ArcGIS Runtime: Analysis

Lucas Danzinger

Mike Branscomb

2019 ESRI DEVELOPER SUMMIT
Palm Springs, CA

ArcGIS Runtime session tracks at DevSummit 2019

- ArcGIS Runtime SDKs share a common core, architecture and design
- Functional sessions promote common capabilities and workflows
  - An Introduction to the API and Architecture
  - Styling maps
  - Building AR and VR Experiences
  - Getting to know Arcade
  - Configuring Webmaps
  - Analysis
  - Preparing your data for offline use
  - Editing your data online and offline
  - Authenticating your applications
  - Working with maps online and offline
  - Working with ArcGIS Enterprise and Portal
  - Building 3D applications
  - Road ahead
  - Everything you wanted to know... Q&A
  - Migrating from Engine
- Product sessions promote specific development experiences
- Demo theaters highlight specific technical capabilities

Shared workflows, any platform, any device

Agenda

- Geometry operations
- Geoprocessing
- Visual Analysis
- Summary & Roadmap

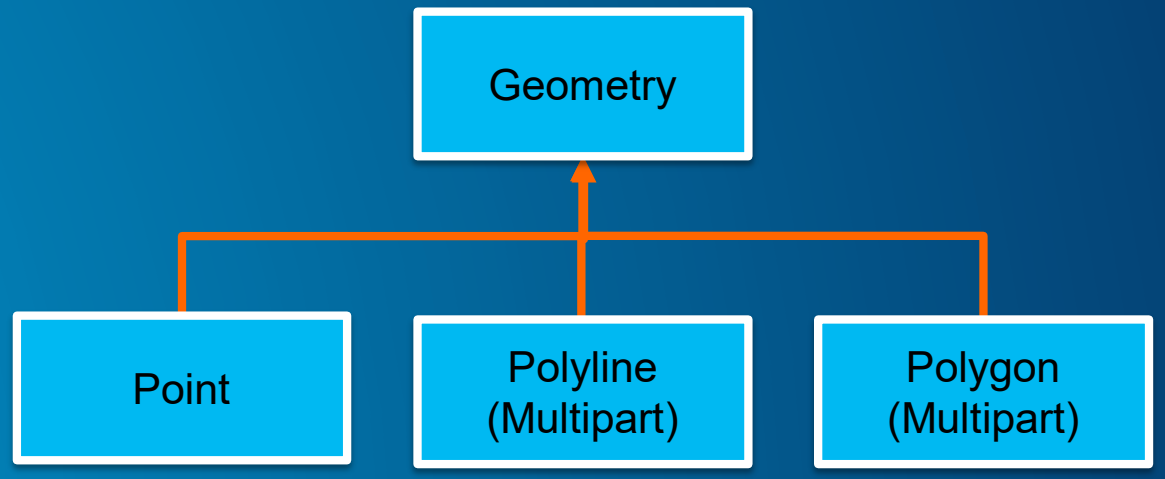
Geometry Operations

Lucas Danzinger

Geometry Engine

- Available on all Runtime SDKs
- Class with static methods for Geometry
- Input Geometries: point, line, polygon, envelope, or multipoint
- From features and graphics
- Fast, synchronous client-side processing

- 3 groups of functions
  - Geometry dimension operations
  - Geometry spatial operations
  - Building new geometries



Geometry dimension `questions`

- How long is this line?
- What is the area of this polygon?
- What is the boundary of this polygon?
- What is the distance between A and B?

Java sample

```
// get planar distance between 2 points
double planarDistance = GeometryEngine.distanceBetween(pointA, pointB);
```

- General information:
  - Results are in the unit of the spatial reference of the geometry
  - All geometries in an operation must have the same spatial reference

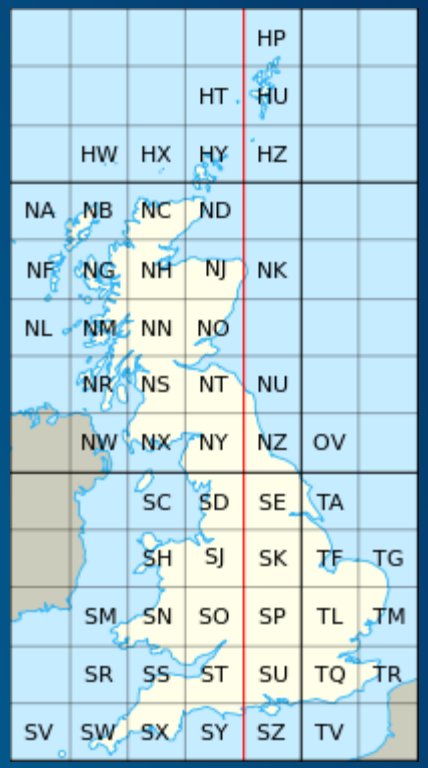
Projecting between spatial references

- Project method
- Convert from Web Mercator into a local coordinate system

Java sample

```
// create a OSGB point from a Web Mercator point
Point reprojectedGeometry = (Point) GeometryEngine.project(point, OSGB);
```

- Default transformation used to project is chosen for you
  - Or specify transformation
- TransformationCatalog class
  - Find suitable transformations



Answer spatial questions about geometries

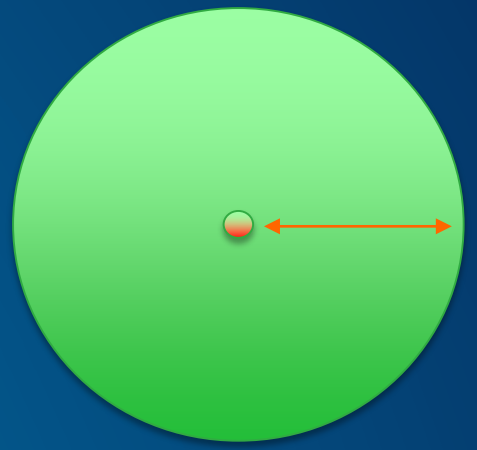
- Contains
- Crosses
- Intersects
- Nearest vertex
- Overlaps
- Touches
- Within

