# Managing Data from the Edge to HPC

Jason Coposky
@jason_coposky
Executive Director, iRODS Consortium

January 30, 2019
Cloud Synchronization and Sharing Serivces
Rome, Italy

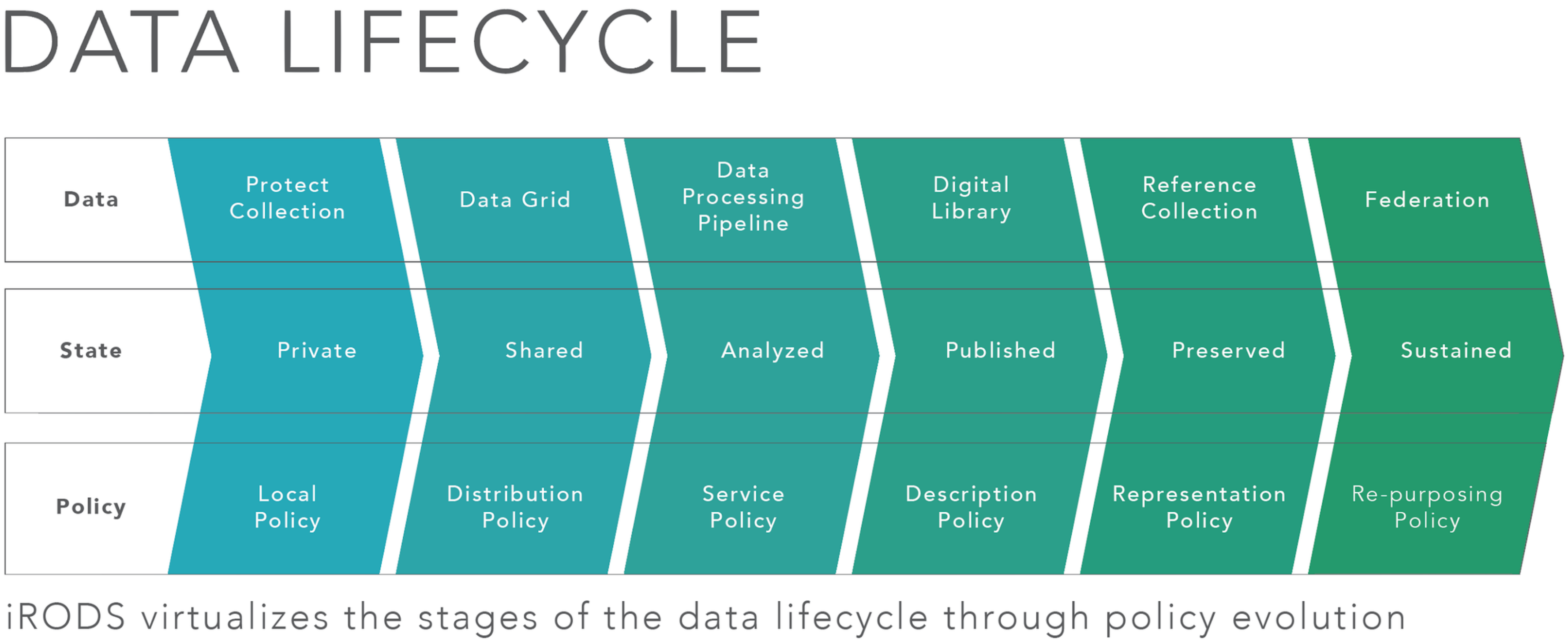
# Data Management

"The development, execution and supervision of plans, policies, programs and practices that control, protect, deliver and enhance the value of data and information assets."

Most organizations are still managing their assets with a collection of small scripts, tribal knowledge, vigilance, and hope.

Organizations, instead, need a future-proof solution to managing data and its surrounding infrastructure.

# Why Data Management Matters



## DATA LIFECYCLE

| | | | | | | |
|---|---|---|---|---|---|---|
| **Data** | Protect Collection | Data Grid | Data Processing Pipeline | Digital Library | Reference Collection | Federation |
| **State** | Private | Shared | Analyzed | Published | Preserved | Sustained |
| **Policy** | Local Policy | Distribution Policy | Service Policy | Description Policy | Representation Policy | Re-purposing Policy |

