Versity

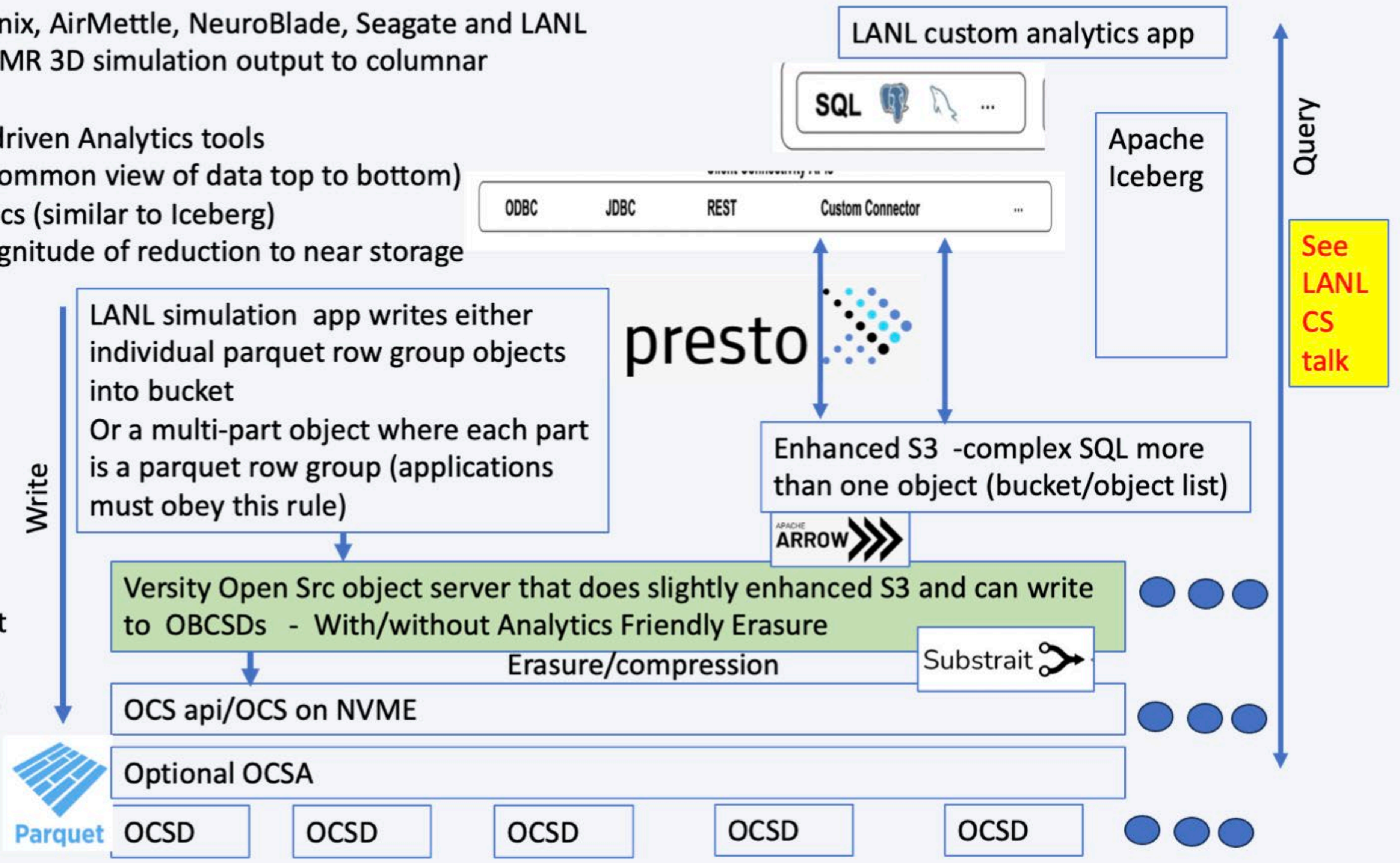
Ben McClelland (Versity)
Gary Grider (LANL)
SDC 2023

LANL use case of Versity Gateway - Object Computational Storage (OCS)

- Partnership: SK hynix, AirMettle, NeuroBlade, Seagate and LANL
- LANL grid-based AMR 3D simulation output to columnar Parquet
- Leverage Apache driven Analytics tools
- Leverage object (common view of data top to bottom)
- Extend S3 semantics (similar to Iceberg)
- Push orders of magnitude of reduction to near storage

- Big Data Analytics
    - You don't know what you are looking for, want compact model
    - You know what you are looking for but don't know where it is

OCS Value Prop


LANL simulation app writes either individual parquet row group objects into bucket
Or a multi-part object where each part is a parquet row group (applications must obey this rule)

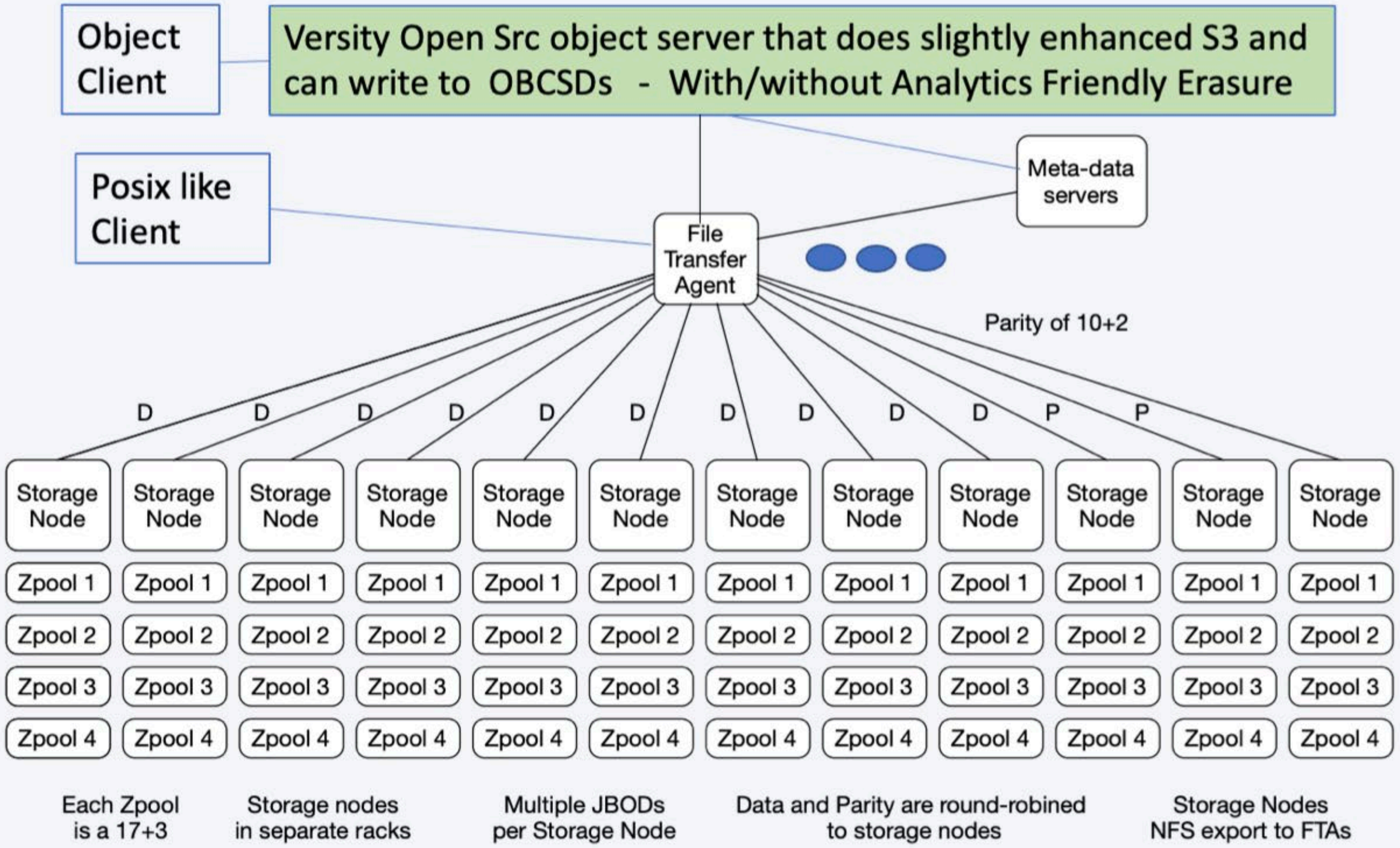
Enhanced S3 -complex SQL more than one object (bucket/object list)

