Frameworks and Collaborative Tools

The request (from Ruth's email)

- … directions in the area of discussion of other scientific communities, open source and commercial environments.
- … provide set of principles to guide future work and a summary of ideas of what and how we might invest effort for this in FY14 and FY15
- … discuss scientific software and global collaboration – including frameworks, databases, analytics, collaboration and analysis tools

Our areas

- Frameworks and analysis tools
  - Environment for code development and application deployment
  - Batch-oriented event processing frameworks
    - reconstruction
    - event selection
    - filtering in real time
  - Physics analysis tools
  - Very experienced in software and physics
- Databases and collaborative tools

Issues to contend with

- Managing divergence
  - Early and during data taking
  - Amongst experiments
  - Lack of expertise (cause)
- Redundant facilities
  - Maintenance burden with low manpower
  - Divergence of features
- Collaboration and contributions
  - Agreeing on what needs to be done (labs, universities, programs, foreign institutions)
  - Accepting and making contributions available
- External changes to computing environment
  - Many-core and multi-core
  - Low power, heterogeneous architectures
  - High speed SSD, networks, and file systems

Direction topics

- How do we make our products easier to use and more readily available?
  - Better tools and processes for development, distribution, deployment, and teaching aids
  - Become more embedded in the experiments to ensure good software is produced
- How do we consolidate our effort?
  - share libraries and tools as much as possible
  - greatly improve integration and configuration with other services within SCD
  - Can we make our development process better through working together in teams?
- How do we expand beyond the FNAL Computing Sector and current function?
  - Internationalization?
  - How much further into astrophysics and cosmology?
  - Reconstruction and analysis formats differ – Better fill the gap between these two worlds?
  - Integrating algorithm development into our department to help experiment developers start out in a good direction?
  - Can we adapt multi-language and layered contributions (i.e. standard library) from other communities?
  - Increase our visits to tool providers and users from other R&D communities?
  - Should this be a priority?
- Can we be prepared for changes in computing?
  - Including changes driven by commercial and research sectors
- So how do we do all of the above and keep up with support and necessary features for the tools we have?
  - Do we have the manpower?

Immediate efforts

- Complete an end-to-end development/release/ distribute process
- Common software framework toolkit
  - Move forward on the initial targets within our department
  - ROOT I/O handling in framework modules
  - Multithreading and many-core capabilities
  - C++ course and training material development and review
- Continue the CosmoSIS development and deployment
  - Includes integration and demonstration of use with the PDACS Galaxy environment
  - Includes introduction of higher-level use cases for LSST DESC beyond running parameter estimation problems
  - Challenge is to introduction structure to development and running of code in this setting
  - Expand on the attribution and contribution properties

Backup slides

Future projects

- Toolkit expanded to cover
  - databases interactions and storage,
  - Standard visualization components,
  - better data model definition and navigation
  - persistency data format more useful for non-interactive applications (real-time requiring network streaming)
- Promote big data test bed and demonstration, including evolved architecture for analysis
  - Data models for use with non-ROOT formats or mapping from ROOT format and alternative analysis tools (There are additional candidates now and users of different tools)
  - Support for fast disk speeds (>750MB/s), which demand changes in data organization (streaming issues, compression issues, file use issues)
  - Explore global file systems and data stores permitting different data organization than conventional files.
- External
  - looking at, HDF5 and friends, R community, Galaxy, other frameworks such as Charm++
  - Should we moving further towards working with outside projects like Belle2 with the offline database interfaces Detector R&D software with NEXT

A few things to think about

- Compartmentalization of processing aspects limits what we can do
  - Framework is separated from workflow, but processing modes are constrained by it
  - Framework is separated from Data handling and the database world, locking us into particular abstractions and ways to manage processing of data
  - Movements towards “big data” may require tight integration with algorithm development and use of the event data store
- Should we consider better integration with workflow and data handling systems?

