**US Army Corps
of Engineers®**
Engineer Research and
Development Center

Engineered Resilient Systems

Integrated Rule-Oriented Data System (iRODS) and High Performance Computing (HPC) Requirements Document

Kevin D. Winters, Mark A. Cowan, Glover E. George,
Megan E. Gonzalez, Brian Priest, Omar Morris,
and Jonathan Landrum

August 2020

The U.S. Army Engineer Research and Development Center (ERDC) solves the nation's toughest engineering and environmental challenges. ERDC develops innovative solutions in civil and military engineering, geospatial sciences, water resources, and environmental sciences for the Army, the Department of Defense, civilian agencies, and our nation's public good. Find out more at www.erdcl.usace.army.mil.

To search for other technical reports published by ERDC, visit the ERDC online library at <http://acwc.sdp.sirsi.net/client/default>.

Integrated Rule-Oriented Data System (iRODS) and High Performance Computing (HPC) Requirements Document

Kevin D. Winters, Mark A. Cowan, Glover E. George, and Megan E. Gonzalez

*U.S. Army Engineer Research and Development Center (ERDC)
Information Technology Laboratory (ITL)
3909 Halls Ferry Road
Vicksburg, MS 39180-6199*

Brian Priest, Omar Morris, and Jonathan Landrum

*Geocent Jackson Mississippi Office
1000 Highland Colony Parkway, Suite 5203
Ridgeland, MS 39157*

Final Report

Approved for public release; distribution is unlimited.

Prepared for Headquarters, U.S. Army Corps of Engineers
Washington, DC 20314-1000

Under Work Unit 92L5D8

Abstract

The purpose of this report is to capture all relevant use cases, functional requirements, and technical requirements of the Integrated Rule-Oriented Data System (iRODS) prototype. The use cases (UCs) define the system interactions an iRODS user, iRODS administrator, and an auditor would expect within the system. The functional requirements define the expected behavior of the system to support the individual use cases; functional requirements are grouped in reference to the use cases supported by the set of functional requirements. The technical requirements are defined last and include references to specific functional requirements and use cases supported by the requirement.

DISCLAIMER: The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

Contents

Abstract.....	ii
Figures	viii
Preface	ix
1 Introduction.....	1
1.1 Background.....	1
1.2 Objectives.....	1
1.3 Approach	1
2 Use Cases (UCs)	2
2.1 Use cases for utilizing the Command Line Interface (CLI).....	2
2.2 UC-1: Manage user-defined rules with the CLI.....	3
2.2.1 Scope	3
2.2.2 Description	3
2.2.3 Actors	3
2.2.4 Process steps.....	3
2.3 UC-2: Manage sharing with the CLI	4
2.3.1 Scope	4
2.3.2 Description	4
2.3.3 Actors	4
2.3.4 Process steps.....	4
2.4 UC-3: Search for data with the CLI.....	5
2.4.1 Scope	5
2.4.2 Description	5
2.4.3 Actors	5
2.4.4 Process steps.....	5
2.5 UC-4: Manage metadata with the CLI.....	6
2.5.1 Scope	6
2.5.2 Description	6
2.5.3 Actors	6
2.5.4 Process steps.....	7
2.6 UC-5: Manage data with the CLI	7
2.6.1 Scope	7
2.6.2 Description	8
2.6.3 Actors	8
2.6.4 Process steps.....	8
2.7 UC-6: Replicate data with the CLI	8
2.7.1 Scope	8
2.7.2 Description	9
2.7.3 Actors	9
2.7.4 Process steps.....	9
2.8 UC-7: Manage collections with the CLI	10

2.8.1 Scope	10
2.8.2 Description	10
2.8.3 Actors	10
2.8.4 Process Steps	10
2.9 UC-8: Manage resources with the CLI	11
2.9.1 Scope	11
2.9.2 Description	11
2.9.3 Actors	11
2.9.4 Process steps.....	12
2.10 UC-9: Manage policies with the CLI	13
2.10.1 Scope	13
2.10.2 Description	13
2.10.3 Actors.....	14
2.10.4 Process steps	14
2.11 UC-10: Manage users with the CLI	14
2.11.1 Scope	14
2.11.2 Description	15
2.11.3 Actors.....	15
2.11.4 Process steps	15
2.12 UC-11: Manage groups with the CLI	16
2.12.1 Scope	16
2.12.2 Description	16
2.12.3 Actors.....	16
2.12.4 Process steps	17
2.13 UC-12: Manage zones with the CLI.....	17
2.13.1 Scope	17
2.13.2 Description	17
2.13.3 Actors.....	18
2.13.4 Process steps	18
2.14 UC-13: Manage tickets with the CLI.....	18
2.14.1 Scope	18
2.14.2 Description	19
2.14.3 Actors.....	19
2.14.4 Process steps	19
2.15 UC-14: Manage usage quotas with the CLI	20
2.15.1 Scope	20
2.15.2 Description	20
2.15.3 Actors.....	20
2.15.4 Process steps	20
2.16 UC-15: Manage reports with the CLI.....	21
2.16.1 Scope	21
2.16.2 Description	21
2.16.3 Actors.....	21
2.16.4 Process steps	21
2.17 UC-16: Search audit records with the CLI.....	22
2.17.1 Scope	22

2.17.2 Description	22
2.17.3 Actors.....	22
2.17.4 Process steps	22
3 UCs for utilizing a GUI	23
3.1 UC-17: Manage sharing with a GUI	24
3.1.1 Scope	24
3.1.2 Description	24
3.1.3 Actors	24
3.1.4 Process steps.....	24
3.2 UC-18: Search for data with a GUI	25
3.2.1 Scope	25
3.2.2 Description	25
3.2.3 Actors	25
3.2.4 Process steps.....	26
3.3 UC-19: Manage metadata with a GUI	26
3.3.1 Scope	26
3.3.2 Description	26
3.3.3 Actors	27
3.3.4 Process steps.....	27
3.4 UC-20: Manage data with a GUI.....	28
3.4.1 Scope	28
3.4.2 Description	28
3.4.3 Actors	28
3.4.4 Process steps.....	28
3.5 UC-21: Replicate data with a GUI.....	29
3.5.1 Scope	29
3.5.2 Description	29
3.5.3 Actors	29
3.5.4 Process steps.....	29
3.6 UC-22: Manage collections with a GUI	30
3.6.1 Scope	30
3.6.2 Description	30
3.6.3 Actors	30
3.6.4 Process steps.....	30
3.7 UC-23: Manage resources with a GUI.....	31
3.7.1 Scope	31
3.7.2 Description	31
3.7.3 Actors	31
3.7.4 Process steps.....	31
3.8 UC-24: Manage users with a GUI	32
3.8.1 Scope	32
3.8.2 Description	32
3.8.3 Actors	33
3.8.4 Process Steps	33
3.9 UC-25: Manage groups with a GUI.....	34

3.9.1 Scope	34
3.9.2 Description	34
3.9.3 Actors	34
3.9.4 Process steps	34
3.10 UC-26: Manage reports with a GUI	35
3.10.1 Scope	35
3.10.2 Description	35
3.10.3 Actors	35
3.10.4 Process steps	35
3.11 UC-27: Search audit records with a GUI	36
3.11.1 Scope	36
3.11.2 Description	36
3.11.3 Actors	36
3.11.4 Process steps	36
3.12 UC-28: Audit requests	37
3.12.1 Scope	37
3.12.2 Actors	37
4 Functional Requirements	38
4.1 Requirements derived from CLI UCs	38
4.1.1 Derived from UC-1: Manage user-defined rules with the CLI	38
4.1.2 Derived from UC-2: Manage sharing with the CLI	38
4.1.3 Derived from UC-3: Search for data with the CLI	38
4.1.4 Derived from UC-4: Manage metadata with the CLI	39
4.1.5 Derived from UC-5: Manage data with the CLI	40
4.1.6 Derived from UC-6: Replicate data with the CLI	40
4.1.7 Derived from UC-7: Manage collections with the CLI	41
4.1.8 Derived from UC-8: Manage resources with the CLI	41
4.1.9 Derived from UC-9: Manage policies with the CLI	42
4.1.10 Derived from UC-10: Manage users with the CLI	42
4.1.11 Derived from UC-11: Manage groups with the CLI	42
4.1.12 Derived from UC-12: Manage zones with the CLI	43
4.1.13 Derived from UC-13: Manage tickets with the CLI	43
4.1.14 Derived from UC-14: Manage usage quotas with the CLI	43
4.1.15 Derived from UC-15: Manage reports with the CLI	43
4.1.16 Derived from UC-16: Search audit records with the CLI	44
4.2 Requirements derived from GUI UCs	44
4.2.1 Derived from UC-17: Manage sharing with a GUI	44
4.2.2 Derived from UC-18: Search for data with a GUI	44
4.2.3 Derived from UC-19: Manage metadata with a GUI	45
4.2.4 Derived from UC-20: Manage data with a GUI	45
4.2.5 Derived from UC-21: Replicate data with a GUI	46
4.2.6 Derived from UC-22: Manage collections with a GUI	46
4.2.7 Derived from UC-23: Manage resources with a GUI	46
4.2.8 Derived from UC-24: Manage users with a GUI	47
4.2.9 Derived from UC-25: Manage groups with a GUI	47

4.2.10	Derived from UC-26: Manage reports with a GUI	47
4.2.11	Derived from UC-27: Search audit records with a GUI	47
4.3	Security and auditing	47
4.3.1	Derived from UC-1: Manage user-defined rules with the CLI	47
4.3.2	Derived from UC-2 and UC-17: Managing sharing with the CLI; Managing sharing with a GUI	48
4.3.3	Derived from UC-3 and UC-18: Search for data with the CLI; Search for data with a GUI	48
4.3.4	Derived from UC-4 and UC-19: Manage metadata with the CLI; Manage metadata with a GUI	49
4.3.5	Derived from UC-5 and UC-20: Manage data with the CLI; Manage data with a GUI	50
4.3.6	Derived from UC-6 and UC-21: Replicate data with the CLI; Replicate data with a GUI	50
4.3.7	Derived from UC-7 and UC-22: Manage collections with the CLI; Manage collections with a GUI	50
4.3.8	Derived from UC-8 and UC-23: Manage resources with the CLI; Manage resources with a GUI	51
4.3.9	Derived from UC-9: Manage policies with the CLI; Manage policies with a GUI	51
4.3.10	Derived from UC-10 and UC-24: Manage users with the CLI; Manage users with a GUI	51
4.3.11	Derived from UC-11 and UC-25: Manage groups with the CLI; Manage groups with a GUI	51
4.3.12	Derived from UC-12: Manage zones with the CLI	51
4.3.13	Derived from UC-13: Manage tickets with the CLI	52
4.3.14	Derived from UC-14: Manage usage quotas with the CLI	52
4.3.15	General requirements	52
5	Technical Requirements	53
5.1	System requirements	53
5.2	Security and auditing	55
5.2.1	Derived from UC-1: Manage user-defined rules with the CLI	55
6	Conclusion	62
	Appendix A: Glossary of iRODS Terms	63
	Acronyms and Abbreviations	64
	Report Documentation Page	

Figures

Figure 1. UCs for utilizing the CLI.	2
Figure 2. UCs for utilizing a web GUI.....	23

Preface

This is a deliverable product under the Engineered Resilient Systems (ERS) Program, Data Analytics Work Package, Collaborative Tradespace Analytics Work Unit 92L5D8. Dr. Owen J. Eslinger was the program manager, and Dr. Robert M. Wallace was the Technical Director of the ERS program.

The work was performed by the Computational Analysis Branch (CAB) of the Computational Science and Engineering Division (CSED), Engineer Research and Development Center (ERDC), Information Technology Laboratory (ITL), Vicksburg, MS. At the time of publication, Dr. Jeffrey L. Hensley was Chief, CAB; Dr. Jerrell R. Ballard, Jr. was the Chief, CSED; Ms. Patti S. Duett was the Deputy Director and the Director was Dr. David A. Horner.

COL Teresa A. Schlosser was the Commander of ERDC, and Dr. David W. Pittman was the Director.

1 Introduction

1.1 Background

The purpose of this report is to capture all relevant use cases, functional requirements, and technical requirements of the Integrated Rule-Oriented Data System (iRODS) prototype. The use cases (UCs) define the system interactions an iRODS user, iRODS administrator, and an auditor would expect within the system. The functional requirements define the expected behavior of the system to support the individual use cases; functional requirements are grouped in reference to the use cases supported by the set of functional requirements. The technical requirements are defined last and include references to specific functional requirements and use cases supported by the requirement.

