PROGRAMME OF THE EUROPEAN UNION

Copernicus
Europe's eyes on Earth

Implemented by
European Commission

Emergency Management

#EUSpace

Accessing GFM using STAC

4th CEMS Global Flood Meeting 2025

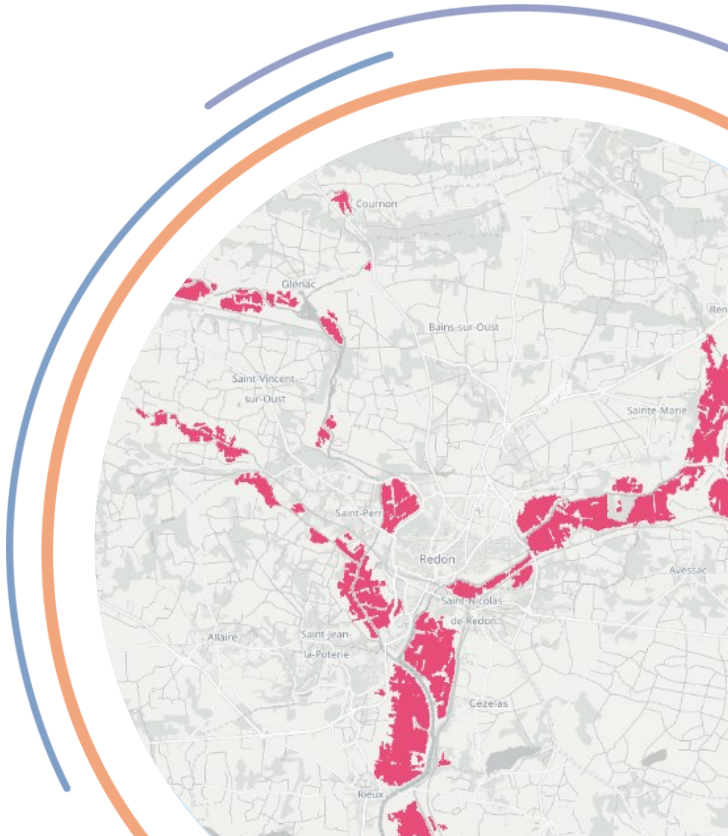
Tobias Stachl, EODC & the GFM consortium

#EUSpace

Emergency Management

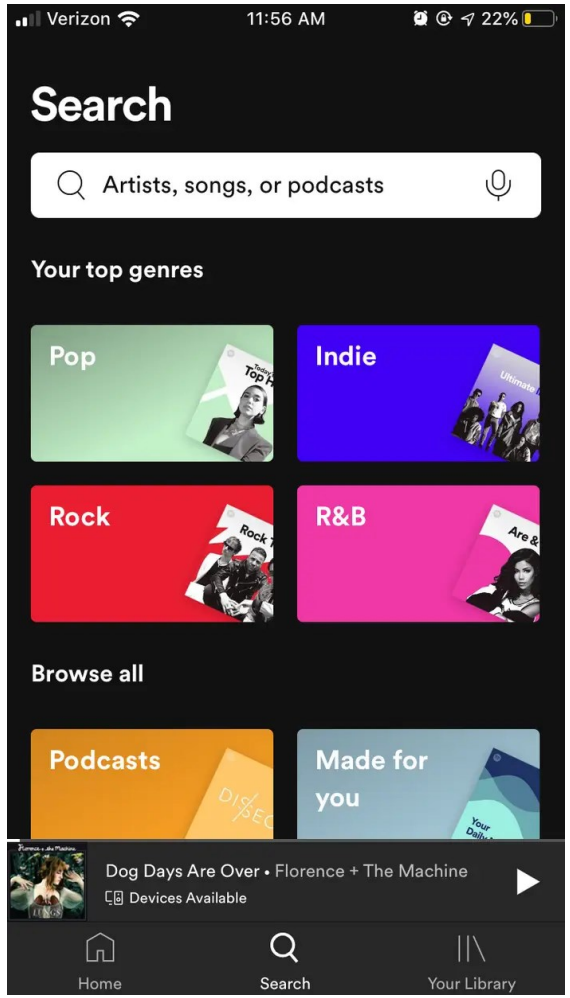
Data Access Points

- Access through GloFAS & EFAS
- GFM UI for the download of all available datasets
- WMS-T service freely accessible for integration in preferred GIS environment
- API service for expert users
- SpatioTemporal Asset Catalog (STAC)



Metadata – What, Why, How?

