OAK RIDGE National Laboratory | LEADERSHIP COMPUTING FACILITY

May 6, 2025 | CUG

Employing a Software-Driven Approach to Scalable HPC System Management

Aaron Barlow

ORNL

U.S. DEPARTMENT OF ENERGY

ORNL IS MANAGED BY UT-BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY

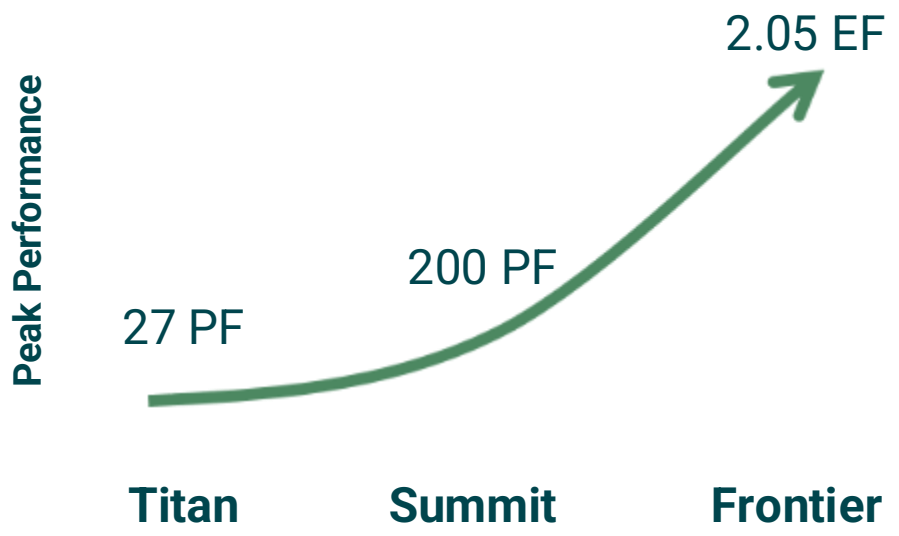
HPC System Management is Challenging

HPC Scale & Complexity

- Multi HPC system management
  - Over *30* allocatable HPE and non-HPE systems
- Thousands of users & research projects
- Massive resource allocations and enforcement
- Filesystem and permission management
- Identity and access management
  - Multi-system authentication (LDAP, OneID, XCAMS)

Operational Scale Challenges

- Manual processes don't scale & admin burden increases as systems and capabilities grow
- Our software supports the business of running HPC systems at our scale and complexity



Data provided by top500.org

System Management Progression

Line-by-Line Configuration

```
ilorest createuser "my_new_user" "password123" --role
  Login,RemoteConsole
useradd -m my_new_user
passwd my_new_user
groupadd user_group
usermod -aG user_group my_new_user
groupadd project123
# Additional group commands here
usermod -aG project007 my_new_user
mkdir -p /gpfs/project123/my_new_user
chown my_new_user:project123
/gpfs/project123/my_new_user
...
```

Abstracting with Bash Scripts

```
bash ./create_user.sh \
--username "my_new_user" \
--password "password123" \
--groups "user_group,project123,project007" \
--directories
"/gpfs/project123/my_new_user,/hpss/project007
/my_new_user" \
--with-filesystem \
--with-home-directory \
--with-directory-permissions
```

Example Manual System Management (formatting for readability)

Problems with Manual System Configuration

1. **Inconsistent Configuration Across Clusters**
    - Repeated manual setup across HPC systems is error-prone, time consuming, and leads to drift
2. **Configuration Complexity**
    - UNIX users, UNIX groups, filesystems, sudo rules, and scheduler configurations (SLURM and LSF)
3. **Slow Policy Enforcement**
    - Manual enforcement delays urgent changes such as ending compute allocations and user access
4. **No Audit Trail**

OLCF Solution

RATS CRM

- Handle operations of multiple large-scale HPC systems from a single application
- Authoritative source of truth that defines state of HPC systems via API endpoints