ROOT activities

- Current experiments most affected by overall performance and multi-threading issues.
  - Performance major bottleneck is compression, and next the streaming.
  - There are ways to improve this by using little-endian instead of big-endian.
  - Multithreading in ROOT 6 and beyond
- broad directions -
  - many-core and multi-core capabilities (already in major direction)
  - Bridge tools with other scientific community tools such as R, HDF5, etc. (some in major direction)
- specific efforts wanted or already started
  - HDF5 or similar libraries
    - read HDF5 files (data produced from other environments)
    - write a subset of data into HDF5 for use in other environments
    - explore a collaboration on HDF5 future file formats for better ROOT integration
  - C++ standard including type introspection, standardizing flexible binary storage of C++ objects
  - reengineering of the core of ROOT to be used more broadly (included threading issues)
  - programming models that include parallelism and hardware that is not necessarily best served by more pure OO techniques. Create tools that hide the details of these additional models.
  - CUDA port of parts of the I/O libraries to be able to operate directly on ROOT data for these types of machines
  - can we invest more effort in the math libraries? (future item)
- Effort statement - still need to do the current operational developments and support well. Can try to team up with NSF and DOE grant effort to pay for part of the additional R&D? All of it requires significant effort.

Additional things to consider

- New programming models
  - Algorithm organization that is stateless, more easily permitting map-reduce
  - Multi-language support
  - Distributed processing within one application (primarily for filtering / real-time areas)
- Mixed processor handling
  - ARM, Phi, GPU, x86, Power8
  - Depends on class of problem
  - Provide tools and methods for clean transitions between them
- Expanding the definition of event
  - Better match how non-collider experiment view their data
  - Permit processing of multiple granularities of processing within one application (e.g. interaction, time window, group of interactions, subruns)
  - Unification of product hierarchy and hierarchy of modules needed

