

Développer avec la webmap et les composants

Pauline Louis, Esri France
plouis@esrifrance.fr



Sommaire

 01

Que configurer dans la webmap ?

 02

Quels avantages pour les développeurs ?
(en + décrire – de code)

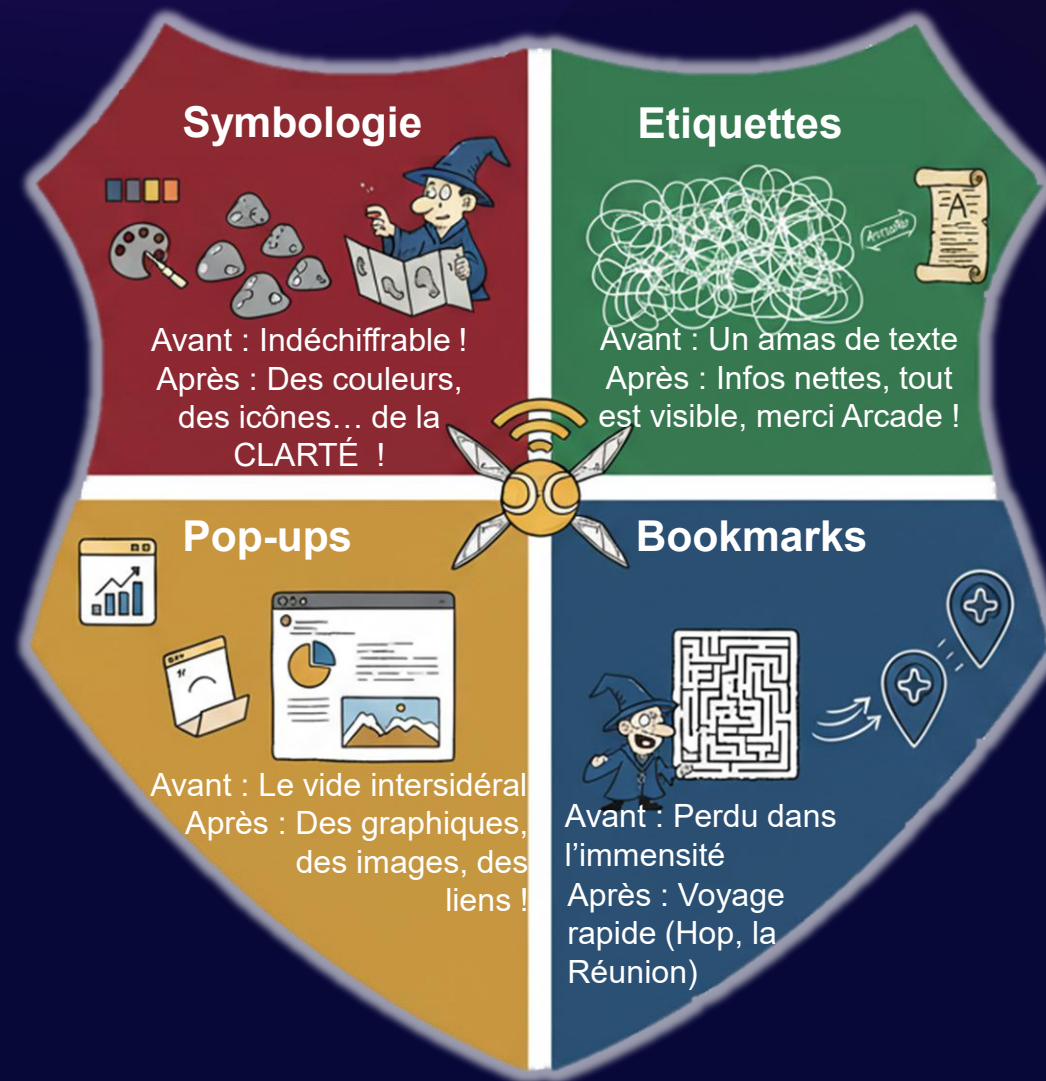
 03

Intégration de la webmap avec les composants
et extension avec le SDK JavaScript d'ArcGIS

The background features a dark blue base with several overlapping, wavy, translucent bands in shades of orange, yellow, and purple. These bands are decorated with a pattern of small, glowing yellow squares. In the lower right, there is a faint, stylized map of a city grid in a lighter blue color.

