



# Improving FERRY: Refactoring, Visualization, and Testing

Antonio Segura – University of Detroit Mercy

Supervisor: Bruno Coimbra – Fermilab

SIST/GEM Final Presentation

7 August 2019

# Project Overview

To Improve the Frontier Experiments RegistRY (FERRY) meta-data management system

- Refactor Application Programming Interface (API) code to be standardized
- Implement a visual interface for user ease of access
- Integrate an automated testing system to test APIs

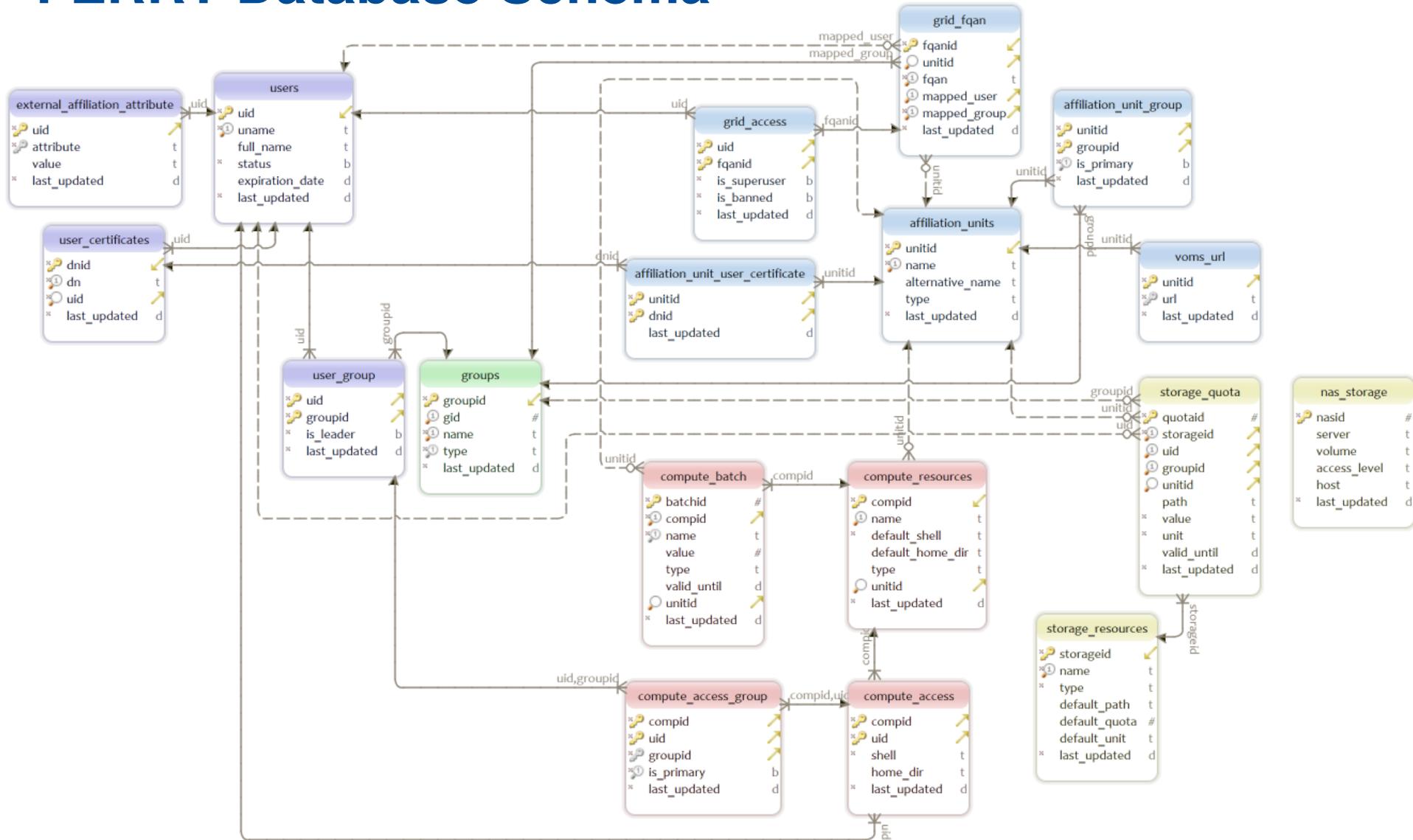


# FERRY

# Description of the Frontier Experiments RegistRY

- FERRY is a meta-database containing information about experiments, users, groups, and storage quotas
- Replaced the usage of GUMS, VOMS, and Vulcan
- Works in collaboration with the SNOW
- Primary used as an account registry system
- Accessible via API

