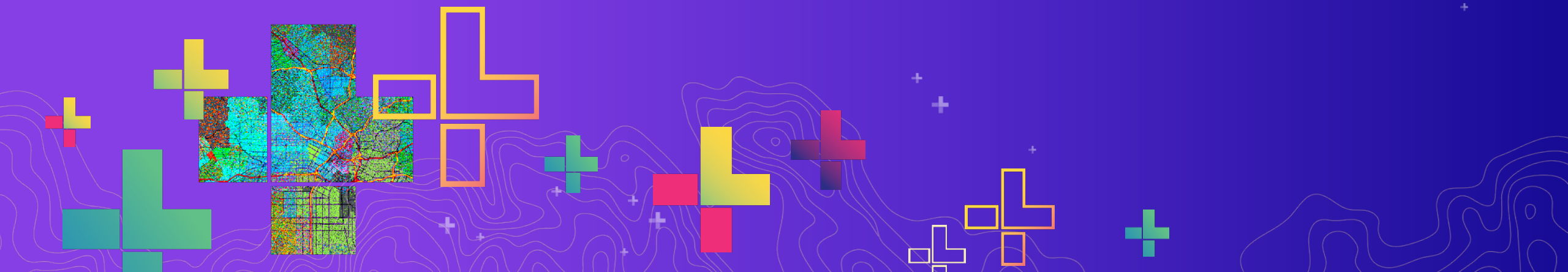




# ArcGIS GeoEvent Server: Applying Real-Time Analytics

Gregory Christakos & RJ Sunderman

2020 ESRI DEVELOPER SUMMIT | Palm Springs, CA



# Agenda:

---

- 1 Performing Analysis in Real-Time
  - 2 Use Case 1: Identifying Conditions
  - 3 Use Case 2: Situational Awareness & Response
  - 4 Summary & Resources
-



# Performing Analysis in Real Time



# ArcGIS GeoEvent Server

*Real-Time GIS for ArcGIS Enterprise*



ingestion

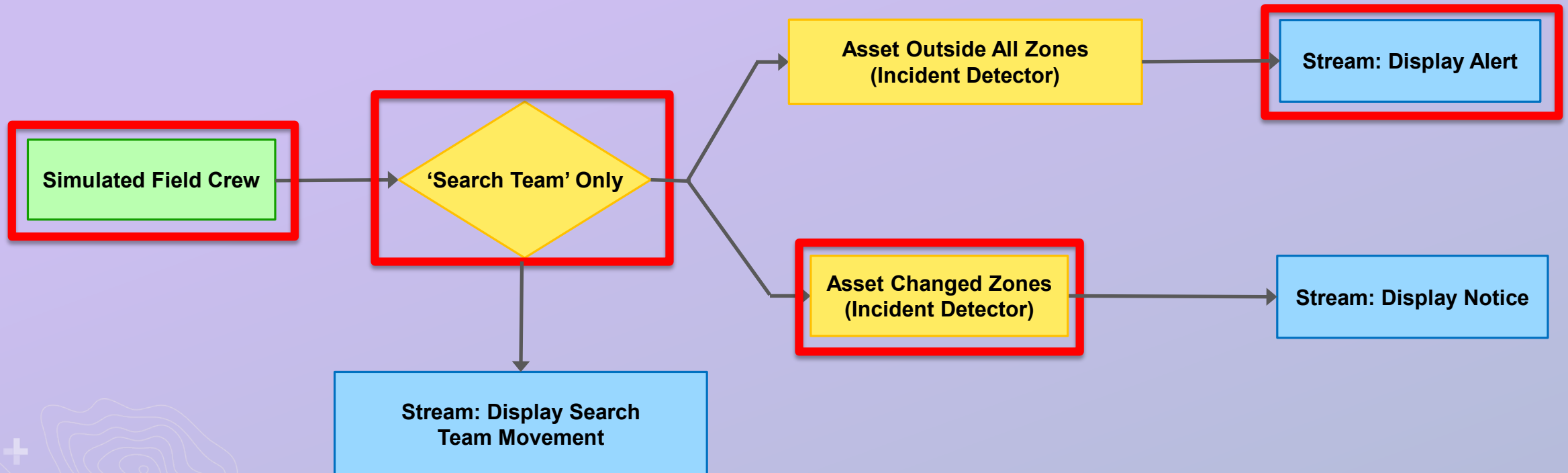
actuation



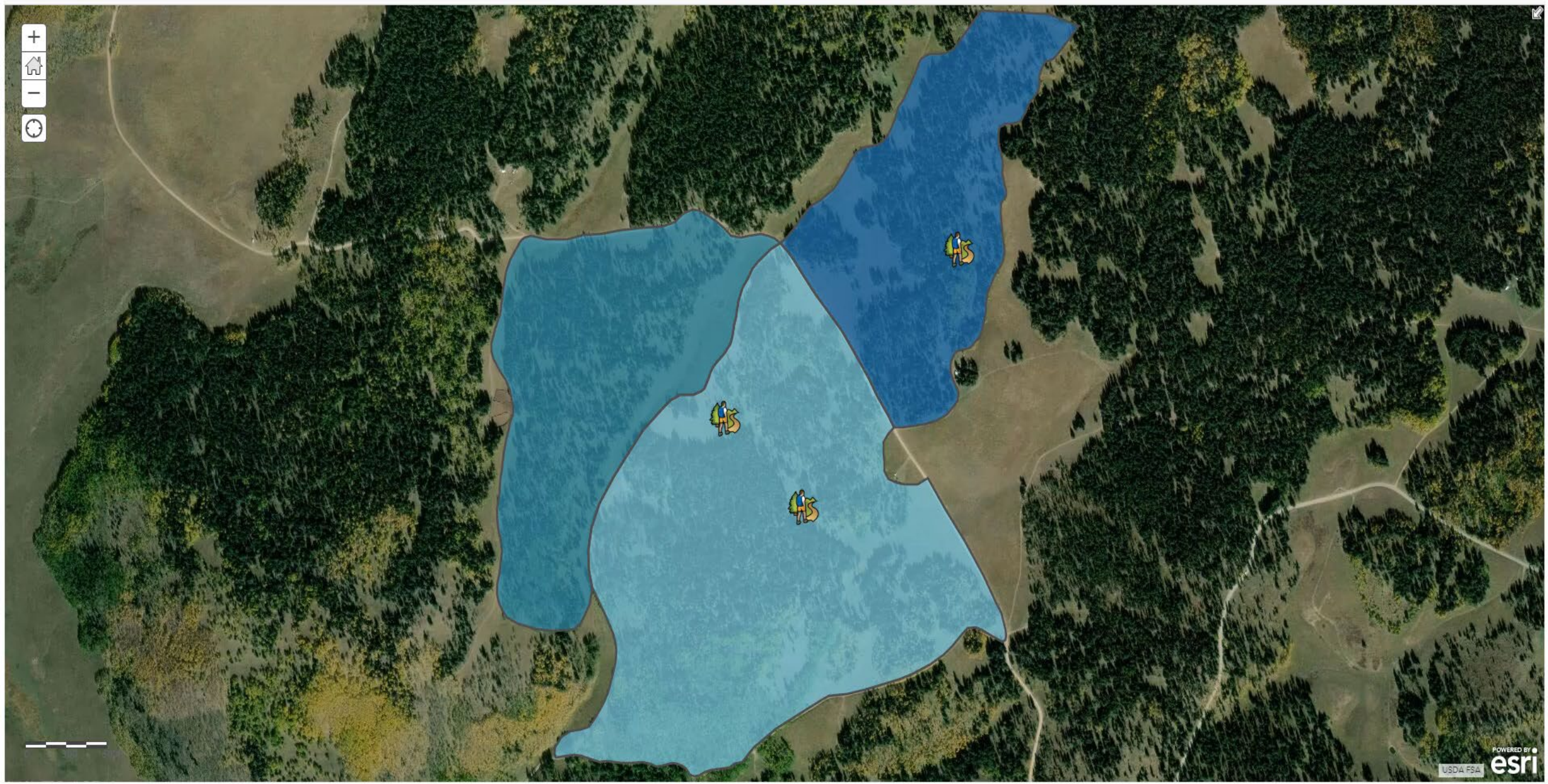
# Real-Time Analysis

## GeoEvent Services

- A **GeoEvent Service** configures the flow of GeoEvents
  - The **Filtering** and **GeoEvent Processing** steps performed
  - The input(s) data comes from and the output(s) to which results are sent





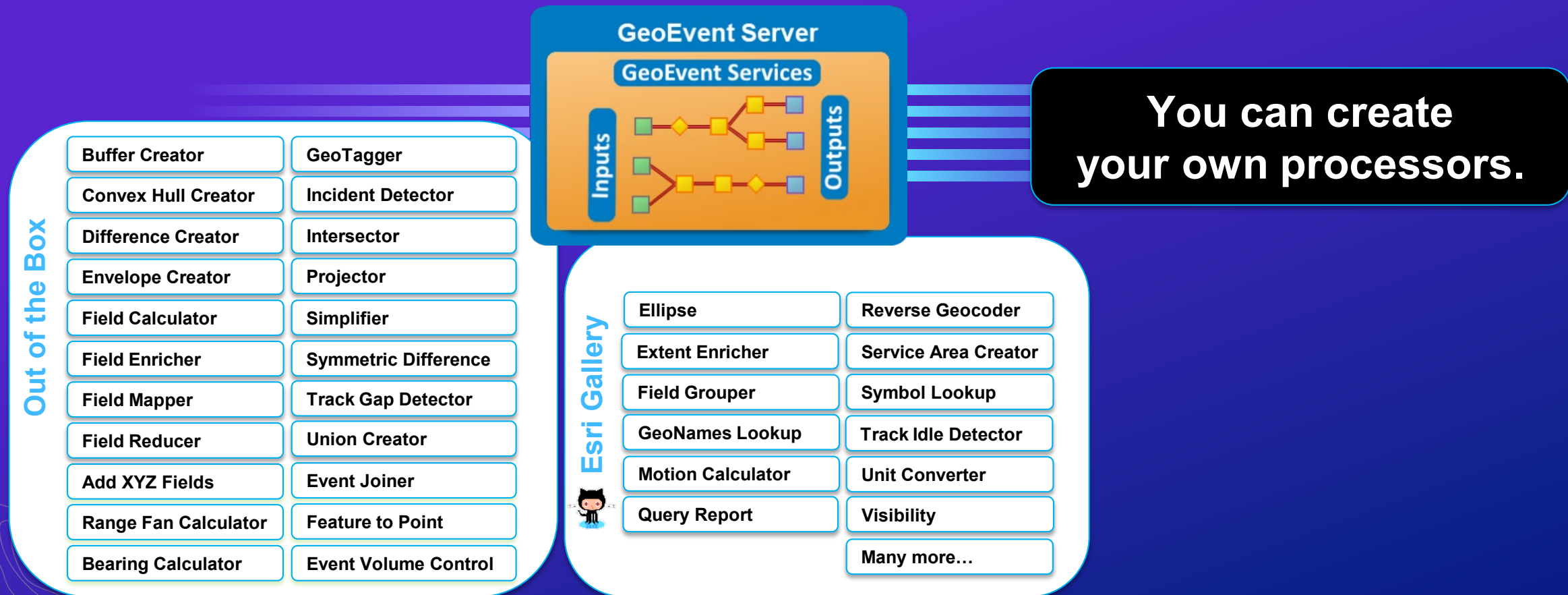




# Real-Time Analysis

## Processors and Filters

- GeoEvent Services apply continuous analysis to event records as they are received using **processors** and **filters**



# Filters

- Allow event records to pass only if a conditional expression evaluates *TRUE*
- The expression can use attributes or geometries established as geofences



Filter Properties

Name:

Ok Cancel

+()

Filter Properties

Name:

Ok Cancel

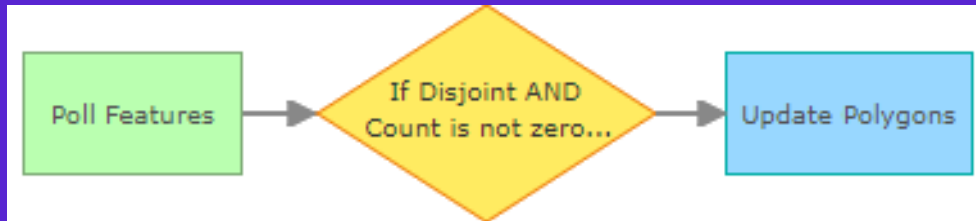
NOT

+()



## Filters (continued...)

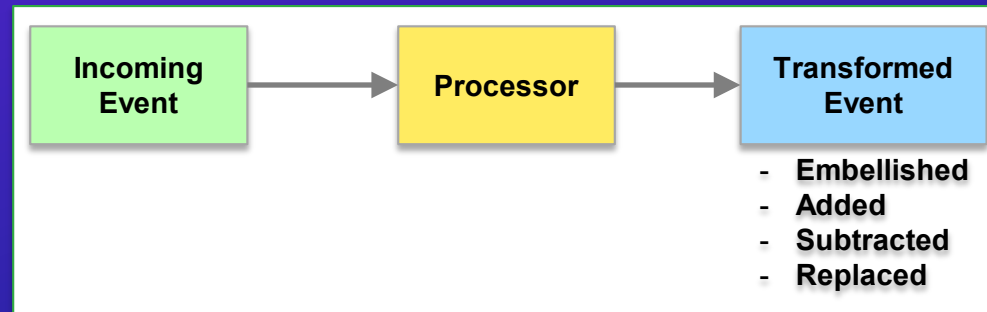
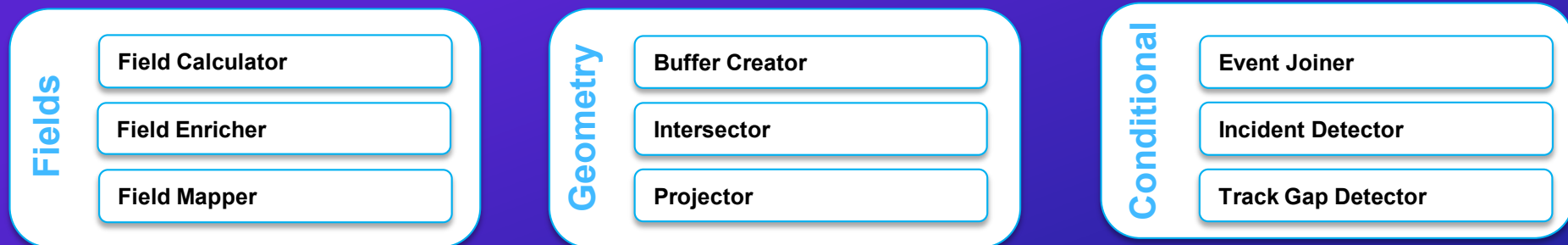
- You can use Boolean logic (*AND*, *OR*, *NOT*) to combine expressions



The screenshot shows a "Filter Properties" dialog box. At the top, the title bar says "Filter Properties". Below the title bar, there is a "Name:" label followed by a text input field containing "If Disjoint AND Count is not zero...". To the right of the input field are "Ok" and "Cancel" buttons. Below the name field, there is a section for the filter expression. On the left of this section is a dropdown menu showing "AND". To the right of the dropdown are two text input fields. The first field contains "GEOMETRY DISJOINT ALL DeliveryArea/\*". The second field contains "count != 0". Below these fields is a green plus icon followed by a blue button with a white plus sign and parentheses "+()".

