

Python For Good

Lightning : 开源无代码Admin及低代码开发框架

@jeff_kit

Founder of Gitmen

2019年PyCon吹下的牛

PyCon China 2019 深圳分会场

324播放 · 1弹幕 2019-12-10 00:04:25



- 分享成果 —— 闪电数据管理
- 分享一种对开发者友好的低代码开发方法
 - 无代码的现代Admin
 - 云函数与触发器是低代码开发的常见形态
 - 以快速开发一个简易CRM为例
- 开源核心代码，回馈社区
 - 开源lightning仓库
 - 开源及演进lightning-crm，用为示范应用



闪电数据

—— 一键赋能企业数据 ——

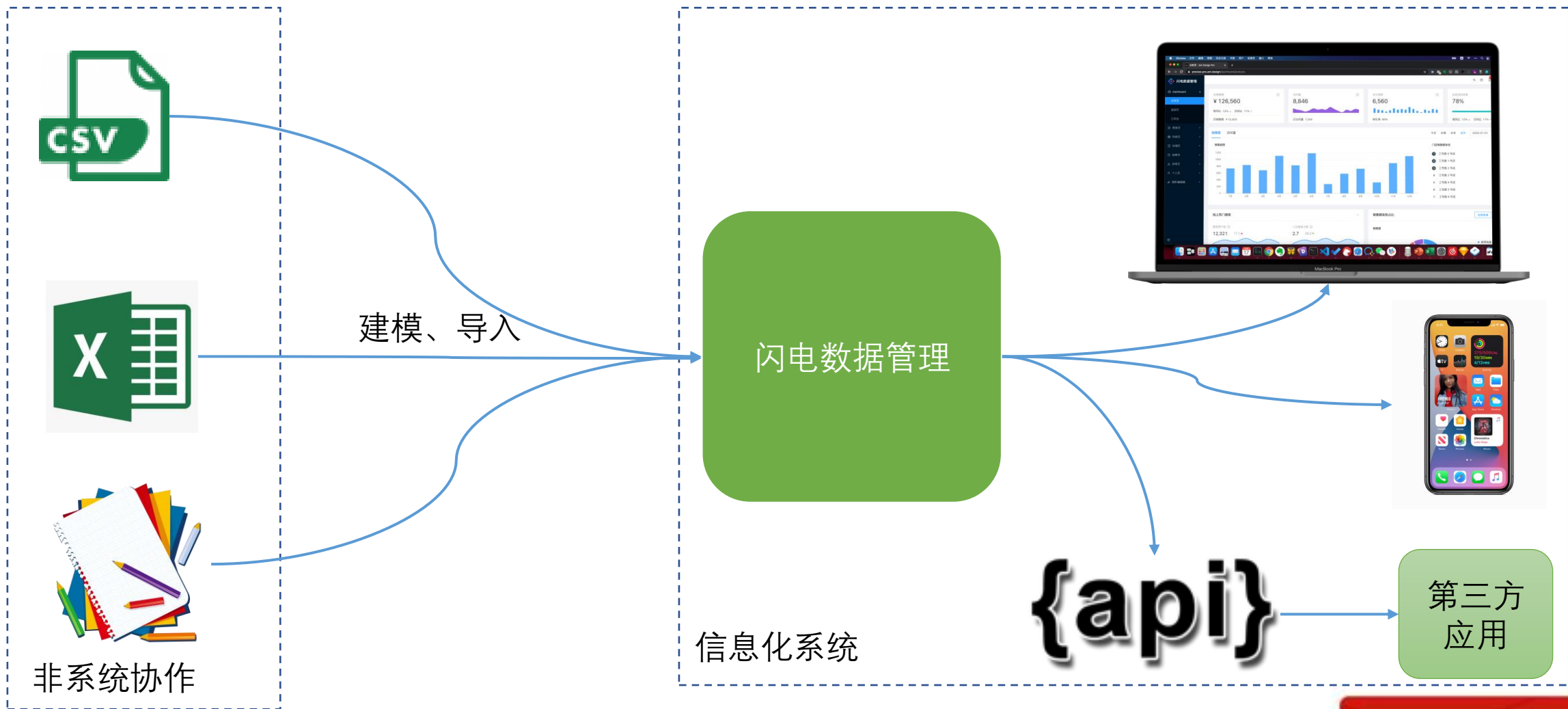
—— 一步开启数字商业 ——

企业全周期一站式低代码数字化运营平台

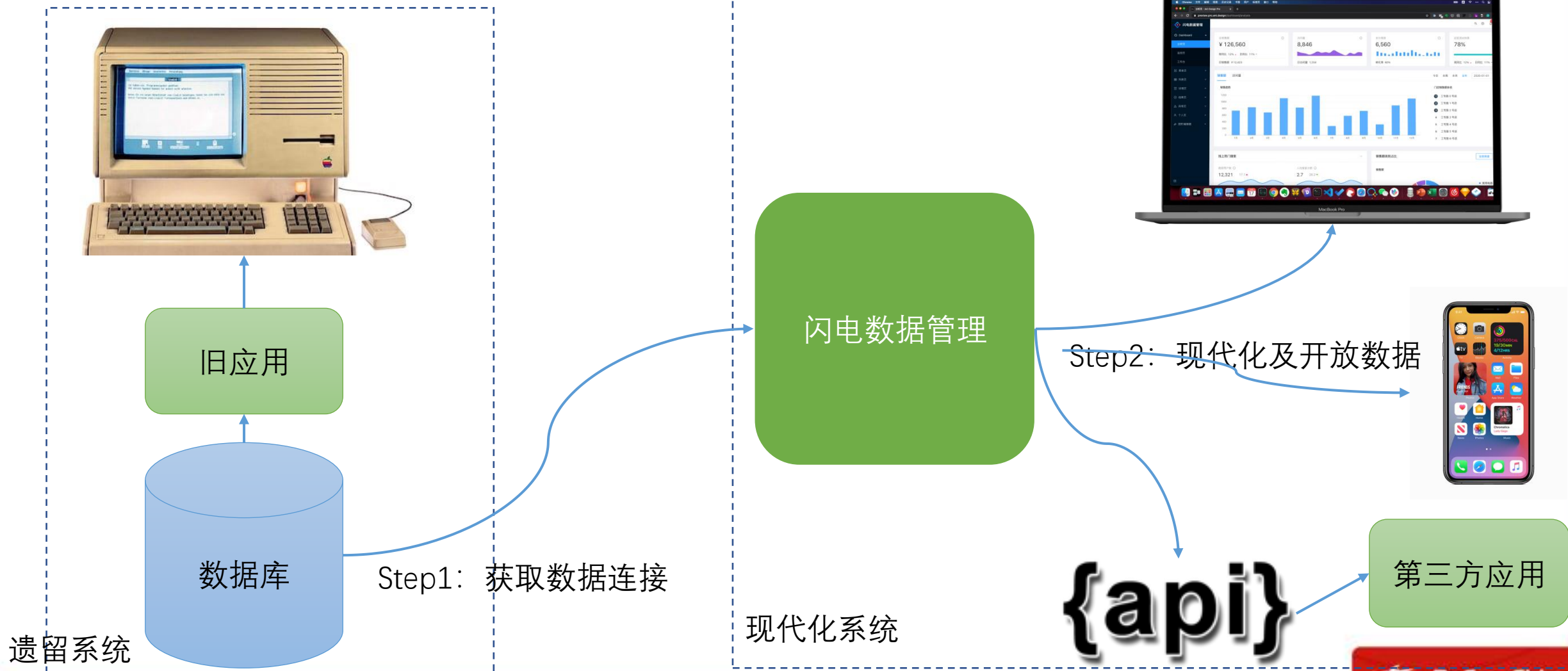
场景1：Admin



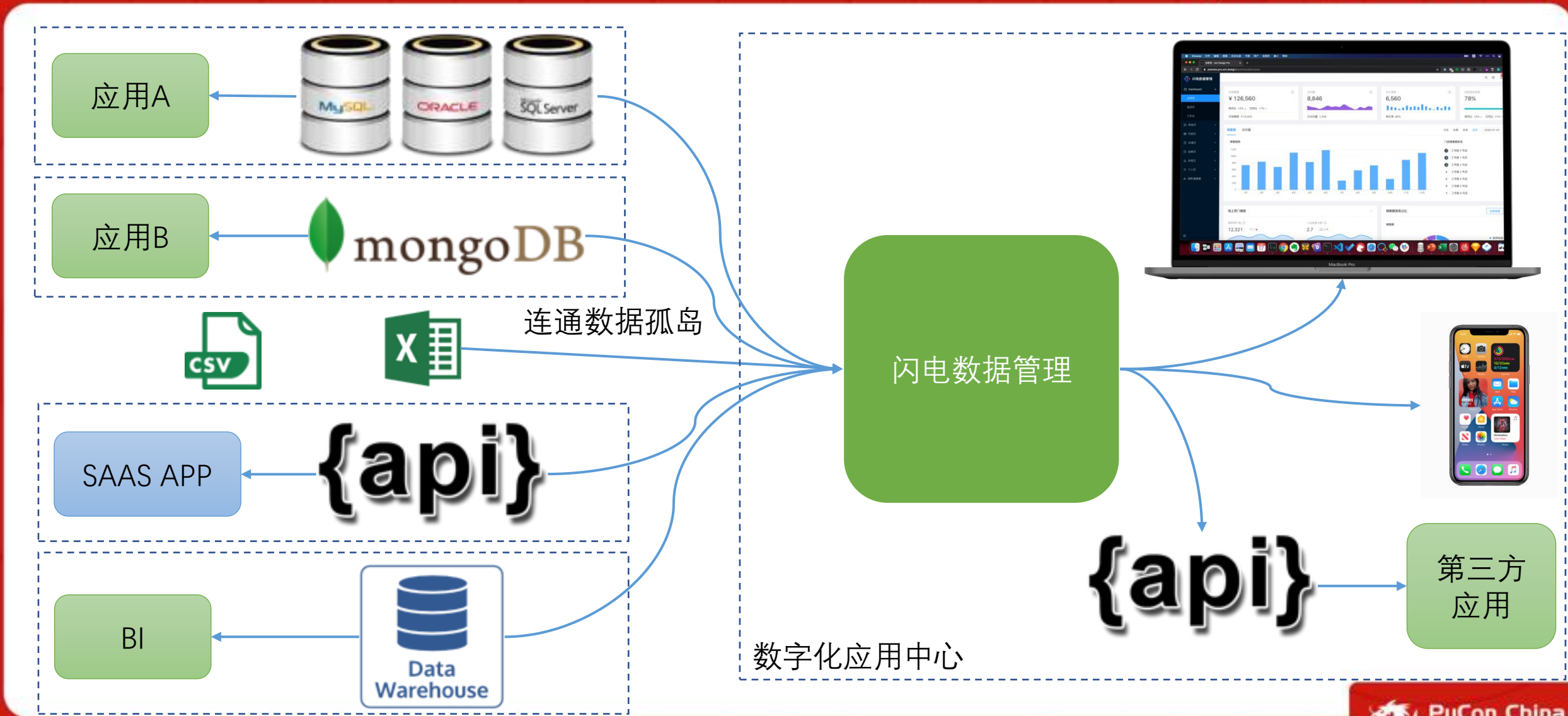
场景2：从0开始信息化



场景3：遗留系统现代化代



场景4：打通数据孤岛，构建数字化应用中心



闪电数据能力架构



开源后端代码
前端打包发布

理想的Admin —— 无代码的现代化的

- 无代码

- 仅编写模型代码，通过模型驱动一切
- 少量配置代码
- 无需编写页面逻辑代码
- 通过可视化配置的方式即配即可
- 无需重新部署

- 现代的

- 前后端分离，响应快速
- 设计精良，如Ant Design
- 友好的导航菜单
- 完善的功能页面，如详情页，可关联显示相关数据
- 图表、数据可视化

Lightning no-code 初体验

- 依赖:
- Python 3.6 +
- Django 2.2 良好测试 , Django3.x有待完整测试

- 安装:
- git clone [git@gitee.com/gitmen/lightning.git](https://gitee.com/gitmen/lightning.git) & cd lightning
- python setup.py install

