Computations Microservice API Documentation

March 10, 2025

1 Overview

The `/metrics` endpoint processes GNSS planning requests by computing satellite visibility and Dilution of Precision (DOP) metrics. It aggregates data from various sources, including DEM data, almanac information, and constellation details (fetched from the Data Integrator microservice), and processes each receiver via a computation pipeline. For differential GNSS applications, common visibility and DOP metrics between a base and rover receivers are computed.

2 Endpoint Details

URL

`/metrics`

HTTP Method

POST

Content Type

application/json

3 Request Format

The request payload is a JSON object with the following fields:

- **start_datetime** (string): ISO 8601 formatted start time (e.g., `"2025-03-01T08:00:00Z"`).
- **duration_hours** (number): Duration (in hours) for which metrics are computed.
- **dem** (object): DEM selection details containing:
  - `type` (string): DEM type (e.g., `"SRTMGL1"`).
  - `source` (string): DEM source key (e.g., `"ot"`).
- **application** (string): The application type (e.g., `"single"` or `"differential_gnss"`).
- **constellations** (array of strings): List of satellite constellations to consider (e.g., `["GPS", "GALILEO"]`).
- **receivers** (array of objects): List of receivers with the following structure:
  - `id` (string): Unique identifier.

- `role` (string, optional): Role (e.g., `"base"` or `"rover"`).
- `coordinates` (object):
  - `latitude` (number)
  - `longitude` (number)
  - `height` (number)
- `obstacles` (array): (Optional) List of obstacles; each must include:
  - `vertices` (array of objects with `latitude` and `longitude`)
  - `height` (number)

Example Request

```
{
  "start_datetime": "2025-03-01T08:00:00Z",
  "duration_hours": 2,
  "dem": {
    "type": "SRTMGL1",
    "source": "ot"
  },
  "application": "differential_gnss",
  "constellations": ["GPS", "GALILEO"],
  "receivers": [
    {
      "id": "receiver_1",
      "role": "base",
      "coordinates": {
        "latitude": 40.712776,
        "longitude": -74.005974,
        "height": 15.0
      },
      "obstacles": [
        {
          "vertices": [
            {"latitude": 40.713000, "longitude": -74.005500},
            {"latitude": 40.713200, "longitude": -74.005700},
            {"latitude": 40.713100, "longitude": -74.005900}
          ],
          "height": 12
        }
      ]
    },
    {
      "id": "receiver_2",
      "role": "rover",
      "coordinates": {
        "latitude": 40.722776,
        "longitude": -74.015974,
        "height": 18.0
      },
      "obstacles": []
    }
  ]
}
```

4 Response Format

On success, the endpoint returns a JSON object with computed metrics for each receiver.

Success Response (200 OK)

- **status** (string): `"success"`.
- **request_id** (string): A unique identifier for the request.
- **planning_details** (object): Contains:
  - `start_datetime` (string): ISO 8601 formatted start time.
  - `duration_hours` (number): Duration in hours.
  - `interval_minutes` (number): Fixed at 30 minutes.
  - `application` (string): Application type.
- **receivers** (array): List of receiver objects with computed metrics. Each receiver includes:
  - `id` (string)
  - `role` (string, if applicable)
  - `coordinates` (object): As provided in the request.
  - `visibility` (object): Satellite visibility counts per constellation at each time step.
  - `dop` (object): DOP metrics containing:
    - `time` (array of strings)
    - `gdop`, `pdop`, `hdop`, `vdop` (arrays of numbers)
  - `common_visibility` (object): For rover receivers in differential GNSS, the common visibility counts computed using the base receiver.
  - `common_dop` (object): For rover receivers in differential GNSS, the common DOP metrics.

