Product User Guide

Near Real Time Active Fire OT-NRT

Document Version 1.0.2 - September 02, 2026

support@ororatech.com | ororatech.com

Beta

Contents

1. Introduction

This user guide document describes OroraTech's Near Real Time Active Fire product (OT-NRT), detailing the product specifications, characteristics and performance metrics. Furthermore, the document gives an overview of the involved satellite missions, and describes the data processors and utilized cal/val methods.

1.1 Product Maturity - Closed Beta Phase

Beta

The OT-NRT product is currently in a closed beta phase. Performance may vary, and the product may not be available at all times.

During this stage, we are continuously optimizing performance, accuracy, and coverage based on real-world data and user feedback.

However, the system is already operating at a level that delivers meaningful value to users, and we are actively working to further enhance its robustness and reliability ahead of full release.

All performance indicators throughout this description are described as targets. The product is delivered as-is.

1.2 Reference Documents and Datasets

[1] OroraTech Level-1 Product User Guide Issue 1.2.1, 25/08/2026

[2] Global Solar Power Tracker, Global Energy Monitor, February 2026 release.

[3] Stid, J. et al. (2025) "GM-SEUS: A harmonized dataset of ground-mounted solar energy in the US with enhanced metadata", Scientific Data. Zenodo. https://doi.org/10.5281/zenodo.14827819.

[4] Stid, J. et al. (2025) "A harmonized dataset of ground-mounted solar energy in the US with enhanced metadata", Scientific Data, 12(1), p. 1586. Available at: https://doi.org/10.1038/s41597-025-05862-4.

[5] Zanaga, D. et al. (2022) "ESA WorldCover 10 m 2021 v200". Zenodo. Available at: https://doi.org/10.5281/zenodo.7254221.

2. Mission

The OT-NRT active fire product is based on OroraTech's proprietary fleet of satellites. At the current state the 8 satellites comprising OroraTech Constellation Phase-1 (OTC-P1) are the sole data source for the OT-NRT product. Newly launched satellites will be added to the system to improve the revisit rate in the future.

2.1 OroraTech Constellation Phase-1 (OTC-P1)

OroraTech Constellation Phase-1 is a thermal-infrared (TIR) satellite constellation operated by OroraTech specialized in wildfire detections. OTC-P1's FOREST satellites (FOREST-4P through FOREST-11P) are equipped with OroraTech's inhouse-developed SAFIRE instrument that features two TIR telescopes with a combined swath of 410 km that generate data in one infrared (LWIR) band (L2) and one medium wave infrared (MWIR) band (M0) with a ground sampling distance (GSD) of 200 m.

The satellites are spaced along a synchronous orbital plane and provide two daily observations for selected regions around the globe (around 4 am and 4 pm local time). Wildfires are automatically detected by the satellites using state-of-the-art onboard AI image processing and notifications are delivered to WFS users as part of the OT-NRT product.

2.2 Mission Characteristics

2.2.1 Satellites

Table 1: *OTC-P1 mission characteristics.*

Number of satellites	8	
Start of operations	Q2 2026	
End of operations	Q1 2029 (TBC)	
Imager Type	SAFIRE	
Orbit	525x550km, Sun-synchronous Mean Local Time at Ascending Node (MLTAN) at 16:00±15 min as of 2026	
Inclination	97.53°	
Revisit time monitoring (average)	0.5 days	
Revisit time tasking	0.5 days	
Detector	Uncooled microbolometer	
NORAD ID	F004: 63357 F005: 63354 F006: 63353 F007: 63358	F008: 63355 F009: 63352 F010: 63356 F011: 63351

COSPAR ID / NSSDCA ID (International Designator)	F004: 2025-061G F005: 2025-061D F006: 2025-061C F007: 2025-061H	F008: 2025-061E F009: 2025-061B F010: 2025-061F F011: 2025-061A

2.2.2 SAFIRE Instrument

The thermal imager deployed on the OTC-P1 satellites was developed by OroraTech and utilizes uncooled microbolometer technology. It was designed for detecting heat signatures of active wildfires with improved spatial resolution and features onboard processing capabilities to detect fires immediately after a given area was observed.

