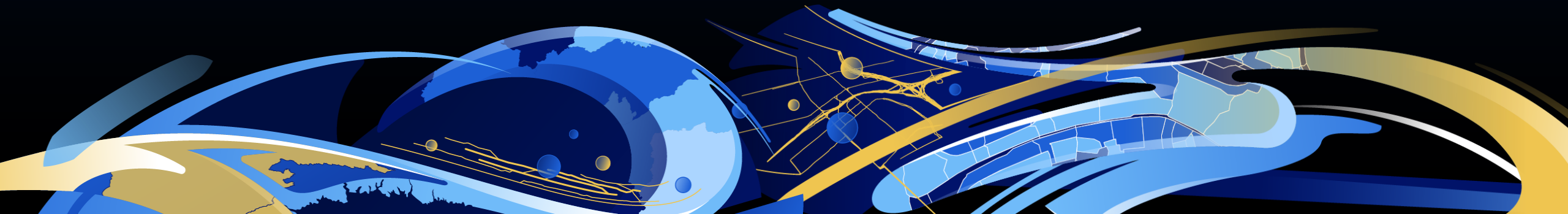


# ArcGIS API for JavaScript 4.x

Fabrice Grataloup – Marc Le Moigne



# Plan de la présentation

- 1. Fondamentaux**
  - 1. Présentation API**
  - 2. Nouvelles couches**
  - 3. Concepts de l'API**
  - 4. Data Visualisation**
- 2. 3D**
  - 1. Vue 2D vers 3D**
  - 2. BIM**
- 3. TypeScript**
- 4. Widgets**
- 5. Environnement et frameworks JS**
- 6. Roadmap**

# Introduction

- 4 mises à jour par an :
- Mars 2019 : 4.11 et 3.28
- Juin 2019 : 4.12 et 3.29
- Octobre 2019 : 4.13 et 3.30
- Décembre 2019 : 4.14 et 3.31
- Avril 2020 : 4.15 et 3.32
- Juillet 2020 : 4.16 et 3.33
- Octobre 2020 : 4.17 et 3.34
- **Décembre 2020 : 4.18 et 3.35**

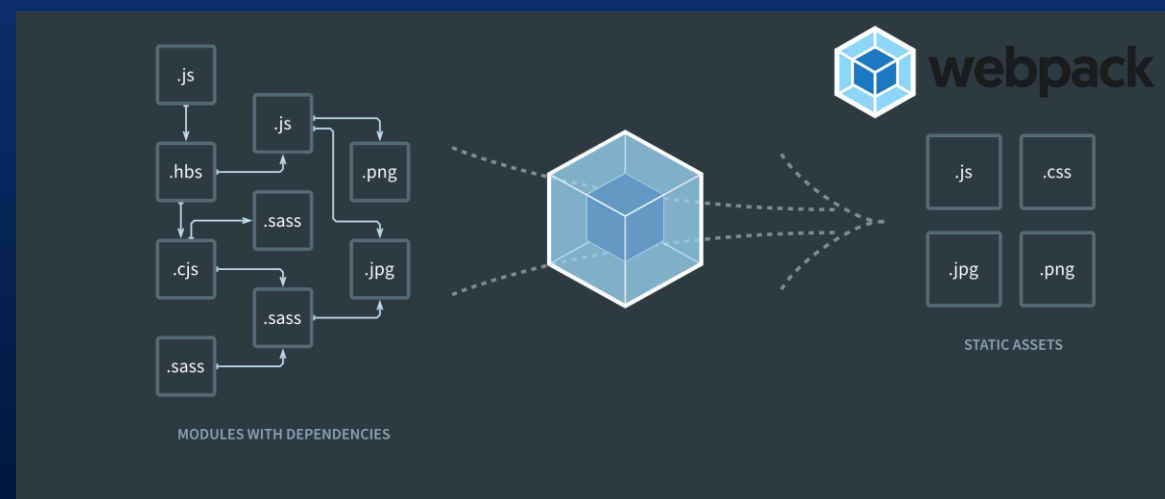
# Fondamentaux

# Présentation API

- **AMD (Asynchronous Module Definition)**
- **Dojo version 5.0.4** (Depuis la version 4.12) – Suppression progressive de la dépendance
  - **require** : ensemble des modules requis
  - **define** : enregistre un nouveau module et ses dépendances
- **ArcGIS CLI**
  - <https://github.com/Esri/arcgis-js-cli>

```
arcgis create my-application
```

```
arcgis create my-react-application -t react
```



# Présentation API

- Séparation des concepts
  - WebMap et WebScene : description
  - MapView et SceneView : représentation
    - Séparation des moteurs 2D et 3D
- WebMap / WebScene
  - Basemap
  - Operational Layers
  - Ground
  - SpatialReference

```
var map = new Map({
  /**streets, satellite, hybrid, topo, gray,
  dark-gray, oceans, national-geographic, terrain, osm,
  dark-gray-vector, gray-vector, streets-vector, streets-night-vector,
  streets-relief-vector, streets-navigation-vector*/
  basemap: "topo",

  ground : "world-elevation"
});
```

```
var webmap = new WebMap({
  portalItem: {
    id: "e691172598f04ea8881cd2a4adaa45ba"
  }
});
```

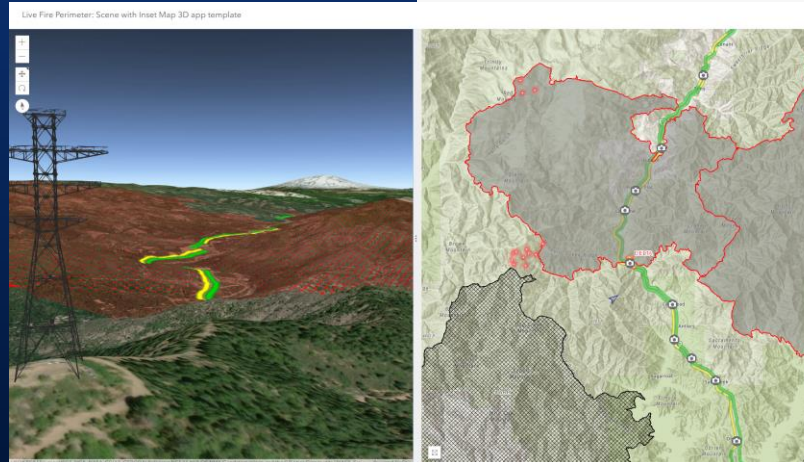
```
/* Set the WebMap instance to the map property in a MapView.
***** */
```

```
var view = new MapView({
  map: webmap,
  container: "viewDiv"
});
```

```
var scene = new WebScene({
  portalItem: {
    id: "3a9976baef9240ab8645ee25c7e9c096"
  }
});
```

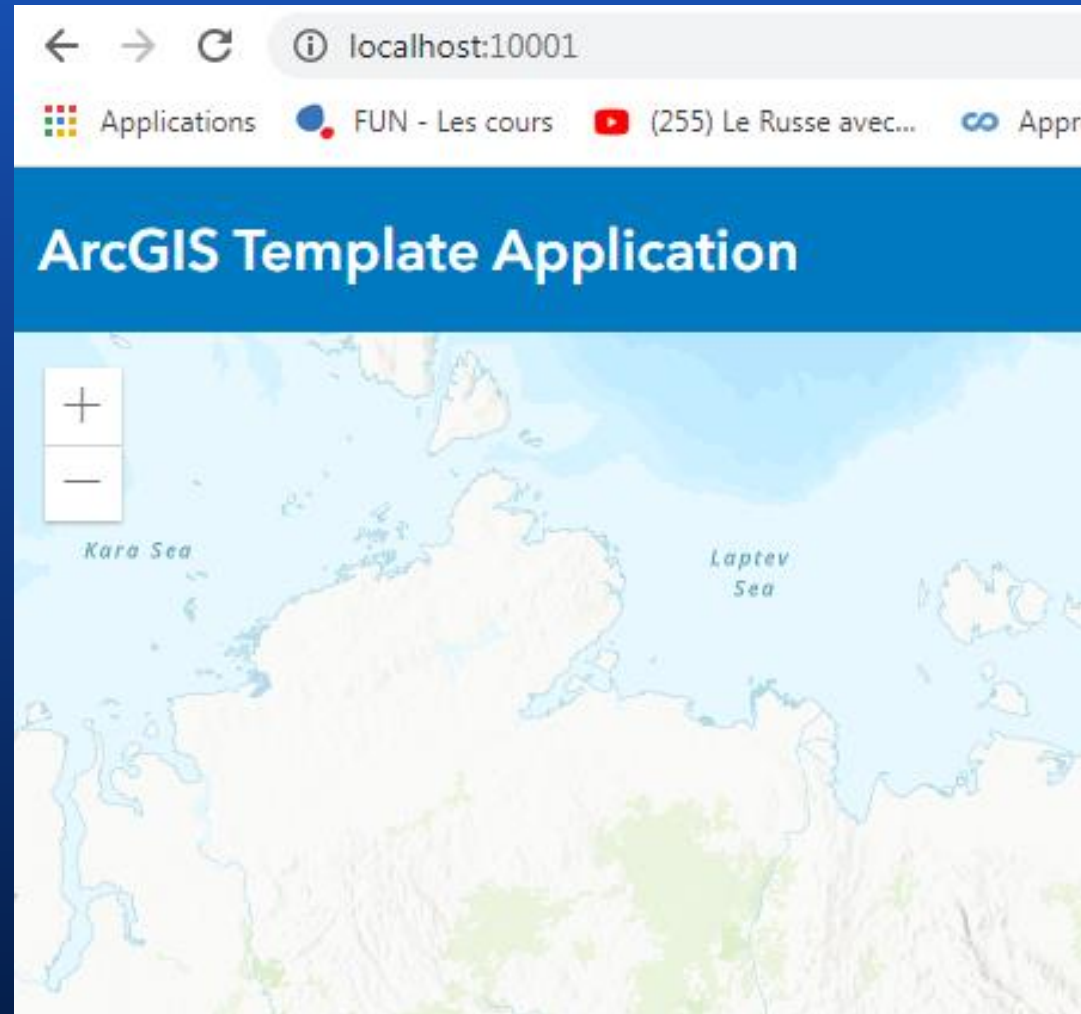
```
/* Set the WebScene instance to the map property in a SceneView.
***** */
```

```
var view = new SceneView({
  map: scene,
  container: "viewDiv"
});
```



