



开源技术构建 Web App 架构

@Paracutin @豌豆荚

2012年底，我们想做一个豌豆荚 Web 版

我们期待的样子：

单页面，实时，稳定，开发效率高，
易于部署，DRY，结构清晰，拿个奖

需要解决的事情：

协同，代码管理，调试环境，开发框架，资源优化，部署

/auth

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

/photo

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

GET	/resource/photos	Fetch photos list.
-----	------------------	--------------------

POST	/resource/photos	Create photos.
------	------------------	----------------

GET	/resource/photos/:id	Fetch a photo by id.
-----	----------------------	----------------------

PUT	/resource/photos/:id	Update a photo by id.
-----	----------------------	-----------------------

DELETE	/resource/photos/:id	Delete a photo by id.
--------	----------------------	-----------------------

GET	/directive/photos/download	Download photo on user's phone.
-----	----------------------------	---------------------------------

POST	/directive/photos/upload	
------	--------------------------	--

/message

/contact

GET	/resource/contacts	
-----	--------------------	--

POST	/resource/contacts	Create contacts.
------	--------------------	------------------

GET	/resource/contacts/:id	Fetch a contact by id.
-----	------------------------	------------------------

PUT	/resource/contacts/:id	Update a contact.
-----	------------------------	-------------------

DELETE	/resource/contacts/:id	Delete a contact by id.
--------	------------------------	-------------------------

POST	/resource/contacts/delete	Batch delete contacts.
------	---------------------------	------------------------

GET	/resource/contacts/search	Search contacts that match the query.
-----	---------------------------	---------------------------------------


/app

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)

/setting





[Show/Hide](#) | [List Operations](#) | [Expand Operations](#) | [Raw](#)


设计 RESTful Data API,
前后端可以并行工作



This repository ▾

ExploreGistBlogHelp


 windyarts   


PUBLIC  wandoulabs / **satan**

[Pull Request](#) [Unwatch](#) [Star](#) 0 [Fork](#) 5


CodeNetworkPull Requests 0Issues 0WikiGraphsSettings

The EVIL side of wandoujia — [Read more](#)

 Clone in Mac


 ZIP

HTTPSSHGit Read-Only













Read+Write access

branch: master ▾FilesCommitsBranches 1Tags 1

satan /  440 commits

Merge pull request #58 from windyarts/master

 windyarts authored 4 hours ago latest commit e6ded93f7f 

 app	4 hours ago	Get contacts suggestion from local cache [windyarts]
 test	3 months ago	Refactor auth flow [windyarts]
 .bowerrc	5 months ago	Built up app structure [Ziming Miao]
 .editorconfig	5 months ago	Built up app structure [Ziming Miao]
 .gitattributes	5 months ago	Built up app structure [Ziming Miao]
 .gitignore	3 days ago	Fix sprites issue [windyarts]
 .gitmodules	5 months ago	Built up app structure [Ziming Miao]
 ishintro	11 days ago	refactor again [windyarts]

既然从来都是裸着的，不如我们开源吧



A JAVASCRIPT
MODULE LOADER

Home 🏠

Start ⏻

Download ⬇

API ⚙

Optimization ⚙

Use with jQuery </>

Use with Node </>

Use with Dojo </>

CommonJS Notes </>

FAQs ?

Common Errors ?

Writing Plugins ⚙

Why Web Modules ?

Why AMD ?

Requirements 📄

History ⌚

Get Help +

Blog ✍

Twitter 🐦

GitHub 🐱

Contributing 🛠

2.1.6 Docs (upgrade info)

1.0 Doc Link

```
/* ---
```

```
RequireJS is a JavaScript file and module  
loader. It is optimized for in-browser use, but  
it can be used in other JavaScript environments,  
like Rhino and Node. Using a modular script  
loader like RequireJS will improve the speed and
```

AMD 能做到的:

定义模块

自动管理依赖

CSS/HTML 加载

灵活的 Build 工具

✓
✓
✓
✓
✓
the API.

Latest Release: 2.1.6

Open source: new BSD or MIT licensed

web design by Andy Chung © 2011

```
1  define([
2      'text!template.html', 'text!style.css'
3  ], function(
4      html, style
5  ) {
6
7      var element = $(html).appendTo(document.body);
8      $('<style>').html(style).appendTo(document.head);
9
10     // Do anything with element
11
12 });
```




Compass is an open-source *CSS Authoring Framework*.

