

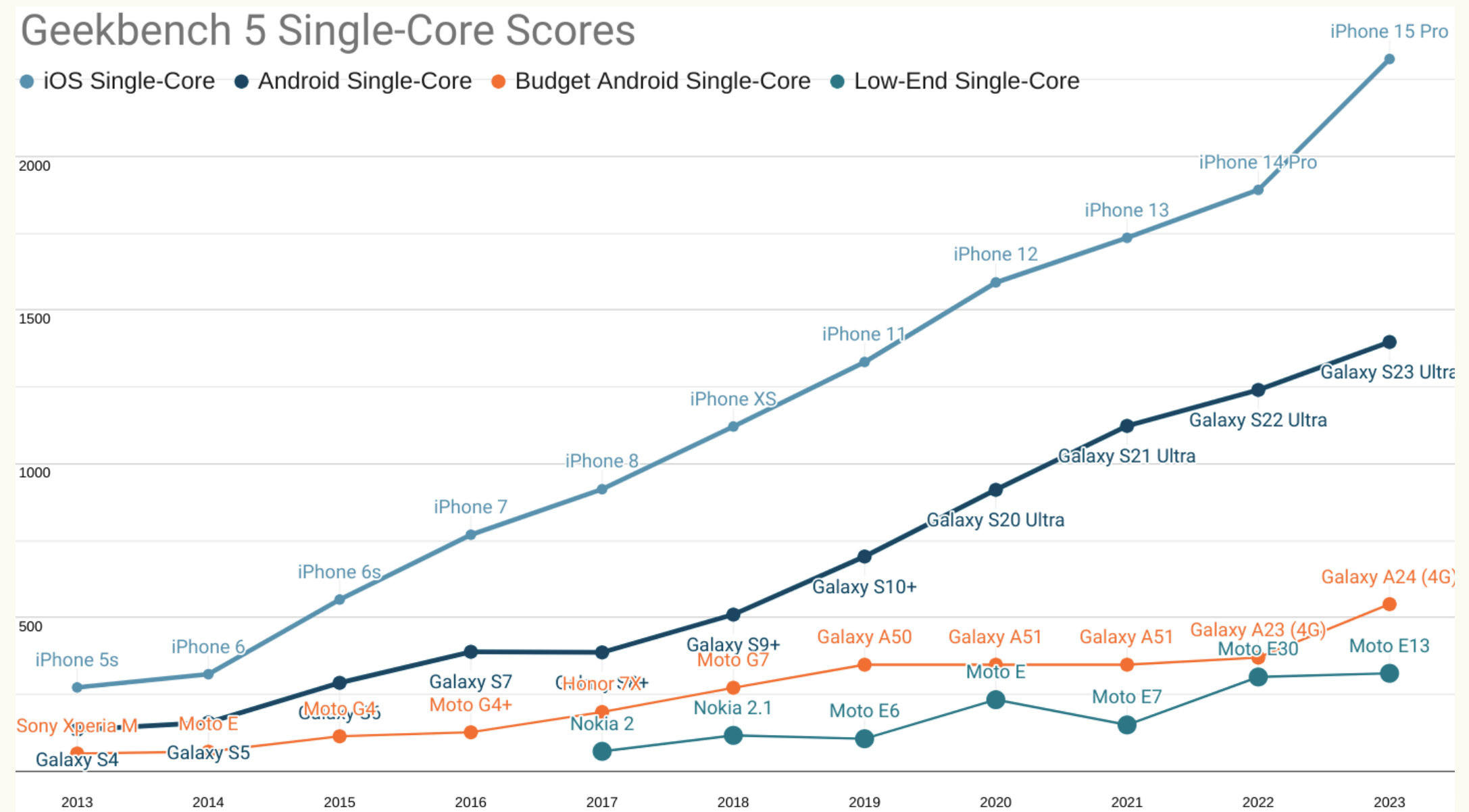
local / offline First

minimalisme radical

Web mobile

Ressources

Autonomie batterie

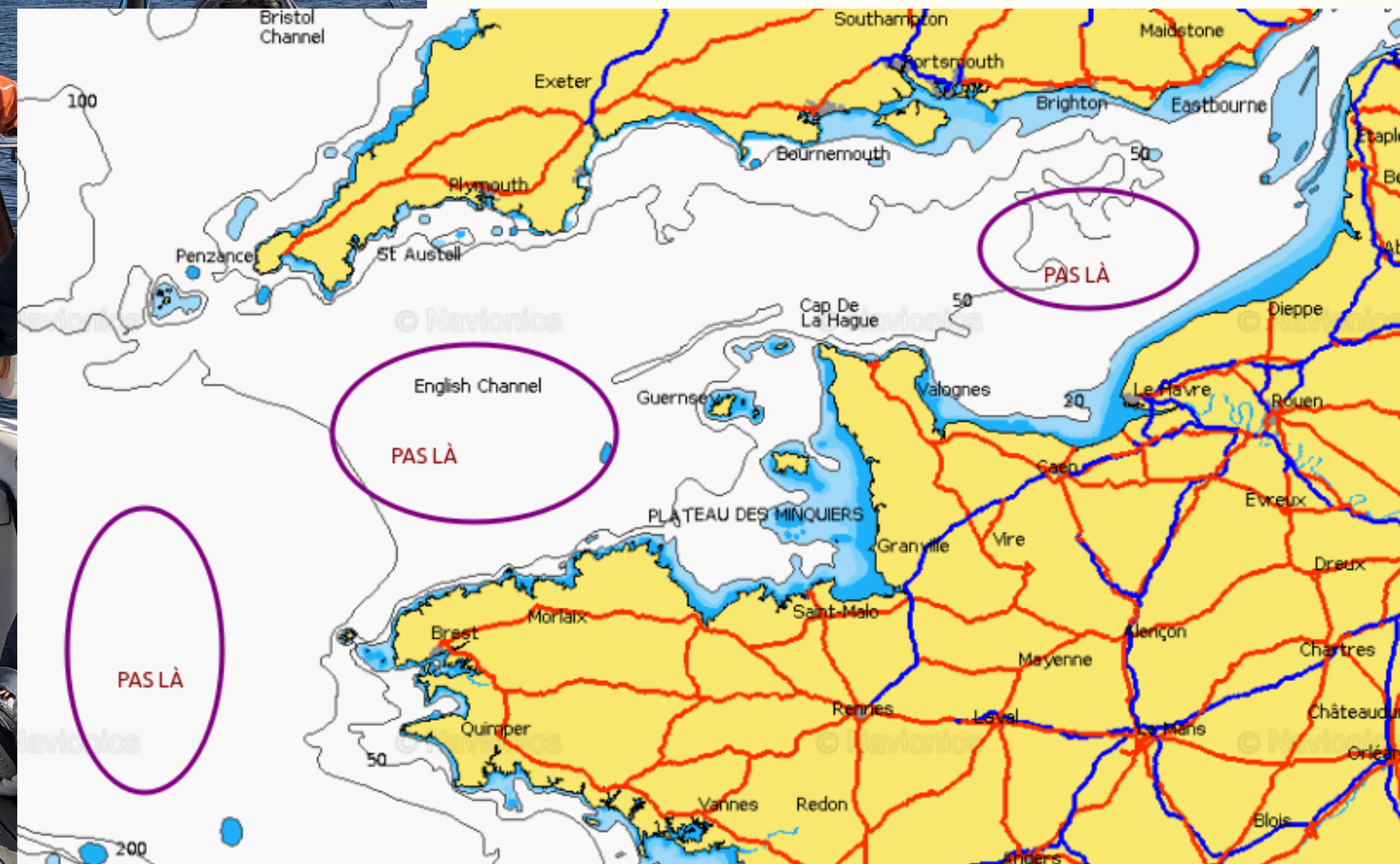


infrequently.org/2024/01/performance-inequality-gap-2024/#mobile

Mon lab R&D

Panneau solaire 100W

