MeshStore: Creating Services on the Edge-Fog-Cloud with Data Containers and Globus Compute

Dante D. Sanchez-Gallegos, Diana Carrizales, J. L. Gonzalez-Compean, Jesus Carretero
dantsanc@pa.uc3m.es

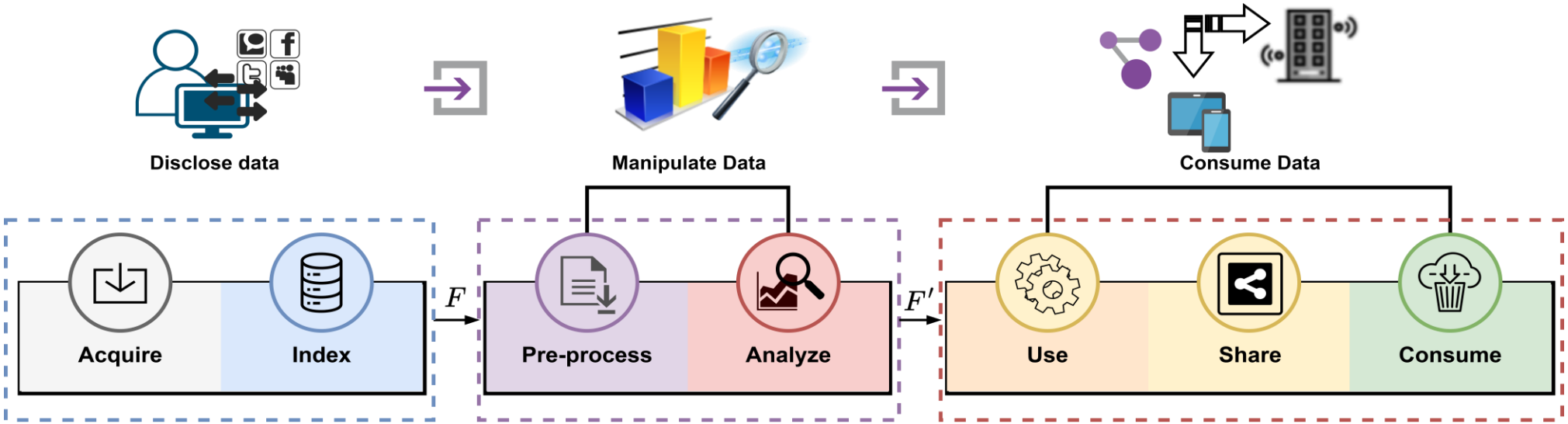
ParslFest 2023
October 19-20, 2023

Motivation

- Many **eScience** problems require very complex and **data intensive cooperation** among multidisciplinary actors.
  - Some challenges are:

