GeoNimbus: A serverless framework to build earth observation and environmental services

Dante Sánchez-Gallegos, Diana Carrizales, Catherine Alessandra Torres Charles, Alejandro De La Rosa Zequeira, Jose Luis Gonzalez-Compean and Jesus Carretero

Universidad Carlos III de Madrid

dantsanc@pa.uc3m.es

EURO-PAR CONFERENCE 2024

1st workshop about High-Performance e-Science

August 26

Context

- **Earth Observation Systems (EOS):**
  - Spatio-temporal studies
  - Large volumes of data
  - Multiple applications processing data

- **Serverless computing**:
  - Cloud model
  - Service provider automatizes:
    - The allocation of resources
    - Deployment of applications/functions
    - Execution based on events
    - Scaling
    - Monitoring
  - Developer: pushes a snippet of code.
  - Examples: AWS Lambda, Google Functions, etc.

Objective:

Create serverless systems to automatize the management of EOS.

Benefits:

Researches only have to focus on creating code and not dealing with infrastructure.

Serverless computing in Earth Observation: challenges and issues

- Most of the serverless environments are on the public cloud.
- The usage of the public cloud have drawbacks to considerate.

Data confidentiality

Vendor lock-in

Loss of control

Data accessibility during outages

Latency to store and access data

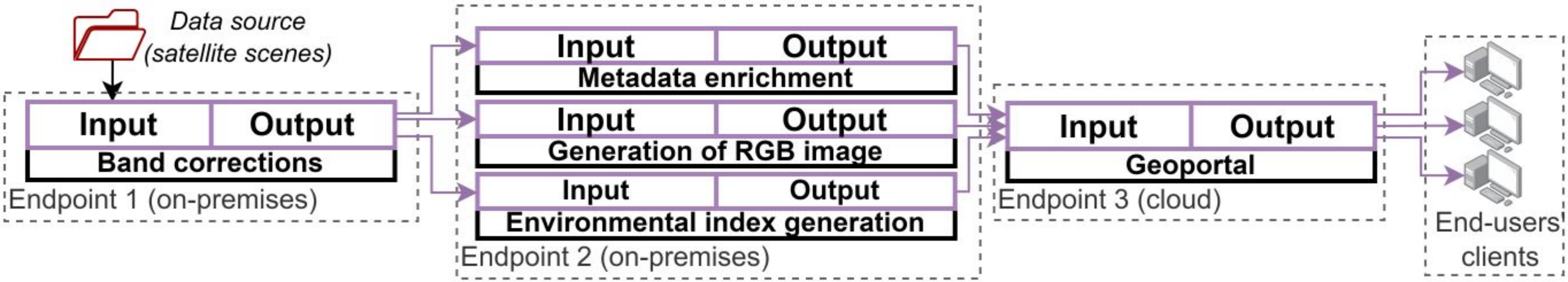
- **Possible solution**:
  - Distribute through different infrastructures on the edge and the fog.

Related work

Work	Scope	Environment of execution	Design	Data management	Execution	Monitoring	Auto-scaling	Dataflows
AWS Lambda	Serverless backend	Public cloud	Functions	S3	Based on events	Automatic	Based on demand and a budget	Using Kafka and S3
Lithops	Serverless backend	Multicloud	Functions	Cloud object storage	Delegated to service provider	Automatic	Based on a demand	By design
Globus compute (funcX) - 2022	Serverless framework	Any computer	Python functions	Object storages	Based on invocations	Automatic	In response to workload	Parsl and/or ProxyStore
Proposed Method	Serverless con the computing continuum	Any computer	High-level configurations	Based on a content delivery network	Events based on inputs	Automatic	To reduce bottlenecks	By design