Réseau = pas partout



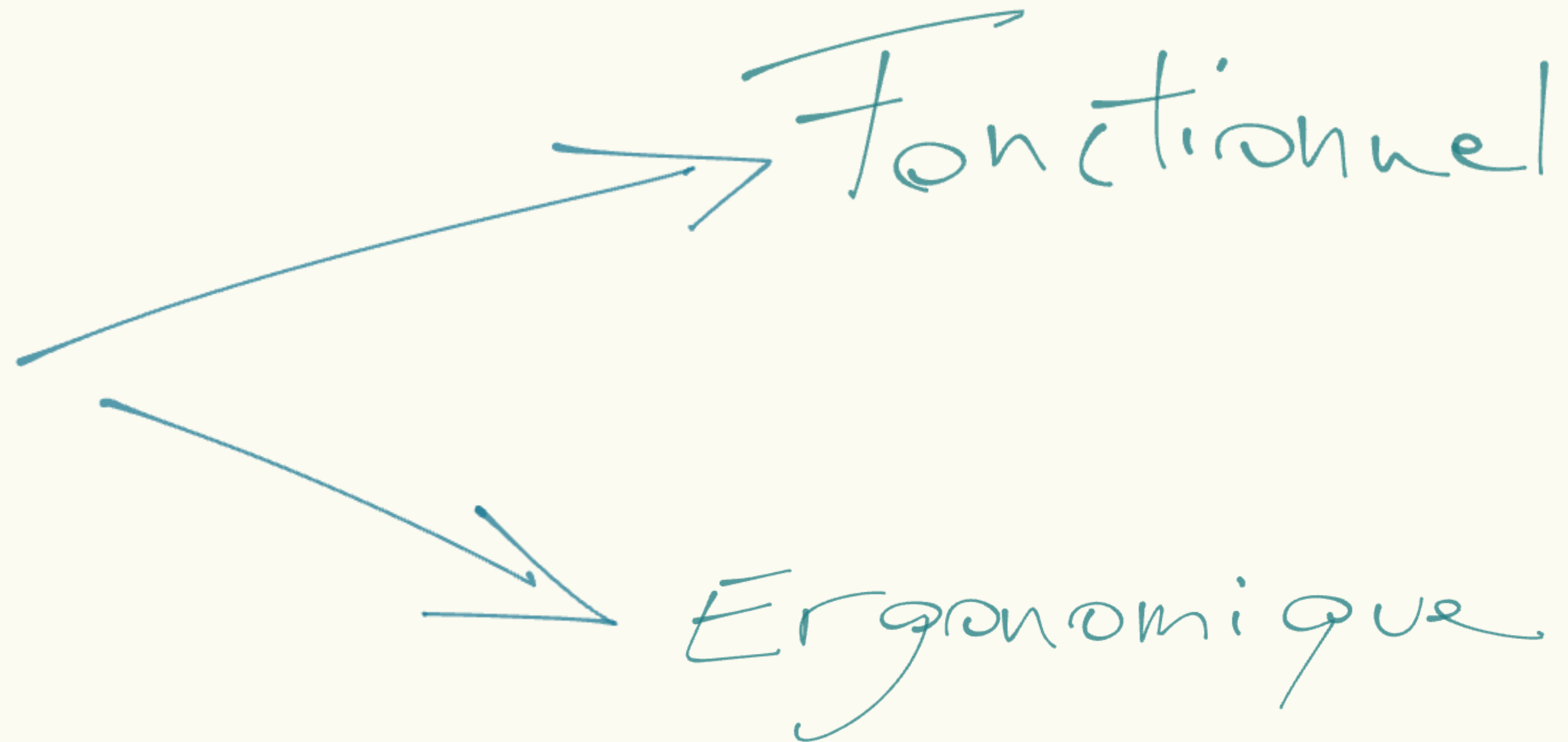
Limiter la complexité

API WEB Natives

Les projets sur lesquels je travaille

Outils métiers

Greenfield



No build

No build
No framework

No build
No framework
No dependencies

No build

presque

No framework

No dependencies

No build

presque

No framework

No dependencies

limitées à 2 🙄

HTML first

HTML first

Déclaratif

HTML first

Déclaratif

Décrit le métier

HTML first

Déclaratif

Décrit le métier / UX inclus

HTML first

Déclaratif

Décrit le métier / UX inclus
“The UX of HTML”

