Microsoft

Building Desktop Applications with Electron

Jfokus 2018

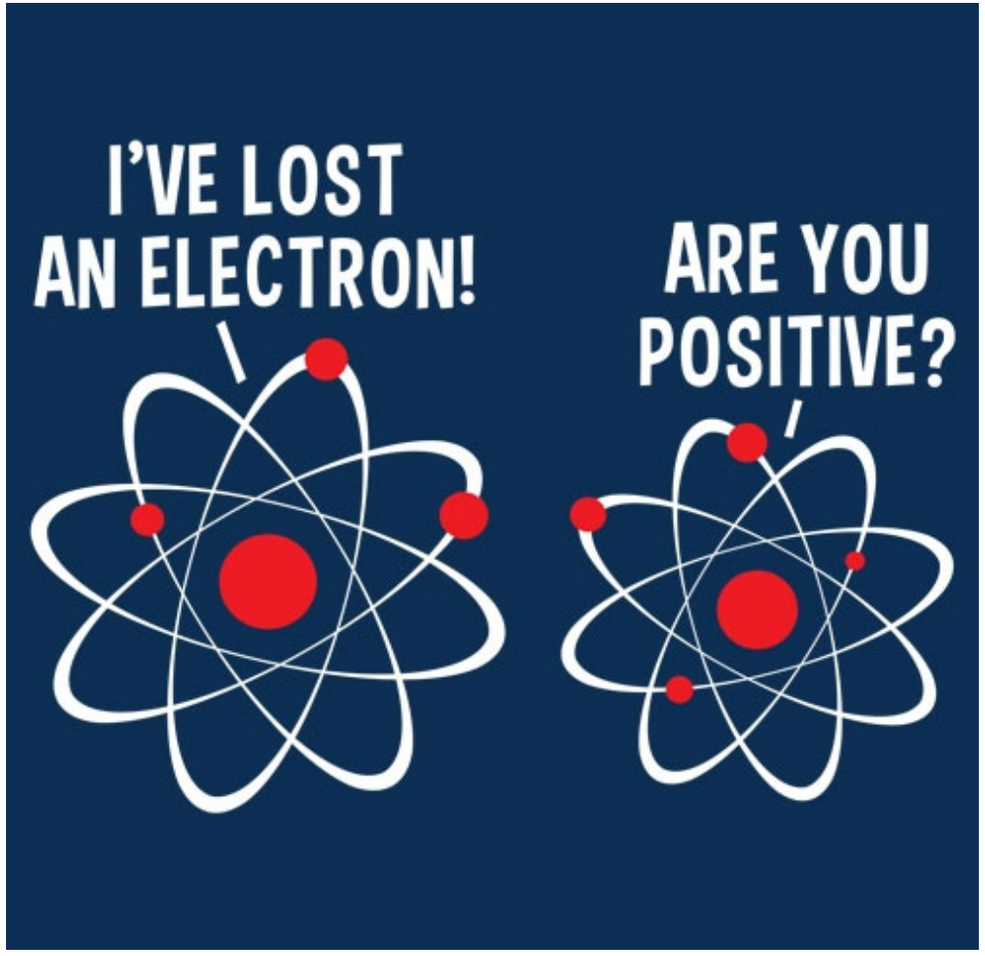
Catalin Fratila
@catalinfratila

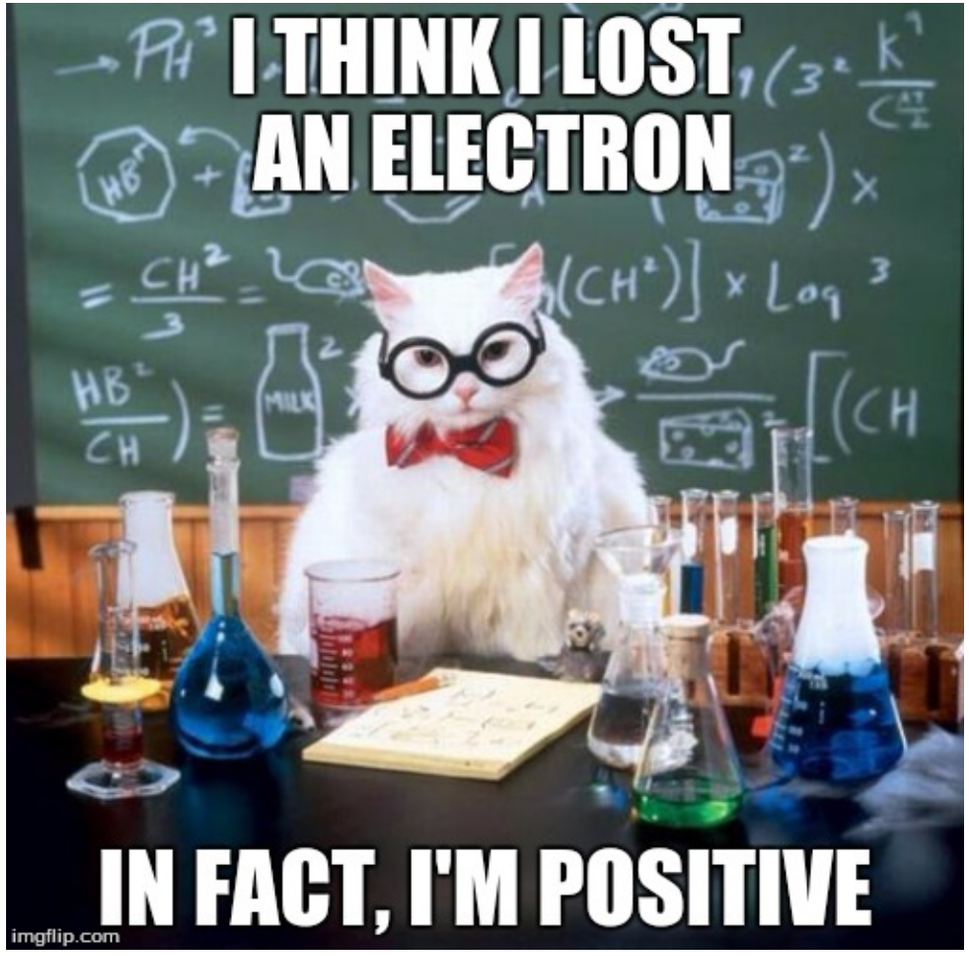
Skype

Roadmap

- What is Electron?
- My first Electron App
- Features – Electron API
- Missing functionality?
- Security
- Microsoft and Electron
- Future of Electron
- Pros and Cons

Roadmap


I'VE LOST AN ELECTRON!
ARE YOU POSITIVE?


I THINK I LOST AN ELECTRON
IN FACT, I'M POSITIVE
imgflip.com

About

ELECTRON

- Open source framework and runtime
- 2013 – Atom Shell
- Chromium, Node.JS
- Build apps with HTML, CSS, JavaScript, C/C++

GitHub

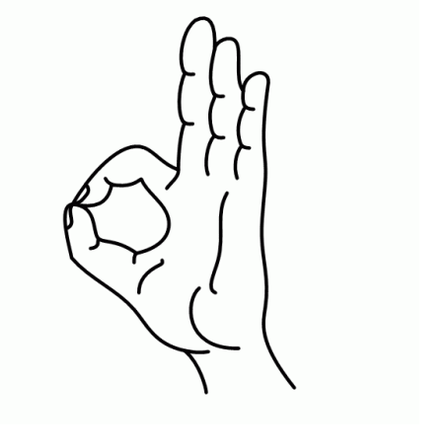
Microsoft

slack

Web Applications for Desktop?

Web Applications for Desktop?

