Decentralizing SHARE
Bringing SHARE closer to the community and the community closer to SHARE
SHARE is a community open-source initiative developing tools and services to connect related, yet distributed, research outputs, enabling new kinds of scholarly discovery.

@SHARE_research

www.share-research.org
Strategy and Focus

» Begin with local problems, local needs, an eye towards shared use
» A shift from 100% dedicated support at Center for Open Science (COS) to Community support and infrastructure model
» Greater institutional control and self-sufficiency
Local
Notre Dame: Campus

Ongoing conversations with Provost Office

» Investigate ways to streamline and improve the annual reporting of centers and institutes. Currently, each center/institute director must manually determine scholarly productivity by polling faculty, reviewing their CVs, etc. Furthermore, automate the process.
Notre Dame: Hesburgh Libraries

» Demonstrate impacts of centers, institutes on campus (i.e., which are the strongest areas/disciplines of impact at Notre Dame).
» Other direct requests from academic depts
» Allow us to increase the deposit rate in our IR
» Demonstrate library role and expertise
Notre Dame: Why Library?

» We have the expertise and content to feed this process
» Bring proof-of-concept
Notre Dame: Goals

» Visualize
» Query
» Manage
Intersection of API Task Force Charge and ND Research Activity

Goal – Increase % of deposit in IR

Phase 1 - API workflow
- Digital Measures
- citation exports
- WoS api

Phase 2 - (RAD)
- Metadata workflow: baseline elements, harvesting, creation, validation, quality assurance, authority control, etc.
- PubMed api

Phase 3 - Deposit workflow
- metadata record + doc + rights; merging & deduping between Ndlectual & CurateND

CurateND
- Self deposit
- ND research activity store
- Ndlectual deposit
- API sourced deposit
- Library sponsored deposit

Analytics/Reporting:
Univ. Admin., Deans, departments, individuals, etc.

Goal – Create local ID Store

Author IDs & Authority Control:
- Aleph
- OCLC
- Vendor Sources
- Other IRs

Departmental data
Notre Dame: SHARE’s Role

Integrate with

» Metadata harvesting and management tools
» Visualization dashboard and other tools
» Share data between institutions
Community
Community Building Overview

» Process
  ◦ Outreach
  ◦ Integrating feedback at every level

» Progress to date
  ◦ Monthly community calls
  ◦ Ramping up additional developers
  ◦ Contribution from 3 separate institutions - Notre Dame, Virginia Tech, 221b
  ◦ Continued involvement from COS
Community & Governance

» Community shared development
  ◦ Core technology
  ◦ Specific Applications (double-meaning here)
  ◦ Github and issue-tracking -> facilitating collaboration, lowering barrier to entry

» Governance

» Near-term trajectory
Technology
Greater Institutional Control
Harvesting Framework
Network and Indexing
Strategy - Shared community infrastructure
Active Community Partners

VT™  COS  UNIVERSITY OF NOTRE DAME
CENTER FOR OPEN SCIENCE  Hesburgh Libraries

ASSOCIATION OF RESEARCH LIBRARIES  221b
Node-Red

A flow-based development tool for visual programming
Main concepts

Nodes
- Wrapped as a Node.js module
- Executes a small subset of code

Flows
- Comprised of many nodes
- Wired together
- Deployed to its runtime to execute the nodes in succession
Creating a Node-Red node
<script type="text/javascript">
RED.nodes.registerType('lower-case', {
    category: 'share',
    color: '#eefafa',
    defaults: {
        name: {value: ''}
    },
    inputs: 1,
    outputs: 1,
    icon: "share.png",
    label: function() {
        return this.name||"lower-case";
    }
});
</script>

<script type="text/x-red" data-template-name="lower-case">
<div class="form-row">
    <label for="node-input-name"><i class="icon-tag"></i> Name</label>
    <input type="text" id="node-input-name" placeholder="Name">
</div>
</script>

<script type="text/x-red" data-help-name="lower-case">
    <p>A simple node that converts the message payloads into all lower-case characters</p>
</script>
module.exports = function (RED) {
  function LowerCaseNode(config) {
    RED.nodes.createNode(this, config);
    var node = this;
    node.on('input', function (msg) {
      msg.payload = msg.payload.toLowerCase();
      node.send(msg);
    });
  }
  RED.nodes.registerType("lower-case", LowerCaseNode);
}
{
  "name": "node-red-example",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" & exit 1"
  },
  "author": "",
  "license": "ISC",
  "node-red": {
    "nodes": {
      "lower-case": "lower-case.js"
    }
  }
}
Adding the node to Node-Red
Real world example
Moving Forward
THANKS!
Any questions?