Table 2: *SAFIRE characteristics.*

Instrument name	SAFIRE
Spectral bands	M0: 3.4 – 4.2 µm
	L2: 10.4 – 12.5 µm
GSD at Nadir	Design: 200 m Threshold (max.): 250 m
Product resolution	200 m
Swath	Per TIR telescope: ● Design: 210 km ● Threshold (min.): 170 km Combined: ● Design: 410 km ● Threshold (min.): 330 km
Dynamic range	M0: 300 - 650 K
	L2: 260 - 640 K
Viewing angle	<50°

2.3 Observation Modes and Coverage Extent

OTC-P1 operates primarily in a near-real-time scanning mode optimized for active fire detection over large areas. The satellite is tasked for several operational use cases, such as systematic monitoring of high-risk regions, regular acquisitions over customer areas, and support of emergency response activities.

In nominal operation, OTC-P1 can acquire images globally with roll angles up to ±30° to decrease the revisit time over regions of interest. All imaging data relevant for fire detection and monitoring are processed directly after acquisition onboard the satellite to generate the OT-NRT product, i.e. on-orbit active fire detections with minimal latency.

3. Product Description

Like other thermal satellites, these scan the surface of the Earth for thermal anomalies that might indicate the presence of an active fire. The sensor captures the thermal data like a camera (but in a long strip) which is subsequently passed directly to the onboard processing unit for detection of hotspots. Each strip is usually between 150,000 to 250,000 km² in area, and the processing unit can identify thousands of individual hotspots at 200 m resolution - much higher than hotspots provided by other satellites, which can vary from up to 2 km at nadir (e.g. Himawari-9) down to 375 m (e.g. VIIRS sensor on SUOMI-NPP, NOAA-20 and NOAA-21).

In its current state OT-NRT provides up to 2 daily scans of a customer's area: One around 4 am and another one around 4 pm local time. Our satellites were launched in this specific orbit to provide data during the peak afternoon burn period, when coverage from other satellites is limited.

The OT-NRT product is provided as part of OroraTech's Wildfire Solution platform, either via the frontend or via the WFS API.

For the following description general knowledge about OroraTech's fire-related terminology is assumed. If needed, please refer to the OroraTech's Help Center page about Clusters and Hotspots before continuing.

3.1 OT-NRT Metadata

Metadata for the OT-NRT product is accessible through the WFS API's fire-related endpoints, mirroring the data structure used for public satellite information within WFS. You can find additional details about the metadata, including unit specifications, in the API documentation.

Table 3: *Relevant OT-NRT metadata available via the WFS API. The "Level" section indicates if the information is provided at the individual hotspot level, the cluster level, or both.*

Field	Level	Description	Example	Default unit/format
acquisition_time	Hotspot	Timestamp in UTC of satellite acquisition	"2026-03-26T12:30:00Z"	ISO 8601
age	Both	Relative age of detection or cluster	120	minutes
algorithm	Hotspot	Algorithm that recognized the hotspot	OT-NRT	-
algorithms	Cluster	Detection algorithms contributing to clusters	["MODIS-Collection6-Active-Fire-Product", "OT-AI"]	-

