HALCYON : Unified HPC Center Operations

PURDUE UNIVERSITY
Information Technology
RESEARCH COMPUTING

Under a shared campus cluster model, with many research groups investing distinct amounts and annual new hardware acquisitions, management of users and resources can become very complex. At Purdue, the Research Computing division set out in 2011 to design a cluster management solution to empower faculty to manage access to their own purchased resources.

As operations at Purdue expanded, the internal portal evolved and took on many aspects of the operation of an HPC center beyond resource allocation and management. Eventually, components included HPC and storage resource management, user management & authorization, customer relationship, communications, documentation, and purchasing.

Halcyon reconstitutes these in a modular, extensible framework to allow for the growth and maintenance necessary to encompass all aspects of HPC center management, from resource allocation to the less talked about but equally important aspects of customer relations. This will allow centers to operate not only more efficiently, but more effectively, and deliver better services to researchers.

Solution

The original Purdue Research Computing portal served its purpose well over the years but difficulty in making changes and the expanding needs of managing an HPC center drove the decision to re-architect the underlying code to:

- Abstract all environment assumptions and settings
  - *Language and site settings pulled into internal database and interfaces for editing*
  - *Environment settings incorporated and consolidated into configuration files*
- Restructure for modularity and easier extension
  - *Laravel as an underlying framework due to its popularity, large community, active development, and inclusion of desired features*
  - *Laravel app design helps secure codebase*

The portal code was separated into modules—logical groupings based on the data being handled, tasks, and interfaces. Modules may communicate and share data via an event dispatcher. Third-party services are integrated as plug-ins through the same event dispatcher system, which allows modules to ingest data from sources such as LDAP or REST APIs, as shown in Fig. 1.

Modules implement a REST API, documented in the code according to the OpenAPI v3 specification, allowing automatically generated human-readable documentation for all available endpoints. This API is one of the essential interaction points for external services and resources. For administrative and scheduled tasks, modules may also implement console commands, providing powerful methods to inspect and interact with data. Some of these interfaces are shown in Fig. 2, Fig. 3, and Fig. 4.

A key aspect to the interaction of external resources such as compute clusters or storage is the use of an internal message queue, implemented through ZeroMQ. When users submit changes such as new directories, permissions changes to a directory, or checks for current resource usage, a message is queued for a remote worker on the appropriate resource. The worker processes this message as soon as possible and submits any response back to Halcyon asynchronously.