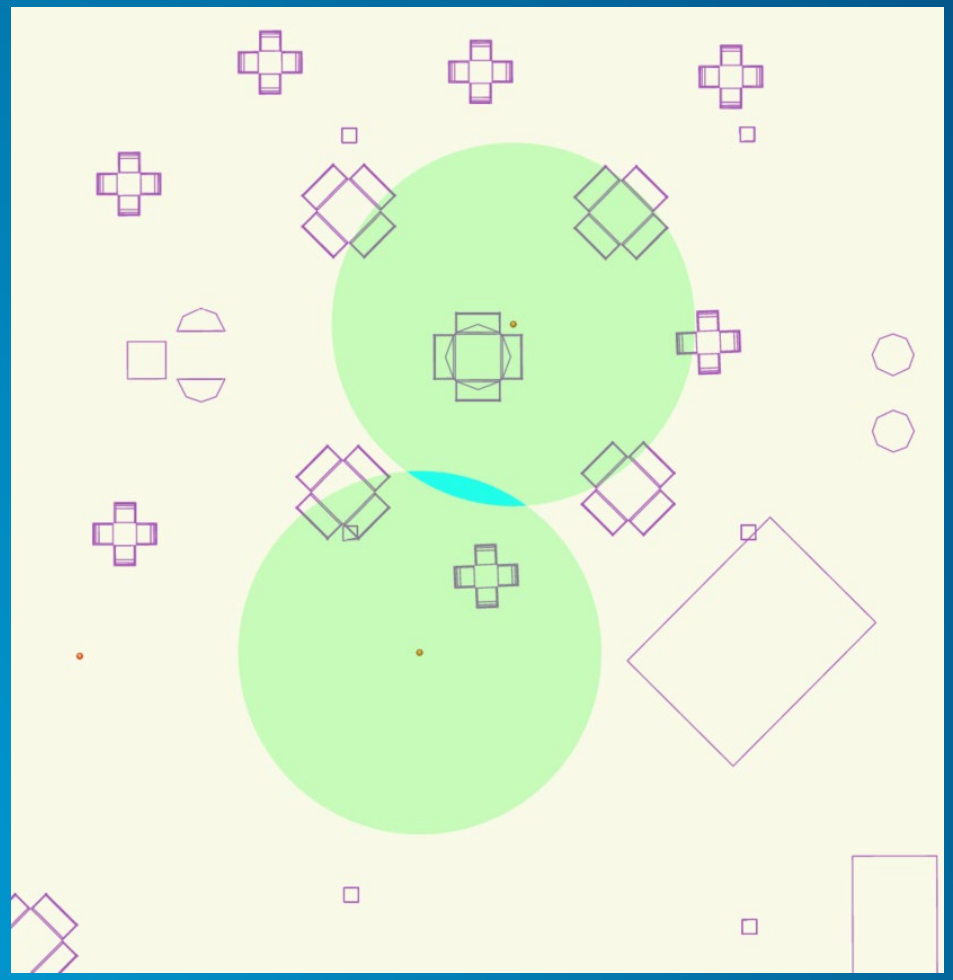
Java sample

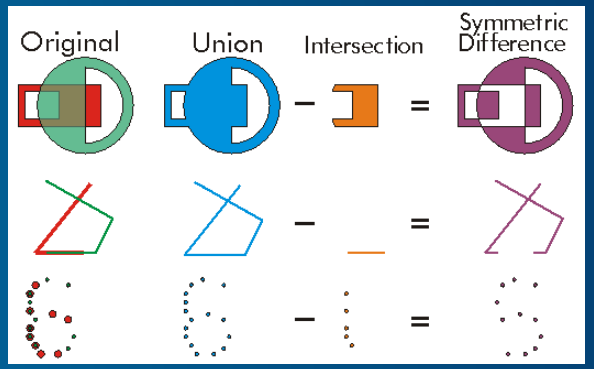
```
// does line cross polygon?
if (GeometryEngine.crosses(polyline, polygon)) {
    // color it green to say yes
    fillSymbol.setColor(0xFF00FF00);
} else {
    // color it red to say no
    fillSymbol.setColor(0xFFFF0000);
}
```

