

复杂Rails系统架构优化

从1到30：如何将单一系统切分成相互协作的小应用集群

Jonathan Palley , idapted CTO
郭磊, idapted 系统架构师

从一个故事开始

双“楼”记

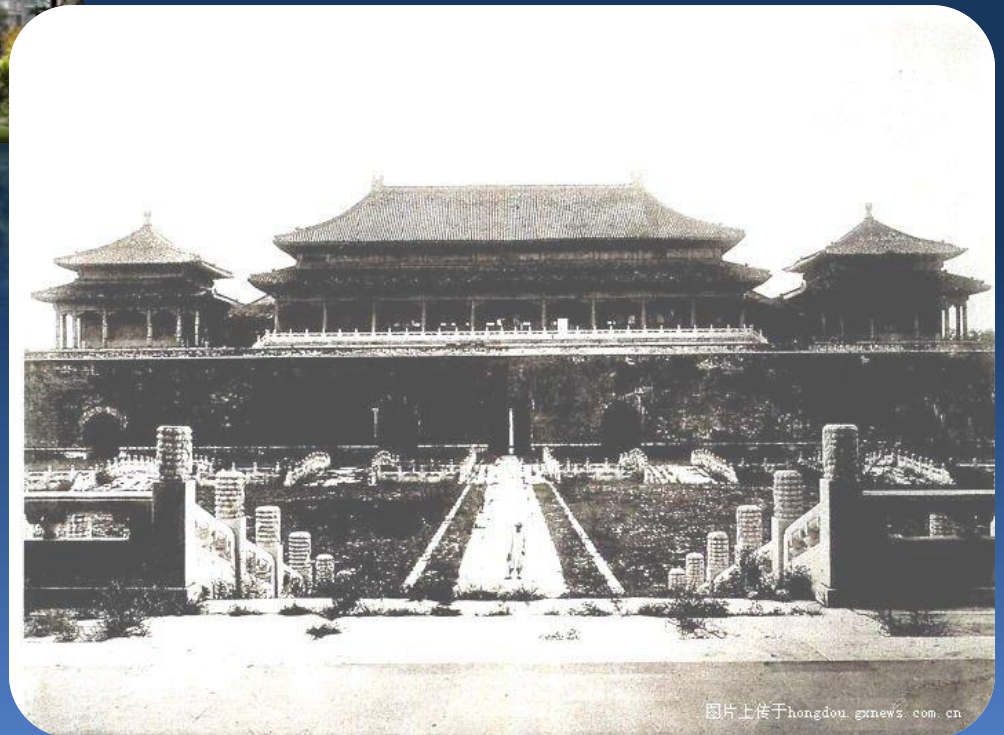
SACC

2010系统架构师大会

2009 上海 莲花河畔景苑



1909 北京 故宫



图片上传于hongdou.gnews.com.cn

1



30

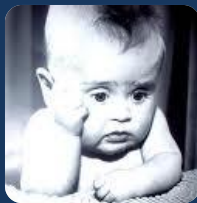


什么是“1”？

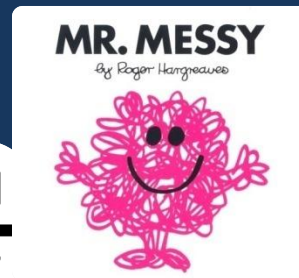
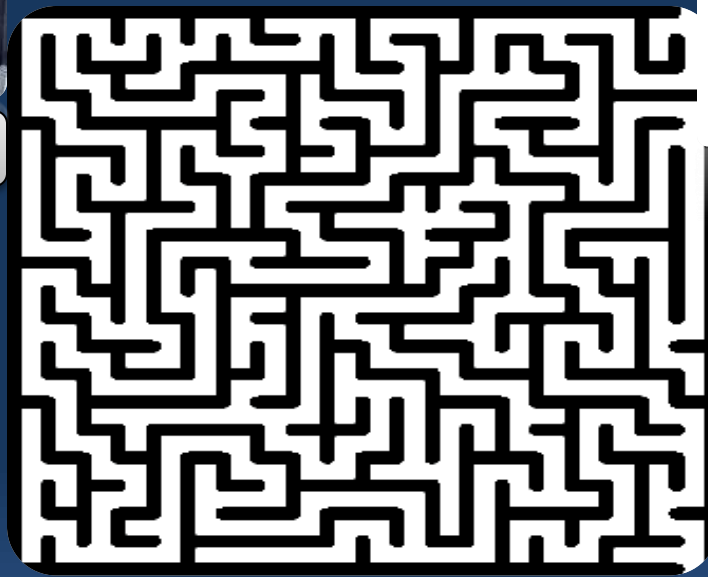
单一Rails应用