# FERRY Database Schema

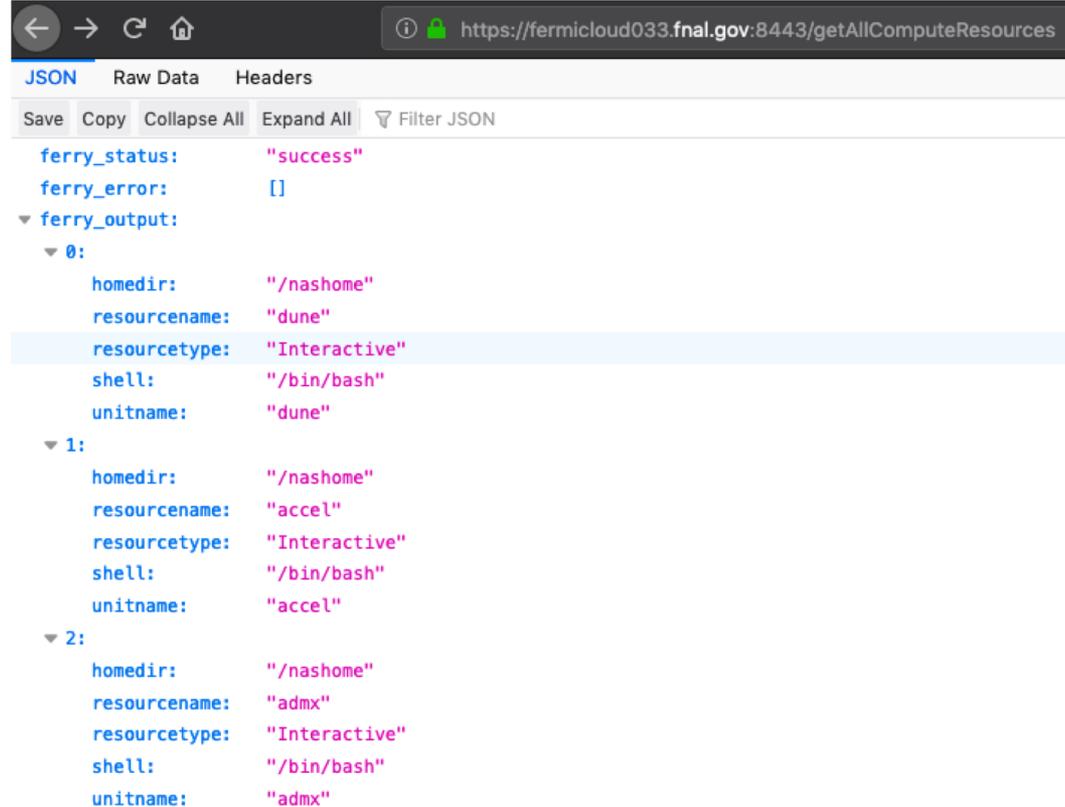


# FERRY as a RESTful API

- Representational State Transfer (REST) is an architectural style for Web based software services
- Services that conform to this architecture are called RESTful Web Services
- These services allow requesting systems to access and manipulate data that is transmitted
- FERRY is a RESTful service that returns data in the JSON format
- Several HyperText Transfer Protocol (HTTP) methods are included, such as GET, PUT, and DELETE

