



Service
Worker

\$WHOAMI

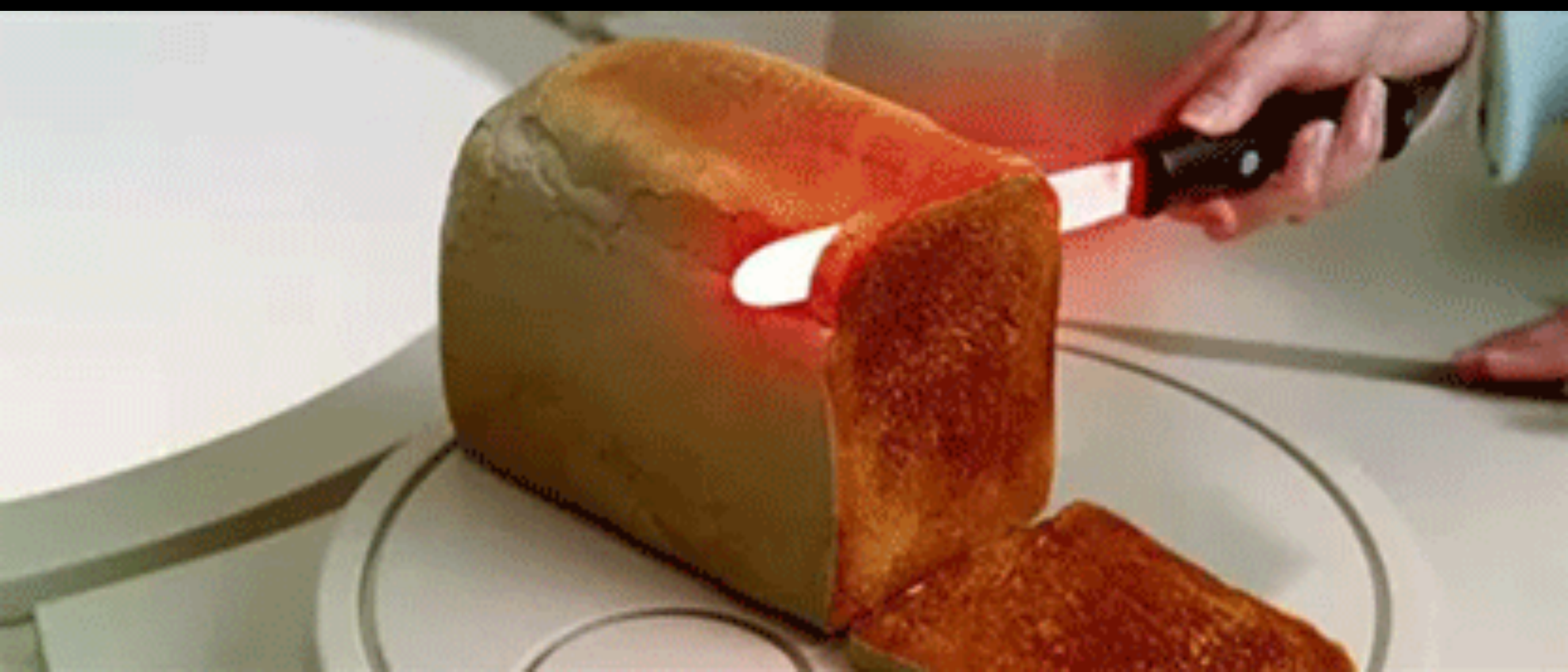


I am Pascal Helfenstein

 nicam

L//P

What is a Service
Worker?