(尤其是指大型Rails应用，涉及到多种角色、非常广泛的业务功能)

问题



困惑的新员工



难以测试/维护/扩展

什么是“30”？

一组应用



独立



无缝连接



每个应用的基本特征

- 独立的数据库
- 能够独立运行（处理完整的业务流程）
- 轻量级(一个开发者维护)
- 高内聚低耦合



如何实现？

UI一致性

- 共享CSS/JS/Styleguide
- Gem中公用方法

所有应用使用同一基础CSS/JS

保证所有应用的风格一致

```
<%= idp_include_js_css %>
```

```
# =>
```

```
<script src ="/assets/javascripts/frame.js" type="text/javascript"></script>
```

```
<link href="/assets/stylesheets/frame.css" media="screen" rel="stylesheet"  
      type="text/css" />
```

CSS 框架

Heading 1

Heading2

Heading3

Heading4

Heading5

Heading6

Paragraph for notice.

Paragraph for warning.

Paragraph for alert.

Want to hide sth? give it class="hidden":

This is a paragraph.

This is for bold font. or

This is for italic font.

This is for underline font.

This is for blockquote.

This is ^{sup}

This is _{sub}

This is red bold font.

This is gray font.

paragraph with 14px size.

Paragraph with 16px size and red color.

center align text

Gem中的公用Helper方法

Model的查询

Advanced Search ID

ID

Released

Rating

Mark Spot Num -

Created At From To

[cancel](#)

am
pm

View:

<%=

```
search_form_for(HistoryRecord, :interaction_id, :released,[:rating,  
{:collection=>assess_ratings}],[:mark_spot_num,{:range=>true}],  
[:created_at, {:ampm=>true}])
```

%>

Controller:

```
@history_records = HistoryRecord.combo_search(params)
```

开发周期

1. 在某一应用中实现方法
2. 放到公用gem中



interface

data

应用间如何共享数据?

- 只读数据库
- Services

data

示例

SACC
2010系统架构师大会



user
用户信息



course
课程管理



purchase
订单管理



learning
学习过程

data



Purchase

SACC
2010系统架构师大会

需求: 列出课程信息供用户选择

但是

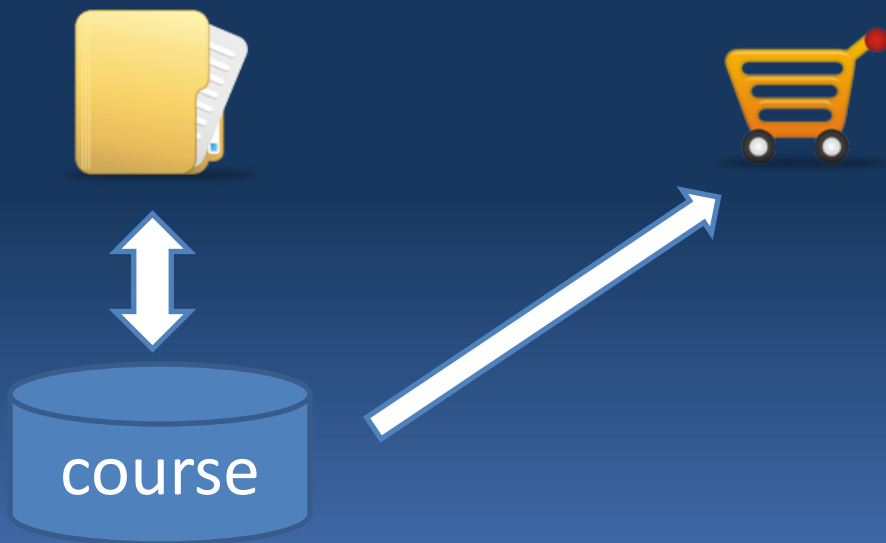
课程信息在“course”存储和管理

data

解决方案

SACC
2010系统架构师大会

只读数据库





Model:

```
class CoursePackage < ActiveRecord::Base
  acts_as_readonly :course
end
```