Versity Open Src object server that does slightly enhanced S3 and can write to OBCSDs - With/without Analytics Friendly Erasure

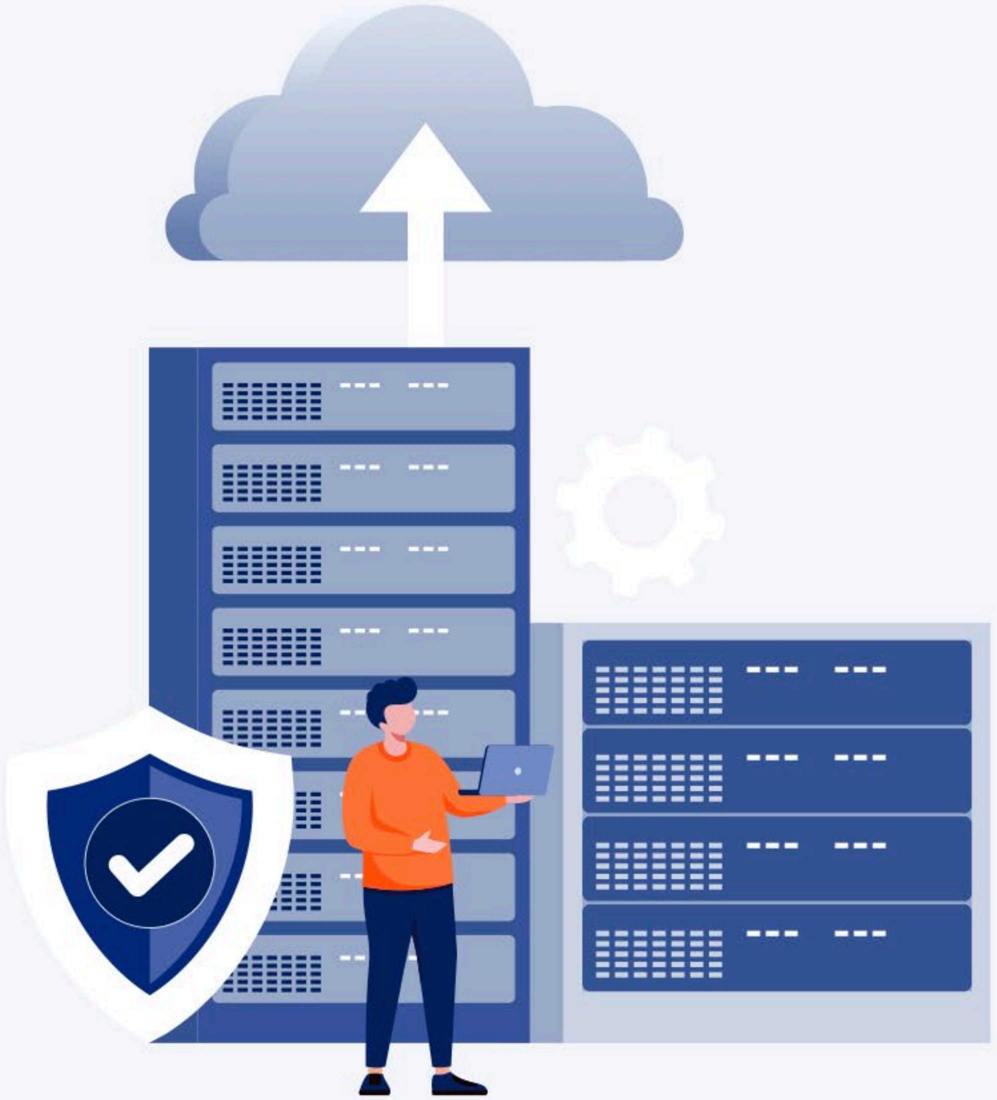
Versity

LANL use case of Versity Gateway Object Access to Science Campaign Data

- Leverage massive 2 tier erasured science campaign storage data lake from posix like users and from object users.
- Data is already in immutable files
- Hundreds of gigabyes/sec
- Stable for years
- Disk cool storage and even erasured tape in the future.


Object Client
Versity Open Src object server that does slightly enhanced S3 and can write to OBCSDs - With/without Analytics Friendly Erasure
Posix like Client
Meta-data servers
File Transfer Agent
Parity of 10+2
D
D
D
D
D
D
D
D
D
D
P
P
Storage Node
Zpool 1
Zpool 2
Zpool 3
Zpool 4
Each Zpool is a 17+3
Storage nodes in separate racks
Multiple JBODs per Storage Node
Data and Parity are round-robined to storage nodes
Storage Nodes NFS export to FTAs

