AlphaFlow

Workflow Applications

What is Workflow?

- It defines the approval process
- It is the routing slip, not the content
- By design, workflow has no knowledge of what it is tracking

Common Workflows & Events

- Workflows
  - Linear
  - Parallel
  - Voting (majority, 2/3, 3/5)
  - Periodic reminders, restarts
- Events
  - Alarm (periodic)
  - Email
  - External (DB logging, …)

Workflow Use Cases

1) Role your own
   - No enterprise data; you manage roles
   - Zope's restricted Python, XML
   - Example: Windows Policy Committee
2) Simple Managerial Approval
   - May be a document, report, etc.
   - Example: CD Tracks
3) Complex Workflows With In-house Support
   - Python, Perl, Java, PL/Sql
   - Example: Leave Request, Miser

1) Role Your Own

- Minimal support from DBI/INF group
- Use the Plone site to create & maintain
  - AlphaFlow XML (states, roles, transitions, events)
  - Persons in roles (via script)
  - Front end application or form
- Maintain using Plone/AlphaFlow
  - Built-in GUIs
  - API calls
  - Zope's restricted Python scripts

Role Your Own - Example

- Windows Policy Committee Roles
  - representatives from different lab entities:
  - did not align with org-chart roles (no Experiments in org chart...)

2) Simple Managerial Approval

- No coding necessary!
- Uses generic workflow (supervisor or management chain)
- Based on Kerberos Principal of requester
- Initiated via simple web page
  - https://appora.fnal.gov/work_flow_ora/www/generic_wf_api.html
  - Select the workflow
  - Enter a title (email subject and title in Plone)
  - Enter the URL (what the approvers see)

3) Complex Workflows With In-house Support

- Write the AlphaFlow XML workflow
  - (we have examples and we can help)
- Create a URL with XML that identifies roles and persons in roles
  - May be a flat file, CGI script, etc.
  - (due to roll out on the 1st of March)
- Workflow APIs available in Python, Perl, Java and PL/SQL for:
  - Creating and activating workflow items
  - Calculating enterprise, person-in-role data

Enterprise Roles

- Leadership roles (head of organizational group)
  - Supervisor
  - Management chain
- Functional Leader
  - Head of a group that "owns" something
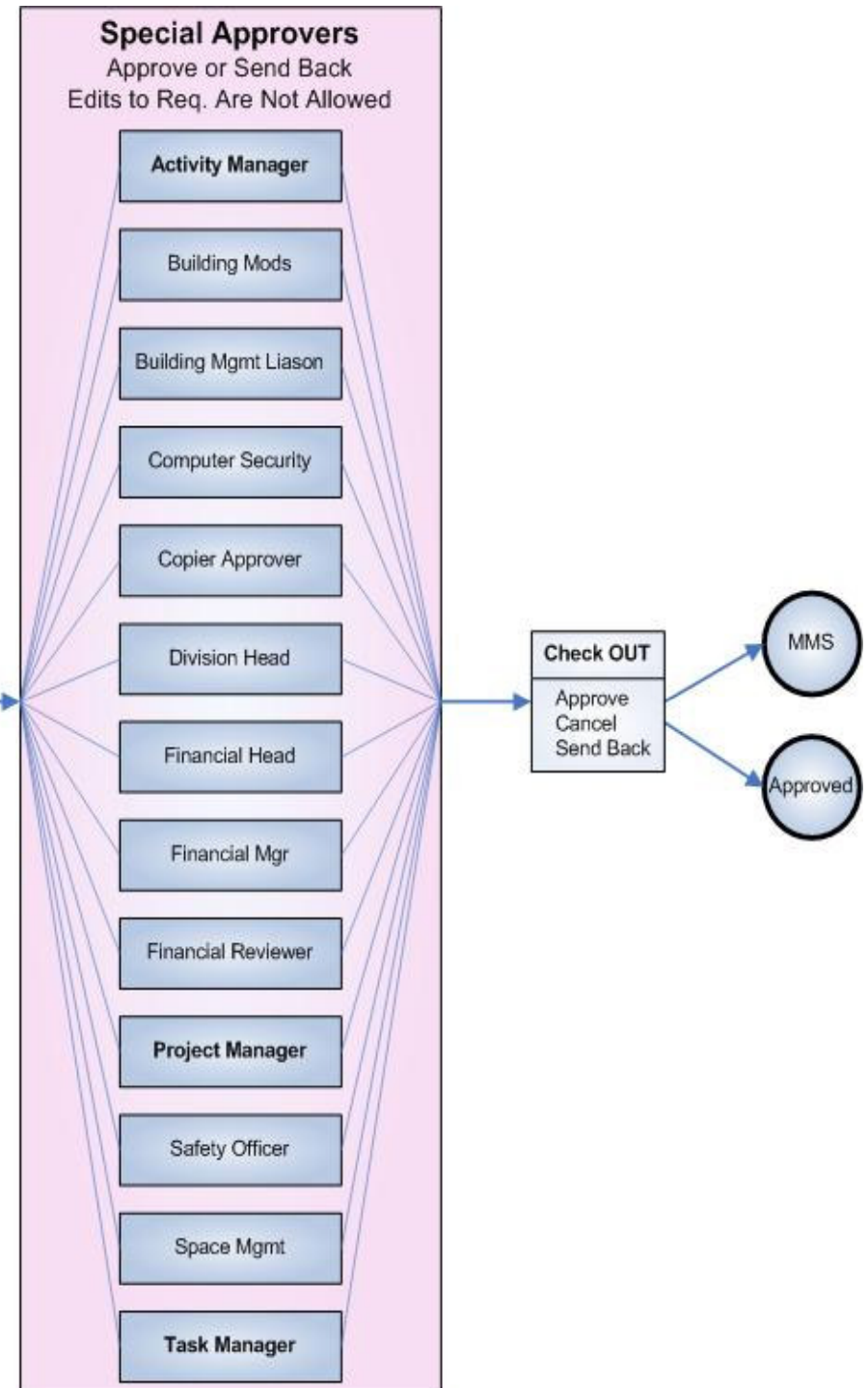 - Task manager, project manager, activity manager
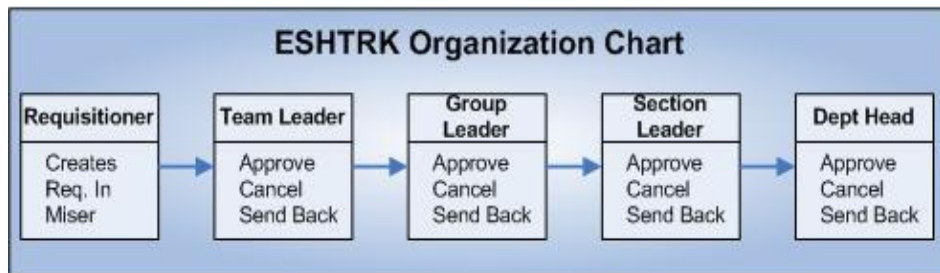
Enterprise Roles