Que configurer dans la webmap ?

Configuration de la webmap



The background features a dark blue field with several overlapping, wavy bands of color: a wide orange band, a thinner pink band, and a purple band. A faint, stylized city map pattern is visible in the lower right. The text 'Quels avantages pour les développeurs ?' is centered in white.

Quels avantages pour les développeurs ?

Intégration

```
<script type="module" src="https://js.arcgis.com/calcite-components/3.0.2/calcite.esm.js"></script>

<link rel="stylesheet" href="https://js.arcgis.com/4.32/esri/themes/light/main.css" />
<script src="https://js.arcgis.com/4.32/"></script>

<script type="module" src="https://js.arcgis.com/map-components/4.32/arcgis-map-components.esm.js"></script>
</head>

<body>
  <arcgis-map item-id="55496d29650f4e11b53b4c9d724519cc">
    <arcgis-zoom position="top-left"></arcgis-zoom>
    <arcgis-expand expanded position="top-left">
      <arcgis-bookmarks ></arcgis-bookmarks>
    </arcgis-expand>
    <arcgis-legend></arcgis-legend>
  </arcgis-map>
</body>

</html>
```

[illegible]

Moins de code – les composants

ArcGIS Maps SDK for JavaScript / References

Home

Sample Code

References

Showcase

Blogs

Search topics

> core

▼ map-components

Area Measurement 2D

Area Measurement 3D

Map components

The ArcGIS Maps SDK for JavaScript's Map components are a collection of pre-built UI components for building web mapping applications with minimal code. This package contains the [Map component](#) (2D), the [Scene component](#) (3D), and other components that encapsulate functionality from the [core API](#) into UI.

On this page

Get started

CDN

npm

```
require(["esri/Map", "esri/views/MapView"], (Map, MapView) => {
  const map = new Map({
    basemap: "streets-vector"
  });

  const view = new MapView({
    container: "viewDiv",
    map: map,
    zoom: 14,
    center: [8.5, 47.37]
  });
});
```