Create geometries based on geometries

- Buffer
- Union
- Intersects
- Difference



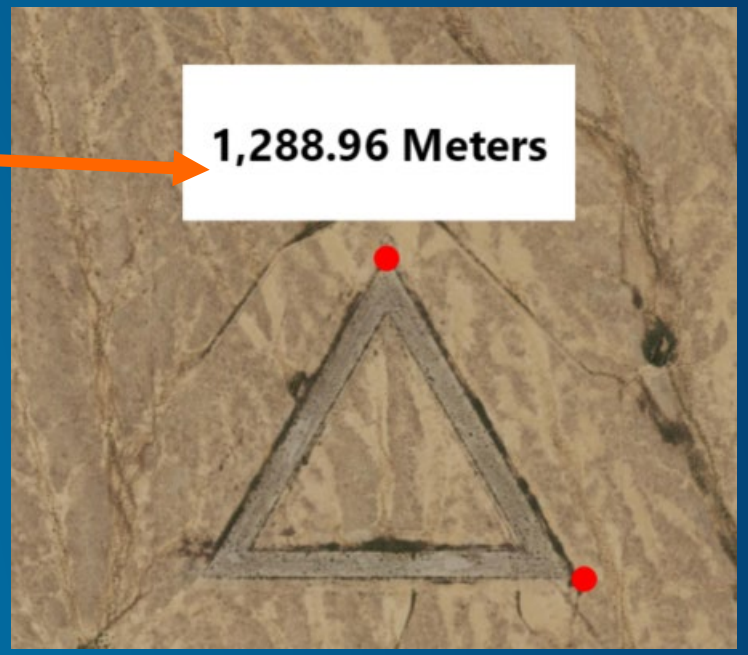


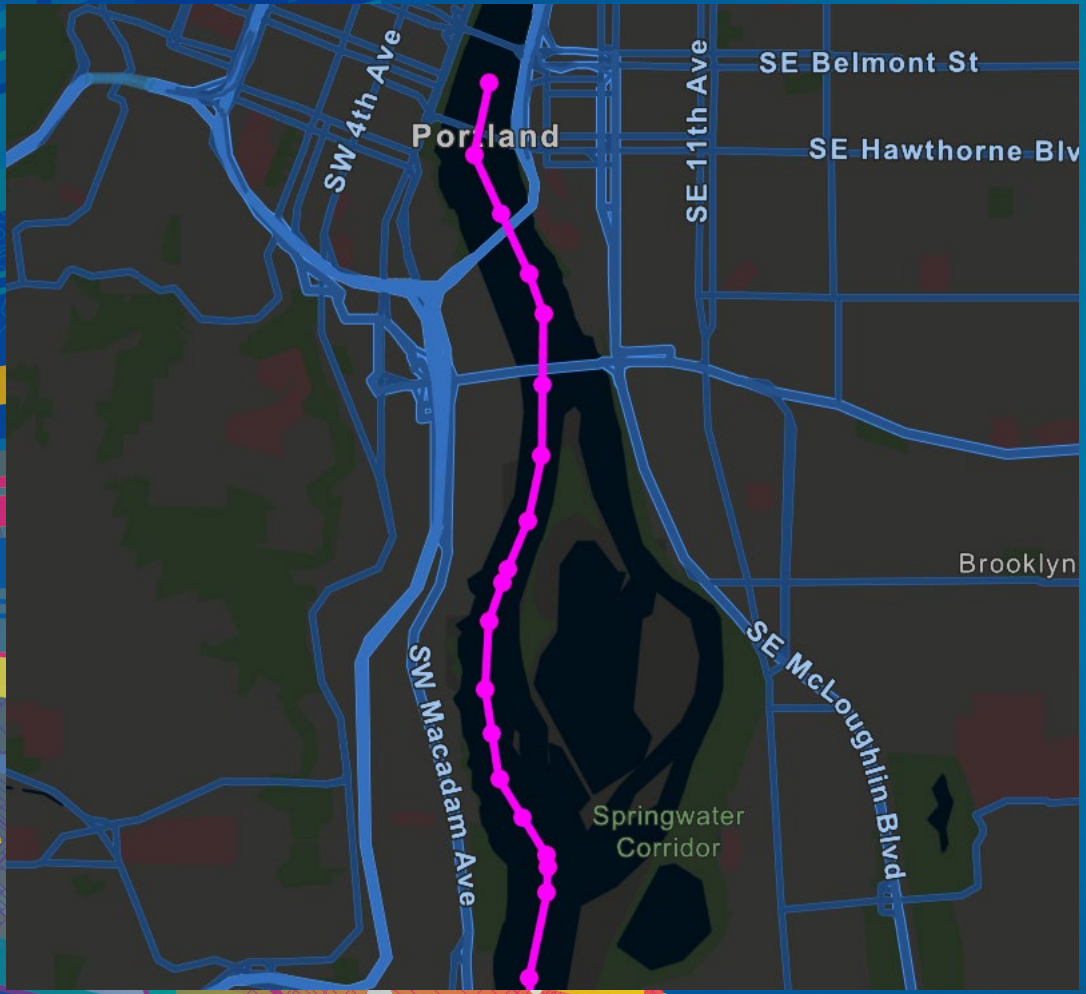


Plus many more…

- Label point

```
GeometryEngine.labelPoint(myPoly)
```





Demo: GeometryEngine

Lucas Danzinger

Geoprocessing

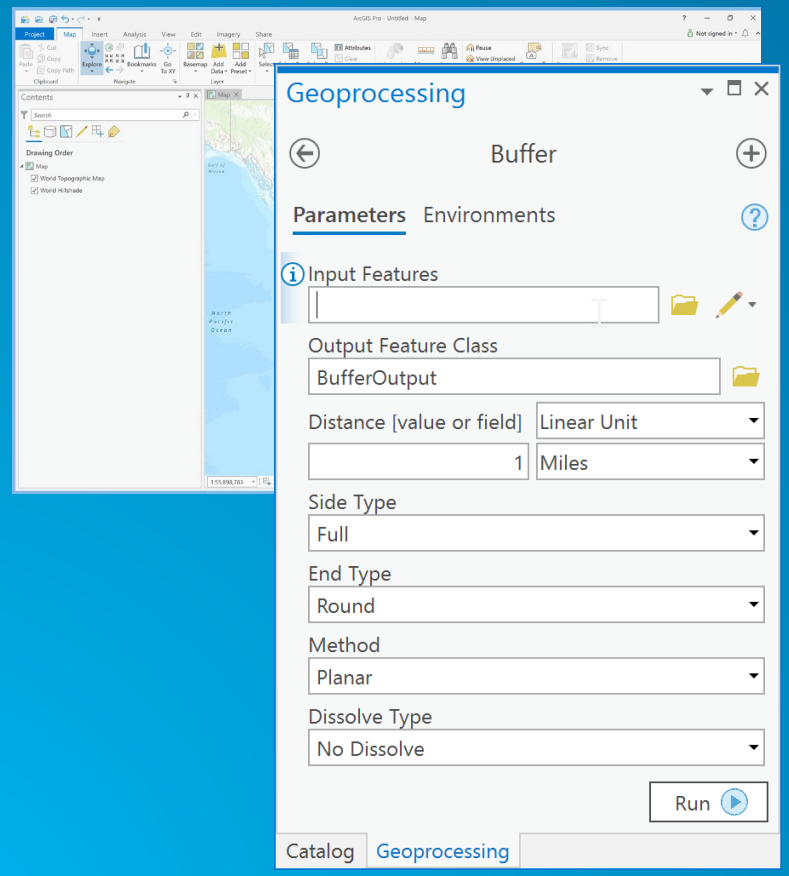
Mike Branscomb

Geoprocessing: Overview

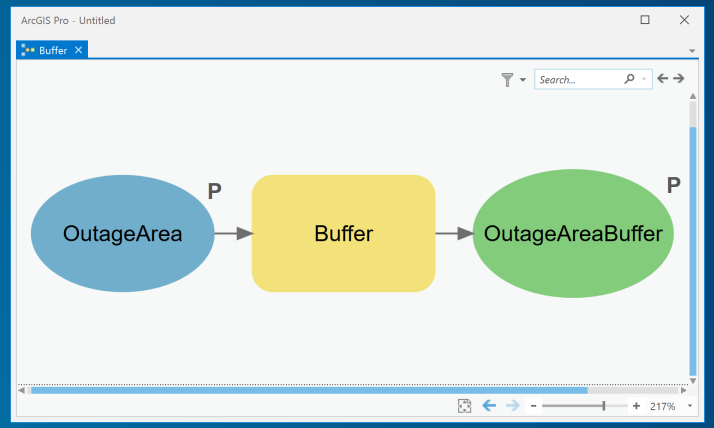
- Framework and tools for processing spatial data
  - Spatial analysis e.g. Buffer, Intersect, Viewshed…
  - Data management e.g. Create Feature Class, Add Field, Add Domain…
- More than 1200 built-in tools installed with ArcGIS Desktop
- Combined into custom tools/workflows
  - Model Builder
  - Python scripts
- Published to ArcGIS Enterprise as services
- Available `offline` as local services with ArcGIS Runtime Local Server
- Provide REST endpoint for service execution