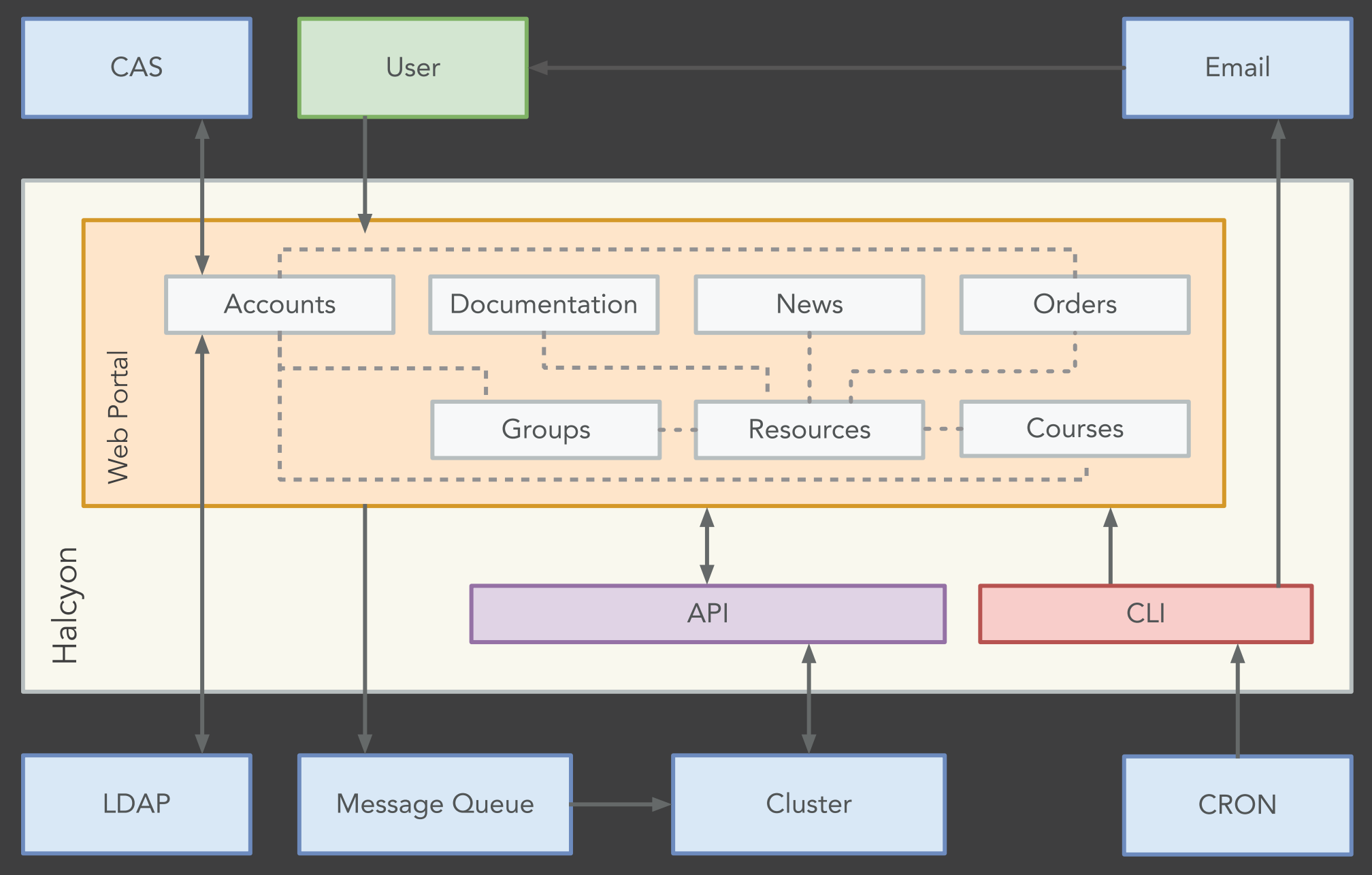