# Démo Modern App

- ArcGIS CLI
- MapView
- WebMap



# Quelles données ?

## Couches supportées

### 2D Layers :

GeoRSSLayer

KMLLayerLayer (4.5)

### 3D Layers

BaseElevationLayer (4.4)

ElevationLayer

IntegratedMeshLayer

PointCloudLayer

SceneLayer

BuildingSceneLayer (4.10)

### 2D/3D Layers

BaseTileLayer (4.4)

BingMapsLayer (4.8)

WebTileLayer

OpenStreetMapLayer

CSVLayer

FeatureLayer

GraphicsLayer

GroupLayer

ImageryLayer

MapImageLayer

StreamLayer

TileLayer

VectorTileLayer

MapNotesLayer (4.4)

WMSLayer (4.4)

WMTSLayer (4.4)

GeoJSONLayer (4.11)

**ImageryTileLayer (4.16) pas de blendMode en 3D**

**OGCFeatureLayer (4.16)**

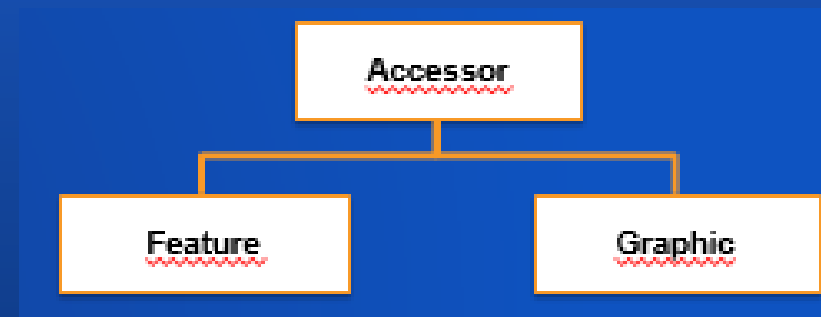
**BaseDynamicLayer (4.4) pas de blendMode en 3D**



# Concepts de l'API

- **Accessor**

- Socle de l'API, la plupart des objets héritent d'accessor
- Permet d'accéder aux propriétés d'un élément rapidement
- Remplace les méthodes get/set



```
map.set("basemap", "topo");  
// is equivalent to accessor  
map.basemap = "topo";
```

- **Autocasting**

- Définir des propriétés typées sans appeler les modules

```
var diamondSymbol = {  
  type: "simple-marker", // autocasts as new SimpleMarkerSymbol()  
};
```

# Concepts de l'API : gestion des événements

- **Promise**

- Traitement asynchrone
- 3 états : pending / resolved / Rejected
- De nombreuses classes héritent de Promise

```
var geoService = new GeometryService( "https://sampleserver6.arcgisonline.com/arcgis/");

var projectParams = new ProjectParameters({
  geometries: [points],
  outSR: outSR,
  transformation = transformation
});

geoService.project(projectParams)
  .then(function(projectedGeoms){
    console.log("projected points: ", projectedGeoms);
  }, function(error){
    console.error(error);
  });
```

- **Loadable**

- Les objets cartographiques ont un cycle de vie
- Propriété loadStatus

- **Watch**

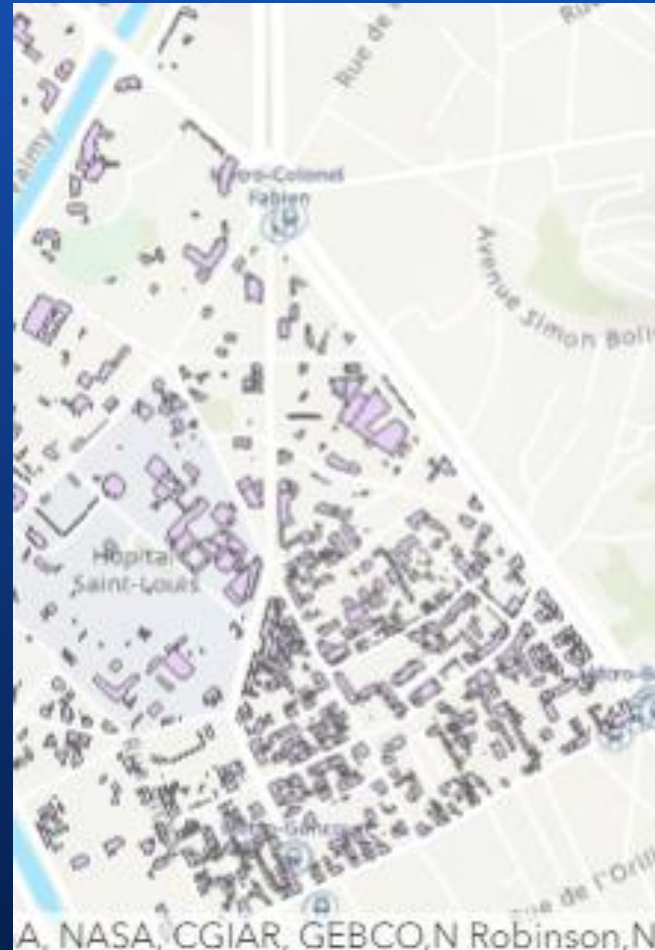
- On peut tester le changement de propriétés d'un objet
- La classe esri/core/watchUtils est un utilitaire permettant d'aider au test de ces propriétés

Value	Description
not-loaded	The object's resources have not loaded.
loading	The object's resources are currently loading.
loaded	The object's resources have loaded without errors.
failed	The object's resources failed to load. See <a href="#">loadError</a> for more details.

```
layer.watch("loadStatus", function(status) {/**/});
```

# Démo Loading Map

- AMD
- Accessor
- Autocasting
- Loadable
- Watch
- Promise



the view center changed. x: 262

6253678.06

the view extent changed:

xmin:258197.73 xmax: 267230.20

ymin:6252395.26 ymax: 6254960

the view center changed. x: 262

6253678.06

the view extent changed:

xmin:258157.31 xmax: 267270.69

ymin:6252547.23 ymax: 6254808

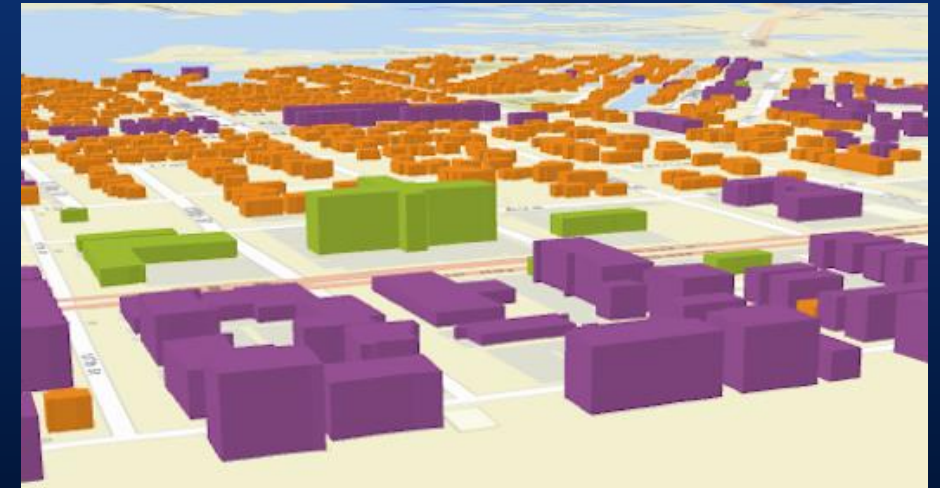
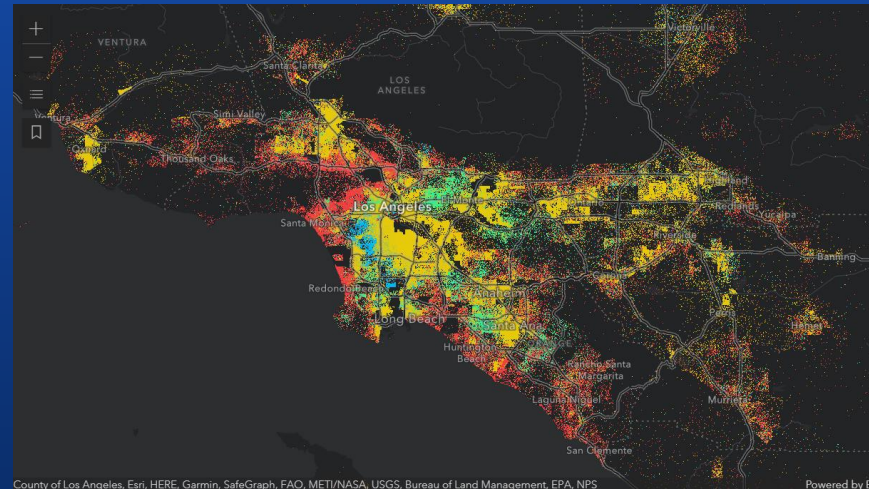
# Visualisation des données

- **Renderer : définit l'aspect visuel des entités d'une couche**

- SimpleRenderer
- UniqueValueRenderer
- ClassBreaksRenderer
- DotDensityRenderer (4.11)

- **Basé sur les valeurs de champs**

- Variation de couleur, taille, opacité
- Extrusion



# Visualisation des données








MapView (2D)	<b>FillSymbol</b> <b>LineSymbol</b> <b>MarkerSymbol</b> <b>TextSymbol</b> <b>CIMSymbol (4.12)</b>
SceneView (3D)	<b>Symbol3D</b>

- **CIMSymbol (Beta)**

- utilisés pour afficher des symboles vectoriels multicouches pour les entités et les graphiques.