<https://fermicloud033.fnal.gov:8443/testBaseAPI?uid=9001&username=asegura>

# FERRY API Call



```
https://fermicloud033.fnal.gov:8443/getAllComputeResources

JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON

{
  "ferry_status": "success",
  "ferry_error": [],
  "ferry_output": [
    {
      "0": {
        "homedir": "/nashome",
        "resourcename": "dune",
        "resourcetype": "Interactive",
        "shell": "/bin/bash",
        "unitname": "dune"
      }
    },
    {
      "1": {
        "homedir": "/nashome",
        "resourcename": "accel",
        "resourcetype": "Interactive",
        "shell": "/bin/bash",
        "unitname": "accel"
      }
    },
    {
      "2": {
        "homedir": "/nashome",
        "resourcename": "adm",
        "resourcetype": "Interactive",
        "shell": "/bin/bash",
        "unitname": "adm"
      }
    }
  ]
}
```

The output is a JSON Object. In Python this can be imported and turned into a nested dictionary/list

# Refactoring FERRY

FERRY was developed by several people and the code is inconsistent in several ways:

- Attributes  $\neq$  Parameters
- Output is not consistent
- Style and strategies to create APIs not standard

The project requires rewriting of each API call to be standardized

# Refactoring FERRY (cont.)

```
setSuperUser := BaseAPI {
  InputModel {
    Parameter{UserName, true},
    Parameter{UnitName, true},
  },
  setSuperUser,
}

func setSuperUser() {
  ...
}
```

BaseAPI handles:

- Connecting to the database host
- Database transactions
- Error calls
- Input

```
// List of valid Attribute names
const (
  UserName      Attribute = "username"
  GroupName     Attribute = "groupname"
  UnitName      Attribute = "unitname"
  FullName      Attribute = "fullname"
  ResourceName  Attribute = "resourcename"
  AlternativeName Attribute = "alternativename"
  GroupType     Attribute = "grouptype"
  UnitType      Attribute = "unittype"
  ResourceType  Attribute = "resourcetype"
  DN            Attribute = "dn"
  UserAttribute Attribute = "attribute"
  Value         Attribute = "value"
  QuotaUnit     Attribute = "quotaunit"
  Path          Attribute = "path"
  Shell         Attribute = "shell"
  HomeDir       Attribute = "homedir"
  FQAN          Attribute = "fqan"
  VOMSURL       Attribute = "vomsurl"
  Role          Attribute = "role"
  CondorGroup   Attribute = "condorgroup"
  UID           Attribute = "uid"
  GID           Attribute = "gid"
  GroupID       Attribute = "groupid"
  DNID          Attribute = "dnid"
  FQANID        Attribute = "fqanid"
  UnitID        Attribute = "unitid"
  ResourceID    Attribute = "compid"
  Quota         Attribute = "quota"
  Status        Attribute = "status"
  Primary       Attribute = "primary"
  Leader        Attribute = "leader"
  GroupAccount  Attribute = "groupaccount"
  Experiment    Attribute = "experiment"
  Suspend       Attribute = "suspend"
  ExpirationDate Attribute = "expirationdate"
  LastUpdated   Attribute = "lastupdated"
  Help          Attribute = "help"
  PasswdMode    Attribute = "passwdmode"
)
```

# Refactoring FERRY (cont.)

## Original API – 114 Lines

```
141 func addCertificateDNToUserLegacy(w http.ResponseWriter, r *http.Request) {  
255 }
```

## Refactored API – 59 Lines

```
1373 func addCertificateDNToUser(c APIContext, i Input) (interface{}, []APIError) {  
1432 }
```

- Shortened Code
- Better documented and organized
- Much easier to read and understand

# Developing a Visual Interface for FERRY – Superset

- APIs are somewhat hard to decipher from a user standpoint
- User Interface (UI) would increase user ability to understand and use FERRY
- Initially wanted to create the UI from scratch
- Found open-source software from the Apache Software Foundation:



# Superset Dashboards

CMSUserInfo ☆

Edit dashboard

**CMSUserFilter**

Time range  
No filter

Unix Name  
Select [Unix Name]

Full Name  
Select [Full Name]

CERN User Name  
Select [CERN User Name]

