The Event Loop Tightrope

Shelley Vohr

@codebytere

Shelley Vohr

@electronjs

@github

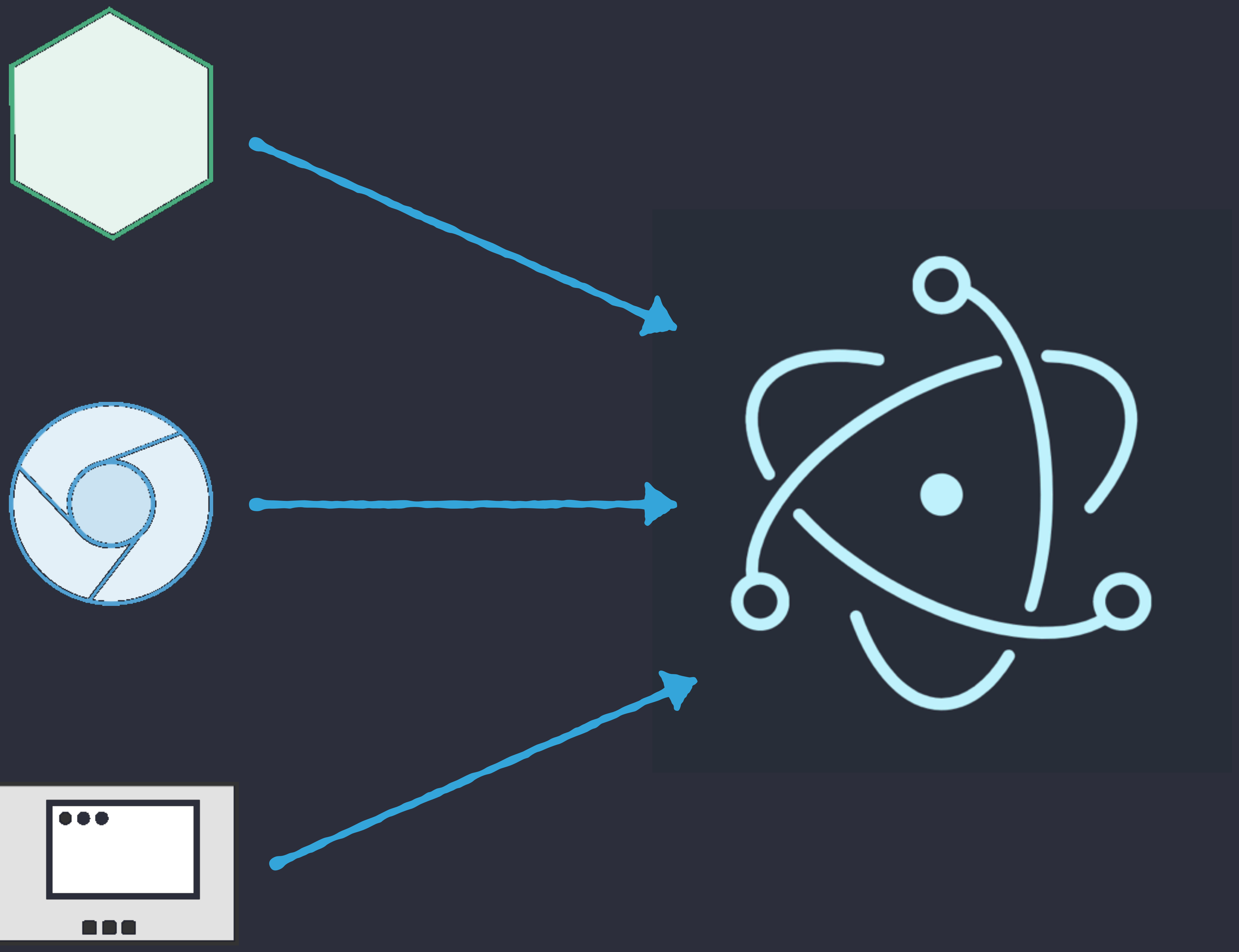
Based in SF 🌉

Electron

Node.js for filesystems and networks

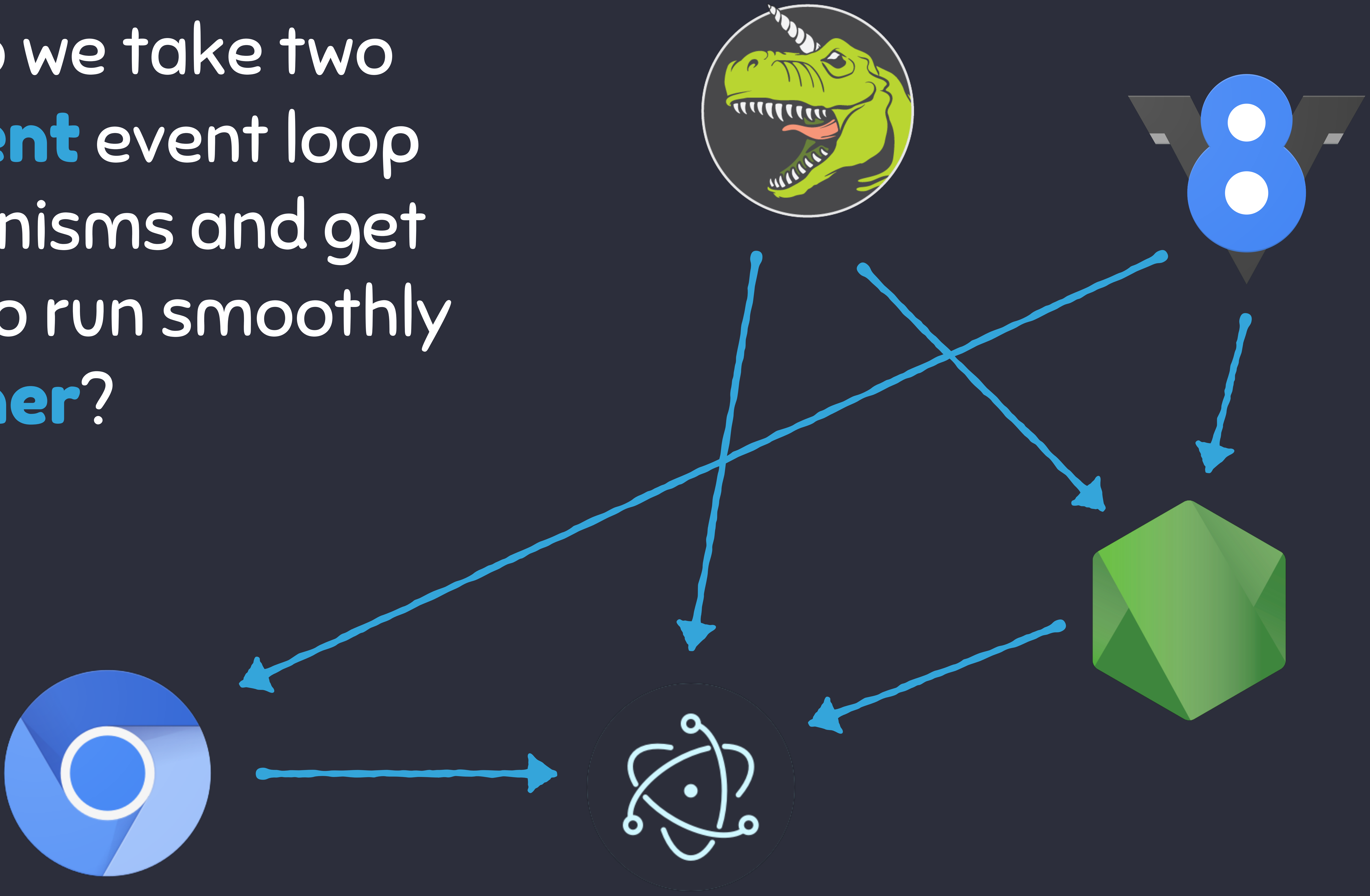