# Processors

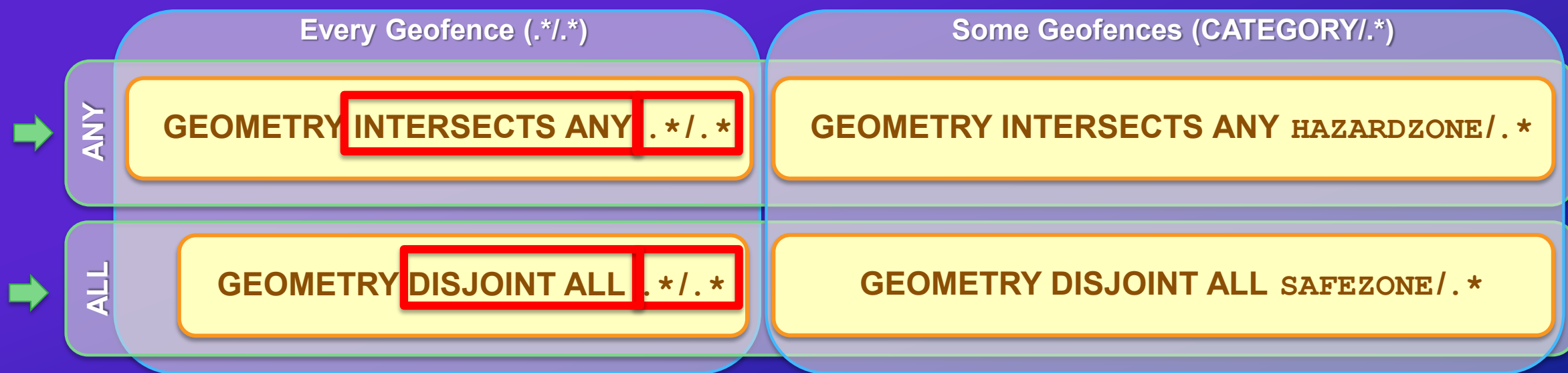
- Processors typically transform the event record being processed in some way
- Many work with attribute values, some apply only to event geometry
- Some watch for or monitor conditions in order to generate new messages



# Spatial Operators

## *GeoFence Selection vs. Spatial Operator Scope*

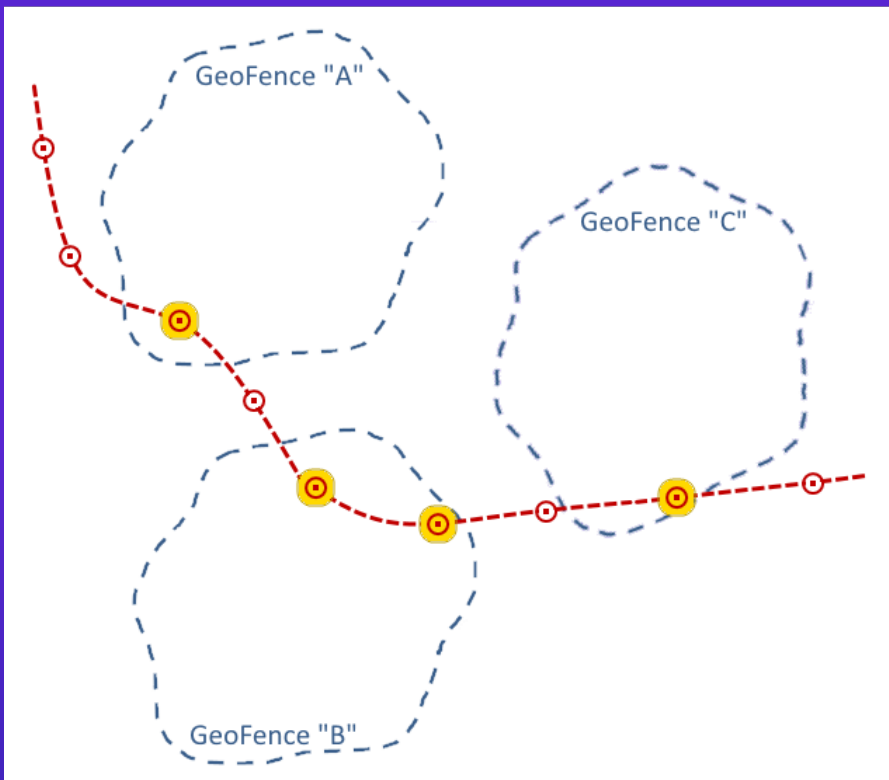
- Do not confuse the regular expression patterns used to select which geofences are considered with the *ANY* or *ALL* qualifier for a specified operation



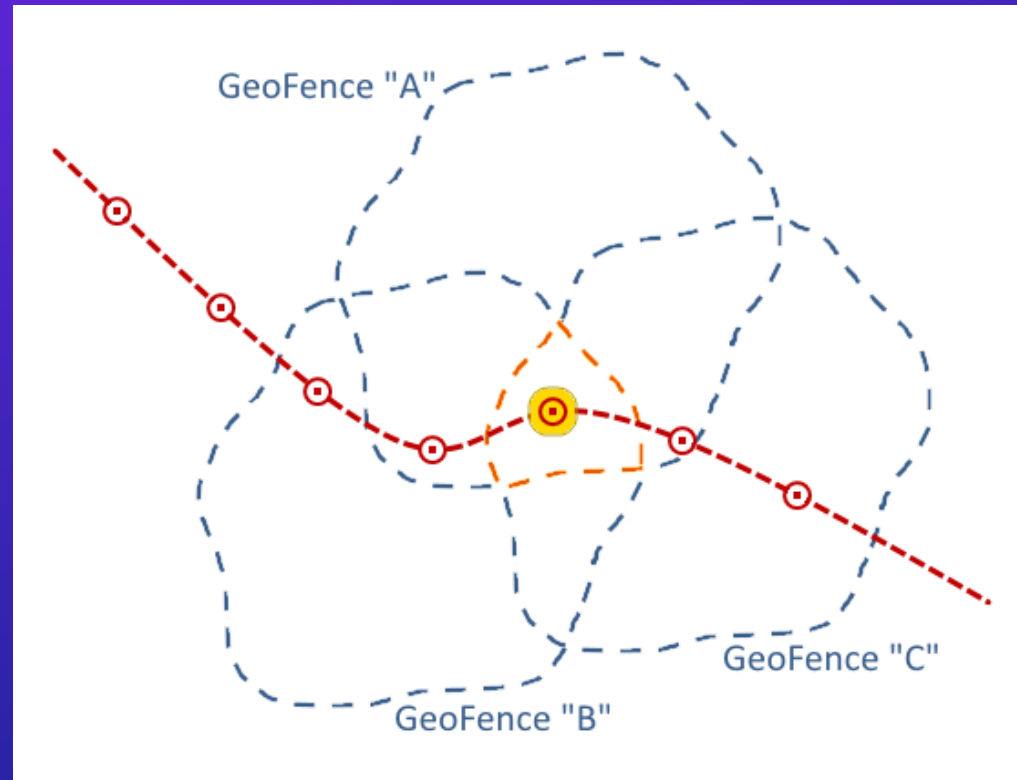
# Spatial Operators

## Overlapping GeoFences

- Intersects *ANY* geofence



- Intersects *ALL* geofences





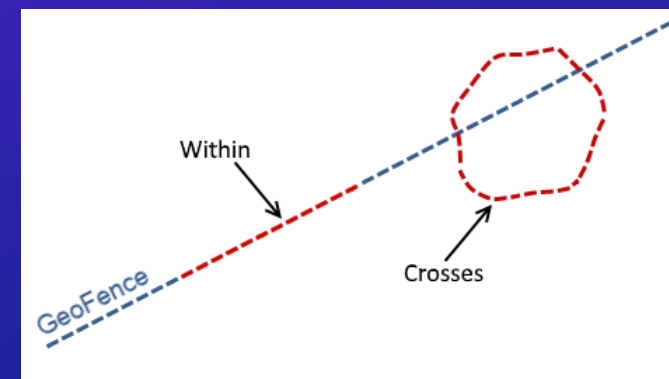
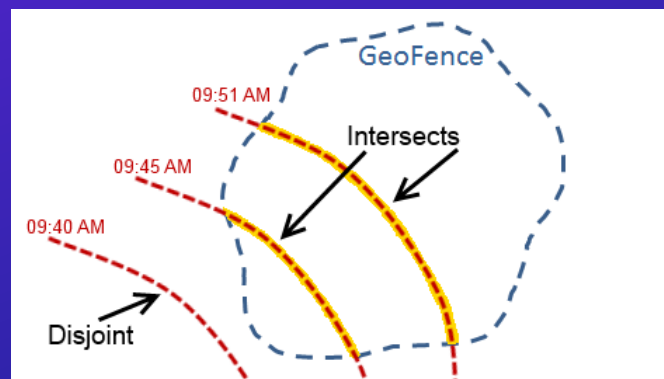
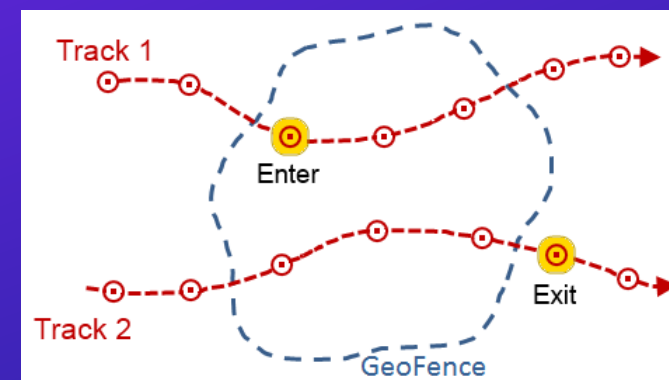
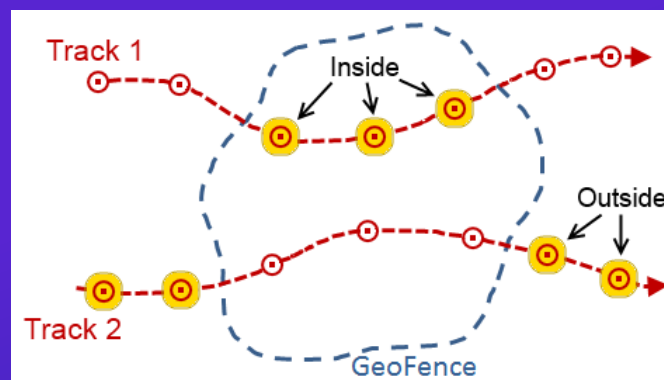
# Spatial Operators

## Stateless

inside	outside
intersect	disjoint
touches	contains
crosses	equals
overlaps	within

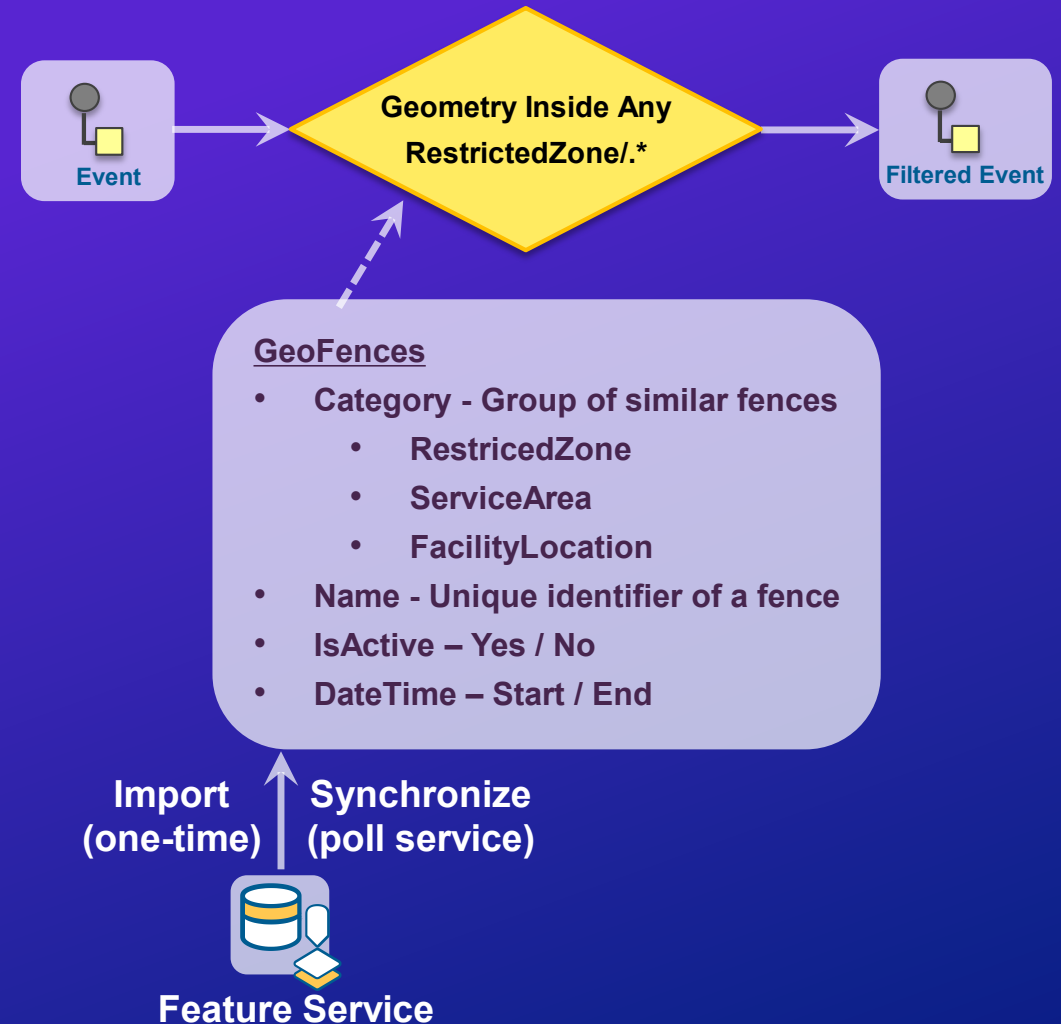
## Stateful

enter	exit
-------	------



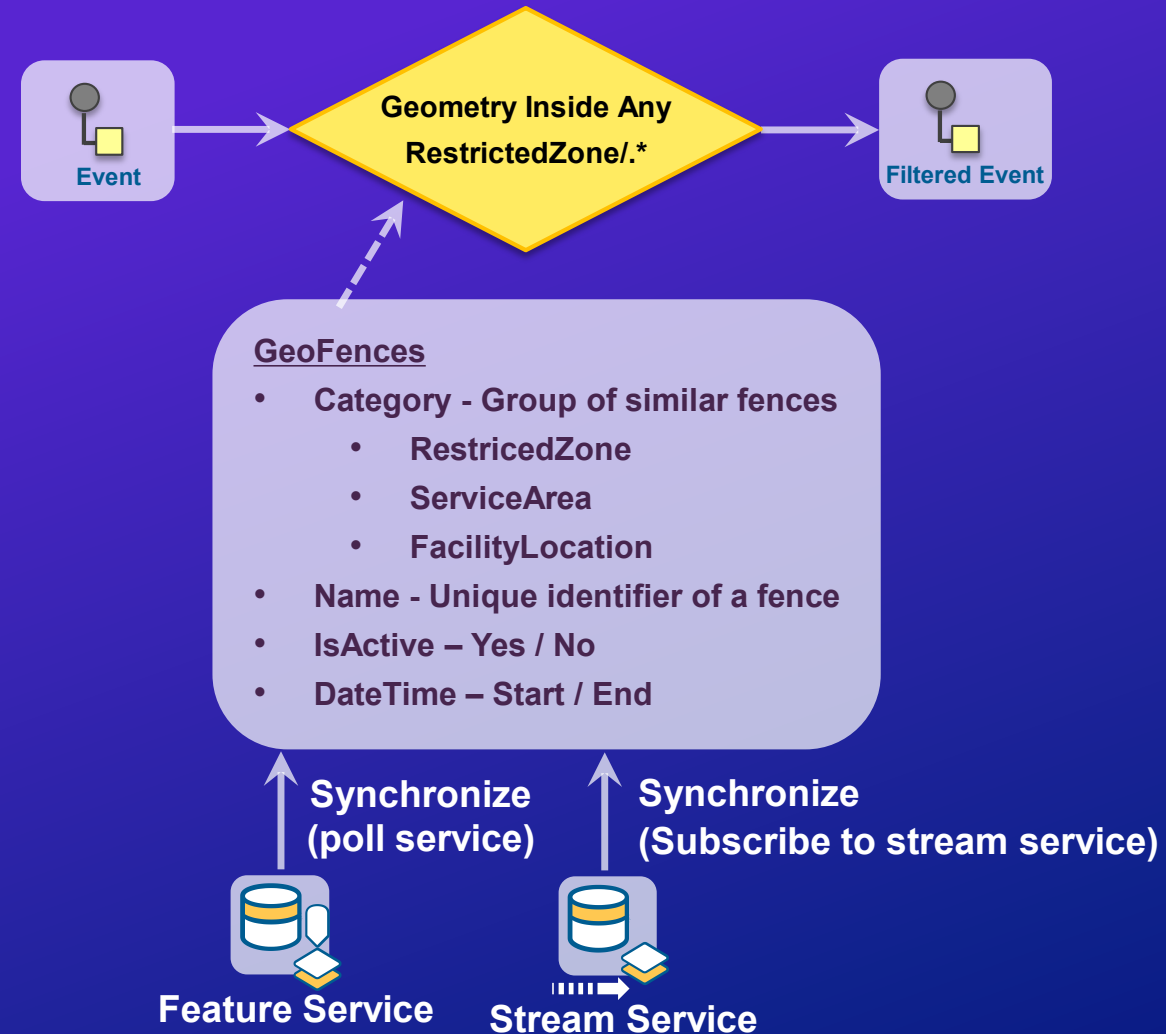
# GeoFence Synchronization

- Import from a feature service
  - Reads once, good for static geofences
- Synchronize with a feature service
  - Periodically refreshes to update geofences
  - Effective when
    - Geometry of a area of interest is changing
    - Date/Time a geofence is effective changes
    - External considerations change when an area should be considered (geofence active / disabled)