♥ Why designers love Compass.

1. Experience cleaner markup without presentational classes.
2. It's chock full of the web's best [reusable patterns](#).
3. It makes creating [sprites](#) a breeze.
4. Compass mixins make [CSS3](#) easy.
5. Create beautiful typographic rhythms.
- 6.

✓ Compass uses Sass.

[Sass](#) is an extension of CSS3 which adds nested rules, variables, mixins, selector inheritance, and more. Sass generates well formatted CSS and makes your stylesheets easier to organize and maintain.

Sass & Compass 能做到：
可编程的 CSS
自动合成 Sprite

```
@import "icons/*.png";

@each $icon in "add", "remove", "update" {
  .icon-#{ $icon } {
    @include inline-block;
    @include transition(opacity, 0.4s);
    @include icons-sprite($icon);
    opacity: 0.7;
    &:hover {
      opacity: 1;
    }
  }
}
```

workflow



YO

脚本任务



GRUNT

类库管理



BOWER

```
// Gruntfile.js

module.exports = function(grunt) {
  grunt.initConfig({
    // compile .scss/.sass to .css using Compass
    compass: {
      dist: {
        options: {
          config: 'compass.config.rb',
          force: false
        }
      }
    },
  });

  grunt.registerTask('default', ['compass']);
};

// Run 'grunt' will compile scss using Compass
```

```
{  
  "name": "satan",  
  "version": "0.0.0",  
  "dependencies": {  
    "angular-ui": "~0.4.0",  
    "moment": "~2.0.0",  
    "angular": "~1.0.6",  
    "angular-resource": "~1.0.6",  
    "socket.io-client": "~0.9.11"  
  }  
}
```

```
// Run 'bower install' will download  
// dependencies.
```