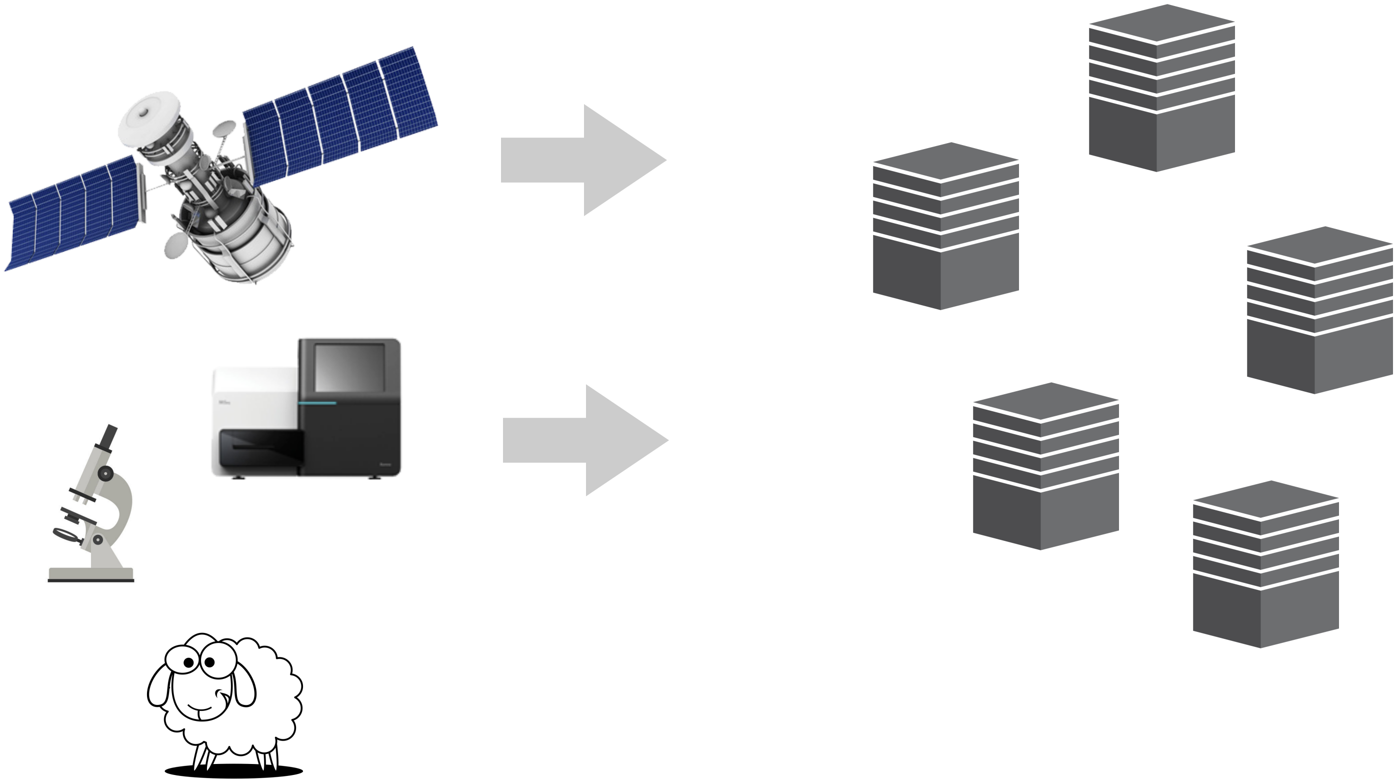
iRODS virtualizes the stages of the data lifecycle through policy evolution

As data matures and reaches a broader community, data management policy must also evolve to meet these additional requirements.

# Typical Data Flow

Devices / Sensors

On Premise / Cloud



Incoming source data from satellites, sequencers, microscopes, ... sheep

# The Problem

Data is coming in with greater...

- Volume
- Velocity
- Variety

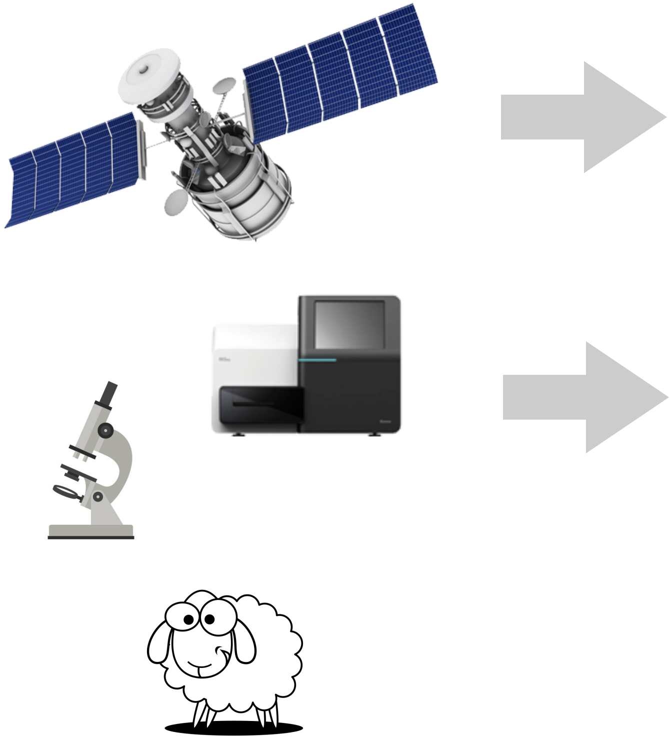
Human-throttled ingestion and cleaning is no longer sufficient.

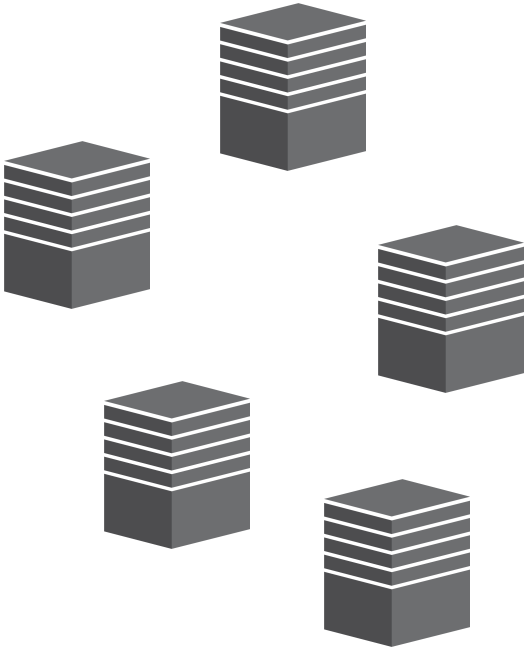
- Should be handled with policy and procedure
- Should be handled with code
- Should be handled closer to point of creation

# Where is the Edge?

Devices / Sensors

On Premise / Cloud





Where does the data come under management?

Where can it be vouched for?

Where can it be trusted?