Chromium for making web pages

Native APIs for three systems



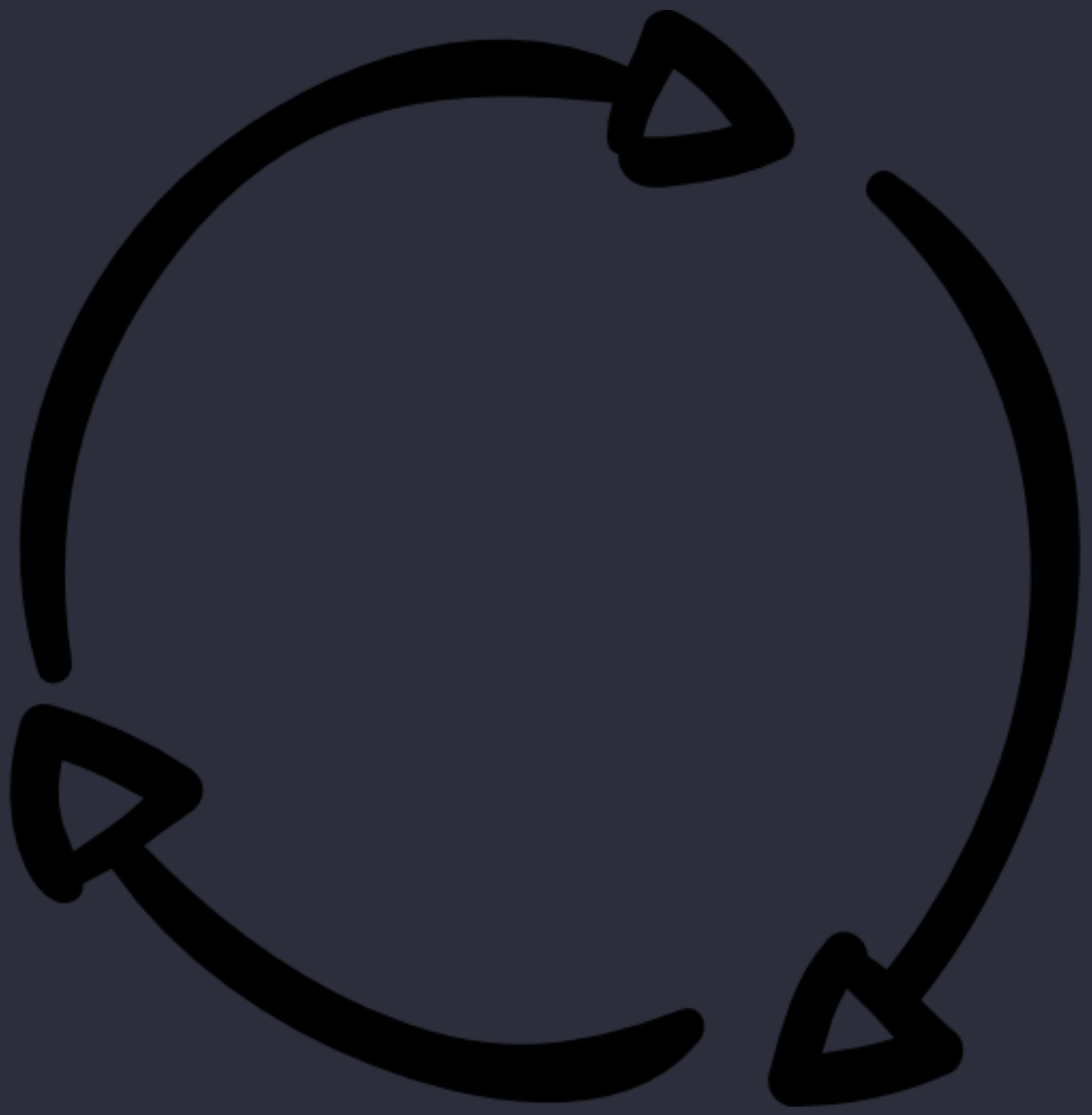
Today's Journey

How do we take two **different** event loop mechanisms and get them to run smoothly **together**?