GeoNimbus

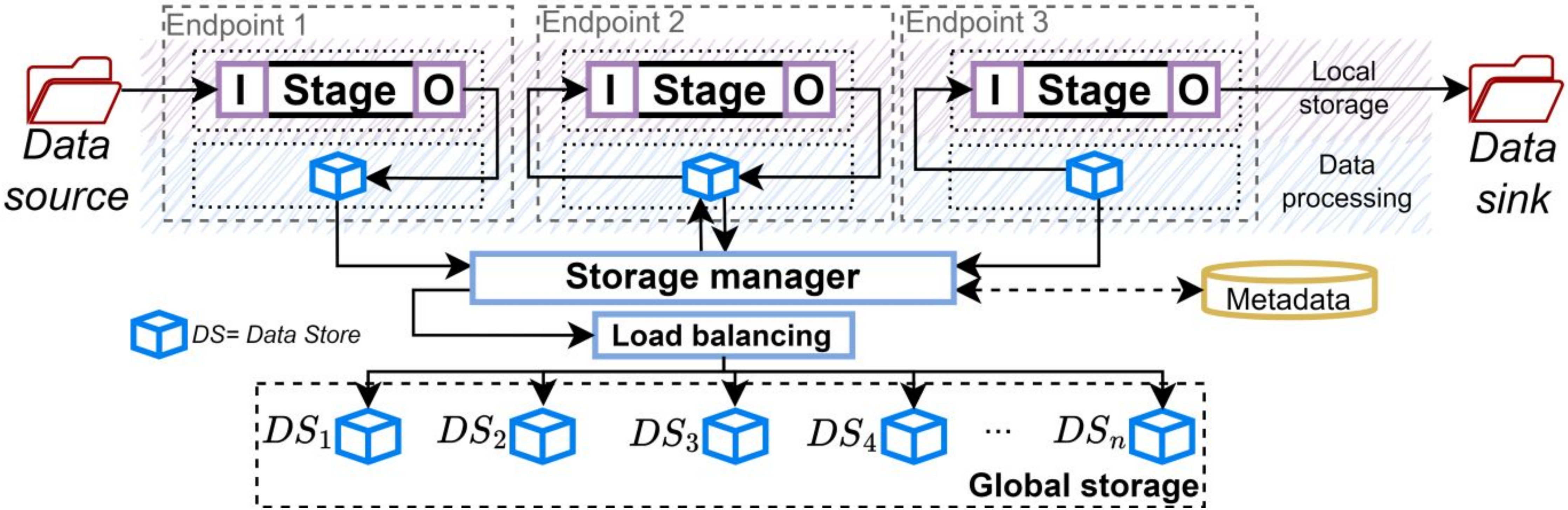
- A **serverless framework** to manage spatio-temporal earth observation services (EOS).
  - *What does GeoNimbus do?*
    - Design, deployment, execution, scaling, and monitoring
    - Data management
    - On any on-premise or cloud infrastructure.
    - Design time: using a design-driven model
    - Deployment and execution times: as serverless units connected using I/O interfaces.
    - Reduce the complexity to create and manage an EOS.