# A Modest Proposal

iRODS is open source data management software

DATA VIRTUALIZATION

DATA DISCOVERY

WORKFLOW AUTOMATION

SECURE COLLABORATION

Provides insurance against your changing infrastructure:

- edge devices
- storage
- compute
- networking
- authentication

# iRODS Core Competencies

The underlying technology categorized into four areas

DATA VIRTUALIZATION

DATA DISCOVERY

WORKFLOW AUTOMATION

SECURE COLLABORATION

# iRODS Policy Examples

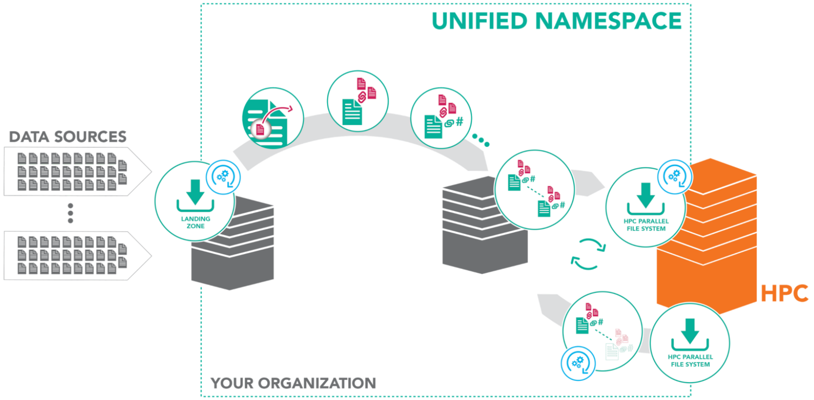
- Data Routing
- Data Movement
- Data Verification
- Data Synchronization
- Data Transformation
- Metadata Capture
- Metadata Application
- Metadata Verification

# iRODS Capabilities

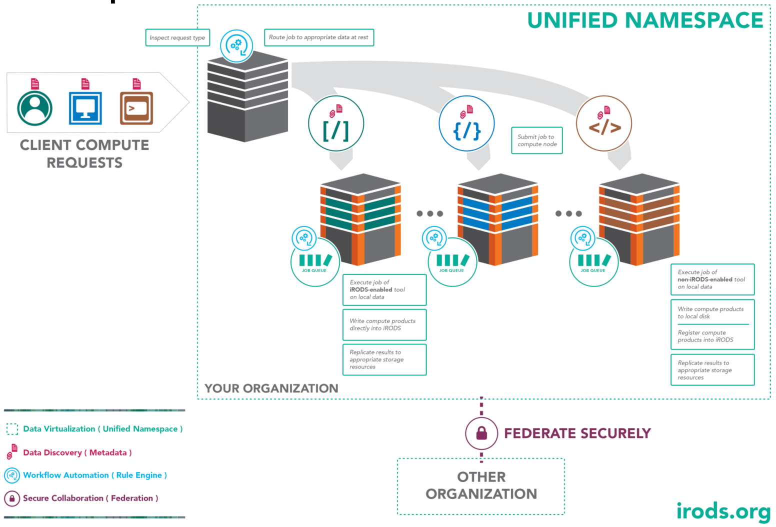
- Automated Ingest
- Auditing
- Indexing
- Data Integrity
- Storage Tiering
- Provenance
- Publishing
- Compliance

# Deployment Patterns

## Data to Compute



## Compute to Data


UNIFIED NAMESPACE
CLIENT COMPUTE REQUESTS
[/]
{/}
</>
JOB QUEUE
JOB QUEUE
JOB QUEUE
YOUR ORGANIZATION
FEDERATE SECURELY
OTHER ORGANIZATION
Data Virtualization ( Unified Namespace )
Data Discovery ( Metadata )
Workflow Automation ( Rule Engine )
Secure Collaboration ( Federation )

Inspect request type

Route job to appropriate data at rest

Submit job to compute node

Execute job of **iRODS-enabled** tool on local data

Write compute products directly into iRODS

Replicate results to appropriate storage resources

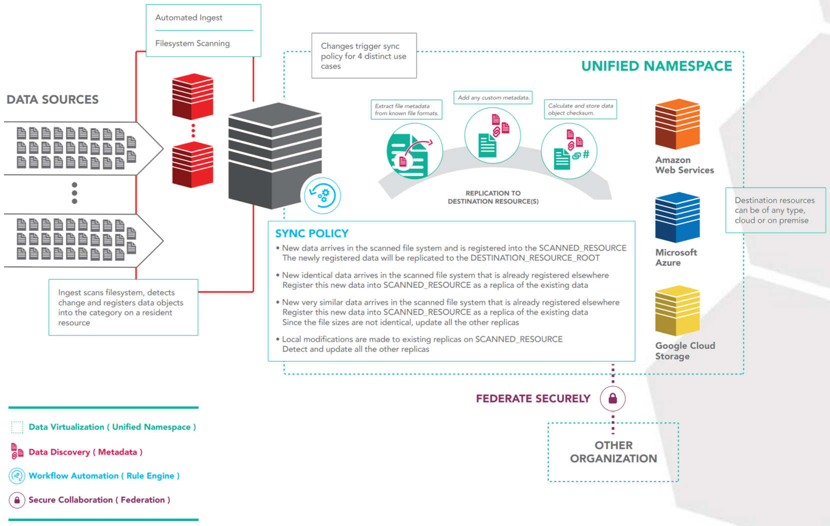
Execute job of **non-iRODS-enabled** tool on local data

Write compute products to local disk

Register compute products into iRODS

Replicate results to appropriate storage resources

## Filesystem Synchronization


DATA SOURCES
UNIFIED NAMESPACE
REPLICATION TO DESTINATION RESOURCE(S)
Amazon Web Services
Microsoft Azure
Google Cloud Storage
FEDERATE SECURELY
OTHER ORGANIZATION
Data Virtualization ( Unified Namespace )
Data Discovery ( Metadata )
Workflow Automation ( Rule Engine )
Secure Collaboration ( Federation )

Automated Ingest

Filesystem Scanning

Ingest scans filesystem, detects change and registers data objects into the category on a resident resource

Changes trigger sync policy for 4 distinct use cases

Extract file metadata from known file formats.