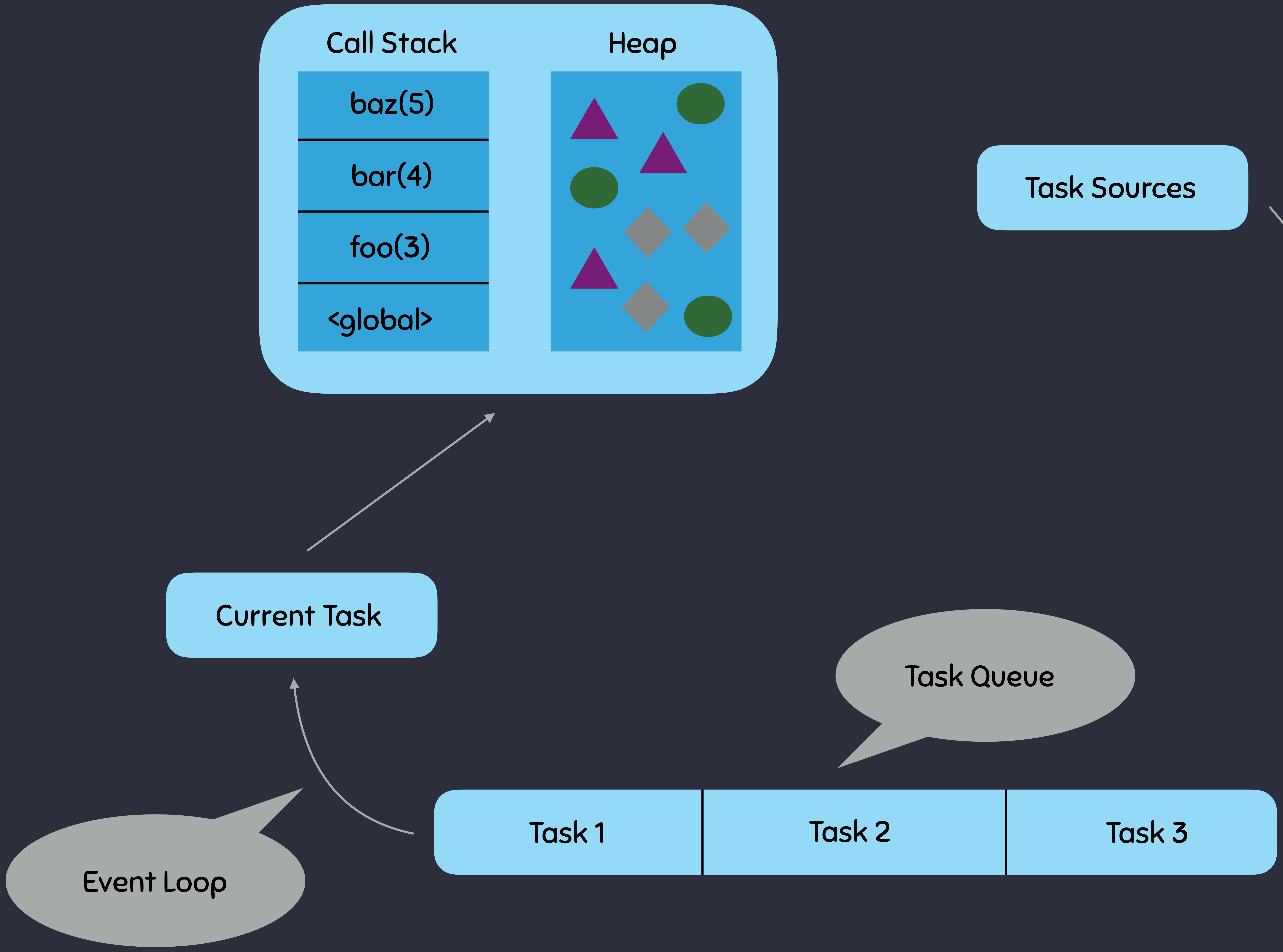
@codebytere

The Event Loop

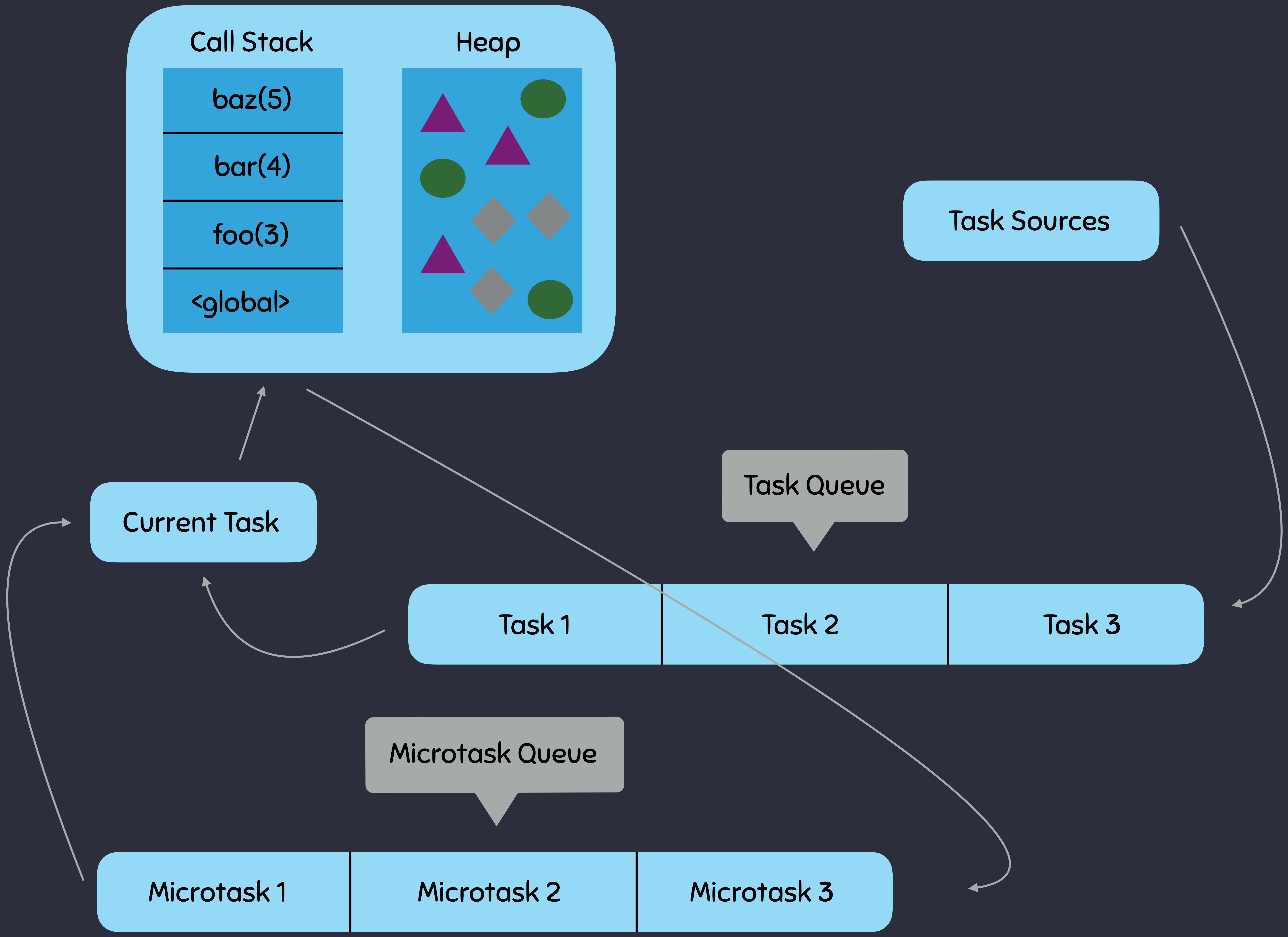


An **endlessly running, singly-threaded** loop, where each iteration runs a small chunk of the code in the program currently being executed.