1.2 Objectives

This report will detail all relevant UCs, functional requirements, and technical requirements of the iRODS prototype. It also outlines the requirements needed to support building and testing the iRODS prototype in a test environment within the Defense Supercomputing Resource Center (DSRC). The specific system components outlined in this requirements document include iRODS, an iRODS web-based graphical user interface (GUI), and the auditing framework (i.e., Elastic Search, Logstash, and Kibana).

1.3 Approach

This document will first detail the use cases in depth, including the use cases specific to the GUI. The use cases will include the scope, description, actors, and process steps as pertaining to the presented use case. Second, this document will describe the functional requirements as outlined by the use cases. Lastly, this document will define the technical requirements for the prototype including references to specific functional requirements and use cases supported by the requirement.

2 Use Cases (UCs)

This chapter describes the identified use cases for interacting with an iRODS system. Because there is a need for auditing system access, the Elastic Stack has been included as an integral part of the system. Also presented are use cases for interacting with iRODS through a web GUI.

2.1 Use cases for utilizing the Command Line Interface (CLI)

The default method for interacting with an iRODS system is through the CLI. The iRODS CLI is a collection of Linux-derived commands to be executed via the command line.

Figure 1. UCs for utilizing the CLI.

2.2 UC-1: Manage user-defined rules with the CLI

2.2.1 Scope

This UC describes managing user-defined rules in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of these rules. For regular users, these rules can only be created to act on data owned by that user. For administrators, rules can be created to act on data system-wide. The tasks identified are logging on to iRODS, creating a new rule, displaying information about pending rules, updating an existing rule, and deleting an existing rule.

2.2.2 Description

The iRODS user creates rules to manage the workflow process of a group of researchers gathering sensor data. The user then modifies and updates the rules as new researchers are added to the group. The group numbers grow to many researchers and a decision is made to divide the group into three teams with different areas of focus. The iRODS user then deletes several of the old rules and creates new ones to reflect the changes in group structure. For regular users, these rules can only be created to act on data owned by that user. For administrators, rules can be created to act on data system-wide.

2.2.3 Actors

iRODS user

2.2.4 Process steps

1. Basic flow
 - a. Initiate iRODS connection
The user initiates a connection to the iRODS CLI using the *iinit* command.
 - b. Create a new rule
The user creates a new user-defined rule in the iRODS system using the *irule* command.
2. Alternative flows
 - a. Display information about pending rules
At basic flow step “Initiate iRODS connection,” the user displays information about pending user-defined rules in the iRODS system using the *iqstat* command.

- b. Update an existing rule
At basic flow step “Initiate iRODS connection,” the user updates values to an existing user-defined rule in the iRODS system using the *iqmod* command.
- c. Delete an existing rule
At basic flow step “Initiate iRODS connection,” the user deletes an existing user-defined rule from the iRODS system using the *iqdel* command.

2.3 UC-2: Manage sharing with the CLI

2.3.1 Scope

This UC describes managing the sharing of data with other users and groups in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of data access permissions. The process steps are largely the same for both data objects and collections, thus, the generic term *item* is used. The tasks identified are logging on to iRODS, granting read access on an item, granting write access, granting ownership access, and removing all access permissions on an item.

2.3.2 Description

The iRODS user grants different permissions to data. Certain data has been cleared for access to the team that needs it. Junior level engineers are granted read access to the files and senior level staff are granted write access. The user also takes away all access to information that has had its level upgraded to classified.

2.3.3 Actors

iRODS user

2.3.4 Process steps

1. Basic flow
 - a. Initiate iRODS connection
The user initiates a connection to the iRODS CLI using the *iinit* command.
 - b. Grant read access on an item
The user grants read access on an item in the iRODS system using the *ichmod* command with the *read* subcommand.

2. Alternative flows

a. Grant write access on an item

At basic flow step “Initiate iRODS connection,” the user grants write access on an item in the iRODS system using the *ichmod* command with the *write* subcommand.

b. Grant ownership access on an item

At basic flow step “Initiate iRODS connection,” the user grants ownership access on an item in the iRODS system using the *ichmod* command with the *own* subcommand.

c. Remove all access permissions on an item

At basic flow step “Initiate iRODS connection,” the user removes all access permissions on an item in the iRODS system using the *ichmod* command with the *null* subcommand.

2.4 UC-3: Search for data with the CLI

2.4.1 Scope

This UC describes searching for data in the iRODS system. The process steps are largely the same for both data objects and collections, thus, the generic term *item* is used. The tasks identified are logging on to iRODS, searching for an item with a known name, searching for an item using wildcards, searching for all items the user has access to, and searching for an item with a known metadata entry.

2.4.2 Description

The iRODS user needs to initiate a simulation model validation. This process requires several input files. The iRODS user knows the exact name of some of the files and knows partial names or metadata attributes of others. The user searches for the necessary files with a variety of methods: exact name, wildcard, and metadata attributes using iCommands.

2.4.3 Actors

iRODS user

2.4.4 Process steps

1. Basic flow

a. Initiate iRODS connection

The user initiates a connection to the iRODS CLI using the *iinit* command.

- b. Search for an item
The user searches for an item with a known name in the iRODS system using the *iquest* command.
- 2. Alternative flows
 - a. Search for an item with wildcards
At basic flow step “Initiate iRODS connection,” the user searches for an item with a partially known name in the iRODS system using wildcards with the *iquest* command.
 - b. Search for all items the user has access to
At basic flow step “Initiate iRODS connection,” the user searches for all items they have access to in the iRODS system using the *ils* command.
 - c. Search for an item with a known metadata entry
At basic flow step “Initiate iRODS connection,” the user searches for an item with a known metadata entry in the iRODS system using the *imeta* command.

2.5 UC-4: Manage metadata with the CLI

2.5.1 Scope

This UC describes managing metadata in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of metadata. The process steps are largely the same for data objects, collections, resources, and users; thus, the generic term *item* is used. The tasks identified are logging on to iRODS, adding metadata to an item, listing the metadata that exists for an item, modifying metadata, and deleting metadata.

2.5.2 Description

After a model validation is complete, the iRODS user needs to add metadata to newly generated outputs and modify existing metadata on the updated input files. The iRODS user logs into the system and adds metadata attributes to the output files. The user then places the new files in their correct collection, changes several existing metadata attributes on the input files, and then deletes certain attributes to reflect the modifications done to them.

2.5.3 Actors

iRODS user

2.5.4 Process steps

1. Basic flow
 - a. Initiate iRODS connection
The user initiates a connection to the iRODS CLI using the *iinit* command.
 - b. Add a metadata entry
The user adds a metadata entry to an item in the iRODS system using the *imeta* command with the *add* subcommand.
2. Alternative flows
 - a. List existing metadata entries
At basic flow step “Initiate iRODS connection,” the user lists existing metadata entries for an item in the iRODS system using the *imeta* command with the *ls* subcommand.
 - b. Modify an existing metadata entry
At basic flow step “Initiate iRODS connection,” the user modifies a metadata entry for an item in the iRODS system using the *imeta* command with the *mod* subcommand.
 - c. Create or update a metadata entry
At basic flow step “Initiate iRODS connection,” the user creates or updates a metadata entry for an item in the iRODS system using the *imeta* command with the *set* subcommand.
 - d. Remove a metadata entry
At basic flow step “Initiate iRODS connection,” the user removes metadata from an item in the iRODS system using the *imeta* command with the *rm* subcommand.

2.6 UC-5: Manage data with the CLI

2.6.1 Scope

This UC describes managing data in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of data. The steps in this process are largely the same for both data objects and collections, thus, the generic term *item* is used. The tasks identified are logging on to iRODS, listing the items that exist in the system, moving items into iRODS from the local workstation, renaming items, updating items by overwriting them, and deleting unwanted items.

2.6.2 Description

The iRODS user logs in to the system and moves several files from a model run from their local work directory into iRODS. After this is complete, the user lists their directory structure and moves to an existing collection to update some of the data objects it contains and move others to different collections.

2.6.3 Actors

iRODS user

2.6.4 Process steps

1. Basic flow
 - a. Initiate iRODS connection
The user initiates a connection to the iRODS CLI using the *iinit* command.
 - b. Upload an item
The user uploads an item from their local machine to the iRODS system using the *iput* command.
2. Alternative flows
 - a. List existing items
At basic flow step “Initiate iRODS connection,” the user lists existing items in the iRODS system using the *ils* command.
 - b. Rename an item
At basic flow step “Initiate iRODS connection,” the user renames an item in the iRODS system using the *imv* command.
 - c. Update an item
At basic flow step “Initiate iRODS connection,” the user overwrites an existing item in the iRODS system using the *iput* command.
 - d. Delete an item
At basic flow step “Initiate iRODS connection,” the user deletes an item from the iRODS system with the *irm* command.

2.7 UC-6: Replicate data with the CLI

2.7.1 Scope

This UC describes replicating data across resources in the iRODS system. The process steps are largely the same for both data objects and collections, thus, the generic term *item* is used. The tasks identified are

logging on to iRODS, replicating an item within an iRODS zone, updating a stale replica of an item within an iRODS zone, updating all stale replicas within an iRODS zone, replicating an item between iRODS zones, updating a stale replica of an item in another iRODS zone, and updating all stale replicas in another iRODS zone.

2.7.2 Description

The iRODS user replicates data generated from a model run. The user places a copy in two separate collections in the same iRODS zone. The collections belong to two different groups that do not have access to each other's data, but both need the output files. A new model run results in the output files needing an update. The iRODS user refreshes the files and places new copies in the two collections. The user also performs the same steps to place copies of the files into a collection in a separate iRODS zone for access to researchers on a different, but related project that need the data.

2.7.3 Actors

iRODS user

2.7.4 Process steps

1. Basic flow
 - a. Initiate iRODS connection
The user initiates a connection to the iRODS CLI using the *iinit* command.
 - b. Replicate an item within an iRODS zone
The user replicates an item to another resource within the same iRODS zone using the *irepl* command.
2. Alternative flows
 - a. Update a stale replica of an item within an iRODS zone
At basic flow step "Initiate iRODS connection," the user updates a stale replica of an item in another resource within the same iRODS zone with the latest version of the item using the *irepl* command.
 - b. Update all stale replicas of an item within an iRODS zone
At basic flow step "Initiate iRODS connection," the user updates all stale replicas of an item in another resource within the same iRODS zone with the latest version of the item using the *irepl* command.
 - c. Replicate an item between separate iRODS zones
At basic flow step "Initiate iRODS connection," the user replicates an

item to another resource in a separate iRODS zone using the *irsync* command.

- d. Update a stale replica of an item between separate iRODS zones
At basic flow step “Initiate iRODS connection,” the user updates a stale replica of an item in another resource in a separate iRODS zones using the *irsync* command.
- e. Update all stale replicas of an item between separate iRODS zones
At basic flow step “Initiate iRODS connection,” the user updates all stale replicas of an item in another resource in a separate iRODS zones using the *irsync* command.

2.8 UC-7: Manage collections with the CLI

2.8.1 Scope

This UC describes managing collections in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of collections. The tasks identified are logging on to iRODS, creating a collection, granting users rights to the collection, listing rights to an existing collection, and deleting a collection.

2.8.2 Description

The iRODS user creates a new collection to store data derived from river basin remote sensors. This new collection will be shared among several groups that require varying levels of access to the data it will contain. With the creation of the new collection, old collections can be deleted and their data, if needed, can be transferred into the new collection. The user lists the Access Control Lists (ACL) of the collections slated for deletion in order to give the correct users their appropriate access to the new collection.

2.8.3 Actors

iRODS user

2.8.4 Process Steps

1. Basic flow
 - a. Initiate iRODS connection
The user initiates a connection to the iRODS CLI using the *iinit* command.
 - b. Create a collection
The user creates an iRODS collection using the *imkdir* command.

2. Alternative flows

a. Display ACL of a collection

At basic flow step “Initiate iRODS connection,” the user lists an iRODS collection and displays its ACL with the *ils -A -r* command.

b. Grant user rights to a collection

At basic flow step “Initiate iRODS connection,” the user grants write access to an existing iRODS collection using the *ichmod -r write* command.

c. Delete a collection

At basic flow step “Initiate iRODS connection,” the user deletes a collection using the *irm -r* command.