- 创建Django项目
- Django-admin startproject crm

配置 settings.py

```
from lightning.settings import *  
import lightning  
INSTALLED_APPS += lightning.APPS
```

配置 urls.py

```
from django.contrib import admin  
from django.urls import path, include
```

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path("", include('lightning.urls')),  
]
```

放在最下面

- 创建CRM应用
 - Python manage.py startapp lightning_crm
- 编辑lightning_crm/models.py, 三个主要模型
 - 客户
 - 联系人
 - 跟进

```
class Customer(models.Model):
    name = models.CharField('名字', max_length=100)
    description = models.CharField('简介', max_length=1024, null=True, blank=True)
    avatar = ImageURLField('头像', blank=True, null=True)
    is_organization = models.BooleanField('是否机构', default=False)
    org_name = models.CharField('机构全称', max_length=200, null=True, blank=True)
    org_code = models.CharField('机构代码', max_length=200, null=True, blank=True)
    phone = models.CharField('电话', max_length=20, null=True, blank=True)
    address = models.CharField('地址', max_length=300, null=True, blank=True)
    email = models.EmailField('邮箱', max_length=100, null=True, blank=True)
    weixin = models.CharField('微信号', max_length=100, null=True, blank=True)
    qq = models.CharField('QQ号', max_length=100, null=True, blank=True)
    identity_code = models.CharField('身份证号', max_length=100, null=True, blank=True)
    stage = models.CharField('阶段', max_length=20, choices=settings.LIGHTNING_CRM_CUSTOMER_STAGES, default='')
    source = models.CharField('渠道来源', max_length=100, choices=settings.LIGHTNING_CRM_CUSTOMER_SOURCE, default='')
    referrer = models.ForeignKey('self', models.SET_NULL, null=True, blank=True, verbose_name='推荐客户')
    follwer = models.ForeignKey(settings.AUTH_USER_MODEL, models.SET_NULL, null=True, blank=True, verbose_name='跟进人',
                                on_delete=models.CASCADE)

    create_time = models.DateTimeField('创建时间', auto_now_add=True)
    update_time = models.DateTimeField('修改时间', auto_now=True)
    create_by = models.ForeignKey(settings.AUTH_USER_MODEL, models.SET_NULL, null=True, blank=True, verbose_name='创建人',
                                  on_delete=models.CASCADE)
```

```
class Contact(models.Model):
    customer = models.ForeignKey(settings.LIGHTNING_CRM_CUSTOMER_MODEL, models.SET_NULL,
        null=True, blank=True, verbose_name='客户')
    description = models.CharField('简介', max_length=1024, null=True, blank=True)
    name = models.CharField('名字', max_length=100)
    title = models.CharField('职位', max_length=100, null=True, blank=True)
    avatar = ImageURLField('头像', blank=True, null=True)
    phone = models.CharField('电话', max_length=20, null=True, blank=True)
    address = models.CharField('地址', max_length=300, null=True, blank=True)
    email = models.EmailField('邮箱', max_length=100, null=True, blank=True)
    weixin = models.CharField('微信号', max_length=100, null=True, blank=True)
    qq = models.CharField('QQ号', max_length=100, null=True, blank=True)
    identity_code = models.CharField('身份证号', max_length=100, null=True, blank=True)

    create_time = models.DateTimeField('创建时间', auto_now_add=True)
    create_by = models.ForeignKey(settings.AUTH_USER_MODEL, models.SET_NULL, null=True,
        blank=True, verbose_name='创建人')
```



```
class FollowUp(models.Model):
    customer = models.ForeignKey(settings.LIGHTNING_CRM_CUSTOMER_MODEL, models.SET_NULL,
        null=True, blank=True, verbose_name='客户')
    action_type = models.CharField('跟进类型', max_length=50,
        choices=settings.LIGHTNING_CRM_FOLLOWUP_TYPE, default='')
    content = models.TextField('跟进内容', null=True, blank=True)
    images = models.ManyToManyField(Image, blank=True, verbose_name='图片')
    contacts = models.ManyToManyField(Contact, blank=True, verbose_name='联系人')
    follow_up_time = models.DateTimeField('创建时间', auto_now_add=True)
    follow_up_by = models.ForeignKey(settings.AUTH_USER_MODEL, models.SET_NULL,
        null=True, blank=True)
```

migrate

(lightning-app) → `crm python manage.py migrate`

Operations to perform:

Apply all migrations: admin, api_basebone, auth, bsm_config, contenttypes, guardian, lightning_crm, sessions, shield

Running migrations:

```
Applying contenttypes.0001_initial... OK
Applying auth.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Applying admin.0003_logentry_add_action_flag_choices... OK
Applying api_basebone.0001_initial... OK
Applying api_basebone.0002_auto_20190524_1155... OK
Applying api_basebone.0003_api_field_filter_parameter... OK
Applying api_basebone.0004_auto_20190910_2242... OK
Applying api_basebone.0005_auto_20190910_2244... OK
Applying api_basebone.0006_auto_20190910_2246... OK
Applying api_basebone.0007_auto_20190910_2247... OK
Applying api_basebone.0008_auto_20190910_2254... OK
Applying api_basebone.0009_auto_20190911_1547... OK
Applying api_basebone.0010_filter_layer... OK
Applying api_basebone.0011_auto_20190924_1142... OK
Applying api_basebone.0012_auto_20190930_1715... OK
Applying api_basebone.0013_auto_20191012_1642... OK
Applying api_basebone.0014_auto_20191014_1518... OK
Applying api_basebone.0015_parameter_is_array... OK
Applying api_basebone.0016_auto_20191022_1928... OK
Applying api_basebone.0017_auto_20191101_1829... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
```

创建帐号, light it up!

```
(lightning-app) → crm python manage.py createsuperuser
```

用户名 (leave blank to use 'jeff'):

电子邮件地址: jeff@gitmen.com

Password:

Password (again):

Superuser created successfully.

创建管理员帐号

```
(lightning-app) → crm python manage.py light lightning_crm
```

lightning admin config successfully generated!