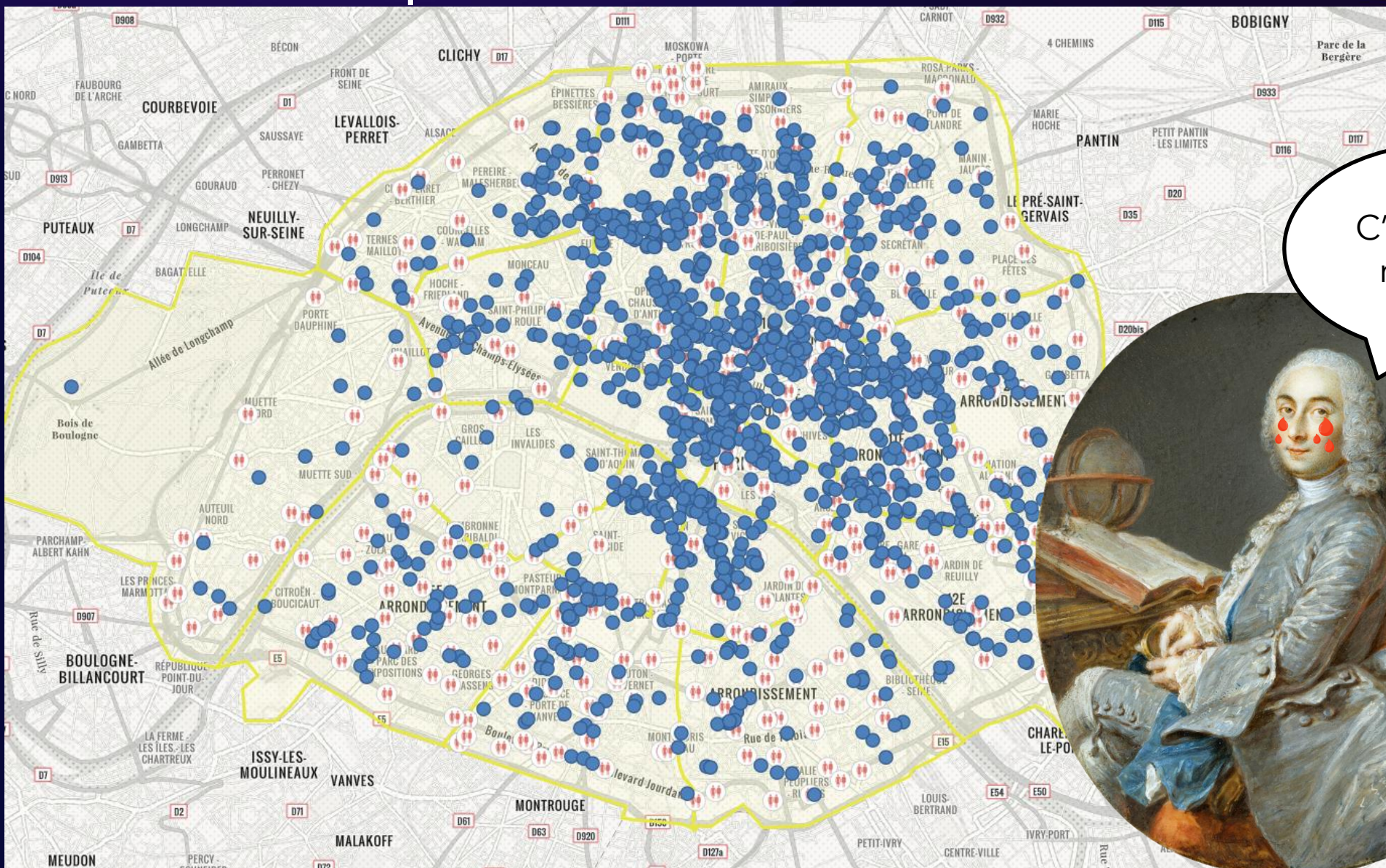
b46cb37eca"></arcgis-map>

<arcgis-map zoom="14" center="8.5,47.37" basemap="streets-vector"></arcgis-map>

Was this page helpful?