MyOLCF

- Reduce administrative burden through management and usage insight

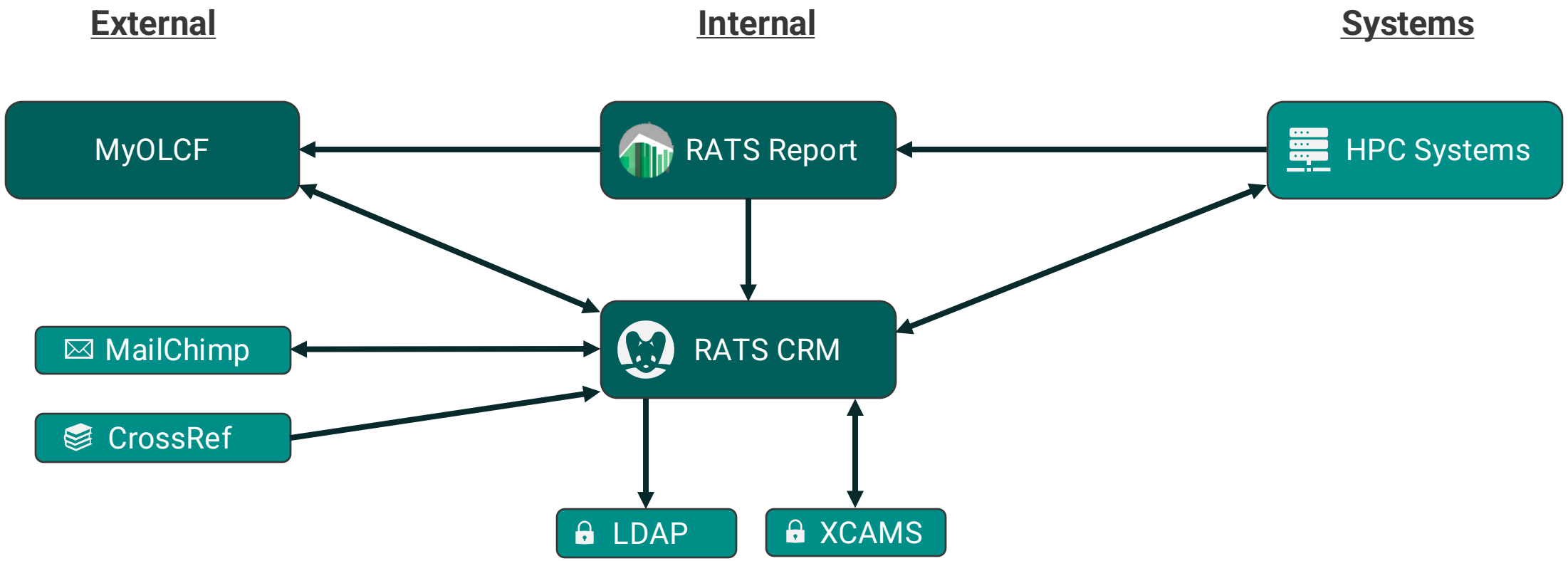
myOLCF

RATS Report

- Near real-time analytics & reporting of system compute & filesystem usage

** **RATS** stands for **R**esource **A**llocation **T**racking **S**ystem*
** **CRM** stands for **C**ustomer **R**elationship **M**anagement*

High-Level RATS CRM, MyOLCF, and HPC Integration



Arrows denote information flow direction

System Architecture Overview

RATS CRM

1. **Built With**
   - Ruby (Rails framework)
   - MariaDB
2. **DevOps & Deployment**
   - GitLab Runners (CI/CD)
   - Kustomize for declarative infrastructure
3. **History**
   - Created in 2012
   - Replaced legacy C++ and Bash scripts

RATS Report

1. **Built With**
   - Ruby (Rails framework)
   - MariaDB
2. **DevOps & Deployment**
   - GitLab Runners (CI/CD)
   - Kustomize for declarative infrastructure
3. **History**
   - Created in 2015