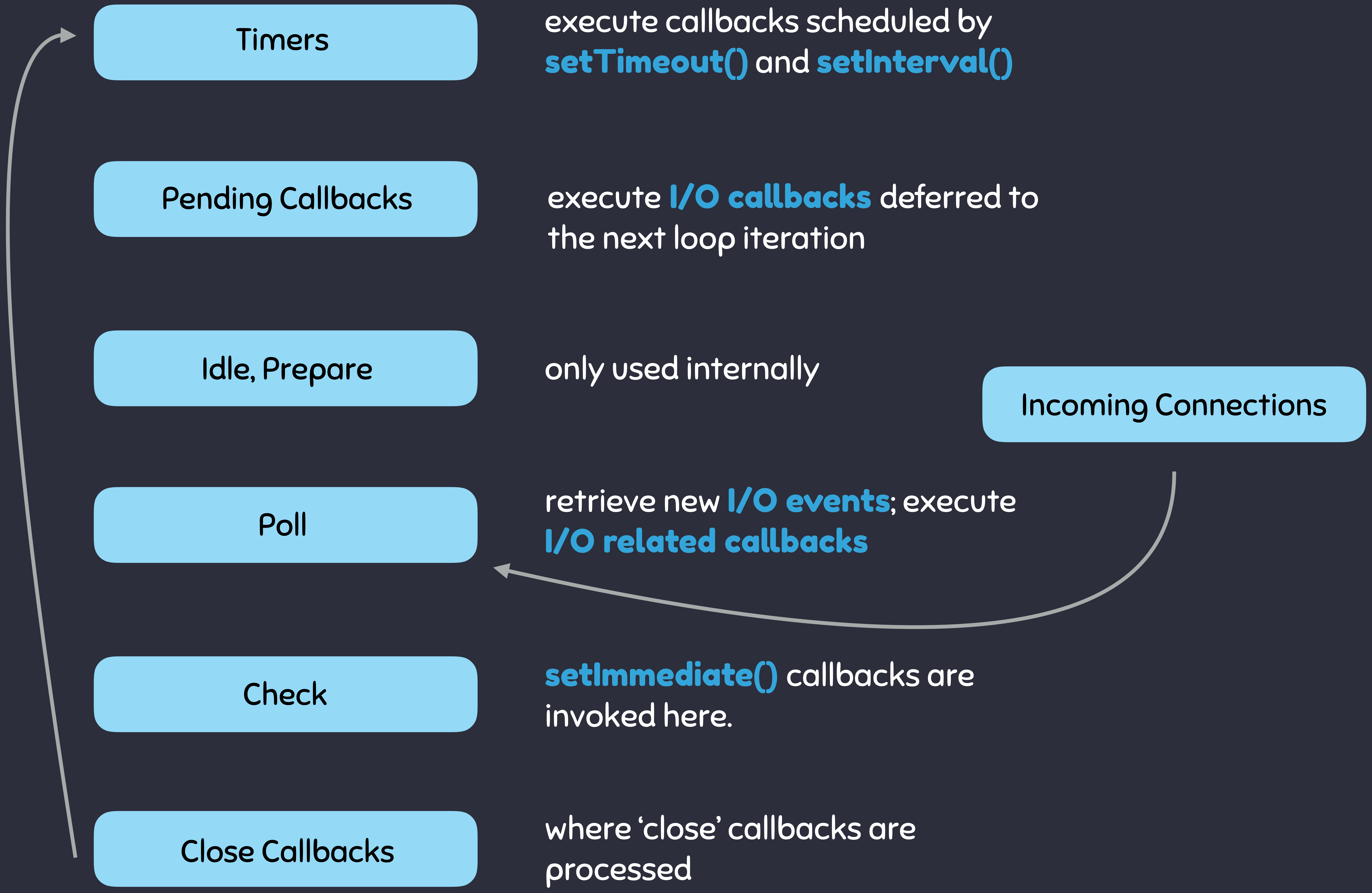
@codebytere


Call Stack
Heap
baz(5)
bar(4)
foo(3)
<global>
Task Sources
Current Task
Task Queue
Event Loop
Task 1
Task 2
Task 3

@codebytere


Call Stack
Heap
baz(5)
bar(4)
foo(3)
<global>
Task Sources
Task Queue
Current Task
Task 1
Task 2
Task 3
Microtask Queue
Microtask 1
Microtask 2
Microtask 3

@codebytere



execute callbacks scheduled by **setTimeout()** and **setInterval()**

execute **I/O callbacks** deferred to the next loop iteration

only used internally

retrieve new **I/O events**; execute **I/O related callbacks**

setImmediate() callbacks are invoked here.