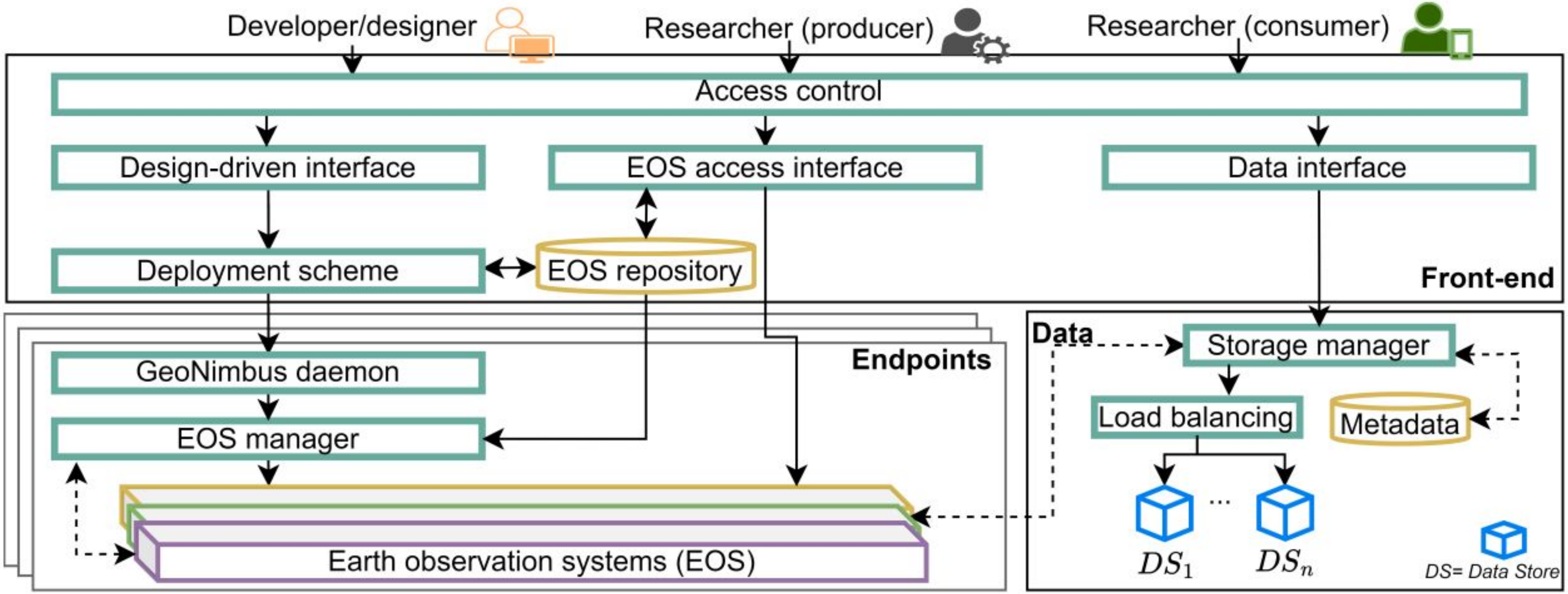
Example of a GeoNimbus EOS.

Data management

- Data is moved through stages using a **wide-area storage system**.
  - **Data stores (DS)**: a virtual storage unit that stores the data required and produced by stages (their applications) in a system.
    - Local or global
  - **Storage Manager**: metadata, load-balancing, reliability.

