Point Clouds and 3D Mesh

Sean William Morrish, ESRI Redlands

Thomas Other, ESRI R&D Center Zürich

Complementary Resource Email

(no marketing)

- A copy of the presentation
- Links to today's web demos
- Links to training materials

Agenda

1. 3D Data on the ArcGIS Platform
   - Introduction to different 3D Scene Layers
   - Indexed 3D Scene Layers
2. 3D Data on the Desktop
   - Data Acquisition & Processing
   - Data Publishing & Value Add
3. 3D Data on the Web
   - Data Viewing & Authoring
   - Current & Future Features

ArcGIS Platform

Indexed 3D Scene Layers —*Fundamentals*

- Stream data format designed to support 3D geospatial content
- Provides level of detail using advanced tree traversal algorithms
- Supports different layer types and data profiles
- Uses request scheduling to achieve optimal performance
- Open standard: I3S Spec

Indexed 3D Scene Layers —*Layer Types*

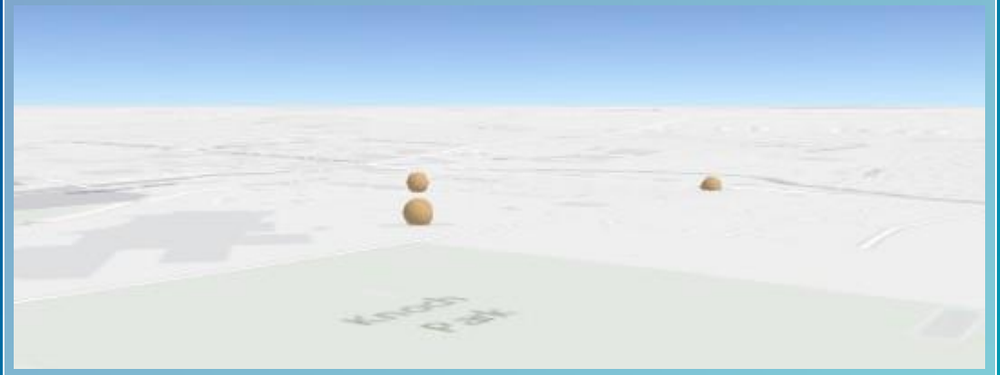
3D Objects

Multipatch

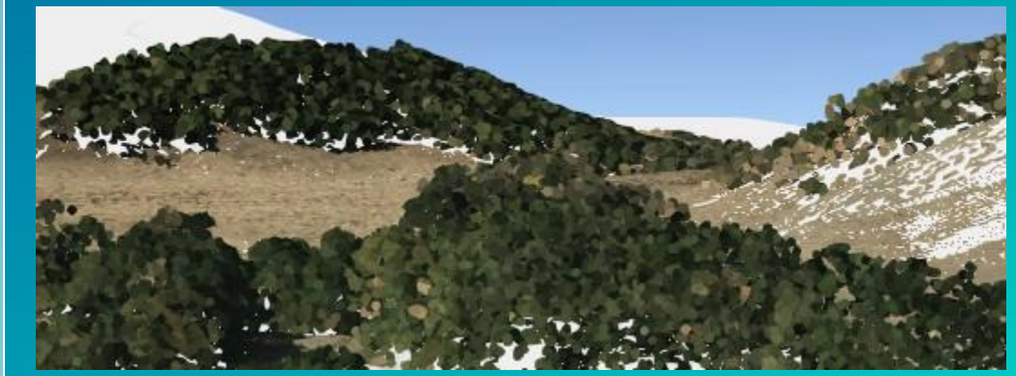
3D Integrated Mesh

VRICON / Drone2Map

3D Points

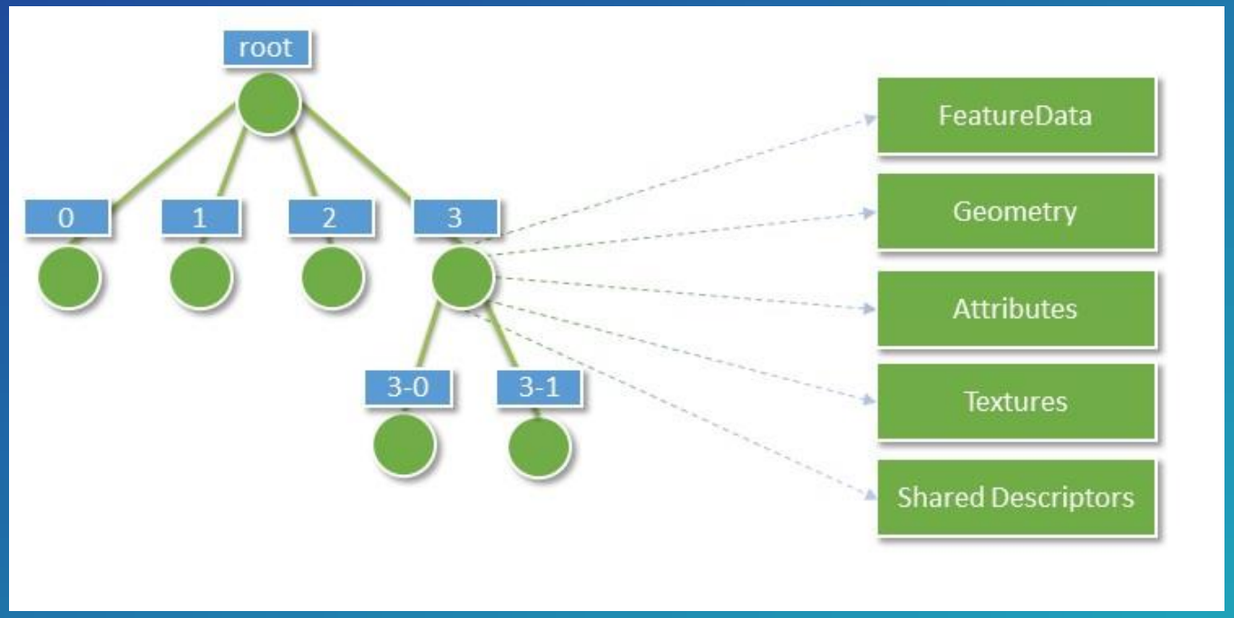


Point Clouds



LAS

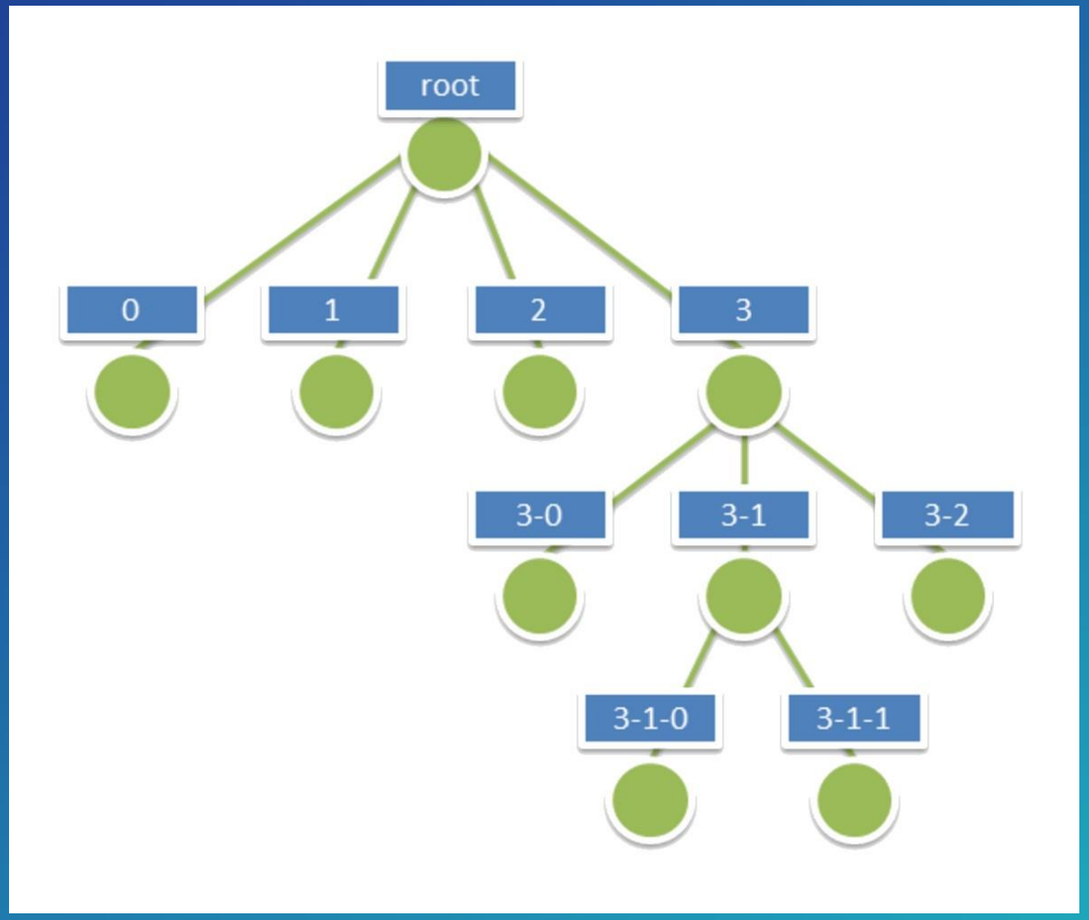
Indexed 3D Scene Layers—*Data Format*



- Data is clustered in LOD nodes
- Each node contains relevant data on
  - Features
  - Attributes
  - Geometry
  - Textures