Add any custom metadata.

Calculate and store data object checksum.

Destination resources can be of any type, cloud or on premise

**SYNC POLICY**

- New data arrives in the scanned file system and is registered into the SCANNED_RESOURCE
  The newly registered data will be replicated to the DESTINATION_RESOURCE_ROOT
- New identical data arrives in the scanned file system that is already registered elsewhere
  Register this new data into SCANNED_RESOURCE as a replica of the existing data
- New very similar data arrives in the scanned file system that is already registered elsewhere
  Register this new data into SCANNED_RESOURCE as a replica of the existing data
  Since the file sizes are not identical, update all the other replicas
- Local modifications are made to existing replicas on SCANNED_RESOURCE
  Detect and update all the other replicas

irods.org

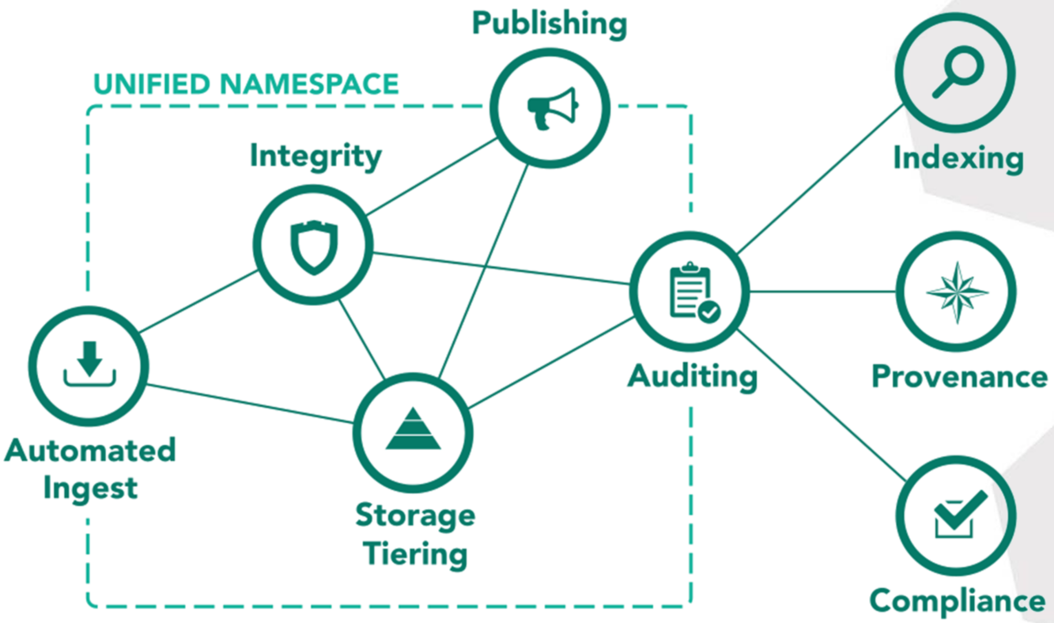
# The Data Management Model

iRODS provides eight packaged capabilities, each of which can be selectively deployed and configured.

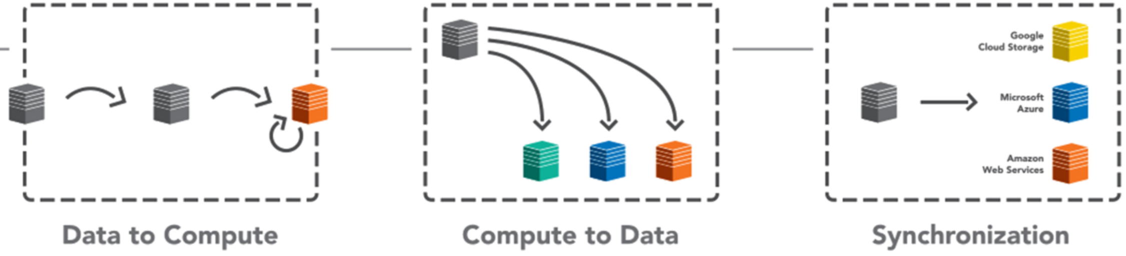
These capabilities represent the most common use cases as identified by community participation and reporting.

The flexibility provided by this model allows an organization to address its immediate use cases.

Additional capabilities may be deployed as any new requirements arise.



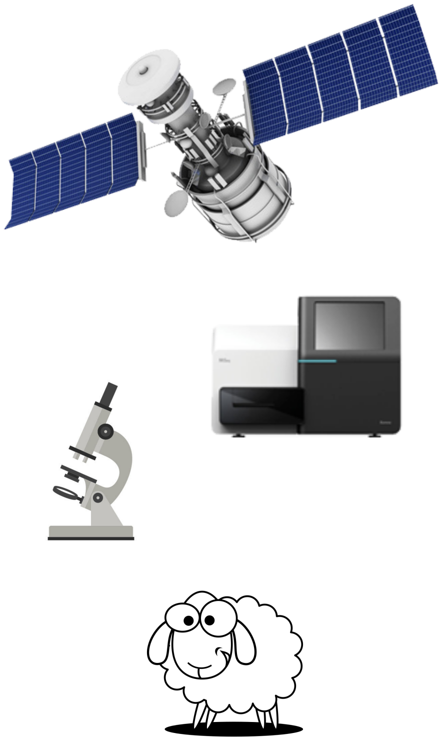
A pattern represents a combination of iRODS capabilities and data management policy consistent across multiple organizations. Three common patterns of iRODS deployment have been observed within the community:

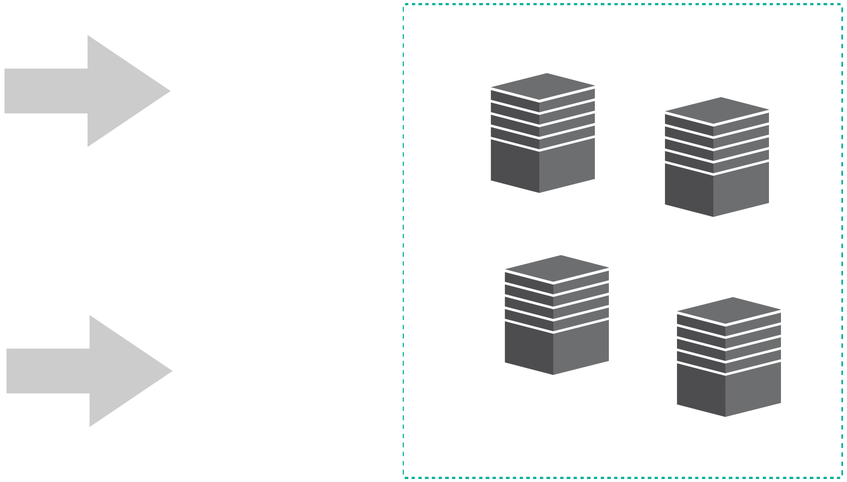


# Where is the Edge?

Devices / Sensors

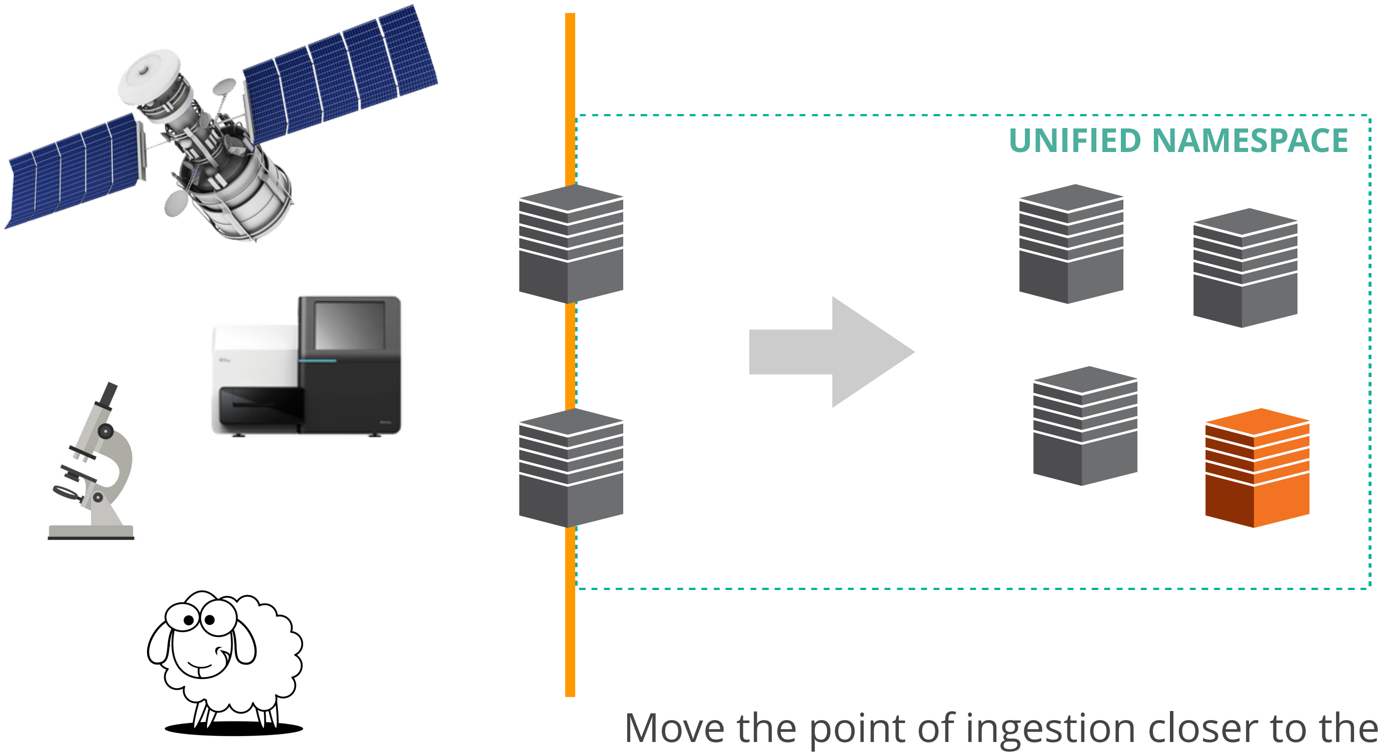
On Premise / Cloud





Create a logical namespace

# Where is the Edge?



Devices / Sensors

Edge

On Premise / Cloud

UNIFIED NAMESPACE

Move the point of ingestion closer to the source.

Ingest on site.  Ingest at the point of data creation.

