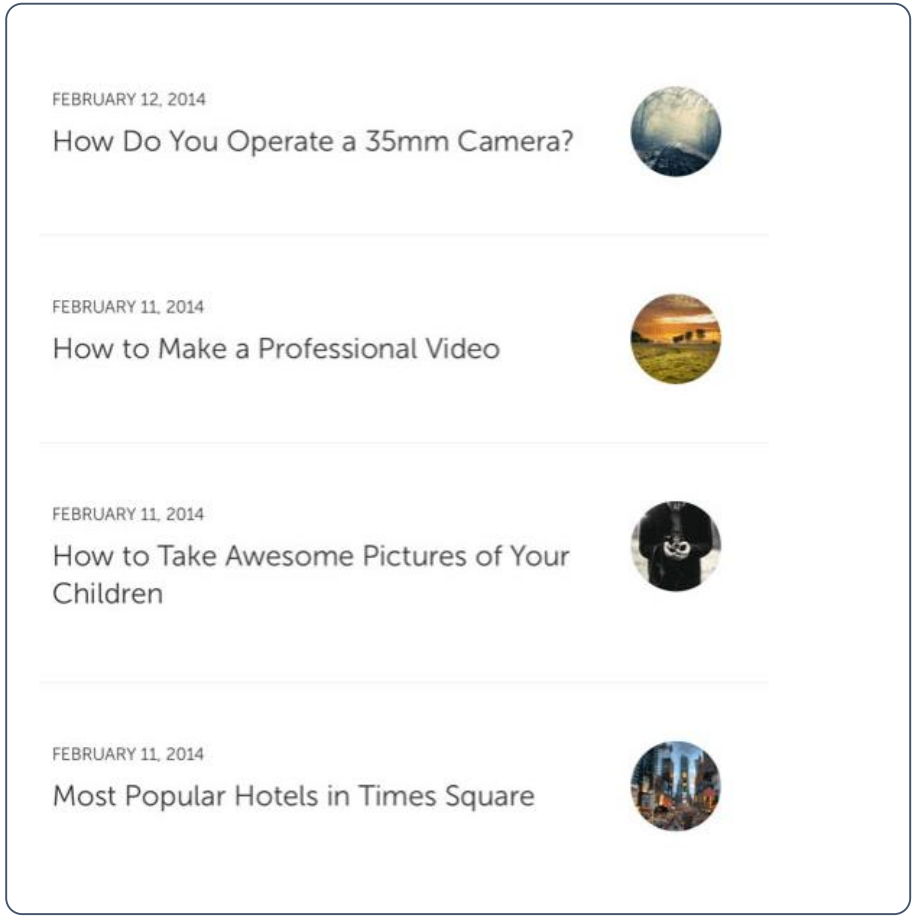
MVC examples

- Games
- Web applications
- …

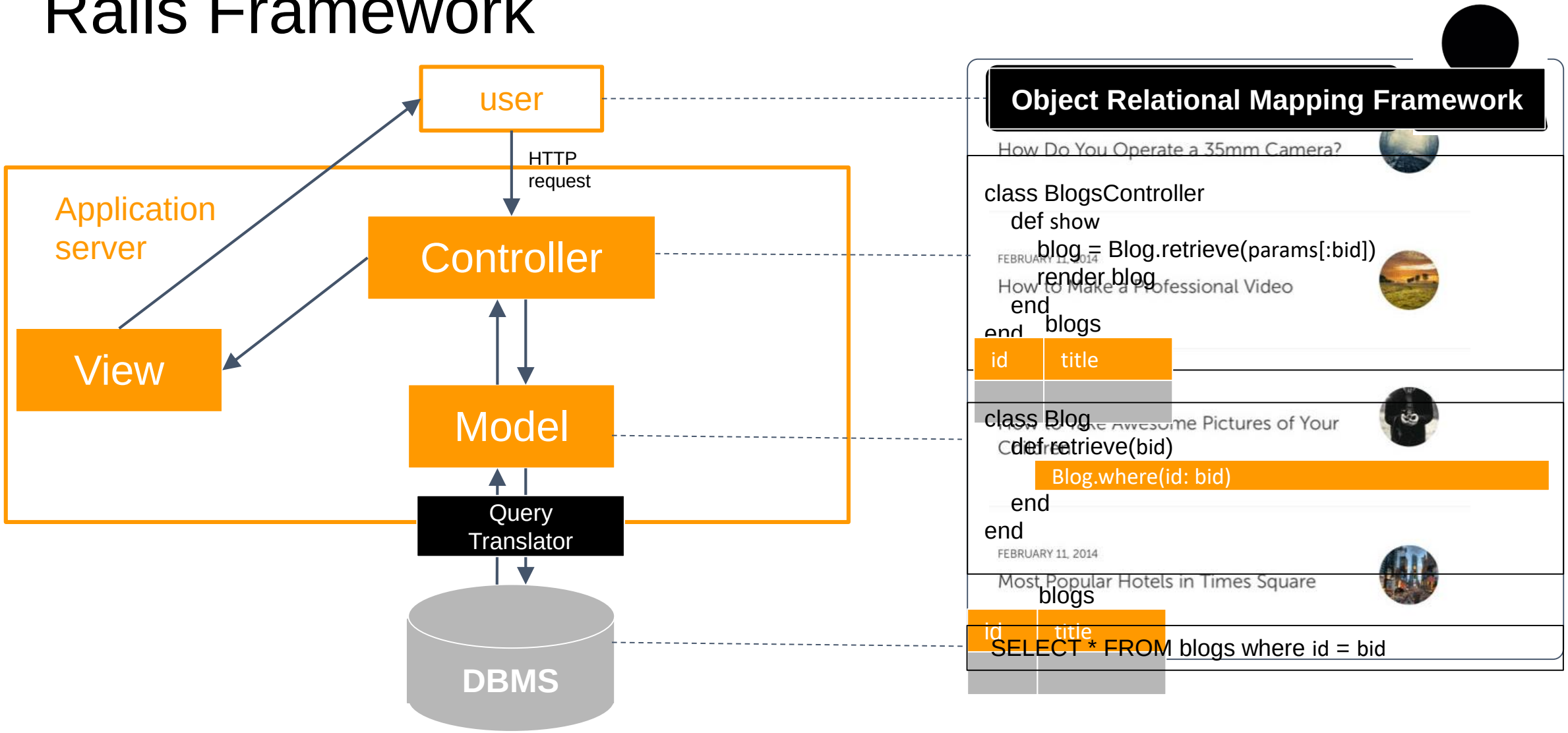
A toy MVC example

http://blogs-jwyoung.c9users.io:8080/blogs/show?bid=1
render blog with id = 1

http://blogs-jwyoung.c9users.io:8080/blogs/show?bid=2
render blog with id = 2



Rails Framework



Controller: BlogsController

```
1  class BlogsController < ApplicationController
2     def show
3           @blog = Blog.retrieve(params[:bid])
4     end
5  end
```