where 'close' callbacks are processed

@codebytere

Pending Callbacks

execute **I/O callbacks** deferred to the next loop iteration

Incoming Connections

Poll

retrieve new **I/O events** & execute **I/O related callbacks**

@codebytere

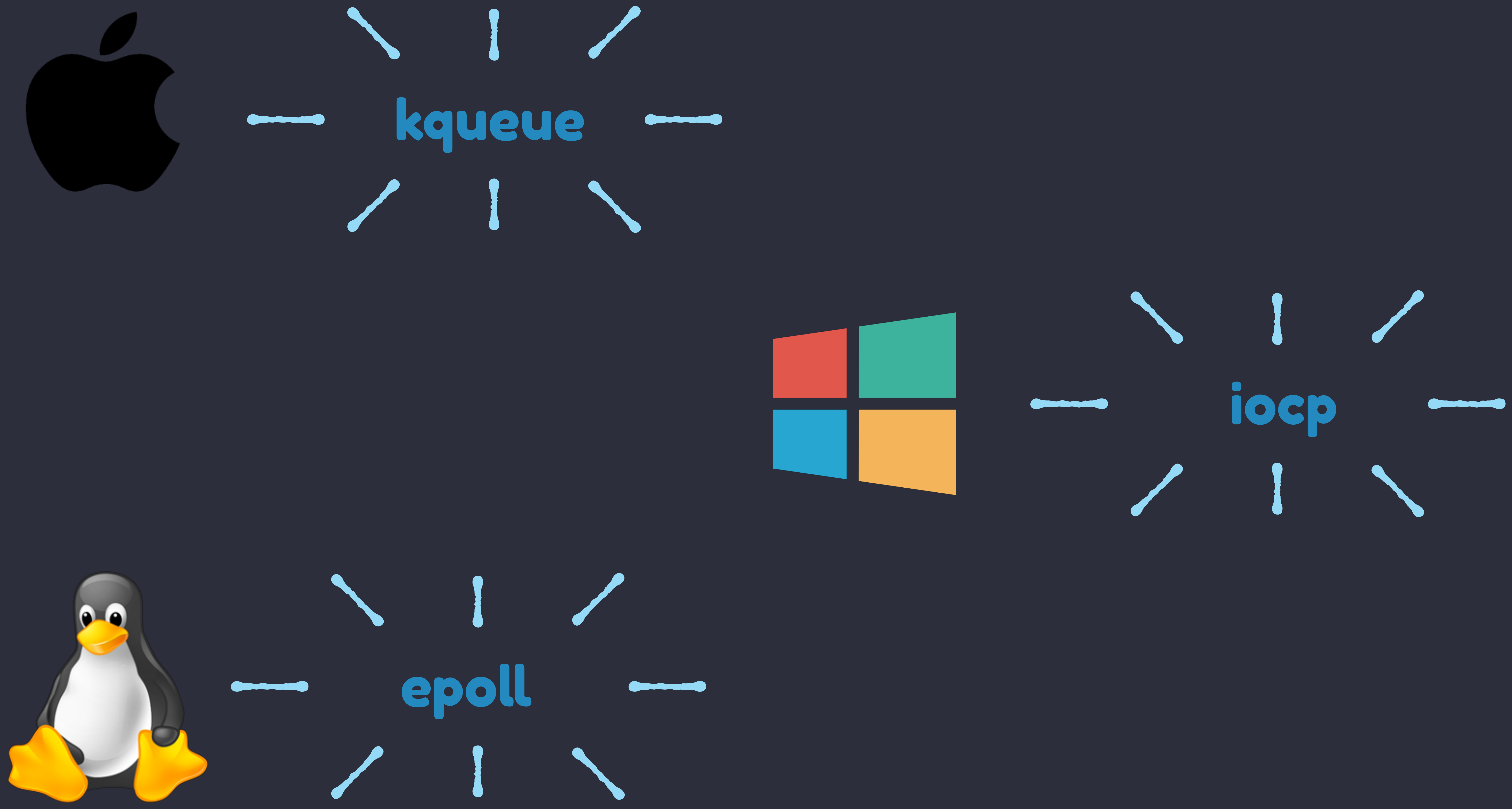
libuv

Multi-platform C library that provides support for **asynchronous I/O** based on event loops

This powers Node.js' event loop!

@codebytere

libuv - Event Notification



@codebytere

File Descriptor

Abstract indicator used to access a **file** or other **I/O** resource.

Non-Negative Integer

0	Standard Input	stdin
1	Standard Output	stdout
2	Standard Error	stderr

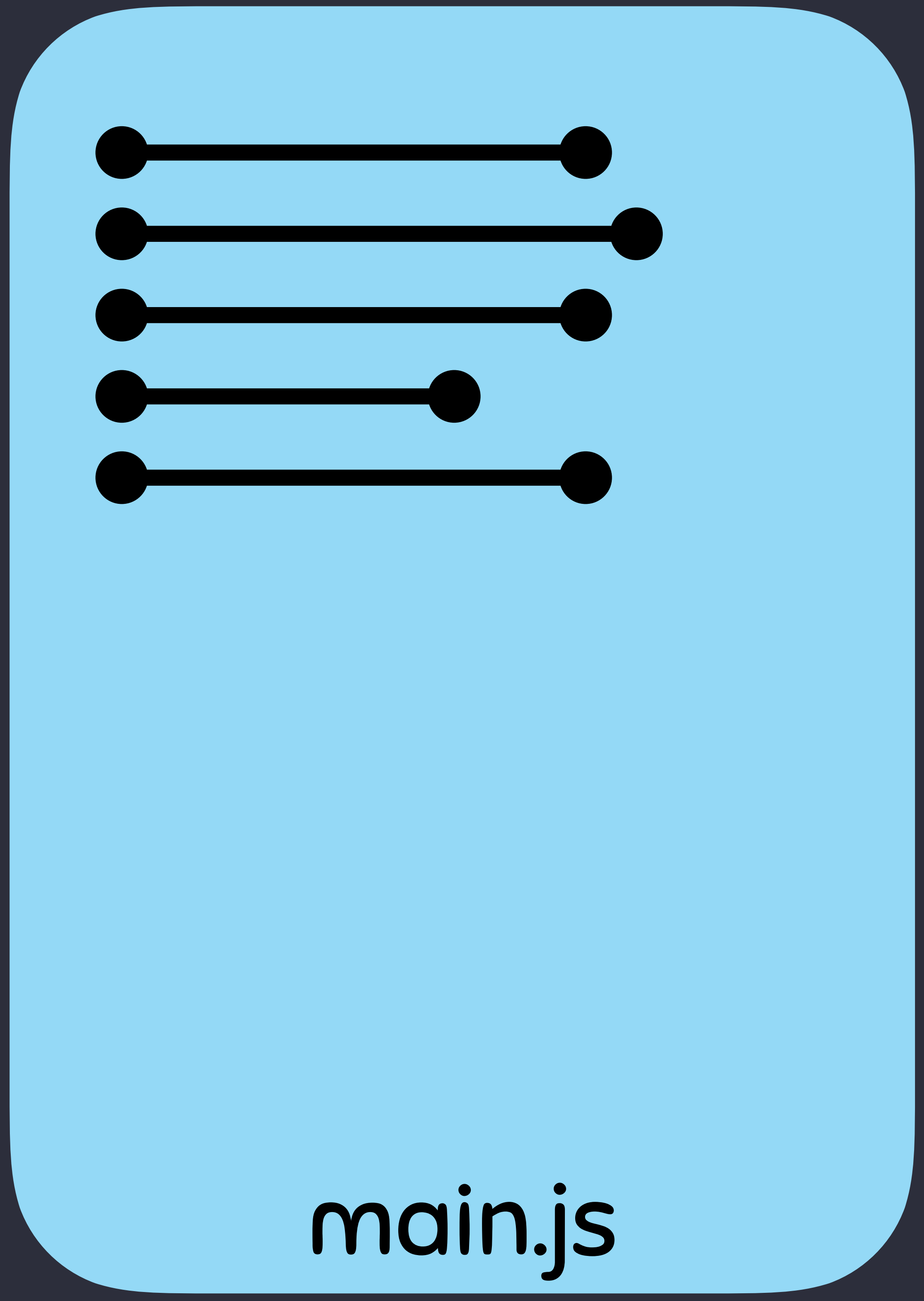
@codebytere

Electron

Let's take a few steps back

- What does Electron need to do with the file descriptor in libuv?
- Why is this **more challenging** than it was in Node.js itself?
  - What's **different**?