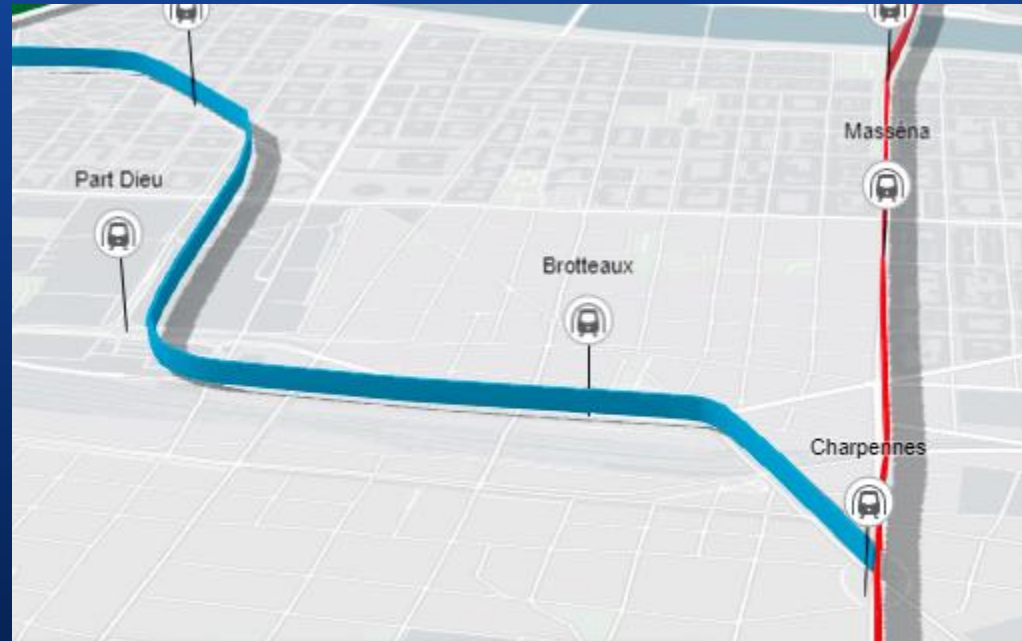
Specification : <https://github.com/Esri/cim-spec/blob/master/docs/v2/Types.md#symbol>

- **SimpleLineSymbol avec marqueurs**

Symbol3DLayer Type	Symbol3D Type	Flat/Volumetric	Size Units	Example
IconSymbol3DLayer	PointSymbol3D, PolygonSymbol3D	flat	points	
ObjectSymbol3DLayer	PointSymbol3D, PolygonSymbol3D	volumetric	meters	
LineSymbol3DLayer	LineSymbol3D, PolygonSymbol3D	flat	points	
PathSymbol3DLayer	LineSymbol3D	volumetric	meters	
FillSymbol3DLayer	PolygonSymbol3D, MeshSymbol3D	flat	-	
ExtrudeSymbol3DLayer	PolygonSymbol3D	volumetric	meters	
TextSymbol3DLayer	PointSymbol3D, LineSymbol3D, PolygonSymbol3D, LabelSymbol3D	flat	points	

# Démo Symbologie

- Unique Value Renderer
- Symbologie dynamique
- CIMSymbol
- 2D/3D





# Visualisation des données

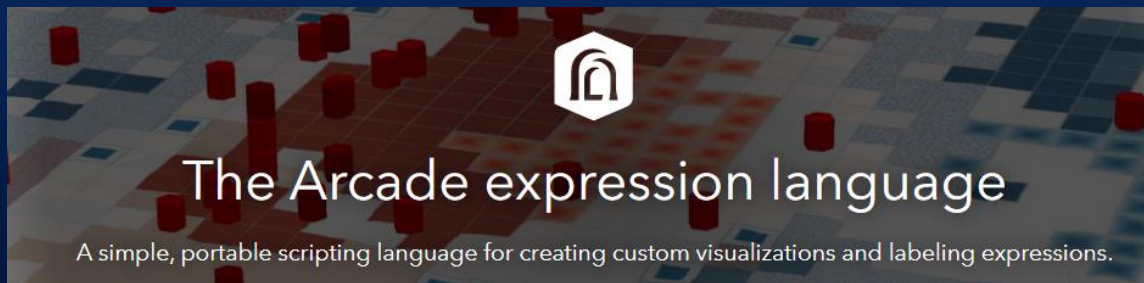
- **Arcade version 1.11**

<https://developers.arcgis.com/arcade/playground/>

<https://developers.arcgis.com/arcade/>

<https://developers.arcgis.com/arcade/guide/version-matrix/>

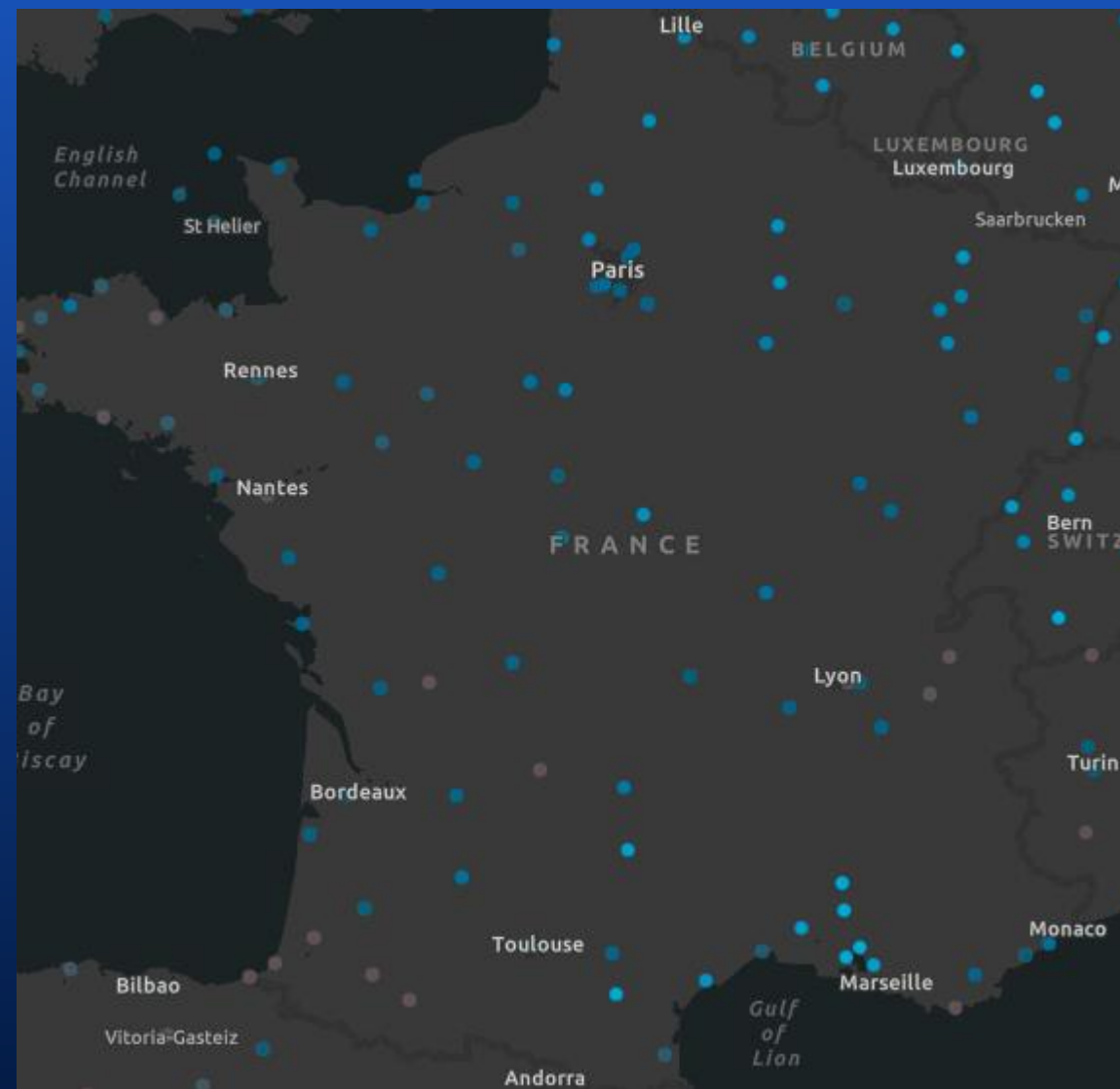
- **Portable** : toute la plateforme ArcGIS
- **Sécurisé**
  - Expressions de scripts ne compromettent pas la sécurité de l'application
- **Léger**
  - Expressions courtes, langage léger et sources non volumineuses
- **Géospatial**
  - Classes orientés géospatiales