Field	Level	Description	Example	Default unit/format
area	Cluster	Estimated affected area	1500	m^2
cause_string	Cluster	Human-readable cause	"Unknown"	-
causes	Cluster	Numeric cause classification	[0]	-
centroid	Cluster	Center point of cluster geometry	{"lat": 37.45, "lon": 14.22}	decimal degrees (WGS84)
cluster_id	Hotspot	ID linking hotspot to a cluster	9876	-
cluster_type	Cluster	Associated type of the cluster (see Section 3.1.1 for more information)	"industry"	-
detection_time	Hotspot	Timestamp when hotspot was detected / processed in WFS	"2026-03-26T12:35:00Z"	ISO 8601
fire_confidence	Cluster	Advanced fire confidence value associated with cluster (see Section 3.1.2 for more information)	0.85	range [0, 1]
gsd	Hotspot	Ground sampling distance	200	meters
id	Both	Unique identifier of hotspot or cluster	12345	-
lat / lon	Hotspot	Geographic coordinates of hotspot	37.45 / 14.22	decimal degrees (WGS84)
is_latest_detection	Hotspot	Indicates whether the detection is the most recent for the hotspot	true	-
lifetime	Cluster	Duration of cluster activity	360	minutes
newest_acquisition	Cluster	Latest hotspot acquisition time in cluster	"2026-03-26T12:30:00Z"	ISO 8601
newest_detection	Cluster	Latest hotspot detection time	"2026-03-26T12:35:00Z"	ISO 8601

Field	Level	Description	Example	Default unit/format
num_fires	Cluster	Number of hotspot events in cluster	25	-
oldest_acquisition	Cluster	First hotspot acquisition time in cluster	"2026-03-26T10:00:00Z"	ISO 8601
oldest_detection	Cluster	First hotspot detection time	"2026-03-26T10:05:00Z"	ISO 8601
orbit_type	Cluster	Orbit type used for cluster generation	3 (= GEO, LEO)	-
satellite_name	Hotspot	Name of satellite detecting the hotspot	"FOREST-5P"	-
satellite_orbit_type	Hotspot	Orbit type of satellite (LEO/GEO)	"LEO"	-
satellite_product_filename	Hotspot	Source product filename	“F009_L2__TA_________2026-04-04T144111.919767Z_2026-04-04T144545.796357Z_45964fec.nc4”	-
satellites	Cluster	Satellites contributing to cluster	["FOREST-5P","FOREST-11P"]	-
type_string	Cluster	Human-readable classification	"Fire"	-
types	Cluster	Numeric classification types	[1]	-

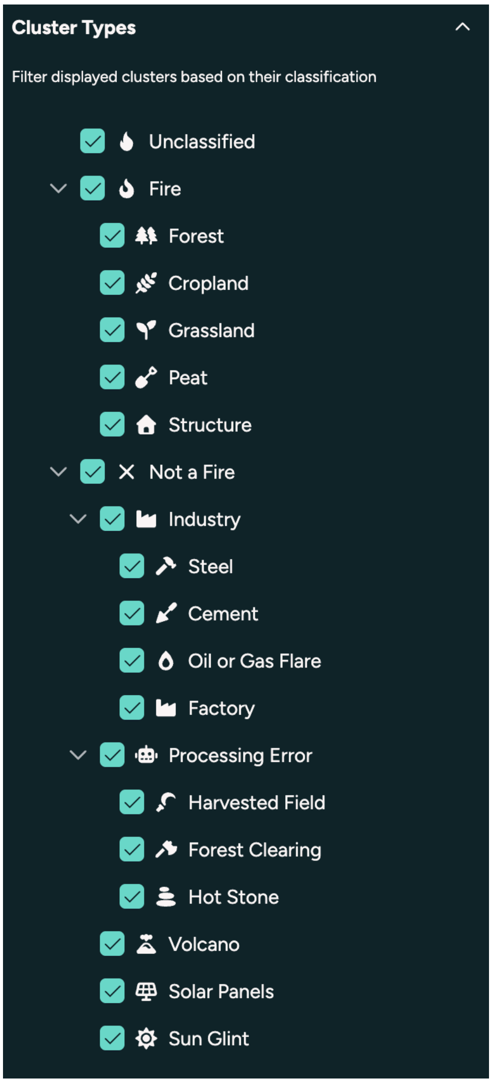
3.1.1 Cluster types

The cluster_type field is an integer on fire clusters (and hotspots) that classifies the type of event a cluster represents — for example, a real fire or an industrial heat source. All available values are listed in Table 4. Note that values 14, 15, and 16 are not used.