Versity



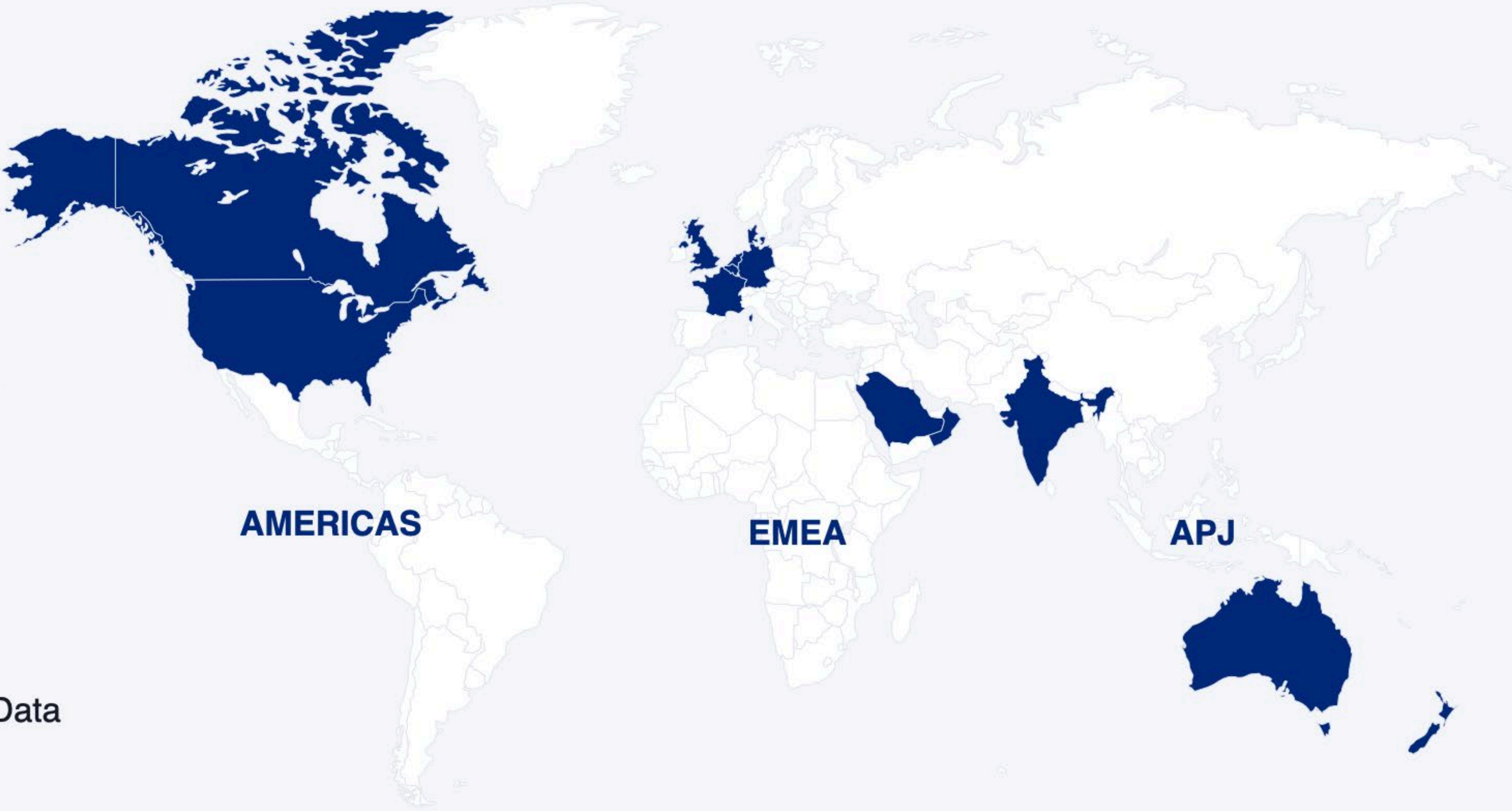
What We Do:

Manage Large Unstructured Data Collections at Low Cost

- Software-Defined Storage Platform
- Mass Storage & Large Archive

Versity

Global Installed Base



Total Data

2 Exabytes Data

80 Sites

Versity

Versity Gateway

S3 -> POSIX
Enables S3 workloads to utilize POSIX filesystems and tape via ScoutAM

Modular
POSIX and ScoutAM backends supported, easy to add more

Scalable
Load balance across multiple instances

High Performance
Written from scratch
Go / GoFiber
Fast processing and response times

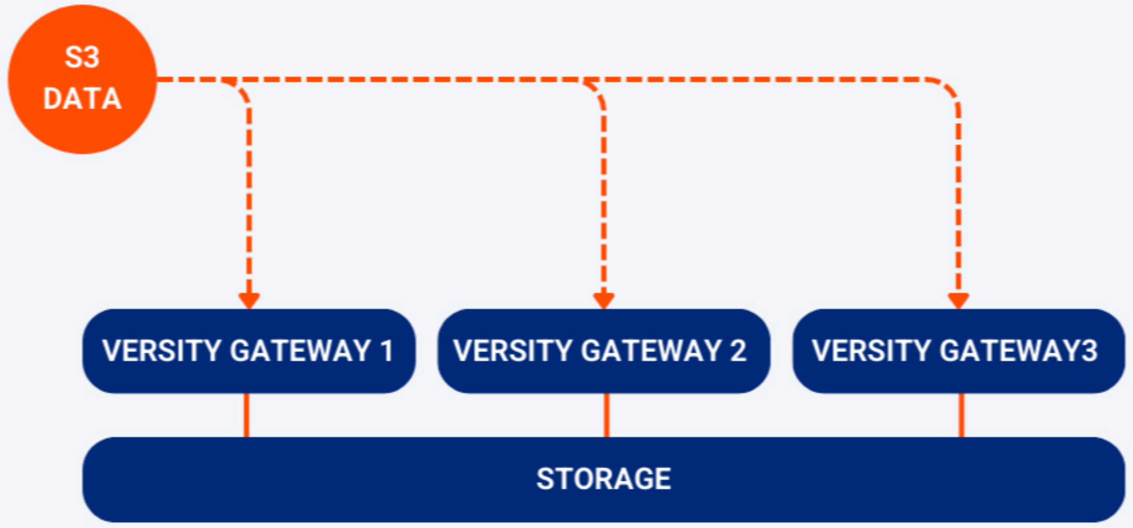
Open-Source
Apache2 licensed, available on GitHub

