

02

OPEN ORIENTED

凹凸实验室

《chrome 扩展程序开发》

— O2HOST

张燕婷
2017.03.20



chrome 扩展程序

作用、组成、用户界面、架构、API、调试、打包、托管/发布



O2HOST 平台

功能、原理、实现、演示



chrome 扩展程序

为 Google Chrome 浏览器增加功能

- 弹出窗口，让用户执行操作，或者显示信息
 - 向网页中注入脚本文件，执行某些功能
- 调用 chrome 提供的 native api（tab 页开关、窗口开关、文件下载等）

[Web Apps（应用）]

独立的用户界面、典型且丰富的用户互动

[扩展程序]

丰富浏览器或网站的功能、一般来说适用于所有网站

组成

- 配置清单文件（1）：manifest.json *
- HTML 文件（1 – N）：除非是一个主题背景 *
- JavaScript 文件（1 – N）
- 其他文件：图片

本质

网页：可以利用浏览器为网页提供的所有 API

标识符

扩展程序生成的唯一标识符

- 打包前（编写）：扩展程序标识符可能会更改
- 打包时：扩展程序将会获得一个永久的标识符

```
{
//=====必选=====
  "manifest_version": 2, // 清单文件格式的版本 2 (版本 1 已弃用)
  "name": "hostPlatform", // 扩展程序名称
  "version": "1.0", // 扩展程序版本号

//=====可选=====
  "description": "Switch hosts! ", // 扩展程序描述
  "permissions": [
    "proxy",
    "tabs",
    "storage" // 此权限支持读取标签 (tab) 中的信息 ( "http://api.gdfans.net/"
    支持发送 Ajax 请求)
  ],
  "browser_action": { // 用户界面按钮形式 (browser_action 或 page_action)
    "default_icon": "icon.png", // 图标
    "default_popup": "popup.html" // html 页面路径
  },
  "icons": {
    "128": "icon120.png" // 128 像素大小的图标的路径
  },
  "content_security_policy": "script-src 'self' 'unsafe-eval' https://
    cdn1.lncld.net/static/js/2.1.3/av-min.js; object-src 'self'" //
  CSP设置
}
```