Main Process



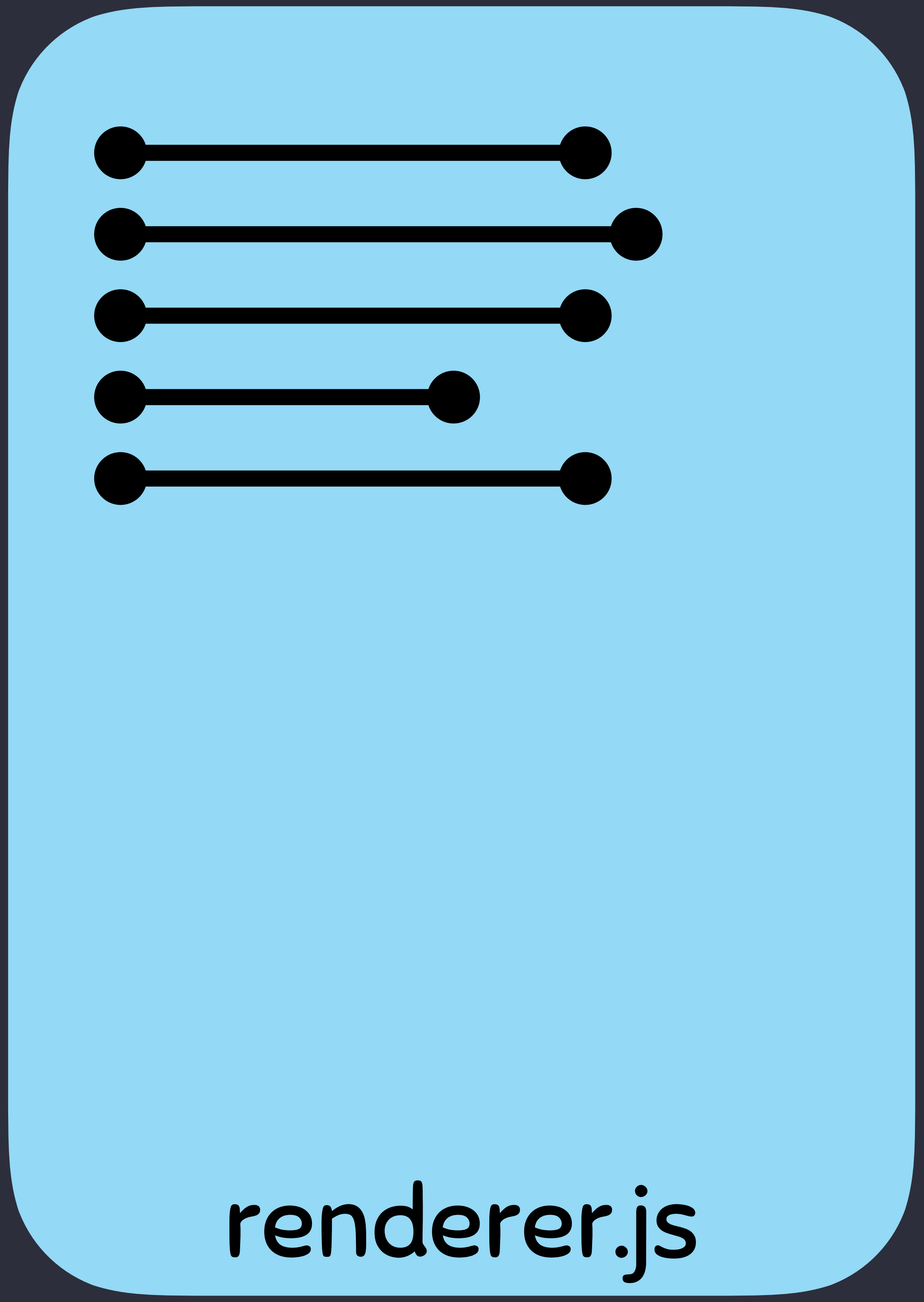
- Node.js APIs (always)
- Electron Main Process Modules
- ONLY **1 (ONE!)**

Main Process

```
const {app, BrowserWindow} = require('electron')

let win

function createWindow () {
  win = new BrowserWindow({
    width: 800,
    height: 600
  })
}

app.on('ready', createWindow)

app.on('window-all-closed', () => {
  if (process.platform !== 'darwin') app.quit()
})

app.on('activate', () => {
  if (win === null) createWindow()
})
```

@codebytere

Renderer Process



- Node.js APIs (optional)
- Electron Renderer Process Modules
- Many
- Independent

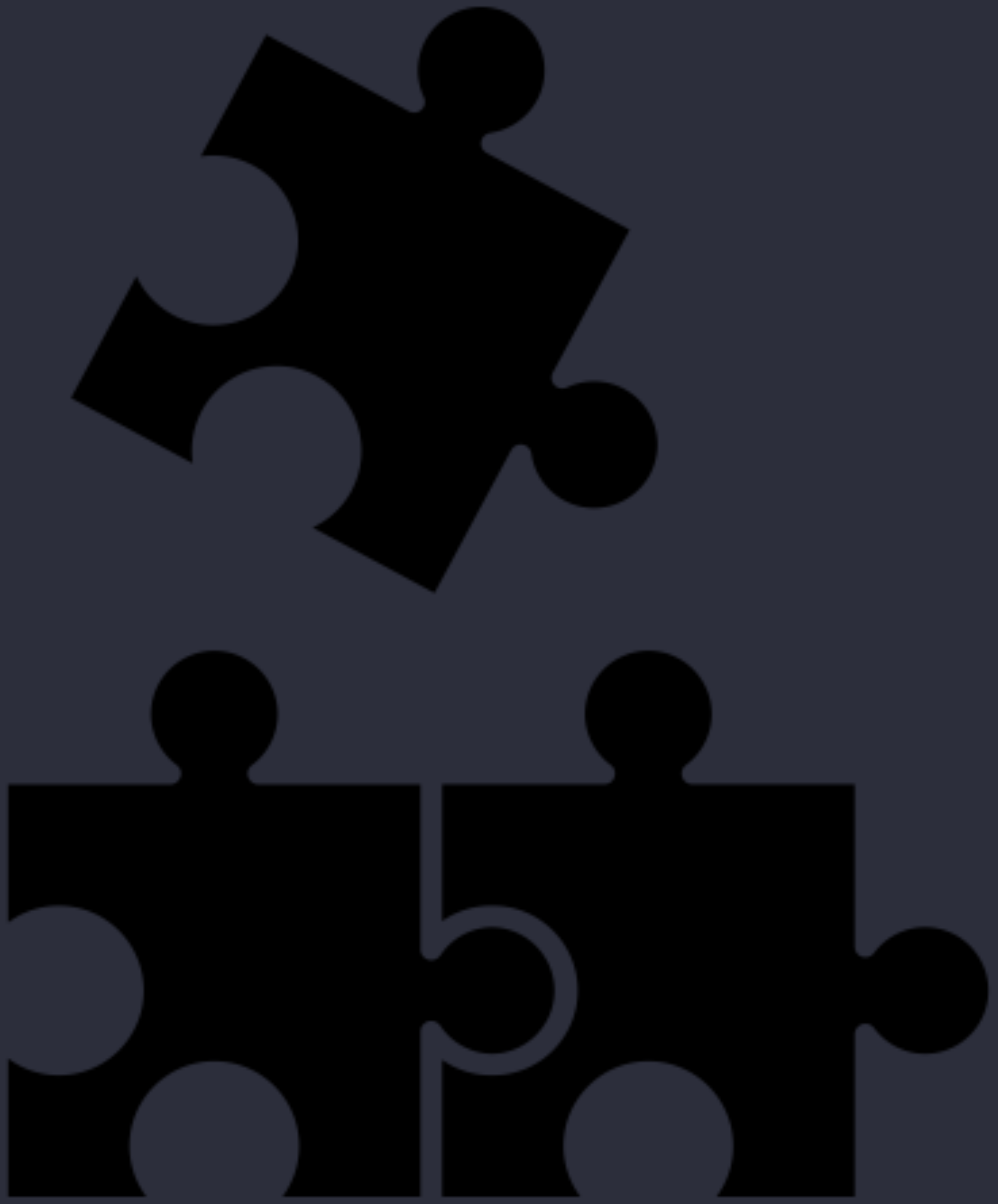
@codebytere

Renderer Process

```
const { remote } = require('electron')
const { BrowserWindow } = remote

const win = new BrowserWindow()
```