View:

```
<ul>
<% CoursePackage.all.each do |package| %>
  <li><%= package.title %> <%= package.price %></li>
<% end %>
</ul>
```


是否造成高耦合？

Model:

```
class CoursePackage < ActiveRecord::Base
  acts_as_readonly :course
end
```

View:

```
<ul>
  <% CoursePackage.all.each do |package| %>
    <li><%= package.title %> <%= package.price %></li>
  <% end %>
</ul>
```

深入acts_as_readonly

```
def acts_as_readonly(name, options = {})  
  config = CoreService.app(name).database  
  establish_connection config[Rails.env]  
  set_table_name(self.connection.current_database +  
    (options[:table_name] || table_name).to_s)  
end
```

深入acts_as_readonly

```
def acts_as_readonly(name, options = {})  
  config = CoreService.app(name).database  
  establish_connection config[Rails.env]  
  set_table_name(self.connection.current_database +  
    (options[:table_name] || table_name).to_s)  
end
```

Core service

```
class CoreService < ActiveRecord::Base
  self.site = :user
  def self.app(app_name)
    CoreService.find(app_name)
  end
end
```

data

配置集中管理

SACC
2010系统架构师大会

Core如何知道所有应用的配置信息？

data

应用启动时将配置信息发送到core



data

config/site_config.yml

```
app: course  
api:  
  course_list: package/courses
```

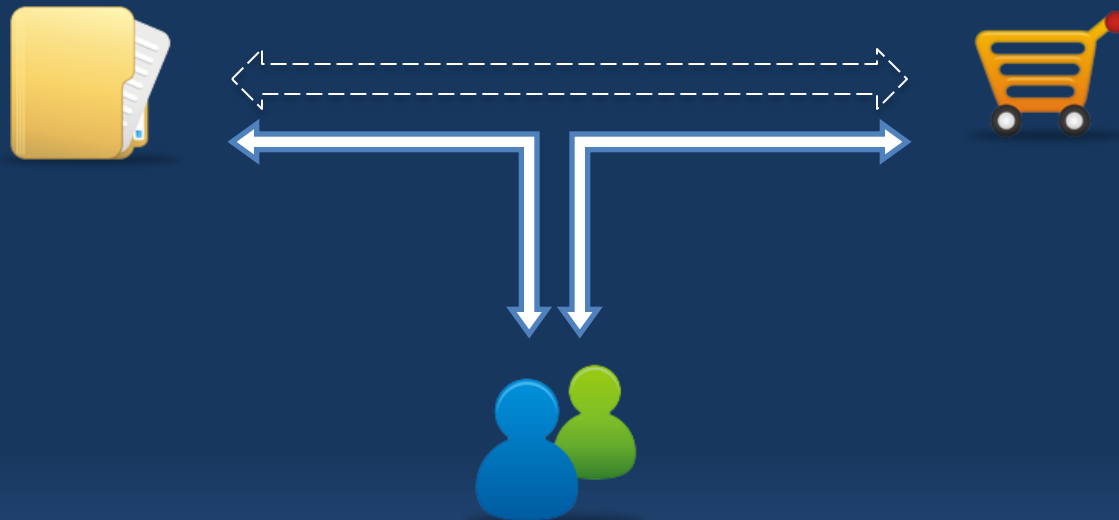
config/initializers/idp_initializer.rb

```
CoreService.reset_config
```

core_service.rb in idp_lib

```
APP_CONFIG =  
  YAML.load(Rails.root.join("config/site_config.yml"))  
  
def reset_config  
  self.post(:reset_config, :app => {  
    :name => APP_CONFIG["app"],  
    :settings => APP_CONFIG,  
    :database => YAML.load_file(  
      Rails.root.join("config/database.yml"))})  
end
```

data



Model in Purchase:

```
class CoursePackage < ActiveRecord::Base
  acts_as_readonly :course
end
```

