



# Building Interactive Web Apps Using the JavaScript API's Geometry Engine

Kristian Ekenes, David Bayer

2018 Esri DEVSummit Conference | Palm Springs, CA

# Session Goals

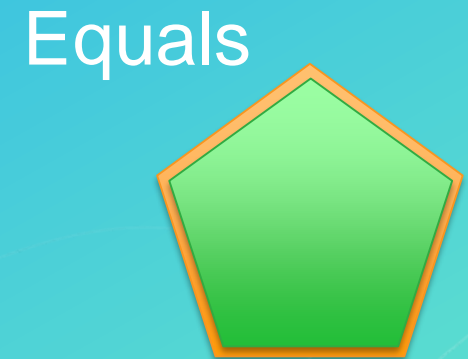
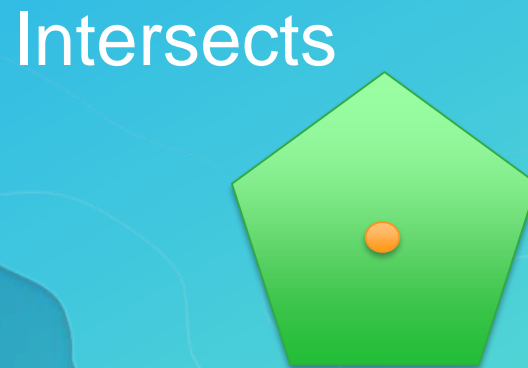
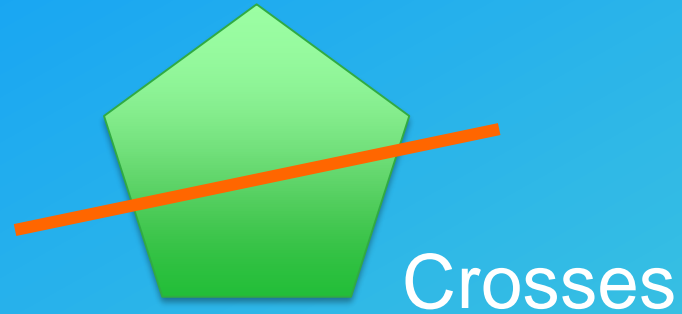
- Overview of the Geometry Engine
  - Available operations
- Coding with the Geometry Engine
  - How to use the Geometry Engine
- Geometry Engine Internals
  - Sync/Async Architecture
- When to use the Geometry Engine
  - When does it make a difference?

# Geometry

Points, lines, polygons, multipoints...

