

Using Neutrino data

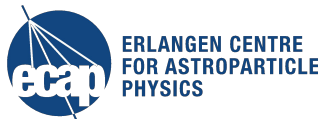
A KM3NeT perspective

J. Schnabel, T. Gal, D. Dornic

ESCAPE WP4 Technology Forum 2

13th April 2021

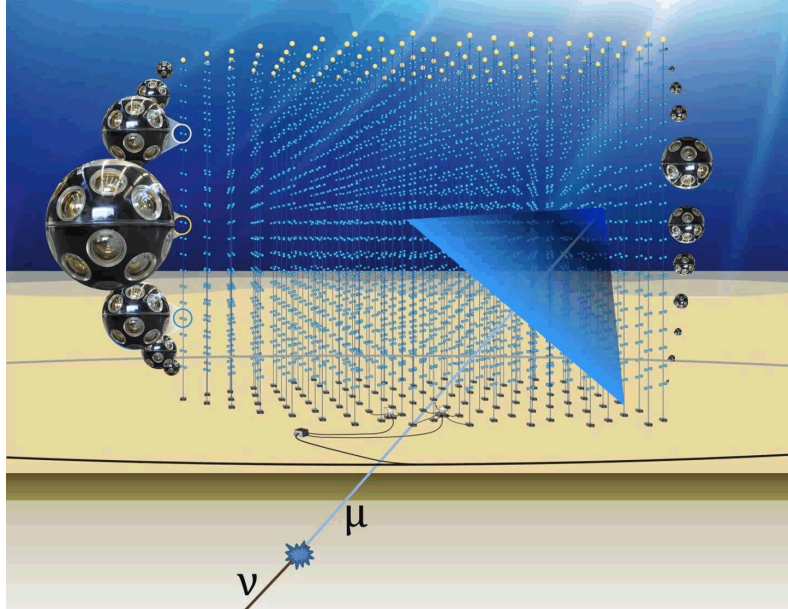
ecap



Making our data “FAIR” - with or without the VO?

- “neutrino data” from a KM3NeT perspective - current formats and content
 - with the data comes the service
- how to get it - interfaces at the development stage
- unify and standardize - but how?
 - approaches with gamma astronomy
 - for integration in machine learning

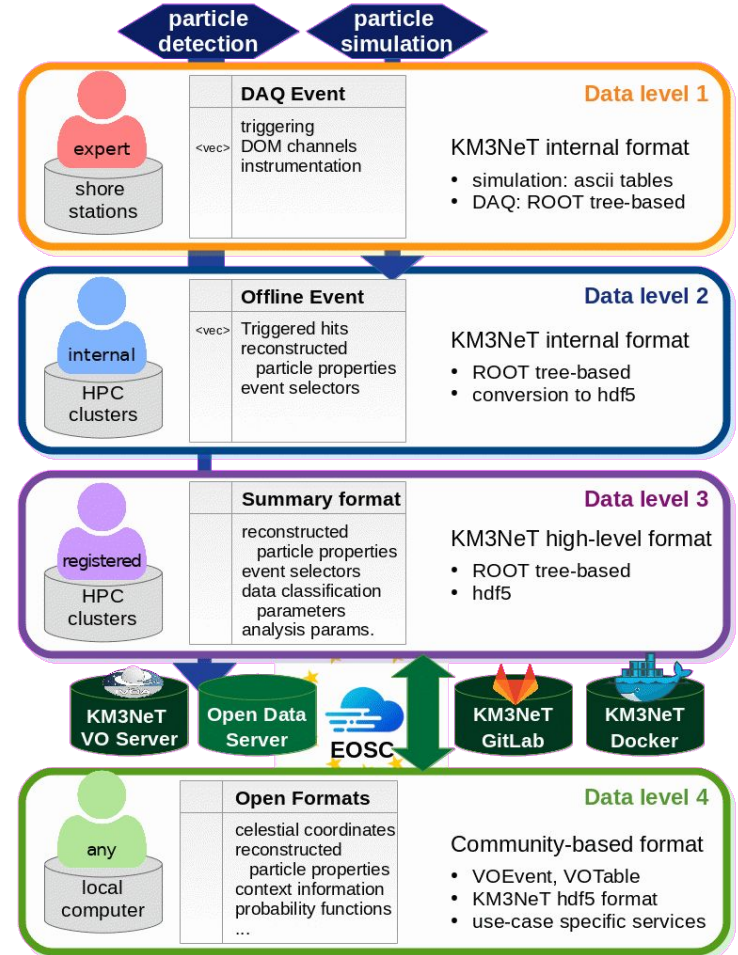
KM3NeT data landscape



relevant information for data sharing:

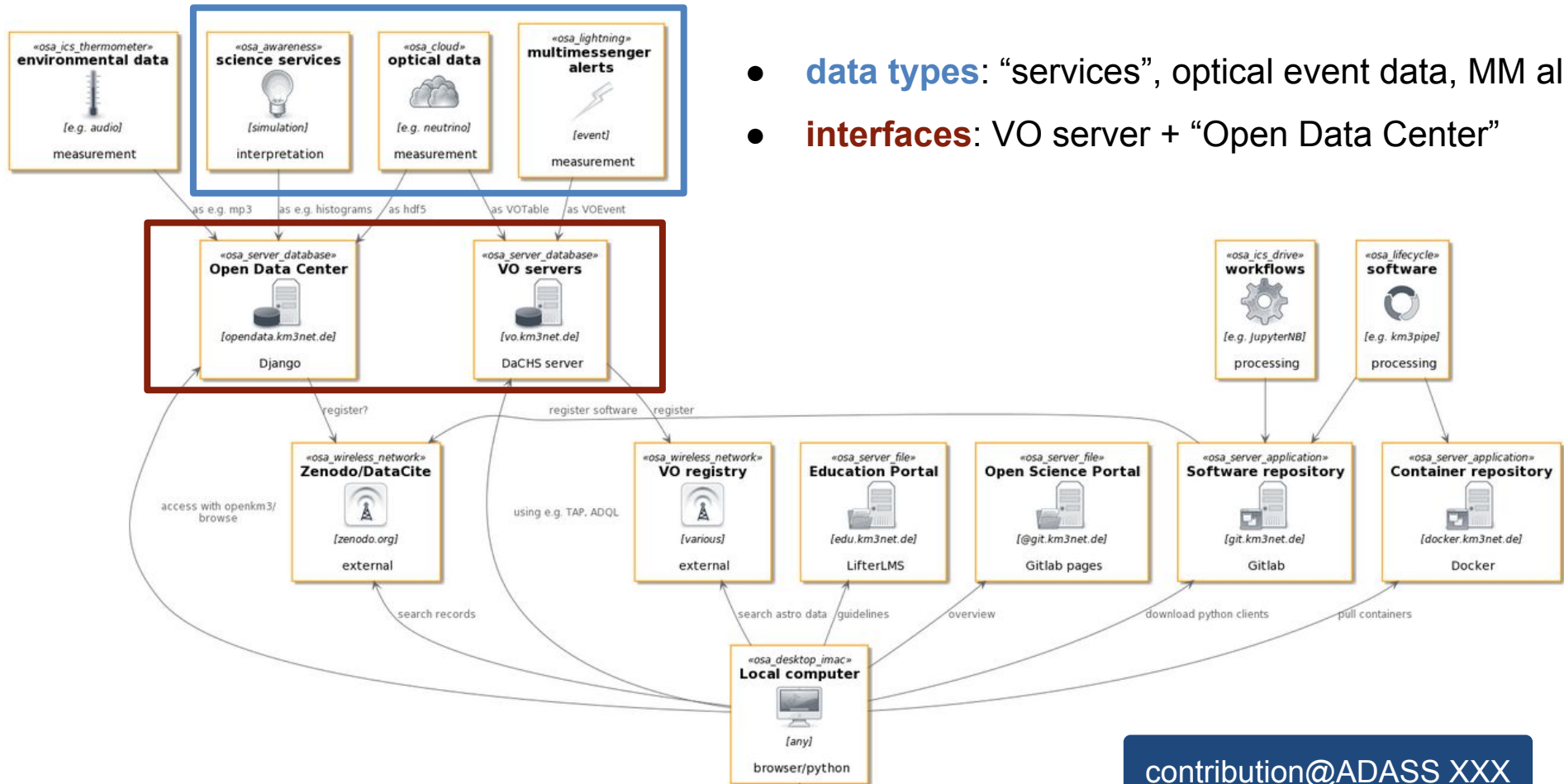
Data level 4: event info: time, direction, energy, properties

Data level 3: photon hits distribution (t, x, amplitude)



Open Science @ KM3NeT

- **data types:** “services”, optical event data, MM alerts
- **interfaces:** VO server + “Open Data Center”

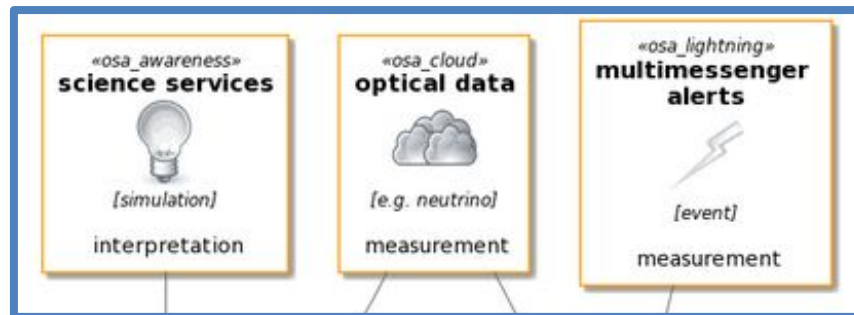


[contribution@ADASS XXX](mailto:contribution@ADASS.XXX)
The KM3NeT open science system

Data content and types

Science services

MC-derived, individual to scientific purpose/data sets: “plots”



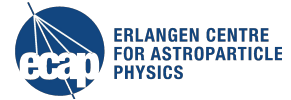
Optical data

Metadata type	content
<i>Provenance</i> information	list of processing steps (referenced by identifier)
<i>Parameter</i> description	parameter name, unit (SI), type, description, identifier
<i>Data taking</i> metadata	start/stoptime, detector, event selection info
<i>Publication</i> metadata	publisher, owner, creation date, version, description

Multimessenger alerts

(Meta)data type	content
Event identification	event identifier, detector
Event description	type of triggers, IsRealAlert
Event coordinates	time, rightascension, declination, longitude, latitude
Event properties	flavor, multiplicity, energy, neutrino type, error box 50%, 90% (TOC), reconstruction quality, probability to be neutrino, probability for astrophysical origin, ranking
Publication metadata	publisher, contact

Common description - metadata



- defined in the [datamodels project](#) (working model)
- **KM3Resource** provides
 - external metadata - made to fit e.g. Zenodo and DataCite
 - links and resource description
- can include VO resources, data from the server, external URLs, Info from Git repositories ...
- KM3Resources can be combined to analysis package, e.g. containing plots, data, paper etc.
- kmeta can be used to make product “server ready” by adding metadata files e.g. to the repository (e.g. integrated workflow with km3pipe)

```
KM3Resource: # basic class to describe a KM3NeT open resource
title: string # descriptive title, required
description: string # could also be an abstract
publisher: # required, should always be the collaboration for official
author: ORCID # becomes creator in VO
datePublication: date # required
dateCreation: date # optional
license: string # required

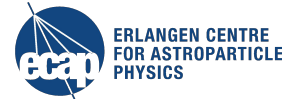
kid: UUID # km3net internal identifier, required
idIVOA: URI # IVOA resource identifier, optional
idDOI: URI # DOI as assigned by DataCite/Zenodo, optional
idALT: string # additional identifier, optional
resourceType: ResourceType # enumerator to identify resource
version: string # version

keywords: list # list of strings according to keyword catalogue, optional
referenceURL: URI # reference to document for further description
parentResource: string # kid for parent resource, optional
relatedResources: list # kids for further related resources, siblings

source: # linking to the actual data, metadata for interoperability
accessURL: URI # required
accessFormat: string # MIME type of the return object, required
filesize: string # estimated size of return object, optional
typeReference: URI # optional, link to description of the content

content: # optional parameters dependent on type
processingLevel: int # optional
startTime: datetime # optional, validity / measurement start time
stopTime: datetime # optional, validity / measurement stop time
instrument: string # identifier for the detector which generated
```

