Provenance Aware Synthesis
Tracking Architecture – PASTA

Mark Servilla – NIS Lead Developer
2012 Mid-term Review
PASTA Roadmap

- Where have we come from?
- Where are we today?
- Where do we go from here?
Where have we come from?

- Humble beginnings in EcoTrends
  - “Pasta” is now “PASTA”

- Key attributes:
  - Ecological Metadata Language
  - Provenance aware
  - Synthesis and derived data enabling
  - Data package = Metadata + Data
  - Limit impact on sites
  - Service Oriented Architecture/Web-service API
  - Transparency and community engagement
Where have we come from?

- Humble beginnings in EcoTrends
  - “Pasta” is now “PASTA”

- Key attributes:
  - Ecological Metadata Language
  - Provenance aware
  - Synthesis and derived data enabling
  - Data package = Metadata + Data
  - Limit impact on sites
  - Service Oriented Architecture/Web-service API
  - Transparency and community engagement
Where have we come from?

- Humble beginnings in EcoTrends
  - “Pasta” is now “PASTA”
- Key attributes:
  - Ecological Metadata Language
  - Provenance aware
  - Synthesis and derived data enabling
  - Data package = Metadata + Data
  - Limit impact on sites
  - Service Oriented Architecture/Web-service API
  - Transparency and community engagement
Where have we come from?

- Humble beginnings in EcoTrends
  - “Pasta” is now “PASTA”

- Key attributes:
  - Ecological Metadata Language
  - Provenance aware
  - Synthesis and derived data enabling
  - Data package = Metadata + Data
  - Limit impact on sites
  - Service Oriented Architecture/Web-service API
  - Transparency and community engagement
Where have we come from?

- Humble beginnings in EcoTrends
  - “Pasta” is now “PASTA”

- Key attributes:
  - Ecological Metadata Language
  - Provenance aware
  - Synthesis and derived data enabling
    - Data package = Metadata + Data
  - Limit impact on sites
  - Service Oriented Architecture/Web-service API
  - Transparency and community engagement
Where have we come from?

- Humble beginnings in EcoTrends
  - “Pasta” is now “PASTA”

- Key attributes:
  - Ecological Metadata Language
  - Provenance aware
  - Synthesis and derived data enabling
  - Data package = Metadata + Data
  - Limit impact on sites
  - Service Oriented Architecture/Web-service API
  - Transparency and community engagement
Where have we come from?

- Humble beginnings in EcoTrends
  - “Pasta” is now “PASTA”

- Key attributes:
  - Ecological Metadata Language
  - Provenance aware
  - Synthesis and derived data enabling
  - Data package = Metadata + Data
  - Limit impact on sites
  - Service Oriented Architecture/Web-service API
  - Transparency and community engagement
Where have we come from?

- Humble beginnings in EcoTrends
  - “Pasta” is now “PASTA”

- Key attributes:
  - Ecological Metadata Language
  - Provenance aware
  - Synthesis and derived data enabling
  - Data package = Metadata + Data
  - Limit impact on sites
  - Service Oriented Architecture/Web-service API
  - Transparency and community engagement
PASTA

Existing LTER metadata infrastructure (Metacat and EML)

Source A

Source B

Source C

Metacat-Harvester

EML

EML.xml

Metadata describing derived data, including data provenance and data versioning – expand on community provenance research

Metadata specific search engine for very fast data discovery

Data Manager Library

Cache

Workflow Engine

Support for multiple scientific workflow engines (e.g., R script, Kepler, Chimera, D2K)

Derived Data

Metadata

Web API

HTML

SOAP

Metadata and derived data products; metadata as EML

Standard interfaces to support various web portals (e.g., Trends, GEOSS, GEON, NEON, WATERS) and web service APIs

Dataset Registry

Event driven data loading for synthetic processing (e.g., new data, metadata change)
Operational Plan

Metadata generation for NIS Data Modules, including provenance and version information.