Geoprocessing: Framework

- ArcGIS tools



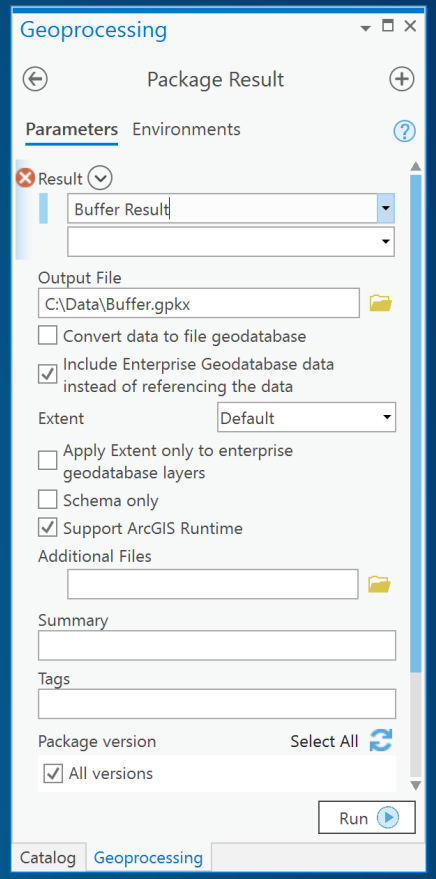
- Custom tools




```python
import os
import arcpy

# Set the workspace for ListFeatureClasses
arcpy.env.workspace = "c:/base"

# Use the ListFeatureClasses function to return a list of
#  shapefiles.
featureclasses = arcpy.ListFeatureClasses()

# Copy shapefiles to a file geodatabase
for fc in featureclasses:
    arcpy.CopyFeatures_management(
        fc, os.path.join("c:/base/output.gdb",
                         os.path.splitext(fc)[0]))
```

Geoprocessing: Services

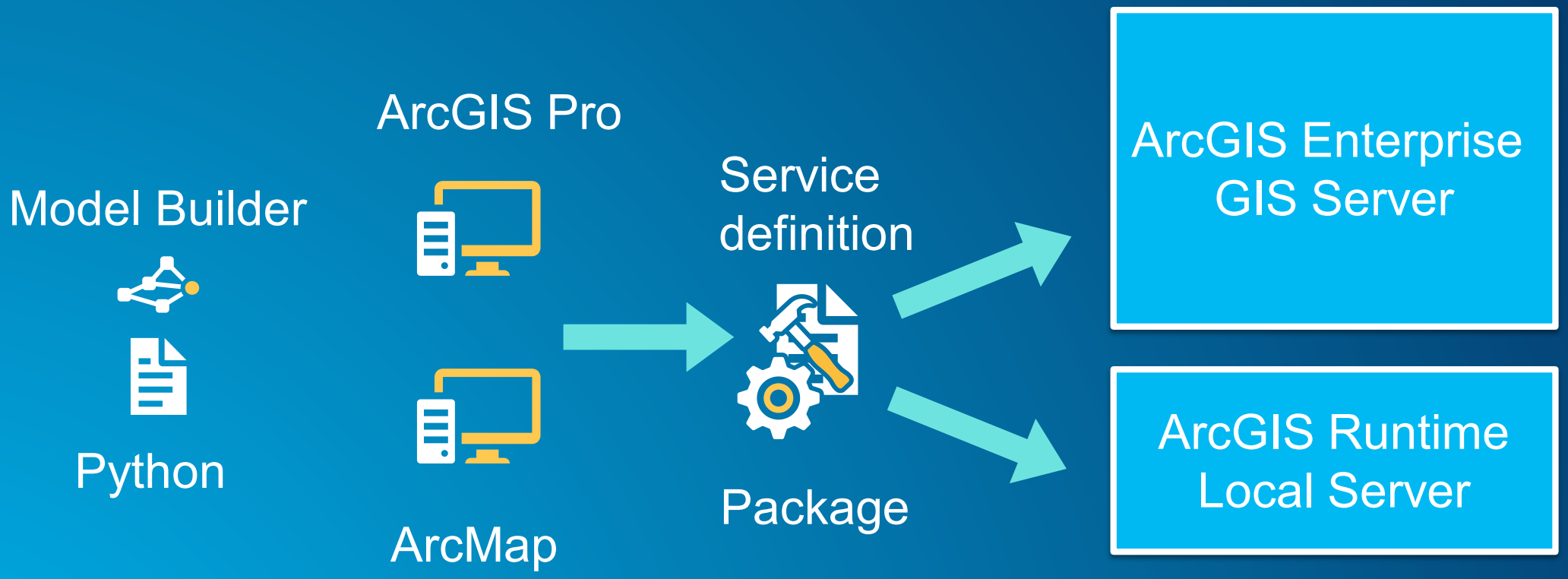
- Must run tool in ArcGIS Desktop before publishing
  - Validates inputs, processing and outputs
- Publish result to ArcGIS Enterprise
  - Result includes input data, output data, project data
- Runtime API provides types to represent geoprocessing
  - Service (task)
  - Specific requests to run (jobs)
- Build parameter objects, send to service, and await results
- Process output parameters
  - e.g. features, tables, raster datasets, values