- Volume 
- Veracity 
- Data distribution 
- Heterogeneity/
- Variety 

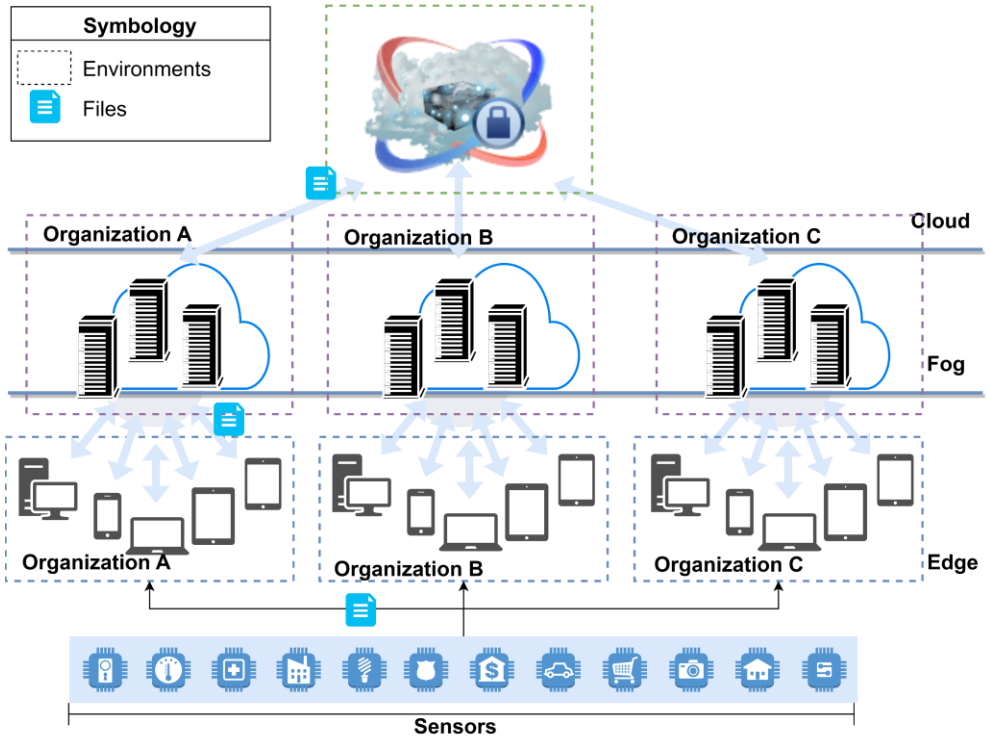
- To cope with this, workflow managers usually create **dataflow processing schemes** statically connected by using mostly input/output files as **synchronization** points following data dependencies.

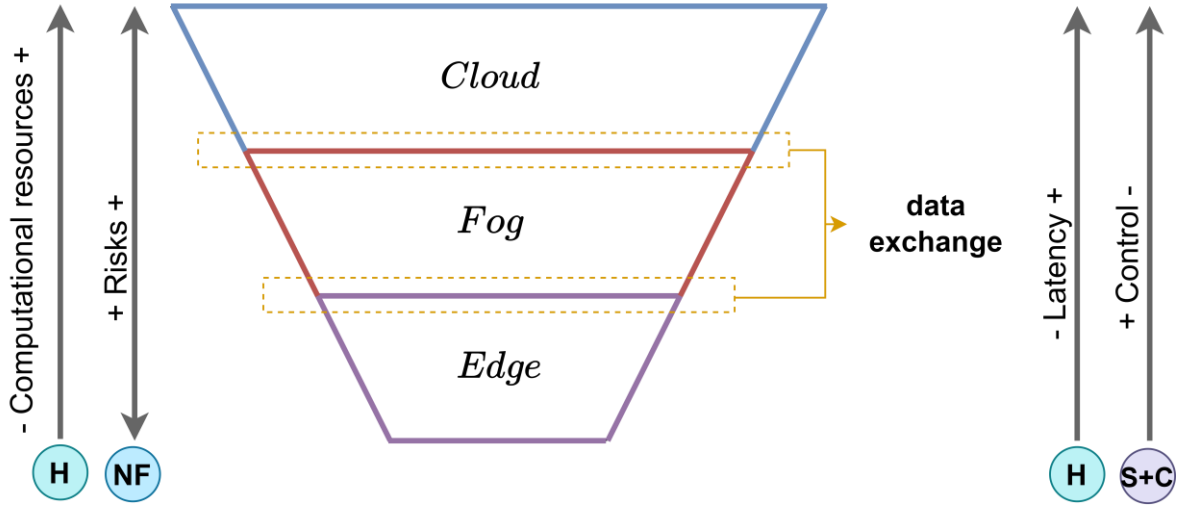


Multi-tier serverless architectures

- Multi-tier serverless architectures allows to create a geographically distributed data service.
  - Deployed dynamically following applications needs

- Challenges:
  - **Latency** between infrastructures.
  - Storage **Capacity** (persistent, volatile)
  - **Synchronization** and global availability of data.
  - To manage the **input/output operations**.
  - Enforcing **Non-Functional Requirements** for the data.