Example Success Response


```
{
  "status": "success",
  "request_id": "e4d1f5a2-9b2a-4c3e-8d7f-123456789abc",
  "planning_details": {
    "start_datetime": "2025-03-01T08:00:00Z",
    "duration_hours": 2,
    "interval_minutes": 30,
    "application": "differential_gnss"
  },
  "receivers": [
    {
      "id": "receiver_1",
      "role": "base",
      "coordinates": {
        "latitude": 40.712776,
        "longitude": -74.005974,
        "height": 15.0
      },
      "visibility": {
        "GPS": {
```

```
        "satellite_count": [
          {"time": "2025-03-01T08:00:00Z", "count": 5},
          {"time": "2025-03-01T08:30:00Z", "count": 6}
        ]
      },
      "GALILEO": {
        "satellite_count": [
          {"time": "2025-03-01T08:00:00Z", "count": 3},
          {"time": "2025-03-01T08:30:00Z", "count": 4}
        ]
      }
    },
    "dop": {
      "time": ["2025-03-01T08:00:00Z", "2025-03-01T08:30:00Z"],
      "gdop": [3.5, 3.6],
      "pdop": [2.8, 2.9],
      "hdop": [1.5, 1.6],
      "vdop": [2.3, 2.4]
    },
    "skyplot_data": {
      "satellites": [
        {
          "constellation": "GPS",
          "satellite_id": "G01",
          "trajectory": [
            {"time": "2025-03-01T08:00:00Z", "azimuth": 45.0, "
               elevation": 30.0, "visible": true},
            {"time": "2025-03-01T08:30:00Z", "azimuth": 47.0, "
               elevation": 32.0, "visible": true}
          ]
        },
        {
          "constellation": "GALILEO",
          "satellite_id": "E02",
          "trajectory": [
            {"time": "2025-03-01T08:00:00Z", "azimuth": 120.0, "
               elevation": 10.0, "visible": false},
            {"time": "2025-03-01T08:30:00Z", "azimuth": 122.0, "
               elevation": 12.0, "visible": true}
          ]
        }
      ]
    }
  },
  {
    "id": "receiver_2",
    "role": "rover",
    "coordinates": {
      "latitude": 40.722776,
      "longitude": -74.015974,
      "height": 18.0
    },
    "visibility": {
      "GPS": {
        "satellite_count": [
          {"time": "2025-03-01T08:00:00Z", "count": 4},
          {"time": "2025-03-01T08:30:00Z", "count": 5}
        ]
```

```
    },
    "GALILEO": {
      "satellite_count": [
        {"time": "2025-03-01T08:00:00Z", "count": 2},
        {"time": "2025-03-01T08:30:00Z", "count": 3}
      ]
    }
  },
  "dop": {
    "time": ["2025-03-01T08:00:00Z", "2025-03-01T08:30:00Z"],
    "gdop": [4.0, 3.9],
    "pdop": [3.2, 3.1],
    "hdop": [1.8, 1.7],
    "vdop": [2.6, 2.5]
  },
  "common_visibility": {
    "GPS": {
      "satellite_count": [
        {"time": "2025-03-01T08:00:00Z", "count": 3},
        {"time": "2025-03-01T08:30:00Z", "count": 4}
      ]
    },
    "GALILEO": {
      "satellite_count": [
        {"time": "2025-03-01T08:00:00Z", "count": 2},
        {"time": "2025-03-01T08:30:00Z", "count": 3}
      ]
    }
  },
  "common_dop": {
    "time": ["2025-03-01T08:00:00Z", "2025-03-01T08:30:00Z"],
    "gdop": [3.8, 3.7],
    "pdop": [3.0, 2.9],
    "hdop": [1.6, 1.5],
    "vdop": [2.4, 2.3]
  },
  "skyplot_data": {
    "satellites": [
      {
        "constellation": "GPS",
        "satellite_id": "G01",
        "trajectory": [
          {"time": "2025-03-01T08:00:00Z", "azimuth": 45.0, "
             elevation": 30.0, "visible": true},
          {"time": "2025-03-01T08:30:00Z", "azimuth": 47.0, "
             elevation": 32.0, "visible": true}
        ]
      },
      {
        "constellation": "GALILEO",
        "satellite_id": "E02",
        "trajectory": [
          {"time": "2025-03-01T08:00:00Z", "azimuth": 120.0, "
             elevation": 10.0, "visible": false},
          {"time": "2025-03-01T08:30:00Z", "azimuth": 122.0, "
             elevation": 12.0, "visible": true}
        ]
      }
```

```
        ]
      }
    }
  ]
}
```

Error Response

- **500 Internal Computation Error:** In case of an unexpected error, the response will contain:

Example Error Response:

```
{
  "error": "Internal computation error"
}
```

5 Edge Cases and Error Handling

- Missing or invalid fields in the request payload will result in an error.
- Unexpected processing errors return a 500 Internal Computation Error.

6 Notes for Developers

- The endpoint retrieves auxiliary data (almanac and available constellations) from the Data Integrator microservice.
- DEM data is fetched per receiver using the `/dem` endpoint.
- Each receiver is processed individually through a computation pipeline that returns a payload with satellite visibility (`raw_visible`) which is then used to compute DOP metrics.
- For differential GNSS applications, common visibility and DOP metrics are computed between a base and rover receivers.
- The internal `raw_visible` field is removed before sending the final response.
- All timestamps are in ISO 8601 UTC format.