# Démo

## Arcade expression

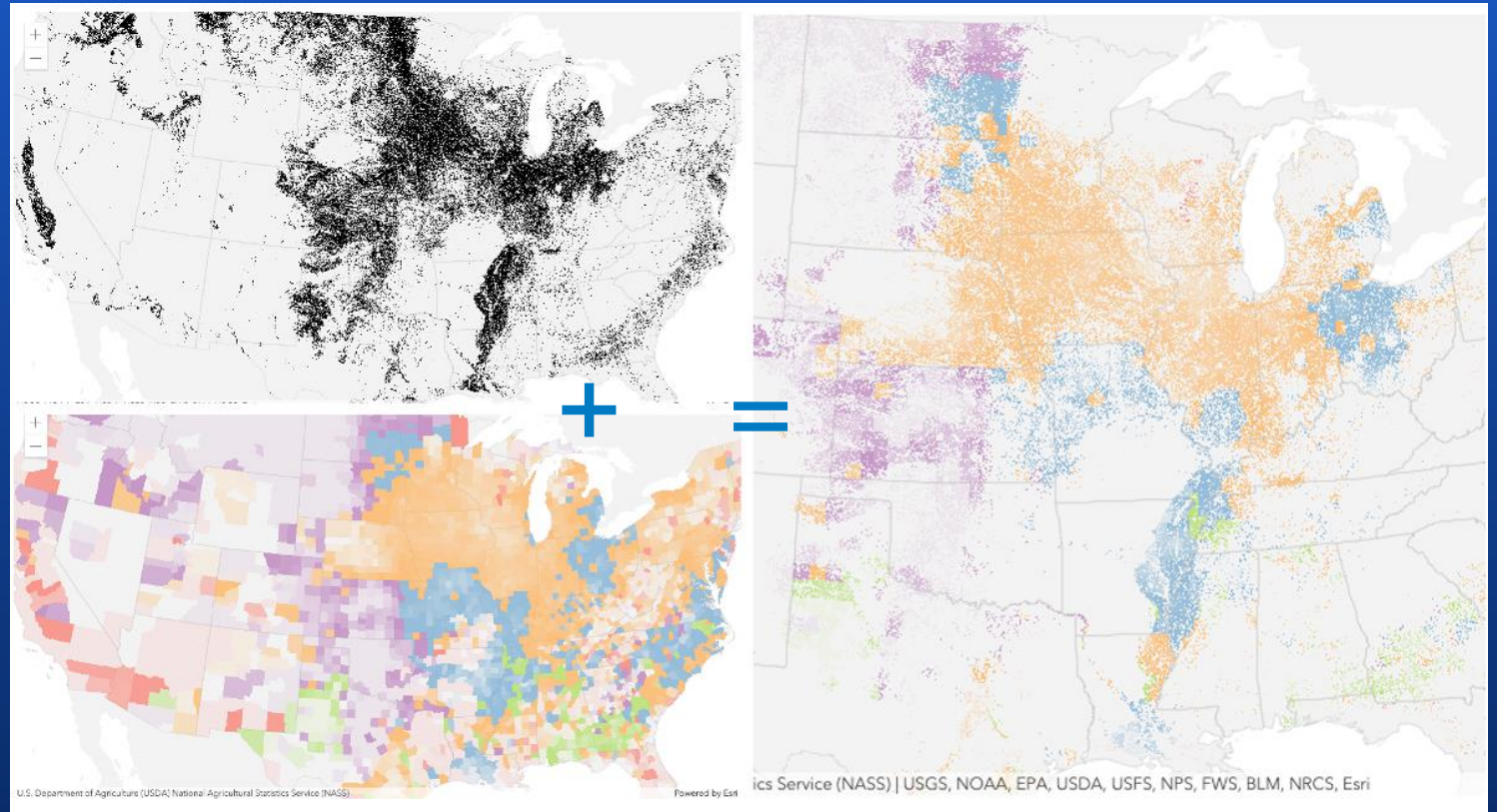
- Arcade
- Symbologie
- PopupTemplate



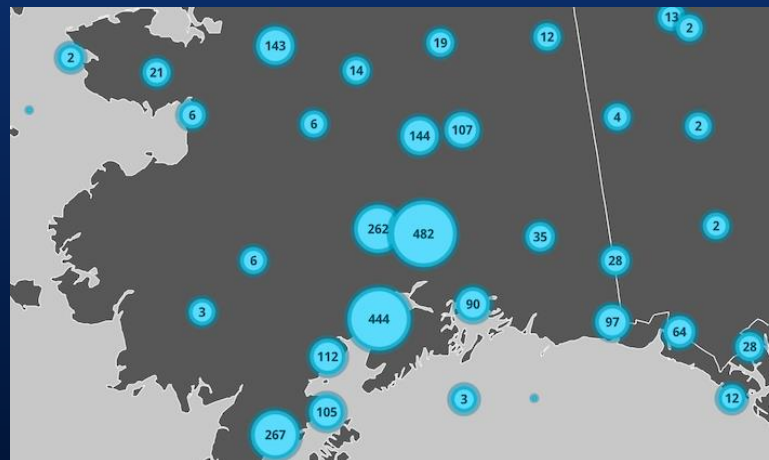


# Visualisation des données

- Layer Blend Mode



- Clustering avancé

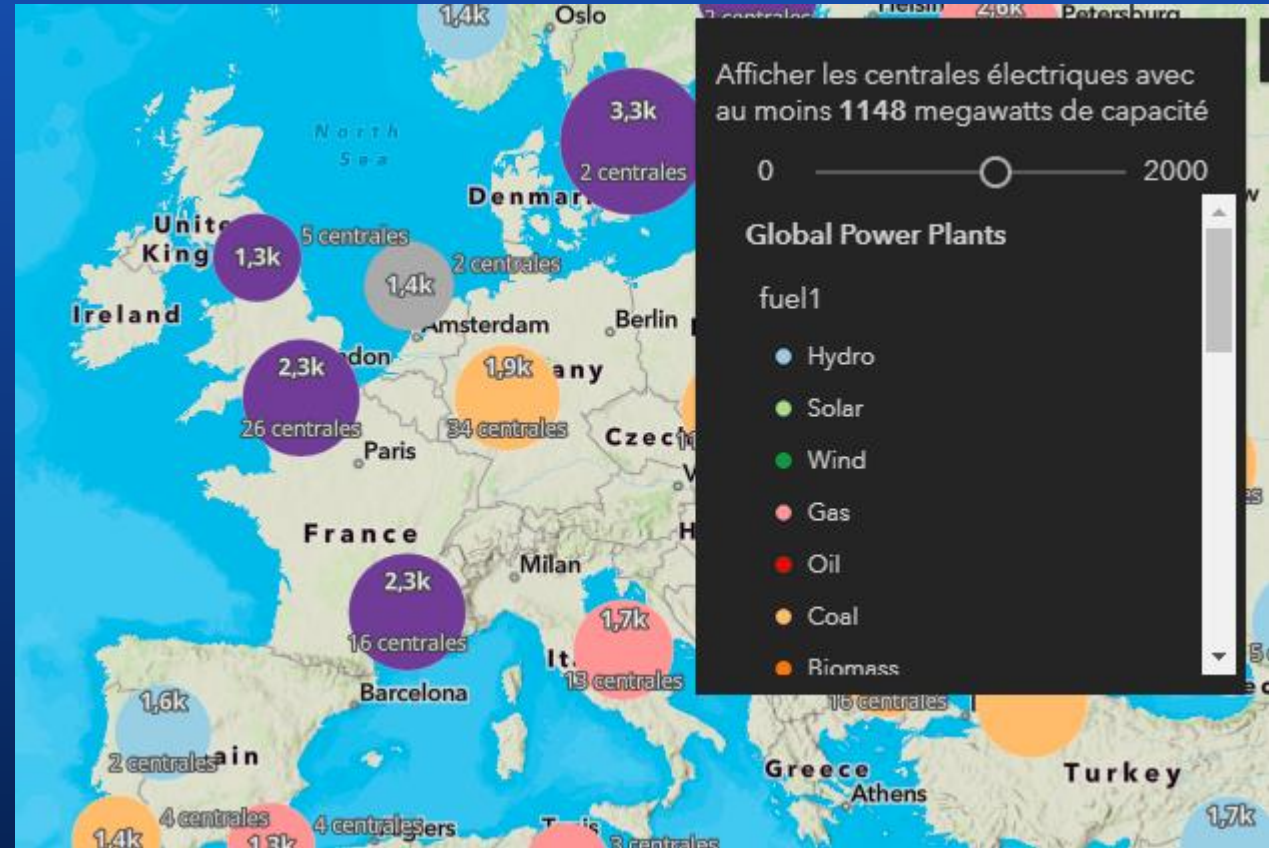


# Démo

# Visualisation

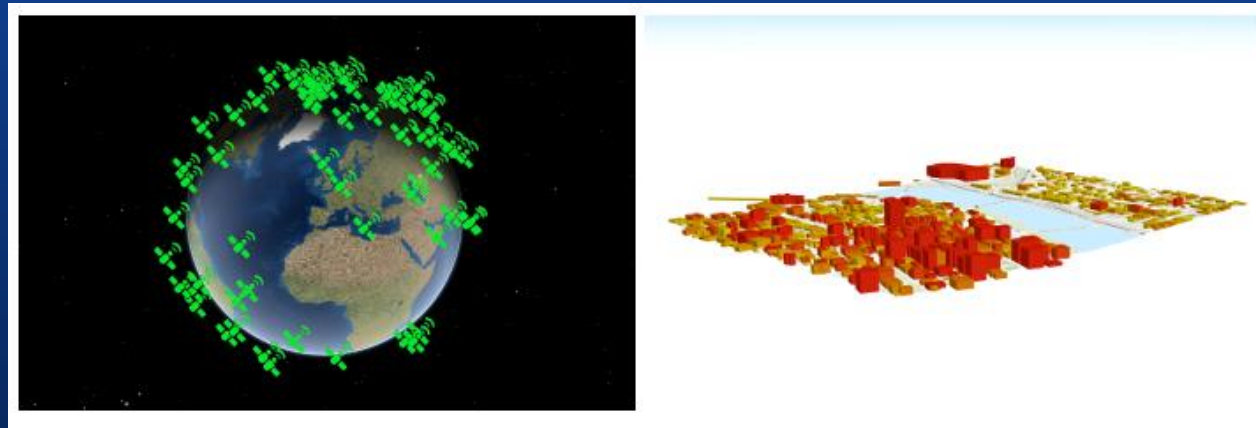
# des données

- **Layer Blend Mode**
- **Clustering avancé**



# De la 2D vers la 3D : concepts

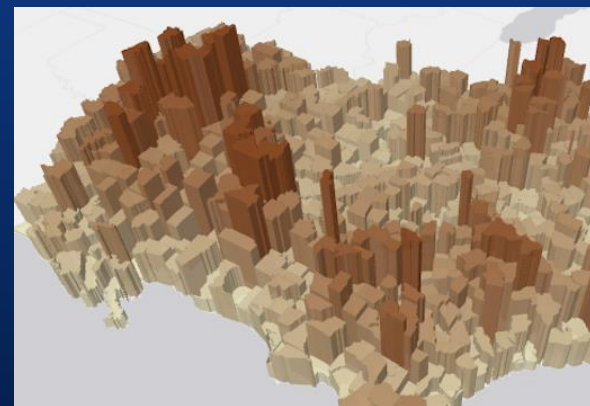
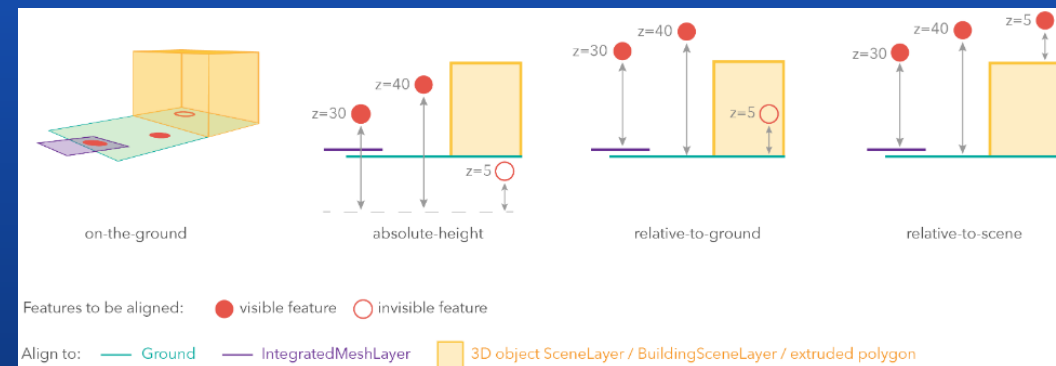
- Mode de visualisation :
  - **Global** : terre représentée comme une sphère, WGS84 ou WebMercator
  - **Local** : terre sur une surface plane, système de coordonnées projetées



- Camera : propriété de la SceneView
  - Position, tilt, heading