2.9 UC-8: Manage resources with the CLI

2.9.1 Scope

This UC describes managing storage resources in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of resources. The tasks identified are logging on to iRODS, creating a new resource, modifying the name of a resource, modifying the type of a resource, modifying the host a resource is attached to, modifying the path to a resource, updating the paths to data objects and collections stored on a resource, modifying the status of a resource, modifying the free space available on a resource, modifying the contextual information about a resource, rebalancing a coordinating resource node, and removing a resource.

2.9.2 Description

After the system administrator creates a new iRODS zone, new storage resources are created and a portion of the data is moved from an existing zone to the new one while deleting another existing zone after completing transferring its data. The paths to the data are then modified to reflect these changes, including changing the free space available on the previously existing resource that remains.

2.9.3 Actors

iRODS administrator

2.9.4 Process steps

1. Basic flow

a. Initiate iRODS connection

The administrator initiates a connection to the iRODS CLI using the *iinit* command.

b. Make a new resource

The administrator makes a new resource available to the iRODS system using the *iadmin* command with the *mkresc* subcommand.

2. Alternative flows

a. Modify the name of a resource

At basic flow step “Initiate iRODS connection,” the administrator modifies the name of a resource in the iRODS system using the *iadmin* command with the *modresc* subcommand and the *name* switch.

b. Modify the type of a resource

At basic flow step “Initiate iRODS connection,” the administrator modifies the type of a resource in the iRODS system using the *iadmin* command with the *modresc* subcommand and the *type* switch.

c. Modify the host of a resource

At basic flow step “Initiate iRODS connection,” the administrator modifies the domain name system (DNS) host name associated with a resource in the iRODS system using the *iadmin* command with the *modresc* subcommand and the *host* switch.

d. Modify the path to a resource

At basic flow step “Initiate iRODS connection,” the administrator modifies the path to a resource in the iRODS system using the *iadmin* command with the *modresc* subcommand and the *path* switch. (See alternative flow 2.e “Update resource data paths.”)

e. Update resource data paths

At basic flow step “Initiate iRODS connection,” the administrator updates the paths to data on a resource in the iRODS system of which the path has been changed using the *iadmin* command with the *modrescdatapaths* subcommand. (See alternative flow 2.d “Modify the path to a resource”.)

f. Modify the status of a resource

At basic flow step “Initiate iRODS connection,” the administrator modifies the status of a resource in the iRODS system using the *iadmin* command with the *modresc* subcommand and the *status* switch.

- g. Modify the free space available on a resource
At basic flow step “Initiate iRODS connection,” the administrator modifies the free space available on a resource in the iRODS system using the *iadmin* command with the *modresc* subcommand and the *free_space* switch.
- h. Modify the contextual information of a resource
At basic flow step “Initiate iRODS connection,” the administrator modifies the contextual information of a resource in the iRODS system using the *iadmin* command with the *modresc* subcommand and the *context* switch.
- i. Rebalance a coordinating resource node
At basic flow step “Initiate iRODS connection,” the administrator rebalances a coordinating resource node in the iRODS system using the *iadmin* command with the *modresc* subcommand and the *rebalance* switch.
- j. Remove a resource
At basic flow step “Initiate iRODS connection,” the administrator removes a resource from the iRODS system using the *iadmin* command with the *rmresc* subcommand.

2.10 UC-9: Manage policies with the CLI

2.10.1 Scope

This UC describes managing iRODS policies. The term *manage* means the creation, reading, editing, and deletion of policies. The tasks identified are logging on to iRODS, adding a new policy, updating an existing policy, and deleting a policy.

2.10.2 Description

As part of a new research collaboration effort between two workgroups, the iRODS administrator implements a policy that replicates new data automatically when placed in certain designated collections. When a file is placed in this collection, a rule is triggered that sends copies to each team lead of the collaborating groups. The research is a short-term project lasting six months. At the end of the research project, the system administrator logs in and removes the duplication policy.

2.10.3 Actors

System administrator

2.10.4 Process steps

1. Basic flow

a. Authenticate into the iRODS server

The system administrator initiates a connection to the iRODS server via `ssh` and authenticates with their user credentials.

b. Add a new policy to the iRODS server

The system administrator adds a new policy rule base to the iRODS server by creating an accompanying `.re` file and referencing that rule base in the correct `re_rulebase_set` array within the `server_config.json` file.

2. Alternative flows

a. Update an existing policy in the iRODS server

At basic flow step “Authenticate into the iRODS server,” the system administrator updates an existing policy by editing the respective `.re` file.

b. Delete an existing policy from the iRODS server

At basic flow step “Authenticate into the iRODS server,” the system administrator deletes an existing policy by removing the reference to it from the correct `re_rulebase_set` array within the `server_config.json` file. The corresponding `.re` file can be kept, archived, or deleted at the system administrator’s discretion.

2.11 UC-10: Manage users with the CLI

2.11.1 Scope

This UC describes managing users in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of users. The tasks identified are logging on to iRODS, creating a new user, modifying the type of an existing user, modifying the default zone of an existing user, modifying the password of an existing user, adding a Kerberos authentication name to an existing user, removing a Kerberos authentication name from an existing user, removing a PAM-derived password from an existing user, and deleting a user.

2.11.2 Description

The Sensor Systems group manager wishes to make several changes to their group. New employees have been added, increasing the group size. The manager decides to divide the work group into several smaller teams and give administrative privileges to senior team leaders over their respective sub-groups. The manager logs in and adds new accounts with passwords for the new researchers and changes the type of accounts of the existing team leads to reflect their expanded status as administrators. The manager also adds Kerberos authentication to all accounts to comply with two-factor authentication requirements.

2.11.3 Actors

iRODS administrator

2.11.4 Process steps

1. Basic flow
 - a. Initiate iRODS connection
The administrator initiates a connection to the iRODS CLI using the *iinit* command.
 - b. Make a new user
The administrator makes a new user in the iRODS system using the *iadmin* command with the *mkuser* subcommand.
2. Alternative flows
 - a. Modify the type of a user
At basic flow step “Initiate iRODS connection,” the administrator modifies the user account type using the *iadmin* command with the *moduser* subcommand and the *type* switch.
 - b. Modify the default zone of a user
At basic flow step “Initiate iRODS connection,” the administrator modifies the user account default zone using the *iadmin* command with the *moduser* subcommand and the *zone* switch.
 - c. Modify the password of a user
At basic flow step “Initiate iRODS connection,” the administrator modifies the user account password using the *iadmin* command with the *moduser* subcommand and the *password* switch.
 - d. Add Kerberos authentication name to a user
At basic flow step “Initiate iRODS connection,” the administrator adds

- a Kerberos authentication name to a user account using the *iadmin* command with the *aua* subcommand.
- e. Remove Kerberos authentication name from a user
At basic flow step “Initiate iRODS connection,” the administrator removes a Kerberos authentication name from a user account using the *iadmin* command with the *rua* subcommand.
- f. Remove a PAM-derived password from a user
At basic flow step “Initiate iRODS connection,” the administrator removes a PAM-derived password from a user account using the *iadmin* command with the *rpp* subcommand.
- g. Remove a user
At basic flow step “Initiate iRODS connection,” the administrator removes a user account using the *iadmin* command with the *rmuser* subcommand.

2.12 UC-11: Manage groups with the CLI

2.12.1 Scope

This UC describes managing user groups in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of user groups. The tasks identified are logging on to iRODS, making a new group, adding a user to a group, removing a user from a group, and removing a group.

2.12.2 Description

A research project under the Data Analytics work group has reached completion. A new effort based on the results of this research is underway. New teams and roles have been created. The manager needs to create, modify, and delete old groups to reflect these organizational changes. They log into iRODS, create new groups and adds existing users to them. Several of the old groups that were relevant to the completed project are deleted. One of the old research groups is kept and transitions its role to the new project, but users are deleted from it that have transferred to other areas.

2.12.3 Actors

iRODS administrator

2.12.4 Process steps

1. Basic flow
 - a. Initiate iRODS connection
The administrator initiates a connection to the iRODS CLI using the *iinit* command.
 - b. Make a new group
The administrator makes a new user group available to the iRODS system using the *iadmin* command with the *mkgroup* subcommand.
2. Alternative flows
 - a. Add a user to a group
At basic flow step “Initiate iRODS connection,” the administrator adds a user to a group using the *iadmin* command with the *atg* subcommand.
 - b. Remove a user from a group
At basic flow step “Initiate iRODS connection,” the administrator removes a user from a group using the *iadmin* command with the *rfg* subcommand.
 - c. Remove a group
At basic flow step “Initiate iRODS connection,” the administrator removes a group using the *iadmin* command with the *rmgroup* subcommand.

2.13 UC-12: Manage zones with the CLI

2.13.1 Scope

This UC describes managing zones in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of zones. The tasks identified are logging on to iRODS, making a new zone, modifying the name of a zone, modifying the connection information of a zone, modifying the access mode of a zone, and removing a zone.

2.13.2 Description

The iRODS administrator has decided to add an additional zone to the iRODS deployment because of an increase in users, data, and hardware. After the new equipment is installed and has gone through the necessary software configurations, the iRODS administrator logs on to the system and adds a zone to iRODS. The administrator modifies one of the existing

zones by transferring some of its data to the new zone, and deletes one of the old zones from the system.

2.13.3 Actors

iRODS administrator

2.13.4 Process steps

1. Basic flow

a. Initiate iRODS connection

The administrator initiates a connection to the iRODS CLI using the *iinit* command.

b. Make a new zone

The administrator makes a new zone available to the iRODS system using the *iadmin* command with the *mkzone* subcommand.

2. Alternative flows

a. Modify the name of a zone

At basic flow step “Initiate iRODS connection,” the administrator modifies the name of a zone using the *iadmin* command with the *modzone* subcommand and the *name* switch.

b. Modify the connection information of a zone

At basic flow step “Initiate iRODS connection,” the administrator modifies the connection information of a zone to the parent iRODS metadata catalog (iCAT) server using the *iadmin* command with the *modzone* subcommand and the *conn* switch.

c. Modify the access mode of a zone

At basic flow step “Initiate iRODS connection,” the administrator modifies the access mode of a zone using the *iadmin* command with the *modzonecollacl* subcommand.

d. Remove a zone

At basic flow step “Initiate iRODS connection,” the administrator removes a zone using the *iadmin* command with the *rmzone* subcommand.

2.14 UC-13: Manage tickets with the CLI

2.14.1 Scope

This UC describes managing access tickets in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of tickets. The

tasks identified are logging on to iRODS, making a new ticket, listing the existing tickets, and deleting a ticket.

2.14.2 Description

As part of a new research collaboration effort between two workgroups, the iRODS administrator creates tickets that will allow the two workgroups to access data placed in a new collection. The two groups will have read access to the collection with select members being granted write access. The research is a short-term project lasting six months. At the end of the research project, the iRODS administrator logs in and removes the access tickets.

2.14.3 Actors

iRODS administrator

2.14.4 Process steps

1. Basic flow
 - a. Initiate iRODS connection
The administrator initiates a connection to the iRODS CLI using the *init* command.
 - b. Make a new ticket
The administrator makes a new ticket available to the iRODS system using the *iticket* command with the *create read/write* subcommand followed by the data object or collection name.
2. Alternative flows
 - a. Show existing tickets
At basic flow step “Initiate iRODS connection,” the administrator lists all existing tickets using the *iticket* command with the *ls-all* subcommand.
 - b. Modify a ticket
At basic flow step “Initiate iRODS connection,” the administrator modifies the access granted by a ticket using the *iticket* command with the *mod Ticket-id add/remove* subcommand followed by the group or user name.
 - c. Delete a ticket
At basic flow step “Initiate iRODS connection,” the administrator removes a ticket using the *iticket* command with the *delete* subcommand.

2.15 UC-14: Manage usage quotas with the CLI

2.15.1 Scope

This UC describes managing usage quotas in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of usage quotas. The tasks identified are logging on to iRODS, setting a user quota, setting a group quota, listing existing quotas, and calculating usage toward quotas.

2.15.2 Description

In order to keep the system from being inundated with unnecessary data, and to make the most of limited storage, a group administrator has imposed storage quota limits on the collections under their supervision (logs in and sets storage quotas for several groups). A few months later, the quotas are reviewed and checked to see how much has been used.

