



2022 SAR ANALYTICS  
SYMPOSIUM

# VHR SAR Imagery for a fast-changing world!

Mack Koepke

*Capella Space*

*Mack.Koepke@capellaspace.com*

# Capella Space is Changing Access to Earth Information



**Frequent Revisit +  
Temporal Diversity**



**Any time,  
Any Weather**



**Very High-  
Resolution Imaging**

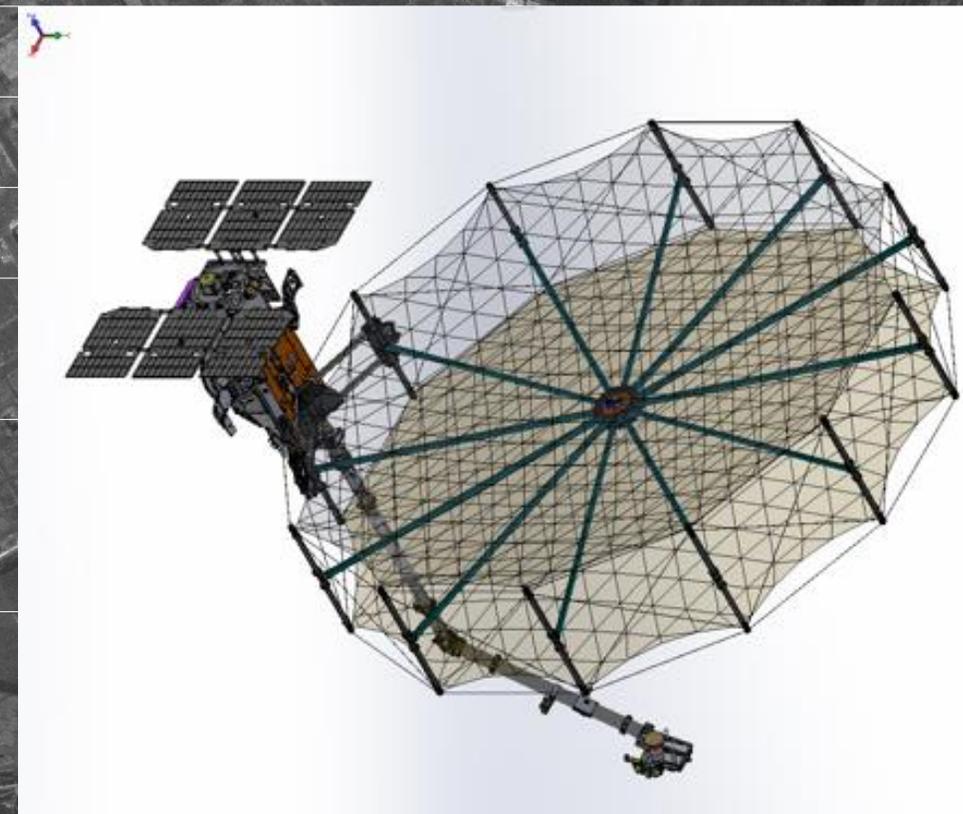


**Fastest From  
Order to Delivery**

• Constellation of Small Radar Satellites – Rapid Path to deployment

# Capella SAR Imaging- Current Generation

<b>Central Frequency</b>	X-Band
<b>Polarization</b>	Single-Pol HH or VV
<b>Imaging Bandwidth</b>	Up to 500 MHz
<b>Acquisition Direction</b>	Ascending+Descending Orbit Direction Left+Right Look Direction
<b>Imaging Modes</b>	Spotlight Sliding Spotlight Stripmap
<b>SAR Imagery Products</b>	Spot (spotlight imaging mode) Site (sliding spotlight imaging mode) Strip (stripmap imaging mode)





