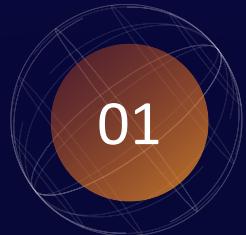


Développer avec la webmap et les composants

Pauline Louis, Esri France
plouis@esrifrance.fr



Sommaire



01

Que configurer dans la webmap ?



02

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



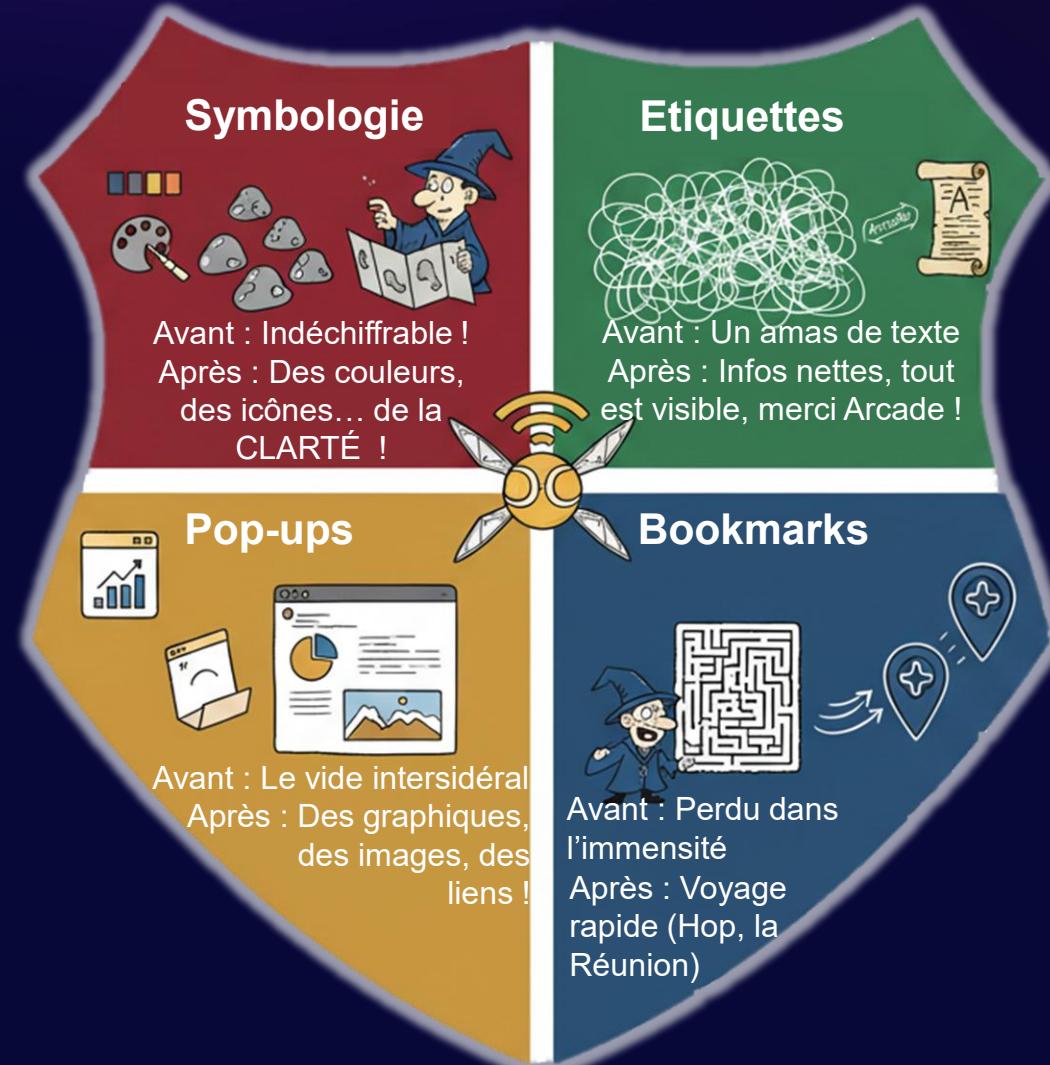
03

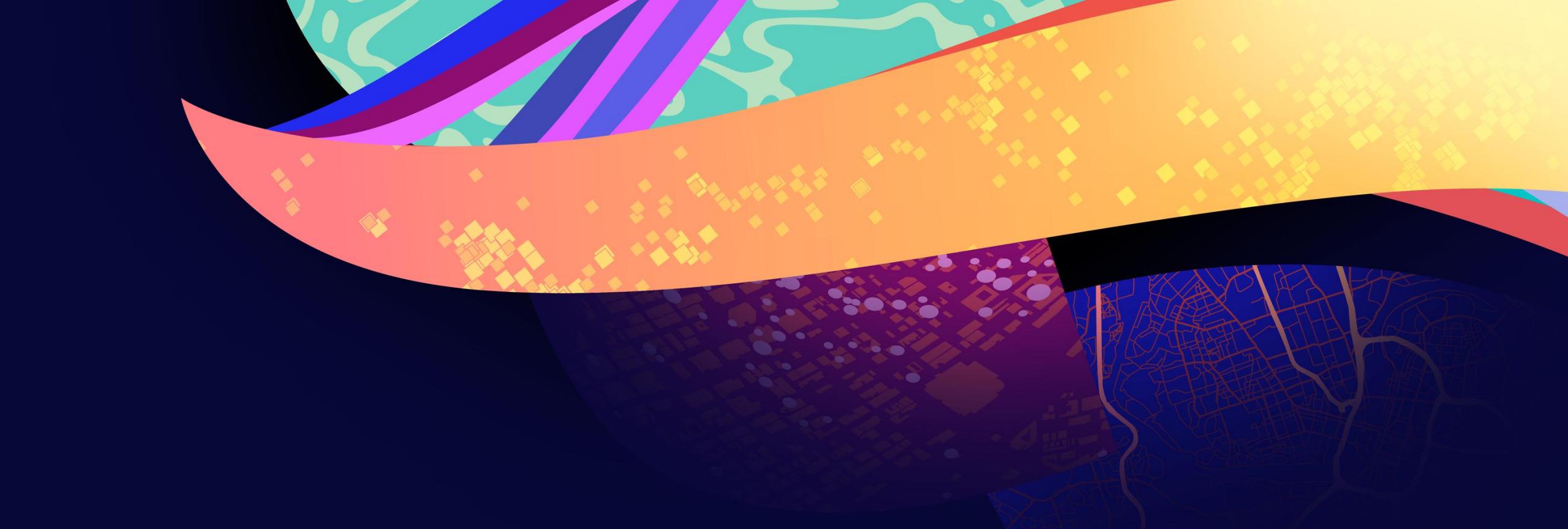
Intégration de la webmap avec les composants
et extension avec le SDK JavaScript d'ArcGIS



Que configurer dans la webmap ?

