# Recommended Standards and Specifications for EarthCube Projects

Authors: Ken Rubin, Mike Daniels, Dave Fulker, Jed Brown, Stephen Richard, Ouida Meier, Ilya Zaslavsky, Craig Willis, Kenton McHenry, Christine Kirkpatrick.

Approved by EarthCube Leadership Council: 7 May 2020, DOI: https://doi.org/10.6075/J0QR4VMG

This living document describes minimal and optimal recommended practices for NSF EarthCube projects, proposed in 2020 and beyond, with the hope that prior funded EarthCube projects voluntarily conform too. The contents of this document are informed by several prior efforts within EarthCube Community Governance to describe programmatic goals (including component architecture and interoperability), open community conditions for standards development, and the needs of EarthCube affiliated data repositories and existing EarthCube projects. Three such efforts are noted and linked in the acknowledgments section.

Standards and Specifications for cyberinfrastructure to support cross-disciplinary science will continue to change as technologies evolve and new science domains join the effort. The ideas expressed here focus on current and future NSF-funded efforts to create new EarthCube and related cyberinfrastructure that promotes interoperability, sustainability, usability and repurposing (often via open-source software).

The Standards and Specifications listed here focus on basic principles and are not comprehensive (i.e., they do not at this time address coding practice, testing, human interfaces, product stability, or other attributes that are currently up to developers to define). Project developers building software are encouraged to engage with the EarthCube Community Office (ECO) Technology Team to take advantage of and incorporate the latest features and recommendations.

This document draws from a recent white paper [LINK] describing some of the rationale for, and nuance of, many of the specifications listed here, as well as additional, options and recommendations regarding  licensing, reuse of existing components, use of notebooks, version control, information exchange procedures (e.g., APIs), and vocabularies/ontologies.

## Primary Guidelines:

1. EarthCube seeks to have all data and data resources produced by and for EarthCube to be FAIR Complaint, in keeping with its status as a co-signatory of the US FAIR data effort.  See http://earthcube.org/fair for additional details.
2. Products require producer attribution and contact, version control, hardware and operating system requirements, and notation of any data format or data resource dependencies.
3. Products shall be properly **registered** in the EarthCube registries documented at: https://www.earthcube.org/geocodes. Currently there are two paths to registration.

    1) For datasets employed in EarthCube: should be made accessible from a data repository whose metadata is exposed and formatted for harvesting by EarthCube's GeoCODES [LINK],  via a repository that is (or easily can be) recorded in the EarthCube Resource Registry [LINK],  -or-  via an existing alternate and interoperable resource

2) For other technical products and resources (e.g., software, models, codes, etc.): the resources should be registered in the EarthCube Resource Registry [LINK].

4. Products need to be **accessible online**. Producers must supply the documentation to the registry indicating how to access the product (e.g., GitHub). Make sure that the licensing terms are clear. Various approaches and recommendations are warranted for different resource types; see next sections.

## Additional Recommended Guidelines

| **Category** | **Minimum Product Specifications** | **Ideal Product Specifications** | **Optional Next-Level Community Specifications** |
|---|---|---|---|
| 1. Developer contact information. | Provide info about EarthCube Resource Registry (ECRR) Responsible Parties. | Keep content current | ORCiD ID for each person identified (author, contributor, responsible party) and keep ORCID info updated. |
| 2. Discoverability of project deliverables, data and digital resources. | Required metadata for registration in the GeoCODES Data Registry and the ECRR for currently supported object types (even if these items sit in a different repository). | Robust metadata including persistent identifiers, URLs, related resources, and other information into ECRR (or Data Registry). | Metadata exported in schema.org structure and JSON-LD format (as implemented in Data Registry and ECRR). |
| 3. Hardware, OS, and other computational requirements. | Provide info as ECRR Dependencies | Point to reference deployment, and a document describing the setup process and the associated setup and maintenance costs | Provide a DockerHub link to the resource (see Item 6. below). |
| 4. Access to code, metadata and/or documentation. | Provide information on the location of code and related documentation in ECRR (to include software licensing) | Open-source software (under an OSI-recommended license), a completely filled out ECRR record (beyond mandatory fields) *and* access to code via version-controlled repository | Software publication: GitHub repos have an end-of-award commit and release with documentation, test input, push to Zenodo including DOI assignment, and availability for functional testing by ECO and others. Where possible, include well-annotated Jupyter notebooks demonstrating access to and re-use of the resources and consider implementing live functionality of notebooks through a Binder [LINK] |

