

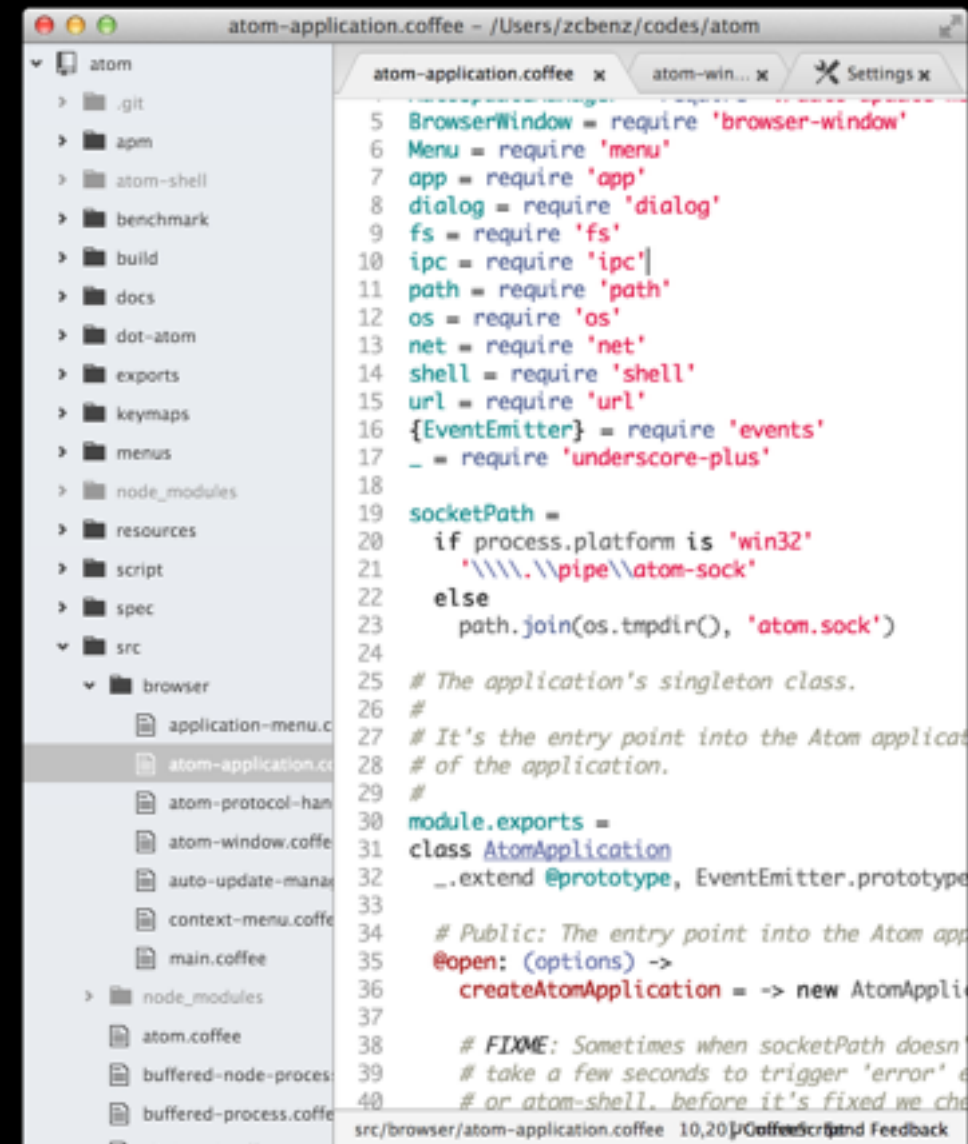
Atom编辑器嵌入 Node.js引擎实践

赵成

<https://github.com/zcbenz>

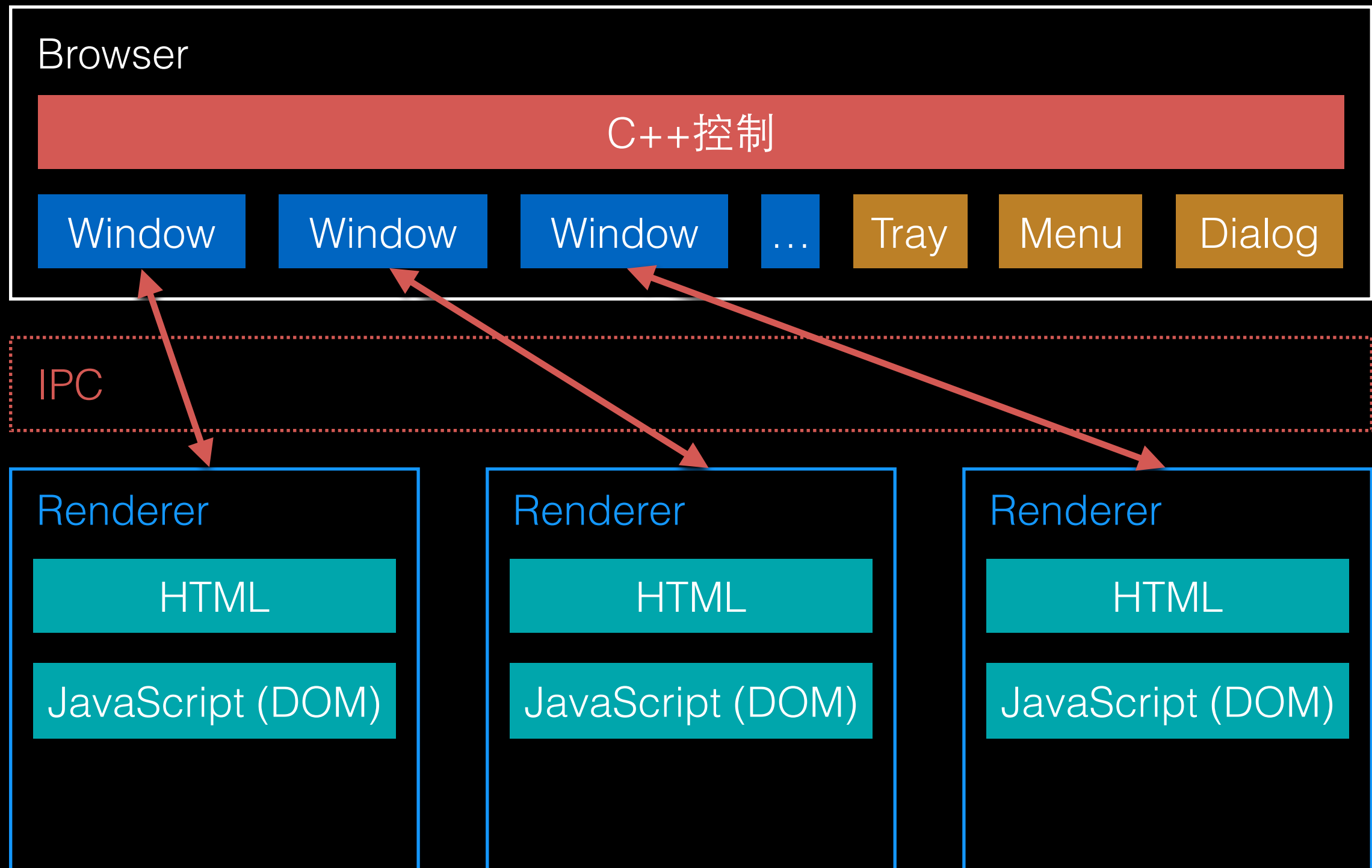
Atom编辑器和Atom Shell

- Atom是GitHub出品的文本编辑器
<https://github.com/atom/atom>
- Atom Shell是基于Chromium和Node.js的桌面软件开发平台
<https://github.com/atom/atom-shell>
- Atom在Atom Shell之上构建，并使用CoffeeScript编写

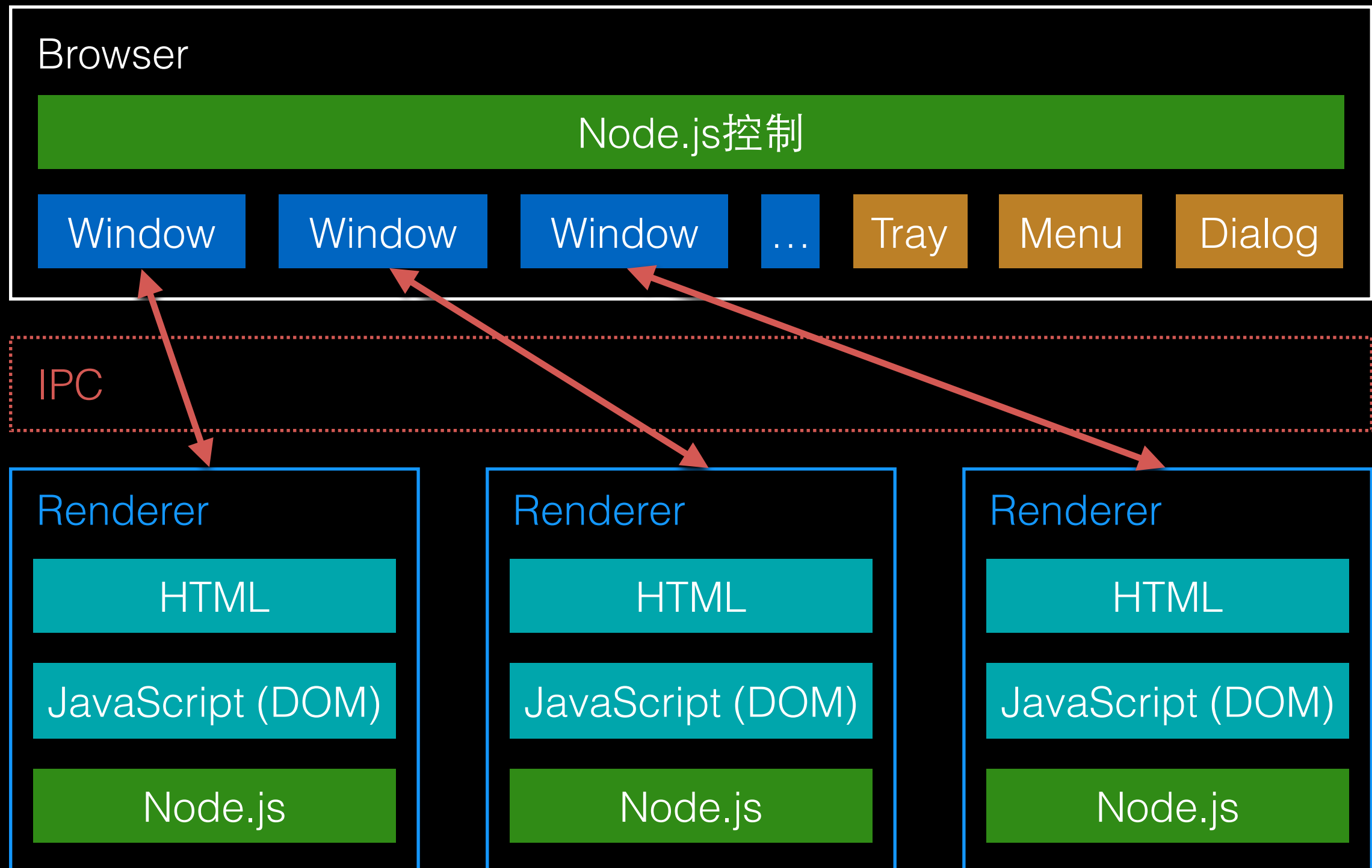
A screenshot of the Atom application source code running in Atom Shell. The left sidebar shows a file explorer with the 'atom' directory expanded, highlighting the 'src' folder. The main editor pane displays the 'atom-application.coffee' file. The code is written in CoffeeScript and includes various module requirements like 'browser-window', 'menu', 'app', 'dialog', 'fs', 'ipc', 'path', 'os', 'net', 'shell', 'url', 'events', and 'underscore-plus'. It defines a 'socketPath' based on the platform and a 'module.exports' block containing a 'class AtomApplication' that extends 'Emitter' and has an '@open' method. The status bar at the bottom indicates the file path 'src/browser/atom-application.coffee' and the line number '10,20'.