有了这些，我们能够在 10 分钟内，
在一台新电脑上搭好开发环境

接下来我们需要寻找一个合适的 MVC 框架

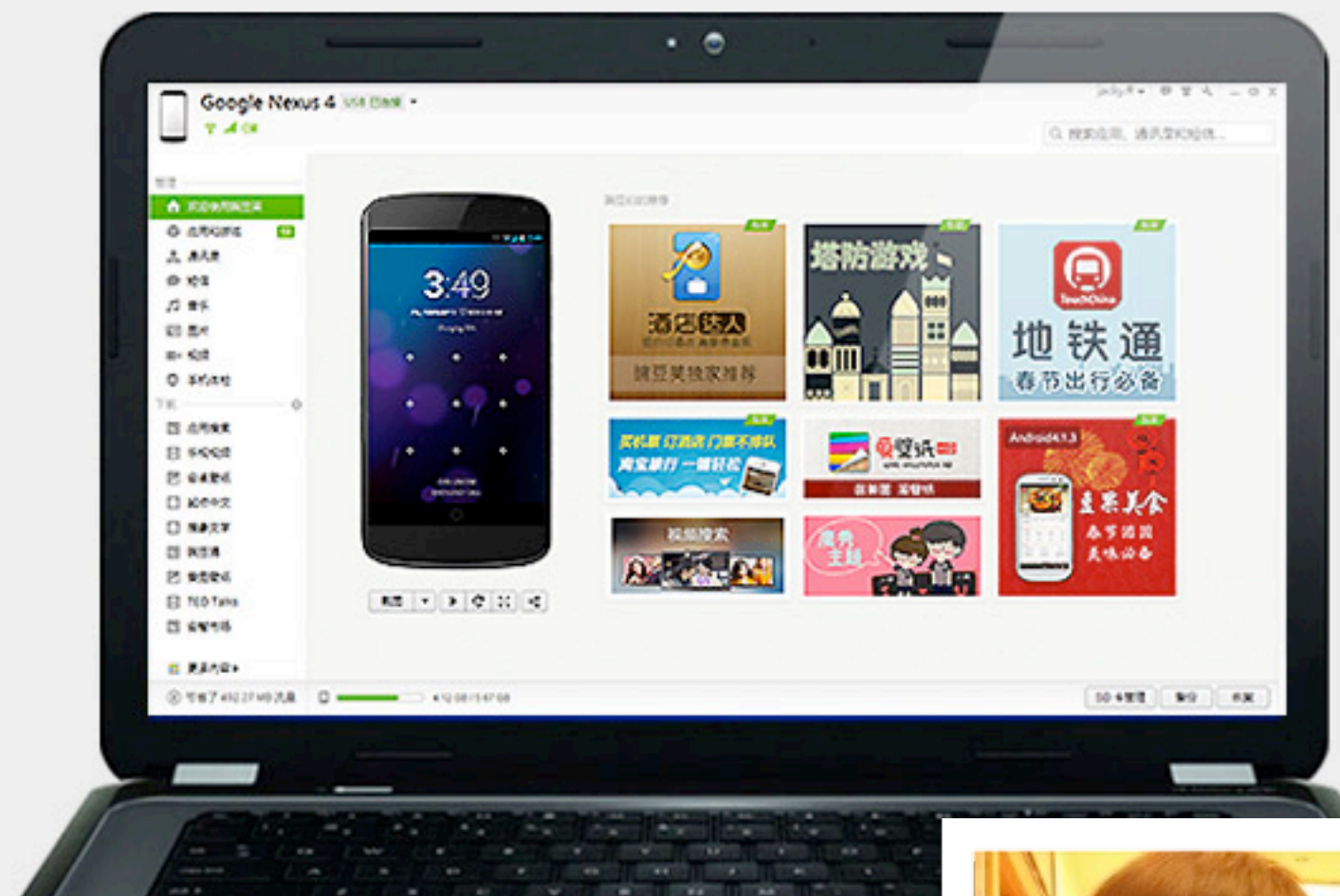


您的安卓手机加油站

最丰富的应用、游戏、音乐、视频和壁纸免费下载到手机。

↓ 下载豌豆荚

其他下载: 完整安装包 Beta 测试版 迷你版
在手机上安装豌豆荚



我们已经试过 Backbone
by wangye

想到, 就能下载

豌豆荚收录了超过 45 万款优质 Android 应用和游戏, 以及
130 多家内容提供商提供的音乐、视频、电子书、壁纸和主题。
让您的手机比您想象中更强大更好玩。

前往豌豆荚应用搜索 →



@赵望野

优势：

1. 精致，抽象的数据同步模型
2. 清晰的结构，易于扩展

缺陷：

1. View 层功能薄弱
2. 缺乏常用功能



餐厅: 麦当劳

菜单

套餐

主食

甜品

小食

+1

温柔将军堡套餐

28.50 元

+1

麦香鱼餐

25.00 元

+1

双层吉士汉堡餐

25.00 元

+1

五色嫩鸡菠菜卷餐

23.50 元

+1

五色至牛菠菜卷

23.50 元

+1

巨无霸餐

26.00 元

+1

巨无霸餐(不要吉士)

26.00 元

+1

麦乐鸡餐

22.50 元

+1

麦辣鸡翅餐

27.00 元

+1

培根蔬萃双层牛堡餐

27.50 元

+1

经典麦辣鸡腿汉堡

+1

珍萃麦辣鸡腿汉堡餐

26.00 元

+1

原味板烧鸡腿堡套餐

26.00 元

+1

麦香鸡餐

21.00 元

下单

清空

← 快去点餐吧 :)

我们又尝试了 Knockout

优势：

1. 双向数据绑定，写 View 很方便
2. 可以通过插件扩展功能

缺陷：

1. 只有 View 层解决方案
2. 基于事件的数据更新，不容易优化



Manage your Android phone from the web.

Your photos, messages, contacts and apps in your web browser.

Sign in

[Where's my passcode?](#)

最后我们用了 Angular

New to SnapPea?

To get your passcode, [install SnapPea](#) on your phone. It's free.



我们觉得 Angular 的一些理念很好，
也许可以用在你的项目里

Model-View-ViewModel

Service

XHR, DB, Worker,
Sync, Calc, etc.

Service

XHR, DB, Worker,
Sync, Calc, etc.

