

**YES, YOUR
BROWSER CAN*
DO THAT**

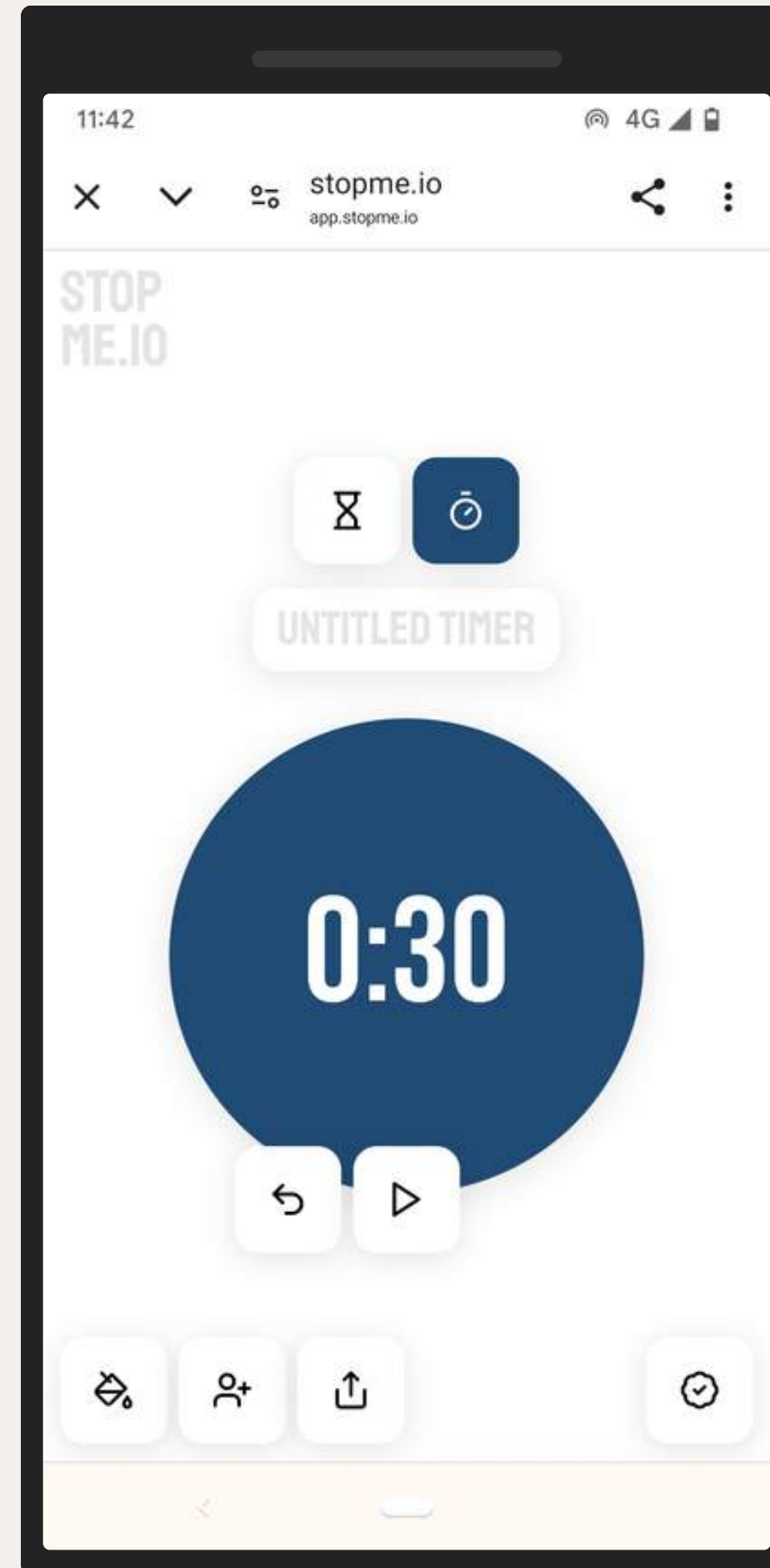
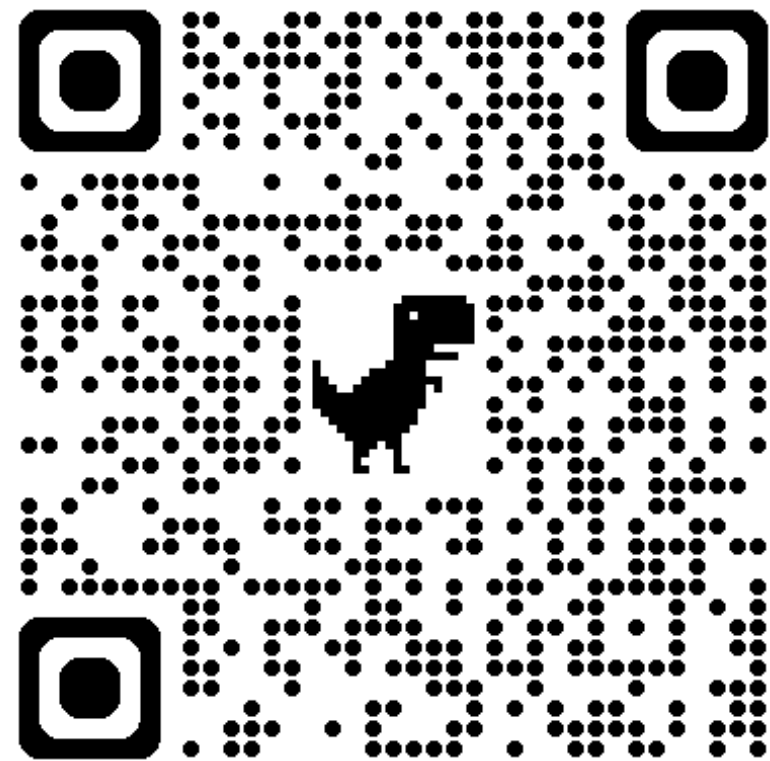
**probably*



Progressive enhancement is a design philosophy that provides a baseline of essential content and functionality to as many users as possible while delivering the best possible experience only to users of the most modern browsers

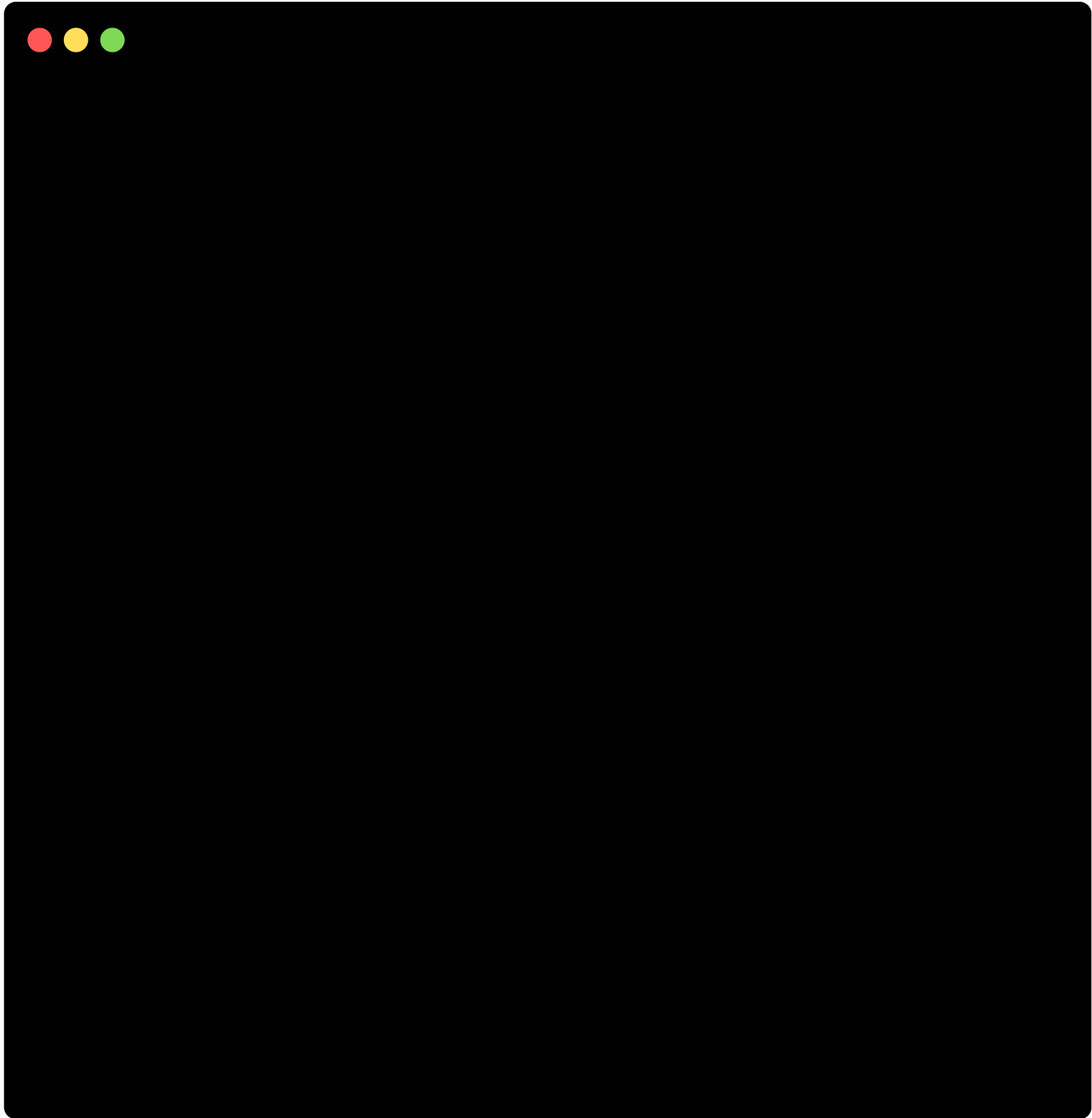
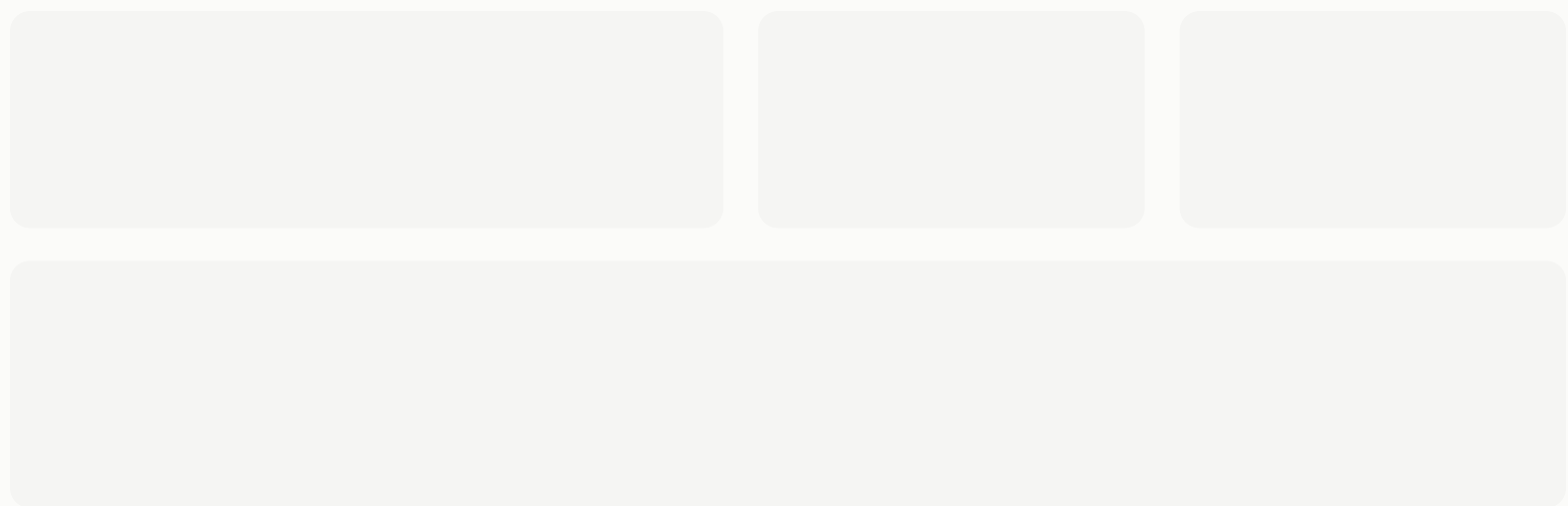
STOPME.IO

The ultimate SaaS (Stopwatch as a Service) product for everyone



**OBSERVE
YOUR APP**

RESIZE & INTERSECTION OBSERVER



RESIZE & INTERSECTION OBSERVER

```
const callback = (entries) => {  
  entries.forEach((entry) => {  
    // *.contentBoxSize  
  });  
}  
  
const observer =  
  new ResizeObserver(callback, options);  
  
const el = document.querySelector("element");  
observer.observe(el);
```

RESIZE & INTERSECTION OBSERVER

```
const callback = (entries) => {
  entries.forEach((entry) => {
    // *.isIntersecting
    // *.intersectionRect
    // *.intersectionRatio
    // ...
  });
}

const observer =
  new IntersectionObserver(callback, options);

const el = document.querySelector("element");
observer.observe(el);
```

RESIZE & INTERSECTION OBSERVER



Baseline Widely available

```
const callback = (entries) => {
  entries.forEach((entry) => {
    // *.isIntersecting
    // *.intersectionRect
    // *.intersectionRatio
    // ...
  });
}

const observer =
  new IntersectionObserver(callback, options);

const el = document.querySelector("element");
observer.observe(el);
```



Baseline Widely available



Baseline Widely available

Features listed as **widely available** have a consistent history of support in each of the Baseline browsers for at least 2.5 years.



Baseline Newly available

Features listed as **newly available** work in at least the latest stable version of each of the Baseline browsers, but may not work with older browsers and devices.