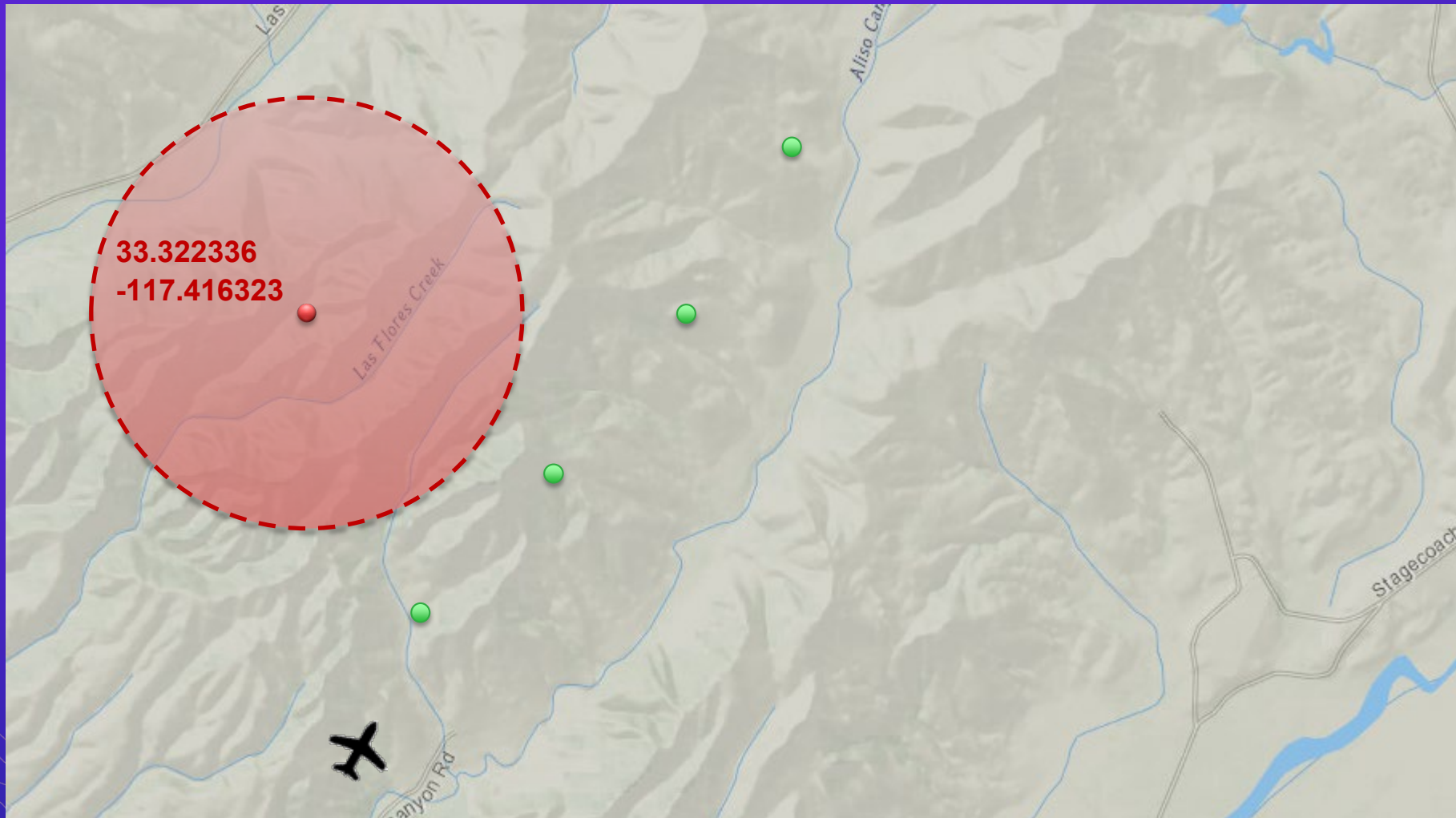
# GeoFence Synchronization

- Import from a feature service
  - Reads once, good for static geofences
- Synchronize with a feature **or stream service**
  - Periodically refreshes to update geofences
  - Effective when
    - Geometry of a area of interest is changing
    - Date/Time a geofence is effective changes
    - External considerations change when an area should be considered (geofence active / disabled)
  - **Requires active management and purging of geofences as they expire**



# Dynamic GeoFences

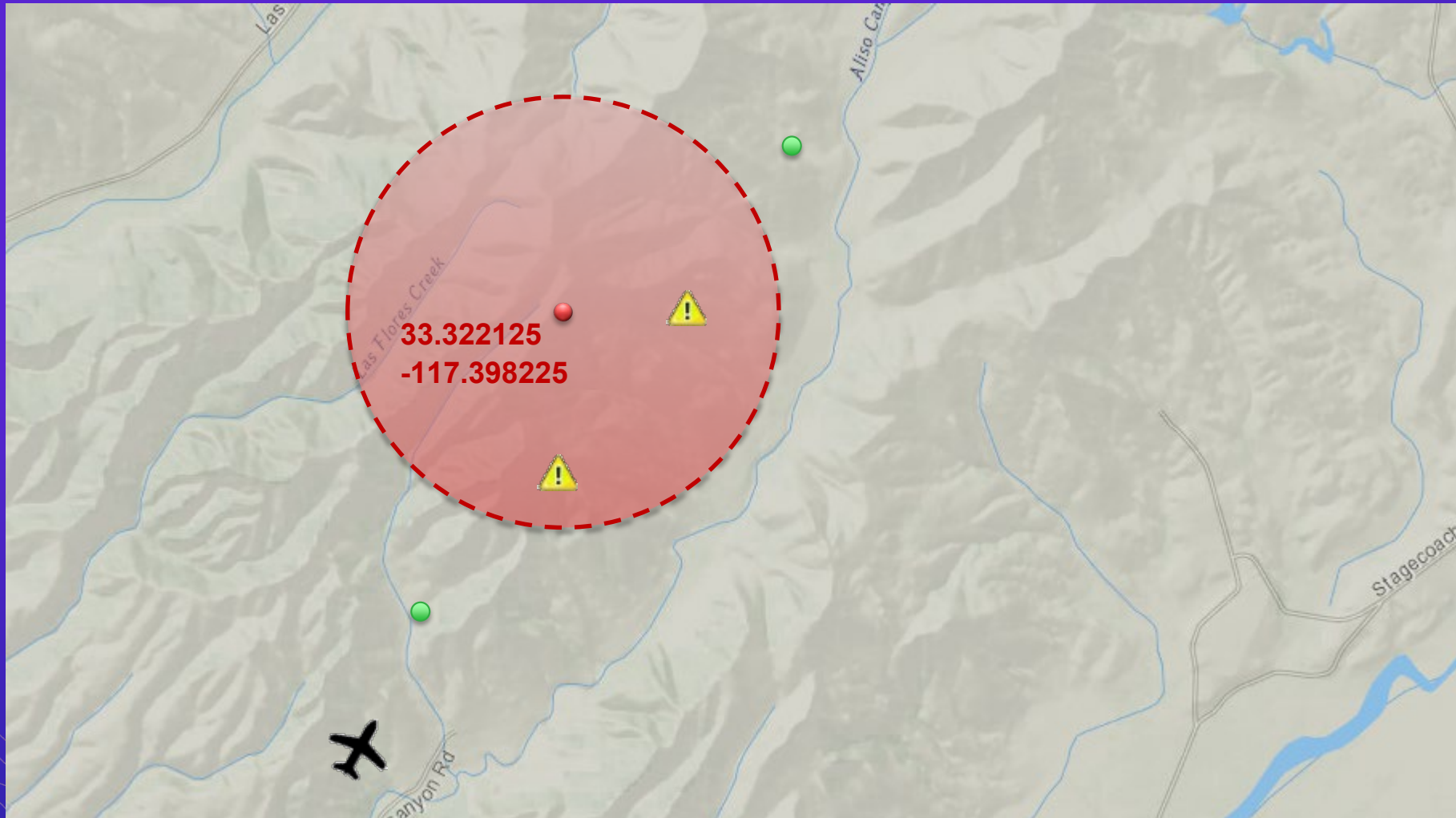
*Check if geometries from two different feeds are spatially coincident*





# Dynamic GeoFences

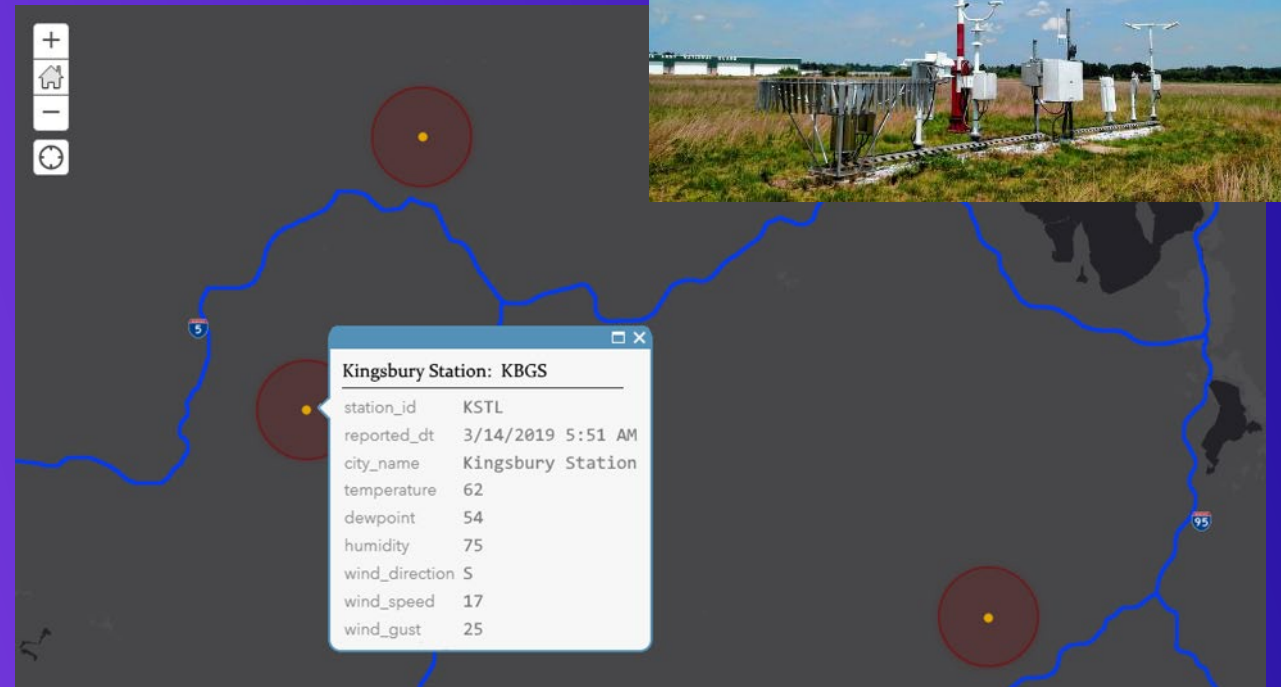
*Check if geometries from two different feeds are spatially coincident*




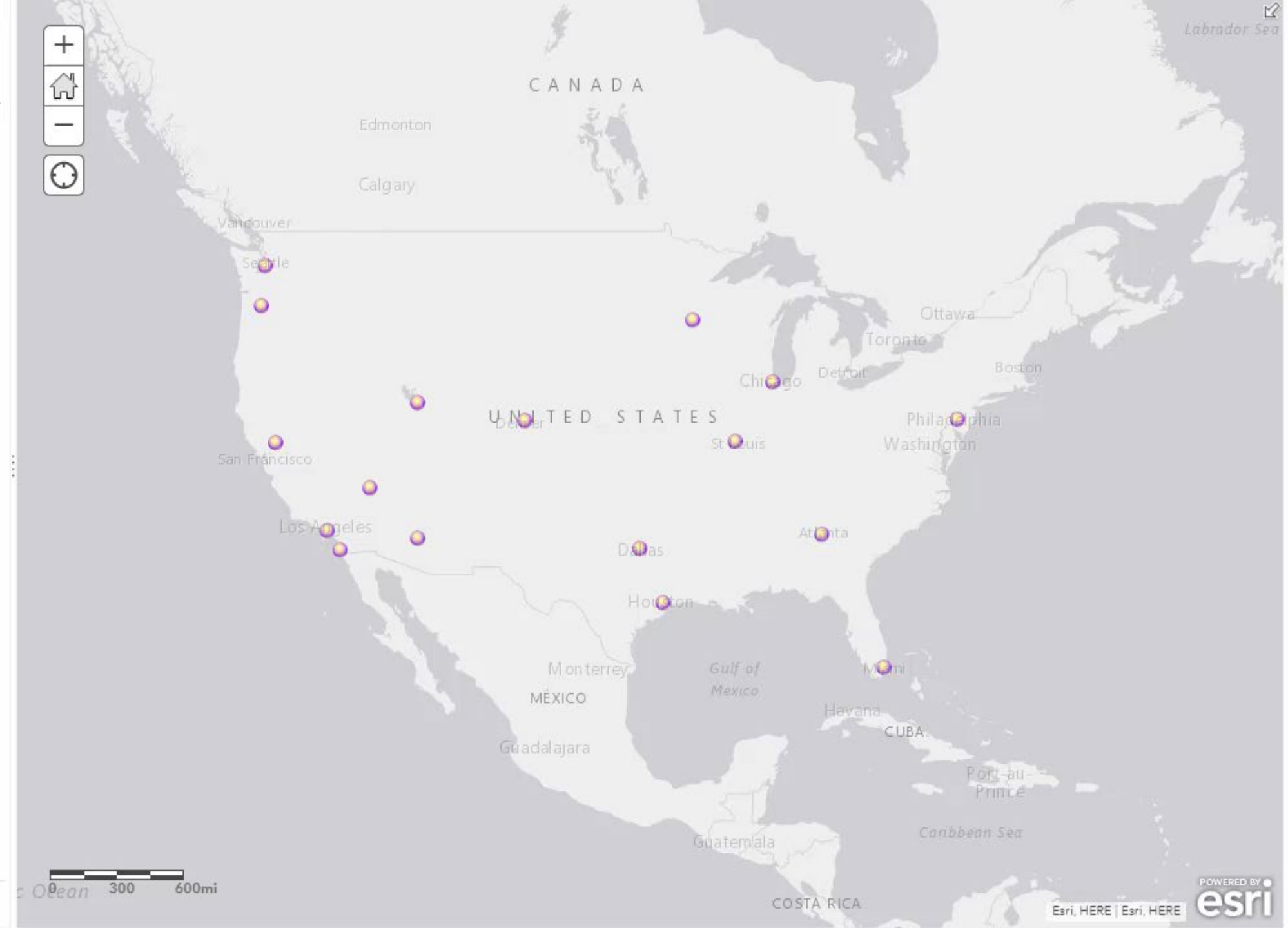
## 2

# Use Case 1: Identifying Conditions

- *Monitor sensors in real-time*
- *Save metrics as feature records*
- *Alert on observed conditions*



