**WORKS 2023**
**18th Workshop on Workflows in Support of Large-Scale Science**
https://works-workshop.org

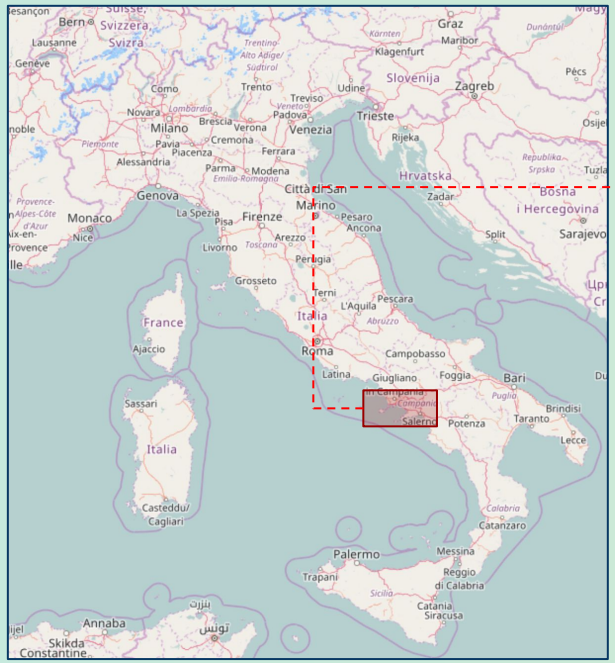

# Workflow Building Blocks: The Success Story of Environmental Modeling, HPC, and AI for Predicting Farmed Seafood Bacteria Contamination

**Raffaele Montella**
High-Performance Scientific Computing Smart Lab
University of Naples “Parthenope” - Naples, Italy
https://raffaelemontella.it raffaele.montella@uniparthenope.it http://hpsclab.uniparthenope.it

MytilEX - Extended Modelling mytilus farming System with High Performance Computing and Artificial Intelligence - team: Diana Di Luccio, Ciro Giuseppe De Vita, Gennaro Mellone, and Raffaele Montella

# Introduction and motivation

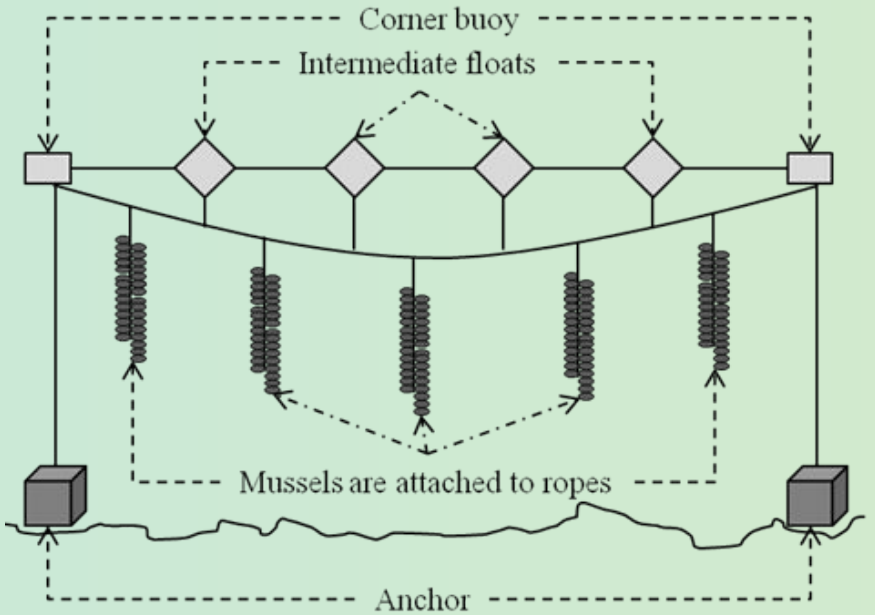
Mussels farming is an outstanding business cornerstone in the most part of Italian coastal regions.




**MARKET SIZE**

- Companies: 263(d)/886(r)
- Tons: ~64235 (⅔ EU prod - ISPRA)
- Euro/Kg: ~1.75 **l** (average)

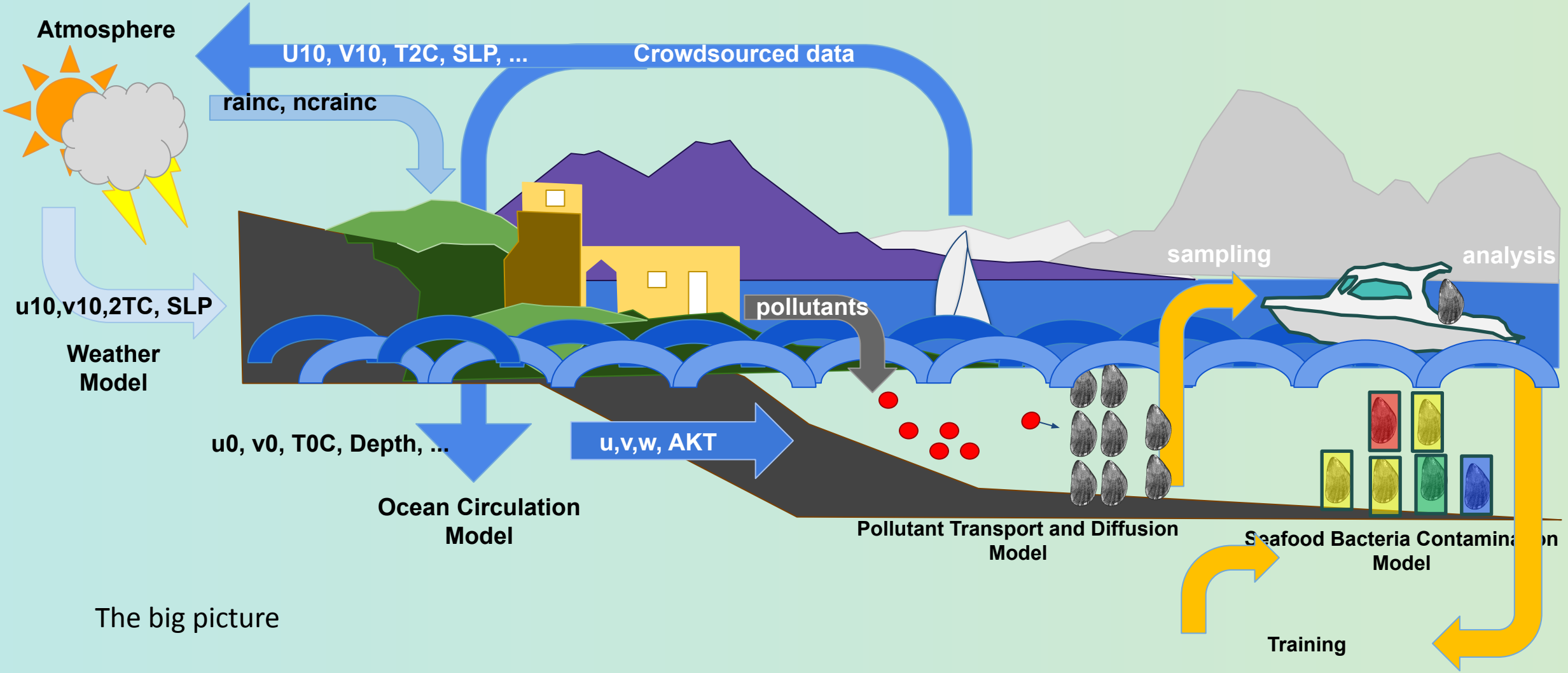
**~113M€ (2023, Italy)**




**Local Healthcare Agency (2012):**

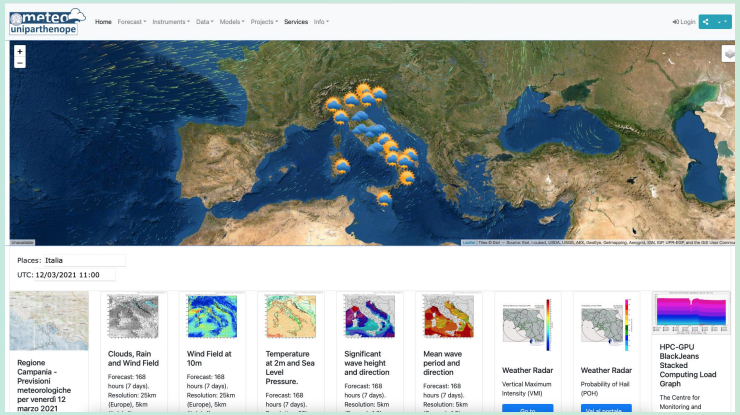
"Is it possible making predictions about the bacteria contamination in farmed mussels in order to limit human gastroenteric disease **and** the gathering interruptions?"

# Contextualization




The big picture

# The timeline



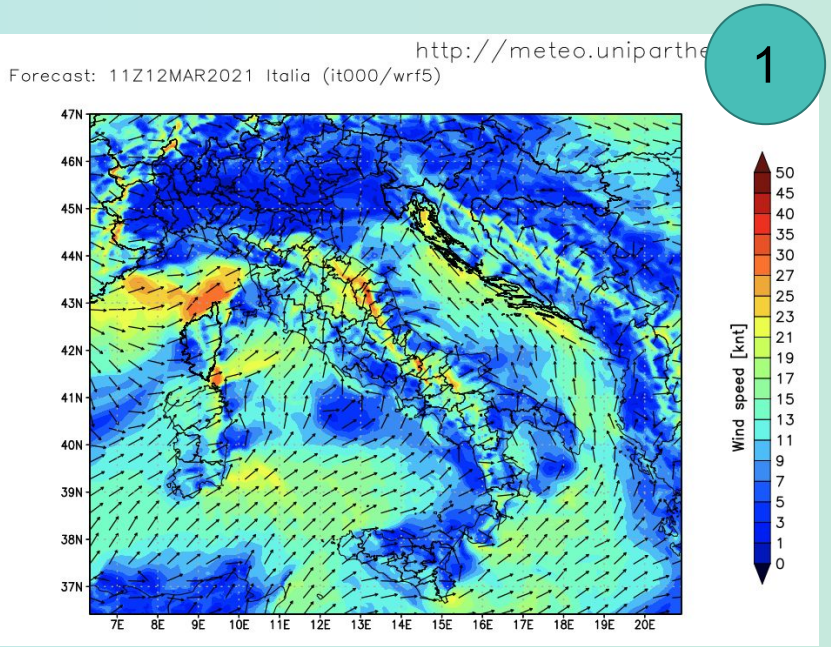

## http://meteo.uniparthenope.it

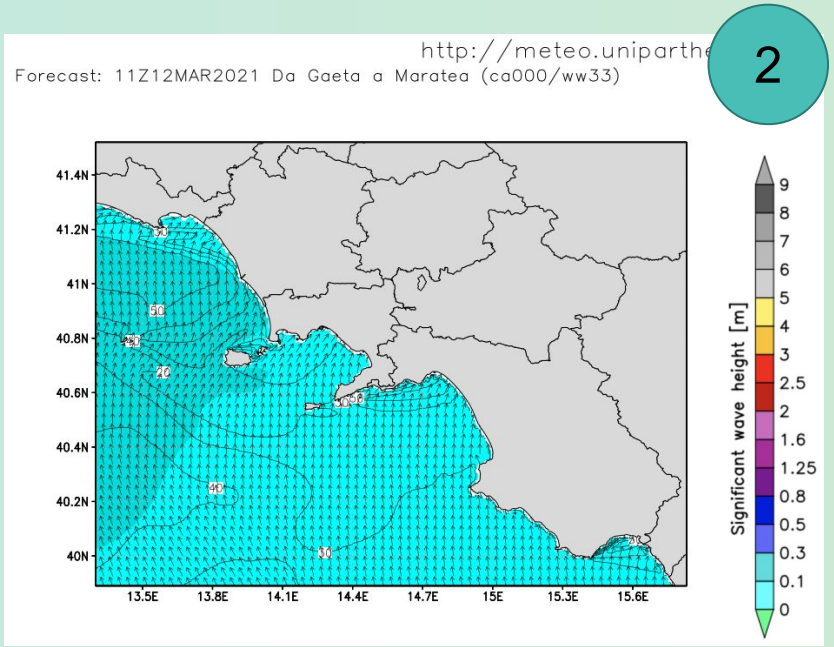
Center for Monitoring and Modeling Marine and Atmospheric applications (CMMMA, 2010)

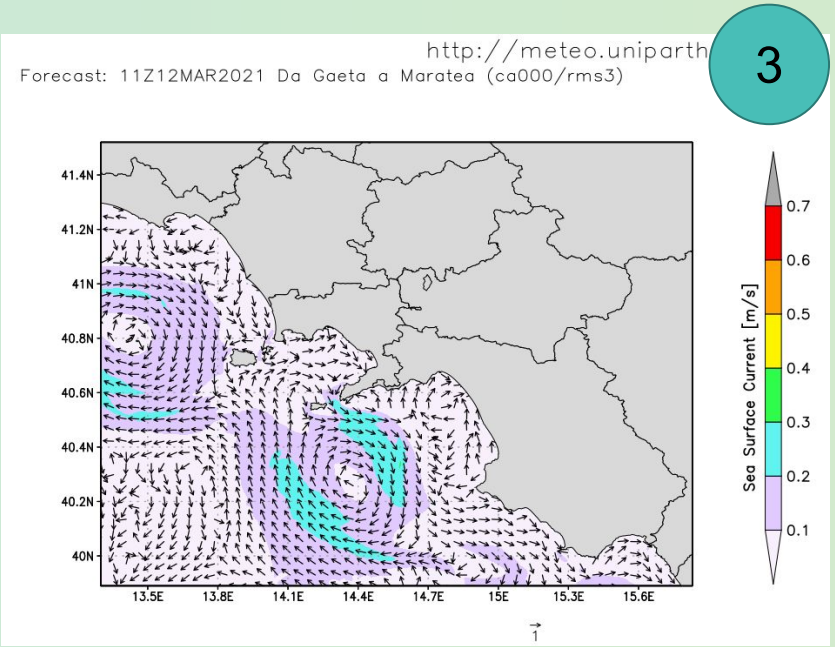
2018: Python ad-hoc workflow engine - DagOnStar.