用户界面（按钮）

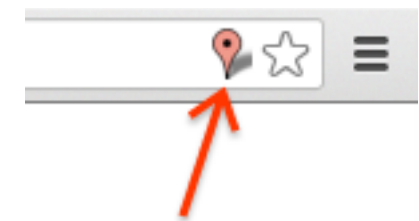
浏览器按钮

- 扩展程序与大部分网页相关
- 在浏览器主窗口地址栏右侧工具栏中添加图标



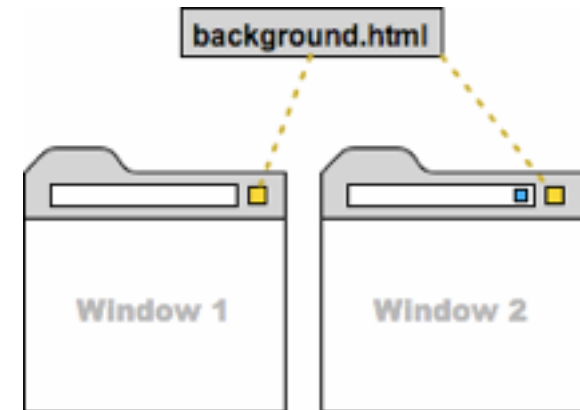
页面按钮

- 扩展程序的图标显示还是消失取决于具体网页
- 代表用于当前网页的操作，不适用于所有网页

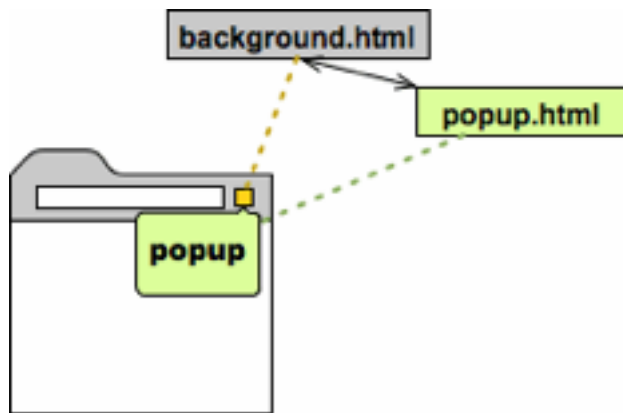


后台页面 扩展程序主要逻辑

- 持续运行的后台网页：保持打开状态
- 事件页面：根据需要打开与关闭（首选）



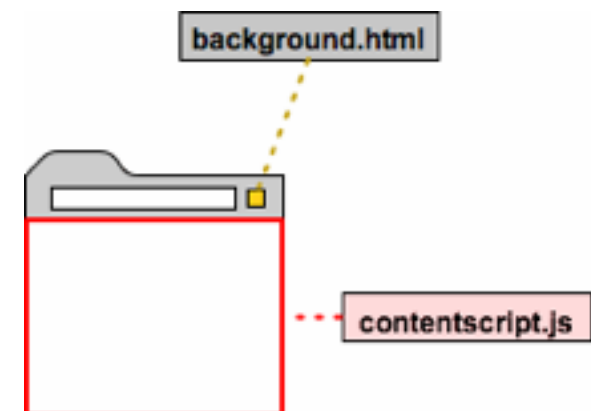
用户界面页面 展现扩展程序



- 浏览器按钮：弹出菜单
- 选项页面：用户自定义扩展程序的工作方式
- 替代页面
- 其他页面：使用 `tabs.create` 或 `window.open()` 来显示

内容脚本 与用户加载的网页交互

- 在浏览器中已加载页面的上下文中执行（看做已加载页面的一部分）
- 不是完全与所属扩展程序隔离的（可以与所属扩展程序交换消息）



- 网页可以使用的所有 API
- `chrome.*` API (Chrome 浏览器的 API)
 - 大部分方法都是异步的

[`chrome.*` API]