- System Claims
  - System administrator
  - System owner
- Purely Functional
  - Safety officer
  - Computer security approver
  - Financial head, manager and reviewer
- New functional roles are easily added
- A list of Enterprise roles is available at
  http://appora.fnal.gov/pls/default/work_flow.wf_maint_lib.valid_roles

Advantages of Enterprise Data

- No redundant data maintenance
- Everyone is using the same **exact** data
- Stored and backed up in a central location on a stable system
- Easily regenerated when
    - A person "puts on a hat"
    - There is a reorganization

Miser Workflow
- What we think it looks like

Special Approvers
Approve or Send Back
Edits to Req. Are Not Allowed

- **Activity Manager**
- Building Mods
- Building Mgmt Liason
- Computer Security
- Copier Approver
- Division Head
- Financial Head
- Financial Mgr
- Financial Reviewer
- **Project Manager**
- Safety Officer
- Space Mgmt
- **Task Manager**

ESHTRK Organization Chart

Requisitioner	Team Leader	Group Leader	Section Leader	Dept Head
Creates Req. In Miser	Approve Cancel Send Back	Approve Cancel Send Back	Approve Cancel Send Back	Approve Cancel Send Back

Check OUT
Approve
Cancel
Send Back

MMS

Approved

Notes:
1) Expedite (via Miser) moves req. directly to Check OUT.
2) Send Back = Demote to any previous state.
3) Attachments are allowed in any state.
4) All roles have alternates
5) No duplicate approvals within areas (ESHTRK/SA)