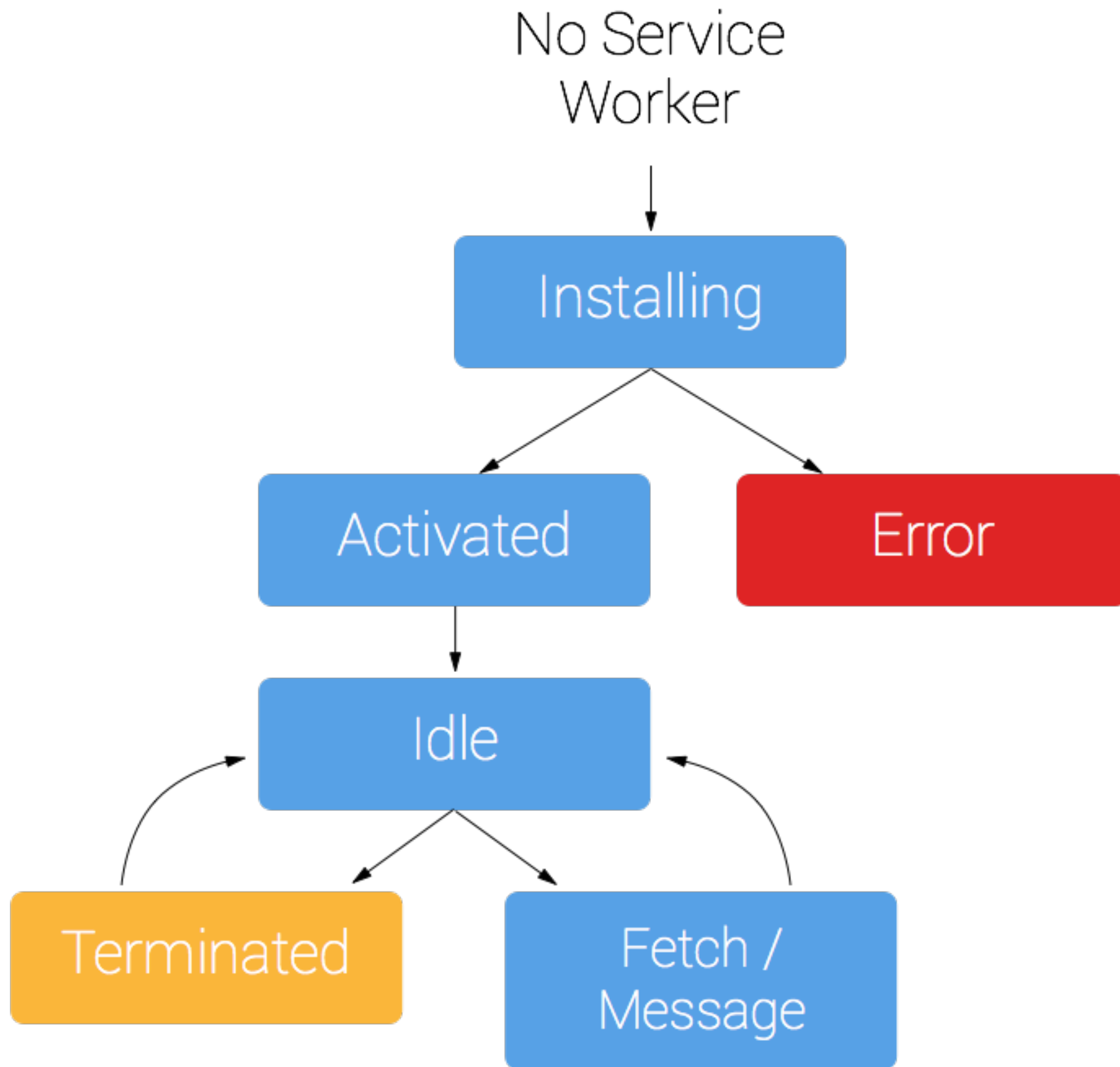


What is it?

- javascript runtime in a separate thread
- controls multiple pages
- interception and modification of requests & cache
- can run without open tab
- is activated by the browser whenever necessary

Prerequisite

- service-worker & registrator on same origin
- **https** only (except localhost)
- doesn't have DOM access
- downloaded every 24 hours (on request)
- lot's of promises (.then().catch())



caveats

- first run doesn't have a worker*
- worker gets switched once all tabs with the old worker have been closed
- you have to write cache upgrade / clear scripts
- can be terminated at any time