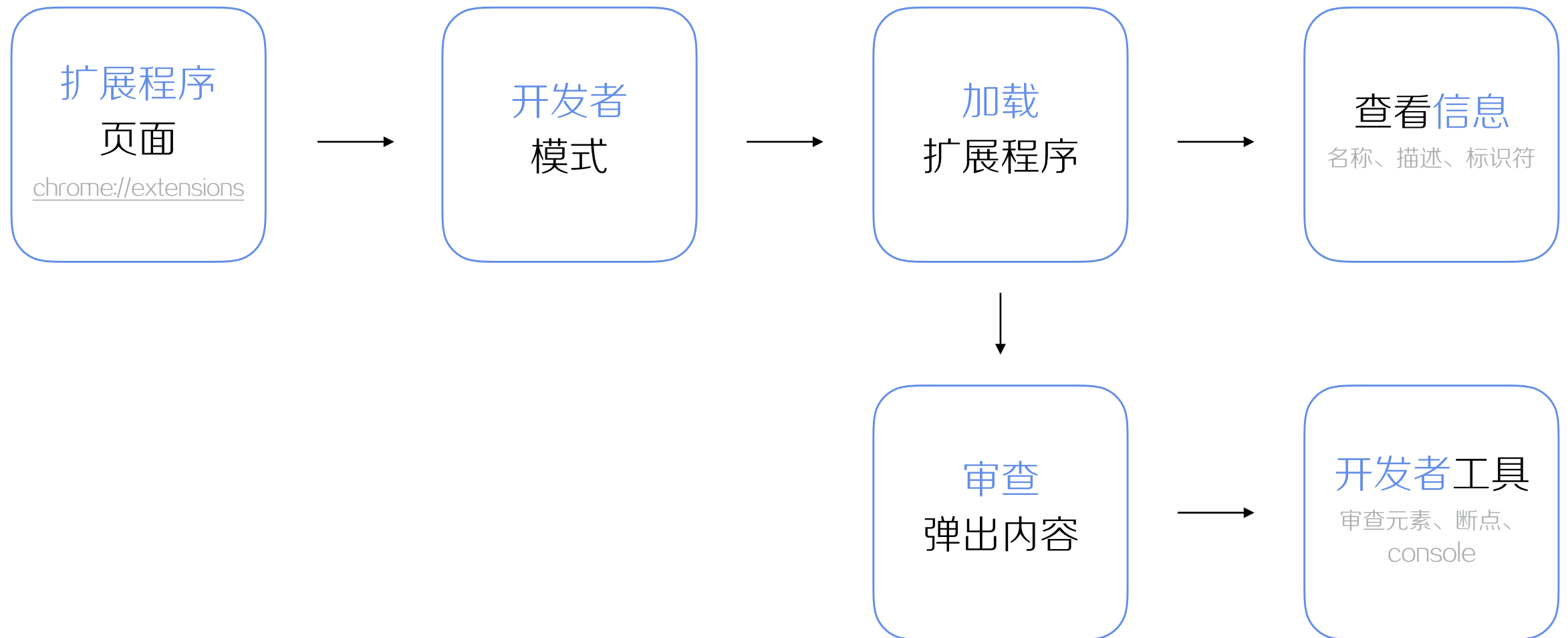
`chrome.proxy`: 管理 Chrome 浏览器的代理服务器设置

`chrome.types`: 用于 Chrome 浏览器的类型声明

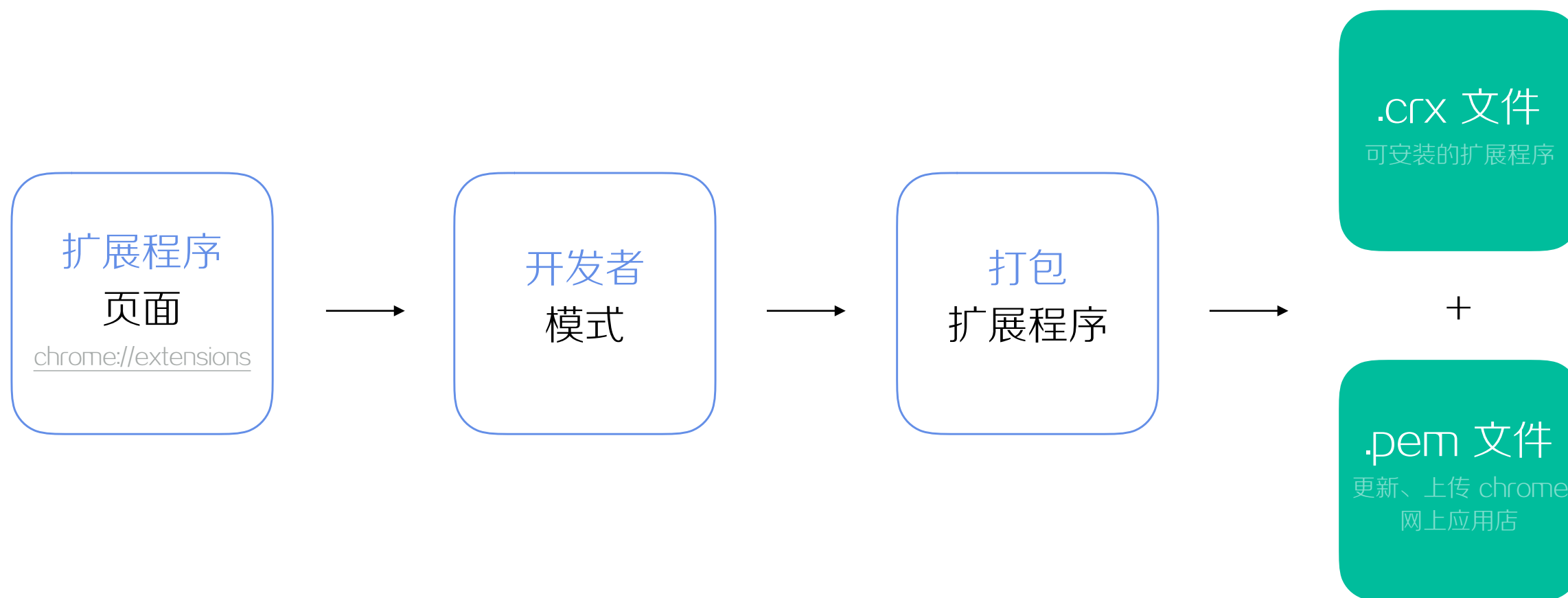
(Chrome 浏览器设置 `ChromeSetting`原型提供一组常用函数 `get()`、`set()` 和 `clear()` 以及一个事件发布者 `onChange`)

`chrome.storage`: 存储、获取用户数据，追踪用户数据的更改 (对象)