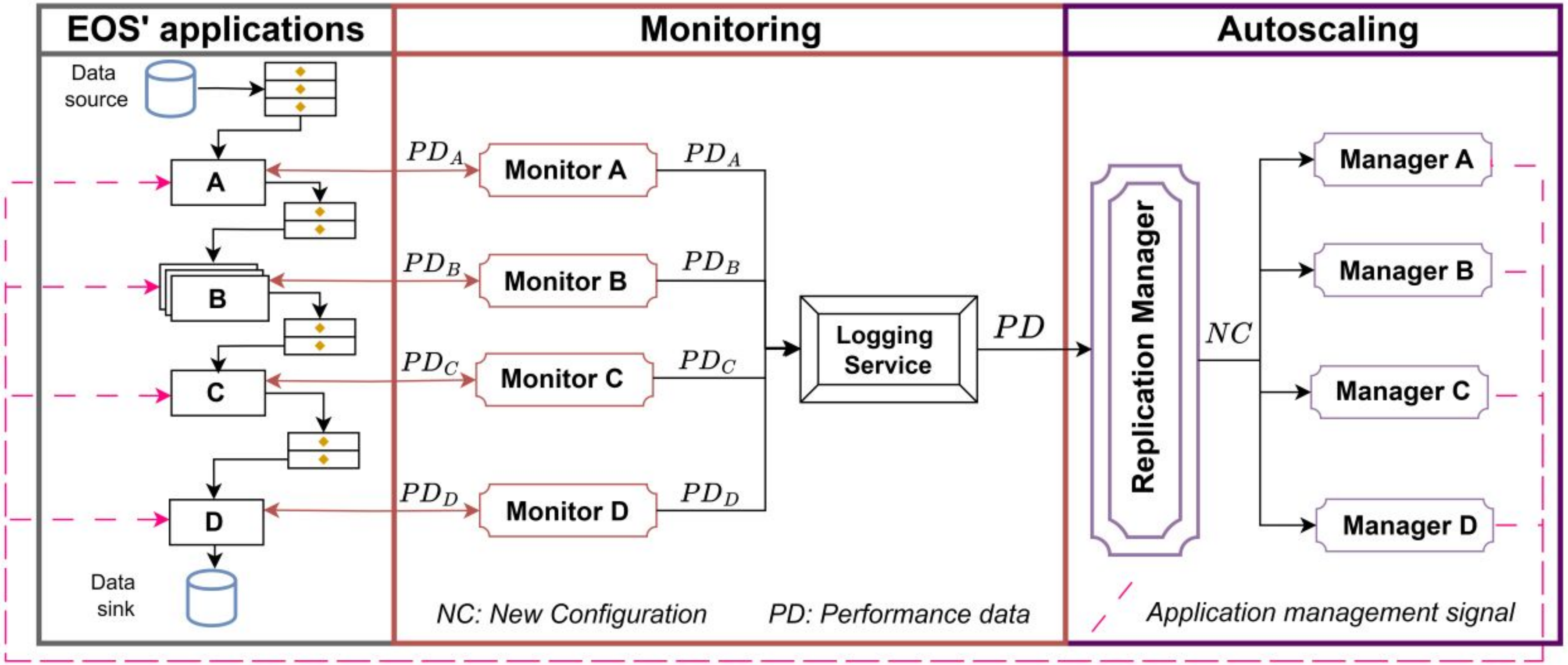


Design principles: architecture



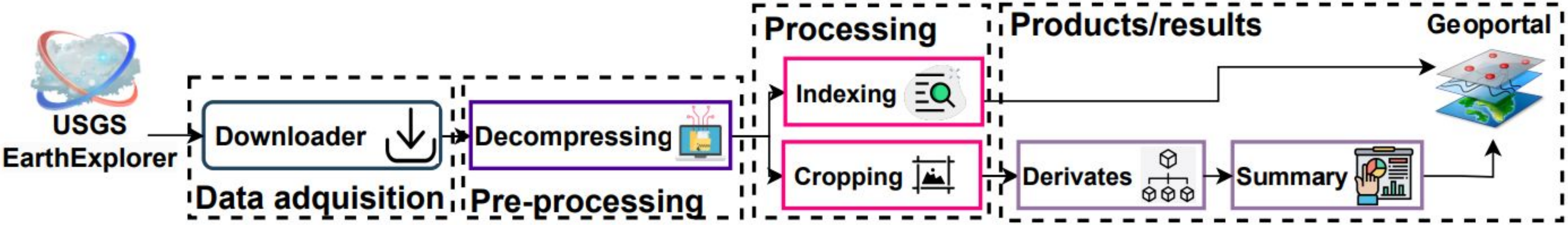
Design principles: autoscaling and monitoring

- Designers can create **manager/worker patterns**.
- The number of workers is chosen:
  - During design time using a configuration file.
  - During execution time by monitoring the performance of applications.
- Bottleneck identification:
  - $Btl = MINTHPOS(thpApps)$
- For each bottleneck a new worker is added.
  - *Restriction: Number of workers < cores*
  - If performance decreases, the system rollback to a previous configuration.



Evaluation. Case study: services to analyze changes in water resources in Mexico

- We conducted a case study based on a system for processing LandSat8 images and generating normalized difference water indexes (NDWI).
- *Zone of study:* Lake Cuitzeo, Michoacan, Mexico.
- *Temporal:* 2013 – 2024.