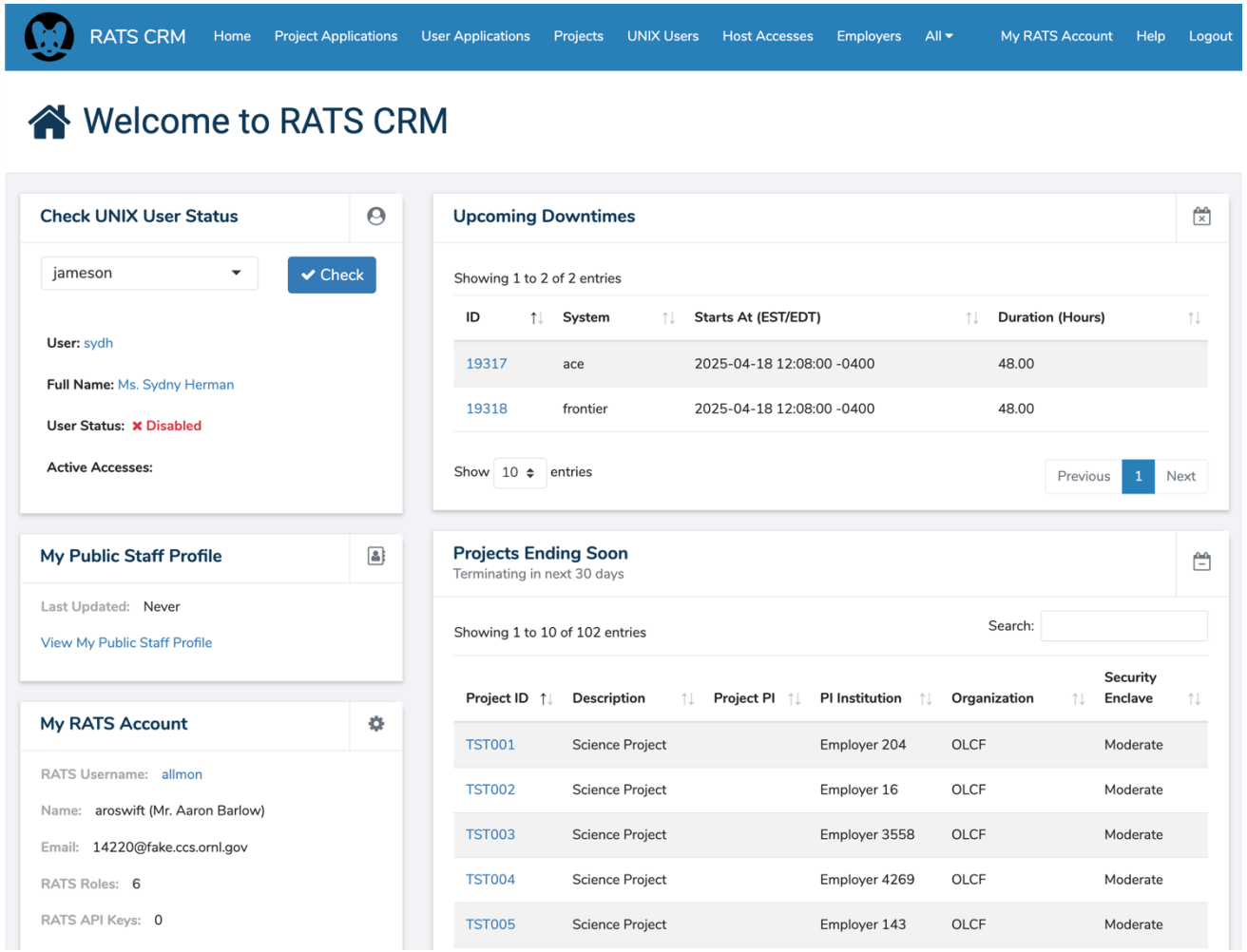
MyOLCF

1. **Built With**
   - JavaScript
   - Vue framework
2. **DevOps & Deployment**
   - GitLab Runners (CI/CD)
   - Kustomize for declarative infrastructure
3. **History**
   - Created in 2020

RATS CRM: Centralizing HPC Operations Management

1. **Core HPC Management**
   - Job scheduler configurations
   - Filesystem directories
   - Node and core allocations
   - Identity & security access
   - User and system access
   - Enforces & automates governance policies

