Architecture for a High Performance Computing Center to Support Science and Engineering

Richard B Rood
University of Michigan
with more than a little acknowledgement to
Daniel Q. Duffy, Computer Sciences Corporation
W. Philip Webster, NASA

November 15, 2005

Context and Purpose

- CONTEXT: These concepts and these figures follow from my tenure as the Chief of the Computational and Information Sciences Technology Office at NASA's Goddard Space Flight Center. Many of the specifics of the content were developed by Daniel Q. Duffy of Computer Sciences Corporation in concert with W. Philip Webster (NASA) and myself.
- PURPOSE: This document presents the basic concepts of a "center" architecture to support science and engineering which requires high-performance computing. These concepts are intended to place "data" and "applications" at the forefront of center design. This stands in the contrast to the traditional, and still common, notion which focuses on "the computer."

Applications (software) and Data

- The next 6 slides discuss why the focus is on applications (software) and data, and why the application suite defines the required computational center and services.

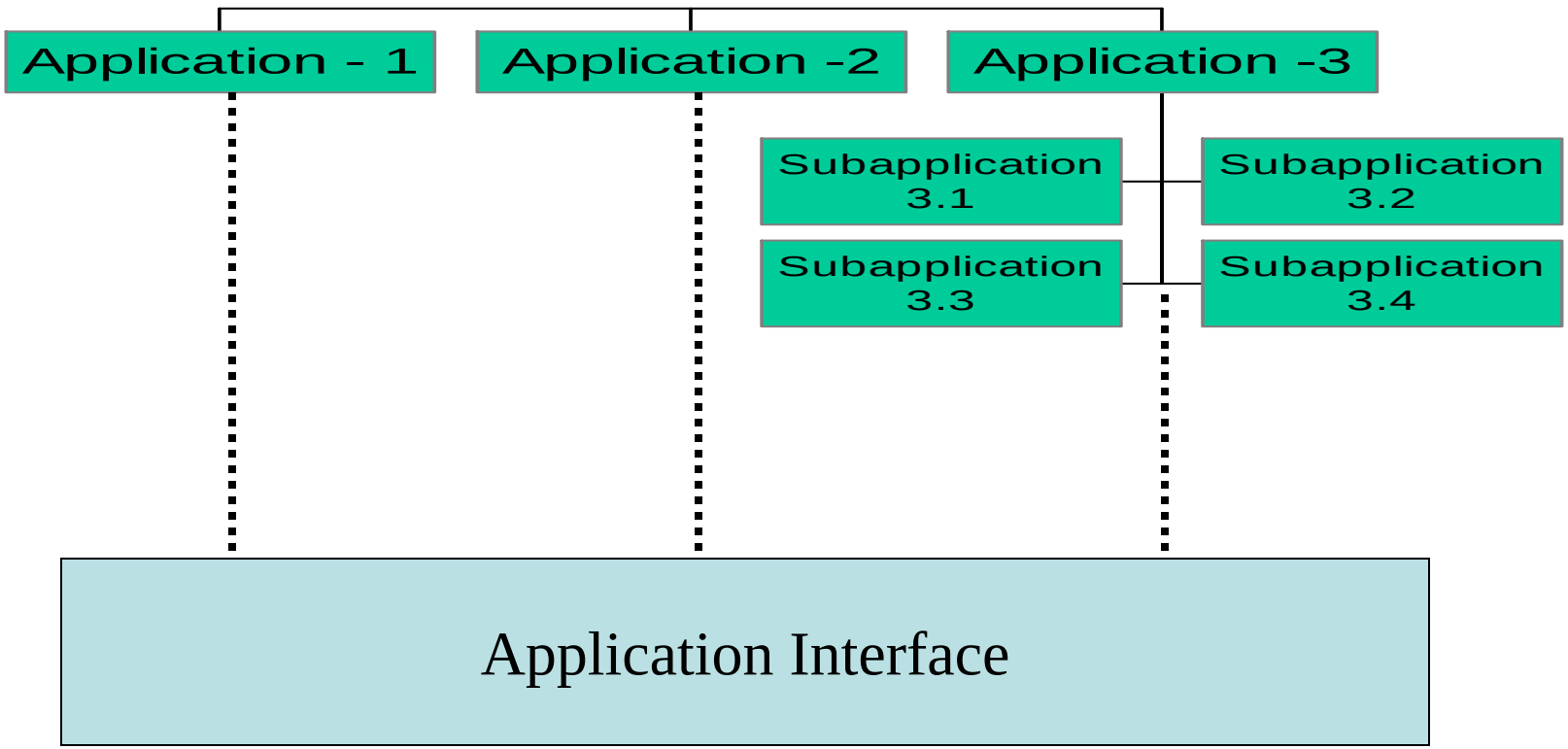
Focus on Applications (Software) and Data

