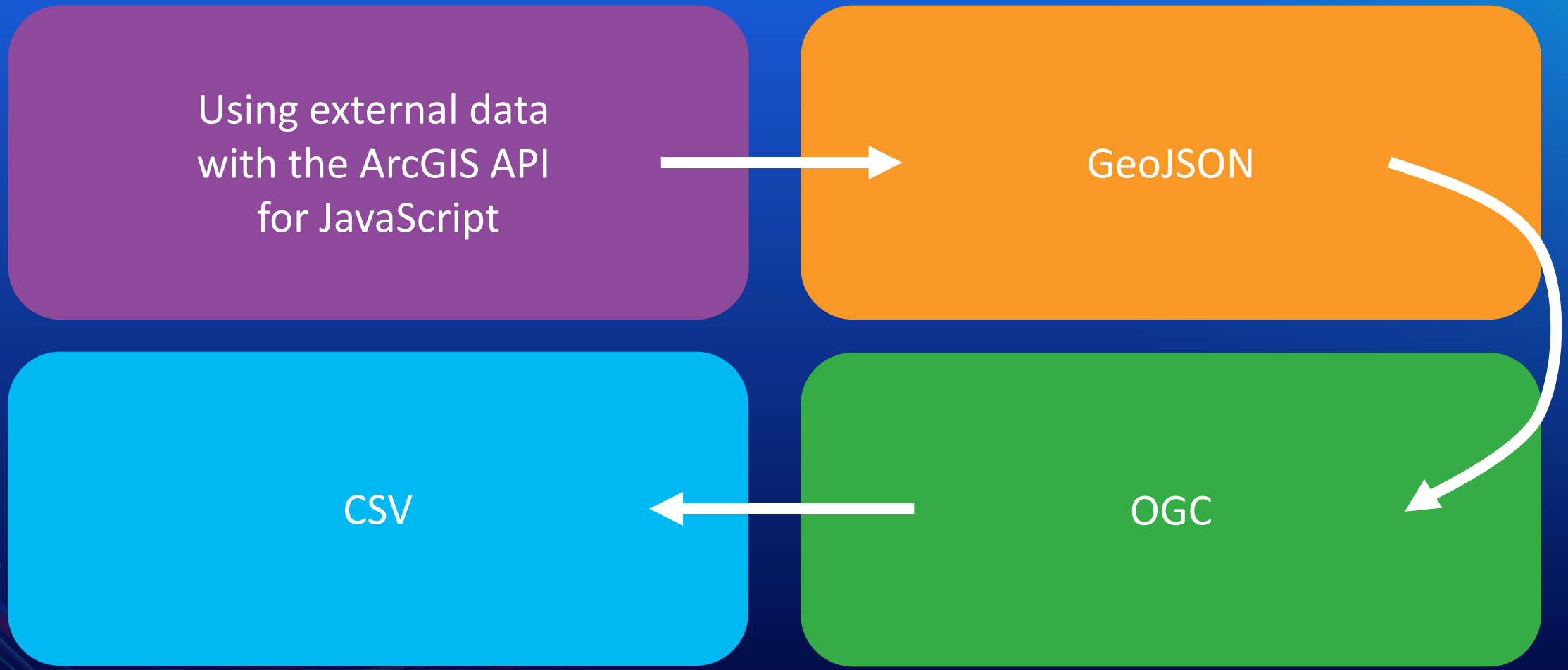
Building Web Apps Using your GeoJSON, CSV, and OGC data

Anne Fitz and Jose Banuelos

2021 ESRI
DEVELOPER SUMMIT

Agenda


Using external data with the ArcGIS API for JavaScript
GeoJSON
OGC
CSV

Using external data with the ArcGIS API for JavaScript

Working with external data sources

Additional types of data and files are directly supported by specific subclasses of `Layer`. These include specific types of layers for working with external files like CSV or GeoJSON files or integrating external services such as Bing Maps.