Controller

Prepare Data &
Interaction handler

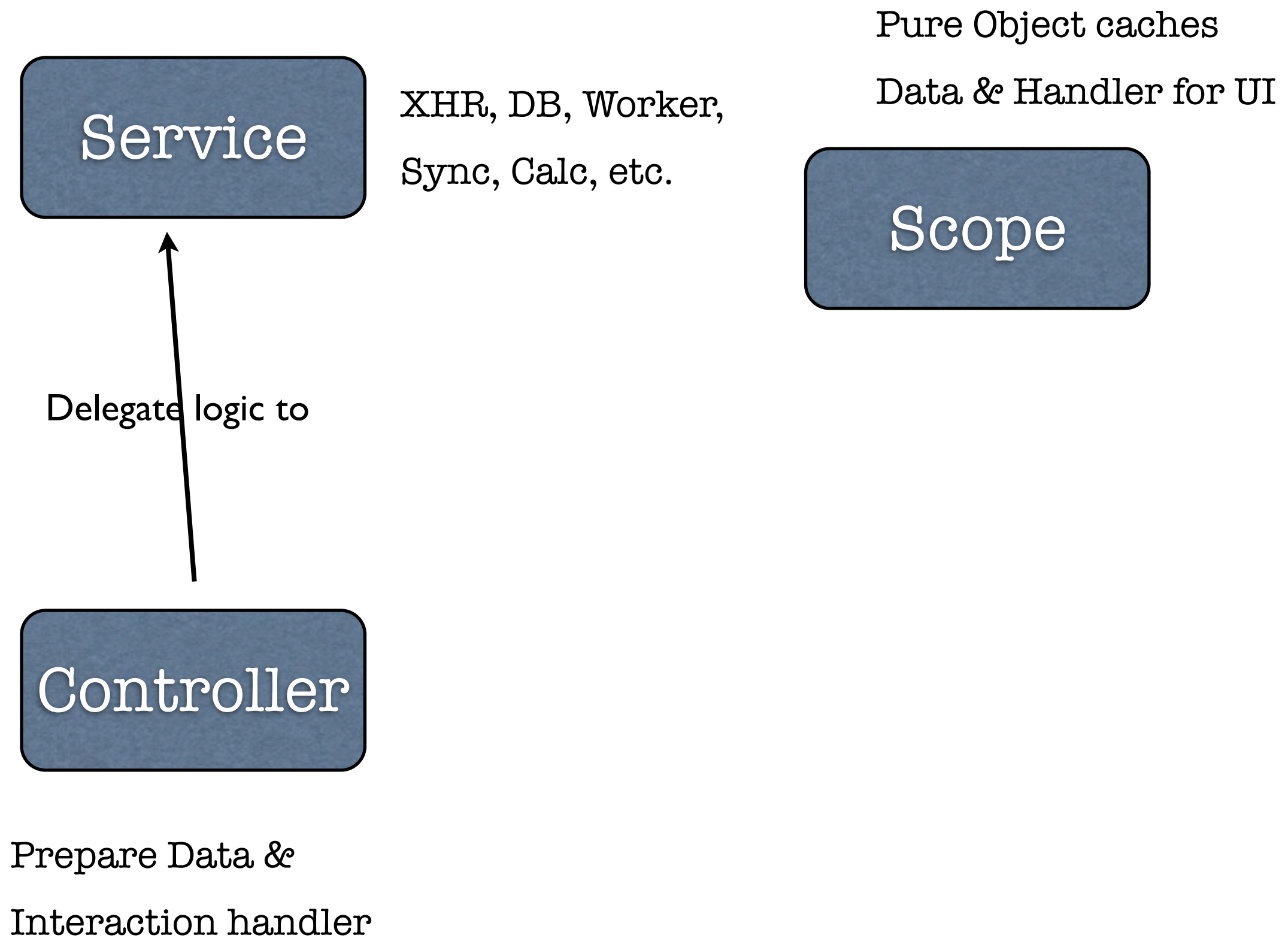