👍 Yes 👎 No

Amélioration de la répartition du travail

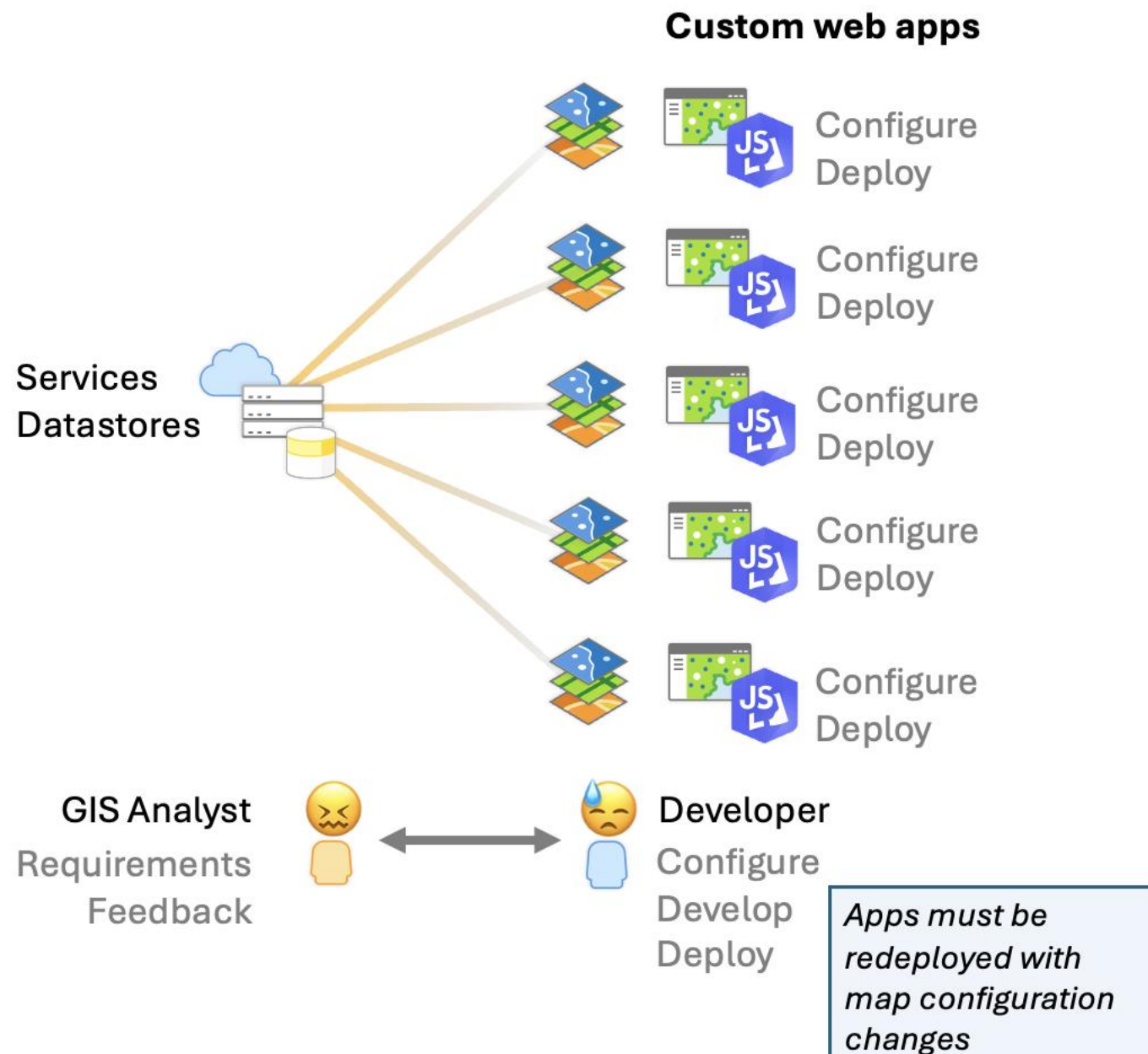


C'est super moche !