Layer Subclass	Data Source	Data Types	Features	Limitations
`CSVLayer`	CSV files	- Vector graphics downloaded as points	Client-side processing, popup templates, renderers with 2D and 3D symbols	May require large download depending on the number of features
`GeoRSSLayer`	GeoRSS feed	Vector graphics as points, polylines, and polygons	- Graphics storage - Popup templates	- No 3D support No support for renderers
`GeoJSONLayer`	GeoJSON file	Vector graphics as points, polylines, and polygons	Create layer from GeoJSON data	- Each GeoJSON Layer accepts a single geometry type - Data must comply with RFC 7946 specification

- Different subclasses of Layer based on the data type

```
// typical programming pattern for working with external data
const layer = new Layer({
  url: "url to your external data",
})
```

GeoJSON

Jose Banuelos

History

- Published in 2008
- Replaced in 2016 (RFC 7946)

Features

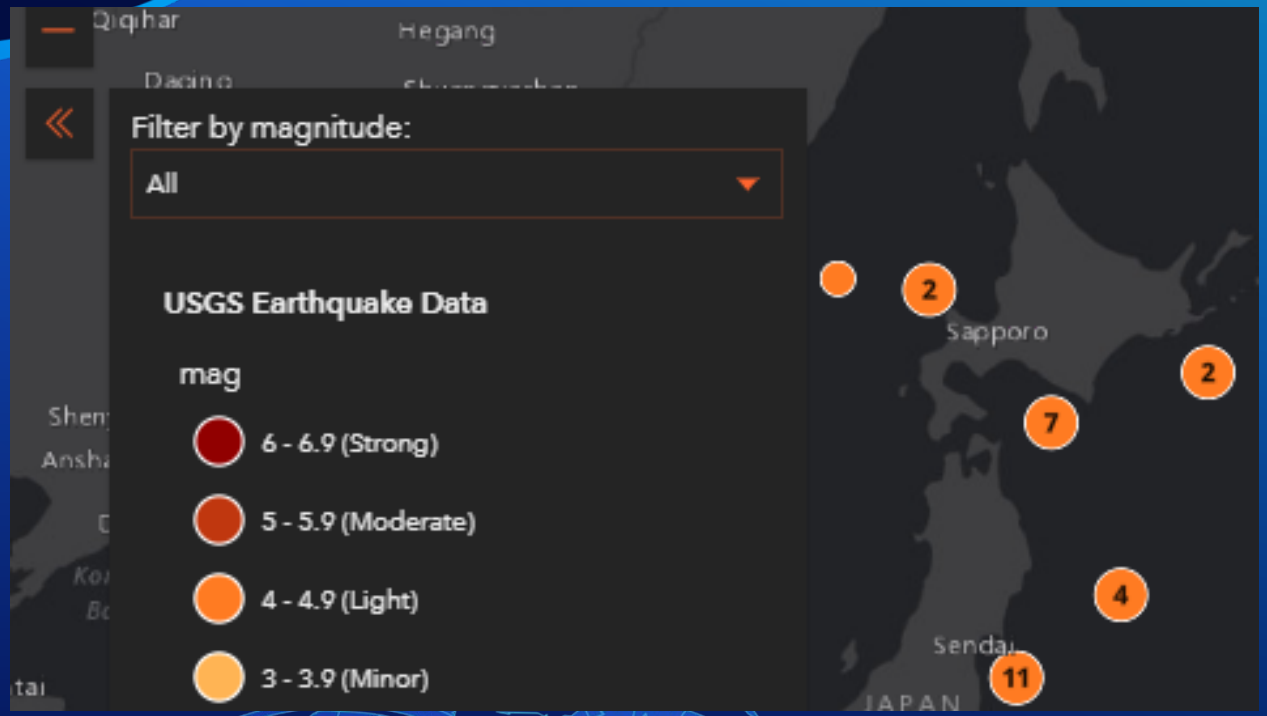
- Geometry, a feature, or a collection of features.
- World Geodetic System 1984 (WGS84) datum
- GeoJSON fixed types

