Simple Storage Service (S3)

Chandra Lingam

Cloud Wave LLC

S3 Storage Classes

Storage Class	Standard	IA	One Zone IA	Glacier	Glacier Deep Archive
Usage	Frequently Accessed	Less frequently accessed		Long term archival	
First byte latency	Immediate	Immediate		Minutes to hours	Several hours
Retrieval Fee		Per GB		Per GB	
Monthly Cost USD 500 GB	11.50	6.25	5.00	2.00	0.50
Minimum		30 days, 128 KB		90 days, 40 KB	180 days, 40 KB
Durability	99.999999999% (11 9's). Average annual expected loss of 0.000000001% of objects.				

Durability

"For example, if you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years."

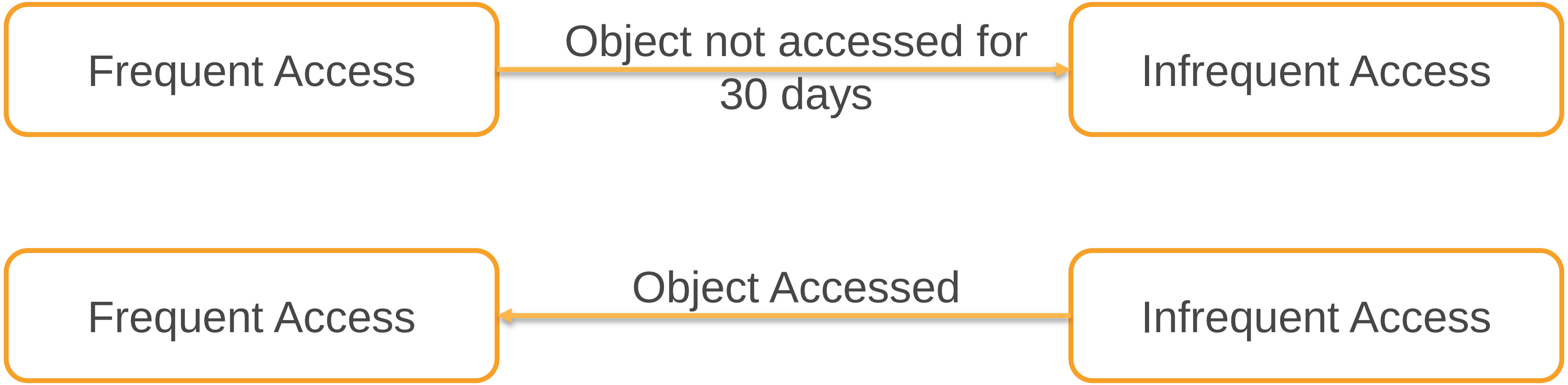
https://aws.amazon.com/s3/faqs/

- Backup
- Secure access permissions
- Replication
- Versioning

S3 Intelligent Tiering

“Automatic cost savings for data with unknown or changing access patterns”

Objects are automatically moved between frequent access and infrequent access storage classes



Frequent Access

Object not accessed for 30 days

Infrequent Access

Frequent Access

Object Accessed

Infrequent Access

S3 Intelligent Tiering Pricing

Cost of storing 500 GB/Month in Intelligent Tiering Storage Class

- Frequent Access: USD 11.50 (same as standard)
- Infrequent Access: USD 6.25 (same as IA)
- Minimum: 30 days

There is a separate monitoring and automation charge/month: $0.0025 per 1,000 objects

Glacier Retrieval Options

	Expedited	Standard	Bulk
Glacier	1 – 5 minutes	3 – 5 hours	5 – 12 hours
Glacier Deep Archive		Within 12 hours	Within 48 hours

Provisioned Capacity:

- If you frequently used expedited retrieval, AWS may reject the request during periods of high demand
- You can purchase provisioned capacity to ensure expedited retrieval capacity is available when you need it

Storage Class Analysis

“One of the challenges of developing and configuring lifecycle rules for the data lake is gaining an understanding of how data assets are accessed over time.”

Reference: Data Lake on AWS,
https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/building-data-lake-aws.html

Storage Class Analysis

“This new Amazon S3 analytics feature observes data access patterns to help you determine when to transition less frequently accessed STANDARD storage to the STANDARD_IA storage class”

Reference: S3, https://docs.aws.amazon.com/AmazonS3/latest/dev/analytics-storage-class.html

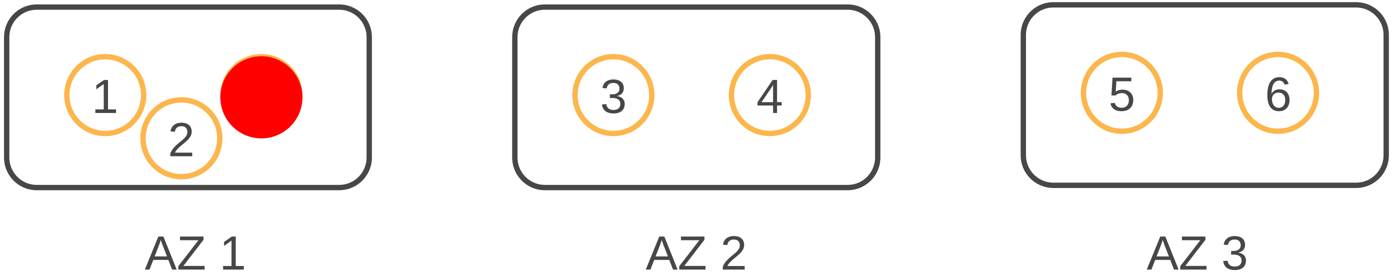
Consistency Model

Durability

"Amazon S3 Standard, S3 Standard-IA, and S3 Glacier storage classes redundantly store your objects on multiple devices across a minimum of three Availability Zones (AZs) in an Amazon S3 Region before returning SUCCESS."

"The S3 One Zone-IA storage class stores data redundantly across multiple devices within a single AZ."

