

irods-go-drs

Federated analysis meets policy-managed data grids

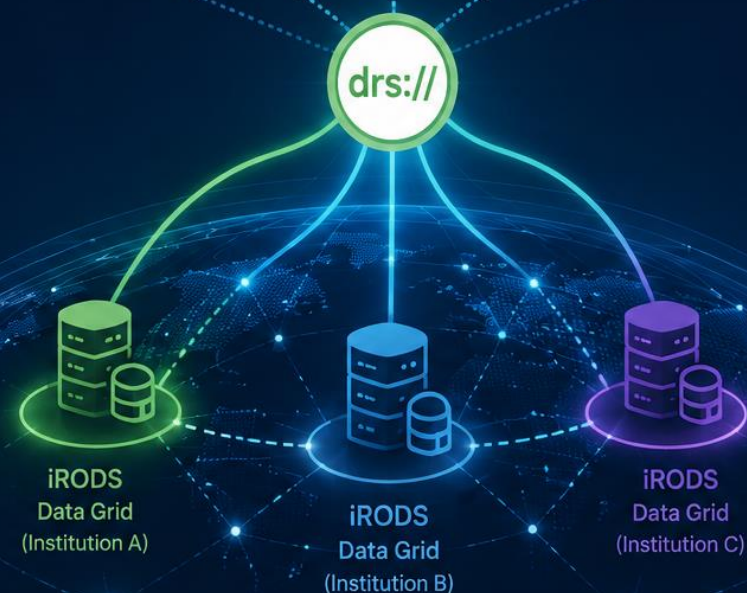
Connecting iRODS-managed data to
GA4GH ecosystems for standards-based
workflows and data federation



github.com/michael-conway/irods-go-drs



iRODS User Group Demo
2026



Data sovereignty at the source. Interoperability at the edge. Insight everywhere.

This presentation was prepared by the author in his personal capacity. The opinions expressed in this article are the author's own and do not reflect the view of the National Institutes of Health, the Department of Health and Human Services or the United States Government.

GA4GH gives the interoperability contracts

Common boundaries for tools, execution, data, and access decisions.



DRS lets workflow systems refer to data without knowing storage-specific details.

Computational tasks can be dispatched to remote locations to minimize data movement and enhance security.

stable object IDs

metadata + checksums

atomic + compound objects

HTTPS / S3 / other access methods

locality-aware access paths

iRODS capabilities + Data Mesh/GA4GH

A policy-managed data grid can become a standards-facing federation node.

CLOUD-HEAVY DRS ECOSYSTEM



Terra / AnVIL

Gen3

BioData Catalyst

Cloud data commons

IRODS OPPORTUNITY

policy-managed namespace
metadata-rich collections
resource hierarchy + replicas
federated zones
REST + S3 + native paths
on-prem / hybrid control

The goal is not to move iRODS data into someone else's platform. The goal is to let iRODS participate on standards-based terms.

Demo stack: irods-grid-stack

github.com/michael-conway/irods-grid-stack

A reproducible local grid with storage, access APIs, auth, UI, and CLI tools.

```
docker compose --profile frontend up -d
```

