An Automated Smart Tasking System to Support NASA Urgent Response

Cathleen E. Jones, Hook Hua, Alexander Torres, Namrata Malarout
Jet Propulsion Laboratory, California Institute of Technology

2021 Earth Science Technology Forum
Earth Surface Deformation and Change
July 1, 2021

Jet Propulsion Laboratory
California Institute of Technology

ESTO
Earth Science Technology Office

Vision

Dynamic Retasking of Satellite Assets

Ground-based sensor webs and on-board processing provide triggers for targeted acquisitions from multiple instruments and missions.

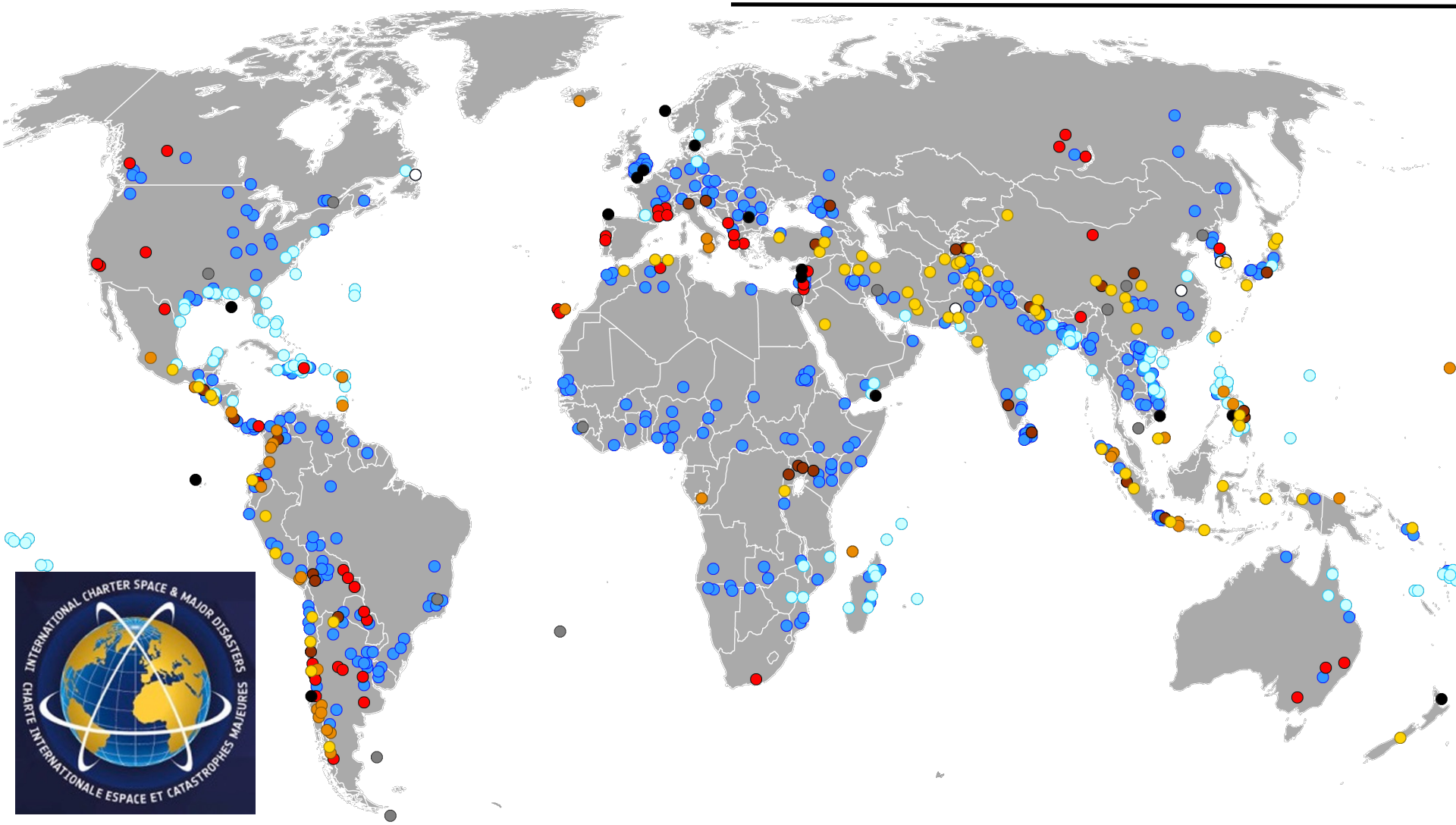
Observe dynamic processes

Support multi-band studies

Disaster response

Limit data volume

Today's Challenge: NASA's Contribution to Disaster Response



Legend: Earthquake · Landslide · Volcano · Storm/hurricane · Flood/ocean wave · Ice/snow hazard · Fire · Oil spill · Other

Background

The idea originated from a need for missions to respond to disasters more quickly => reduce or eliminate human-in-the-loop. This reduces the time between when a disaster occurs and when information is in the hands of responders.