Geoprocessing: Local Services

- ArcGIS Runtime Local Server includes subset of tools
  - Approx. 500 tools (data management, analysis, extensions)
- Result must be packaged
  - Geoprocessing Package (.gpk / .gpkx)
- Package includes all data required for execution of tool as a service
  - Input data, project data, output data
- LocalServer API includes administration
  - e.g. LocalGeoprocessingService.StartAsync(ServiceType.Asynchronous)

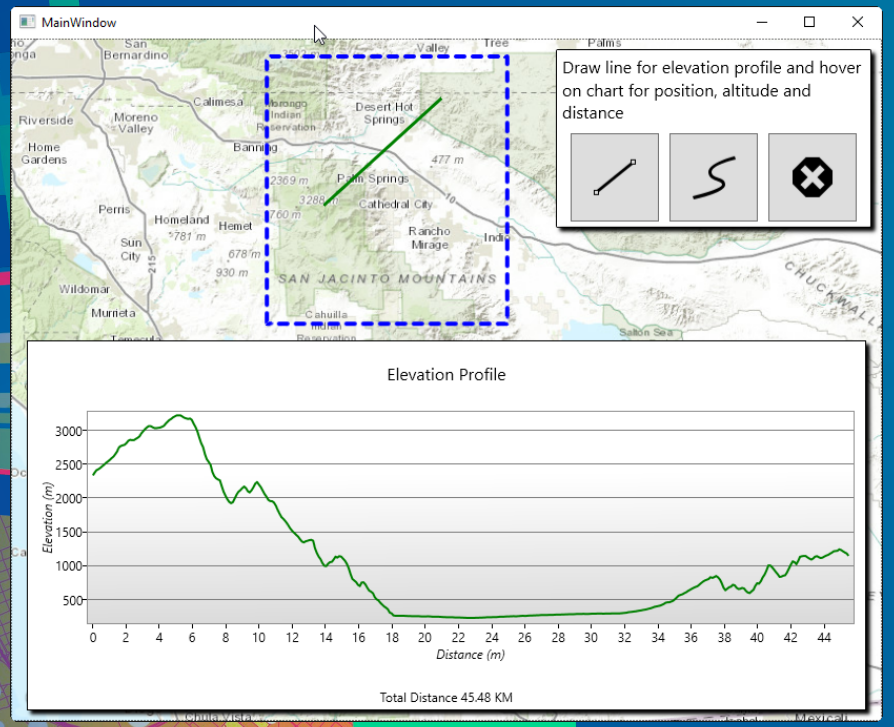
Publishing Geoprocessing services



Geoprocessing: Tips for building geoprocessing tools

- Keep input data and output data small
  - Determines size of the .gpk/.gpkx or Service Definition
- Use local data to ArcGIS Server / LocalServer
- Write intermediate data to memory
- Add attribute and spatial indexes
- Avoid unnecessary coordinate transformations
- Reduce data size