Common description - metadata



- defined in the [datamodels project](#) (working model)
- **KM3Resource** provides
 - external metadata - made to fit e.g. Zenodo and DataCite
 - links and resource description
- can include VO resources, data from the server, external URLs, Info from Git repositories ...
- KM3Resources can be combined to analysis package, e.g. containing plots, data, paper etc.
- kmeta can be used to make product “server ready” by adding metadata files e.g. to the repository (e.g. integrated workflow with km3pipe)

```
KM3Resource: # basic class to describe a KM3NeT open resource
title: string # descriptive title, required
description: string # could also be an abstract
publisher: # required, should always be the collaboration for official publications
author: ORCID # becomes creator in VO
datePublication: date # required
dateCreation: date # optional
license: string # required

kid: UUID # km3net internal identifier, required
idIVOA: URI # IVOA resource identifier, optional
idDOI: URI # DOI as assigned by DataCite/Zenodo, optional
idALT: string # additional identifier, optional
resourceType: ResourceType # enumerator to identify resource
version: string # version

keywords: list # list of strings according to keyword catalogue, optional
referenceURL: string # URL for further description
parentResource: URI # parent resource, optional
relatedResources: list # other related resources, siblings

source: # linking to the actual data, metadata for interoperability
accessURL: URI # required
accessFormat: string # MIME type of the return object, required
filesize: string # estimated size of return object, optional
typeReference: URI # optional, link to description of the content type

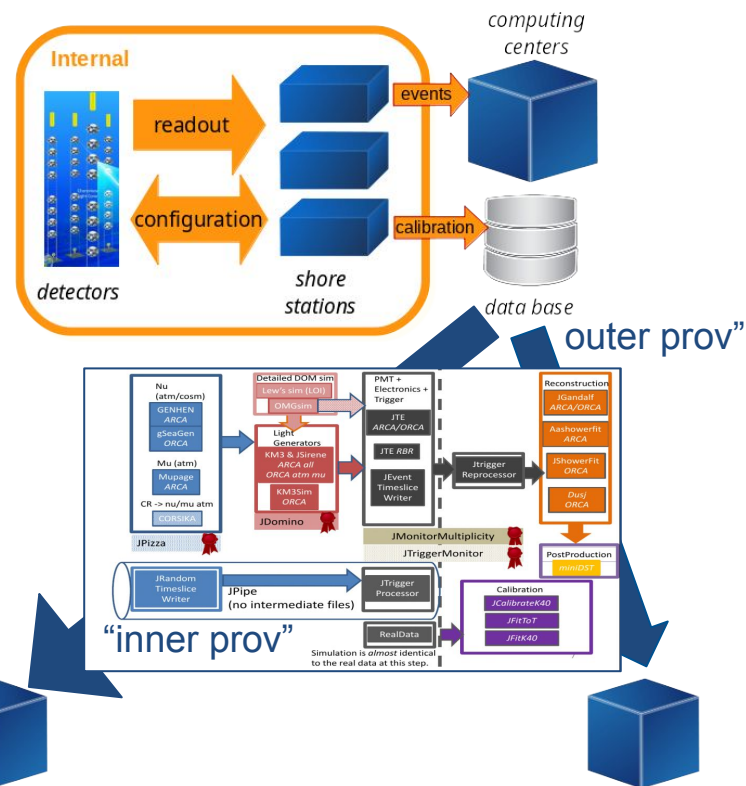
content: # optional parameters dependent on type
processingLevel: int # optional
startTime: datetime # optional, validity / measurement start time
stopTime: datetime # optional, validity / measurement stop time
instrument: string # identifier for the detector which generated the data
```

content type

Aside: Questions on provenance

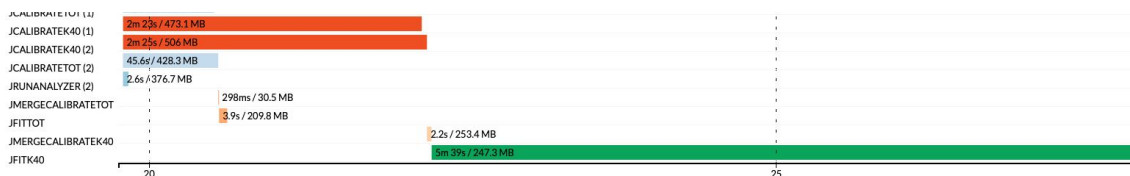
- How do we capture provenance information along the processing chain?
 - main purpose: **reproducibility** and **documentation for us!**
- How to convey the history of various data levels and multiple data reduction steps in **a short line?**

→ Not only “outer” black-box provenance but also “inner” gray-box provenance needed

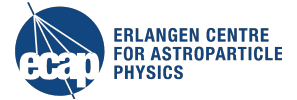


Working point: workflow management

- **Workflow management** for reproducibility and provenance generation
- Standardising the way data is processed and produced (especially within the collaboration), mostly related to Tier-0 and Tier-1 but also applicable to Tier-2
- Two workflow management solutions under investigation: Nextflow and CWL
 - **Nextflow**: all-in-one solution (workflow description, executor, reporter, provenance tracker). First prototype implementations and test runs already performed.
 - **CWL**: abstract description of tools and workflows, many different tools which understand how to execute, report and track the provenance. No suitable executor found yet which fits most of our basic needs (need further investigations and likely the development of such a tool)
- Provenance of low-level processing provided by the workflow management will be linked to the high-level provenance **data via UUID**



Provenance data at high level



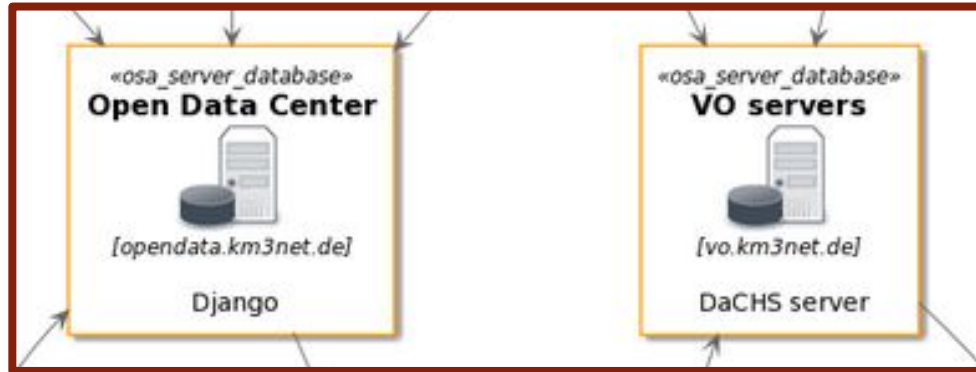
Information is stored directly in final file as “header” table, or attributes to the file or table - no querying to database for “high level” necessary!

```
table = ks.get("one_week_orca")
table.get_origin()
```

```
{'provenance': [{'refID': '5749ebad-f620-11ea-868d-a4c3f052a927',
  'name': 'JConvertEvt',
  'settings': None,
  'software': None},
{'refID': '5749ebae-f620-11ea-868d-a4c3f052a927',
  'name': 'JEnergy',
  'settings': None,
  'software': None},
{'refID': '5749ebaf-f620-11ea-868d-a4c3f052a927',
  'name': 'JStart',
  'settings': None,
  'software': None}],
```