**On-orbit view of Antenna**



# SAR Imagery Products

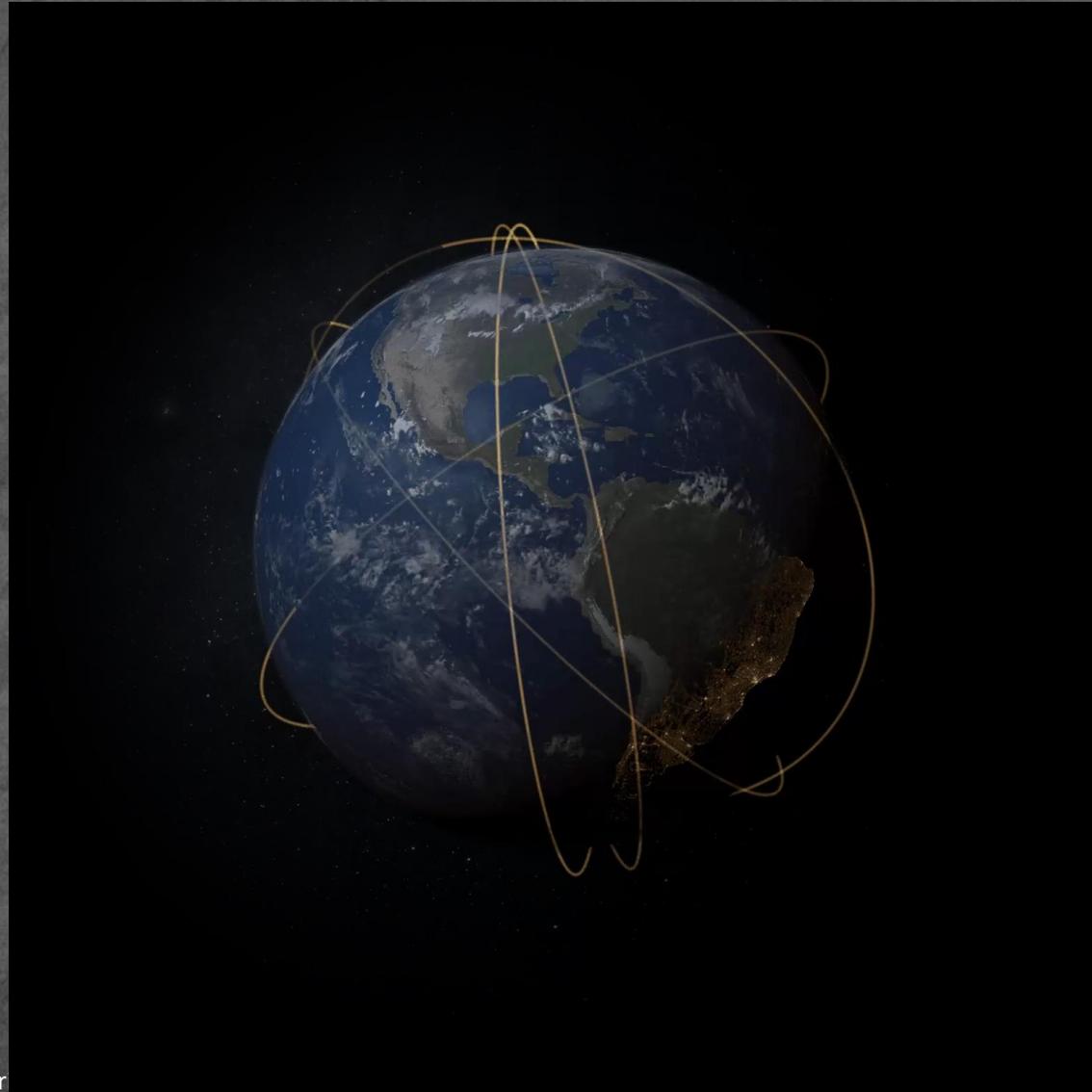
# SAR Imagery Product Scenes



Priority Ordering Tiers	1-Day, 3-Day, 7-Day		
Orbital Diversity	45° INC, 53° INC, 97° SSO		
Collection Geometries	Ascending, Descending, Left, Right		
Imaging Modes	Spotlight	Sliding Spotlight	Stripmap
Scene Length(s)	5 km	10 km – 50 km	20 km – 200 km
Scene Width(s)	5 km	5 km	5 km – 10 km
Look Angle Range	5° – 40°	5° – 45°	5° – 45°
Azimuth Resolution	0.5 m	1.0 m	1.2 m
Slant Range Resolution	0.3 m	0.5 m	0.75 m
Ground Range Resolution	0.5 m – 3.1 m	0.7 m – 5.0 m	1.1 m – 11.5 m
Number of Looks	1 – 9	1 – 5	1