# The Data Lifecycle begins at Data Generation

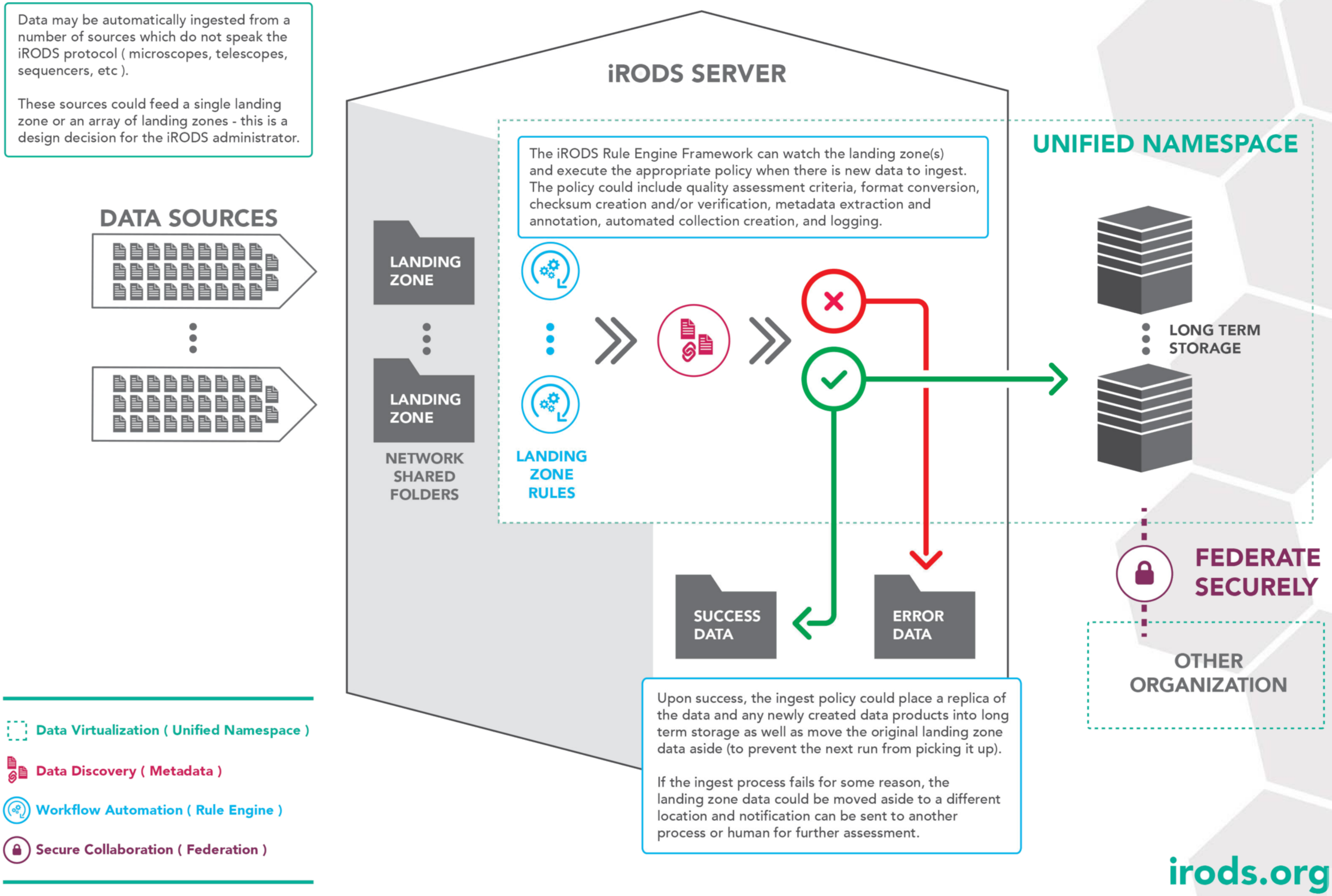
By bringing data management to the point of data generation

(and extending the programmatic surface out to the instruments),

a system with this architecture can address other hard problems:

- Data Harmonization
- Data Movement
- Data Integrity
- Geographic Distribution
- Network Capacity
- Network Reliability
- Variety of Data Sources
- Variety of Data Formats

# Automated Ingest - Landing Zone



Data may be automatically ingested from a number of sources which do not speak the iRODS protocol ( microscopes, telescopes, sequencers, etc ).

These sources could feed a single landing zone or an array of landing zones - this is a design decision for the iRODS administrator.