`chrome.tabs`: 与浏览器标签页交互，创建、修改或重新排列浏览器中的标签页



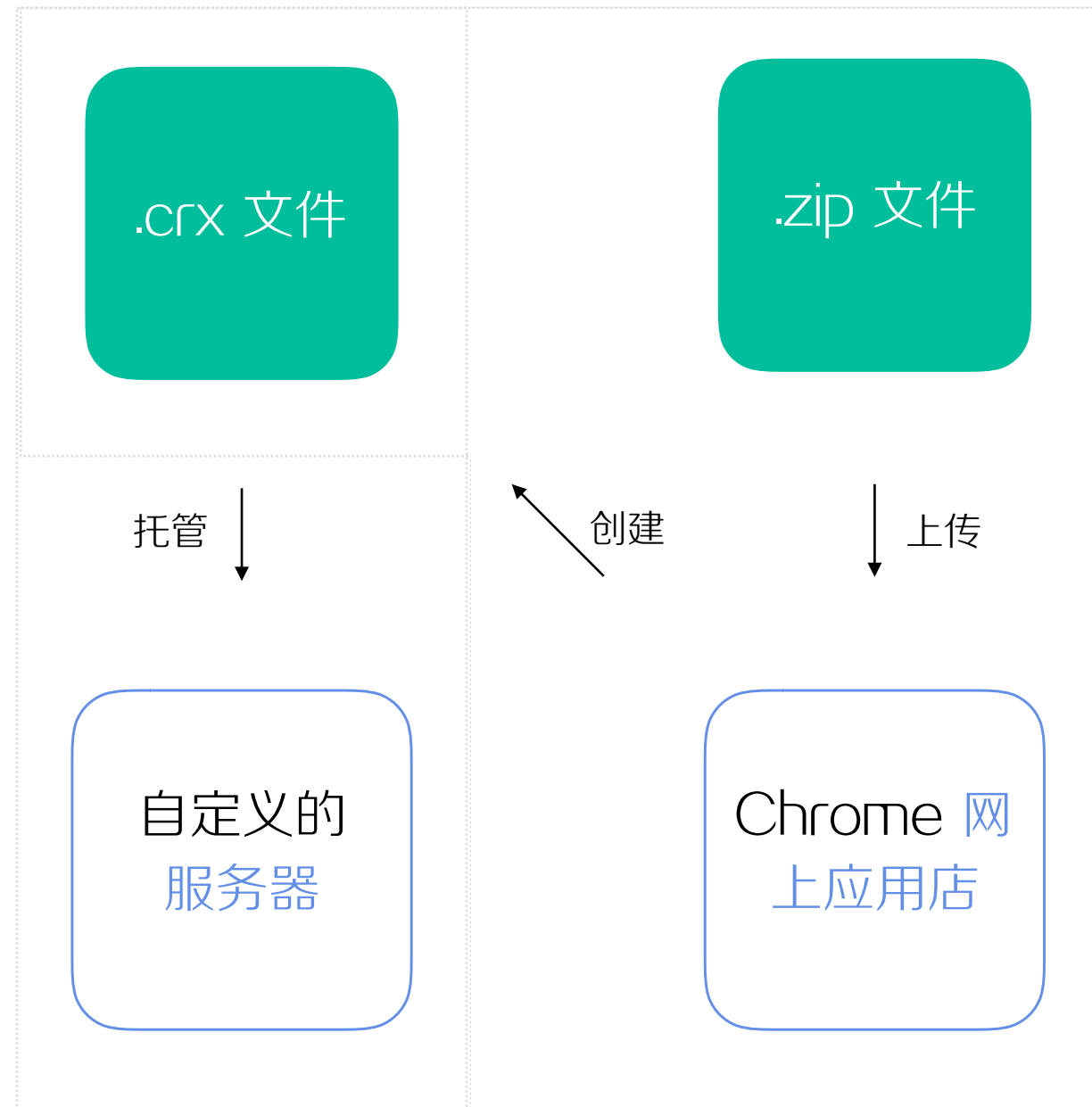
打包



托管/发布

02

- 扩展程序
- 可安装的网上应用
- 主题背景





O2HOST 平台

HOST PLATFORM

配置不同业务下多个方案的 host 规则

Chrome 扩展程序

查看/设置当前生效的 host 规则

[解决问题]