# Capella's Orbits provide unmatched Temporal & Orbital Diversity



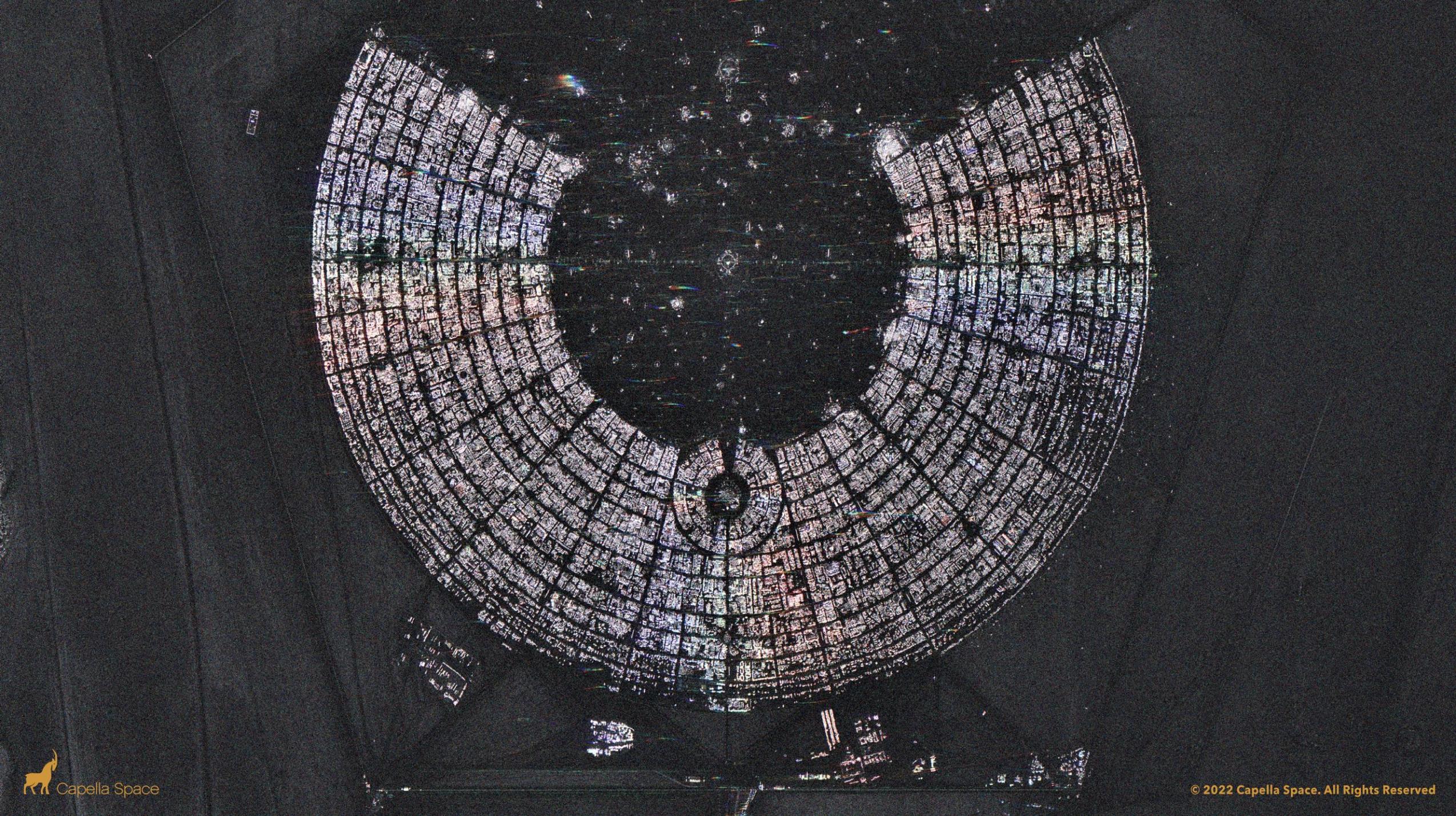
# Unmatched Temporal & Orbital Diversity

Monitor all day, throughout the day, every single day. A capability unique to the Capella SAR constellation

## Ukraine Conflict AOR, 2022

- 2022-03-05T23:34:38.292Z
- 2022-03-05T19:31:30.090Z
- 2022-03-05T19:26:49.361Z
- 2022-03-05T19:24:54.499Z
- 2022-03-05T12:50:21.650Z
- 2022-03-05T12:23:08.159Z
- 2022-03-05T11:52:37.341Z
- 2022-03-05T10:43:42.998Z
- 2022-03-05T10:12:19.350Z
- 2022-03-05T08:41:25.168Z
- 2022-03-05T08:40:20.321Z
- 2022-03-05T07:13:33.203Z
- 2022-03-05T07:11:21.169Z
- 2022-03-05T06:52:29.894Z