Cyclic Redundancy Check (CRC) and Checksums to detect data corruption and repairs corruption using redundant data



Reference: https://aws.amazon.com/s3/faqs/

S3 Consistency

Atomic at Object level

Data is safely stored (PUT)

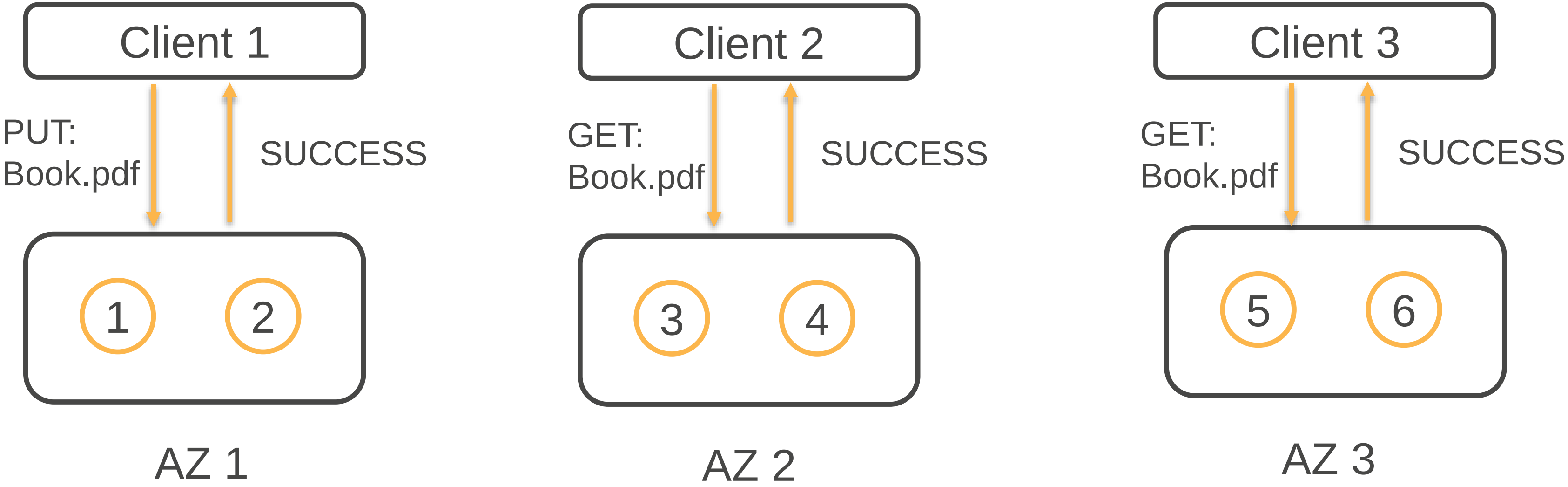
Never stores or returns partial or corrupted data

Replicated across S3 asynchronously

Consistency Model – New Objects

Read-After-Write-Consistency - Any new object that was written can be read immediately

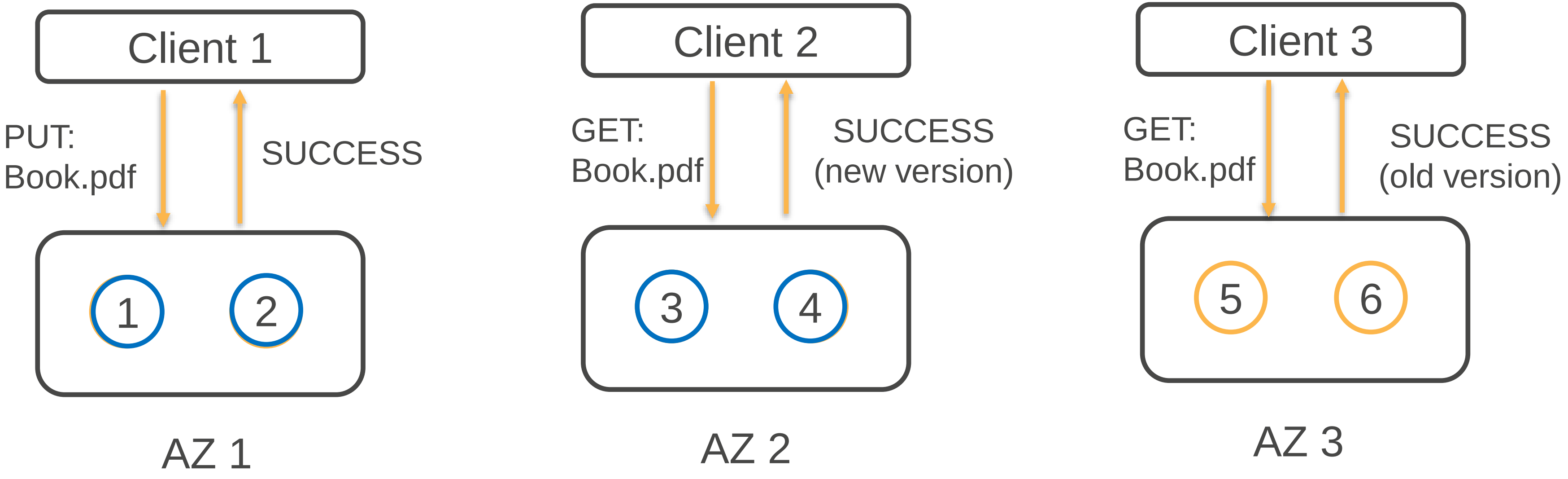
Reference: https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html#ConsistencyModel



Consistency Model – Update Objects

Eventual Consistency - For updates, subsequent read may return old data until change is fully propagated (but it never returns corrupted or partial data)