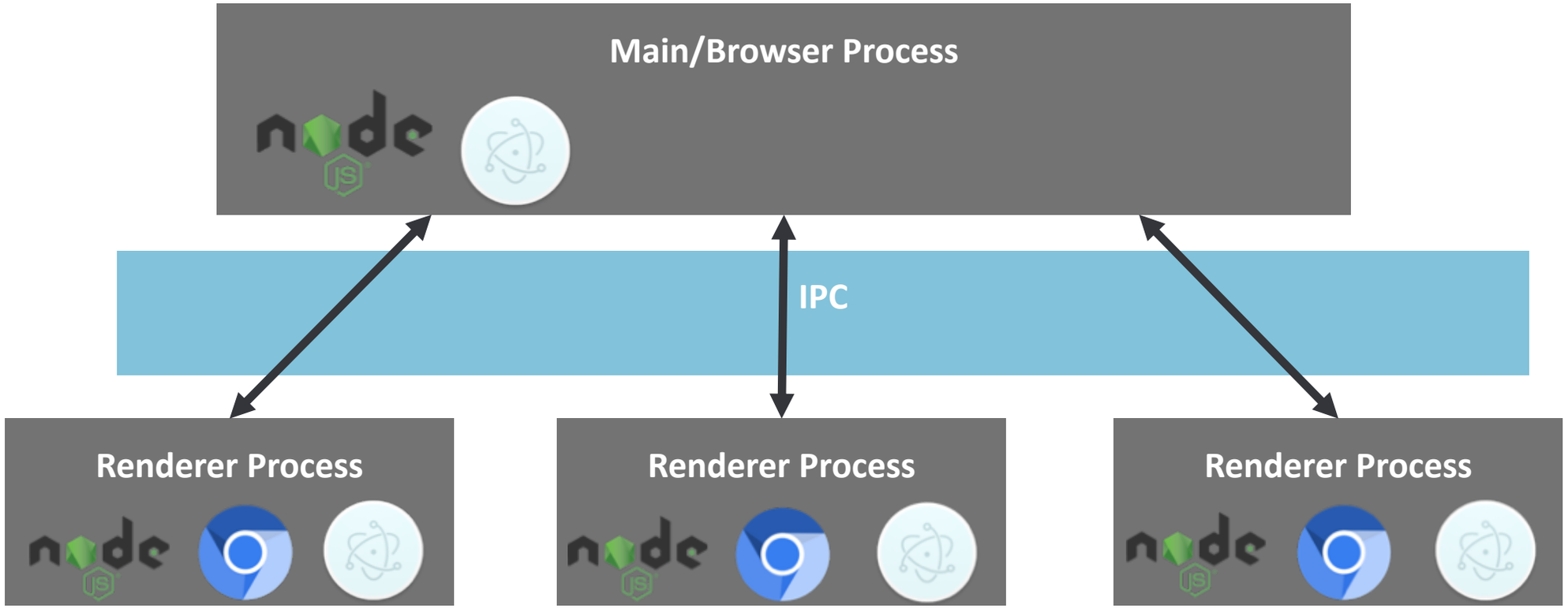
- Access peripheral devices or other hardware
- Work with files
- Notifications
- App Store
- Windows, MAC, Linux, ARM, ARM64, MIPS64L



Components

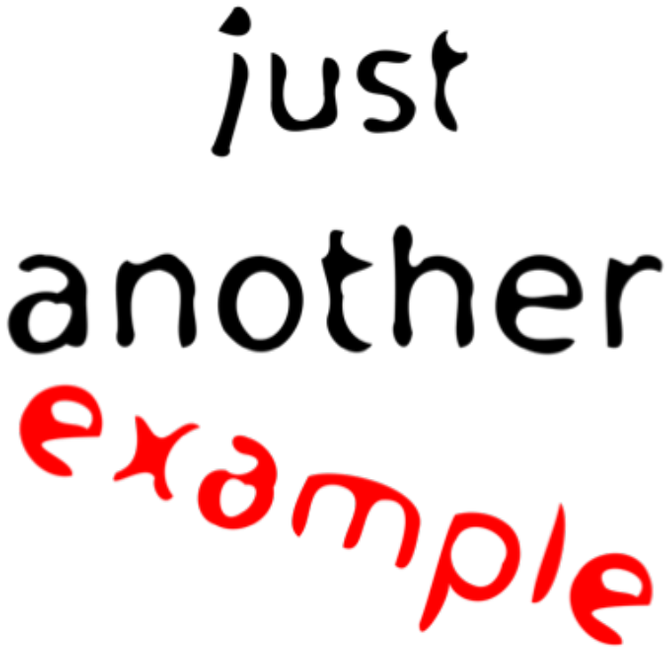
- Chromium
  - open-source web browser from Google
  - Uses V8
  - Uses Blink as layout engine
  - Electron uses only the Chromium Content API (codename: LIBCC)
- Node.JS
  - Open-source JavaScript runtime
  - Uses V8
  - Event-driven, non-blocking I/O
  - npm