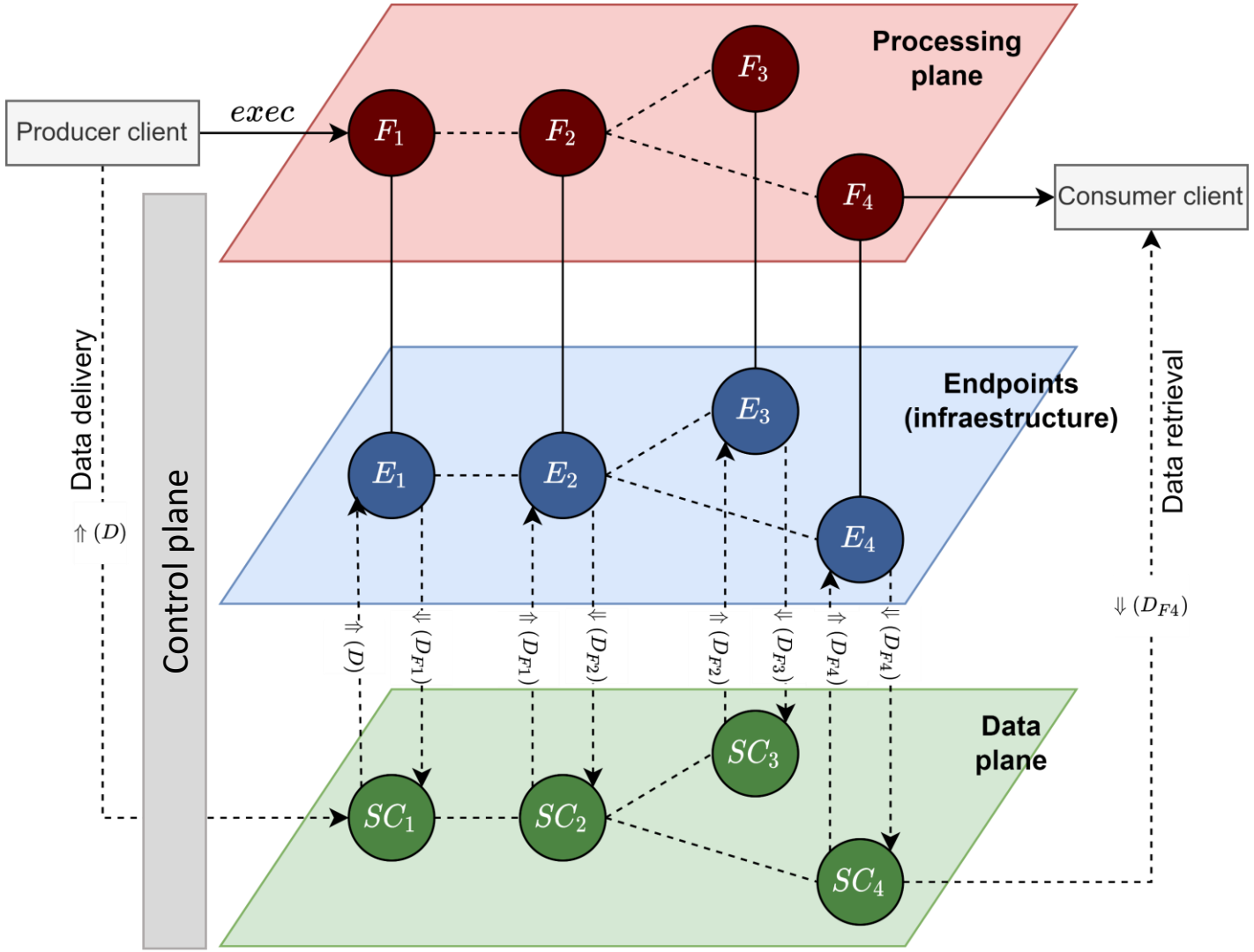




General architecture

- MeshStore is organized as an overlay architecture composed of 4 layers:

1. **Processing plane**: serverless functions. Implemented using funcx (Globus Compute)
2. **Data plane**: an in-memory CDN storage composed of storage containers.
3. **Control plane**: Deployment, policies, control points
4. **Endpoint layer**: infrastructure to deploy data and process data.