Value-adding applications
- LTER Data Portal
- DataONE Member Node
- CUAHSI Hydro-data
- EcoTrends Portal
1. Data Manager, Catalog, Loader, & Cache
2. Workflow Manager
3. Metadata Factory
4. NIS Data Portal
5. Discovery/Access API
6. Identity Management Services
7. Persistent Identifier Services
8. System Monitor
9. Audit Services
PASTA – Operational Plan

1. Data Manager, Catalog, Loader, & Cache
2. Workflow Manager
3. Metadata Factory
4. NIS Data Portal
5. Discovery/Access API
6. Identity Management Services
7. Persistent Identifier Services
8. System Monitor
9. Audit Services
PASTA – Operational Plan

1. Data Manager, Catalog, Loader, & Cache
2. Workflow Manager
3. Metadata Factory
4. NIS Data Portal
5. Discovery/Access API
6. Identity Management Services
7. Persistent Identifier Services
8. System Monitor
9. Audit Services
PASTA – Operational Plan

1. Data Manager, Catalog, Loader, & Cache
2. Workflow Manager
3. Metadata Factory
4. NIS Data Portal
5. Discovery/Access API
6. Identity Management Services
7. Persistent Identifier Services
8. System Monitor
9. Audit Services
PASTA – Operational Plan

1. Data Manager, Catalog, Loader, & Cache
2. Workflow Manager
3. Metadata Factory
4. NIS Data Portal
5. Discovery/Access API
6. Identity Management Services
7. Persistent Identifier Services
8. System Monitor
9. Audit Services
PASTA – Operational Plan

1. Data Manager, Catalog, Loader, & Cache
2. Workflow Manager
3. Metadata Factory
4. NIS Data Portal
5. Discovery/Access API
6. Identity Management Services
7. Persistent Identifier Services
8. System Monitor
9. Audit Services
PASTA – Operational Plan

1. Data Manager, Catalog, Loader, & Cache
2. Workflow Manager
3. Metadata Factory
4. NIS Data Portal
5. Discovery/Access API
6. **Identity Management Services**
7. Persistent Identifier Services
8. System Monitor
9. Audit Services
PASTA – Operational Plan

1. Data Manager, Catalog, Loader, & Cache
2. Workflow Manager
3. Metadata Factory
4. NIS Data Portal
5. Discovery/Access API
6. Identity Management Services
7. Persistent Identifier Services
8. System Monitor
9. Audit Services
PASTA – Operational Plan

1. Data Manager, Catalog, Loader, & Cache
2. Workflow Manager
3. Metadata Factory
4. NIS Data Portal
5. Discovery/Access API
6. Identity Management Services
7. Persistent Identifier Services
8. System Monitor
9. Audit Services
PASTA – Operational Plan

1. Data Manager, Catalog, Loader, & Cache
2. Workflow Manager
3. Metadata Factory
4. NIS Data Portal
5. Discovery/Access API
6. Identity Management Services
7. Persistent Identifier Services
8. System Monitor
9. Audit Services
Design Evolution
Noteworthy Changes...

- **Persistent Identifier Services**
  - PASTA resource URI’s already a persistent identifier
  - DOI integration to come in next phase

- **Workflow Manager**
  - Support directed toward “event management”
  - User managed workflow much greater flexibility
  - Reduced development complexity

- **Discovery/Access API**
  - Not a developed component
  - Rather, the collective of all PASTA service APIs

- **Metadata Factory**
  - Generates only metadata fragment for provenance, thus “Provenance Factory”
  - Level-1 metadata created by Data Package Manager
Noteworthy Changes...

- **Persistent Identifier Services**
  - PASTA resource URI’s already a persistent identifier
  - DOI integration to come in next phase

- **Workflow Manager**
  - Support directed toward “event management”
  - User managed workflow much greater flexibility
  - Reduced development complexity

- **Discovery/Access API**
  - Not a developed component
  - Rather, the collective of all PASTA service APIs

- **Metadata Factory**
  - Generates only metadata fragment for provenance, thus “Provenance Factory”
  - Level–1 metadata created by Data Package Manager
Noteworthy Changes...

- **Persistent Identifier Services**
  - PASTA resource URI’s already a persistent identifier
  - DOI integration to come in next phase

- **Workflow Manager**
  - Support directed toward “event management”
  - User managed workflow much greater flexibility
  - Reduced development complexity

- **Discovery/Access API**
  - Not a developed component
  - Rather, the collective of all PASTA service APIs

- **Metadata Factory**
  - Generates only metadata fragment for provenance, thus “Provenance Factory”
  - Level–1 metadata created by Data Package Manager
Noteworthy Changes...