**CMSUserInfoList**  
Show 50 entries

uname	full_name	cern_uname	home_dir	shell
3wolf3	Evan Wolfe	swolfe	/uscms/home/3wolf3	/bin/tcsh
aagarwal	Ayush Agarwal	ayagarwa	/uscms/home/aagarwal	/bin/bash
aakpinar	Alp Akpinar	aakpinar	/uscms/home/aakpinar	/bin/tcsh
aapyan2	Aram Apyan	arapyan	/uscms/home/aapyan2	/bin/bash
aarond	Aaron Dominguez	NA	/uscms/home/aarond	/bin/bash
aarulana	Adhav Arulanandan	NA	/uscms/home/aarulana	/bin/tcsh
aattikis	Alexandros Attikis	attikis	/uscms/home/aattikis	/bin/tcsh

Previous 1 2 3 4 5 ... 23 Next

**CMSUserGroupsList**  
Show 50 entries

uname	full_name	name	is_leader
butler	Joel Butler	us_cms	false
butler	Joel Butler	inujj	false
butler	Joel Butler	lpcllj	false
butler	Joel Butler	lpcsa14	false
stoyan	Stoyan Stoynev	us_cms	false
stoyan	Stoyan Stoynev	lpczgz	false
stoyan	Stoyan Stoynev	lpcmuon	false
cheung	Harry Cheung	us_cms	false
cheung	Harry Cheung	lpcllj	false
banerjee	Sudeshna Banerjee	us_cms	false
lammel	Stephan Lammel	us_cms	false
lammel	Stephan Lammel	lpcvms	false

Previous 1 2 3 4 5 ... 13 Next

**CMSUserStorageQuotaList**  
Show 50 entries

uname	full_name	name	path	TB	valid_until
3wolf3	Evan Wolfe	EOS	/eos/uscms/store/user/3wolf3	2	
3wolf3	Evan Wolfe	NOBACKUP3	/uscms_data/d3/3wolf3	0	
aagarwal	Ayush Agarwal	EOS	/eos/uscms/store/user/aagarwal	2	
aagarwal	Ayush Agarwal	NOBACKUP3	/uscms_data/d3/aagarwal	0	
aakpinar	Alp Akpinar	NOBACKUP3	/uscms_data/d3/aakpinar	0.1	
aakpinar	Alp Akpinar	NOBACKUP2	/uscms_data/d2/aakpinar	0.1	
aakpinar	Alp Akpinar	EOS	/eos/uscms/store/user/aakpinar	0.9	
aapyan2	Aram Apyan	EOS	/eos/uscms/store/user/aapyan2	2	
aapyan2	Aram Apyan	NOBACKUP3	/uscms_data/d3/aapyan2	0.1	
aattikis	Alexandros Attikis	NOBACKUP3	/uscms_data/d3/aattikis	0.1	
aattikis	Alexandros Attikis	EOS	/eos/uscms/store/user/aattikis	10.9	
abakshi	Amandeep Singh Bakshi	NOBACKUP3	/uscms_data/d3/abakshi	0.1	
abakshi	Amandeep Singh Bakshi	EOS	/eos/uscms/store/user/abakshi	2	
abkh1004	Akhshak Pasa	EOS	/eos/uscms/store/user/abkh1004	2	

Previous 1 2 3 4 5 ... 15 Next

**CMSUserDNList**  
Show 50 entries

uname	full_name	dn
3wolf3	Evan Wolfe	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=ewolfe/CN=739620/CN=Evan Michael Wolfe
3wolf3	Evan Wolfe	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Evan Wolfe/CN=UID:3wolf3
aagarwal	Ayush Agarwal	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Ayush Agarwal/CN=UID:aagarwal
aagarwal	Ayush Agarwal	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=ayagarwa/CN=828978/CN=Ayush Bhushan Agarwal
aakpinar	Alp Akpinar	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Alp Akpinar/CN=UID:aakpinar
aakpinar	Alp Akpinar	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=aakpinar/CN=834208/CN=Alp Akpinar
aapyan2	Aram Apyan	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=arapyan/CN=455306/CN=Aram Apyan