  - Mise à jour avec `.clone()` et `view.goTo()`

# De la 2D vers la 3D : concepts

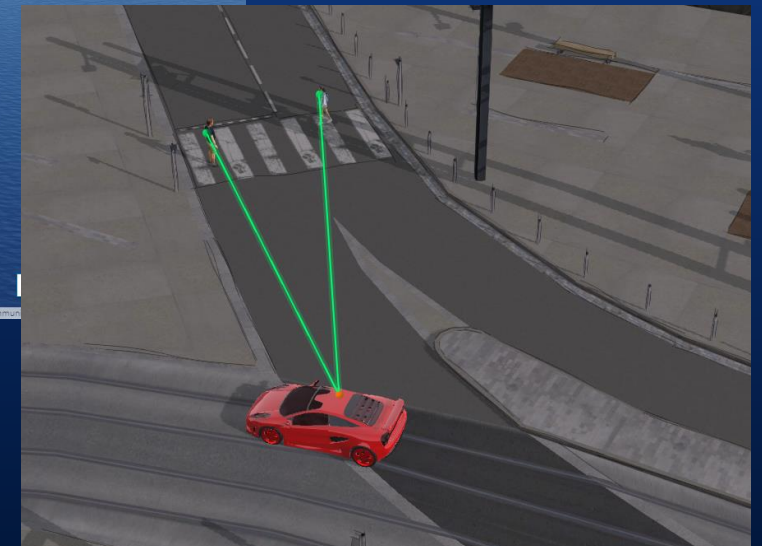
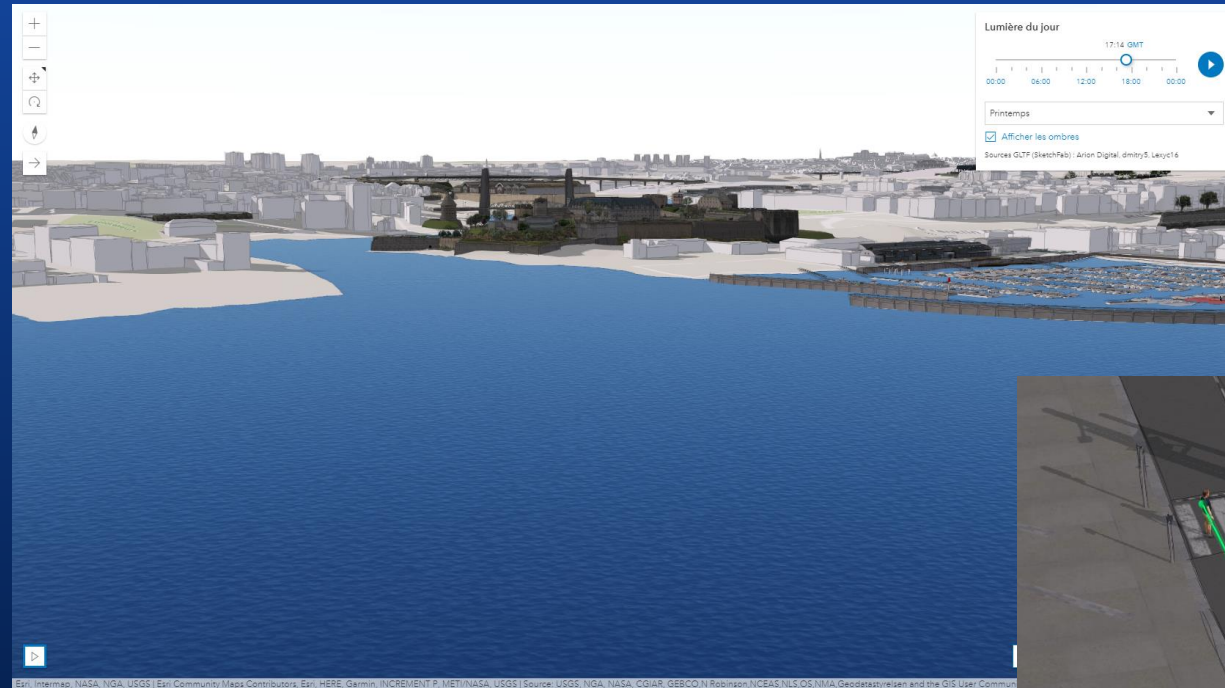
- Informations d'élévation:
  - 4 modes : sur le sol, absolu, relatif sol/scène
  - pour les données 2D ou 3D
- Extrusion :
  - Type de symbologie ExtrudeSymbol3DLayer
  - Représente des polygones en volume 3D
- Environnement :
  - luminosité : jour, heure, ombre, eau
  - atmosphère, arrière-plan, étoiles





# Démo 3D

- Environnement
- Scene
- Camera
- Object3DSymbol
- Line of sight



# Démo 3D BIM

- Revit BuildingSceneLayer
- Filter widget
- Slice Widget
- Measure



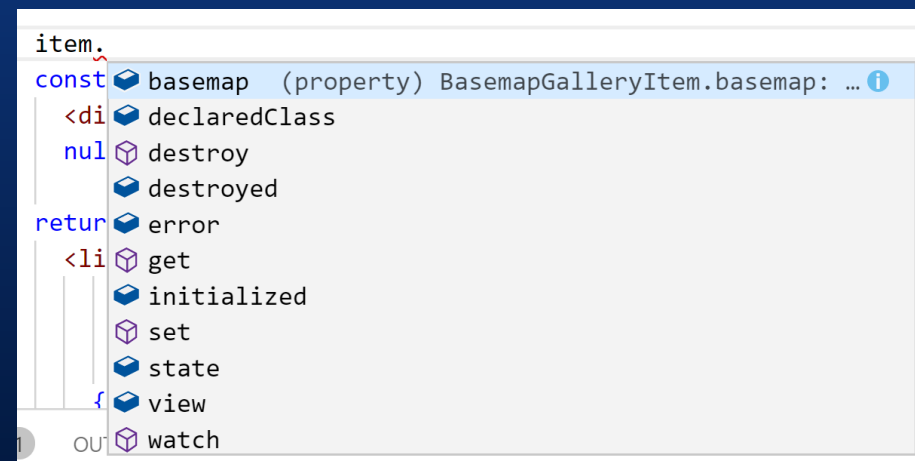
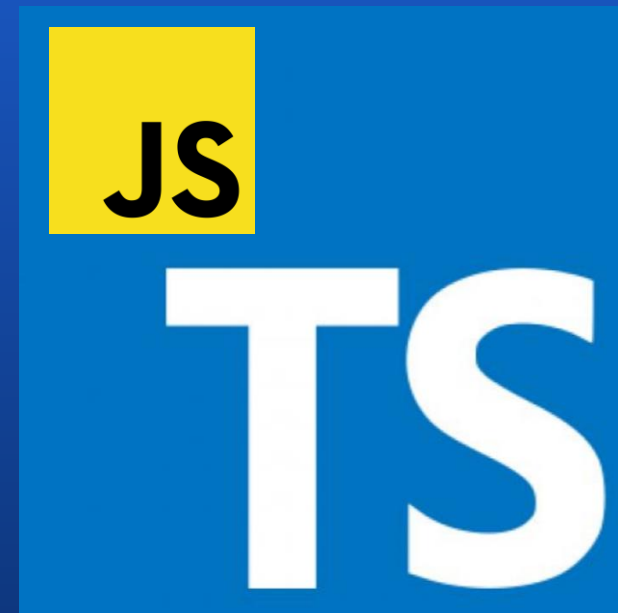
# TypeScript