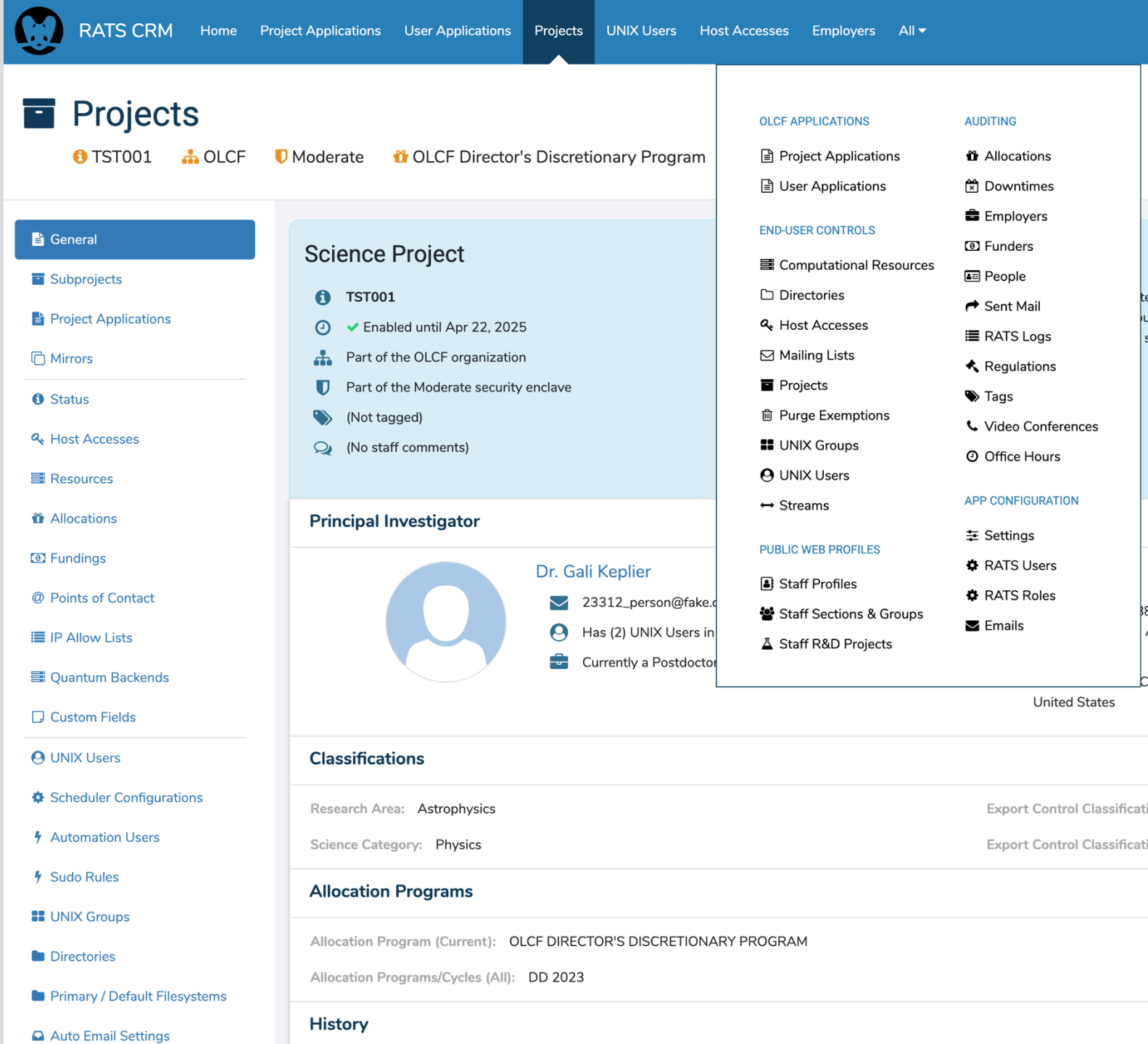
2. **Data Dissemination**
   - 224 API endpoints used by HPC systems and services



RATS CRM: first look with anonymized data

RATS CRM: Project-Centric Approach

1. Simplifies management
2. Definitive configuration source
3. Enables life-cycle automation



RATS CRM: From Proposal to Access

1. **Proposal Submission**
   - Researcher submits proposal via MyOLCF
2. **Review & Approval**
   - Admin reviews and applies policy
3. **Automated Setup**
   1. **Project-level setup**
      - Scheduler configuration & filesystem creation
      - Sub-projects enable delegation and subdivision
   2. **User-level setup**
      - UNIX user & group creation, filesystem creation, identity access sync, myOLCF access
      - Cascaded resource & filesystem access