```
(lightning-app) → crm python manage.py runserver
```

为app生成管理界面

浏览 http://localhost:8000

Python For Good

菜单管理 - 客户管理系统

localhost:8000/content/bsm_config__menu

客户管理系统

- 权限管理
- 开发管理
- 客户管理
 - 客户管理
 - 跟进管理
 - 联系人管理
 - 图片管理

+ 新建

名称	页面类型	关联模型	操作
+ 权限管理			编辑 删除
- 开发管理			编辑 删除
菜单管理	列表页	bsm_config__menu	编辑 删除
页面配置	页面配置面板		编辑 删除
- 客户管理			编辑 删除
客户管理	列表页	lightning_crm__customer	编辑 删除
跟进管理	列表页	lightning_crm__followup	编辑 删除
联系人管理	列表页	lightning_crm__contact	编辑 删除
图片管理	列表页	lightning_crm__image	编辑 删除

自动生成的功能页面

Python For Good

客户管理系统

localhost:8000/content/lightning_crm__customer/detail/3

客户管理系统

权限管理

开发管理

客户管理

客户管理

跟进管理

图片管理


联系人管理

首页 / 客户管理 / 客户管理

← Jeff

编辑 删除 详情

详情

ID: 3	名字: Jeff	简介: 无	头像: 
是否机构: <input checked="" type="checkbox"/>	机构全称: 无	机构代码: 无	电话: 13570410587
地址: 无	邮箱: 无	微信号: 无	QQ号: 无
身份证号: 无	阶段: 试用	渠道来源: 公众号	引荐客户: 无
跟进人: 无	创建时间: 2020-11-27 21:09:57	修改时间: 2020-11-27 21:09:57	创建人: 2

闪电数据管理 Gitmen.Inc

Copyright © 2018 广东杰明科技出品

在线调整后的功能页面

Python For Good

The screenshot displays a web browser window with the following elements:

- Browser Tab:** 客户管理系统
- Address Bar:** localhost:8000/content/lightning_crm__customer/detail/3/50zDes
- System Header:** 客户管理系统
- Left Sidebar:**
 - 权限管理
 - 开发管理
 - 客户管理
 - 客户管理 (highlighted)
 - 跟进管理
 - 图片管理
 - 联系人管理
- Modal Form (新建跟进):**
 - * 跟进类型: [Dropdown menu]
 - 跟进内容: [Text area]
 - 联系人: [Input field with placeholder text "搜索联系人"]
 - 图片: [Image upload area with a "+" icon and "上传" text]
 - Buttons: 提交 (Submit), 重置 (Reset)
- Background Page (Customer Detail):**
 - Buttons: 编辑 (Edit), 删除 (Delete), 详情 (Details)
 - Search: 查询 (Search), 重置 (Reset)
 - Table (partially visible):

时间	操作
11-27 22:32:39	编辑 删除
11-27 22:05:57	编辑 删除
 - Page Navigation: 1 / 30条/页

页面配置面板

Python For Good

客户管理 - 客户管理系统

localhost:8000/content/lightning_crm__customer

客户管理系统




首页 / 客户管理 / 客户管理

客户列表

名字: 是否

渠道来源: 跟进

[+ 新建](#)

客户	是否机构	机构全称
 Jeff	<input checked="" type="checkbox"/>	
 Google 谷歌	<input checked="" type="checkbox"/>	Google
 jtme 杰明科技	<input checked="" type="checkbox"/>	深圳杰明科技有