2.15.3 Actors

iRODS administrator

2.15.4 Process steps

1. Basic flow
 - a. Initiate iRODS connection
The administrator initiates a connection to the iRODS CLI using the *iinit* command.
 - b. Set a user quota
The administrator sets a quota for a user account using the *iadmin* command with the *suq* subcommand.
2. Alternative flows
 - a. Set a group quota
At basic flow step “Initiate iRODS connection,” the administrator sets a quota for a group using the *iadmin* command with the *sgq* subcommand.
 - b. List existing quotas
At basic flow step “Initiate iRODS connection,” the administrator lists all existing quotas or quotas for an individual user or group using the *iadmin* command with the *lq* subcommand.
 - c. Calculate usage toward quotas
At basic flow step “Initiate iRODS connection,” the administrator calculates current usage toward quotas using the *iadmin* command with the *cu* subcommand.

2.16 UC-15: Manage reports with the CLI

2.16.1 Scope

This UC describes managing reports about the iRODS system. The term *manage* means the creation, reading, editing, and deletion of reports. The tasks identified are logging on to iRODS, creating a new report, and updating an existing report. Both the reading and deleting of generated reports happens fully on the local machine, thus, those UCs do not exist within iRODS.

2.16.2 Description

There is an upcoming infrastructure review to determine if new hardware is needed for the next year. The branch chiefs have directed their ERS team leads to generate resource allocation and use reports for budget review. The ERS team lead logs on to the system and generates a report to reflect their teams' use of storage resources and team sizes.

2.16.3 Actors

ERS team lead

2.16.4 Process steps

1. Basic flow
 - a. Authenticate into the iRODS server
The ERS Team Lead initiates a connection to the iRODS server via *ssh* and authenticates with their user credentials.
 - b. Create a new report
The ERS Team Lead generates a new report using the *curl* command line utility.
2. Alternative Flows
 - a. Update an existing report
At basic flow step "Authenticate into the iRODS server," the ERS Team Lead updates an existing report using the *curl* command line utility.

2.17 UC-16: Search audit records with the CLI

2.17.1 Scope

This UC describes searching for patterns in the audit records of the iRODS system. The tasks identified are logging on to iRODS and searching audit records.

2.17.2 Description

The data owner of the Black Hawk rotorcraft data performs an audit of the access records of the flight sensor database files to determine which users have been accessing the data, their frequency of access, and how much data they have been using.

2.17.3 Actors

Auditor

2.17.4 Process steps

1. Basic flow
 - a. Authenticate into the iRODS server
The Auditor initiates a connection to the iRODS server via *ssh* and authenticates with their user credentials.
 - b. Audit system activity and conditions
The Auditor searches audit records using the *curl* command line utility.

3 UCs for utilizing a GUI

A more user-friendly method for interacting with the iRODS system is through a web-enabled GUI. The benefits a web GUI provides are an intuitive user interface, a cross-platform code base, the ability to access the system from remote devices, portability across devices, and immediate updates when changes are made to the system. The UCs in this chapter are those found to be fully implemented in the iRODS web GUI, each examined for standard users and administrators, and those UCs to be performed with Elasticsearch by an auditor (Figure 2).

Figure 2. UCs for utilizing a web GUI.

3.1 UC-17: Manage sharing with a GUI

3.1.1 Scope

This UC describes managing sharing of data with other users and groups of users in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of data access permissions. The process steps are largely the same for both data objects and collections, thus, the generic term *item* is used. The tasks identified are navigating to the web GUI, authenticating with a Common Access Card (CAC), granting read access on an item, granting write access on an item, granting ownership access on an item, and removing all access permissions on an item.

3.1.2 Description

The iRODS user grants different permissions to data. Certain data has been cleared for access to teams that need it. Junior level engineers are granted read access to the files, and senior level staff are granted write access. The user also takes away all access to information that has been classified.

3.1.3 Actors

iRODS user

3.1.4 Process steps

1. Basic flow
 - a. Navigate to the web GUI
The user accesses the web GUI with a web browser.
 - b. Authenticate with CAC
The user logs into the web GUI using their CAC.
 - c. Grant read access on an item
The user grants read access on an item in the iRODS system using the web GUI by selecting the item to which access should be granted, clicking on the link for managing permissions, adding the appropriate user or group, and assigning the *read* permission.
2. Alternative Flows
 - a. Grant write access on an item
At basic flow step “Authenticate with CAC,” the user grants write access on an item in the iRODS system using the web GUI by selecting the item to which access should be granted, clicking on the link for

managing permissions, adding the appropriate user or group, and assigning the *write* permission.

- b. Grant ownership access on an item

At basic flow step “Authenticate with CAC,” the user grants ownership access on an item in the iRODS system using the web GUI by selecting the item to which access should be granted, clicking on the link for managing permissions, adding the appropriate user or group, and assigning the *own* permission.

- c. Remove all access permissions on an item

At basic flow step “Authenticate with CAC,” the user removes all access permissions on an item in the iRODS system using the web GUI by selecting the item from which access should be removed, clicking on the link for managing permissions, selecting the appropriate user or group, and assigning the null permission.

3.2 UC-18: Search for data with a GUI

3.2.1 Scope

This UC describes searching for data in the iRODS system. The process steps are largely the same for both data objects and collections, thus, the generic term *item* is used. The tasks identified are navigating to the web GUI, authenticating with a CAC, searching for an item with a known name, searching for an item with a known metadata entry, and searching for an item with a known property.

3.2.2 Description

The iRODS user needs to initiate a simulation model validation. The process requires several input files. The user knows the exact name of some of the files and knows partial names or metadata attributes of others. The user searches for the necessary files with a variety of methods: exact name, wildcard, and metadata attributes by entering in the search terms in the appropriate GUI fields.

3.2.3 Actors

iRODS user

3.2.4 Process steps

1. Basic flow

a. Navigate to the web GUI

The user accesses the web GUI with a web browser.

b. Authenticate with CAC

The user logs into the web GUI using their CAC.

c. Search for an item

The user searches for an item with a known name in the iRODS system using the web GUI by entering their search terms in the search bar, which filters the items by the terms entered, displaying those results that match the query.

2. Alternative flows

a. Search for an item with a known metadata entry

At basic flow step “Authenticate with CAC,” the user searches for an item with a known metadata entry in the iRODS system using the web GUI by entering their search terms in the search bar and specifying that the search is on metadata.

b. Search for an item with a known property

At basic flow step “Authenticate with CAC,” the user searches for an item with a known property in the iRODS system using the web GUI by entering their search terms in the search bar and specifying that the search is on item properties.

3.3 UC-19: Manage metadata with a GUI

3.3.1 Scope

This UC describes managing metadata in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of metadata. The process steps are largely the same for data objects, collections, resources, and users, thus, the generic term *item* is used. The tasks identified are navigating to the web GUI, authenticating with a CAC, adding metadata to an item, listing the metadata that exists for an item, modifying metadata, and deleting metadata.

3.3.2 Description

After a model validation is complete, the iRODS user needs to add metadata to newly generated outputs and modify existing metadata on the updated input files. The iRODS user logs into the system through the GUI

and adds metadata attributes to the output files. The user then places the new files in their correct collection and changes several existing metadata attributes on the input files and deletes other attributes to reflect the modifications done to them.

3.3.3 Actors

iRODS user

3.3.4 Process steps

1. Basic flow

- a. Navigate to the web GUI
The user accesses the web GUI with a web browser.
- b. Authenticate with CAC
The user logs into the web GUI using their CAC.
- c. Add a metadata entry
The user adds a metadata entry to an item in the iRODS system using the web GUI by clicking the link to add metadata to the item.

2. Alternative flows

- a. List existing metadata entries
At basic flow step “Authenticate with CAC,” the user lists existing metadata entries for an item in the iRODS system using the web GUI by clicking the link to display existing metadata entries for the item.
- b. Modify an existing metadata entry
At basic flow step “Authenticate with CAC,” the user modifies an existing metadata entry for an item in the iRODS system using the web GUI by clicking the link to display existing metadata entries for the item, and then clicking the link to edit the metadata entry to be modified.
- c. Remove a metadata entry
At basic flow step “Authenticate with CAC,” the user removes an existing metadata entry for an item in the iRODS system using the web GUI by clicking the link to display existing metadata entries for the item, and then clicking the link to delete the metadata entry to be removed.

3.4 UC-20: Manage data with a GUI

3.4.1 Scope

This UC describes managing data in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of data. The process steps are largely the same for both data objects and collections, thus, the generic term *item* is used. The tasks identified are navigating to the web GUI, authenticating with a CAC, uploading an item, renaming an item, updating an item, and deleting an item.

3.4.2 Description

The iRODS user logs into the system and moves several files from a model run from their local work directory into iRODS. After this is complete, the user views their directory structure and moves it to an existing collection to update some of the data objects it contains, and then, moves others to different collections.

3.4.3 Actors

iRODS user

3.4.4 Process steps

1. Basic flow
 - a. Navigate to the web GUI
The user accesses the web GUI with a web browser.
 - b. Authenticate with CAC
The user logs into the web GUI using their CAC.
 - c. Upload an item
The user uploads an item from their local machine to the iRODS system using the web GUI by clicking the “Upload” button and selecting the item to upload.
2. Alternative flows
 - a. Rename an item
At basic flow step “Authenticate with CAC,” the user renames an item in the iRODS system using the web GUI by selecting the item, choosing the “Rename” option, changing the name appropriately, and clicking the “Save” button.
 - b. Update an item
At basic flow step “Authenticate with CAC,” the user updates an existing item in the iRODS system using the web GUI by clicking the

“Upload” button and selecting the item on the local machine to replace the item in the iRODS system.

c. Delete an item

At basic flow step “Authenticate with CAC,” the user deletes an item from the iRODS system using the web GUI by selecting the item, choosing the “Delete” option, and clicking “Yes” in the delete confirmation dialog box.

3.5 UC-21: Replicate data with a GUI

3.5.1 Scope

This UC describes replicating data across resources in the iRODS system. The process steps are largely the same for both data objects and collections, thus, the generic term *item* is used. The tasks identified are navigating to the web GUI, authenticating with a CAC, and replicating an item within an iRODS zone.

3.5.2 Description

The iRODS user replicates data generated from a model run. The iRODS user places a copy in two separate collections in the same iRODS zone. The collections belong to two different groups that do not have access to each other’s data, but both need the output files. A new model run results in the output files needing an update. The iRODS user refreshes the files and places new copies in the two collections. The user also performs the same steps to place copies of the files into a collection in a separate iRODS zone for access to researchers on a different, but related project that need the data.

3.5.3 Actors

iRODS user

3.5.4 Process steps

1. Basic flow

a. Navigate to the web GUI

The user accesses the web GUI with a web browser.

b. Authenticate with CAC

The user logs into the web GUI using their CAC.

c. Replicate an item within an iRODS zone

The user replicates an item to another resource within the same iRODS

zone using the web GUI by selecting the item to replicate and then clicking the link for replicating items and selecting the appropriate resource to which the item should be replicated.

3.6 UC-22: Manage collections with a GUI

3.6.1 Scope

This UC describes managing collections in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of collections. The tasks identified are navigating to the web GUI, authenticating with a CAC, creating a collection, granting users rights to the collection, and deleting a collection.

3.6.2 Description

The iRODS user creates a new collection to store data derived from river basin remote sensors. This new collection will be shared among several groups that require varying levels of access to the data it will contain. With the creation of the new collection, old collections can be deleted and their data, if needed, can be transferred into the new collection. The user lists the ACL of the collections slated for deletion in order to give the correct users their appropriate access to the new collection.

3.6.3 Actors

iRODS user

3.6.4 Process steps

1. Basic flow
 - a. Navigate to the web GUI
The user accesses the web GUI with a web browser.
 - b. Authenticate with CAC
The user logs into the web GUI using their CAC.
 - c. Create a collection
The user creates a new collection by clicking the link for adding a new collection.
2. Alternative flows
 - a. Grant access to a collection
At basic flow step “Authenticate with CAC,” the user, navigates to their existing collections using the web GUI, clicks on an iRODS collection and grants a user write access.

- b. Delete a collection

At basic flow step “Authenticate with CAC,” the user, navigates to their existing collections using the web GUI, clicks on an iRODS collection to select the collection to be removed, and clicks on the link for deleting the collection.

3.7 UC-23: Manage resources with a GUI

3.7.1 Scope

This UC describes managing storage resources in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of resources. The tasks identified are navigating to the web GUI, authenticating with a CAC, creating a new resource, displaying information about a resource, and removing a resource.