The cluster_type is determined by user classification, automated classification, or association with an external incident. Historical cluster_type data from the same location may also influence the assigned value.

__Table 4:__ List of different cluster_types and image taken from the filter menu on the platform UI.

cluster_type	Description/name
0	unclassified
1	fire
2	industry
3	volcano
4	falseDetection
5	forest
6	cropland
7	grassland
8	structure
9	factory
10	steel
11	flare
12	cement
13	peat
17	solar
18	sun
19	processing
20	harvest
21	forestClearing
22	hotStone



3.1.2 Fire Confidence

"Fire Confidence", is a high-level metric describing the level of certainty with which we consider a heat anomaly to be a fire. It is derived from aggregating various key metrics that report on the local fire weather, fire radiative power and the detecting satellites and algorithms. The result is one of the following categorical values: Insufficient data (0%), Low (25%), Medium (50%), High (75%), Very high (90%). A detailed description of the fire confidence can be found on OroraTech's Help Page.

3.2 Generation of the OT-NRT product

The OT-NRT product is generated through a complex processing chain that combines machine learning algorithms deployed on our satellites, high priority downlink and multiple post-processing steps.
The following flowchart (Figure 1) shows the high-level data and processing flow. Thermal data is acquired by our satellites. Using onboard processing capabilities we generate an on-orbit Level-1 product (see [1]) that contains the medium wave infrared signal collected by our sensor mapped on the surface of the Earth. This data is fed into a state-of-the-art machine learning algorithm that detects thermal anomalies in the data on the pixel-level (so-called hotspots). The algorithm operates on the MWIR signal alone and does not currently take a cloud mask or land cover classification as input. The hotspots are then packaged and downlinked via a ground station network.

As soon as the data arrives in our ground segment, we ingest the data into our Wildfire Solution (WFS) system. False-positive hotspots over open water, as well as those associated with known data artefacts, are removed. Hotspots located in close proximity to large solar parks are flagged as potential solar farm reflections but retained (see Section 3.1.1 and Section 3.5), as reflected sunlight can produce signals similar to genuine fire detections.

In the next step, the OT-NRT hotspot data is spatiotemporally clustered with hotspot data from public satellites. The cluster now represents a collection of hotspots and is a representation of the wildfire as a whole (whereas hotspots are usually only a part of a wildfire). On the cluster level, we compute our Fire Confidence metric (see Section 3.1.2). In the next and final step, notifications about the newly detected events or a new cluster activity in a monitored area are then sent to our users via WFS Frontend or WFS API.