- Applications (Software)
  - Applications Software
  - Analysis Software
- Data
  - Model simulation
  - Observations
  - Assimilation and analysis of Observation and Model Simulation

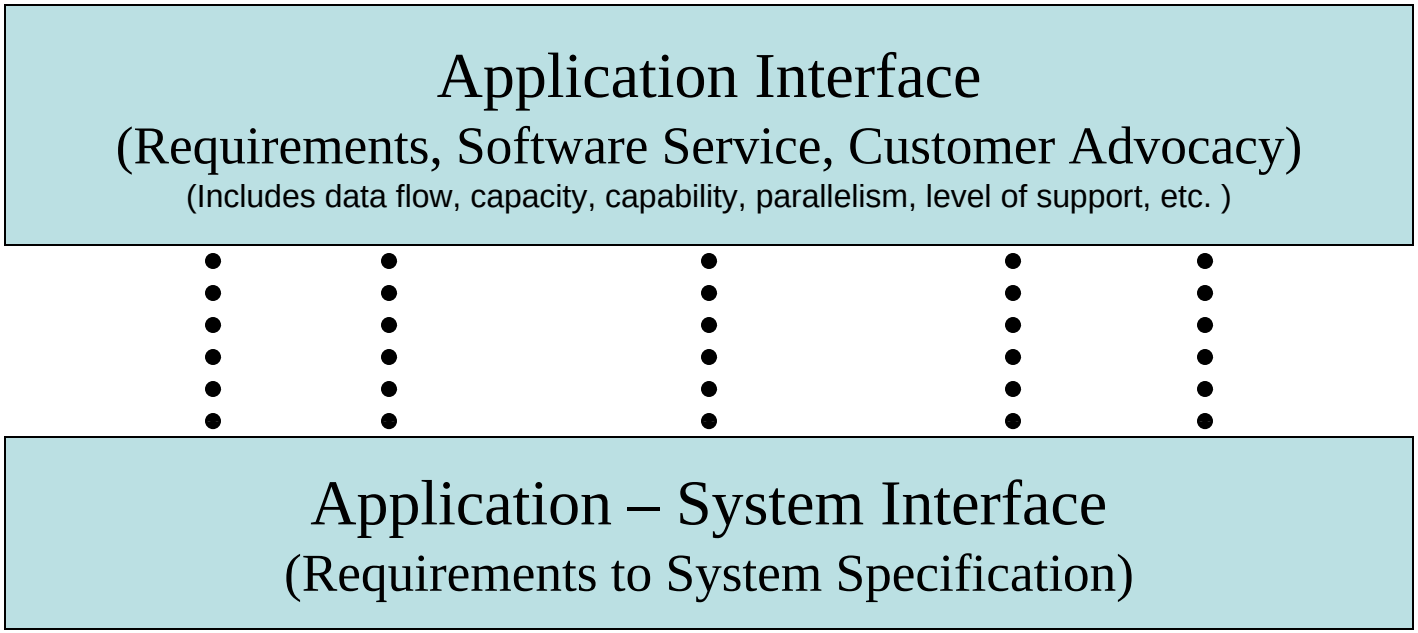
Why Applications Software and Data?