Configuration de la webmap





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>
```

Moins de code



Moins de code – les composants

ArcGIS Maps SDK for JavaScript / References

Home Sample Code References **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.

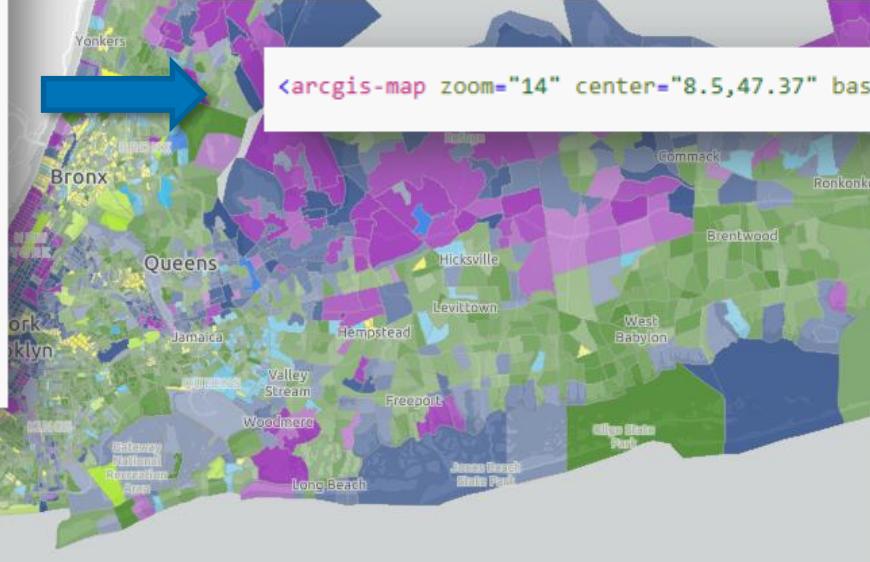
[On this page](#)

- [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

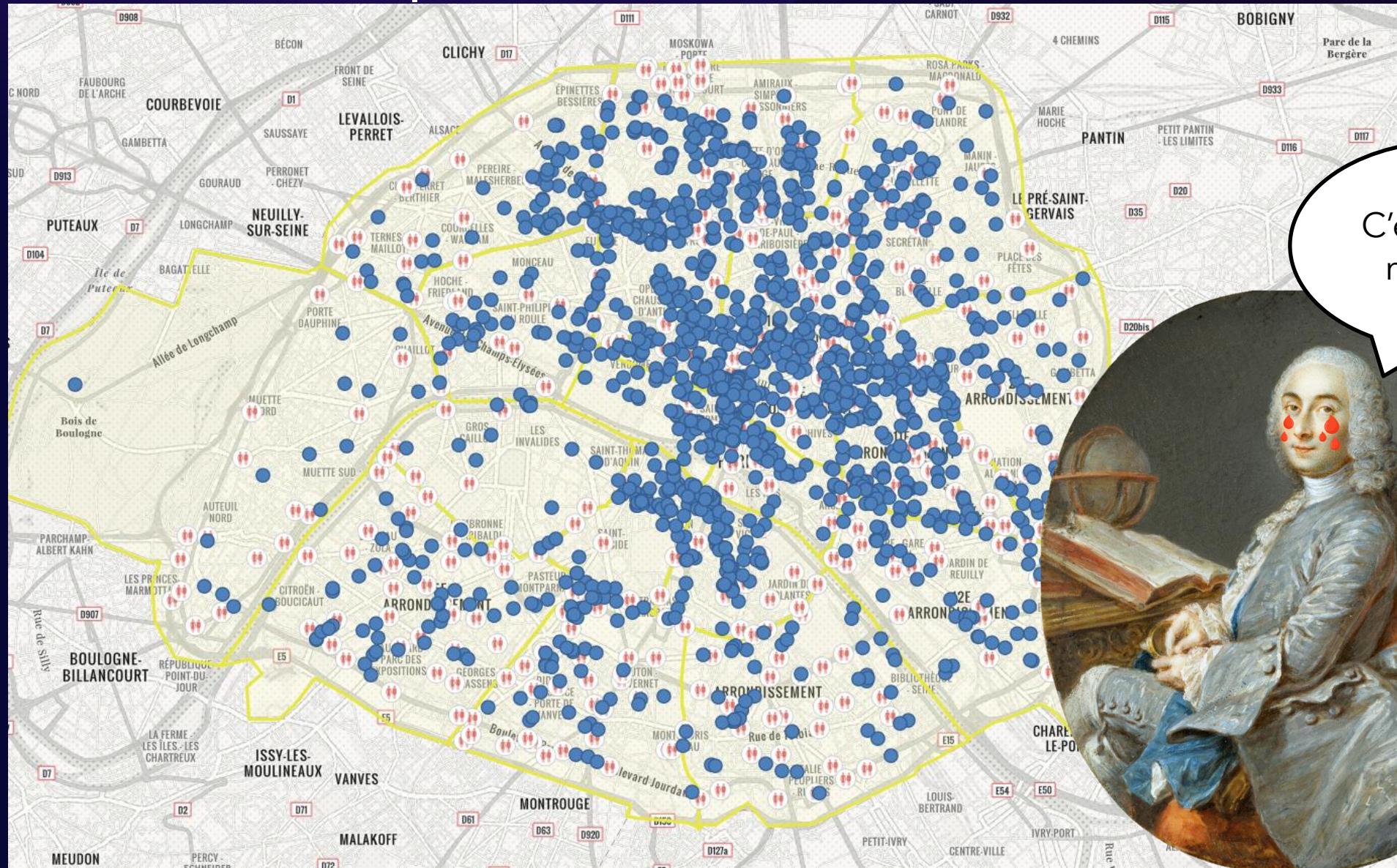