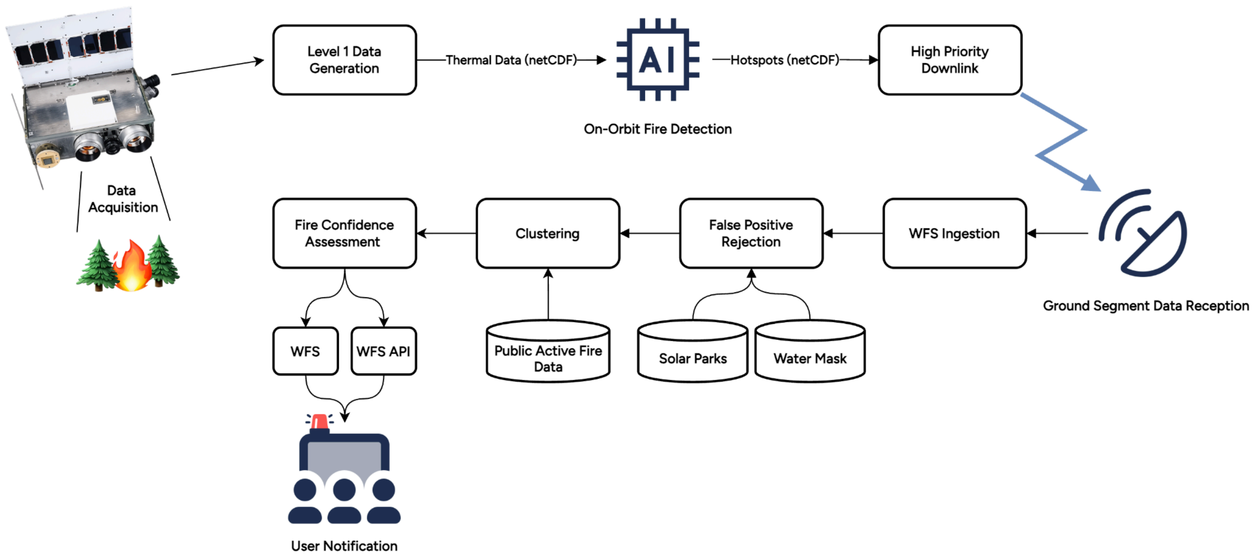


Figure 1: *High level processing workflow of the OT-NRT product generation pipeline*

3.3 Product Performance

The following performance characteristics refer primarily to the OT-NRT product generated from OroraTech's Constellation Phase 1. Where applicable, system-level aspects of the Wildfire

Solution are referenced to provide context. The performance of the OT-NRT product is primarily defined by its latency, spatial resolution, and detection capability, which depend on the underlying satellite systems and processing pipelines.

The spatial resolution of OroraTech data is 200 m, enabling the detection of smaller and lower-intensity fires compared to many public satellite systems. Once acquired, the data is processed through highly optimized pipelines, typically generating detections within minutes. The primary source of latency stems from waiting for ground station overpasses rather than from the processing itself.

The ability to correctly and accurately detect fires is the single most important performance indicator for the OT-NRT product. Our detection data is regularly assessed by the Optical Mission Performance Cluster (OPT-MPC) of the European Space Agency within the Category 1 Copernicus Contributing Missions (CCM) Program and the MPC-CCM framework. A recent assessment (June 2026) evaluated OT-NRT against the VIIRS Active Fire product and the Meteosat Third Generation (MTG) Active Fire Product. One of the reported metrics is the joint detection rate which is defined as the number of fires identified by both OT-NRT and the reference product, divided by the total number of fires detected by the reference product. The joint detection rate reached 77.8% against VIIRS, and 96.1% against MTG, indicating "reasonably good performances for fire detections for both sensors". These evaluations are still ongoing and are currently based on a relatively limited dataset. As additional data becomes available and the analysis is extended, the reported performance metrics may be refined and are therefore subject to change.

This means that you can expect OT-NRT to reliably detect the majority of fires that would be detected by other commonly used active fire products such as VIIRS, MODIS, GOES and MTG but at times of the day, where many of these satellites are not available and at a higher spatial resolution

3.4 Product Validation

Validation aspects of the OT-NRT product are complemented by higher-level processing within the Wildfire Solution, including clustering and multi-source aggregation. The operational validation of the OT-NRT product is based on confidence metrics, multi-source data integration, and the characterization of known limitations inherent to satellite-based fire detection.

Each detected hotspot is associated with a Fire Confidence metric (see Section 3.1.2), which expresses the likelihood that a thermal anomaly corresponds to an actual fire. This metric is derived from a combination of indicators, including environmental conditions (fuel availability, weather, and human interactions, e.g. suppression efforts), and the number and diversity of contributing satellite detections and algorithms. Confidence is provided in categorical levels ranging from low to very high, allowing users to assess detection reliability.

Validation is further strengthened through the integration of multiple satellite systems and algorithms, as well as the clustering of individual hotspots into fire events. Consistent detections across independent sources increase confidence and reduce the likelihood of false positives.