**provenance summary
pointing to full info (kid)**

Interfaces: two servers, one goal



+ MM alert
pipeline

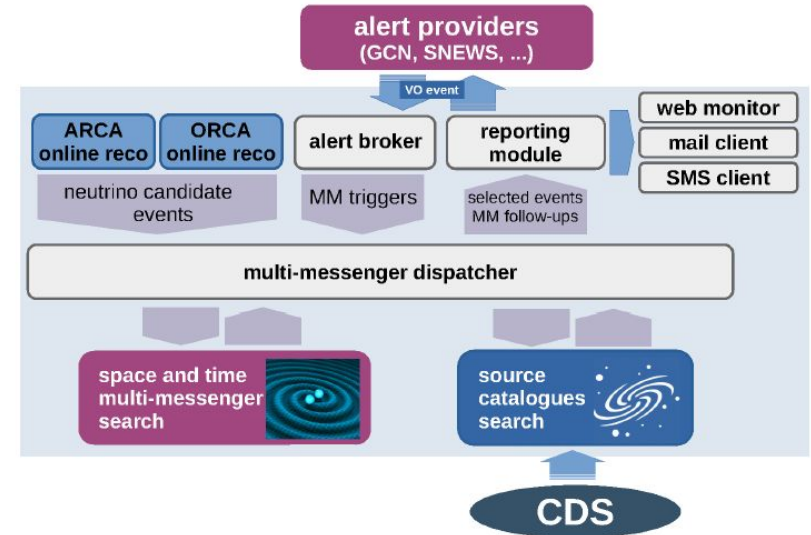
non-VO data & interfaces

VO interfaces



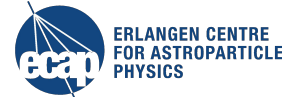
Status of the MM developments

- Important effort on setting an efficient real-time analysis framework:
 - KM3NeT alert sending using Comet and VO events (~10-20 s)
 - Online cross-match of interesting external triggers received via the GCN, SNEWS, TNS, ZTF/LSST brokers...
- Standardize KM3NeT VO alert format (Include provenance in the VO, storage in the VO server)
- Working on the definition of an open public alert program



contribution@ADASS XXX
real-time MM program at KM3NeT

Data in VO: Virtual Observatory server



- Running server with **DaCHS** software
- Registered as data provider to the VO (ivo://km3net.org)
- Can publish data sets to the VO registry - done with Antares 2007-2017 data
- Data accessible through widely used tools in **astrophysics** (Aladin, Topcat ...)

Information on resource 'ANTARES 2007-2017'

Neutrino candidates from full-sky search from 2007-2017

Services defined within this resource descriptor

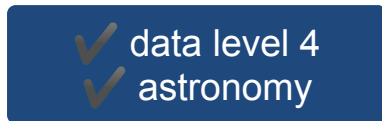
- [ANTARES 2007-2017](#)

Tables defined within this resource descriptor

- [ant20_01.main](#) – queryable through [TAP](#) and [ADQL](#)

[\[Manage RD\]](#)

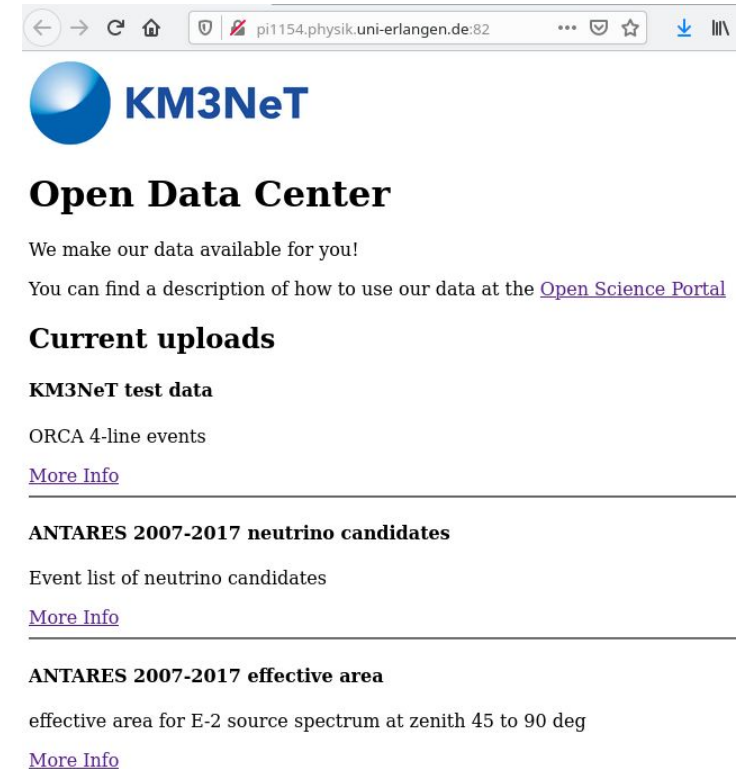
report errors and problems to the [site operators](#). Thanks.