Results: performance evaluation

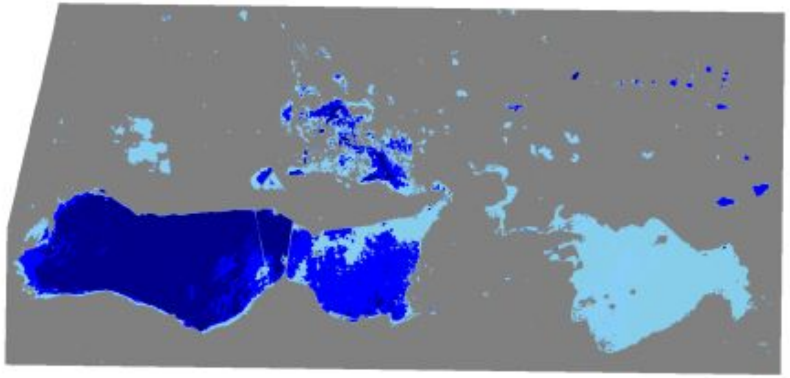
- This experiment aims to show how GeoNimbus scales when managing large workloads.
- Improvement of until 85.10% when passing from one worker to 32.

(a) 32 images.
39 GB

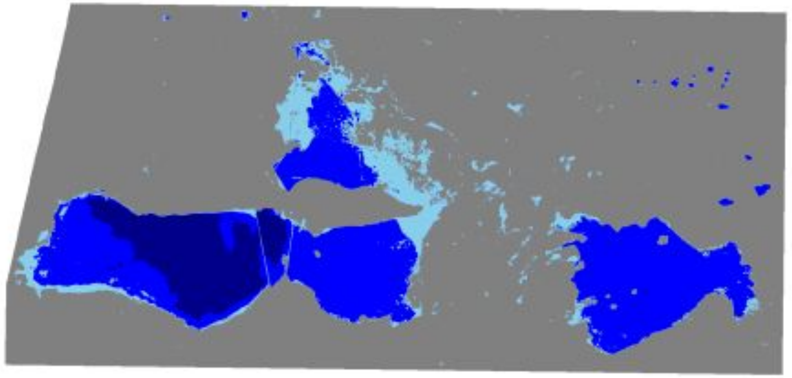
(b) 213 images.
251 GB

Results: temporal study

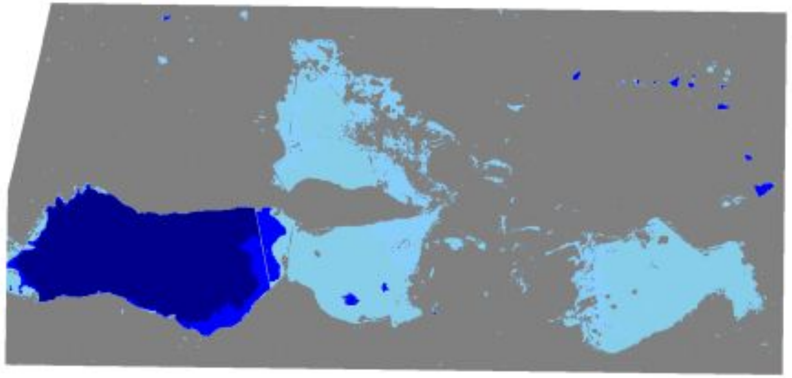
- NDWI$_{red}$ produced with a combination of the bands 7 (SWIR 2) and 4 (red).}



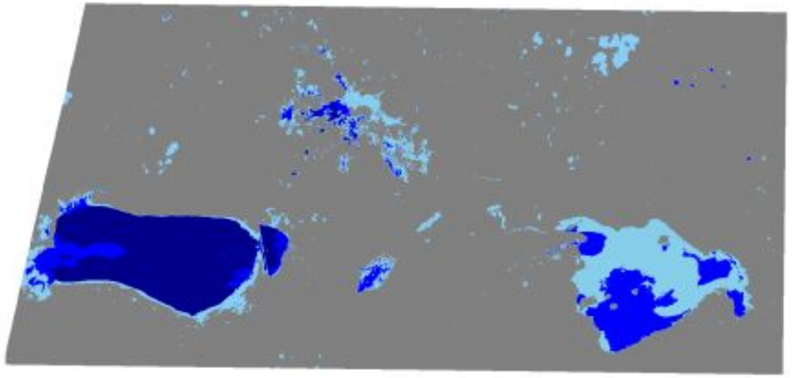
(a) 2013.