Indexed 3D Scene Layers —*Level Of Detail*

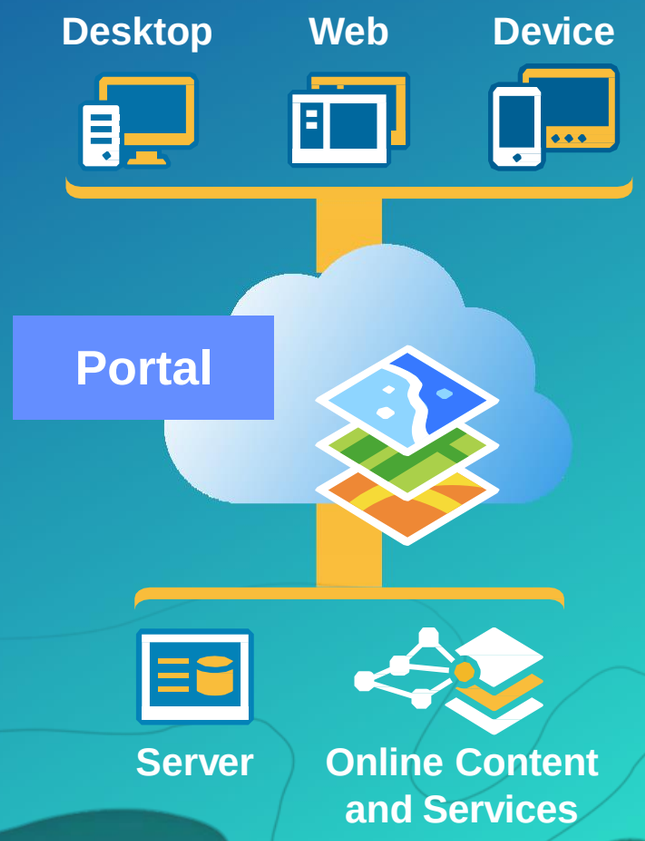
Node Structure



3D Data on the Desktop

3D GIS Platform

- Synthesize 2D and 3D in web **GIS** architecture
- Default Elevation Service
- Multiuse dynamic services across clients
- Securely manage large enterprise geodatabases
- Analyze across real-time and historical data



ArcGIS 3D Scene Layers

- 3D Objects Scene Layer
- 3D Point Scene Layer
- 3D Integrated Mesh Layer
- 3D Point Cloud Scene Layer
- Elevation Layer

Esri Indexed 3d Scene (*.I3s) and Scene Package (*.slpk) formats

Requirements for a 3D GIS visualization format:

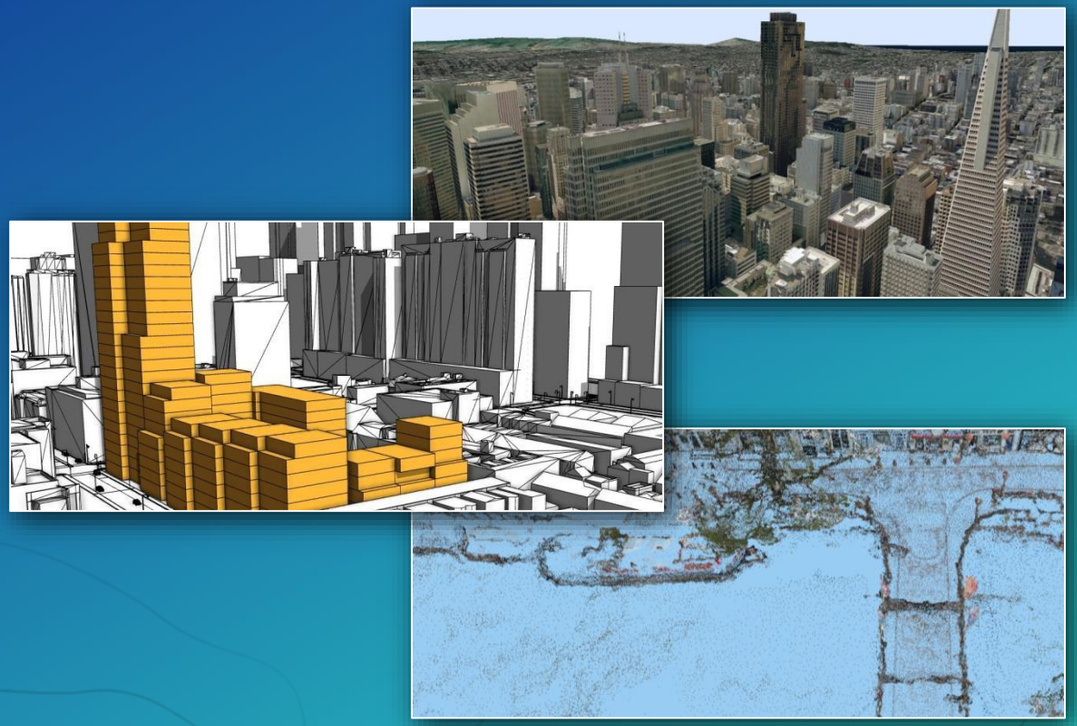
1. **Web friendly:** JSON + Typed Arrays
2. **Mobile friendly:** Works good with varying bandwidth
3. **Extensible:** Support different types of content
4. **Declarative:** Reduce required implicit knowledge
5. **Efficient:** Use spatial indexing for quick delivery
6. **Scalable:** Provide Level of Detail Support
7. **Protected:** Ensure that content is protected
8. **Open:** Full Specification publicly accessible

Now available @ https://github.com/Esri/i3s-spec

Content Profiles

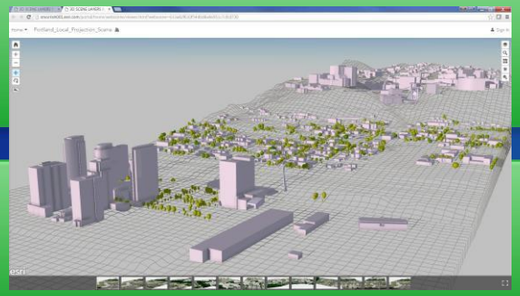
- Support different geometry types

  - Individual Features
    - Points
    - Multipatches
  - Integrated Meshes
  - Pointclouds
  - Point Symbols
  - Analytics

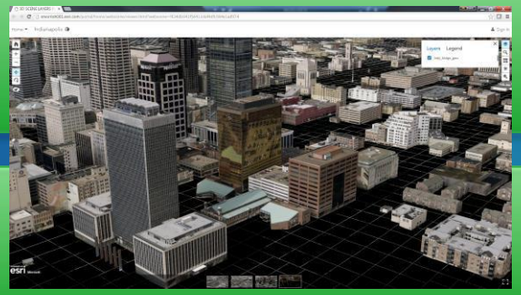


Parts of a Scene

Feature Service Layers
*(< 2000)



2D Layers
Draped or *Absolute**



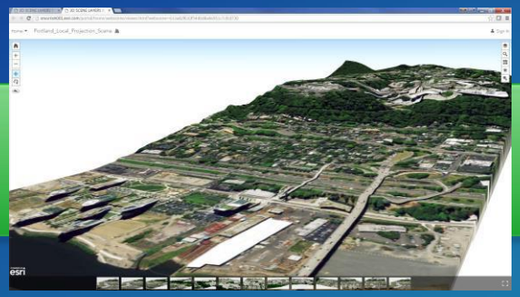
Feature Service Layers
*(< 2000)

Web Scene Layers
(i3S)

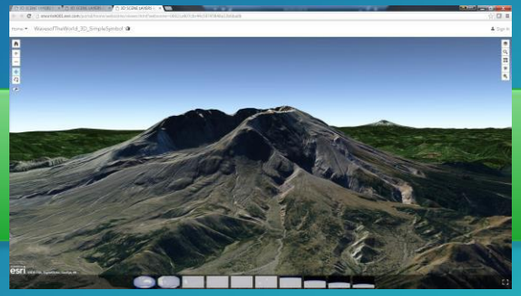
3D Layers
(Absolute Z')

Web Scene Layers
(i3S)