Dist. [arcsec]	MJD	Beta	number of hits	right ascension	declination	ID
2032.96	55926.0166	0.5	24	359.5	0.3	ANT4536
3020.36	57940.5225	0.5	29	359.3	-0.1	ANT8455
5359.52	56178.5081	0.3	54	0.4	-1.2	ANT4999
6001.19	55841.6068	0.9	27	0.3	1.9	ANT4092
8913.93	55954.107	0.3	44	2.2	-1.0	ANT4680
8953.08	56340.0737	0.2	39	358.4	2.1	ANT5618
10580.47	54462.2357	0.4	35	1.1	3.0	ANT0238
11096.03	54510.0353	0.2	43	2.6	-1.5	ANT0415
11303.16	56284.2811	0.7	64	2.6	2.1	ANT5411
11708.80	55968.0641	0.4	37	1.8	3.0	ANT4734
12383.00	58059.5025	0.5	26	357.3	-1.8	ANT8630

Virtual observatory server
<http://vo.km3net.de/>

- Server based on Django to build REST API
- Can interface with data objects stored in the server with requests (returns json), but also upload files and create webpages and forms
- At the moment, list of “KM3Resources” are displayed
- additional classes are possible and planned, eg.
 - a “ktype” and “kid” finder to return info on registered data objects (kid) and object definitions (ktype)



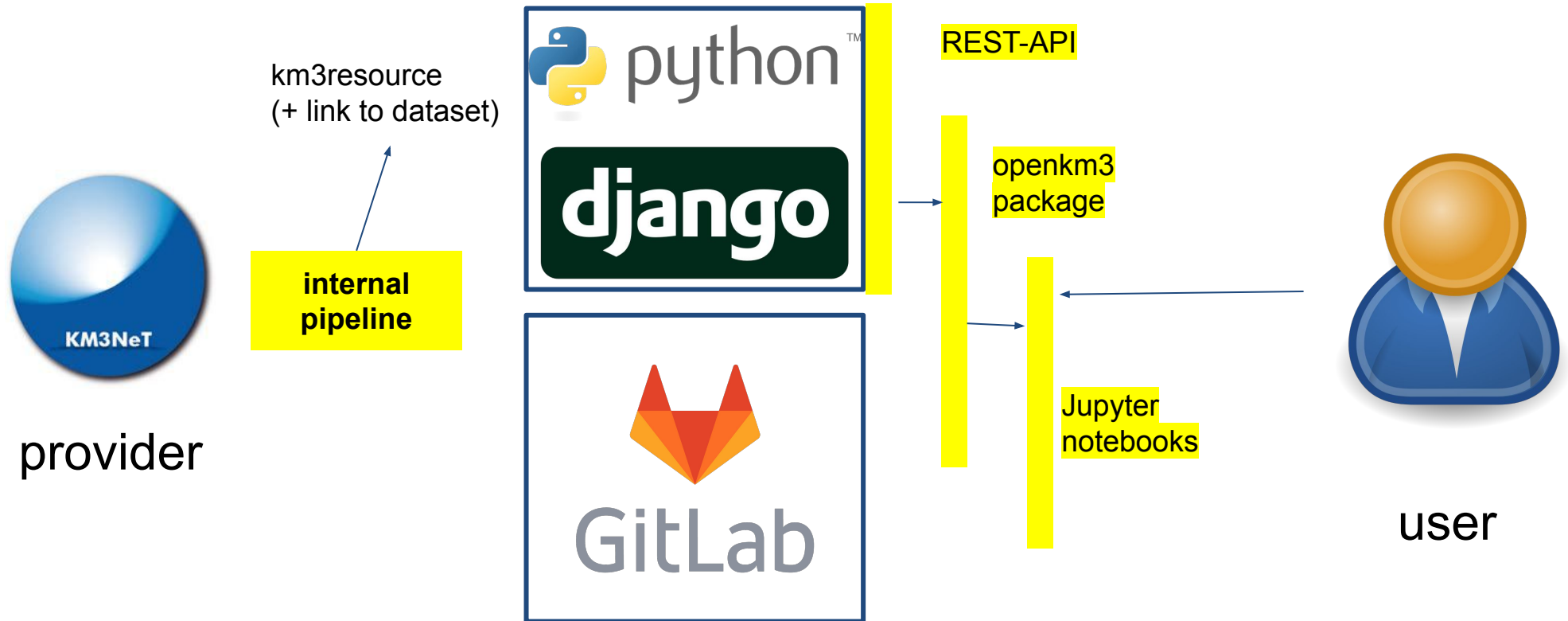
The screenshot shows a web browser window with the URL `pi1154.physik.uni-erlangen.de:82`. The page features the KM3NeT logo and the title "Open Data Center". Below the title, it states "We make our data available for you!" and provides a link to the "Open Science Portal" for more information on data usage. The page lists "Current uploads" with three entries: "KM3NeT test data" (ORCA 4-line events), "ANTARES 2007-2017 neutrino candidates" (Event list of neutrino candidates), and "ANTARES 2007-2017 effective area" (effective area for E-2 source spectrum at zenith 45 to 90 deg). Each entry includes a "More Info" link.

✓ data level 3+4
✓ astronomy + particle
phys + ...

Access: Python package + REST API



ERLANGEN CENTRE
FOR ASTROPARTICLE
PHYSICS



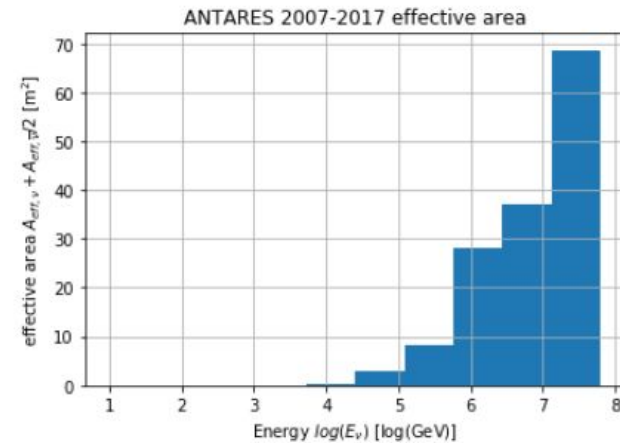
openkm3 functions

- Small python package to directly use open data in python from local computer
- Interfaces with data center API, allows to query datasets (km3resources) in the “KM3Store”
- Provides functions to download & interpret data products
- Current capability
 - Loading hdf-File as pandas DataFrame, reading additional parameter infos & provenance
 - Reading histogram data as plain table, pandas Dataframe or automatically build plot
 - uses pyvo for VO interfaces

```
table = ks.get("annotated_aeff0", loadoption = "pandas")  
table.data
```

	Log(E1/GeV)	Log(E2/GeV)	AEFF/m2
0	1.0	1.2	5.153940e-12
1	1.2	1.4	1.614220e-10

```
nice = ks.get("annotated_aeff0", loadoption="plot")
```