NFS Home Areas

ID	Path	Policy/Template	Owner/UID	Group/GID	User	Project	Permissions	Quota	State
65665	/fake/directory/path	OLCF Moderate Project Home (nccs_mod_filer_proj) project home area	root / 0	gen110 / 27121	--	GEN110	0770	50	RECONCILED

GPFS Filesystems

ID	Path	Policy/Template	Owner/UID	Group/GID	User	Project	Permissions	Quota	State
54190	/gpfs/alpine2/gen110	Alpine2	root / 0	gen110 / 27121	--	GEN110	2755		PENDING
54191	/gpfs/alpine2/gen110/scratch	Alpine2	root / 0	gen110 / 27121	--	GEN110	2750		PENDING
54193	/gpfs/alpine2/gen110/proj-shared symlink: /gpfs/alpine2/proj-shared/gen110	Alpine2	root / 0	gen110 / 27121	--	GEN110	2770		PENDING
54194	/gpfs/alpine2/gen110/world-shared symlink: /gpfs/alpine2/world-shared/gen110	Alpine2	root / 0	gen110 / 27121	--	GEN110	2775		PENDING

RATS CRM: directories with anonymized data

RATS CRM: A Closer Look at Scheduler Configurations

1. **Centralized policy control for each HPC system:**
   - Queues / partitions
   - Job size limits & default wall times
   - QOS policies
2. **Defined once, applied everywhere:**
   - System-wide defaults
   - Project and user specific overrides
   - Supports cascading rules by project, group, or user
3. **Simplifies multi-scheduler environments:**
   - Abstracts SLURM, LSF, KUBE, MOAB config differences

Defaults

ID	Context	Granularity	Target	Perpetual?	Precedence
1915	Resource: Summit	Per User	User: All Users	Yes	50 / 100
310	Resource: Summit	Per Project	Project: All Projects	Yes	0 / 100
311	Resource: Summit	Per User	User: All Users	Yes	0 / 100