Your Local Imagery

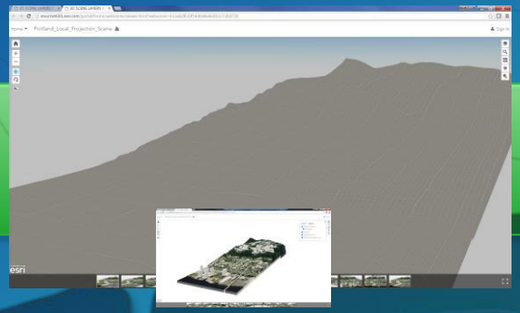


Draped Imagery

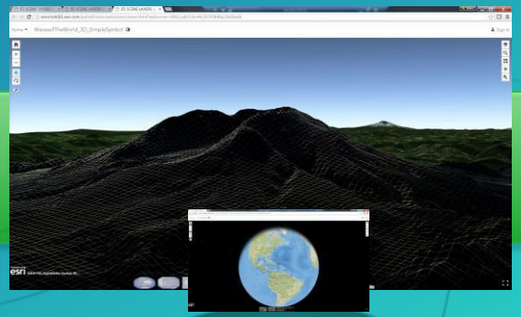


ArcGIS Online

Your Local Terrain



Terrain

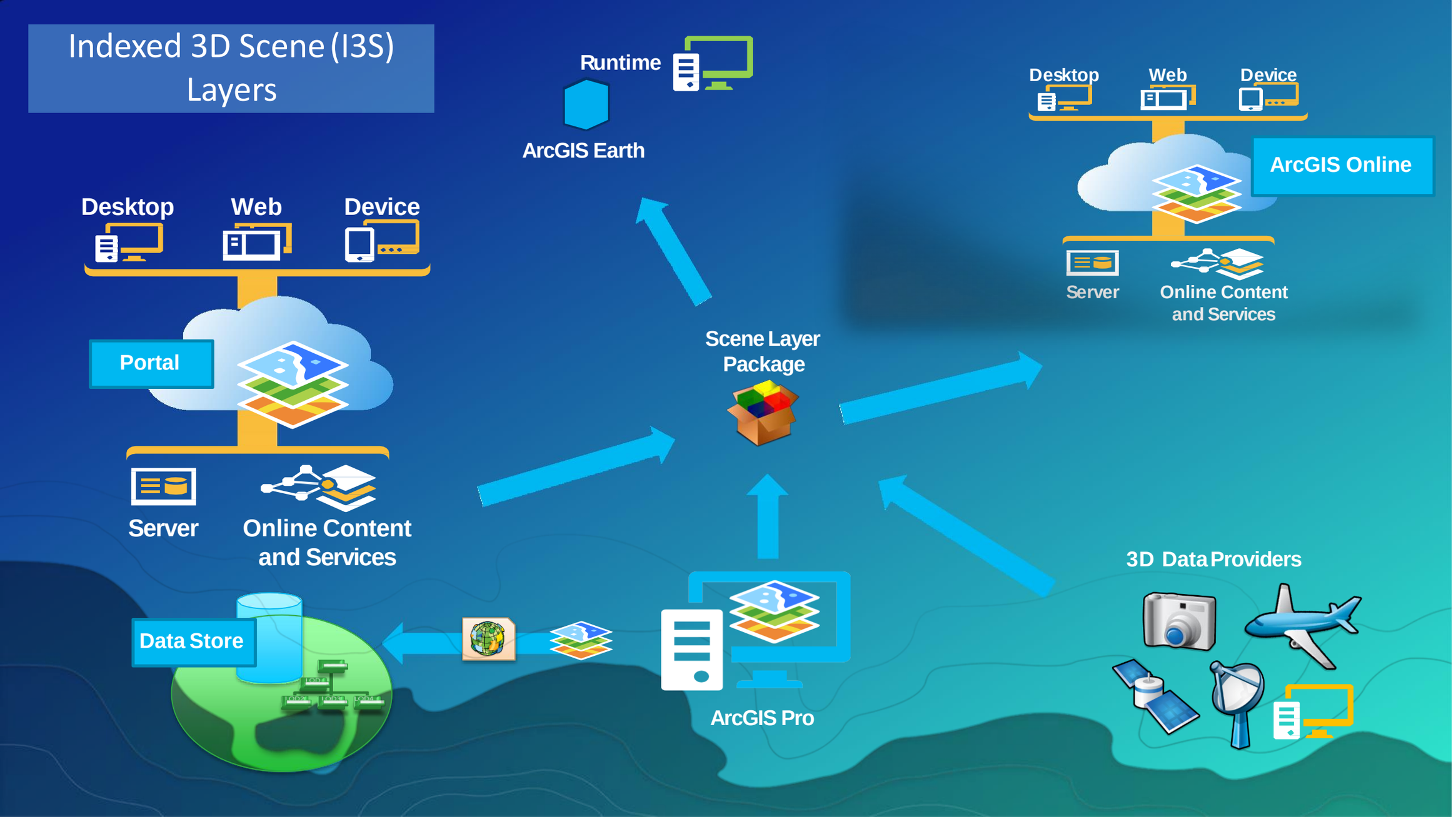


ArcGIS Online

Local Scene

Global Scene

Indexed 3D Scene (I3S) Layers



Point Clouds

What are Point Clouds

Integrated Mesh

What is an Integrated Mesh

Data Acquisition

SUBHEAD INFORMATION

Data Acquisition

Platforms for the collection of LiDAR and Integrated Mesh data.

Data Processing

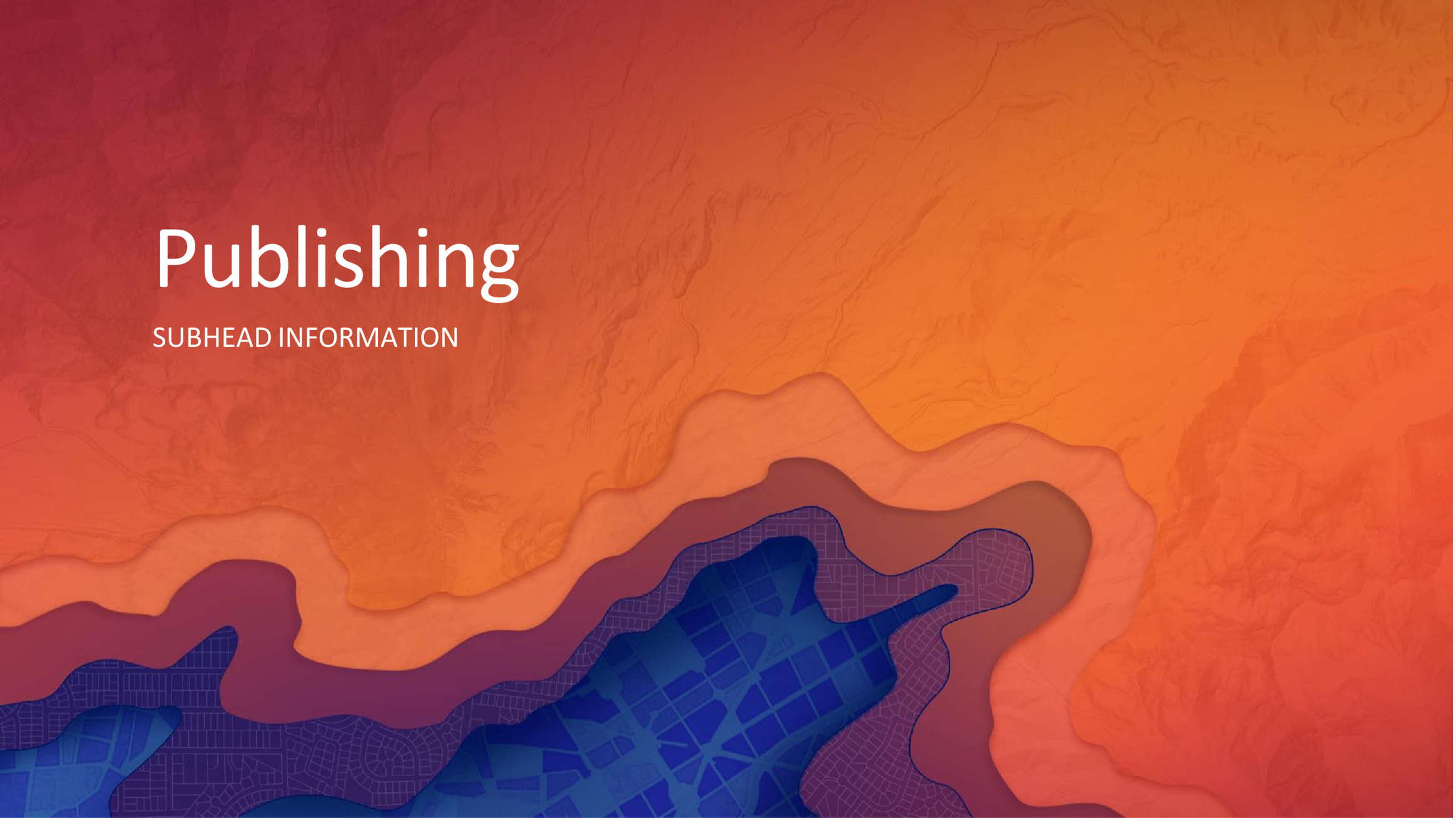
SUBHEAD INFORMATION

Data Processing

Workflows to process, create and analysis LiDAR & Integrated Mesh data.