Limited availability

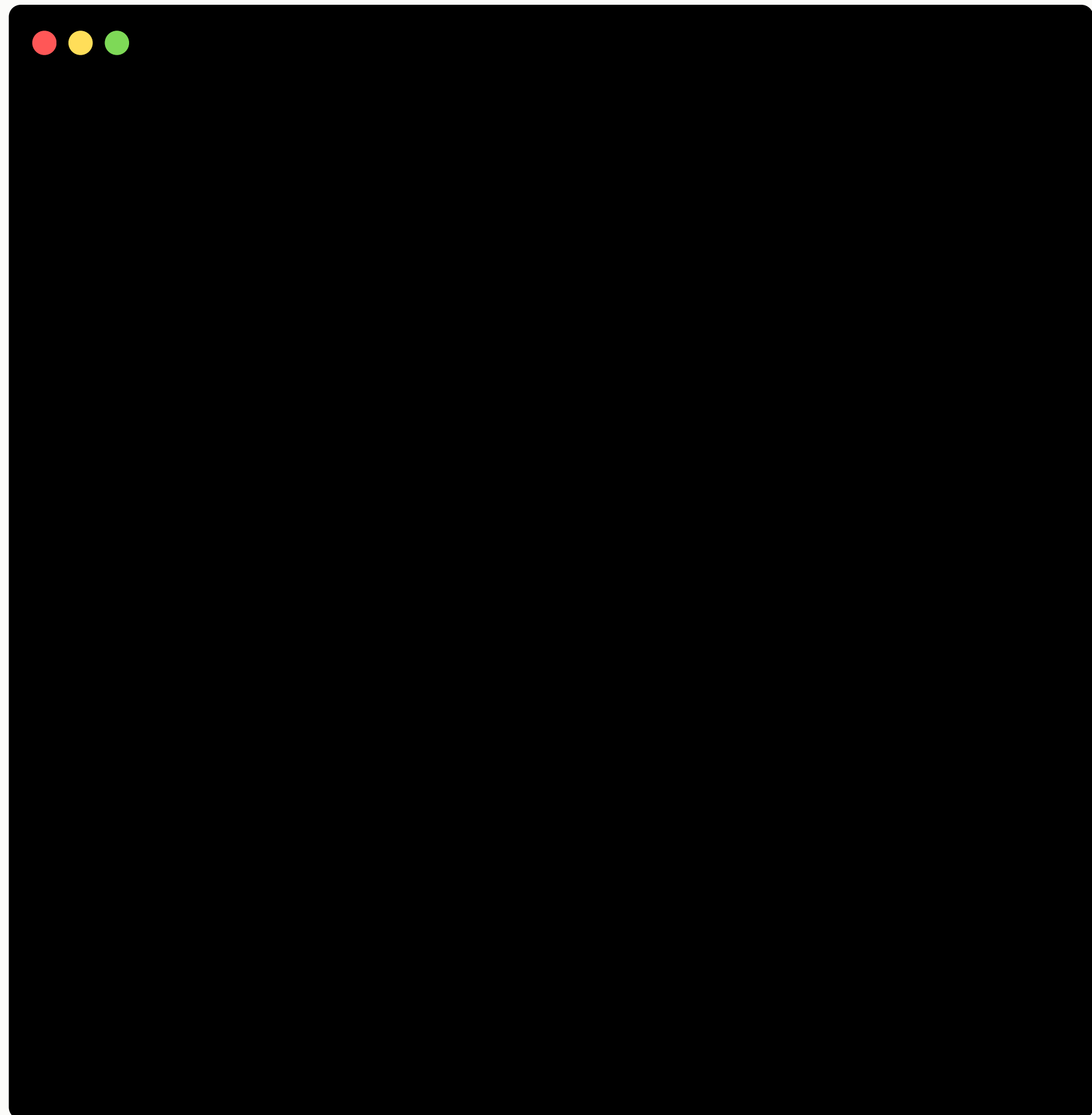
Features listed with **limited availability** are not yet available in all browsers.

**OBSERVE THE
USERS DEVICE**

PAGE VISIBILITY API



Baseline Widely available



[HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/WEB/API/PAGE_VISIBILITY_API](https://developer.mozilla.org/en-US/docs/Web/API/Page_Visibility_API)

PAGE VISIBILITY API



Baseline Widely available

```
let isHidden = document.hidden;

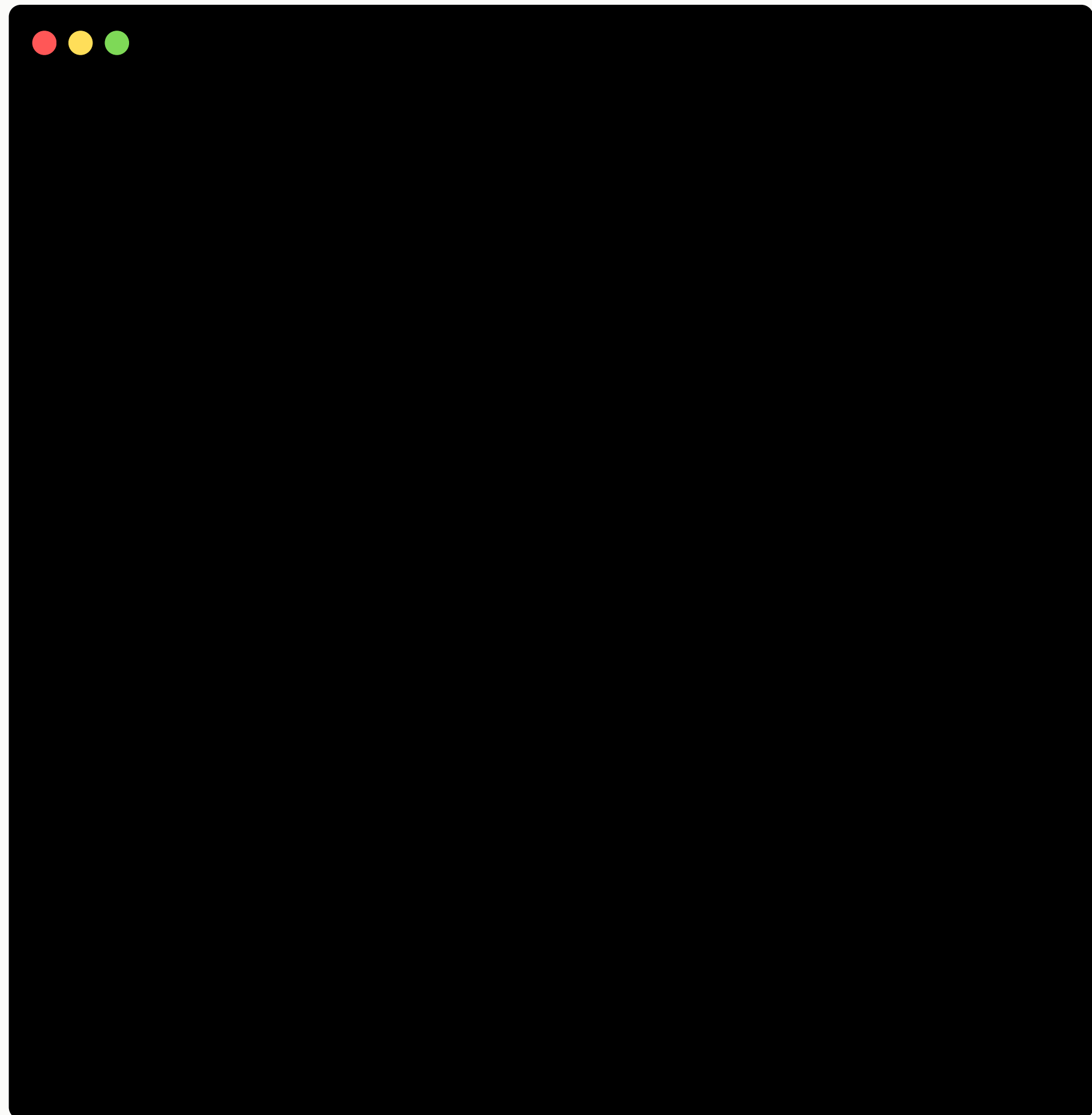
const callback = () => {
  isHidden = document.hidden;
}

document.addEventListener(
  "visibilitychange",
  callback
);
```

NETWORK INFORMATION API



Baseline Widely available



NETWORK INFORMATION API



Baseline Widely available

```
// Online status
let isOnline = navigator.onLine

const callback = () => {
  isOnline = navigator.onLine;
}

window.addEventListener("online", callback);
window.addEventListener("offline", callback);
```

NETWORK INFORMATION API



Baseline Widely available

```
let userConnection = navigator.connection;
// *.type (e.g. wifi, cellular, ...)
// *.effectiveType (e.g. 2g, 3g, ...)
// *.downlink
// *.downlinkMax
// *.rtt
// *.saveData

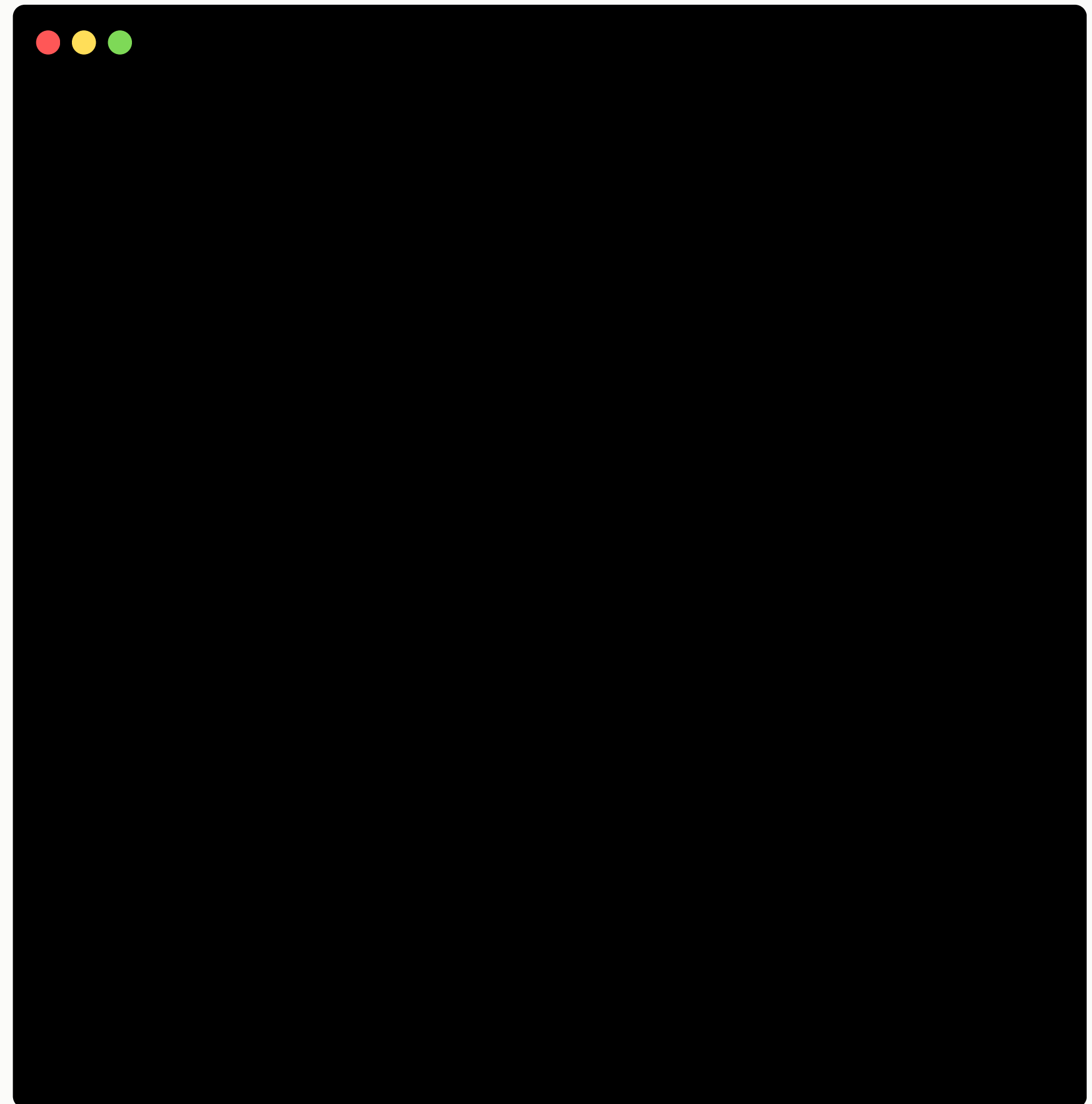
const callback = () => {
  userConnection = navigator.connection;
}

navigator.connection
  .addEventListener("change", callback);
```

BATTERY STATUS API



Limited availability



[HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/WEB/API/BATTERY_STATUS_API](https://developer.mozilla.org/en-US/docs/Web/API/Battery_Status_API)

BATTERY STATUS API



Limited availability

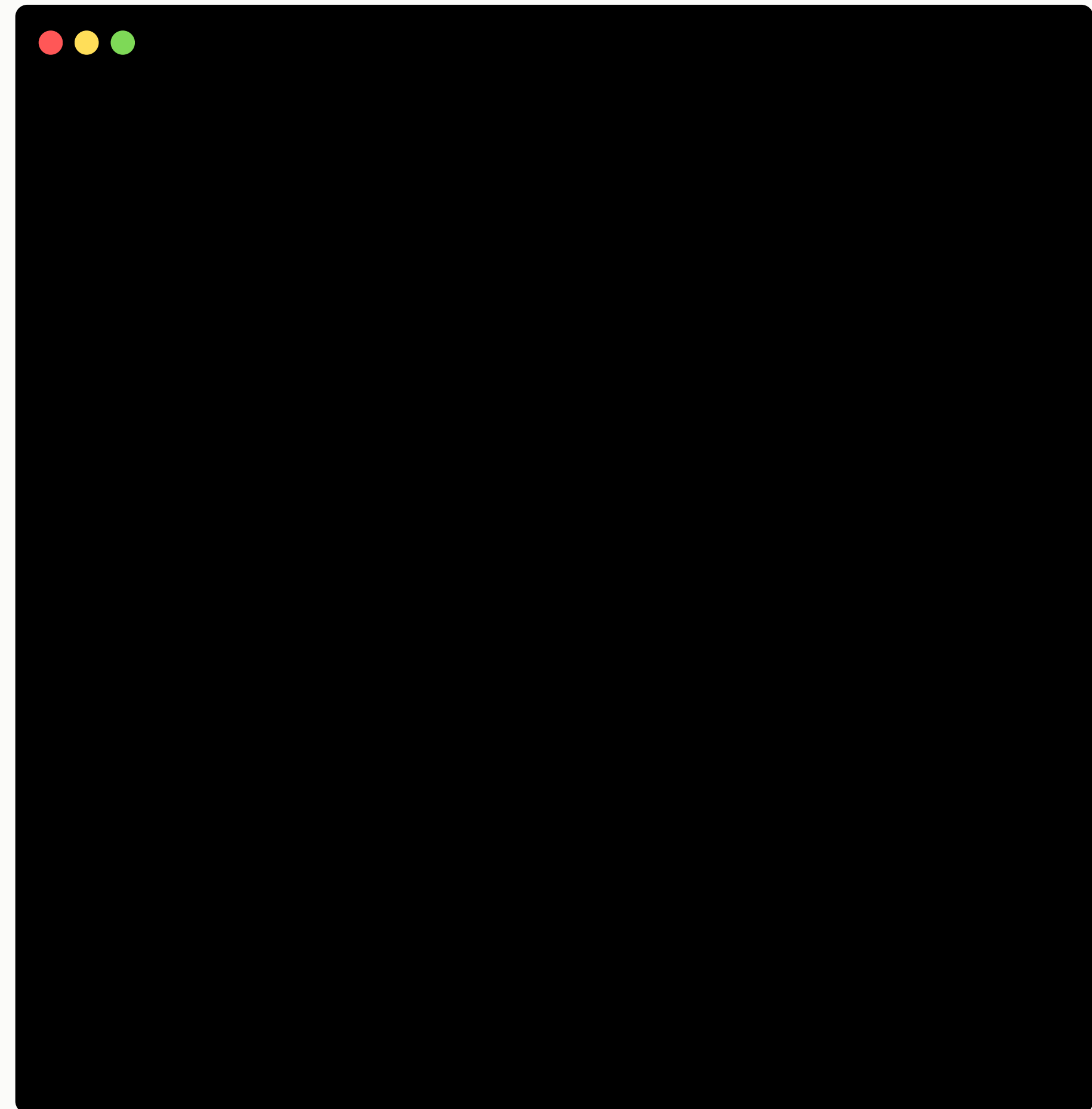
```
const info = await navigator.getBattery();  
// *.level  
// *.charging  
// *.chargingTime  
// *.dischargingTime  
  
info.addEventListener("levelchange", ...);  
info.addEventListener("chargingchange", ...);  
// ...
```

