

Functionality:

1. Informal, private in-group data sharing
2. Collection creation
3. Publishing collections into a repository
4. Data access via repositories
5. Data-literature linking
6. Authentication and group management
7. Reliable data transfer
8. Analysis platforms
9. Data Discovery

Which of these can be served readily by vanilla services?

Which are important for community-specific specialization?

How should a framework balance dependency on standards versus many-to-many custom connections?

Should NDS recommend (to RDA) the development of new standards or recommendations?

What cross-disciplinary discovery capabilities will be important to enable?

What do developers and community projects need to understand how to plug into the framework?

What do you see as the most challenging interoperability issues we will need to address?

How do we maximize reuse of existing capabilities?

What mechanisms should be put into place to measure/track impact?

Questions behind disc:

Should NDS be the default service for the unprovided for (archive of last resort)?

How does NDS encourage providers and not compete them out of existence?

NDS vanilla services as reference implementations: demonstrates interoperable services

Standard way of associating data & metadata

Standard way of delivering data-metadata between system

NDS would have core services that could be assembled in different ways, e.g. into a new facility, customized to a community. NDS as an incubator

What is NDS?

- communities are not willing to give up on their community services
- NDS as an IETF Interop for RDA

#5 Data linking: good to have vanilla solution

Are we asking the wrong questions? How much of a gap analysis is needed?

Not sure if those 9 things are the ones they cared about?

Hard to focus on 9 things at once. Recommend a phased approach.

NDS working hand-in-hand with RDA

Metadata Problem:

Standards as it relates to discovery

Vanilla: data discovery at top

Also 5,6,7; should be common across arch.

Others closer to the community

Dataset to dataset linking (dependencies, derivations)

Reference implementations for maximizing re-use

Take best of breeds and link things together into a reference implementation, using standards

Tracking: citations, usage via common framework

Need for tiered data access. Users should be able to decide their own roles, controls.

Vanilla services: Globus Nexus , ORCHID (note difference b/w authentication and authorization)

Standards for passing data around (OAI-ORE, SWORD)

Architecture: microservice: many small services exposed through rest interfaces

[Reference: Heroau 12 principles for horizontal scaling]

Technologies that prevent people from realizing where things happen

Questions: what role is NDS playing? What is a collection, repository, ...

Impact metrics: like github model: can users easily fork a dataset that naturally tracks provenance. Visually display “activeness” (e.g. downloads, linking).

Good idea: provide a framework more important than being **the** solution.

Metadata, standards important. Mechanisms to “force” generation of standard metadata.

Look to see data sharing/publishing become the common

Need very easy to use tool for capturing metadata **early** in the process, build a metadata wrapper.

Encourage the publication of campus data to national repositories. Expect will require standard packaging mech.

Discussion touches on governance issues.

#2 one of the best suited for a vanilla implementation

Observation: metadata extraction is a key capability and good starting point for many interoperability issues.