通过gem简化代码

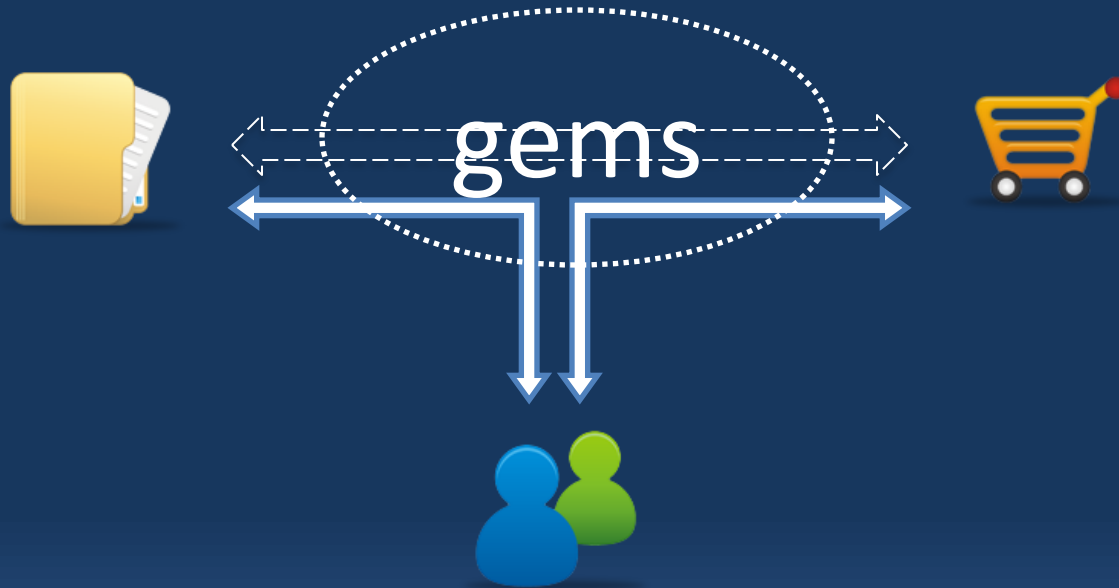
config/environment.rb

```
config.gem 'idp_helpers'
```

```
config.gem 'idp_lib'
```

data

SACC
2010系统架构师大会



- 写操作: Web services

```
class CoursePackageService < ActiveSupport::Base
  self.site = :course
end
```

data

示例

SACC
2010系统架构师大会



用户购买课程后生成学习信息

learning: app/controllers/roadmap_services_controller.rb

```
def create
  Roadmap.generate(params[:user_id], params[:course_id])
end
```

Purchase: app/models/roadmap_service.rb

```
class RoadmapService < ActiveSupport::Base
  self.site = :course
end
```

Purchase: app/models/order.rb

```
def activate_roadmap
  RoadmapService.create(self.user_id, self.course_id)
end
```