Architecture



Roadmap checkpoint

- What is Electron?
- **My first Electron App**
- Features – Electron API
- Missing functionality?
- Security
- Microsoft and Electron
- Future of Electron
- Pros and Cons



What do you need?

index.html

main.js

package.json

package.json

```
{
  "name": "my-app",
  "version": "1.0.0",
  "description": "A minimal Electron application",
  "main": "main.js",
  "scripts": {
    "start": "electron ."
  },
  "devDependencies": {
    "electron": "~1.8.2-beta.5"
  }
}
```

main.js

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

const path = require('path')
const url = require('url')

let window = null

function createWindow () {
  mainWindow = new BrowserWindow({
    width: 800,
    height: 600,
    show: false,
})

mainWindow.loadURL(url.format({
  pathname: path.join(__dirname, 'index.html’),
  protocol: 'file:’,
  slashes: true
}))

mainWindow.on('closed', function () {
    mainWindow = null
  })
}

app.on('ready', createWindow)

// In this file you can include the rest of your app's specific main process
// code. You can also put them in separate files and require them here.
```

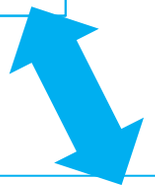
index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <!-- All of the Node.js APIs are available in this renderer process. -->
    We are using Node.js <script>document.write(process.versions.node)</script>,
    Chromium <script>document.write(process.versions.chrome)</script>,
    and Electron <script>document.write(process.versions.electron)</script>.
    <script>
      // You can also require other files to run in this process
      require('./renderer.js’)
    </script>
  </body>
</html>
```

Main ↔ Renderer

```
const { ipcMain } = require('electron');

ipcMain.on('hello-async', (event, arg) => {
  console.log(arg) // print 'hello-renderer'
  event.sender.send('hello-async-reply', 'hello-main')
});

ipcMain.on('hello-sync', (event, arg) => {
  console.log(arg)
  event.returnValue = 'hello-main'
});
```