Distance Measurement 2D

Editor

Elevation Profile

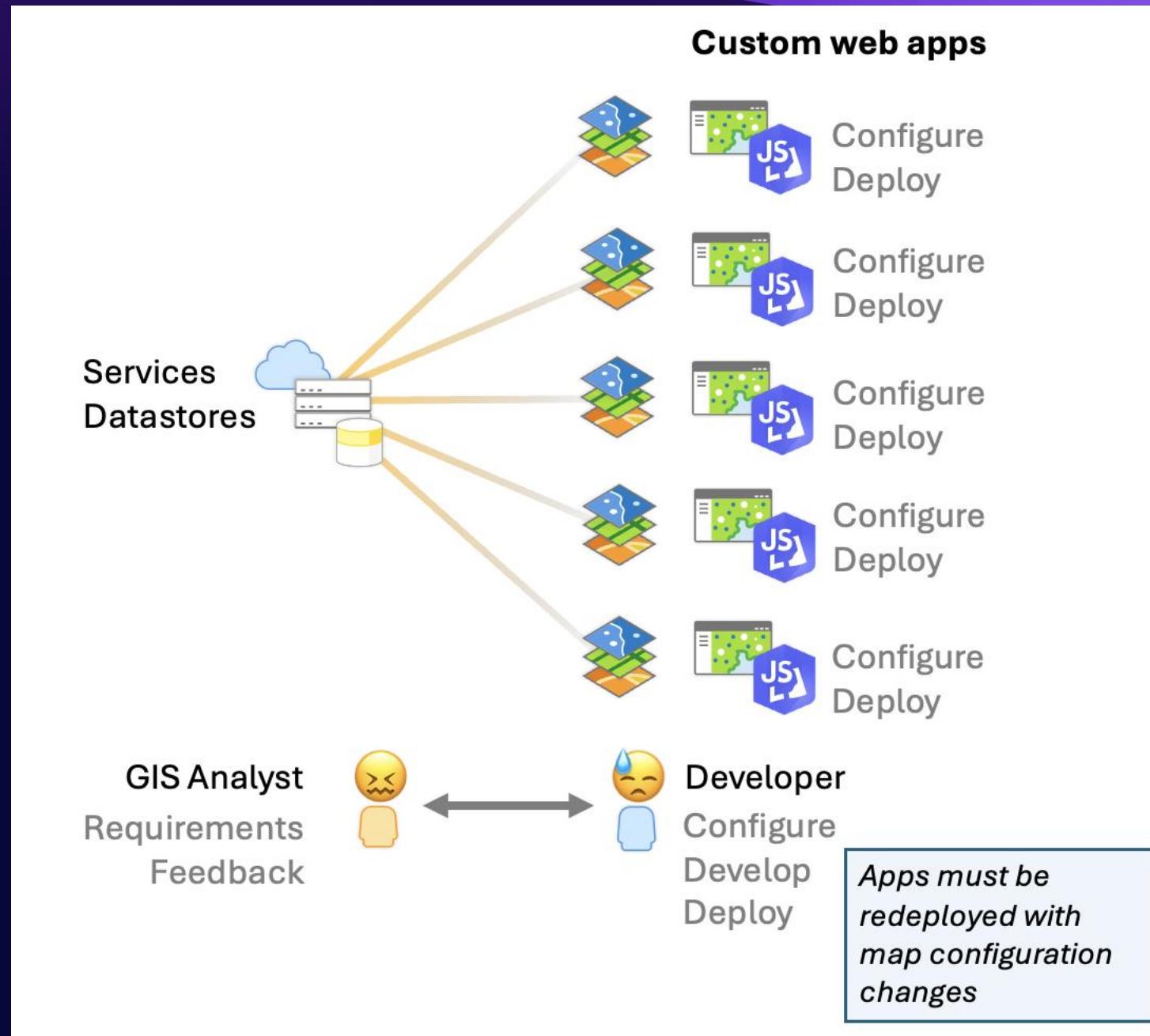
Expand

Amélioration de la répartition du travail



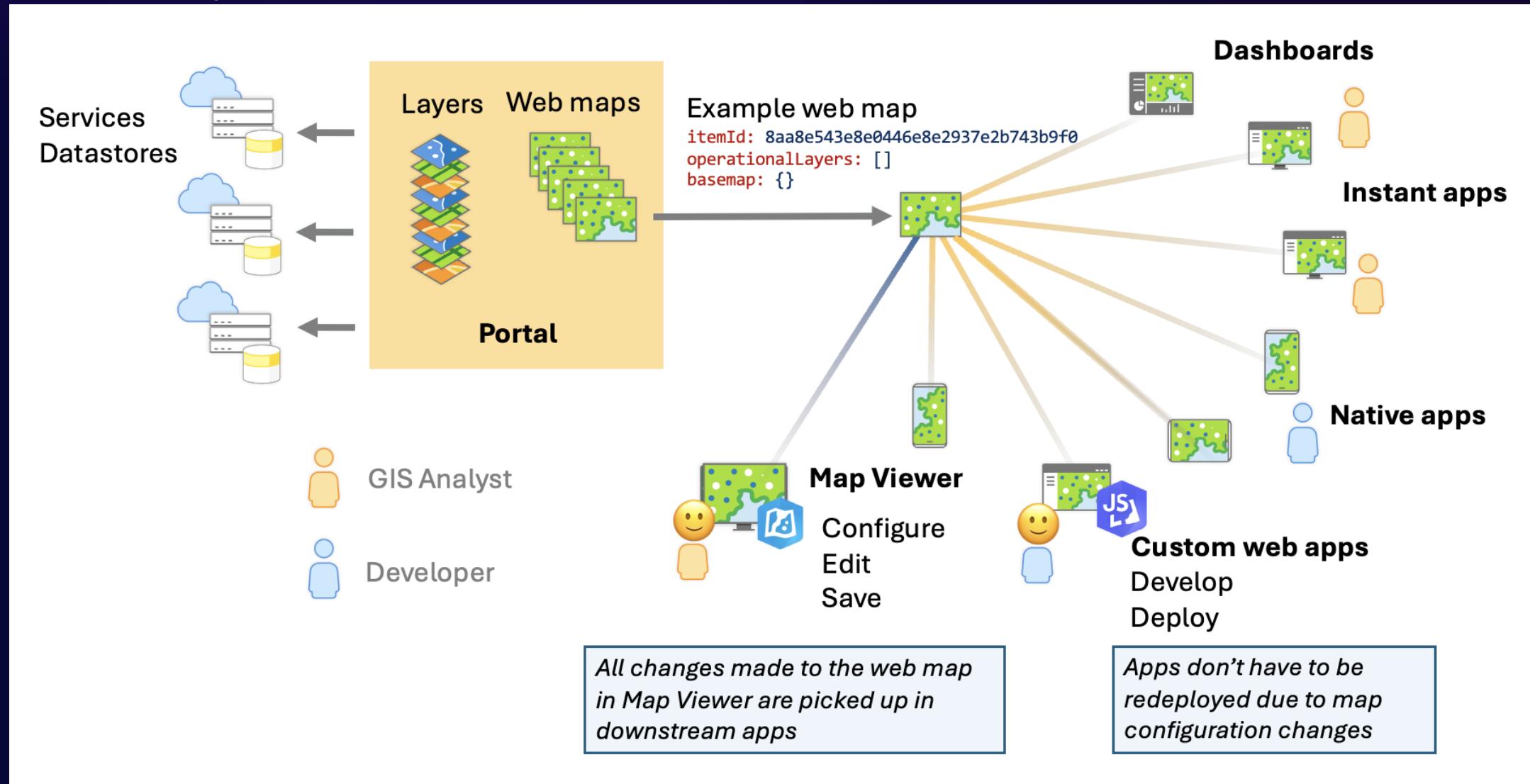
Maintenabilité