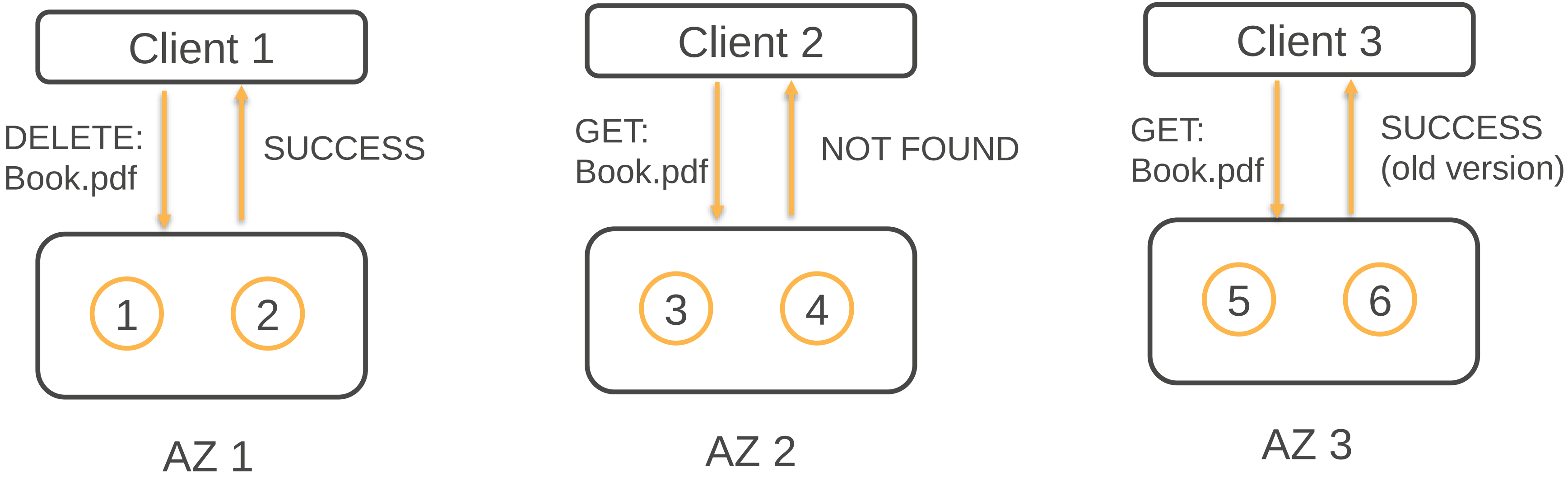
Reference: https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html#ConsistencyModel



Consistency Model – Delete Objects

Eventual Consistency - For deletes, subsequent read may return old data until change is fully propagated (but it never returns corrupted or partial data)

Reference: https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html#ConsistencyModel



S3 Consistency Model - Summary

Read-after-write-consistency – New objects can be read immediately

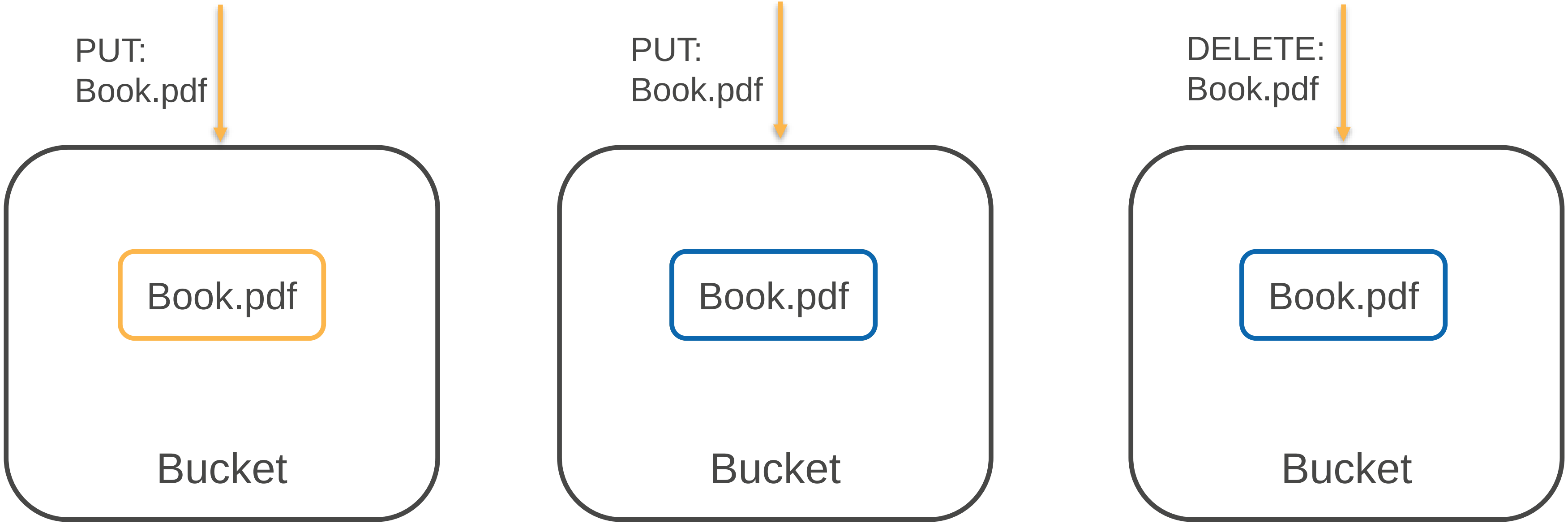
Eventual Consistency

- Stale read possible after updates and deletes (until change is fully propagated)
- Lowest read latency
- Highest read throughput