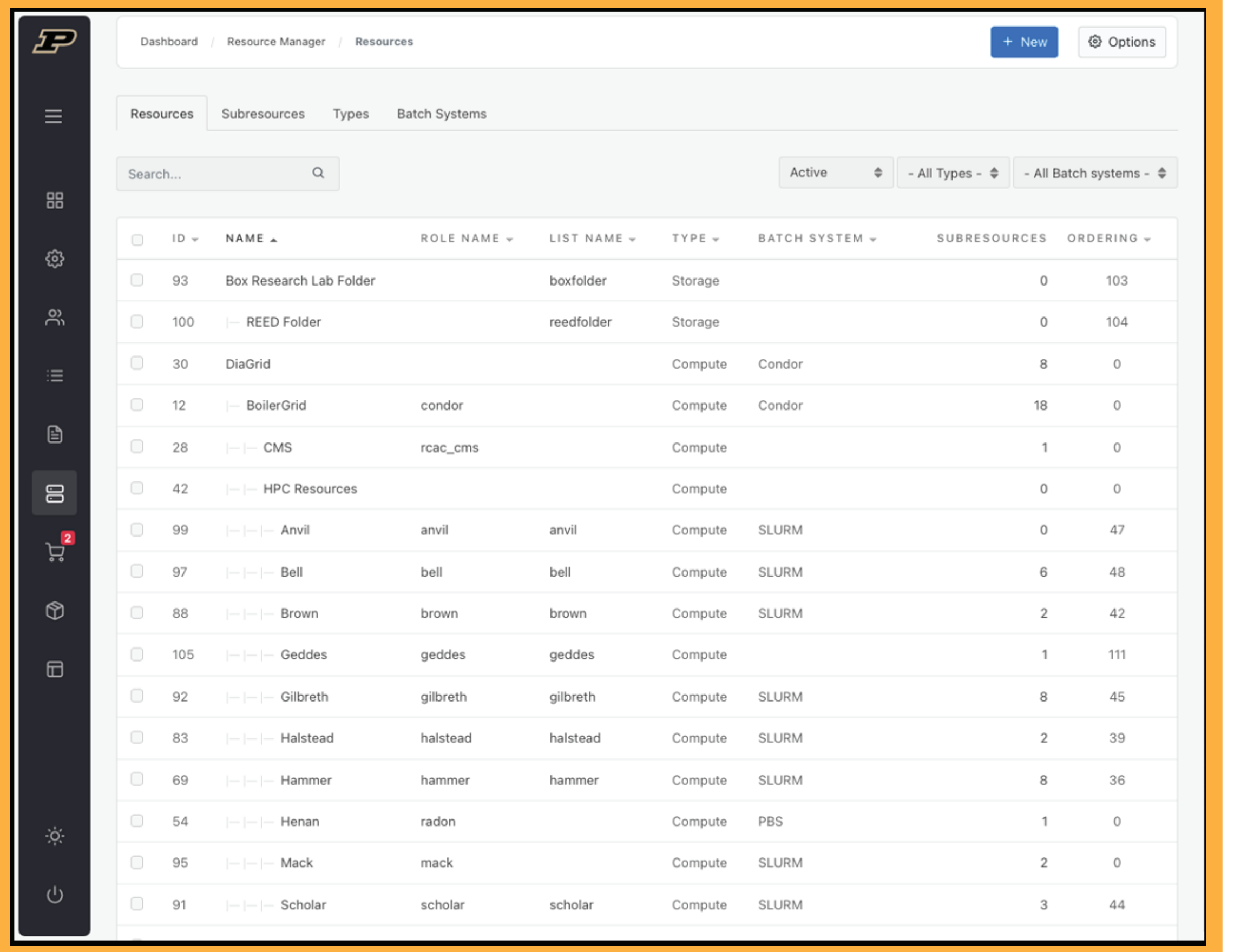


