BUILDING SECURE ELECTRON APPLICATIONS

Naz Gayoom

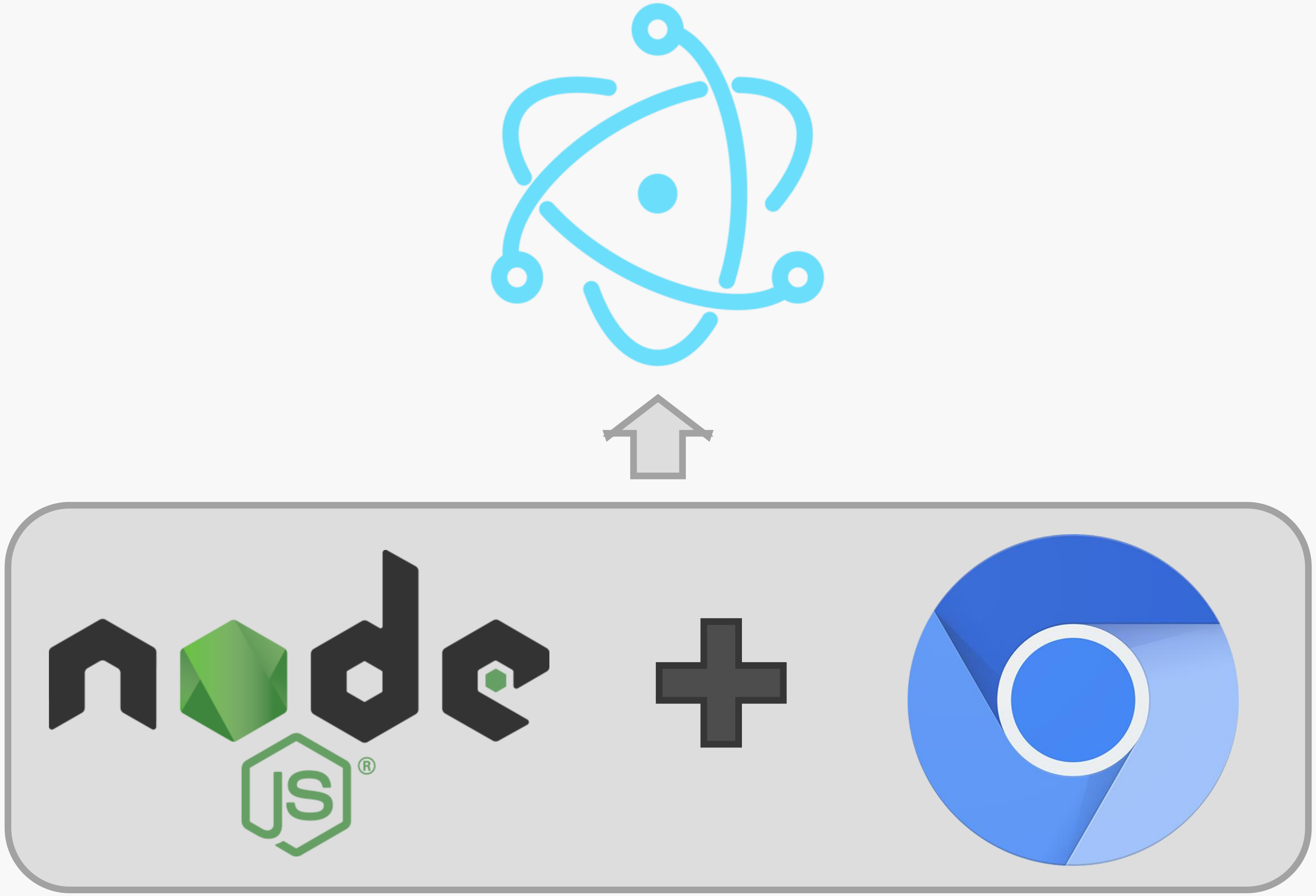
@ngayoom

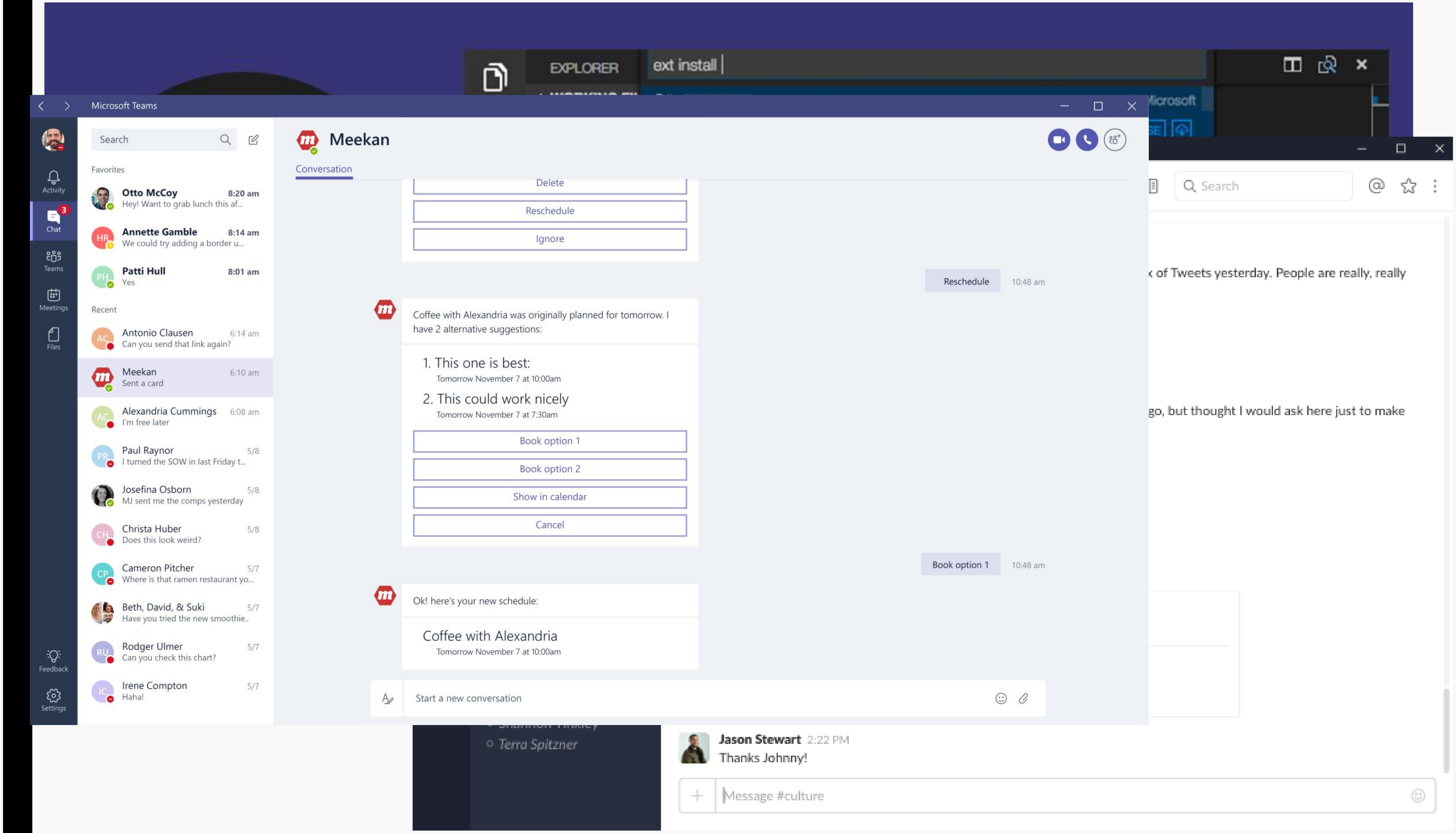
Agenda

- What is Electron
- Node Access
- Permissions
- The Webview Tag
- Demo
- A few other things…
- Q & A

What is Electron?