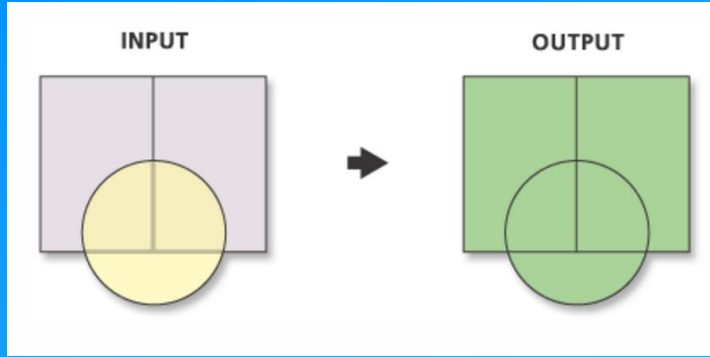


# Operations: Geometry Operations

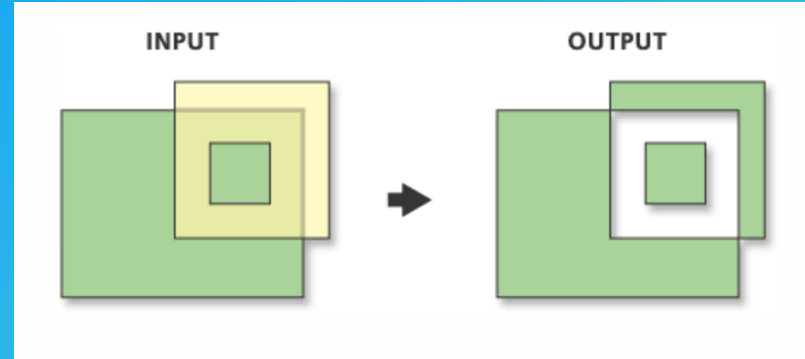


# Operations: Overlay

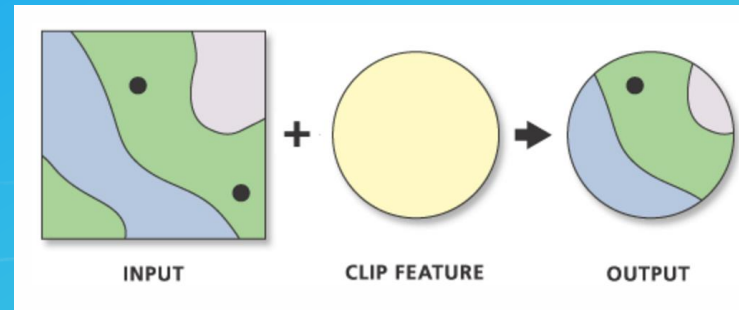
## Union



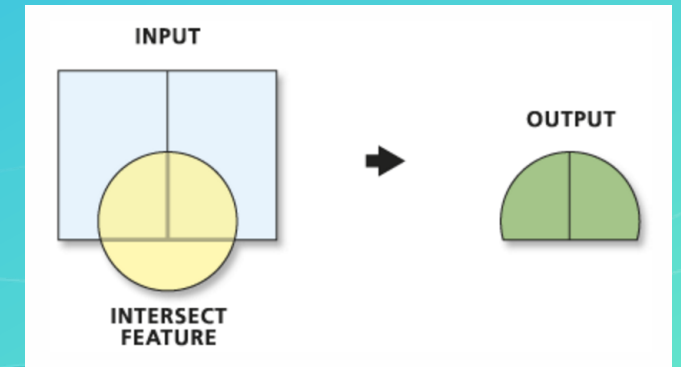
## Symmetric Difference



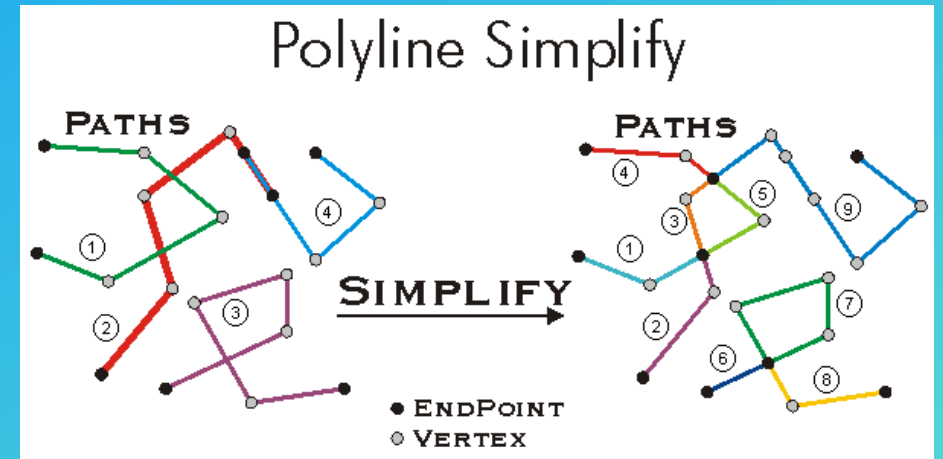
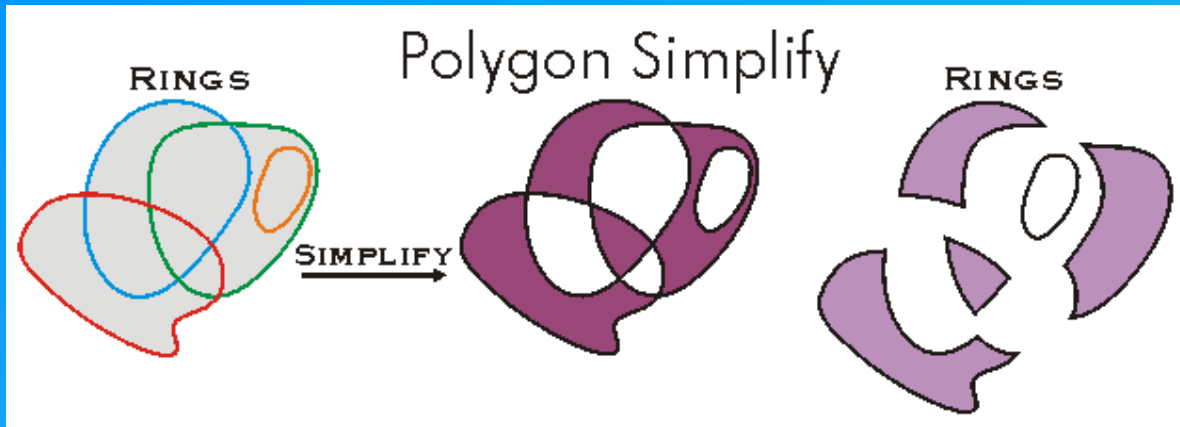
## Clip