3.7.2 Description

The iRODS administrator decides to add additional hardware to the iRODS deployment to accommodate an influx of new researchers, projects, and data. The administrator creates a new iRODS zone with new storage resources and moves a portion of the data from an existing zone to the new one and deletes another existing zone after completing transferring its data to new storage resources. The iRODS administrator then modifies the paths to the data to reflect these changes, and also changes the free space available on the previously existing resource that remains.

3.7.3 Actors

iRODS administrator

3.7.4 Process steps

1. Basic flow

- a. Navigate to the web GUI

The administrator accesses the web GUI with a web browser.

- b. Authenticate with CAC

The administrator logs into the web GUI using their CAC.

- c. Make a new resource

The administrator makes a new resource available to the iRODS system using the web GUI by navigating to the resources list and clicking the link for adding a new resource.

2. Alternative Flows

a. Display information about a resource

At basic flow step “Authenticate with CAC,” the administrator displays information about a resource in the iRODS system using the web GUI by navigating to the resources list and clicking the information link for the appropriate resource.

b. Remove a resource

At basic flow step “Authenticate with CAC,” the administrator removes a resource from the iRODS system using the web GUI by navigating to the resources list and clicking the delete link for the appropriate resource.

3.8 UC-24: Manage users with a GUI

3.8.1 Scope

This UC describes managing users in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of users. The tasks identified are navigating to the web GUI, authenticating with a CAC, making a new user, modifying the type of an existing user, modifying the default zone of an existing user, modifying the password of an existing user, and deleting a user.

3.8.2 Description

The Sensor Systems group manager has hired two new researchers, transferred several members from another group into Sensor Systems, and one researcher transferred out to work on a separate project. The group has grown large enough that the manager decides to divide the work group into several smaller teams and give administrative privileges to senior team leaders over their respective sub-groups. The manager logs in and adds new accounts with passwords for the new researchers, changes the type of accounts of the existing team leads to reflect their expanded status as admins, and deletes the account of the researcher that is moving to a new team. The manager also adds Kerberos authentication to all accounts to comply with two-factor authentication requirements.

3.8.3 Actors

iRODS administrator

3.8.4 Process Steps

1. Basic flow

- a. Navigate to the web GUI
The administrator accesses the web GUI with a web browser.
- b. Authenticate with CAC
The administrator logs into the web GUI using their CAC.
- c. Make a new user
The administrator makes a new user in the iRODS system using the web GUI by navigating to the users list and clicking the link for adding a new user.

2. Alternative flows

- a. Modify the type of a user
At basic flow step “Authenticate with CAC,” the administrator modifies the user account type by navigating to the users list, selecting the user to modify, clicking the link for editing the user, and modifying the user account type accordingly.
- b. Modify the default zone of a user
At basic flow step “Authenticate with CAC,” the administrator modifies the user account default zone by navigating to the users list, selecting the user to modify, clicking the link for editing the user, and modifying the user account default zone accordingly.
- c. Modify the password of a user
At basic flow step “Authenticate with CAC,” the administrator modifies the user account password by navigating to the users list, selecting the user to modify, clicking the link for editing the user, and modifying the user account password accordingly.
- d. Remove a user
At basic flow step “Authenticate with CAC,” the administrator removes a user account by navigating to the users list, selecting the user to remove, and clicking the link for deleting the account.

3.9 UC-25: Manage groups with a GUI

3.9.1 Scope

This UC describes managing user groups in the iRODS system. The term *manage* means the creation, reading, editing, and deletion of user groups. The tasks identified are navigating to the web GUI, authenticating with a CAC, making a new group, adding a user to a group, removing a user from a group, and removing a group.

3.9.2 Description

A research project under the Data Analytics work group has reached completion. A new effort based on the results of this research is underway. New teams and roles have been created. The manager needs to create, modify, and delete old groups to reflect these organizational changes. The iRODS administrator logs into iRODS, creates new groups, and adds existing users to them. Several of the old groups that were relevant to the completed project are deleted. Also, the iRODS administrator keeps one of the old research groups and transitions its role to the new project, but deletes users from it that have transferred to other areas.

3.9.3 Actors

iRODS administrator

3.9.4 Process steps

1. Basic flow
 - a. Navigate to the web GUI
The administrator accesses the web GUI with a web browser.
 - b. Authenticate with CAC
The administrator logs into the web GUI using their CAC.
 - c. Make a new group
The administrator makes a new user in the iRODS system using the web GUI by navigating to the groups list and clicking the link for adding a new group.
2. Alternative flows
 - a. Add a user to a group
At basic flow step “Authenticate with CAC,” the administrator adds a user to a group in the iRODS system using the web GUI by navigating to the groups list, selecting the group to which the user should be

added, clicking the link for editing the group, and adding the user to the list of users for this group.

b. Remove a user from a group

At basic flow step “Authenticate with CAC,” the administrator removes a user from a group in the iRODS system using the web GUI by navigating to the groups list, selecting the group from which the user should be removed, clicking the link for editing the group, and removing the user from the list of users for this group.

c. Remove a group

At basic flow step “Authenticate with CAC,” the administrator removes a group from the iRODS system using the web GUI by navigating to the groups list, selecting the group that should be removed, and clicking the link for deleting the group.

3.10 UC-26: Manage reports with a GUI

3.10.1 Scope

This UC describes managing reports about the iRODS system. The term *manage* means the creation, reading, editing, and deletion of reports. The tasks identified are navigating to the reporting dashboard, authenticating with a CAC, creating a new report, and updating an existing report. Both the reading and deleting of generated reports happens fully on the local machine, thus, those UCs do not exist within iRODS.

3.10.2 Description

There is an upcoming infrastructure review to determine if new hardware is needed for the next year. The branch chiefs have directed their ERS team leads to generate resource allocation and use reports for budget review. The ERS team lead logs on to the system and generates a report to reflect their teams’ use of storage resources and team sizes.

3.10.3 Actors

ERS team lead

3.10.4 Process steps

1. Basic flow

a. Navigate to the reporting and auditing dashboard

The ERS Team Lead accesses the reporting and auditing dashboard with a web browser.

- b. Authenticate with CAC

The ERS Team Lead logs into the reporting and auditing dashboard using their CAC.

- c. Create a new report

The ERS Team Lead generates a new report by navigating to the reporting page and clicking the link for generating a new report, which can be saved to the local machine as a portable document format (PDF) file.

2. Alternative flows

- a. Update an existing report

At basic flow step “Authenticate with CAC,” the ERS Team Lead updates an existing report by navigating to the reporting page and clicking the link for generating a new report, which can be saved with the same name as the previous report and this will overwrite the file with the new data.

3.11 UC-27: Search audit records with a GUI

3.11.1 Scope

This UC describes searching for patterns in the audit records of the iRODS system. The tasks identified are navigating to the auditing dashboard, authenticating with a CAC, and searching audit records.

3.11.2 Description

The data owner of the Black Hawk rotorcraft data performs an audit of the access records of the flight sensor database files to determine which users have been accessing the data, their frequency of access, and how much of the data they have been using.

3.11.3 Actors

Auditor

3.11.4 Process steps

1. Basic flow

- a. Navigate to the reporting and auditing dashboard

The Auditor accesses the reporting and auditing dashboard with a web browser.

- b. Authenticate with CAC
The Auditor logs into the reporting and auditing dashboard using their CAC.
- c. Audit system activity and conditions
The Auditor audits system activity and conditions by navigating to the auditing page and querying the saved audit reports that have been generated by the auditing function of the iRODS system.

3.12 UC-28: Audit requests

3.12.1 Scope

This UC begins when a user initiates any of the previous GUI-based use cases. Because a web GUI must utilize the iRODS Application Programming Interface (API) for managing users, groups, data, metadata, etc., every request to iRODS from the web GUI is duly audited with the iRODS Auditing Plugin, just as if the command were executed through iRODS directly. This UC is fully automated and neither requires, nor allows, for interaction from the user, thus, there are no process steps for this use case. The UC ends when the triggering event is complete.

3.12.2 Actors

The iRODS auditing plugin

4 Functional Requirements

4.1 Requirements derived from CLI UCs

4.1.1 Derived from UC-1: Manage user-defined rules with the CLI

- **FR-1:** The system will allow users to create system-wide rules defining actions to be performed by the system using the CLI.
- **FR-2:** The system will allow users to list information about pending user-defined rules using the CLI.
- **FR-3:** The system will allow users to update rules that they have created using the CLI.
- **FR-4:** The system will allow users to update rules that apply to data that they own using the CLI.
- **FR-5:** The system will allow users to delete rules that they have created using the CLI.

4.1.2 Derived from UC-2: Manage sharing with the CLI

- **FR-6:** The system will allow users to grant read access on data objects that they own to another user account using the CLI.
- **FR-7:** The system will allow users to grant read access on collections that they own to another user account using the CLI.
- **FR-8:** The system will allow users to grant write access on data objects that they own to another user account using the CLI.
- **FR-9:** The system will allow users to grant write access on collections that they own to another user account using the CLI.
- **FR-10:** The system will allow users to grant ownership access on data objects that they own to another user account using the CLI.
- **FR-11:** The system will allow users to grant ownership access on collections that they own to another user account using the CLI.
- **FR-12:** The system will allow users to revoke access permissions on a data object that they own from another user account using the CLI.
- **FR-13:** The system will allow users to revoke access permissions on a collection that they own from another user account using the CLI.

4.1.3 Derived from UC-3: Search for data with the CLI

- **FR-14:** The system will allow users to search for data objects with exact name queries using the CLI.

- **FR-15:** The system will allow users to search for collections with exact name queries using the CLI.
- **FR-16:** The system will allow users to search for data object with wildcard characters using the CLI.
- **FR-17:** The system will allow users to search for collections with wildcard characters using the CLI.
- **FR-18:** The system will allow users to search for data objects with known metadata attributes using the CLI.
- **FR-19:** The system will allow users to search for collections with known metadata attributes using the CLI.
- **FR-20:** The system will allow users to list a data object's attributes using the CLI.
- **FR-21:** The system will allow users to list a collection's directory structure using the CLI.

4.1.4 Derived from UC-4: Manage metadata with the CLI

- **FR-22:** The system will allow users to add metadata entries to data objects that they own using the CLI.
- **FR-23:** The system will allow users to add metadata entries to collections that they own using the CLI.
- **FR-24:** The system will allow users to add metadata entries to resources that they control using the CLI.
- **FR-25:** The system will allow administrators to add metadata entries to other user accounts using the CLI.
- **FR-26:** The system will allow users to list existing metadata entries for data objects using the CLI.
- **FR-27:** The system will allow users to list existing metadata entries for collections using the CLI.
- **FR-28:** The system will allow users to list existing metadata entries for resources using the CLI.
- **FR-29:** The system will allow users to list existing metadata entries for other user accounts using the CLI.
- **FR-30:** The system will allow users to modify existing metadata entries on data objects that they own using the CLI.
- **FR-31:** The system will allow users to modify existing metadata entries on collections that they own using the CLI.
- **FR-32:** The system will allow users to modify existing metadata entries on resources that they control using the CLI.
- **FR-33:** The system will allow administrators to modify existing metadata entries on other user accounts using the CLI.

- **FR-34:** The system will allow users to remove metadata entries from data objects that they own using the CLI.
- **FR-35:** The system will allow users to remove metadata entries from collections that they own using the CLI.
- **FR-36:** The system will allow users to remove metadata entries from resources that they control using the CLI.
- **FR-37:** The system will allow administrators to remove metadata entries from other user accounts using the CLI.

4.1.5 Derived from UC-5: Manage data with the CLI

- **FR-38:** The system will only allow users to make changes to data they have permission to modify.
- **FR-39:** The system will allow users to upload files as data objects using the CLI.
- **FR-40:** The system will allow users to upload directories as collections using the CLI.
- **FR-41:** The system will allow users to rename data objects using the CLI.
- **FR-42:** The system will allow users to rename collections using the CLI.
- **FR-43:** The system will allow users to move data objects between collections using the CLI.
- **FR-44:** The system will allow users to move collections between collections using the CLI.
- **FR-45:** The system will allow users to update data objects using the CLI.
- **FR-46:** The system will allow users to update collections using the CLI.
- **FR-47:** The system will allow users to delete existing data objects using the CLI.
- **FR-48:** The system will allow users to delete existing collections using the CLI.

4.1.6 Derived from UC-6: Replicate data with the CLI

- **FR-49:** The system will allow users to replicate data objects between storage resources using the CLI.
- **FR-50:** The system will allow users to replicate collections between storage resources using the CLI.