Model: blogs.rb

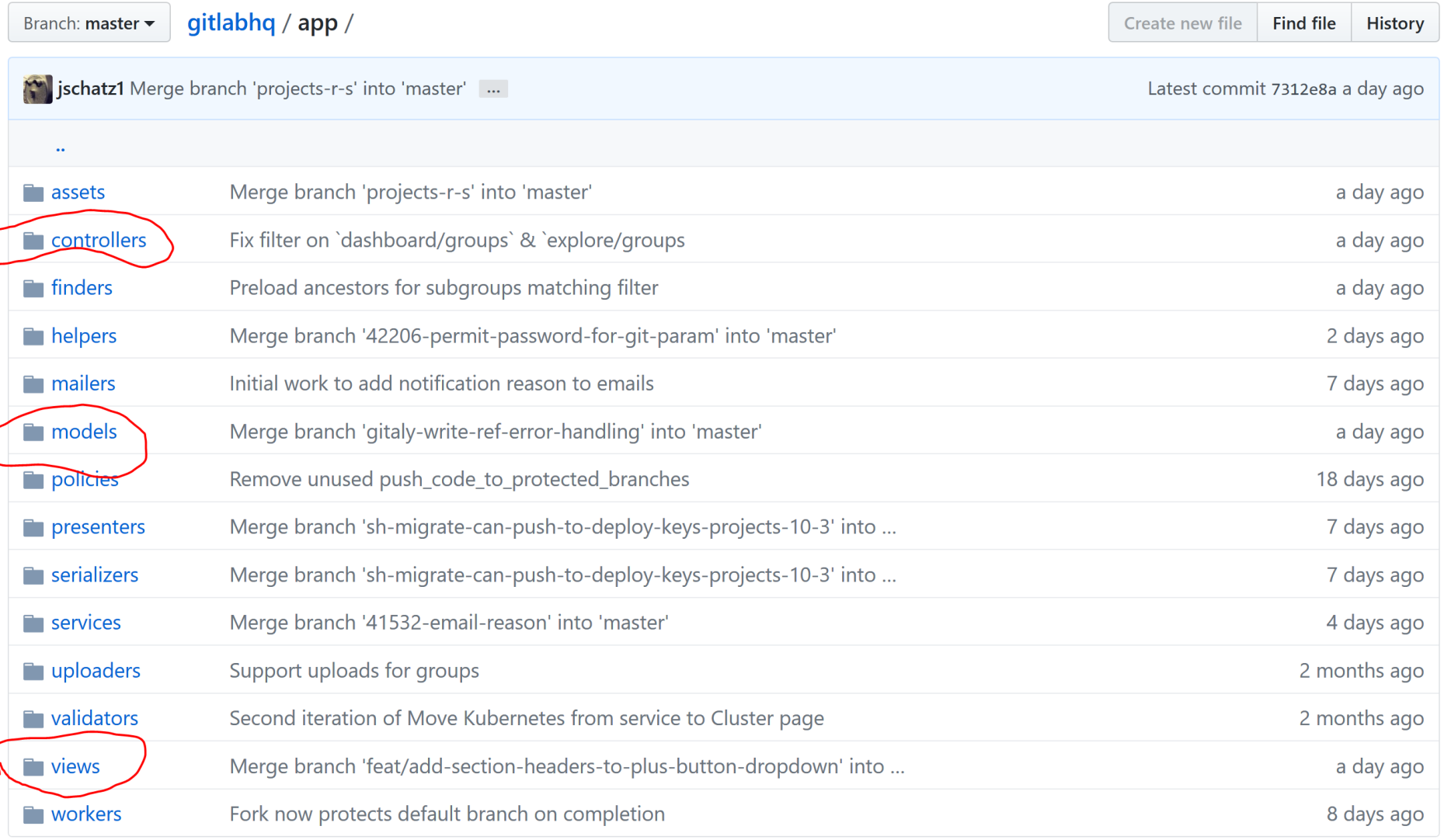
```ruby
class Blog < ActiveRecord::Base
    def self.retrieve(bid)
        self.where(id: bid).first
    end
end
```

View: blogs/show.html.erb

```
1
2     <div id="main-container" class="container">
3       <div class="row">
4         <div id="sidebar" class="col-xs-3">
5           <!-- sidebar content -->
6         </div>
7         <div id="content" class="col-xs-9">
8             <div class="card">
9               <p>
10                <strong>Title:</strong>
11                <%= @blog.title %>
12              </p>
13
14              <p>
15                <strong>Text:</strong>
16                <%= @blog.text %>
17              </p>
18            </div>
19        </div>
20      </div>
21    </div>
```