GeoJSON Geometry Object	API Geometry Type
Point	Point
MultiPoint	Multipoint
LineString/MultiLineString	Polyline
Polygon/MultiPolygon	Polygon

The ArcGIS API for JavaScript GeoJSONLayer

- Features
  - Clustering
  - Client-side querying
  - Widgets
  - Projection

```
const geoJSONLayer = new GeoJSONLayer({
const blob = new Blob([JSON.stringify(geojson)], {type: "application/json"});
const url  = URL.createObjectURL(blob);
const layer = new GeoJSONLayer({ url });
map.add(geoJSONLayer);  // adds the layer to the map
```



GeoJSONLayer

Jose Banuelos

OGC data

- WMS
- WMTS
- WCS
- OGC API Features

OGCFeatureLayer

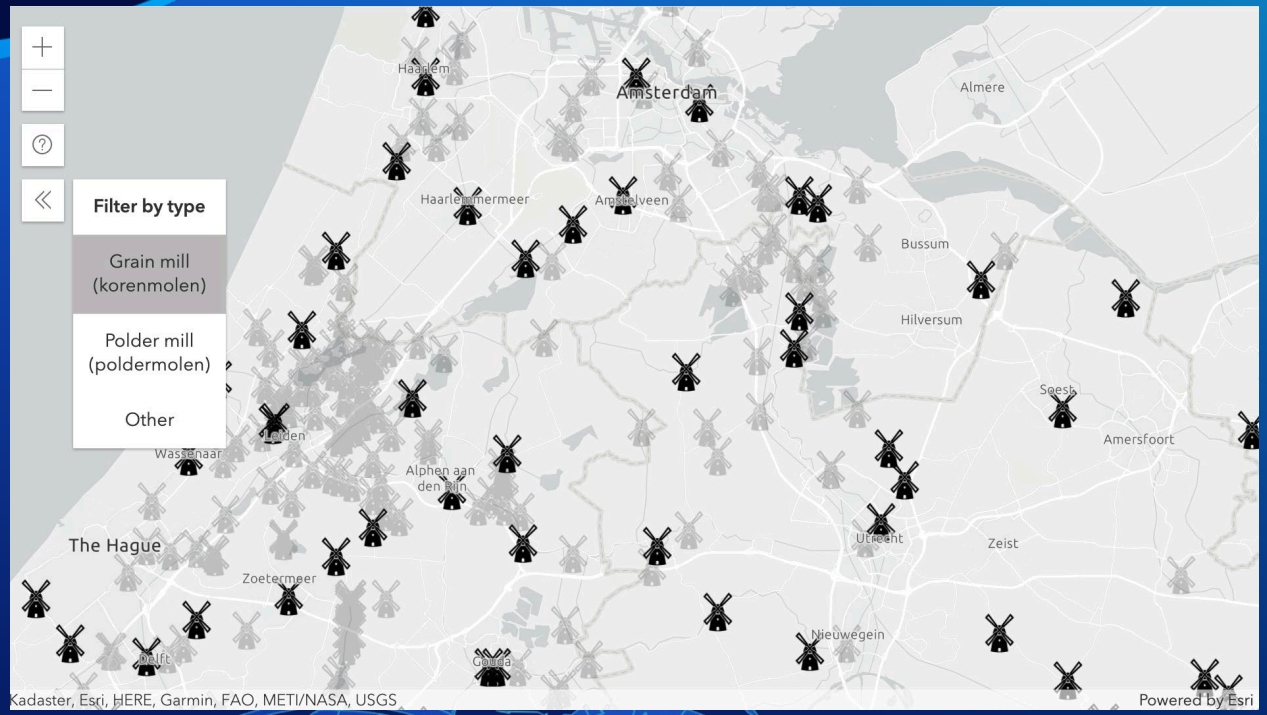
Based on an OGC API – Features service

- Geospatial data is stored in “collections”
  - Part 1 published October 2019
  - Previously referred to as WFS3
- Support added in our API at version 4.16 (Summer 2020)

```
// require["esri/layers/OGCFeatureLayer"]
const windmillsLayer = new OGCFeatureLayer({
  url: "https://demo.pygeoapi.io/stable", // url to the OGC service
  collectionId: "dutch_windmills" // unique id of the collection
})
```