Versioning

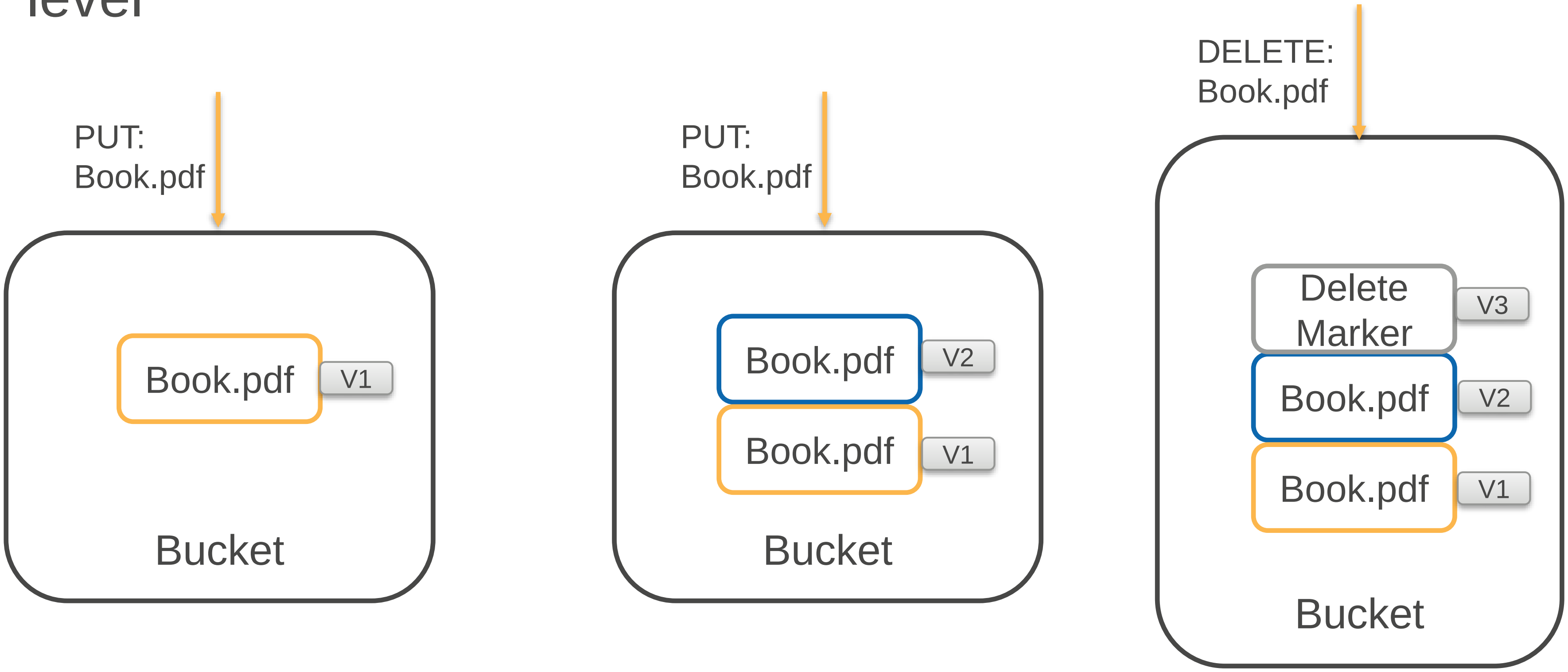
Without Versioning

Changes are not reversible

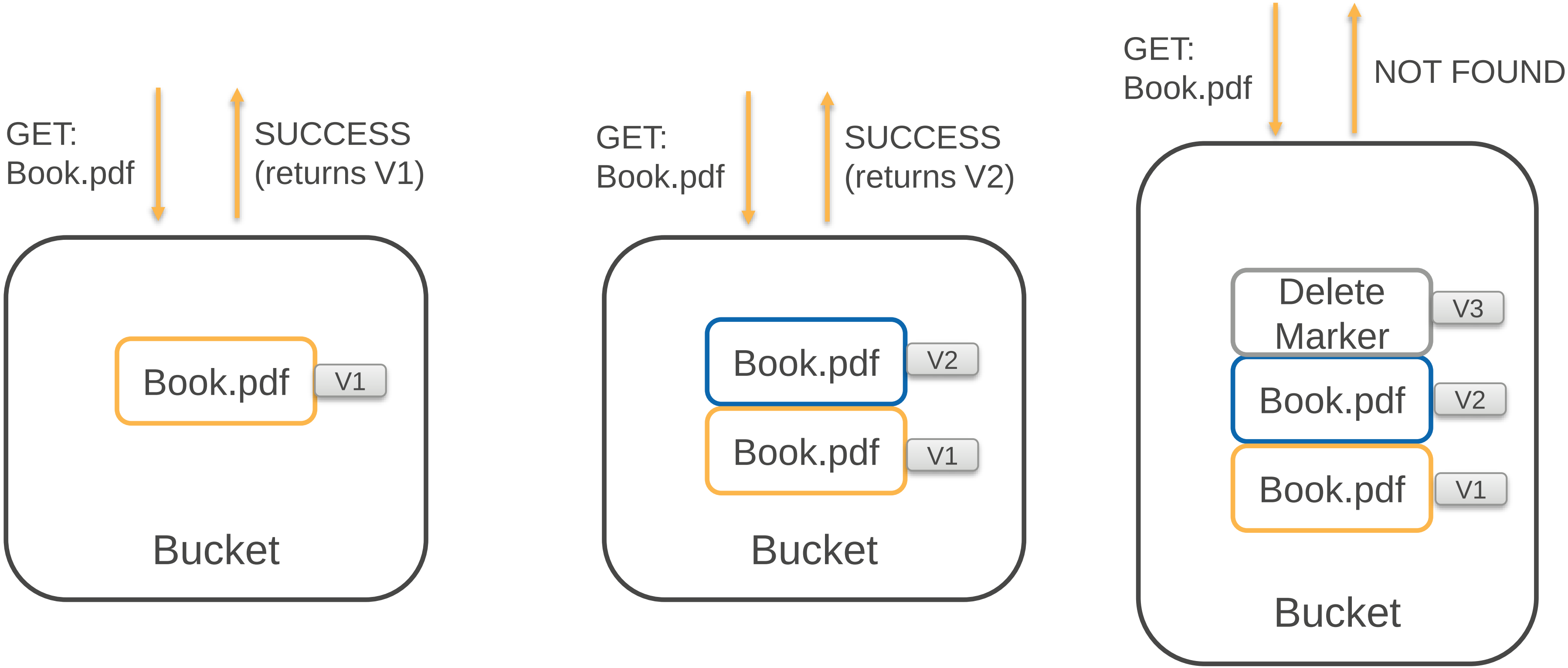