@codebytere

Integrating Event Loops

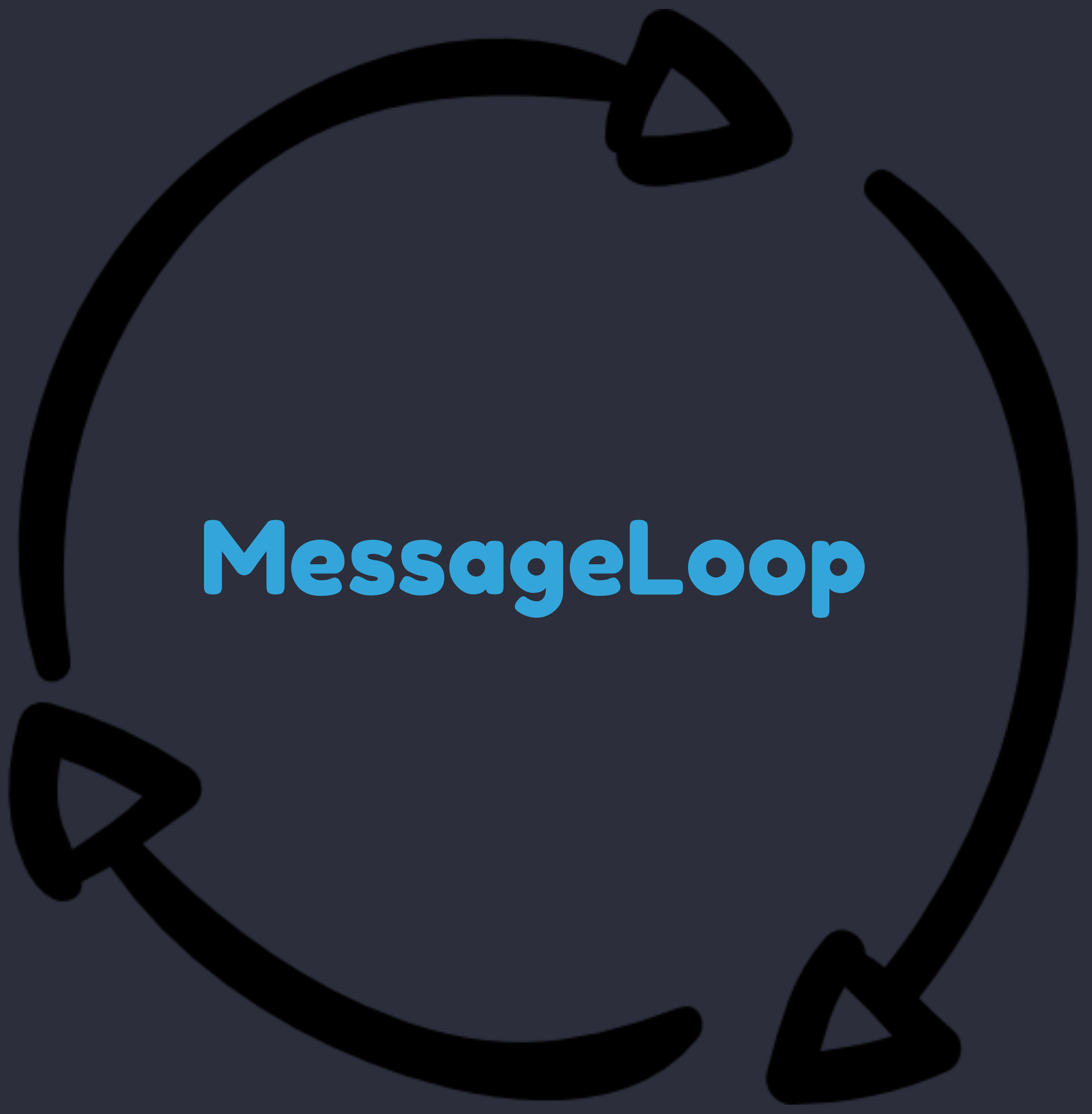


Why are we talking about **processes** anyway?

Event loop integration was **slightly different** for each!

Chromium's Message Loop

Chromium implements its own event loop (they term it **Message Loop**) in both the **renderer** and **browser** processes!