*a well considered user experience starts with
well considere HTML*

vasilis.nl/nerd/the-ux-of-html/

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <nav><a href="/about">about</a><a href="/services">services</a></nav>
  <main>
    <header>
      <h1>Nos services</h1>
    </header>
    <div>
      <form>
        <label>Email
          <input type="email" name="email">
        </label>
        <!-- 100 x input -->
      </form>
    </div>
    <table>
      <tr>
        <th>col1</th> <th>col2</th> <th>col3</th> <th>col4</th> <th>col5</th>
      </tr>
      <tr>
        <td>1111</td> <td>stuff</td> <td>other stuff</td> <td>123456.23</td> <td>2024-05-025</td>
      </tr>
      <!-- 1000 x tr -->
    </table>
  </main>
</body>
</html>
```

← Navigateur optimisé pour parser et afficher du HTML

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <nav><a href="/about">about</a><a href="/services">services</a></nav>
  <main>
    <header>
      <h1>Nos services</h1>
    </header>
    <div>
      <form>
        <label>Email
          <input type="email" name="email">
        </label>
        <!-- 100 x input -->
      </form>
    </div>
    <table>
      <tr>
        <th>col1</th> <th>col2</th> <th>col3</th> <th>col4</th> <th>col5</th>
      </tr>
      <tr>
        <td>1111</td> <td>stuff</td> <td>other stuff</td> <td>123456.23</td> <td>2024-05-025</td>
      </tr>
      <!-- 1000 x tr -->
    </table>
  </main>
</body>
</html>
```

← Navigateur optimisé pour parser et afficher du HTML

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <nav><a href="/about">about</a><a href="/services">services</a></nav>
  <main>
    <header>
      <h1>Nos services</h1>
    </header>
    <div>
      <form>
        <label>Email
          <input type="email" name="email">
        </label>
        <!-- 100 x input -->
      </form>
    </div>
    <table>
      <tr>
        <th>col1</th> <th>col2</th> <th>col3</th> <th>col4</th> <th>col5</th>
      </tr>
      <tr>
        <td>1111</td> <td>stuff</td> <td>other stuff</td> <td>123456.23</td> <td>2024-05-025</td>
      </tr>
      <!-- 1000 x tr -->
    </table>
  </main>
</body>
</html>
```


← Navigateur optimisé pour parser et afficher du HTML

```
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <nav><a href="/about">about</a><a href="/services">services</a></nav>
  <main>
    <header>
      <h1>Nos services</h1>
    </header>
    <div>
      <form>
        <label>Email
          <input type="email" name="email">
        </label>
        <!-- 100 x input -->
      </form>
    </div>
    <table>
      <tr>
        <th>col1</th> <th>col2</th> <th>col3</th> <th>col4</th> <th>col5</th>
      </tr>
      <tr>
        <td>1111</td> <td>stuff</td> <td>other stuff</td> <td>123456.23</td> <td>2024-05-025</td>
      </tr>
      <!-- 1000 x tr -->
    </table>
  </main>
</body>
</html>
```



SPA ?

MPA ?

Service Workers !

Service Workers !

Navigation locale avec

`<a href=""></a>`

Pages pré-générées server side

killer combo

killer combo

pages statiques

```
// config/routes.yaml
app_mypage:
  path: /my-page
  controller: Symfony\Bundle\FrameworkBundle\Controller\TemplateController
  defaults:
    template: 'pages/my-page.html.twig'
    context:
      title: 'My page'
```

killer combo

pages statiques

```
// config/routes.yaml
app_mypage:
  path: /my-page
  controller: Symfony\Bundle\FrameworkBundle\Controller\TemplateController
  defaults:
    template: 'pages/my-page.html.twig'
    context:
      title: 'My page'
```

Workbox developer.chrome.com/docs/workbox/modules

← dependance n°1

killer combo

pages statiques

```
// config/routes.yaml
app_mypage:
  path: /my-page
  controller: Symfony\Bundle\FrameworkBundle\Controller\TemplateController
  defaults:
    template: 'pages/my-page.html.twig'
    context:
      title: 'My page'
```

Workbox developer.chrome.com/docs/workbox/modules

← dependance n°1

```
// config/workbox-dev-config.js
"templatedURLs": {
  '/': [
    '../templates/pages/my-page.html.twig',
    '../templates/app_base.html.twig',
  ],
  ...
}
```

killer combo