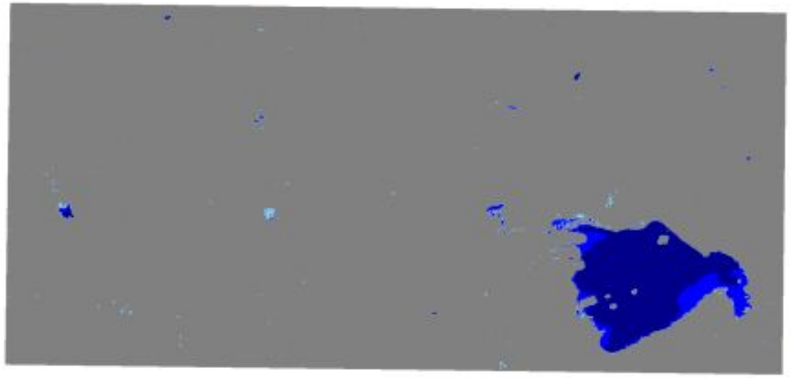
(b) 2015.



(c) 2018.



(d) 2021.

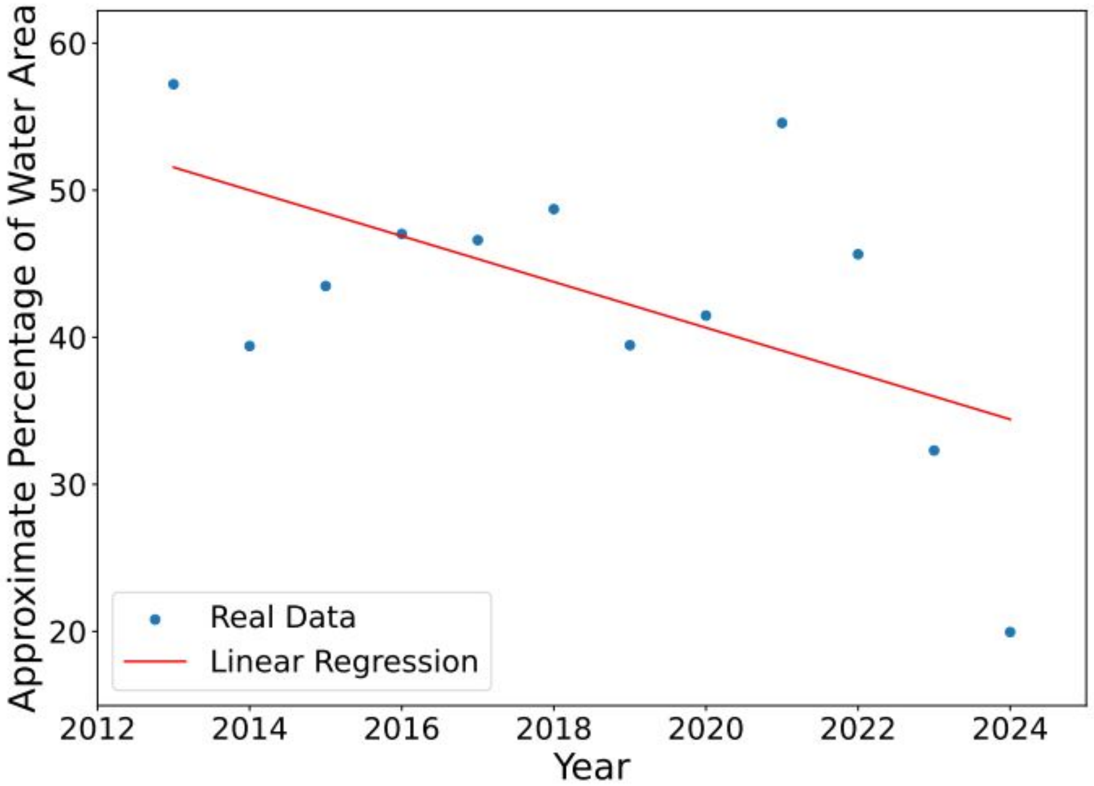


(e) 2024.