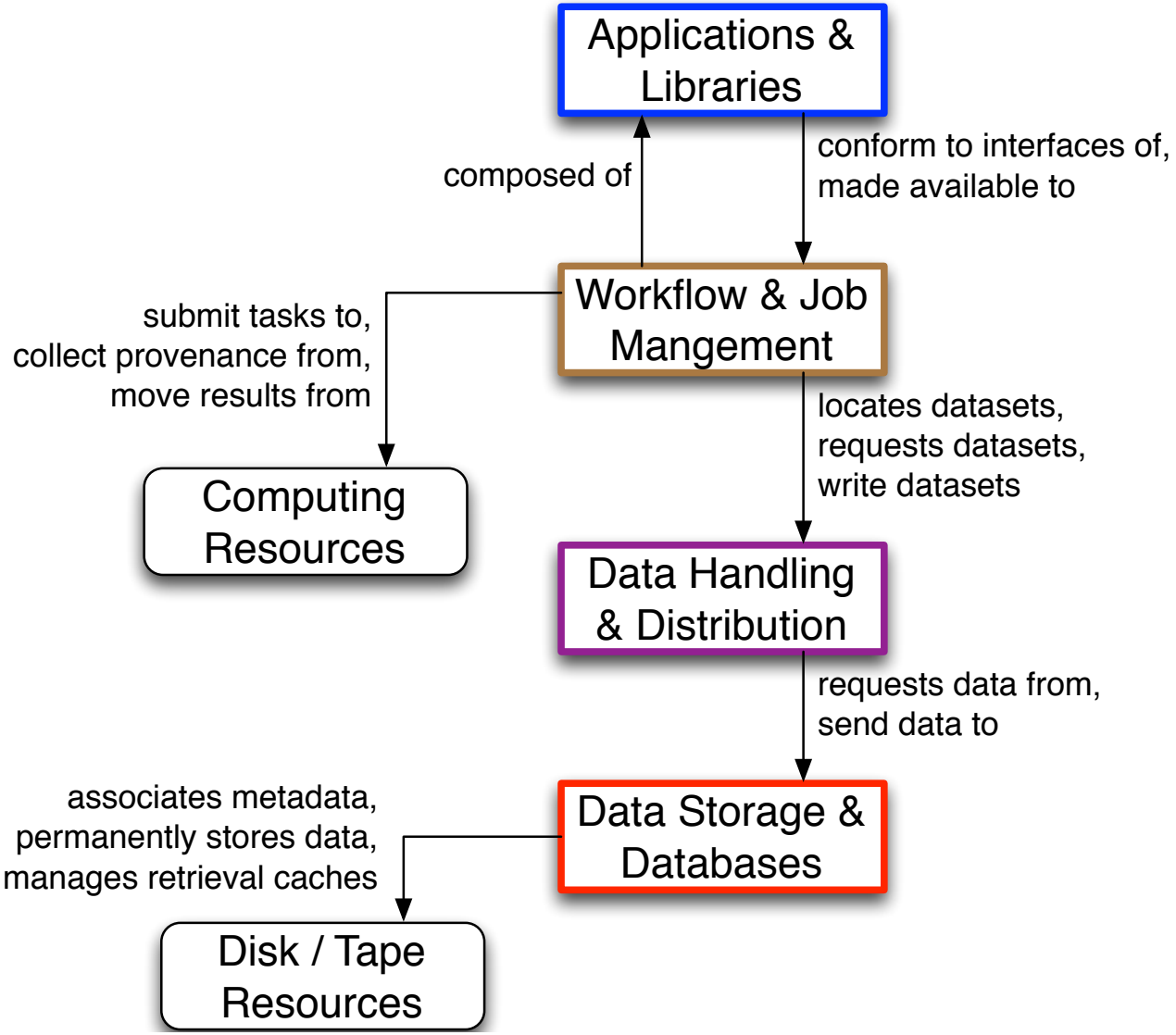
Customers

- HEP detector experiments
  - LHC (CMS)
  - Neutrino program (NovA, uBooNE, mu2e, g-2, LBNE)
- Dark matter
  - First installation within Darkside-50
- Dark energy
  - Excellent forward motion in bringing HEP concepts to the DES analysis world