A example large software organized in MVC

- GitLab



Branch: master ▾ gitlabhq / app / Create new file | Find file | History

jschatz1 Merge branch 'projects-r-s' into 'master' ... Latest commit 7312e8a a day ago

..

assets	Merge branch 'projects-r-s' into 'master'	a day ago
controllers	Fix filter on `dashboard/groups` & `explore/groups	a day ago
finders	Preload ancestors for subgroups matching filter	a day ago
helpers	Merge branch '42206-permit-password-for-git-param' into 'master'	2 days ago
mailers	Initial work to add notification reason to emails	7 days ago
models	Merge branch 'gitaly-write-ref-error-handling' into 'master'	a day ago
policies	Remove unused push_code_to_protected_branches	18 days ago
presenters	Merge branch 'sh-migrate-can-push-to-deploy-keys-projects-10-3' into ...	7 days ago
serializers	Merge branch 'sh-migrate-can-push-to-deploy-keys-projects-10-3' into ...	7 days ago
services	Merge branch '41532-email-reason' into 'master'	4 days ago
uploaders	Support uploads for groups	2 months ago
validators	Second iteration of Move Kubernetes from service to Cluster page	2 months ago
views	Merge branch 'feat/add-section-headers-to-plus-button-dropdown' into ...	a day ago
workers	Fork now protects default branch on completion	8 days ago

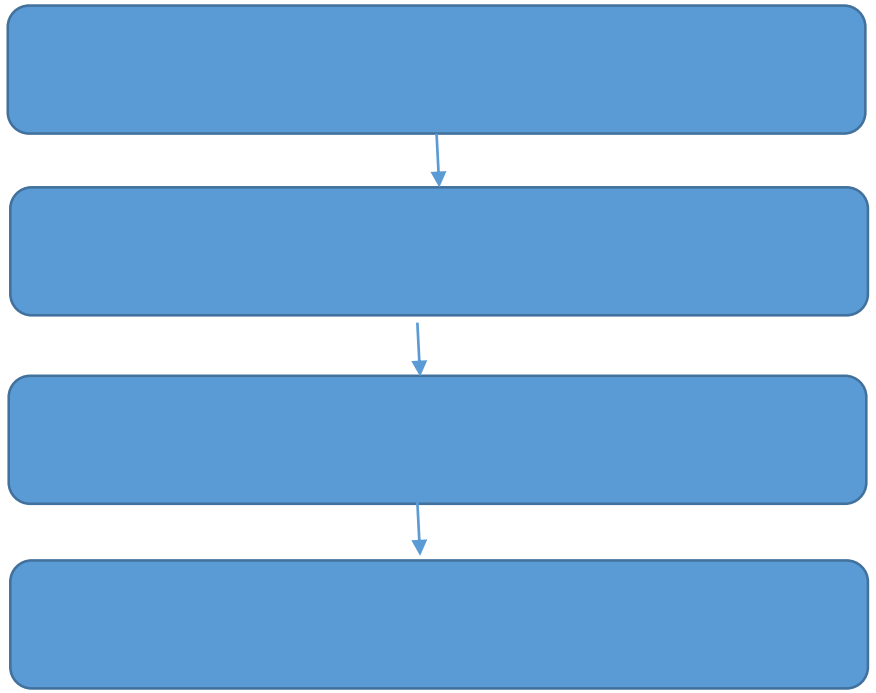
A example large software organized in MVC

- https://gitlab.com/explore/projects/trending
- View
  - https://github.com/gitlabhq/gitlabhq/tree/master/app/views/dashboard/projects
- Controller
  - https://github.com/gitlabhq/gitlabhq/blob/master/app/controllers/explore/projects_controller.rb
- Model
  - …

Design concerns in MVC

- Where to put M, V, C, given multiple nodes?
  - M is most suitable for server machines
  - C is most suitable for client machines
  - V depends on network bandwidth between serve and client