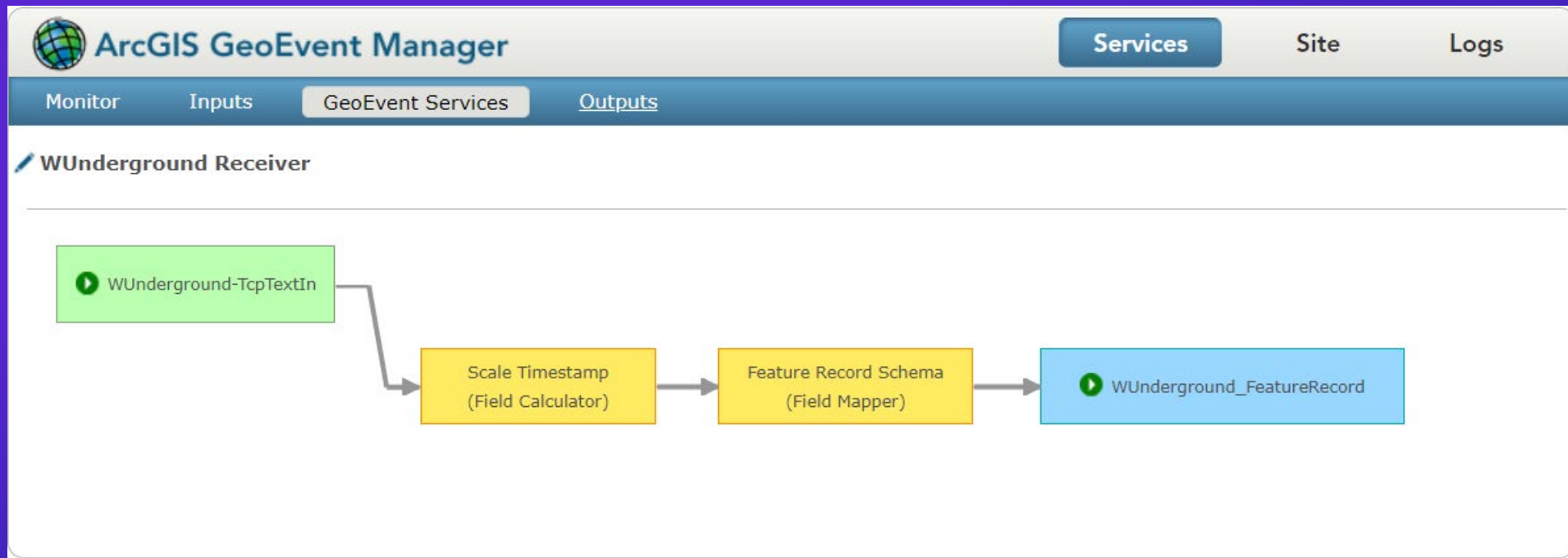
- ☒ WUnderground FeatureRecord
- 
-  Calm Winds
-  High Wind Advisory
-  Other
-  Light Gray Canvas



# Identifying Conditions with Real-Time Data

*Demo 1: Collect sensor network data and persist as feature records using a feature service*

- Key Processors: Field Calculator, Field Mapper



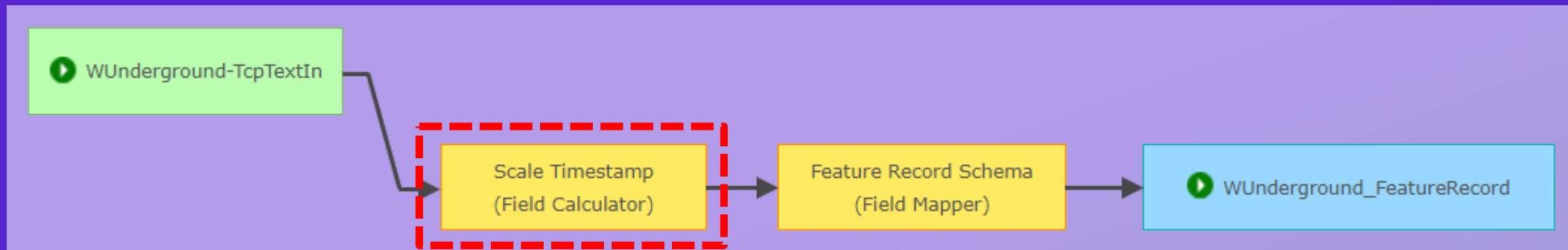


# Field Calculator

- Use a **Field Calculator** when you need to:
  - Calculate new values using data from a received event record
- An expression is evaluated and used to calculate the new values
  - Results can be written to a new field or used to update an existing attribute
  - Expressions can be mathematical or perform string manipulation
- Expressions can also invoke functions
  - Some functions support powerful regular expression pattern matching



# Field Calculator



Processor Properties

Name\*: Scale Timestamp

Processor: Field Calculator

Expression\*: Epoch \* 1000

Target Field\*: Existing Field

Existing Field Name*	Definition	Field
WUnderground_TcpText		Epoch

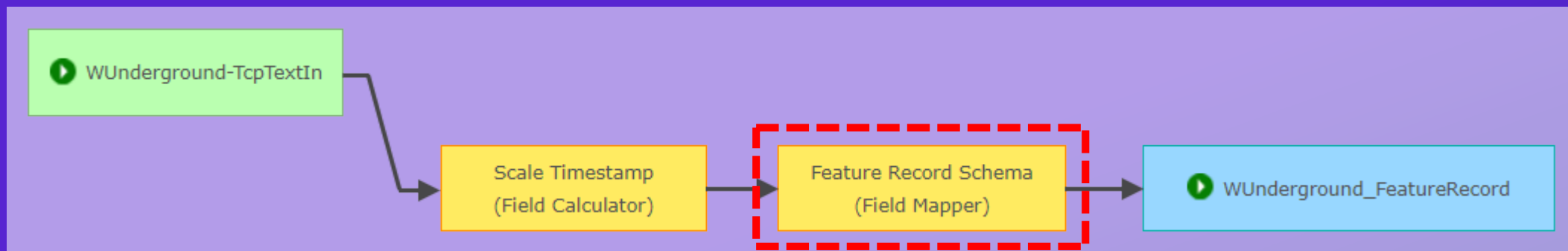
Ok Cancel Help

# Field Mapper

- Use a **Field Mapper** when you need to:
  - Change the schema or structure of an event record
  - Translate from one GeoEvent Definition to another
  - Specify how attribute values map from an inbound to an outbound event record



# Field Mapper



## Processor Properties

Name:\*

Feature Record Schema

Processor:

Field Mapper

Source GeoEvent Definition\*:

WUnderground\_TcpText

Target GeoEvent Definition\*:

WUnderground\_StationReport

### Source Fields

StationID

Epoch

DateTime\_String

DewPoint

### Target Fields

station\_id *String*

reported\_dt *Date*

local\_datetime *String*

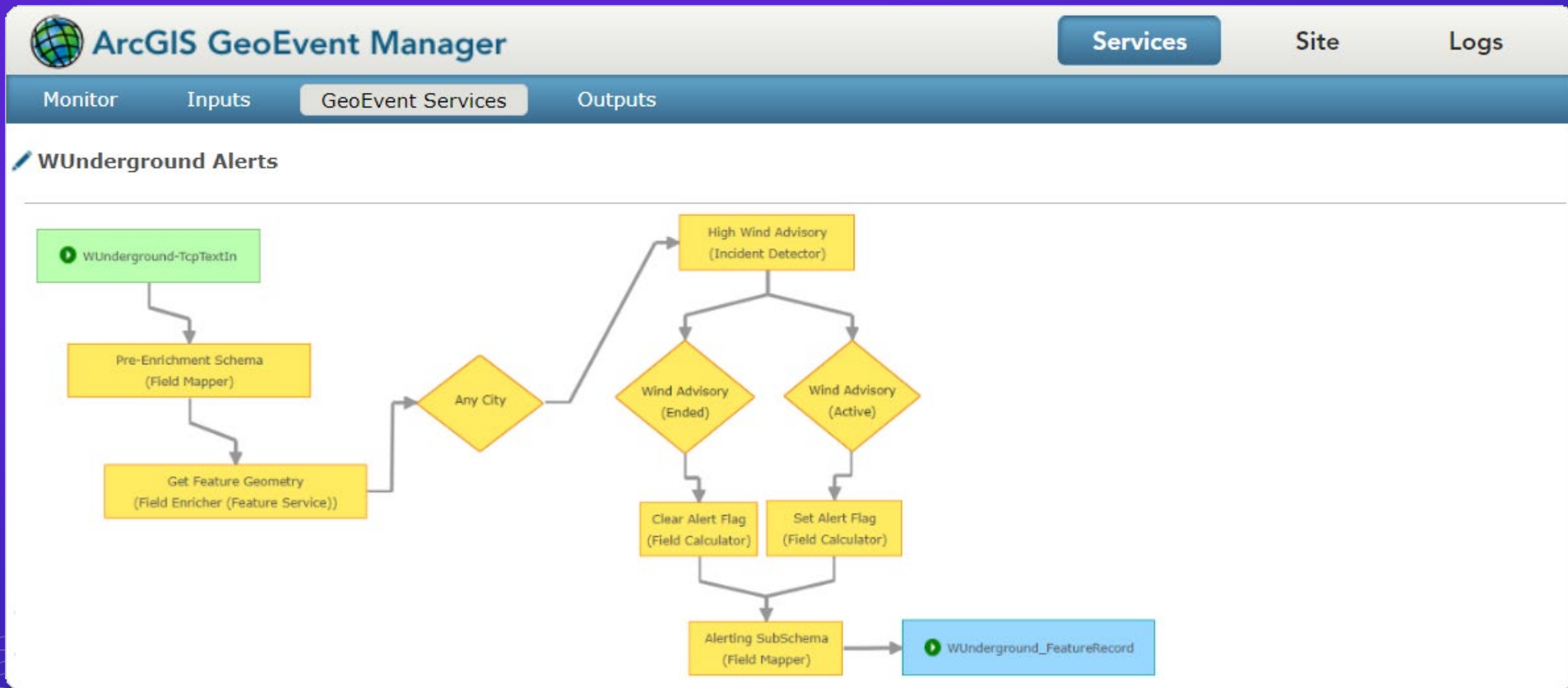
dewpoint *Integer*



# Identifying Conditions with Real-Time Data

*Demo 2: Process event records looking for patterns of interest or alerting thresholds*

- Key Processors: Field Mapper, Field Enricher, Incident Detector

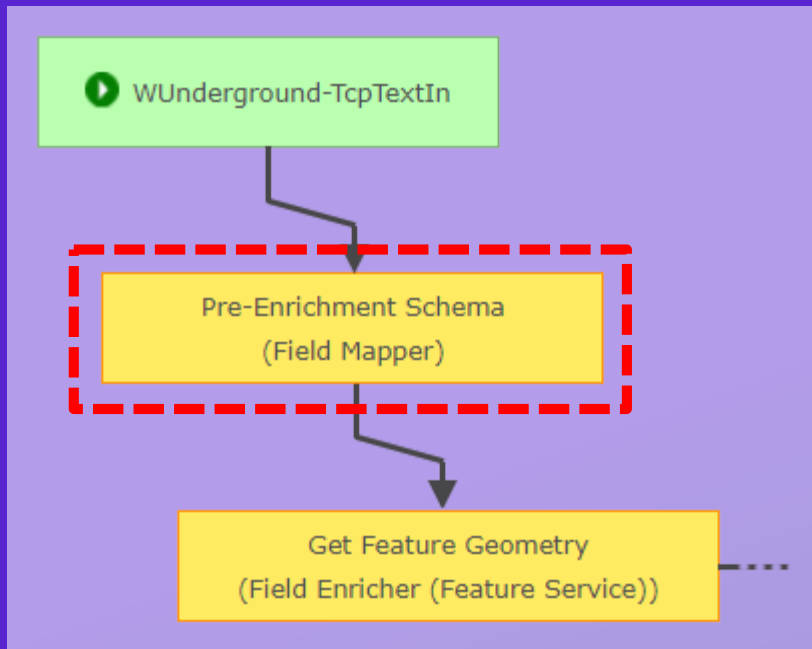


## Field Mapper (Revisited)

- Should field mapping be done at the beginning or at the end of an event processing workflow?
  - You can use a **Field Mapper** to “pre-map” an event schema so other processors (e.g. Field Enricher or Field Calculator) can write values into existing fields
  - Often you will want to use a **Field Mapper** to guarantee an event record’s schema matches the schema expected by an ArcGIS feature service
  - You can also use a **Field Mapper** to simplify an event record’s schema, removing attribute fields you do not want updated in a feature record



# Field Mapper



- Prior to enriching an event record, pre-map the event schema so that you can write to “existing” fields
- Leaving a field unmapped places a null value in the unmapped field

Source GeoEvent Definition*:	<input type="text" value="WUnderground_TcpText"/>
Target GeoEvent Definition*:	<input type="text" value="WUnderground_PreEnrichment"/>
<b>Source Fields</b>	<b>Target Fields</b>
<input type="text" value="StationID"/>	StationID <i>String</i>
<input type="text" value="Epoch"/>	Epoch <i>Date</i>
<input type="text" value="WindGust"/>	WindGust <i>Integer</i>
<input type="text" value=""/>	geometry <i>Geometry</i>

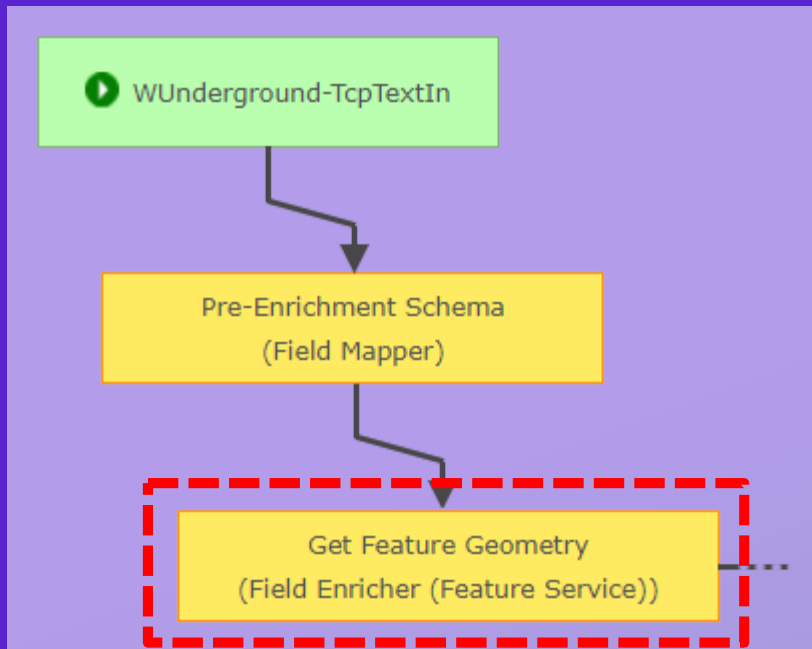
# Field Enricher

- Use a **Field Enricher** when you need to:
  - Enrich an event record with new attribute or geometry from a secondary source
  - An attribute join is used to retrieve values from a feature service or system file

The processor retrieves the specified data values and then enriches an event record by either appending new fields to the record or writing the data to existing fields



# Field Enricher



**Processor Properties**

Name\*: Get Feature Geometry

Processor: Field Enricher (Feature Service)

ArcGIS Server Connection\*: Default [Register ArcGIS Server](#)

Folder: Root

Service\*: WUnderground\_FeatureRecord (FeatureServer)

Layer\*: WUnderground\_FeatureRecord (0)

Feature Layer Join Field\*: station\_id

Target Fields\*: Existing Fields

Enrichment Fields: geometry [Select Fields](#)

Definition	Field
*	StationID

Cache Refresh Time Interval (minutes): 1

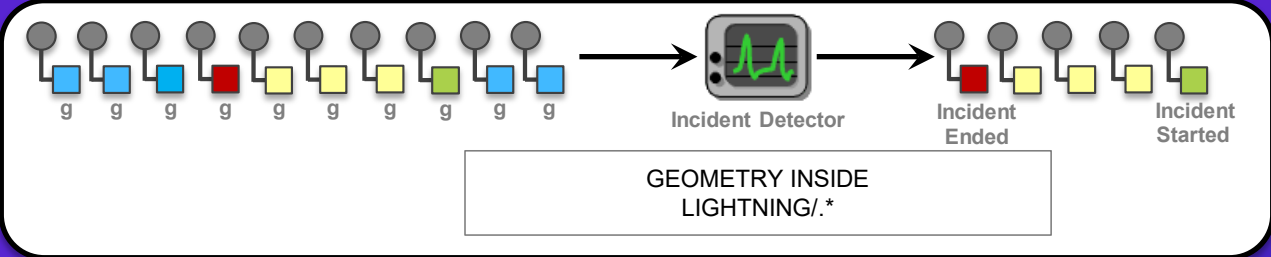
Maximum Number of Feature Records: 1000



# Incident Detector

- Use an **Incident Detector** when you need to:
  - Detect that a condition has occurred and monitor its duration
- Filter expressions are used to specify opening and closing conditions
- This processor maintains state for the duration of the incident
  - Incidents are created when an event is received which satisfies an opening condition
  - Incidents are updated only when new event records are received by the processor
  - Incidents are closed when an event is received which satisfies a closing condition
    - Incidents may close (expire) when no further events are received for the *TRACK\_ID* associated with the incident

