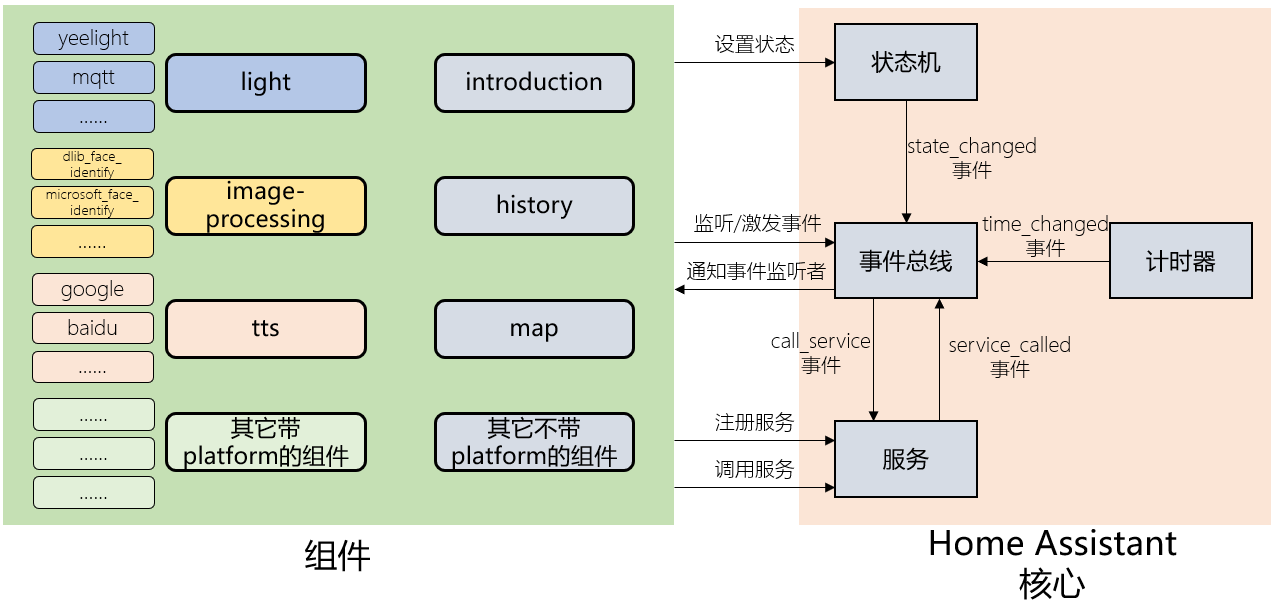
组件的工作原理

【操作步骤】

1. HomeAssistant核心与它的API
2. 第一个组件程序
3. 加载第一个组件程序
4. 组件的加载过程
5. 什么是platform？

【参考】

* HomeAssistant内部架构图



* 组件样例程序

def setup(hass, config):

# 设置实体hachina.hello\_world的状态

hass.states.set('hachina.hello\_world', '太棒了！')

# 调用服务persistent\_notification.create

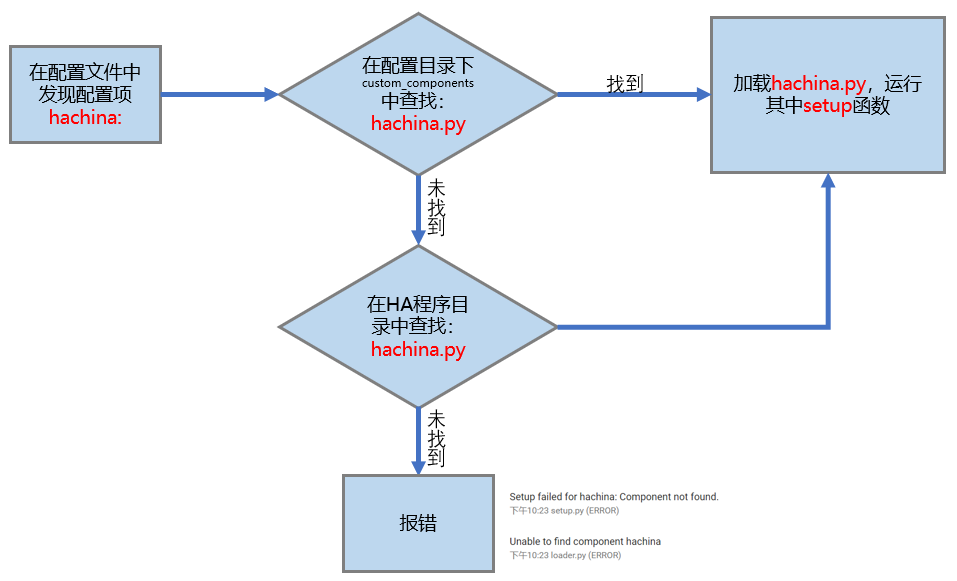
service\_data = { "title":"服务被调用",

"message":"这是调用服务后显示的信息"}

hass.services.call('persistent\_notification', 'create', service\_data)

return True

* 组件加载过程



* HomeAssistant的核心机制

<https://www.hachina.io/docs/342.html>

* HomeAssistant带有平台的组件域

|  |  |  |  |
| --- | --- | --- | --- |
| alarm\_control\_panel | binary\_sensor | calendar | camera |
| climate | cover | device\_tracker | fan |
| image\_processing | light | lock | mailbox |
| media\_player | notify | remote | scene |
| sensor | switch | telegram\_bot | tts |
| vacuum | weather |  |  |