1220e-09
5090e-07
7530e-07
2520e-06
9460e-05

openkm3 project

<https://git.km3net.de/open-data/openkm3/>

Using hdf5 full event files

```
ks = KM3Store()
```

```
ks.print_index(include_technical=True)
```

```
one_week_orca
=====
tables: ['events', 'group_info', 'header']
header:
  author:          b'The KM3NeT collaboration'
  contact:         b'opendata@km3net.de'
  instrument:      b'ORCA'
  license:         b'Creative Commons 4.0 International'
  measurement_start: b'2019-10-03T06:00:00.544000000'
  measurement_stop: b'2019-09-17T06:00:00.202000000'
  reference:       b'http://www.km3net.org/'
url:  http://vo.km3net.de:82/storage/one_week_orca.h5
type:  application/x-hdf5
local: /home/jutta/Desktop/openkm3/examples/orca_data/.openkm3/one_week
```

```
table = ks.get("one_week_orca", ["events"], "pandas")
table.data
```

	angular_error	azimuth	dirz	energy	internalID
0	0.004341	5.108108	-0.968124	22.017775	km3net.44.61f
1	0.001181	3.358250	-0.990520	128.639694	km3net.44.61f
2	0.003534	6.146142	-0.744035	24.362326	km3net.44.61f
3	0.008247	2.954967	-0.784588	15.714084	km3net.44.61f
4	0.006967	5.618736	-0.529338	137.616933	km3net.44.61f

getting info on file content

load data as pandas dataframe

Services - using “plot data”

with KM3NeT data server:

Information is provided as annotated text files, containing extended header (can be interpreted by kmeta) and actual table (various formats possible)

```
ks.print_index()

annotated_aeff0
=====
header:
  name:           ANTARES 2007-2017 effective area
  description:    effective area for E-2 source sp
  contact:        antares.spokesperson@in2p3.fr
  instrument:     ANTARES
  license:        Creative Commons 4.0 Internatio
  reference:      https://antares.in2p3.fr/public
```

```
plain = ks.get("annotated_aeff0") # not so nice, make i
plain.data
```

```
{'Log(E1/GeV)': [1.0,
1.2,
1.4,
1.6,
1.8,
2.0,
2.2,
2.4,
2.6,
2.8,
3.0,
3.2,
3.4,
3.6,
...]}
```

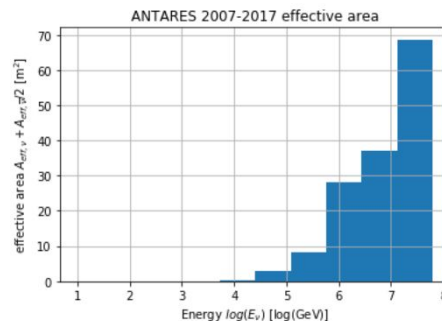
only get the table

```
table = ks.get("annotated_aeff0", loadoption = "pandas")
table.data
```

	Log(E1/GeV)	Log(E2/GeV)	AEFF/m2
0	1.0	1.2	5.153940e-12
1	1.2	1.4	1.614220e-10
2	1.4	1.6	4.891220e-09
3	1.6	1.8	1.245090e-07
4	1.8	2.0	9.157530e-07
5	2.0	2.2	4.372520e-06

get a Dataframe

```
nice = ks.get("annotated_aeff0", loadoption="plot")
```



get the plot

```
table.get_origin()
```

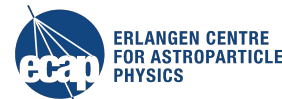
```
{'range': {'time': {'measurement_start': datetime.datetime(2007, 1, 1, 0, 0, 0), 'measurement_stop': datetime.datetime(2018, 1, 1, 0, 0, 0), 'ktype': 'tbd'}, 'zenith': {'min': -45, 'max': 0, 'ktype': 'tbd'}}}
```

```
table.get_paraminfo()
```

```
{'lower_edge': {'columnname': 'Log(E1/GeV)', 'name': 'Energy', 'description': 'logarithmic reconstructed energy of the neutrino', 'unit': 'log(GeV)', 'symbol': 'log(E_{\nu})'}, 'upper_edge': {'columnname': 'Log(E2/GeV)', 'name': 'Energy', 'description': 'logarithmic reconstructed energy of the neutrino', 'unit': 'log(GeV)', 'symbol': 'log(E_{\nu})'}}
```

get additional info

Which “services” do we need?



- highly dependent on science use case
- starting from a user perspective **inside the collaboration**
 - template git project to make plots and underlying data available
 - standardized way to provide high-level data to be derived from this
- when providing non-publication related datasets, usability will depend on the **general services** which are provided alongside the data
 - still, we need the science goal - **use case driven!**

The screenshot shows a GitHub repository page for a group named 'Templates'. The group ID is 235 and there is a 'Leave group' link. Below the group name, it says 'Template collection based on Cookiecutter.' There are three tabs: 'Subgroups and projects' (selected), 'Shared projects', and 'Archived projects'. A list of projects is shown, each with a bookmark icon, a letter in a colored box, the project name, and a shield icon:

- K** KM3NeT Analysis Template
- P** Python Project
- K** KM3NeT Public Plots Demo
Demonstration for public plots template.
- C** Code Style
Simple code style examples for a few languages used in KM3NeT.
- K** KM3NeT Public Plots
A cookiecutter template to generate KM3NeT public plots
- K** KM3NeT Document Maintainer
A template for generating KM3NeT documents from Emacs Org files.

Use Case description

The scientist can calculate the probability of neutrino observation for a given point source, using a neutrino event list (measurement) and additional services for instrument responses (simulation).

Goal

Integrate full neutrino analysis

- Neutrino data shared through the VO
- provide services to interpret the neutrino data
 - move from “homemade” solution to interoperable solution with VO?

Data & Software

- ANTARES 2007-2017 point source set ([KM3NeT VO server](#))
- Background and sensitivity estimates ([KM3NeT data center](#))
- Jupyter notebook with example analysis

ESCAPE WPs: WP4 (VO discussions) / WP3 (common data formats, just starting)

Use Case description

The scientist can find and read KM3NeT event data through ESAP and perform analyses on public data sets.