Atom Shell是可用
Node.js编程的
最小化Chromium浏览器

Chromium浏览器



Atom Shell



Atom Shell里的Node.js

- Atom Shell同时在Chromium浏览器的Browser端和Renderer端嵌入了Node.js引擎
- Browser端可以用JavaScript操作GUI，比如创建窗口、建立窗口菜单等，同时也能使用Node.js的API
- Renderer端则是在DOM的基础上增加了Node.js的API

Node.js引擎是如何
嵌入到Atom Shell中的?

三步



1. 初始化V8引擎

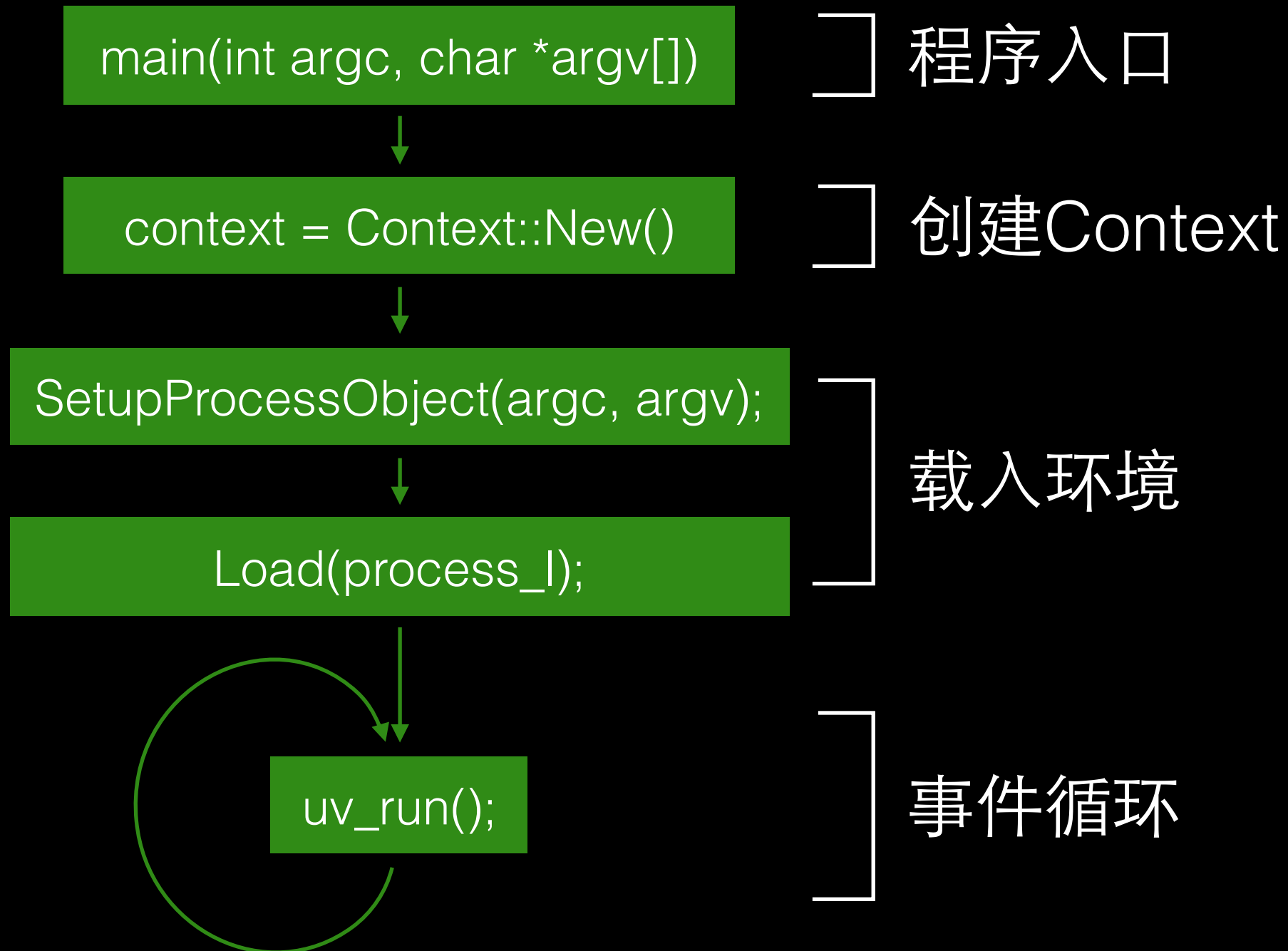
2. 载入Node.js环境

3. 进入事件循环

今天的内容

2. 载入Node.js环境

Node.js \leq 0.10.x



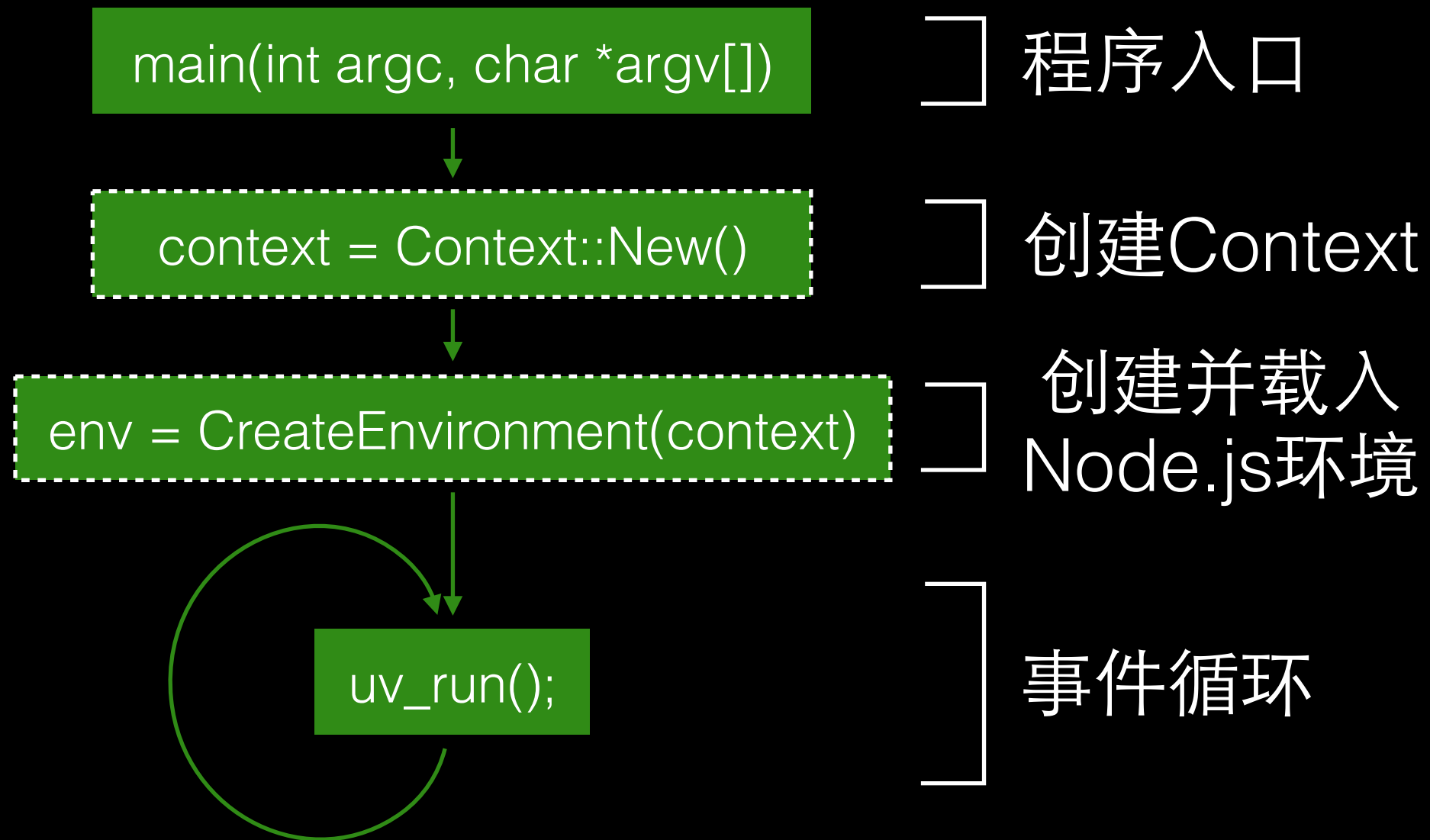
只允许有一个
全局context



multi-context patch

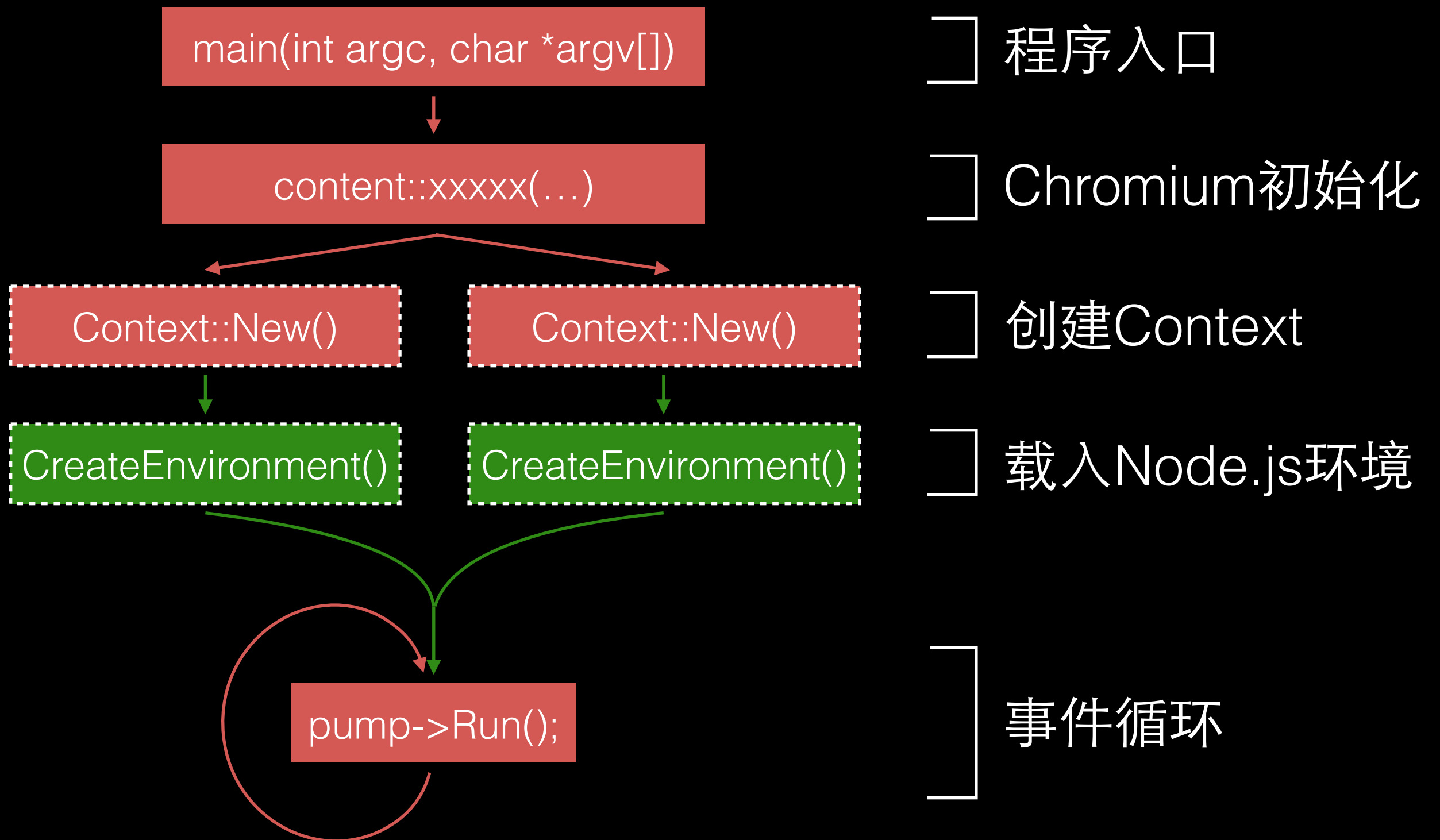
<http://strongloop.com/strongblog/whats-new-node-js-v0-12-multiple-context-execution/>