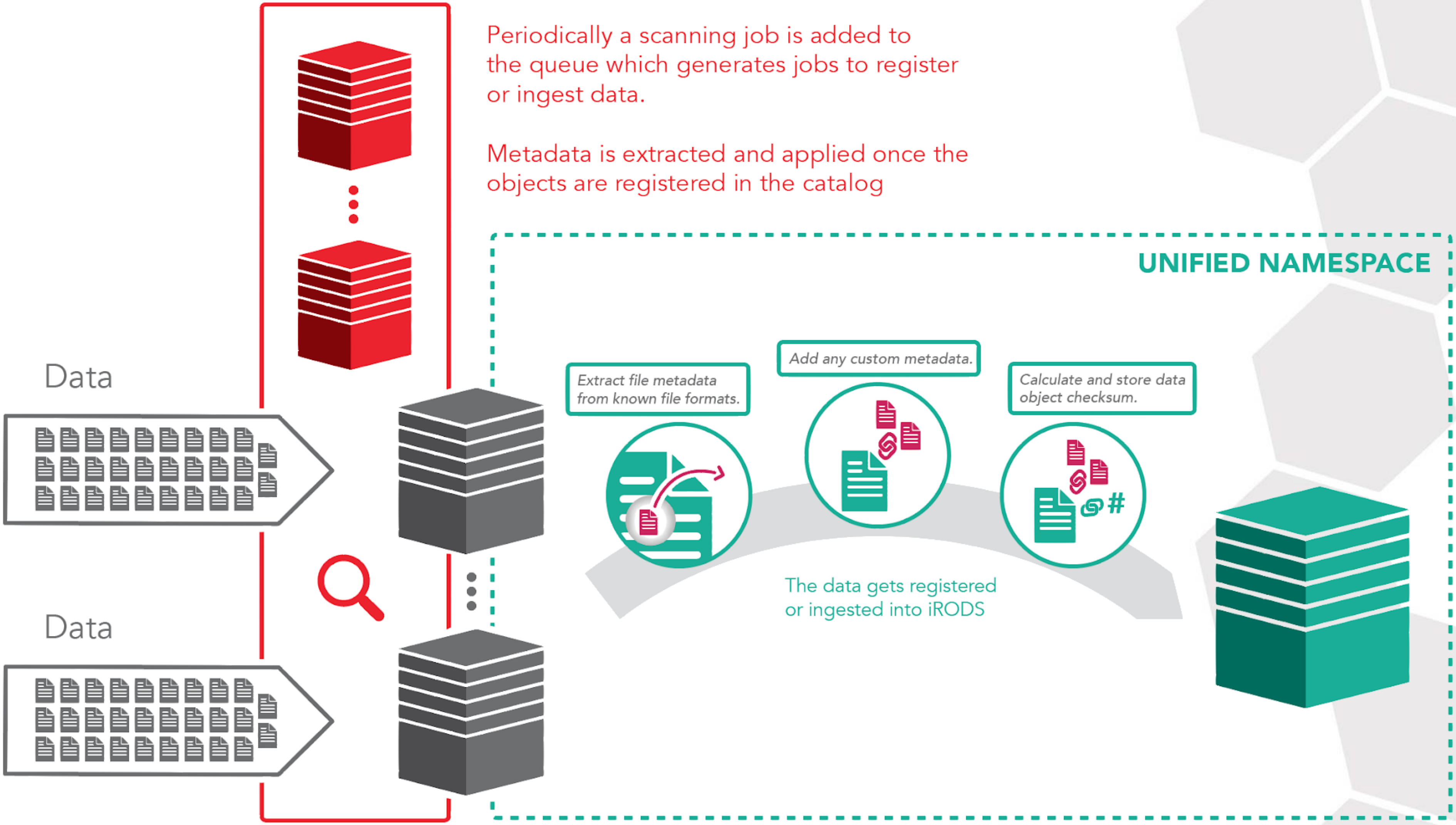
The iRODS Rule Engine Framework can watch the landing zone(s) and execute the appropriate policy when there is new data to ingest. The policy could include quality assessment criteria, format conversion, checksum creation and/or verification, metadata extraction and annotation, automated collection creation, and logging.

Upon success, the ingest policy could place a replica of the data and any newly created data products into long term storage as well as move the original landing zone data aside (to prevent the next run from picking it up).

If the ingest process fails for some reason, the landing zone data could be moved aside to a different location and notification can be sent to another process or human for further assessment.

- Data Virtualization ( Unified Namespace )
- Data Discovery ( Metadata )
- Workflow Automation ( Rule Engine )
- Secure Collaboration ( Federation )

# Automated Ingest - Filesystem Scanning



Periodically a scanning job is added to the queue which generates jobs to register or ingest data.

Metadata is extracted and applied once the objects are registered in the catalog

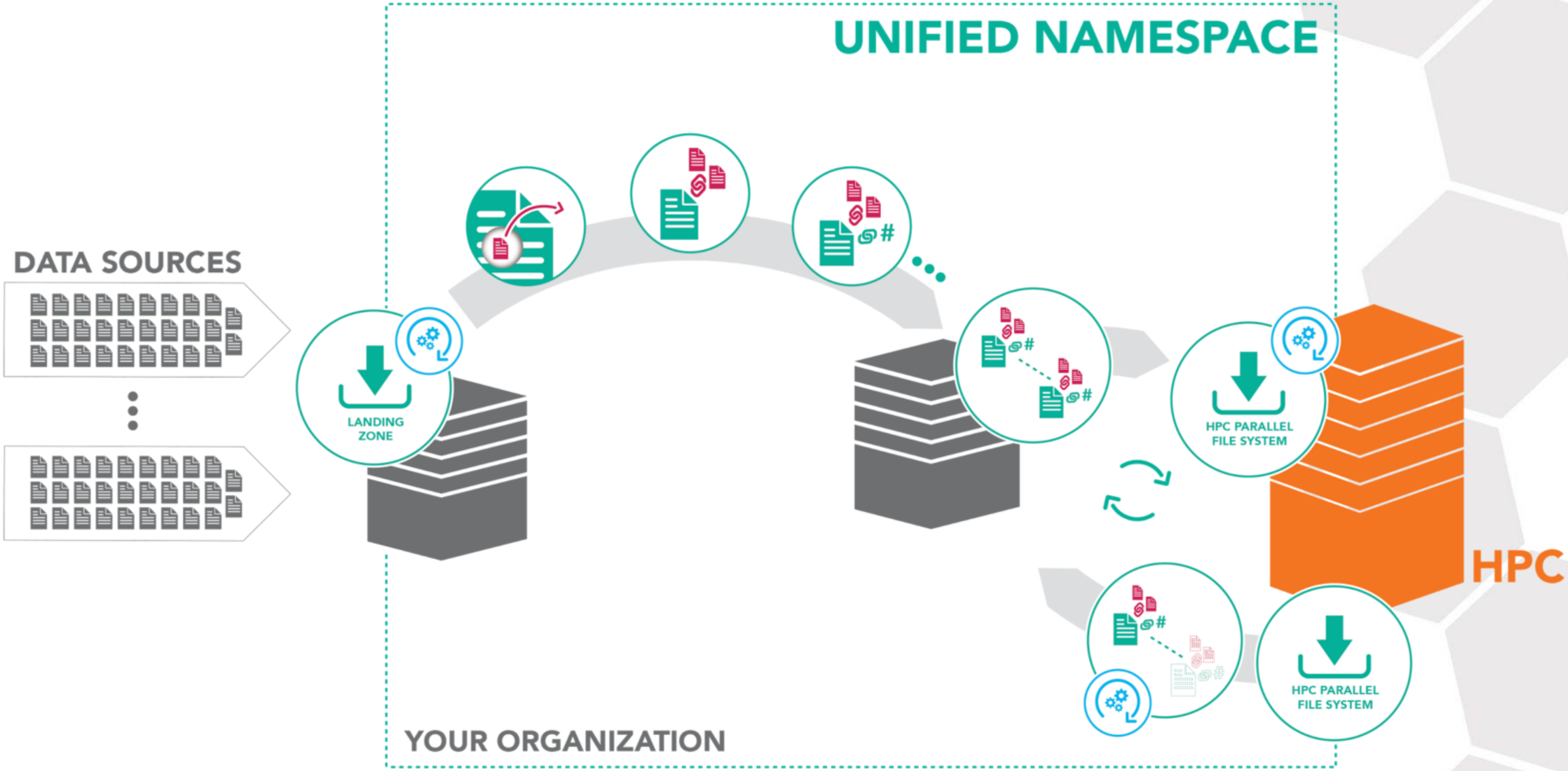
*Extract file metadata from known file formats.*