**ENHANCE YOUR
COMPONENTS**

SCREEN WAKE LOCK API



Baseline Newly available



SCREEN WAKE LOCK API



Baseline Newly available

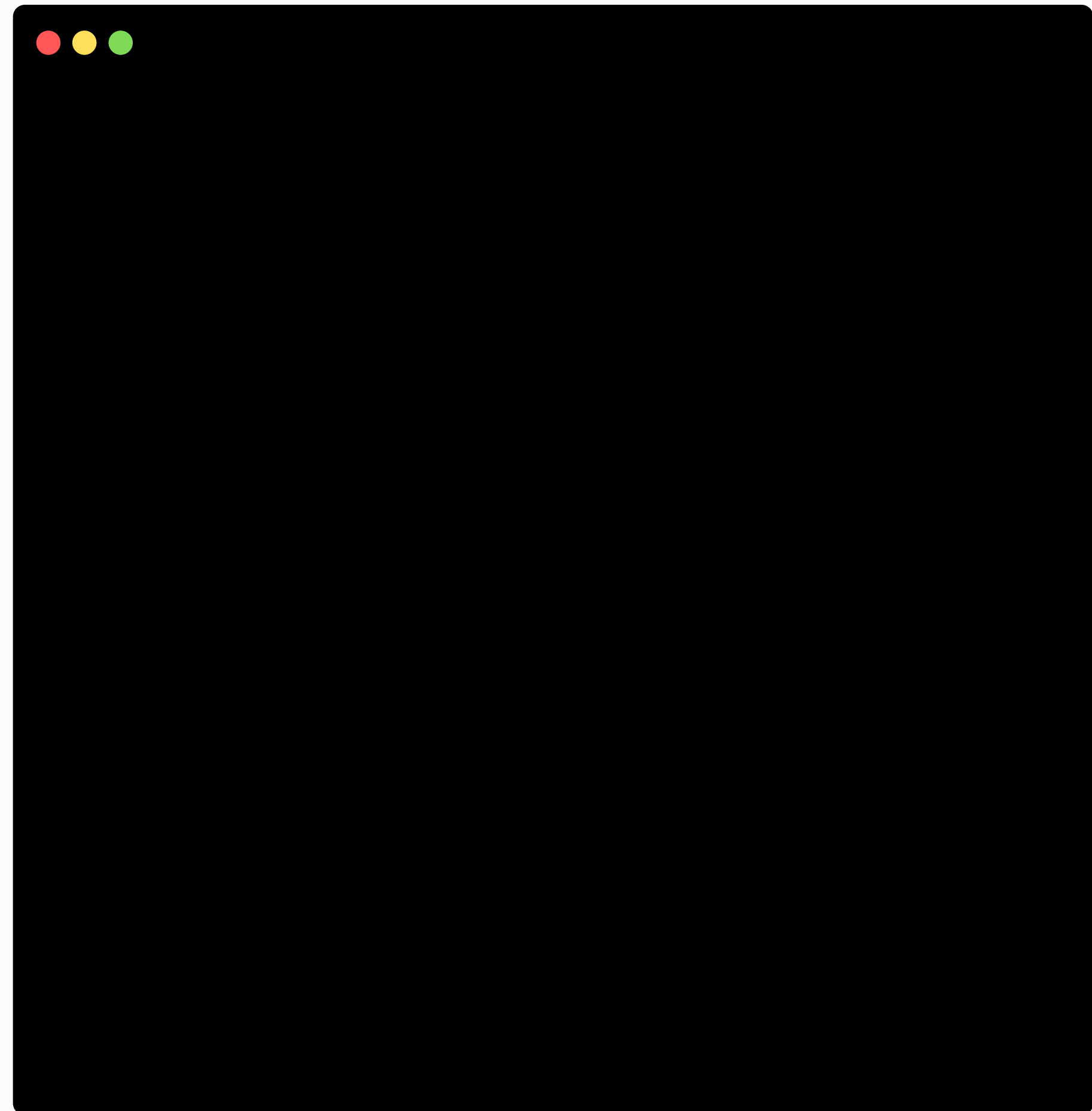
```
const wakeLock = await navigator
  .wakeLock
  .request("screen")
  .catch((e) => {
    // Lock request failed, often because
    // of low battery etc
  });

await wakeLock.release();
```

VIBRATION API



Limited availability



VIBRATION API



Limited availability

```
navigator.vibrate(duration);

// Patterns
// e.g. vibrate for 100ms with 50ms pauses
navigator.vibrate([100, 50, 100, 50, 100]);

// Stop long vibration or pattern
navigator.vibrate(0);
```

VIBRATION API



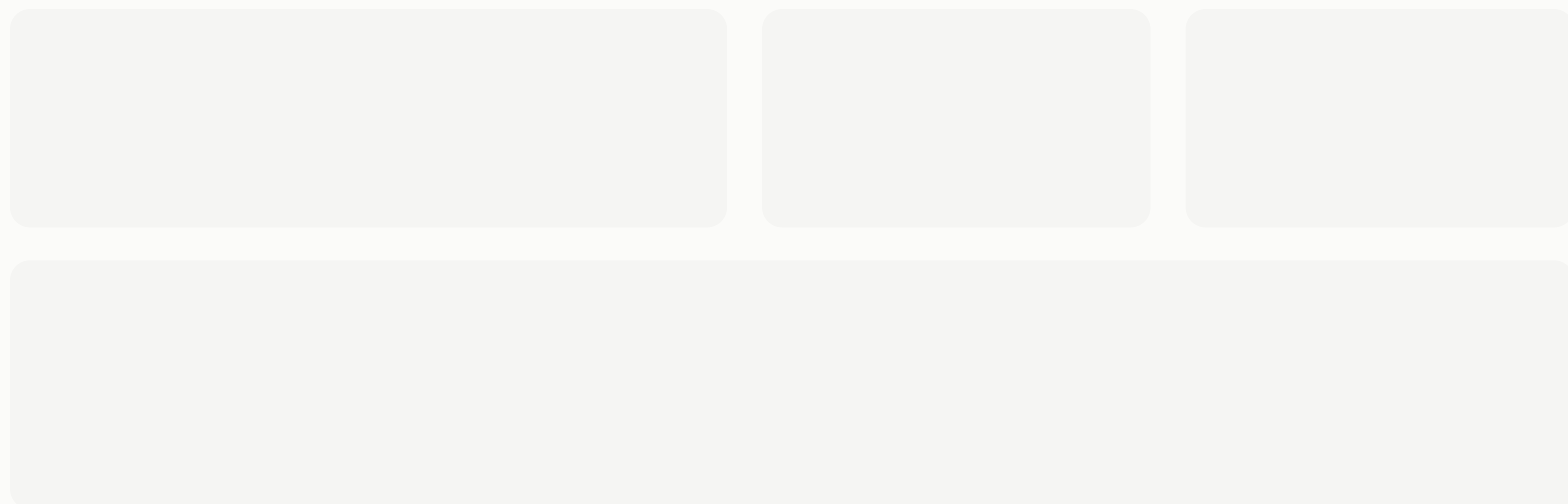
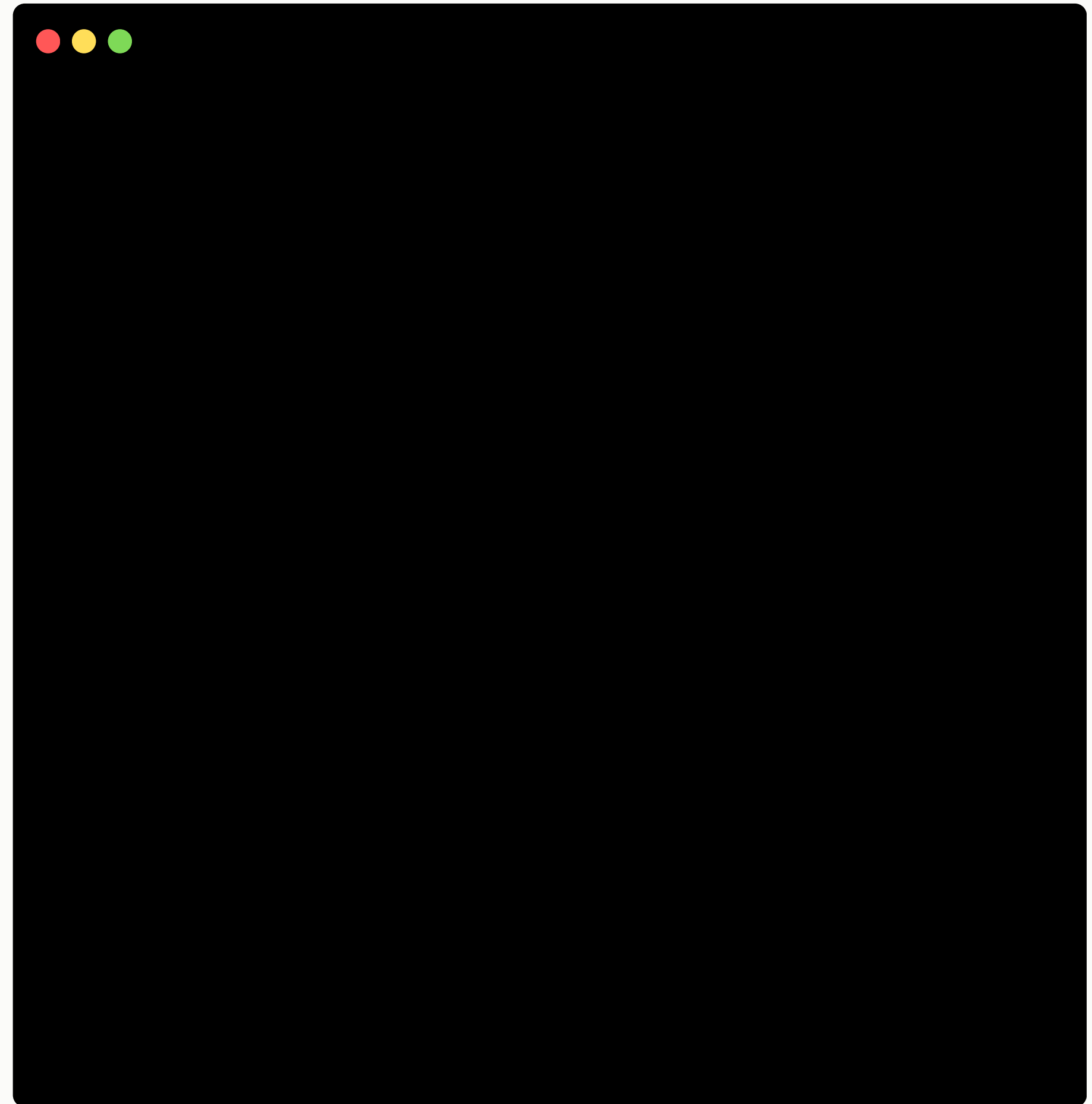
Limited availability

```
// SOS in Morse
navigator.vibrate([
  100, 30, 100, 30, 100, 30, 200, 30, 200,
  30, 200, 30, 100, 30, 100, 30, 100
]);

// Super Mario
navigator.vibrate([
  125, 75, 125, 275, 200, 275, 125, 75,
  125, 275, 200, 600, 200, 600
]);

// Star Wars
navigator.vibrate([
  500, 110, 500, 110, 450, 110, 200, 110,
  170, 40, 450, 110, 200, 110, 170, 40, 500
]);
```

FILE SYSTEM & CLIPBOARD API



FILE SYSTEM & CLIPBOARD API



Limited availability

```
const pickerOpts = {  
  // You can filter by file types & control  
  // whether you want single or multi select  
  types: [...],  
  multiple: false,  
};  
  
const [fileHandle] = await window  
  .showOpenFilePicker(pickerOpts);  
  
const fileData = await fileHandle.getFile();
```

FILE SYSTEM & CLIPBOARD API



Baseline Newly available

```
button.addEventListener("click", () => {  
  // Read from the clipboard  
  const contents = await navigator.clipboard  
    .read();  
  
  // Write to the clipboard  
  const data = {  
    ["text/plain"]: "Hello world"  
  };  
  const item = new ClipboardItem(data);  
  await navigator.clipboard.write([item]);  
});
```