## Intersection

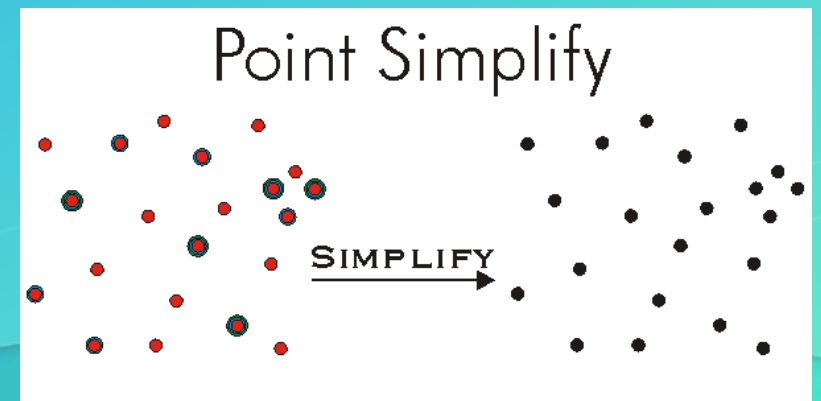


# Operations: Topological correctness



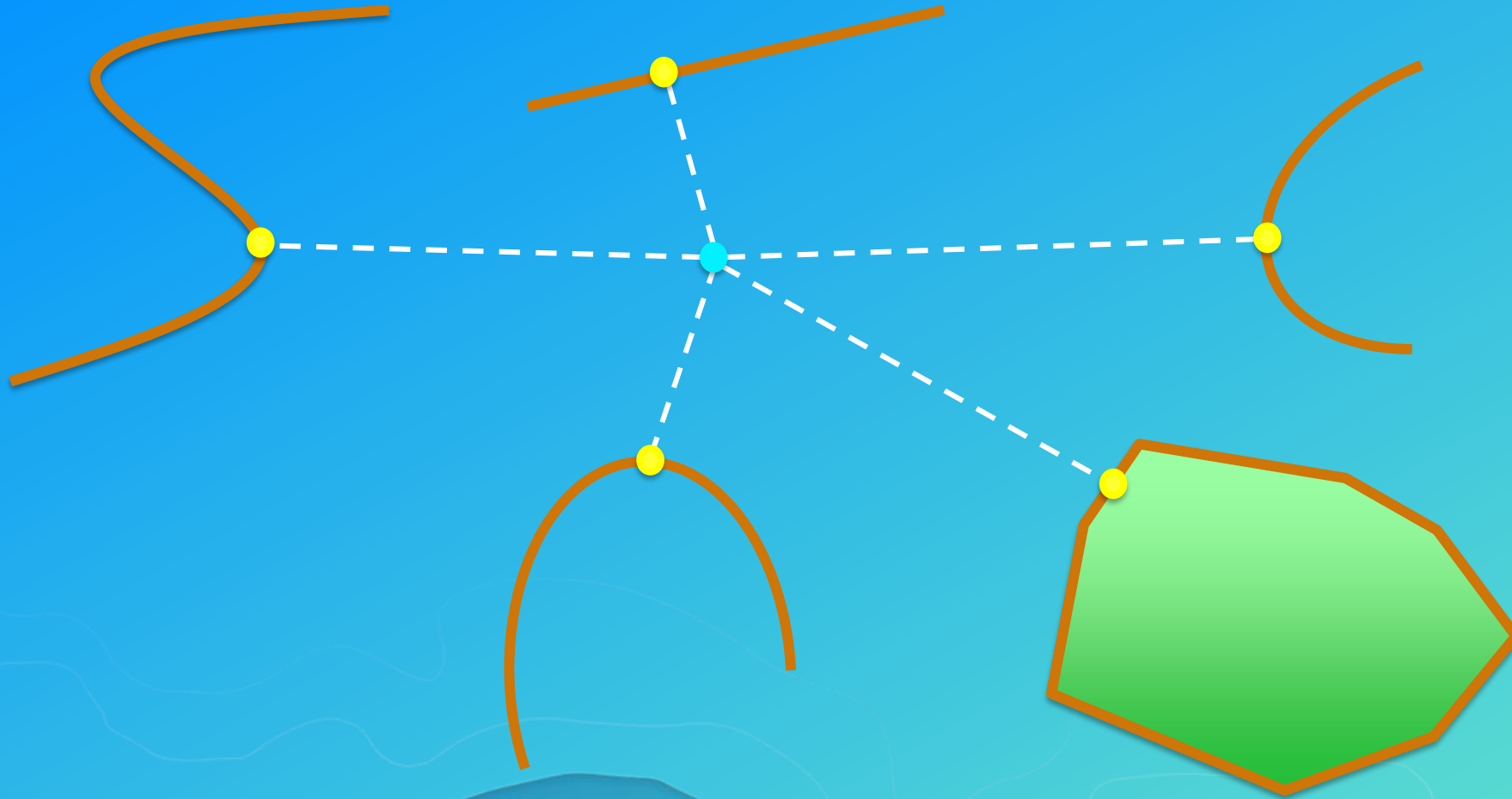
Simplify

isSimple



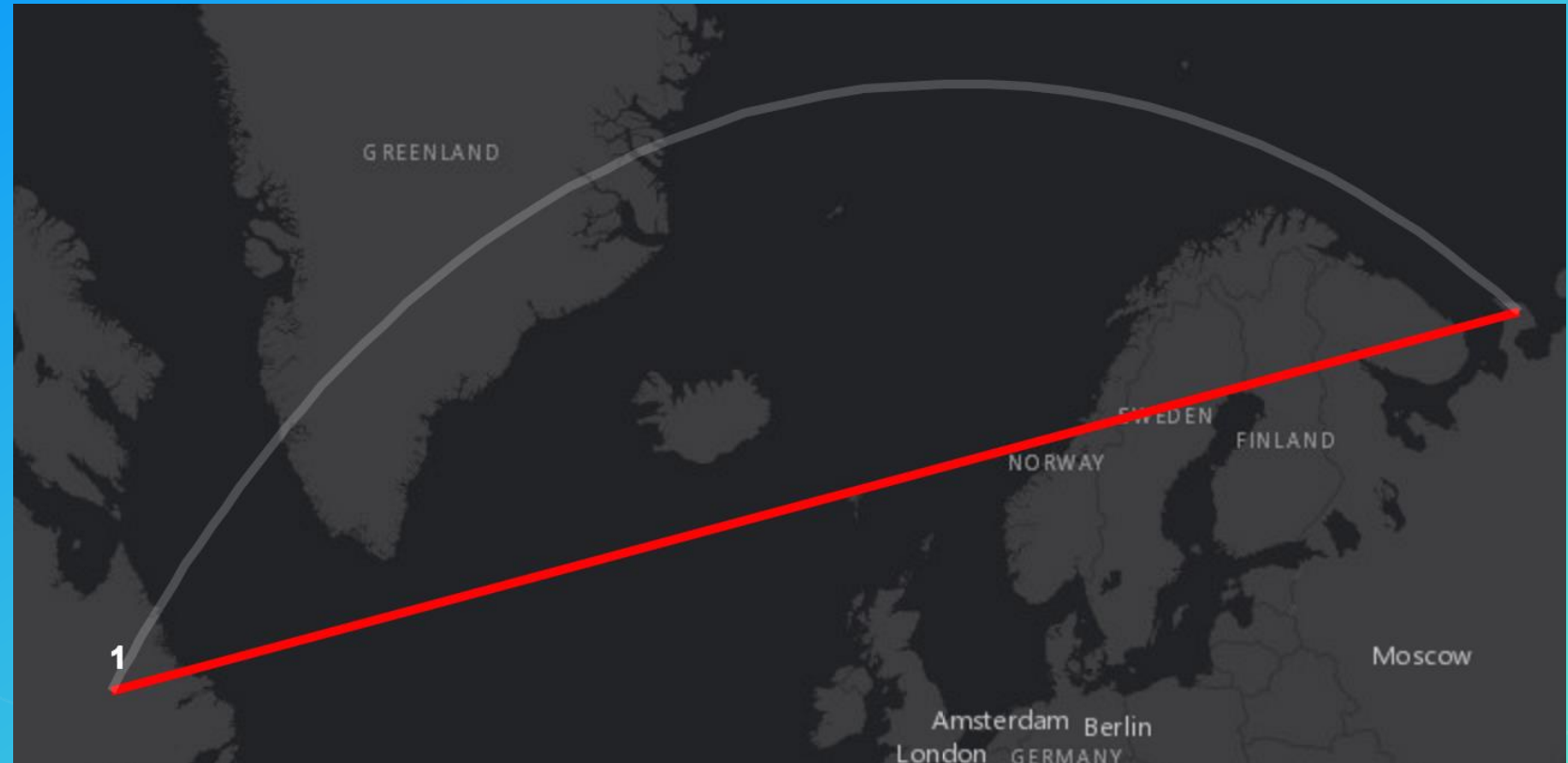


# Operations: Nearest vertices/coordinates



# Operations: Measurement

- geodesicLength
- planarLength
- geodesicArea
- planarArea





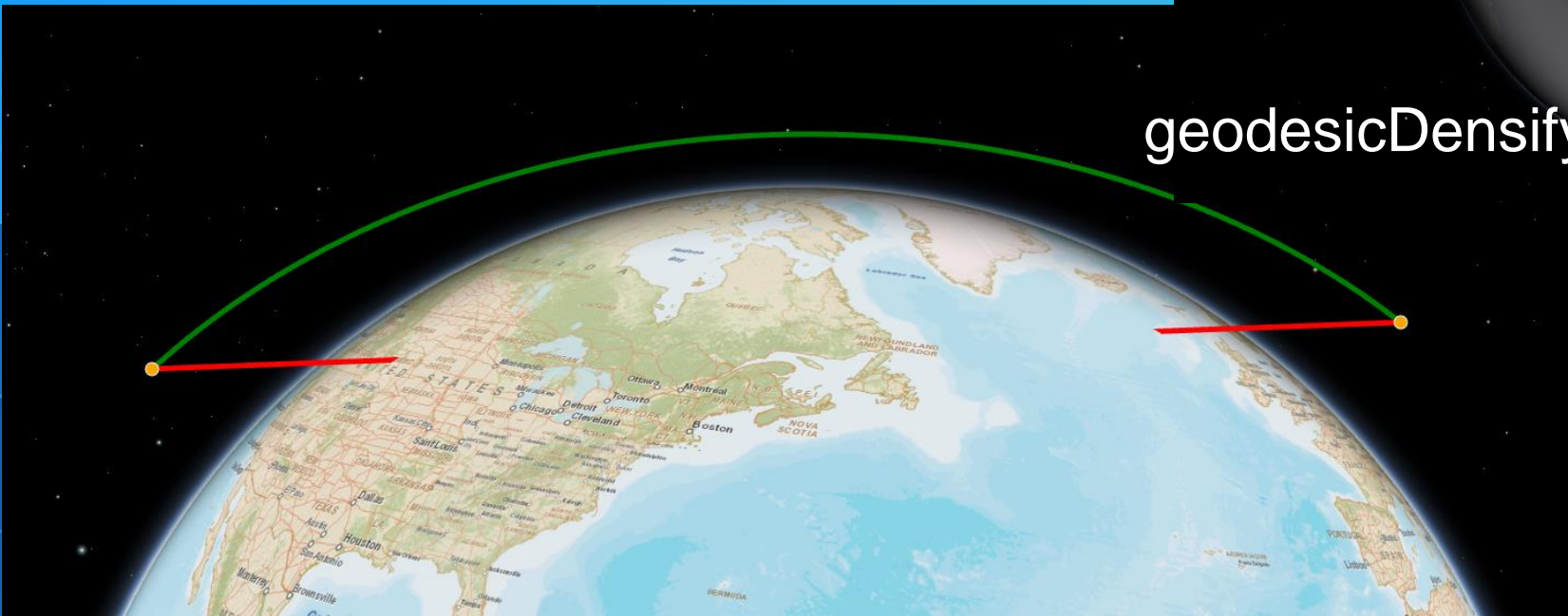
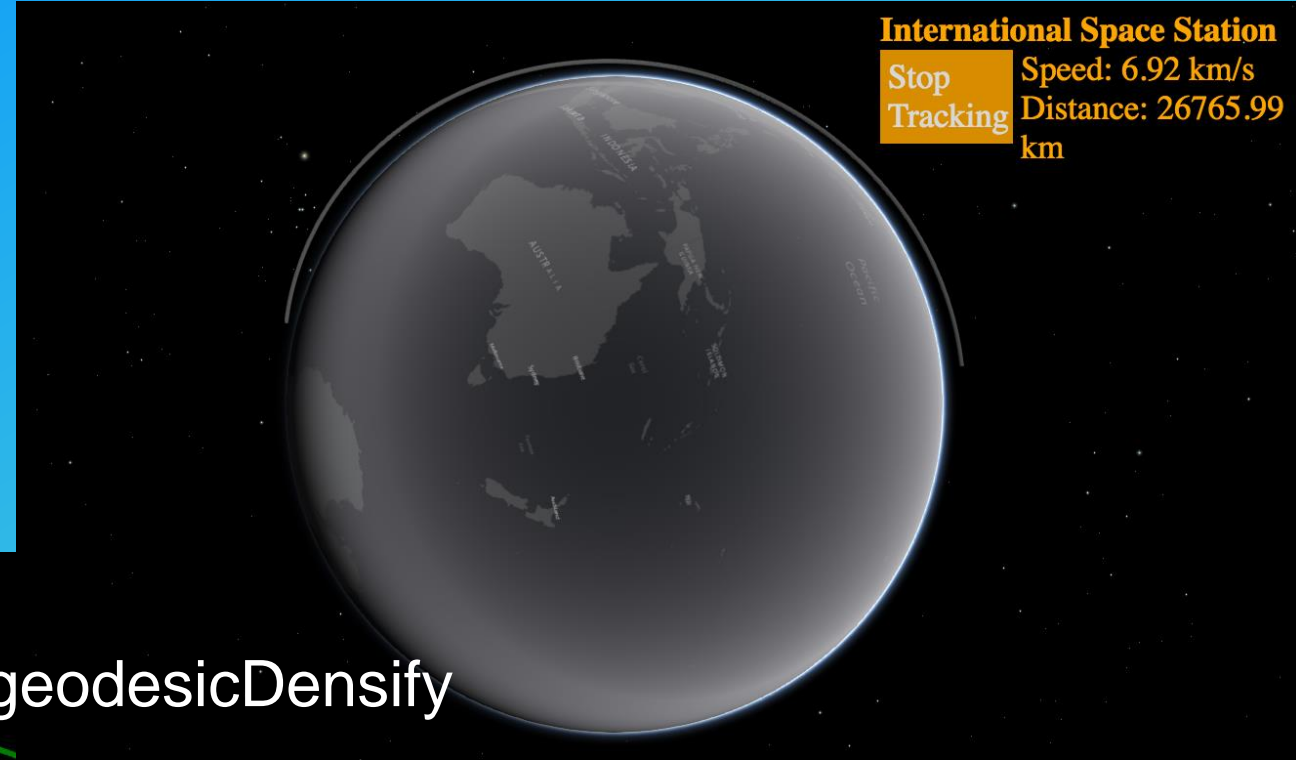
# Operations: Buffering

## Planar and Geodesic Buffering



# Other Operations

- Generalize
- Rotate
- Densify
- Flip
- ConvexHull



The background features a blue-to-teal gradient. In the upper right, there are faint, light blue topographic contour lines. In the lower left, there is a stylized, layered mountain range with colors ranging from dark blue to purple.