# Incident Detector

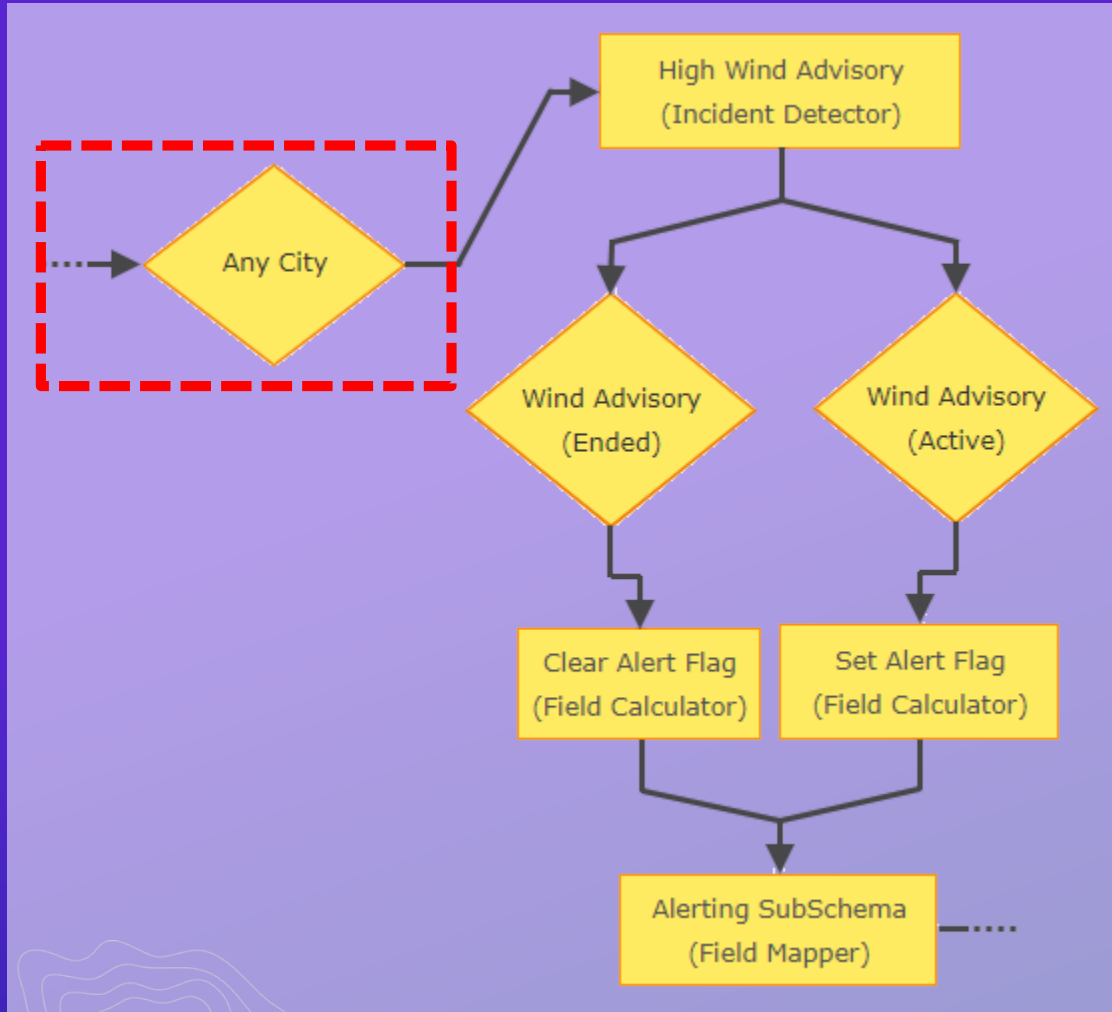


TrackID	J7890
Date	1405176935553
BatteryLevel	Medium
Distance	0.01
DurationMin	1.03
SpeedMPH	0.62
Geometry	-117.1....., 36.0.....



id	c982db54-...-3bbb61211eb6
name	Lightning Hazard
type	Cumulative
status	Ended
alertType	Warning
openCondition	INSIDE(LIGHTNING/*)
closeCondition	
description	Ended at 7/12/14 10:54 AM and lasted for 40 seconds
timestamp	1405176905553
definitionName	Incident
definitionOwner	com.esri.ges.processor/Incident Detector/10.7.0
trackId	J7890
geometry	-117.123..., 36.064...
duration	40000
dismissed	False
assignedTo	
note	

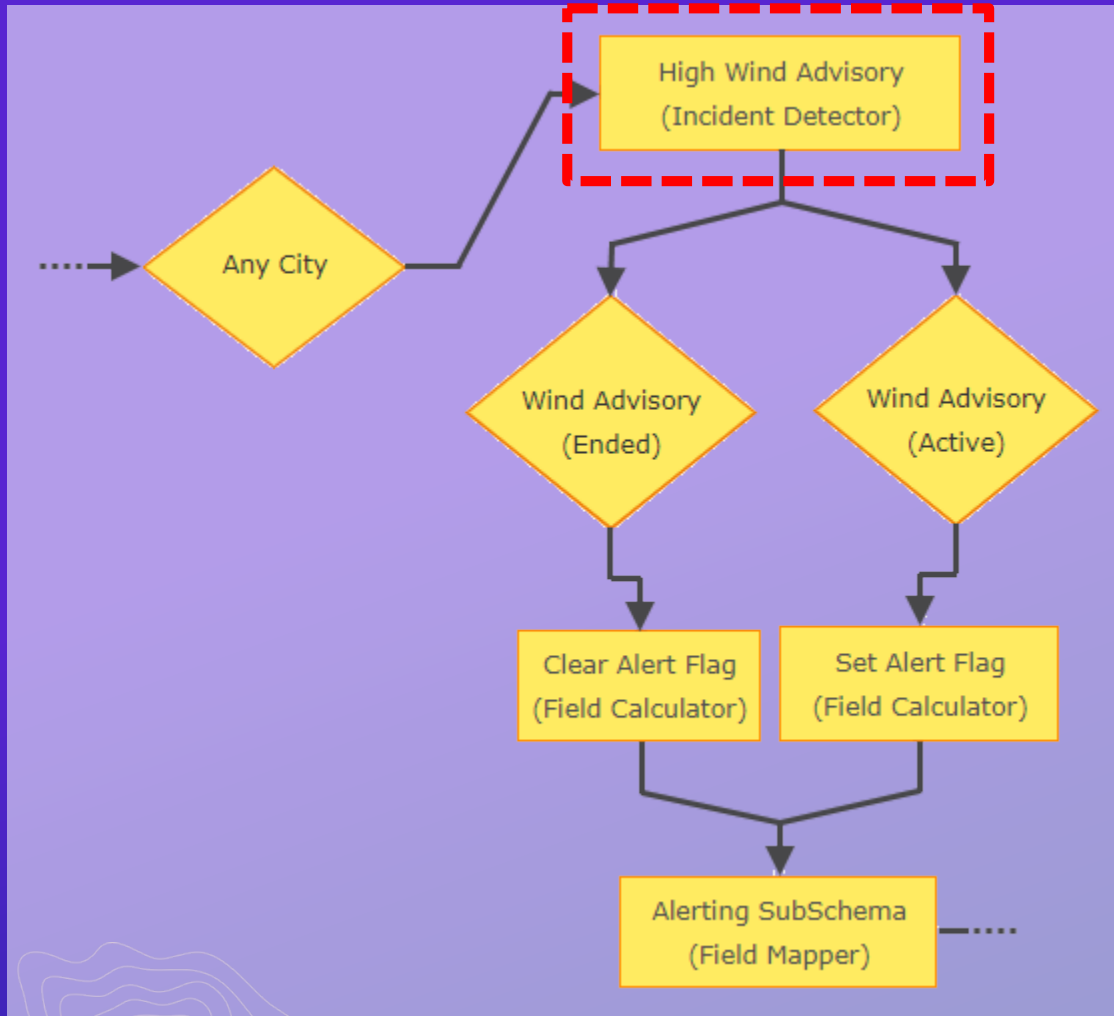
# Incident Detector



- Enriching each sensor's event record with a geometry would enable a spatial filter to focus downstream logic to a specific area of interest
- For this demo I chose to configure the filter to check only that a specific field exists, allowing all event records to pass through the filter

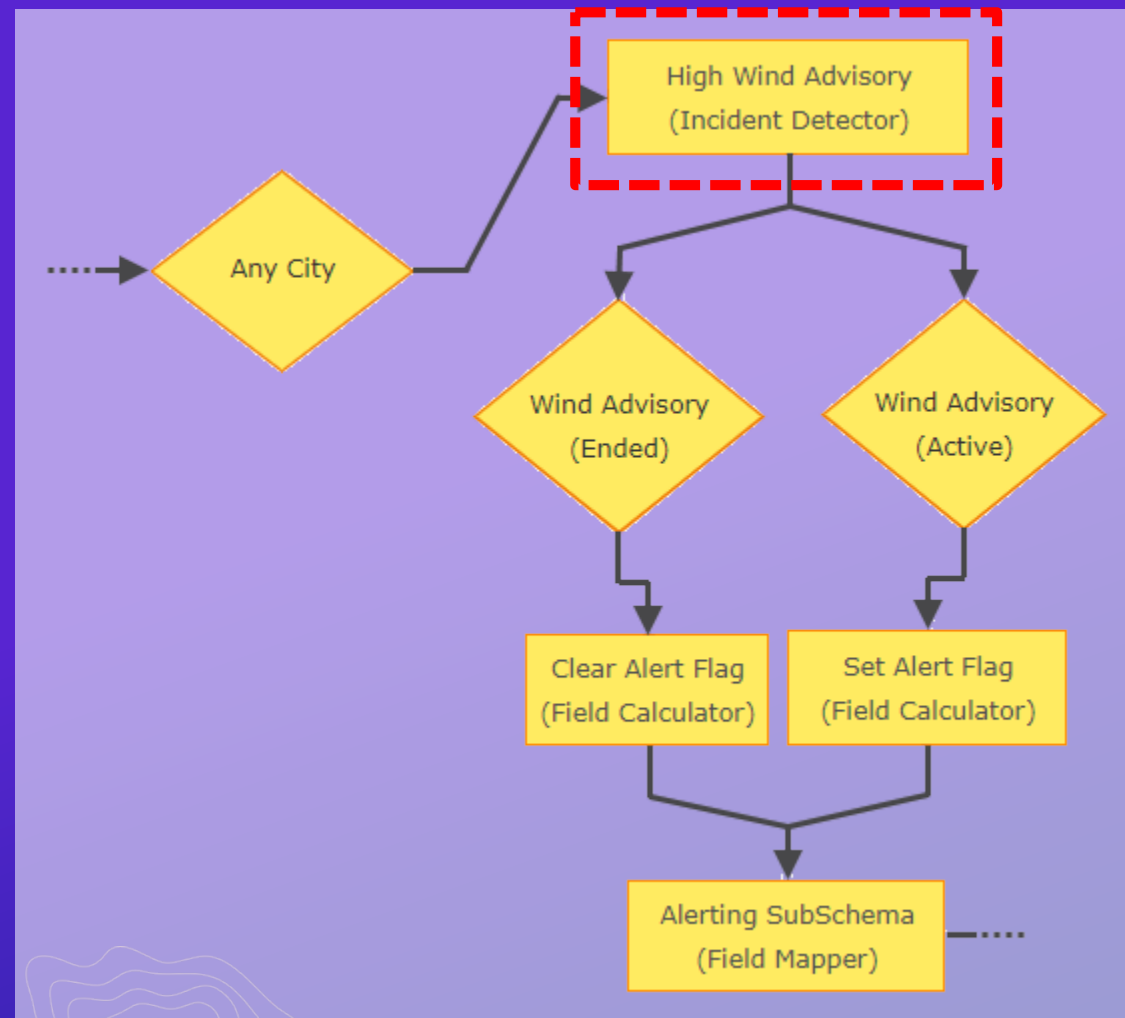
The "Filter Properties" dialog box shows the configuration for the "Any City" filter. The "Name" field is set to "Any City". The filter condition is defined as "StationID EXISTS". Below the condition, there is a green plus icon and a blue button labeled "+()". The "Ok" and "Cancel" buttons are located in the top right corner.

# Incident Detector



- An Incident Detector looks at the *TRACK\_ID* of a received event record
- If an incident exists whose state is 'Started' or 'Ongoing' and the processor's opening condition is satisfied the incident is updated

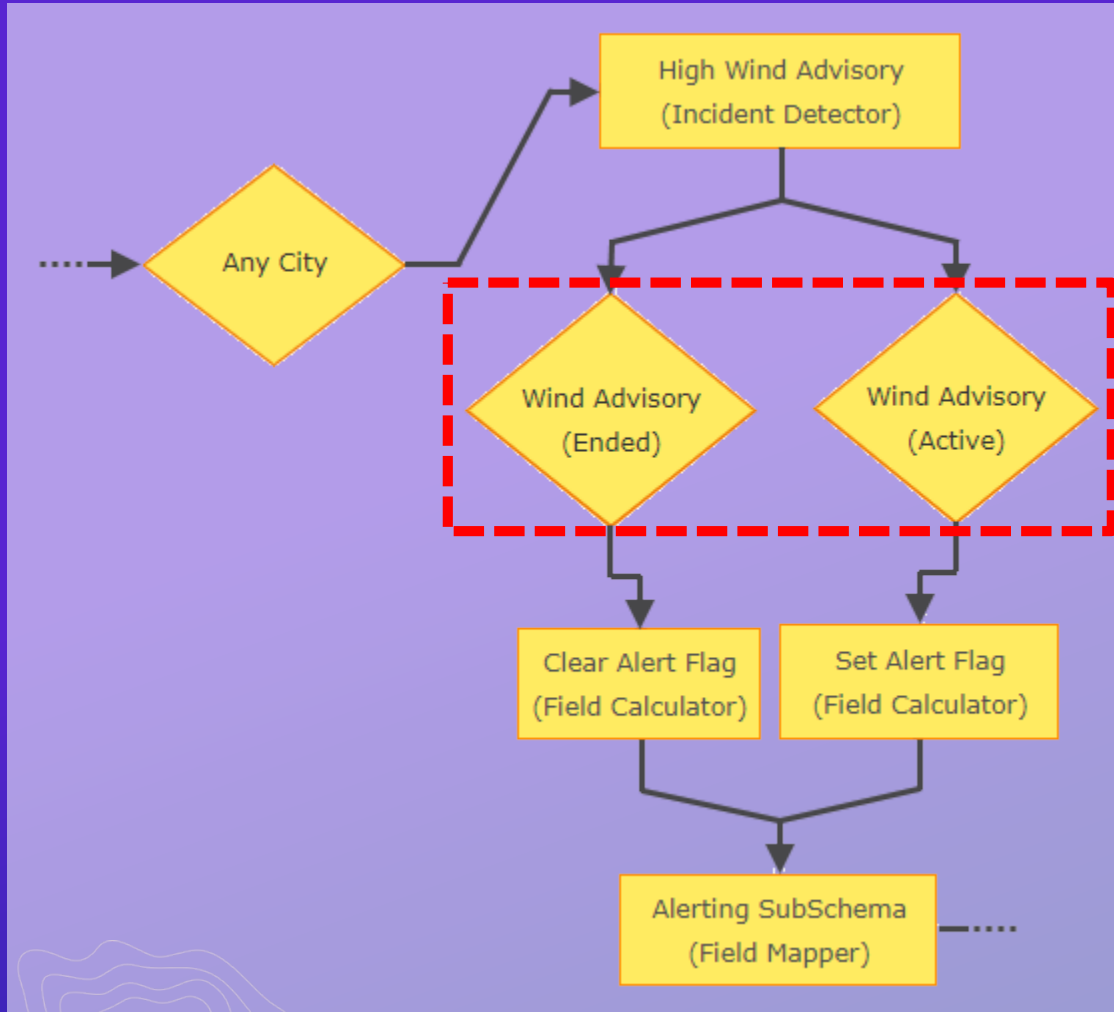
# Incident Detector



Processor Properties	
Name*:	High Wind Advisory
Processor:	Incident Detector
Incident Name*:	High Wind Advisory
Opening Condition*:	WindGust >= 30
	Opening Condition
Closing Condition:	WindGust < 30
	Closing Condition
Severity*:	Notification
Incident Type*:	Cumulative
Geometry Type*:	Point
Expiry Time (seconds)*:	0



# Incident Detector



- A pair of filters look specifically for *incident* event records whose status is either 'Started' 'Ongoing' ... or 'Ended'