**FROM NATIVE
TO WEB**



```
// ???  
const networkState = navigator.connection.type;  
document.addEventListener("online", ...);  
document.addEventListener("offline", ...);  
  
// ???  
window.addEventListener("batterystatus", (status) => {  
    ...  
});  
  
// ???  
navigator.vibrate(3000);  
navigator.vibrate([3000]);  
  
// ???  
window.requestFileSystem(..., (fs) => {  
    fs.root.getFile("test.txt", ..., (fileEntry) => {  
        writeFile(fileEntry);  
    });  
});
```



```
// cordova-plugin-network-information
const networkState = navigator.connection.type;
document.addEventListener("online", ...);
document.addEventListener("offline", ...);

// cordova-plugin-battery-status
window.addEventListener("batterystatus", (status) => {
  ...
});

// cordova-plugin-vibration
navigator.vibrate(3000);
navigator.vibrate([3000]);

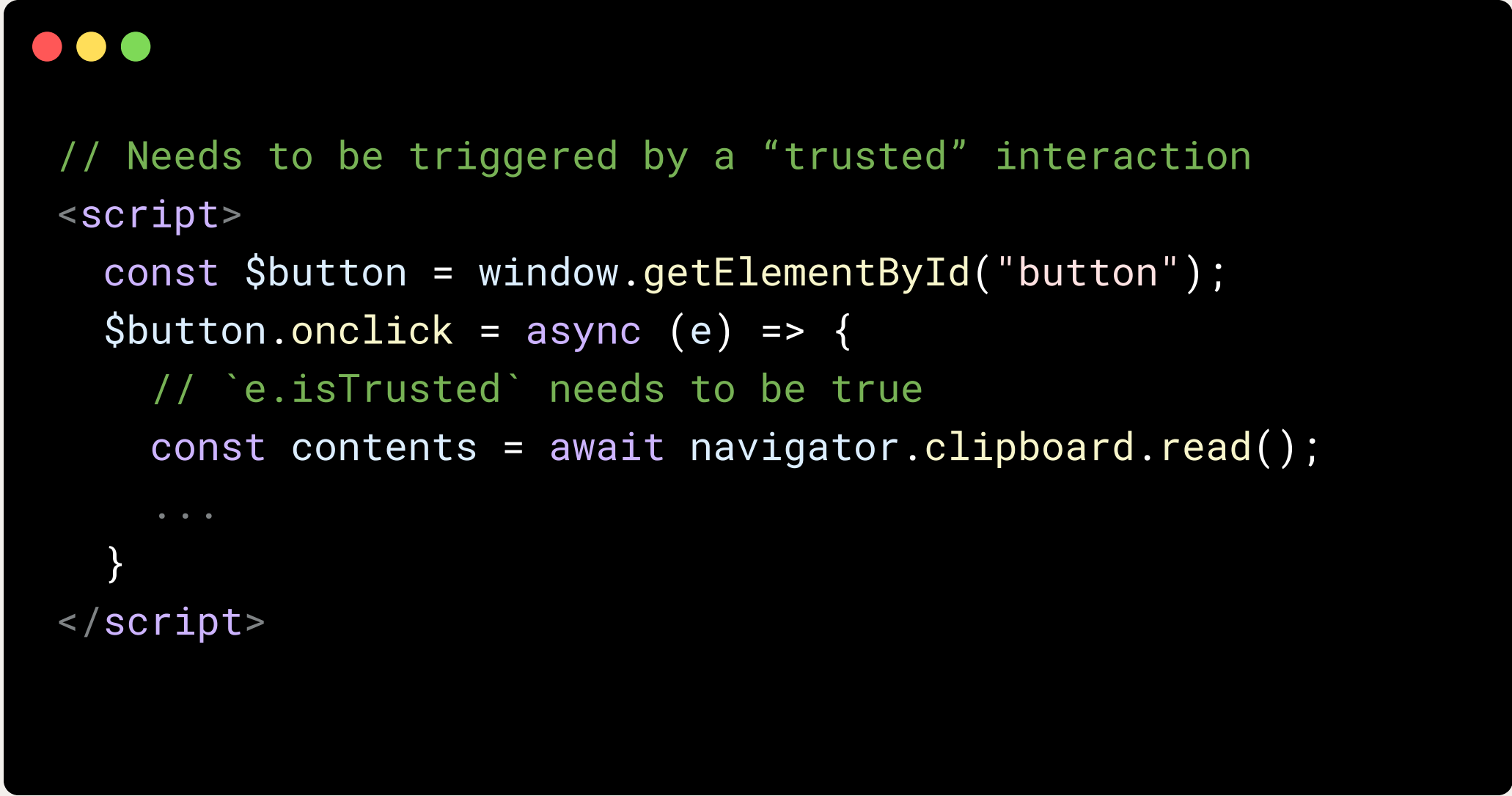
// cordova-plugin-file
window.requestFileSystem(..., (fs) => {
  fs.root.getFile("test.txt", ..., (fileEntry) => {
    writeFile(fileEntry);
  });
});
```

NO ACTIVATION REQUIRED



```
// No user interaction required
<script>
  console.log(navigator.connection.type);
  console.log(navigator.connection.downlink);
</script>
```

TRANSIENT ACTIVATION



```
// Needs to be triggered by a "trusted" interaction
<script>
  const $button = window.getElementById("button");
  $button.onclick = async (e) => {
    // `e.isTrusted` needs to be true
    const contents = await navigator.clipboard.read();
    ...
  }
</script>
```

“STICKY” ACTIVATION

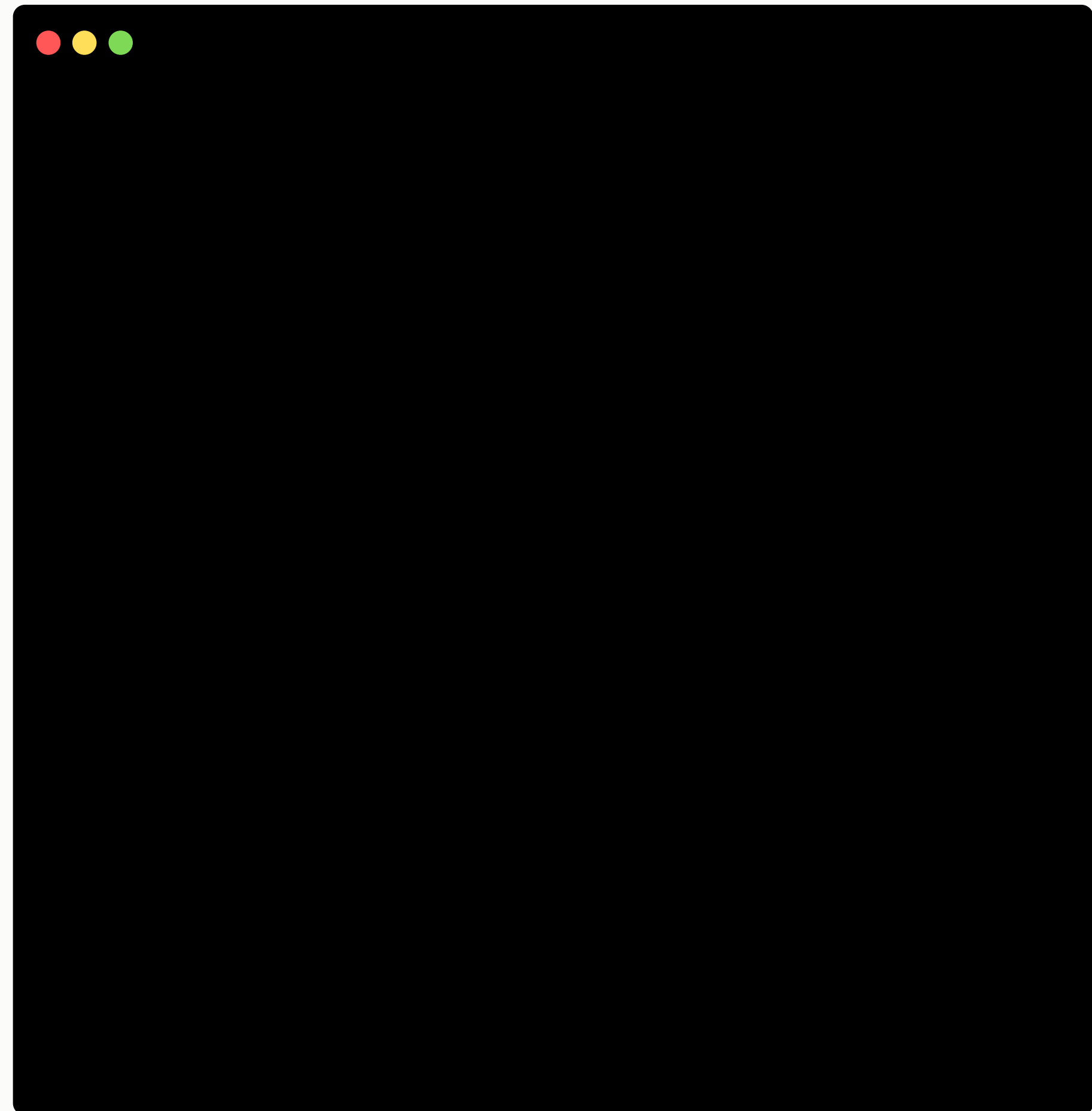


```
// Requires an initial transient activation on the site
<script>
  // This still works outside of trusted event handlers
  // as long as the user has triggered a transient
  // activation on the site before
  navigator.vibrate(500);
</script>
```

EYE DROPPER API



Limited availability



[HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/WEB/API/EYEDROPPER_API](https://developer.mozilla.org/en-US/docs/Web/API/Eyedropper_API)

EYE DROPPER API



Limited availability

```
const eyeDropper = new EyeDropper();

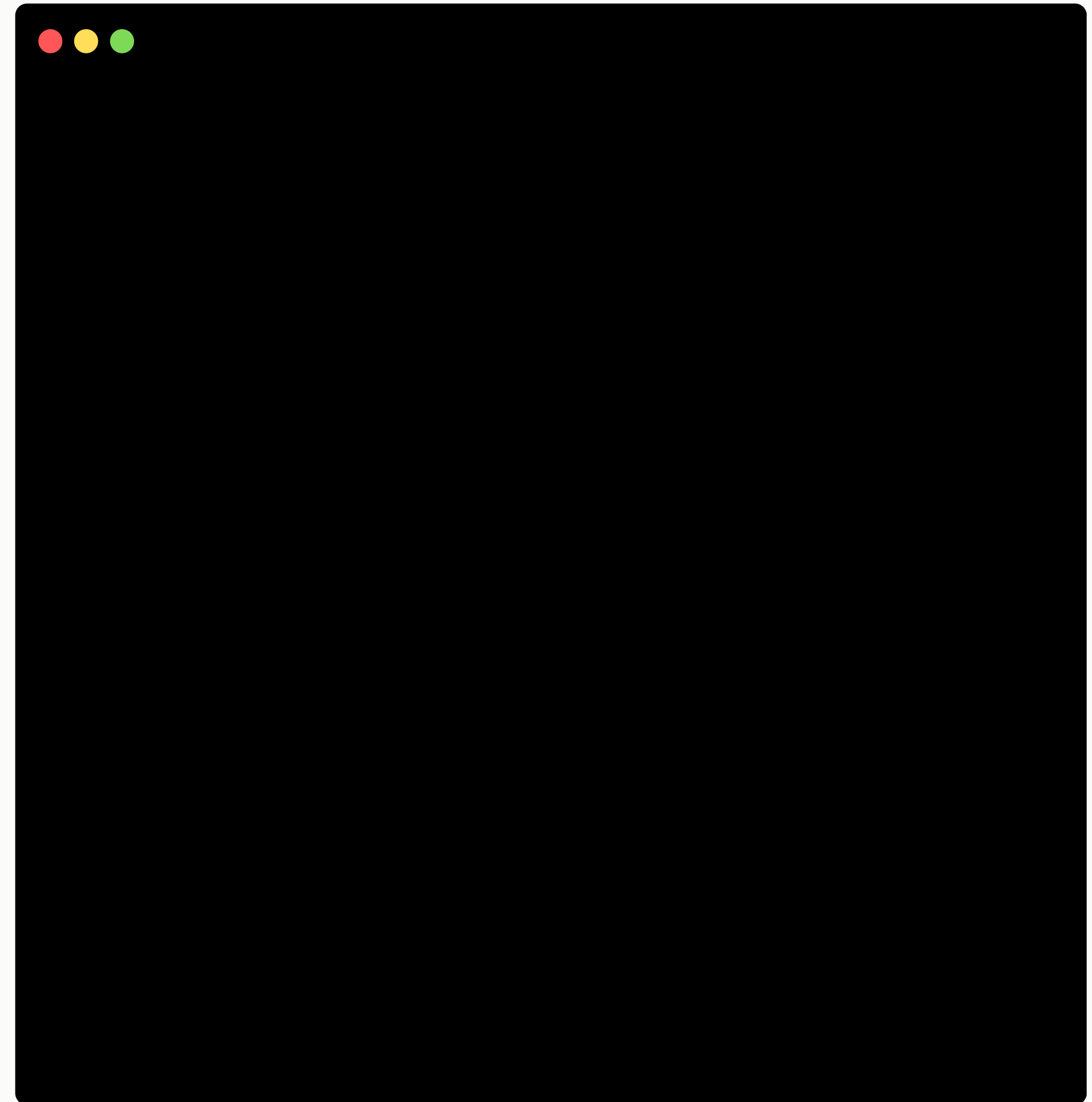
document.getElementById("btn")
  .addEventListener("click", () => {
    // Needs to be triggered by user action
    eyeDropper.open()
      .then((result) => {
        // Returns the selected color
        // *.sRGBHex
      })
      .catch((e) => {
        // Catches any errors, including
        // when the user cancels selection
      });
  });
```

**ALMOST AS
GOOD AS NATIVE**

WEB SHARE API



Limited availability



[HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/WEB/API/WEB_SHARE_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Share_API)