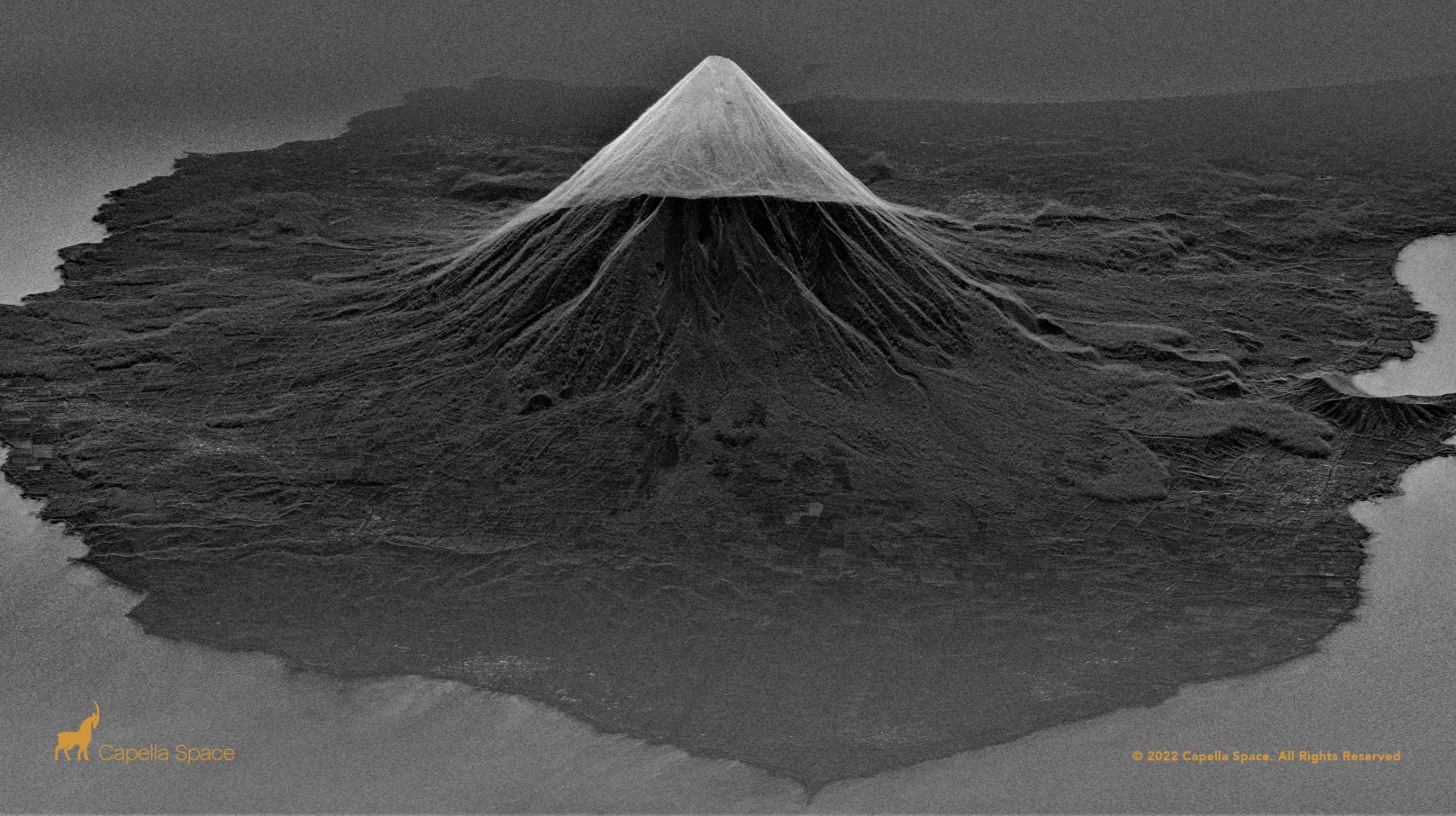
**Port of Odesa**

46.5036 Lat | 30.7444 Long

04/13/22 19:18:00 UTC











# Building

# Automation:

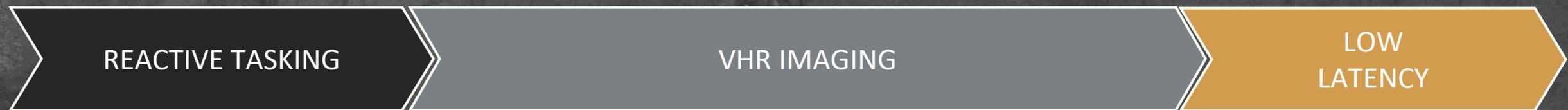
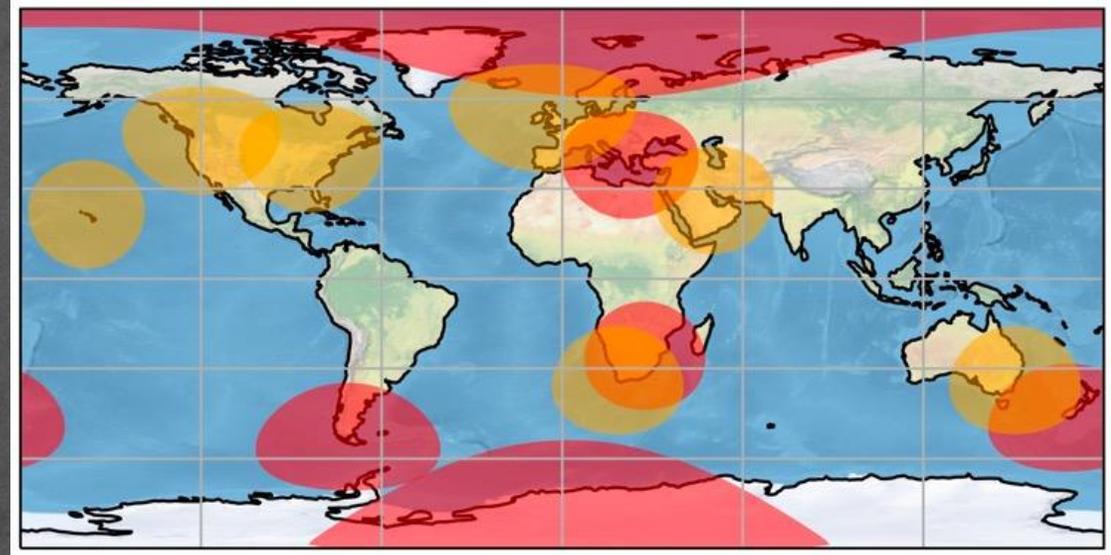
CONOPS, Ground Segment &  