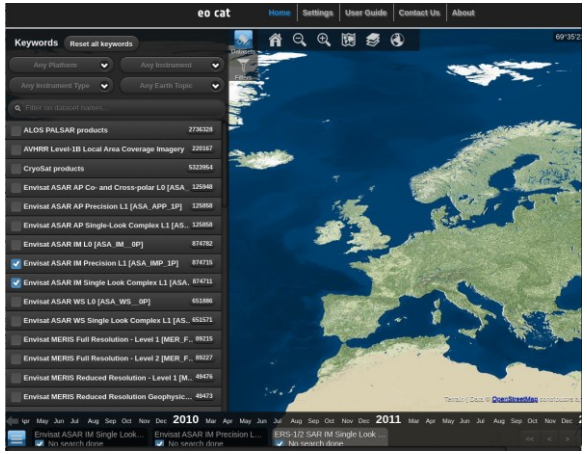
- Metadata is information about data.
- Metadata makes finding data and discovering resources easier
- Metadata standards provide a common way to structure and describe data

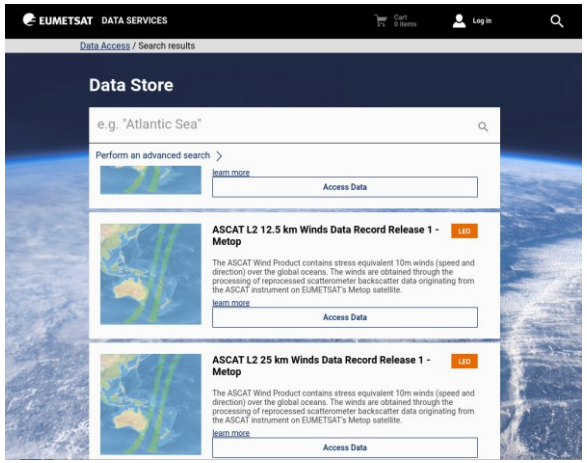


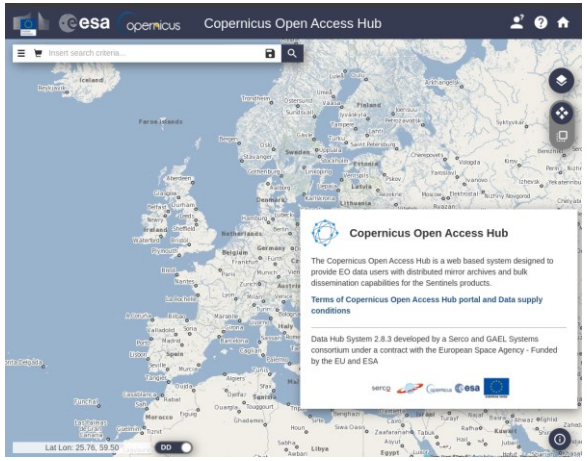
Introducing STAC

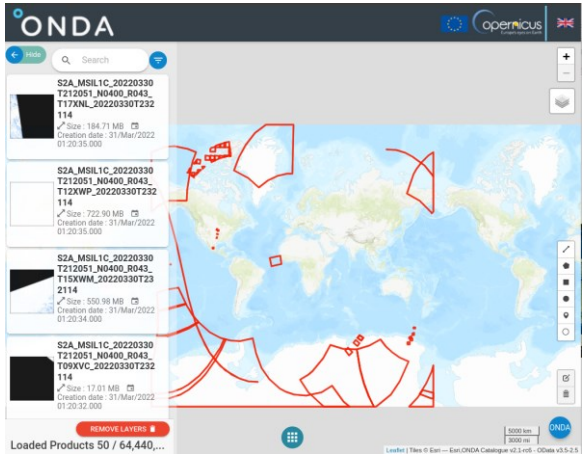
STAC
SpatioTemporal
Asset Catalog

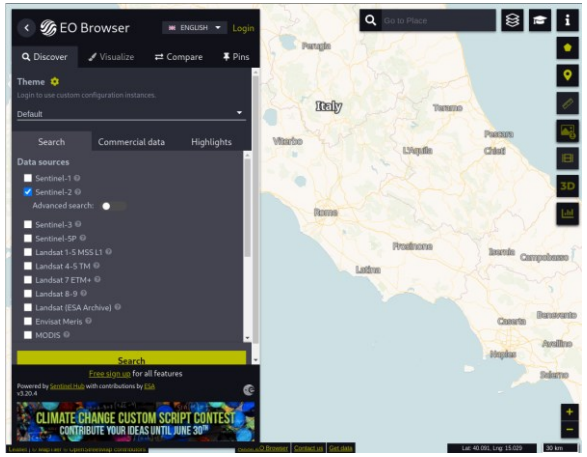
STAC Motivation





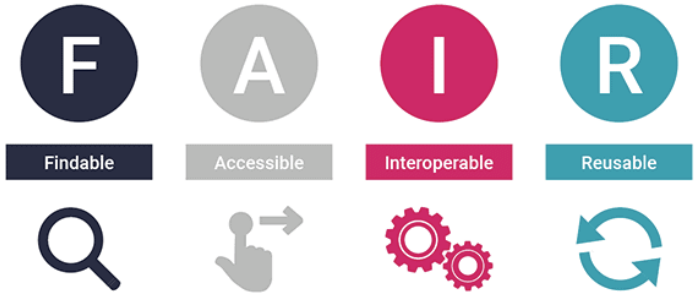






STAC Motivation

- Users should be able to quickly and easily find the desired data