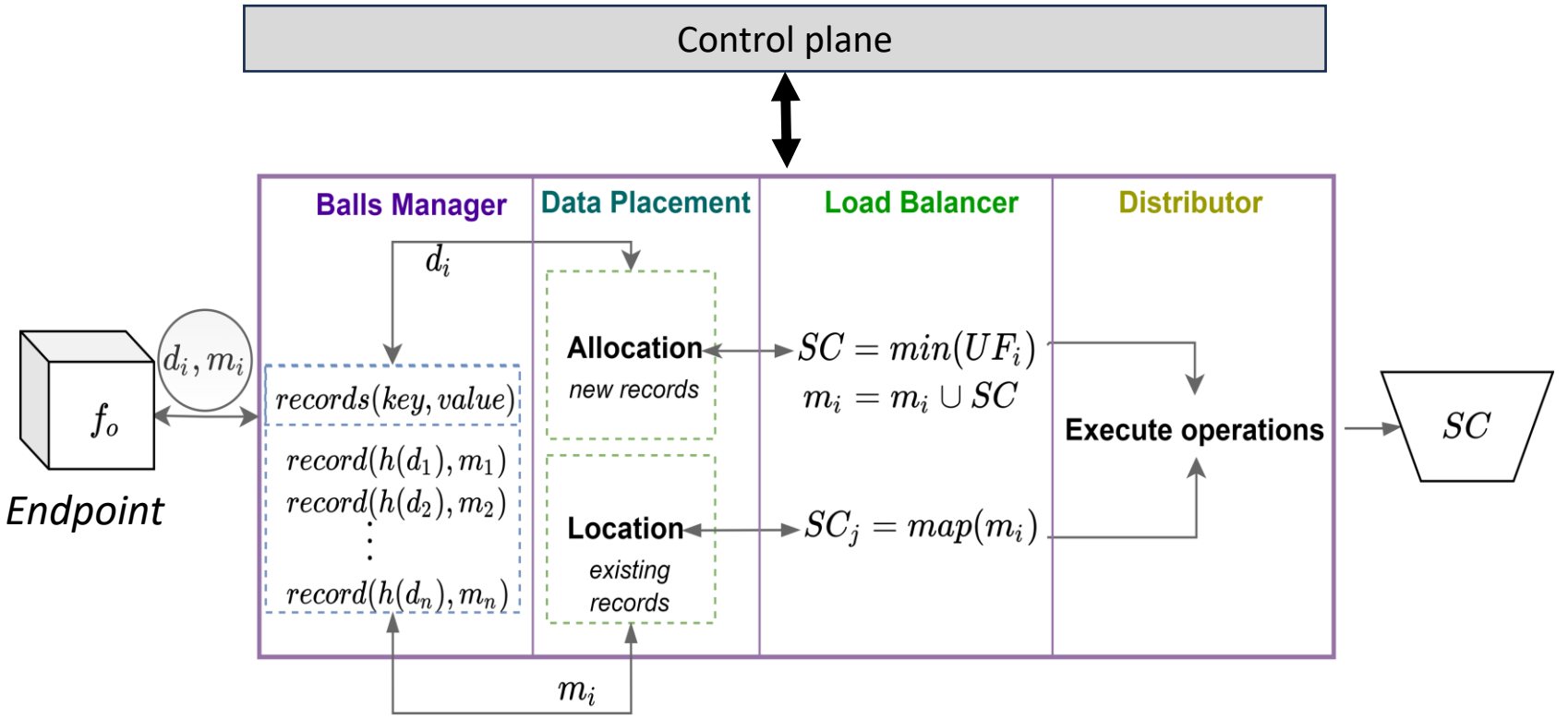


Data allocation/location scheme

- The allocation/location of data is based on a balls-into-bins metaphor.