Node.js \geq 0.11.x



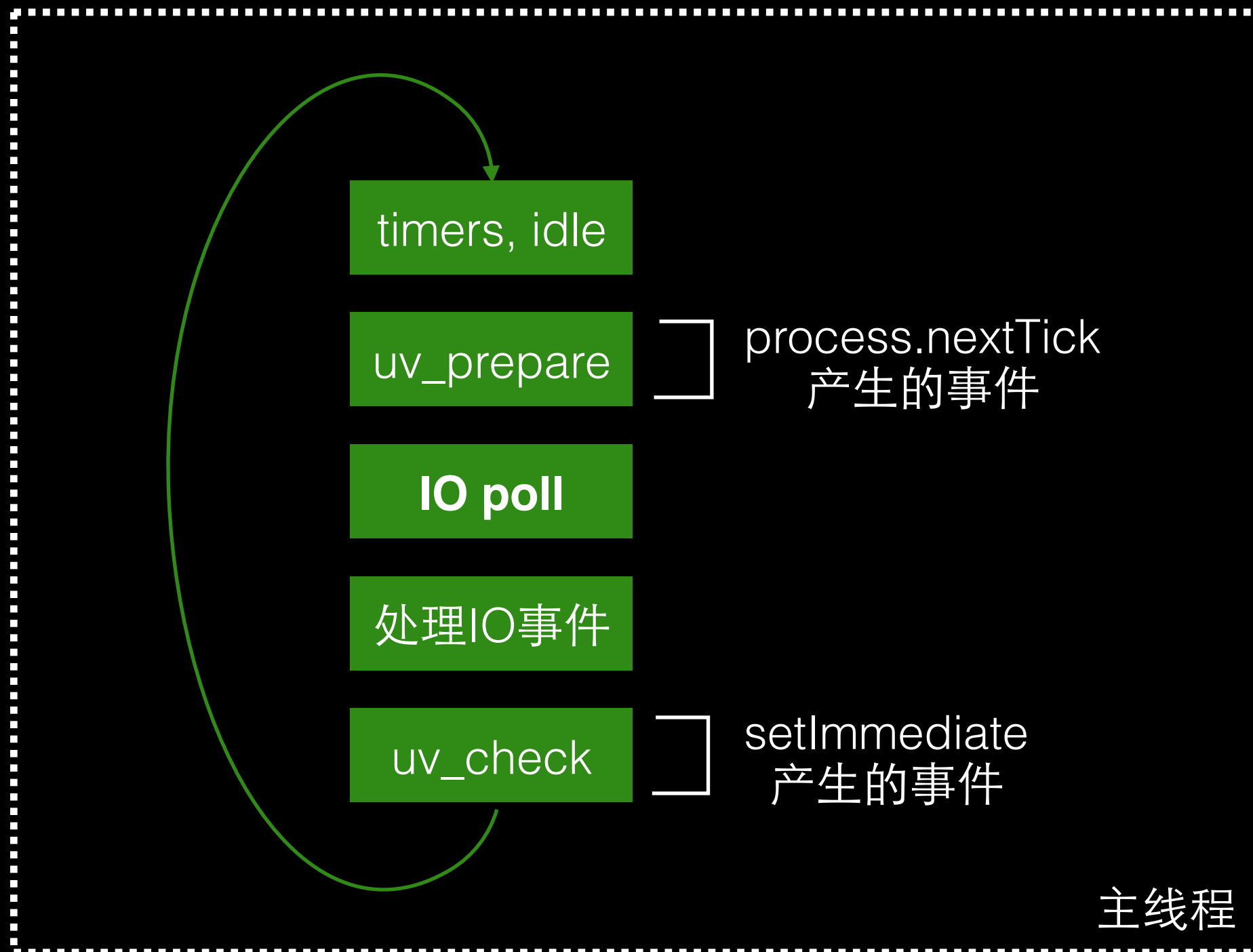
一个线程内允许
多个context

Atom Shell中载入Node.js