Previous 1 2 3 4 5 ... 20 Next

# Superset Querying

The screenshot displays the Superset SQL Lab interface. At the top, the navigation bar includes the FERRY logo, a 'Security' dropdown, 'Sources', 'Manage', 'Charts', 'Dashboards', and 'SQL Lab' (selected). A '+ New' button and user profile icons are on the right. The main area shows an 'Untitled Query' with a dropdown arrow. On the left sidebar, there are controls for 'Database: postgresql ferry', 'Select a schema (2)', and 'See table schema (0 in )' with a 'Select table' dropdown. The query editor contains the SQL: `1 select full_name, dn from users join user_certificates using(uid)`. Below the editor are buttons for 'Run Query', 'Save Query', 'Share Query', a 'LIMIT 100000' input, and a 'parameters' section with a green timer showing '00:00:02.41'. The 'Results' tab is active, showing a table with columns 'full\_name' and 'dn'. The table contains 10 rows of user data. At the bottom of the results section are buttons for 'Explore', '.CSV', and 'Clipboard', and a 'Filter Results' input field.

full_name	dn
Andreas Kronfeld	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Andreas Kronfeld/CN=UID:ask
Andrew Baden	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Andrew Baden/CN=UID:drew
Eileen Berman	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Eileen Berman/CN=UID:berman
Brenna Flaughner	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Brenna Flaughner/CN=UID:brenna
Elizabeth Buckley-Geer	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Elizabeth Buckley-Geer/CN=UID:buckley
Joel Butler	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=jbutler/CN=527378/CN=Joel Butler
Joel Butler	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Joel Butler/CN=UID:butler
Frank Chiebana	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Frank Chiebana/CN=UID:chiebana
Stoyan Stoynev	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=stoynev/CN=597450/CN=Stoyan Emilov Stoynev
Stoyan Stoynev	/DC=org/DC=cilogon/C=US/O=Fermi National Accelerator Laboratory/OU=People/CN=Stoyan Stoynev/CN=UID:stoyan

# Developing a Visual Interface for FERRY – Grafana

- The obvious alternative was a system already in use: Grafana
- Since it is in use throughout Fermilab already, Grafana has a secure authentication system set up
- No heavy memory usage
- Retains all of the good features from Superset



# Grafana Dashboards

Group Name

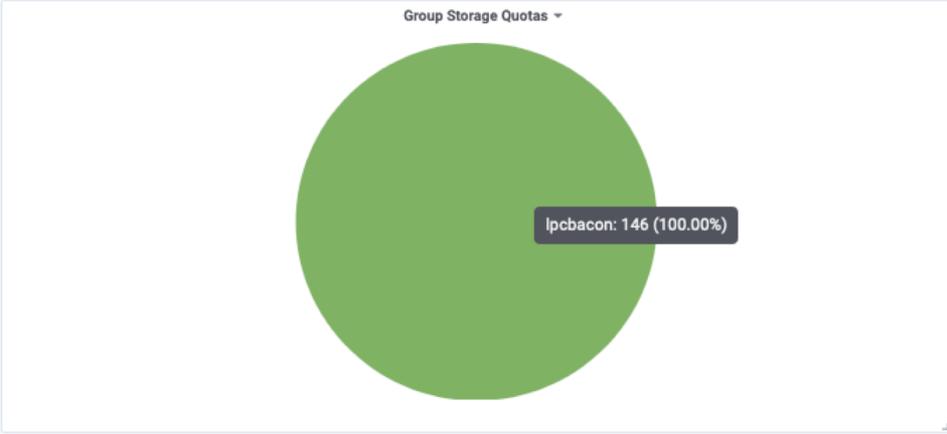
- lpcb
- lpcbbacon
- lpcbphy
- lpcbprime
- lpcbbril
- lpcbtag
- lpcbtagging