*Typed JavaScript at Any Scale.*

*TypeScript extends JavaScript by adding types.*

*By understanding JavaScript, TypeScript saves you time catching errors and providing fixes before you run code.*

*Any browser, any OS, anywhere JavaScript runs.  
Entirely Open Source..*



# TypeScript

- Pourquoi l'utiliser ?
  - Ajout du typage à JavaScript
  - Intégration aux IDE
  - Utilisation des dernières capacités de JavaScript
  - Réduit le risque de bugs
  - Augmente la productivité
  - Microsoft planifie un support long terme de TypeScript
  - Valide JS -> Valide TS (simple integration)
  - Adhérence aux spécifications JS

```
import MapView from 'esri/views/MapView';
import Map from 'esri/Map';

const myMap: Map = new Map();

const loadMap = (container: HTMLDivElement, map: Map): MapView => {
  const mapView = new MapView({
    map,
    container,
  });

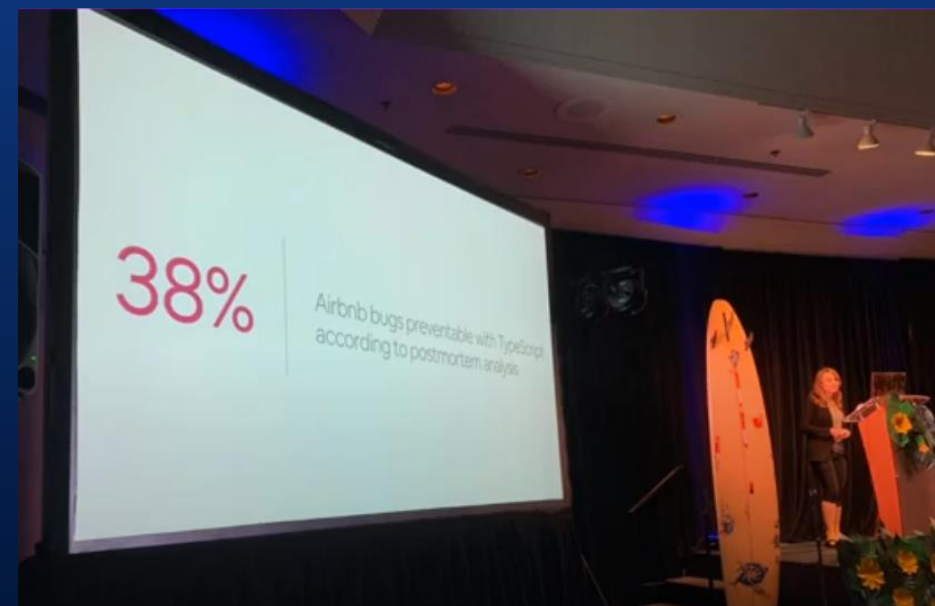
  return mapView;
};

loadMap('maDiv', myMap);
```

Argument of type '"maDiv"' is not assignable to parameter of type 'HTMLDivElement'. ts(2345)

[Peek Problem](#) No quick fixes available

```
const loadMap = async (container: HTMLDivElement): Promise<MapView> => {
  if (container) {
    const mapping = await import('../data/map');
    return mapping.initialize(state.container);
  }
};
```





# TypeScript

- + de 99% de l'API écrite en TypeScript
- installation recommandée avec node.js et npm

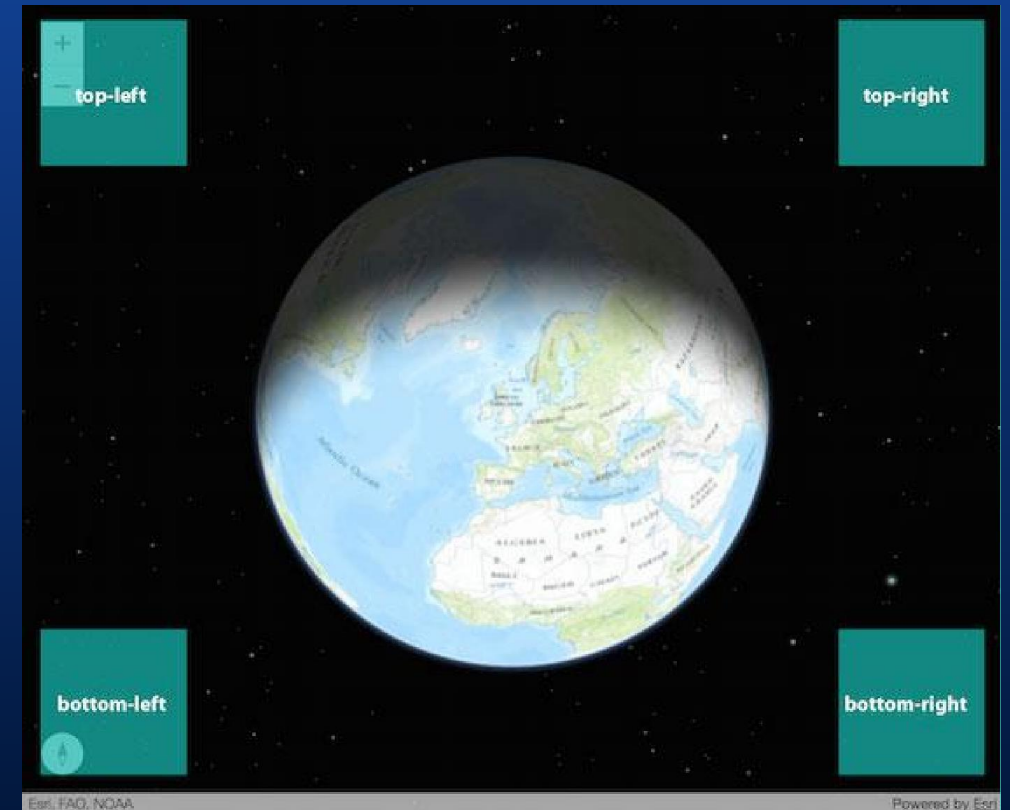
*npm install -g typescript*

*npm install --save @types/arcgis-js-api*

*npm install --save dojo-typings*

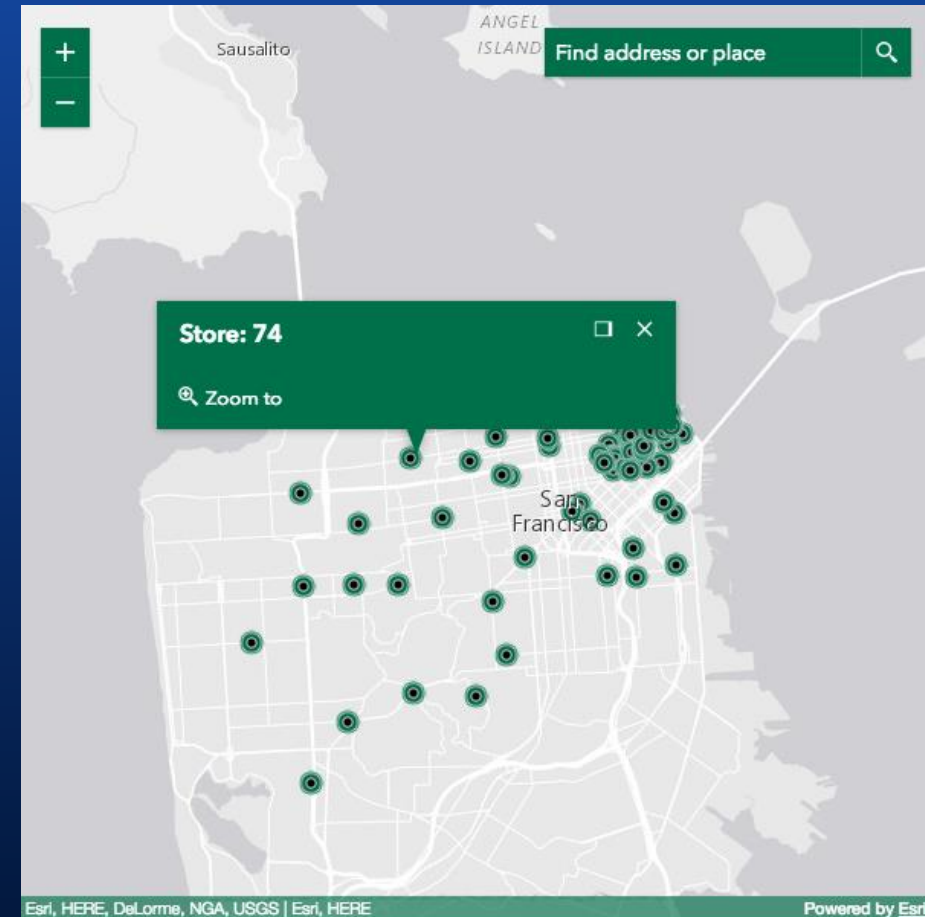
# Widget

- Composant pour interagir avec la carte
- Modèle vue (view) et contrôleur (viewModel)
  - Séparation des responsabilités
  - Création, extensibilité
  - Facilité de maintenance
  - Intégration de framework
- `esri/widgets/Widget`
  - Cycle de vie
    - `constructor`
    - `postInitialize`
    - `render`
    - `destroy`