Publishing

SUBHEAD INFORMATION

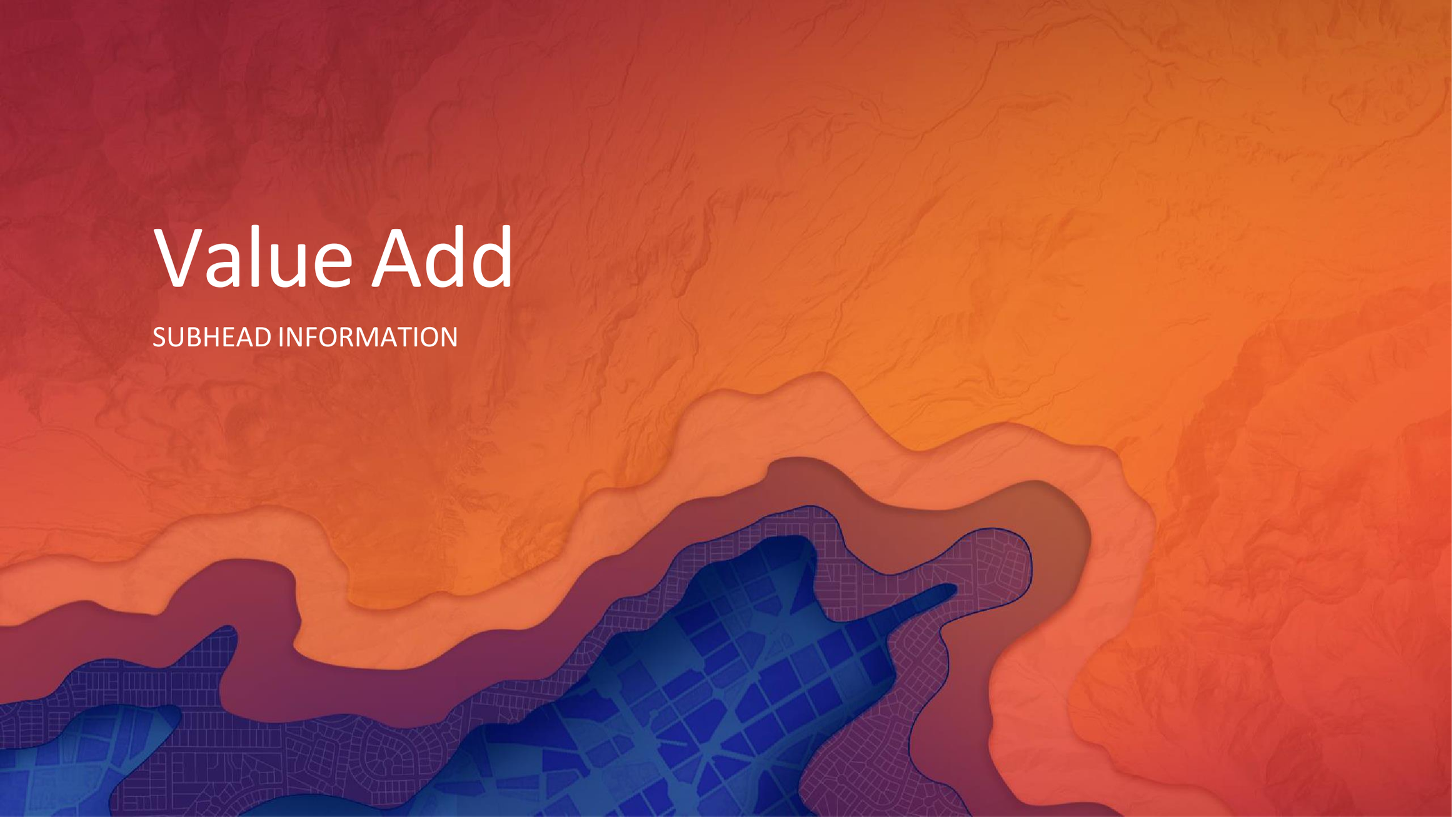


Publishing

Workflows for the sharing and publishing of LiDAR and Integrated mesh data.

Value Add

SUBHEAD INFORMATION



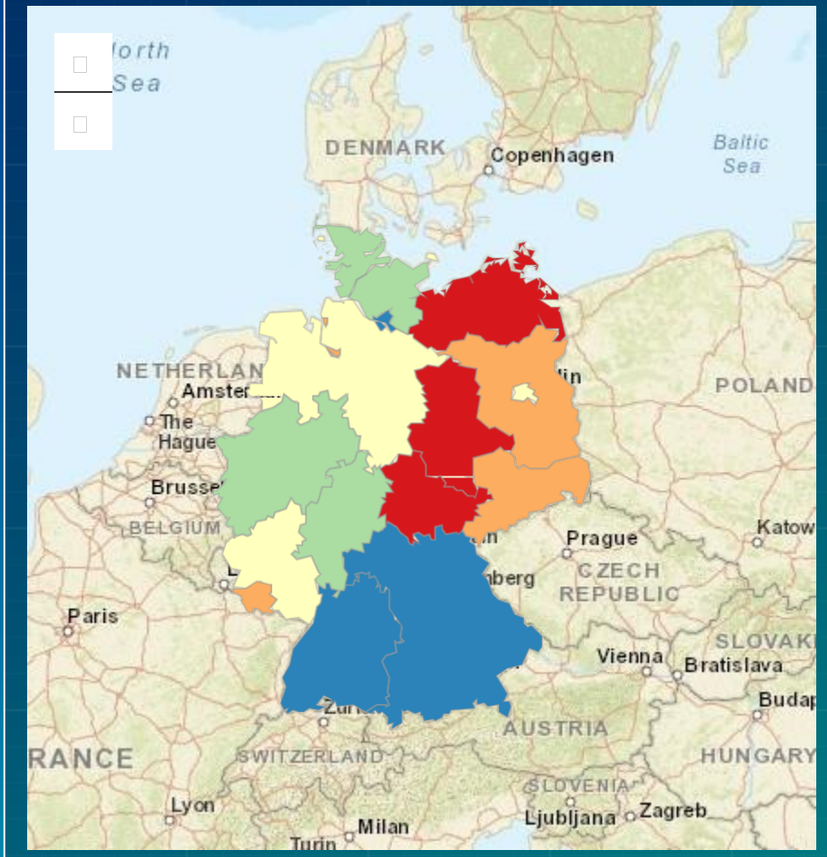
Added Value of LiDAR & Integrated Mesh

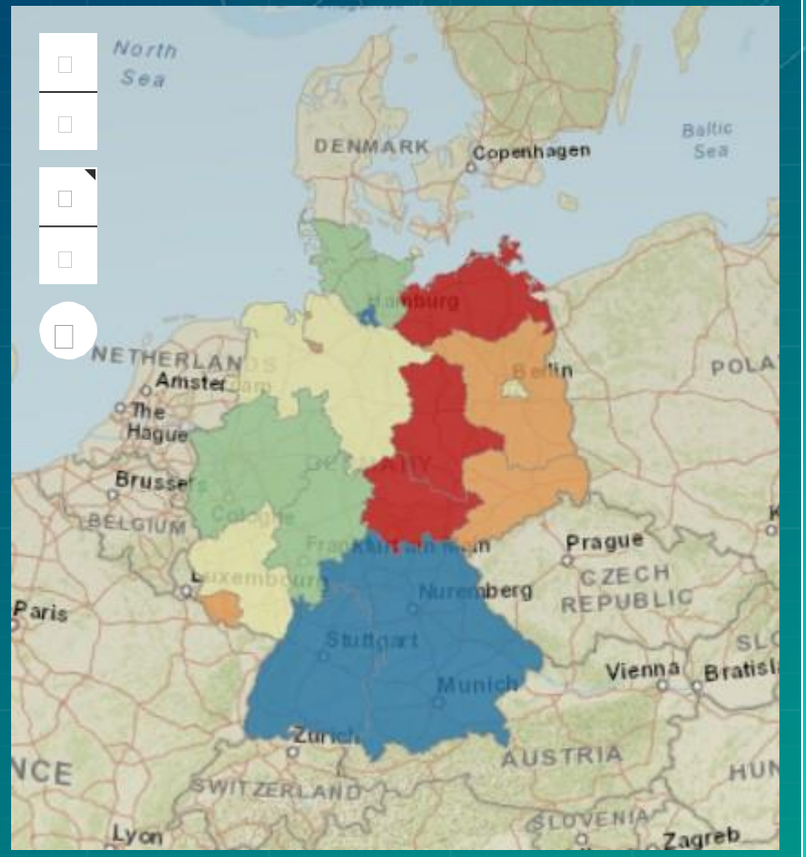
Added value of LiDAR & 3D Integrate Mesh to existing 3D.

3D Data on the Web

JavaScript API —*2D & 3D Viewing*

```
var map = new Map({
  basemap: "streets",

  layers: [new FeatureLayer(
    "...Germany/FeatureServer/0"
  )]
});

viewLeft = new MapView({
  container: "viewDivLeft",

  map: map
});

viewRight = new SceneView({
  container: "viewDivRight",

  map: map
});
```