- They are long-lived, actually do our work, provide our information, and are what we control.
- Hardware is a tool to store and execute, it is short-lived, and not under our control.
- We have to assure that hardware is available for us to run our applications and that we can store and retrieve our data.

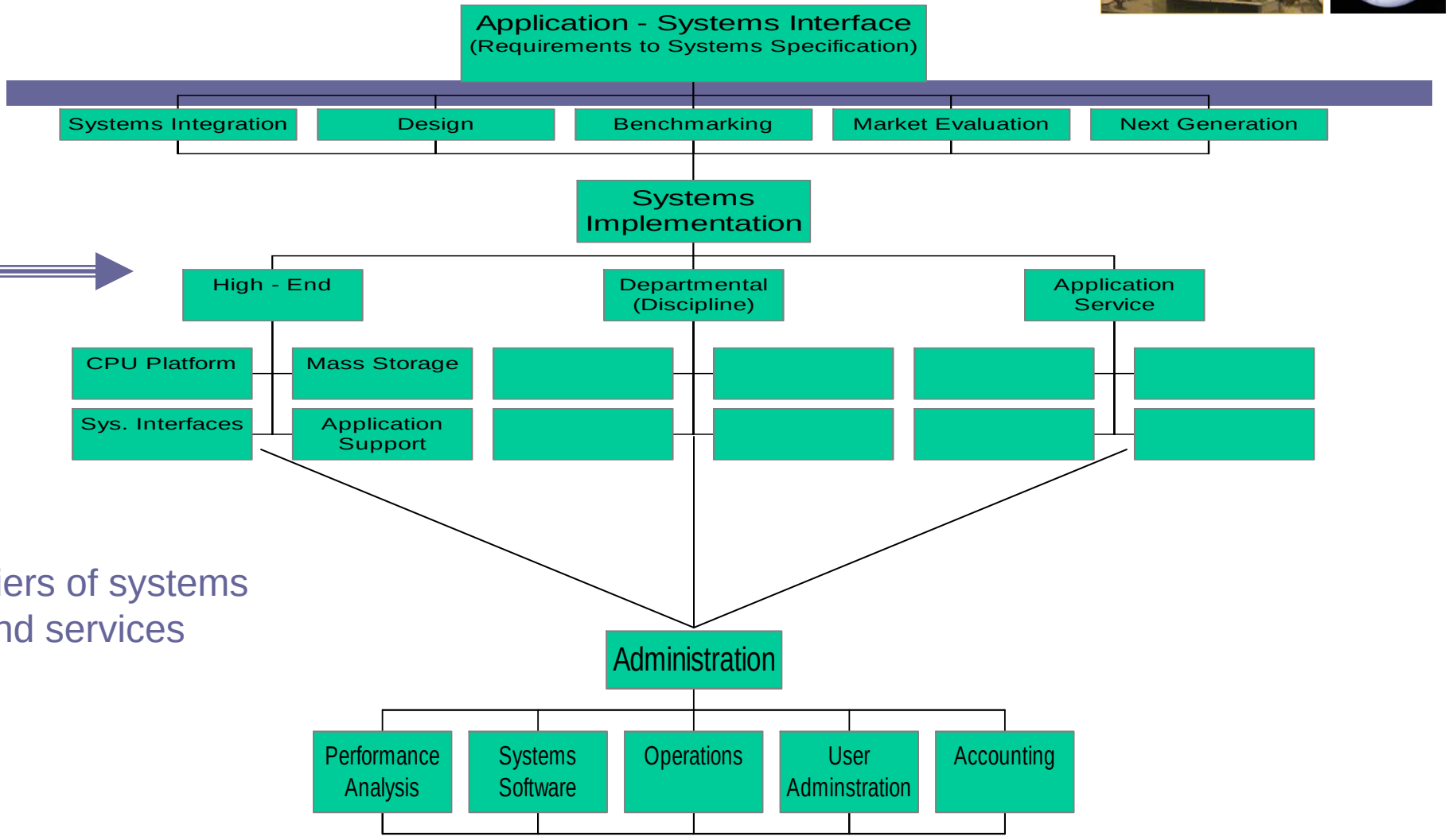
Application suite and user community provide a specific set of characteristics