FIGURE 1

Results

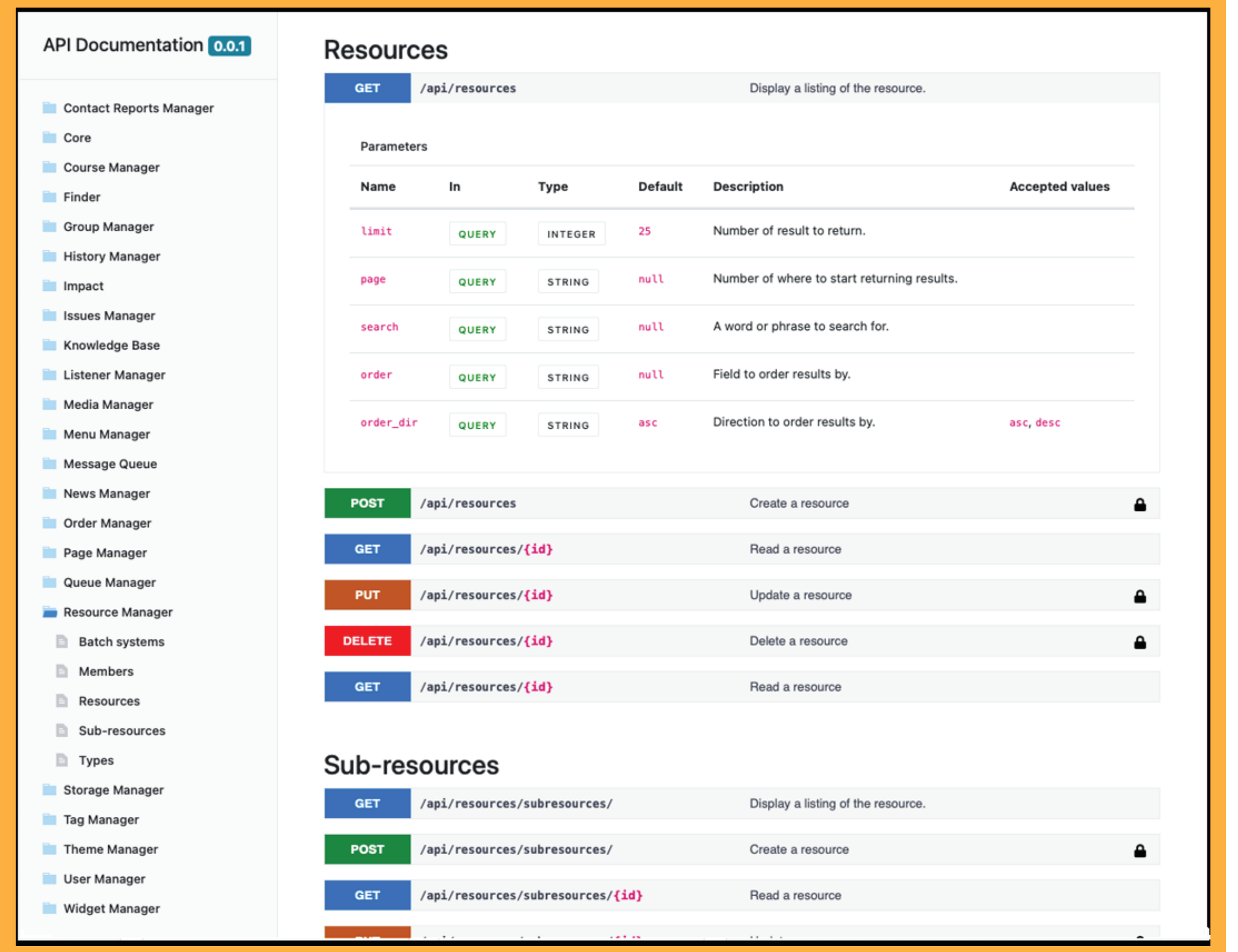
In its initial incarnation, the Purdue Research Computing portal provided faculty, students, and center staff an immediate reduction in time and effort to request, provision, and manage allocations. The user allocation/authorization alone was estimated as saving 6.7 hours of center staff time per week over the first 18 months. New storage spaces were able to reduce time-to-completion from 8 days to 1 hour within the first year of adoption of storage management. Pushing those tools beyond the resource to larger center operations has proven still more valuable, but all this originally came at a high maintenance cost.

Reconstituting the portal features into a modular, extensible framework was an essential next step to allow for the growth and maintenance necessary. These aspects of Halcyon have already allowed improvements to user interfaces, workflows, and integrations to proceed at an ever-increasing pace. To date, Halcyon is garnering praise from both center staff and researchers for time savings in additional self-service utilities and automation of repeated tasks.



ID	NAME	ROLE NAME	LIST NAME	TYPE	BATCH SYSTEM	SUBRESOURCES	ORDERING
93	Box Research Lab Folder		boxfolder	Storage		0	103
100	|— REED Folder		reedfolder	Storage		0	104
30	DiaGrid			Compute	Condor	8	0
12	|— BoilerGrid	condor		Compute	Condor	18	0
28	|—|— CMS	rcac_cms		Compute		1	0
42	|—|— HPC Resources			Compute		0	0
99	|—|—|— Anvil	anvil	anvil	Compute	SLURM	0	47
97	|—|—|— Bell	bell	bell	Compute	SLURM	6	48
88	|—|—|— Brown	brown	brown	Compute	SLURM	2	42
105	|—|—|— Geddes	geddes	geddes	Compute		1	111
92	|—|—|— Gilbreth	gilbreth	gilbreth	Compute	SLURM	8	45
83	|—|—|— Halstead	halstead	halstead	Compute	SLURM	2	39
69	|—|—|— Hammer	hammer	hammer	Compute	SLURM	8	36
54	|—|—|— Henan	radon		Compute	PBS	1	0
95	|—|—|— Mack	mack		Compute	SLURM	2	0
91	|—|—|— Scholar	scholar	scholar	Compute	SLURM	3	44

FIGURE 2



Resources

GET	/api/resources	Display a listing of the resource.