- **Persistent Identifier Services**
  - PASTA resource URI’s already a persistent identifier
  - DOI integration to come in next phase

- **Workflow Manager**
  - Support directed toward “event management”
  - User managed workflow much greater flexibility
  - Reduced development complexity

- **Discovery/Access API**
  - Not a developed component
  - Rather, the collective of all PASTA service APIs

- **Metadata Factory**
  - Generates only metadata fragment for provenance, thus “Provenance Factory”
  - Level–1 metadata created by Data Package Manager
PASTA Service Oriented Architecture

- NIS Data Portal
- Gatekeeper
  - Provenance Factory
  - Audit Manager
  - Event Manager
- Data Package Manager
  - Quality Engine
    - Metadata Manager
    - Data Manager
      - Metadata Catalog
      - Data Catalog
Where are we today?

1. NIS Data Portal
2. Gatekeeper
3. Audit Manager
4. Data Package Manager
   - Quality Engine
   - Metadata Manager/Catalog
   - Data Manager/Catalog
5. Event Manager
6. Provenance Factory
7. System Monitor
Audit Manager

PASTA

System Monitor

NIS Data Portal

Gatekeeper

Provenance Factory

Audit Manager

Event Manager

Data Package Manager

Quality Engine

Metadata Manager

Data Manager

Metadata Catalog

Data Catalog
Data Package Manager

- NIS Data Portal
  - System Monitor
  - Gatekeeper
    - Provenance Factory
    - Audit Manager
    - Event Manager
    - Data Package Manager
      - Quality Engine
        - Metadata Manager
        - Data Manager
          - Metadata Catalog
          - Data Catalog

PASTA
Event Manager
Provenance Factory

NIS Data Portal

Gatekeeper

PASTA

System Monitor

Provenance Factory

Audit Manager

Event Manager

Data Package Manager

Quality Engine

Metadata Manager

Data Manager

Metadata Catalog

Data Catalog
System Monitor
Practical Innovations

- Interoperability
- Provenance
- Access
- Quality

- Workflow Demonstration (recorded)
Development In Practice

- Community Engagement
- PASTA Service Software Stack
- Software Process
Community Engagement

- Tiger Team
- IM Product Oriented Working Groups
- NIS Community Drupal website
- NIS Monthly Updates
NIS Tiger Teams

- Small groups (4–6 each) of LTER scientists, IMs, and students
- Short duration 4–8 months for 2–3 hrs/week
- To provide feedback, direction, & sign-off on specific software component
- Nine groups, one for each original OP PASTA component
Community Engagement

- Tiger Team
- IM Product Oriented Working Groups
- NIS Community Drupal website
- NIS Monthly Updates
LTER IM Working Groups

- GeoNIS – Nov 2011, Feb 2012
  - Provided requirements for geospatial data support

- EML Metrics – Mar 2012
  - Guided development of the Data Package Manager – Quality Engine

- Workflow – Mar 2012
  - Guided development in Data Package Manager “read resource”
  - Guided development in Event Manager
Community Engagement

- Tiger Team
- IM Product Oriented Working Groups
- NIS Community Drupal website
- NIS Monthly Updates
Community Engagement

- Tiger Team
- IM Product Oriented Working Groups
- NIS Community Drupal website
- NIS Monthly Updates
PASTA Service Software Stack

Filesystem (LINUX) -> RDBMS (PostgreSQL) -> Service Logic -> Web-Service API (JAX-RS) -> Application Server (Tomcat, Jetty) -> Java 1.6 SDK -> Web Server (Apache) -> Web-Service Interface
Software Process

- Agile RUP/OpenUP
  - Inception, Elaboration, Construction, & Transition
- Subversion code repository – public read
  - https://svn.lternet.edu
- Trac project management
  - https://trac.lternet.edu
- Confluence design wiki
  - https://nis.lternet.edu:8443/display/pasta/Home
Software Process

- Agile RUP/OpenUP
  - Inception, Elaboration, Construction, & Transition
- Subversion code repository – public read
  - https://svn.lternet.edu
- Trac project management
  - https://trac.lternet.edu
- Confluence design wiki
  - https://nis.lternet.edu:8443/display/pasta/Home
Agile RUP/OpenUP

- An agile version of Rational Unified Process
  - Inception
  - Elaboration
  - Construction
  - Transition

![Diagram showing project phases and milestones](image-url)
Inception

Resource Read:

Client

Gatekeeper

Service

GET: URI

isAuthN'd (token)

isAuthZ'd (service)

isAuthZ'd (URI)

readResource (URI)