```
// public/service-worker.js
importScripts('js/vendor/workbox-v6.5.1/workbox-sw.js');
const {
  strategies,
  routing,
  precaching
} = workbox;

workbox.precaching.precacheAndRoute(self.__WB_MANIFEST, {
  // Ignore all URL parameters.
  ignoreURLParametersMatching: [/.*?/],
})
```


killer combo

```
// public/service-worker.js
importScripts('js/vendor/workbox-v6.5.1/workbox-sw.js');
const {
  strategies,
  routing,
  precaching
} = workbox;

workbox.precaching.precacheAndRoute(self.__WB_MANIFEST, {
  // Ignore all URL parameters.
  ignoreURLParametersMatching: [/.*?/],
})
```

```
$ workbox injectManifest config/workbox-dev-config.js ← build
```

killer combo

```
// public/service-worker.js
importScripts('js/vendor/workbox-v6.5.1/workbox-sw.js');
const {
  strategies,
  routing,
  precaching
} = workbox;

workbox.precaching.precacheAndRoute(self.__WB_MANIFEST, {
  // Ignore all URL parameters.
  ignoreURLParametersMatching: [/.*?/],
})
```

```
$ workbox injectManifest config/workbox-dev-config.js ← build
```

```
// public/sw.js
workbox.precaching.precacheAndRoute([
  {"revision": "d1f4feaf44d6062fc360e2115fb4fc5e", "url": "/my-page"},
  // ...
], {
  ignoreURLParametersMatching: [/.*?/],
})
```

Offline first = Mini backend pour 1 personne

CRUD + Sync



IndexedDB → github.com/jakearchibald/idb

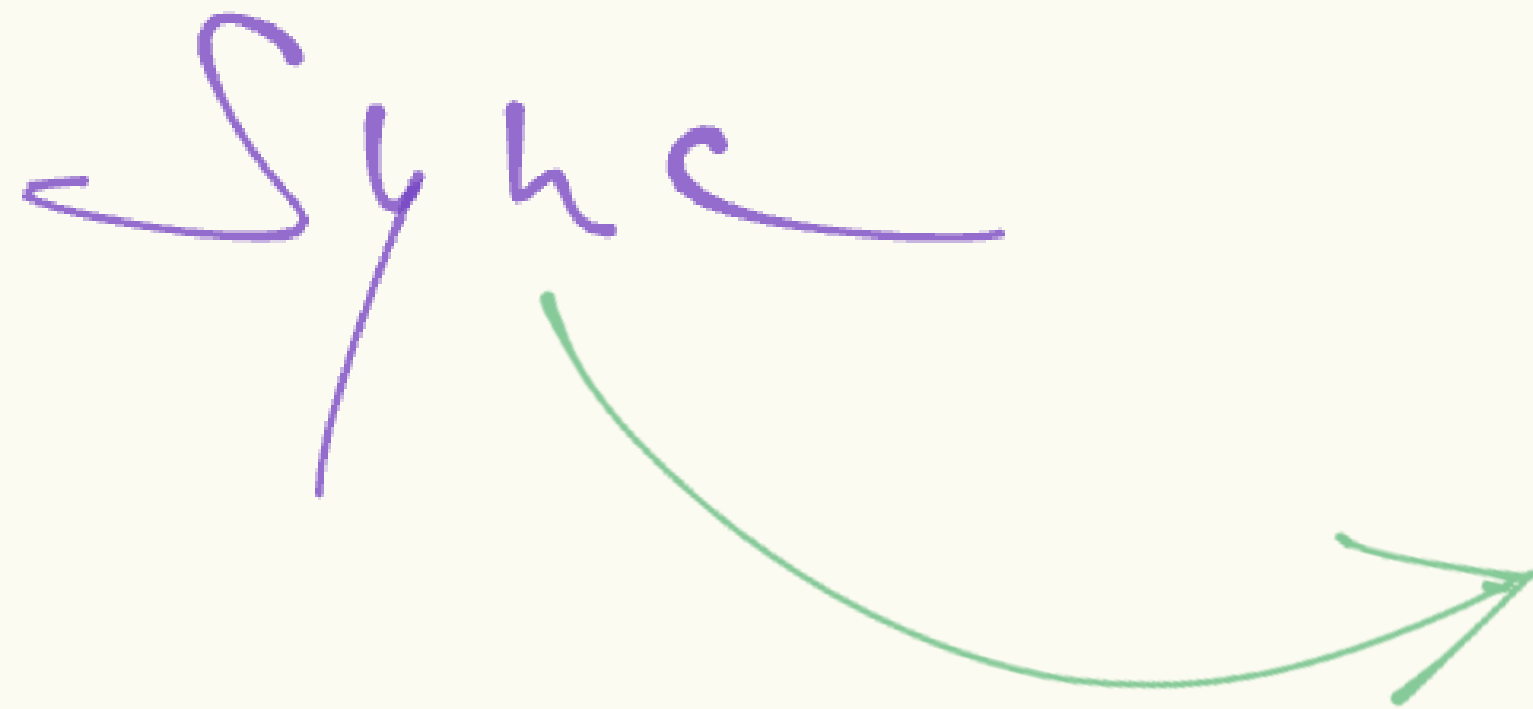
↑ dépendance n°2

Sync

1. Export CSV
2. local → distant (one way) (json, binary)
3. json-p
4. intent based → Replicache, Kinto
5. CRDT / OT

SW Background-sync (pas toujours fiable)

Sync



- check existence sur le serveur avant sync
- Envoi de tous les fichiers dans le même cycle → Promise.allSettled
- formData.append('aPhoto', File, 'my-photo.jpg') SW/iOS ❌

CRUD ⇒ Autosave

JS : langage événementiel

`<input />` → change

`<button>clickme</button>`
→ click

data- attribute pour gérer les events

`<input data-change=changeThing />`

`<button data-click=doStuff>`