Seamless Integration
Familiar interfaces

Collaborative
Versity
Los Alamos
NATIONAL LABORATORY
pawsey

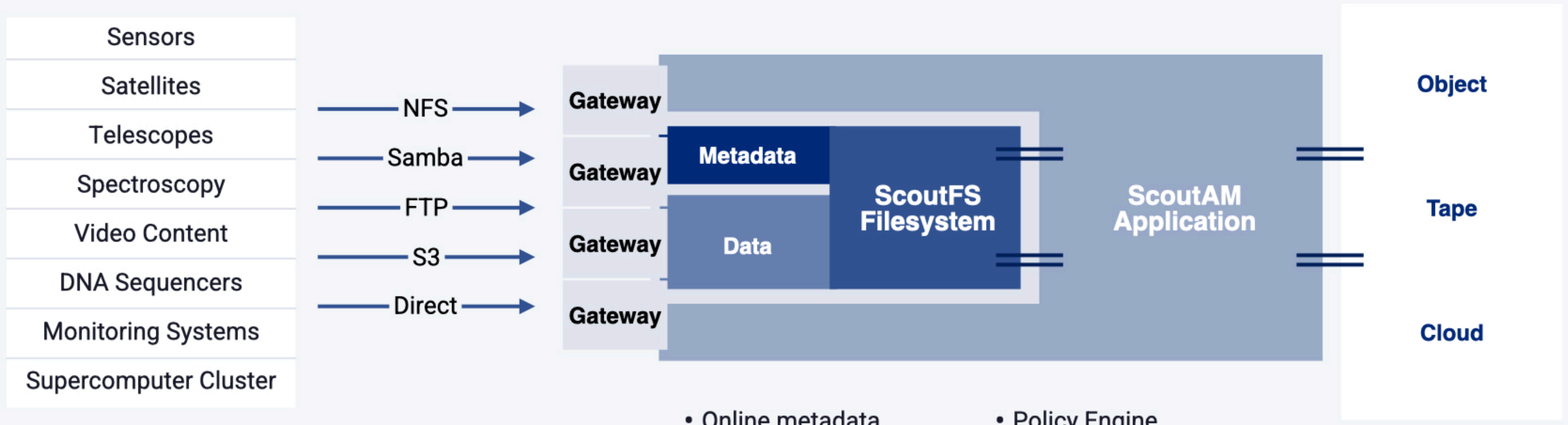
Versity Gateway

- Scalable S3 Service
- High Performance
- Modular Backend Support
- Flexible Open Source Licensing (Apache 2.0)



Scale-Out Archive Manager: ScoutAM

A Modern Data Management Platform

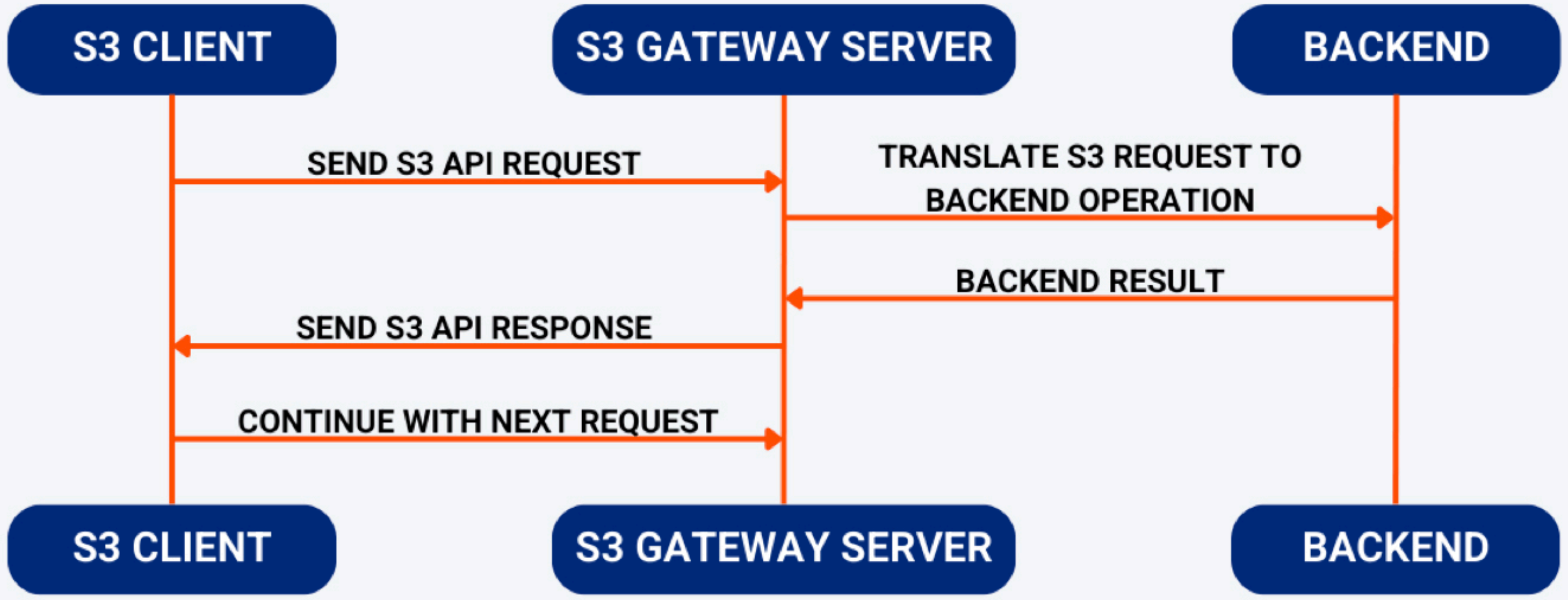


- Online metadata
- Indexed attributes
- Scalable namespace

- Policy Engine
- Scheduling
- Parallel Data Movement

Versity

Versity Gateway



Versity

Modular Backend

Current Backends:

- POSIX - filesystem with xattr support
- ScoutFS - (https://github.com/versity/scoutfs)