- Data should be easily accessible to users
- Data formats for metadata and data should be self-explanatory and easy to understand

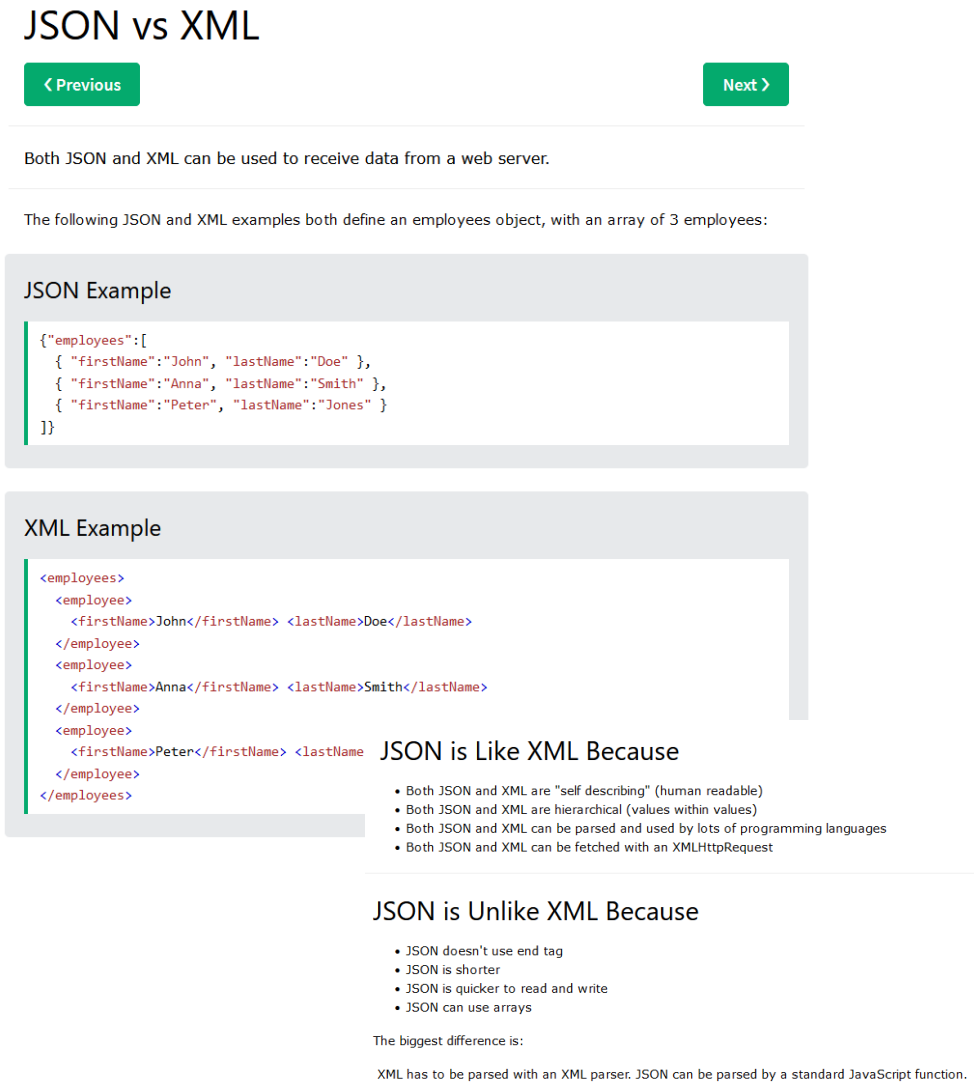


STAC Motivation

- **Simplicity**:
  STAC uses JSON for straightforward metadata management.

- **Web Compatibility**:
  By adopting JSON, STAC aligns with web standards, promoting interoperability.

- **Client Support**:
  Supported by a variety of client libraries.



JSON vs XML

‹ Previous Next ›

Both JSON and XML can be used to receive data from a web server.