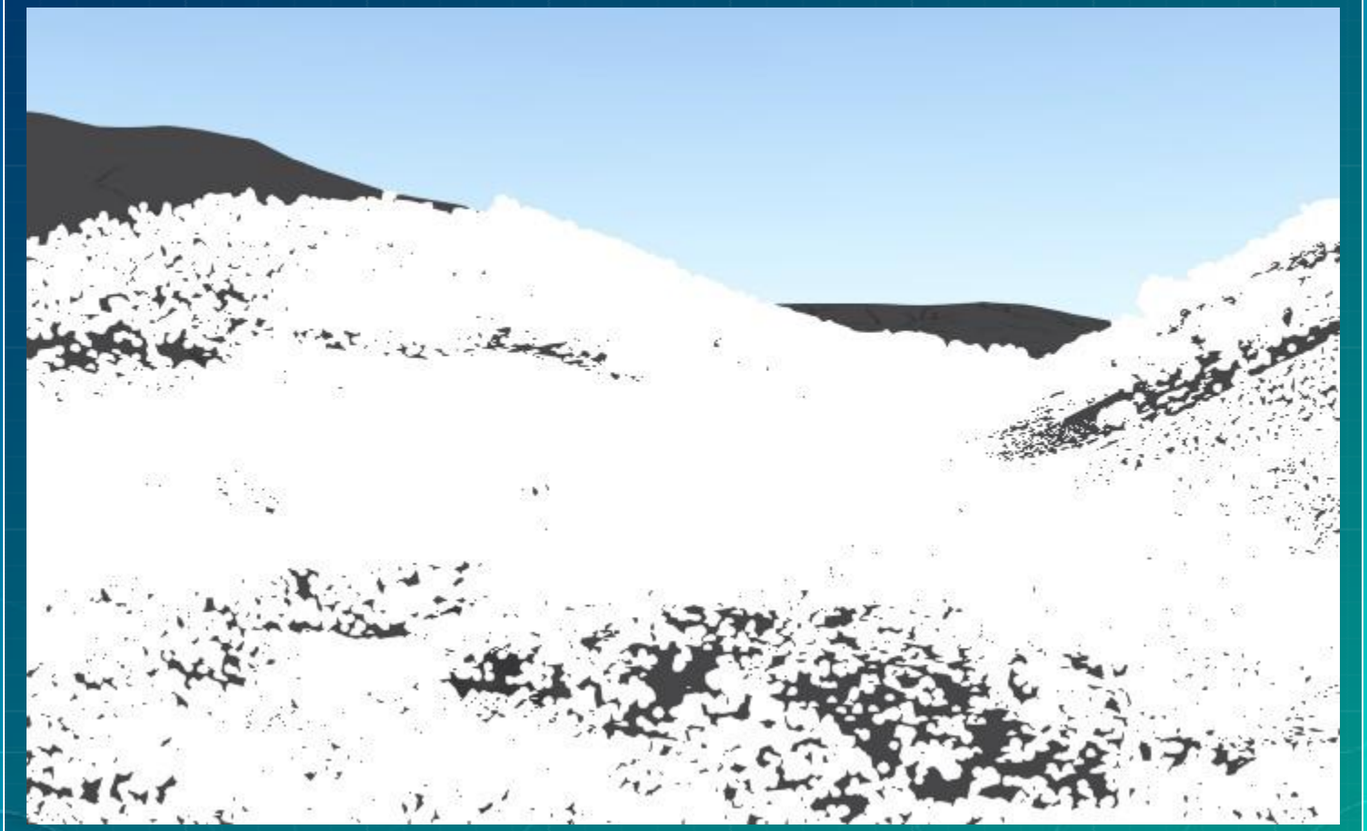


PointClouds —*Renderers*

- *RGB Renderer*
  - Use attribute value directly as color for point rendering
  - Only useful when data set contains RGB values in attributes
- *Stretch Renderer*
  - Define a color ramp, driven by a specific attribute
- *ClassBreaks Renderer*
  - Define value ranges that map to specific colors
- *UniqueValue Renderer*
  - Color by classification that is already present in a field

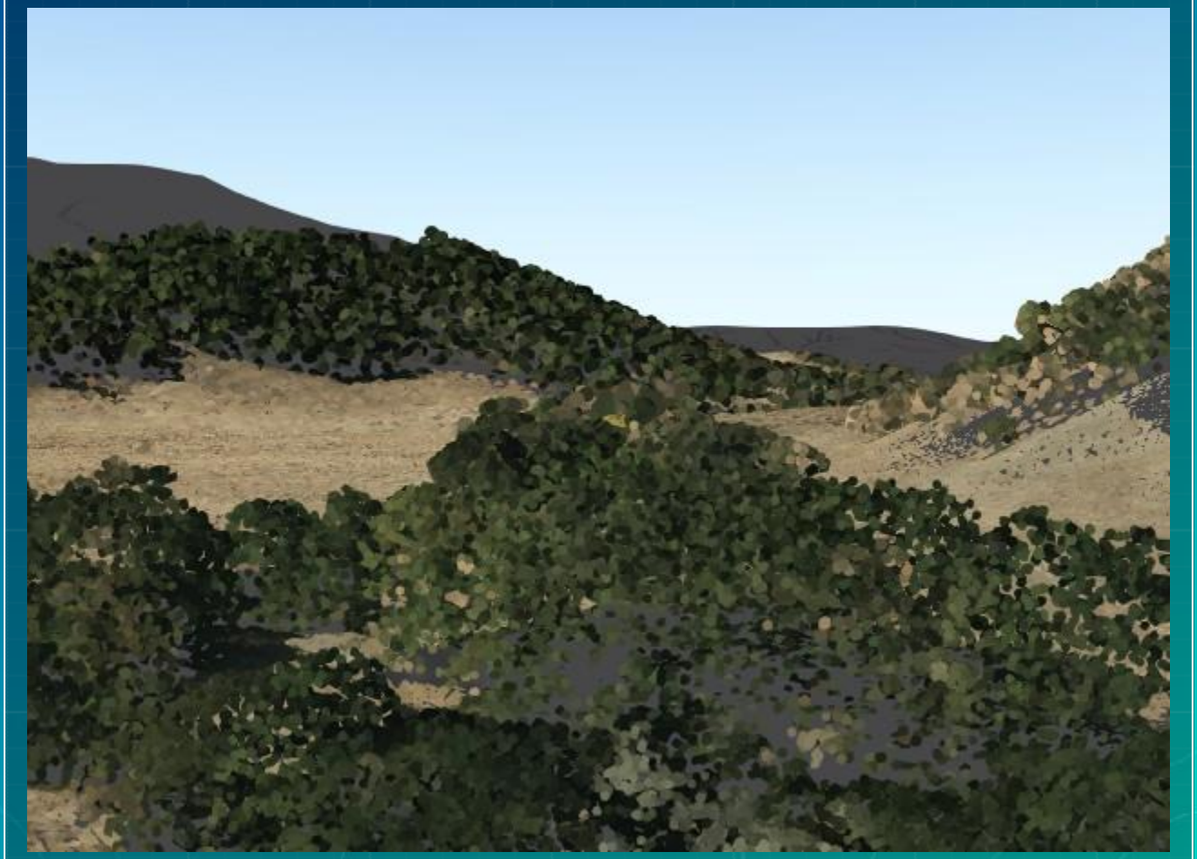
PointClouds —*RGBRenderer*

```
var layer = new PointCloudLayer({
  url: "...sonoma8_lepcc/SceneServer",
  renderer: new PointCloudRGBRenderer()
});

function changeRenderer() {
  layer.renderer = new PointCloudRGBRenderer({
    field: "RGB"
  });
}
```



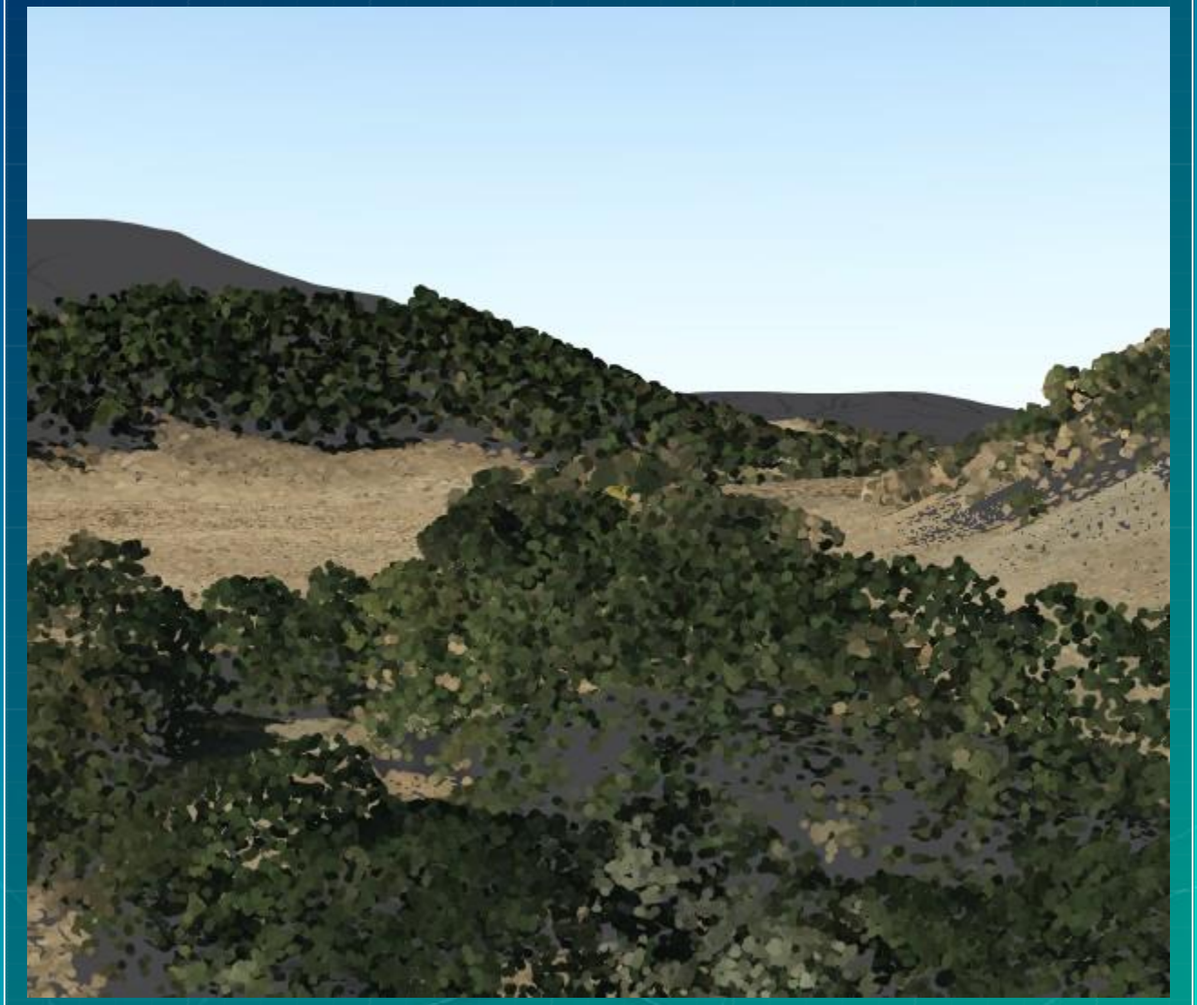
PointClouds — *StretchRenderer*

```
var layer = new PointCloudLayer({
  url: "...sonoma8_lepcc/SceneServer",
  renderer: new PointCloudRGBRenderer({ field: "RGB" })
});

function changeRenderer() {
  layer.renderer = new PointCloudStretchRenderer({
    field: "ELEVATION",
    stops: [{
      value: 0,
      color: "green"
    },{
      value: 50,
      color: "yellow"
    },{
      value: 100,
      color: "red"
    }]
  });
}
```