Filter Properties

Name:\* Wind Advisory (Active) Ok Cancel

OR ☒ ☐

☒ status = Started X

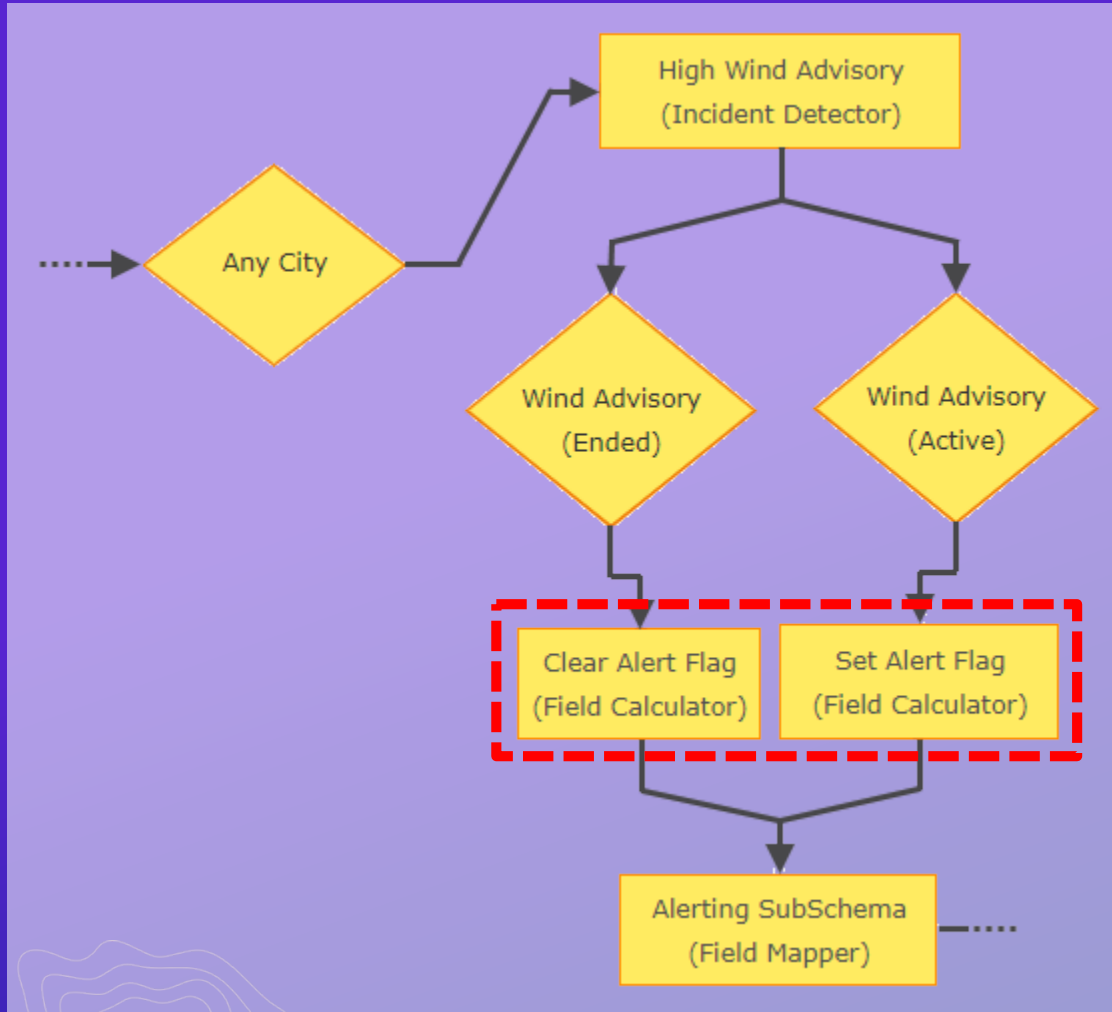
☒ status = Ongoing X

Filter Properties

Name:\* Wind Advisory (Ended)

☐ status = Ended X

# Incident Detector



- A pair of Field Calculators write coded values into an attribute field enabling the web map to symbolize detected incidents with flags

Processor Properties

Name\*: Set Alert Flag

Processor: Field Calculator

Expression\*: toString(1)

Target Field\*: Existing Field

Definition	Field
Existing Field Name*: incident	note

Processor Properties

Name\*: Clear Alert Flag

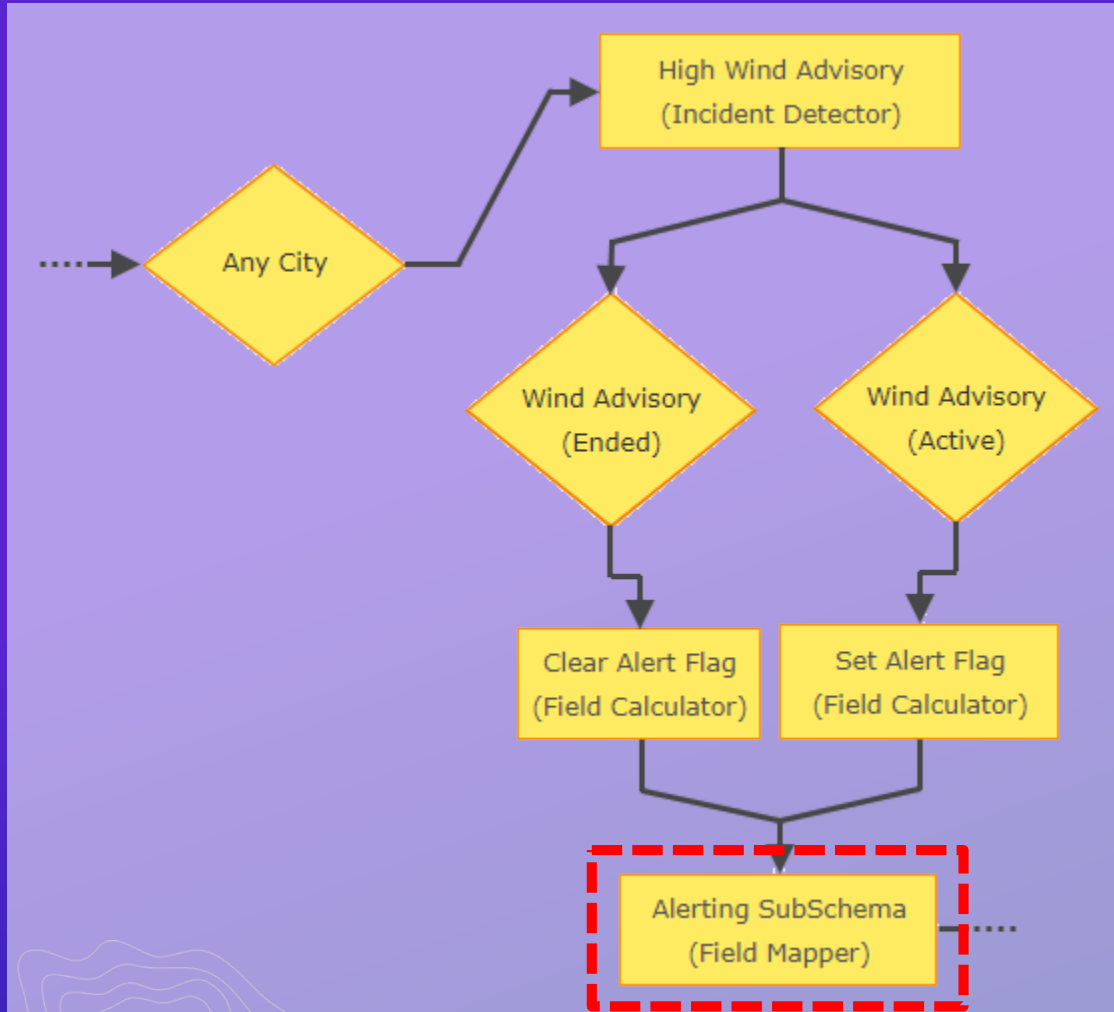
Processor: Field Calculator

Expression\*: toString(0)

Target Field\*: Existing Field

Definition	Field
Existing Field Name*: incident	note

# Incident Detector



- A final Field Mapper prepares an event record whose schema matches a subset expected by the feature service
- Attribute values whose fields are not included will not be updated

Processor Properties

Name:\* Alerting SubSchema

Processor: Field Mapper

Source GeoEvent Definition\*: incident

Target GeoEvent Definition\*: WUnderground\_AlertRecord

Source Fields	Target Fields
trackId	station_id <i>String</i>
note	display_code <i>Integer</i>
duration	alert_duration <i>Long</i>

# 3

## Use Case 2: Situational Awareness & Response

- *Monitor drones and operators*
- *Situational alerting*
- *Real-time response*





[Details](#) [Add ▾](#) [Basemap](#) [Analysis](#)

[Save ▾](#) [Share](#) [Print ▾](#) [Directions](#) [Measure](#) [Bookmarks](#)  [Q](#)

Legend

**Outside Line-Of-Sight Warning**

- Current observations
- Previous observations

**Altitude Exceeded Warning**

- Current observations
- Previous observations

**No Fly Zone Warning**

- Current observations
- Previous observations

**Operator**

- 

**Missing Drone**

- 

**Drone**

- Current observations
- Previous observations

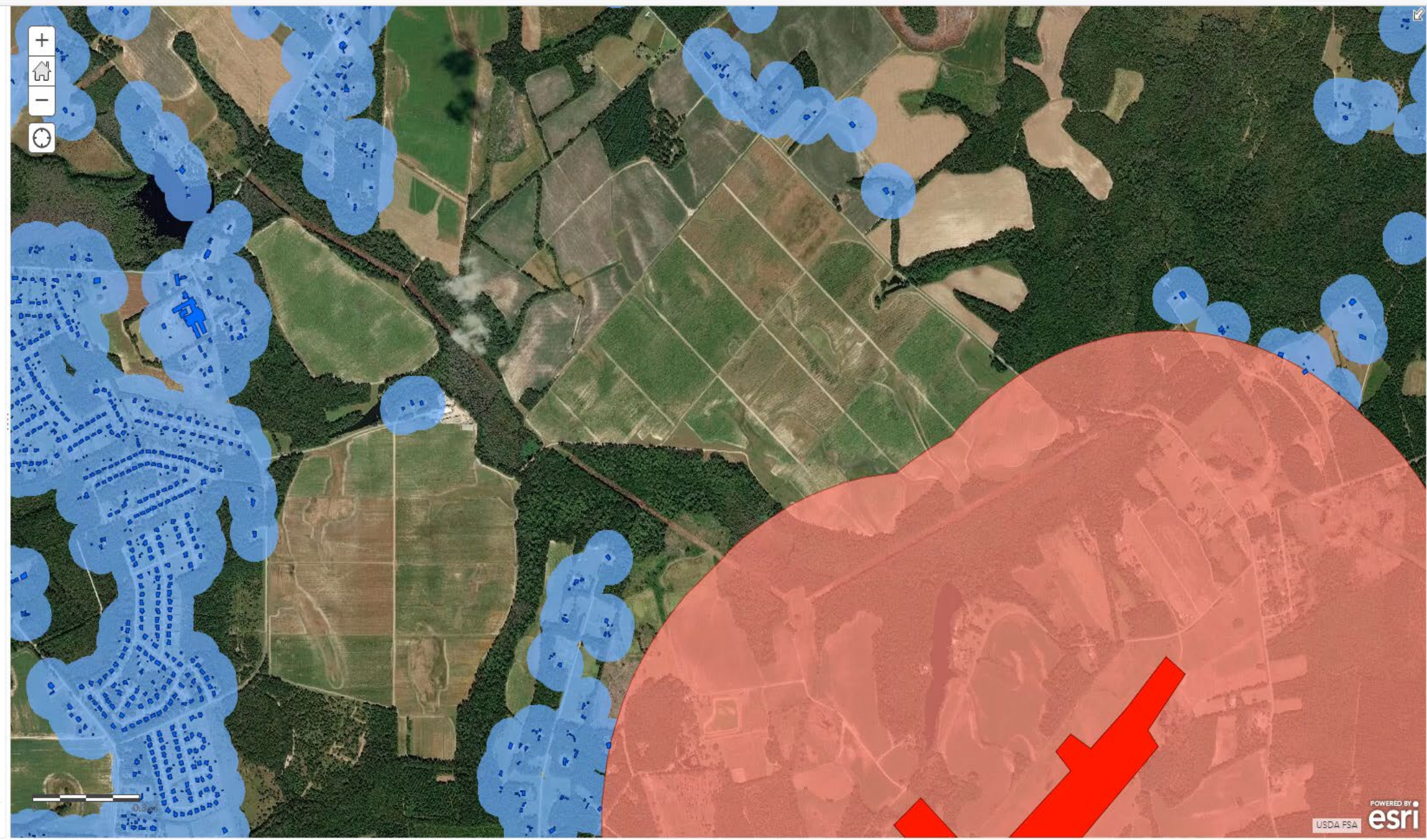
**No Fly Zone - Airspace - 1 Mile**

- 

**No Fly Zone - Building - 400 Feet**

- 

[Trust Center](#) [Contact Esri](#) [Report Abuse](#)  
[Contact Us](#)

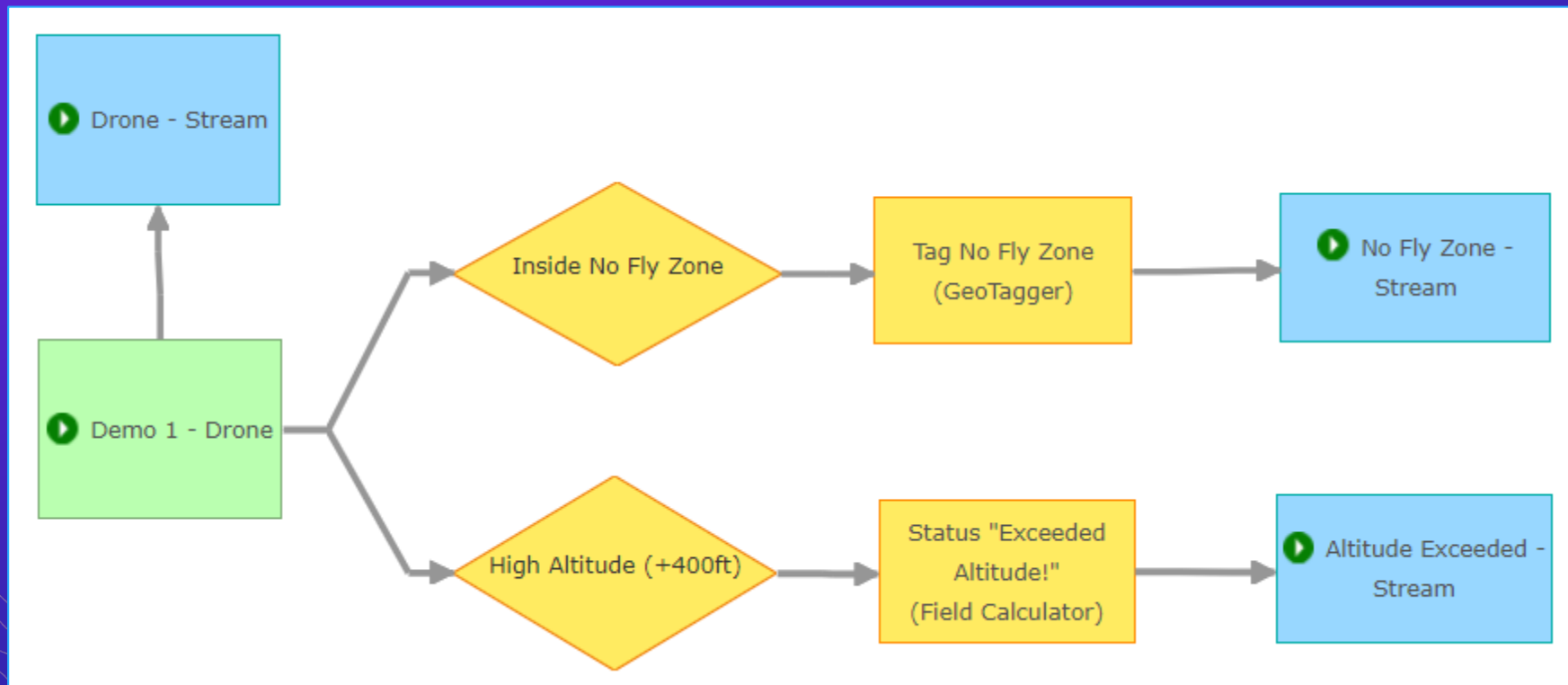




# Situational Awareness & Response with Real-Time Data

## *Demo 1: Collect drone positions and perform spatial and attribute detection*

- Input: Drone position and ancillary attribute data (altitude, speed)
- Output: Stream service to visualize spatial and attribute conditions
- Key Processor: GeoTagger



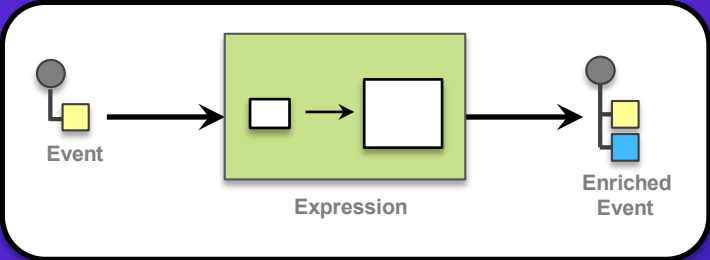
# GeoTagger

- Use a **GeoTagger** when you need to:
  - Enrich an event record with the name of a geofence with which the event record's geometry shares a spatial relationship
- The processor uses a spatial expression to identify related geometries.
- The unique identifier (or “name”) of a related geofence is appended to the event record – essentially performing a spatial join with a geofence.