列表页设置

表单设置

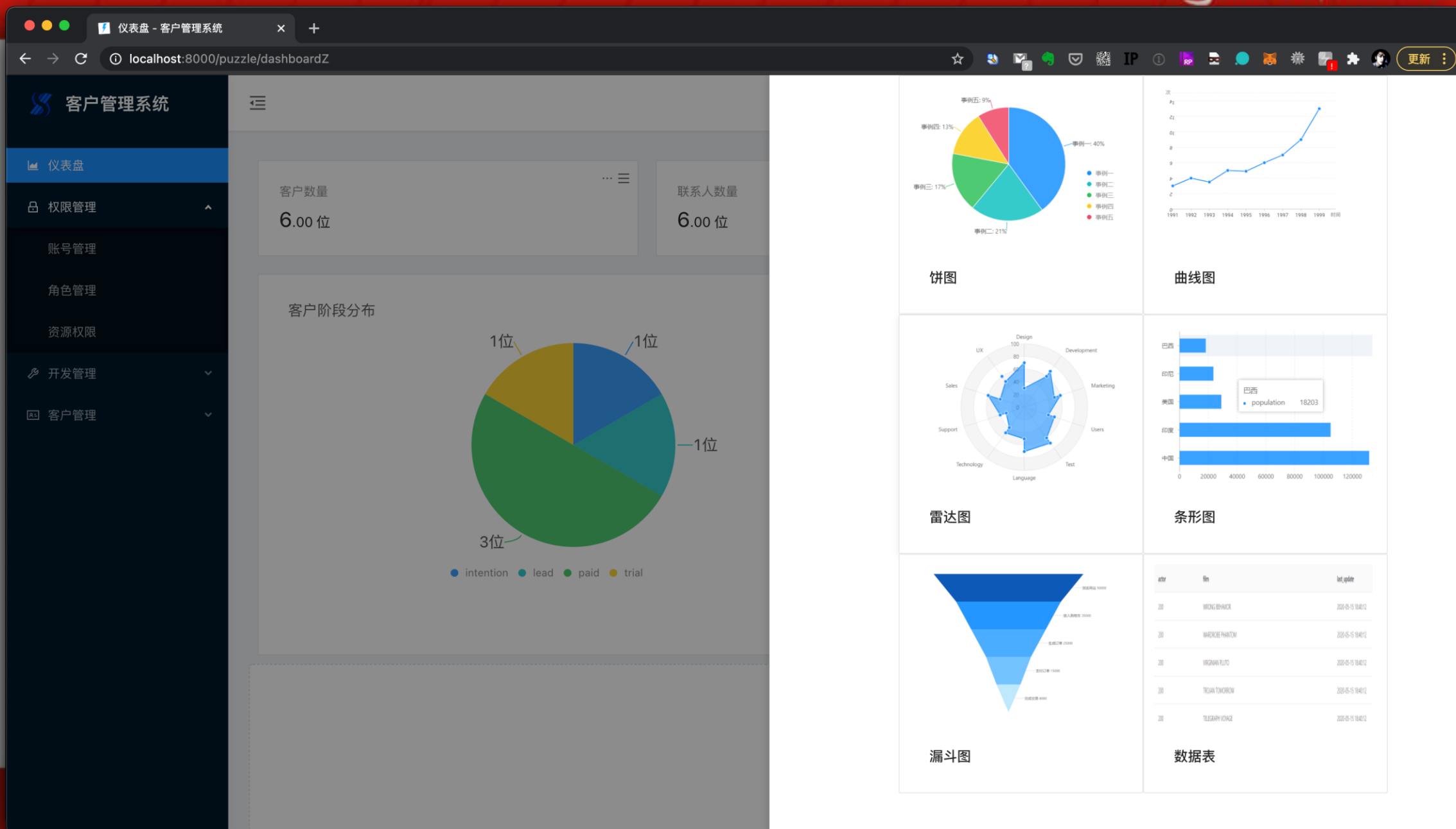
详情页设置

导出设置

列表字段

	字段	操作
+	请选择 <input type="text"/>	删除
+	是否机构 <input type="text"/>	删除
+	机构全称 <input type="text"/>	删除
+	阶段 <input type="text"/>	删除
+	渠道来源 <input type="text"/>	删除
+	跟进人 <input type="text"/>	删除
+	创建人.用户名 <input type="text"/>	删除

噢，在线配置图表



- 常规API, 在线配置生成
- 云函数, 非常规API, 通过编写函数的方式提供
- 触发器 (或信号)
- No more ...
 - MVC
 - MTV
 - Template, HTML

常规API的配置

Python For Good

2020

API管理 - 客户管理系统

localhost:8000/content/api_db__api

客户管理系统

仪表盘

权限管理

开发管理

菜单管理

API管理

客户管理

首页 / 开发管理 / API管理

Api接口模型列表

+ 新建

ID	操作	接口标识	名称	操作
1	列表	customers	所有客户	编辑 删除 请求地址 查看文档

共1条数据 < 1 > 30条/页

闪电数据管理 Gitmen.Inc
Copyright © 2018 广东杰明科技出品

China

自动生成 Open API 互动文档

Python For Good

POST /api/create_post/ 创建一张海报吧

DELETE /api/delete_text/

GET /api/film_detail/ 读取电影的详情

GET /api/films/ 列出所有电影清单

Parameters Try it out

Name	Description
pageNumber integer (query)	页码参数
<input type="text" value="pageNumber - 页码参数"/>	
pageSize integer (query)	页长参数
<input type="text" value="pageSize - 页长参数"/>	

Responses

云函数 – 解决常规API未能满足的接口

```
@bsm_func(login_required=True, name='user_share_qr_code', model=Agency)
def user_share_qr_code(user, appid, **kwargs):
    return wechat.wxa(appid).get_wxa_code_unlimited(
        scene=f'{user.source or ""};{user.id}', page='pages/home/index',
        is_hyaline=True
    )
```

```
curl -X POST -H "Content-Type: application/json" -d '{"appid":"abc"}' \
http://localhost:8000/api_basebone/shop_agency/func/
```

触发器 – 数据变更时的自动化流程机制

```
from django.dispatch import Signal

post_bsm_create = Signal(providing_args=['instance', 'create', 'request',
    'old_instance'])

before_bsm_create = Signal(providing_args=['instance', 'create', 'request', 'old_instance'])

post_bsm_delete = Signal(providing_args=['instance', 'request'])

before_bsm_delete = Signal(providing_args=['instance', 'request'])
```

开源计划 12.3

- 团队正在做大量的开源前准备工作
- 12月3日正式发布
- MIT授权
- 后端代码开源，前端作为资源打包入内。
- 招募各种志愿者，求PR



THANK YOU



Jeff Kit

广东 广州



扫一扫上面的二维码图案，加我微信