With Versioning

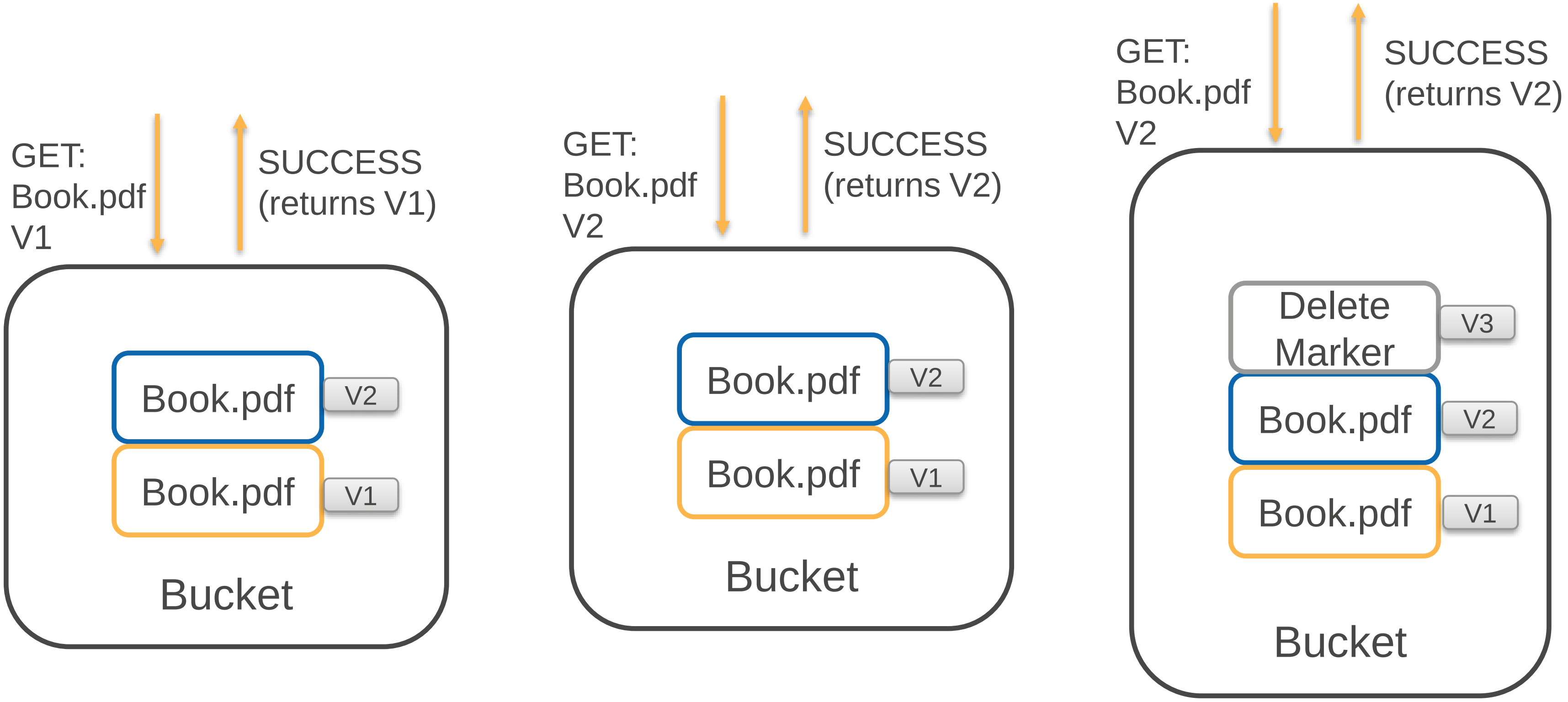
Complete history of changes - Enable or Suspend at bucket level