Open to collaboration on new backends

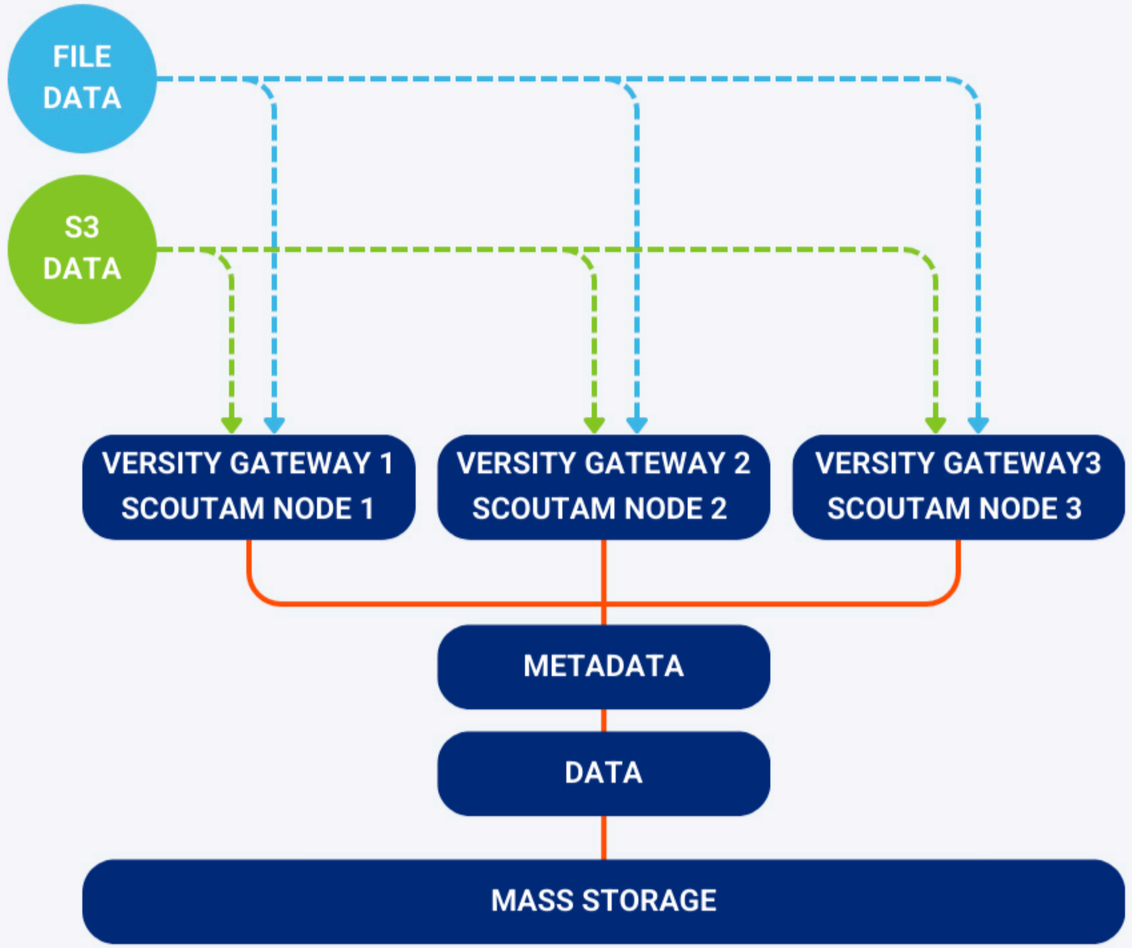
ScoutFS Optimized Backend

Optimized Writes

- Multi-part upload segments are written once to the underlying storage then
- Parts of data upload are combined into a single file with a system call.
- **Eliminates one full read/write cycle**
- Speeds up the process and reduces overall upload time.

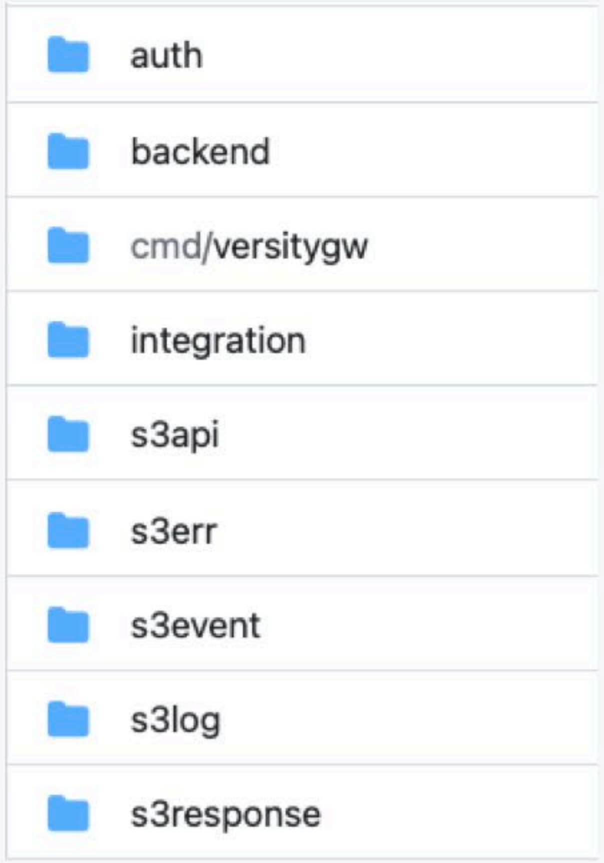
S3 Glacier Mode

- Cold storage feature for data archiving and long-term data retention.
- Data may be stored on local tape systems using Versity's ScoutAM platform.
- Manage storage costs more effectively by accessing low cost storage.

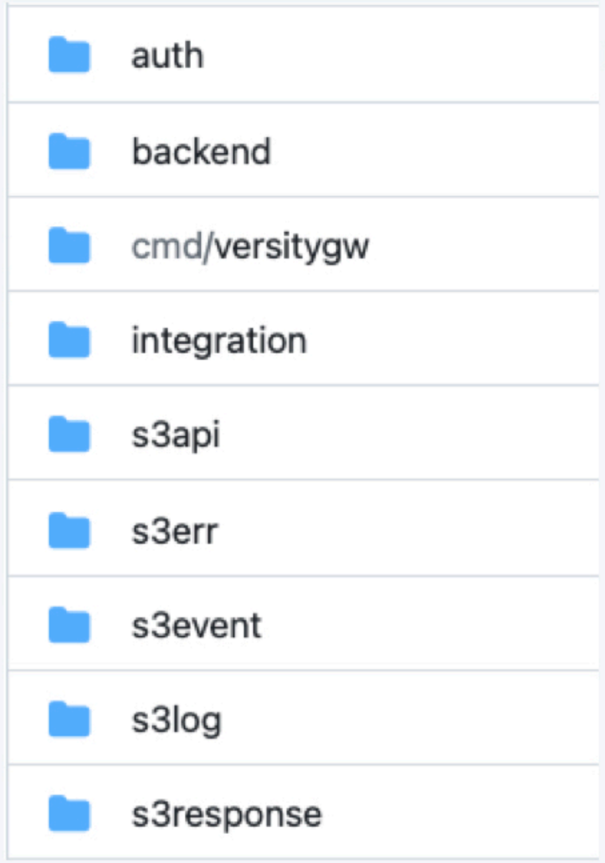

FILE DATA
S3 DATA
VERSITY GATEWAY 1 SCOUTAM NODE 1
VERSITY GATEWAY 2 SCOUTAM NODE 2
VERSITY GATEWAY3 SCOUTAM NODE 3
METADATA
DATA
MASS STORAGE