| | | | |
|---|---|---|---|
| 5. Data and resource usage, longevity, and sustainability. | Provide info as ECRR Maturity/Status and Expected Lifetime | Especially for large datasets, access via (open-source) services that offer subsetting. | Provide usage statistics and a narrative describing status and stewardship/ maintenance arrangements and costs. |
| 6. Packaging and distribution. | Resource interoperates in a packaged or containerized way for portability, scalability, ease of installation.<br>If a resource is suitable for containerized or cloud deployment, include a statement, URL and parameters in ECRR entry. | Software development should be complete and packaged, with well described and versioned web APIs. | Provide use case demonstration for a geoscience instance; example: Docker containers with JupyterHub Notebooks, documented workflows, testable input, and robust description and documentation. |
| 7. Interoperability: tools, platforms, or capabilities that the resource interoperates with. | Provide info in the ECRR about Related Resources (e.g., Python, Matlab, data center holdings). | Update as needed | Include references to projects that re-use the resource. |
| 8. Flexibility with domain or community-specific practices. | Use existing or add vocabularies and other specifications (e.g., spatial and temporal frameworks, etc.) used in the ECRR | Deposit new vocabularies in EC-affiliated ontology repositories (e.g., at the ESIP Community Ontology Repository [LINK]). | Use and identify community-accepted semantic, spatial, temporal and other reference frameworks that the deliverables rely on. Use modular specifications that have well defined extension points and processes. |
| 9. Sustainability | Store code in a trusted repository and base developments on widely adopted standards. Allow for community contributions where appropriate and completely document all code. | Store, distribute and maintain developments in organizational infrastructure using in-house staff support. Provide regular updates to code for security patches and evolving technology. | Provide active links to all resources identified and perform regular continuous integration. Continue to develop features as needed and update ECRR information over time. |

**Notes: “***Minimum” refers to declared specifications that projects use and register in the appropriate registry. There is no requirement that these specifications are endorsed by EC or follow EC interoperability/ reproducibility expectations. Compliance with “Ideal” Specifications Enhances interoperability with other EarthCube products. “Optional Next-Level Community” Specifications allow products to conform to specifications that enhance compatibility with selected EC frameworks such as Geocodes and other best practices (e.g., DOI and ORCID assignment, code management in GitHub, a validation suite, a demonstration suite with notebooks, ESIP Community Ontology Repository for semantics, etc.)*

**Acknowledgments:** The authors wish to acknowledge several important efforts that led to this Specifications document, including a 2019 white paper from the Earth Cube Technology and Architecture

Committee following work by their Standards and Specifications Working Group [LINK], a 2018 report by the Architecture Refinement Workshop [LINK], and a 2020 white paper co-authored by members of the Council of Funded Projects and the EarthCube Office [LINK].

**Cite as**: Rubin, K.; Daniels, M.; Fulker, D.; Brown, J.; RIchard, S.; Meier, O.; Zaslavsky, I.; Willis, C.; McHenry, K.; Kirkpatrick, C. (2020). Recommended Standards and Specifications for EarthCube Projects. EarthCube Leadership Council: 7 May 2020. In EarthCube Organization Materials. UC San Diego Library Digital Collections. https://doi.org/10.6075/J0QR4VMG

## OTHER RESOURCES - see also [LINK]

### EarthCube Resources:

1. GeoCODES [Link]
2. EarthCube Resource Registry [Link] [entry form] [vocabularies] (see also information fields available at registration in Table 2 of [Link])
3. Earth Cube Data Discovery Tool [link]
4. Pangeo Cloud Notebook deployments [Link] and data analysis examples [Link]

### FAIR Resources

5. FAIR Data Principles - seminal paper [Link]
6. FAIR Data - background and implementation (AGU's list of resources) [Link]
7. EarthCube position paper regarding FAIR data and data resources: http://earthcube.org/fair
8. COPDESS Enabling FAIR Data Project (AGU Endorsed) [Link]
9. FORCE 11 Data Citation Principles [Link]

## Licensing Information:

10. Open Source Software Licensing [Link]
11. Trends in popularity of open source software licensing [Link]
12. Creative Commons licenses (including CC0 and CC-BY) [Link]

### Related Community Data and Software Resources:

13. Schema.org development for geoscience use (Science on schema.org) [Link]
14. Data on the Web Best Practices (W3C Recommendation) [Link]
15. Spatial Data on the Web Best Practices (W3C & OGC Recommendation) [Link]
16. Data Citation Guidelines v2 (ESIP Assembly Endorsed) [Link]
17. Software and Services Citation Guidelines and Examples (ESIP Assembly Endorsed) [Link]
18. Science Software Guidelines (ESIP Assembly Endorsed) [Link]
19. Earth Science Data Analytics Techniques - type list (ESIP Assembly Endorsed) [Link]
20. Registered Ontologies (ESIP COR - Community Ontology Repository) [Link]
21. Computational Infrastructure for Geodynamics Community Software Development Best Practices [Link]
22. xSDK Community Package Policies [Link] or [Link]