Two-fold: Mission instrument tasking/retasking and Processing tasking

Potential Programs Supported:

- ARIA – Disaster Response
- NASA Applied Science Disasters Program (David Green)
- NISAR – Urgent Response L1 Requirement (NISAR Program, Gerald Bawden)
- NASA Multi-Mission Disaster Response (Disasters, David Green)
- New Observing Strategies
- NASA Mission Formulation & Study Teams
- R&A Program science & applications w/ short-timescale system dynamics

Smart Tasking Overview

Automated Smart Instrument Tasking for NASA Urgent Response (Smart Tasking)

ESTO Funding, Aug. 2019-March 2021

Objective

- Develop a prototype urgent response tasking request system suitable for use by multiple NASA spaceborne missions and urgent response clients.
- Design the system to be compatible with the NASA NISAR mission and prepare for infusion.
- Demonstrate the system with the ARIA processing platform.

Smart Tasking Overview

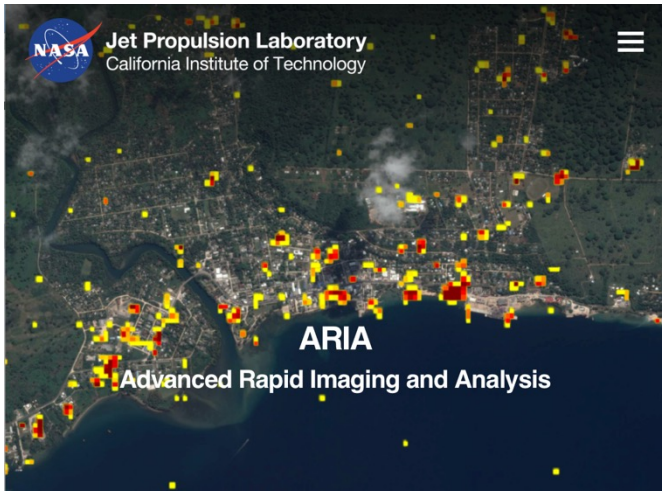
Automated Smart Instrument Tasking for NASA Urgent Response (Smart Tasking)

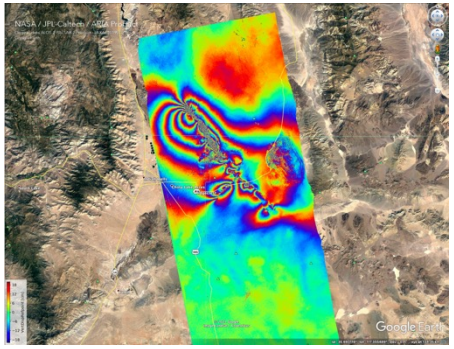
Automatically requests ***urgent response*** tasking for

1) *Earthquakes (global extent)*
2) *Major fires (U.S.)*
3) *Volcanic eruption*

Include **web interface** to handle ***manual requests,*** e.g., for *International Disasters Charter activations*.

Leveraged team's experience with ARIA





Smart Tasking Implementation

The Smart Tasking disaster response system has these features:

- Flexible architecture that can accommodate multiple inputs and multiple clients (expandable)
- Uses available triggers maintained by major U.S. federal agencies for automated tasking
- Implemented in the cloud for reliability and resilience
- Provides a database to store and track all requests
- Collects performance and usage metrics
- Communication protocol consistent with NASA missions' cybersecurity requirements

Automated request handlers developed for **four levels of cyber-sophistication**:

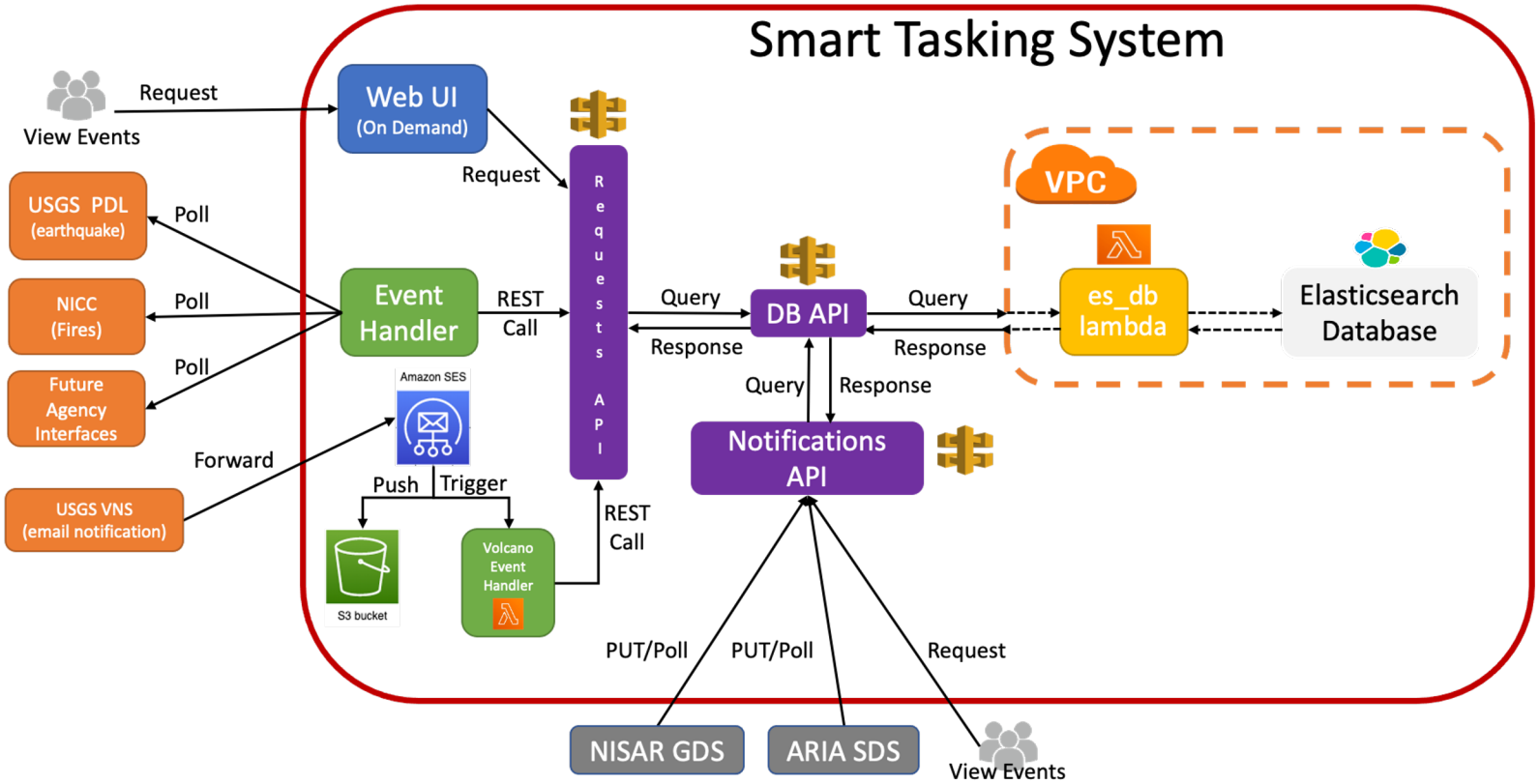
High tech: Earthquakes, USGS Product Distribution Layer (API to access EQ database)

Medium tech: Volcanos, USGS Volcano Notification Service (automated email re. volcanic activity)

Low tech: U.S. Forest Service: Internet accessible file updated daily

(Almost) No tech: Web-based manual user interface for making and tracking requests