GET Object - Returns current version

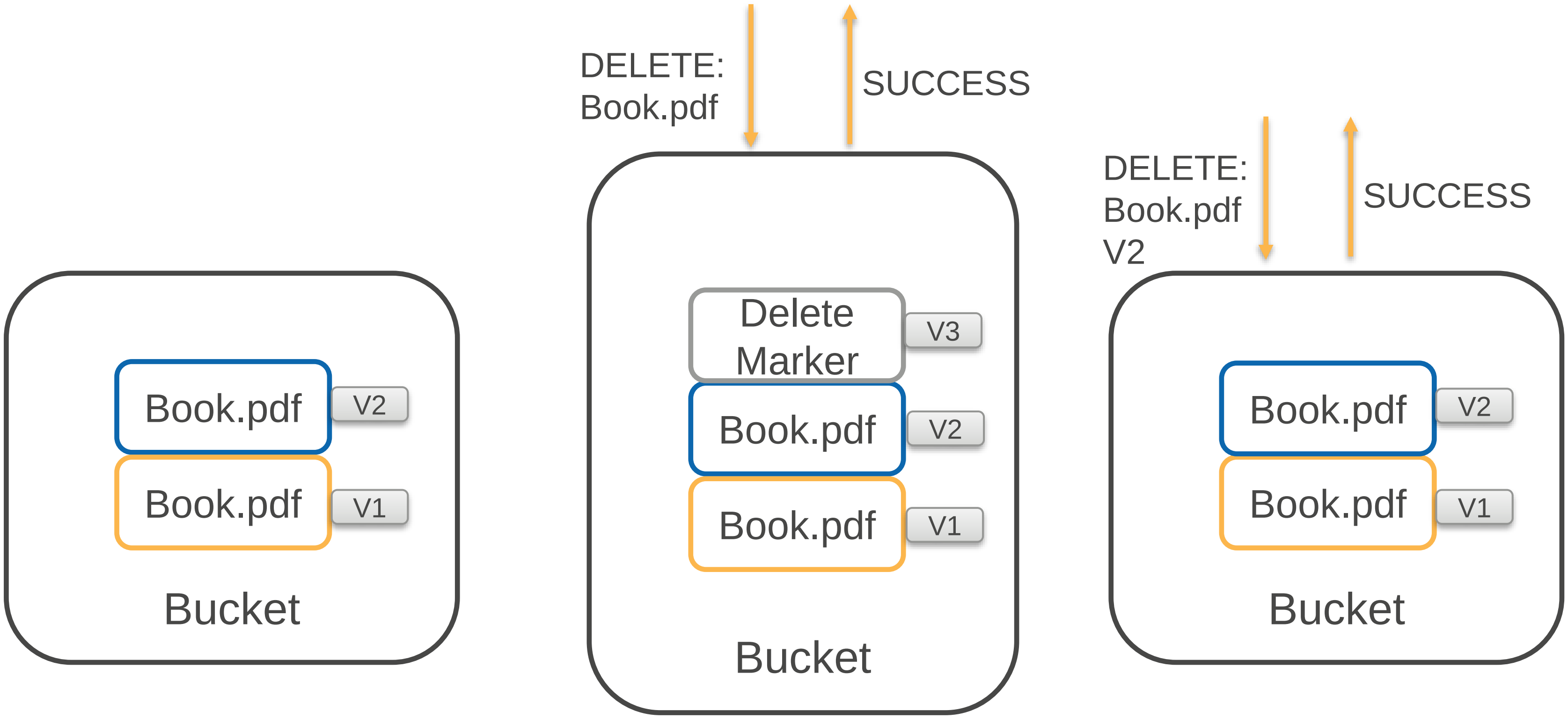


GET Object Version - Returns specified version



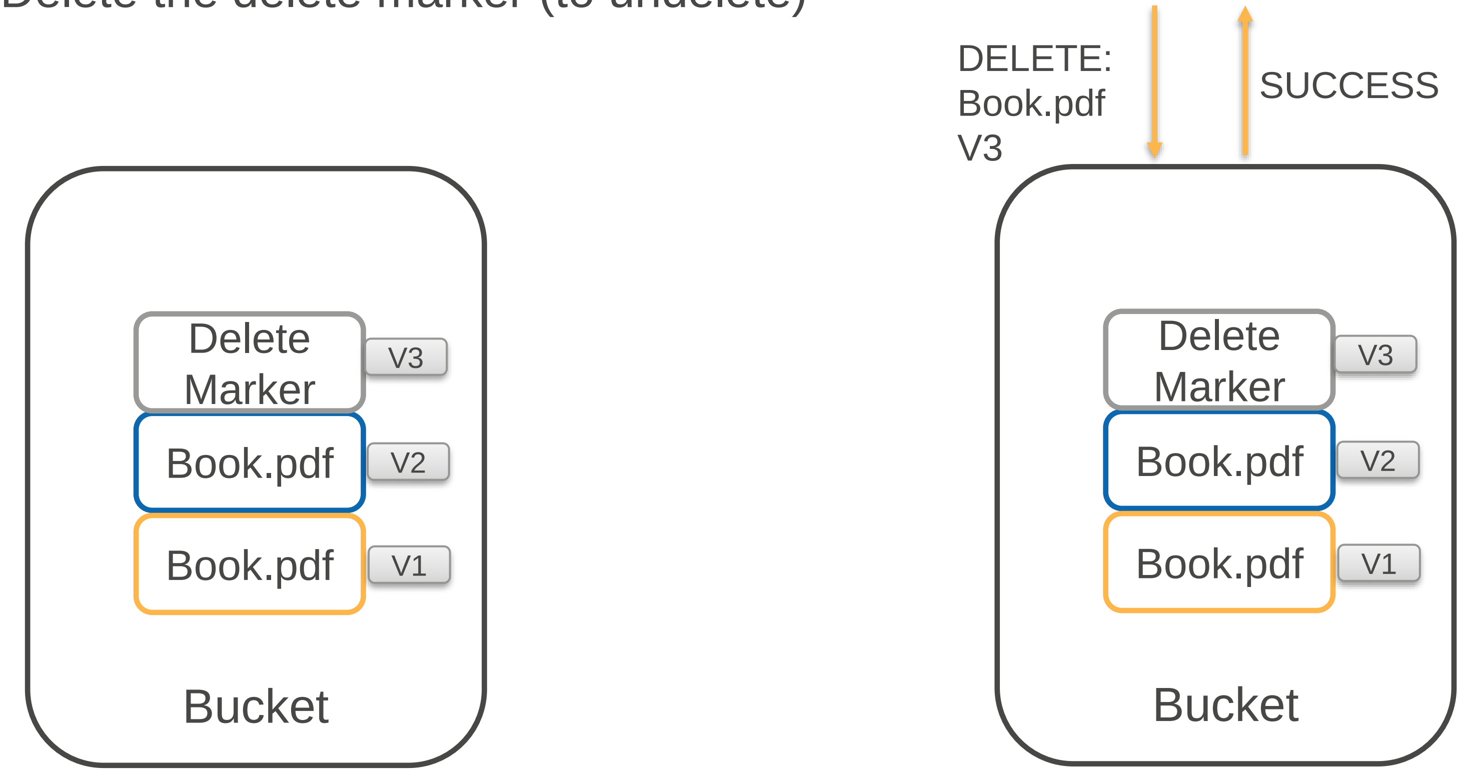
Delete version

Delete a specific version (permanent)



Undelete

Delete the delete marker (to undelete)