Parameters

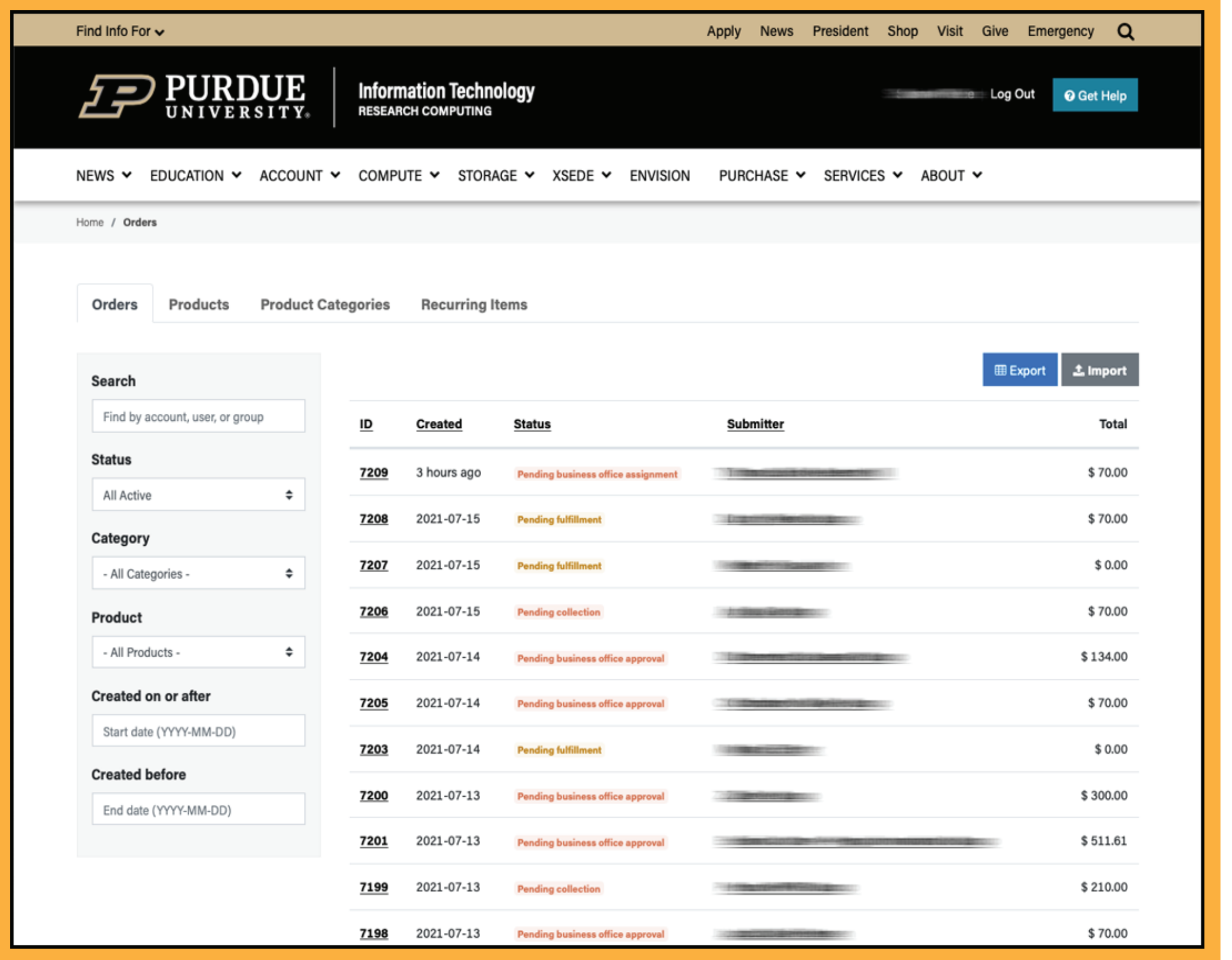
Name	In	Type	Default	Description	Accepted values
limit	QUERY	INTEGER	25	Number of result to return.	
page	QUERY	STRING	null	Number of where to start returning results.	
search	QUERY	STRING	null	A word or phrase to search for.	
order	QUERY	STRING	null	Field to order results by.	
order_dir	QUERY	STRING	asc	Direction to order results by.	asc, desc