Smart Tasking Architecture



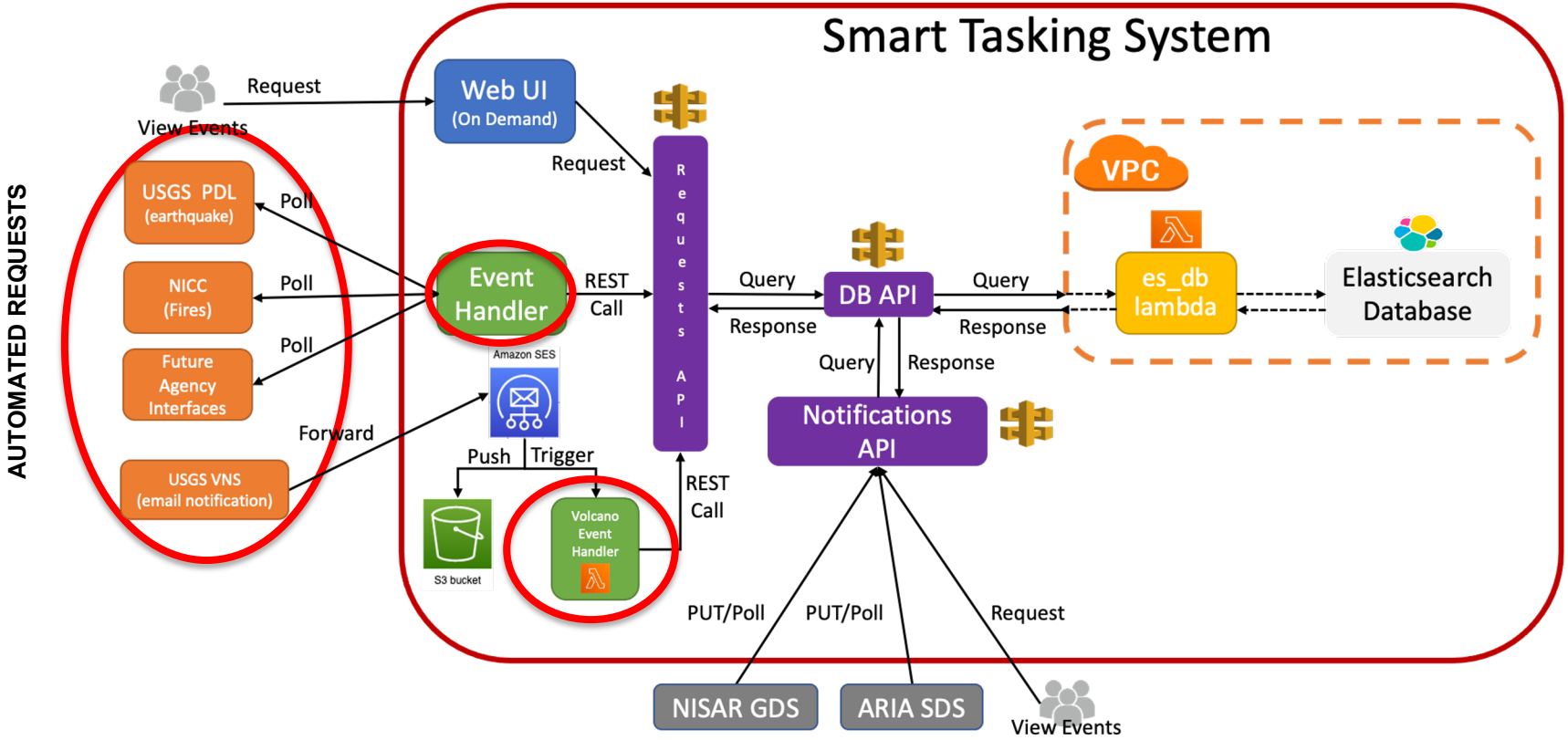
Implemented entirely in AWS

Smart Tasking Request Flowdown (Automated Requests)

Automated Requests –

Event Handlers compare to threshold (pre-negotiated and changeable)

- Above => Request accepted, no additional vetting needed
- Below => Ignored, no further action taken



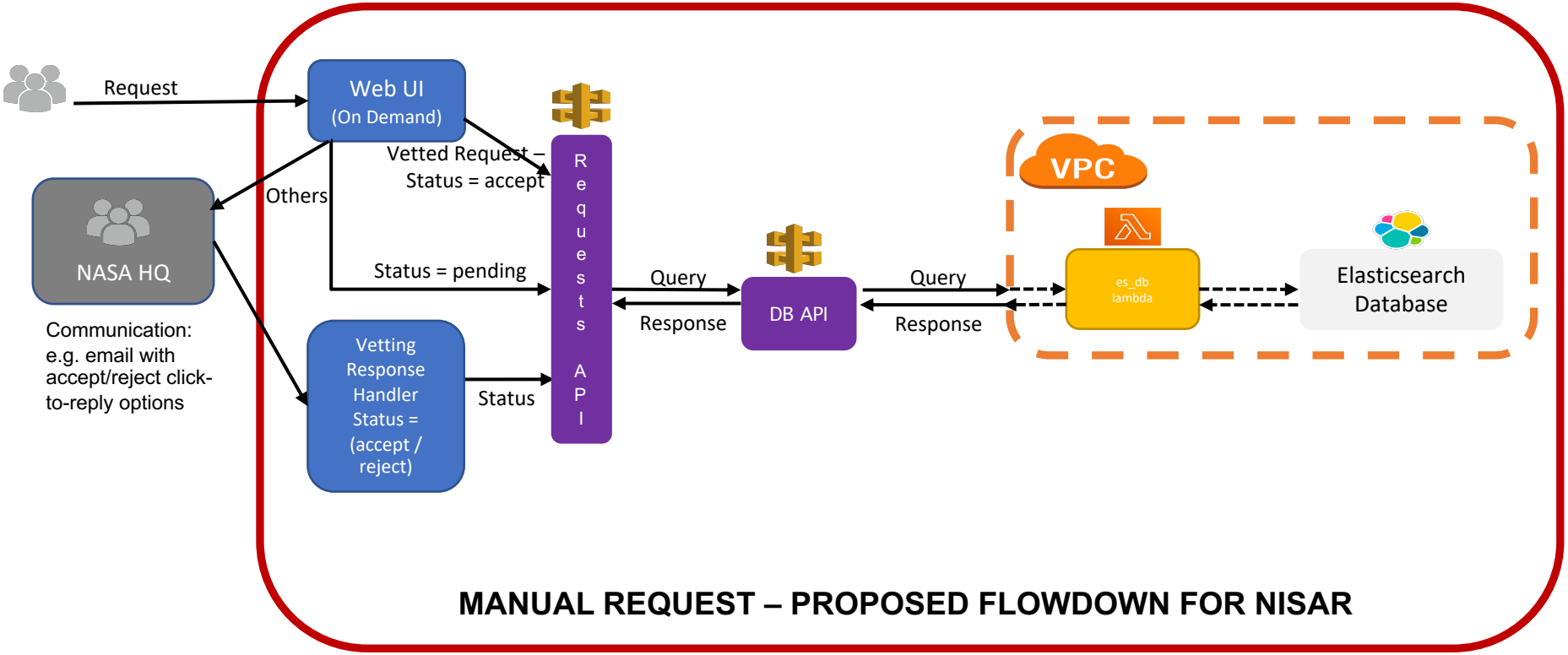
Smart Tasking Request Flowdown (Manual Requests)

