NSF EarthCube Conceptual Design Project
Scalable Community-Driven Architecture
Award Period (July 2014 - June 2016)

S.Caltagirone, D.Crichton, S.G.Djorgovski, T.Huang, S.Hughes, E.Law, A.Mahabal, D.Pilone, T.Pilone
September 16, 2015

Outline
I. Introduction
II. Drivers / Requirements
III. Use cases
IV. Conceptual Architecture
V. Benchmark Systems
VI. Lessons Learned
VII. Recommendations

I. Introduction

A complex cyberinfrastructure such as EarthCube requires a well thought out software and system architecture in order to identify the principles, patterns, and relationships between the software, services, and data and their deployment and integration. As a distributed, national infrastructure, such an architecture will help to guide the implementation, integration, and governance for EarthCube at a broad level. This paper describes our approach, methodology, key drivers, use cases, conceptual design context, alternatives considered, lessons learned, as well as a set of recommendations based on our work on this project so far and related experience.

Our development of a conceptual architecture follows a lifecycle that identifies a set of iterative steps for generating the conceptual architecture, report and recommendations. The elements of the lifecycle include identification of a methodology for capturing the architecture, identification of stakeholders, key use cases, and architectural drivers. From these we can identify the architectural principles, develop architectural models and engage the community to collect feedback. This feedback is then used to drive benchmarking activities and generate recommendations for adopting the architecture into the EarthCube program.
This methodology has been used several times, in our experience, to build distributed cyberinfrastructures in space science, earth science, and biomedicine. The crux of our approach is recognizing that an architecture must be able to provide strategic guidance to support the evolution of such a cyberinfrastructure. This requires understanding how different aspects of such an architecture, the processes, data, and software system itself can both evolve independently and come together to support the scientific community. As a result, our design focuses on mapping the drivers, use cases, and principles to three key views of the architecture:

1. Process: The process describes the functions that are supported by EarthCube to enable a system to operate. Understanding the data lifecycle is critical for development of a comprehensive set of functions needed by EarthCube.
2. Data: The information architecture describes the data, its structures, and organization. It defines the data, their taxonomy, ontology and vocabulary for EarthCube.
3. Technology: Describes the technology components necessary to construct a system facilitating the flow of data lifecycle from bits to information to knowledge.

II. Drivers and Goals

For the last two years, geoscience communities have been engaged in a series of workshops to articulate and document their cyberinfrastructure drivers, challenges and needs. These community inputs are the keys to shape and elucidate our scalable architecture. The drivers described below identify those challenges that the EarthCube architecture should address. Relative to each driver, multiple issues identified by the community will be used to define architectural principles that will form the basis of our architectural approach. Three overarching challenges have been identified as core drivers that will steer future geoscience research:

1. Transform and accelerate research and discovery by turning data into knowledge and enabling interdisciplinary data integration. Data governance is a critical hurdle to enabling increased accessibility, interoperability, and data integration. Much of the current data is difficult to use due to variable data quality and uncertainty errors; lack of common vocabulary/ontology with various formats/structures; lack of rich and standardized metadata, documentation; highly distributed data; and provenance of the data itself. Governance needs to consider the importance of common standards, sustainability, and operations for future systems to ensure data is usable, data and systems can integrate, and analysis needs can be met.
2. Provide critically needed data, tools, and computational resources and frameworks for cross-domain scientific collaboration, analysis and with long-term geoscience software and data preservation, discovery and use. Lack of real time collaboration and sharing capabilities inhibits scientists from having data and computational
services that are needed. Limited use of standards and accessible data repositories make use of the data difficult. Limited discoverability of the data limits the use. Long-term challenges with sustainability and maintenance threaten the preservation and long-term use of the data.

3. Provide a geosciences cyberinfrastructure and architecture that is scalable, extensible and sustainable. The broad scope of EarthCube and geoscience data means significant heterogeneity of data formats, data size, computational and processing demands, and users. Given that along with the long timespan envisioned, EarthCube requires an architecture that can evolve with technology and funding while meeting geoscience and sustainability needs.