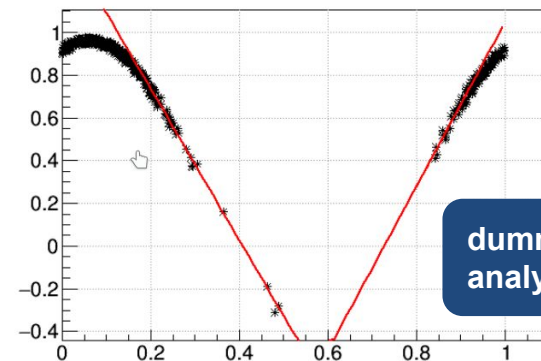
Goal

- Develop interfaces for reading of KM3NeT event data
- explore the use of the Data lake and develop KM3NeT's multisite computing environment

ESCAPE WPs: WP2 (bring data to data lake), WP3 (onboard software), WP5 (access) - **WP4?**

Data & Software

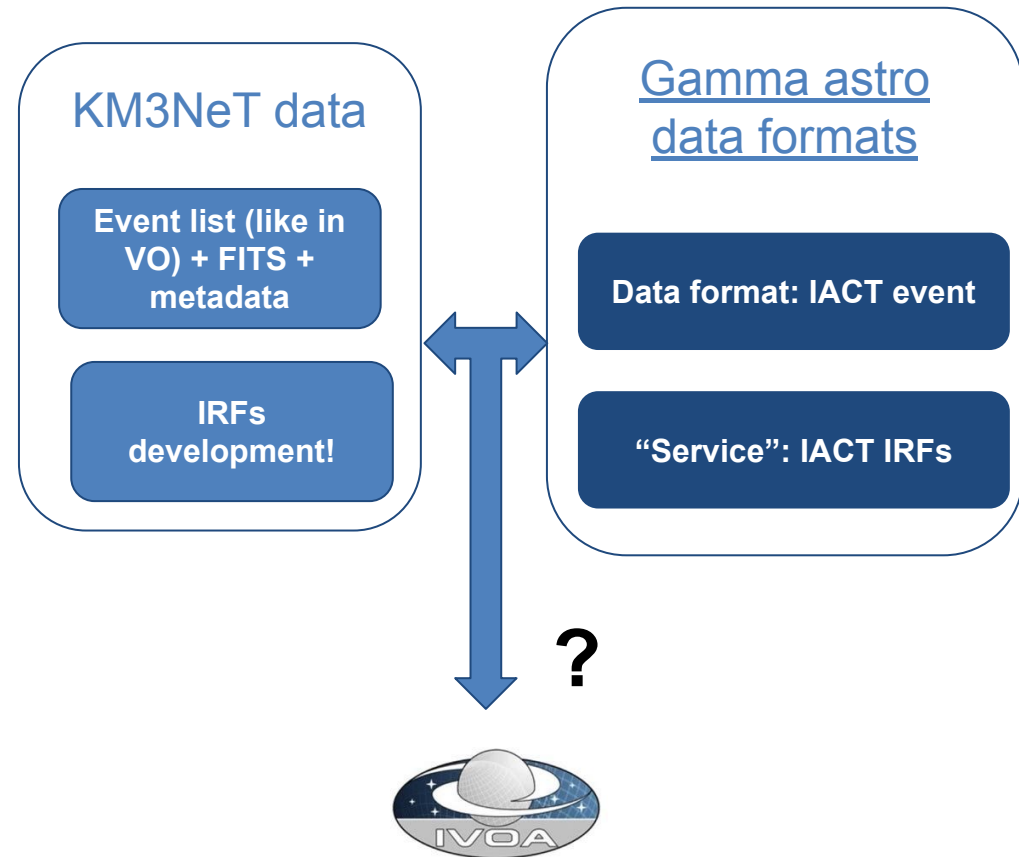
- One week ORCA4 data ([KM3NeT data center](#), full sample to be provided)
 - simple event table (h5)
 - fully reconstructed event files (root/hdf5)
- Example notebooks
- km3py package for access from KM3NeT data center



So between 9:00 and 20:00 we have only 3 events in that region which are upward reconstructed. 3 events in 2 sec: which seems low

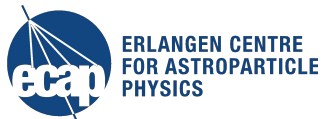
Developing standards

- Scientifically close use case: **gamma rays**
- ground-based Cherenkov telescopes already defined exchange formats - map to standards from neutrino side in common analysis
 - FITS format
 - explore IRFs for KM3NeT
 - interface - python-based
- link to VO?



- Working on integration of use cases in ESCAPE - relatively bypassing WP4 at the moment, driven by use cases based on Jupyter notebooks
- not sure how to proceed with the integration of the developments for “services” in the VO - is it a priority?
- Data level 3 (+hits) might be interesting for common ML efforts - starting discussion on this (in WP3) and towards the test science projects - probably not in the scope of VO
- stay tuned!

Thank you for your attention!