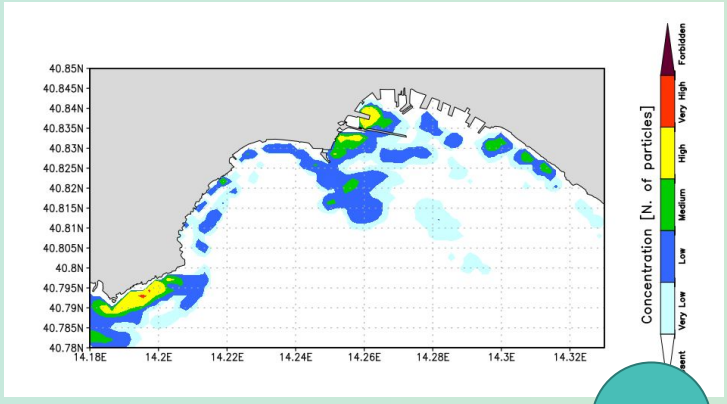
| 2012 | 2016 | 2021-22 | 2021-24 | 2023-28 |
|---|---|---|---|---|
| **Preliminary studies**<br><br>Production: Java ad-hoc workflow engine.<br><br>On demand: Face-IT Galaxy-ES | **MytiluSE** - **M**odelling m**ytilu**s farming **S**ystem with **E**nhanced web technologies.<br><br>Water quality Community Model | **MytilAI** - **M**odelling **mytil**us farming with **A**rtificial **I**ntelligence technologies<br><br>Water quality Community Model (MPI, OMP, GPU)<br><br>Artificial Intelligence water Quality Model | **ADMIRE** project EuroHPC - H2020<br><br>Workflow Engine “DagonStar” Environmental Application<br><br>Computational & Storage Malleability<br>Water quality Community Model | **MytilEX** - **Ex**tended **M**odelling **mytil**us farming System with High Performance Computing and Artificial Intelligence<br><br>**Planned improvements:**<br>HPC resources<br>DagOnStar<br>WaComM<br>AIQUAM |

# Products

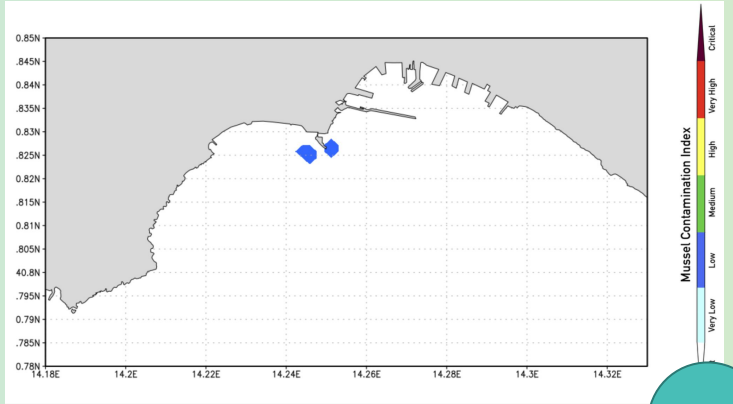


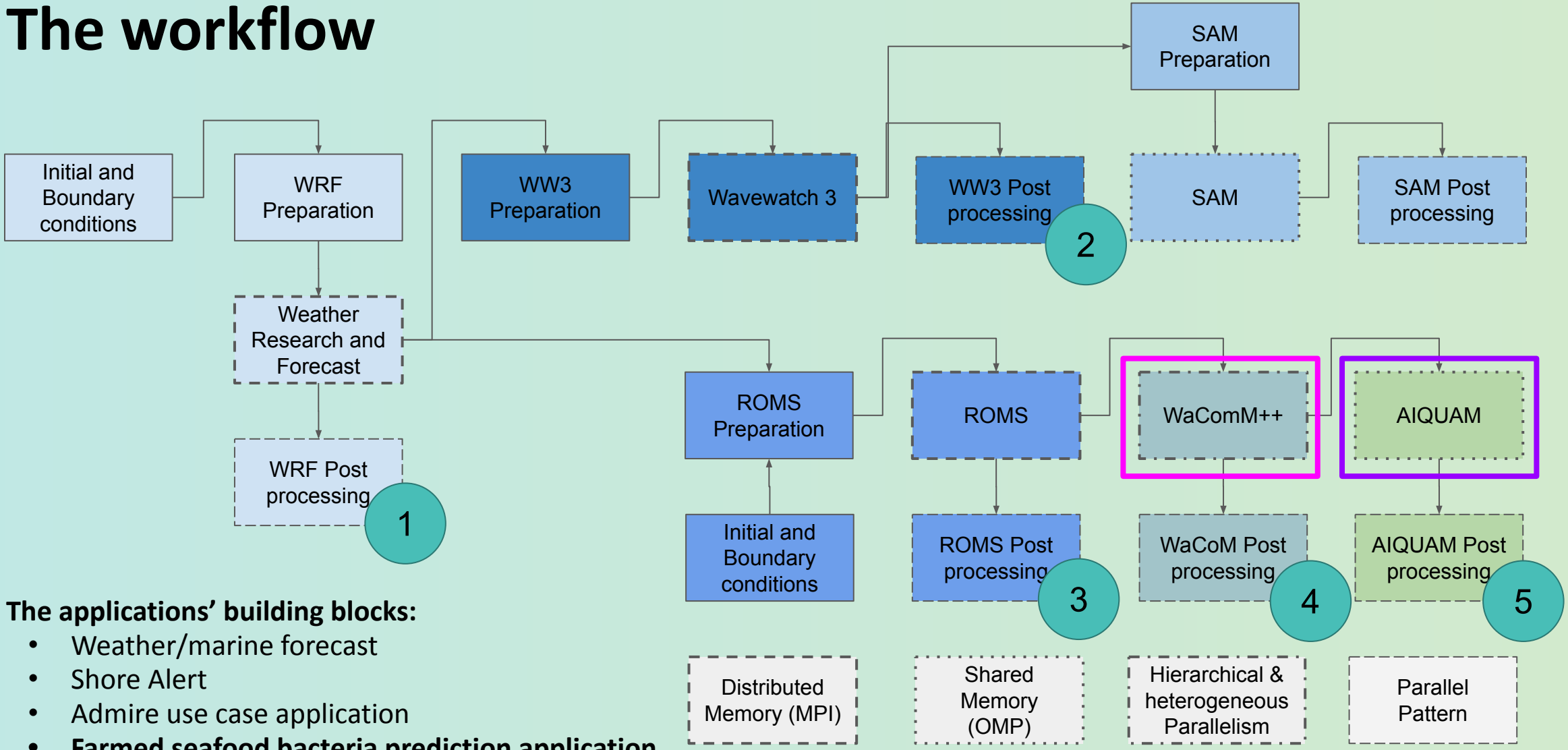

4




5

1 - Weather forecasts (168h, 1Km)
2 - Wind drive sea waves (168h, 1Km)
3 - Sea currents (168h, 160m)
4 - Inerts tracing (168, 160m)
5 - Seafood Bacteria Contamination Model (168h, 160m)

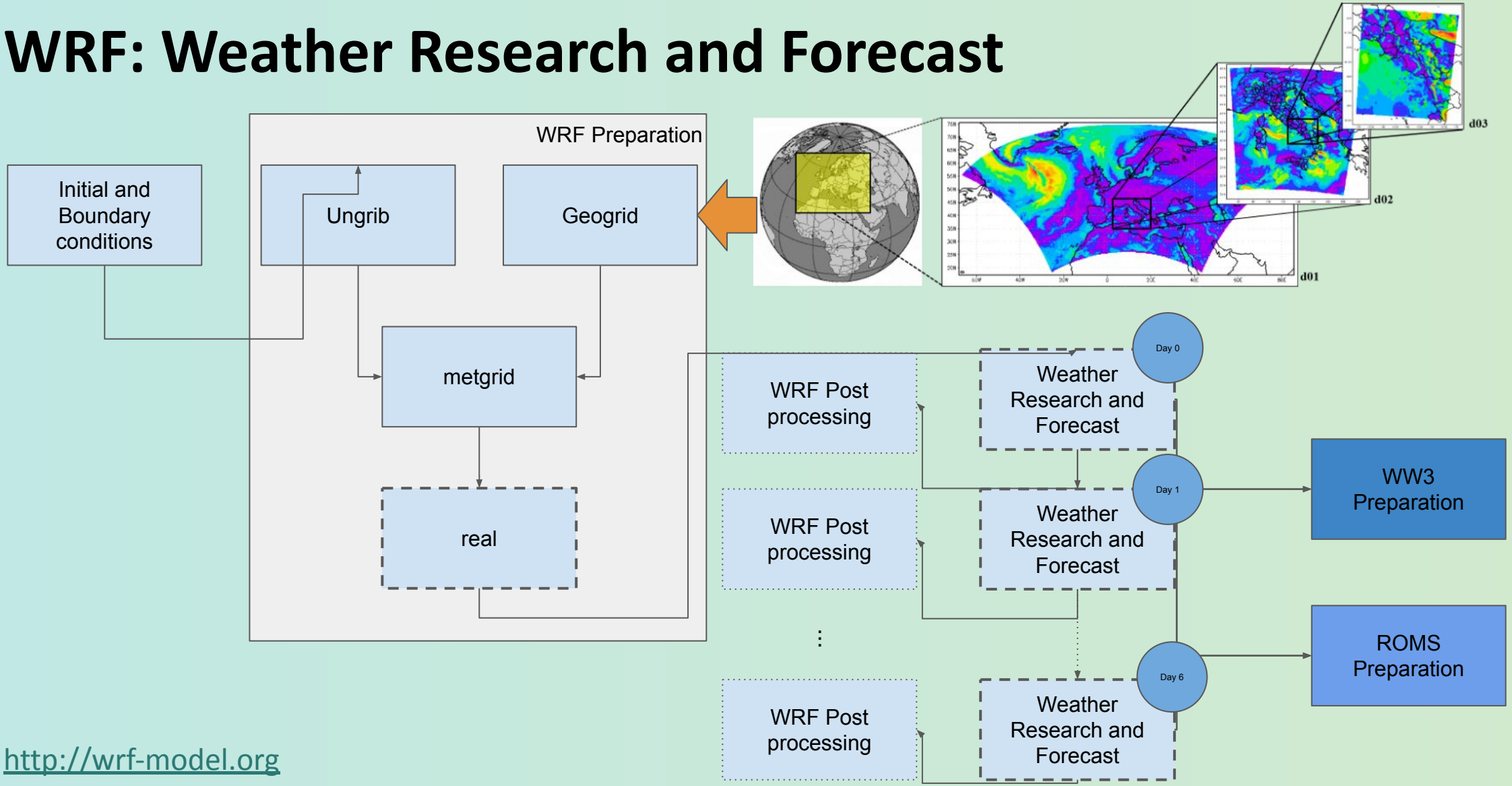
# The workflow




**The applications' building blocks:**

- Weather/marine forecast
- Shore Alert
- Admire use case application
- **Farmed seafood bacteria prediction application**

# WRF: Weather Research and Forecast



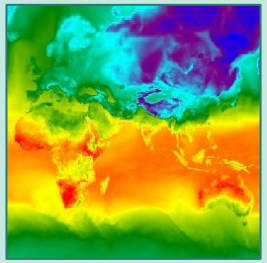

http://wrf-model.org



https://www.mmm.ucar.edu/models/wrf

# WRF: Input & Output

## input



Global Forecast System
National Centers for Environmental Prediction

Resolution: 0.5 degrees/3h
4 dataset per day
39 GB/run - 5.6 GB/day.