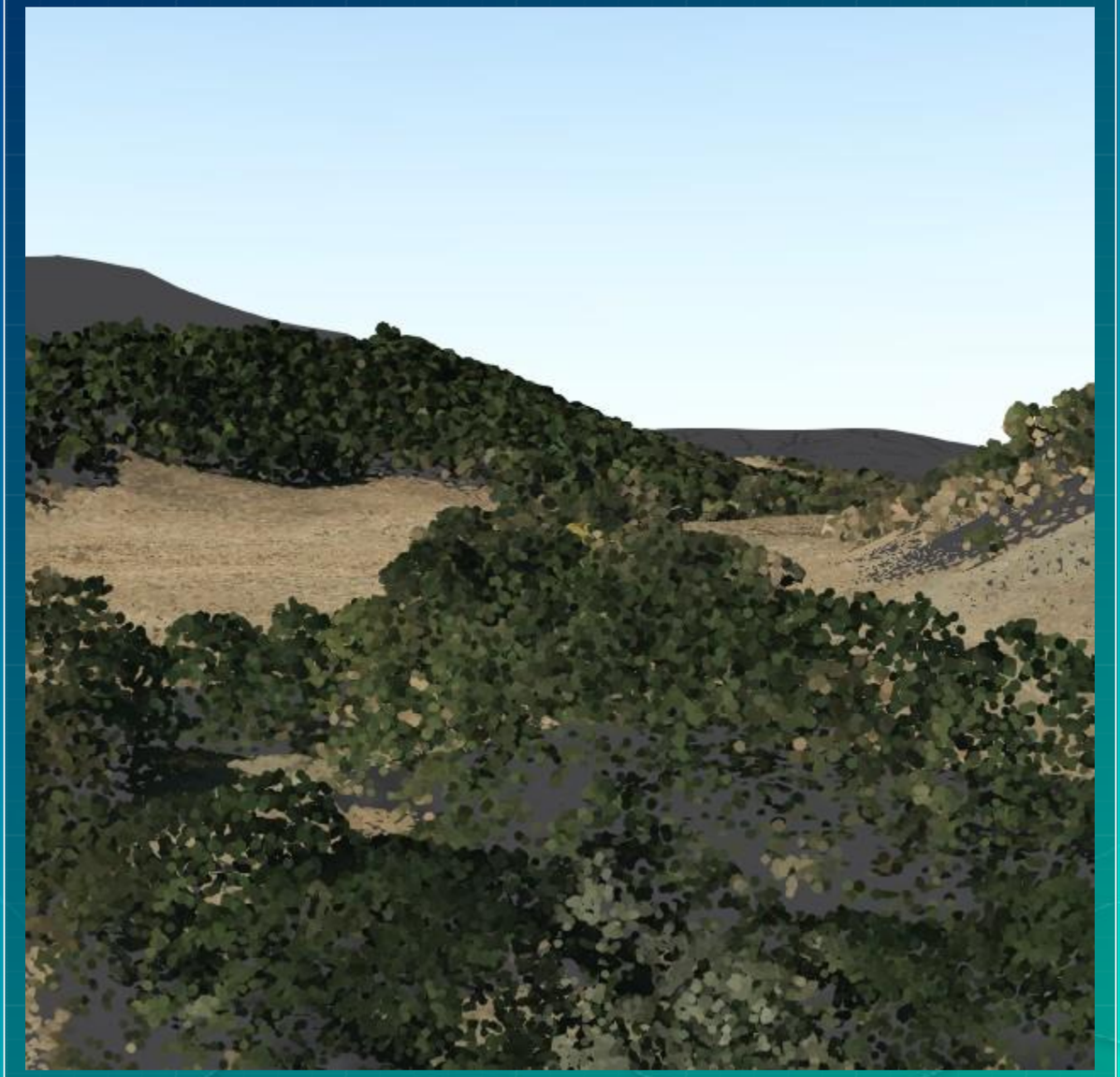
PointClouds —*ClassBreaksRenderer*

```
var layer = new PointCloudLayer({
  url: "...sonoma8_lepcc/SceneServer",
  renderer: new PointCloudRGBRenderer({ field: "RGB" })
});

function changeRenderer() {
  layer.renderer = new PointCloudClassBreaksRenderer({
    field: "INTENSITY",
    colorClassBreakInfos: [
      {
        minValue: 0,
        maxValue: 100,
        color: [100, 0, 0]
      }, {
        minValue: 100,
        maxValue: 200,
        color: [150, 0, 0]
      }, {
        minValue: 200,
        maxValue: 300,
        color: [200, 0, 0]
      }]
  });
}
```



PointClouds — *UniqueValueRenderer*

```
var layer = new PointCloudLayer({
  url: "...sonoma8_lepcc/SceneServer",
  renderer: new PointCloudRGBRenderer({ field: "RGB" })
});

function changeRenderer() {
  layer.renderer = new PointCloudUniqueValueRenderer({
  field: "CLASS_CODE",
  colorUniqueValueInfos: [
    {
      values: [2],
      label: "Ground",
      color: [222, 184, 135]
    }, {
      values: [3, 4, 5],
      label: "Vegetation",
      color: [200, 232, 171]
    }, {
      values: [6],
      label: "Building",
      color: [158, 40, 17]
    }, {
      values: [7, 8, 9, 10, 11, 12],
      label: "Other",
      color: [50, 50, 50]
    }
  ]});
}
```

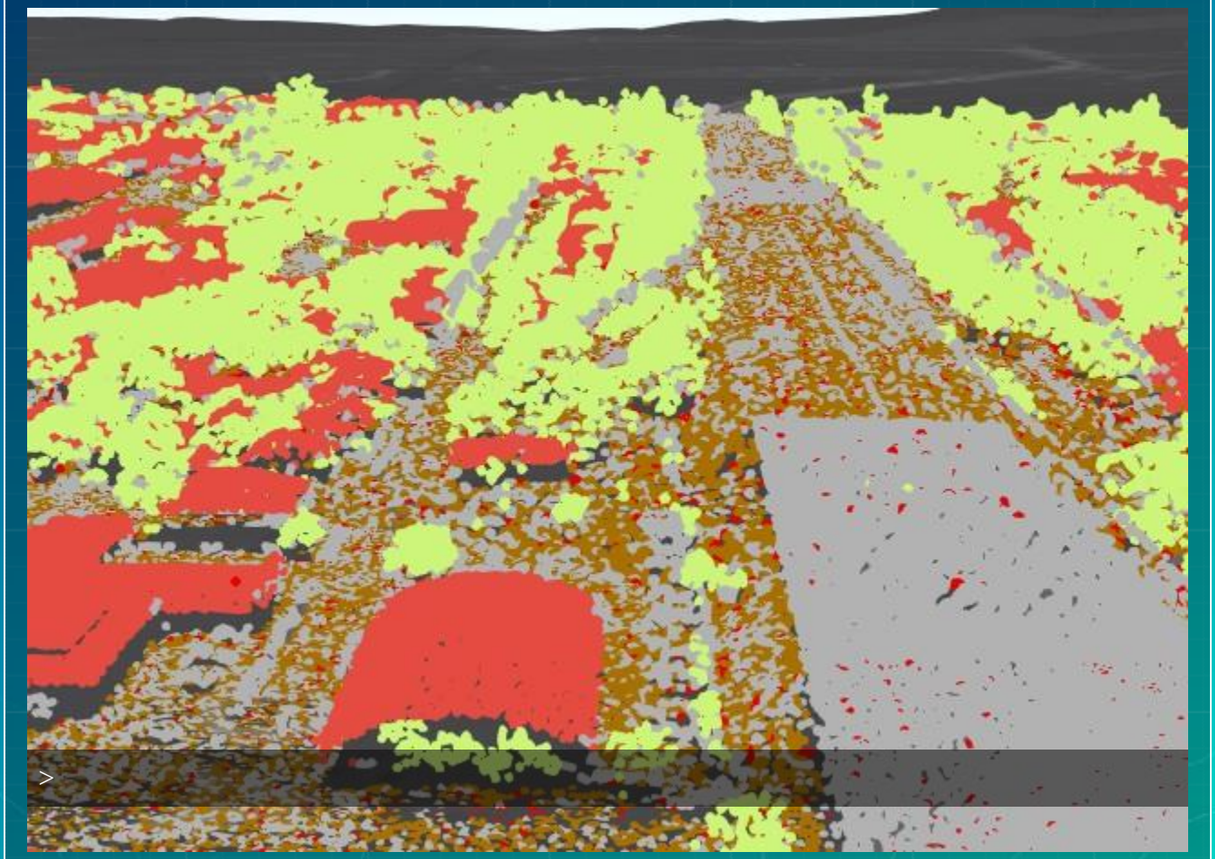


PointClouds —*Extended Properties*

- *colorModulation*
  - Alters brightness of points based on a data field
  - Displays scanned surfaces in a more realistic way
- *pointsPerInch*
  - Lets you configure the density of points on screen
  - Never exceeds a hard cap on the total number of points
- *pointSizeAlgorithm*
  - Two modes: splat and fixed-size
  - *Splat* automatically picks a size based on point density
  - *Fixed size* uses a constant size for all splats

PointClouds —*colorModulation Property*

```
var layer = new PointCloudLayer({
  url: "...sonoma8_lepcc/SceneServer",
  renderer: new PointCloudUniqueValueRenderer(...)
});

var enabled = false; var
colorModulation ={
  field: "INTENSITY",
  minValue: 35,
  maxValue: 211
};

function changeRenderer() {
  enabled = !enabled

  layer.renderer = new PointCloudUniqueValueRenderer({
    ...
    colorModulation: enabled ? colorModulation : null
  });
}
```