index.js

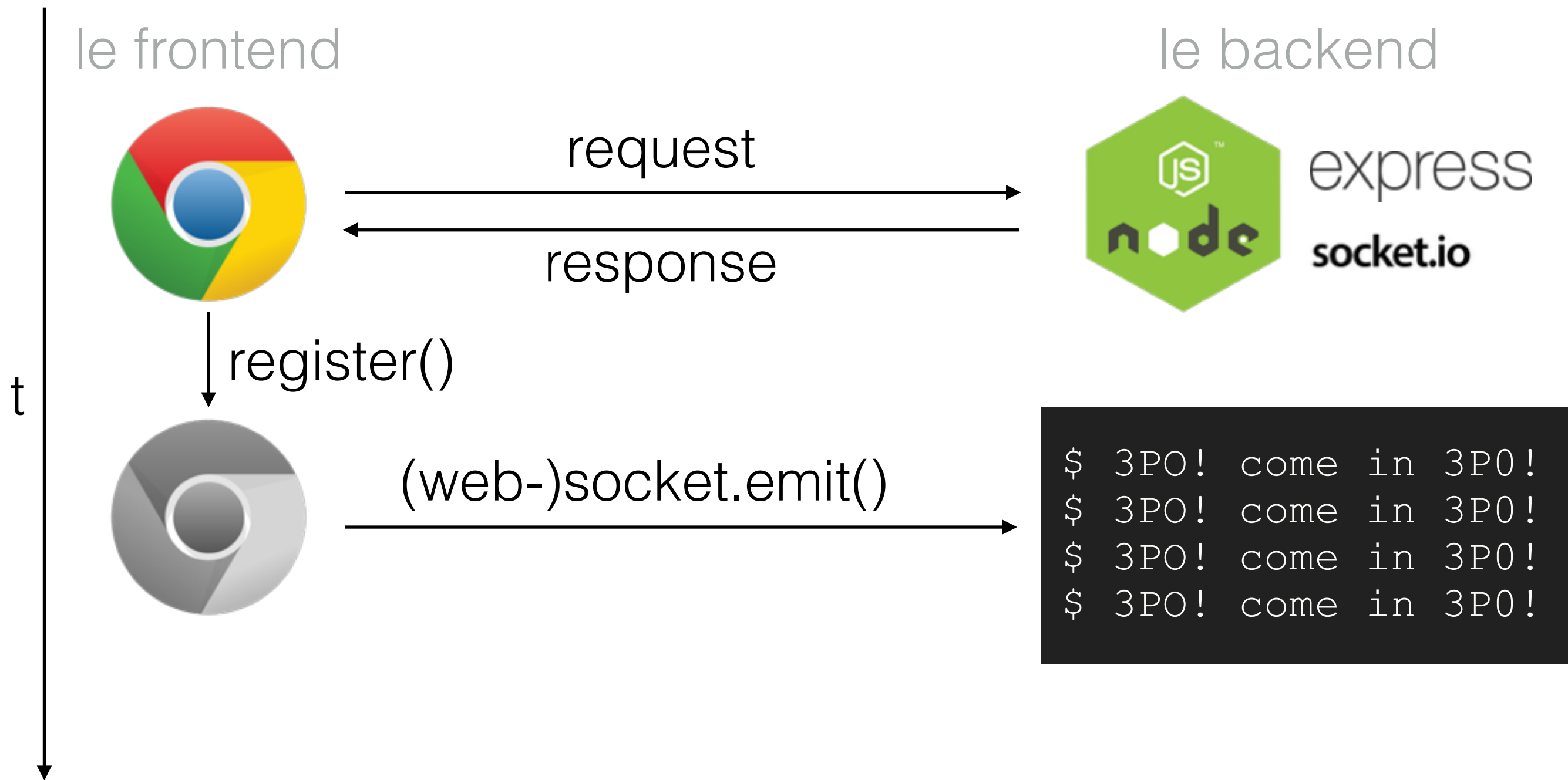
```
if ('serviceWorker' in navigator) {  
  navigator.serviceWorker.register('/sw.js', {  
    scope: '/'  
  }).then(function(reg) {  
    console.log('Yey!', reg);  
  }).catch(function(err) {  
    console.log('Boo!', err);  
  });  
}
```



basic sw.js

```
self.addEventListener('install', function(event) {  
  // Do cache your stuff!!!  
});  
  
self.addEventListener('activate', function(event) {  
  // Clean up old things!!!  
});
```

demo #1 setup



Demo 1





SW.js

```
this.addEventListener('install', function(event) {  
  event.waitUntil(self.skipWaiting());  
});  
  
this.addEventListener("activate", function(event) {  
  event.waitUntil(self.clients.claim().then(function () {  
    importScripts('/socket.io/socket.io.js');  
    var socket = io.connect('http://localhost:3000', {  
      jsonp: false  
    });  
  
    setInterval(function () {  
      socket.emit('timer', "3P0!, come in 3P0!");  
    }, 3000);  
  }));  
});
```

gotchas

- the service worker won't be running all the time
- chrome will spin it up when necessary