Overallocation Auto-Penalties

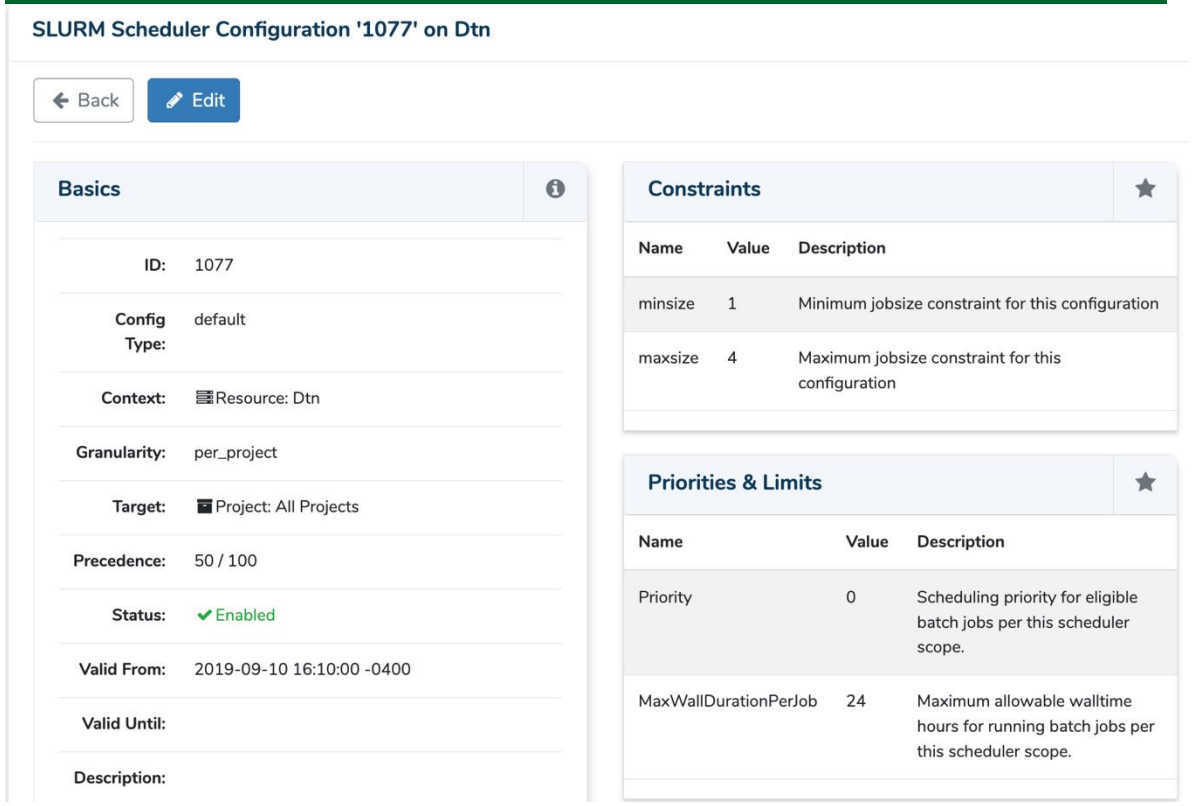
Showing 1 to 10 of 303 entries

ID	Context	Granularity	Target	Perpetual?	Precedence
2048	Resource: Summit	Per Project	Project: TST001	Yes	75 / 100
5632	Resource: Summit	Per Project	Project: TST002	Yes	75 / 100
2052	Resource: Summit	Per Project	Project: TST003RSCH	Yes	75 / 100
5380	Resource: Summit	Per Project	Project: TST004	Yes	75 / 100

RATS CRM: HPC scheduler configurations with anonymized data

RATS CRM: A Closer Look at Scheduler Configurations Cont.

RATS CRM: Individual Scheduler Configuration



Equivalent Scheduler Configuration via API


```
"config_type": "default",
"resource": "dtn",
"source_id": 1077,
"description": "",
"precedence": 50,
"primary_scope": {
  "id": 52,
  "type": "Resource",
  "name": "Dtn"
},
"secondary_scope": {
  "id": 9,
  "type": "SchedulerCredential",
  "name": "All Projects"
},
"granularity": "per_project",
"constraints": [
  {
    "type": "minsize",
```


Example RATS CRM SLURM scheduler configuration with anonymized data

Managing HPC Systems via API: Scheduler Configurations

Retrieving Scheduler Configuration Data

Request

```
GET /api/scheduler_configurations
```

Response

```
[{  ...,
    "attributes": {
        "config_type": "default",
        "resource": "frontier",
        "precedence": 50,
        "primary_scope": { ... },
        ...
    }, ...
}]
```

HPC Systems Enforcement

```
# Assume API response parsing above
for entry in "${config_lines[@]}"; do
   read -r part min_n max_n wall_h <<< "$entry”
   for host in "${SLURM_HOSTS[@]}"; do
      ssh admin@"$host" "sacctmgr --immediate update
          partition name=$part \
          set MinNodes=$min_n MaxNodes=$max_n \
          MaxWall=${wall_h}:00:00 \
          || sacctmgr --immediate add partition name=$part \
          default=NO target=ALL MinNodes=$min_n \
          MaxNodes=$max_n MaxWall=${wall_h}:00:00”
          ...
```

Simplified example API and HPC system bash script based on RATS CRM & Integration

Managing HPC Systems via API: UNIX Groups

Retrieving UNIX Group Data

Request

```
GET /api/groups?filter[‘system]=frontier
```

Response

```
[{
    "data": {
        "type": "groups",
        "id": 1,
        "attributes": {
            "name": "testgroup",
            "gid": 10000,
            "organization": "olcf",
            "security_enclave": "moderate”
        }, ...
}]
```

