NDS@NCSA Hackathon

an experiment in community development

Ray Plante, Matt Turk

National Center for Supercomputing Applications

University of Illinois

Why a hackathon?

- To cultivate a open community of developers
  - Assist with building out needed software
  - Explore requirements for a development framework
  - Inspire innovative ideas
- Envisioning a series of hackathons
  - Gather developers from across the consortium for 2-4 days of side-by-side, collaborative development
  - Establish some open-source projects
    - Collect developer teams around those projects
    - Support continued development beyond meeting
  - Grow participation over time
- Start with a small, informal proto-hackathon

NDS Hackathon at NCSA

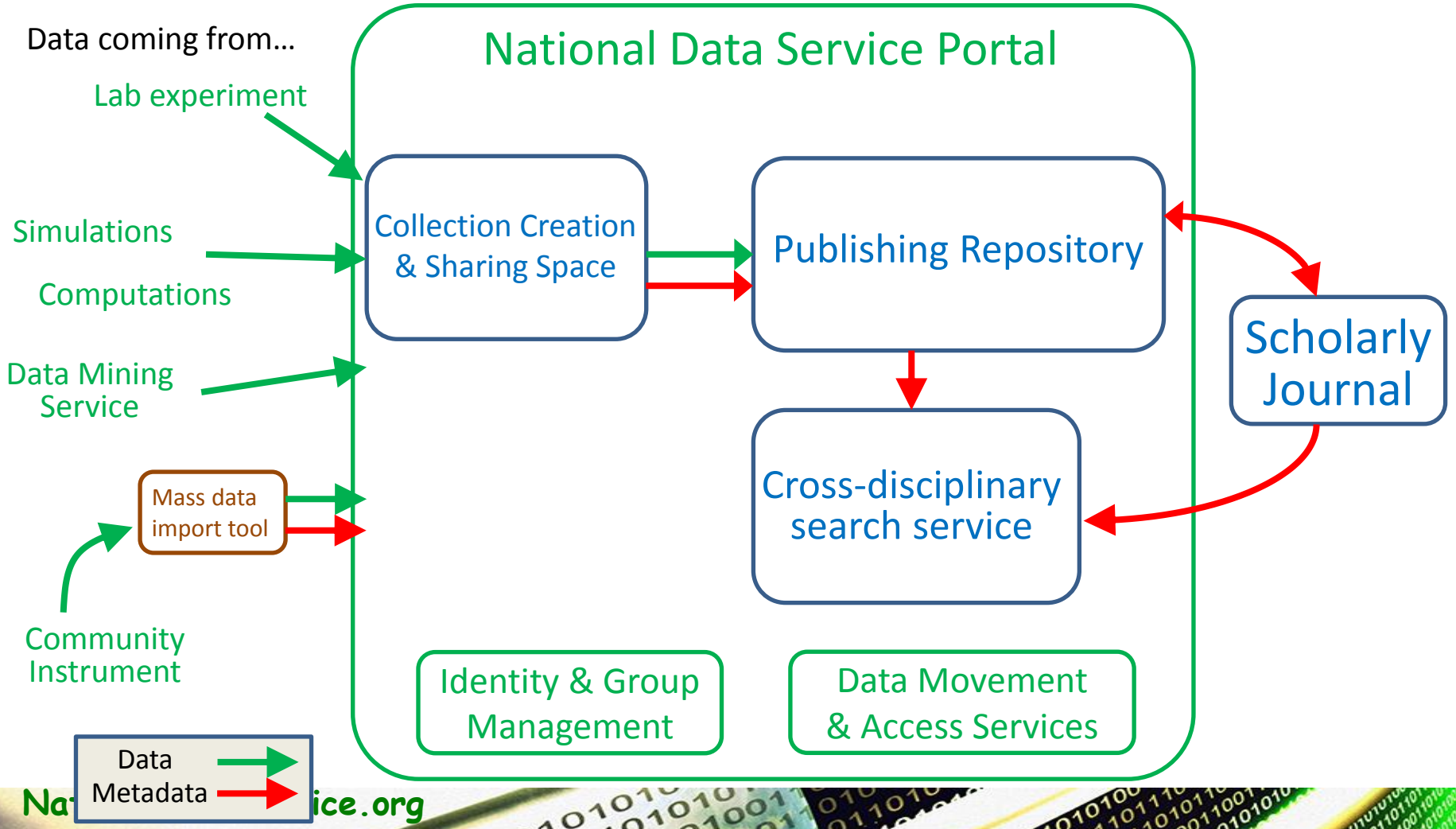
- September 17-19, 2014
- External Participants
  - Jim Myers (U of Michigan)
    - SEAD Project enables scientists to create and publish collections
  - Dmitry Mishin (SDSC/JHU)
    - Primary developer for SciDrive, a “Dropbox” for science data
  - Deoyani Nantrekar (JHU)
    - Developer in JHU-IDIES lab
  - Kacper Kowalik (UTexas Austin)
    - Contributing developer to yt community software package
  - Amit Chourasia (SDSC)
    - Leads SEEDME, a service for sharing research results quickly
- NCSA-local developers