Keycloak

Starbase

irods

irods-go-drs

terminal
gocmd + drscmd

REST provider/resource

S3 provider/resource

iRODS provider
providerResc

REST provider/resource

S3 provider/resource

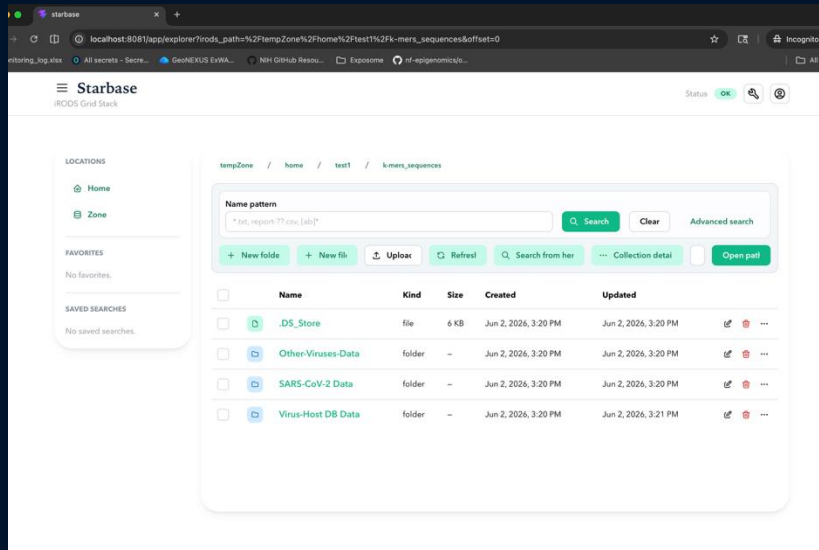
iRODS resource
resourceResc

Starbase + REST Service

Visual metadata editing and a lightweight DRS 1.5 endpoint for the demo.

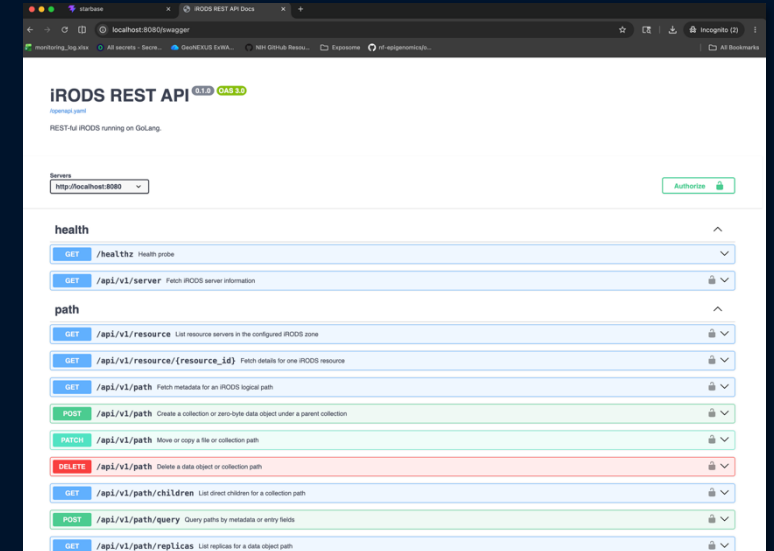
STARBASE IN THE DEMO

Browse demo collections
Edit DRS and child-object metadata
Exercise REST/OIDC/S3-adjacent workflows



REST IN THE DEMO

Lightweight REST service
HATEOAS
Basic and Bearer auth
Swagger + OpenAPI



Starbase is a useful companion: it makes the metadata that drives the DRS demo visible and editable.

Demo path

Follow managed iRODS data through DRS metadata, access methods, and locality.

- 1. Inspect iRODS object and collection**
- 2. Add DRS metadata with drscmd**
- 3. Resolve atomic object**
- 4. Show provider/resource affinity**
- 5. Show HTTPS + S3 access methods**
- 6. Resolve compound object**

DRS service info

A standards-shaped entry point that self-describes the service and implementation.

Responses

Curl

```
curl -X 'GET' \
  'http://localhost:8888/ga4gh/drs/v1/service-info' \
  -H 'accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2kiOiA6Iiw3BicpLWZlg3M0ZQz8HdmpfOERaOU4tWjJaeFk3LXU4MDdJZrVZYnNZIn0.eyJleHAiOjE3ODAxMTQwMDIsImh0dCI6MTc4MDUxMzcxMl0' \
  http://localhost:8888/ga4gh/drs/v1/service-info
```

Request URL

http://localhost:8888/ga4gh/drs/v1/service-info

Server response

Code

Details

200

Response body

```
{
  "contactUrl": "https://irods.org/contact/",
  "createdAt": "2026-06-03T19:10:21Z",
  "description": "Local DRS service metadata for the iRODS grid stack.",
  "documentationUrl": "https://github.com/michael-conway/irods-go-drs",
  "drs": {
    "maxBulkRequestLength": 1000,
    "objectCount": 0,
    "totalObjectSize": 0
  },
  "environment": "dev",
  "id": "org.irods.grid-stack-drs",
  "maxBulkRequestLength": 1000,
  "name": "iRODS Grid Stack DRS",
  "organization": {
    "name": "iRODS Grid Stack",
    "url": "https://irods.org/"
  },
  "type": {
    "artifact": "drs",
    "group": "org.ga4gh",
    "version": "1.5.0"
  },
  "updatedAt": "2026-06-03T19:10:21Z",
  "version": "1.5.0"
}
```



Download

Response headers

```
content-length: 584
content-type: application/json; charset=UTF-8
date: Wed, 03 Jun 2026 19:10:43 GMT
```

Returns information about the DRS service along with stats pertaining to total object count and cumulative size in bytes.