Tasking Platform

# Nominal Tasking & Delivery

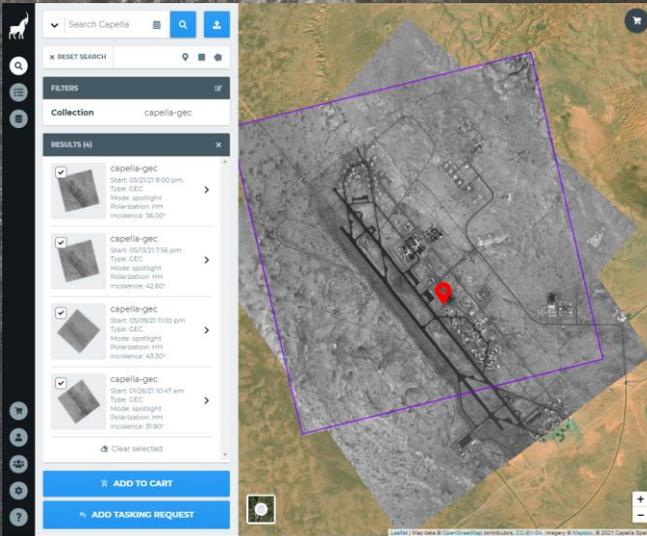


# Capella Rapid Response And Low Latency Delivery

- Reactive low-latency imaging system
- Fast data turn around times
- Fully-automated, and Secure