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

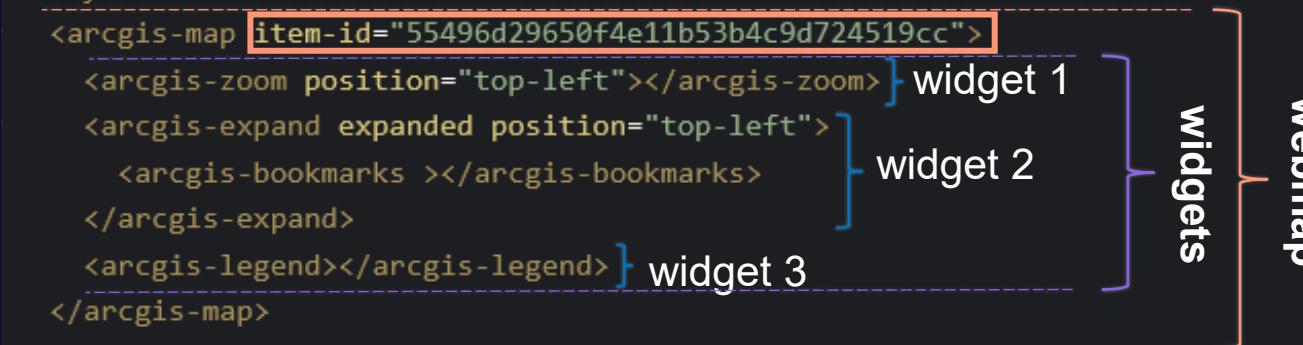
```
<script type="module" src="https://js.arcgis.com/calcite-components/3.0.2/calcite.esm.js"></script>