Extends the v1.0.0 GA4GH Service Info specification as the standardized format for GA4GH web services to self-describe.

drscmd for DRS object administration

```
conwaymc@adimi02012643 temp % gocmd ls
TYPE      PATH
collection /tempZone/home/test1/k-mers_sequences/SARS-CoV-2 Data

TYPE      NAME      SIZE      OWNER  MODIFY TIME
data-object RawData.mat    14423897 test1  2026-06-04.11:03
data-object RawData.xlsx 14245491 test1  2026-06-04.11:03
data-object RawDataTable.mat 14423822 test1  2026-06-04.11:03
collection Images      test1  2026-06-04.11:03
collection kmers        test1  2026-06-04.11:03
collection kmers2D       test1  2026-06-04.11:03

conwaymc@adimi02012643 temp % drscmd drsmake -d "demo1" RawData.mat
{
  "drsId": "a2dde654-9263-4e1c-90c3-d9b218697e9c",
  "path": "/tempZone/home/test1/k-mers_sequences/SARS-CoV-2 Data/RawData.mat"
}
conwaymc@adimi02012643 temp %
```

drsmake creates a new DRS object. This is a lightweight op that decorates an object with AVU metadata.

Note the opaque DRS id that is assigned.

Starbase
iRODS Grid Stack

Status: OK

RawData.mat

← Back to explorer

test1 / k-mers_sequences / SARS-CoV-2 Data / RawData.mat

TEMPZONE ☆ Command hints

test1/k-mers_sequences/RawData.mat

SIZE: 13.8 MB

RESOURCE: providerResc

Binary file
Preview unavailable

Actions

- Download object
- Replace object
- Rename file
- Move file
- Copy file
- Delete file

Delete will return to the explorer root.

Overview Storage AVUs Permissions Tickets

AVU metadata + Add AVU

Attribute	Value	Unit	Actions
iRODS:DRS:ID	a2dde654-9263-4e1c-90c3-d9b218697e9c	iRODS:DRS	🗑️
iRODS:DRS:VERSION	kldhpUqA5qPxIZlIK4n6qj3awCgxPpLavT84OCuKqfs=	iRODS:DRS	🗑️
iRODS:DRS:DESCRIPTION	demo1	iRODS:DRS	🗑️

Resolve an atomic DRS object

A single iRODS data object becomes a standard DRS object with access methods.

```
"access_methods": [  
  {  
    "type": "https",  
    "access_id": "irods-go-rest-https-providerResc",  
    "cloud": "irods:tempZone",  
    "region": "providerResc",  
    "authorizations": {  
      "supported_types": [  
        "BasicAuth",  
        "BearerAuth"  
      ],  
      "bearer_auth_issuers": [  
        "https://keycloak:8443"  
      ]  
    },  
  },  
],  
},
```

Note the https access method has affinity with the iRODS providerResc

REST
provider/resource

S3 provider/resource

iRODS provider
providerResc

GET /objects/{object_id} Get info about a DrsObject.