The computational system defined by the characteristics of the application interface

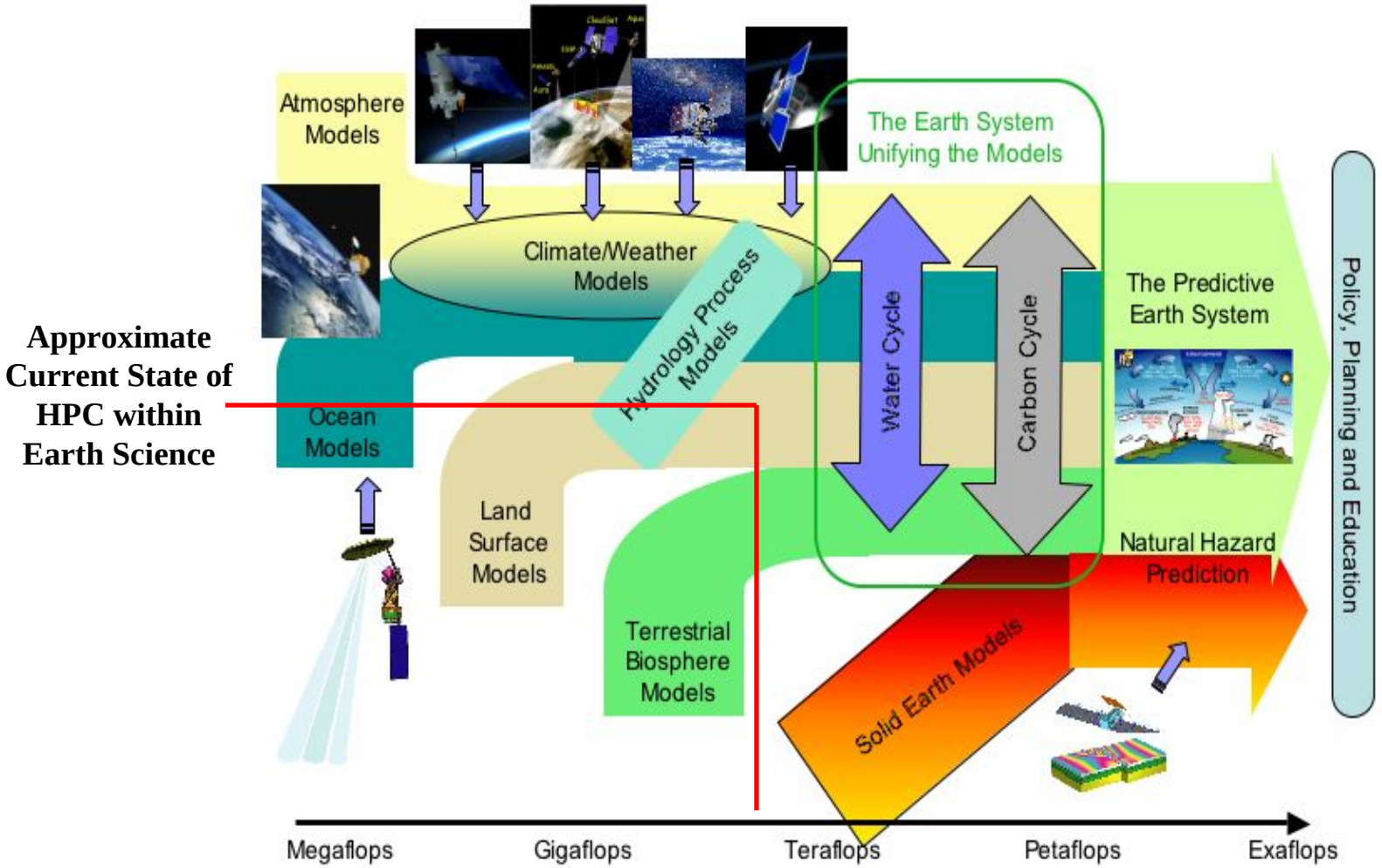


The needed computational service follows from the Applications-Systems Interface



Tiers of systems and services

Earth Science Modeling as an example of a high performance computing problem.



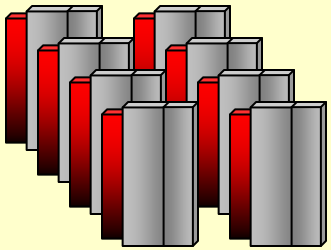
Center Architecture

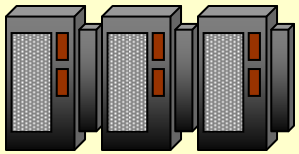
- The following viewgraphs introduce the notion of a center architecture to support the applications and data needs of the user community – defined by the suite of applications.

Typical Computer Centers Architecture

Loosely Coupled over 1 Gbit Network.

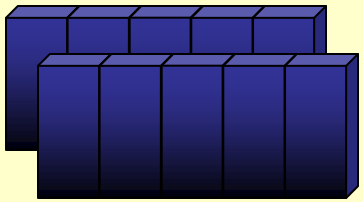
Compute Engine

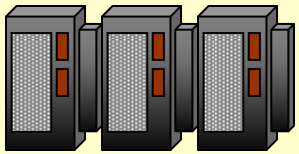




Data

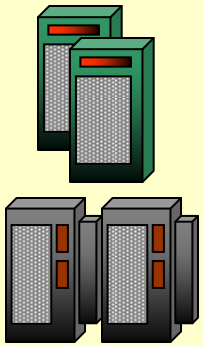
Compute Engine



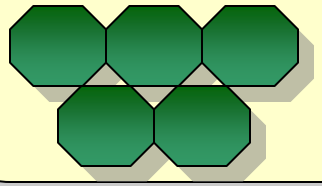


Data

Hierarchical Storage Management (HSM)



Data



Multiple interfaces and copies of data.