Manual Requests –

Web User Interface requires login credentials (recommended by Science Team Urgent Response Working Group to be limited to specific agencies) --- UI implemented fully with login requirement, security & credential A&A being worked with NISAR

1) From vetted requester authorized to place request: Request accepted => database + NISAR GDS --- DONE
2) From others: Request forwarded to NASA HQ (NISAR Program & Disasters Program) for vetting --- NOT DONE

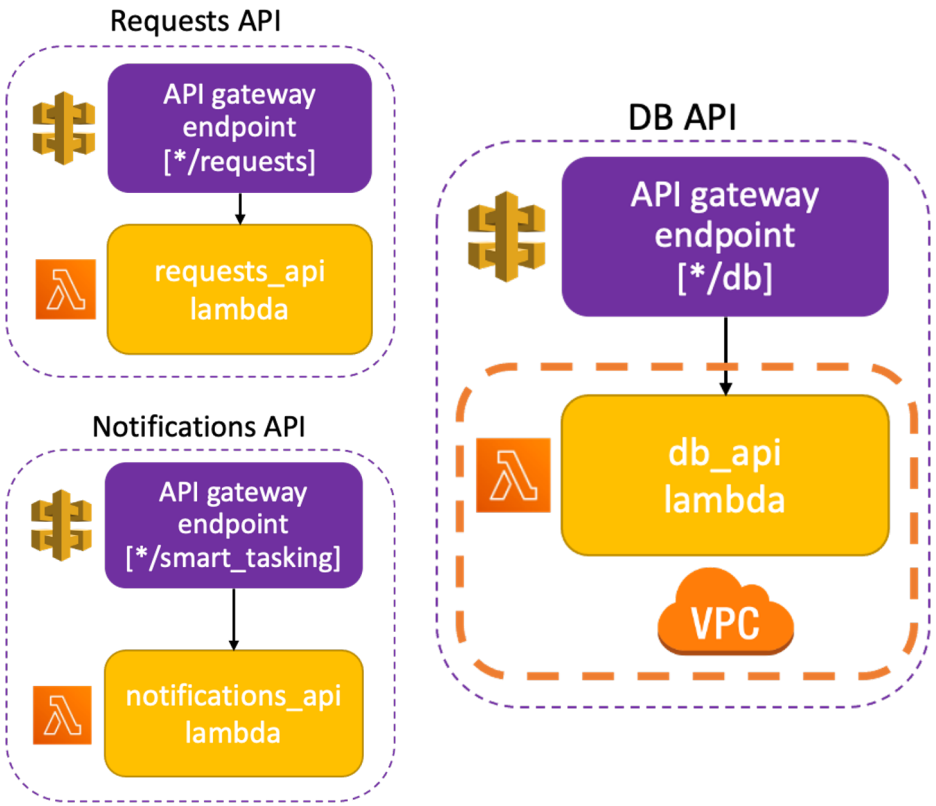
Only accepted requests sent to NISAR GDS, other clients --- DONE