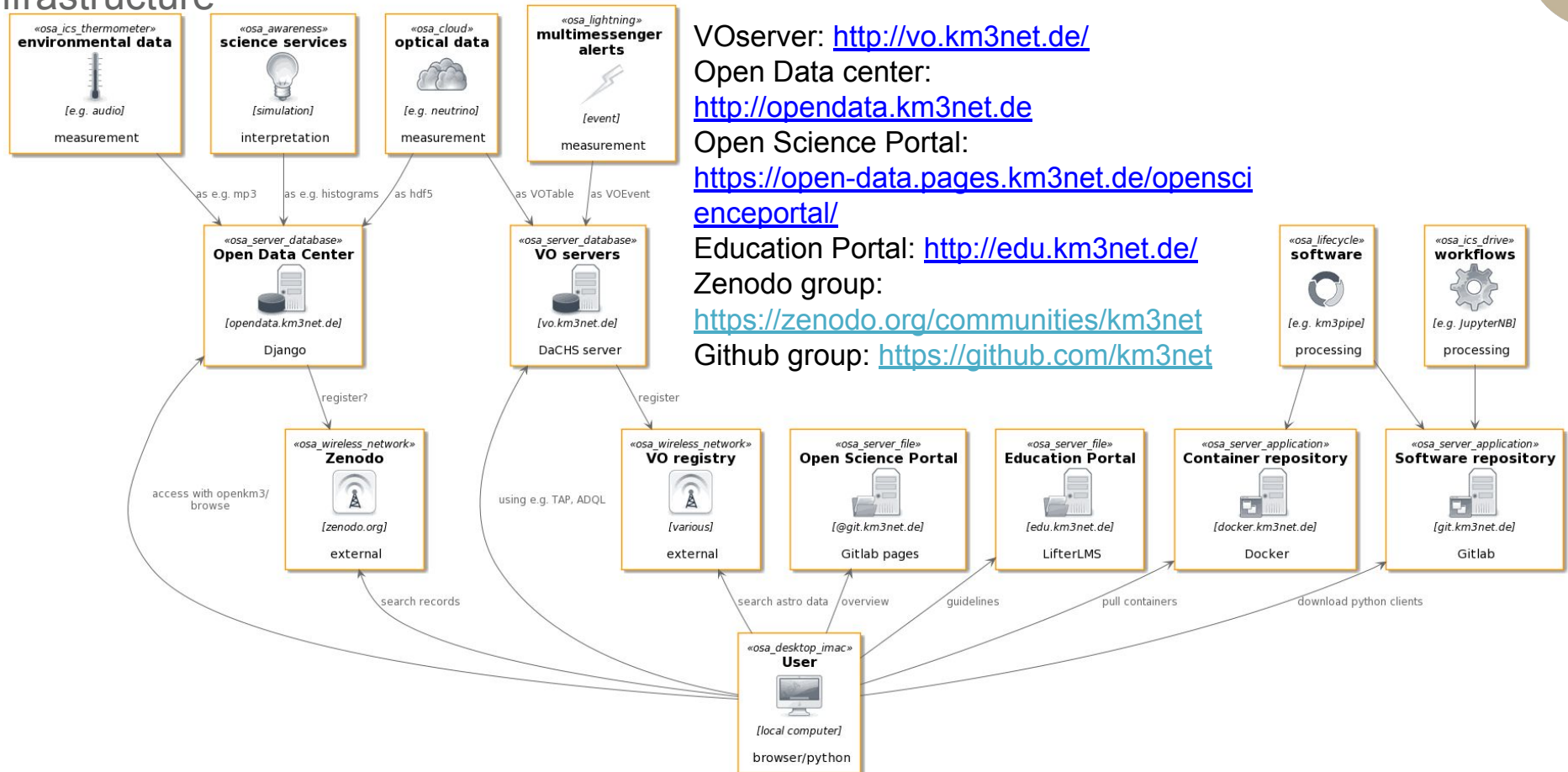
ERLANGEN
FOR ASTROPHYSICS



Infrastructure

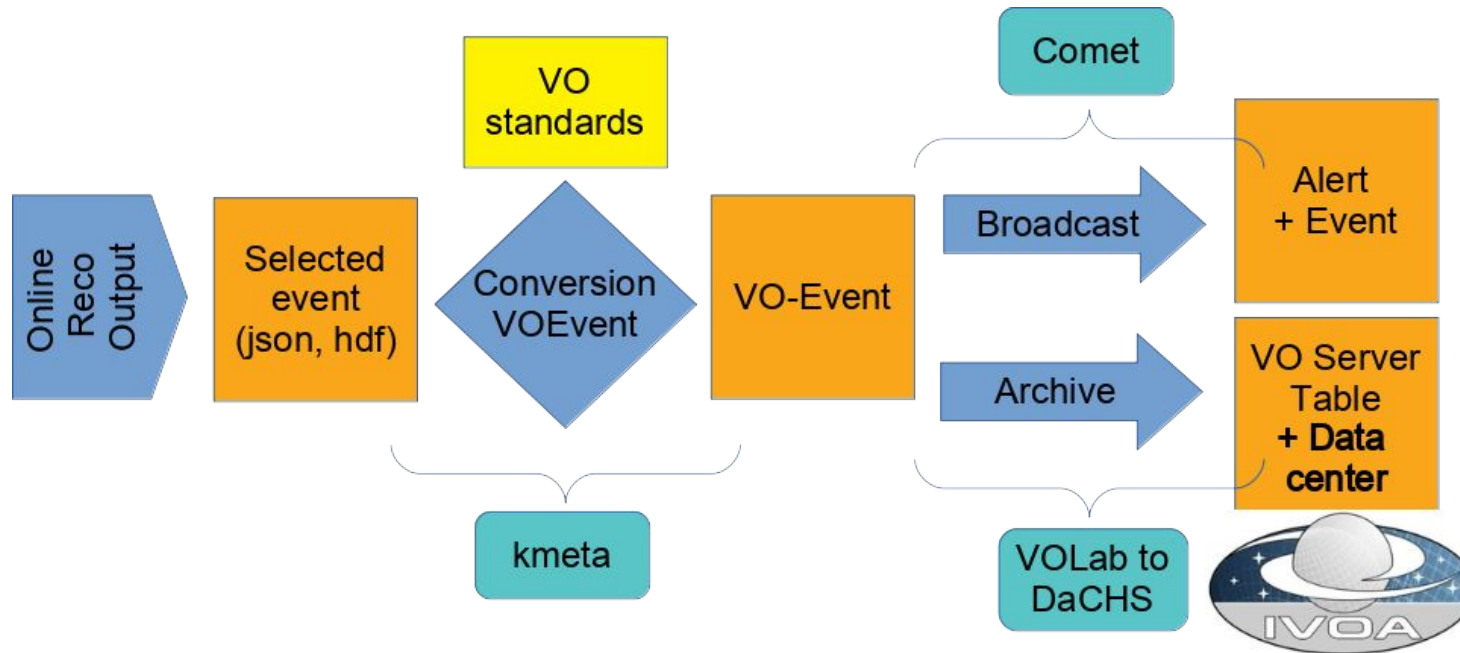
The KM3NeT Open Science System

VOserver: <http://vo.km3net.de/>
 Open Data center: <http://opendata.km3net.de>
 Open Science Portal: <https://open-data.pages.km3net.de/openscienceportal/>
 Education Portal: <http://edu.km3net.de/>
 Zenodo group: <https://zenodo.org/communities/km3net>
 Github group: <https://github.com/km3net>



rendered with PlantUML version 1.2020.20beta2
2020, The KM3NeT Collaboration, CC-BY4.0

Multimessenger alerts



Converting alert info (json) to VOEvent
Access additionally provided through **data center**