- A framework for building cross-platform desktop applications





Favorites

Otto McCoy 8:20 am
Hey! Want to grab lunch this af...

Annette Gamble 8:14 am
We could try adding a border u...

Patti Hull 8:01 am
Yes

Recent

Antonio Clausen 6:14 am
Can you send that link again?

Meekan 6:10 am
Sent a card

Alexandria Cummings 6:08 am
I'm free later

Paul Raynor 5/8
I turned the SOW in last Friday t...

Josefina Osborn 5/8
MJ sent me the comps yesterday

Christa Huber 5/8
Does this look weird?

Cameron Pitcher 5/7
Where is that ramen restaurant yo...

Beth, David, & Suki 5/7
Have you tried the new smoothie...

Rodger Ulmer 5/7
Can you check this chart?

Irene Compton 5/7
Haha!

Delete

Reschedule

Ignore

Reschedule 10:48 am

Coffee with Alexandria was originally planned for tomorrow. I have 2 alternative suggestions:

1. This one is best:
Tomorrow November 7 at 10:00am

2. This could work nicely
Tomorrow November 7 at 7:30am

Book option 1

Book option 2

Show in calendar

Cancel

Book option 1 10:48 am

Ok! here's your new schedule:

Coffee with Alexandria
Tomorrow November 7 at 10:00am

k of Tweets yesterday. People are really, really

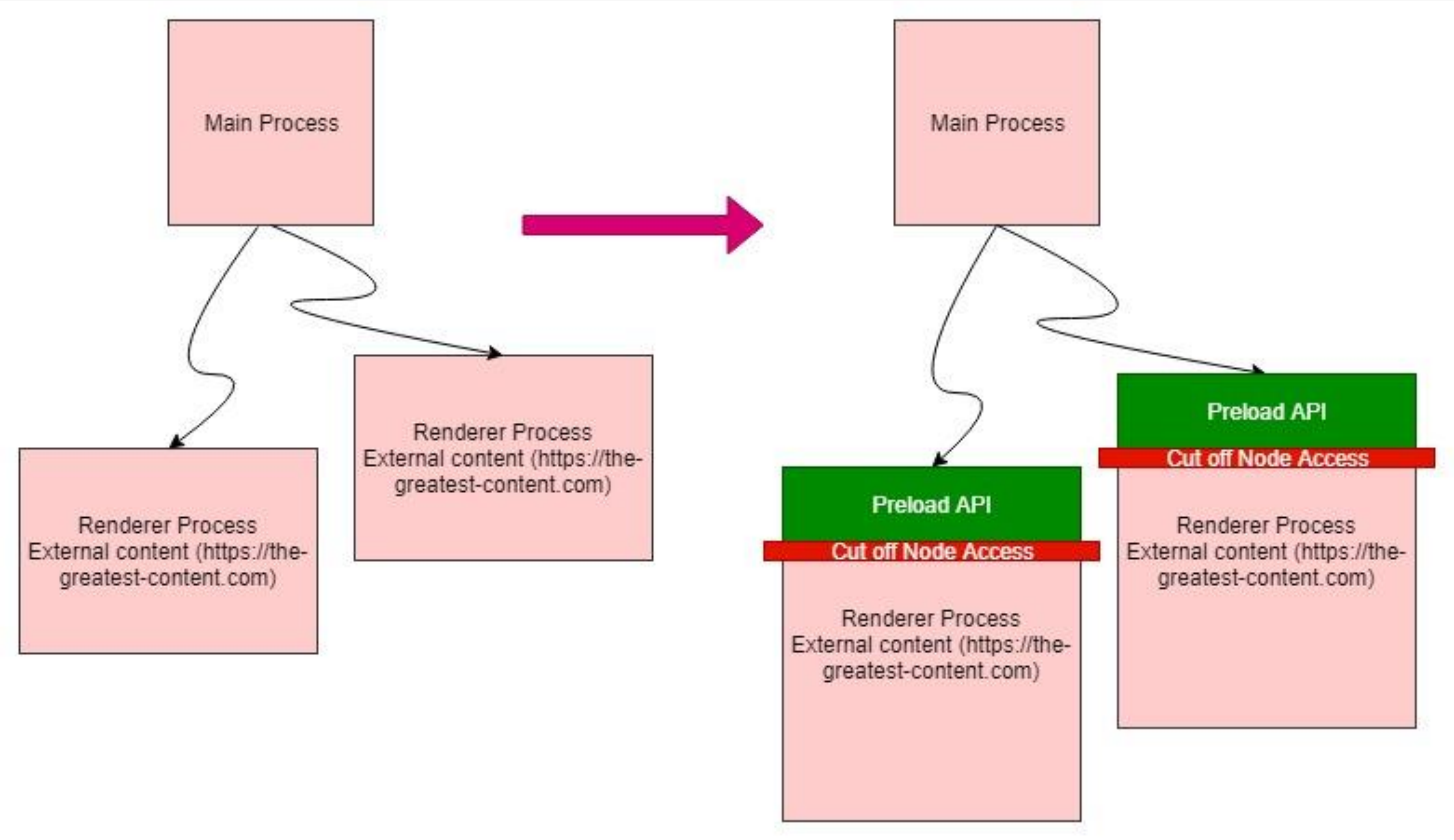
go, but thought I would ask here just to make

