Thought Leadership paper - "Metadata as Infrastructure" the data source that's been there all along

About the Author

Andrew MacKay - President & CTO of Superna

Abstract

This paper is a continuation of Securing the Future of Media Production, Post and Creative Technologies with Superna

Overview

What's old is new again. Metadata is a datasource that has always been available but rarely leveraged. The files we work with every day have rich metadata that few applications expose, use, enrich or extract value from. A good example is the lowly pdf, a basic pdf has 50+ metadata fields that describe the files contents. All of this rich data is available without reading the file's contents.

This paper will explore using file agnostic metadata to provide the foundation for advanced hybrid cloud workflows with metadata aware infrastructure that manages file and object data.

Some Background before we dig in: Data streams? File forks? Extended attributes?

These concepts are important to understand this proposal and the following references below provide additional background.

At a high level, these file system features allow files to carry file type agnostic metadata designed to signal infrastructure components how to secure, protect, manage data within a workflow.

File System Features

A good summary of file forks is available here https://en.wikipedia.org/wiki/Fork_(file_system)

Alternate Data Streams in Microsoft NFS NTFS file system allows storing and accessing one or more file forks all using the same file name.

Multi fork file system feature in MacOS HST+ (examples on how to create, access data forks)

Linux file systems have extended

attributes. They are usually abbreviated as xattr or xattrs.

Object Data features

Object data (S3) has metadata builtin from day one and allows name value pairs to be added to any object. Several loosely agreed on properties exist like version attribute, storage tier, etc.. The S3 protocol allows creating customer name value pairs within the header of every object and has been supported across all vendors for years. This metadata transport function has not been leveraged beyond simple data specific properties.

The Hybrid Cloud "Metadata as Infrastructure" Proposal

This paper proposes to use metadata, combined with file forks to carry json or xml metadata to signal infrastructure how to treat the data and bridge between file system and object custom properties. A set of common verbs with a schema that slows the possibility of vendor extensions to allow innovation outside of the common functions.

The key value of this solution allows any file type on any Operating system, over any local or network attached file system to carry application independent metadata.

The proposed solution would be protocol independent and would operate over SMB, NFS or any new protocol for accessing files.

What problem does this solve?

File formats and new applications change frequently but workflows are fairly static. Data life cycle functions such as archive, backup, signing, encryption, DR, storage tiering etc.. are very static data requirements.

The applications that provide these functions all have "policy databases" that store rules specific to the function they offer. Each vendor implements policies in a way that makes them incompatible with each other. Any synergies between applications offering the same or different functions are lost with vendor specific policy databases.

File to object or object to file data creation workflows are here to stay which means data transparency and infrastructure needs to treat all the data the same no

matter where it began life and throughout its lifecycle. A "single source of policy truth" needs to exist to move vendor innovation away from policy database implementations into higher value workflow based solutions that add value to the workflow versus the infrastructure.

If the policies lived inside the data itself, the scaling of a monolithic single point of failure database to handle millions or billions of files disappears.

It is clear, a better approach is needed to keep up with file and object application innovations and allow the infrastructure to get out of the way of innovation at the application layer.

Why would a common infrastructure metadata approach be useful?

The proposed solution would allow "**Data Inspectors**", functionality that allows infrastructure to inspect data within a workflow and apply policies or decisions on the data by reading the metadata within the file or object. A file type agnostic solution is required with an open format that allows for standardized properties.

The Data Inspector function can reside inside a business application, backup software, security software, archive software or any application that "**touches**" the data in any way. Each touch point along the data's life cycle, allows policies to be applied to the data, or new policies to be updated within the files metadata to signal downstream processes how to treat the data.

The xml or json metadata payload stored in the file fork would use a versioned agreed industry standard schema that allows vendors to extend the schema into proprietary fields for specific applications, features or workflows.

Using xml or json also has the advantage of human readable text for troubleshooting issues.

Using this approach to carry "Infrastructure instructions" within each file allows file to object mapping between file fork schema properties and S3 object custom properties.

Infrastructure instructions that should be covered in the common standard schema:

1. Retention time
2. Storage Tier
3. Security profile (none, low, medium

, high)
    a. Each profile would trigger automated protections such as DLP policies, audit logging of access to the data, security triggers and event notification)
4. DR requirement (protected, unprotected)
5. Indexing (content, metadata or both)
6. Backup number of copies to maintain and locations (local, cloud)
7. Cyber vault requirement true false
8. Application specific workflows where vendors can differentiate their use cases.

How would this proposal improve security?

File forks are files and can be read and written like any other file, allowing the properties contents to be verified with integrity checksums or field level encryption. A signature that can be applied across all the fields would allow data integrity, encryption of the fields, authentication using x.509 certificates or any similar method. The metadata properties can carry the encryption algorithm and signing algorithm to be implemented using metadata fields.

Encryption of the main file contents and its integrity checksum can be stored in the file fork adding another layer of storage independent data integrity to travel with the file via the metadata.

"**Data Inspectors**" could validate the main files integrity and ensure encryption is applied to all data within a workflow.

Conclusion

The hybrid cloud file and object workflow shift is needed to modernize application architectures to leverage the cloud and on premise infrastructure.

The complexity and incompatibility of file and object and related infrastructure products that manage , protect and secure data have no common method to be file and object agnostic.

If file/object data transparency is not addressed, hybrid cloud application adoption will not accelerate and the inherent advantages of file and object cannot be maximized.

More importantly, adoption of native hybrid cloud solutions will be slowed with continued use of home grown brittle building blocks without universal support

United States
3630 Peachtree Road, Suite 920,
Atlanta, GA 30326

from infrastructure application vendors that agree infrastructure as metadata policies and format.

A future paper will look at implementation details of this proposal.. Stay tuned.

United States
3630 Peachtree Road, Suite 920,
Atlanta, GA 30326