Main

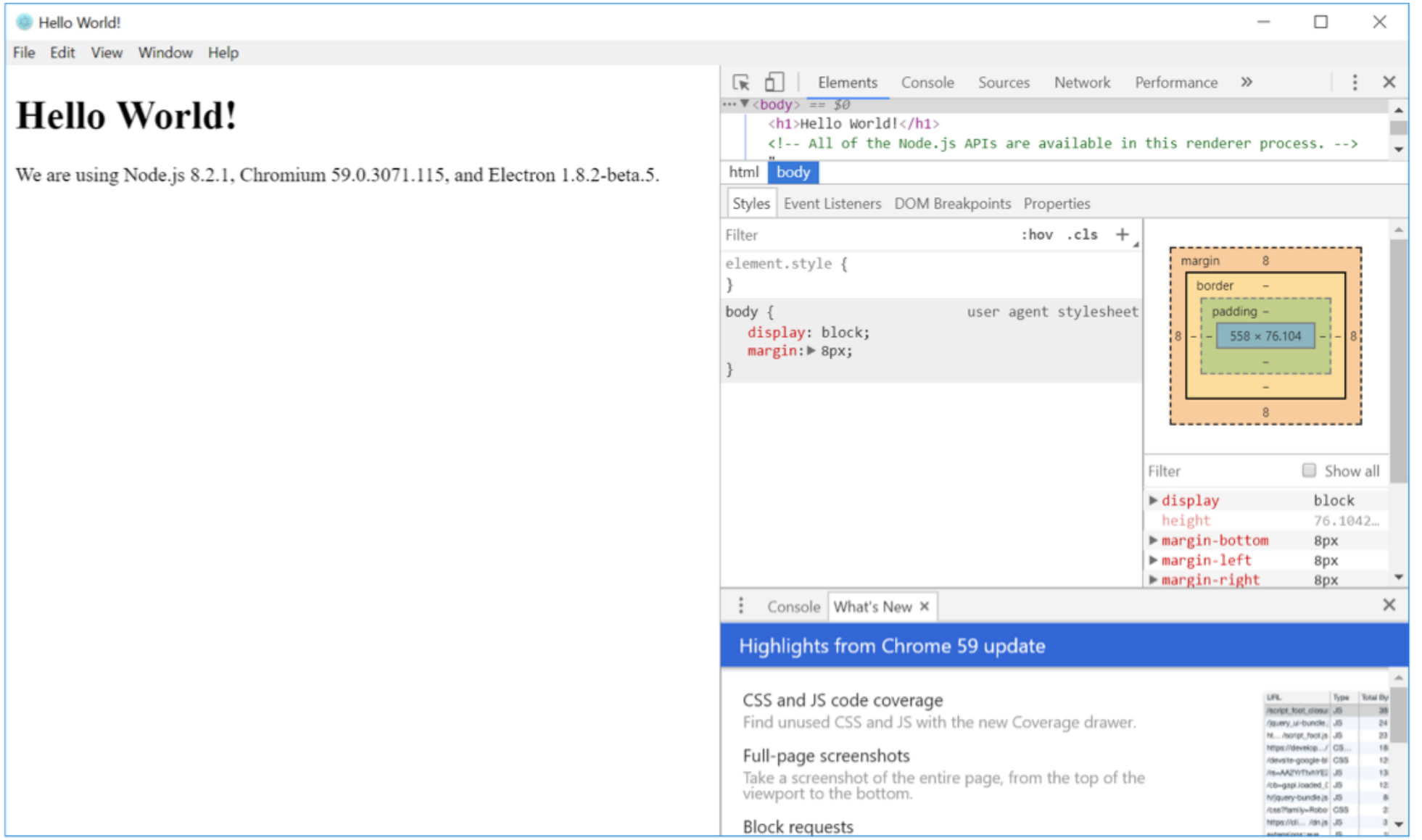


```
const { ipcRenderer } = require('electron');

console.log(ipcRenderer.sendSync('hello-sync', 'hello-renderer')) // prints "hello-main"

ipcRenderer.on('hello-async-reply', (event, arg) => {
  console.log(arg) // prints "hello-main"
});

ipcRenderer.send('hello-async', 'hello-renderer');
```

Renderer

Result

```
> npm install
> npm start
```



Roadmap checkpoint

- What is Electron?
- My first Electron App
- **Features – Electron API**
- Missing functionality?
- Security
- Microsoft and Electron
- Future of Electron
- Pros and Cons

Main process

- app
- autoUpdater
- dialog
- ipcMain
- powerMonitor
- Tray
- TouchBar*

Renderer process

- ipcRender
- remote
- webFrame

Shared

- clipboard
- crashReporter
- nativeImage
- process
- screen
- shell

Where to find more knowledge

- Electron Docs – https://electronjs.org/docs
- Introducing Electron by Felix Rieseberg https://www.safaribooksonline.com/library/view/introducing-electron/9781491996041/
- Electron Userland - https://github.com/electron-userland
- Pluralsight course - https://www.pluralsight.com/courses/electron-fundamentals

Roadmap checkpoint

- What is Electron?
- My first Electron App
- Features – Electron API
- **Missing functionality?**
- Security
- Microsoft and Electron
- Future of Electron
- Pros and Cons

Going native

- Native node modules
- How do I integrate:
  - npm + setting some environment variables
  - electron-rebuild: https://github.com/paulcbetts/electron-rebuild
  - Manually using node-gyp: https://www.npmjs.com/package/node-gyp
- Troubleshooting:
  - V8 headers incompatibility
  - 32 vs 64 bit
  - *When in doubt run electron-rebuild*

Going native

- Keytar
- Sqlite3
- Spellchecker
- … many more
- Custom?