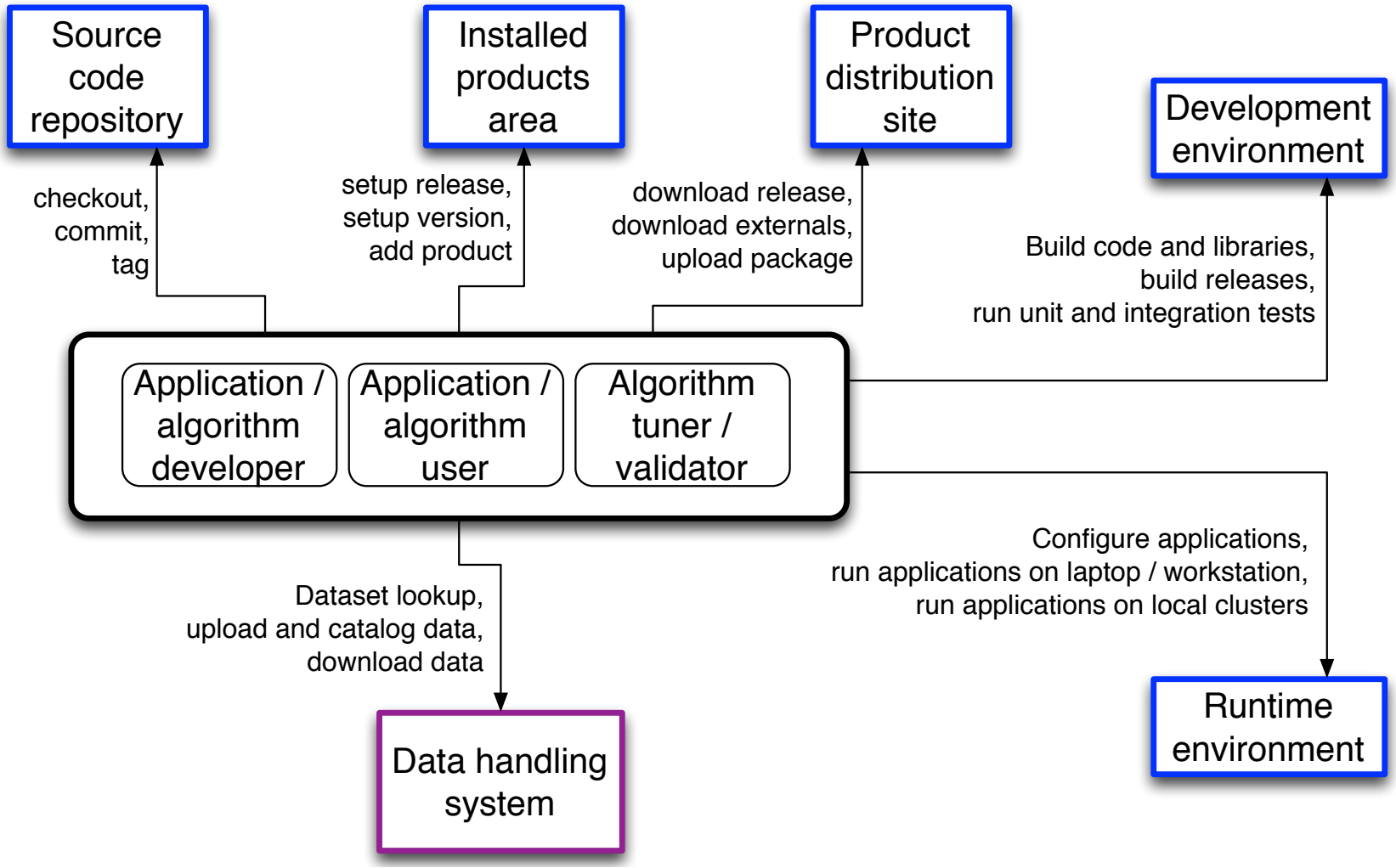
Current framework properties

- Coverage
  - Currently provides facilities for physicists to inspect and process data up to the ntuple or end analysis
  - Collaborative science: Defines a development and runtime model, along with ways to build, test, install, package, and distribute packages
- The framework we are discussing
  - Is scheduled by the workflow management tools
  - Interacts with file and metadata handling systems
  - Has real-time and batch capabilities
  - Has limited interactive abilities
  - Has tools for defining complex data models
  - Permits algorithms to be defined and access additional facilities through agreed upon APIs