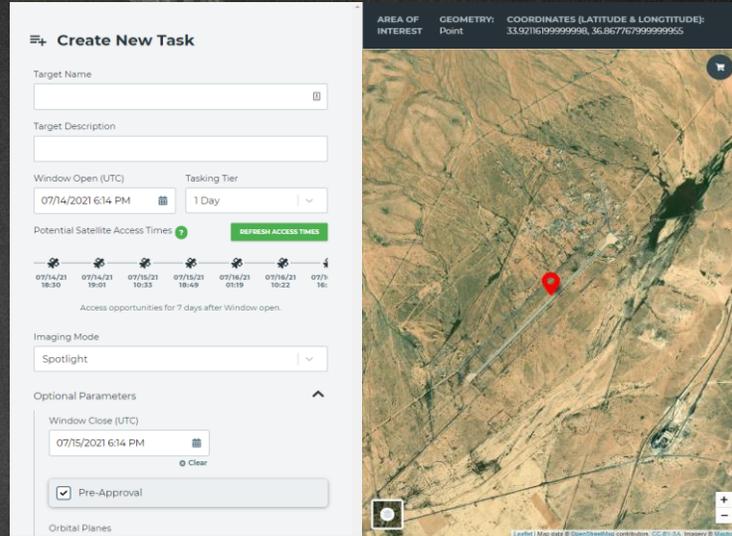
# Capella User Console Interface & API



## Catalog Browsing & API Integration

Full catalog of historical collection for baseline metrics & pattern of life analysis.

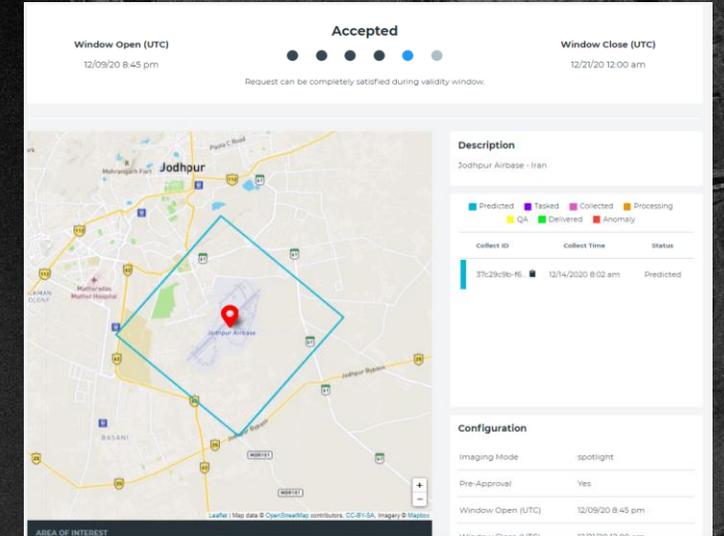
Tip-and-cue scenario for immediate responsiveness via API integration.



## Simple-to-Use GUI

Task or purchase archived imagery via coordinates, AOI creation tool or shapefile upload.

Fully automated and secure operations: Satellite ops, SAR processing and data storage are cloud based, fully confidential.



## Real-Time Status Updates

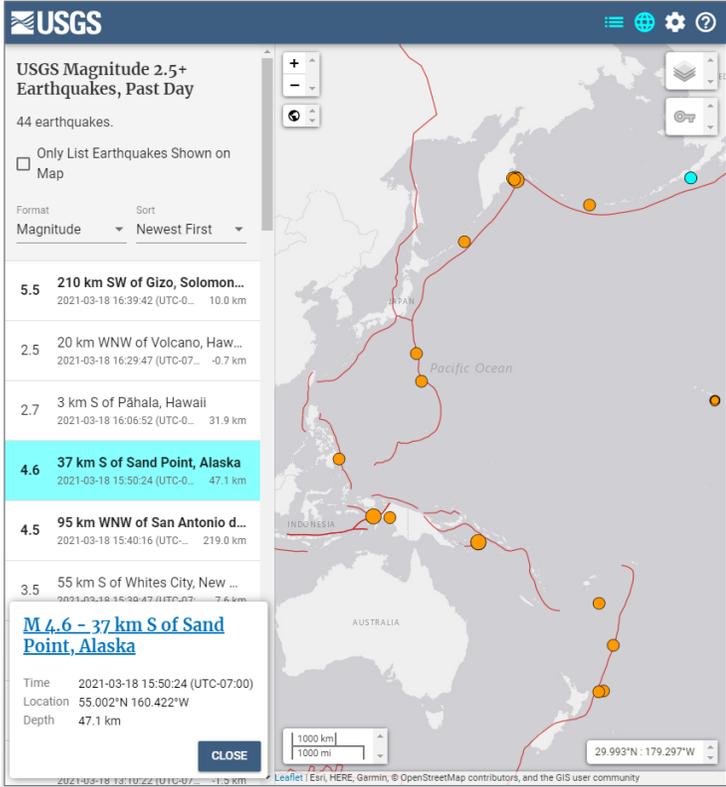
New tasking scheduling in  $\leq 15$  minutes and users are provided real-time status updates.