Storage: 107.4 GB/run

Scratch: 105 GB/run

## output

**Domain 1, 25Km, 1h, 168h**

History:
2.2 GB/day, 15.4 GB/run

Archive:
539 MB/day, 3.8 GB/run

Restart:
0.4 GB/day, 2.8 GB/run

**Domain 2, 5Km, 1h, 168h**

History:
4.3 GB/day, 30.1 GB/run

Archive:
2 GB/day, 14 GB/run

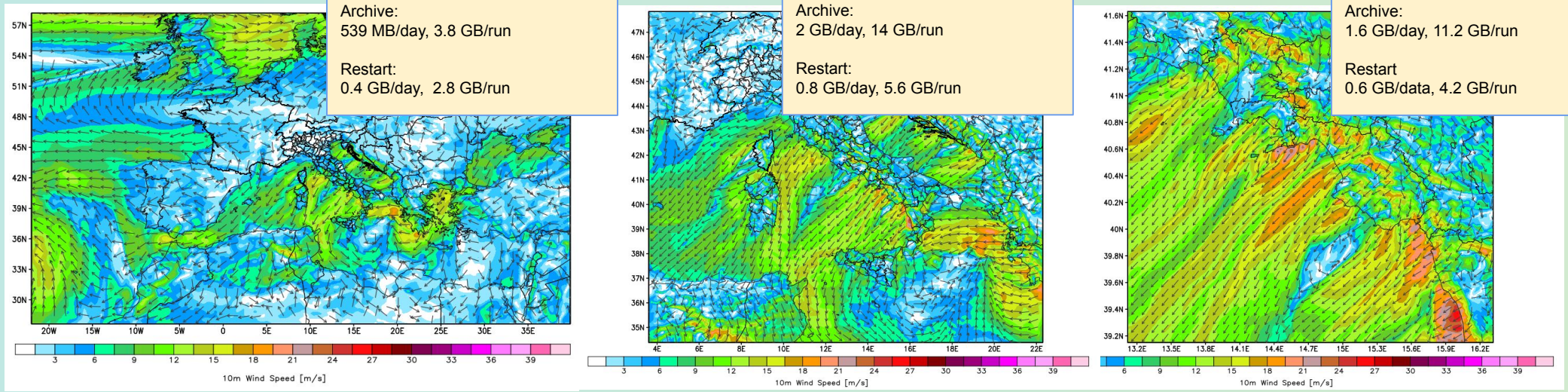
Restart:
0.8 GB/day, 5.6 GB/run

**Domain 3, 1Km, 1h, 168h**

History:
2.9 GB/day, 20.3 GB/run

Archive:
1.6 GB/day, 11.2 GB/run

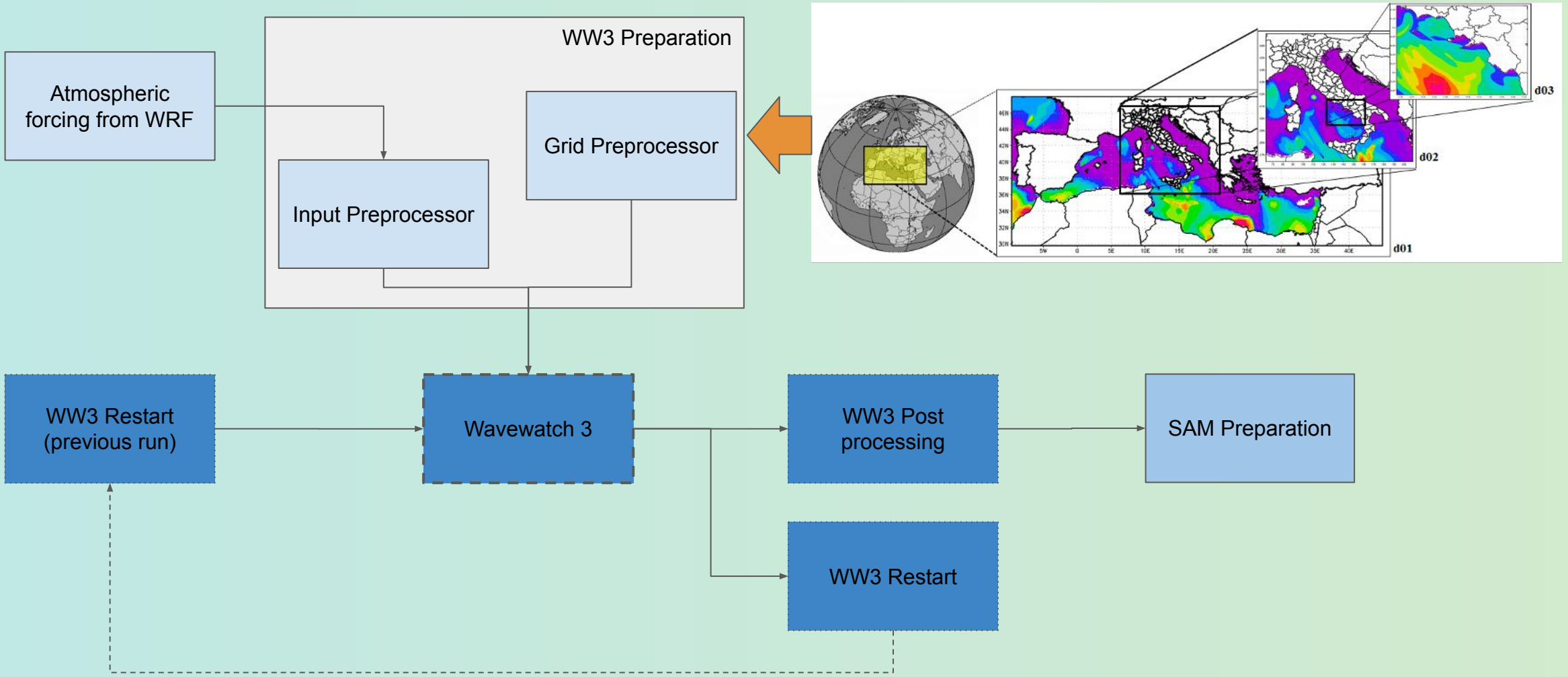
Restart
0.6 GB/data, 4.2 GB/run






https://www.mmm.ucar.edu/models/wrf

# WW3: Wavewatch III

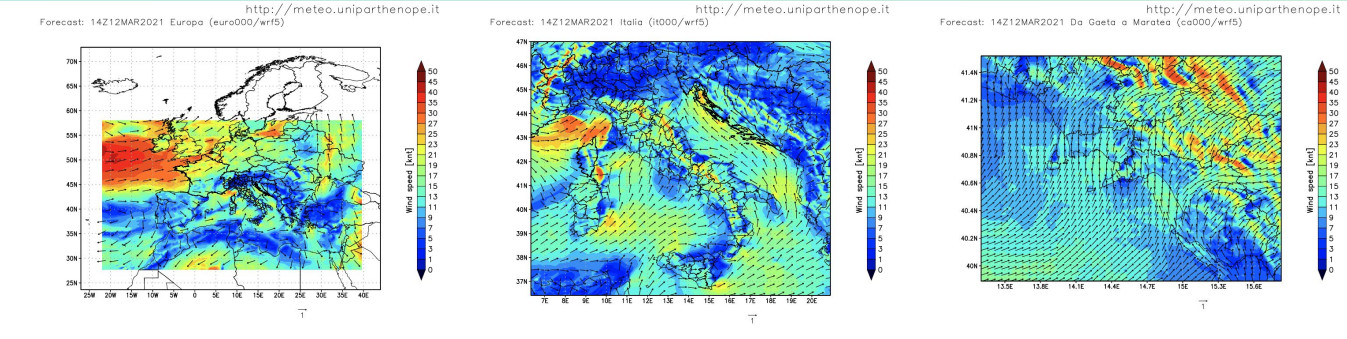





https://polar.ncep.noaa.gov/waves/wavewatch/

# WW3: Input & Output

## input



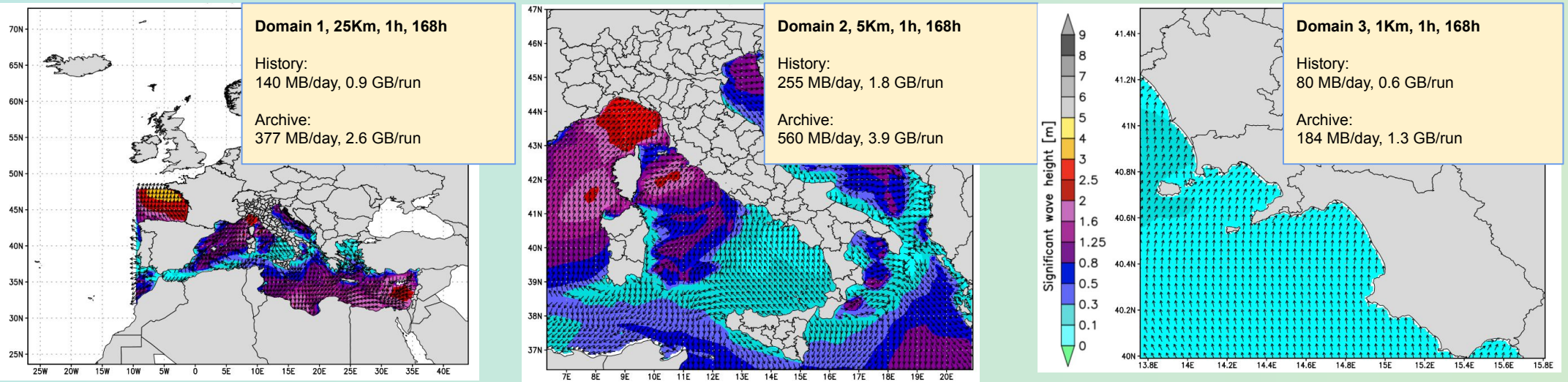

WRF - CMMMA

67 GB/run

Storage: 11 GB/run

Scratch: 7.7 GB/run

## output




**Domain 1, 25Km, 1h, 168h**

History:
140 MB/day, 0.9 GB/run

Archive:
377 MB/day, 2.6 GB/run

**Domain 2, 5Km, 1h, 168h**

History:
255 MB/day, 1.8 GB/run

Archive:
560 MB/day, 3.9 GB/run

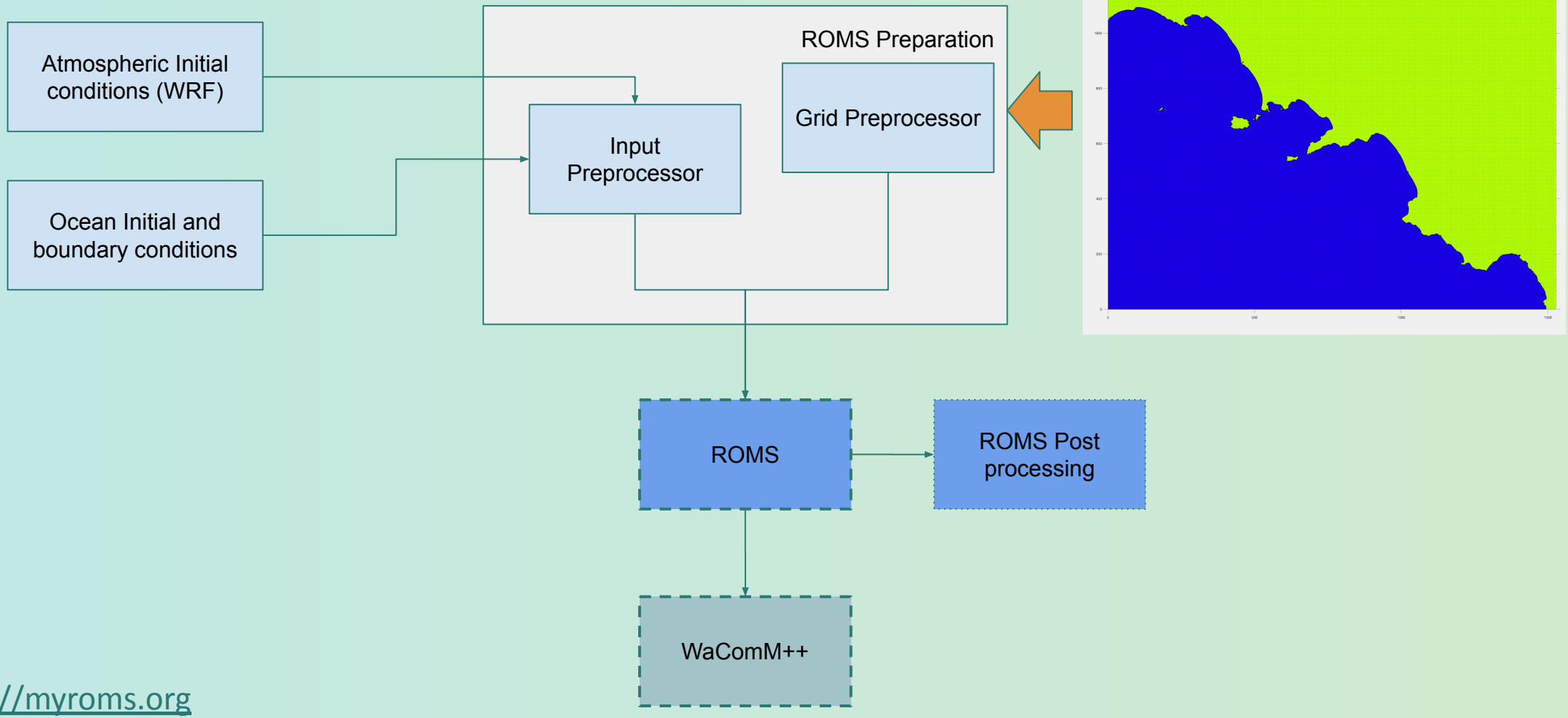
**Domain 3, 1Km, 1h, 168h**

History:
80 MB/day, 0.6 GB/run

Archive:
184 MB/day, 1.3 GB/run

Storage: 11 GB/run Scratch: 71 GB/run

# ROMS: Regional Ocean Model System



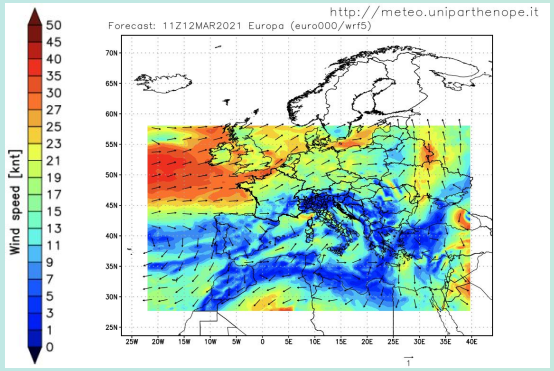

http://myroms.org



https://www.myroms.org

# ROMS: Input & Output

**input**

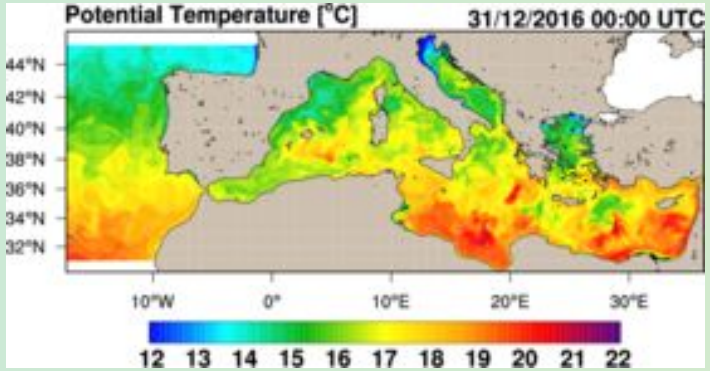



WRF - CMMMA

27 GB/run

Copernicus

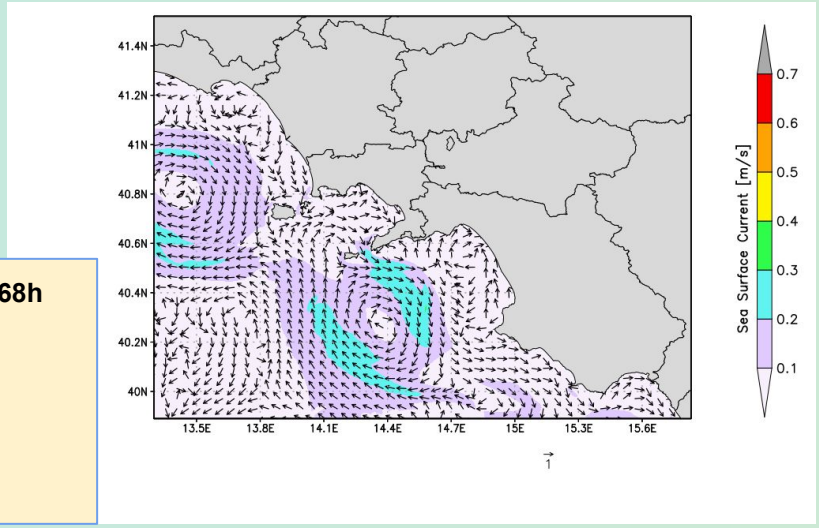
0.4 GB/run




**output**

Storage: 492 GB/run

Scratch: 332.5 GB/run




**Domain 3, 160 m, 1h, 168h**

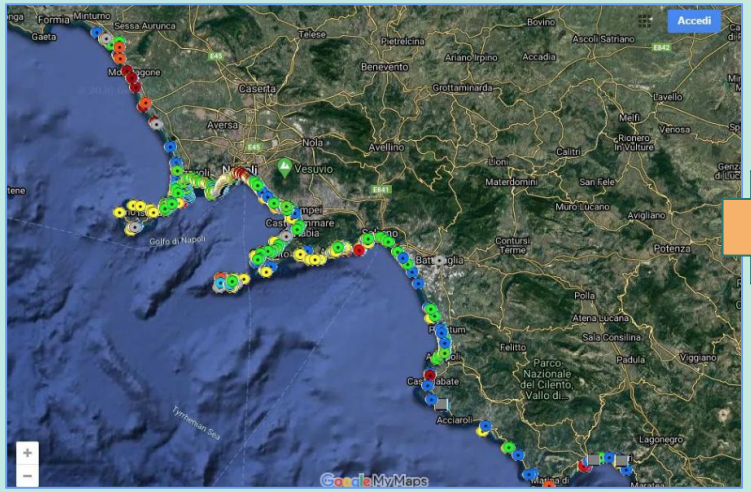
History:
63 GB/day, 441 GB/run

Archive:
7.3 GB/day, 51.1 GB/run



https://www.myroms.org

# WaComM++: Water quality Community Model ++




Initial Conditions (ROMS)

Emission Sources

WaCoM restart (previous run)

WaComM++

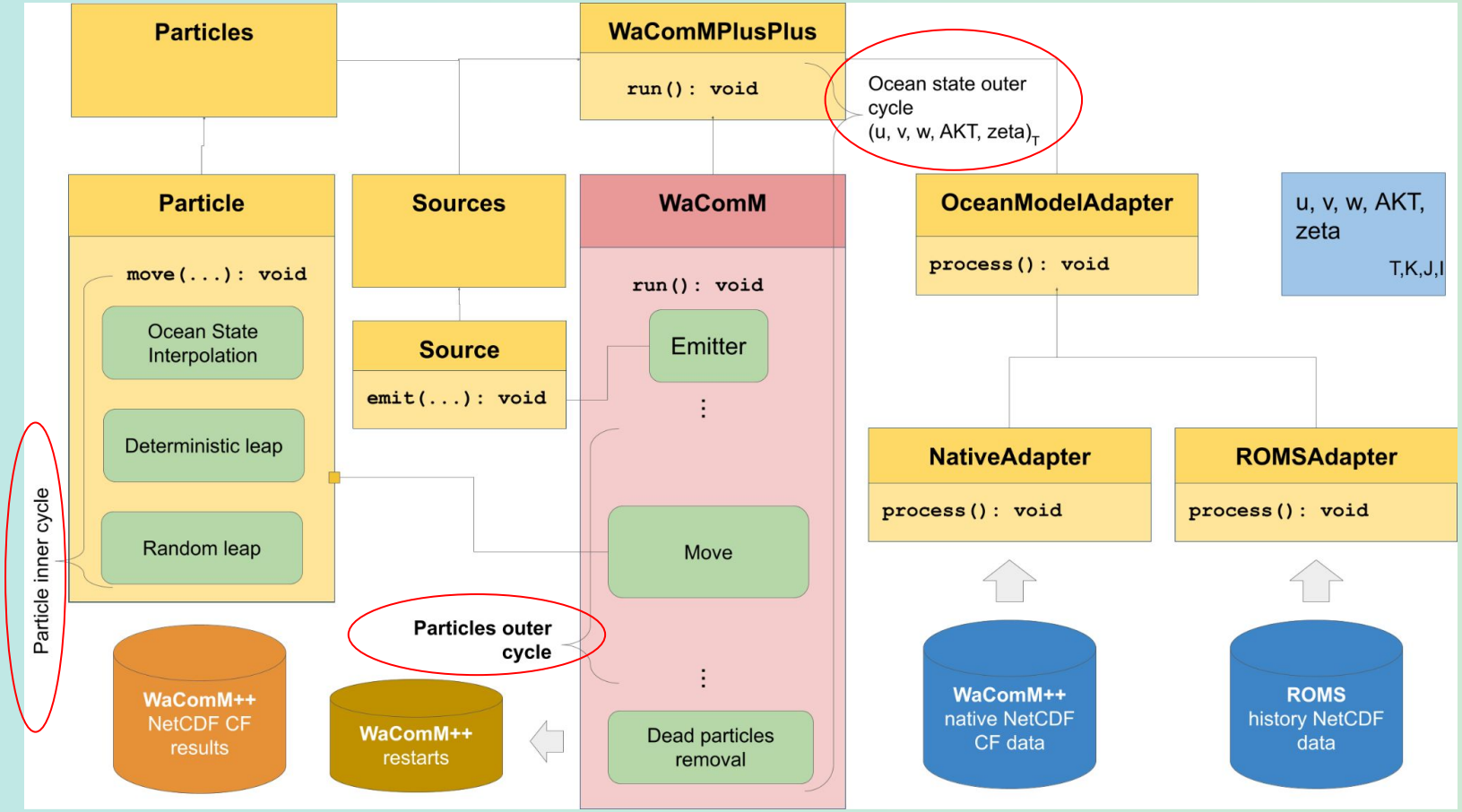
WaCoM Post processing

WaCoM restart



https://github.com/ccmmma/wacommplusplus

# WaComM++ architecture




The overall computation is performed over three nested cycles:

Ocean state outer cycle: for each time-referenced dataset (usually 1-hour), a WaComM component is instanced.