Returns object metadata, and a list of access methods that can be used to fetch object bytes. For collection-backed compound DRS objects in `irods-go-drs`, the returned `https` access method uses a direct `access_url` that points to `/ga4gh/drs/v1/ext/compound/{object_id}` and does not require an `access_id` follow-up request. Example (compound object access method shape): `{ "type": "https", "access_url": { "url": "https://<drs-host>/ga4gh/drs/v1/ext/compound/<object_id>" } }`

Parameters

Name	Description
object_id * required string (path)	DrsObject identifier a2dde654-9263-4e1c-90c3-d9b218697e9c
expand boolean (query)	If false and the object_id refers to a bundle, then the ContentsObject array contains only those objects directly contained in the bundle. That is, if the bundle contains other bundles, those other bundles are not recursively included in the result. If true and the object_id refers to a bundle, then the entire set of objects in the bundle is expanded. That is, if the bundle contains other bundles, then those other bundles are recursively expanded and included in the result. Recursion continues through the entire sub-tree of the bundle. If the object_id refers to a blob, then the query parameter is ignored. false

Execute Clear

Responses

Curl

```
curl -X 'GET' \  
  'http://localhost:8888/ga4gh/drs/v1/objects/a2dde654-9263-4e1c-90c3-d9b218697e9c?expand=false' \  
  -H 'accept: application/json' \  
  -H 'Authorization: Basic dGVzdDE6dGVzdA=='
```

Request URL

```
http://localhost:8888/ga4gh/drs/v1/objects/a2dde654-9263-4e1c-90c3-d9b218697e9c?expand=false
```

Server response

Code	Details
200	Response body <pre>{ "id": "a2dde654-9263-4e1c-90c3-d9b218697e9c", "name": "/tempZone/home/test1/k-mers_sequences/SARS-CoV-2_Data/RowData.mat", "self_url": "drs://localhost:8888/a2dde654-9263-4e1c-90c3-d9b218697e9c", "size": 14423897, "created_time": "2026-06-04T15:03:18Z", "updated_time": "2026-06-04T15:03:19Z", "version": "kLdhpUqASpXlZlIK4n6qJ3wCgxPpLavT840CuKqfs=", "checksums": [{ "checksum": "908761a54a80e6a3f1959225b89faa3ddac028313e92dabd3f38382b8aa9fb", "type": "sha-256" }], "access_methods": [{ "type": "https", "access_id": "irods-go-rest-https-providerResc", "cloud": "irods:tempZone", "region": "providerResc", "authorizations": { "supported_types": ["BasicAuth", "BearerAuth"] } }] }</pre> Response headers <pre>content-length: 913 content-type: application/json; charset=UTF-8 date: Thu, 04 Jun 2026 15:46:10 GMT</pre>

DRS access method location affinity

Access methods can reflect where the data lives in the iRODS resource topology.

GET

/objects/{object_id}/access/{access_id}

Get a URL for fetching bytes

🔒

⬆

Returns a URL that can be used to fetch the bytes of a **DrsObject**. This method only needs to be called when using an **AccessMethod** that contains an **access_id** (e.g., for servers that use signed URLs for fetching object bytes).

Parameters

Cancel

Name	Description
object_id *required	DrsObject identifier
string (path)	a2dde654-9263-4e1c-90c3-d9b218697e9c
access_id *required	An access_id from the access_methods list of a DrsObject
string (path)	irods-go-rest-https-providerResc

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8888/ga4gh/drs/v1/objects/a2dde654-9263-4e1c-90c3-d9b218697e9c/access/irods-go-rest-https-providerResc' \
-H 'accept: application/json' \
-H 'Authorization: Basic dGVzdDE6dGVzdA=='
```

Request URL

