D7.3 Final Digital Asset Store Supporting Geolocation

Grant Agreement nr	780470
Project acronym	SAUCE
Project start date (duration)	January 1st 2018 (36 months)
Document due:	30th September 2020
Actual delivery date	30th September 2020
Leader	Sam Hudson
Reply to	peri.friend@foundry.com
Document status	Submission Version

Project funded by H2020 from the European Commission

Project ref. no.	780470
Project acronym	SAUCE
Project full title	SAUCE
Document name	D7.3 Final Digital Asset Store Supporting Geolocation
Security (distribution level)	PU
Contractual date of delivery	30th September 2020
Actual date of delivery	30th September 2020
Deliverable name	Final Digital Asset Store Supporting Geolocation
Type	Demo
Status & version	Submission Version
Number of pages	12
WP / Task responsible	WP7 Task 2
Other contributors	-
Author(s)	Sam Hudson, Tom Mulvaney, Dan Ring, Peri Friend
EC Project Officer	Ms Adelina Cornelia Dinu - adelina-cornelia.dinu@ec.europa.eu
Abstract	WP7T2 has taken the prototype developed in WP5T5 and extended it to cater for handling of assets that are being shared by projects that have multiple locations globally. To handle this, development of the store is built on top of a modern REST architecture. This allows hybrid user technologies (mobile, web desktop) to be used so that users can collaborate on the same project using their preferred device
Keywords	Asset store, asset pipeline, cloud
Sent to peer reviewer	Yes

Peer review completed	Yes
Circulated to partners	No
Read by partners	No
Mgt. Board approval	No

Document History

Version and date	Reason for Change
1.0 24-04-20	document created by Peri Friend
1.1 23-09-20	Version for internal review (14 days before submission date)
1.2 28-09-20	Revisions in response to review: final versions submitted to Commission

Table of Contents

1 EXECUTIVE SUMMARY

We demonstrate the new Flix feature 'Regions' which will allow for each Flix Server to be configured to exist within a Region. This facilitates server grouping, allowing for Flix Clients to connect only to a subset of Flix Servers. This improves user experience across geographical locations so requests are only made to servers in a geographically optimal location. Flix also replicates assets across regions using a configurable algorithm to ensure high availability and redundancy.

1.1 Relationship to Self Assessment and other Deliverables

This deliverable follows on from D7.2 Prototype of Asset Store, which is at TRL8/9 as the refactor enabling this workflow is already in use with customers and being continually improved. The asset store is the basis for D7.3 Final Digital Asset Store Supporting Geolocation at M33 and D5.6 Transcode Mechanism, both of which go from TLR2 Basic Research to TLR6 Technology Demonstration.

The final digital asset store will reduce asset retrieval time, allowing for capacity and scalability as well as enabling high availability. These are laid out in D1.2 Self Assessment plan. Details of how this will be assessed are included in D8.3, also submitted at M30.

2 BACKGROUND

Flix is a storyboard-development software for collaborative work in animated film, TV and gaming that promotes fast-paced collaboration by removing technical barriers that slow productions down. Winner of a 2013 HPA Engineering Excellence Award, Flix was developed as an in-house solution at Sony Pictures Imageworks before FO acquired it. It has been used in production since 2008 on more than 15 projects including Cloudy with a Chance of Meatballs 2, The Smurfs and Hotel Transylvania. It is currently integrated with specific industry leading applications such as Photoshop, Storyboard Pro, Avid, Premiere, Final Cut Pro. Flix will be extended (beyond just storyboards) so that the data model can be used in new contexts, for different post-production tasks and purposes. It will be used by users in varying environments and will require supporting new types of assets for parts of the pipeline such as lighting & texturing, modelling, animation, FX and compositing. Flix has been extended in a modular way so as to cater for the variety of types of new assets that need to be supported. This structure will additionally allow for other applications to interface with Flix.

3 INTRODUCTION

A common and growing problem is access to data. With more of the workforce becoming remote workers, freelancers or in co-located offices, the challenge of making data accessible to a user wherever they are becomes more desirable. Currently, many studios simply use network shares, NFS or SMB to make filesystems available to their users. This is OK for more traditional on-premise network topologies, but poses problems with federating secure access to smaller groups, or remote users due to VPN requirements. We have taken this requirement into consideration and devised a mechanism to allow for the asset store to be outside of a firewall and available securely to end-users across the internet. This is accomplished without the need to expose insecure file shares and complex VPN management.