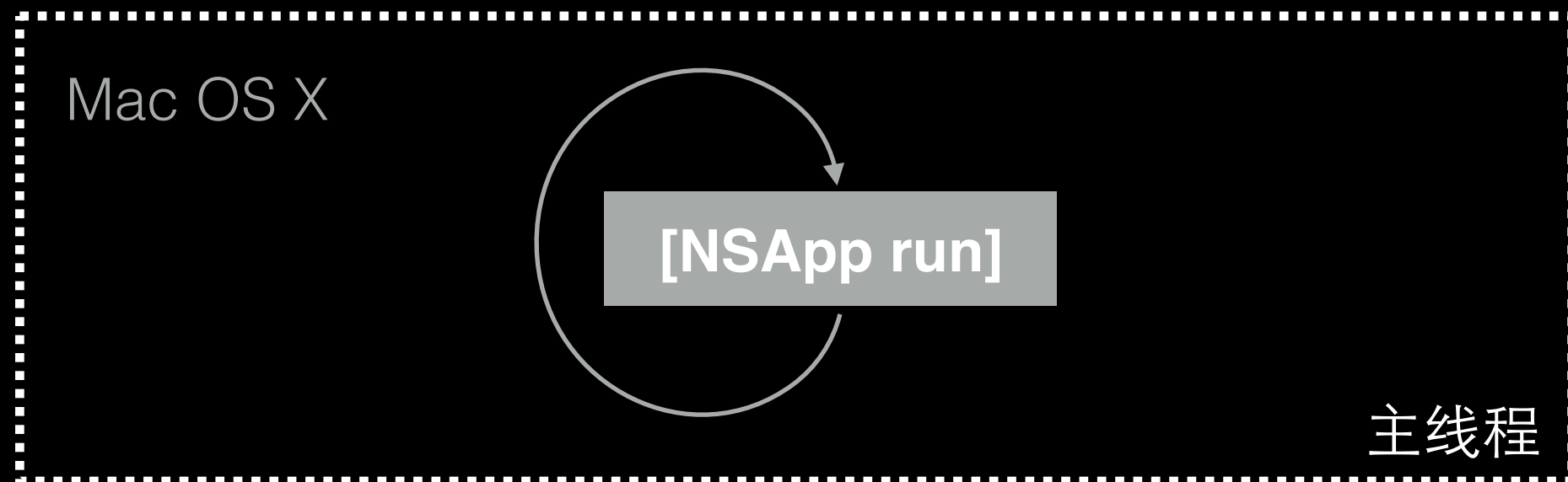
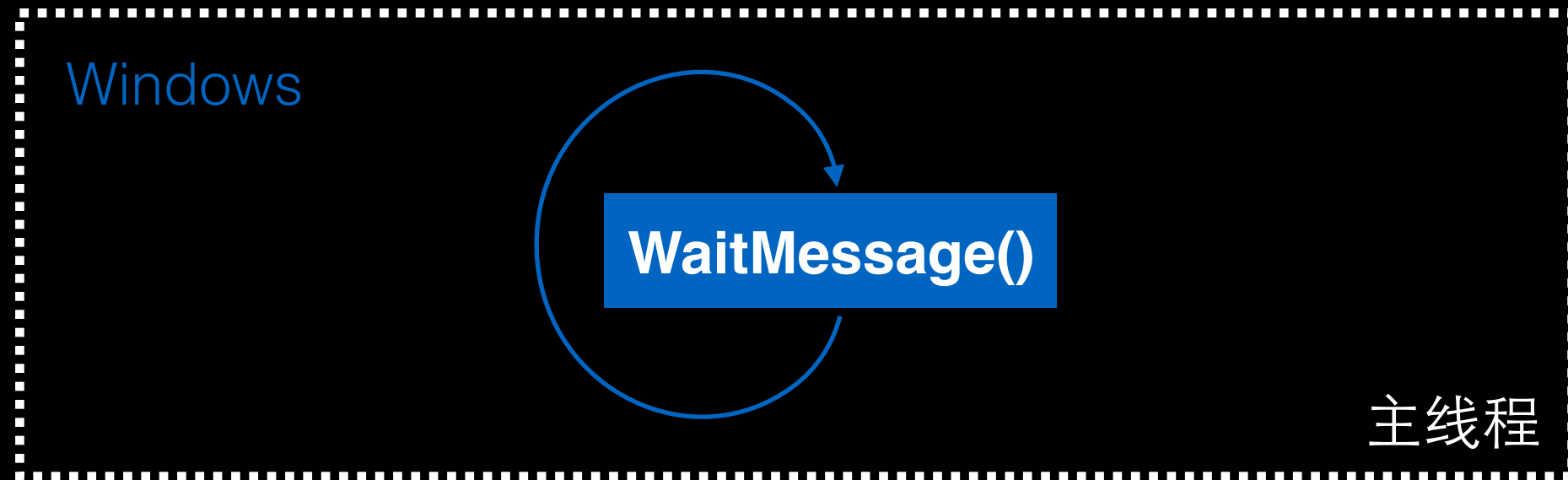