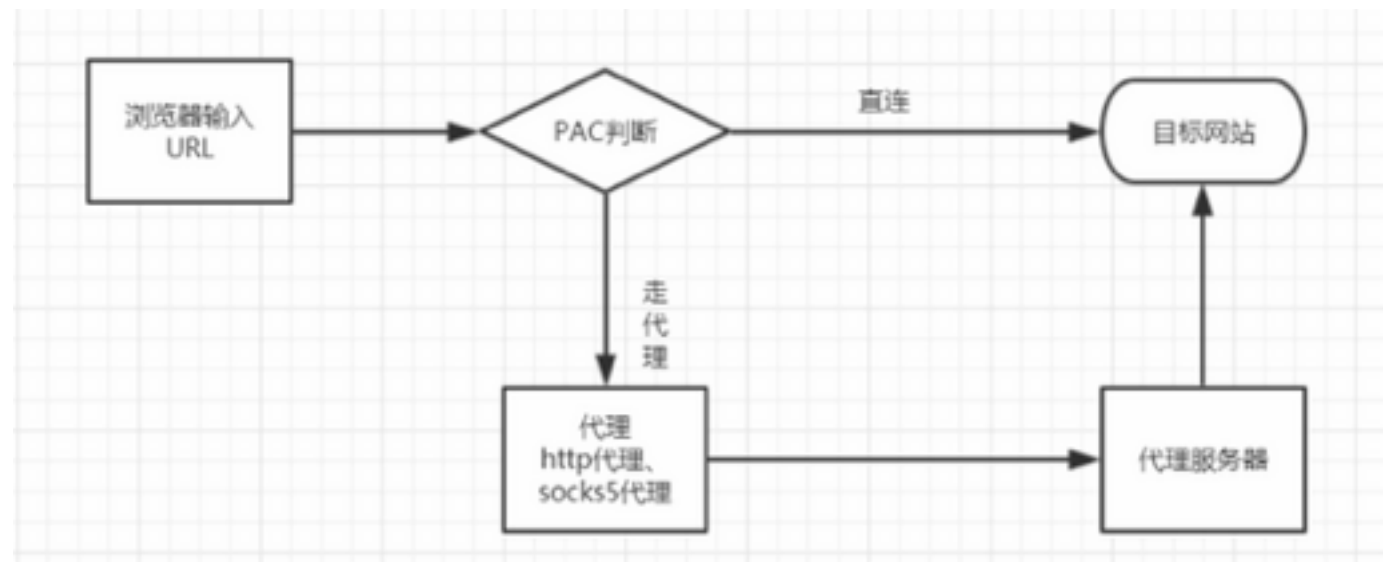
开发人员：开发过程中过程中 HOST 配置繁琐、多程序

非开发人员：不熟悉 HOST 配置

PAC (Proxy auto-config)

— 代理自动配置

- 定义浏览器该如何自动选择适当的代理服务器来访问一个网址



[PAC文件]

包含一个JavaScript形式的函数 “FindProxyForURL(url, host)”

返回一个包含 一个或多个访问规则的字符串

```
function FindProxyForURL(url, host) {  
    // our local URLs from the domains below example.com don't need a proxy:  
    if (shExpMatch(url, "*.example.com/*")) {return "DIRECT";}   
    if (shExpMatch(url, "*.example.com:*/*")) {return "DIRECT";}   
  
    // URLs within this network are accessed through  
    // port 8080 on fastproxy.example.com:  
    if (isInNet(host, "10.0.0.0", "255.255.248.0")) {  
        return "PROXY fastproxy.example.com:8080";  
    }  
  
    // All other requests go through port 8080 of proxy.example.com.  
    // should that fail to respond, go directly to the WWW:  
    return "PROXY proxy.example.com:8080; DIRECT";  
}
```