- The data placement (allocation) is based on a two choices load-balacing algorithm with an **utilization factor (UF_i)**.
  - SC_j=min($UF_i = 1 - \left(\frac{C_i - Ui}{C}\right)$)
  - U_i=SC_i usage
  - $C = \Sigma c_{i,\ i = 1...n}$
  - C_i = SC_i capacity

- Metadata maps are generated for each content (m_i) to be stored in a storage container.
  - Location, NFR, …



Experimental evaluation

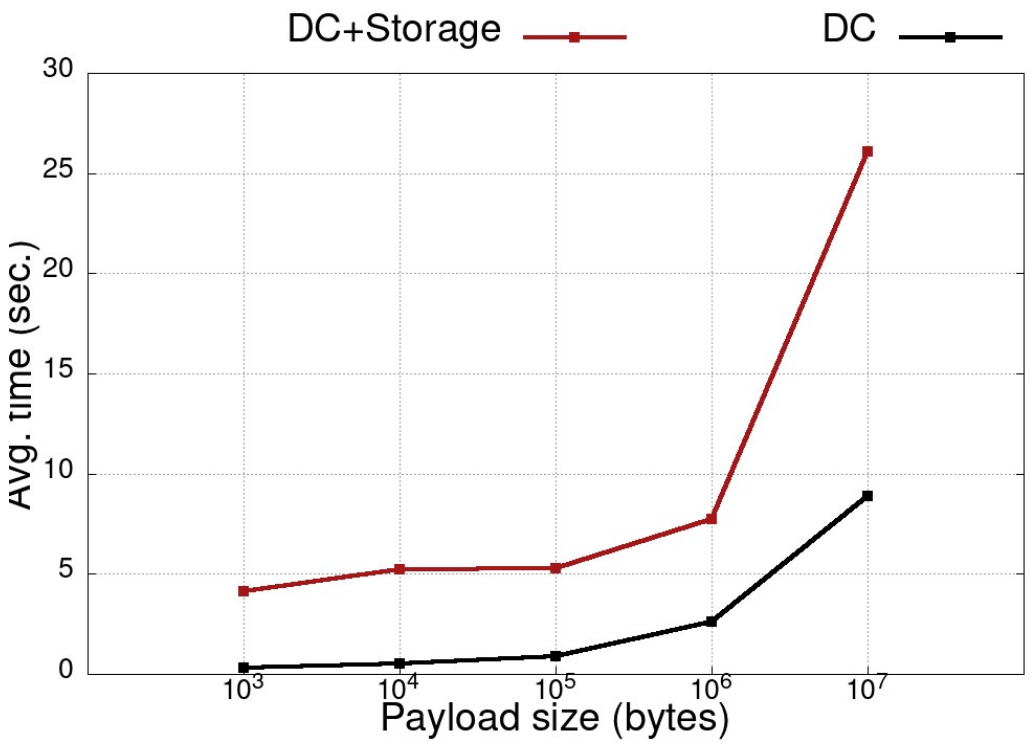
- Evaluation performed using synthetic data and real meteorological traces.

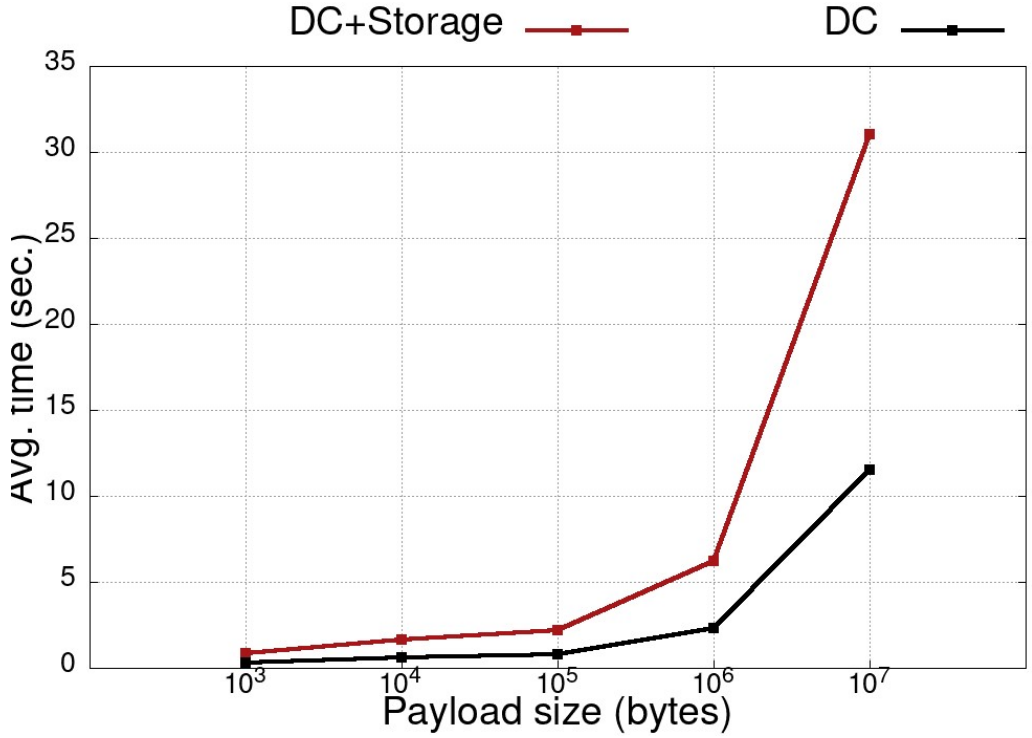
- Evaluated using simultaneously distributed infrastructure available at Mexico, Spain, and Amazon AWS.
  - Mexico. 1 edge, 3 fog.
  - Spain. 1 edge, 2 fog.
  - AWS. Shared storage instance.