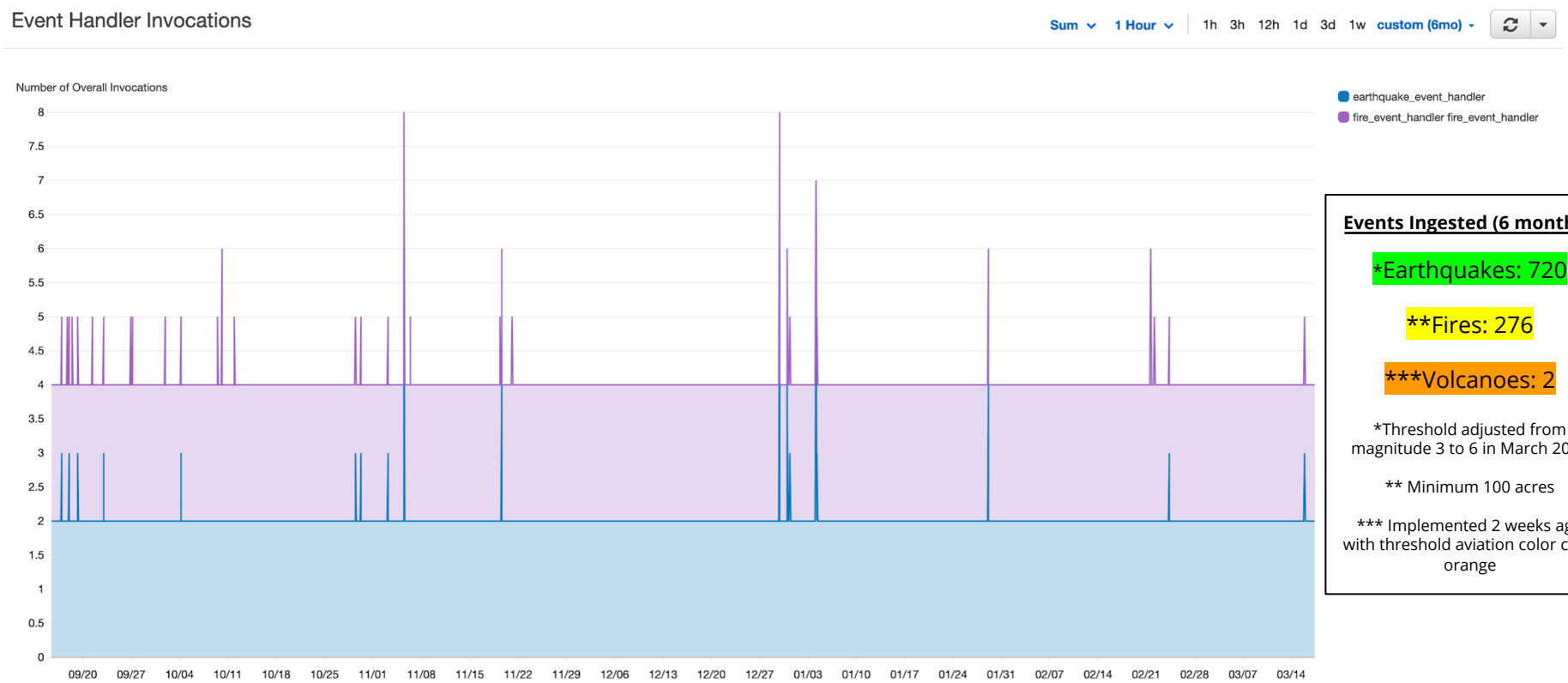
Smart Tasking Services

Application Programming Interfaces for Services

- Event Requests Service:
  - Purpose: Submit requests
  - Interfaces: Public API, UI
- Event Catalog Service:
  - Purpose: Ingest and catalog requests
  - Interfaces: Internal API
- Notifications and Decision Response Service
  - Purpose: Query requests
  - Interfaces: Public API, UI



Automated Event Metrics



Events Ingested (6 months):

*Earthquakes: 720

**Fires: 276

***Volcanoes: 2

*Threshold adjusted from magnitude 3 to 6 in March 2021

** Minimum 100 acres

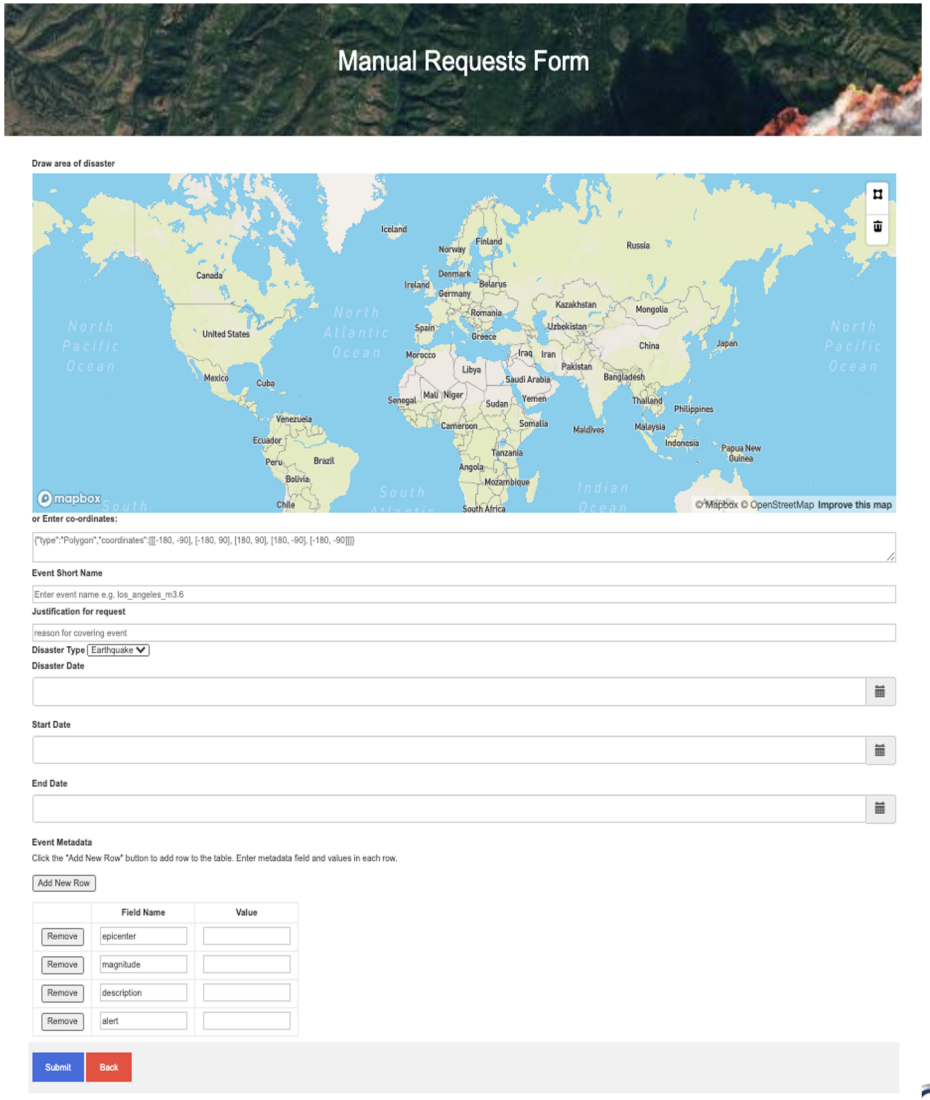
*** Implemented 2 weeks ago with threshold aviation color code orange