The current architecture of Flix ensures that there is high availability, which means that multiple servers are always available to users to ensure that there is no single point of failure. This is currently implemented in a way that ALL Flix servers in the cluster are accessible to the Client application. This design requires further development to cater for some more advanced setup scenarios. A Studio may wish to spread their Flix servers over multiple geographical locations, for example an office in New York, and an office in Los Angeles. Currently, the traffic of all Flix users would be spread across all of the available servers. They would

experience a differing quality of service depending on their distance from each location. We have solved this problem by introducing a concept of 'Regions' within Flix, to enable our Geolocations feature.

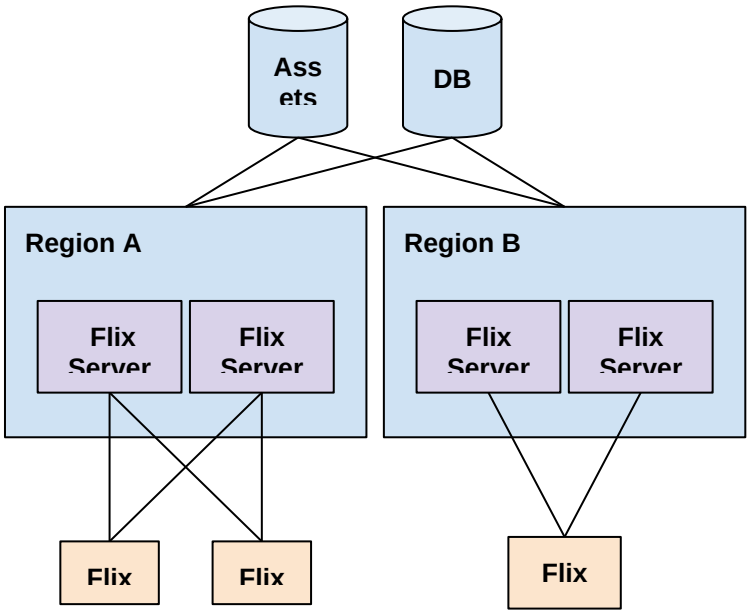
Regions would allow for servers to be tagged with a specific Region identifier, which would then allow the Client application to decide which region is optimal, and limit its requests to the subset of servers which exist within the region specified. This would ensure that users always have an optimal experience while still ensuring Flix can offer multi-site environments.

3.1 Main objectives and goals

The main objective for this work is to ensure that users of Flix in a multi-site environment enjoy a stable experience without varying latencies during their session. This means ensuring that connections to Flix servers would be maintained to one geographical location, instead of load-balanced across all Flix Servers as it currently is.

Due to the initial design criteria we have followed during Flix server development,there are some complexities with segregating Flix servers into sub-groups. All Flix servers are intended to be able to accept any task available in the pool of tasks, whether that is accept a HTTP request, or render out a thumbnail for a panel. This allows servers to be ephemeral, and somewhat stateless. However, with dividing the cluster into smaller groups we would need to implement further changes to all of these job scheduling algorithms to take the new regions into consideration.

Likewise for asset replication. Flix can be configured in two different ways for asset replication. The simplest is 'shared storage' in which Flix assumes its filesystem for storing assets is available across all Flix Servers and therefore every server always has access to every asset. This relies on external services to ensure concurrency of the fileshare. Flix can take care of asset replication itself when 'shared storage' is turned off, and this has various implications. Flix needs access to assets for processing, or simply serving them up to clients. So with shared storage turned off, Flix Server will maintain its own cache of assets locally on disk. This then requires some extra steps involved when Flix Server handles requests as it may not have the required asset available to it. So it would then request the asset from another server to store in its local cache.



D1: This diagram displays 3 different Flix Clients and their differing connections to Flix Servers across Region A or B.