用户界面

浏览器
按钮

+

弹出
窗口

脚本

popup.js
主要逻辑

+

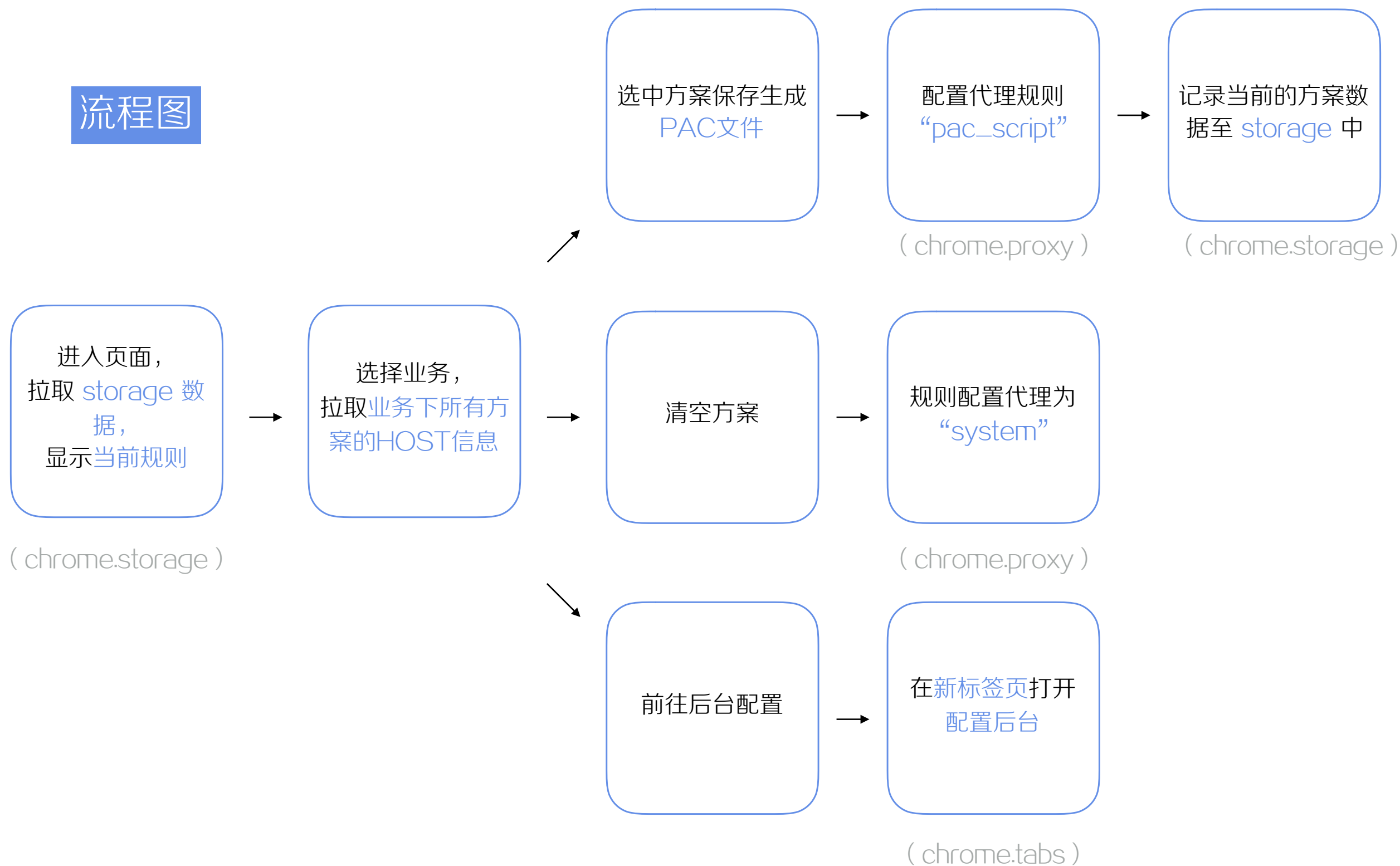
leancloud.js
leancloud配置

+

leancloud
外链sdk

CSP问题:
《Chrome Extension 中的
CSP (Content Security
Policy) 开发小记》

流程图



《入门：建立 Chrome 扩展程序》系列

《Chrome（谷歌）浏览器扩展开发初探》

《vue与chrome浏览器插件开发》

《Chrome Extension 开发初探》

《分享Chrome Extension（扩展程序/插件）开发的一些小经验》

《PAC自动代理文件格式，教你如何写PAC文件》

《Chrome Extension 中的 CSP（Content Security Policy）开发小记》

《unsafe-eval security error in chrome extension》

THANKS
FOR YOUR WATCHING

02

OPEN ORIENTED

凹凸实验室