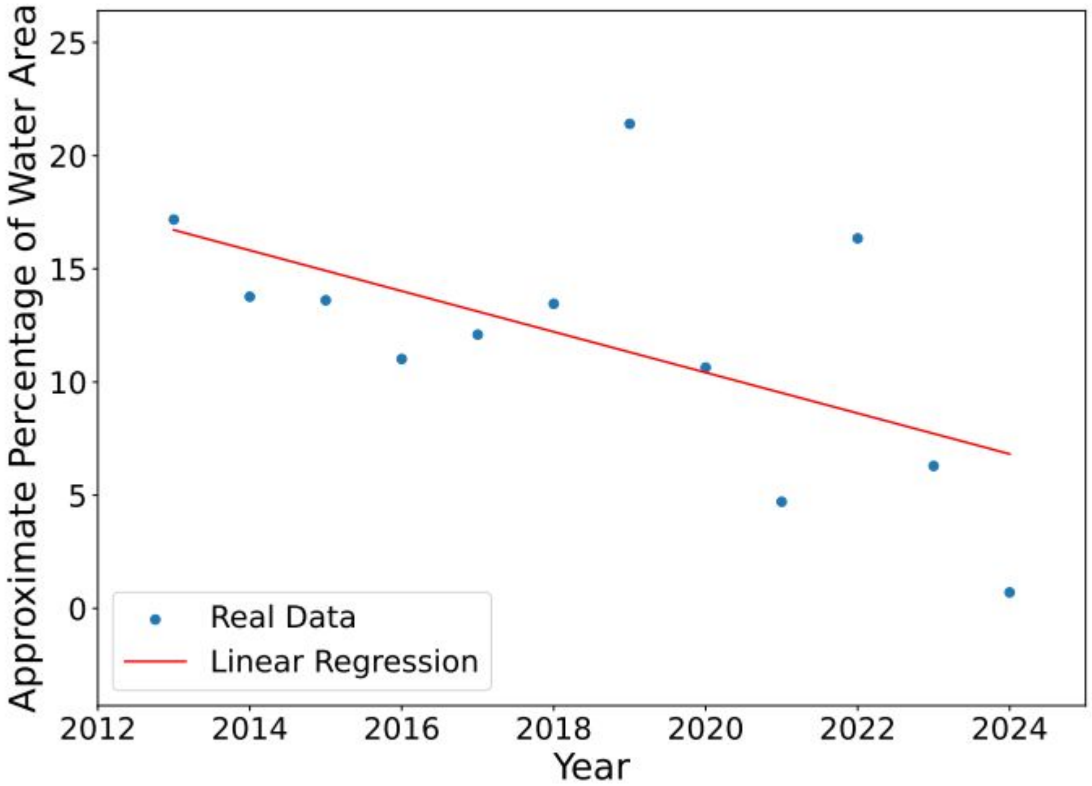
Results: temporal study

- *Approximate percentage of water area*:

$$\frac{Number\ of\ pixels\ >\ umbral\ (0.65)}{Total\ number\ of\ pixels} * 100$$



(a) Path 27, Row 46.



(b) Path 28, Row 46.

Conclusions and future work

- We presented GeoNimbus, a framework for designing EOS that follows the design principles of serverless computing.
  - Manages the deployment, scaling, monitoring, and execution of functions and applications.
- GeoNimbus's goal is to reduce the complexity of creating serverless EOS.
- We conducted a case study based on the processing and analysis of LandSat8 images corresponding to Lake Cuitzeo.
  - Performance: GeoNimbus decreases by almost 85% the time required to process the data in comparison with a non-parallel configuration.
- In future work, we plan to perform other environmental studies to measure changes in vegetation and urban areas.

GeoNimbus: A serverless framework to build earth observation and environmental services

Dante Sánchez-Gallegos, Diana Carrizales, Catherine Alessandra Torres Charles, Alejandro De La Rosa Zequeira, Jose Luis Gonzalez-Compean and Jesus Carretero

Universidad Carlos III de Madrid

dantsanc@pa.uc3m.es

EURO-PAR CONFERENCE 2024

1st workshop about High-Performance e-Science

August 26