Manual Requests (Via UI)

Overview:

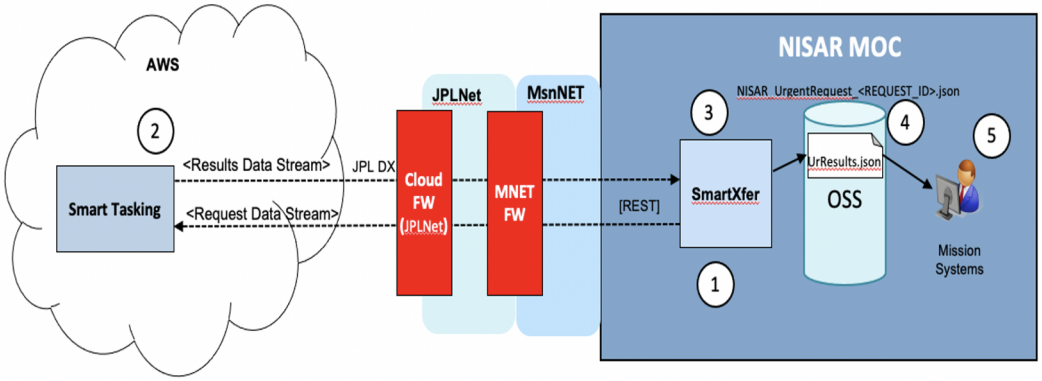
This web page is intended for registered Smart Tasking users to manually request disaster events based on the following:

- area of interest (GeoJSON polygon)
- event short name
- disaster type (i.e. earthquake, fire, volcano, etc.)
- disaster date
- observation start date (optional)
  - Defaults to disaster date
- observation end date (optional)
  - Defaults to disaster date + 30 days
- event metadata

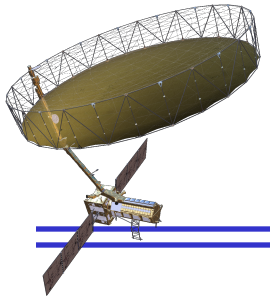


Prep for NISAR Infusion

- NISAR Infusion used as a proxy for other NASA missions
- Communication protocol developed and prototyped with the NISAR Ground Data System
- Message content and handling consistent with the NISAR Urgent Response Request Software Interface Specification Document (SIS)
- Security, authentication, and authorization method developed in conjunction with NISAR Mission Systems & JPL IT Cybersecurity Office

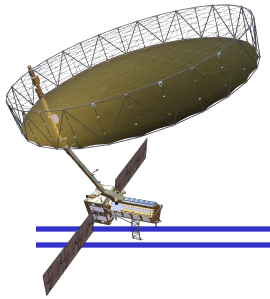


NISAR Mission System plan for Urgent Response with Smart Tasking



Smart Tasking for NISAR

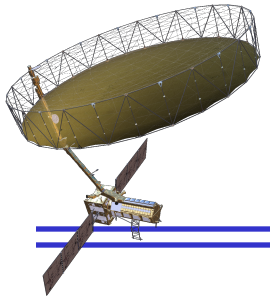
- Smart Tasking has been adopted by NISAR Missions Systems as their Urgent Response request interface.
    - Will be managed by the NISAR Science Data System
- The NISAR MS intends for all requests for Urgent response instrument retasking or urgent response data processing to come through this interface
- Enables full automation of request processing within Mission Systems software.
- Preparing now for delivery of the initial version Smart Tasking for the December 2021 Joint Thread Test
- NISAR is scheduled to launch in Jan. 2023



Summary

Keys to Smart Tasking success

- Flexible architecture
- Forward-thinking implementation
- Coordination with NISAR and JPL IT Cybersecurity to address mission security issues (very different from ARIA)
- Development effort coordinated with NISAR project early on to improve infusion potential.