Service

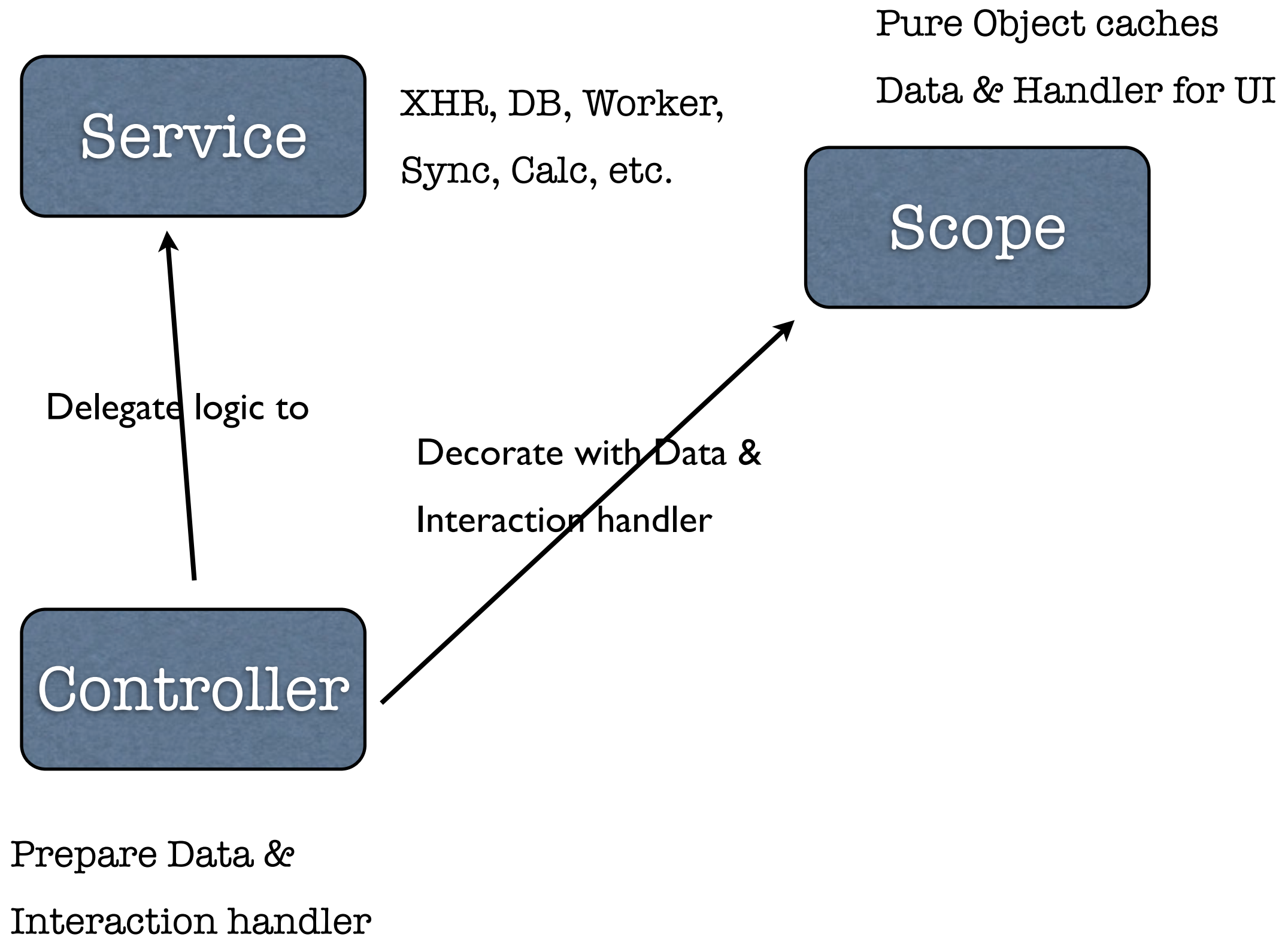
XHR, DB, Worker,
Sync, Calc, etc.