WEB SHARE API



Limited availability

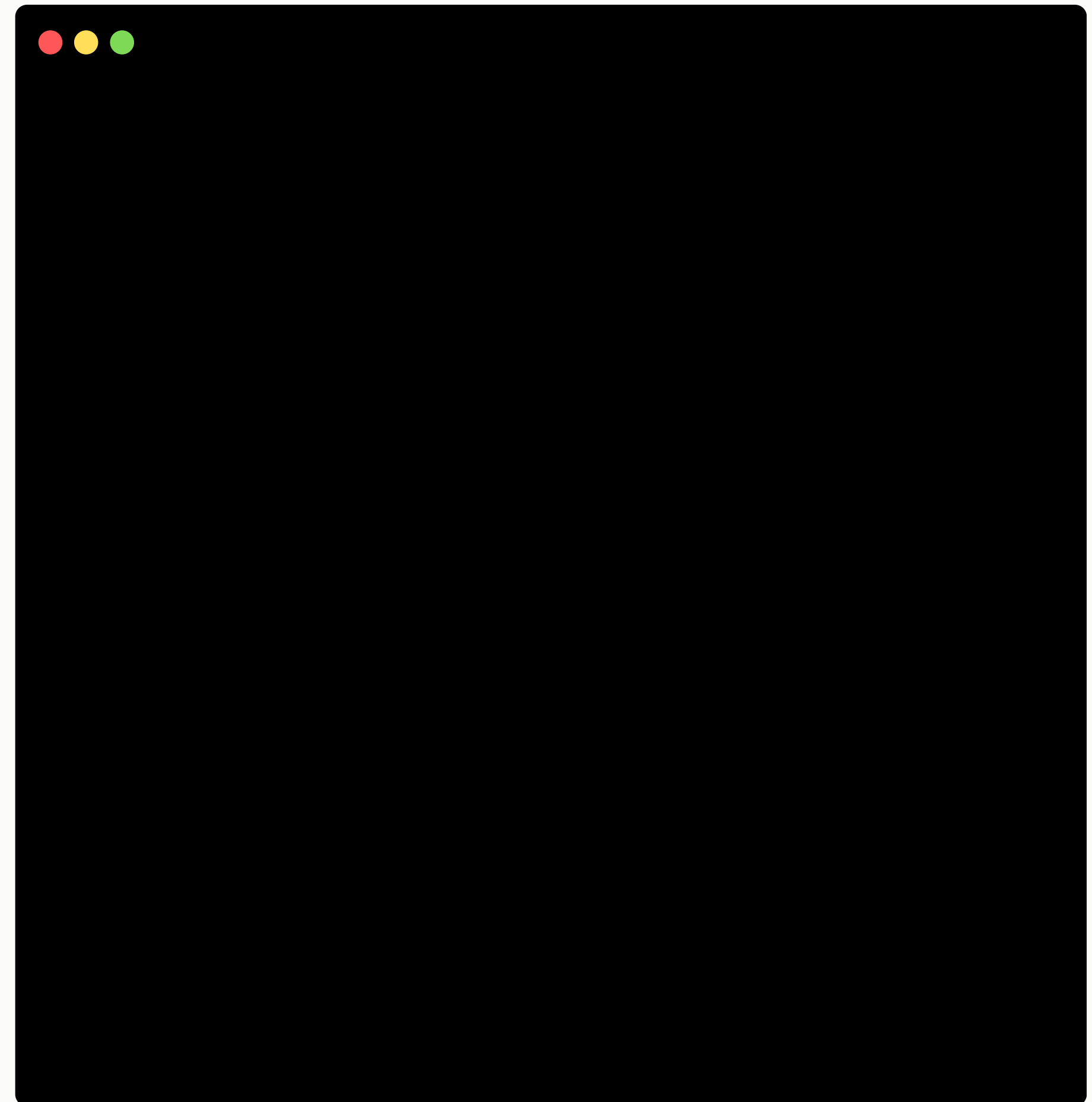
```
const shareData = {
  title: "../NEW",
  text: "Celebrate all things technology",
  url: "https://slashnew.tech"
  files: [imageFile, videoFile]
};

document.getElementById("btn")
  .addEventListener("click", () => {
    navigator.share(shareData)
      .catch((e) => {
        // Handle any errors
      });
  });
```

CONTACT PICKER API



Limited availability



[HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/WEB/API/CONTACT_PICKER_API](https://developer.mozilla.org/en-US/docs/Web/API/Contact_Picker_API)

CONTACT PICKER API



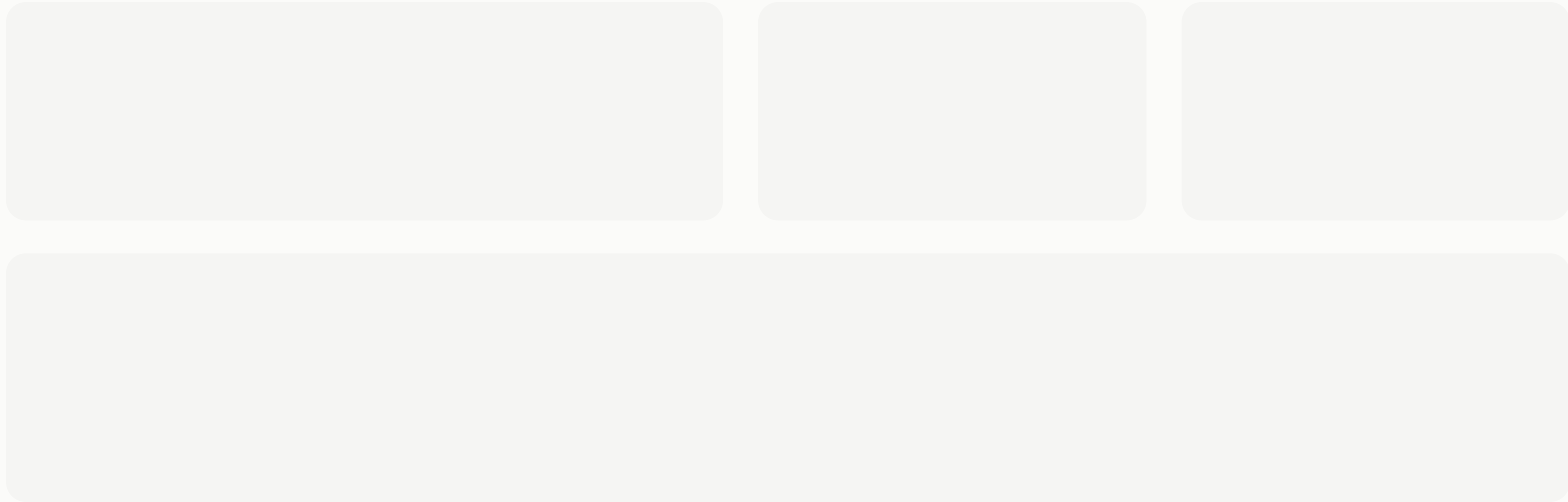
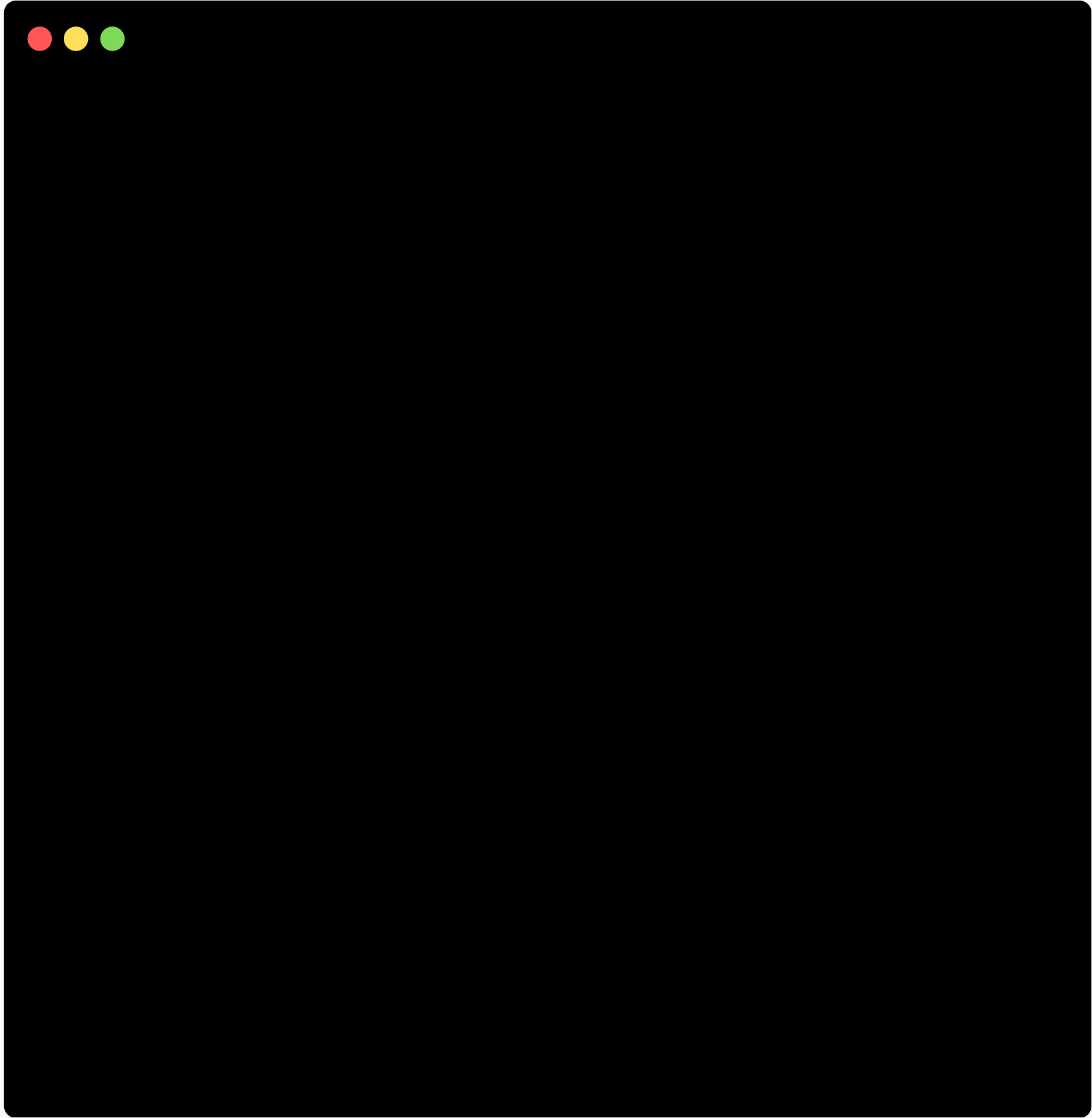
Limited availability

```
const props = ["name", "email", "tel", "icon"];
const opts = { multiple: true };

const contacts = await navigator.contacts
  .select(props, opts);
.catch((e) => {
  // Handle any errors
});
```

HONORABLE MENTIONS

AUTHENTICATION, BUT BETTER:

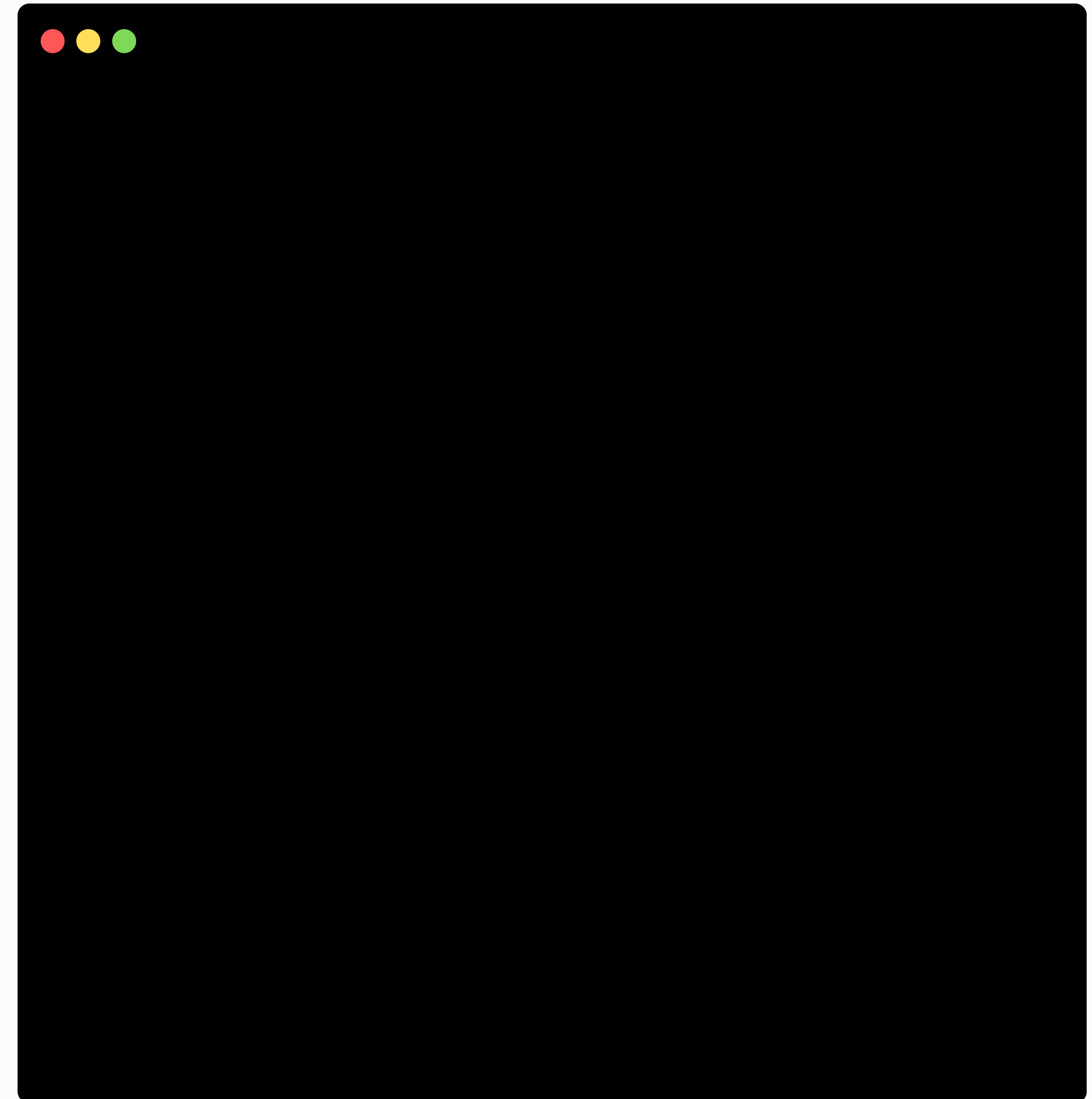


AUTHENTICATION, BUT BETTER:

WEB OTP API



Limited availability



[HTTPS://DEVELOPER.MOZILLA.ORG/EN-US/DOCS/WEB/API/WEBOTP_API](https://developer.mozilla.org/en-US/docs/Web/API/WebOTP_API)

AUTHENTICATION, BUT BETTER:

WEB OTP API



Limited availability

```
navigator.credentials
  .get({
    otp: { transport: ["sms"] },
    signal: ac.signal,
  })
  .then((otp) => {
    // *.code
  })
  .catch((e) => {
    // Handle any errors
  });

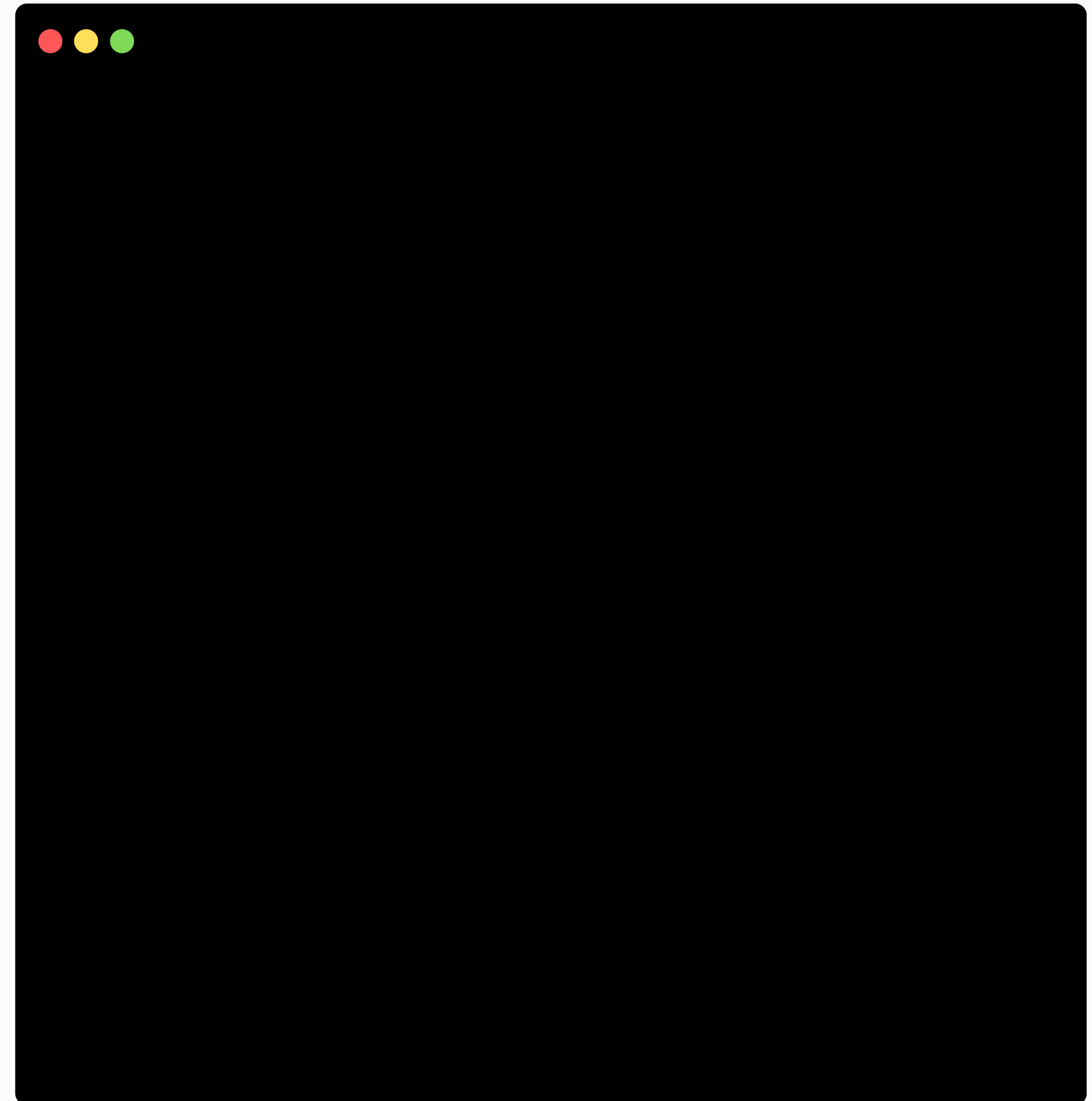
// Format of the SMS so it can be processed
//   Your verification code is 123456.\n\n
//   @app.stopme.io #123456
```

AUTHENTICATION, BUT BETTER:

CREDENTIALS API



Baseline Widely available



AUTHENTICATION, BUT BETTER:

CREDENTIALS MANAGEMENT API



Baseline Widely available

```
const cred = new PasswordCredential({
  id,
  password,
  name
});

// Store credentials
await navigator.credentials.store(cred);

// Get stored credentials
const user = await navigator.credentials.get();

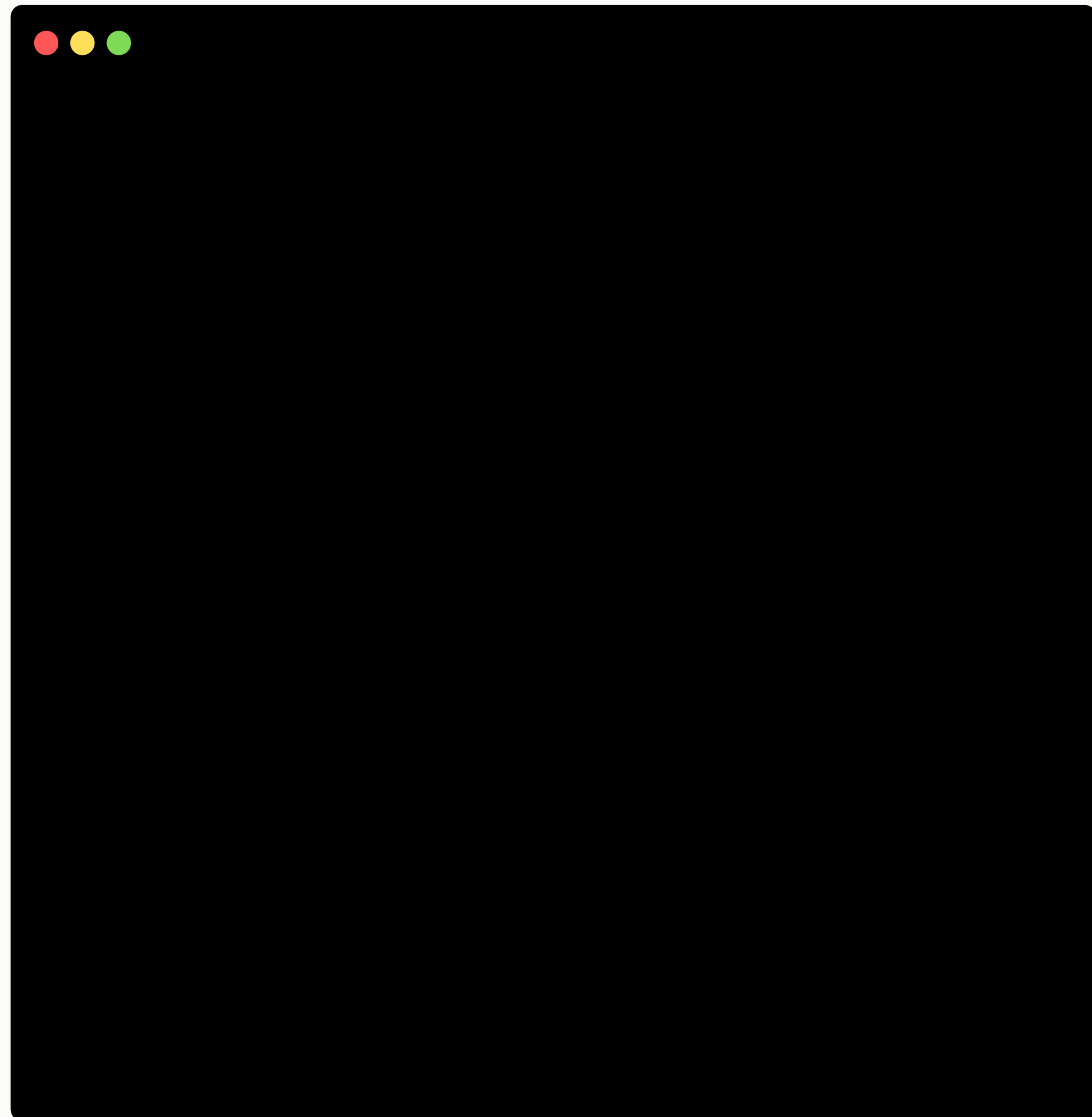
// Prevent automatic login when user signs out
await navigator
  .credentials
  .preventSilentAccess();
```

AUTHENTICATION, BUT BETTER:

AUTHN API



Baseline Widely available



AUTHENTICATION, BUT BETTER:

AUTHN API



Baseline Widely available

```
// Create credentials object on the client
// using challenge generated on the server
const registerCredential =
  await navigator.credentials.create({
    publicKey
  });

// Credentials are stored with the user
// identity are sent back and stored on the
// server

// Authenticate again using server challenge
const authCredential =
  await navigator.credentials.get({
    publicKey
  });

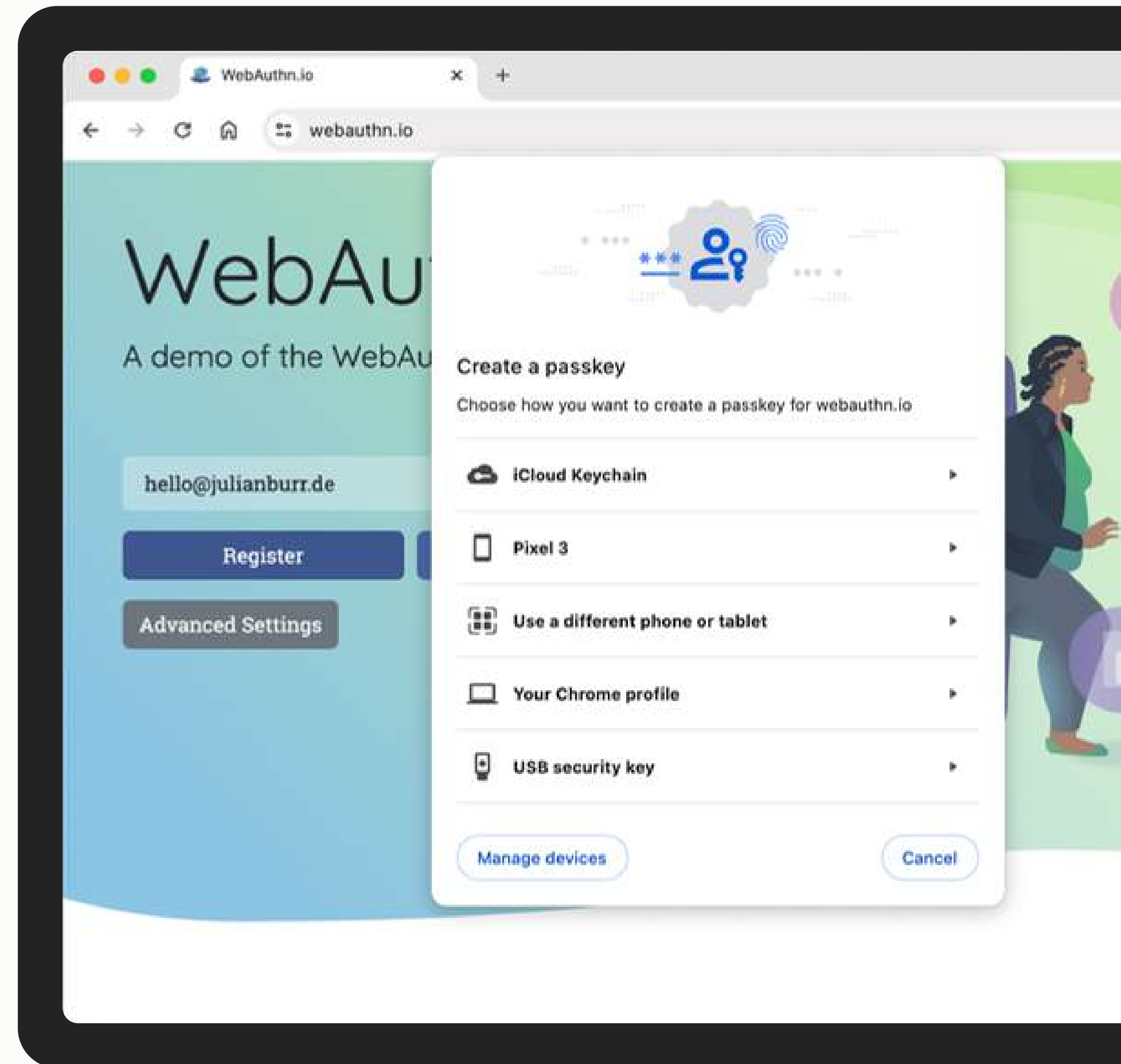
// Use stored public key to verify validity
// of auth credentials
```

AUTHENTICATION, BUT BETTER:

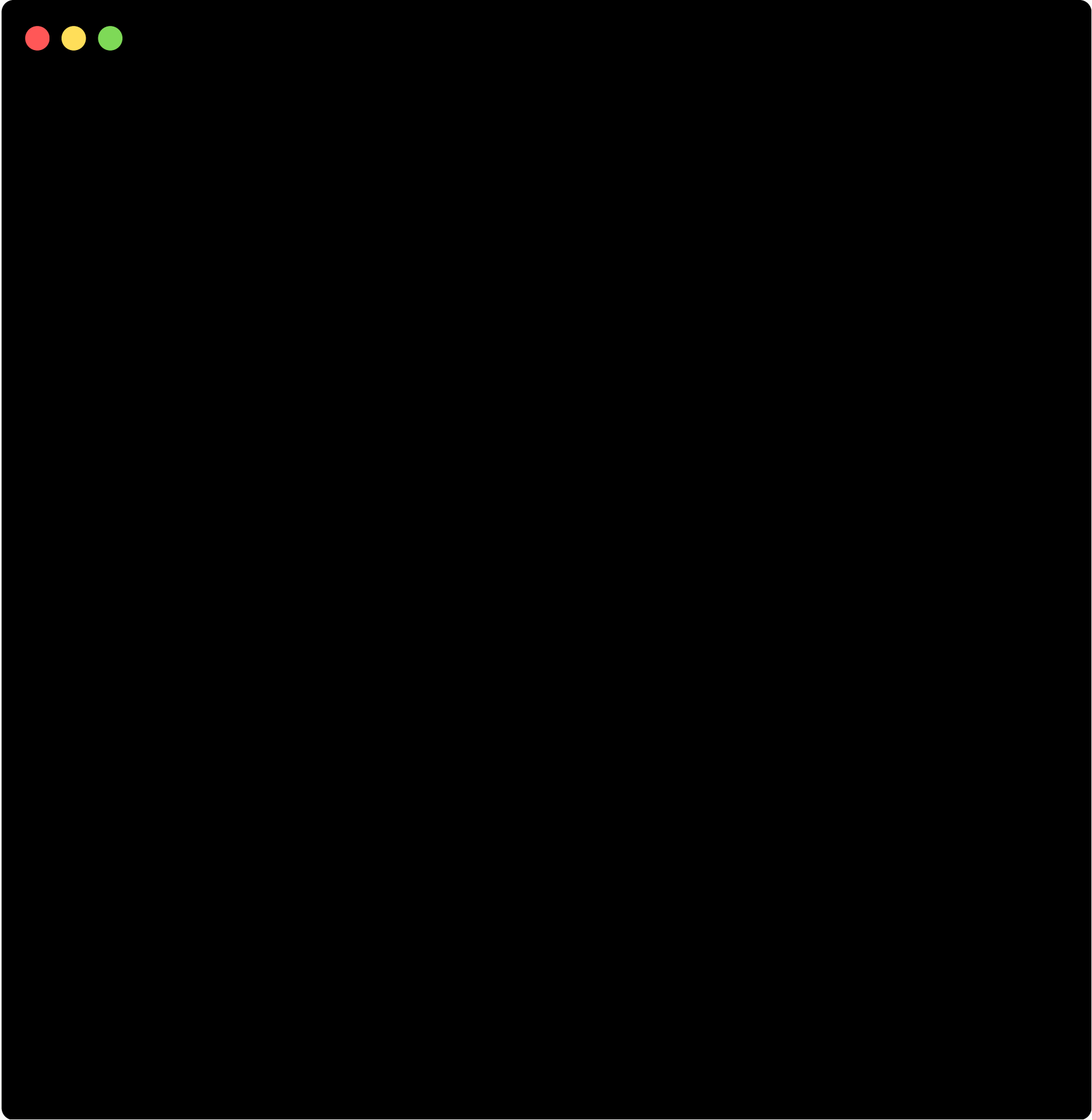
AUTHN API



Baseline Widely available



IMPROVE THE EXISING:



IMPROVE THE EXISING:

WEB INTL API



Baseline Widely available

```
const options = {
  dateStyle: "full",
  timeStyle: "long",
  timeZone: "Australia/Sydney"
});

new Intl.DateTimeFormat("en-US", options)
  .format(date);
// Friday, December 2, 2022 at 12:21:40 PM
// GMT+11

// Relative time
const fmt = new Intl.RelativeTimeFormat(
  "en",
  { style: "narrow" }
);
fmt.format(3, "day"); // in 3 days
fmt.format(-2, "year"); // 2 years ago
```

IMPROVE THE EXISING:

WEB INTL API



Baseline Widely available



```
const au = new Intl.NumberFormat("en-AU");  
au.format(123_456.79);  
// 123,456.79
```

```
const de = new Intl.NumberFormat("de-DE");  
de.format(123_456.79);  
// 123.456,79
```

```
const fmt = new Intl.NumberFormat(  
  "de-DE",  
  { style: "currency", currency: "EUR" }  
);  
fmt.format(123_456.79)  
// 123.456,79 €
```

IMPROVE THE EXISING:

TEMPORAL API



Limited availability

```
// Get exact current system time
Temporal.Now.instant();

// Get time zone
Temporal.Now.timeZoneId();

// Useful utilities
const date = Temporal.PlainDate.from(dateStr);
// *.year
// *.inLeapYear
// *.toString()
// ...

// Manipulate time
date.add({ hours: 1 });
```

IMPROVE THE EXISING:

NAVIGATION API



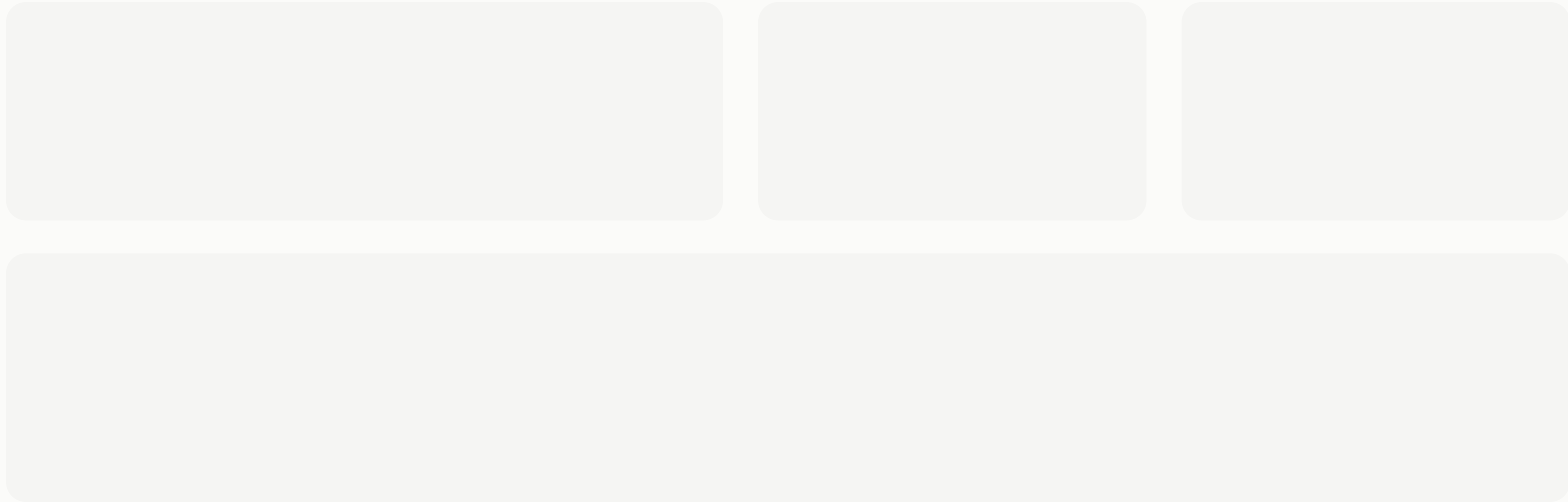
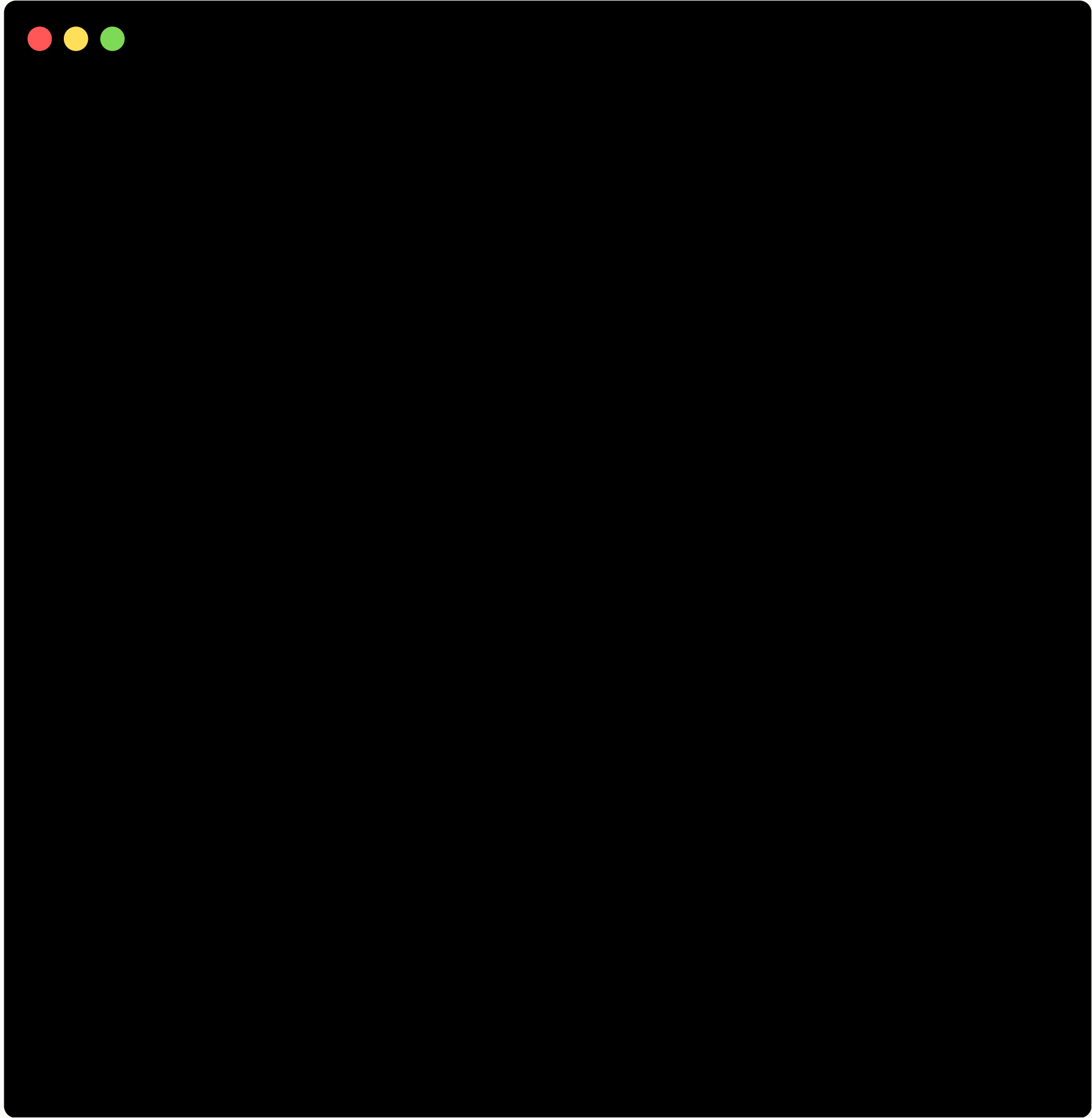
Baseline Newly available

```
// Promises for the win (finally!)
await navigation.reload({ info, state });

// Navigate around
await navigation.navigate(url, options);
await navigation.back(options);
await navigation.forward(options);
await navigation.traverseTo(key, options);

// Events
navigation
  .addEventListener("currententrychange", ...)
  .addEventListener("navigate", ...)
  .addEventListener("navigatesuccess", ...)
  .addEventListener("navigateerror", ...);
```

MANDATORY MENTION OF AI:

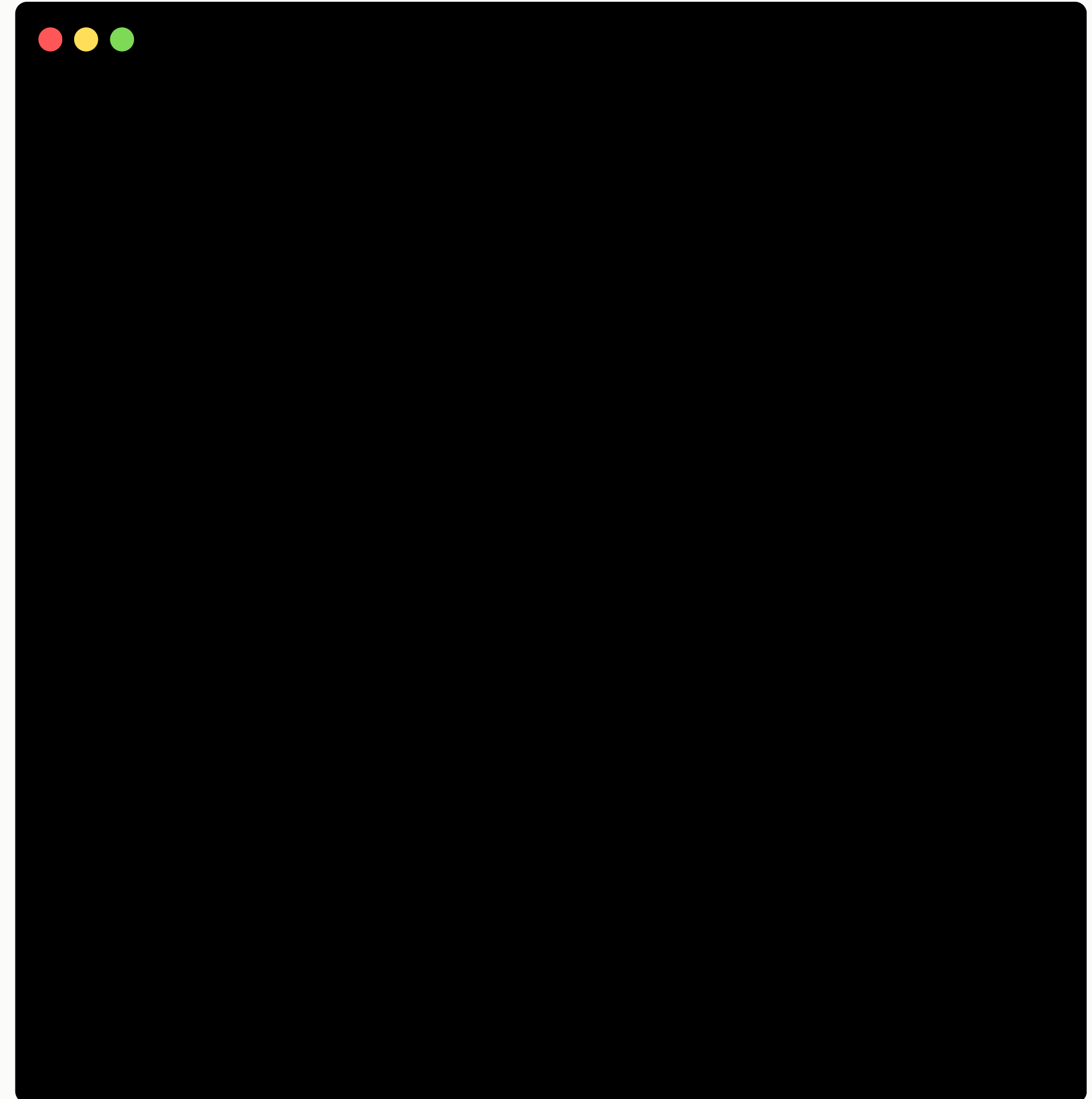


MANDATORY MENTION OF AI:

AI, BUT BUILT-IN?