  Mario Falarca, Tom Habing, Ray Plante, Tom Redman, Matt Turk, Venkat Yekkirala
- Theme: Can we connect these tools in a useful way?

Winding up the developers

- Before meeting
  - Posted ideas to a Trello page (http://trello.com/b/CA66J4cB/september-hackathon)
  - Established NDS presence in open-source repositories
    - Use not required but encouraged
    - GitHub: https://github.com/nds-org
    - BitBucket: https://bitbucket.org/nds-org
- Start of meeting
  - Plante: NDS Context, motivation, NDS framework
  - Developers introduced relevant work
  - Development ideas pitched and discussed
- Development
  - Broke into 2 teams, 1.5 days of development
- Wrap-up
  - Report on results

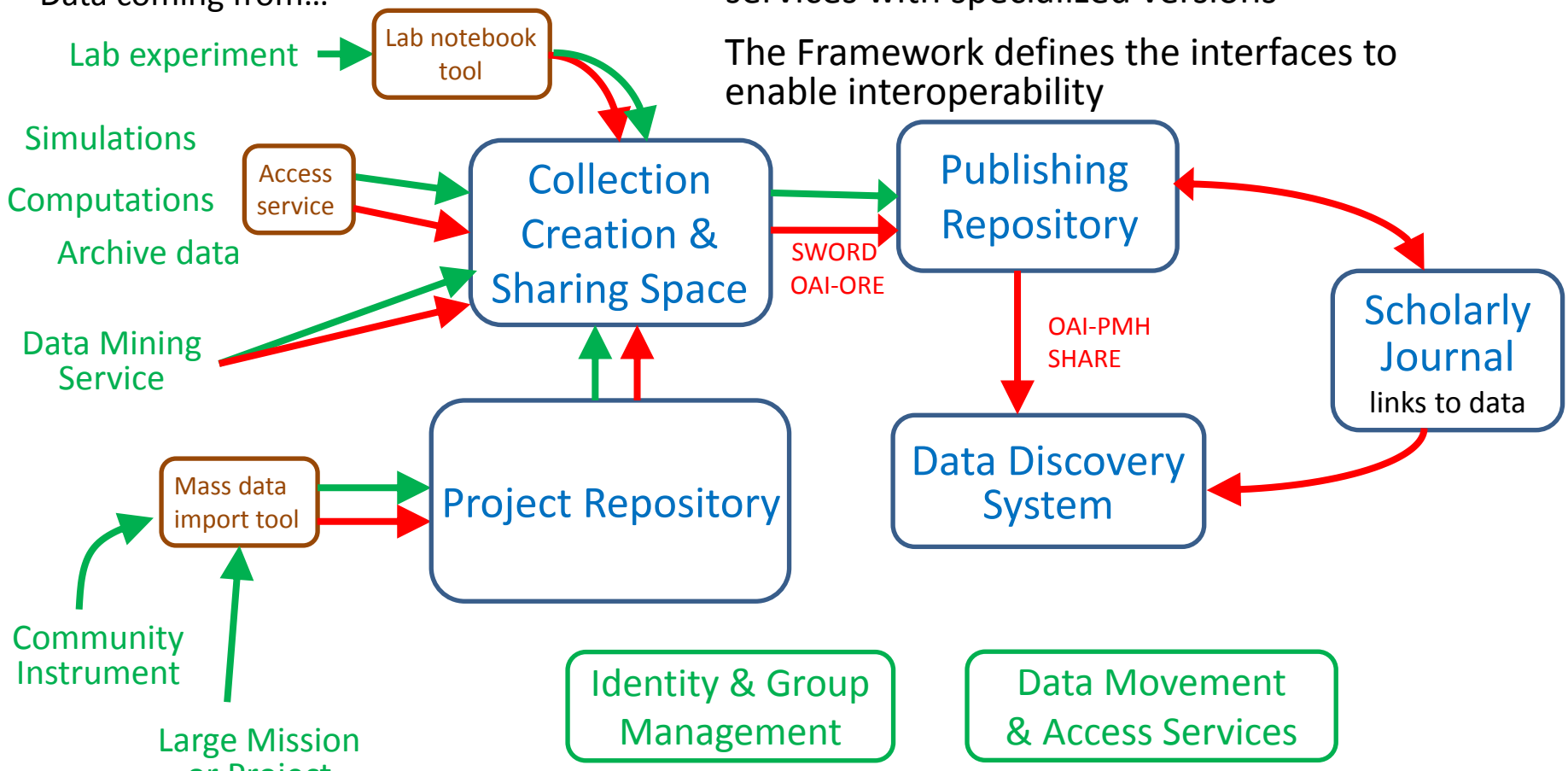
Thinking about the Framework



Thinking about the Framework

Communities can replace any/all of the vanilla services with specialized versions

The Framework defines the interfaces to enable interoperability



NationalDataService.org

Project 1: Connecting SciDrive to SEAD

- Background
  - SEAD =
    - Provides service for creating publishable collections (via Medici)
    - Delivers collections to one of several possible repositories (via Virtual Archiver)
  - SciDrive
    - "Dropbox" for scientific data
    - Features plugin mechanism for automatically executing operations on data in a folder
      - Used, e.g., to extract metadata, load tables into database
- Scenario
  - Research group uses SciDrive to share data products informally
  - Some metadata for products are extracted/created in SciDrive
  - Want to move data *and metadata* to SEAD to prepare for publishing
- What we built
  - Plugin for SciDrive for creating and editing metadata
  - Defined simple "standard" for accessing metadata
    - REST service: give PID, get back metadata in JSON-LD format
  - Implemented service in both SciDrive and SEAD

Project 2: Attaching Processing to archived data

- Motivation
  - Emerging Epicyte Pilot (see next talk)
  - Make large simulation result accessible for analysis
- What we built
  - iRODS-based data archive
  - Use ownCloud to pull data from different systems, including DropBox and SciDrive
  - Docker containers hosting IPython notebooks
    - Uploaded scripts can access portions of simulation data
  - SEEDME storage that can collect analysis products along with viewers and metadata

Shedding light on the framework

- Demonstrated 2 mechanisms for interoperability
  - (simple, well-defined) standards
    - REST API for accessing metadata
    - Exiting standard format: JSON-LD
  - Leveraging existing, non-standard but well-documented APIs
    - ownCloud's support for multiple storage systems
    - Can aggregate several tools through a few custom connections
- Developer communities can be cultivated around open (source) development