Server Help > Performance Tips for Geoprocessing Services



Demo: Geoprocessing

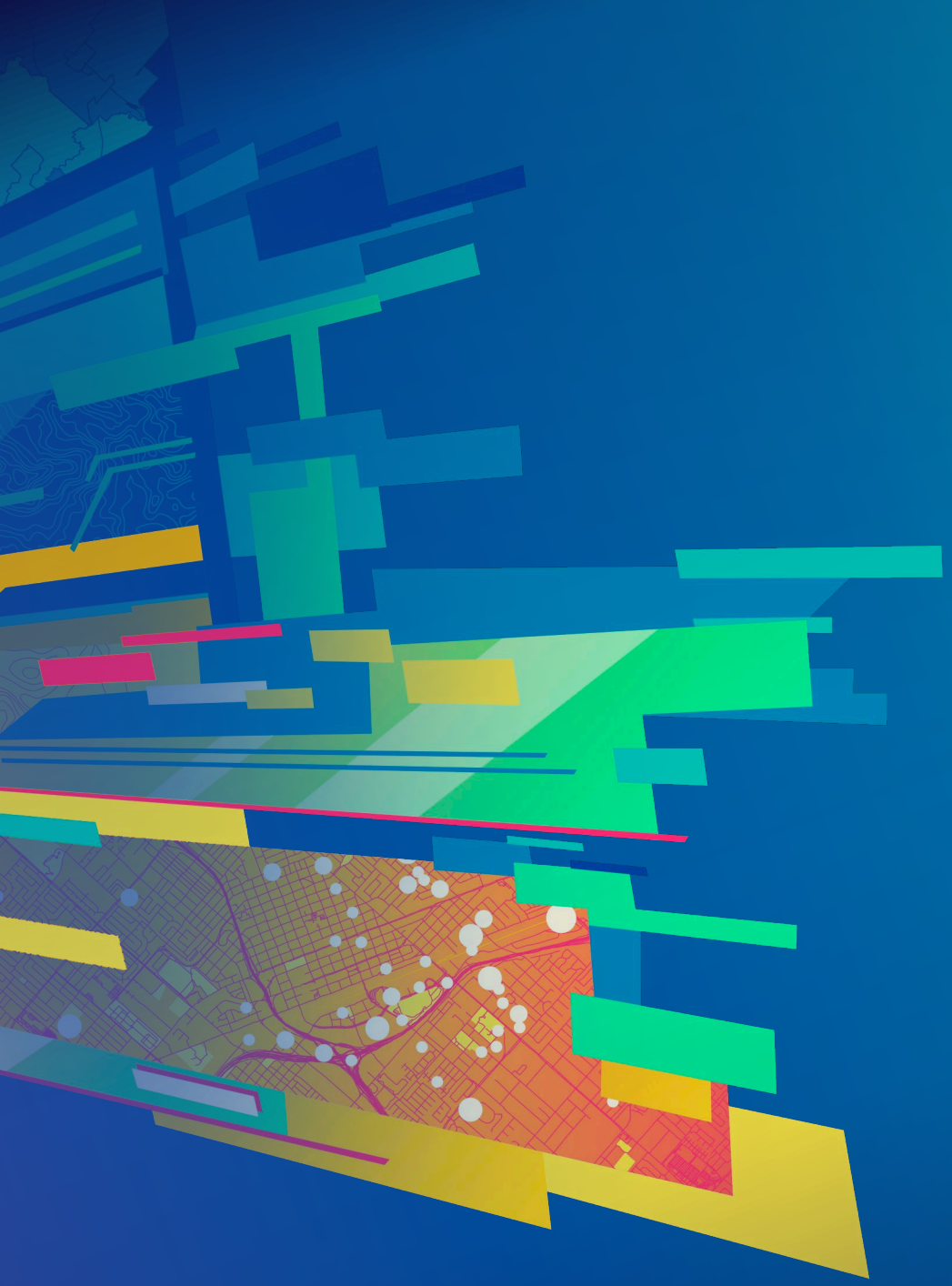
Mike Branscomb

Interactive Visual Analysis

Mike Branscomb

Interactive Visual Analysis

- Available analyses:
  - Line of Sight
  - Viewshed
  - Location Distance Measurement
- Fast GPU-based operations
- Acts on current visible resolution of data
- Results are displayed in SceneView
- Contrasts with geoprocessing
  - Acts on full resolution of data
  - Results persisted as new datasets

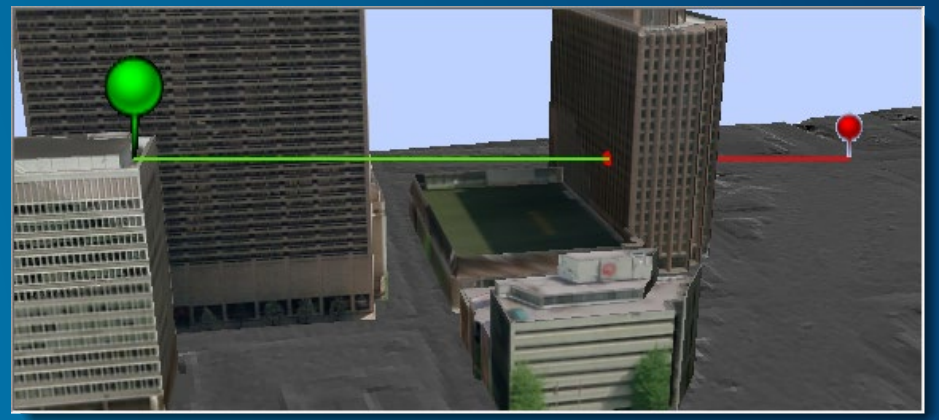


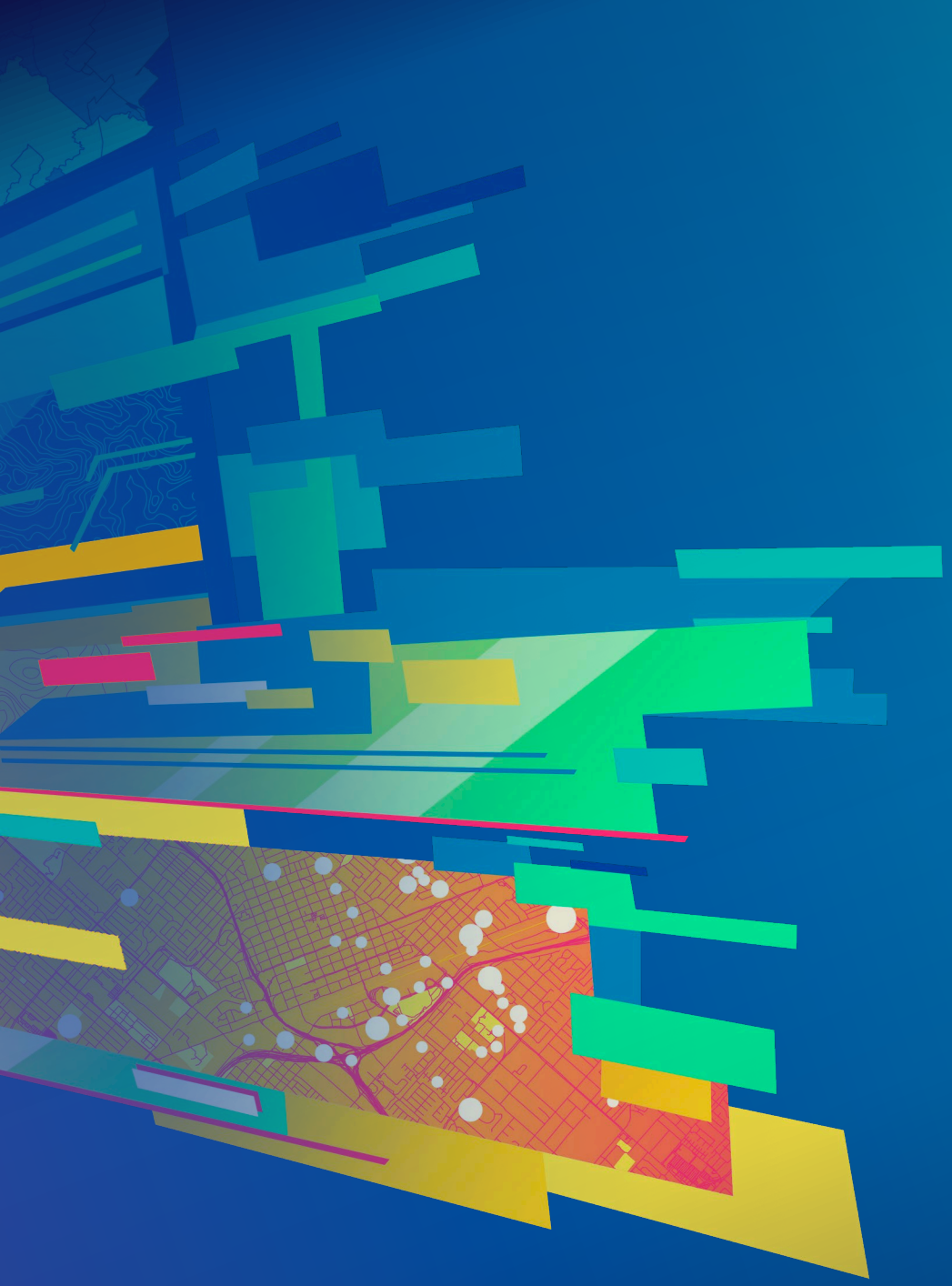
Demo: Line of Sight

Mike Branscomb

Line of Sight – Interactive Visual Analysis

- Determines the visible and not visible portions of an imaginary line between a target and an observer
- Output is a line graphic symbolized by visible areas and obstructed areas
- LocationLineOfSight – calculates against 2 points
- GeoElementLineOfSight – calculates between 2 GeoElements
  - Can attach to 2 GeoElements to auto update
- Events triggered when target is visible