3.2 Terminology

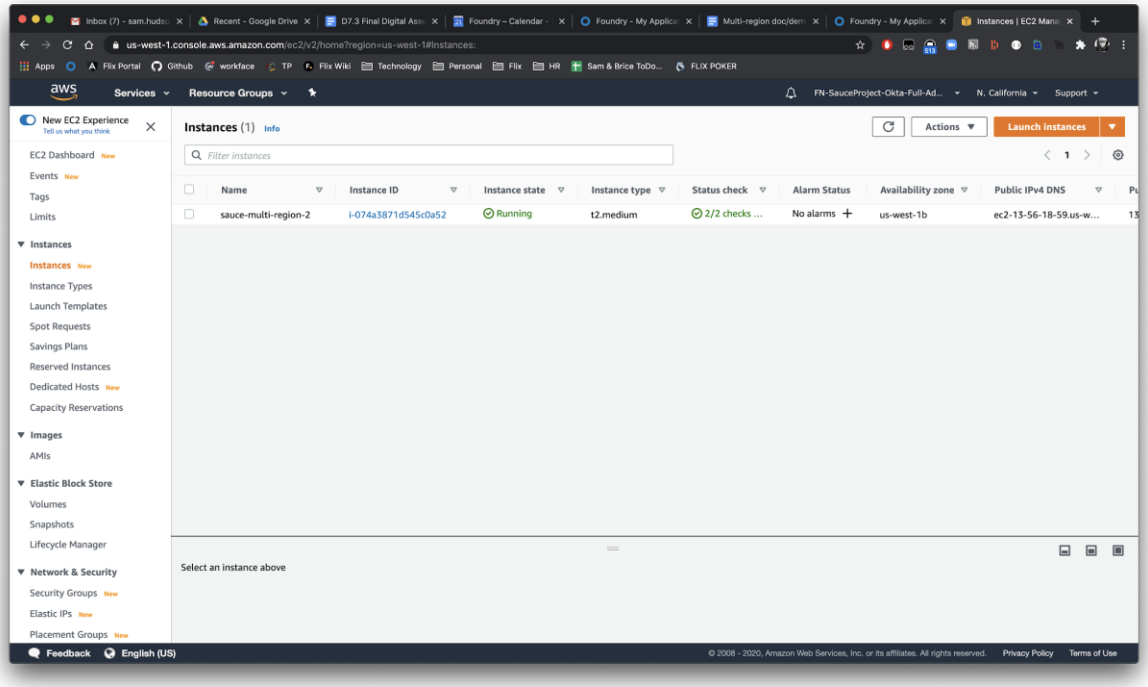
Term	Description
Asset	An *asset* is an entity which can be uniquely identified in the storage system. An asset does not directly represent a 'file', 'image', or blob of data. An *asset* can and usually will have a relationship with one or many underlying media objects. Underlying media objects may be different representations of a single concept which collectively or individually make up the asset.
Location	Locations are physical locations in which a Media Object may reside. The locations are usually a unique identifier to a particular *Server* in the system.
Server	A server is an instance of the Flix Server executable. It can be assumed that the relationship between the running software, and the hardware is a single concept. The server hardware, and the server application are one.
Region	A *region* is a logical grouping of *Servers*. This is simply a way to group *servers* into smaller clusters. The most common use-case for this grouping would be the segregate *servers* by their geographical location, hence the name *region*.

4 Flix Server Multi Region

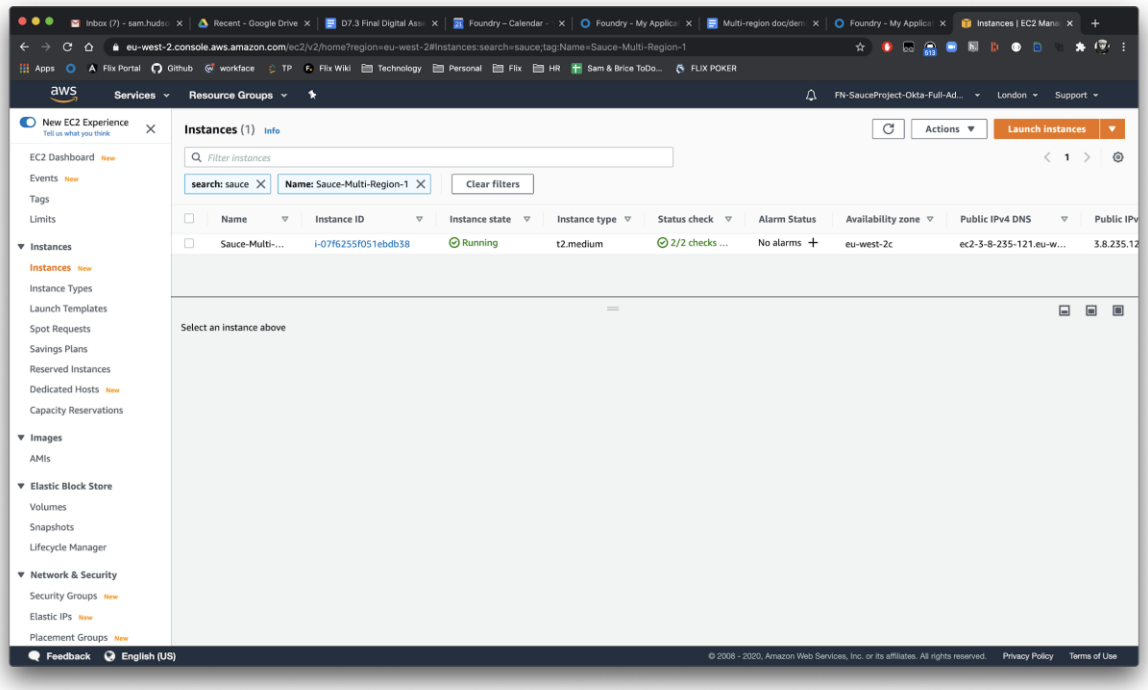
This deliverable will be a demonstration of the multi-region feature within Flix. We demonstrate the use of Amazon Web Services to provide virtual machine servers in physically different geo locations to demonstrate a Flix Client connecting to different regions and the quality of experience compared with Flix without this feature.