data

开源gem

SACC
2010系统架构师大会



http://github.com/idapted/eco_apps



interface

data

user



User的功能

- 用户注册与登录
- 用户信息管理
- 基于角色的权限控制

user

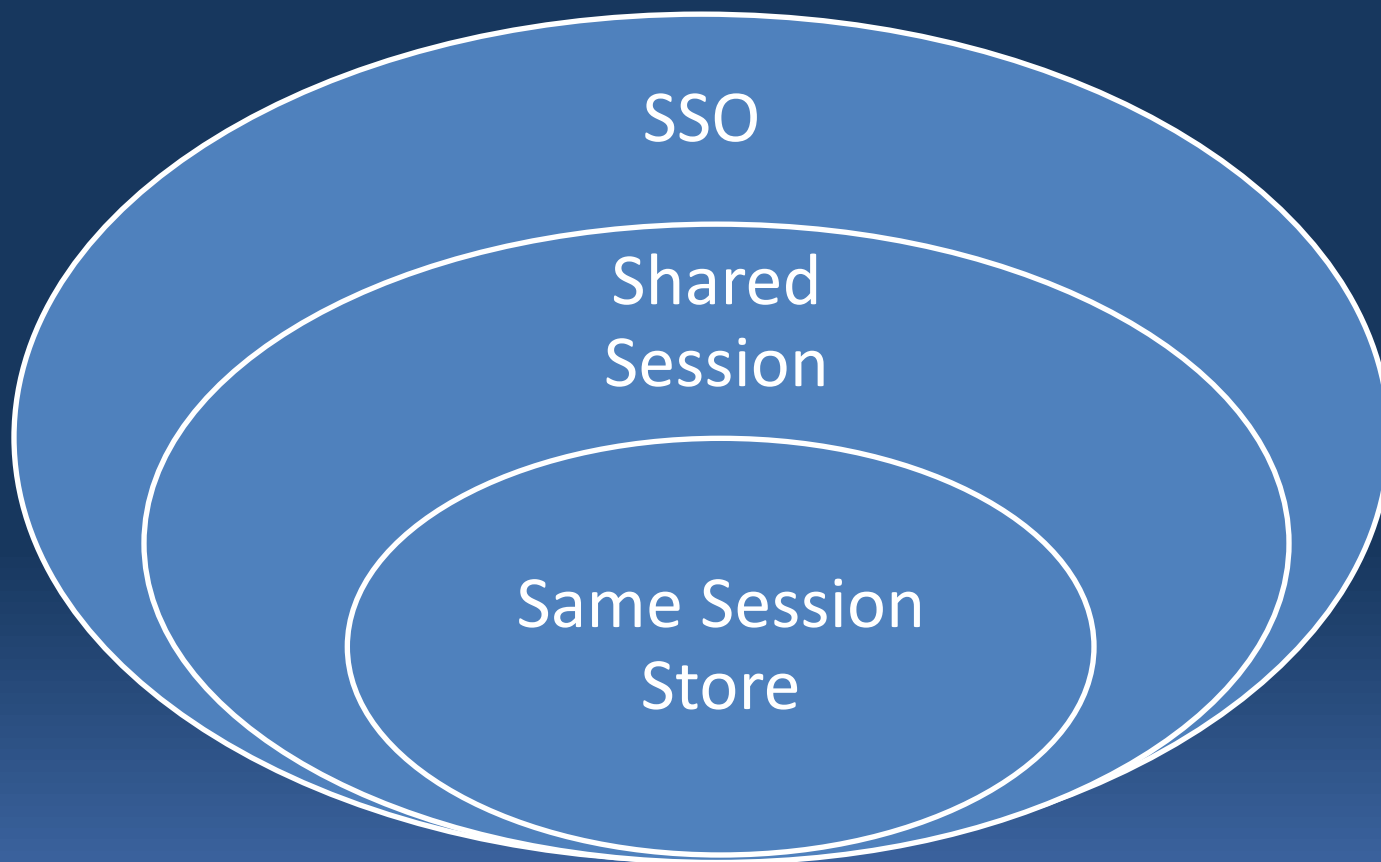


用户信息共享

user

登录验证

SACC
2010系统架构师大会



user

登录验证



config/initializers/idp_initializer.rb

```
ActionController::Base.session_store = :active_record_store  
ActiveRecord::SessionStore::Session.acts_as_remote :user,  
  :readonly => false
```

权限控制

Controller做为“节点”