Particles outer cycle: moves the particle to process using ocean data.

Particle inner cycle: moves the particle within the considered time slice, applying the Lagrangian transport and diffusion equations integrated on a given time step.

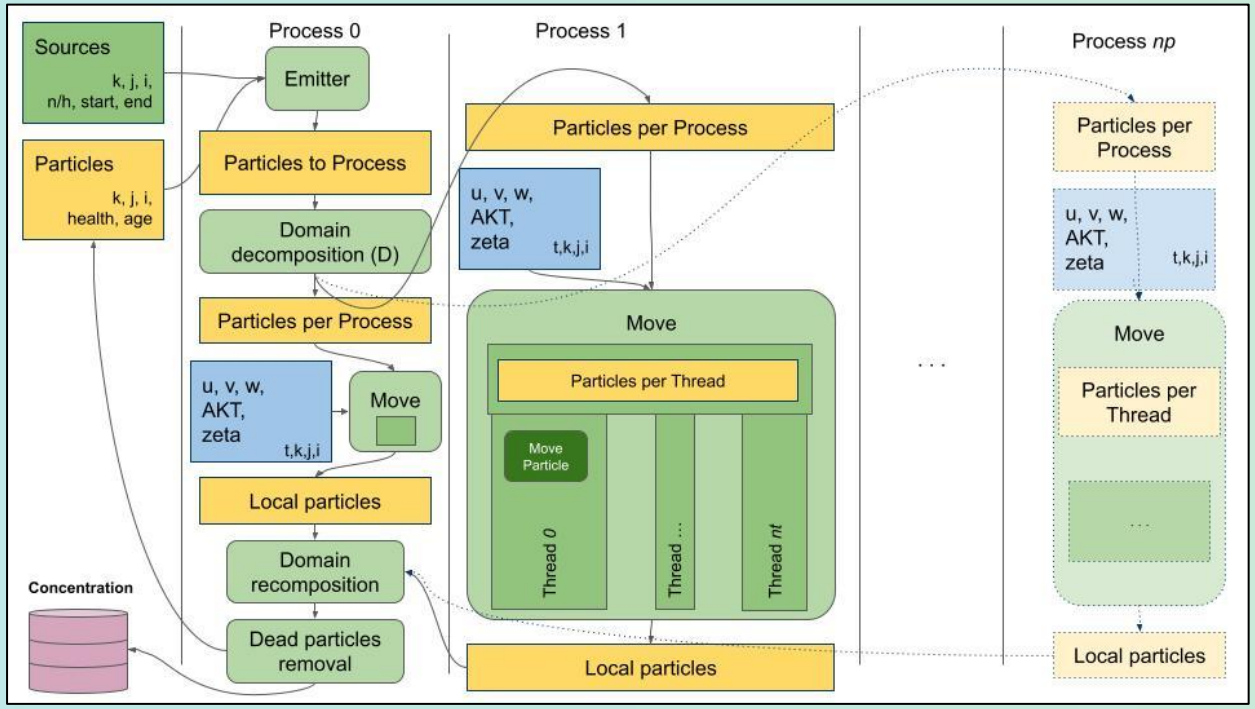
While time-dependent iterations characterize the ocean state outer cycle and the inner particle cycle, the particles' outer cycle has been hierarchically parallelized because each particle movement is independent of the others.



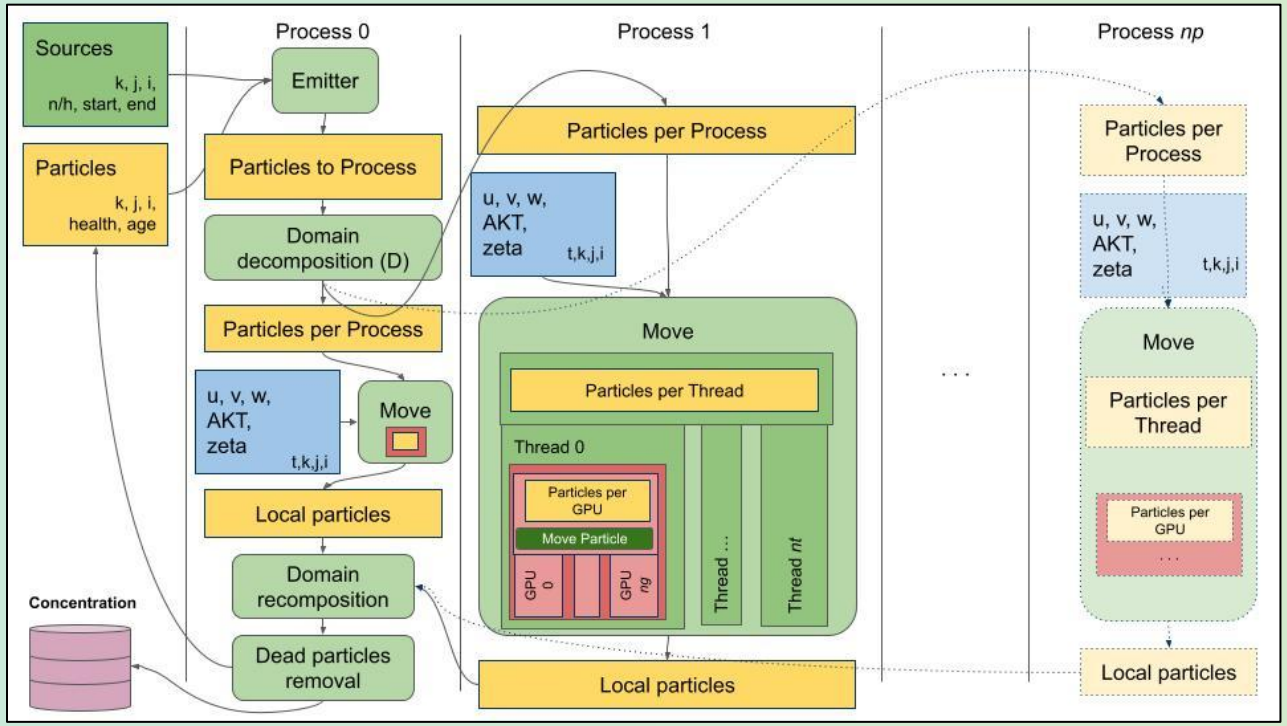
https://github.com/ccmmma/wacommplusplus

# WaComM++ hierarchical parallelization schema

Without multi-GPU paradigm.




With multi-GPU paradigm.

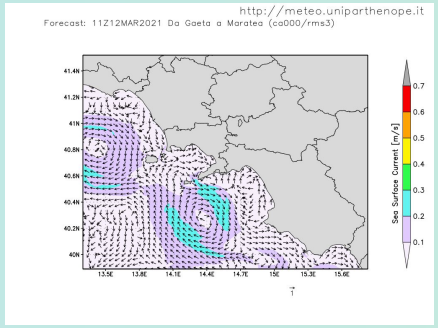





https://github.com/ccmmma/wacommplusplus

# WaComM++: Input & Output

## input



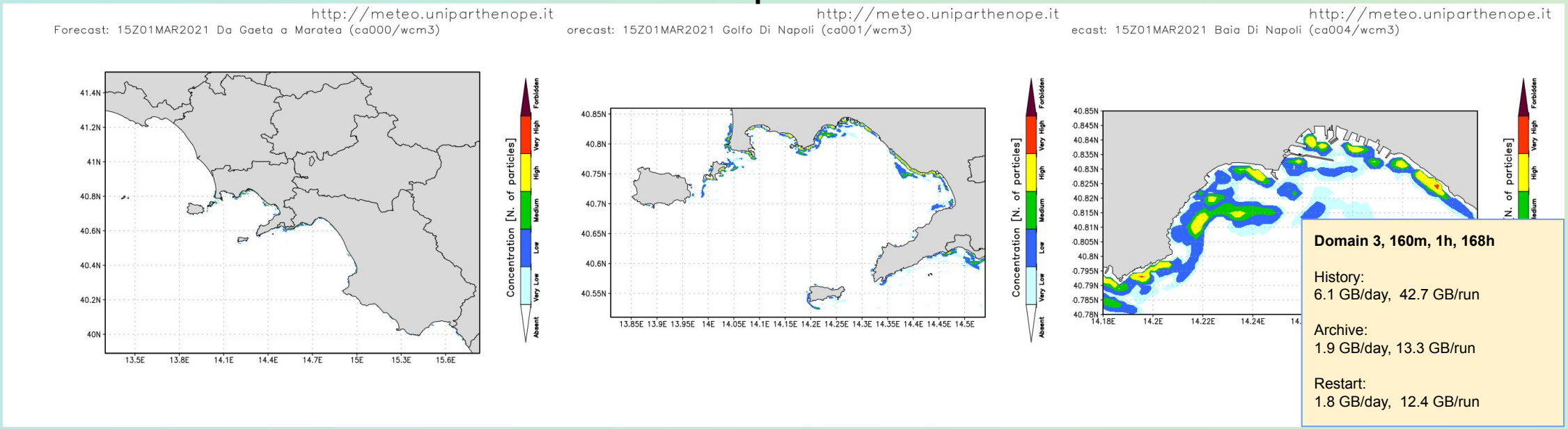

ROMS - CMMMA

305 GB/run

Storage: 68.4 GB/run

Scratch: 353 GB/run

## output




**Domain 3, 160m, 1h, 168h**

History:
6.1 GB/day, 42.7 GB/run

Archive:
1.9 GB/day, 13.3 GB/run

Restart:
1.8 GB/day, 12.4 GB/run



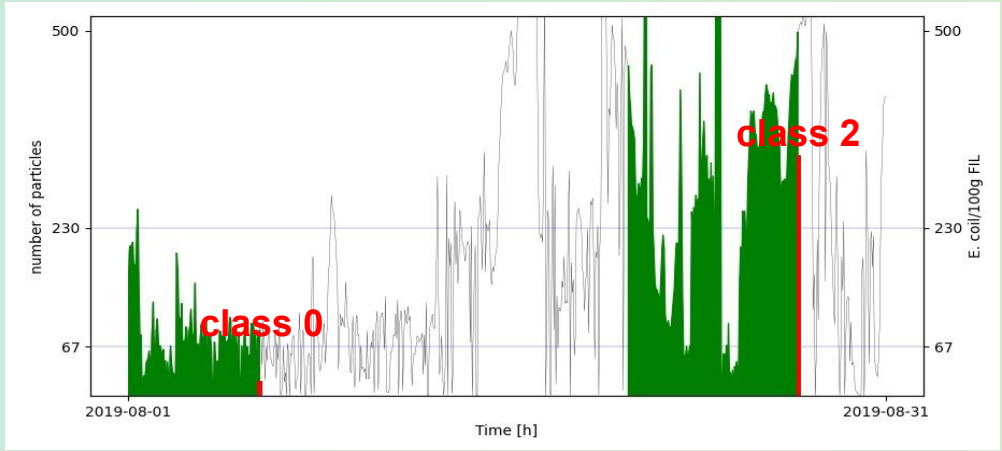
https://github.com/ccmmma/wacommplusplus

# Predicting bacteria contamination: the idea

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | NUMERO SCHEDA | ANNO AC | NUMERO ACCETTAZIONE | IZS ACCETTAZIONE | DATA PRELIEVO | DATA ARRIVO | DATA ESITO | REGIONE | Codice SITO | SITO | ESITO |
| 1295 | 1043A-101608-B | 2019 | 101608 | ISTITUTO ZOOPROFILATTICO SPERIMENTALE DEL MEZZOGIORNO | 30/09/19 | 30/09/19 | 04/10/19 | CAMPANIA | 1500016 | VARCATURO GIACOBBE | 230 |
| 1296 | 1043A-101610-B | 2019 | 101610 | ISTITUTO ZOOPROFILATTICO SPERIMENTALE DEL MEZZOGIORNO | 30/09/19 | 30/09/19 | 04/10/19 | CAMPANIA | 1500009 | MONTE DI PROCIDA | 230 |
| 1297 | 1043A-101611-B | 2019 | 101611 | ISTITUTO ZOOPROFILATTICO SPERIMENTALE DEL MEZZOGIORNO | 30/09/19 | 30/09/19 | 04/10/19 | CAMPANIA | 1500038 | ACQUAMORTA | 18 |
| 1298 | 1043A-101611-B | 2019 | 101611 | ISTITUTO ZOOPROFILATTICO SPERIMENTALE DEL MEZZOGIORNO | 30/09/19 | 30/09/19 | 04/10/19 | CAMPANIA | 1500038 | ACQUAMORTA | 45 |
| 1299 | 1043A-101611-B | 2019 | 101611 | ISTITUTO ZOOPROFILATTICO SPERIMENTALE DEL MEZZOGIORNO | 30/09/19 | 30/09/19 | 04/10/19 | CAMPANIA | 1500038 | ACQUAMORTA | 18 |
| 1300 | 1043A-102056-B | 2019 | 102056 | ISTITUTO ZOOPROFILATTICO SPERIMENTALE DEL MEZZOGIORNO | 01/10/19 | 01/10/19 | 04/10/19 | CAMPANIA | 1500026 | TORRE DI PESCOPAGANO | 45 |
| 1301 | 1043A-102125-B | 2019 | 102125 | ISTITUTO ZOOPROFILATTICO SPERIMENTALE DEL MEZZOGIORNO | 01/10/19 | 01/10/19 | 04/10/19 | CAMPANIA | 1500012 | PUNTA CAVALLO, NISIDA | 18 |
| 1302 | 1043A-102133-B | 2019 | 102133 | ISTITUTO ZOOPROFILATTICO SPERIMENTALE DEL MEZZOGIORNO | 01/10/19 | 01/10/19 | 04/10/19 | CAMPANIA | 1500012 | PUNTA CAVALLO, NISIDA | 18 |
| 1303 | 1043A-102583-B | 2019 | 102583 | ISTITUTO ZOOPROFILATTICO SPERIMENTALE DEL MEZZOGIORNO | 02/10/19 | 02/10/19 | 07/10/19 | CAMPANIA | 1500038 | ACQUAMORTA | 110 |

**MPN (Most Probable Number):** analytical methodology for counting the microbial burden of an organic sample.

- The features are built by the concentration of particles per hour (168) produced by the **WaComM++ model**
- The time of assimilation of mussels can be expressed by $\int_{t_0}^{t_0-\Delta t} f\, dt$
- Dataset is composed as follows:
  - **feature: produced by WaComM++ as time series**
  - **labels: produced by analytical microbiological counting**
    - **class 0 (0-67 MPN / 100 g)**
    - **class 1 (67-230 MPN / 100 g)**
    - **class 2 (230-4600 MPN / 100 g)**
    - **class 3 (> 4600 MPN / 100 g)**



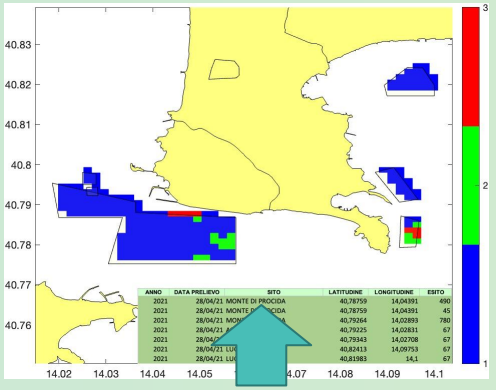

https://github.com/ccmmma/aiquam/

# AIQUAM: Artificial Intelligence water Quality Model

**AIQUAM** implements an AI model for seawater quality predictions.
The model performs **time series classification** leveraging various and different algorithms and then performs a weighted majority report for predicting the best result.