```
http://localhost:8888/ga4gh/drs/v1/objects/a2dde654-9263-4e1c-90c3-d9b218697e9c/access/irods-go-rest-https-providerResc
```

Server response

Code	Details
200	Response body <pre>{ "url": "http://localhost:8888/api/v1/path/contents?irods_path=%2FtempZone%2Fhome%2Ftest1%2Fk-mers_sequences%2FSARS-CoV-2+Data%2FrawData.mat", "headers": ["Authorization: Bearer irods-ticket:ticket_d8gq4ls0qvs73bqq3g0"] }</pre> Download Response headers <pre>content-length: 219 content-type: application/json; charset=UTF-8 date: Thu, 04 Jun 2026 16:06:02 GMT</pre>

DRS id plus the selected access method results in a resolvable reference. In this case an https:// reference with a generated iRODS ticket.

The ticket is a valid bearer token in the REST endpoint.

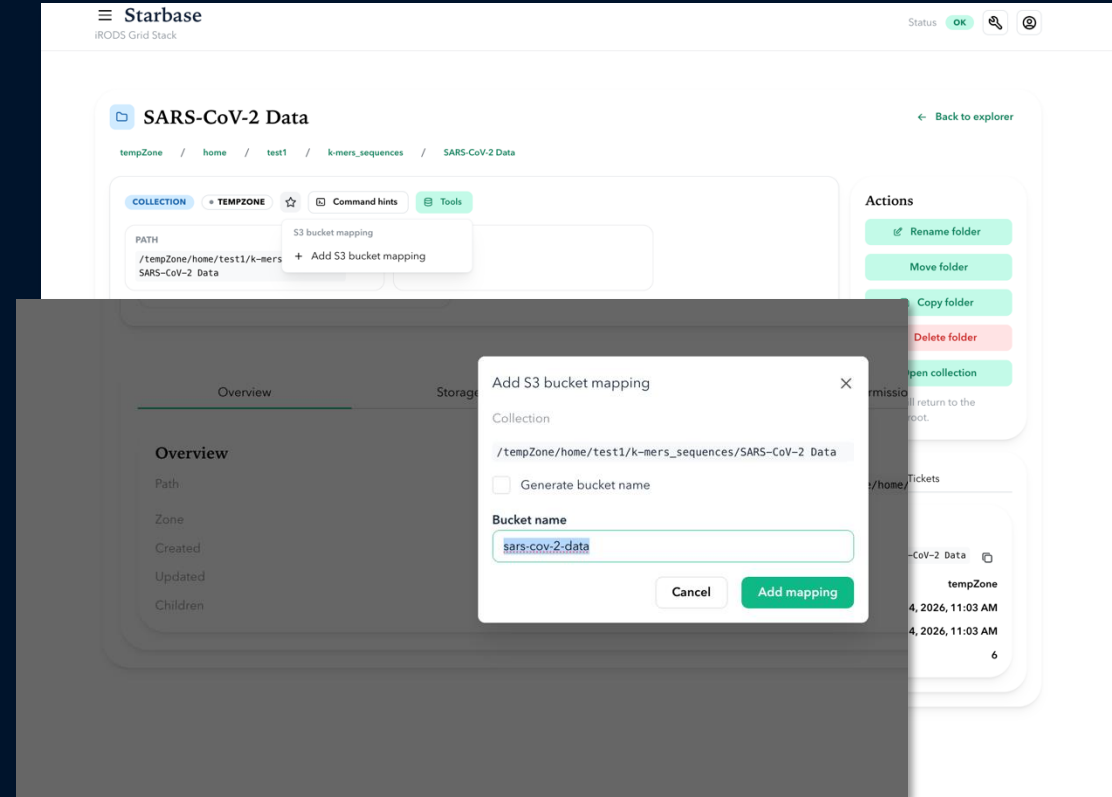
As replicas appear on different iRODS resources, DRS can expose access methods aligned with those resource locations.

Other access methods: focus on S3

S3 access matters because many workflow and cloud-adjacent tools already know how to consume it.

```
{
  "type": "s3",
  "access_url": {
    "url": "s3://sars-cov-2-data/RawData.mat"
  },
  "cloud": "irods:tempZone",
  "region": "providerResc",
  "available": true
}