However, satellite-based fire detection is subject to inherent uncertainties. These include both false positives and missed detections, which may occur due to limitations in sensor capability, environmental conditions, or observation geometry. In particular:

- Not all hotspots correspond to actual fires (e.g. industrial heat sources)
- Fires may remain undetected if they are too small, too cool, or obscured
- Detection depends on satellite overpass timing and coverage

Besides the operational validation we continue to monitor our detection algorithms based on a large global dataset of expert-labelled fires to detect any deviations in detection performance caused by updates of the processing logic before they are deployed to our satellites.

4. Data Exploration and Delivery

This section describes how OT-NRT data can be explored interactively via the Wildfire Solution (WFS) and accessed programmatically through the WFS API. For a complete description of WFS functionality, data layers, filtering capabilities, API keys, guides and more, refer to the Wildfire Solution Service Description document and the WFS API documentation collection OroraTech Docs.

4.1 Data Exploration via WFS

The Wildfire Solution (WFS) is OroraTech's web-based platform for visualizing and analyzing near real-time wildfire activity, providing access to hotspot detections, clustered fire events, and associated geospatial and environmental data through an interactive interface.

4.1.1 Accessing and Exploring OT-NRT Data in WFS

OT-NRT data is accessible directly within the WFS map interface:

- **Hotspots** are displayed as point detections representing thermal anomalies derived from satellite observations.
- **Clusters** aggregate spatially and temporally correlated hotspots into active fire events, enabling tracking of fire progression and characteristics.

Users can explore data through:

- Map-based visualization (Figure 3)
- Cluster detail views (left)
- Hotspot detail views (right)

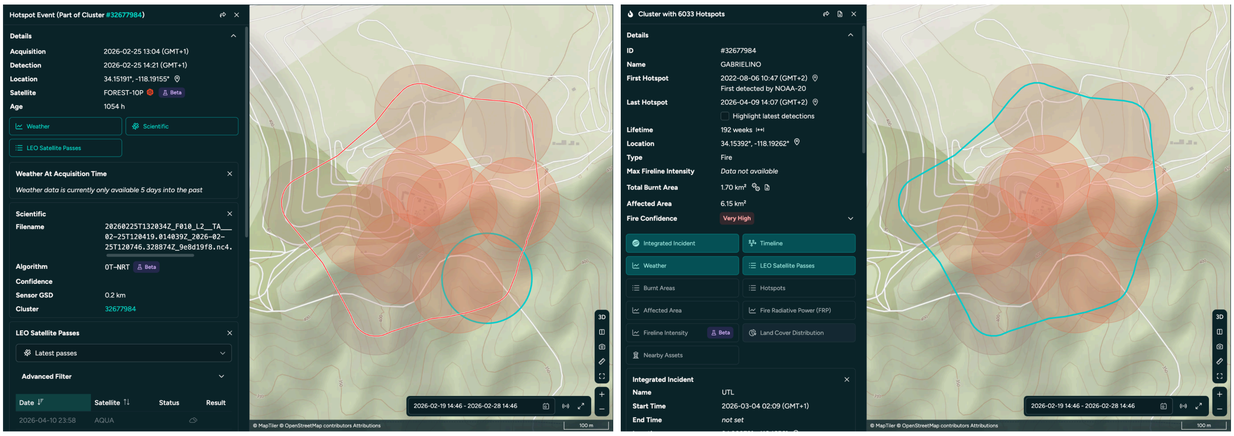


Figure 3: *Hotspot Detail View (left) and Cluster Detail view (right) in WFS, highlighted in turquoise*

4.1.2 Filtering for NRT Hotspots

WFS provides multiple filtering mechanisms to isolate relevant NRT detections:

- **Time filter (Live Mode)** for continuous updates of recent detections
- **Satellite filter** to restrict detections to specific sources (e.g. OroraTech vs. third-party)
- **Fire confidence filter** to focus on high-confidence detections
- **Algorithm filter** to select specific detection pipelines

These filters allow users to approximate OT-NRT data selection, although the system aggregates multiple data sources by design.