Delete Marker V3

Book.pdf V2

Book.pdf V1

Bucket

DELETE: Book.pdf V3

SUCCESS

Delete Marker V3

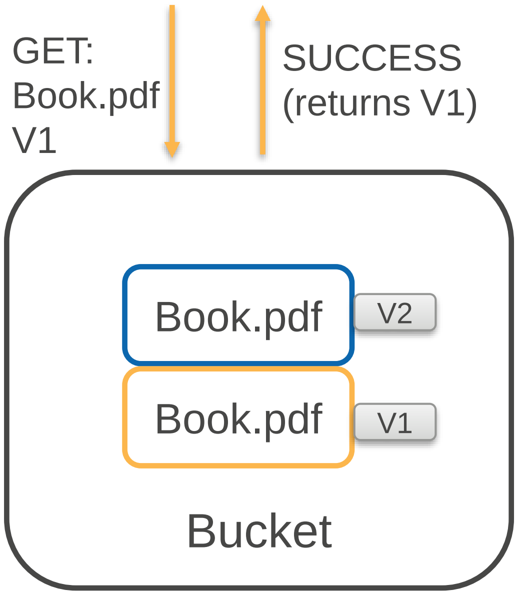
Book.pdf V2

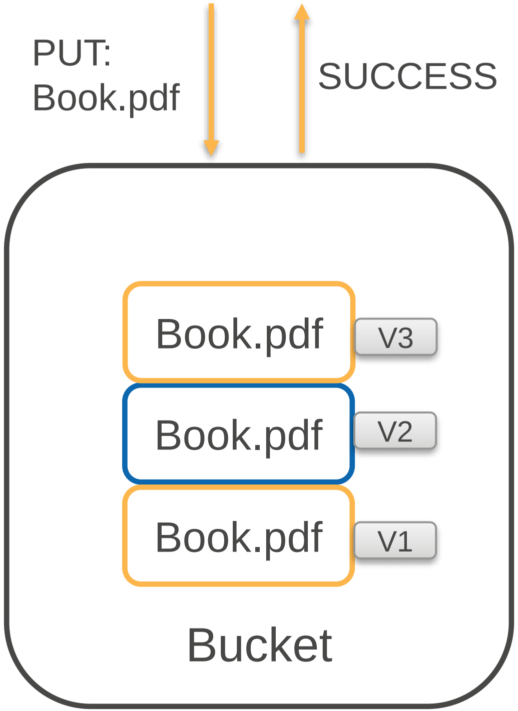
Book.pdf V1

Bucket

Restore a version

Read the version and write it back

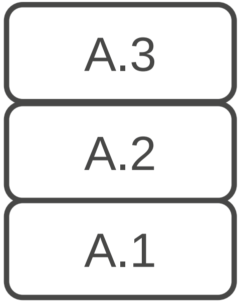




S3 Versioning

Protection against accidental and malicious deletes

S3 maintains versions of objects (full copy)



A.3

A.2

A.1

Configure Lifecycle Rules for current and previous versions

Multi-Factor Authentication (MFA) for additional layer of authentication

Lifecycle Management

Lifecycle Management

- Tiering - Transition to lower cost storage
- Expiration – Remove objects that are not needed
- Archiving - For long term retention
- Versioning – Handle current and previous versions

Examples: Log files might be needed only for a few days. Data files that are frequently accessed for a first days and then infrequently accessed. Archive data for long term retention

Reference: https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html

Cost Considerations

Storage Class	Minimum Size	Minimum Duration
Infrequent Access	128 KB	30 days
Infrequent Access (One Zone)	128 KB	30 days
Glacier	40 KB	90 days
Glacier Deep Archive	40 KB	180 days

- Aggregate smaller objects to few larger objects
- Transition to lower cost storage only if you plan to keep beyond minimum duration

Lifecycle Management

Define rules based on:

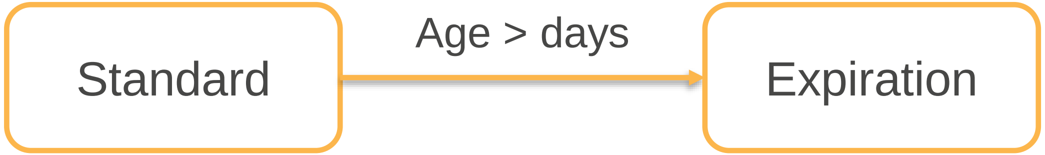
- Object Age
- Current and previous versions

Filter based on:

- Prefix (**images**/, **logs**/)
- Object Tags (**Name=PHI**)

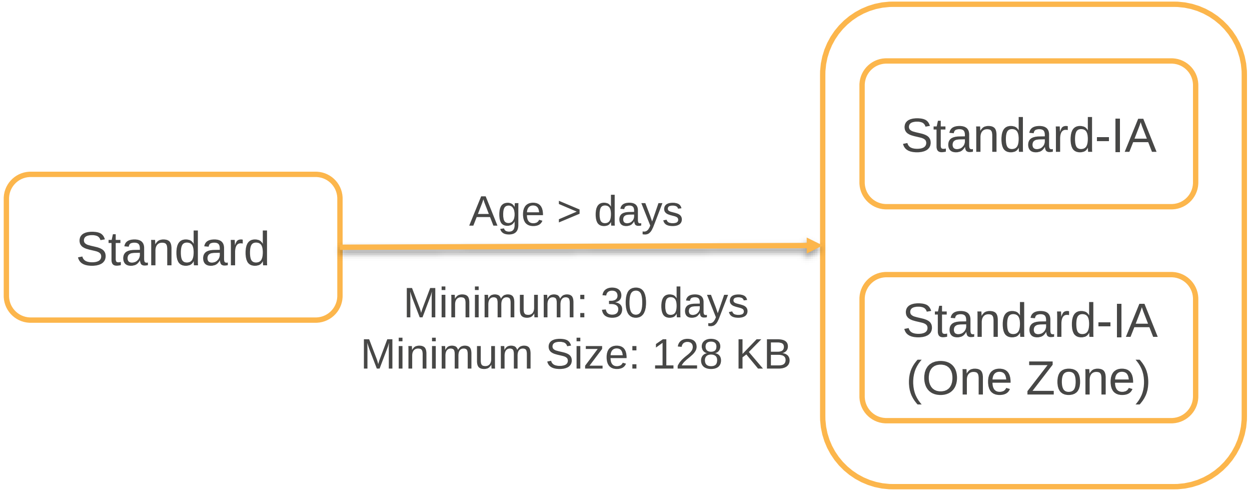
Scenario - Expiration

Remove objects few days after creation



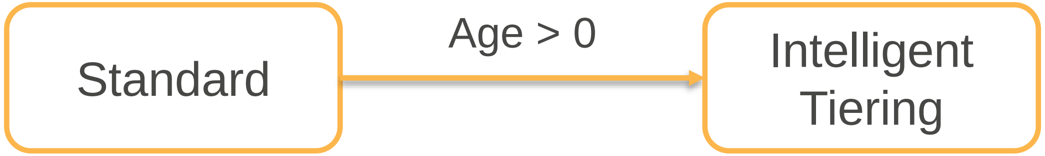
Scenario – Lower Cost Storage

- Move object to infrequent access tier after few days
- Object must remain in Standard tier for at least 30 days and size > 128 KB
- Object must be kept in infrequent tier for at least 30 days