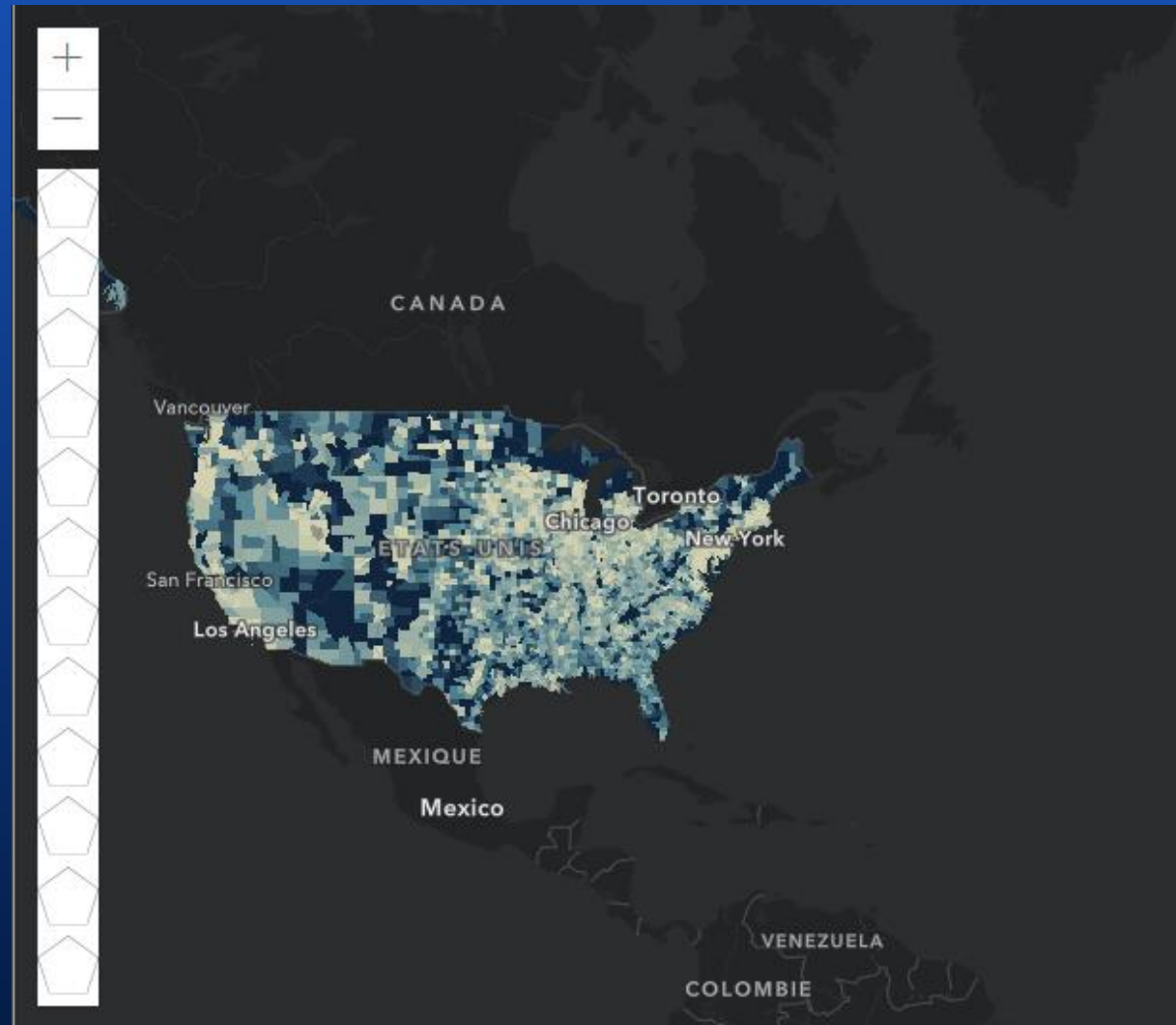
# Widget stratégies de customisation

- **Modifier le thème d'un widget**
  - Thème : changer style : size, color, font...
- **Modifier la présentation d'un widget**
  - Nouvelle vue
- **Recréer une vue**
- **Etendre une vue**
  - Surcharger une fonctionnalité
  - Ajouter une fonctionnalité



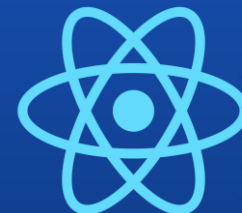
# Démo Widget

- Custom widget
- New widget



# Intégration dans les frameworks

- Désir d'utiliser un autre framework que Dojo
  - Fonctionnalités, méthode
  - Souvent accompagné **d'outils de développement & build**
  - Documentation et écosystème riche
- Deux solutions pour charger les modules
  - **esri-loader**: librairie qui permet de charger des modules à partir d'une version hébergée (web ou local) de l'API ArcGIS pour JavaScript.
  - **Plugin Webpack**: un loader AMD léger pendant le processus de bundle de Webpack.



# Intégration dans les frameworks : plugin webpack

- plugin développé par esri : <https://github.com/Esri/arcgis-webpack-plugin>
  - API JS 4.7+
  - Basé sur dojo-webpack-plugin
  - Loader AMD pendant le build
  - Configurable (loader d'asset, exclusion de module...)
- Quand l'utiliser ?
  - Webpack et personnalisation de la configuration
  - Version de ArcGIS API for JavaScript > 4.7
- Comment ça fonctionne ?
  - npm install --save-dev @arcgis/webpack-plugin
  - Loader externe pour les workers à renseigner

```
plugins: [  
  new CleanWebpackPlugin(),  
  
  new ArcGISPlugin({  
    features: {  
      '3d': true,  
      userDefinedExcludes: [ ...  
    ],  
    has: {  
      'esri-native-promise': true,  
    },  
  }),  
  
  new HtmlWebPackPlugin({ ...  
  })  
]
```

## Intégration dans les frameworks

Démo Webpack preact : faire rapidement une pwa

# Intégration dans les frameworks : esri-loader

- Librairie développée par esri : <https://github.com/Esri/esri-loader>

- API JS 3.x ou 4.x, choix de la version de l'API
- “Lazy Loading” par défaut
- Utilisation de Promise
- IE 11 non supporté

```
function createScript(url) {  
  const script = document.createElement('script');  
  script.type = 'text/javascript';  
  script.src = url;  
  script.setAttribute('data-esri-loader', 'loading');  
  return script;  
}
```

- Quand l'utiliser ?

- Outils de build/loader issus des frameworks CLI
- Rendu côté serveur
- Dans tous les cas ou webpack n'est pas utilisable

- Comment ça fonctionne ?

- Chargement du css = ajout d'un tag <style>
- Chargement des modules = ajout d'un tag <script>

```
// create a script object whose source points to the API  
script = createScript(url);  
// _currentUrl = url;  
// once the script is loaded...  
handleScriptLoad(script, () => {  
  // update the status of the script  
  script.setAttribute('data-esri-loader', 'loaded');  
  // return the script  
  resolve(script);  
}, reject);  
// load the script  
document.body.appendChild(script);
```



# Démo esri-loader

- Github Jobs API integration
- Server-side render

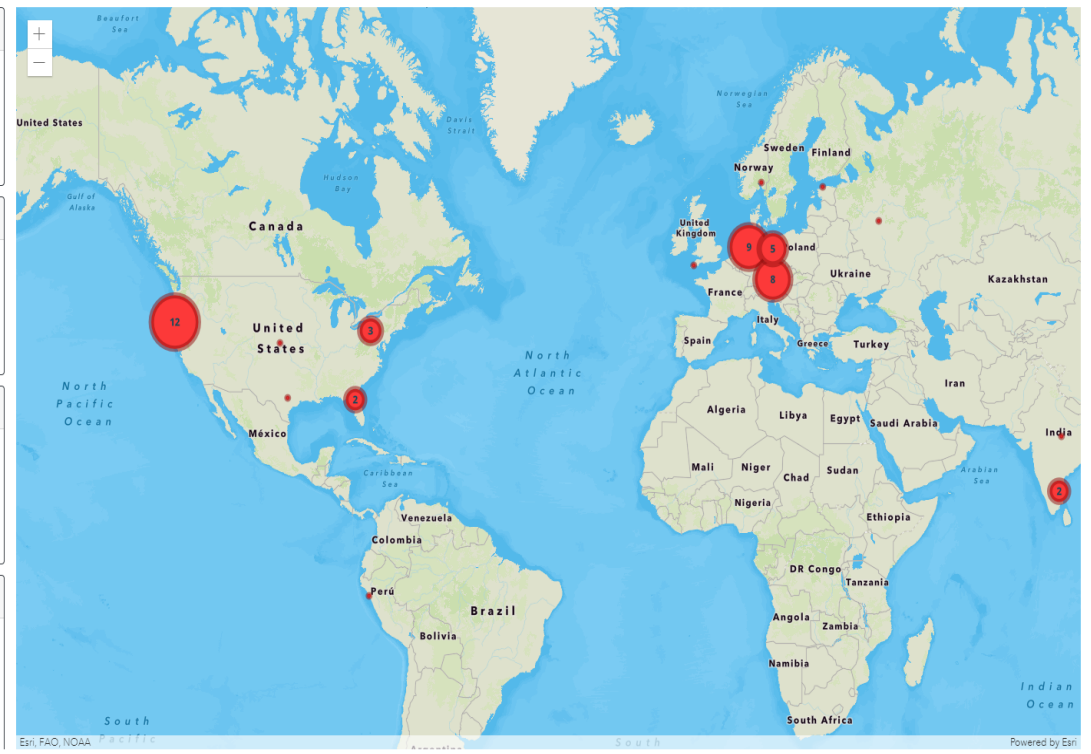
Démonstration esri-loader : rendu côté serveur