- A storage mesh was created using that infrastructure.

Data movement evaluation

- **MeshStore-direct**: a direct transmission of the data (Point to Point data transmission).
- **MeshStore-storage**: including the storage of the data for their long-time preservation on storage containers (serverless).





Conclusions

- MeshStore is based on storage structures that represent maps of storage resources available on multiple infrastructures.
- Automatically manages the data required and produced by serverless functions.
- A unified storage layer is added in a transversal manner to serverless functions.

Ongoing work

- Integration of MeshStore with a blockchain model to keep the traceability of the data and exploitation through smart contracts.
- Study of self-adaptable mechanisms to choose the number of workers and virtual containers in a storage mesh.
- Enhancing data distribution by alleviating I/O bottlenecks.
- Using ad-hoc storage deployments per workflow to enhance I/O in HPC systems

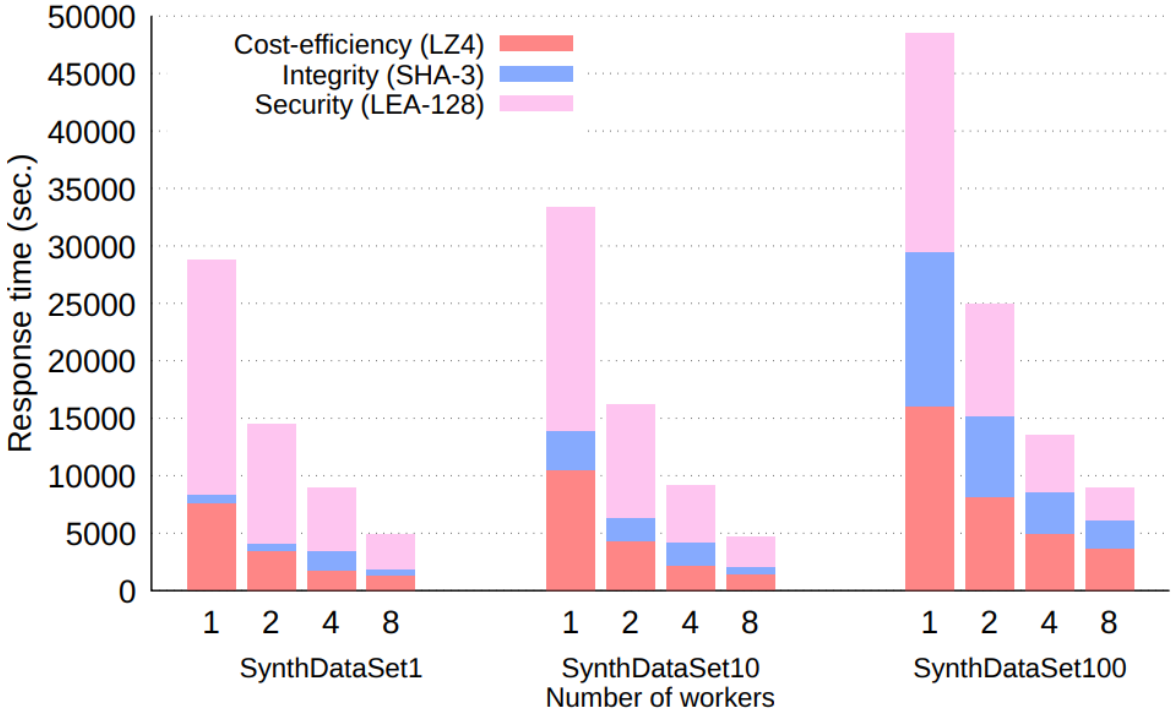
Creating Services on the Edge-Fog-Cloud with Data Containers and Globus Compute