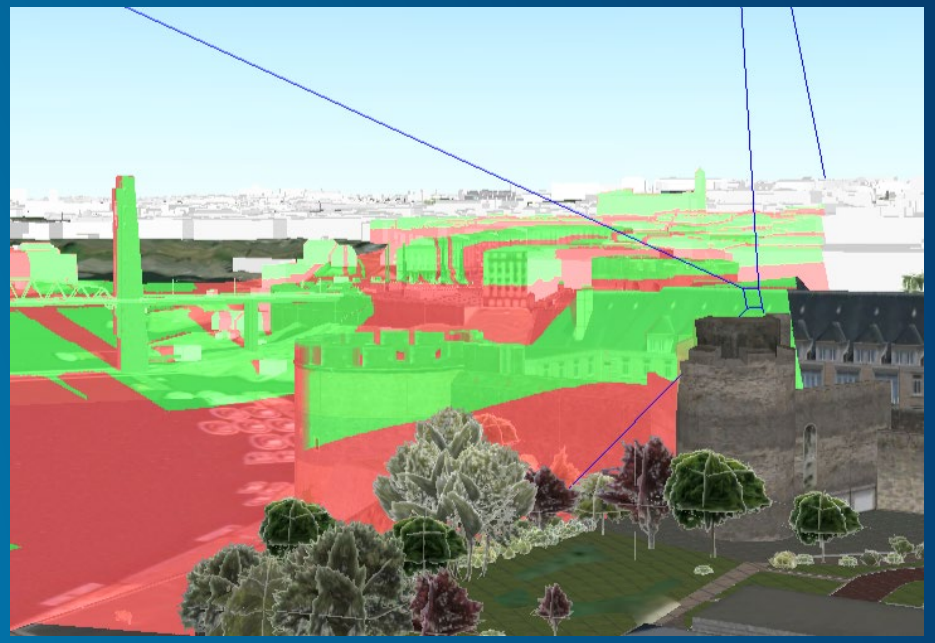


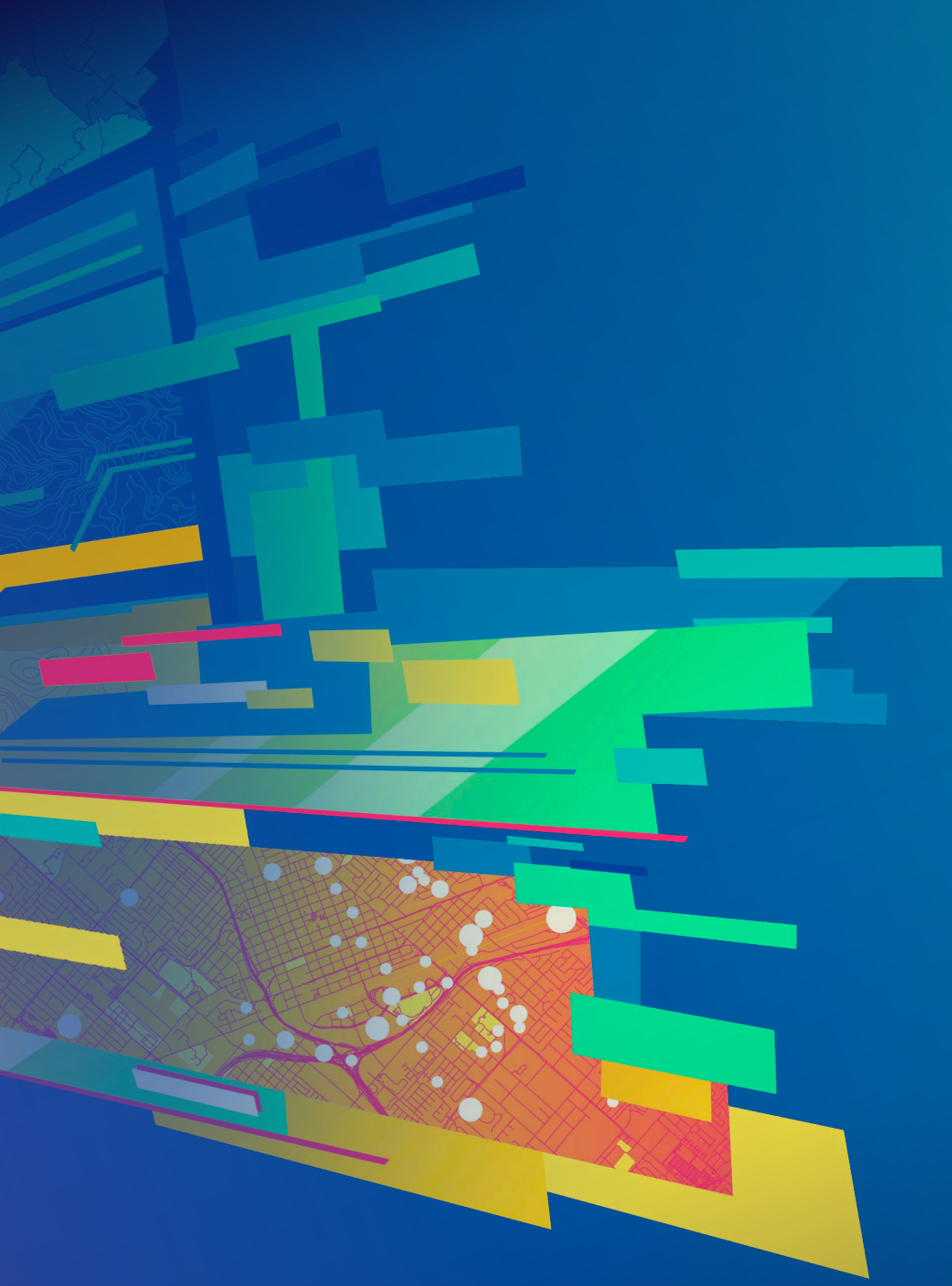
Demo: Viewshed

Mike Branscomb

Viewshed – Interactive Visual Analysis

- Determines the visible and not visible portions of a surface based on an observer point
- Output is a raster symbolized by visible areas and obstructed areas
- LocationViewshed
  - Camera
- LocationViewshed
  - Point
- GeoElementViewshed
  - Attach to a GeoElement