Versity

Project Layout


auth
backend
cmd/versitygw
integration
s3api
s3err
s3event
s3log
s3response

Project Layout


auth
backend
cmd/versitygw
integration
s3api
s3err
s3event
s3log
s3response

$\longrightarrow$

Multi-Tenant
IAM account integrations

Extensible to support new authentication systems

Currently Supported:
Local accounts

Versity

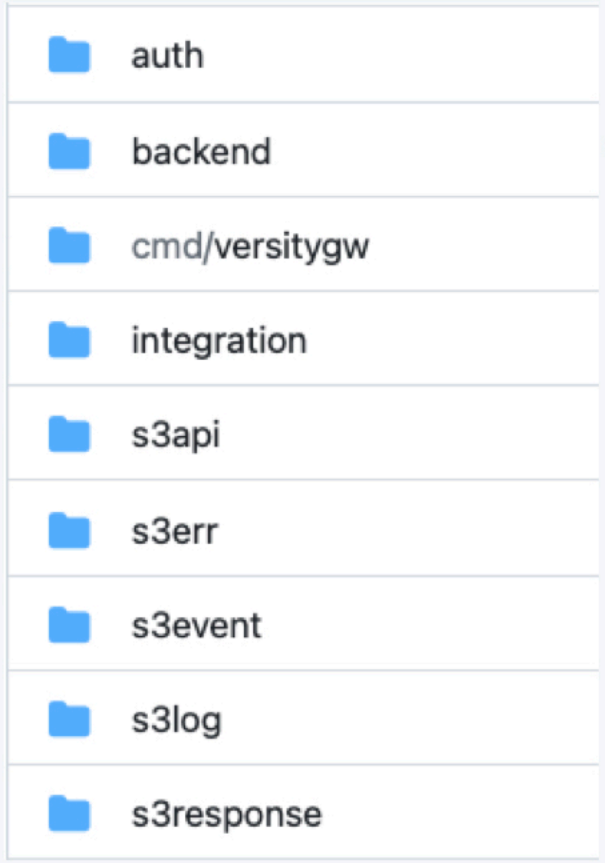
Multi-Tenant

root account defined with cli flags/env vars
admin/user accounts stored in IAM service

	See All Buckets	Create New Buckets	Create New Users	Assign Bucket Ownershi	See Only Owned Buckets
root	X	X	X	X	
admin		X	X	X	X
user					X

Versity

Project Layout



auth
backend
cmd/versitygw
integration
s3api
s3err
s3event
s3log
s3response

Storage Backends

Extensible to support new storage systems

Currently Supported:
POSIX
ScoutFS

Versity

Backend

- Embed backend.BackendUnsupported
- Implement backend.Backend methods as needed to satisfy interface
- Any unimplemented will return s3err.ErrNotImplemented back to client

```
package mystorage

type MyStorage struct {
    backend.BackendUnsupported
}
```

```
type Backend interface {
    // bucket operations
    ListBuckets(_ context.Context, owner string, isRoot bool) (s3response.ListAllMyBucketsResult, error)
  …

    // multipart operations
    CreateMultipartUpload(context.Context, *s3.CreateMultipartUploadInput) (*s3.CreateMultipartUploadOutput, error)
  …

    // standard object operations
    PutObject(context.Context, *s3.PutObjectInput) (string, error)
  …

    // special case object operations
    RestoreObject(context.Context, *s3.RestoreObjectInput) error
  …

    // object tags operations
    GetTags(_ context.Context, bucket, object string) (map[string]string, error)
  …
}
```

Versity

ScoutFS/Glacier Example

ScoutFS/ScoutAM - Archiving Filesystem

- POSIX filesystem
- Supports offline files where data only resides on tape/mass storage
- Automatic recall when data requested

When in Glacier Emulation Mode,
scoutfs backend enables following:

HEAD offline object returns
storage-class: GLACIER
x-amz-restore: (transition state)

GET of offline file returns
Invalid Object State

Restore Object
triggers backend data recall

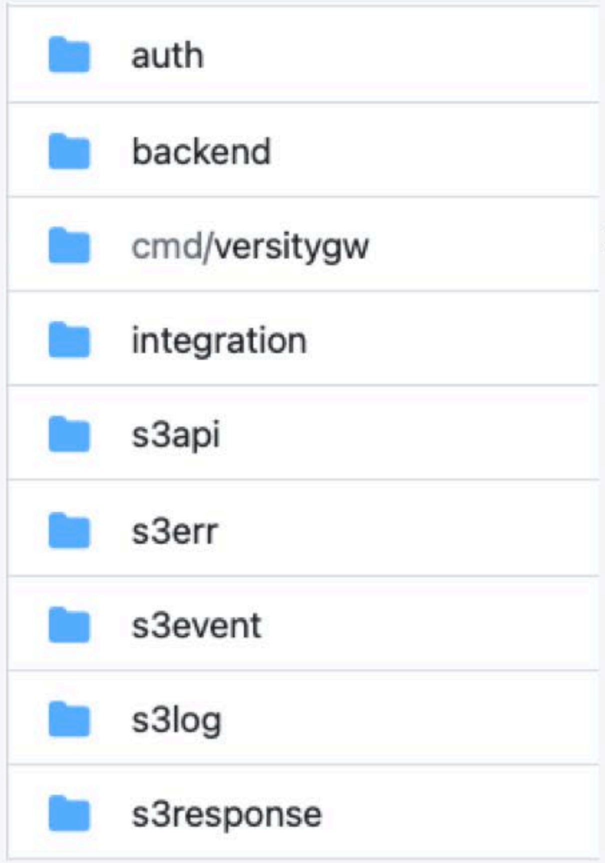
```
type Backend interface {
  ...

    // special case object operations
    RestoreObject(context.Context, *s3.RestoreObjectInput) error
    ...

}
```

Versity

Project Layout

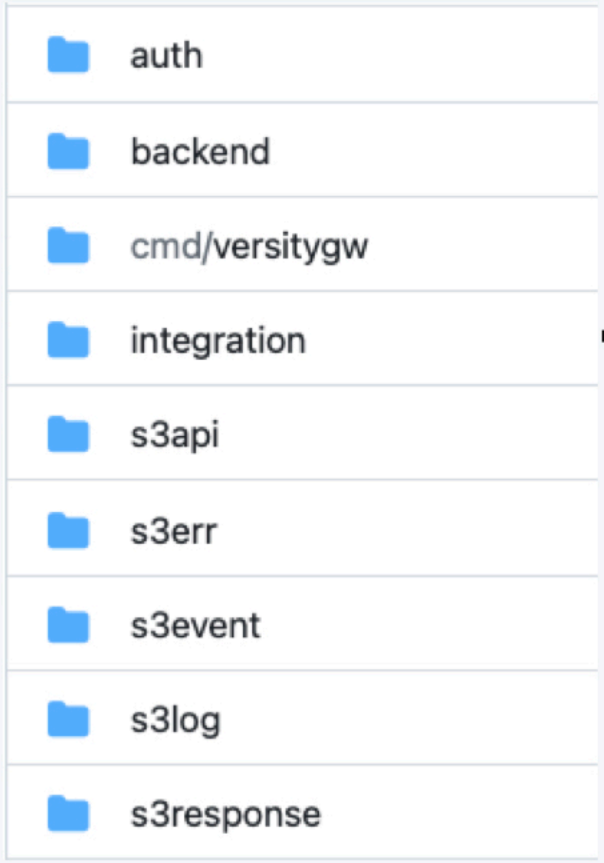


- auth
- backend
- cmd/versitygw
- integration
- s3api
- s3err
- s3event
- s3log
- s3response