- **FR-51:** The system will allow users to update replicated copies of data objects using the CLI.
- **FR-52:** The system will allow users to update replicated copies of collections using the CLI.

4.1.7 Derived from UC-7: Manage collections with the CLI

- **FR-53:** The system will allow users to create collections using the CLI.
- **FR-54:** The system will allow users to grant access to collections they control using the CLI.
- **FR-55:** The system will allow users to create symbolic links to existing collections they control using the CLI.
- **FR-56:** The system will allow users to list ACLs to collections they control using the CLI.
- **FR-57:** The system will allow the user to delete collections they control using the CLI.

4.1.8 Derived from UC-8: Manage resources with the CLI

- **FR-58:** The system will allow administrators to create a new resource using the CLI.
- **FR-59:** The system will allow administrators to modify the name of an existing resource using the CLI.
- **FR-60:** The system will allow administrators to modify the type of an existing resource using the CLI.
- **FR-61:** The system will allow administrators to modify the host of an existing resource using the CLI.
- **FR-62:** The system will allow administrators to modify the path of an existing resource using the CLI.
- **FR-63:** The system will allow administrators to update the paths to data objects located on a resource of which the path has been changed using the CLI.
- **FR-64:** The system will allow administrators to modify the status of a resource using the CLI.
- **FR-65:** The system will allow administrators to modify the amount of free space available on a resource using the CLI.
- **FR-66:** The system will allow administrators to modify the contextual information of a resource using the CLI.
- **FR-67:** The system will allow administrators to rebalance a coordinating resource node using the CLI.

- **FR-68:** The system will allow administrators to remove a resource using the CLI.

4.1.9 Derived from UC-9: Manage policies with the CLI

- **FR-69:** The system will allow administrators to authenticate via SSH from the CLI.
- **FR-70:** The system will allow administrators to add a new policy rule base using the CLI.
- **FR-71:** The system will allow administrators to update an existing policy rule base using the CLI.
- **FR-72:** The system will allow administrators to delete an existing policy rule base using the CLI.

4.1.10 Derived from UC-10: Manage users with the CLI

- **FR-73:** The system will allow administrators to create new user accounts using the CLI.
- **FR-74:** The system will allow administrators to modify the user account type using the CLI.
- **FR-75:** The system will allow administrators to modify the user account default zone using the CLI.
- **FR-76:** The system will allow administrators to modify the user account password using the CLI.
- **FR-77:** The system will allow administrators to add a Kerberos authentication name to a user account using the CLI.
- **FR-78:** The system will allow administrators to remove a Kerberos authentication name from a user account using the CLI.
- **FR-79:** The system will allow administrators to remove user accounts using the CLI.

4.1.11 Derived from UC-11: Manage groups with the CLI

- **FR-80:** The system will allow administrators to create new groups using the CLI.
- **FR-81:** The system will allow administrators to add users to a group using the CLI.
- **FR-82:** The system will allow administrators to remove users from a group using the CLI.
- **FR-83:** The system will allow administrators to delete groups using the CLI.

4.1.12 Derived from UC-12: Manage zones with the CLI

- **FR-84:** The system will allow administrators to make a new zone using the CLI.
- **FR-85:** The system will allow administrators to modify the zone name using the CLI.
- **FR-86:** The system will allow administrators to modify the connection information of a zone to its parent iCAT server using the CLI.
- **FR-87:** The system will allow administrators to modify the access mode of an existing zone using the CLI.
- **FR-88:** The system will allow administrators to remove an existing zone using the CLI.

4.1.13 Derived from UC-13: Manage tickets with the CLI

- **FR-89:** The system will allow administrators to create new access tickets using the CLI.
- **FR-90:** The system will allow administrators to list the types of tickets available using the CLI.
- **FR-91:** The system will allow administrators to list detailed information about a ticket using the CLI.
- **FR-92:** The system will allow administrators to remove access tickets using the CLI.

4.1.14 Derived from UC-14: Manage usage quotas with the CLI

- **FR-93:** The system will allow administrators to set usage quotas for user accounts using the CLI.
- **FR-94:** The system will allow administrators to set usage quotas for groups using the CLI.
- **FR-95:** The system will allow administrators to list all existing quotas for user accounts using the CLI.
- **FR-96:** The system will allow administrators to list all existing quotas for a group using the CLI.
- **FR-97:** The system will allow administrators to calculate current usage toward quota limits using the CLI.

4.1.15 Derived from UC-15: Manage reports with the CLI

- **FR-98:** The system will allow ERS Team Leads to generate new reports about the system using the CLI.

4.1.16 Derived from UC-16: Search audit records with the CLI

- **FR-99:** The system will allow Auditors to search audit records using the CLI.

4.2 Requirements derived from GUI UCs

4.2.1 Derived from UC-17: Manage sharing with a GUI

- **FR-100:** The system will allow users to grant read access on data objects that they own to another user account using a GUI.
- **FR-101:** The system will allow users to grant read access on collections that they own to another user account using a GUI.
- **FR-102:** The system will allow users to grant write access on data objects that they own to another user account using a GUI.
- **FR-103:** The system will allow users to grant write access on collections that they own to another user account using a GUI.
- **FR-104:** The system will allow users to grant ownership access on data objects that they own to another user account using a GUI.
- **FR-105:** The system will allow users to grant ownership access on collections that they own to another user account using a GUI.
- **FR-106:** The system will allow users to revoke access permissions on a data object that they own from another user account using a GUI.
- **FR-107:** The system will allow users to revoke access permissions on a collection that they own from another user account using a GUI.

4.2.2 Derived from UC-18: Search for data with a GUI

- **FR-108:** The system will allow users to search for data objects with exact-name queries using a GUI.
- **FR-109:** The system will allow users to search for collections with exact-name queries using a GUI.
- **FR-110:** The system will allow users to list a collection's directory structure using a GUI.
- **FR-111:** The system will allow users to search for data objects with known metadata attributes using a GUI.
- **FR-112:** The system will allow users to search for collections with known metadata attributes using a GUI.
- **FR-113:** The system will allow users to search for data objects with a known property using a GUI.
- **FR-114:** The system will allow users to search for collections with a known property using a GUI.

4.2.3 Derived from UC-19: Manage metadata with a GUI

- **FR-115:** The system will allow users to add metadata entries to data objects that they own using a GUI.
- **FR-116:** The system will allow users to add metadata entries to collections that they own using a GUI.
- **FR-117:** The system will allow users to add metadata entries to resources that they control using a GUI.
- **FR-118:** The system will allow administrators to add metadata entries to other user accounts using a GUI.
- **FR-119:** The system will allow users to list existing metadata entries for data objects using a GUI.
- **FR-120:** The system will allow users to list existing metadata entries for collections using a GUI.
- **FR-121:** The system will allow users to list existing metadata entries for resources using a GUI.
- **FR-122:** The system will allow users to list existing metadata entries for other user accounts using a GUI.
- **FR-123:** The system will allow users to modify existing metadata entries on data objects that they own using a GUI.
- **FR-124:** The system will allow users to modify existing metadata entries on collections that they own using a GUI.
- **FR-125:** The system will allow users to modify existing metadata entries on resources that they control using a GUI.
- **FR-126:** The system will allow administrators to modify existing metadata entries on other user accounts using a GUI.
- **FR-127:** The system will allow users to remove metadata entries from data objects that they own using a GUI.
- **FR-128:** The system will allow users to remove metadata entries from collections that they own using a GUI.
- **FR-129:** The system will allow users to remove metadata entries from resources that they control using a GUI.
- **FR-130:** The system will allow administrators to remove metadata entries from other user accounts using a GUI.

4.2.4 Derived from UC-20: Manage data with a GUI

- **FR-131:** The system will allow users to list a collection's directory structure using a GUI.
- **FR-132:** The system will allow users to upload files as data objects using a GUI.

- **FR-133:** The system will allow users to upload directories as collections using a GUI.
- **FR-134:** The system will allow users to rename data objects using a GUI.
- **FR-135:** The system will allow users to rename collections using a GUI.
- **FR-136:** The system will allow users to place data objects into collections using a GUI.
- **FR-137:** The system will allow users to move data objects from one collection to another using a GUI.
- **FR-138:** The system will allow users to update data objects using a GUI.
- **FR-139:** The system will allow users to update collections using a GUI.
- **FR-140:** The system will allow users to delete existing data objects using a GUI.
- **FR-141:** The system will allow users to delete existing collections using a GUI.

4.2.5 Derived from UC-21: Replicate data with a GUI

- **FR-142:** The system will allow users to replicate data objects to multiple storage resources using a GUI.
- **FR-143:** The system will allow users to replicate collections to multiple storage resources using a GUI.

4.2.6 Derived from UC-22: Manage collections with a GUI

- **FR-144:** The system will allow users to create directories using a GUI.
- **FR-145:** The system will allow users to delete collections using a GUI.

4.2.7 Derived from UC-23: Manage resources with a GUI

- **FR-146:** The system will allow administrators to make new storage resources using a GUI.
- **FR-147:** The system will allow administrators to display resource information using a GUI.
- **FR-148:** The system will allow administrators to remove storage resources using a GUI.

4.2.8 Derived from UC-24: Manage users with a GUI

- **FR-149:** The system will require users to authenticate via CAC from a GUI.
- **FR-150:** The system will allow administrators to create new user accounts using a GUI.
- **FR-151:** The system will allow administrators to modify the user account type using a GUI.
- **FR-152:** The system will allow administrators to modify the user account default zone using a GUI.
- **FR-153:** The system will allow administrators to modify the user account password using a GUI.
- **FR-154:** The system will allow administrators to remove user accounts using a GUI.

4.2.9 Derived from UC-25: Manage groups with a GUI

- **FR-155:** The system will allow administrators to create new groups using a GUI.
- **FR-156:** The system will allow administrators to add users to a group using a GUI.
- **FR-157:** The system will allow administrators to remove users from a group using a GUI.
- **FR-158:** The system will allow administrators to delete groups using a GUI.

4.2.10 Derived from UC-26: Manage reports with a GUI

- **FR-159:** The system will allow ERS Team Leads to generate new reports about the system using a GUI.

4.2.11 Derived from UC-27: Search audit records with a GUI

- **FR-160:** The system will allow Auditors to search audit records using a GUI.

4.3 Security and auditing

4.3.1 Derived from UC-1: Manage user-defined rules with the CLI

- **FR-161:** The system will audit the creation of user-defined rules.

- **FR-162:** The system will audit the listing of pending user-defined rules.
- **FR-163:** The system will audit the modifying of user-defined rules.
- **FR-164:** The system will audit the deleting of user-defined rules.

4.3.2 Derived from UC-2 and UC-17: Managing sharing with the CLI; Managing sharing with a GUI

- **FR-165:** The system will audit the granting of read access on data objects.
- **FR-166:** The system will audit the granting of read access on collections.
- **FR-167:** The system will audit the granting of write access on data objects.
- **FR-168:** The system will audit the granting of write access on collections.
- **FR-169:** The system will audit the granting of ownership access on data objects.
- **FR-170:** The system will audit the granting of ownership access on collections.
- **FR-171:** The system will audit the revocation of access permissions on data objects.
- **FR-172:** The system will audit the revocation of access permissions on collections.

4.3.3 Derived from UC-3 and UC-18: Search for data with the CLI; Search for data with a GUI

- **FR-173:** The system will audit the searching for data objects with exact-name queries.
- **FR-174:** The system will audit the searching for collections with exact-name queries.
- **FR-175:** The system will audit the searching for data objects with wildcard queries.
- **FR-176:** The system will audit the searching for collections with wildcard queries.
- **FR-177:** The system will audit the searching for data objects with metadata attribute queries.
- **FR-178:** The system will audit the searching for collections with metadata attribute queries.
- **FR-179:** The system will audit the listing of a data object's attributes.

- **FR-180:** The system will audit the listing of a collection's directory structure.

4.3.4 Derived from UC-4 and UC-19: Manage metadata with the CLI; Manage metadata with a GUI

- **FR-181:** The system will audit the addition of metadata entries to data objects.
- **FR-182:** The system will audit the addition of metadata entries to collections.
- **FR-183:** The system will audit the addition of metadata entries to resources.
- **FR-184:** The system will audit the addition of metadata entries to user accounts.
- **FR-185:** The system will audit the listing of metadata entries for data objects.
- **FR-186:** The system will audit the listing of metadata entries for collections.
- **FR-187:** The system will audit the listing of metadata entries for resources.
- **FR-188:** The system will audit the listing of metadata entries for user accounts.
- **FR-189:** The system will audit the modification of metadata entries for data objects.
- **FR-190:** The system will audit the modification of metadata entries for collections.
- **FR-191:** The system will audit the modification of metadata entries for resources.
- **FR-192:** The system will audit the modification of metadata entries for user accounts.
- **FR-193:** The system will audit the deleting of metadata entries from data objects.
- **FR-194:** The system will audit the deleting of metadata entries from collections.
- **FR-195:** The system will audit the deleting of metadata entries from resources.
- **FR-196:** The system will audit the deleting of metadata entries from user accounts.