{
  "type": "s3",
  "access_url": {
    "url": "s3://sars-cov-2-data/RawData.mat"
  },
  "region": "resourceResc"
}
```



Compound DRS objects

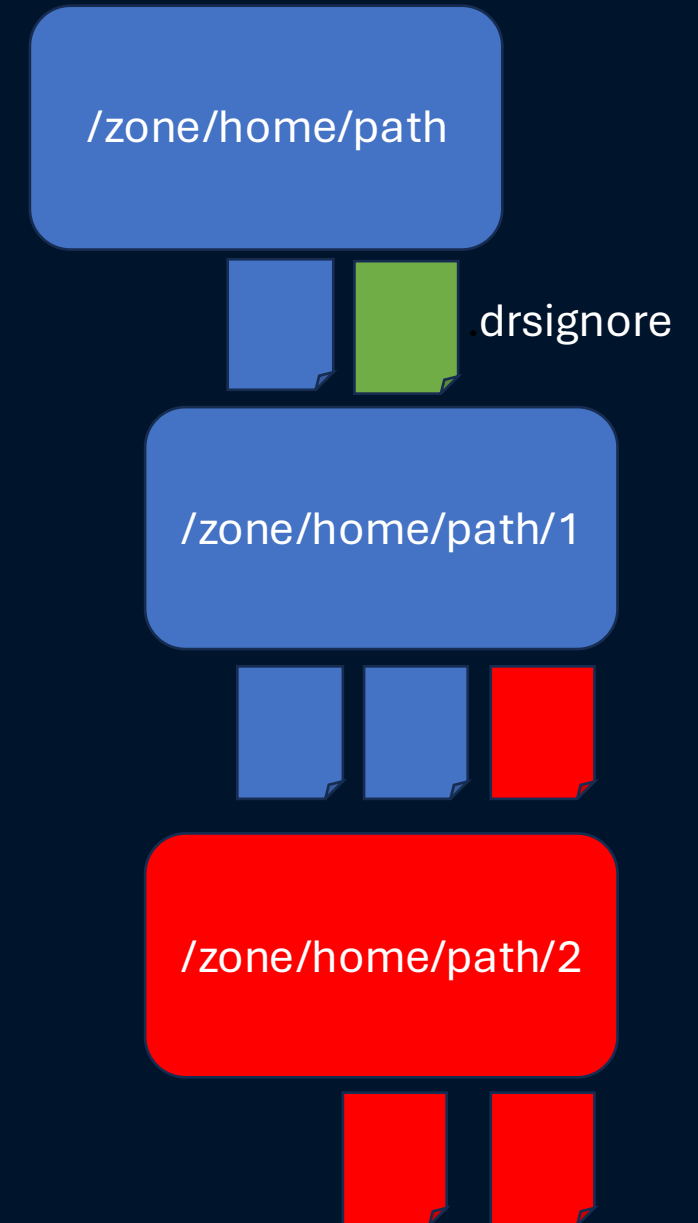
iRODS collections can represent related data packages, not just individual files.

EXAMPLES

BAM + BAI genomic read sets
DICOM images with multiple raw image objects
Study bundles with contents generated together
Workflow input or result packages

In this DRS implementation

- Compound objects are structured around iRODS Collections
- A .gitignore style .drsignore allows selection of exposed objects
- Compound objects have a DRS id and produce a JSON manifest with objects and additional metadata



Compound object workflow

.drsignore and preflight make compound exposure explicit and reviewable.

1. **.drsignore**

Select what is exposed

2. **Child metadata**

Add AVUs to member objects

3. **Preflight**

Show ignored paths + manifest preview

4. **Make compound**

Create the skeleton DRS object

5. **Edit metadata**

Manifest reflects current AVUs at access time

```
drsmakecompound --preflight /tempZone/home/demo/study-001  
  
# no data changes  
# shows ignored paths  
# previews generated manifest
```

Generated compound manifest

The manifest is generated dynamically from DRS AVUs and selected child metadata.

```
GET /ga4gh/drs/v1/ext/compound/{object_id}

{
  "id": "study-001",
  "contents": [
    { "name": "RawData.mat", "drs_id": "...", "path": "RawData.mat" },
    { "name": "metadata.json", "drs_id": "...", "path": "metadata.json" },
    { "name": "QC/report.html", "drs_id": "...", "path": "QC/report.html" }
  ],
  "metadata": { ... }
}
```

WHY IT MATTERS

Relates multiple iRODS objects as one logical unit
Makes collection-level DRS identity possible
Allows metadata edits without rebuilding a static package
Fits workflow inputs, studies, and result bundles

Next steps

iRODS as a data mesh node supporting federated analysis and discovery.

NEAR-TERM ARCHITECTURE

Integrate task running via TES/Funnel
Use affinity for resource-aware task execution
Connect DRS-aware genomic commons to
SLURM/local compute
Explore Data Connect / GenQuery discovery
Consider Mesh Card / service advertisement

DRS gives iRODS a standards-facing way to expose data access, data relationships, and data locality — the bridge from policy-managed data grids to federated analysis.