# I can do this already...

It's called the Geometry Service!



# User Experience: Interactivity

## Geometry Service REST Call



Encoding  
Geometry  
as JSON

Sending Request over Network

Running Geometry  
Operation

Sending Response back  
over Network

Decoding  
Response

5ms

200+ ms

100+ ms

200+ ms

5ms

# User Experience: Interactivity

## Geometry Engine API Call



Running Geometry Operation

200+ ms

# Demo

How many REST calls can I save?





# Writing Apps with Geometry Engine



# Geometry Engine : Writing Code

## Calling the methods

```
require( [ "esri/geometry/geometryEngine" ],  
function( geometryEngine ) {  
  
    var buffer = geometryEngine.buffer(mapPoint, 10, "miles");  
  
});
```

# Geometry Service : Code Comparison

```
require( [ "esri/tasks/GeometryService", "esri/tasks/BufferParameters" ],  
function( GeometryService, BufferParameters ) {  
  
    var geomService = new GeometryService("//.../GeometryServer");  
    var bufferParams = new BufferParameters();  
    bufferParams.geometries = [mapPoint]  
    bufferParams.distances = [10]  
    bufferParams.unit = GeometryService.UNIT_MILE;  
    bufferParams.outSpatialReference = map.spatialReference;  
  
    geomService.buffer( bufferParams, function(buffer) {  
  
        });  
  
});
```