	<u>Admin</u>	<u>Trainer</u>	<u>Student</u>
application	<input type="checkbox"/> Admin	<input type="checkbox"/> Trainer	<input type="checkbox"/> Student
admin			
students	<input checked="" type="checkbox"/> Admin	<input type="checkbox"/> Trainer	<input type="checkbox"/> Student
reports	<input checked="" type="checkbox"/> Admin	<input type="checkbox"/> Trainer	<input type="checkbox"/> Student
detail			
base	<input type="checkbox"/> Admin	<input type="checkbox"/> Trainer	<input type="checkbox"/> Student
self_studies	<input checked="" type="checkbox"/> Admin	<input type="checkbox"/> Trainer	<input type="checkbox"/> Student
reviews	<input checked="" type="checkbox"/> Admin	<input type="checkbox"/> Trainer	<input type="checkbox"/> Student
interactions	<input checked="" type="checkbox"/> Admin	<input type="checkbox"/> Trainer	<input type="checkbox"/> Student
data			
reviews	<input type="checkbox"/> Admin	<input type="checkbox"/> Trainer	<input checked="" type="checkbox"/> Student
page_times	<input type="checkbox"/> Admin	<input type="checkbox"/> Trainer	<input checked="" type="checkbox"/> Student
test_results	<input type="checkbox"/> Admin	<input type="checkbox"/> Trainer	<input checked="" type="checkbox"/> Student
time_on_pages	<input type="checkbox"/> Admin	<input type="checkbox"/> Trainer	<input checked="" type="checkbox"/> Student
lpr_services	<input type="checkbox"/> Admin	<input checked="" type="checkbox"/> Trainer	<input type="checkbox"/> Student

* 应用启动时将controller结构发送至user

权限控制

```
before_filter :check_access_right
```

```
def check_access_right
```

```
  unless xml_request? or inner_request?
```

```
    access_denied unless has_page_right?(params[:controller])
```

```
  end
```

```
end
```

一定程度上强迫设计良好的controller结构

user

权限控制



has_page_right?

只读user数据库

权限控制

```
class IdpRoleRight < ActiveRecord::Base
  acts_as_readonly :user, :table_name => "role_rights"
end

def has_page_right?(page)
  roles = current_user.roles
  roles_of_page = IdpRoleRight.all(:conditions => ["path = ?",
page]).map(&:role_id)
  (roles & roles_of_page).size > 0
end
```

user

gems!



config/environment.rb

```
config.gem 'idp_helpers'
```

```
config.gem 'idp_lib'
```

```
config.gem 'idp_core'
```



interface

data

user

service

支持系统

- File
- Mail
- Comet service

File

Specify Class that
Has Files

```
class Article < ActiveRecord::Base  
  has_files  
end
```

Upload File in
Background to
FileService

Idp_file_form

Store with
app_name,
model_name,
model_id

Use readonly
magic to easily
display

@article.files.first.url

Comet

```
class ChatRoom < ActiveRecord::Base  
  acts_as_realtime  
end
```

```
<%= realtime_for(@chat_room, current_user.login) %>
```

```
<%= realtime_data(dom_id, :add, :top) %>
```

```
@chat_room.realtime_channel.broadcast  
("hi world", current_user.login)
```

service

mail



Mail services

```
MailService.send("test@idapted.com", :welcome,  
:user => "test")
```


所有App在同一域下

www.eqenglish.com/user

www.eqenglish.com/studycenter

www.eqenglish.com/scenario

Projects

- ▼ v2-00_IdpGem
 - 1_TestServer
 - 2_Production
- ▶ v2-01_Assets
- ▶ v2-01_EQ_Portal
- ▶ v2-01_Flex
- ▶ v2-02_Board
- ▶ v2-02_CRM
- ▶ v2-02_Lpr
- ▶ v2-03_File
- ▶ v2-03_Mail
- ▶ v2-03_Realtime
- ▶ v2-03_System
- ▶ v2-03_User
- ▶ v2-04_Dialer
- ▶ v2-04_FS_Control
- ▶ v2-05_Interaction
- ▶ v2-05_Queue

Projects > Project v2-00_IdpGem > Stage 1_TestServer > Deployment

Deployment: _0_gem_install_idp_lib

Task: _0_gem_install_idp_lib

Stage: 1_TestServer

Deployed by: guolei

Revision: 24758

Started: 2010-05-25 14:42 (Beijing)