Dante D. Sanchez-Gallegos, Diana Carrizales, J. L. Gonzalez-Compean, Jesus Carretero
dantsanc@pa.uc3m.es

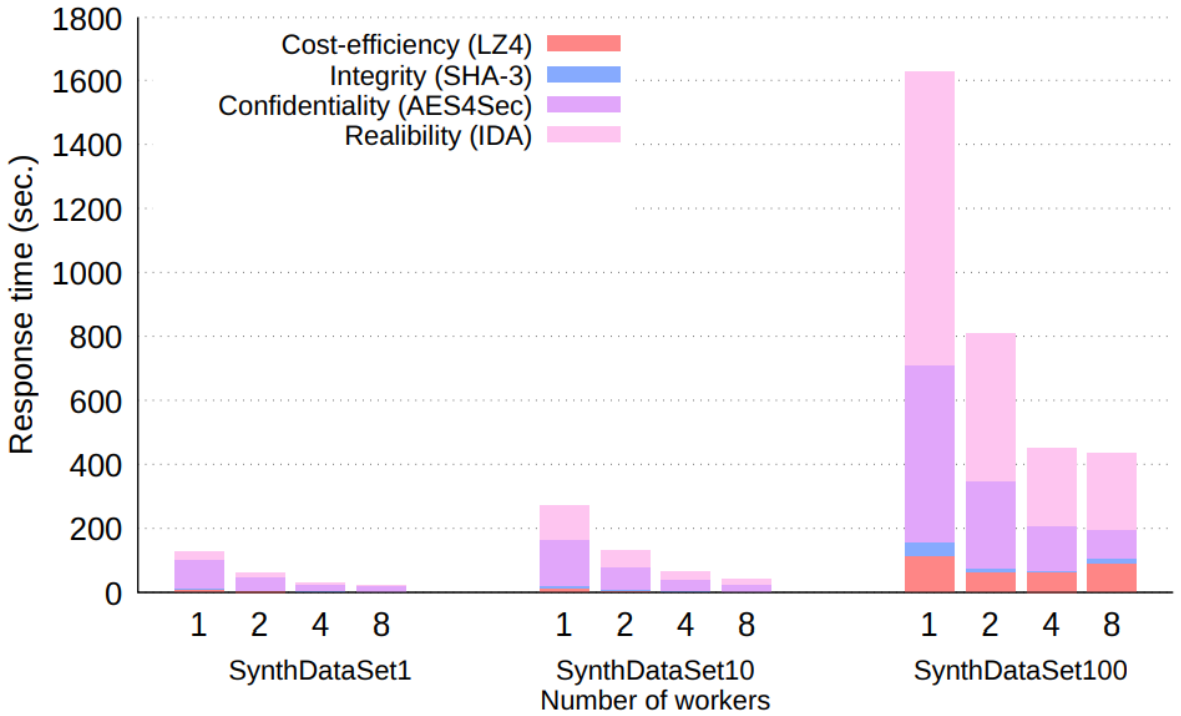
ParslFest 2023
October 19-20, 2023

Management of non-functional requirements

- We measured the **costs of adding NFR characteristics** to 100 files of 1, 10, and 100 MB on edge and fog environments.
- Bottlenecks are mitigated using parallelism patterns.



Edge machine



Fog machine