*Add any custom metadata.*

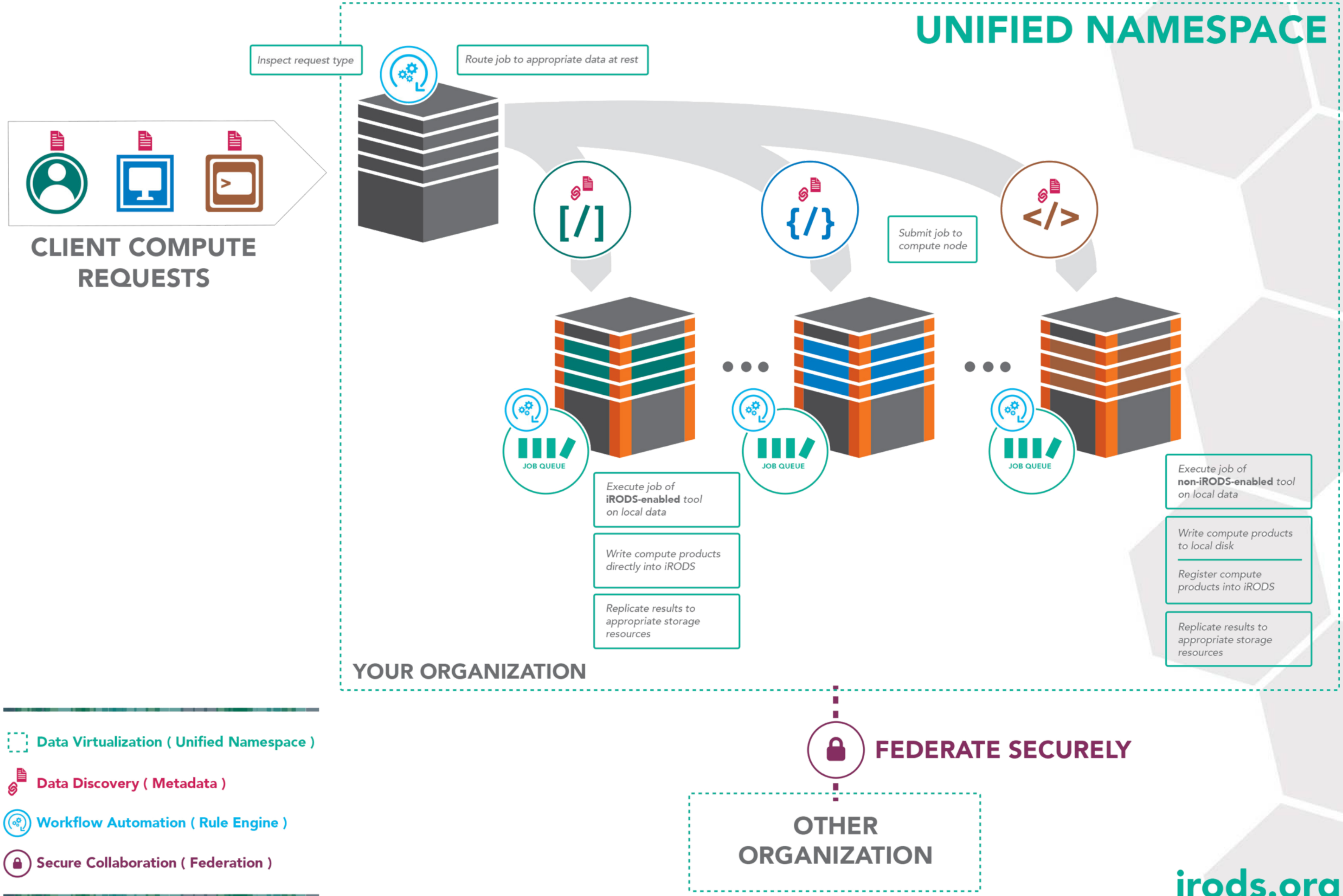
*Calculate and store data object checksum.*

The data gets registered or ingested into iRODS

# Data to Compute



# Compute to Data



Inspect request type

Route job to appropriate data at rest

Submit job to compute node

Execute job of **iRODS-enabled** tool on local data

Write compute products directly into iRODS

Replicate results to appropriate storage resources

Execute job of **non-iRODS-enabled** tool on local data

Write compute products to local disk

Register compute products into iRODS

Replicate results to appropriate storage resources

Data Virtualization ( Unified Namespace )

Data Discovery ( Metadata )

Workflow Automation ( Rule Engine )

Secure Collaboration ( Federation )

# Resources

iRODS Overview and Diagrams

https://irods.org/documentation

Official Documentation

https://docs.irods.org

iRODS Training Materials and Presentations

https://slides.com/irods

iRODS User Group

https://irods.org/ugm2019

# Questions?