3. 进入事件循环

Node.js的事件循环



原生GUI程序的事件循环

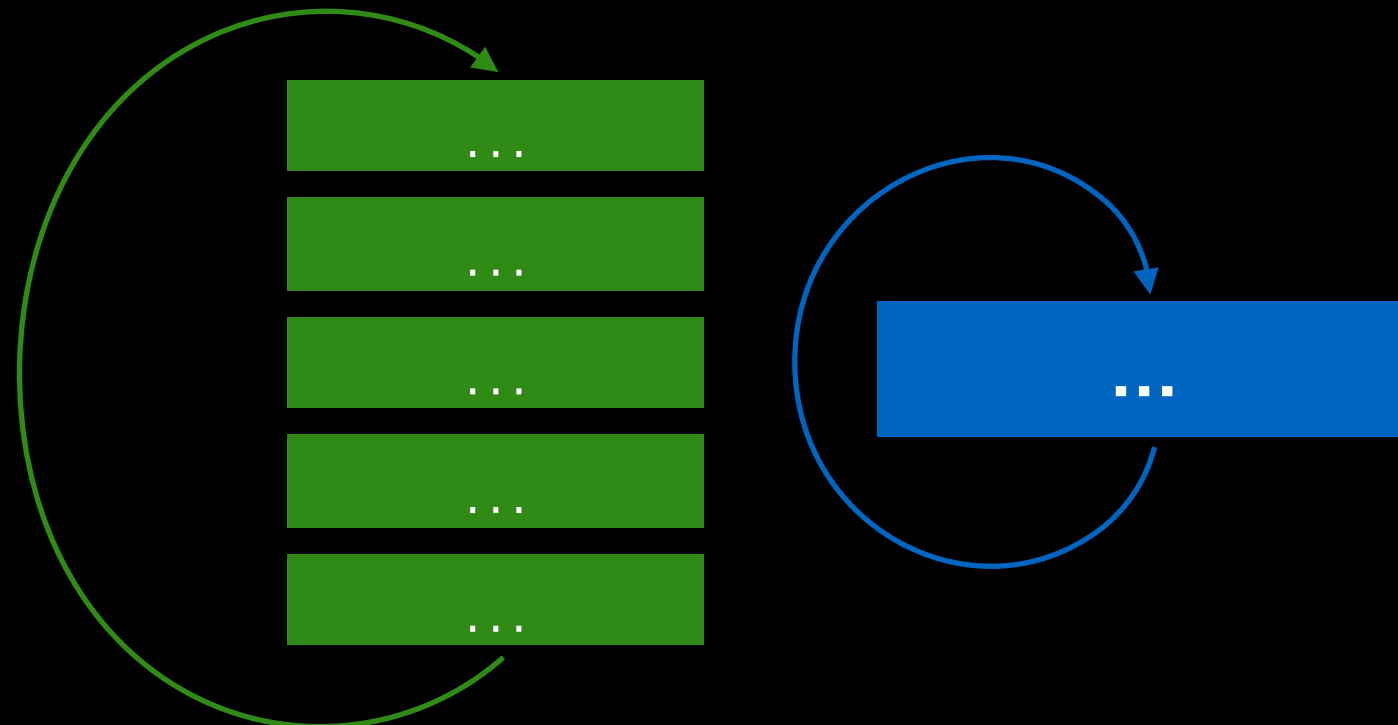


一个主线程，
两个事件循环

```
main(int argc, char *argv[])
```



?????

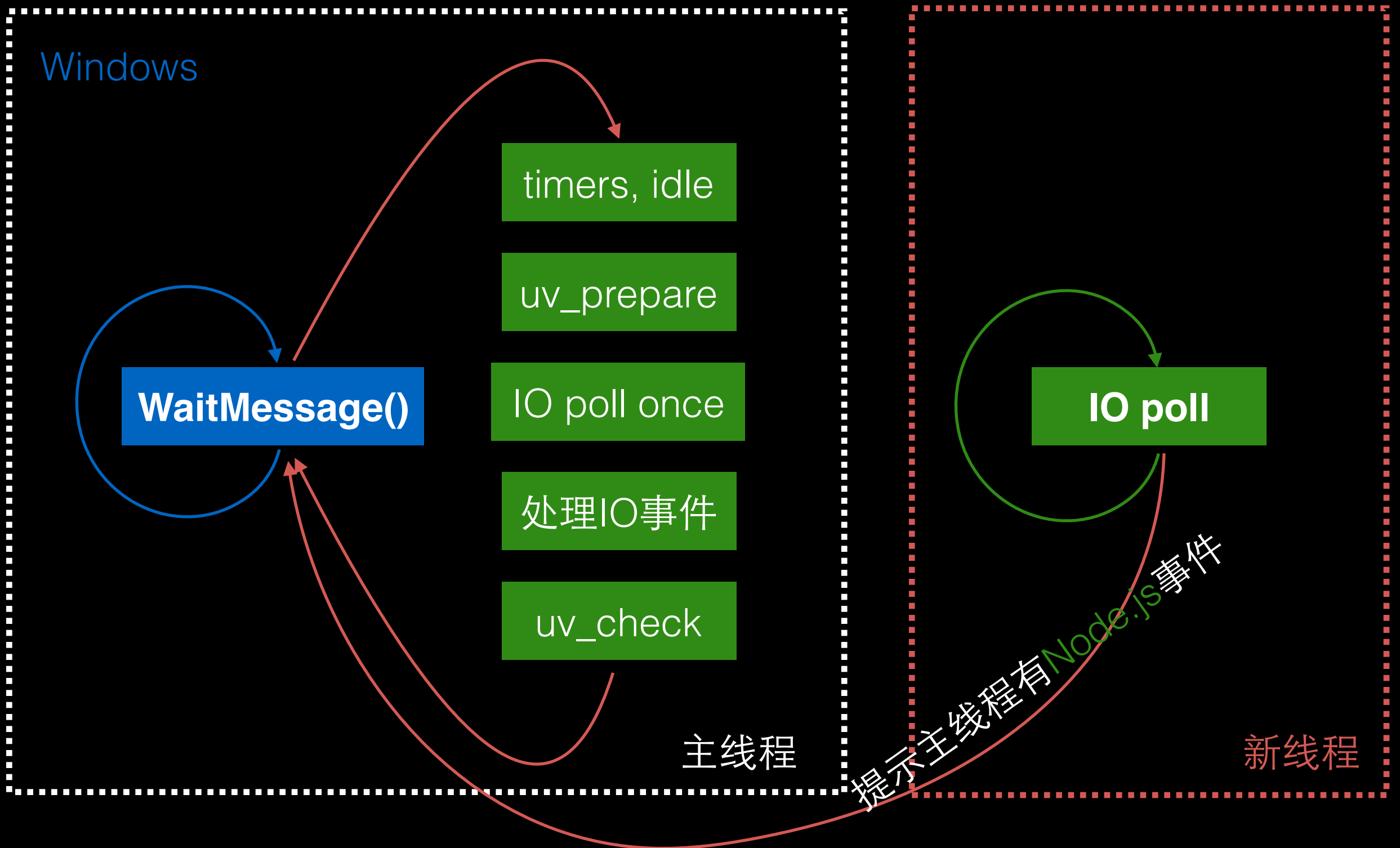


主线程

事件循环的融合

[https://github.com/atom/atom-shell/blob/master/atom/common/
node_bindings.cc](https://github.com/atom/atom-shell/blob/master/atom/common/node_bindings.cc)

Atom Shell的事件循环



问题？