

# How to handle new devices

## ⌘ Problem

- New storage devices need to be part of the storage/memory hierarchy
  - NVRAM
  - SCM
  - ...
- These devices are closer to memory than traditional storage
  - Low latencies
  - High bandwidth
  - Byte addressable
- It does not make sense to use them as “block devices”



## ⌘ dataClay proposal

- Store data as objects like in OO programming models



# Link computation and data

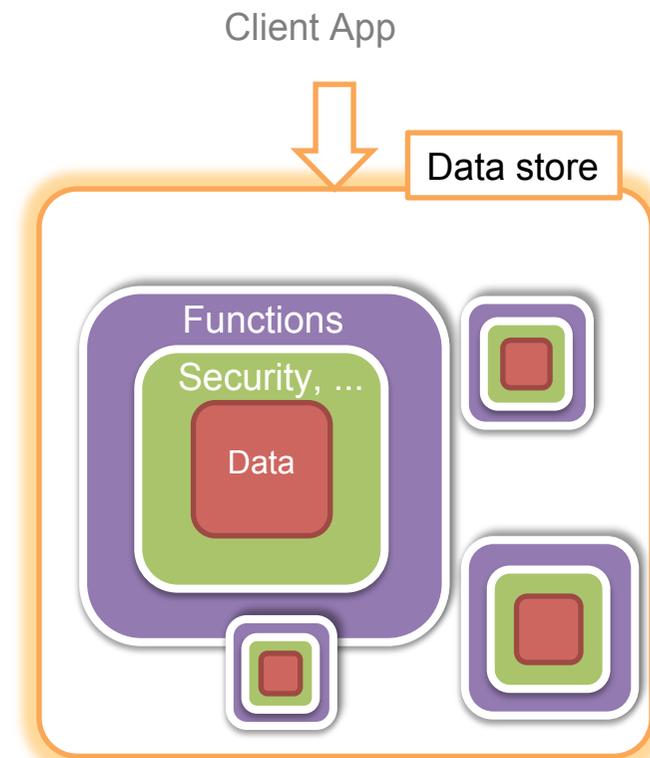
## ⌘ Problem

- Depending on the computation and data sizes
  - Computation needs to be moved to the data, or vice versa
- Moving computation to data is sometime difficult
- Moving data may “break object behavior”



## ⌘ dataClay proposal

- Include into the object
    - Data
    - **Methods**
    - **Behavior polices**
      - security, integrity, privacy, lifecycle, ...
- ... and create ***Self-contained objects***



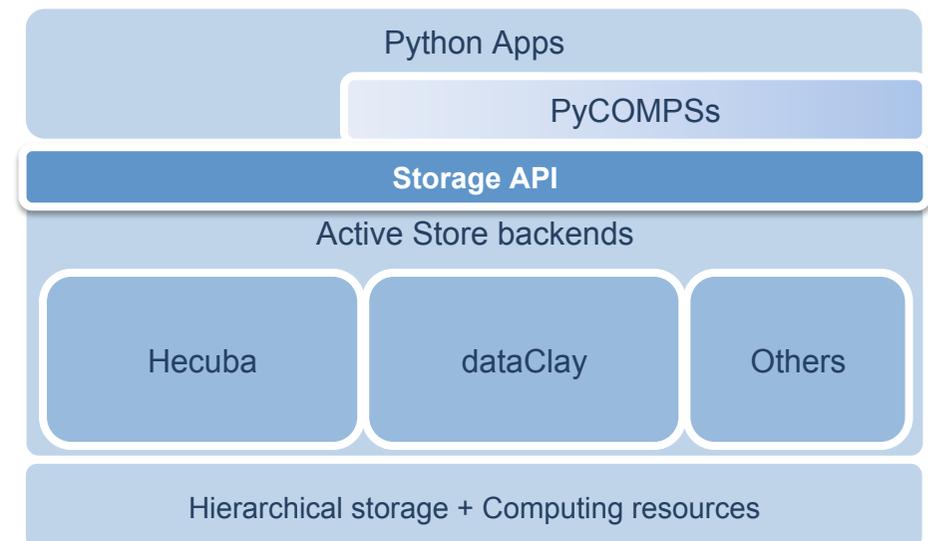
# Integration with programming model

## ⌘ Problem

- Two data models (persistent and volatile)
- Not fully integrated with programming model

## ⌘ dataClay proposal

- **Objects** as the storage abstraction **enable a single data model**
- Self-contained Objects **integrated with PyCOMPSs** enables
  - Iterators aware of data locality  
*((More in Rosa's talk))*



# Data sharing

## Problem

- Though real value comes from sharing, today there is no good solution
  - Real sharing → too dangerous
  - Data copy → owner loses control
  - Data services → too inflexible for “clients”

## dataClay proposal

- **Objects can be enriched** by third parties
  - New fields,
  - New methods
  - New implementations
- Owner still keeps control