4.1 Demonstration Server Architecture

We configured our AWS account to have two Flix Servers running on Linux EC2 instances in different physical locations. We used 1 server in the EU region, and 1 server in the US region. This should provide a clear demonstration that Flix can work across geographical locations. The servers share a database which is installed on one of the servers, and accessible to the other. This could be configured in a more optimal way to improve performance, but this would be outside of the scope of this example.



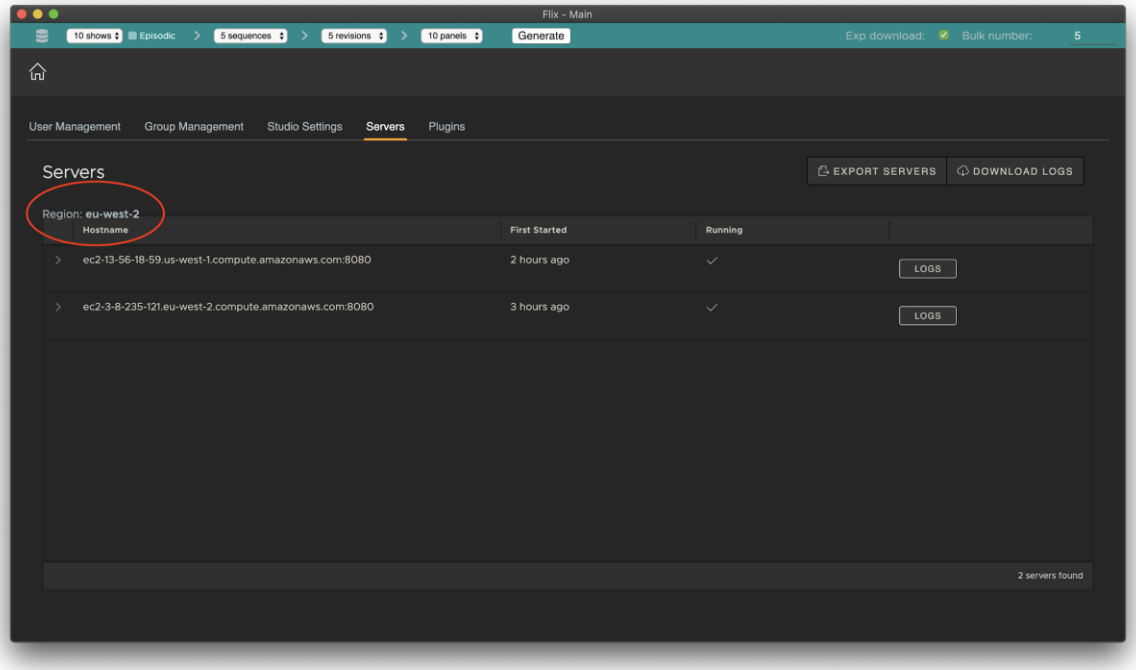
D2: Diagram displays an EC2 VM running in California AWS Region