React to emergencies in real-time.

Deliver to teams on the ground hours after capture.

# Capella API Integration

- Tip-and-cue scenario for immediate responsiveness via API integration. Existing system alerts can push task requests.
- React to emergencies in real-time. Deliver data to teams on the ground hours after image capture.



Queue from Existing Systems

```

Access Request

In [3]: #Posting an access request POINT TARGET
accessrequest = {
    "geometry": {
        "type": "Point",
        "coordinates": [
            -76.018611, 7.248361
        ]
    },
    "properties": {
        "userId": user_id,
        "orgId": org_id,
        "accessrequestDescription": "Aleutian Islands Earthquake",
        "accessrequestName": "Aleutian Islands Earthquake",
        "windowOpen": "2021-03-17T00:00:00.000Z",
        "windowClose": "2021-03-17T00:00:00.000Z",
        "accessConstraints": {
            "type": "Feature"
        }
    }
}
headers = {'Authorization': 'Bearer ' + accesstoken}
r = requests.post("https://api.capellaspace.com/ma/accessrequests/", json=accessrequest, headers=headers)
requestid = r.json()["properties"]["accessrequestid"]

#Print the response
p(r.json())

{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [
      -76.018611,
      7.248361
    ]
  },
  "properties": {
    "accessrequestid": "98f68d1f-c847-4b36-b34d-69a7d8c8e4d3",
    "accessrequestName": "Aleutian Islands Earthquake",
    "accessrequestDescription": "Aleutian Islands Earthquake",
    "orgId": "6e4d9d5d-9e4e-474e-855b-78a2a5cc3893",
    "userId": "587f099a-22da-4a89-965e-ba0531938718",
    "windowOpen": "2021-03-17T00:00:00.000Z",
    "windowClose": "2021-03-17T00:00:00.000Z",
    "processingstatus": "queued",
    "accessibilitystatus": "unknown",
    "accessibilitymessage": "Accessibility Pending Processing",
    "accessconstraints": {
      "loaddirection": "either",
      "ascDesc": "either",
      "localTime": [
        0,
        86400
      ]
    },
    "offNadirMin": 5,
    "offNadirMax": 40
  }
}

In [ ]: #Get the status of the access request
r = requests.get("https://api.capellaspace.com/ma/accessrequests/" + requestid, headers=headers)
p(r.json())

In [ ]: #Get the accesses
r = requests.get("https://api.capellaspace.com/ma/accesses/" + requestid, headers=headers)
p(r.json())
  
```

Task the Capella Constellation

```

1 # Post search filters
2 filters = {
3   "datetime": "2020-01-01T00:00:00Z/2020-06-01T12:31:12Z", # start date/end date and
4   "limit": 200, # overwrite the default pagination limit of 10, adjust as necessary
5   "collections": ["capella-aerial"], # specify the desired collection
6   "sort": [{"field": "dtr:start_datetime", "direction": "desc"}],
7 }
8 headers = {'Content-Type': 'application/json',
9   'Accept': 'application/geo+json', 'Authorization': 'Bearer ' + accesstoken}
10 r = requests.post(URL + catsearch, json=filters, headers=headers)

1 # Display the results on a folium map
2 import folium
3
4 m = folium.Map(location=[32.834686, -117.130775], zoom_start=9)
5 folium.GeoJson(
6   r.text,
7 ).add_to(m)
8
9 m
  
```

Leaflet | Data by © OpenStreetMap, under ODBL.

Pull Scenes & Metadata





# Custom SAR Imagery Products

Power to control SAR imaging acquisition parameters to get the exact data products that meet your specific needs

Custom Imaging Parameters			
Window Open	Window Close	Tasking Tier	Imaging Mode
Scene Length	Look Direction	Look Angle Minimum	Look Angle Maximum
Ascending / Descending		Azimuth Resolution Minimum	Azimuth Resolution Maximum
Number Of Looks	Ground Range Resolution Minimum		Ground Range Resolution Maximum



# Demo