Terra Spitzner

Jason Stewart 2:22 PM
Thanks Johnny!

Access to Node.js

- Disable Node.js
- Use a preload script
- Don’t expose node handles on your preload scripts.


Main Process
Renderer Process
External content (https://the-greatest-content.com)
Renderer Process
External content (https://the-greatest-content.com)
Main Process
Preload API
Cut off Node Access
Renderer Process
External content (https://the-greatest-content.com)
Preload API
Cut off Node Access
Renderer Process
External content (https://the-greatest-content.com)

```
webPreferences: {
        nodeIntegration: false,
        preload: path.join(__dirname, 'preload.js’)
}


<webview preload=“./preload.js” src=“https://www.externalsite.com”></webview>
```

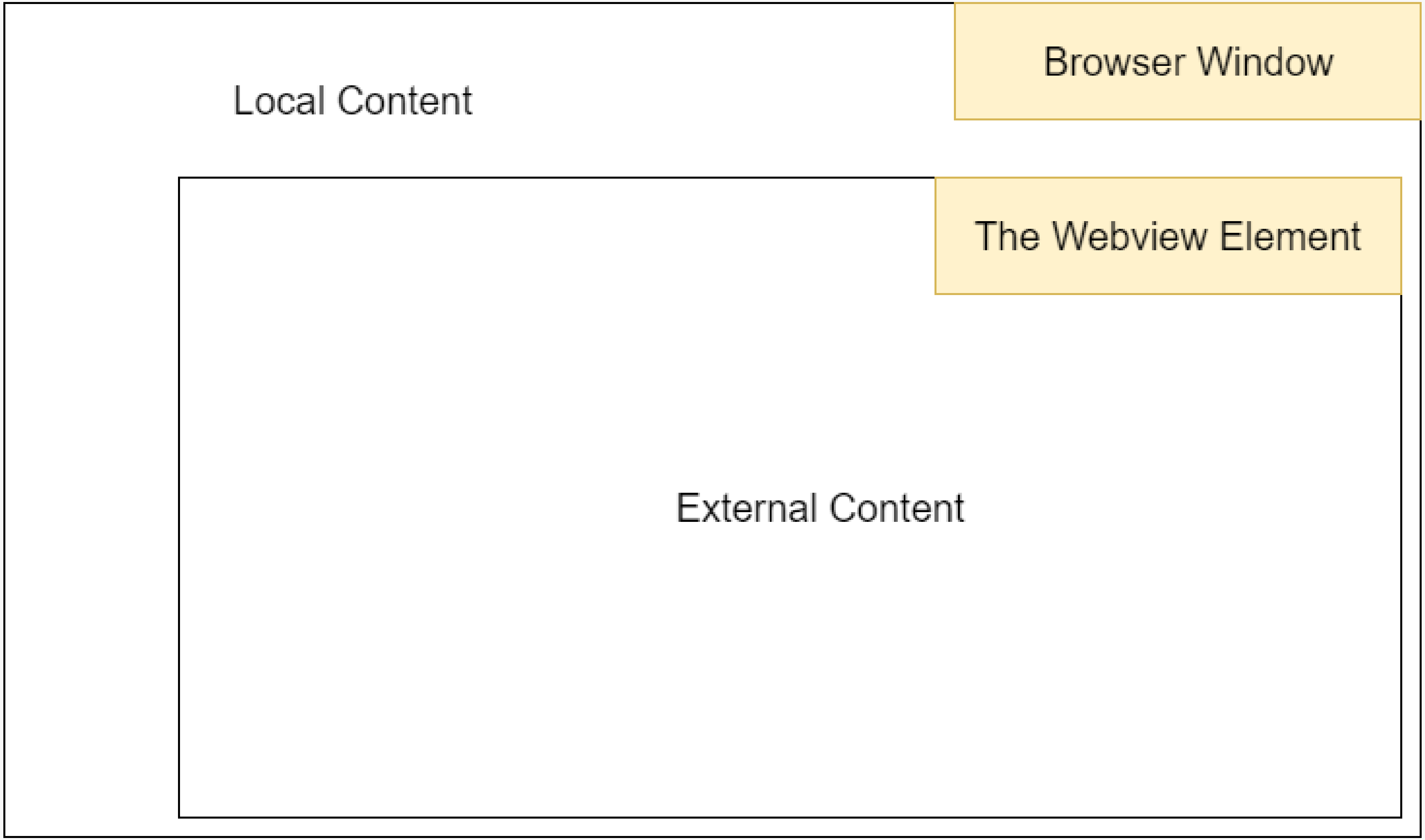
Permissions handling

- HTML5 API: Geolocation, Camera, Microphone, Notifications
- Browser normally asks the user, electron does not
- We want to have custom validations.

```
const currentSession = mainWindow.webContents.session

currentSession.setPermissionRequestHandler((webContents,
permission, callback) => {

      const resultOfSomeCondition = false;

      return callback(resultOfSomeCondition);

});
```

The <Webview> tag

- Has its own process and memory allocation
- Configurable like the browser window
- Always validate dynamically created Webviews



Browser Window

Local Content

The Webview Element

External Content

```
<body>

      <button onclick="getDirectory()">Get the directory from
here...</button>

      <p id="list-of-directories"></p>

      <webview id="pineapple-machine"
src=http://localhost:8080/ preload="./preload.js"
nodeIntegration></webview>

</body>
```

```
app.on('web-contents-created', (event, contents) => {
    contents.on('will-attach-webview', (event,
webPreferences, params) => {
        webPreferences.nodeIntegration = false
    })
  })
```

Demo...

A few other things…

- Using context isolation for JS global objects
- Allowing popups on a webview
- Mixed content (insecure content loaded on a securely loaded page)
- Setup a content security policy

Demo repo and other links

- https://github.com/nawazg/OWASP_NZDay_2019_Talk
- https://slack.engineering/
- https://electronjs.org/docs/tutorial/security

PROVOKE

WE ARE HIRING DEVELOPERS!

THANKS FOR LISTENING