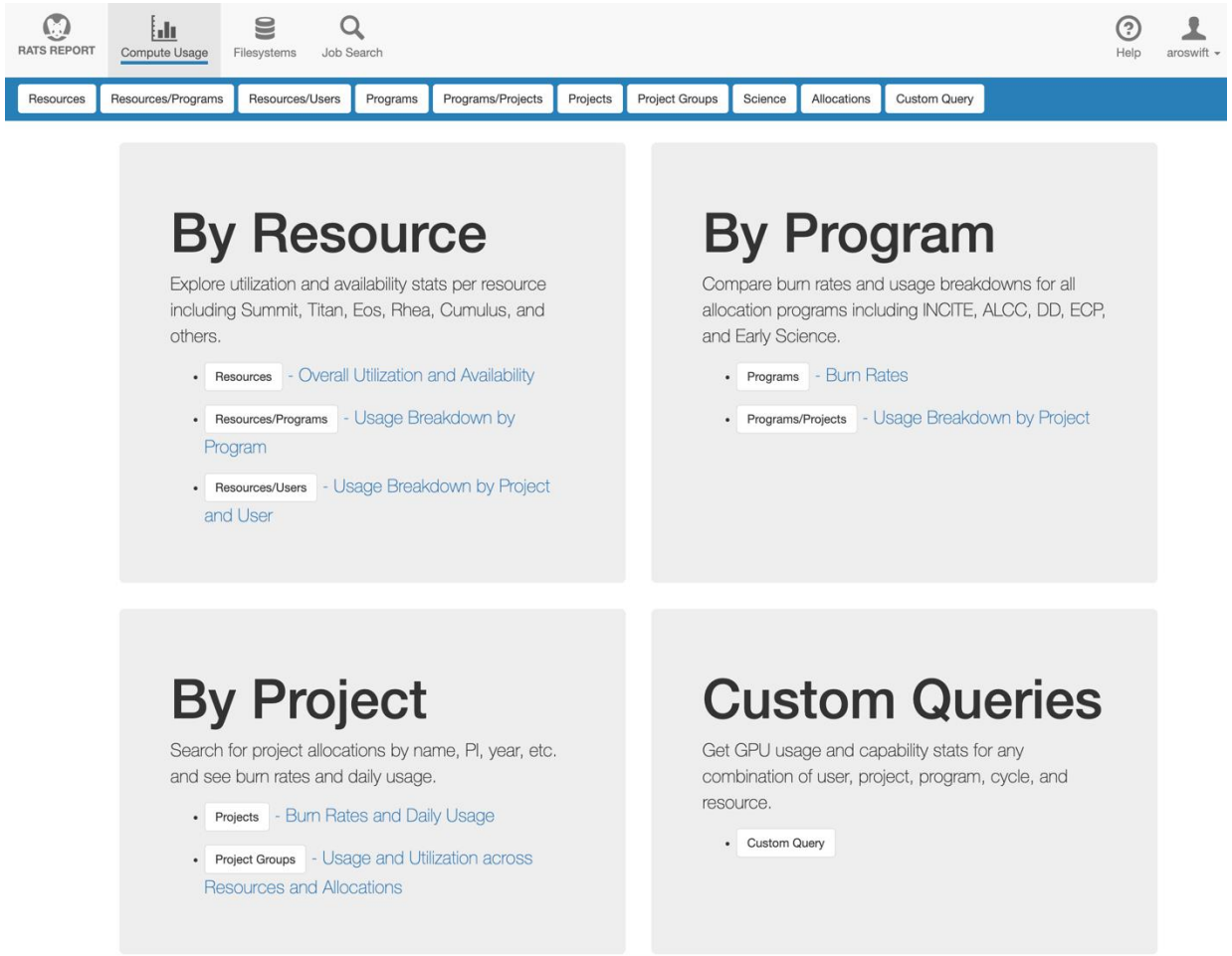
HPC Systems Enforcement

```
#!/bin/bash
# Triggered via cronjob every 30 minutes:
# */30 * * * * /opt/scripts/group_enforcement.sh

groups_json=$(curl -s -H "Authorization: Bearer
 $API_TOKEN" "$BASE_URL")

echo "$groups_json" | jq -r '.[] | .data.attributes | "\(.name)
\(.gid)”'

while read -r group gid; do
    for system in "${HPE_SYSTEMS[@]}"; do
        ssh admin@"$system" "sudo groupadd -g $gid $group
    done
done
```