Roadmap checkpoint

- What is Electron?
- My first Electron App
- Features – Electron API
- Missing functionality?
- **Security**
- Microsoft and Electron
- Future of Electron
- Pros and Cons

Change your mindset

- Electron is not a Browser
- The code wields the power
  - JS that can access I/O, shell and many more
- Do not use remote content
- Be on the latest Electron to avoid Chromium and Node.JS security issues

Checklist

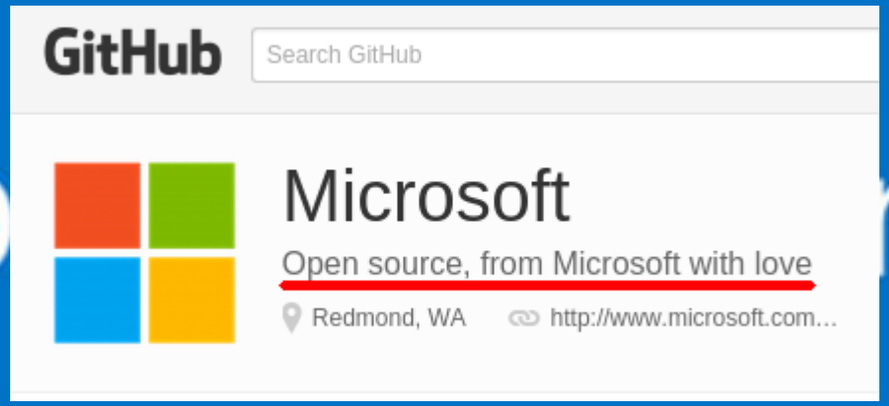
https://electronjs.org/docs/tutorial/security

- Only display secure content
- Disable Node integration on renderers (remote content)
- Enable contextIsolation
- Do not disable webSecurity
- Use CSP
- Override and disable eval
- Many more…

Roadmap checkpoint

- What is Electron?
- My first Electron App
- Features – Electron API
- Missing functionality?
- Security
- **Microsoft and Electron**
- More…
- Future of Electron
- Pros and Cons

Micro



nSource

Microsoft and Electron

- Electron Mini-Summit
  - San Francisco 2017
  - Tokyo 2017 (hosted at MS Tokyo Office)
  - Prague 2018 (will be hosted at MS Prague Office)
- Electron Core-Summit (Prague 2017)
- Security

Microsoft and Electron

- Chromium upgrades
- Build infrastructure
- Native Notifications on Windows 7
- Security patches
- Chromium DCHECK

What did we learn?

- How to optimize and profile web apps
- Grow our skill set
- The pain of upgrading to newer Electron versions
- Electron Community – open-source

Roadmap checkpoint

- What is Electron?
- My first Electron App
- Features – Electron API
- Missing functionality?
- Security
- Microsoft and Electron
- **Future of Electron**
- Pros and Cons

The future of Electron

- Electron 2.0 and beyond
- Security
- Modularity
- Faster releases and Chromium integration
- GitHub issues…

Roadmap checkpoint

- What is Electron?
- My first Electron App
- Features – Electron API
- Missing functionality?
- Security
- Microsoft and Electron
- Future of Electron
- **Pros and Cons**

The Cons

- Size!
- HTML/CSS/JS
- Not a standard for developing Desktop Applications
- Dependencies on Node.JS and Chromium
- Multi-process = higher memory usage

The Pros

- Vibrant Community
- One source code for all platforms
- HTML/CSS/JS
- Developer Tools
- Make use of the latest features
- Multitude of native modules