It consists of training a dataset classifier to map possible inputs to a probability distribution over the class variable values (labels). We tested three ML models:

- **KNN: K-Nearest Neighbors Algorithm**
- **KNN + DTW: Dynamic Time Warp distance (best results, more than 90%(**
- **CNN: Convolutional Neural Network**

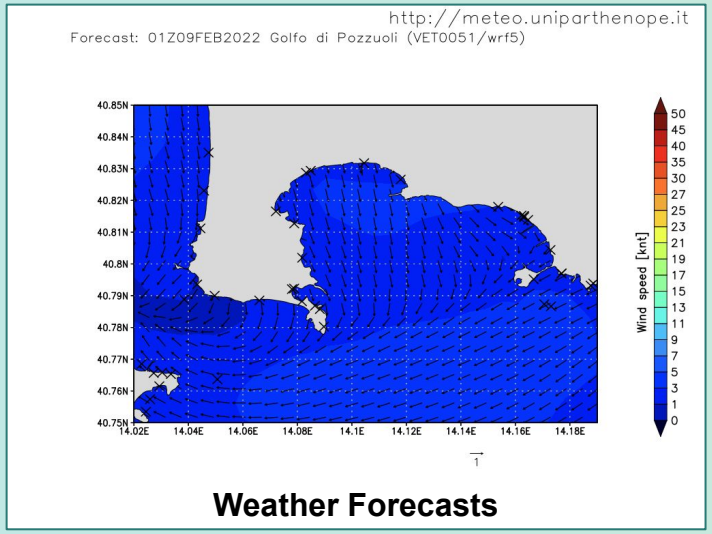


**Mussel Contamination Prediction**

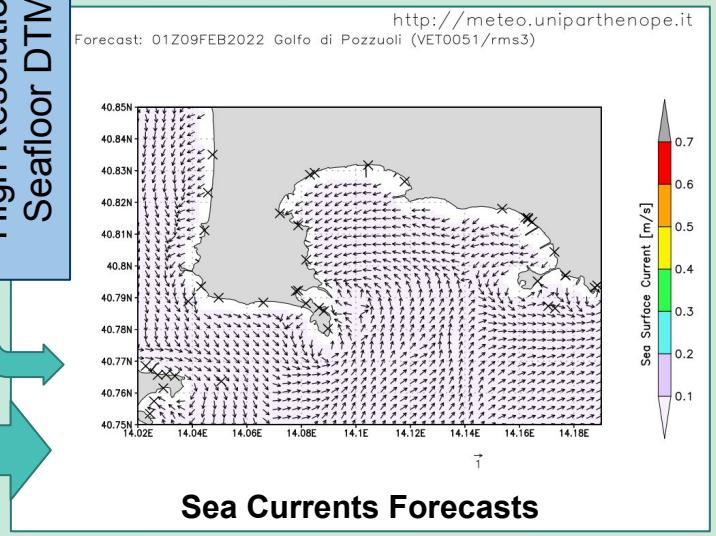
K.N.N. + DTW

High Resolution Seafloor DTM

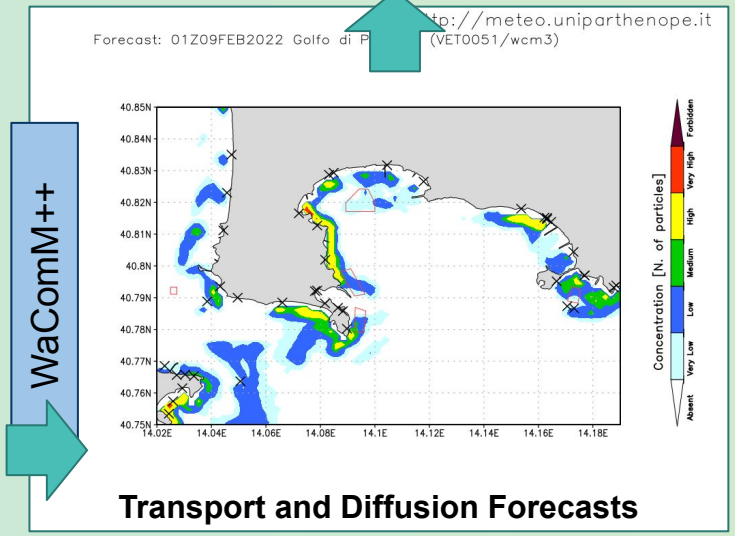
WaComM++




**Weather Forecasts**




**Sea Currents Forecasts**

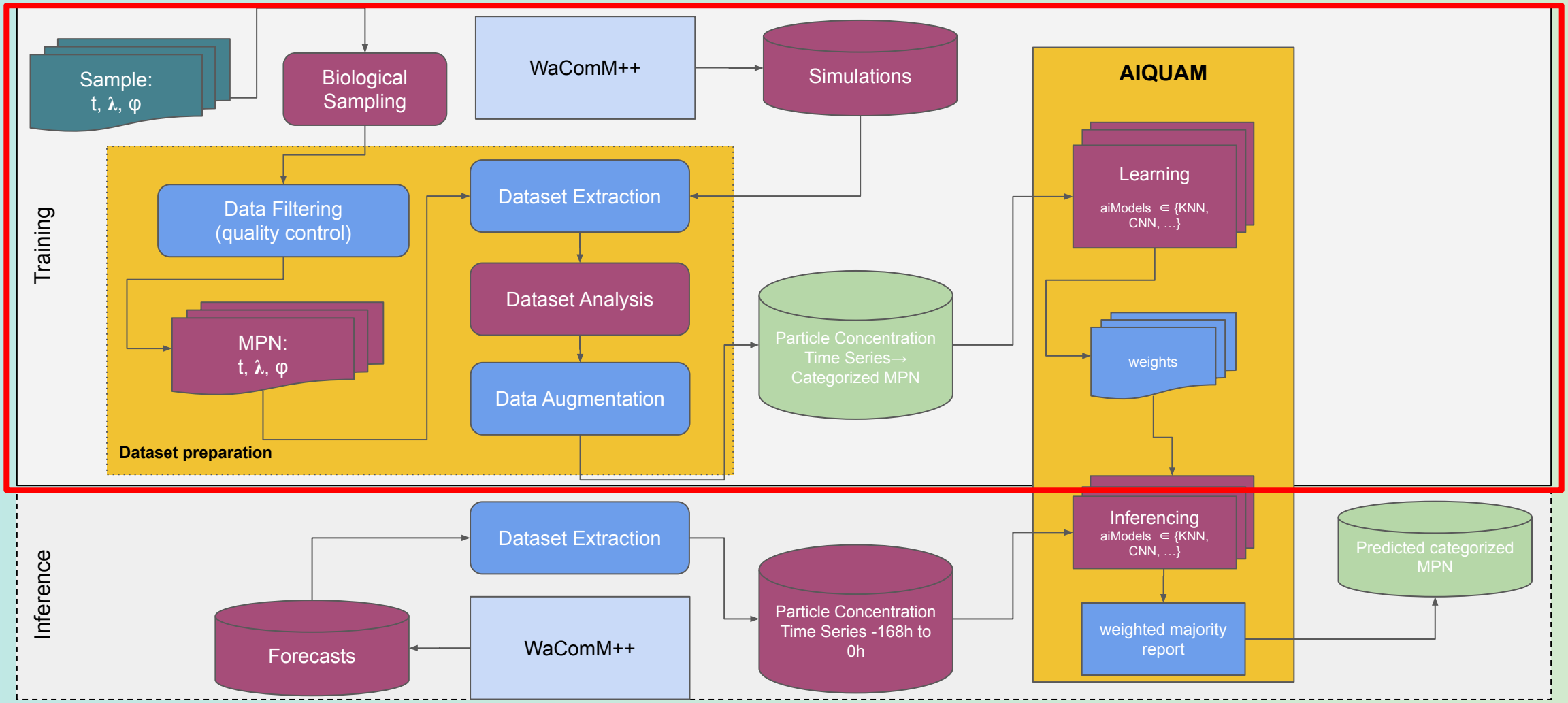



**Transport and Diffusion Forecasts**



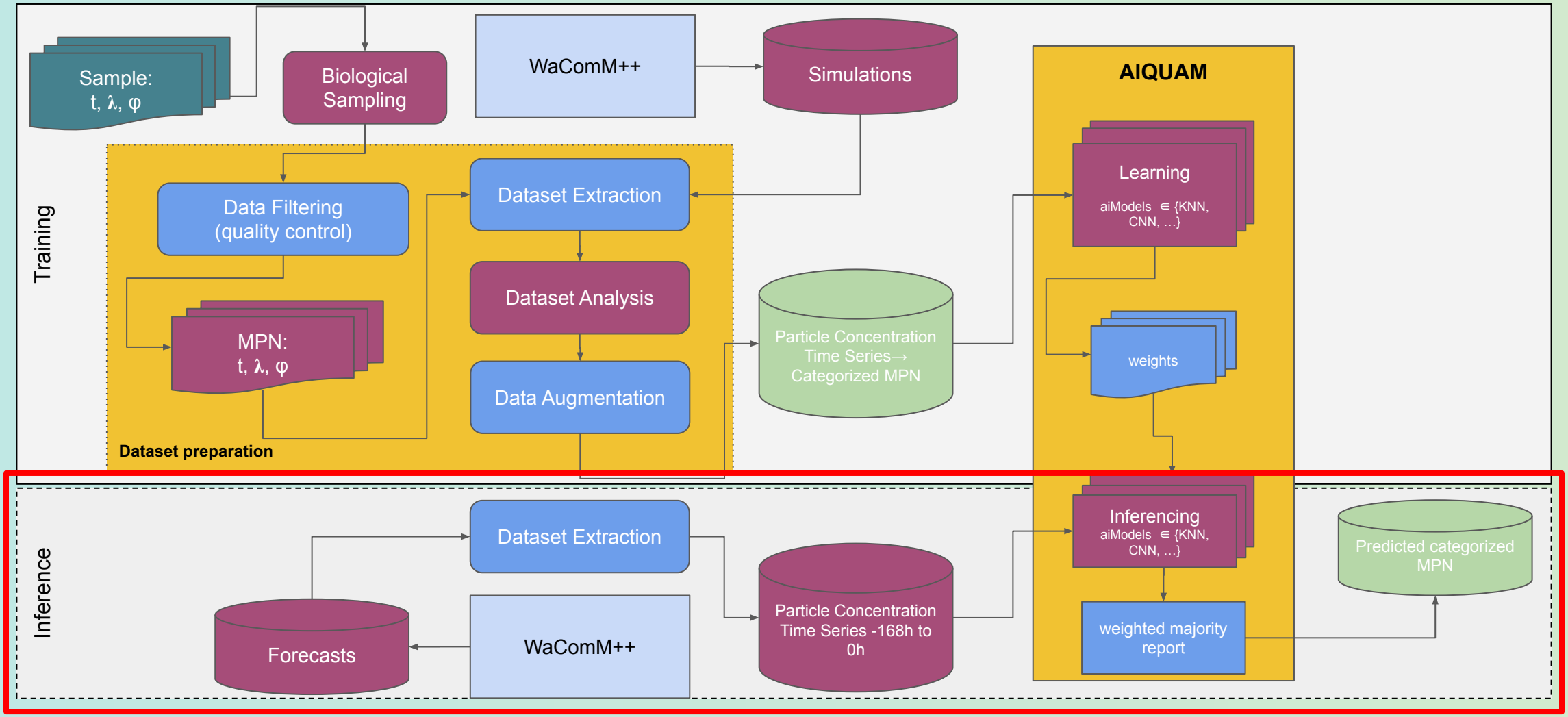
https://github.com/ccmmma/aiquam/

# AIQUAM: Architecture - Training phase




https://github.com/ccmmma/aiquam/

# AIQUAM: Architecture - Prediction phase

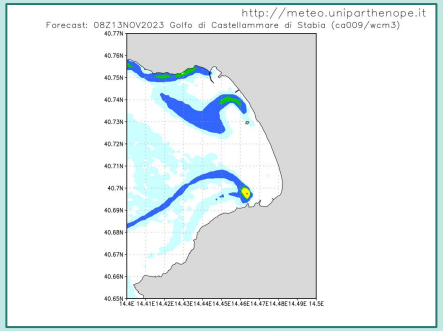





https://github.com/ccmmma/aiquam/

# AIQUAM: Input & Output

## input



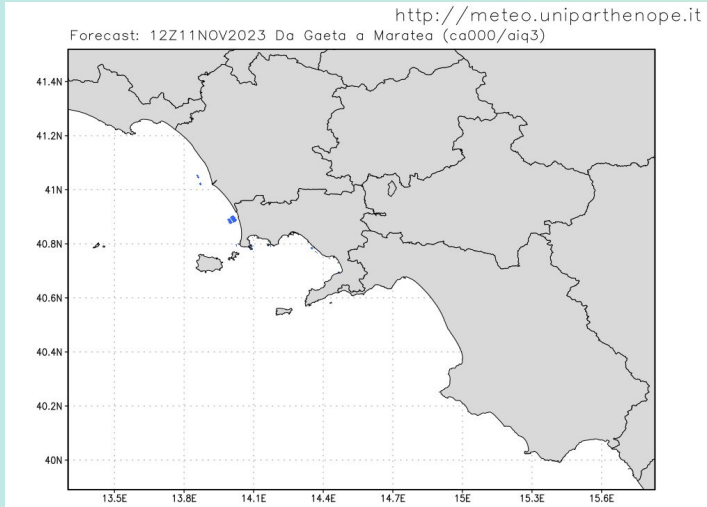

WaCom M++ - CMMMA

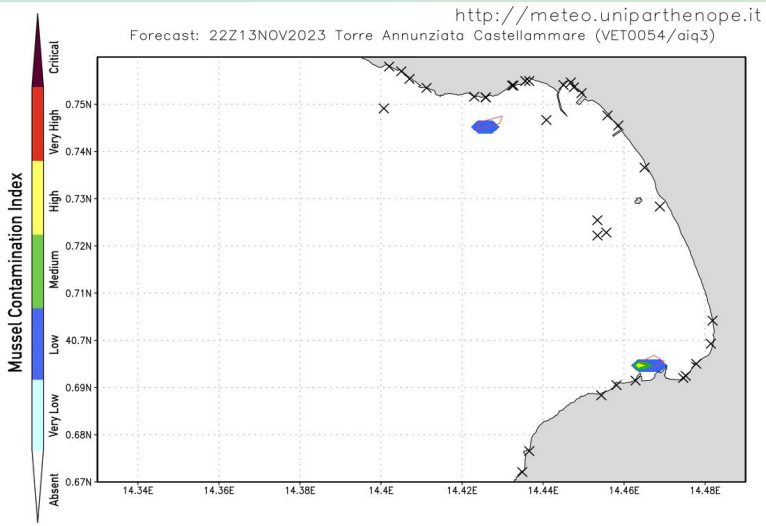
9.4 GB/run

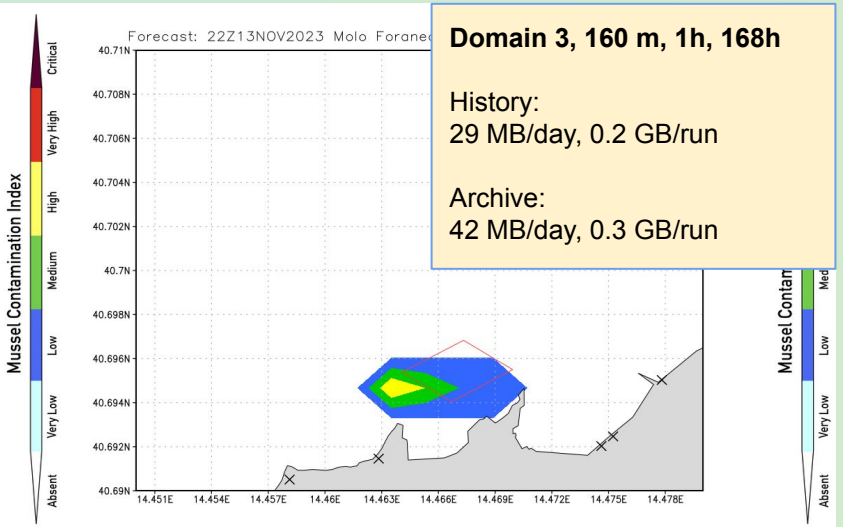
Storage: 0.5 GB/run

Scratch: 9.6 GB/run

## output




**Domain 3, 160 m, 1h, 168h**

History:
29 MB/day, 0.2 GB/run

Archive:
42 MB/day, 0.3 GB/run



https://github.com/ccmmma/aiquam/

# Direct Acyclic Graphs as parallel jobs on anything

**DagOnStar is a production-oriented workflow engine:**

- **Integration** in the Python environment.
- **Minimal** footprint for external software components execution.
- **Avoiding** any **centered data management**.
- **Straightforward** definition of tasks:
  - Python scripts.
  - Web interaction.
  - External software components.
  - Parallel patterns.
- **Execution** sites **independence**:
  - Local / scheduler (SLURM).
  - Containers (Docker).
  - Clouds (AWS, OpenStack, DigitalOcean).
- **Similar products (short incomplete list):** Parsl, StreamFlow, …


Named after the Phoenician god-fish *Dagon* known by ancient Greeks as *Triton*.