The First Attempt

Re-Implement
MessageLoop

@codebytere

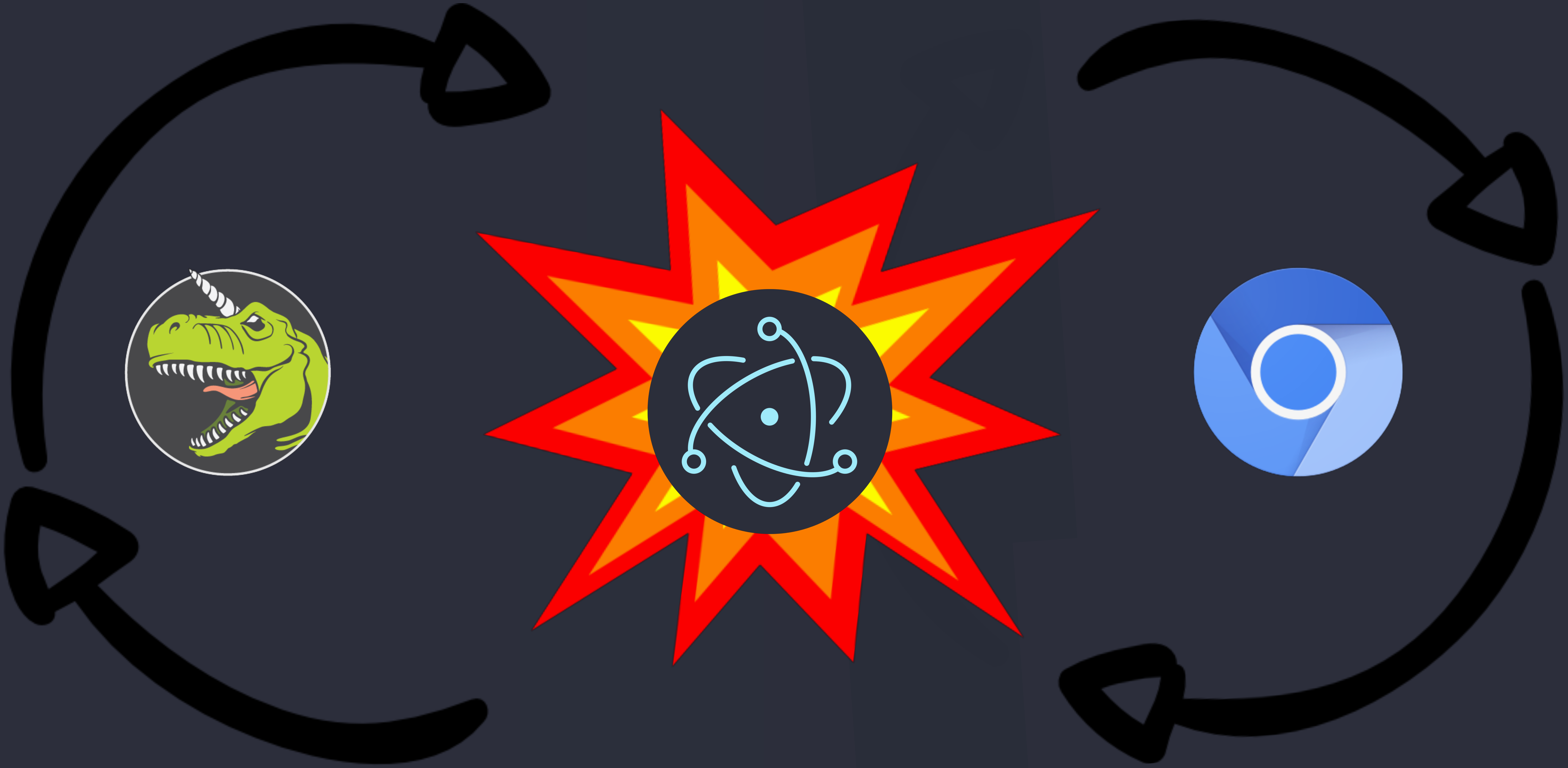
Chromium’s Message Loop

Main Thread (UI Thread) event loop messages are often **platform-specific**

Windows	MessageLoopForUI	MessagePumpWin
macOS	MessageLoopForUI	MessagePumpMac
Linux	MessageLoopForUI	MessagePumpGtk

@codebytere

Second Attempt

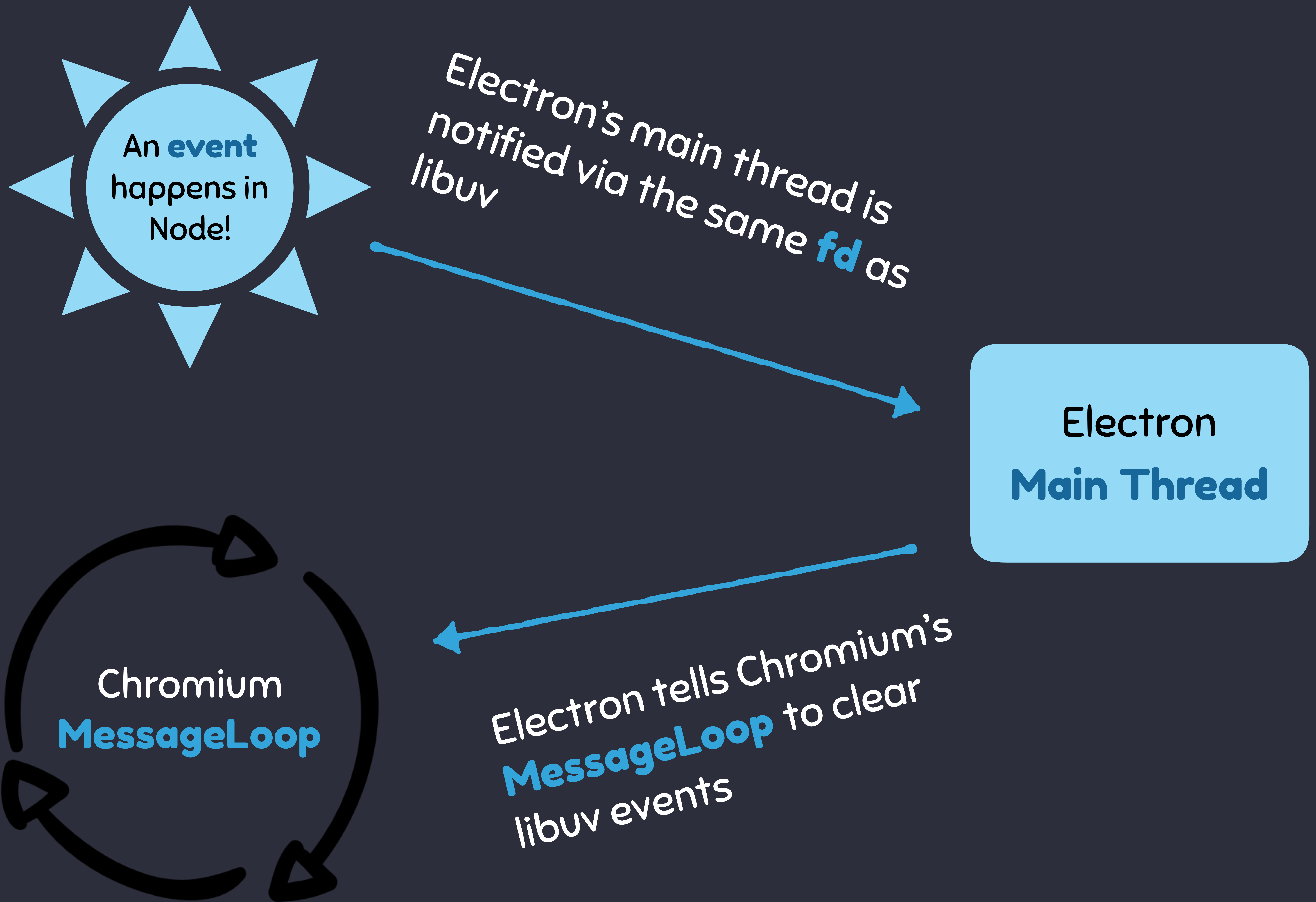


@codebytere

uv_backend_fd()

- Get Node.js' event loop **fd**
- Use this to **learn about events** in the libuv event loop we want to embed into another event loop!

The Final Solution


An **event** happens in Node!
Electron's main thread is notified via the same **fd** as libuv
Electron
Main Thread
Electron tells Chromium's **MessageLoop** to clear libuv events
Chromium
MessageLoop

@codebytere

Further Reading

Multi-Process Architecture: **http://bit.do/m-p-a**

Electron Application Architecture: **http://bit.do/e-p-a**

Node.js' Event Loop: **http://bit.do/n-e-l**

Come Talk to Me!

Follow-up **questions**?

Want to **contribute**?

Want to **chat about OSS**?

@codebytere

THANK YOU!

@codebytere