4.2 Data Delivery via the WFS REST API

While the Wildfire Solution enables interactive exploration, the same underlying data can be accessed programmatically via the REST API, offered as part of WFS. It provides programmatic access to active and historical wildfire data, enabling retrieval of hotspot detections, fire clusters, burnt area data, and related geospatial information in standardized formats such as GeoJSON. It also allows users to run fire spread information, task images from OroraTech's satellites, and query satellite products. It supports:

- Server-side filtering and aggregation
- Geographic queries via bounding boxes
- Integration into GIS tools and custom applications

4.2.1 Fire API & Get Endpoints

The Fire API allows retrieving information about active and past fires, such as their location, estimated perimeter, and other relevant data. Results may include false detections and other thermal anomalies (such as industry heat signatures). The Fire API's [GET] endpoints allow retrieval of wildfire-related data, primarily hotspot detections and clustered fire events. These endpoints form the core interface for accessing OT-NRT data programmatically. The Fire API's two most relevant endpoints "get fire clusters" and "get hotspots" are further explained. For full documentation and further guides, refer to the online documentation collection OroraTech Docs.

Get Hotspots

- Provides individual hotspot detections
- Includes location, timestamp, confidence, and intensity-related attributes
- Represents the closest equivalent to raw OT-NRT detections

Get Clusters

- Provides aggregated fire events (spatio-temporal clustering of hotspots)
- Includes fire evolution, affected area, and cluster-level metrics
- Represents the operational fire product derived from hotspots

All endpoints support a common set of filtering parameters, including geographic bounding boxes, time ranges, satellite sources, detection algorithms, and confidence thresholds. All get endpoints of the fire API are summarized in Table 5.

Table 5: *Fire API [GET] endpoints available in WFS API (see detailed API documentation here).*

Endpoint	Description	Key Functionality	Output Format
/v1/hotspots/	Retrieve hotspot detections (thermal anomalies) for a geographic region	Filter by time, satellites, algorithms and confidence. Returns individual detections	GeoJSON FeatureCollection
/v1/hotspots/:id	Retrieve detailed information for a specific hotspot	Provides full hotspot properties including FRP, confidence, location and acquisition time	JSON
/v1/clusters/	Retrieve fire event clusters for a geographic region	Returns full cluster shapes with filtering by time, satellites, algorithms, type and cause	GeoJSON FeatureCollection
/v1/clusters/centroids/	Retrieve fire cluster centroids for a geographic region	Returns simplified cluster representation (centroids only) for efficient queries	GeoJSON FeatureCollection
/v1/clusters/:id	Retrieve detailed information for a specific cluster	Returns cluster geometry, events and optional extended attributes via select parameter	GeoJSON Feature
/v1/clusters/:id/history/	Retrieve temporal evolution of a cluster	Provides time-resolved cluster shapes and properties per satellite overpass	GeoJSON FeatureCollection

Endpoint	Description	Key Functionality	Output Format
/v1/clusters/types/	Retrieve supported cluster type classifications	Provides mapping of type IDs to human-readable names	JSON
/v1/clusters/causes/	Retrieve supported cluster cause classifications	Provides mapping of cause IDs to human-readable names	JSON
/v1/monitored_areas/my/clusters/	Retrieve clusters within all monitored areas of the user	Applies user-defined monitoring areas with standard filtering options	GeoJSON FeatureCollection
/v1/monitored_areas/:id/clusters/	Retrieve clusters within a specific monitored area	Returns clusters intersecting a defined monitored area	GeoJSON FeatureCollection

4.2.2 Filtering for NRT Hotspots (REST API)

Unlike WFS, filtering is explicit and query-based. Filtering OT-NRT data via API is typically done using:

- **Time range filters** (recent timestamps → NRT)
- **Bounding box (bbox)** for geographic selection
- **Confidence thresholds** (if supported)
- **Satellite/source filters** (if available in endpoint parameters)

4.2.3 REST-API Metadata

The REST API of this product provides event-based and aggregated attributes relevant for geospatial querying and analysis. For a summary of fields, descriptions and examples of the Level-2 NRT AF product's main endpoints "get hotspots" and "get fire clusters", refer to section 3.2 OT-NRT Metadata or the online documentation collection [OroraTech Docs](#).