III. Key Use Cases

This preliminary set of use cases was derived from End User Workshops, feedback from the community, and our experience in building similar systems for multiple science disciplines. These use cases provide an overview of anticipated needs EarthCube should meet. Use cases, drivers, and principles will be used to derive EarthCube’s architectural functional requirements. The conceptual architecture itself does not have a prescribed set of requirements, but rather is derived based on these drivers and key principles from our experience.

1. **Big Science**: Scientists use EarthCube to discover, access, and synthesize data from multiple disciplines to support research efforts.
2. **Data / Information Discovery**: Earth Science instructors use EarthCube to identify data sets relevant to a course curriculum.
3. **Comparison**: Earth Science professionals use EarthCube to compare a data set to a simulated model.
4. **Provenance**: An EarthCube Research Curator wants to ensure a dataset is available and valid.
5. **Model & Visualization**: Scientists use EarthCube to access and use a model, then communicate model parameters associated with their findings.
6. **Collaborative Science**: Two research groups use EarthCube to collaborate on a project.
7. **Dark Data Contribution**: A researcher is encouraged to make their data available through EarthCube.
8. **Tools Contribution and Ongoing Software Maintenance, Support, and Hosting**: A technologist uses EarthCube to distribute a new tool they developed.
9. **Data Documentation:** A data owner uses EarthCube to prepare data into an accessible format.

10. **Models Sharing:** A researcher uses EarthCube to host a model and get community feedback.

11. **High Performance Computing Resources & Additional Storage Ability for Data:** A scientist uses EarthCube to process and host field data.

12. **Data Center:** A researcher uses EarthCube to connect to other repositories to discover relevant data sets.

13. **Using Real Time Data:** A scientist uses EarthCube to discover and interact with real time data.

14. **Physical Sample Curation and Discoverability:** A researcher uses EarthCube to search for information related to a physical sample to support their work.

15. **Sustainability:** A research team migrates their project to EarthCube to leverage its infrastructure.

### IV. Conceptual Architecture

The drivers and use cases provide a context for developing the EarthCube architecture. Figure 1 below illustrates the context for our conceptual architecture. It shows an integrated cyber-infrastructure where data, algorithms, computation, and visualization are brought together across a highly distributed data environment to quickly construct new analytic capabilities for different stakeholders, measurements, science questions, and applications. It is critical to point out that this is considered a high level diagram for the purpose of setting the context of EarthCube. The assumption that there are collection systems on the left of the diagram, that focus on data acquisition, processing and preparation. The right side of the diagram focuses on data use. We define EarthCube to provide the unifying cyberinfrastructure that brings together the left and right side communities in order to support data access, integration, and analytics of highly distributed geosciences observational and model data, captured and managed within in a federated and distributed community.
In order to translate this high level concept, we have found it useful in our experience to define the architecture as a set of views. Those views are most generically data/process lifecycle, data architecture, and technology architecture which is detailed below.

1. **Data and Process Lifecycle** - describes how data moves across the EarthCube cyberinfrastructure. As a distributed geosciences cyberinfrastructure, a major challenge is defining the data lifecycle that addresses the data generation, curation, transport, ingest, management, search, distribution, analysis, and visualization. Given that EarthCube doesn’t govern the entire data lifecycle with data generated in various forms, there is a need to define how the data itself comes together. In particular, understanding the issues around curation of data, distribution and access of data, computation on data, etc, will be critical to laying out the architectural topology and making decisions that affect the governance, management, operations, performance and usability of the system.

2. **Information Architecture** - describes data within EarthCube. The Information Architecture represents the information elements that are required by the process and technology aspects of the system architecture to define the data within the EarthCube geosciences cyberinfrastructure. These elements encompass the architectural elements that can be used to describe the data (e.g., measurements and model output) in EarthCube including the information models, both for bringing data together across EarthCube as well as discipline specific models to describe specific types of data, core dictionaries for constructing and annotating data, and
standards around the formatting of data itself. In our experience, decoupling the representation of the data from the software systems and services is key to allow the data to be used by different software capabilities overtime.