The following JSON and XML examples both define an employees object, with an array of 3 employees:

JSON Example

```
{"employees":[
  { "firstName":"John", "lastName":"Doe" },
  { "firstName":"Anna", "lastName":"Smith" },
  { "firstName":"Peter", "lastName":"Jones" }
]}
```

XML Example

```
<employees>
  <employee>
    <firstName>John</firstName> <lastName>Doe</lastName>
  </employee>
  <employee>
    <firstName>Anna</firstName> <lastName>Smith</lastName>
  </employee>
  <employee>
    <firstName>Peter</firstName> <lastName
  </employee>
</employees>
```

JSON is Like XML Because

- Both JSON and XML are "self describing" (human readable)
- Both JSON and XML are hierarchical (values within values)
- Both JSON and XML can be parsed and used by lots of programming languages
- Both JSON and XML can be fetched with an XMLHttpRequest

JSON is Unlike XML Because

- JSON doesn't use end tag
- JSON is shorter
- JSON is quicker to read and write
- JSON can use arrays

The biggest difference is:

XML has to be parsed with an XML parser. JSON can be parsed by a standard JavaScript function.

#EUSpace Emergency Management

What is STAC?

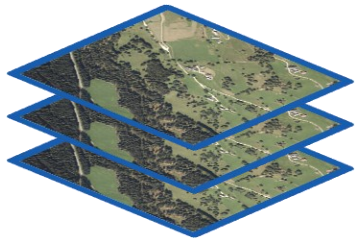
STAC
SpatioTemporal
Asset Catalog

- STAC = SpatioTemporal Asset Catalog
- STAC is a **specification** to describe geospatial information using JSON
- The focus of STAC is **search** and **discovery**
- The focus of STAC is on **raster/array data**
- The philosophy of STAC is to keep it **simple**, yet **flexible** and **extensible**
- STAC is **not a definitive metadata standard** like ISO or OGC
- Search geospatial data sets like GFM by **space**, **time**, **and more**.
- **Data Access** is not covered by STAC
  - Assets may contain links to the actual data

#EUSpace

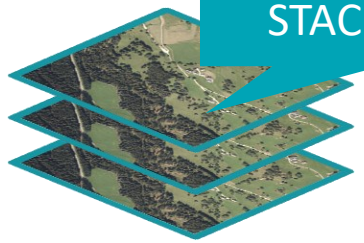
Emergency Management

STAC Catalog

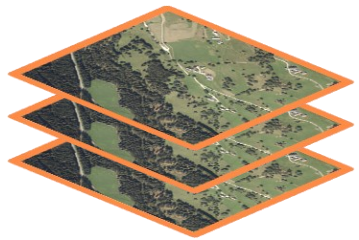


Sentinel-1 GRD

STAC Collection



GFM



Sentinel-2 L1C

STAC Collection

STAC Item

STAC Assets

ensemble_flood.tif

STAC Asset

ensemble_likelihood.tif

reference_water_mask.tif

Data Repository Storage

Who can do what with STAC?