NB: The **star** symbol * is the wildcard for **anything**.



https://github.com/dagonstar/

# Programming Model

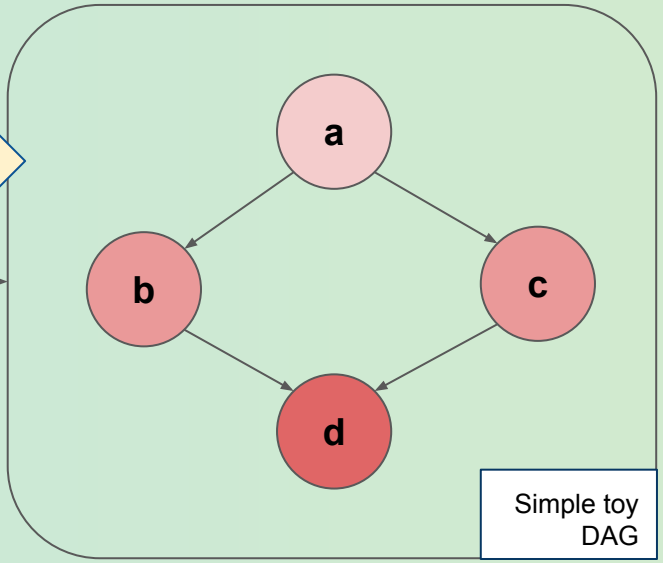
**Python Script: "DagOnStar Hello World App"**

```
from dagon import Workflow
from dagon.task import TaskType, DagonTask
…
workflow = Workflow("myapp")
workflow.add_task(DagonTask(TaskType.BATCH, "a", "..."))
workflow.add_task(DagonTask(TaskType.BATCH, "b", "workflow:///a"))
workflow.add_task(DagonTask(TaskType.BATCH, "c", "workflow:///a"))
workflow.add_task(DagonTask(TaskType.BATCH, "d", "workflow:///b
workflow:///c"))
workflow.run()
sys.exit(0)
```

Task Flow — Defined by task dependencies.

Data Flow — Defined by data dependencies.

Data Flow



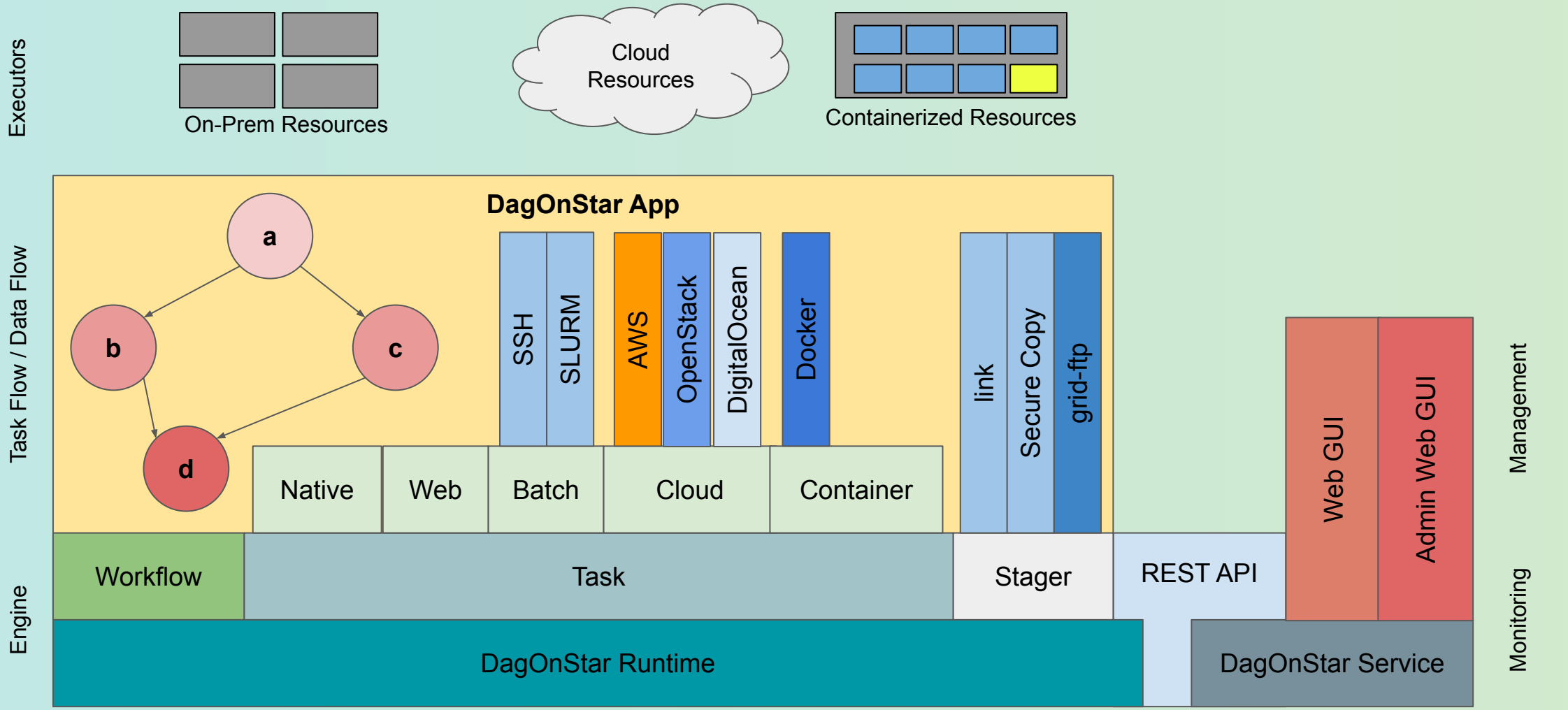

- Dealing with actual data files instead of high-level defined datasets.
- Performing backward data references in order to create dependencies.
- Having more Workflow instances in the same Python application.

**DagOnStar has been designed by a computational environmental application friendly programming model.**



https://github.com/dagonstar/

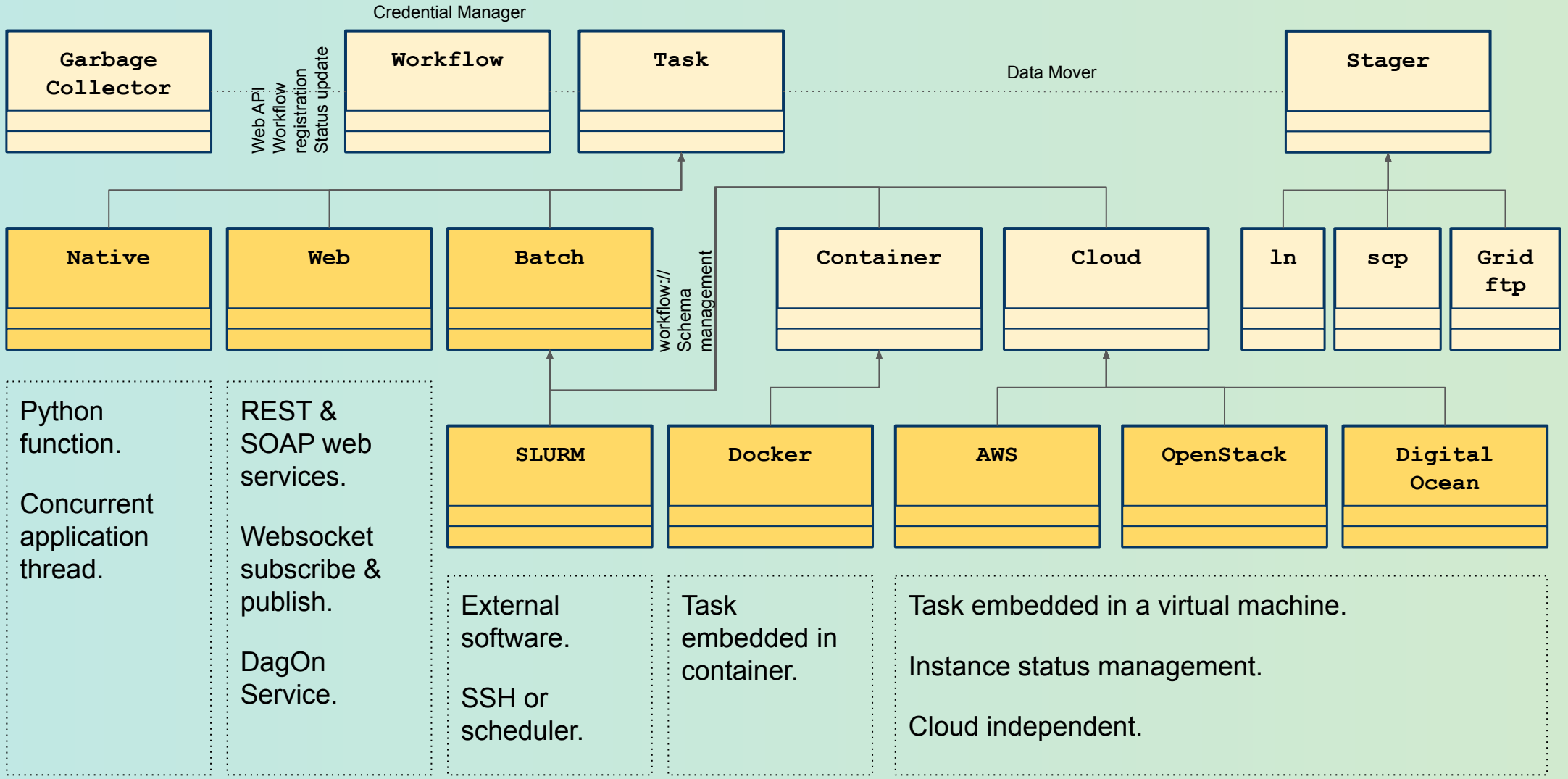
# Architecture






https://github.com/dagonstar/

# Components

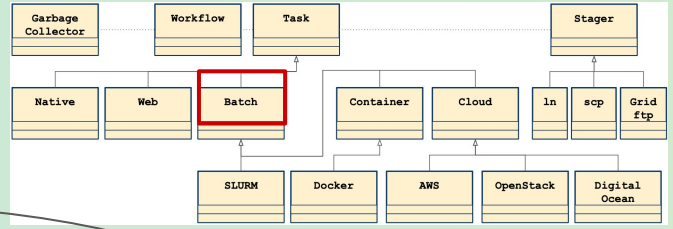





https://github.com/dagonstar/

# The workflow:// schema

The **Batch** component takes charge of the management of data dependencies using the **workflow:// schema**.



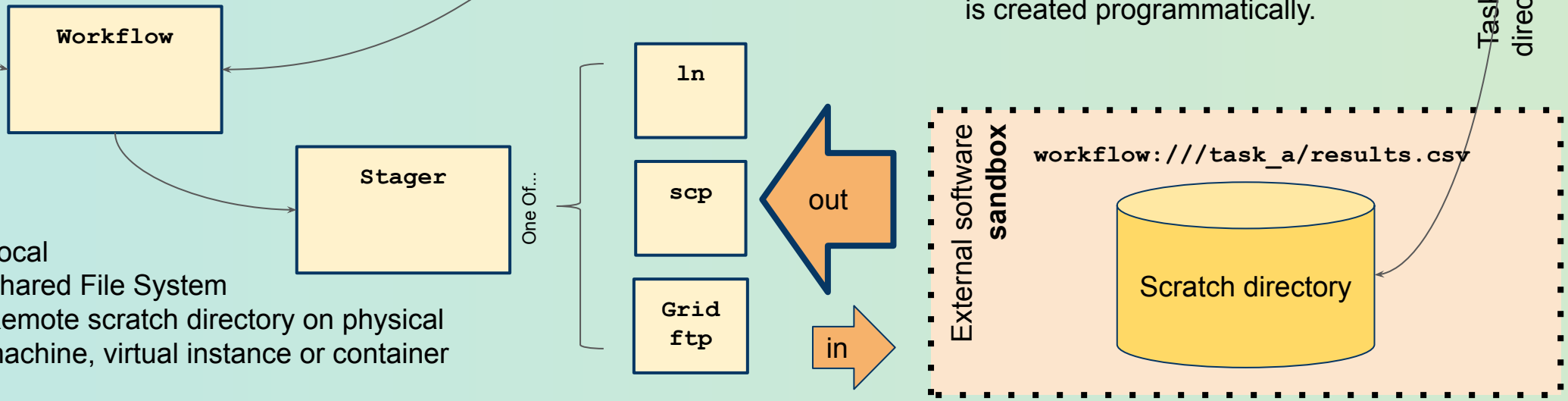

**workflow://workflow_unique_name/task_unique_name/**

The schema label

The workflow unique name
An UUID could be used
If empty means “current workflow”

The task unique name
Can be dynamically generated by the Python script when the workflow is created programmatically.

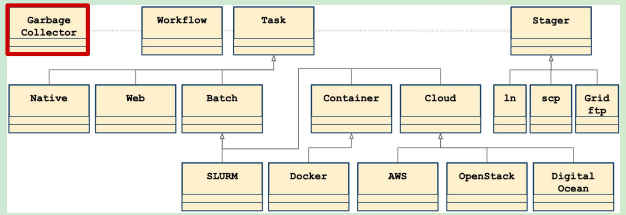
Task scratch directory root




- Local
- Shared File System
- Remote scratch directory on physical machine, virtual instance or container

# The garbage collector

- Tracks the storage and computational resources allocated during tasks execution.
- Proceeds to dispose them when no longer needed.