4.3.5 Derived from UC-5 and UC-20: Manage data with the CLI; Manage data with a GUI

- **FR-197:** The system will audit the uploading of files as data objects.
- **FR-198:** The system will audit the uploading of directories as collections.
- **FR-199:** The system will audit the renaming of data objects.
- **FR-200:** The system will audit the renaming of collections.
- **FR-201:** The system will audit the moving of data objects between collections.
- **FR-202:** The system will audit the moving of collections between collections.
- **FR-203:** The system will audit the updating of data objects.
- **FR-204:** The system will audit the updating of collections.
- **FR-205:** The system will audit the deleting of data objects.
- **FR-206:** The system will audit the deleting of collections.

4.3.6 Derived from UC-6 and UC-21: Replicate data with the CLI; Replicate data with a GUI

- **FR-207:** The system will audit the replication of data objects between storage resources.
- **FR-208:** The system will audit the replication of collections between storage resources.
- **FR-209:** The system will audit the updating of replicated copies of data objects.
- **FR-210:** The system will audit the updating of replicated copies of collections.

4.3.7 Derived from UC-7 and UC-22: Manage collections with the CLI; Manage collections with a GUI

- **FR-211:** The system will audit the creating of storage resources.
- **FR-212:** The system will audit the deletion of storage resources.
- **FR-213:** The system will audit the creation of symbolic links to collections.
- **FR-214:** The system will audit the granting of access to collections.
- **FR-215:** The system will audit the removal of access to collections.

4.3.8 Derived from UC-8 and UC-23: Manage resources with the CLI; Manage resources with a GUI

- **FR-216:** The system will audit the creation of new resources.
- **FR-217:** The system will audit the modification of resources.
- **FR-218:** The system will audit the updating of data object paths on modified resources.
- **FR-219:** The system will audit the rebalancing of coordinating resources.
- **FR-220:** The system will audit the removal of resources.

4.3.9 Derived from UC-9: Manage policies with the CLI; Manage policies with a GUI

- **FR-221:** The system will audit the creation of policy rule bases.
- **FR-222:** The system will audit the modification of policy rule bases.
- **FR-223:** The system will audit the deleting of policy rule bases.

4.3.10 Derived from UC-10 and UC-24: Manage users with the CLI; Manage users with a GUI

- **FR-224:** The system will audit the creation of new user accounts.
- **FR-225:** The system will audit the modification of the user accounts.
- **FR-226:** The system will audit the addition of a Kerberos authentication name to a user account.
- **FR-227:** The system will audit the removal of a Kerberos authentication name from a user account.
- **FR-228:** The system will audit the deleting of user accounts.

4.3.11 Derived from UC-11 and UC-25: Manage groups with the CLI; Manage groups with a GUI

- **FR-229:** The system will audit the creation of new groups.
- **FR-230:** The system will audit the addition of users to groups.
- **FR-231:** The system will audit the removal of users from groups.
- **FR-232:** The system will audit the deleting of groups.

4.3.12 Derived from UC-12: Manage zones with the CLI

- **FR-233:** The system will audit the creation of new zones.
- **FR-234:** The system will audit the modification of zones.
- **FR-235:** The system will audit the deleting of zones.

4.3.13 Derived from UC-13: Manage tickets with the CLI

- **FR-236:** The system will audit the creation of new access tickets.
- **FR-237:** The system will audit the modification of access tickets.
- **FR-238:** The system will audit the revocation of access tickets.

4.3.14 Derived from UC-14: Manage usage quotas with the CLI

- **FR-239:** The system will audit the creation of usage quotas for user accounts.
- **FR-240:** The system will audit the creation of usage quotas for groups.
- **FR-241:** The system will audit the listing of existing usage quotas for user accounts.
- **FR-242:** The system will audit the listing of existing usage quotas for groups.
- **FR-243:** The system will audit the calculation of current usage toward quota limits.

4.3.15 General requirements

- **FR-244:** The system will require all users to be authenticated prior to allowing access to the system.

5 Technical Requirements

5.1 System requirements

- **TR-1:** The system will integrate with any POSIX-compliant storage (*default system capability*)
- **TR-2:** The system will use Red Hat Enterprise Linux Server version 7 as the operating system (*Infrastructure Requirement*)
- **TR-3:** The system will have 10-gigabit ethernet available to it, and will utilize this connectivity for inter-server communication (*Infrastructure Requirement*)
- **TR-4:** The system will use parallel file transfer for files larger than 32 megabytes (*Default System Capability*)
- **TR-5:** The system will allocate 30 terabytes of storage to each resource server (*Infrastructure Requirement*)
- **TR-6:** The system will require network connectivity on client workstations (*Default System Requirement*)
- **TR-7:** The system will require a POSIX-compliant host operating system on client workstations for installation of the iRODS iCommands CLI (*Default System Requirement*)
- **TR-8:** The system will require an up-to-date, modern web browser on client workstations for access to the web-based GUI (*Default System Requirement*)
- **TR-9:** The system will require JavaScript to be enabled on the web browser on client workstations (*Default System Requirement*)
- **TR-10:** The system will require users to authenticate before access is granted (*Infrastructure Requirement*)
- **TR-11:** The system will require two-factor authentication (*Infrastructure Requirement*)
- **TR-12:** The system will utilize the user's CAC and PIN for two-factor authentication (*Infrastructure Requirement*)
- **TR-13:** The system will require a valid, non-expired CAC for two-factor authentication (*Infrastructure Requirement*)
- **TR-14:** The system will require users to be within the DSRC network (*Infrastructure Requirement*)
- **TR-15:** The system will integrate with the existing DSRC Kerberos system (*Infrastructure Requirement*)
- **TR-16:** The system will be installed on an isolated virtual local area network (VLAN), and only those ports specified in the Architecture Diagram will be open (*Infrastructure Requirement*)

- **TR-17:** The system will only contain data up to and including Unclassified FOUO (*Infrastructure Requirement*)
- **TR-18:** The system will provide a Python-based API for scripted access (*Design Requirement*)
- **TR-19:** The system will have two iCAT servers behind a single HAProxy load balancer (*Design Requirement*)
- **TR-20:** The system will provide automated ingest by way of landing zone directories (*Design Requirement*)
- **TR-21:** The system will have the capability of enforcing metadata requirement policies (*Design Requirement see: TR-42, TR-43, TR-44, TR-45*)
- **TR-22:** The system will utilize the “Strict” ACL policy (*Design Requirement*)
- **TR-23:** The system will allow access tickets to be created for use by authorized users only (*Design Requirement see: TR-85, TR-86, TR-87*)
- **TR-24:** The system will utilize the user’s default resource for storing data objects if a resource is not specified in the upload process (*Design Requirement*)
- **TR-25:** The system will stage a copy of a data object in the cache resource prior to opening it when the data object is requested for opening (*Design Requirement*)
- **TR-26:** The system will calculate the SHA256 checksum of a data object after it is successfully uploaded or copied (*Design Requirement*)
- **TR-27:** The system will utilize a “trash” folder for each user for recovering accidentally deleted items (*Design Requirement*)
- **TR-28:** The system will be configured such that anonymous and public access will be prohibited (*Design Requirement*)
- **TR-29:** The system will be configured to add the logical path to the storage vault of the resource when generating the physical path for a data object (*Design Requirement*)
- **TR-30:** The system will be configured such that users will be given some leeway with resource quotas; specifically, resource quotas will not be strictly enforced (*Infrastructure Requirement*)
- **TR-31:** The system will have enabled the option for executing post-processing for bulk-put operations (*Design Requirement*)
- **TR-32:** The system will not have a policy in place for purging expired data (*Design Requirement*)

5.2 Security and auditing

5.2.1 Derived from UC-1: Manage user-defined rules with the CLI

- **TR-33:** The create rule audit message will consist of a timestamp, the user issuing the command, the logical path to the rule file (if specified), the name of the rule called (if specified), the input parameters (if specified), the output parameters (if specified), and the operation of “Create rule.” (*UC-1, FR-1*)
- **TR-34:** The list rules audit message will consist of a timestamp, the user issuing the command, the user listed (if specified), and the operation of “List rules.” (*UC-1, FR-2*)
- **TR-35:** The modify rule audit message will consist of a timestamp, the user issuing the command, the ID of the rule modified, the name of the property modified, the value of the property modified, and the operation of “Modify rule.” (*UC-1, FR-3, FR-4*)
- **TR-36:** The delete rule audit message will consist of a timestamp, the user issuing the command, the ID of the rule deleted, and the operation of “Delete rule.” (*UC-1, FR-5*)
- **TR-37:** The grant access audit message will consist of a timestamp, the user issuing the command, the user or group affected, the logical path to the data object or collection to which access is granted, the type of access granted (one of read, write, or own), and the operation of “Grant access.” (*UC-2, UC-17, FR-6, FR-7, FR-8, FR-9, FR-10, FR-11, FR-100, FR-101, FR-102, FR-103, FR-104, FR-105*)
- **TR-38:** The revoke access audit message will consist of a timestamp, the user issuing the command, the user or group affected, the logical path to the data object or collection from which access is revoked, and the operation of “Revoke access.” (*UC-2, UC-17, FR-12, FR-13, FR-106, FR-107*)
- **TR-39:** The search audit message will consist of a timestamp, the user issuing the command, the search terms, and the operation of “Search.” (*UC-3, UC-18, FR-14, FR-15, FR-16, FR-17, FR-18, FR-19, FR-108, FR-109, FR-11, FR-12, FR-13, FR-14*)
- **TR-40:** The data object listing audit message will consist of a timestamp, the user issuing the command, the logical path to the data object, and the operation of “List data object.” (*UC-3, UC-18, FR-20*)
- **TR-41:** The collection listing audit message will consist of a timestamp, the user issuing the command, the logical path to the collection, and the operation of “List collection.” (*UC-3, UC-18, FR-21, FR-110*)

- **TR-42:** The metadata addition audit message will consist of a timestamp, the user issuing the command, the object to which the metadata entry applies (logical path to the data object, collection, or resource; or the name of the user account, in the case of user metadata), the metadata entry, and the operation of “Add metadata.” (*UC-4, UC-19, FR-22, FR-23, FR-24, FR-25, FR-115, FR-116, FR-117, FR-118*)
- **TR-43:** The metadata listing audit message will consist of a timestamp, the user issuing the command, the object from which metadata will be listed (logical path to the data object, collection, or resource; or the name of the user account, in the case of user metadata), and the operation of “List metadata.” (*UC-4, UC-19, FR-26, FR-27, FR-28, FR-29, FR-119, FR-120, FR-121, FR-122*)
- **TR-44:** The metadata modification audit message will consist of a timestamp, the user issuing the command, the object to which the metadata entry applies (logical path to the data object, collection, or resource; or the name of the user account, in the case of user metadata), the metadata entry, and the operation of “Modify metadata.” (*UC-4, UC-19, FR-30, FR-31, FR-32, FR-33, FR-123, FR-124, FR-125, FR-126*)
- **TR-45:** The metadata deletion audit message will consist of a timestamp, the user issuing the command, the object from which the metadata will be removed (logical path to the data object, collection, or resource; or the name of the user account, in the case of user metadata), the metadata entry removed, and the operation of “Remove metadata.” (*UC-4, UC-19, FR-34, FR-35, FR-36, FR-37, FR-127, FR-128, FR-129, FR-130*)
- **TR-46:** The upload file audit message will consist of a timestamp, the user issuing the command, the logical path to the resulting data object, and the operation of “Upload file.” (*UC-5, UC-20, FR-39, FR-132*)
- **TR-47:** The upload directory audit message will consist of a timestamp, the user issuing the command, the logical path to the resulting collection, and the operation of “Upload directory.” (*UC-5, UC-20, FR-40, FR-133*)
- **TR-48:** The rename data object audit message will consist of a timestamp, the user issuing the command, the logical path to the data object, and the operation of “Rename data object.” (*UC-5, UC-20, FR-41, FR-134*)
- **TR-49:** The rename collection audit message will consist of a timestamp, the user issuing the command, the logical path to the