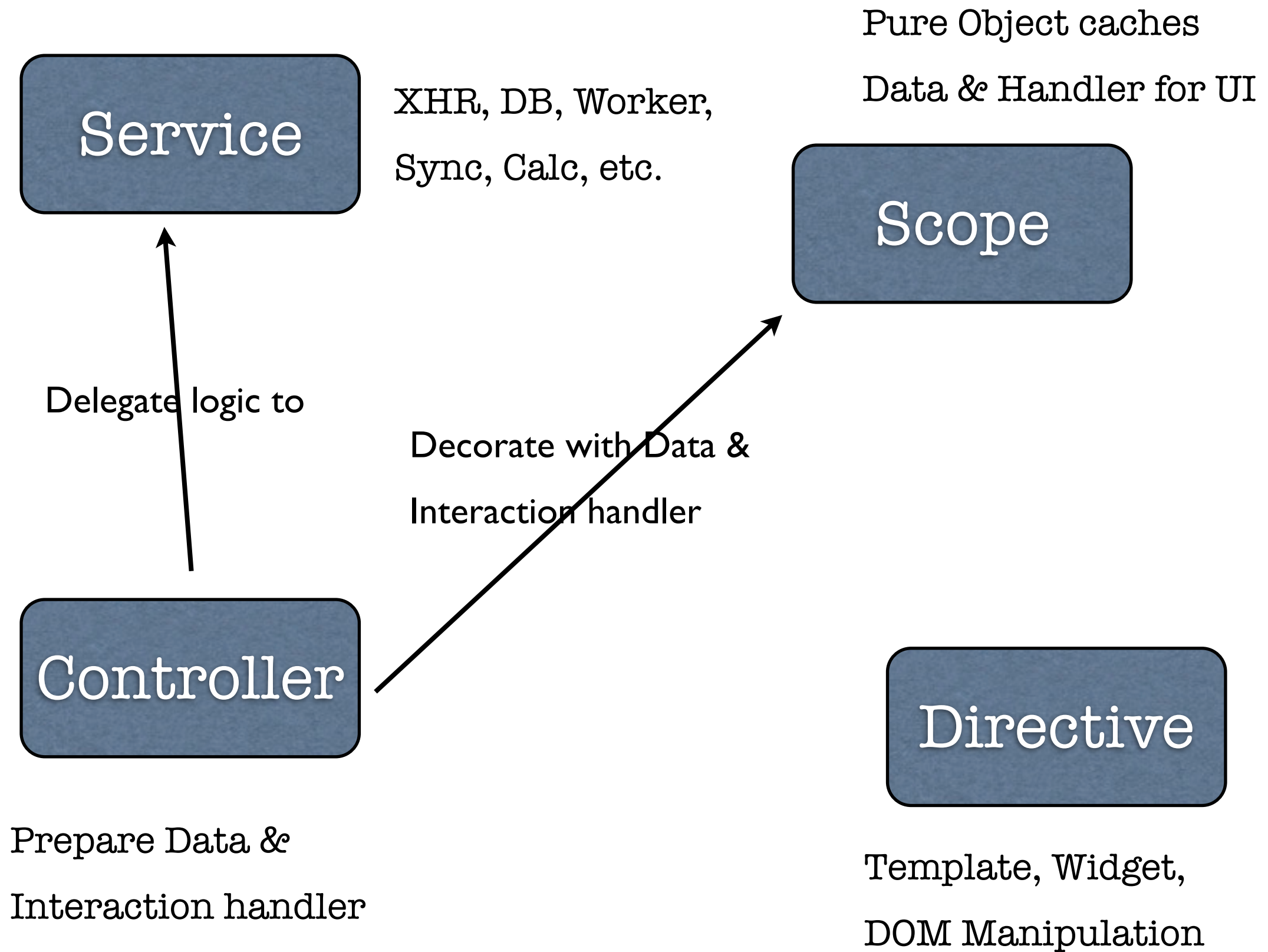
Delegate logic to

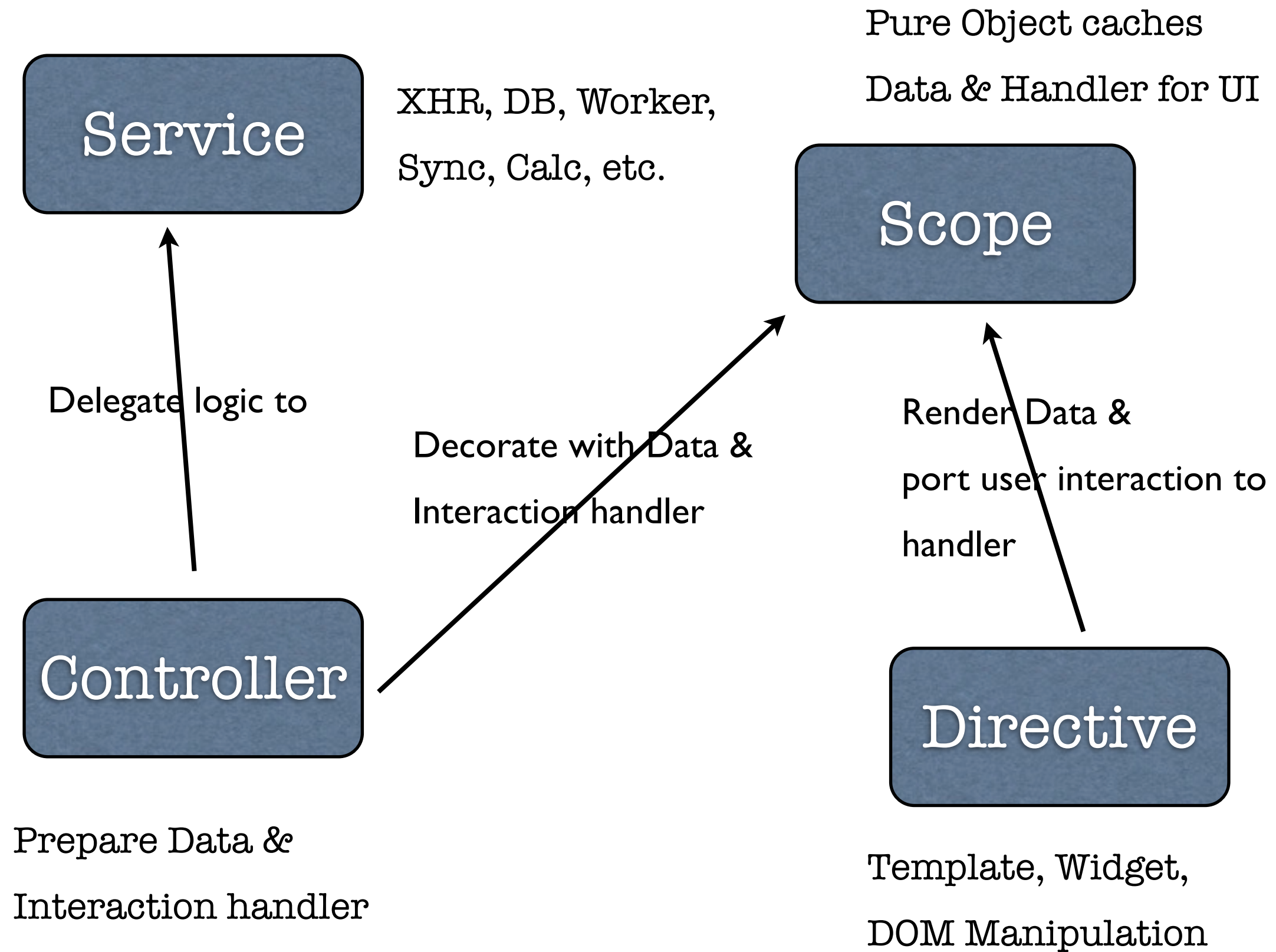
Controller

Prepare Data &
Interaction handler









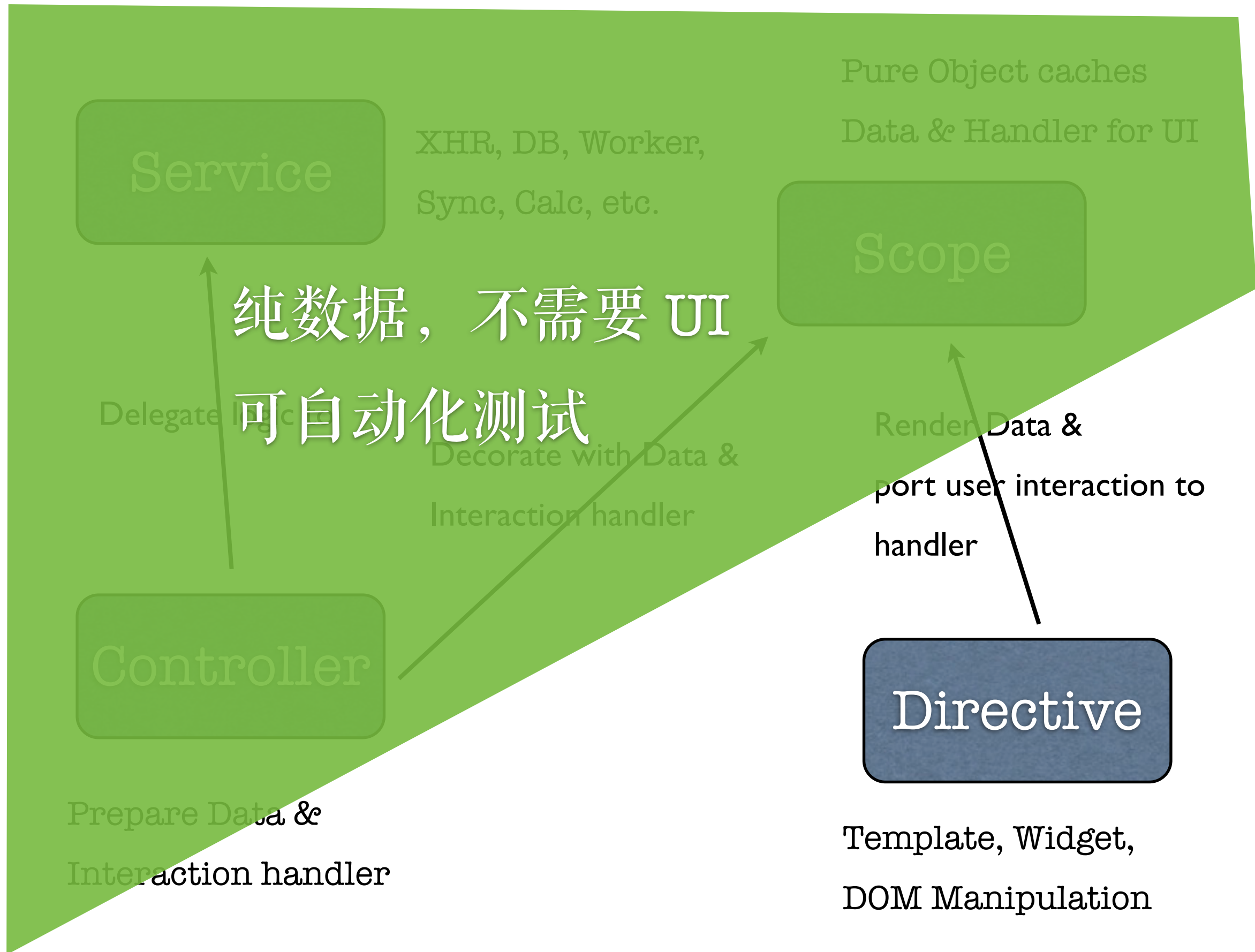
依赖注入


```
angular.module('snappea', ['ng'])
  .service('photos', function($http) {
    this.fetch = function() {
      return $http.get('http://snappea.com/photos');
    };
  })
  .controller('galleryController', function($scope, photos) {
    $scope.photos = [];
    $scope.fetch = function() {
      $scope.photos = Photos.fetch();
    };
  })
  .directive('counter', function() {
    return function($scope, $element) {
      $element.on('click', function() {
        $scope.fetch();
      });
      $scope.$watch('photos.length', function(value) {
        $element.text('Total ' + value + ' photos. ');
      });