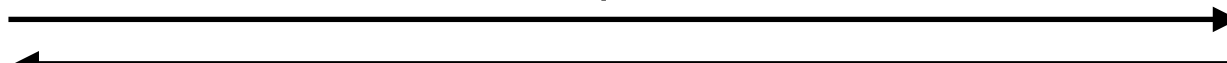
demo #2 setup

le frontend

le backend



request



response



express

register() && activate()



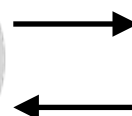
index.html



index.html



index.html



t



Demo 2





cache stuff up front

```
this.addEventListener('install', function(event) {  
  event.waitUntil(  
    caches.open('v1').then(function(cache) {  
      return cache.addAll([  
        '/',  
        '/index.html',  
        '/images/offline.jpg',  
        '/index.js'  
      ]);  
    })  
  );  
});
```



intercept requests

```
this.addEventListener('fetch', function (event) {  
  return event.respondWith(caches.match(event.request)  
    .then(function(response) {  
      return response || fetch(event.request);  
    }) .then(function(response) {  
      caches.open('v1').then(function(cache) {  
        cache.put(event.request, response);  
      });  
      return response.clone();  
    }) .catch(function() {  
      return caches.match('/images/offline.jpg');  
    }));  
});
```

gotchas

- Cleanup your old caches during activation
- remember all the possible cache versions a client can have
- split page cache and asset cache
- have different fallbacks for different types of files
- fetch doesn't send cookies by default

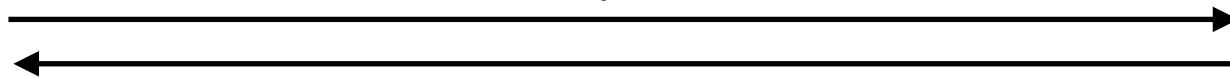
demo #3 setup

le frontend

le backend



request



response



express

register() && activate() && subscribe()



t



push



push



```
$ curl http://GCM
```

spinUp() && receiveMessage()



Demo 3





index.js

```
navigator.serviceWorker.register('/sw.js', { scope: '/' })
  .then(function(reg) {})
  .catch(function(error) {
    console.log('Registration failed with ' + error);
  });
navigator.serviceWorker.ready.then(function (reg) {
  reg.pushManager.subscribe({userVisibleOnly: true})
    .then(function(sub) {
      console.log('endpoint:', sub.endpoint);
    });
});
```



SW.js

```
this.addEventListener('push', function(event) {  
  console.log('Push message received', event);  
  var title = 'The empire needs you!';  
  event.waitUntil(  
    self.registration.showNotification(title, {  
      body: 'Join the dark side',  
      icon: 'images/trooper.jpg',  
      tag: 'my-tag'  
    }));  
});
```

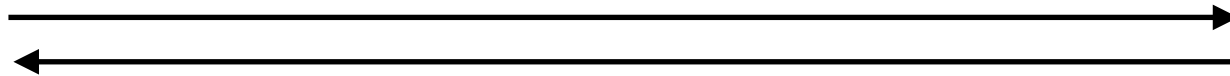
demo #4 setup

le frontend

le backend



request



response



express



register()



t



postMessage()



postMessage()



Demo 4



index.js

```
var registration;
navigator.serviceWorker.register('/sw.js',
{ scope: '/' }).then(function(reg) {
  navigator.serviceWorker.ready.then(function(reg) {
    registration = reg;
  });
});

navigator.serviceWorker.onmessage = function(event) {
  console.log("Got reply", event.data);
};

function sendMessageToWorker() {
  registration.active.postMessage({
    text: "Hi!"
  });
}
```




SW.js

```
this.addEventListener('message', function (event) {  
  console.log("event.data"); // Here's your data  
  event.source.postMessage("Woop! (via source)");  
  // OR  
  clients.matchAll().then(function(clients) {  
    for (var client of clients) {  
      client.postMessage("Whoop! (via client api)");  
    }  
  });  
});
```

Browser Support



The Future

- Background Sync
 - schedule a worker to spin up periodically
- Geofencing
 - trigger a worker at certain locations

Questions?





Thank you!

liip.to/serviceworker