Demo: Location Distance Measurement

Mike Branscomb

Location Distance Measurement – Interactive Visual Analysis

- Interactive distance measurements between two locations in a SceneView
- Input locations are 3D points
- Measure between locations on, above, or below the surface
- Measurements are planar
  - Accurate for relatively short distances
  - Terrain not considered in measurement
- Direct distance
  - Straight-line distance between start and end locations
- Horizontal distance
  - Planar distance between the start and end locations as projected onto the surface of the earth
- Vertical distance
  - The difference in elevation (z-value) between the start and end locations

Summary & Roadmap

Mike Branscomb

Summary

- ArcGIS Runtime enables spatial analysis in your native apps
- Common to combine approaches
  - Geometry Engine > Geoprocessing > Visualization
- Local to the device
  - Offline workflows
- In your enterprise or the cloud
  - Online service-based workflows

Roadmap for analysis in ArcGIS Runtime 100.x (Dev Summit 2018)

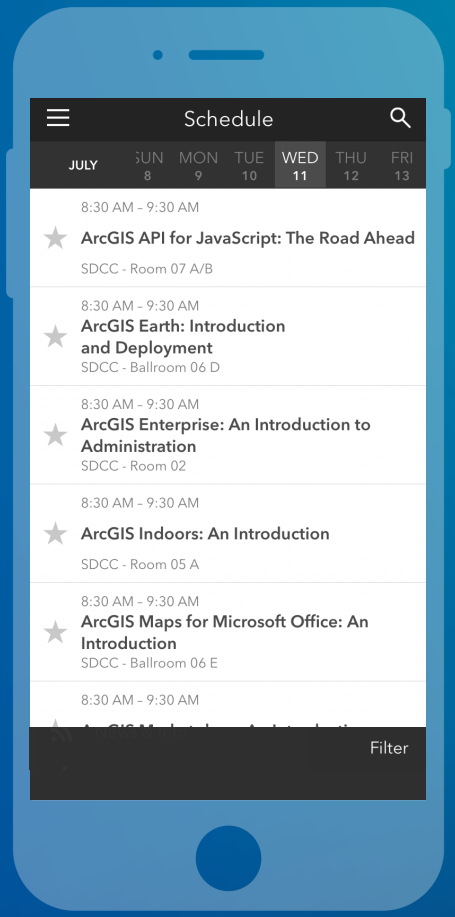
- ✓ Geoprocessing additional tools for Local Server
- ✓ More 3D exploratory analysis tools e.g. 3D distance
- ✓ Local Server support for ArcGIS Pro packages
- ✓ Local Server support for ArcMap 10.6 packages

- 2019: ArcGIS Runtime: Road Ahead
  - Thursday March 7th 5:30 pm - 6:30 pm
  - Primrose B

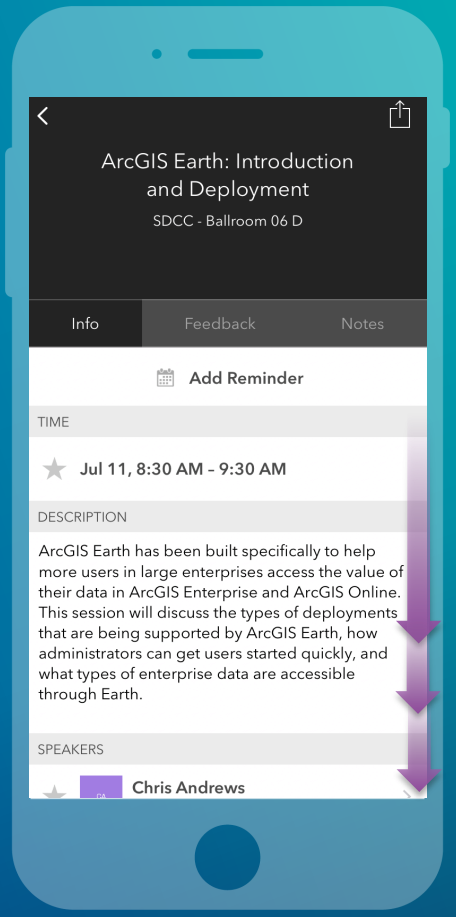
Please Take Our Survey on the App

Download the Esri Events app and find your event

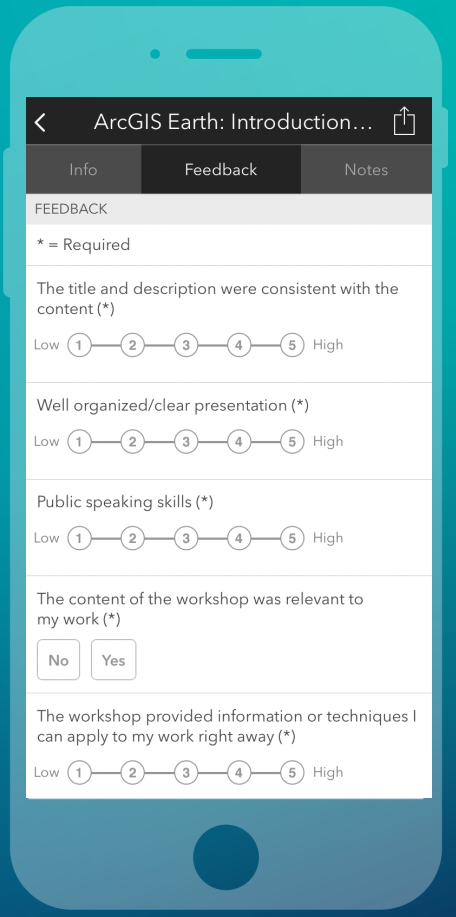
Select the session you attended



Scroll down to find the feedback section



Complete answers and select “Submit”



esri

THE SCIENCE OF WHERE