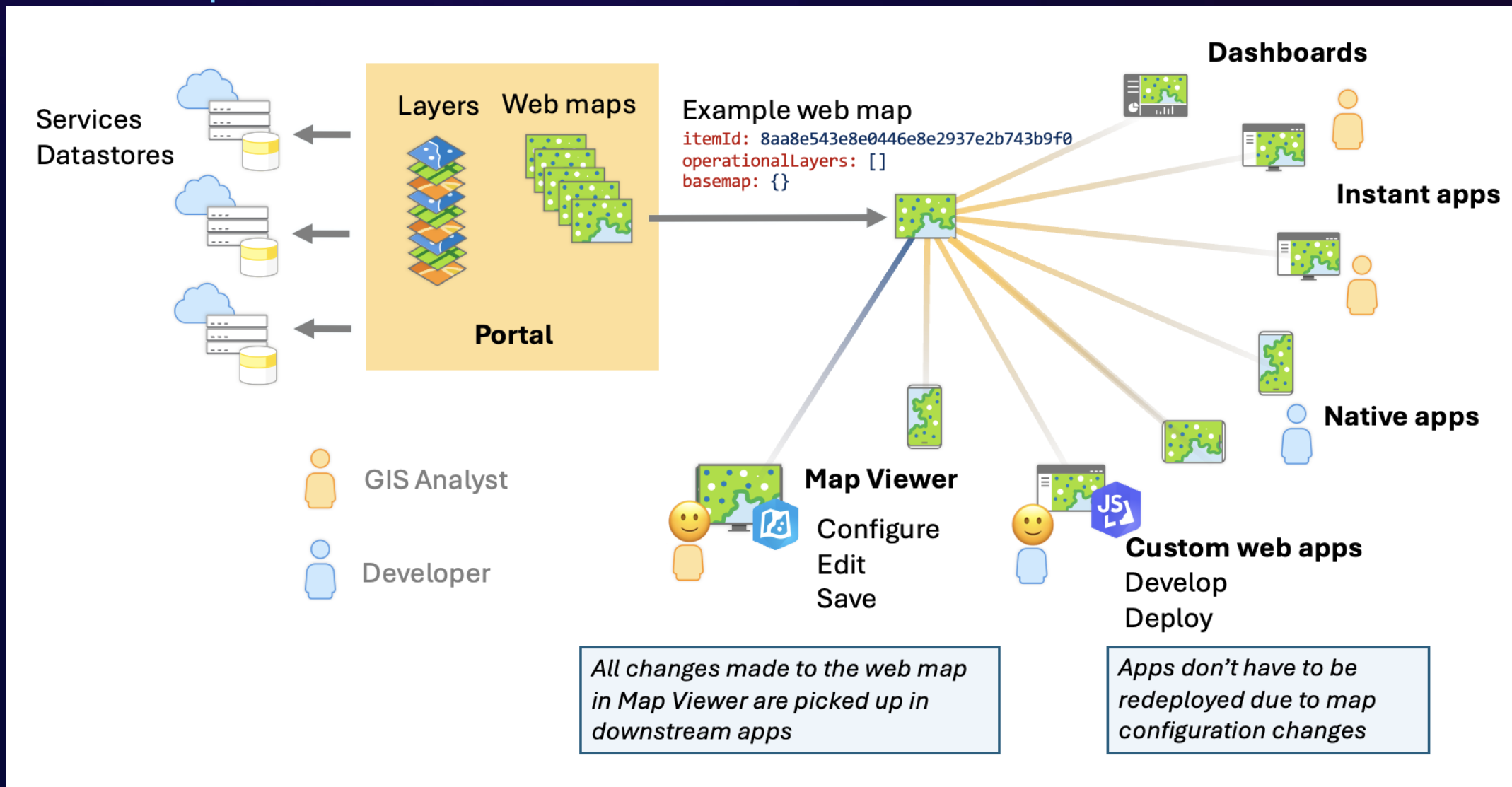
Maintenabilité

Sans la webmap




Maintenabilité

Avec la webmap



Réutilisation

```
<arcgis-map item-id="3ecc498e68cd4022a882363424794422">
```



Intégration de la webmap avec les composants et extension avec le SDK JavaScript d'ArcGIS et Calcite

Intégration

```

<script type="module" src="https://js.arcgis.com/calcite-components/3.0.2/calcite.esm.js"></script>

<link rel="stylesheet" href="https://js.arcgis.com/4.32/esri/themes/light/main.css" />
<script src="https://js.arcgis.com/4.32/"></script>

<script type="module" src="https://js.arcgis.com/map-components/4.32/arcgis-map-components.esm.js"></script>
</head>

<body>
  <arcgis-map item-id="55496d29650f4e11b53b4c9d724519cc">
    <arcgis-zoom position="top-left"></arcgis-zoom> } widget 1
    <arcgis-expand expanded position="top-left"> } widget 2
      <arcgis-bookmarks ></arcgis-bookmarks>
    </arcgis-expand>
    <arcgis-legend></arcgis-legend> } widget 3
  </arcgis-map>
</body>

</html>

```

Diagram illustrating the structure of the webmap integration:

- The `<arcgis-map>` element contains three widgets:
 - `<arcgis-zoom>` (widget 1)
 - `<arcgis-expand>` (widget 2)
 - `<arcgis-legend>` (widget 3)
- The widgets are grouped under the label **widgets**.
- The entire structure is part of the **webmap**.