    };
  });
```

```
<body ng-app>
  <div ng-controller="galleryController">
    <counter>Click to fetch photos!</counter>
    <ol ng-click="fetch()">
      <li ng-repeat="p in photos">
        
      </li>
    </ol>
  </div>
</body>
```


由此我们可以做到：

1. MVC 严格分层
2. 依赖注入控制访问权限
3. 良好的可测试性



基于 DOM 的模板

```
<body ng-app>
  <div ng-controller="galleryController">
    <counter>Click to fetch photos!</counter>
    <ol ng-click="fetch()">
      <li ng-repeat="p in photos">
        
      </li>
    </ol>
  </div>
</body>
```

Before compile phase

```
<body ng-app>
  <div ng-controller="galleryController">
    <counter>Click to fetch photos!</counter>
    <ol ng-click="fetch()">
      <!-- ng-repeat: p in photos -->
      <li class="ng-scope">
        
      </li>
      <li class="ng-scope">
        
      </li>
      <li class="ng-scope">
        
      </li>
    </ol>
  </div>
</body>
```

After compile phase, before link phase,
non-interactive

```
<body ng-app>
  <div ng-controller="galleryController">
    <counter>Total 3 photos.</counter>
    <ol ng-click="fetch()">
      <!-- ng-repeat: p in photos -->
      <li class="ng-scope">
        
      </li>
      <li class="ng-scope">
        
      </li>
      <li class="ng-scope">
        
      </li>
    </ol>
  </div>
</body>
```

After link phase, all ready

基于 dirty-check 的数据更新机制

事件缺少 context，可能导致一系列问题：

- 难于 debug 和定位
- 事件泛滥，容易冲突，维护难度大
- 难于批量优化

dirty-check 会循环的检查所有数据，
直到不再有变化，最后一次性渲染，
难点在于如何保证即时触发

可能导致数据变化的事件：

- DOM Event
- Timeout
- XHR, Worker, etc.

最大的收益：

控制 UI 变化的时机和节奏，优化渲染性能



部署是件很幸福的事

S3 & Cloud Front

2

1

4

2 个人 × 1 个月 × 4 个功能模块

Q & A