**Make Dependencies**

```
For each batch task in the <workflow> ...
    For each workflow://<workflow>/<task>/ reference in the task command line ...
        Increment the number of reference to <task>
```

**On Task Finish**

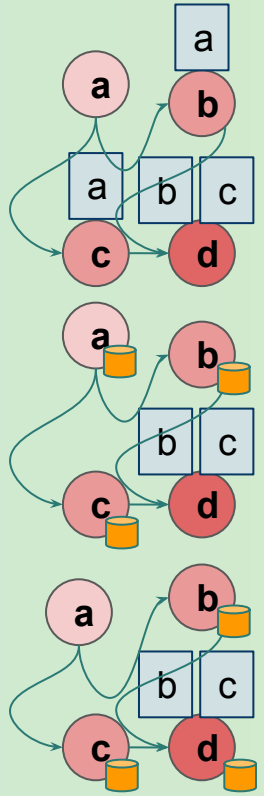
```
For each workflow://<workflow>/<task>/ reference in the task command line ...
    Decrement the number of reference to <task>
    If the number of reference to <task> is 0, clean up the involved resource
```

**Clean Up**

**Local, remote or shared file system:**
Remove the scratch directory.

**Virtual machine instance:**
Stop the instance.

**Container:**
Stop the container.

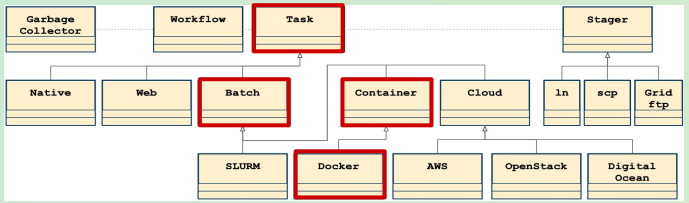


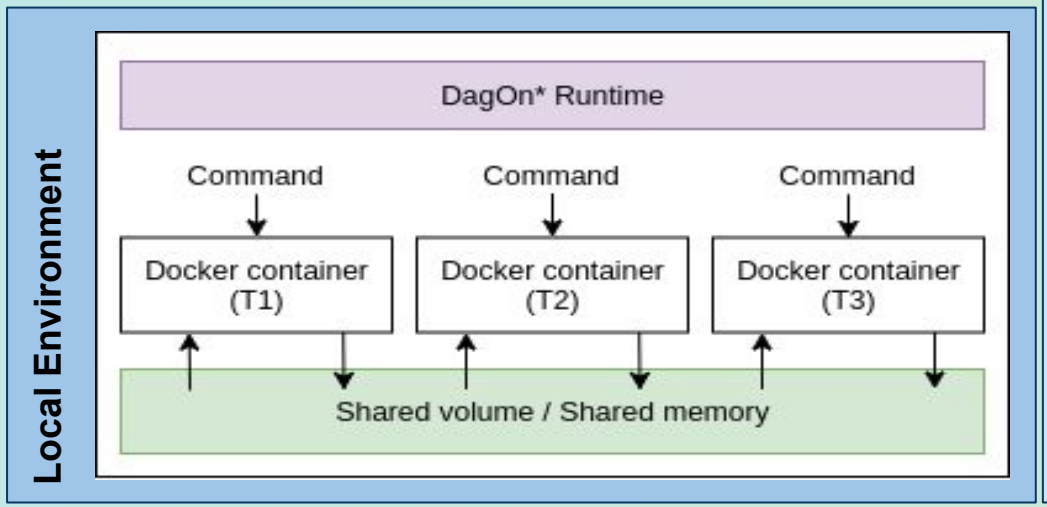


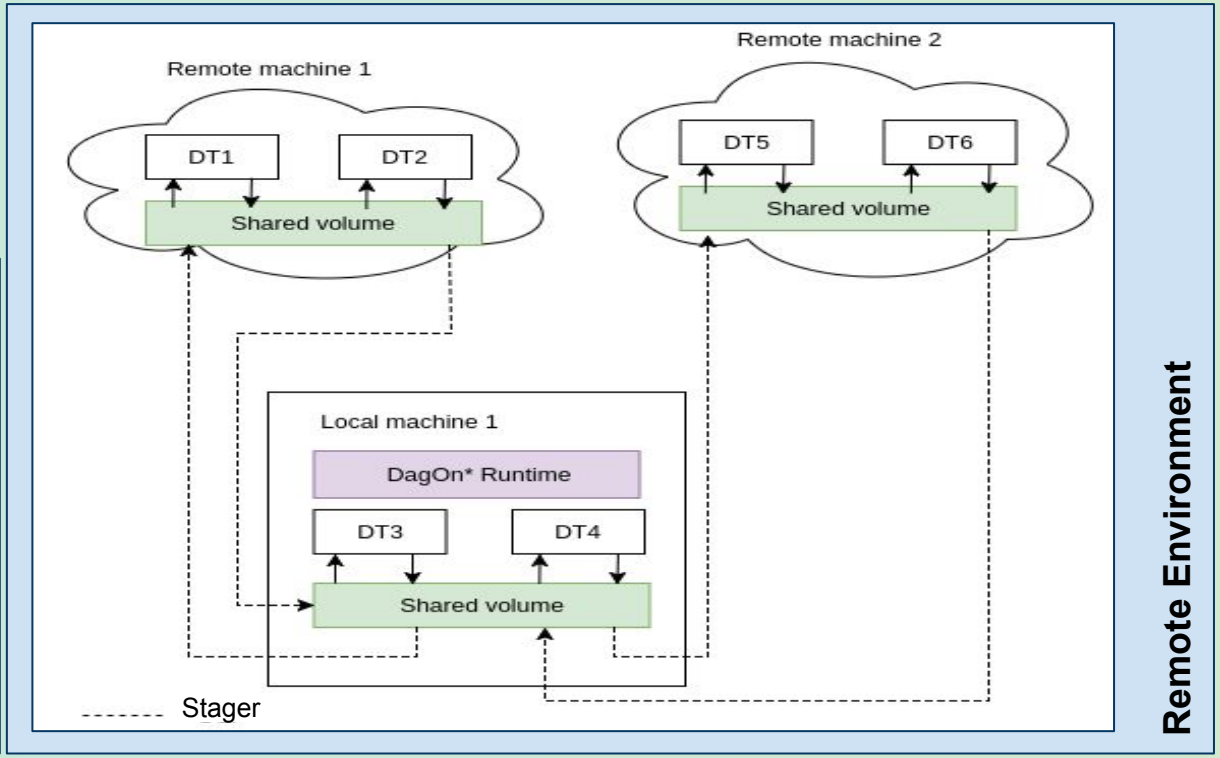

https://github.com/dagonstar/

# Container tasks

- Deployable in any Docker machine.
- Share a volume with the operating system host file system.
- If the tasks are on the same machine, the data transfer is done using shared memory.
- In a remote environment, data is copied to the volume shared between containers.



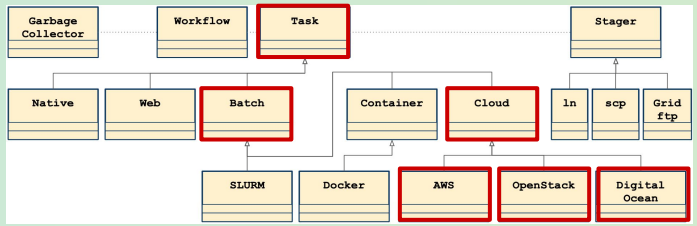



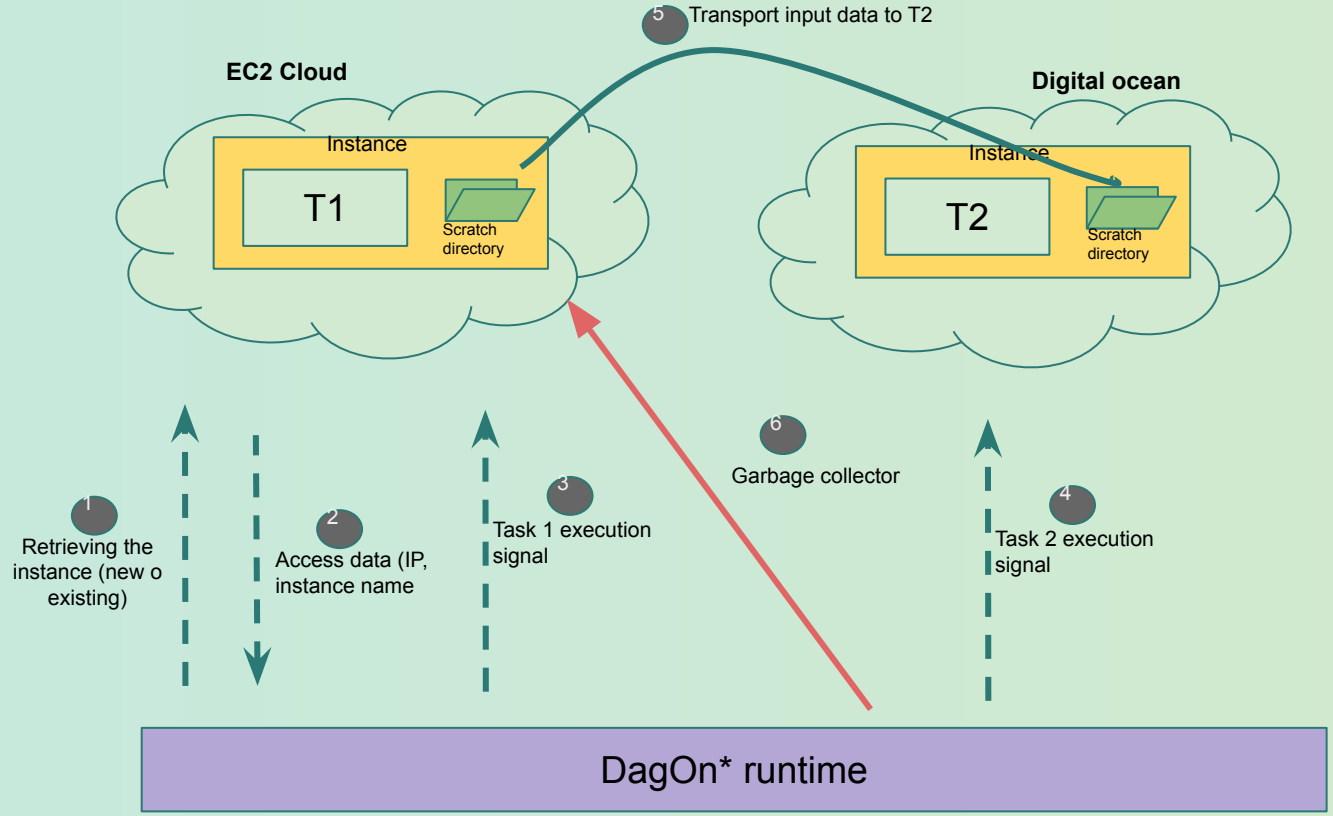



https://github.com/dagonstar/

# Cloud tasks

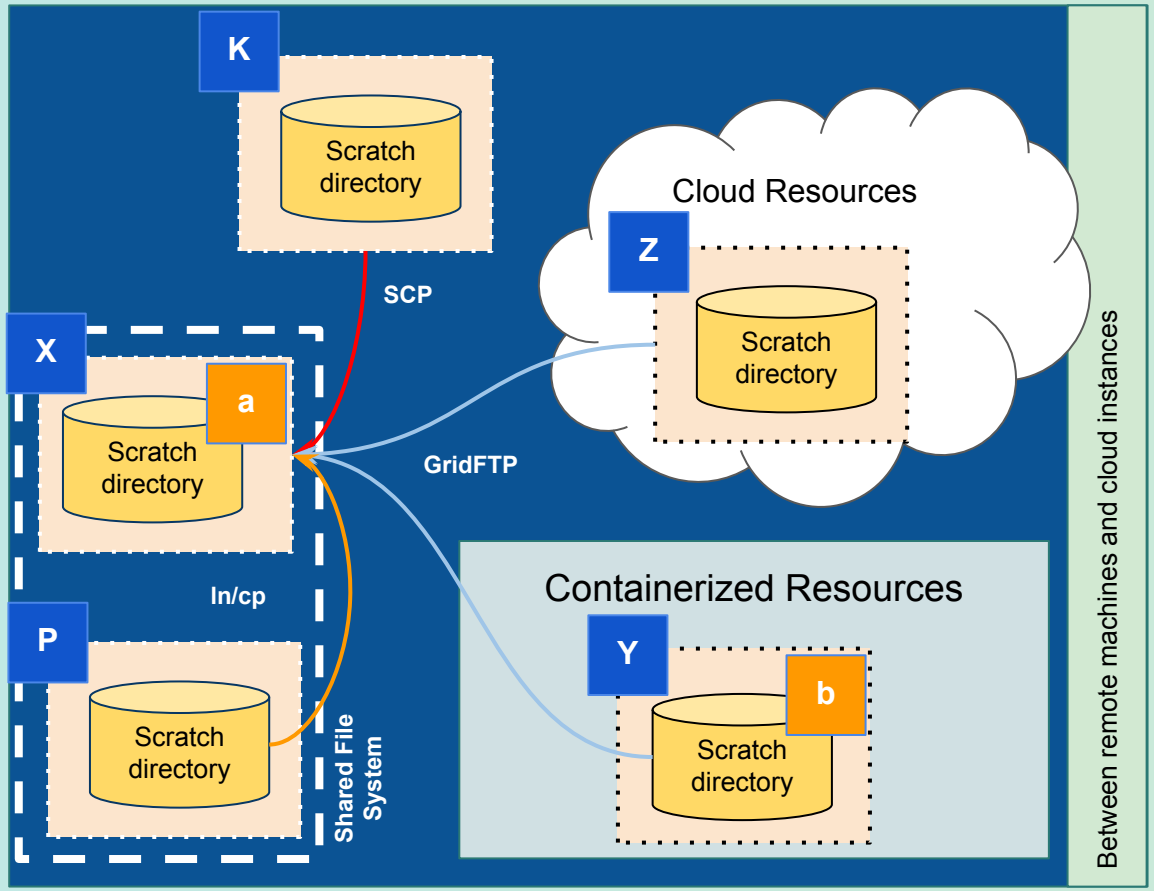
- Deployable in private, public and hybrid clouds.
- Define programmatically the flavour and configuration of the instance.
- Interoperable with other types of tasks (batch, containers, etc).
- SSH is used to make the DagOn app controlling the virtual machine instance.
- Data is transferred between tasks using the Stager component.
- Leverage on Apache Libcloud
- Tested with:
  - AWS
  - OpenStack
  - Digital Ocean
  - **Google Cloud**



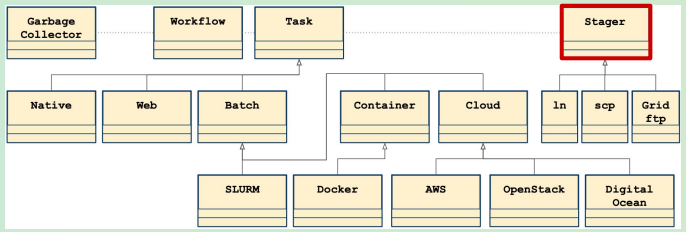

https://github.com/dagonstar/

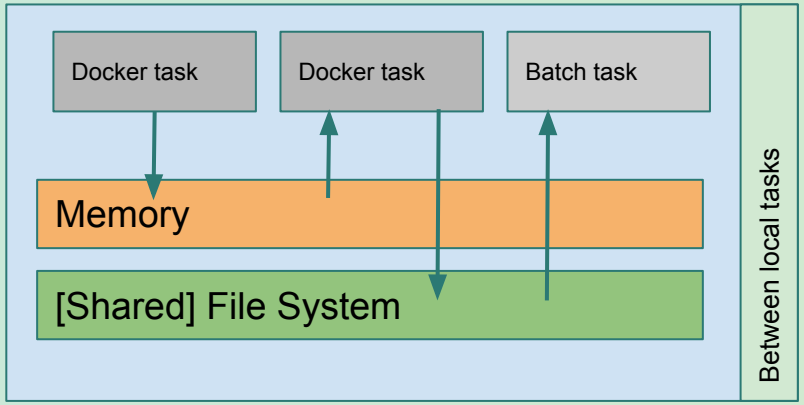
# Staging




```
globus-url-copy -vb -p 4 gsiftp://X/tmp/a/f1 gsiftp://Y/tmp/b/f2
```