No availability

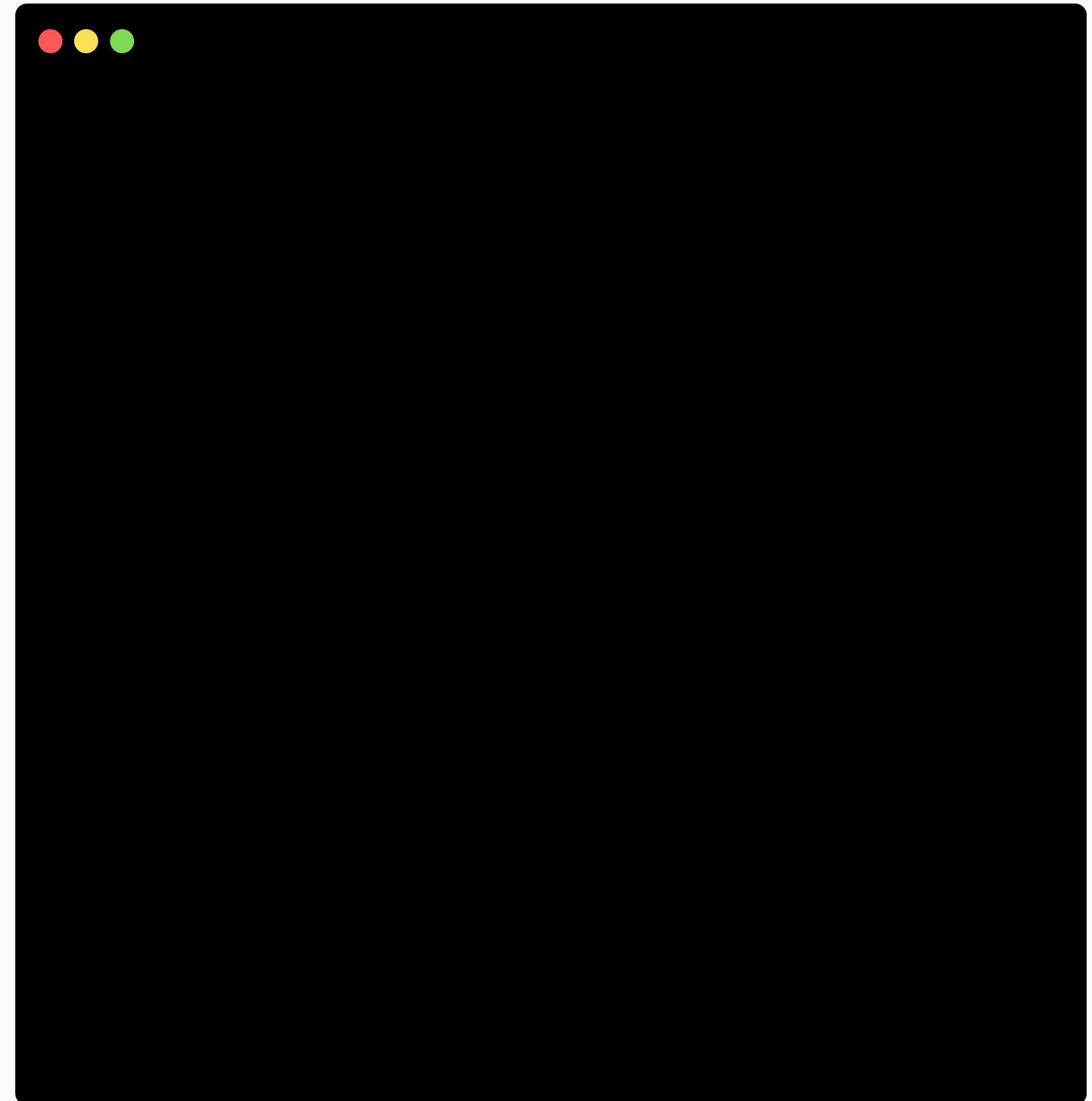


MANDATORY MENTION OF AI:

AI, BUT BUILT-IN?



No availability



MANDATORY MENTION OF AI:

AI, BUT BUILT-IN?



No availability



```
// Check availability status
await window.LanguageModel.availability();

// Create a session and use the AI features
const session = await window
  .LanguageModel
  .create();

const result = await session.prompt("...");
```

MANDATORY MENTION OF AI:

AI, BUT BUILT-IN?



No availability

```
// goo.gle/chrome-ai-dev-preview-index

// Prompt API
await window.LanguageModel.create();

// Writer and Rewriter API
await window.Writer.create();
await window.Rewriter.create();

// Summarization API
await window.Summarizer.create();

// Language Detection API
await window.LanguageDetector.create();

// Translation API
await window.Translator.create();
```

MANDATORY MENTION OF AI:

WEBNN



No availability *

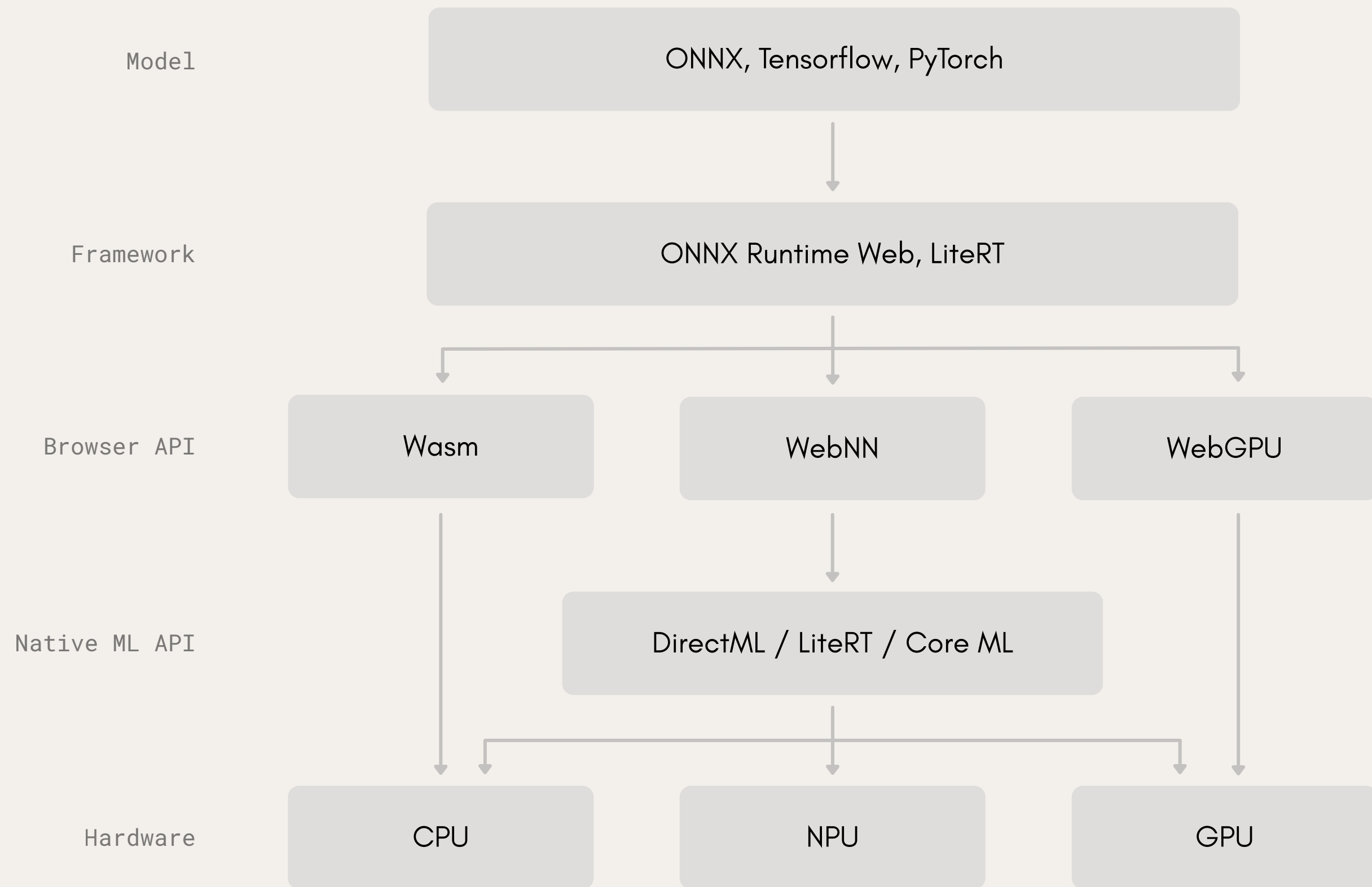
still a TC39 draft

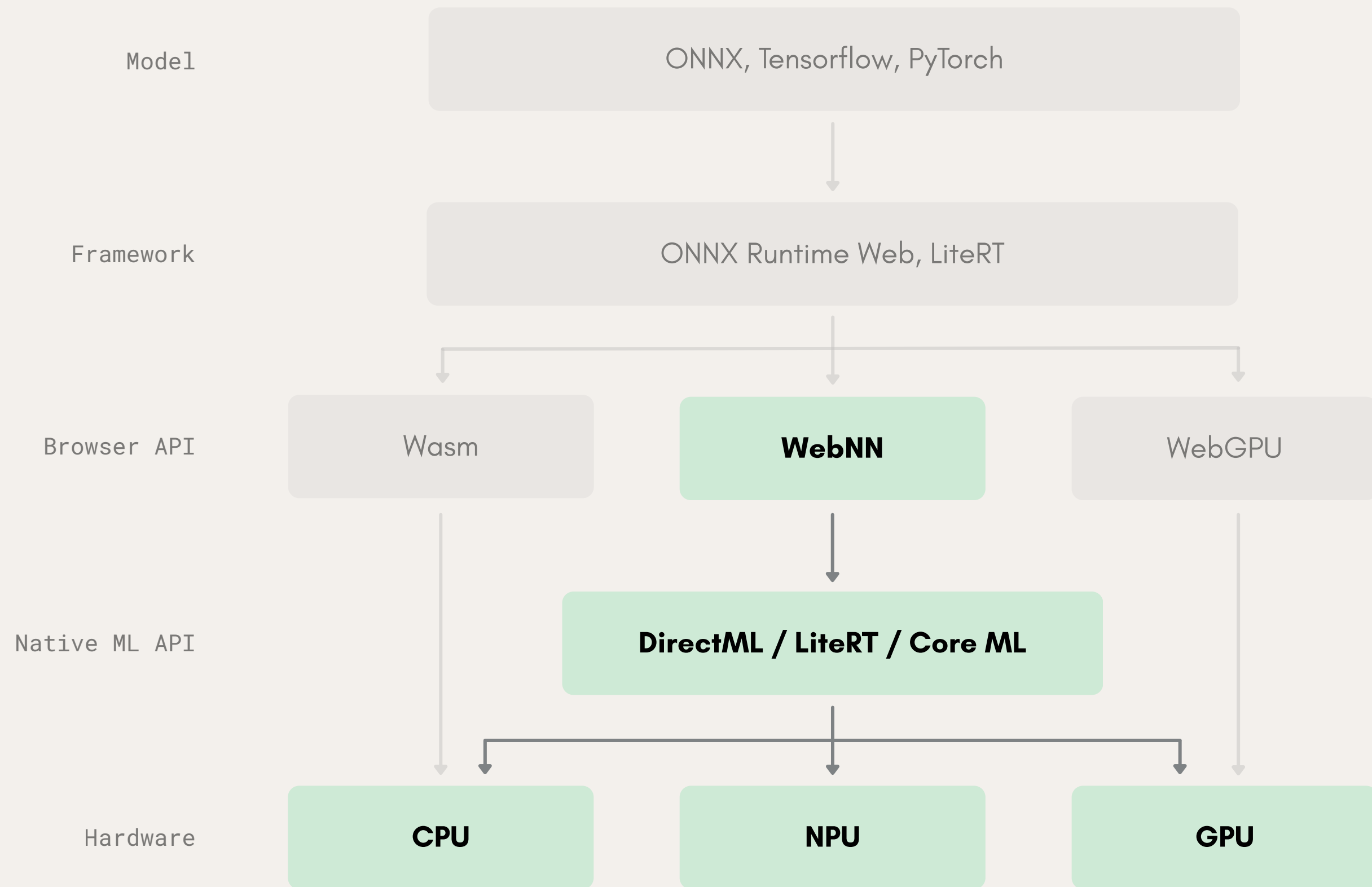
```
const descriptor = {
  dataType: "float32",
  shape: [2, 2]
};
const ctx = await navigator.ml.createContext();
const builder = new MLGraphBuilder(ctx);

// 1. Create a computational graph
builder.constant(...)
builder.input(...)
builder.add(...)

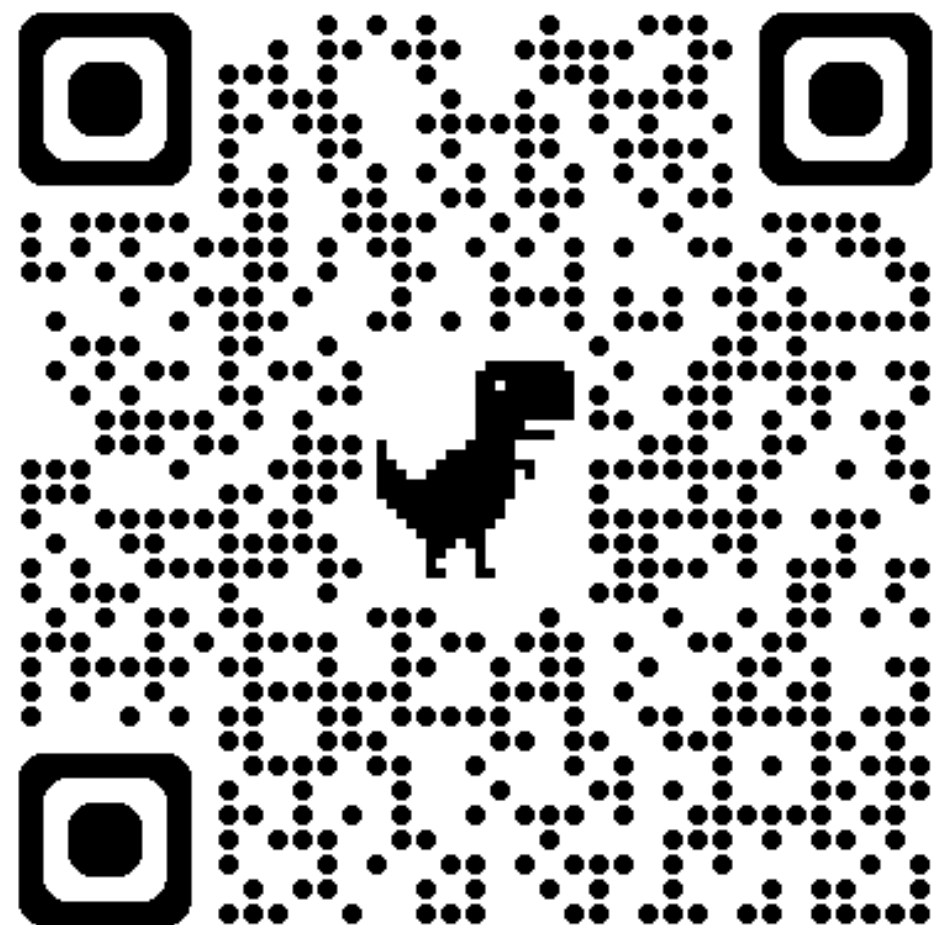
// 2. Compile the graph
builder.build(...)

// 3. Create reusable input and output tensors
// 4. Initialize the inputs
// 5. Execute the graph & read results
ctx.dispatch(graph, inputs, outputs);
ctx.readTensor(...);
```





**STAY
CURIOUS!**



Link to the slides:

**[https://www.julianburr.de/
devbcn-web-2026-slides.pdf](https://www.julianburr.de/devbcn-web-2026-slides.pdf)**

<https://github.com/julianburr/stopme>

<https://www.linkedin.com/in/julianburr/>

<https://bsky.app/profile/julianburr.de>