- collection, and the operation of “Rename collection.” (*UC-5, UC-20, FR-42, FR-135*)
- **TR-50:** The move data object audit message will consist of a timestamp, the user issuing the command, the current logical path to the data object, the new logical path to the data object, and the operation of “Move data object.” (*UC-5, UC-20, FR-43, FR-136*)
 - **TR-51:** The move collection audit message will consist of a timestamp, the user issuing the command, the current logical path to the collection, the new logical path to the collection, and the operation of “Move collection.” (*UC-5, UC-20, FR-44, FR-137*)
 - **TR-52:** The update data object audit message will consist of a timestamp, the user issuing the command, the logical path to the data object, and the operation of “Update data object.” (*UC-5, UC-20, FR-45, FR-138*)
 - **TR-53:** The update collection audit message will consist of a timestamp, the user issuing the command, the logical path to the collection, and the operation of “Update collection.” (*UC-5, UC-20, FR-46, FR-139*)
 - **TR-54:** The data object deletion audit message will consist of a timestamp, the user issuing the command, the logical path to the data object, and the operation of “Delete data object.” (*UC-5, UC-20, FR-47, FR-140*)
 - **TR-55:** The collection deletion audit message will consist of a timestamp, the user issuing the command, the logical path to the collection, and the operation of “Delete collection.” (*UC-5, UC-20, FR-48, FR-141*)
 - **TR-56:** The replicate data object audit message will consist of a timestamp, the user issuing the command, the logical path to the data object, the logical paths to the resources replicated to, and the operation of “Replicate data object.” (*UC-6, UC-21, FR-49, FR-142*)
 - **TR-57:** The replicate collection audit message will consist of a timestamp, the user issuing the command, the logical path to the collection, the logical paths to the resources replicated to, and the operation of “Replicate collection.” (*UC-6, UC-21, FR-50, FR-143*)
 - **TR-58:** The update data object audit message will consist of a timestamp, the user issuing the command, the logical path to the data object, the logical paths to the resources updated, and the operation of “Update data object.” (*UC-6, UC-21, FR-51*)
 - **TR-59:** The update collection audit message will consist of a timestamp, the user issuing the command, the logical path to the

collection, the logical paths to the resources updated, and the operation of “Update collection.” (*UC-6, UC-21, FR-52*)

- **TR-60:** The create collection audit message will consist of a timestamp, the user issuing the command, the physical path to the storage device, the logical path to the resulting resource, and the operation of “create collection.” (*UC-7, UC-22, FR-53, FR-144*)
- **TR-61:** The delete collection audit message will consist of a timestamp, the user issuing the command, the physical path to the storage device, the logical path to the resource, and the operation of “delete collection.” (*UC-7, UC-22, FR-57, FR-145*)
- **TR-62:** The create symbolic link to collection audit message will consist of a timestamp, the user issuing the command, the logical path to the collection, the logical path to the resulting symbolic link, and the operation of “Create symbolic link to collection.” (*UC-7, UC-22, FR-55*)
- **TR-63:** The grant access collection audit message will consist of a timestamp, the user issuing the command, the logical path to the collection, and the operation of “grant collection access.” (*UC-7, UC-22, FR-54, FR-101, FR-103, FR-105*)
- **TR-64:** The removal of collection access audit message will consist of a timestamp, the user issuing the command, the logical path to the collection, and the operation of “remove collection access.” (*UC-7, UC-22, FR-106, FR-107*)
- **TR-65:** The resource creation audit message will consist of a timestamp, the user issuing the command, the physical path to the storage resource, the logical path visible to users, and the operation of “Create resource.” (*UC-8, UC-23, FR-58, FR-146*)
- **TR-66:** The resource modification audit message will consist of a timestamp, the user issuing the command, the logical path to the resource, the name of the property modified, the value of the modified property, and the operation of “Modify resource.” (*UC-8, UC-23, FR-59, FR-60, FR-61, FR-64, FR-65, FR-66*)
- **TR-67:** The data object paths update audit message will consist of a timestamp, the user issuing the command, the old logical path to the resource, the new logical path to the resource, and the operation of “Update resource paths.” (*UC-8, UC-23, FR-62, FR-63*)
- **TR-68:** The resource rebalance audit message will consist of a timestamp, the user issuing the command, a list of the logical paths to each coordinating resource, and the operation of “Rebalance resources.” (*UC-8, UC-23, FR-67*)

- **TR-69:** The resource removal audit message will consist of a timestamp, the user issuing the command, the logical path to the resource, and the operation of “Remove resource.” (*UC-8, UC-23, FR-68, FR-148*)
- **TR-70:** The policy rule creation audit message will consist of a timestamp, the user issuing the command, the directory path to the rule file, and the operation of “Create policy rule.” (*UC-9, FR-70*)
- **TR-71:** The policy rule modification audit message will consist of a timestamp, the user issuing the command, the directory path to the rule file, and the operation of “Modify policy rule.” (*UC-9, FR-71*)
- **TR-72:** The policy rule deletion audit message will consist of a timestamp, the user issuing the command, the physical path to the rule file, and the operation of “Delete policy rule.” (*UC-9, FR-72*)
- **TR-73:** The user account creation audit message will consist of a timestamp, the user issuing the command, the name of the account created, and the operation of “Create user.” (*UC-10, UC-24, FR-73, FR-150*)
- **TR-74:** The user account modification audit message will consist of a timestamp, the user issuing the command, the name of the account modified, the property modified, the value of the modified property, and the operation of “Modify user.” (*UC-10, UC-24, FR-74, FR-151, FR-152, FR-153*)
- **TR-75:** The user account Kerberos principal addition audit message will consist of a timestamp, the user issuing the command, the name of the account to which the principal is added, the value of the principal added, and the operation of “Add Kerberos principal.” (*UC-10, UC-24, FR-77*)
- **TR-76:** The user account Kerberos principal removal audit message will consist of a timestamp, the user issuing the command, the name of the account from which the principal is removed, the value of the principal removed, and the operation of “Remove Kerberos principal.” (*UC-10, UC-24, FR-78*)
- **TR-77:** The user account removal audit message will consist of a timestamp, the user issuing the command, the name of the account removed, and the operation of “Remove user.” (*UC-10, UC-24, FR-79, FR-154*)
- **TR-78:** The group creation audit message will consist of a timestamp, the user issuing the command, the name of the group created, and the operation of “Create group.” (*UC-11, UC-25, FR-80, FR-155*)

- **TR-79:** The addition of a user to a group audit message will consist of a timestamp, the user issuing the command, the name of the group, the name of the user being added to the group, and the operation of “Add user to group.” (*UC-11, UC-25, FR-81, FR-156*)
- **TR-80:** The removal of a user from a group audit message will consist of a timestamp, the user issuing the command, the name of the group, the name of the user being removed from the group, and the operation of “Remove user from group.” (*UC-11, UC-25, FR-82, FR-157*)
- **TR-81:** The group deletion audit message will consist of a timestamp, the user issuing the command, the name of the group deleted, and the operation of “Delete group.” (*UC-11, UC-25, FR-83, FR-158*)
- **TR-82:** The zone creation audit message will consist of a timestamp, the user issuing the command, the name of the zone created, and the operation of “Create zone.” (*UC-12, FR-84*)
- **TR-83:** The zone modification audit message will consist of a timestamp, the user issuing the command, the name of the zone modified, the name of the property modified, the value of the modified property, and the operation of “Modify zone.” (*UC-12, FR-85, FR-86, FR-87*)
- **TR-84:** The zone deletion audit message will consist of a timestamp, the user issuing the command, the name of the zone deleted, and the operation of “Delete zone.” (*UC-12, FR-88*)
- **TR-85:** The access ticket creation audit message will consist of a timestamp, the user issuing the command, the access permissions granted by the ticket, the data object or collection to which access is granted, the ticket ID, and the operation of “Create ticket.” (*UC-13, FR-89*)
- **TR-86:** The access ticket modification audit message will consist of a timestamp, the user issuing the command, the name of the property modified, the value of the modified property, the ticket ID, and the operation of “Modify ticket.” (*UC-13, FR-91*)
- **TR-87:** The access ticket revocation audit message will consist of a timestamp, the user issuing the command, the ticket ID, and the operation of “Revoke ticket.” (*UC-13, FR-92*)
- **TR-88:** The usage quota creation audit message will consist of a timestamp, the user issuing the command, the user or group to which the quota is applied, the quota limit, and the operation of “Create quota.” (*UC-13, FR-93, FR-94*)

- **TR-89:** The usage quota listing audit message will consist of a timestamp, the user issuing the command, the user or group specified, and the operation of “List quota.” (*UC-13, FR-95, FR-96*)
- **TR-90:** The usage quota calculation audit message will consist of a timestamp, the user issuing the command, and the operation of “Calculate usage toward quota.” (*UC-13, FR-97*)

6 Conclusion

This report detailed the UCs, the functional requirements, and the technical requirements for the iRODS prototype. The UCs included the scope, description, actors, and process steps as pertaining to the presented use case. The listed functional requirements were organized by the relevant use case(s) and were grouped by relevant objectives. Technical requirements of the prototype cited the relevant use case(s) and functional requirement(s).

Appendix A: Glossary of iRODS Terms

These are terms used throughout this document.

Collection

Represents a logical directory in the iRODS system

Data Object

Represents a logical file in the iRODS system

iCAT

The iRODS Metadata Catalog, a relational database to organize the content of an iRODS zone and to maintain the metadata thereof

PEP

Policy Enforcement Point, a section of code in the iRODS system that is executed when a certain event occurs

Resource

Represents a data container, such as a hard disk or logical drive, in the iRODS system

Zone

An independent iRODS system, consisting of its own iCAT server and resource servers

Acronyms and Abbreviations

Acronym	Meaning
ACL	Access Control Lists
APIs	Application Programming Interfaces
CAC	Common Access Card
CLI	Command Line Interface
CWFS	Center-Wide File System
DNS	Domain Name System
DoD	Department of Defense
DSRC	Defense Supercomputing Resource Center
ERDC	Engineer Research and Development Center
ERS	Engineered Resilient Systems
GUI	Graphic User Interface
HPC	High-Performance Computing
iCAT	iRODS Metadata Catalog
IA	Information Assurance
iRODS	Integrated Rule-Oriented Data System
PDF	Portable Document Format
SRG	Security Requirements Guides
STIGS	Security Technical Implementation Guide
UCs	Use Cases
USACE	U.S. Army Corps of Engineers
VLAN	Virtual Local Area Network

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.				
1. REPORT DATE (DD-MM-YYYY) August 2020		2. REPORT TYPE Final report		3. DATES COVERED (From - To)
4. TITLE AND SUBTITLE Integrated Rule-Oriented Data System (iRODS) & High Performance Computing (HPC) Requirements Document		5a. CONTRACT NUMBER		
		5b. GRANT NUMBER		
		5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Kevin D. Winters, Mark A. Cowan, Glover E. George, Megan E. Gonzalez, Brian Priest, Omar Morris, and Jonathan Landrum		5d. PROJECT NUMBER		
		5e. TASK NUMBER		
		5f. WORK UNIT NUMBER 92L5D8		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Engineer Research and Development Center Information Technology Laboratory 3909 Halls Ferry Road, Vicksburg, MS 39180-6199		8. PERFORMING ORGANIZATION REPORT NUMBER ERDC/ITL SR-20-10		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Headquarters, U.S. Army Corps of Engineers Washington, DC 20314-1000		10. SPONSOR/MONITOR'S ACRONYM(S)		
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT The purpose of this report is to capture all relevant use cases, functional requirements, and technical requirements of the Integrated Rule-Oriented Data System (iRODS) prototype. The use cases (UCs) define the system interactions an iRODS user, iRODS administrator, and an auditor would expect within the system. The functional requirements define the expected behavior of the system to support the individual use cases; functional requirements are grouped in reference to the use cases supported by the set of functional requirements. The technical requirements are defined last and include references to specific functional requirements and use cases supported by the requirement.				
15. SUBJECT TERMS		iRODS (Computer system) High performance computing Computer architecture Computer science		
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT UNCLASSIFIED	b. ABSTRACT UNCLASSIFIED	c. THIS PAGE UNCLASSIFIED	SAR	76
			19a. NAME OF RESPONSIBLE PERSON	
			19b. TELEPHONE NUMBER (include area code)	