Globus Connect Server



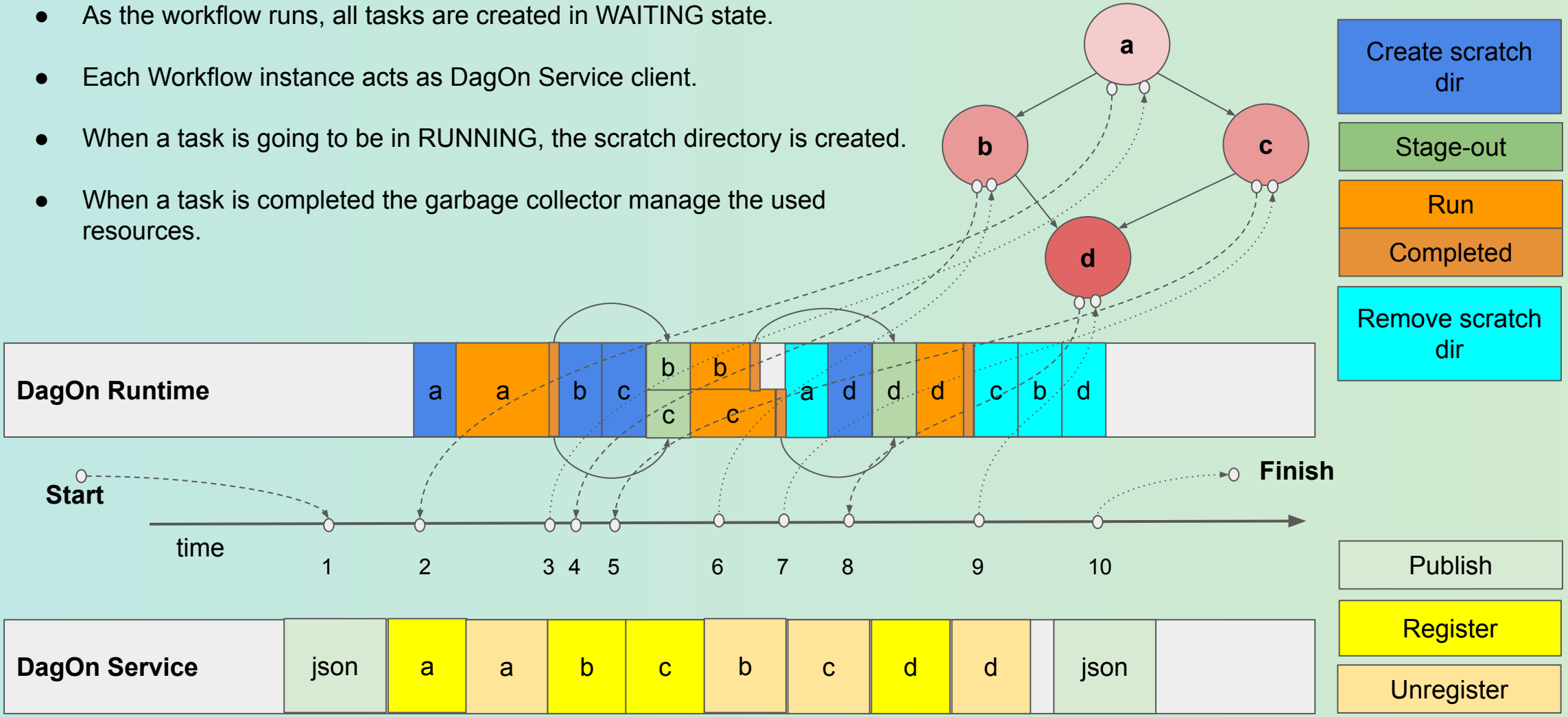




- Manages the data movement between all type of tasks.
- Fallback strategy:
  a. GridFTP
  b. Secure Copy
- Local tasks: memory, [shared] file system.

https://github.com/dagonstar/

# Application lifecycle

- As the workflow runs, all tasks are created in WAITING state.
- Each Workflow instance acts as DagOn Service client.
- When a task is going to be in RUNNING, the scratch directory is created.
- When a task is completed the garbage collector manage the used resources.



# A real-world (workflow) app!

The application is in routinary production on dedicated HPC resources (HPC-GPU BlackJeans, 650 CPU cores, 1 PB long term storage, http://rcf.uniparthenope.it )

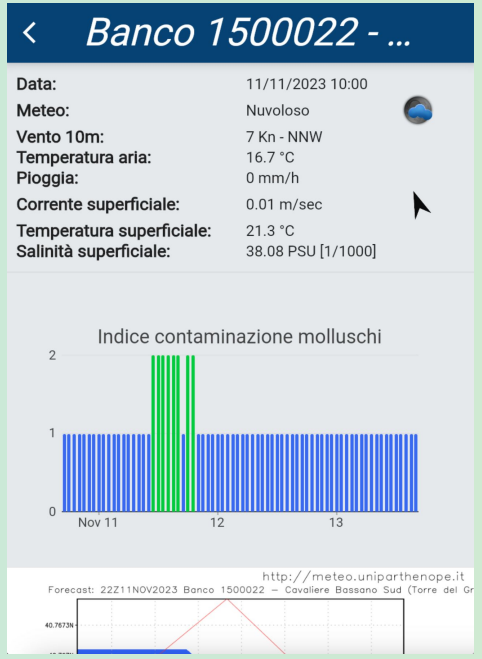
*Almost* fail-safe.

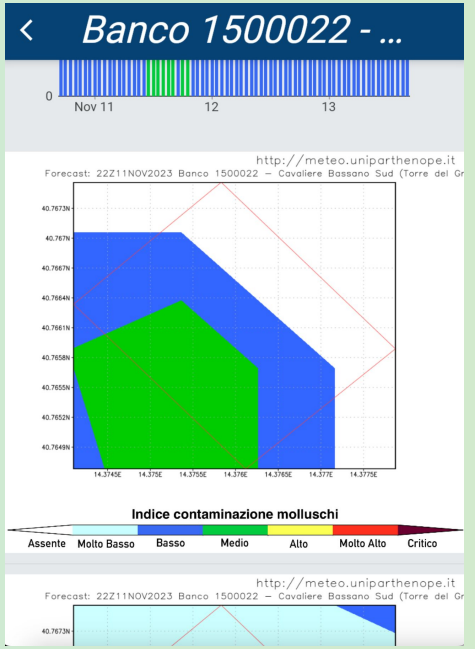
Used by the local healthcare agencies of the Campania Region, Italy.

Dissemination:

- Technical web portal
- Progressive Web Application
- Opendap Server
- Http
- Web APIs



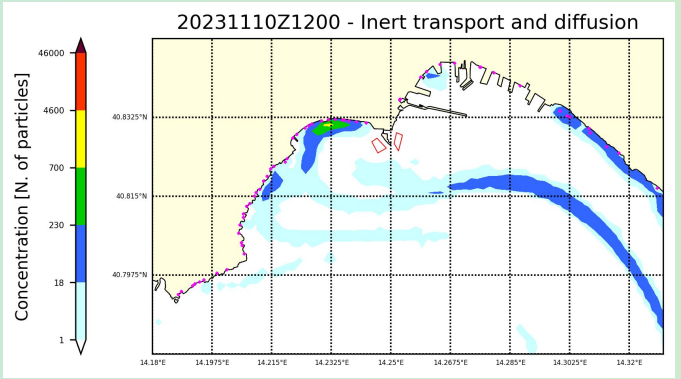




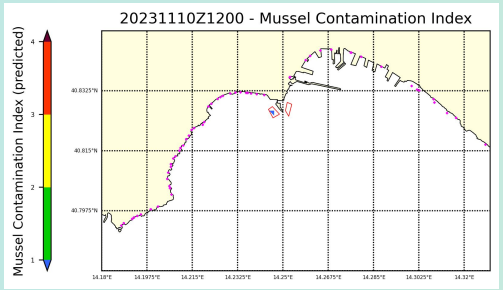

http://meteo.uniparthenope.it/mytilex/

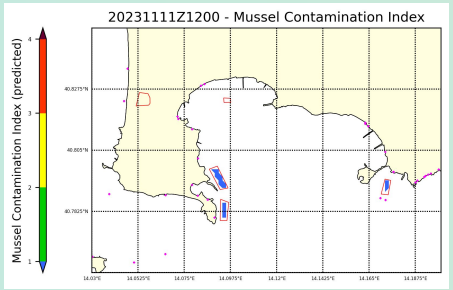
# Conclusion




The MytilEx workflow application is definitively a success story:

- It helps the local healthcare administration to avoid human gastroenteric human disease saving social costs.
- It runs basically uninterrupted since 2012 (whe have archive data since 2018)
- *Almost* failsafe: time to recovery after a catastrophic event (full storage loss), less than 72 hours.



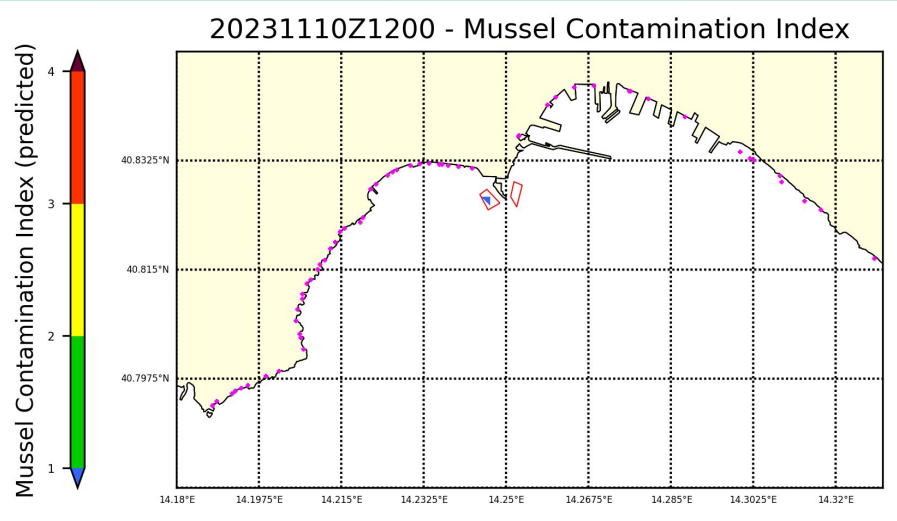



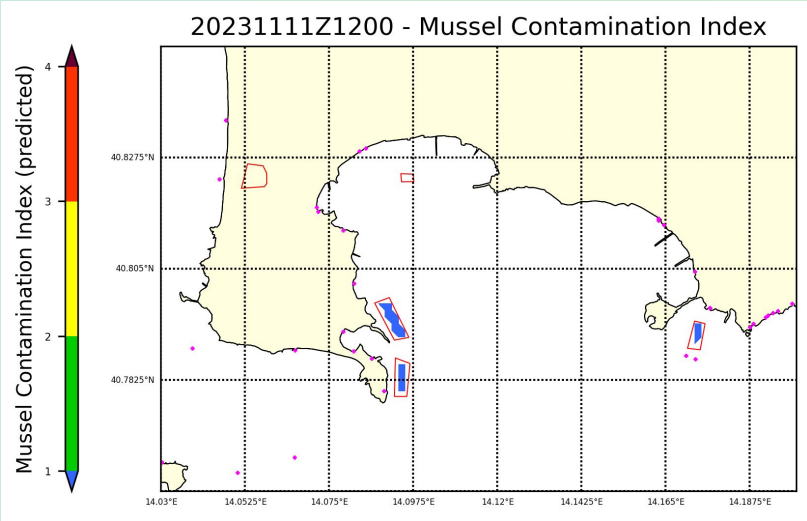

- Runs as DagOnStar application: DagOnStar has been designed for operational computational environmental science applications. It is open source since 2018.
- The WaComM++ transport and diffusion model has already used for different applications (ADMIRE use case, real life on demand search and rescue, real world accidental pollutant spill assessment)
- The AIQUAM artificial intelligent water quality model is the application's younger building block, but it will be used for other applications (prediction of bathing quality in the touristic marine areas of the Campania Region)

# Conclusion

The MytilEx workflow application is definitively a success story:

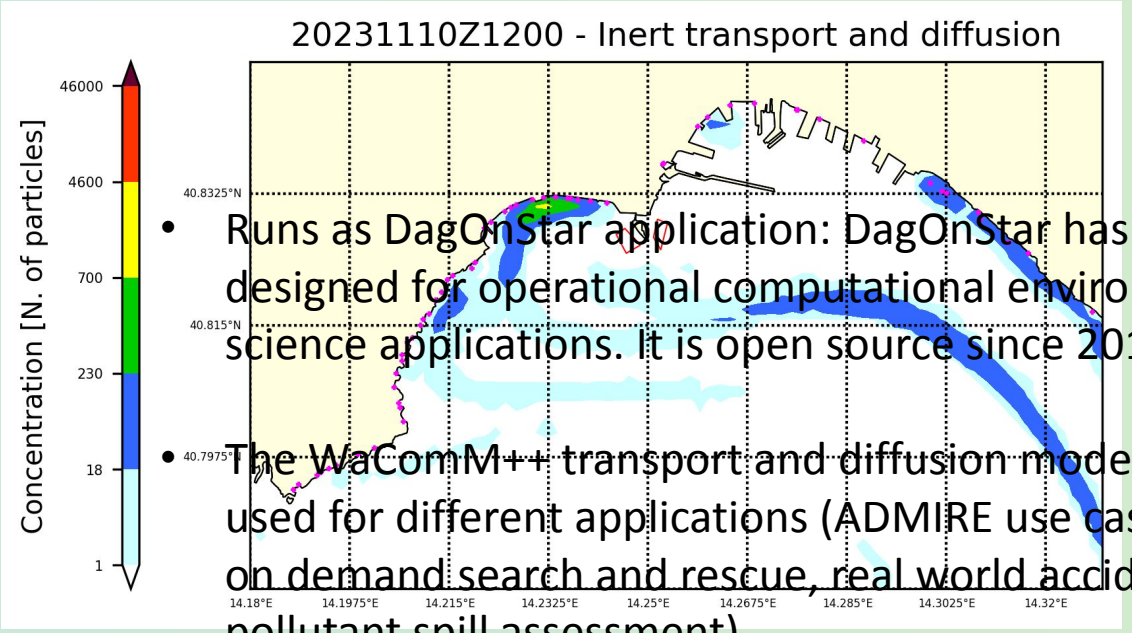
- It helps the local healthcare administration to avoid human gastroenteric human disease saving social costs.
- It runs basically uninterrupted since 2012 (whe have archive data since 2018)
- *Almost* failsafe: time to recovery after a catastrophic event (full storage loss), less than 72 hours.



# Conclusion

- Runs as DagOnStar application: DagOnStar has been designed for operational computational environmental science applications. It is open source since 2018.
- The WaComM++ transport and diffusion model has already used for different applications (ADMIRE use case, real life on demand search and rescue, real world accidental pollutant spill assessment).





- Runs as DagOnStar application: DagOnStar has bee designed for operational computational environm science applications. It is open source since 2018.
- The WaComM++ transport and diffusion model ha used for different applications (ADMIRE use case, on demand search and rescue, real world accident pollutant spill assessment).


- The AIQUAM artificial intelligent water quality mo application's younger building block, but it will be other applications (prediction of bathing quality in touristic marine areas of the Campania Region)



http://meteo.uniparthenope.it/mytilex/