Pros and Cons

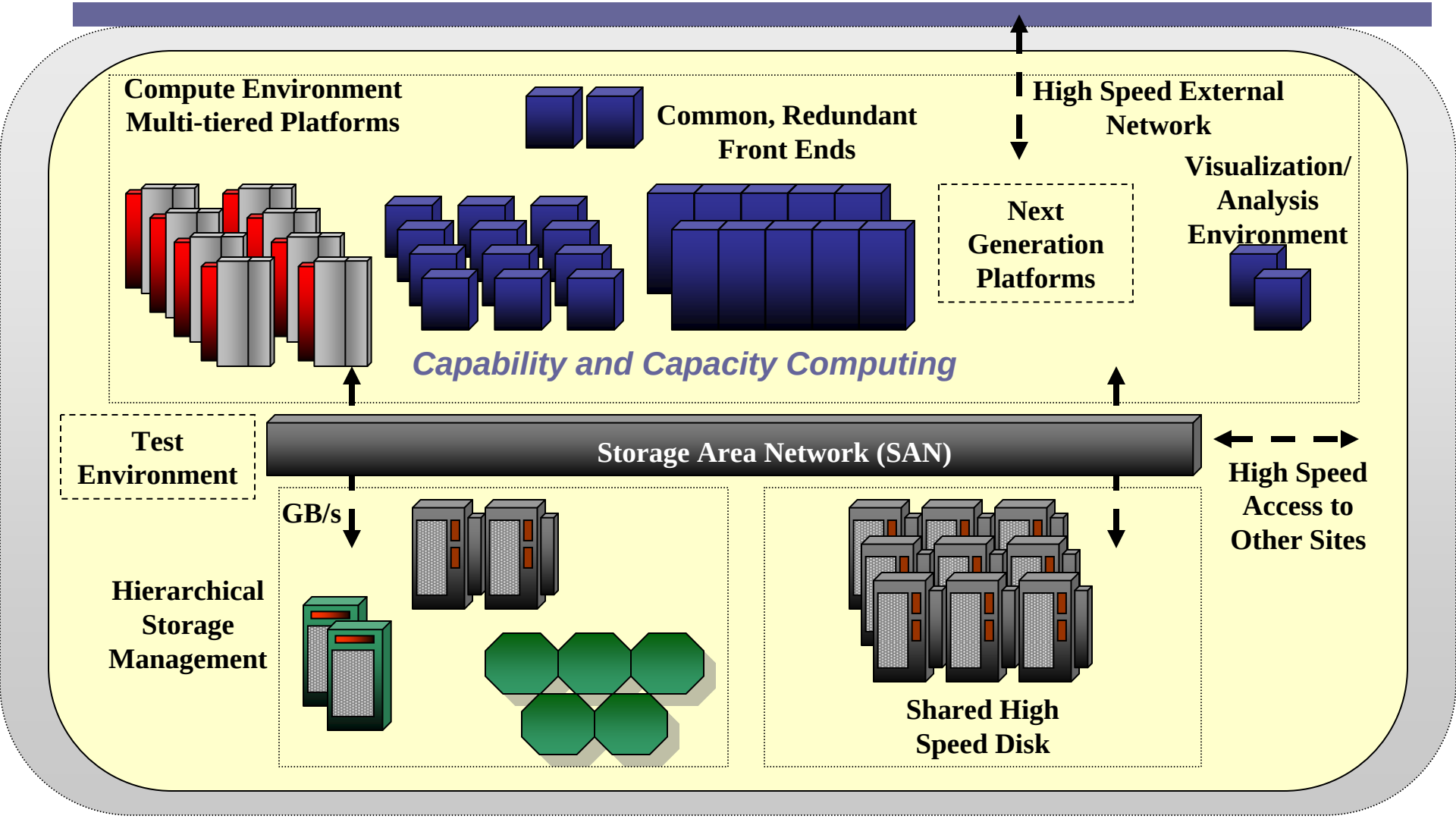
- Pros
  - Highly optimized storage resources
  - Applications take advantage of platform specific optimizations to increase performance
  - HSM provides consolidated long term storage management
- Cons
  - No common architecture between machines or centers
  - Local attached storage leads to multiple interfaces and copies of data
  - Platform specific optimizations minimize portability making it difficult for users and applications to move between machines
  - Little flexibility for the center to schedule and match resources to applications
  - Lost productivity due to the complexity of the resources
  - Visualization and analysis resources are remote and difficult to access, and data movement to these resources is difficult

Recent Changes in the HPC Landscape

- Clusters
  - The price performance of cluster computing continues to improve.
  - Maturation of Linux and the development of tools to better manage and use clusters
- Data
  - The amount of input and output data needed for modeling and simulation continues to increase seemingly without bounds.
  - The sources and format of data are increasing, and research instruments do not adhere to the same standards as the operational community.
  - Users and centers need better ways to manage, transport, and effectively use their data.
- Storage Area Networks (SAN)
  - Multiple vendors are supporting software to manage a SAN in a heterogeneous environment.
  - These products have also matured enough to begin using them in a production computing environment.
- Grid Computing
  - The software to facilitate resource sharing has been developed and implemented at various centers.

Center architecture needs to take into account these HPC trends and improvements while maintaining flexibility with respect to applications and future technological changes.