3. **Technical Architecture- technical components and services within EarthCube.**

The Technology Architecture represents the logical software and hardware capabilities that make up the EarthCube Cyberinfrastructure (CI). The cyberinfrastructure describes the research environments in which capabilities of the highest level of computing tools enable both data providers and researchers to interoperate with data and services over the network. It is more than just hardware and connected systems. The software services implement the functions of the data lifecycle and support the information architecture to bring the CI together. It involves process policies, interfaces, and standards clearly documented, followed, and evolved. As a virtual community of multidisciplinary contributors, the community-driven cyberinfrastructure of EarthCube has two fundamental foci: efficient access to data and to enable science research through data analysis. A distributed architecture describes the common infusion pattern used by enterprises to create an interconnected network of data and services to address various domain-specific needs. The key aspects of distributed architecture include applications and services interfaces, service and data accessibility, availability, security, scalability, reliability, and dependencies.

V. **Benchmark Systems**

As part of our conceptual architecture document several existing systems were evaluated and reviewed for best practices. These systems were all evaluated based on key aspects of distributed architectures, including interfaces, accessibility, availability, security, scalability, reliability and dependencies.

- **DOE Earth System Grid Federation (ESGF)** - Allows users access to Earth Science research data stores and is an international federated system using peer-to-peer protocols with standardized metadata specifications, no single points of failure, information is shared across nodes and architecture is fault tolerant.
- **NIH/NCI Early Detection Research Network (EDRN)** - Is a program established by the National Cancer Institute, a research network for scientists from 40 institutions focused on identifying and validating cancer biomarkers. Common information models are used for sharing data, public repository with automatic failover, LDAP security, and leveraging of Apache OODT.
● **NASA’s Earth Observing System Data and Information System** (EOSDIS) - Is made up of 12 Discipline-oriented Distributed Active Archive Centers (DAACs) and 7 Science Investigator-led Processing Systems (SIPS) that are geographically distributed throughout the US. Interoperability is achieved through the use of various centralized services that encompass both user facing and system facing services. Enterprise-scale infrastructure and centralized authentication are key components of the system.

● **NASA’s Planetary Data System** (PDS) - The purpose of PDS is to collect, archive and make accessible digital data and documentation produced from NASA’s exploration of the solar system. It is a highly distributed infrastructure with planetary science data repositories implemented at major government labs and academic institutions.

● **NSF Virtual Astronomical Observatory** (VAO) - The purpose of the VAO was to provide astronomers a means to discover, access, process astronomy data seamlessly using toolkits as well as distributed analysis environments for cross-matching catalogs, and incorporating models from theory and simulations.

Based on insights gleaned from these systems, we have developed a list of eight major tenets that we recommend are part of the EarthCube system: common software stack, common model, standard interfaces, Service Oriented Architecture (SOA), decoupled storage, compute and data management, federated search, analytic services, and visualization.

**VI. Lessons learned**

Below is an example list of best practices that the team has learned through our experience with this and other directly applicable projects. They have been taken into consideration and incorporated in our conceptual design.

● It is important to gather drivers and use cases, by working with external existing projects of different sizes.

● One should begin by identifying existing standards in conjunction with the community, providing for interconversion of data-formats and smaller flexible toolkits, and metadata incorporation into existing datasets.

● Working groups need to be formed with the community for constructing domain specific information architectures. Having an agreed upon common information architecture allows separate groups to work on different products independently.

● Building the common information model takes longer than expected and involves training domain scientists in data modeling.
• Close connection between scientists (with user stories) and architecture designers (and implementers/developers) is crucial. This also helps with matching vocabularies and semantics.
• Planning early uptake by users through appropriate tools, and proper advertising is important.

VII. Recommendations

Based on our experience including lessons learned from other projects and development to date for the EarthCube conceptual architecture, we recommend the following:

• In addition to the conceptual architecture, a reference architecture including specifications of core software services, tools (including infrastructure monitoring and administration), data models, and interoperability standards should be developed.
• The conceptual architecture should be used to establish a strategy for implementation and governance of EarthCube leading to a set of pilot projects to validate the conceptual design.
• The detailed reference architecture should be developed in parallel with the pilot projects that develop specifications for EarthCube as part of an agile software development process.
• Testbed should be launched that help to integrate existing EarthCube tools, data centers and cyberinfrastructure capabilities based on the conceptual architecture.
• Pilot projects including development of information models and relevant key software services and tools should help to further define the reference architecture throughout the EarthCube project recognizing that an architecture is a living artifact.