Acronyms

- API — Application Programming Interface
- ARIA — Advanced Rapid Imaging and Analysis
- AWS — Amazon Web Services
- DB — Database
- GDS — Ground Data System
- GUI — Graphical User Interface
- MS — Mission Systems
- NISAR — NASA-ISRO Synthetic Aperture Radar
- PDL — Product Distribution Layer
- REST — Representational State Transfer
- SDC — Surface Deformation and Change
- STV — Surface Topography and Vegetation
- ST — Smart Tasking
- UI — User Interface
- VNS — Volcano Notification System

Extra from Demo

Demo: Query Request ID via Notifications API

Overview:

This page is designed to instruct users how to use the Smart Tasking Notifications API. With this API, users will be able to query events registered within the Smart Tasking database based on:

- **request id** (request_id)
- **event type** (type)
- **temporal range** (start_time, end_time)
- **event ingestion time** (ingest_since)
- **event time** (event_time_since)

Requirements:

- Connect to JPL **Full** Tunnel VPN.
- A Command Line Interface (Examples were used with OS and Linux Terminals)

Endpoint:

This URL endpoint will be used to construct queries that gather events from Smart Tasking database via a GET method.

URL endpoint: https://er4e4wetd8.execute-api.us-west-2.amazonaws.com/smart_tasking/events

Example:

```
$ curl https://er4e4wetd8.execute-api.us-west-2.amazonaws.com/smart_tasking/events?ingest_since=2021-02-01T00:00:00
```

Demo: Query Request ID via Notifications API

Output Response:

The output of the query response will be represented in format stated in 3.1.1 NISAR SIS Urgent Response Request (JSON).


```
[
  {
    "REQUEST_ID": "event-fire-2021-03-08T00:00:00.000Z-Rifle_Pit_Ranch-nicc_usda_cloudvault",
    "DISASTER_TYPE": "fire",
    "DISASTER_DATE": "2021-03-08T00:00:00.000Z",
    "OBS_START_DATE": null,
    "OBS_END_DATE": null,
    "REQUEST_INITIATOR_NAME": "nicc_usda_cloudvault",
    "REQUEST_INITIATOR_EMAIL": "new.cloudvault.usda.gov",
    "LOCATION": {
      "coordinates": [[...]],
      "type": "Polygon"
    },
    "EVENT_METADATA": {...},
    "PROJECT_DECISION": pending
  }
]
```

Demo: Post Project Decision via Notifications API

Overview:

This page is designed to instruct users how to use the Smart Tasking Notifications API. With this API, users will be able to query events registered within the Smart Tasking database based on:

- **request id** (REQUEST_ID): "event-<event_type>-<event_time>-<location>-<source>"
- **project decision** (PROJECT_DECISION): "accept|reject"

Requirements:

- Connect to JPL **Full** Tunnel VPN.
- A Command Line Interface (Examples were used with OS and Linux Terminals)
- Valid JPL LDAP credentials

Endpoint:

This URL endpoint will be used to construct queries that gather events from Smart Tasking database via a PUT method.

URL endpoint: https://er4e4wetd8.execute-api.us-west-2.amazonaws.com/smart_tasking/project_decision

Example:

```
$ curl curl -X PUT -d '{"REQUEST_ID":"event-fire-2020-01-01T01:00:00.000Z-test-anonymos", "PROJECT_DECISION" : "acceptlreject"}'  https://er4e4wetd8.execute-api.us-west-2.amazonaws.com/smart_tasking/project_decision
```

Documentation: https://wiki.jpl.nasa.gov/display/smarttasking/How+to+Post+Project+Decision+to+Smart+Tasking

Demo: Post Project Decision via Notifications API

Output Response:

The output of the query response will be represented in format stated in 3.1.2 NISAR SIS Project Response (JSON).


```
{
  "success": true,
  "response": "Successfully updated REQUEST_ID: event-fire-2020-01-01T01:00:00.000Z-test-anonymos"
}
```

```
[
  {
   "REQUEST_ID": "event-fire-2020-01-01T01:00:00.000Z-test-anonymos",
    "DISASTER_TYPE": "fire",
    "DISASTER_DATE": "2020-01-01T01:00:00.000Z",
    "OBS_START_DATE": "2020-01-01T00:00:00.000Z",
    "OBS_END_DATE": "2020-01-02T00:00:00.000Z",
    "REQUEST_INITIATOR_NAME": "anonymos",
    "REQUEST_INITIATOR_EMAIL": "manual request portal",
    "LOCATION": {
      "coordinates": [[...]],
      "type": "Polygon"
    },
    "EVENT_METADATA": {...},
    "PROJECT_DECISION": accept
  }
]
```


Documentation: https://wiki.jpl.nasa.gov/display/smarttasking/How+to+Use+Notifications+API

ARIA Smart Tasking Listener

- ARIA PGE that pulls events (i.e. earthquakes, fires, or all) from Smart Tasking system into ARIA system.
- Uses parameters to constrain query:
  - **disaster_type**
  - **disaster_source**
  - **query_since**
  - **starttime**
  - **endtime**
  - **geojson_polygon**
- Can be used automatically (via crontab)

  or on-demand (via ARIA.on-demand UI)