Microsoft

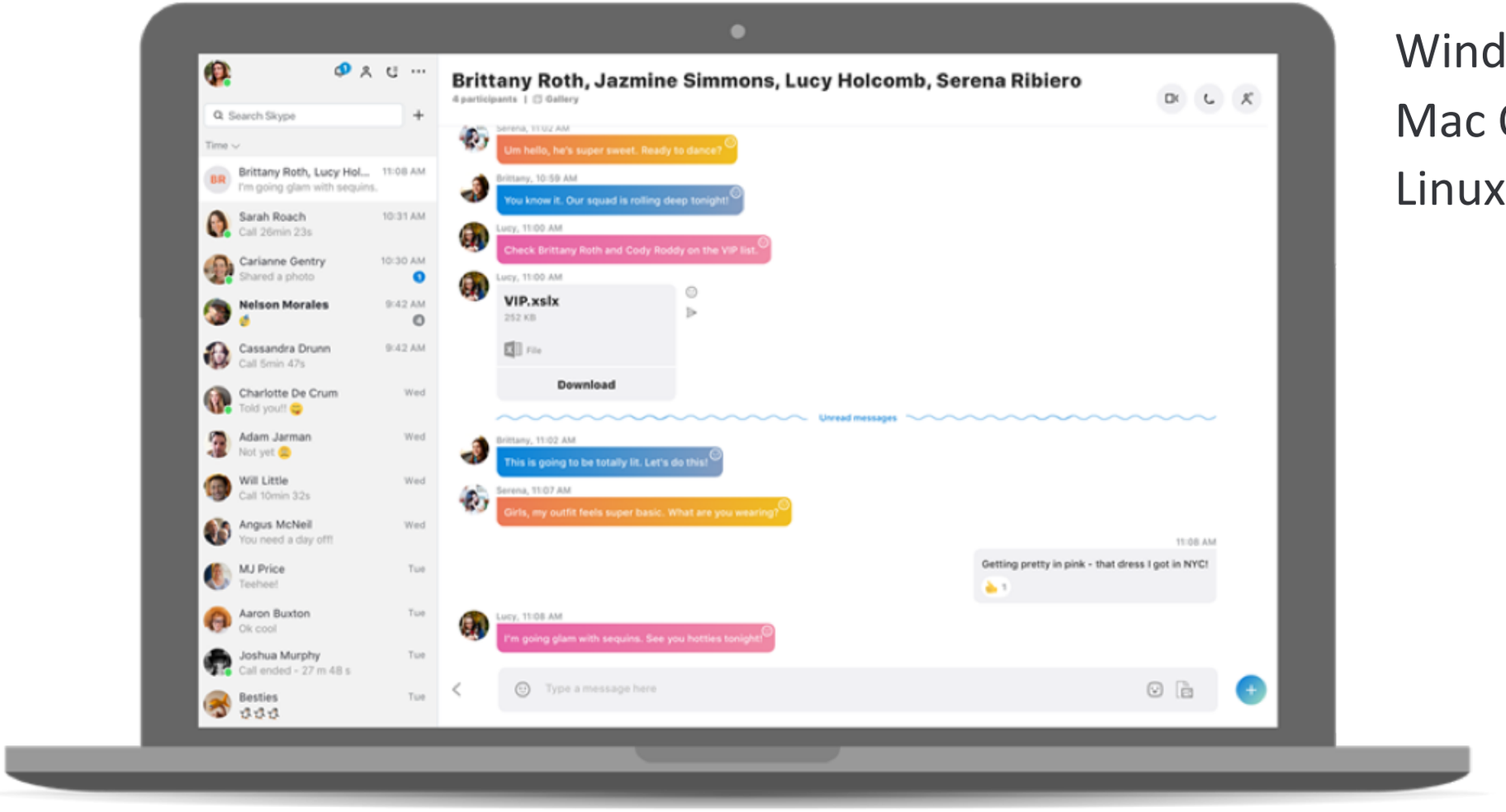
Thank you!
Bear with me a bit more ☺

Catalin Fratila
@catalinfratila

Skype

One more thing…

https://www.skype.com/en/



Windows 7, 8, 8.1, 10 TH1/2

Mac OS X 10.9+

Linux

Microsoft  Linux

Maybe one more ☺

- https://snapcraft.io/
- https://snapcraft.io/skype
- https://insights.ubuntu.com/2018/02/01/skype-now-available-as-a-snap-for-linux-users/

- Since 1st February 2018 Skype is now also availble as a Snap package

Microsoft

Try out the new Skype and let us know how you like it

Catalin Fratila
@catalinfratila

Skype