POST	/api/resources	Create a resource
GET	/api/resources/{id}	Read a resource
PUT	/api/resources/{id}	Update a resource
DELETE	/api/resources/{id}	Delete a resource
GET	/api/resources/{id}	Read a resource

Sub-resources

GET	/api/resources/subresources/	Display a listing of the resource.
POST	/api/resources/subresources/	Create a resource
GET	/api/resources/subresources/{id}	Read a resource

FIGURE 3



Search: Find by account, user, or group
Status: All Active
Category: - All Categories -
Product: - All Products -
Created on or after: Start date (YYYY-MM-DD)
Created before: End date (YYYY-MM-DD)

ID	Created	Status	Submitter	Total
7209	3 hours ago	Pending business office assignment	[redacted]	$ 70.00
7208	2021-07-15	Pending fulfillment	[redacted]	$ 70.00
7207	2021-07-15	Pending fulfillment	[redacted]	$ 0.00
7206	2021-07-15	Pending collection	[redacted]	$ 70.00
7204	2021-07-14	Pending business office approval	[redacted]	$ 134.00
7205	2021-07-14	Pending business office approval	[redacted]	$ 70.00
7203	2021-07-14	Pending fulfillment	[redacted]	$ 0.00
7200	2021-07-13	Pending business office approval	[redacted]	$ 300.00
7201	2021-07-13	Pending business office approval	[redacted]	$ 511.61
7199	2021-07-13	Pending collection	[redacted]	$ 210.00
7198	2021-07-13	Pending business office approval	[redacted]	$ 70.00

FIGURE 4

Future Work

While there has been significant re-architecture in the creation of Halcyon to increase modularity and extensibility and to remove hard-coded values, there still remain a few values and interactions built into code which are specific to the environments and implementations of Research Computing at Purdue and which must be removed.

Implementing Halcyon at another organization is a current goal, which will further highlight where flexibility and configuration can be introduced to better accommodate different infrastructure and HPC center policies. This will provide an opportunity to develop more integrations and plugins to further extend Halcyon's capabilities.

References

[1] Kevin D. Colby, Daniel T. Dietz, Preston M. Smith, and Donna D. Cum- berland, "Self-service Queue and User Management in Shared Clusters," in *Proceedings of the 1st International Workshop on HPC User Support Tools, HUST14, Nov. 2014, New Orleans, LA, USA*, IEEEPress, 2014.

[2] Mohammed Zia, et al., "ColdFront: An Open Source HPC Resource Allocation System," Presented at the 5th International Workshop on HPC User Support Tools, HUST18, Nov. 2018, Dallas, TX, USA. 2018.

[3] Laravel LLC, "The PHP Framework for Web Artisans", laravel.com. [Online]. Available: https://laravel.com/. [Accessed: Jul. 16, 2021].

[4] iMatrix Corporation, "Distributed Messaging", *zeromq.org*. [Online]. Available: https://zeromq.org/get-started/. [Accessed: Jul. 16, 2021].

[5] Donna D. Cumberland, "Six Sigma: Research Account Queue Manage-ment Process Improvement", West Lafayette: Purdue University Press, 2011.

[6] Kevin D. Colby, Daniel T. Dietz, and Lev A. Gorenstein, "Shared Research Group Storage Solution with Integrated Access Management," in *Proceedings of PEARC17, 2017*, New York: ACM, 2017.

Kevin D. Colby
Research Computing
Purdue University

Shawn Rice
Research Computing
Purdue University