# Building Editing Apps



# Building Analysis Apps





# Synchronous versus Asynchronous

You can use Geometry Engine Async





# Geometry Engine Async

- **Asynchronous Geometry Engine**
  - Uses Web Workers to perform Geometry Engine
  - Operations return Promises.
- **Benefits**
  - Browser is not blocked whilst operations run
  - Greater throughput. Operations can run in parallel
  - Easier substitution of existing GeometryService code
- **Drawbacks**
  - Older Browsers do not support WebWorkers

# Asynchronous Patterns

- Uses Deferred Pattern

```
require( [ "esri/geometry/geometryEngineAsync" ],  
function( geometryEngineAsync ) {  
  
    geometryEngineAsync.buffer(mapPoint, 10, "miles").then( function(buffer) {  
  
    },  
    function(err) {  
  
    });  
  
});
```

# Geometry Engine internals

What is the Geometry Engine doing for me!

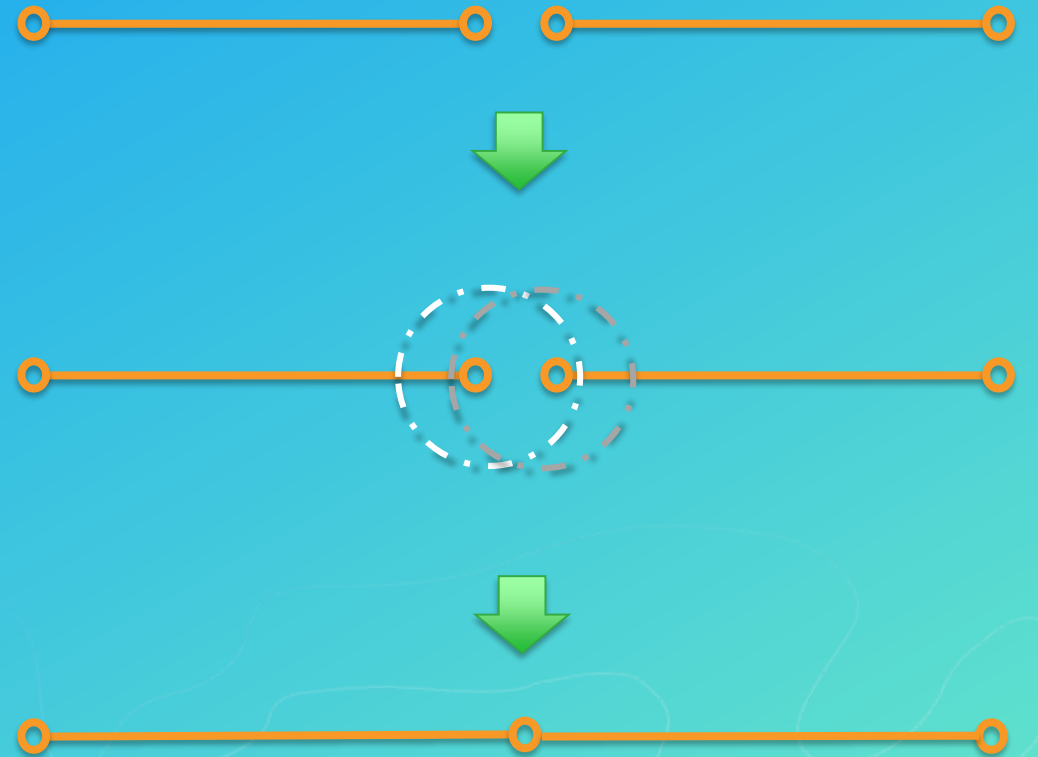


# Spatial Tolerance

Spatial References have different Tolerances.  
GeometryEngine uses X/Y Tolerance  
in its calculations