Completed: 2010-05-25 14:42 (Beijing)



success

Status: success » Repeat

Comment

Log

Log

Comment

结对部署

Advanced Search ID <input type="text"/> <input type="button" value="Search"/>								
Created At	Creator	Owner	App	Revision/Description	Has Migration	Merged At	Test Server Deployed At	Finished At
2010-05-25 14:10	denny	seven	trainer	24746 : add amanda pic to staff page 24748 : new trainer can not see the sp_shift 24755 : score for trainer on the base of assess 24756 : add the cronjob to score	true			
2010-05-24 16:06	denny	seven	trainer	24744 : point not work and late or absence	false	2010-05-24 16:07	2010-05-24 16:07	2010-05-25 14:12
2010-05-24 15:20	blade	guolei	crm	24739 : improve from page prompt	false	2010-05-24 15:20	2010-05-24 15:20	2010-05-24 15:20
2010-05-24 15:14	denny	guolei	interaction	24733 : get assess comment rake 24738 : assess comment task	false	2010-05-24 16:08	2010-05-24 16:08	2010-05-24 16:08
2010-05-24 14:51	blade	guolei	crm	24721 : improve advance search box in success story 24729 : improve stories search 24731 : improve advance search box in stories 24732 :	false	2010-05-24 15:08	2010-05-24 15:08	2010-05-24 15:20

如何拆分？

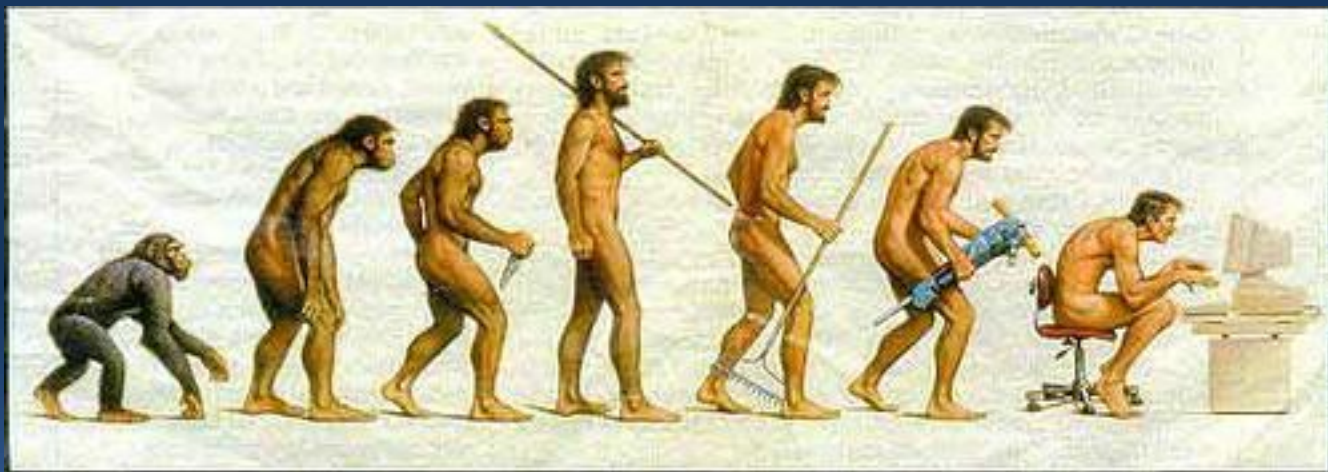
按照功能

每个App的功能应该是一组有限的、类似的
业务

示例

- User Management
- Course package
- Purchase
- Learning process
- ...

迭代



开始时尽可能细地拆分

如果发现

- 某些App之间交互太过频繁
- 为处理交互需要写很多代码

将其合并成一个

拆分准则

- 应用的核心功能不应该通过service来实现
- 应用不需要知道过多的其他应用的业务逻辑
- 独立的业务模型

缺点

- 应用需要在同一局域网下
- 某些情况下没有理想的解决方案

优点

- 更快的开发部署周期
- 更多的自主性
- 更灵活的应用新技术
- Scalable
- 更容易“jump in”
- 更舒服的开发，消除开发人员的惰性！

其他技术

- FreeSWITCH
 - VoIP
 - <http://www.freeswitch.org.cn>
- Flex
- Erlang
 - concurrent tasks
- Titanium
 - mobile

加入我们

1. VoIP
2. JavaScript

发送任何可以证明自己能力的材料至：
`tech-team-jobs@idapted.com`

谢谢大家！

Q&A

<http://developer.idapted.com>

jpalley@idapted.com (@jpalley)

guolei@idapted.com (@fiyuer)