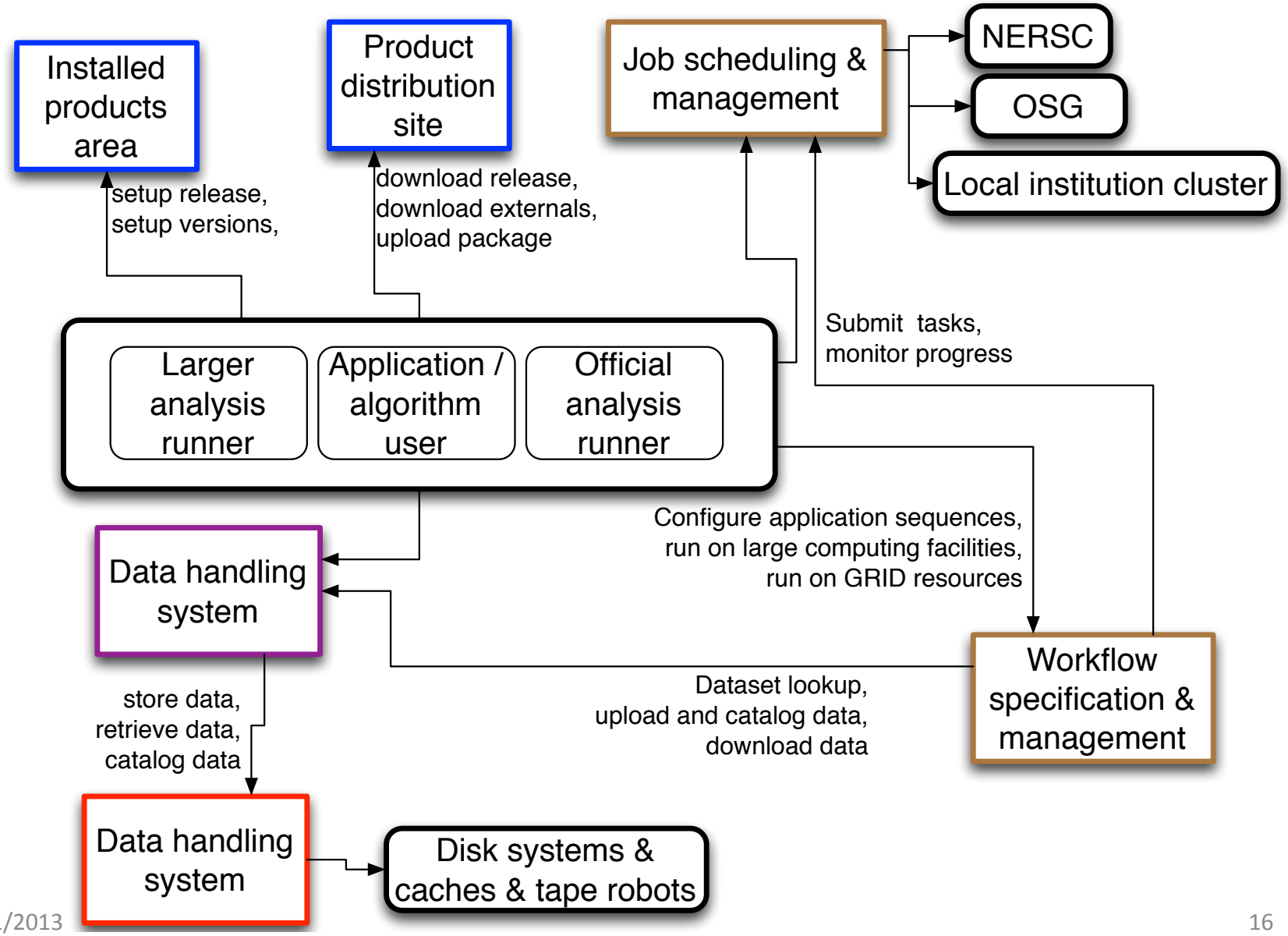
Main relationships amongst pieces



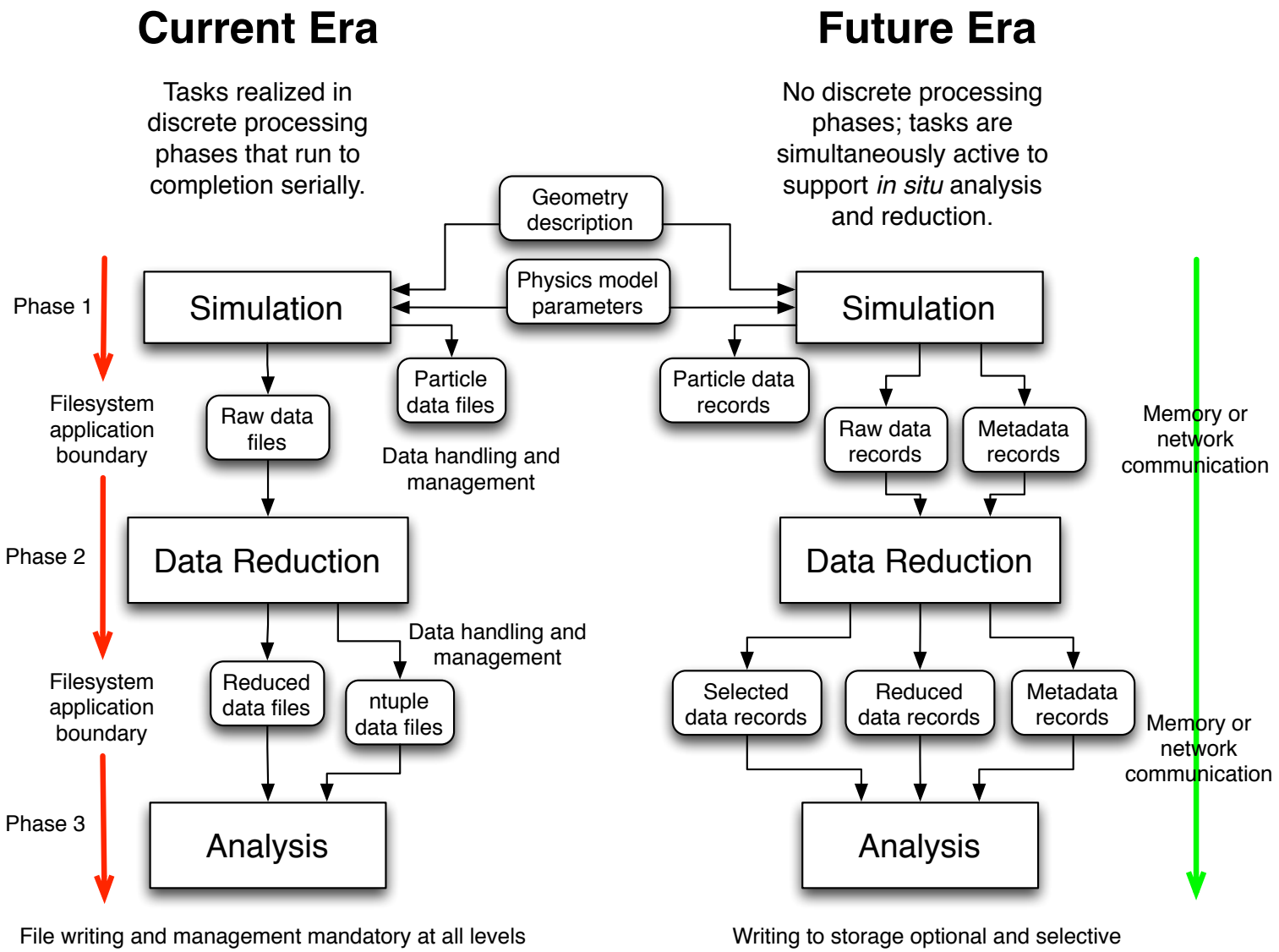
Local resource interactions



Larger analysis role interactions



Possible future processing scenario



Possible future integrated system view