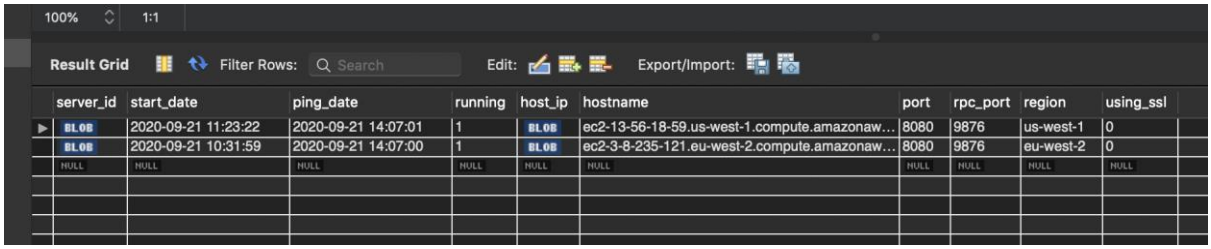
D3: Diagram displays an EC2 VM running in London AWS Region

4.1.1 Demonstration of Flix Server Application Configuration

We display that Flix Server is configured to have two running servers, and that they are divided by a "region" name. The Flix Client Administrator Console will display all of the servers available in the installation. It also displays the current regions which the user is logged in to. We have some further UI work to do to make understanding which regions are configured, and which servers exist within each region.



D4: Image shows our current region name in the Flix Admin Console screen

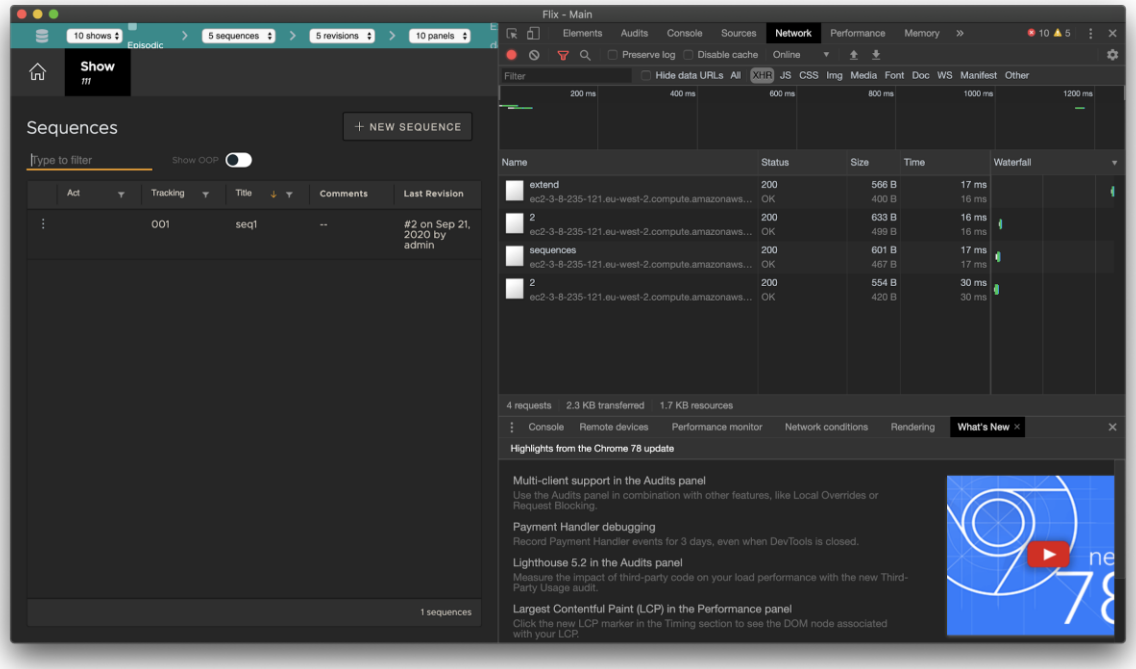


server_id	start_date	ping_date	running	host_ip	hostname	port	rpc_port	region	using_ssl
BLOB	2020-09-21 11:23:22	2020-09-21 14:07:01	1	BLOB	ec2-13-56-18-59.us-west-1.compute.amazonaw...	8080	9876	us-west-1	0
BLOB	2020-09-21 10:31:59	2020-09-21 14:07:00	1	BLOB	ec2-3-8-235-121.eu-west-2.compute.amazonaw...	8080	9876	eu-west-2	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

D5: Image shows the mysql database which contains information about all Flix Servers in the installation, it displays that there are two servers, and each exists within a different region name.