# GeoTagger

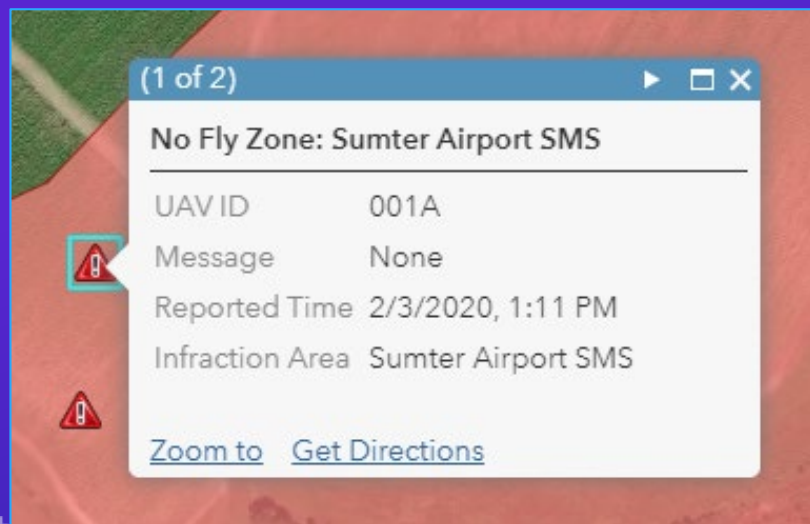
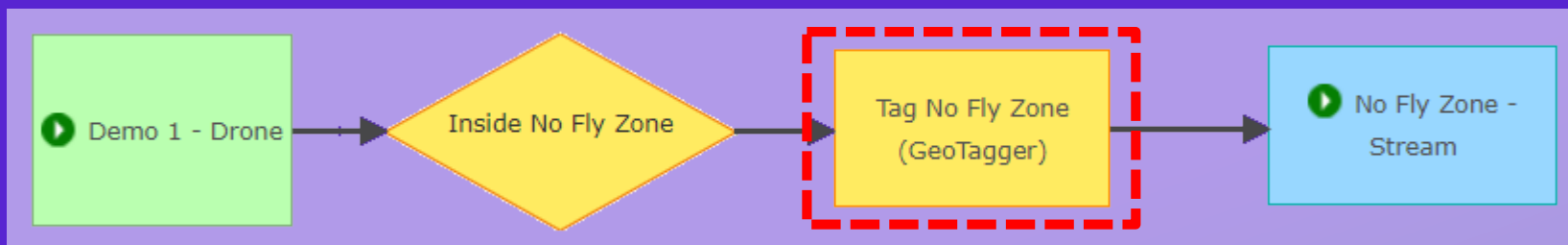
TrackID	001A
Date	1405176845553
Status	In-Flight
Latitude	36.064
Longitude	-117.123
Altitude	301.0
Speed	12.4
Heading	90.23
Geometry	-117.123..., 36.064...
Category	UAV



TrackID	001A
Date	1405176845553
Status	In-Flight
Latitude	36.064
Longitude	-117.123
Altitude	301.0
Speed	12.4
Heading	90.23
Geometry	-117.123..., 36.064...
Category	UAV
IsInside	Sumter Airport SMS



# GeoTagger






### Processor Properties

Name:	Tag No Fly Zone
Processor:	GeoTagger
Geofence(s):	No Fly Zone/.*
Spatial Operator:	Inside Any
Geometry Field:	GEOMETRY
Target Field:	New Field
GeoTag Field Name:	Infraction_Area
GeoTag Format:	Delimited Value
New GeoEvent Definition Name:	Drone_GeoTagged
Include Geofence Category in GeoTag:	<input type="radio"/> Yes <input checked="" type="radio"/> No





 Details  Add ▾  Basemap  Analysis

 Save ▾  Share  Print ▾  Directions  Measure  Bookmarks





Legend



**Outside Line-Of-Sight Warning**

-  Current observations
-  Previous observations


**Altitude Exceeded Warning**

-  Current observations
-  Previous observations


**No Fly Zone Warning**

-  Current observations
-  Previous observations



**Operator**

- 


**Missing Drone**

- 


**Drone**

-  Current observations
-  Previous observations

**No Fly Zone - Airspace - 1 Mile**

- 

**No Fly Zone - Building - 400 Feet**

- 

[Trust Center](#) [Contact Esri](#) [Report Abuse](#)  
[Contact Us](#)

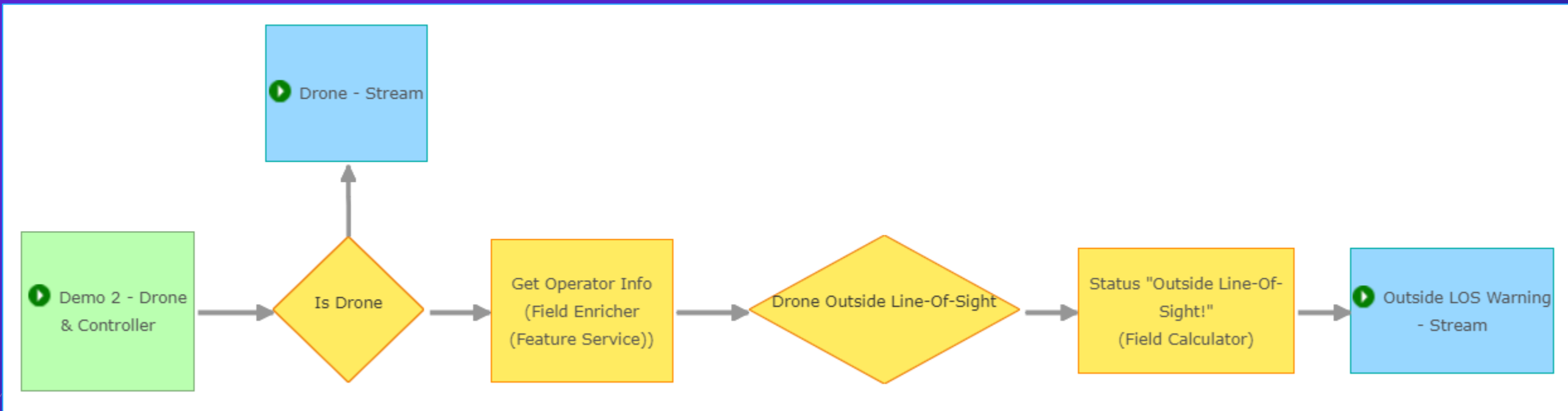




# Situational Awareness & Response with Real-Time Data

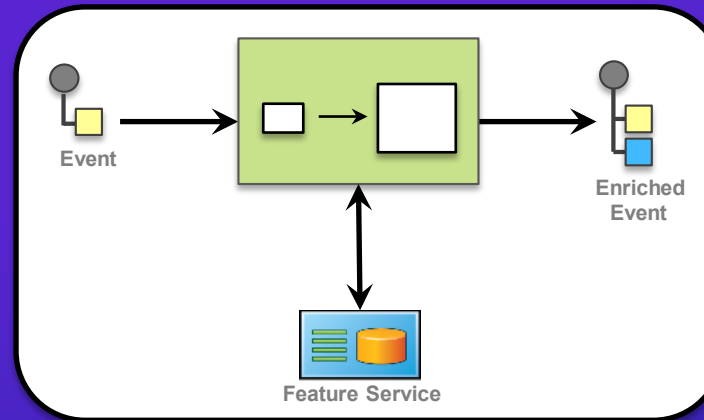
*Demo 2: Enrich event records with operator information; provide context for spatial conditions*

- Input: Drone position and ancillary attribute data (altitude, speed)
- Output: Stream service to visualize drone's position and line-of-sight warning
- Key Processor: Field Enricher (Feature Service)
- Key Filter: Spatial Filter – “Drone Outside Line-of-Sight”



# Field Enricher

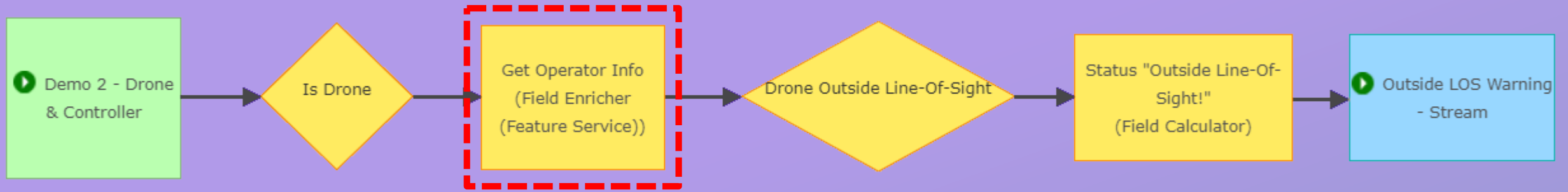
UAV_ID	003C
Date	1405176845553
Status	In-Flight
Altitude	153.0
Speed	14.7
Heading	94.2
geometry	-117.123..., 36.064...



UAV_ID	003C
Date	1405176845553
Status	In-Flight
Altitude	153.0
Speed	14.7
Heading	94.2
geometry	-117.123..., 36.064...
Operator	Jane Doe
Email	janedoe@example.com
Phone	867-5309

Assigned_UAV	Operator	Email	Phone
002B	John Doe	johndoe@example.com	123-4567
003C	Jane Doe	janedoe@example.com	867-5309
...	...	...	...

# Field Enricher



Target Fields\*:

New Fields

Enrichment Fields:

Name,Email,Phone,Task

Select Fields

Field Tags:

New GeoEvent Definition Name:

Drone\_and\_Operator

GeoEvent Join Field\*:

Definition

Field

\*

UAV\_ID

Cache Refresh Time Interval (minutes):

480

Maximum Number of Feature Records:

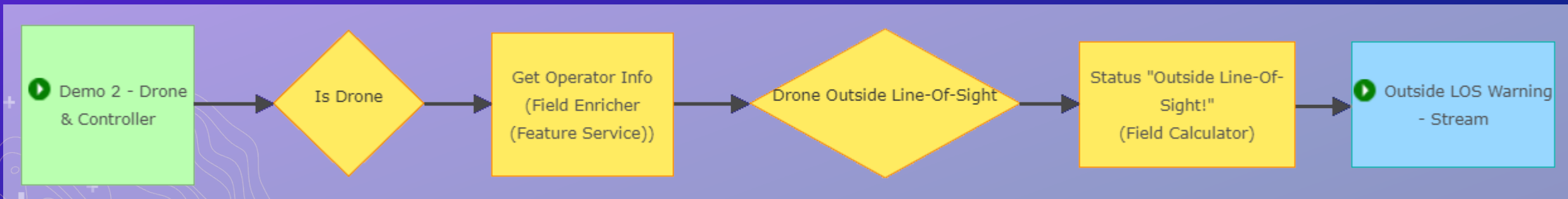
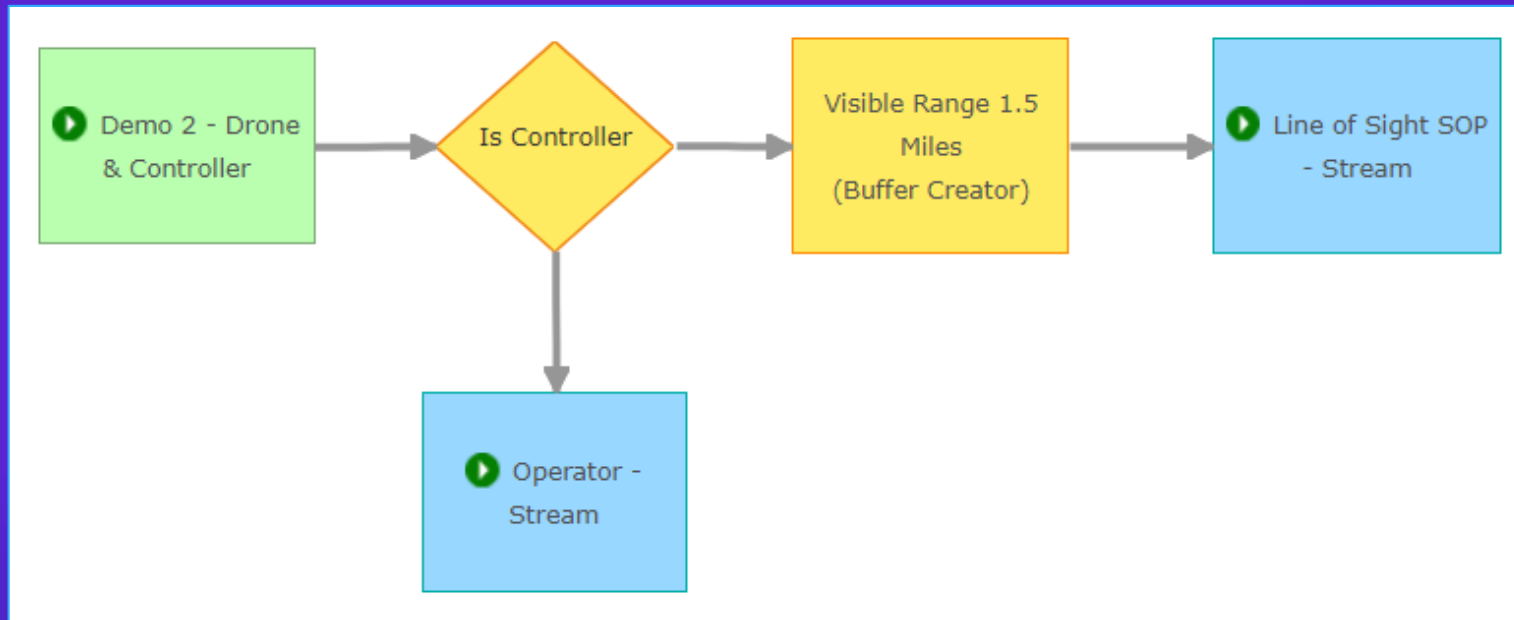
1000



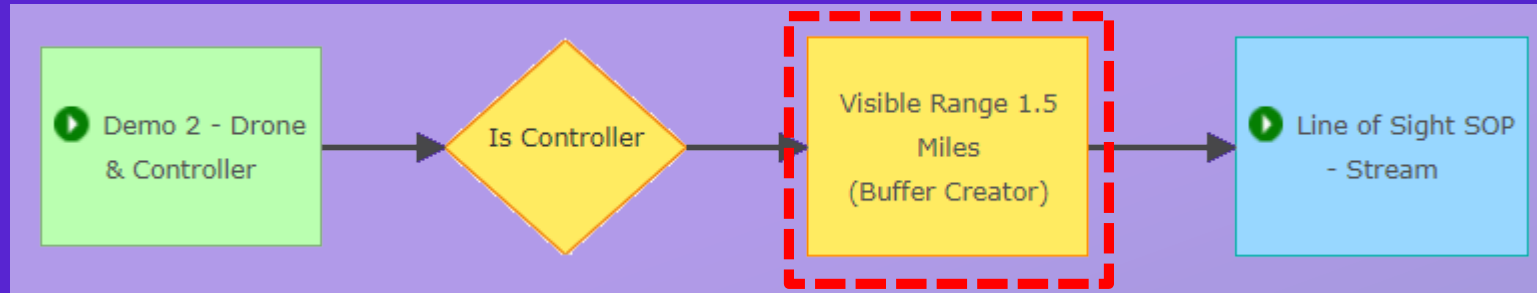
# Situational Awareness & Response with Real-Time Data