5. Python Examples

This section provides example implementations for accessing Near Real-Time Active Fire (OT-NRT) data via the OroraTech Fire API.

5.1 Retrieve hotspots

Python
```python
import requests

BASE_URL = "https://app.ororatech.com"
API_KEY = "YOUR_API_KEY"

headers = {
    "apikey": API_KEY
}

params = {
    "xmin": -123.0,
    "ymin": 36.3,
    "xmax": -119.0,
    "ymax": 38.5,
    "minutes": 1440,  # last 24 hours
    "epsg": 4326,
    "select": [
        "acquisition_time",
        "detection_time",
        "normalized_frp",
        "cluster_id",
        "algorithm"
    ]
}

response = requests.get(
    f"{BASE_URL}/v1/hotspots/",
    headers=headers,
    params=params
)

response.raise_for_status()
hotspots = response.json()

print(f"Returned {len(hotspots.get('features', []))} hotspots")
```

5.2 Retrieve Fire Clusters

Python
```python
import requests

BASE_URL = "https://app.ororatech.com"
API_KEY = "YOUR_API_KEY"

headers = {
```

```
    "apikey": API_KEY
}

params = {
    "xmin": -123.0,
    "ymin": 36.3,
    "xmax": -119.0,
    "ymax": 38.5,
    "minutes": 1440,  # last 24 hours
    "epsg": 4326,
    "unit_area": "ha",
    "select": [
        "lifetime",
        "area",
        "oldest_acquisition",
        "newest_acquisition",
        "satellites",
        "algorithms",
        "centroid"
    ]
}

response = requests.get(
    f"{BASE_URL}/v1/clusters/",
    headers=headers,
    params=params
)

response.raise_for_status()
clusters = response.json()

print(f"Returned {len(clusters.get('features', []))} clusters")
```

5.3 Filtering for OT-NRT Data

```python
import requests

BASE_URL = "https://app.ororatech.com"
API_KEY = "YOUR_API_KEY"

headers = {
    "apikey": API_KEY
}

# Common spatial filter
bbox = {
```

```python
    "xmin": -123.0,
    "ymin": 36.3,
    "xmax": -119.0,
    "ymax": 38.5,
    "epsg": 4326
}

# Hotspot filtering
hotspot_params = {
    **bbox,
    "minutes": 180,  # last 3 hours
    "satellites": ["FOREST-2"],
    "algorithms": ["OT-NRT"],
    "min_fire_confidence": "low",
    "include_no_fire_confidence": False,
    "select": [
        "acquisition_time",
        "detection_time",
        "normalized_frp",
        "cluster_id"
    ]
}

hotspot_response = requests.get(
    f"{BASE_URL}/v1/hotspots/",
    headers=headers,
    params=hotspot_params
)

hotspot_response.raise_for_status()
nrt_hotspots = hotspot_response.json()

# Fire Cluster filtering
cluster_params = {
    **bbox,
    "minutes": 180,
    "algorithms": ["OT-NRT"],
    "min_fire_confidence": "medium",
    "num_fires": 2,
    "orbit_type": ["LEO"],
    "unit_area": "ha",
    "select": [
        "lifetime",
        "area",
        "newest_acquisition",
        "satellites",
        "algorithms",
        "types",
        "causes",
        "centroid"
```

```
    ]
}

cluster_response = requests.get(
    f"{BASE_URL}/v1/clusters/",
    headers=headers,
    params=cluster_params
)

cluster_response.raise_for_status()
filtered_clusters = cluster_response.json()

print(f"NRT hotspots: {len(nrt_hotspots.get('features', []))}")
print(f"Filtered clusters: {len(filtered_clusters.get('features', []))}")
```

End of document