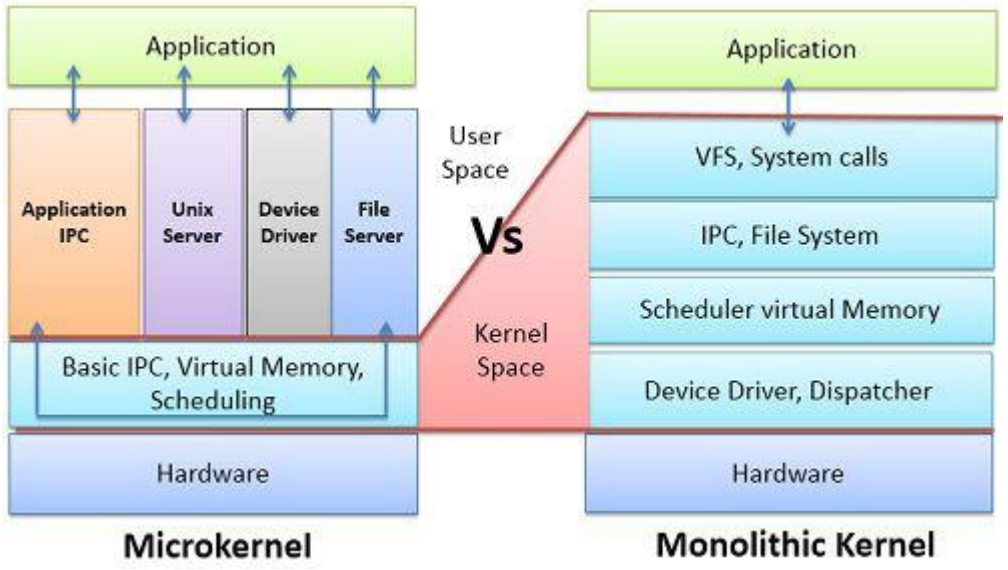
Layered pattern



Examples

- Operating systems

- Strength?
- Weakness?



Examples

- Operating systems and user applications

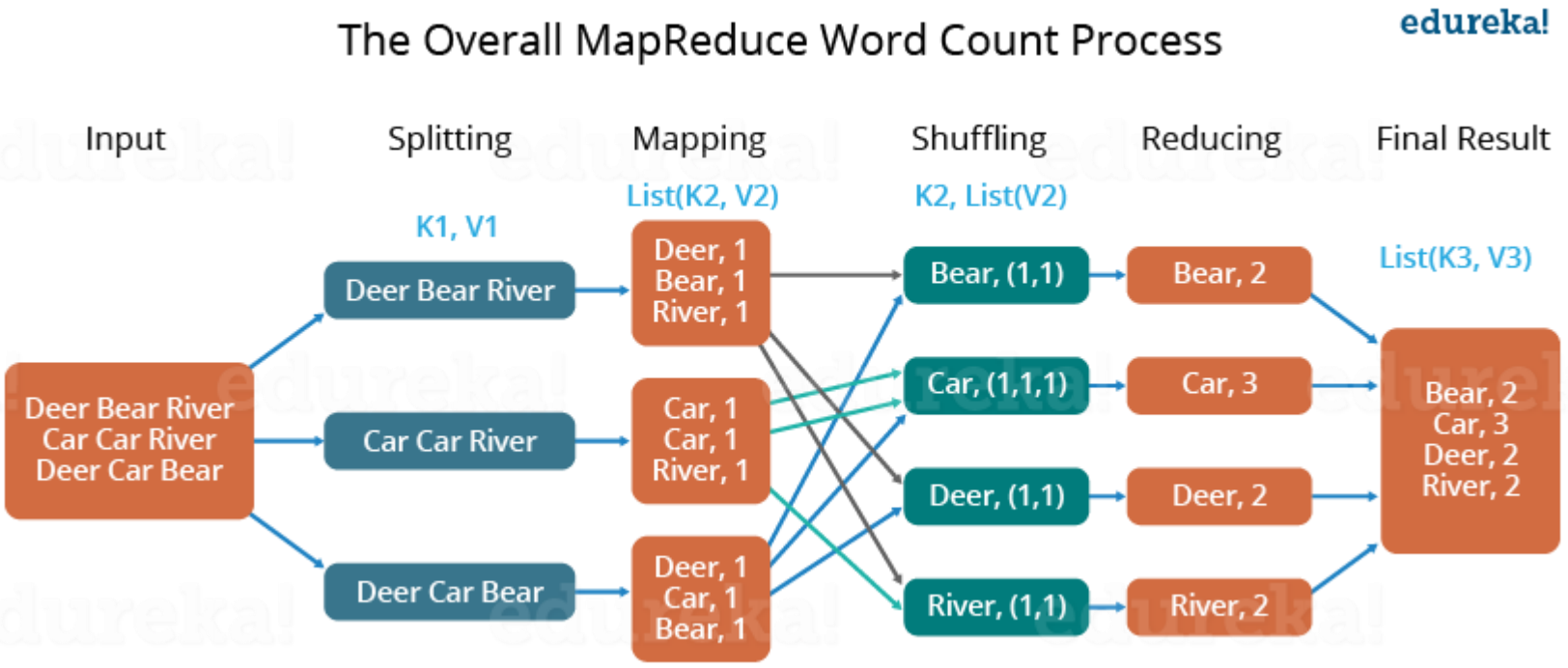
- Strength
  - Easy testing
  - Modify one layer won’t affect the whole system
- Weakness
  - Performance
  - Strict layering may be difficult in practice