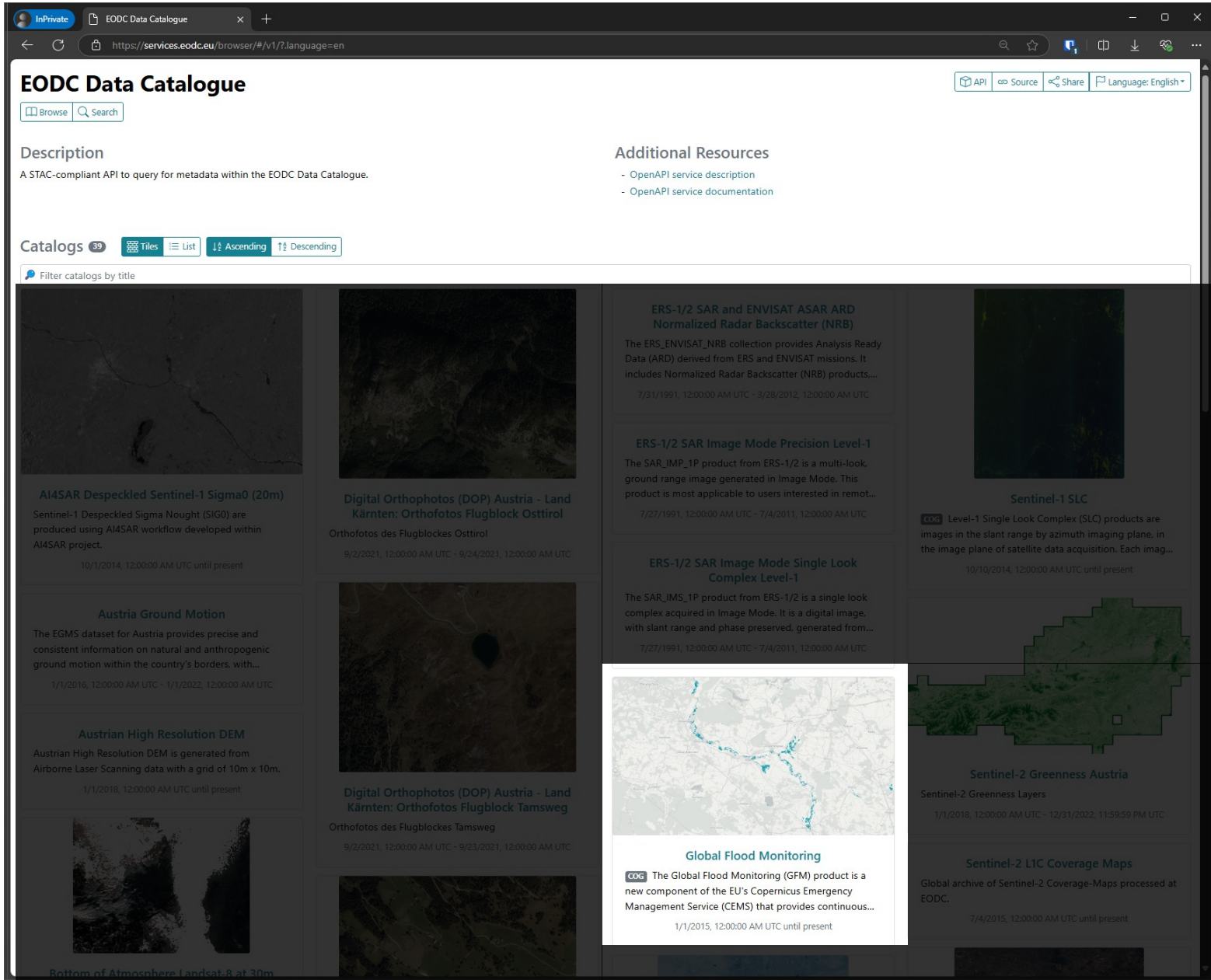
- **Data Providers**
  - Standardised way to expose collections of spatial temporal data
  - Can be as simple as just putting JSON files on the cloud
- **Developers**
  - Can make use of a growing set of tools to work with STAC
  - For interacting in different languages (Python, R, …)
  - For hosting (STAC Server, stac-fastapi, …)
  - As clients (odc-stac, pystac-client, STAC Browser, rstac, …)
  - For validating (STAC Validator, STAC Lint, …)
- **Data Users**
  - Reduce burden of finding geospatial data
  - Browse STAC catalogs on the web
  - Explore satellite imagery with the QGIS STAC API Browser plugin
  - Download data

#EUSpace

Emergency Management

Data Browsing

- STAC Browser
  - **browser.stac.dataspace.copernicus.eu**
  - **ewds.climate.copernicus.eu/stac-browser/**
  - **services.eodc.eu/browser**
  - **radiantearth.github.io/stac-browser**



#EUSpace

Emergency Management

STAC API + Python

Utilize well-known **Python** libraries to **find** and **use** the data you **need**

```python
filt = {
    "op": "gt",
    "args": [
        {"property": "ratio_after_blob_removal"}, 0.9
    ]
}

search = eodc_catalog.search(collections="GFM", filter=filt)
```

```python
# Python imports
from shapely.geometry import box
from pystac_client import Client

# Define asset name to use
asset_name = "ensemble_flood_extent"

# Define bounding box
aoi = box(16.77, 49.91, 18.62, 51.25)

# Define time range
time_range = (datetime(2024, 9, 18), datetime(2024, 9, 28))

# EODC STAC API URL
api_url = "https://stac.eodc.eu/api/v1"
eodc_catalog = Client.open(api_url)

# Define search query using pystac_client
search = eodc_catalog.search(
    max_items=1000,
    collections="GFM",
    intersects=aoi,
    datetime=time_range
)

# Get STAC items
items = search.item_collection()

print("We found", len(items), "items, that match your filter criteria.")

# Prints:
# We found 30 items, that match your filter criteria
```

#EUSpace

Emergency Management

GFM STAC Collection

Full GFM NRT + archive accessible via STAC

Fair-use data policy

Basic training materials such as Jupyter notebooks

Cloud Optimized GeoTiff

Stay tuned on EFAS and GFM channels for news and updates

#EUSpace

Emergency Management

Thank you!

Learn more in the interactive session!