Proposed Architecture Vision
Data Centric, Multi-Tiered



Pros and Cons

- Pros
  - Ease of use leads to higher user productivity and better support
  - Common front ends allow for greater utilization of resources
  - Initial steps toward a common center architecture, which leads to
    - Higher user productivity
    - Lower support costs
  - Storage area network provides large, fast storage from all compute platforms
  - Multi-tiered HSM environment integrated into the SAN
  - Multi-tiered computational engines provides the appropriate platform for the application, rather than the other way around
  - Extensible architecture makes it more adaptable to new architectures and changing requirements
  - Visualization Environment can be tightly coupled to the data using the SAN
  - Interface with other centers can be simplified.
- Cons
  - ***Vendor specific SAN software could limit future integration efforts***
  - ***HSM is typically tightly coupled to the SAN software***
  - Local attached storage may still be needed for certain applications
  - Multiple operating systems increases complexity

High Performance Computing across multiple centers

Scientist Workstation

Resource Sharing

Data Management

Collaboration

Common Policies

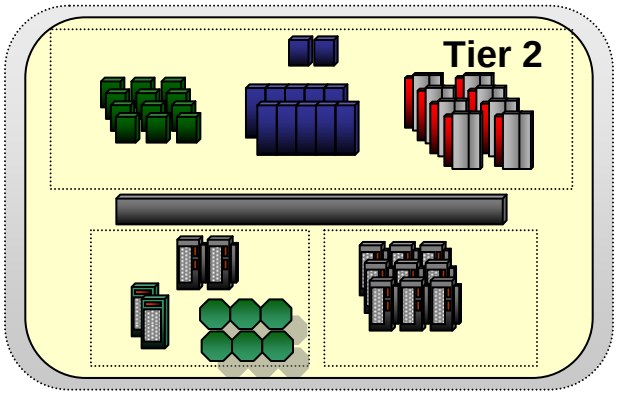
Information Power Grid

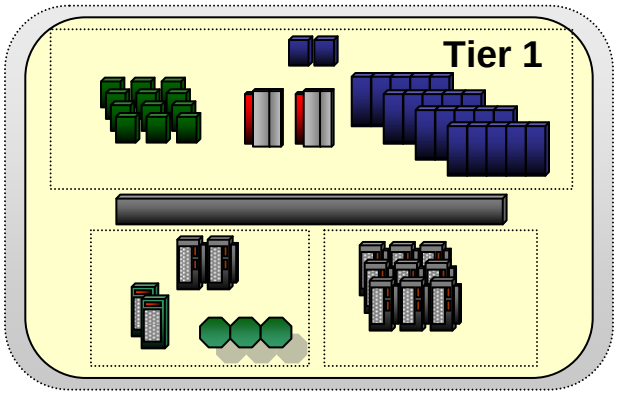
Data Management System (DMS)

High Speed Network Between Centers

Common Center Architecture

Common Architectural Elements





Adaptable, easy to learn and use environments facilitate the end user to compute from anywhere to the right resources.

Example of Current Job Flows

Application Development

Data In

Code

Data Out

Editors, Compilers, Libraries
Local and Mass Storage Access

Application Runs

Data In

Run Jobs

Data Out

Batch Queues
Local and Mass Storage Access

Data Analysis

Data In

Analysis

Data Out

Local and Mass Storage Access
Analysis and/or Visualization Packages
Typically at the desktop level