CMS Groups		
GID	Group Name	Number of Users
1311	lpcbprime	21

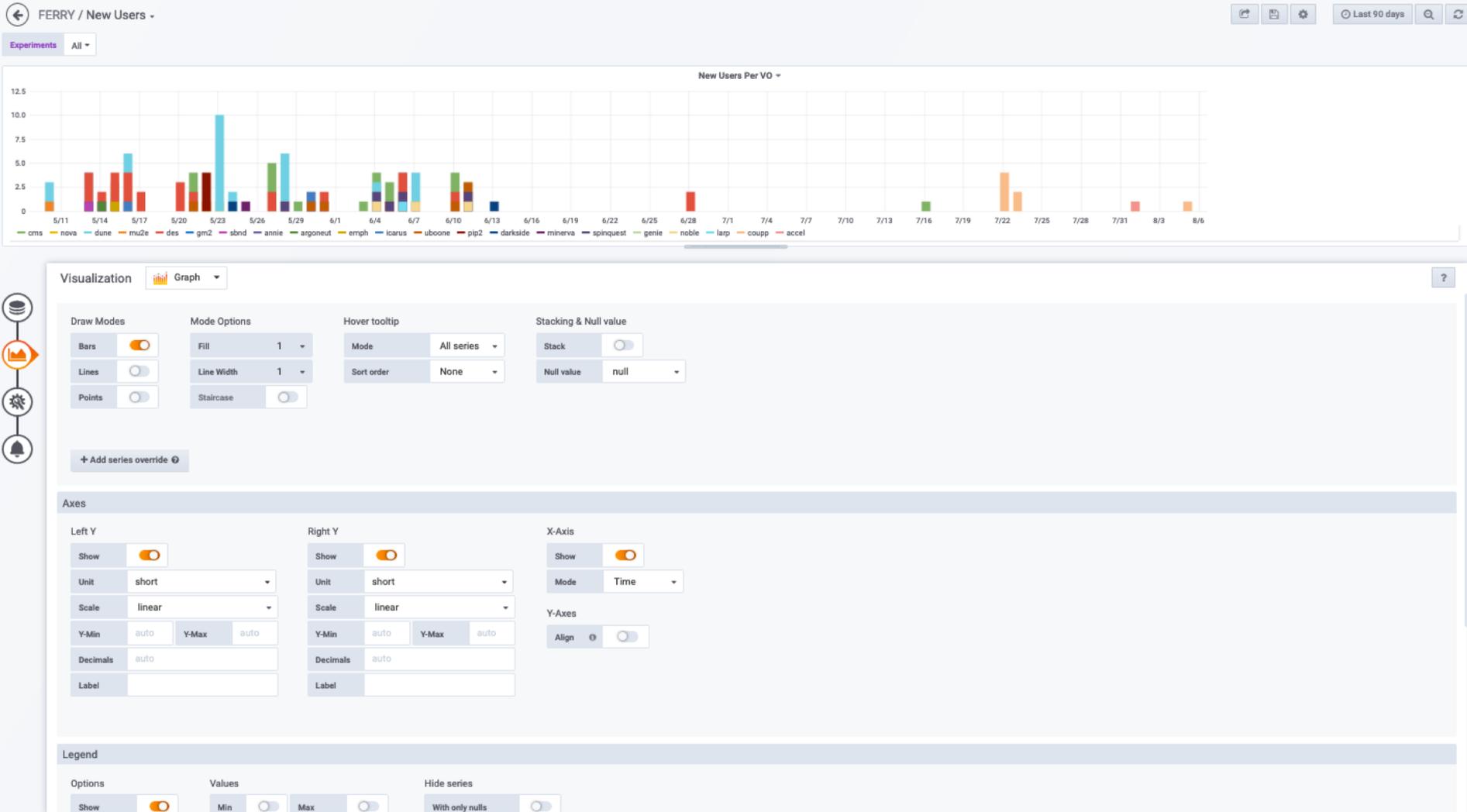
CMS Groups Users			
User ID	Username	Full Name	Group Name
52430	aapyan2	Aram Apyan	lpcbbacon
52698	lpcbbacon	Cms groups	lpcbbacon
49845	cmantill	Cristina Mantilla	lpcbbacon
47452	drankin	Dylan Rankin	lpcbbacon
50795	jbueghly	James Bueghly	lpcbbacon
54733	jkrupa	Jeffrey Krupa	lpcbbacon
53787	jrainbol	Jessica Rainbolt	lpcbbacon
48827	kmondal	Kuntal Mondal	lpcbbacon

Group Leaders		
User ID	Username	Full Name
52536	tonjes	Marguerite Tonjes
42518	jduarte1	Javier Duarte