GET + URI = Service

URI = ResourceId
Elaboration

PASTA AuthN/AuthZ - 1 Sep 2010

Authorization - 1 September 2010

Access rules as applied in current Metacat instances and explicit Metadata Language document for both metadata content and data access (see:
ecsc.org/software/eml/eml-2.1.0/eml-access.html#access):
- The metadata document retain full or "all" access rights (implied).
- Applied in EML access rules can perform authorized rights, including updating metadata content with the exception of changing (explicit).
- Element does not include the optional "access" element, then user has access rights to the package (implied).
- An "All", "public", exists to denote any user or group -- anonymous
- Applies to both the Metadata Management and Data Management
- In an LTER LDAP registry - currently, no group associations exist.
- Both individuals and groups.
- Associated with "roles".
- By default, belong to a minimum of 1 group

"Core service" will require a valid group assignment; actions on metadata or data may require individual authentication.
1. Non-authenticated users will belong to the "anonymous" group.
2. A "revoked" group will exist to identify users whose access rights have been revoked; revocation can affect implied rights given to the submitters of metadata and data (see 1.1).
3. Authorization is the responsibility of each PASTA "core service".
1. Authorization rules are "role-based"; roles are applied only to groups (with the exception of individual access rules applied to metadata and data access as stated in a.1).
2. Authentication should be extensible to external groups, if possible (i.e., federation of authentication, and therefore the use of external identity authorities, should be considered in the overall design).

Example of access rule in EML
< acest
hsystem="ldap://ldap.ecoinformatics.org:389/dc=ecoinformatics,dc=org"
for="allowFirst">
  <allow>
    <principal uid="alice",o=NASA,dc=ecoinformatics,dc=org>/principal>
package edu.internet.pasta.common.security.authorization;

import java.io.*;
import java.util.ArrayList;
import java.util.Hashtable;
import java.util.Set;
import edu.internet.pasta.common.security.token.*;
import edu.internet.pasta.common.security.authorization.Rule.Permission;

/**
 * @author servilla
 * 
 * Create a PASTA access matrix object based on either an XML access element or
 * an existing access matrix rule set.
 */
public class AccessMatrix {

/* Class Fields */

/* Instance Fields */

1.1 (0/7024)
Submission #57

Form: Tiger Team Acceptance Testing Confirmation
Submitted by sbohn
Wednesday, December 14, 2011 - 10:36
207.229.236.211

Service:
Identity Manager - Authentication

Accept or Decline:
Accept with Conditions (state conditions in Comments field)

Comments:
We should return 401 errors rather than 500 errors on authentication failure. When I modify a token, the failure returns a 500 error. To test: "sucubus features/authentication/feature" from https://github.com/...
Software Process

- Agile RUP/OpenUP
  - Inception, Elaboration, Construction, & Transition
- Subversion code repository – public read
  - https://svn.lternet.edu/svn/NIS
- Trac project management
  - https://trac.lternet.edu
- Confluence design wiki
  - https://nis.lternet.edu:8443/display/pasta/Home
Software Process

- Agile RUP/OpenUP
  - Inception, Elaboration, Construction, & Transition
- Subversion code repository – public read
  - https://svn.lternet.edu/svn/NIS
- Trac project management
  - https://trac.lternet.edu
- Confluence design wiki
  - https://nis.lternet.edu:8443/display/pasta/Home
Software Process

- **Agile RUP/OpenUP**
  - Inception, Elaboration, Construction, & Transition

- **Subversion code repository – public read**
  - https://svn.lternet.edu/svn/NIS

- **Trac project management**
  - https://trac.lternet.edu

- **Confluence design wiki**
  - https://nis.lternet.edu:8443/display/pasta/Home
Measurements of Success

- Sign-off by Tiger Teams
- Annual review by NISAC
- Report back to Executive Board
- Feedback from community
Where do we go from here?

- Support DOI assignment to metadata and data objects
- Refine NIS Data Portal
- Standup DataONE member node
- Accept InCommon/CILogon user credentials
- Optimize Data Package Manager core
  - Metadata Manager/Catalog integration
  - Improved query processing
- Evolve our Community Engagement
June–Sept 2012
   ◦ Refine existing prototype
   ◦ Beta–release 2012 LTER All Scientists Meeting
   ◦ Lessons learned/self–evaluation
   ◦ Planning for Phase 2

Oct 2012 – June 2014 (1.75 yrs)
   ◦ Execute Phase 2
Thank you!