Layer capabilities

- Rendering (visual variables)
- Labeling
- Popups
- Querying
- Clustering
- Layer blending

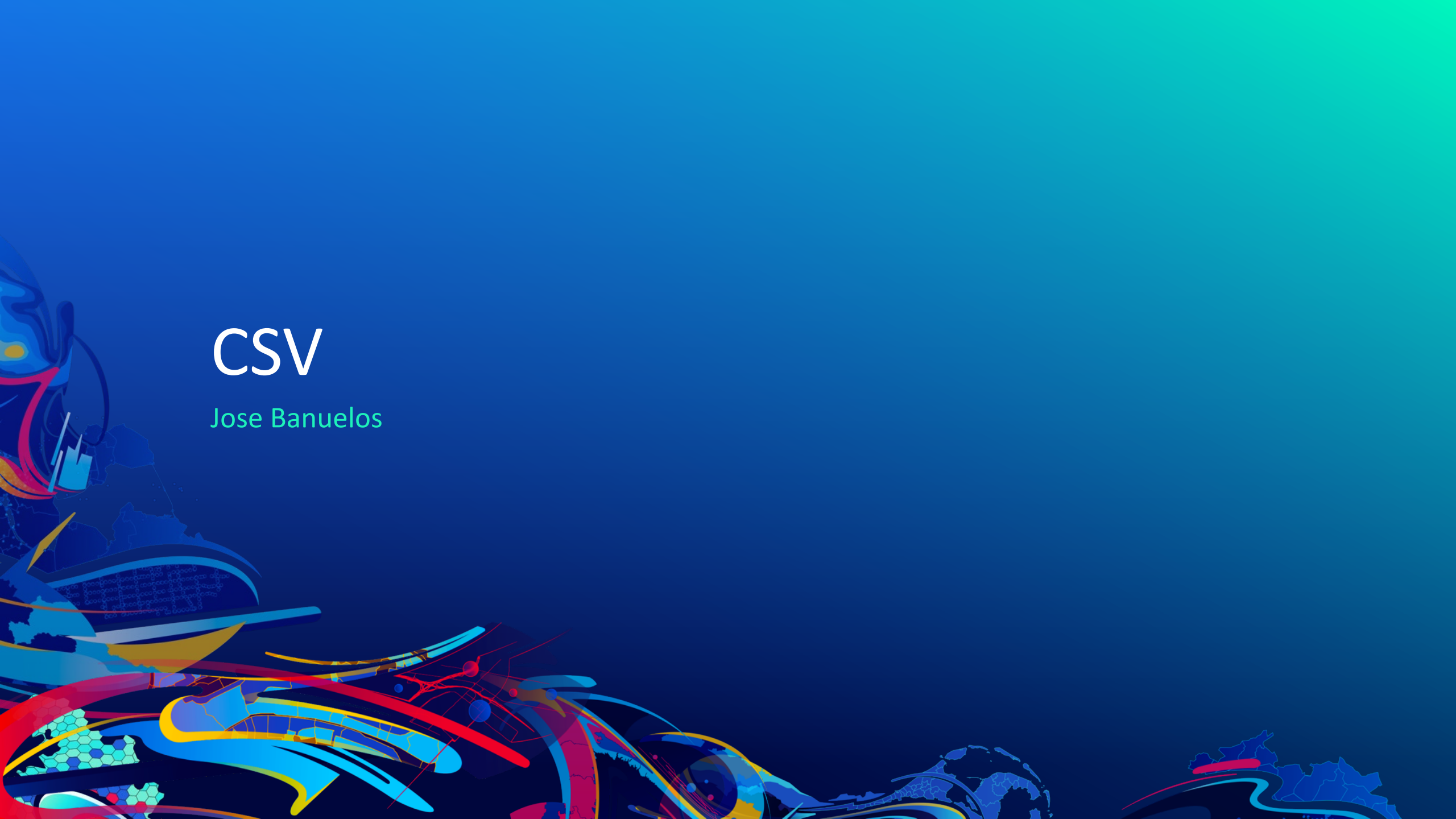


OGCFeatureLayer

Anne Fitz

CSV

Jose Banuelos



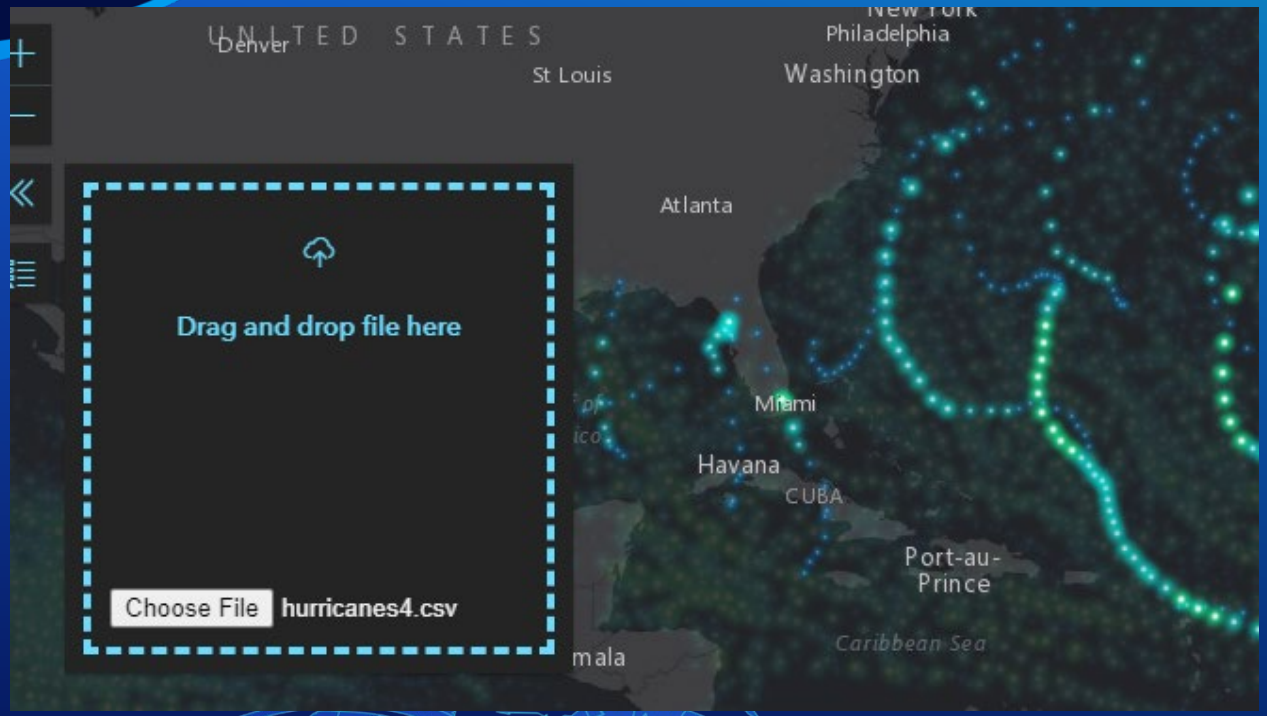
CSVLayer

- Lat / Long automatic detection
  - CSVLayer.latitudeField
  - CSVLayer.longitudeField

```
const csvLayer = new CSVLayer({
  url: "https://developers.arcgis.com/javascript
    /latest/sample-code/layers-csv/live
    /earthquakes.csv",
  copyright: "USGS Earthquakes"
});
```

CSVLayer Features

- Same as a GeoJSONLayer
- Upload a .csv file and display the data immediately



Demo Title

Presenter(s)

Demos and slides available at
https://arcg.is/0qne0G

Please provide your feedback for this session by clicking on the session survey link directly below the video.