Group Storage Quotas	
Name	Terabytes
lpcbbacon	145.5

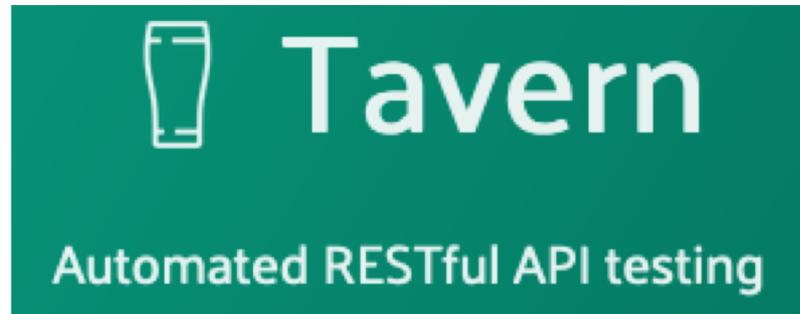


# Grafana Editor



# FERRY Automated Testing

- Refactored APIs and future additions require extensive unit testing to ensure their stability
- Manually testing would take unreasonably long, with 99 API calls, and dozens of parameter combinations, both correct and incorrect
- Tavern is a testing plugin for Pytest in Python that is used to test RESTful APIs
- Runs automated tests, but cannot create tests for each API by itself.



# Tavern Test Structure for FERRY APIs

Call test creation script with desired API to test

Make API call to get parameters

Find all possible parameter combinations

## Execute Tests

```
stages:  
- name: getAllUsers  
  request:  
    cert:  
    - /Users/asegura/certs/hostcert.pem  
    - /Users/asegura/certs/hostkey.pem  
    method: GET  
    url: https://fermicloud033.fnal.gov:8443/getAllUsers  
    verify: false  
  response:  
    body:  
      ferry_error: []  
      ferry_output:  
      - expirationdate: '0001-01-01T00:00:00Z'  
        fullname: System Administrators  
        status: false  
        uid: 44673  
        username: 131.225.107.4  
      - expirationdate: '0001-01-01T00:00:00Z'  
        fullname: System Administrators  
        status: false  
        uid: 44671  
        username: 131.225.107.82  
      - expirationdate: '0001-01-01T00:00:00Z'  
        fullname: System Administrators  
        status: false  
        uid: 44672  
        username: 131.225.107.83  
      - expirationdate: '0001-01-01T00:00:00Z'
```

Create Tavern files containing a test per combination

## Logged Output

```
[INFO] Testing setStorageQuota?expirationdate=2027-07-11&groupaccount=Tr  
[CRITICAL] 512/512 tests were successfully performed for setStorageQuota
```

# Conclusions

Overall we have made FERRY easier to use by:

- Making it easier to develop and read the code
- Creating a visual interface for user ease
- Building an automated testing system that will work for all and future APIs

Future work:

- Create more APIs for user requirements
- Create more dashboards based on user requirements
- Tweak Tavern automation to be more versatile

# Acknowledgements

Special thanks to:

My supervisors Bruno Coimbra and Tanya Levshina; mentors Carrie McGivern, Charlie Orozco, and Mehreen Sultana; Judy Nunez, Laura Fields, Sandra Charles, and the rest of SIST Committee for giving me this opportunity.

## References:

- Bruno Coimbra. *Ferry, What is Next*. <https://cdcvs.fnal.gov/redmine/attachments/download/52614/Ferry20What%20Is%20Next.pdf>
- Antonio Segura, Bruno Coimbra. *Grafana Dashboards*. <https://tinyurl.com/VisualFerry>
- Apache Software Foundation. *Apache Superset*. <https://superset.incubator.apache.org/>
- *Tavern: Automated RESTful API Testing*. <https://taverntesting.github.io/>
- Grafana Labs. *Grafana*. <https://grafana.com/>
- Wikipedia. *Representational State Transfer*. [https://en.wikipedia.org/wiki/Representational\\_state\\_transfer](https://en.wikipedia.org/wiki/Representational_state_transfer)

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.