XY Tolerance = 0.001



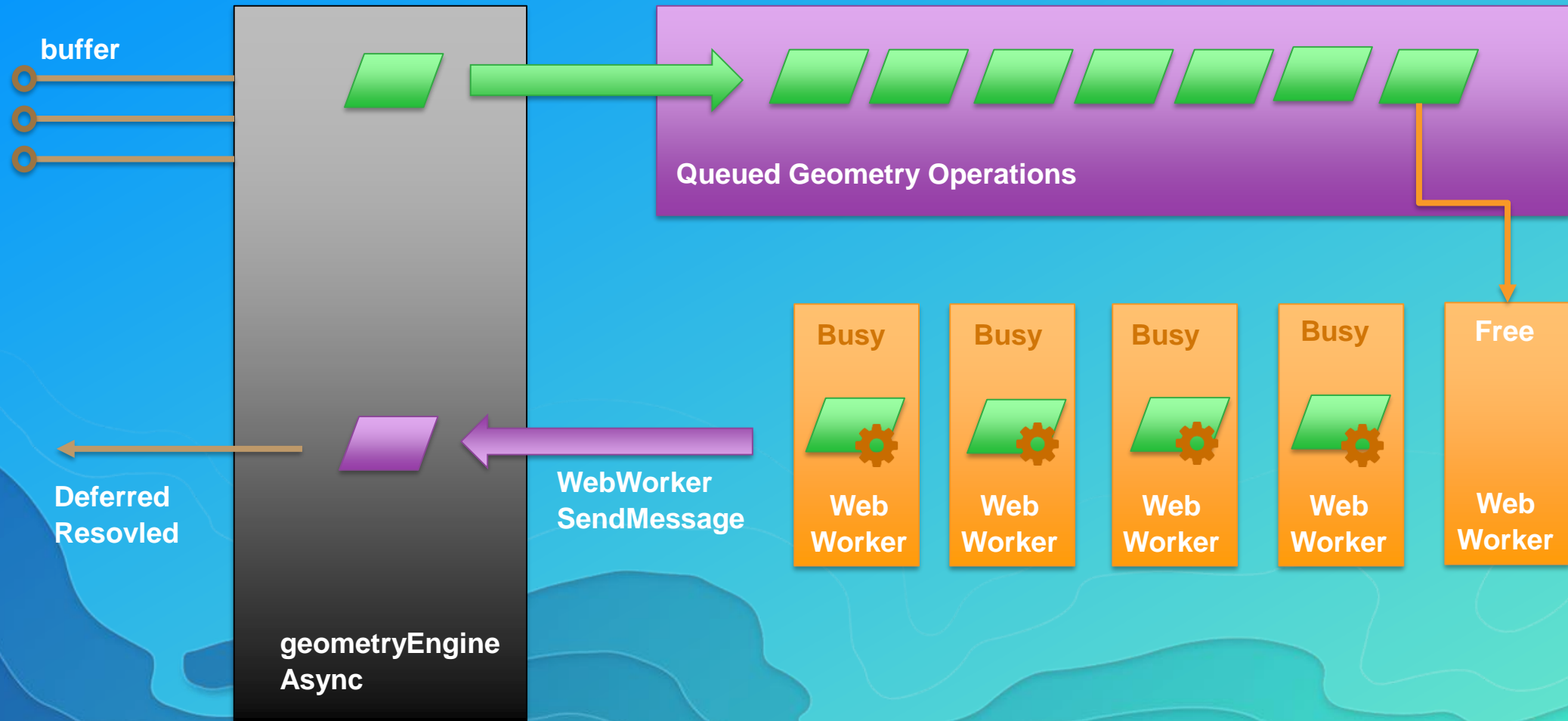
# Spatial Tolerance

## Inside the Geometry Methods

```
require( [ "esri/geometry/geometryEngine" ],  
function( geometryEngine ) {  
  
    var mapPointA = new esriPoint(1.22, 2.0, map.spatialReference)  
    var mapPointB = new esriPoint(1.220000000001, 2.0, map.spatialReference)  
  
    var areequal = geometryEngine.equals(mapPointA, mapPointB);  
  
    // Returns True  
});
```

# Geometry Engine Async

## Architecture and Throughput





# Coordinate Systems

- **Projected and Geographic**
  - **No Project Operation**
  - **Certain methods will only work in WebMercator or Geographic Coordinate Systems**
    - **GeodesicArea**
    - **GeodesicLength**
    - **GeodesicBuffer**

# When to use the Geometry Engine

When will it make a difference in my app



# Geometry Engine: Trade-offs

- **Trade-off between Geometry Service and Geometry Engine**
  - Large AMD download for Geometry Engine**versus**
  - Network latency and round trip cost of Geometry Service
- **Other Considerations: Apps Requirements**
  - Unsupported operations : projections
  - Frequency of calls : user interactivity
  - Offline or poor network connection
  - Ease of coding

# GeometryEngine Resources

Documentation, Blogs, Samples



# Resources

- **Documentation**

- <https://developers.arcgis.com/javascript/latest/api-reference/esri-geometry-geometryEngine.html>
- <https://developers.arcgis.com/javascript/latest/api-reference/esri-geometry-geometryEngineAsync.html>

- **Blogs**

- [Testing spatial relationships and editing](#)
- [Measurement](#)
- [Overlay analysis](#)