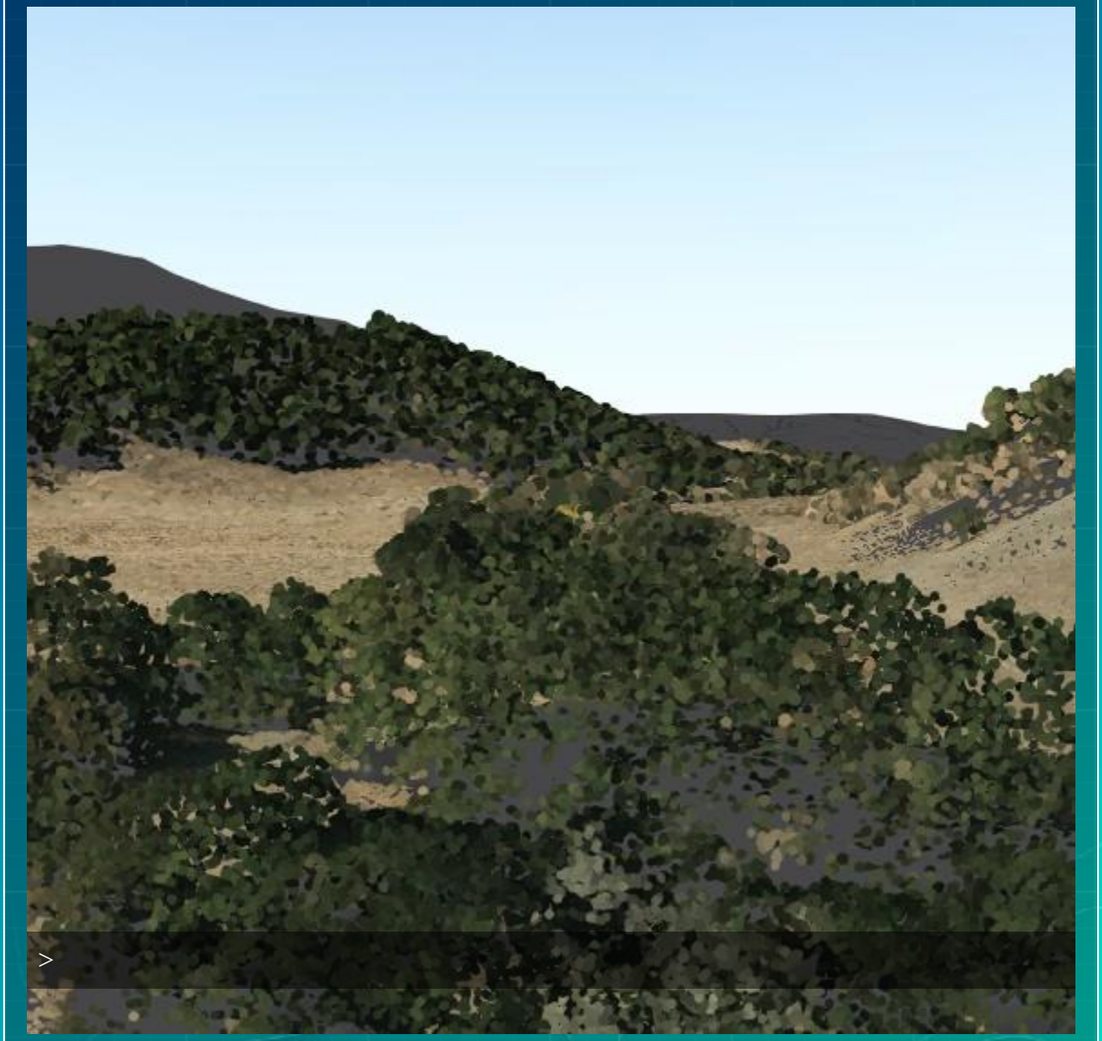


PointClouds —*pointsPerInchProperty*

```
var layer = new PointCloudLayer({
  url: "...sonoma8_lepcc/SceneServer",
  renderer: new PointCloudRGBRenderer({ field: "RGB" })
});

var ppi = 40;

function changeRenderer() {
  switch (window.ppi) {
    case 10:
      ppi = 1;
      break;
    case 1:
      ppi = 40;
      break;
    default:
      ppi = 10;
      break
  }

  layer.renderer = new PointCloudRGBRenderer({
    field: "RGB",
    pointsPerInch: ppi
  });

  log("pointsPerInch:", ppi)
}
```

PointClouds —*pointSizeAlgorithm Property*

```
var layer = new PointCloudLayer({
  url: "...sonoma8_lepcc/SceneServer",
  renderer: new PointCloudRGBRenderer({ field: "RGB" })
});

var algorithm = "splat";

function changeRenderer() {
  if (algorithm === "splat") {
    layer.renderer = new PointCloudRGBRenderer({
      field: "RGB",
      pointSizeAlgorithm: { type: "splat" }
    });
  } else {
    layer.renderer = new PointCloudRGBRenderer({
      field: "RGB",
      pointSizeAlgorithm: { type: "fixed-size" }
    });
  }

  window.log("pointSizeAlgorithm:", algorithm);

  algorithm = algorithm === "splat" ? "fixed-size" : "splat"
}
```

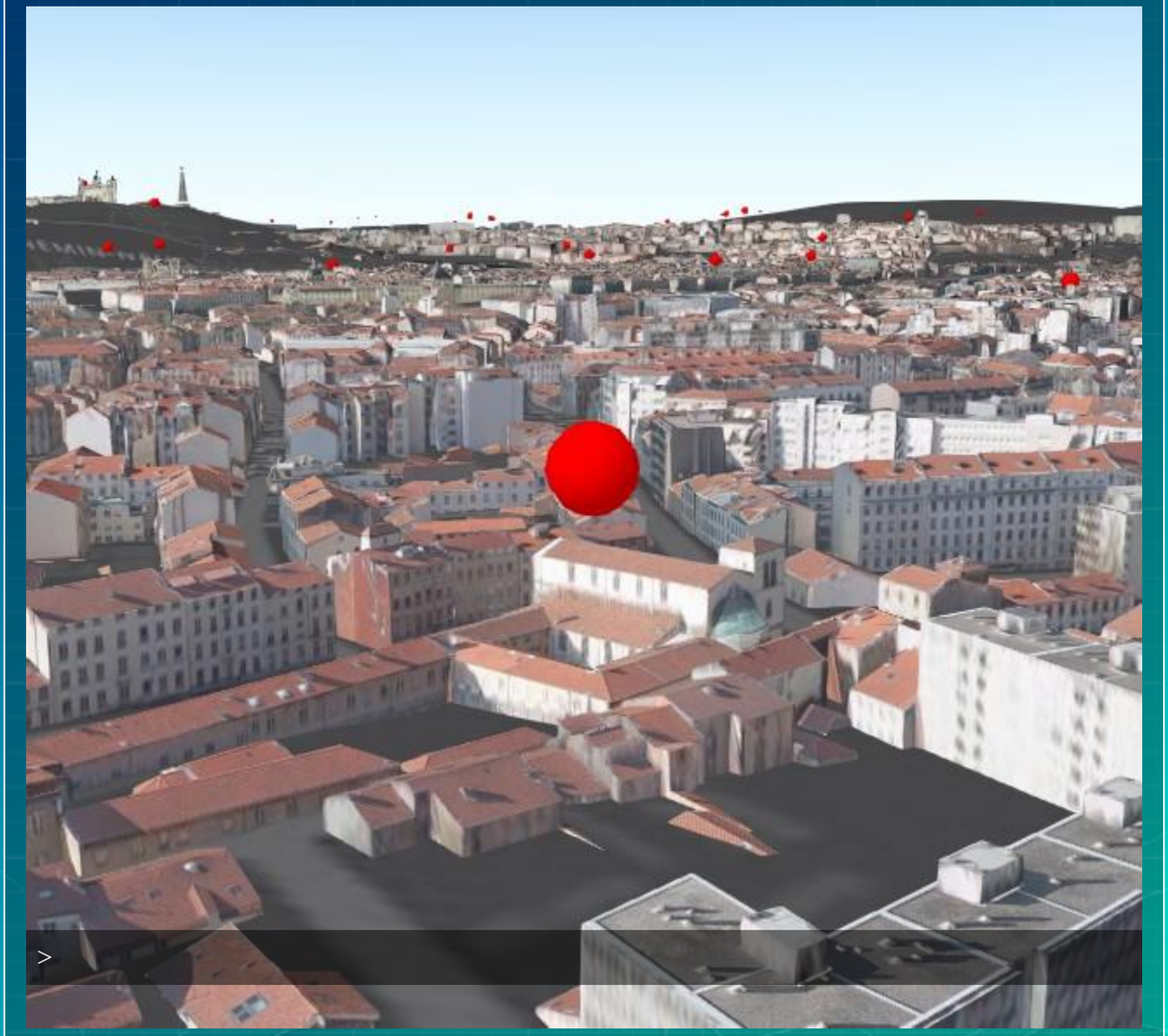


Layers —*Common Properties*

- *elevationInfo*
  - Well known modes: *on-the-ground*, *relative-to-ground*, *absolute-height*
  - New mode: *relative-to-scene*
- *verticalOffset*
  - Allows offsetting all points in a layer by a given amount
- *callout*
  - Gives visual clues between features and map points