Constant feedback drives more job flow processes

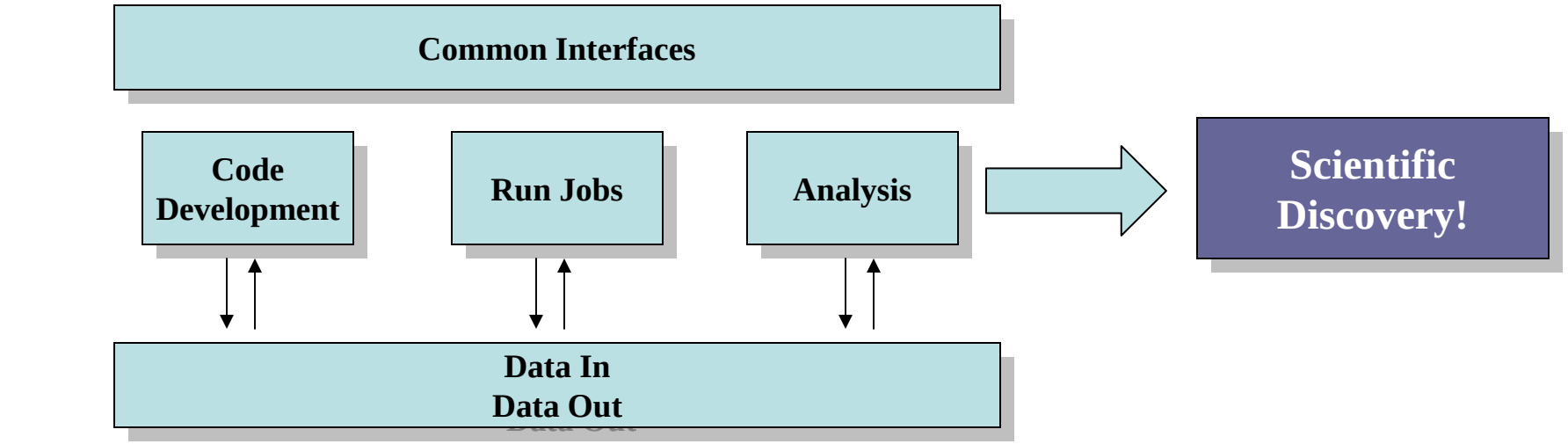
Current computing centers are usually designed with focus on one part of the job flow process: running applications.

However, Application Development and Data Analysis are generally more computationally demanding --- but messy and hard to define.

An integrated architecture must take all of the job flow process into account!

Scientific Discovery!

Resulting Job Flow



- **Transitions between the different steps within the job flow is intuitive, quick, and simple.**
- **Common interfaces to resources provide a more usable environment.**
- **Jobs are run on appropriate and available resources.**
- **Data exists in a single storage area network at all times.**
- **Visualization Environments are readily available and easily assessable.**

Reduced Time to Discovery!

Business Model

- More return on investment
  - Consolidating storage into a storage area network
  - Reduce the overall requirement for disk capacity
  - The money saved could go directly into more computational capacity
- Lower support costs
  - Duplication of services and operating systems lowers support costs
  - Common center practices assist in trouble shooting and decrease help desk support while increase the level of service to the users
- Increase in user productivity
  - Common center practices simplifies the user interface to the HPC resources
  - Jobs run on appropriate resources and maximizes utilization
  - Resource Sharing: jobs run on available resources to minimize wait time

The overall goal of adopting this architectural approach is to provide a framework in which to make computational investments that most effectively enable science and reduce the time to solution in a cost effective manner.

Combating Itinerate Requirements

Computational Requirements

Many requirements that need to be captured, documented, and measured throughout the compute cycle.

Computing Facilities: Research and Production

Solutions are implemented to satisfy requirements:

- Compute
- Storage
- Networking
- Applications
- Etc.

Continuous Feedback over time allows for a sustained cycle of measuring how well the production facilities satisfy the requirements.

Future Requirements

Science and engineering productivity is the most important metric, and the computing resources are necessary to enable discovery.

Common metrics need to be constantly analyzed to ensure the computational environments are adequately supporting the applications.

Scientific Discovery

Metrics

Conclusions

- The resulting architecture and business model creates the necessary framework in which to:
  - Provide adequate computational capacity and adapt to changing application space.
  - Enable IT vendors to focus on relevant application space.
  - Support collaborative efforts within College and University improve the ability to interact outside of the University and with Federal Centers.