<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>

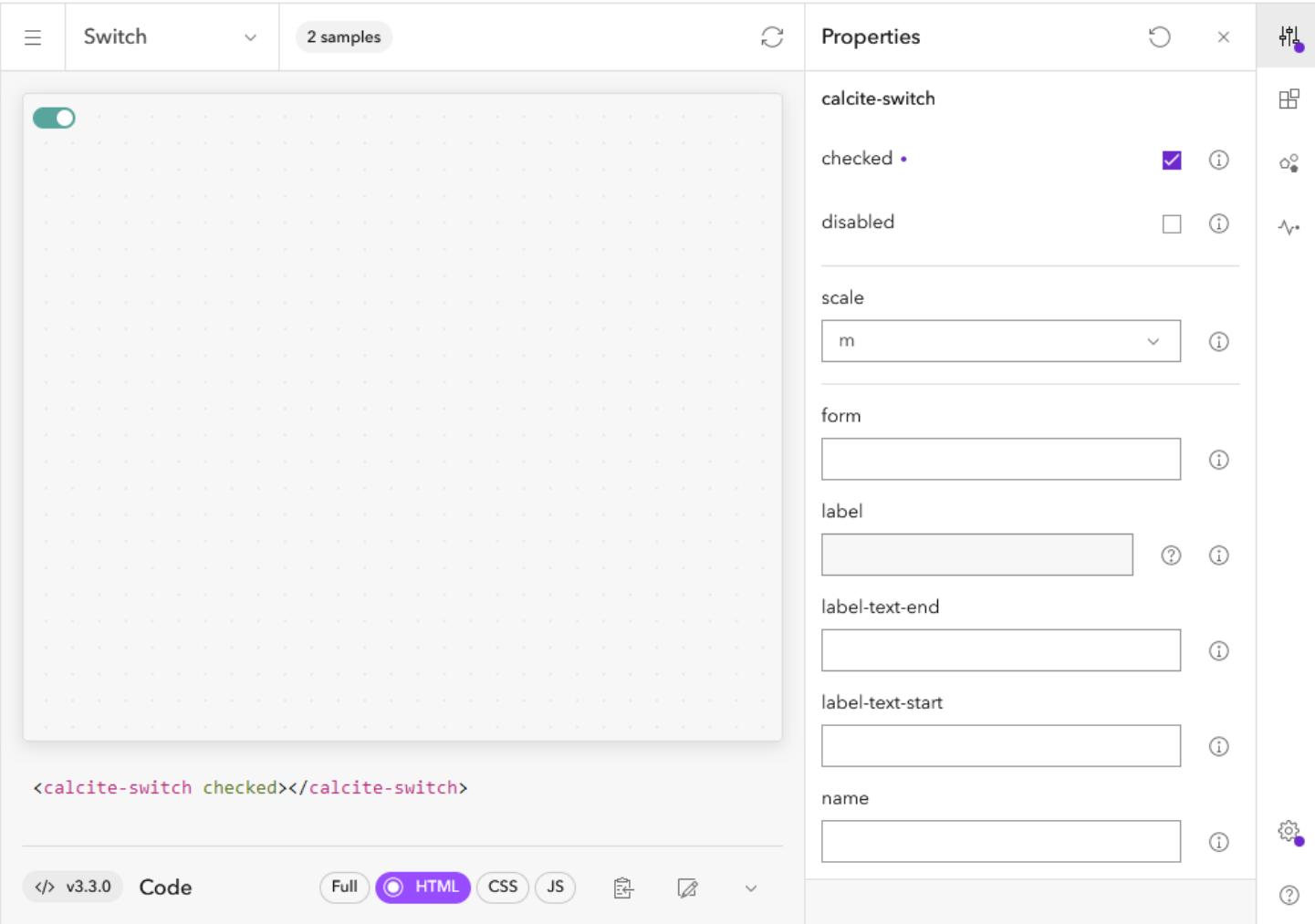
<body>
  <arcgis-map item-id="55496d29650f4e11b53b4c9d724519cc">
    <arcgis-zoom position="top-left"></arcgis-zoom> } widget 1
    <arcgis-expand expanded position="top-left">
      <arcgis-bookmarks ></arcgis-bookmarks> } widget 2
    </arcgis-expand>
    <arcgis-legend></arcgis-legend> } widget 3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