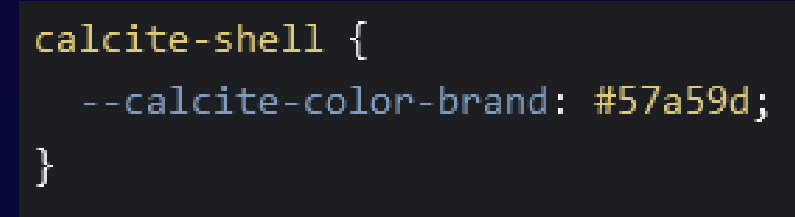
The screenshot shows the VS Code interface with the Calcite Switch component. The Design view on the left displays a visual representation of the switch, which is currently in the 'checked' state. The Properties panel on the right lists the following attributes:

- checked**: A checkbox that is checked, indicating the switch is on.
- disabled**: A checkbox that is unchecked, indicating the switch is not disabled.
- scale**: A dropdown menu set to 'm' (medium).
- form**: A text input field.
- label**: A text input field.
- label-text-end**: A text input field.
- label-text-start**: A text input field.
- name**: A text input field.

The Code view at the bottom shows the HTML markup for the switch:

```
<calcite-switch checked></calcite-switch>
```

```
<div id="switch-container">
  <h4>
    Afficher la fréquentation
  </h4>
  <calcite-switch id="switchbutton" checked></calcite-switch>
</div>
```

Rajouter des comportements personnalisés avec JavaScript

Ouverture du script et attente du chargement complet

```
<script> Ouverture de la balise script => on écrit maintenant en JS
  customElements.whenDefined("arcgis-map").then(() => { Attend que le composant map soit chargé et prêt à être manipulé
    const arcgisMap = document.querySelector("arcgis-map"); Sélectionne la webmap dans le DOM
    arcgisMap.addEventListener("arcgisViewReadyChange", async (event) => { Attend que la vue associée à la carte soit prête avant d'exécuter la suite du code
```

Rajouter des comportements personnalisés avec JavaScript

querySelector – sélection du composant dans le DOM

```
const switchbutton = document.querySelector("calcite-switch");
```

Rajouter des comportements personnalisés avec JavaScript

```
const layer_location = arcgisMap.map.layers.at(0);
const layer_frequentation = arcgisMap.map.layers.at(1);
const legendExpand = document.querySelector("arcgis-expand arcgis-legend").closest("arcgis-expand");
```

🔍 Search topics

- > tables
- > time
- > versionManagement
- > views
- > webdoc
- > webmap
- > webscene
- > widgets
- arcade
- AttributeBinsGraphic
- Basemap
- Camera
- CameraLayout

Map

ESM: `import Map from "@arcgis/core/Map.js";`

CDN: `const Map = await $arcgis.import("@arcgis/core/Map.js");`

Class: `@arcgis/core/Map`

Inheritance: Map → [Accessor](#)

Subclasses: [WebDocument2D](#) , [WebScene](#)

Since: ArcGIS Maps SDK for JavaScript 4.0

On this page

- Constructors
- Properties
- Methods

The Map class contains properties and methods for storing, managing, and overlaying [layers](#) common to both 2D and 3D viewing. Layers can be added and removed from the map, but are rendered via a [MapView](#) (for viewing data in 2D) or a [SceneView](#) (for viewing data in 3D). Thus a map instance is a simple container that holds the layers, while the [View](#) is the means of displaying and interacting with a map's layers and basemap.

Rajouter des comportements personnalisés avec JavaScript

Carte blanche, c'est vous les dévs

```
const changeVisibility = () => {  
  const isShowingFrequentation = layer_frequentation.visible;  
  layer_location.visible = isShowingFrequentation;  
  layer_frequentation.visible = !isShowingFrequentation;  
  legendExpand.expanded = !isShowingFrequentation;  
};
```

Rajouter des comportements personnalisés avec JavaScript

Event

Sample

The screenshot displays the Calcite UI sample application. On the left, a 'Switch' component is shown with a green toggle. Below it, the HTML code is visible: `<calcite-switch checked></calcite-switch>`. On the right, the 'Events' panel for the 'calcite-switch' component is open, showing a single event: 'calciteSwitchChange'. Below the event list, a button says 'Open developer tools to review event logs'. At the bottom of the interface, there are tabs for 'Full', 'HTML', 'CSS', and 'JS', with 'HTML' currently selected. A footer bar contains the text 'v3.3.0 Code' and a 'What are events?' link.

```
switchbutton.addEventListener("calciteSwitchChange", changeVisibility);
```



esri®

THE
SCIENCE
OF
WHERE®