程序样例：二维码识别组件

【操作步骤】

1. 使用pillow与pyzbar库
2. HomeAssistant的image\_processing域
3. qr.py程序
4. 配置与加载image\_processing域中的qr平台

【参考】

* Python下的pyzbar库

<https://github.com/NaturalHistoryMuseum/pyzbar/>

* HomeAssistant中组件程序源代码

<https://github.com/home-assistant/home-assistant/tree/dev/homeassistant/components>

* 配置文件样例

camera:

- platform: rpi\_camera

name: cam5

vertical\_flip: 1

image\_processing:

- platform: qr

scan\_interval: 2

source:

- entity\_id: camera.cam5

name: QRCode

Python程序员的成长与代码规范

【操作步骤】

1. 一些教程
2. github与HomeAssistant源程序
3. python代码规范

【参考】

* 廖雪峰Python教程

<https://www.liaoxuefeng.com/wiki/0014316089557264a6b348958f449949df42a6d3a2e542c000>

* HomeAssistant组件编写StepByStep

<https://www.hachina.io/docs/1891.html>

* 一些检查程序规范的应用安装

sudo pip3 install pylint flake8 pydocstyle

把你的代码贡献给组织

【操作步骤】

1. fork home-assistant项目
2. 构建本地开发环境
3. 本地库修改与测试
4. 提交你的贡献
5. 提交网站文档修改

【参考】

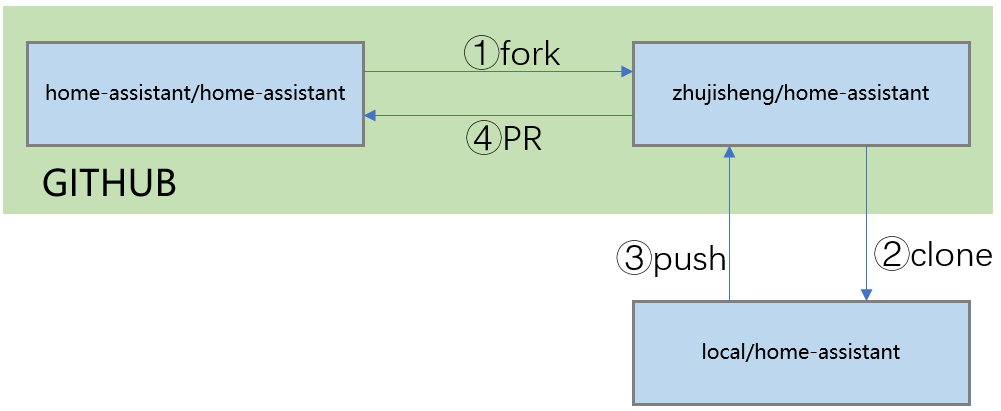
* git工具的介绍

<https://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000>

* HomeAssistant开发者指导

<https://developers.home-assistant.io/docs/en/development_index.html>

* 核心过程



* 构建开发环境命令
* 安装基础软件库

sudo apt update

sudo apt-get install python3-pip python3-dev python3-venv

sudo apt-get install autoconf libssl-dev libxml2-dev libxslt1-dev libjpeg-dev libffi-dev libudev-dev zlib1g-dev

* 创建/进入开发目录

mkdir my\_development

cd my\_development

* 克隆远程库

git clone https://github.com/YOUR\_GIT\_USERNAME/home-assistant.git

cd home-assistant

git remote add upstream <https://github.com/home-assistant/home-assistant.git>

* 构建python虚拟环境，从源码安装HomeAssistant

python3 -m venv .

source bin/activate

pip3 install wheel

script/setup

* 修改与本地测试（以增加image\_processing.qrcode为例）
* 创建工作分支

git checkout -b qrcode

* 修改

vi homeassistant/components/image\_processing/qrcode.py # 编辑组件程序

script/gen\_requirements\_all.py # 生成新的[requirements\_all.txt](https://github.com/zhujisheng/home-assistant/blob/some-feature/requirements_all.txt)

vi .coveragerc # 编辑测试覆盖配置文件

* 代码规范检查

pip3 install -r requirements\_test.txt

flake8 homeassistant/components/image\_processing/qrcode.py

pydocstyle homeassistant/components/image\_processing/qrcode.py

pylint homeassistant/components/image\_processing/qrcode.py

* 完整测试

tox # 完整测试命令，运行时间非常长

* 确认修改

git add .

git commit -m "Added some-feature" --amend --author="YourName <youremail@xxx.xxx>"

* 提交你的贡献
* 获取最新官方版本内容

git fetch upstream dev # 获取官方的当前分支

git rebase upstream/dev # 调基

* 将修改发布到你fork的项目中

git push origin HEAD

* 创建Pull Request（在github网站上进行）

本视频中创建的PR链接：<https://github.com/home-assistant/home-assistant/pull/20215>