Miser Workflow
- What it looks like in AlphaFlow
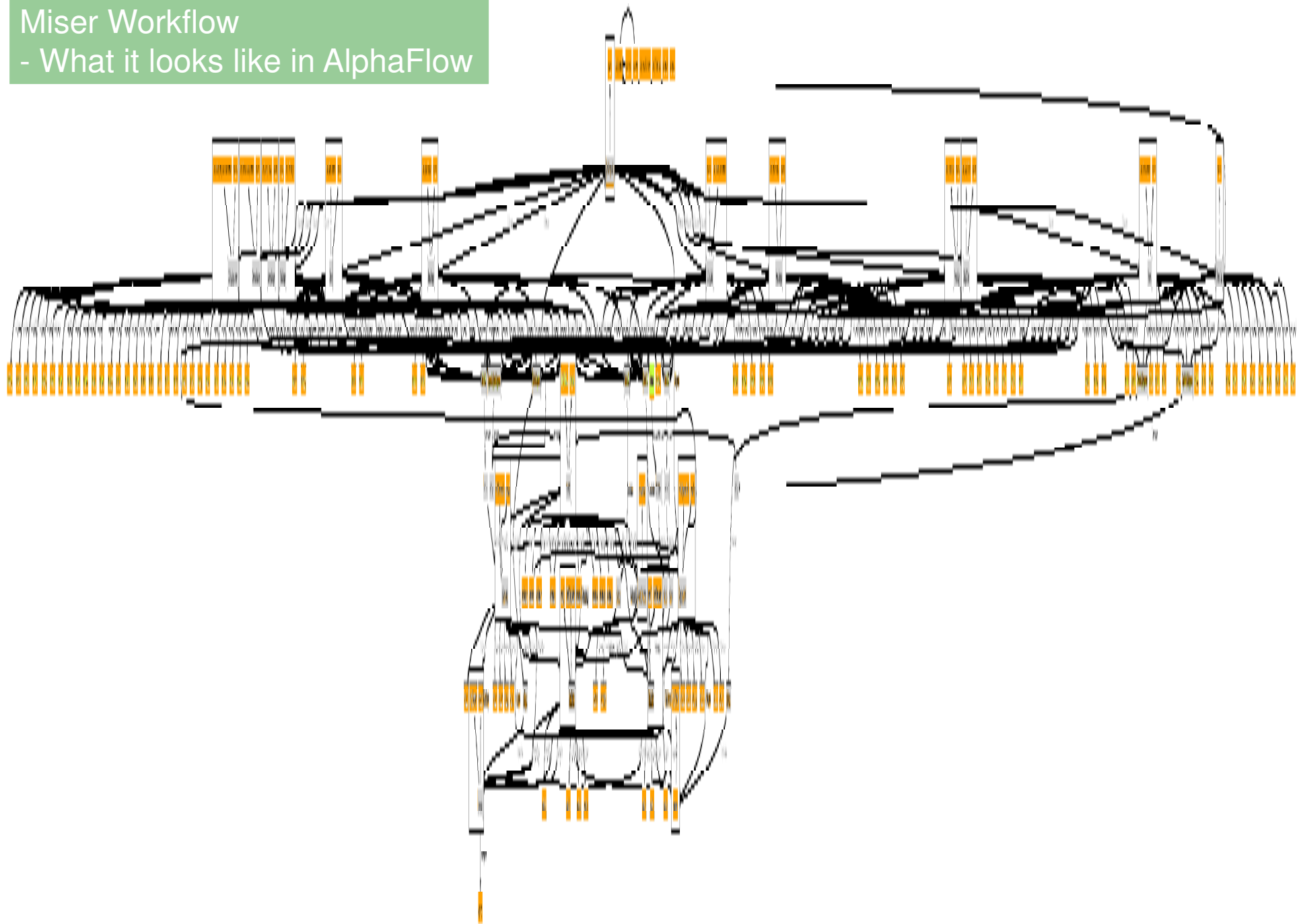
AlphaFlow XML Example

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE workflow SYSTEM "../../schemas/alphaflow.dtd">
<workflow
        id="leave_request_v1_9_10"
        title="Leave Request v1.9.10"
        startActivity="email_confirmation"
        description="Leave Request"
        object_name="cd_leave_request"
        onlyAllowRoles="Owner">
```

AlphaFlow XML Example

```
<!-- notify the owner -->
<email id="email_confirmation"
    title="Notify Owner"
    template="db_wf_template"
    mailSubject="Your Leave Request Has Been Submitted"
    continue_activity="supervisor_approval">
    <recipient type="actual_role" roles="Owner"/>
</email>
```

AlphaFlow XML Example

```xml
<ntask id="supervisor_approval"    title="Leave Request"
                startActivity="enter_supervisor notify_supervisor">
    <assignees kind="actual" roles="Supervisor" />
    <exit id="yes" title="Approve" activities="exit_yes
                notify_requester_yes" />
    <exit id="yes_wo" title="Approve without comments"
                activities="exit_yes notify_requester_yes" />
    <exit id="no" title="Deny"  activities="exit_no notify_requester_no" />
    <exit id="maybe" title="Request info" activities="exit_maybe
                notify_requester_maybe" />
</ntask>
```

AlphaFlow XML Example

```xml
<email id="notify_requester_yes"  title="Notify Leave Request Approved"
    template="db_wf_template"
    mailSubject="Leave Request Approved"
    continue_activity="enter_approved" >
    <recipient type="actual_role" roles="Owner"/>
    <recipient type="actual_role" roles="Supervisor"/>
</email>
```

Role XML

- Use enterprise APIs to generate XML for <u>known roles</u>
- Returns all XML between <ROLE> tags

```
<ROLE>
    <ROLE_NAME>Group_Leader</ROLE_NAME>
    <KERBEROS_PRINCIPAL>arenam</KERBEROS_PRINCIPAL>
    <IS_IN_ROLE>1</IS_IN_ROLE>
    <GROUP_NAME>FL/CD/LSC/DBI/INF</GROUP_NAME>
    ……   (remaining tags omitted)
</ROLE>
```

Role XML

- Add XML for your specific workflow

```
<ROLE>
    <ROLE_NAME>YourRoleNameHere</ROLE_NAME>
    <KERBEROS_PRINCIPAL>lauri</KERBEROS_PRINCIPAL>
    <IS_IN_ROLE>1</IS_IN_ROLE>
    <HEADED_BY_KP>arenam</HEADED_BY_KP>
</ROLE>
```

Web Documentation

- http://wwwserver2.fnal.gov/cfdocs/AlphaFlow