Data-flow pattern (pipe-and-filter)

Example?

Example 1

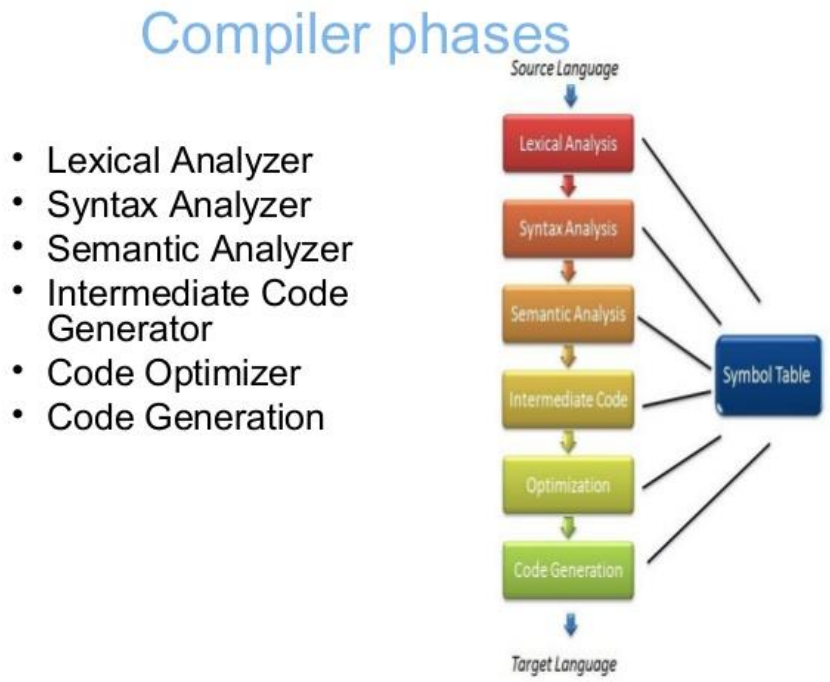
- Map Reduce



More examples

- Pipe & filter

- Compiler



Pros & Cons

- Strengths
  - ??
- Weaknesses
  - ??

Pros & Cons

- Strengths
  - Filters can be re-used
- Weaknesses
  - Data format compatibility among filters is crucial; format violation will break down the system