```
const changeHandler = {
  changeThing: (event) => {
    // stuff
  },
  handleEvent: (event) => {
    let handler = event.target.dataset[event.type]

    if (!handler || !this[handler]) {
      return;
    }

    this[handler](event);
  }
}
element.addEventListener('change', changeHandler)
```

Light DOM custom element

meyerweb.com/eric/thoughts/2023/11/01/blinded-by-the-light-dom

```
<import-photo display-elt="photos-display">
  <label class="dropimage photo">
    <input type="file" name="photos" accept="image/*" multiple>
  </label>
</import-photo>
```

```
export default class importPhoto extends HTMLElement {
  connectedCallback() {
    this.displayElt =
      document.getElementById(
        this.getAttribute('display-elt'))

    this.fileInput =
      document.querySelector(
        'input[name=photos]')

    this.addEventListener('change', this)
  }

  handleEvent(event) {
    // do stuff
  }
}
```




Arborescence / Organiser son code

- templates pages avec Twig

```
{% extends 'base.html.twig' %}
```
- public/js/pages/myPage.js
 - ➔ orchestre les interactions
- public/js/components/importPhoto.js
 - ➔ Custom Element pour modulariser
- public/js/libs/DBStorage.js ➔ i/o indexedDB
- public/service-worker.js, public/sw/controllers.js

Controller js ➡ ES Module

```
//public/js/pages/myPage.js
// 1. imports
import * as iDBStorage from './libs/iDBStorage.js'
import importPhoto from './components/importPhoto.js'
// 2. cache DOM refs
const elts = {
  myEl: _d.qs('[data-something]'),
  // ...
}
// 3. const, let + 'theThing'
// ...
// 4. Initialize web components
customElements.define('import-photo', importPhoto)
```

```
// 5. Templates function
function entryRowTpl(row) {
  return `
    <tr>
      <td>${row.col1}</td>
      <td>${row.col2}</td>
      <td>${row.col3}</td>
      <td class="entry-actions">
        <button data-click="showActionsMenu">Menu</button>
      </td>
    </tr> `
}
// 6. eventHandlers
// 7. eventListener
// 8. init()
function init() {
  // get theThing from iDBStorage
  // doStuff
}
// 10. local utility functions
```


A close-up, low-angle shot of a sailboat's mast and rigging against a clear blue sky. The mast is white and has various ropes and pulleys attached to it. The ocean is visible in the background, with small waves. The text is overlaid on the right side of the image.

Ça marche

Ça va vite

Les bugs sont les tiens :D



Formation en septembre 2024

Contact

Aleth Gueguen
conseil@alethgueguen.com
+33 (0)675 90 86 06