## *Demo 2: Secondary GeoEvent Service for line-of-sight geofencing*

- Key Processor: Buffer Creator

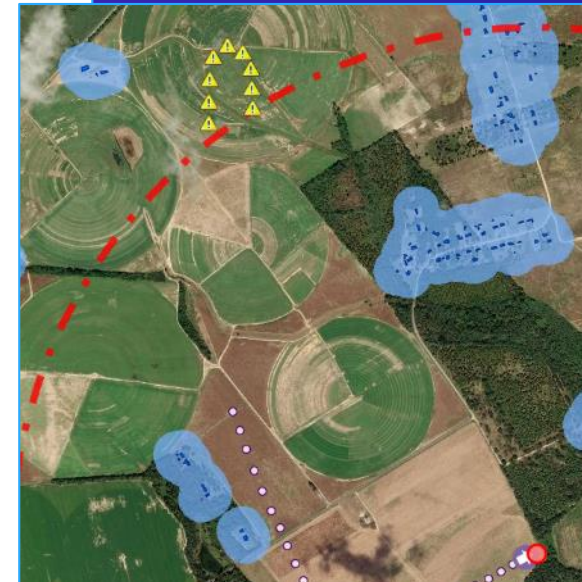


# Buffer Creator

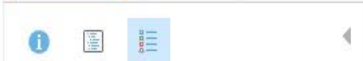


## Processor Properties

Name*:	<input type="text" value="Visible Range 1.5 Miles"/>
Processor:	<input type="text" value="Buffer Creator"/>
Geometry Field*:	<input type="text" value="GEOMETRY"/>
Replace Geometry*:	<input checked="" type="radio"/> Yes <input type="radio"/> No
Buffer Size Units*:	<input type="text" value="Mile"/>
Buffer Size*:	<input type="text" value="1.5"/>
Buffer WKID*:	<input type="text" value="102100"/>











## Legend



### Outside Line-Of-Sight Warning

-  Current observations
-  Previous observations

### Altitude Exceeded Warning

-  Current observations
-  Previous observations

### No Fly Zone Warning

-  Current observations
-  Previous observations



### Operator



### Missing Drone



### Drone

-  Current observations
-  Previous observations

### No Fly Zone - Airspace - 1 Mile



### No Fly Zone - Building - 400 Feet

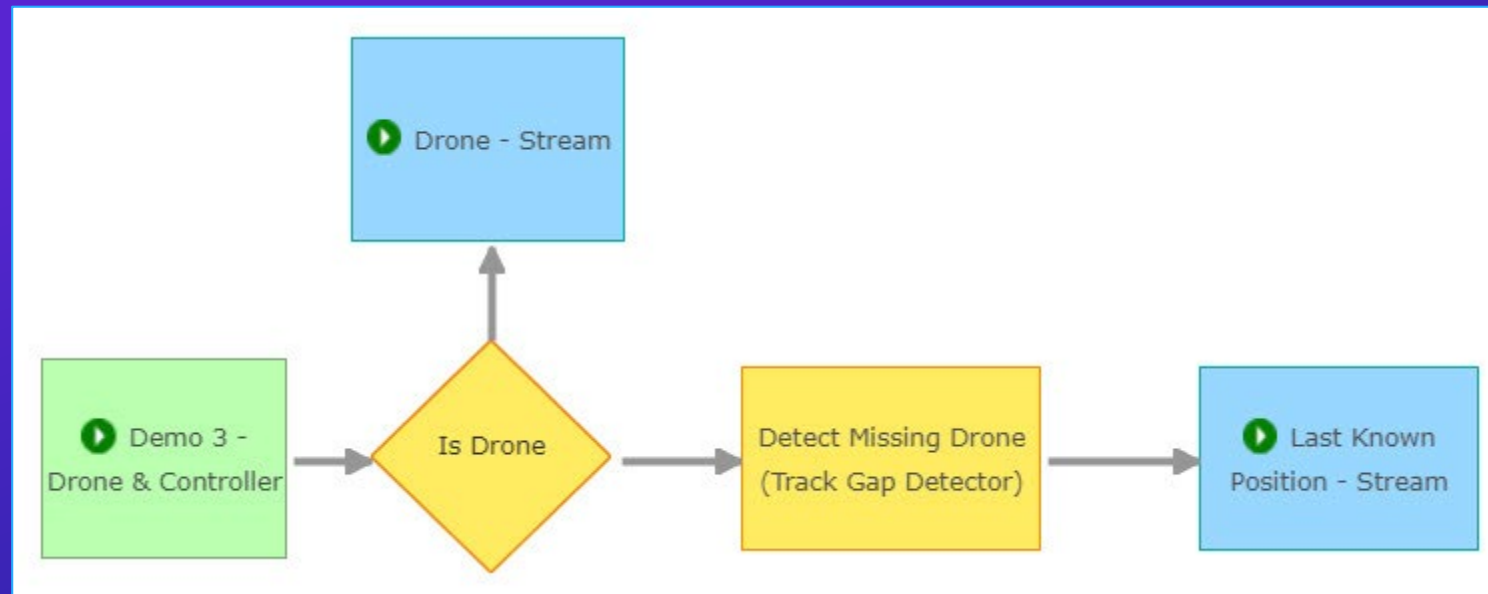




# Situational Awareness & Response with Real-Time Data

## *Demo 3: Detect missing drones and provide alerting*

- Input: Drone position and ancillary attribute data (altitude, speed)
- Output: Stream service to visualize drone's last reported position
- Key Processor: Track Gap Detector

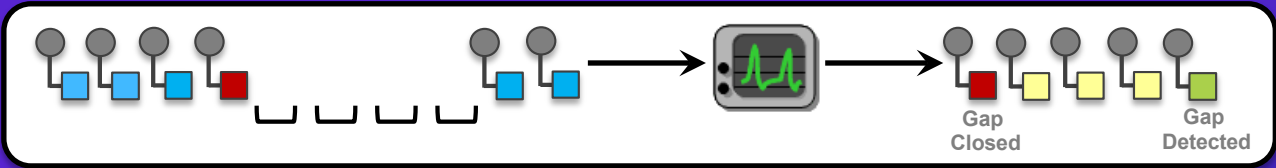


# Track Gap Detector

- Use a **Track Gap Detector** when you need to:
  - Detect the absence of event record reporting
  - Alert or notify someone that expected data was not received



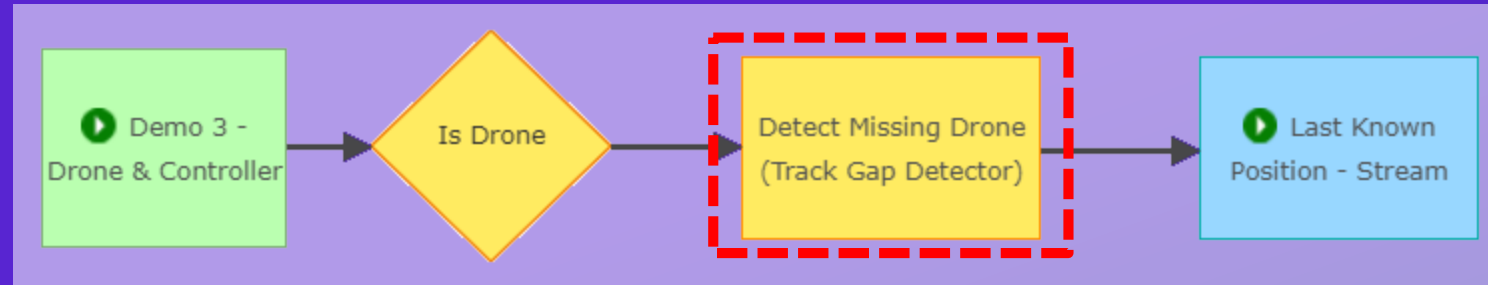
# Track Gap Detector



TrackID	004D
Date	1405176945553
Geometry	-117.123..., 36.064...

trackId	004D
gap	false
lastReceived	1405176915553
geometry	-117.123..., 36.064...

# Track Gap Detector



## Processor Properties

Name\*: Detect Missing Drone

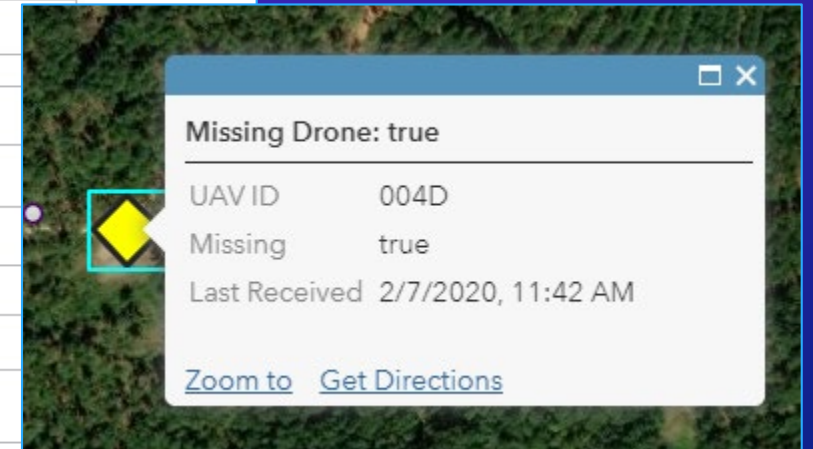
Processor: Track Gap Detector

Gap Notification Mode\*: On Change

Gap Duration (seconds)\*: 5

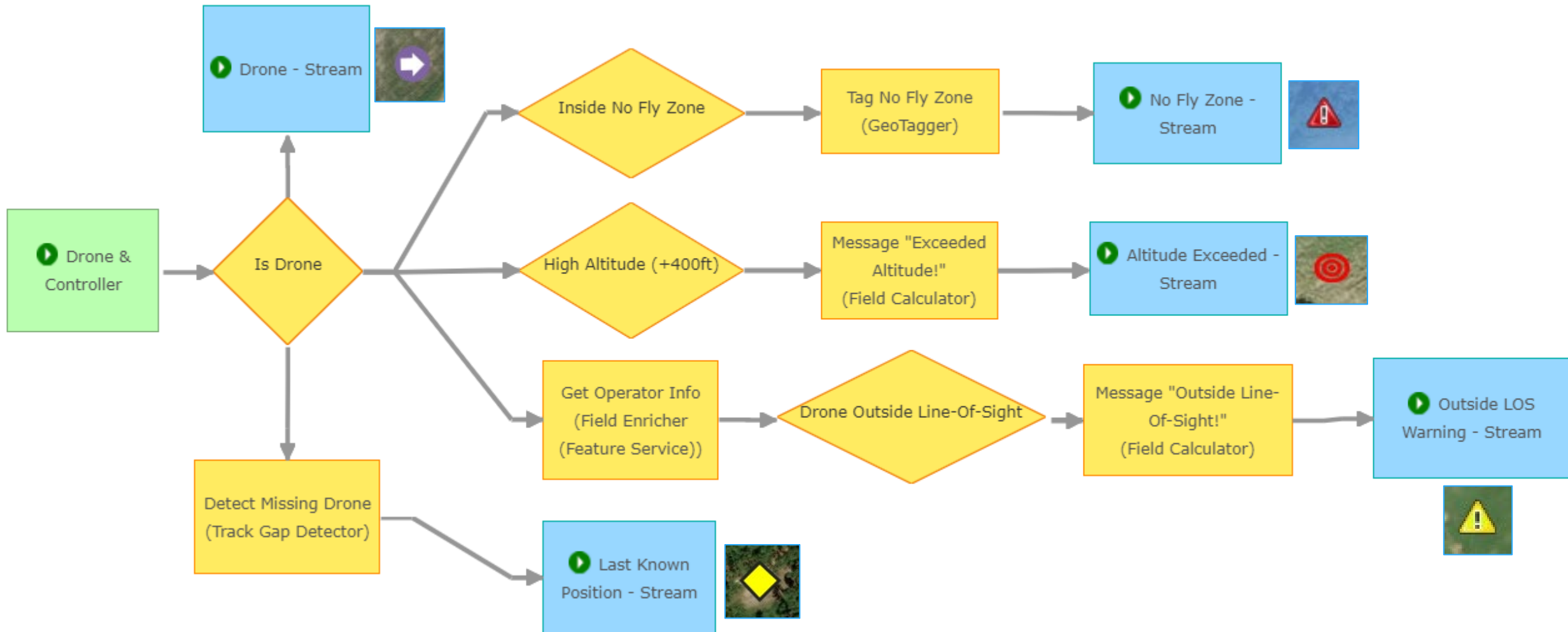
Gap Detection Interval (seconds)\*: 1

Geometry Field\*: Geometry



# Situational Awareness & Response with Real-Time Data

## *GeoEvent Service Design*





4

# Summary & Resources





# Summary

## *GeoEvent Server – Real-Time Analysis for your ArcGIS Enterprise*

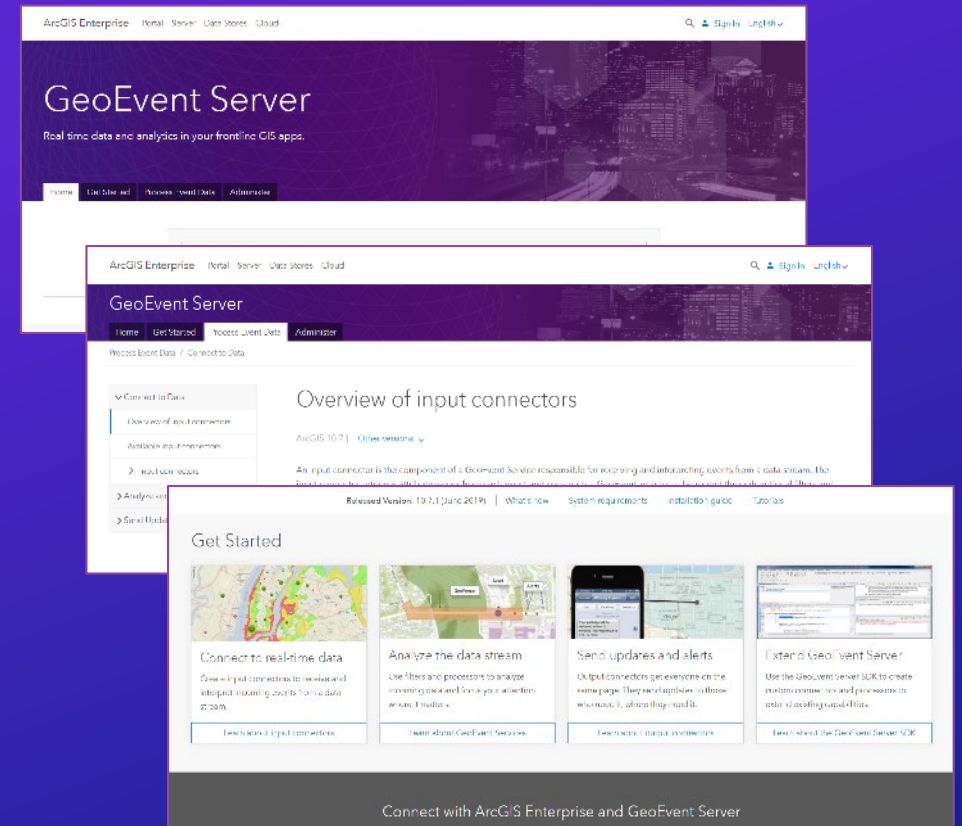
- ArcGIS is a dynamic platform that enables continuous analysis and real-time visualization for better understanding of our world
- The ArcGIS GeoEvent Server allows you to:
  - Know what is happening, as it happens
  - React and make smarter decisions faster
  - Be notified when interesting events occur



# Resources

## *Self-Paced Training and Resources*

- ArcGIS GeoEvent Server resources
  - <http://enterprise.arcgis.com/en/geoevent>
    - Updated documentation
    - Quick Start Guide (PDF)
    - Installation Guides
    - System Requirements
    - Tutorials
- Blogs and discussion forum
  - <http://links.esri.com/geoevent-forum>
- Video recordings of technical workshops
  - <http://www.esri.com/videos>





# Questions?

2020 ESRI DEVELOPER SUMMIT | Palm Springs, CA



**RJ Sunderman**  
ArcGIS GeoEvent Server  
Product Engineer  
[rsunderman@esri.com](mailto:rsunderman@esri.com)



**Gregory Christakos**  
ArcGIS GeoEvent Server  
Product Engineer  
[gchristakos@esri.com](mailto:gchristakos@esri.com)