Slite
Senior Growth Engineer
Lieu : Remote
<a href="#">Référence de l'annonce</a>

Lenus eHealth
Senior Web Developer
Lieu : Copenhagen
<a href="#">Référence de l'annonce</a>

Compute Software
Senior Software Engineer (Clojure Required)
Lieu : USA
<a href="#">Référence de l'annonce</a>

iPaper
Senior Frontend Developer
Lieu : Aarhus/ Århus
<a href="#">Référence de l'annonce</a>



# Calcite Maps / Bootstrap

- **Calcite Maps**

- Theme pour la creation d'apps modernes
- Map app framework + components + styles
- Calcite - themes, colors, fonts, icons...
- ArcGIS JS 3.x, 4.x, Esri-Leaflet
- Desktop and mobile

- **Application style-explorer**

- Visualiser différents styles avec vos données
- Inspecter le HTML et récupérer les classes CSS pour votre application

- <https://esri.github.io/calcite-maps/samples/index.html>

- <https://esri.github.io/calcite-maps/samples/style-explorer/index.html>

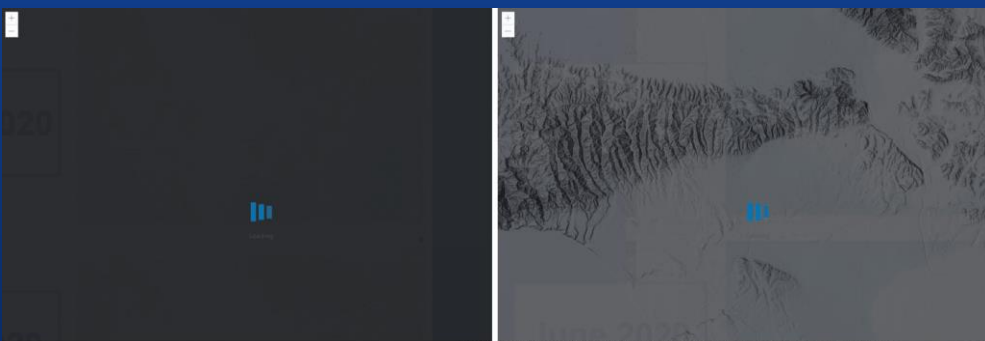


# Nouveautés (4.17) – octobre 2020

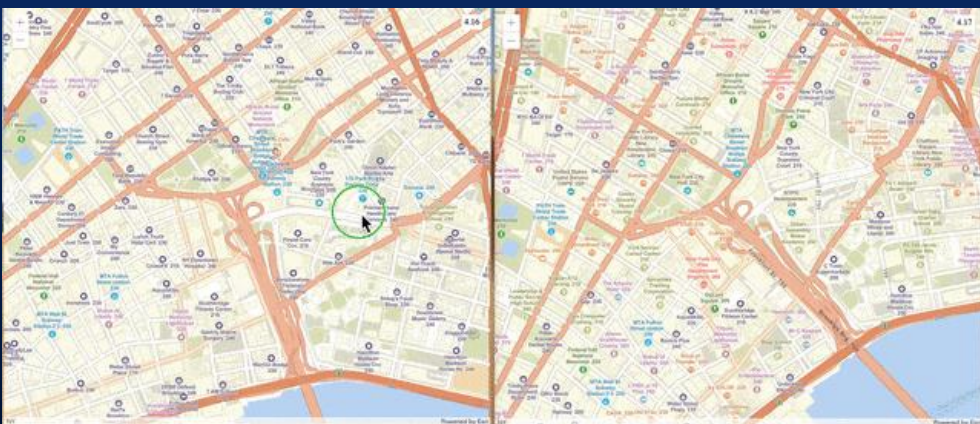
## Améliorations des performances

- Chargement et prétraitement FeatureLayer
- CSVLayer , GeoJSONLayer et FeatureLayer : stockage des entités créés côté client dans un worker
- SceneLayer : prise en compte version 1.7 de la spécification I3S

<https://www.ogc.org/standards/i3s>

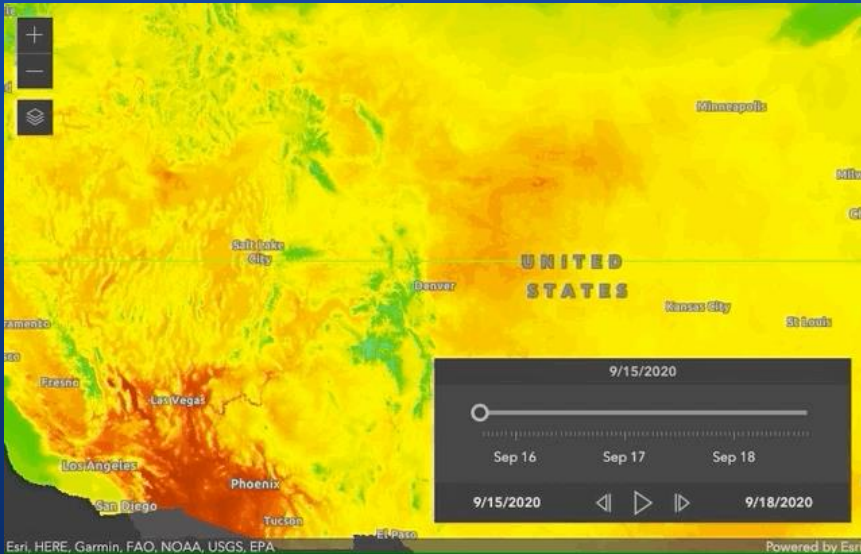


## Expérience améliorée pour VectorTileLayer et LayerView



# Nouveautés (4.17) – octobre 2020

WMSLayers prise en charge du temps



Symbol remplissage en 3D



Autres nouveautés ...

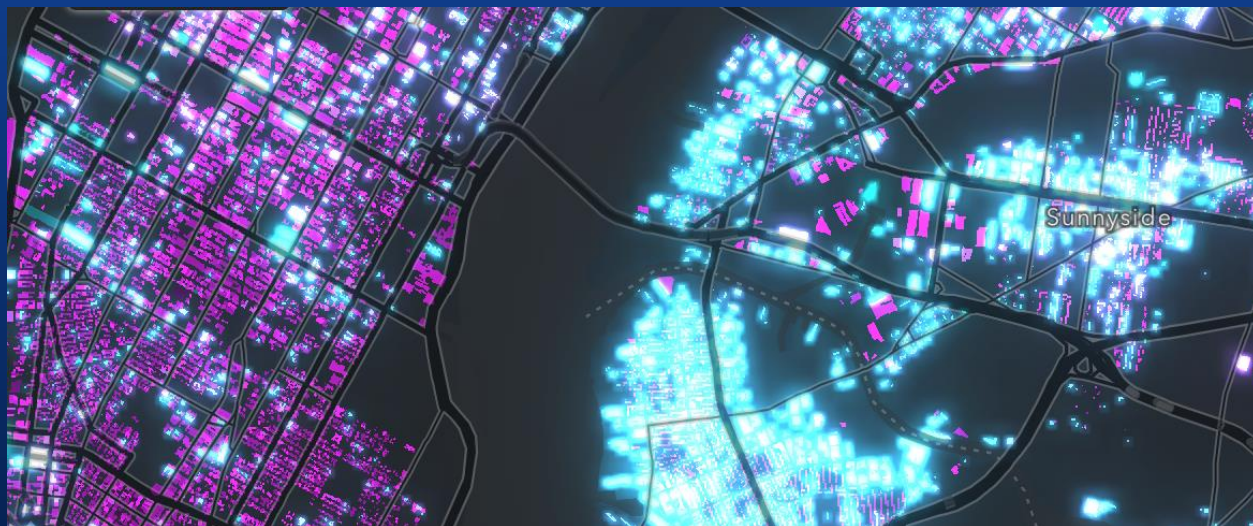
<https://developers.arcgis.com/javascript/latest/guide/release-notes/>



# RoadMap

## Spécifique

- Support du snapping
- Ajout d'effets sur les couches (drop shadow, grayscale, saturation...)



- Ajout du support des item feature collections
- Mise à jour du Direction Widget pour intégration dans la MapView

# RoadMap

## Général

Amélioration des performances

- WebGL Rendering

Modernisation de l'API pour une meilleure productivité :

- remplacer Dojo par du JS natif car les navigateurs disposent maintenant de :

- WEBGL pour l'affichage graphique
- Native promises
- Internationalisation

<https://developers.arcgis.com/javascript/latest/guide/localization/>

- Passage sur Typescript

Simplifier la façon dont l'api est consommée

- Permettre à l'api d'être prise en compte par n'importe quel outil de Build nativement et sans intermédiaire
- ES 6 Modules arrivent en option supplémentaire de AMD modules



# Ressources

## Projets Esri GitHub

- <https://github.com/Esri>

## Migration

- <https://developers.arcgis.com/javascript/latest/guide/migrating/>
- <https://developers.arcgis.com/javascript/latest/guide/functionality-matrix/>

## Tutoriels en ligne

- <https://developers.arcgis.com/javascript/>

## esri-loader

- <https://developers.arcgis.com/javascript/latest/guide/esri-loader/>
- <https://github.com/Esri/esri-loader>

## Webpack

- <https://developers.arcgis.com/javascript/latest/guide/webpack/>
- <https://github.com/Esri/arcgis-webpack-plugin>

## Next API

- <https://github.com/Esri/feedback-js-api-next>
- <https://js.arcgis.com/next>

**Merci...**

**Questions / Réponses**