Another example API and HPC system bash script based on RATS CRM & Integration

RATS Report: HPC Analytics & Reporting

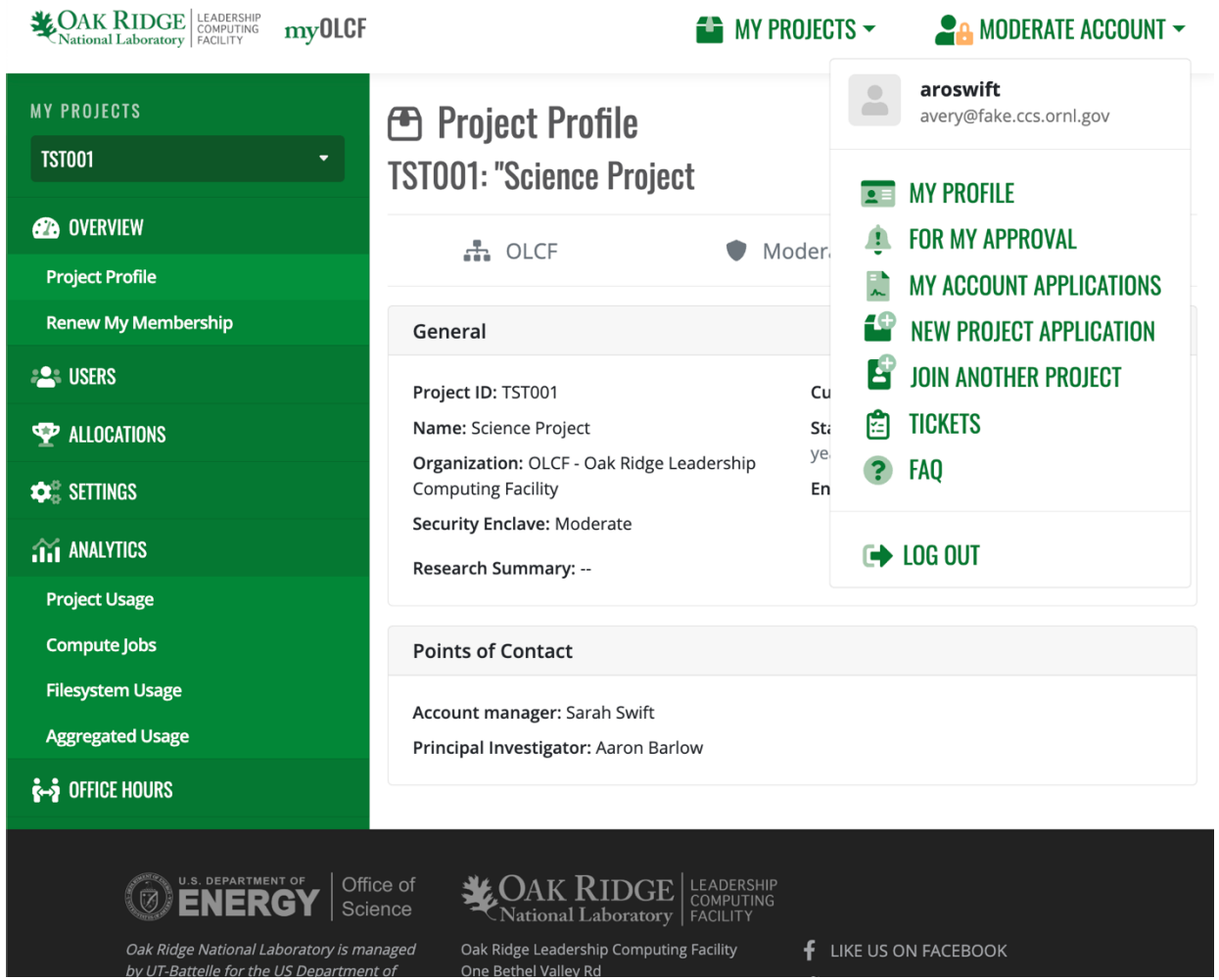
- **Key capabilities:**
  1. Extracts, transforms, and loads (ETL) HPC data
  2. Monitoring of compute resource consumption
  3. Allocation burn-down analysis
  4. Custom queries for job and storage analytics
  5. Aid in auditing
- 4 API endpoints with filtering to disseminate information



RATS Report homepage

MyOLCF: Public Self-Service Application

- Publicly available application that extends management capabilities and provides insights into available resources for users
  - my.olcf.ornl.gov
- Some key features:
  - Apply for and renew projects and accounts on HPC systems
  - Check the status of their applications
  - Monitor their submitted issues
  - Allocation usage monitoring
  - Compute usage report generation
  - Schedule office hours for help



MyOLCF GUI with anonymized data

OLCF Acknowledgement

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.


Questions?