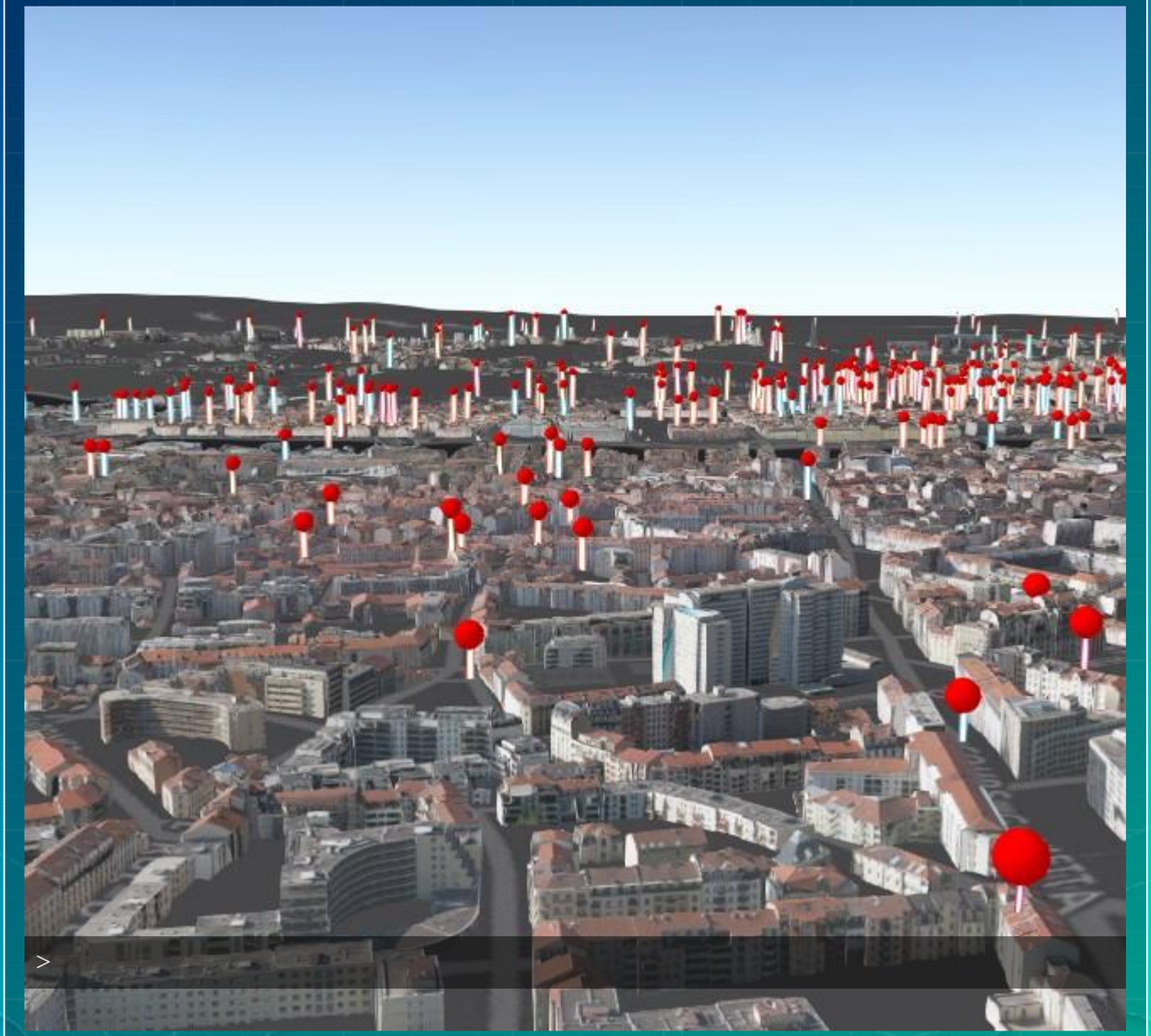
SceneLayer — *Relative To Scene Elevation*

```
var sceneLayer = new SceneLayer({
  url: "...Batiments_Lyon_3D_2012/SceneServer",
  renderer: new SimpleRenderer(...)
});

var elevationMode = "relative-to-scene";

var featureLayer = new SceneLayer({
  url: "...LyonPointsOfInterest/FeatureServer",
  renderer: new UniqueValueRenderer(...),
  elevationInfo: {
    mode: elevationMode
  }
});

function changeLayer() {
  if (elevationMode === "relative-to-scene") {
    elevationMode = "relative-to-ground";
  }
  else {
    elevationMode = "relative-to-scene";
  }

  featureLayer.elevationInfo = { mode: elevationMode }
  window.log("elevationInfo.mode:", elevationMode)
}
```



SceneLayer — *Vertical Offset & Callouts*

```
var featureLayer = new SceneLayer({
  url: "...LyonPointsOfInterest/FeatureServer",
  renderer: new UniqueValueRenderer(...)
});

var offset = 20; function

changeLayer() {
  if (offset < 60) { offset += 20; }
  else { offset = 20; }

  featureLayer.renderer = new UniqueValueRenderer({
    uniqueValueInfos: [
      symbol: {
        symbolLayers: [
          new PointSymbol3D({
            verticalOffset: { screenLength: offset },
            callout: new LineCallout3D(...)
          })
        ]

      }
    ]
  });

  window.log("offset:", offset)
}
```



3D GIS Jumpstart

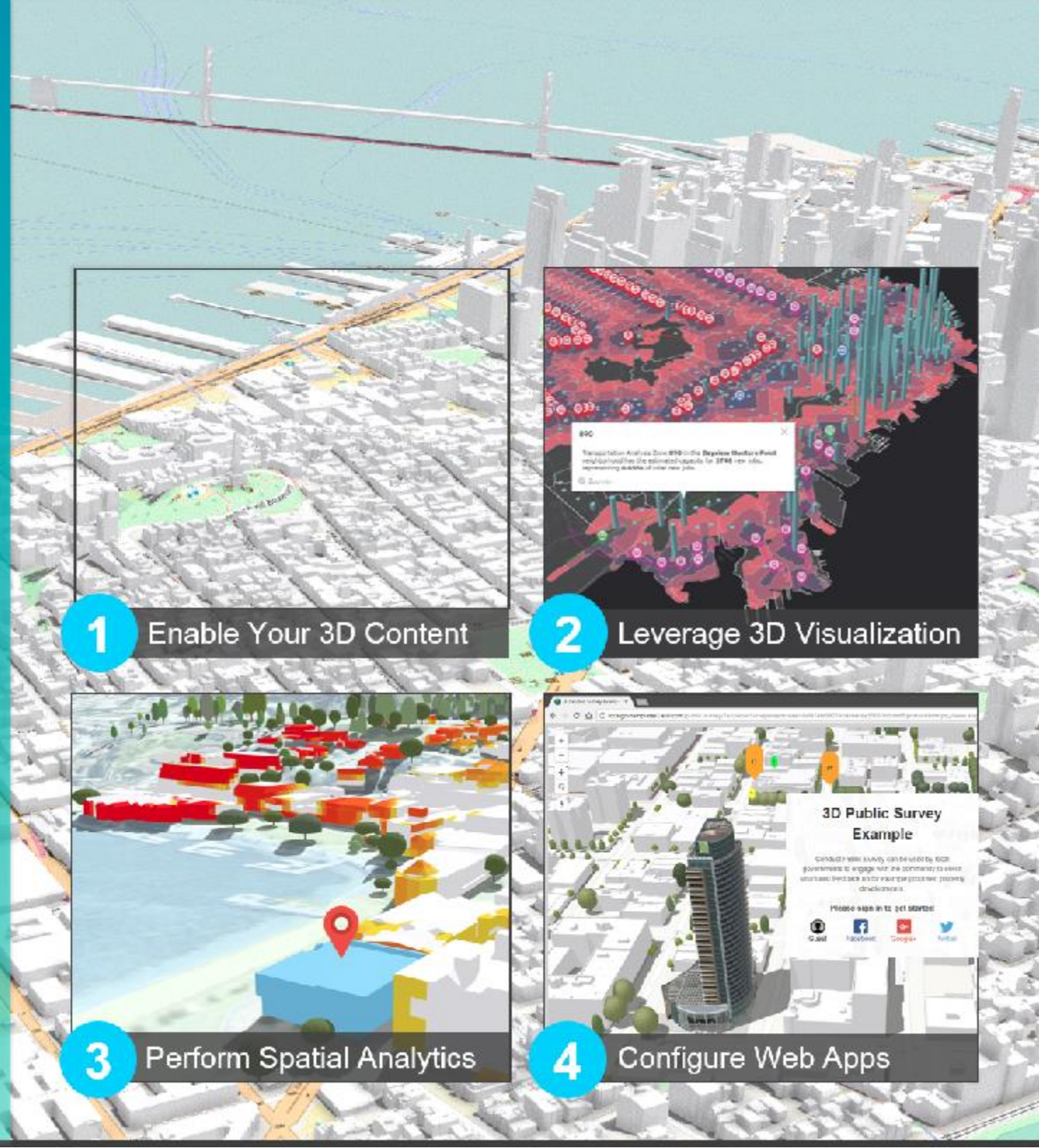
Organized as a phased set of workshop activities

Dive deep into working with 3D in ArcGIS

- Learn advanced 3D workflows and techniques
- Hands-on, one-on-one
- Your data in your environment
- Align with a current project for immediate ROI

Connect with an Expert: 3DConsulting@esri.com

Learn more about Esri's strategy and vision for Enterprise 3D GIS bit.ly/Enterprise_3D_GIS



esri

THE SCIENCE OF WHERE