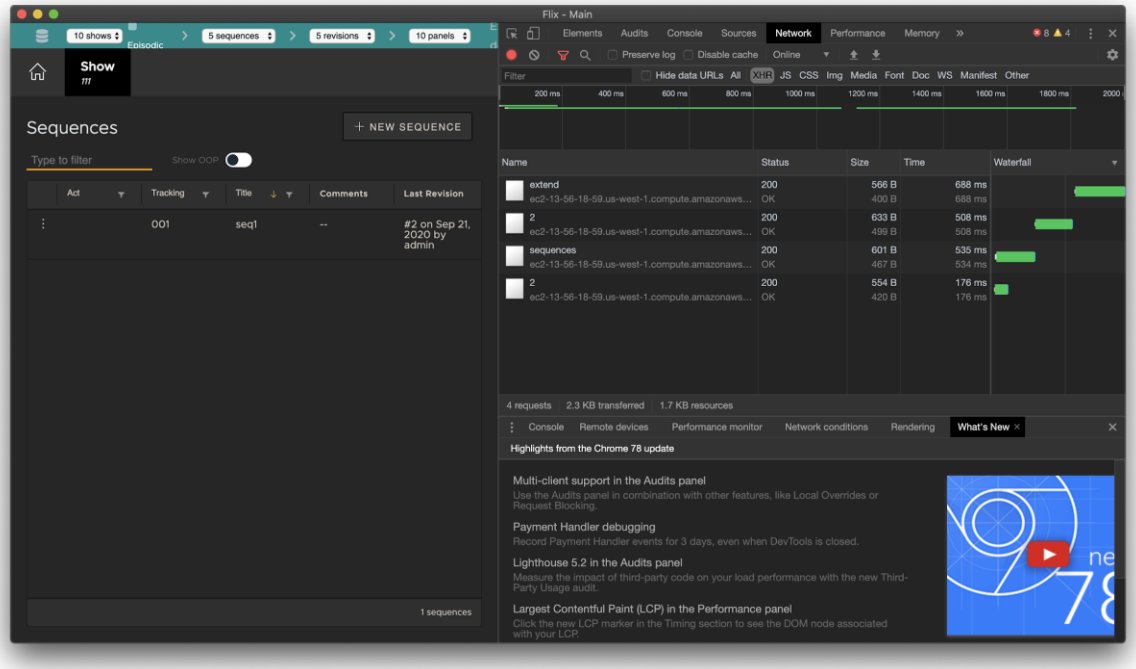
The following images show while using the Flix Client with the regions feature enabled, that all HTTP requests are limited to only servers within the users region. This is indicated by the urls in the console window on the right hand side. The two images also show a considerable latency saving between using a server in the US, vs a server in the UK. The UK server has approximately 20ms response times, whereas the US server has 500ms response time.

D6: Logging into Flix using the URL for the Server in the London Region



D7: Shows Flix Client loading 'Sequence' information, and the console logging debug window on the right hand side which shows 4 HTTP requests averaging 20ms round trip timing each

D8: Logging into Flix using the URL for the California Region, while being geographically located in London



4.1.2 Media Object Replication Strategies

We have introduced the concept of 'replication strategies' into this new feature. A replication strategy is the type of algorithm used which will control how media objects are replicated and made available across different regions and servers. Our aim was for the replication algorithms to be extensible so we can continue our research into the future to devise more advanced replication methods, potentially utilizing machine learning, to predict when and where assets would be required geographically.

Initially we implemented two different replication methods.

1. **Lazy Replication** will not replicate media objects to other regions, until they are requested from within a different region. This serves some useful purposes. Our customers will often have multiple offices across different locations, but each location will be working on a different show. It would be rare that they would require assets from other locations, however it may be required sometimes. So this replication method would be a good fit.
2. **Pro-Active Replication** utilises a set configuration to ensure a certain number of replications are made for each media object across regions. This is suitable for cross site collaboration on the same projects, and ensures there are always copies of assets in each location for redundancy and time to access.

5 Conclusion

The introduction of this feature allows for Flix to be further suitable for a wider variety of customer use cases and environment configurations. Previously Flix would not provide an optimal user experience when used in a multi-site configuration. This new feature will allow customers who previously could not use Flix for the demands they have to now more closely match their requirements. This feature is targeted to a Flix release within Q1 of 2021.

FO will continue to improve this feature with further replication algorithms to fit more scenarios in customer environments. There is further scope to provide AI driven algorithms.

6 References

Flix API Documentation: http://docs.flix-dev.com/
Flix Documentation: https://learn.foundry.com/flix

7 Acronyms and abbreviations

FO: Foundry
DNEG: DNEG (Formerly Double Negative)
USAAR: University of Saarland Informatics Campus
UPF: Universitat Pompeu Fabra
FA: Filmakademie
TCD: Trinity College Dublin
DRZ: Disney Research Zurich
BUT: Brno University of Technology