→ CLI command support

Versity

Project Layout



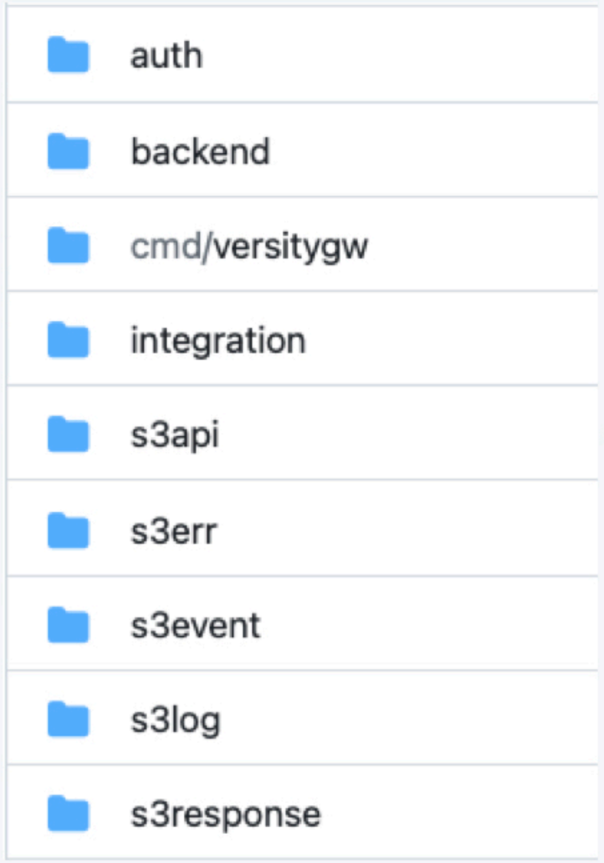
auth
backend
cmd/versitygw
integration
s3api
s3err
s3event
s3log
s3response

→ Integration Tests

GitHub CI runs test suite against POSIX for PRs

Manually run test suite against any storage backend

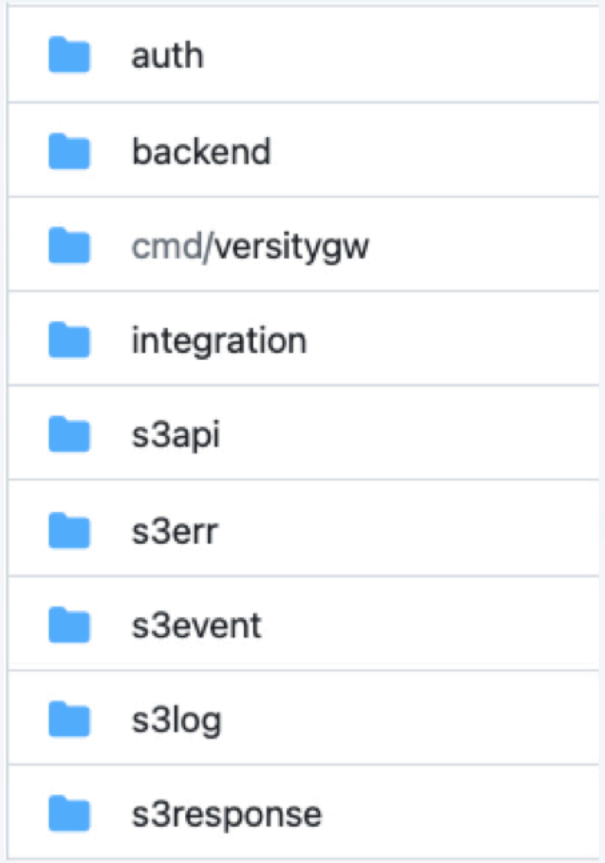
Project Layout


auth
backend
cmd/versitygw
integration
s3api
s3err
s3event
s3log
s3response

→ Frontend API Handlers

Go Fiber framework for HTTP(s) handlers
Supporting S3 and Admin API

Project Layout


auth
backend
cmd/versitygw
integration
s3api
s3err
s3event
s3log
s3response

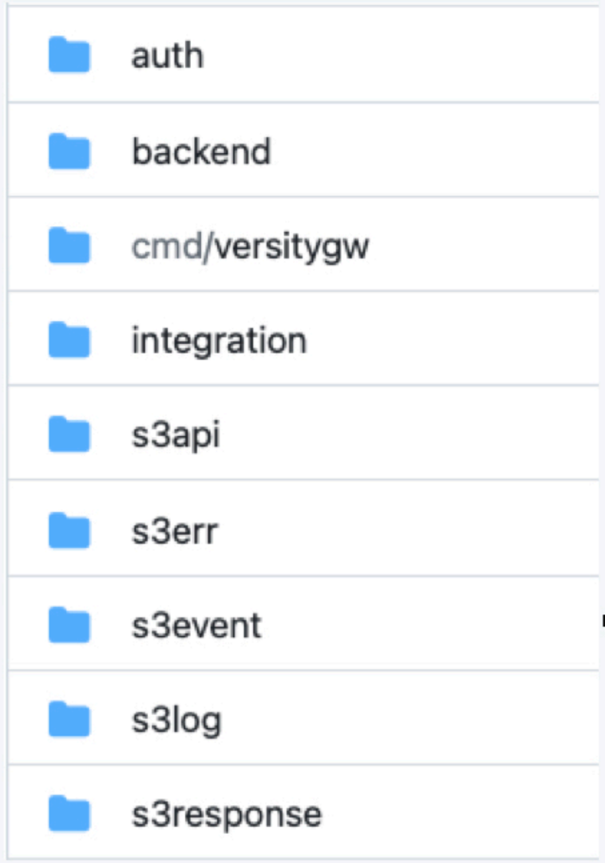
S3 errors contains all errors that clients would expect from AWS S3

Non-s3err types returned from backend treated as Internal Server Error

→ S3 error response types

Versity

Project Layout


auth
backend
cmd/versitygw
integration
s3api
s3err
s3event
s3log
s3response

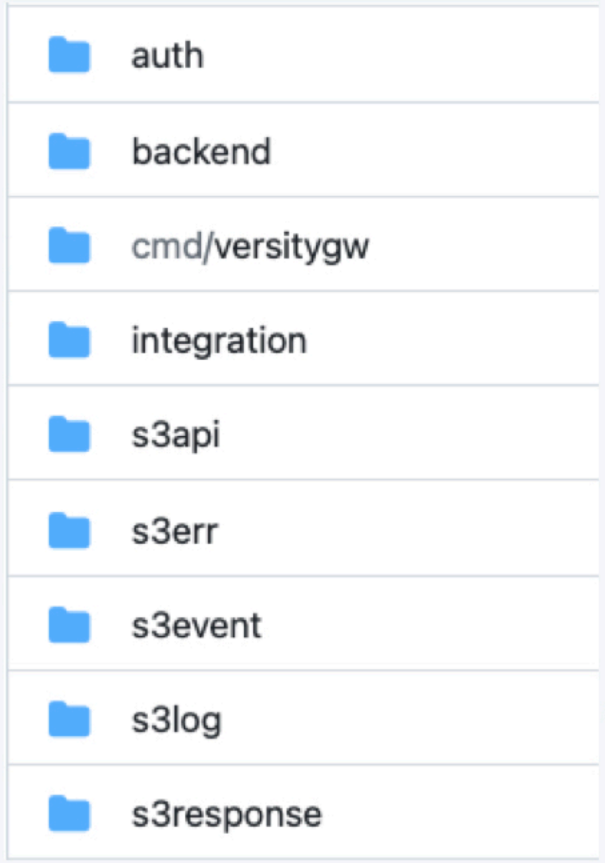
Same event structure as AWS

Extensible event integrations, currently supported:

- NATS
- Kafka

S3 Event Notifications

Project Layout



auth
backend
cmd/versitygw
integration
s3api
s3err
s3event
s3log
s3response

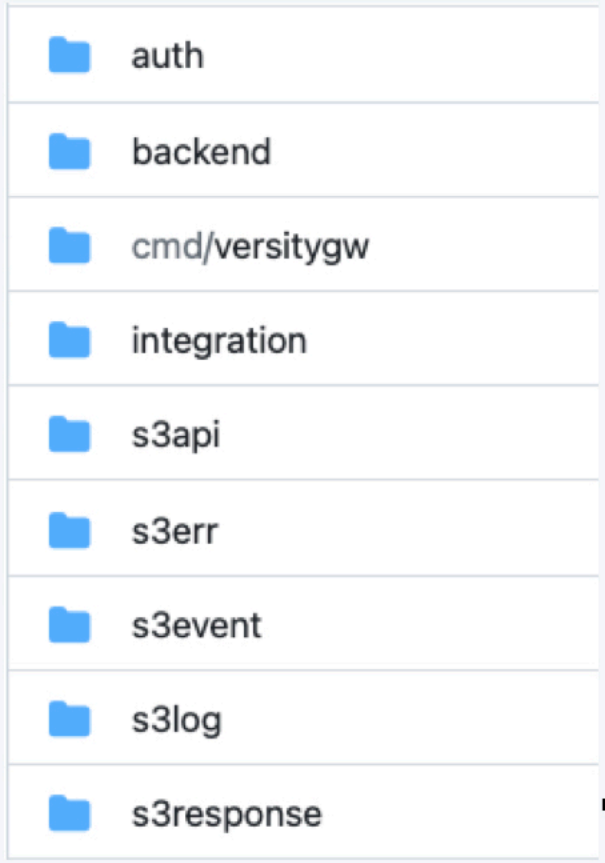
Same event structure as AWS

Extensible log integrations, currently supported:

- file
- webhook

→ S3 Server Access Logs

Project Layout



auth
backend
cmd/versitygw
integration
s3api
s3err
s3event
s3log
s3response

AWS defines specific xml structure for some requests/responses

For these requests, use well specified struct field members and xml struct tags

→ S3 Response formatting

Versity

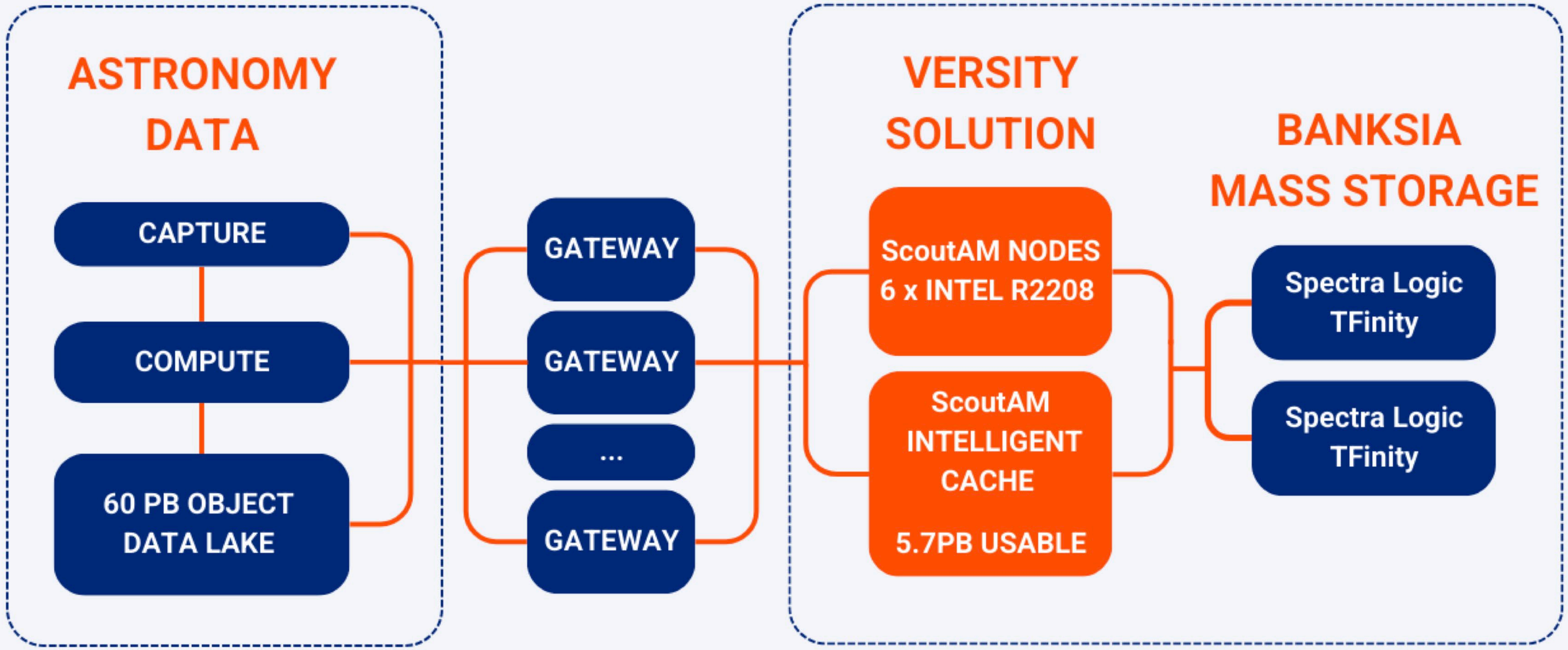
Gateway enables S3 workloads for Pawsey's Large Scale Archive

- Versity's ScoutAM managing tape-based archive, Banksia
  - 150 PB tape
  - 34 tape drives
  - 2 tape libraries
  - 5 PB cache
- Incoming data is S3 via gateway
- Cluster load balanced

Banksia
nimbus
inter-face
acacia
setonix

Versity

Pawsey Solution Architecture



Versity

Developed in the Open on GitHub

https://github.com/versity/versitygw

- Bugs/Features tracked in GitHub Issues
- Documentation in project wiki
- CI using GitHub Actions

Thank You

info@versity.com

@versitysoftware

https://github.com/versity/versitygw

Versity