</html>
```



Rajouter des éléments d'UI avec Calcite

Sample



The screenshot shows the Calcite UI Designer interface. On the left, there's a preview area displaying a green switch button. Below it is the component's code: <calcite-switch checked></calcite-switch>. To the right is the 'Properties' panel, which lists several configuration options:

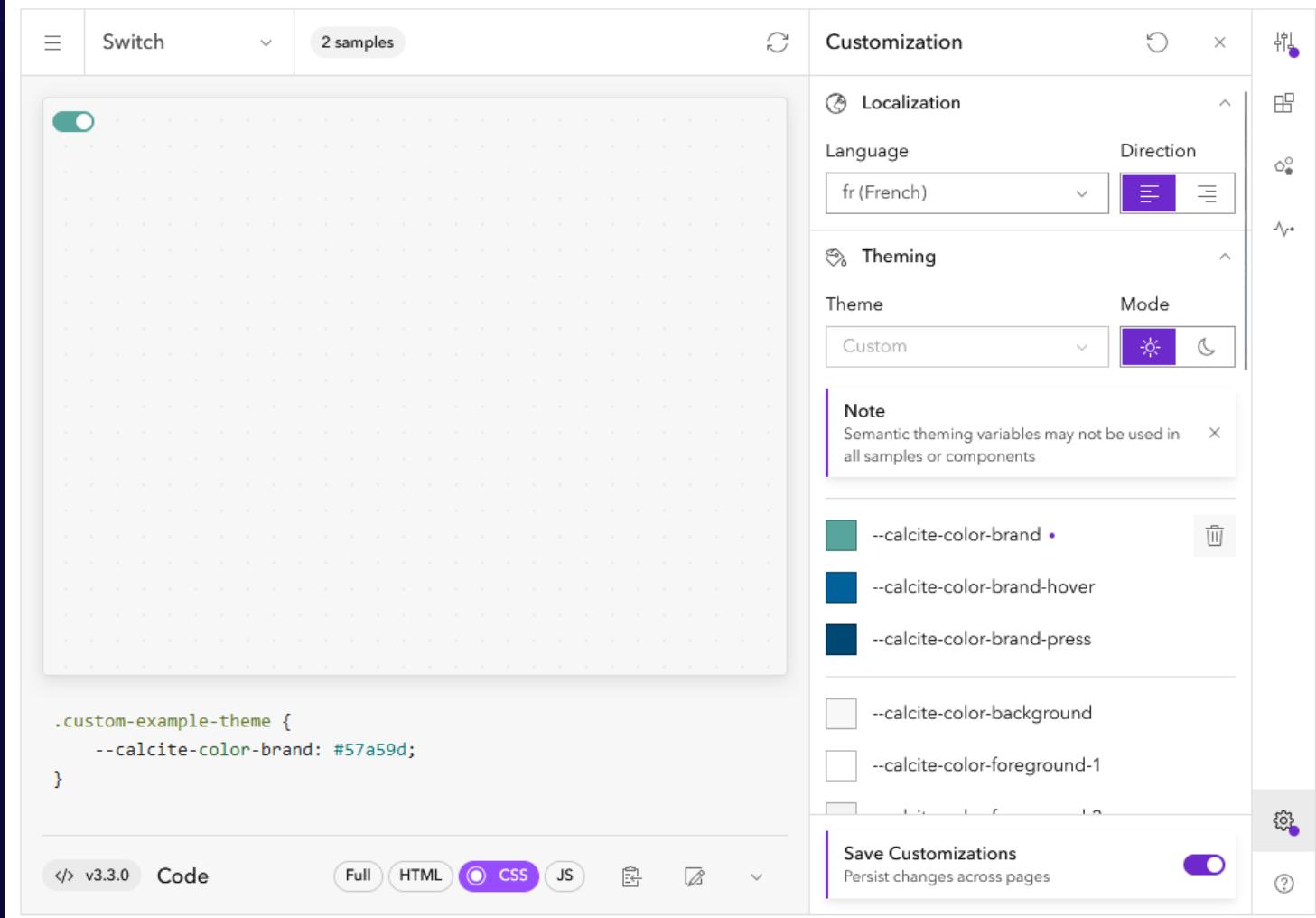
- calcite-switch
- checked •
- disabled
- scale
- form
- label
- label-text-end
- label-text-start
- name

At the bottom of the properties panel, there are tabs for 'Code' (selected), 'Full', 'HTML' (highlighted in purple), 'CSS', 'JS', and other icons.

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

Rajouter des éléments d'UI avec Calcite – avec style

Sample



The screenshot shows the Calcite UI customization interface. On the left, there's a preview window displaying a switch component with its toggle button turned on. Below the preview, the CSS code for the switch is shown:

```
.custom-example-theme {  
  --calcite-color-brand: #57a59d;  
}
```

On the right, the "Customization" panel is open, showing the "Localization" and "Theming" sections. In the "Theming" section, under the "Mode" dropdown, the "Custom" theme is selected. A note below the mode dropdown states: "Semantic theming variables may not be used in all samples or components". A list of semantic theming variables is provided, with the first one, "--calcite-color-brand", being highlighted with a teal square icon. The "Save Customizations" toggle at the bottom is turned on.

```
calcite-shell {  
  --calcite-color-brand: #57a59d;  
}
```

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  
        // Code à exécuter lorsque la vue est prête  
    })  
})
```

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

Map

Search topics

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

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 isShowingFrequmentation = layer_frequmentation.visible;
    layer_location.visible = isShowingFrequmentation;
    layer_frequmentation.visible = !isShowingFrequmentation;
    legendExpand.expanded = !isShowingFrequmentation;
};
```

Rajouter des comportements personnalisés avec JavaScript

Event

Sample

The screenshot shows the Apache Calcite UI developer tool interface. On the left, there is a preview area displaying a single switch component with the 'checked' attribute set. Below the preview is the corresponding HTML code: <calcite-switch checked></calcite-switch>. At the bottom, there are tabs for 'Code', 'Full', 'HTML' (which is selected), 'CSS', and 'JS'. A modal window titled 'Events' is open over the preview. It lists an event for the 'calcite-switch' component: 'calciteSwitchChange'. The event details show it is triggered by '(+) calciteSwitchChange'. Below the event list is a note: 'Open developer tools to review event logs'. In the bottom right corner of the modal, there is a block of JavaScript code: switchbutton.addEventListener("calciteSwitchChange", changeVisibility);. At the very bottom of the interface, there is a footer with links: 'What are events?' and 'Learn about core concepts'.

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



esri

THE
SCIENCE
OF
WHERE®