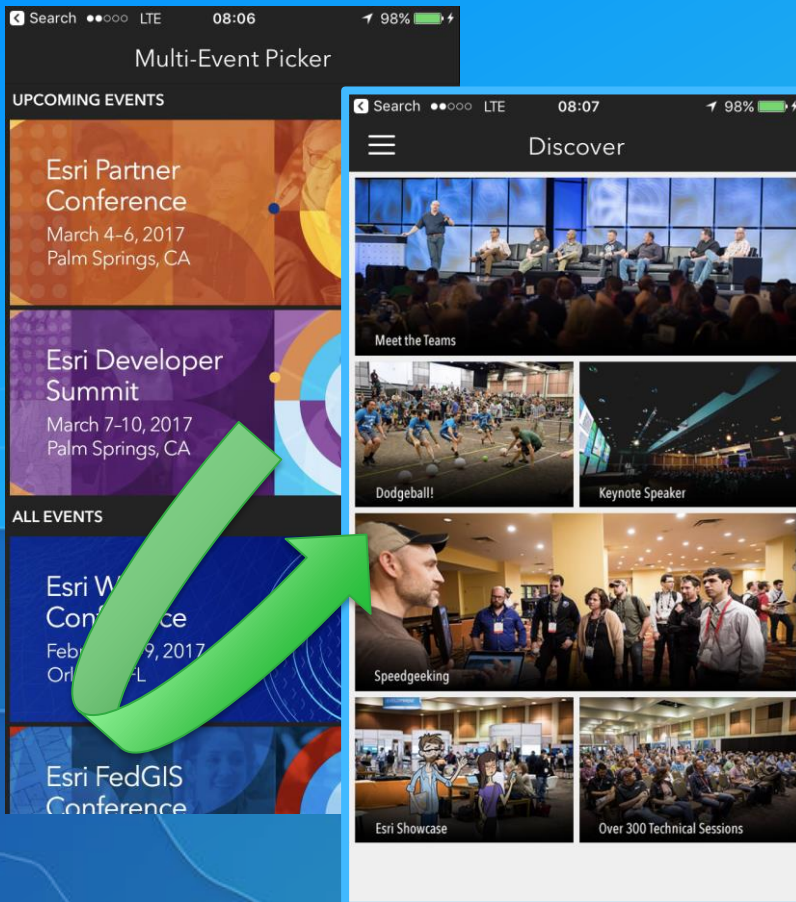
- **Demos/samples**

- <https://github.com/ekenesc/conferences/tree/master/ds-2017/geometry-engine>
- <https://github.com/ekenesc/esri-js-samples>

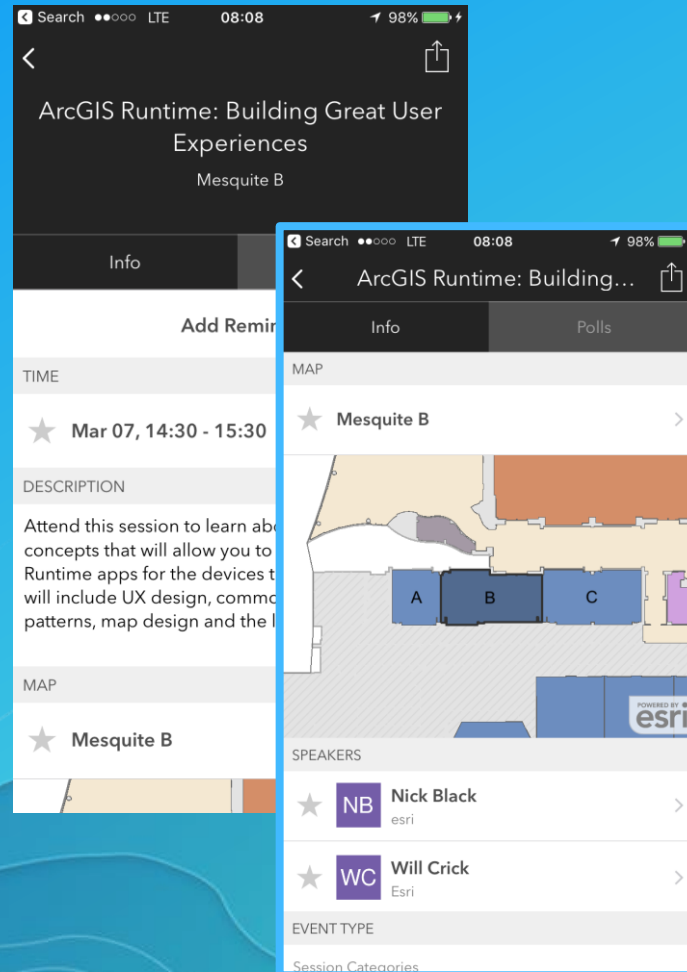


# Please Take Our Survey!

Download the Esri Events app  
and go to DevSummit

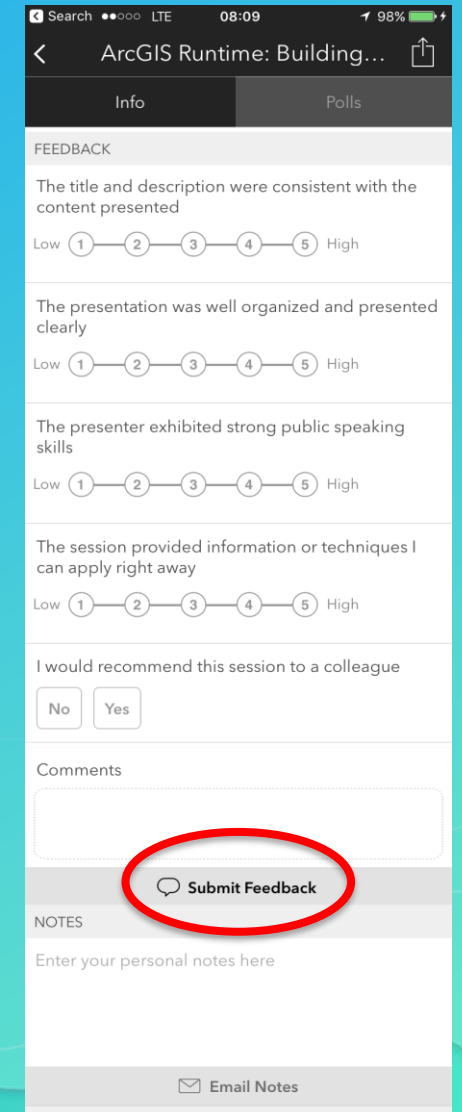


Select the session you attended



Scroll down to the  
“Feedback” section

Complete Answers,  
add a Comment,  
and Select “Submit”







esri

THE  
SCIENCE  
OF  
WHERE