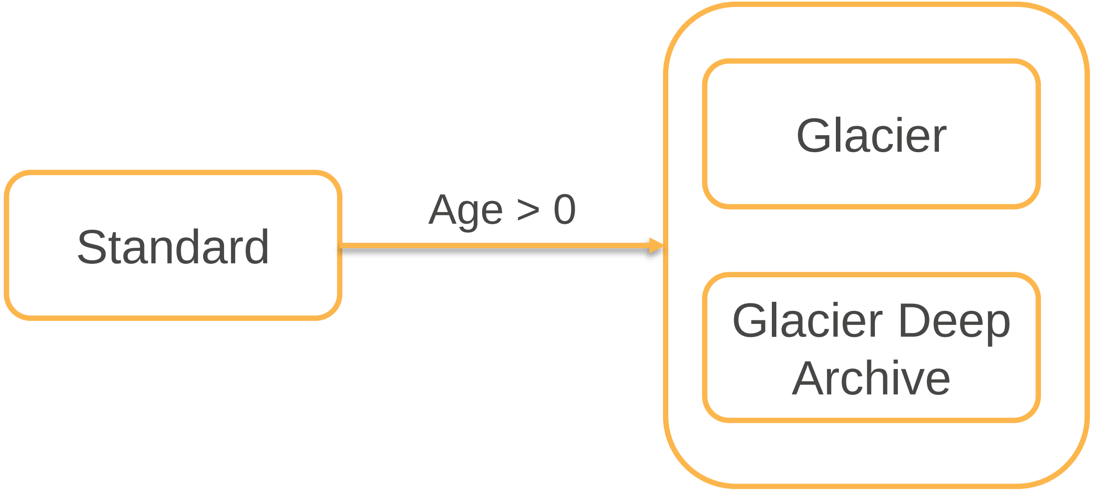
Scenario – Intelligent Tiering

- Move objects to intelligent tiering immediately after it was created
- Intelligent Tiering has a minimum charge for 30 days



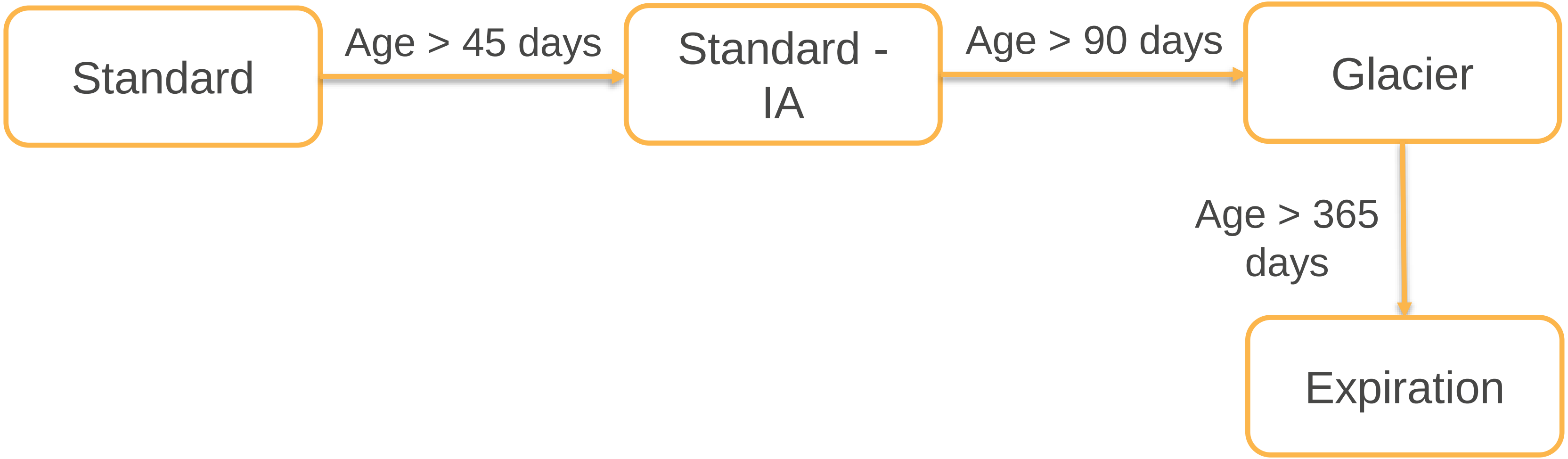
Scenario – Archive

- Move objects to Glacier immediately after it was created
- Glacier has a minimum charge for 90 days and Deep Archive 180 days



Scenario – Tiered Storage and Expiration

Optimize cost - Standard, Infrequent, Glacier, Expiration



S3 Access Control

S3 Access Control

User-based Policy, Roles (IAM)

Resource-based

- Bucket Policy
- Bucket Access Control List (ACL)
- Object Access Control List (ACL)

Bucket Policy

- Grant permissions to users, services, roles in the same account
- Cross-account access to the bucket
- Network origin control

Bucket ACL

Only recommended use for Bucket ACL

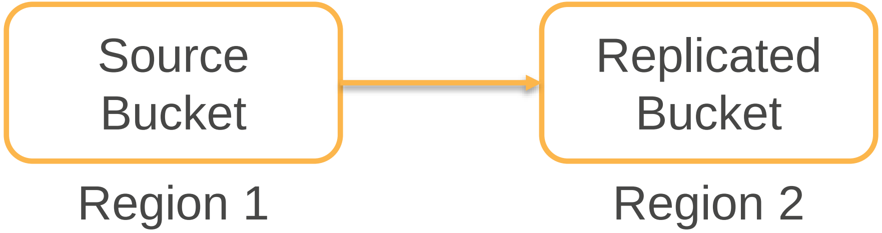
- Grant access to S3 `Log Delivery Group` to write S3 access logs to your bucket
- Bucket ACL is the only way in which `Log Delivery Group` can be granted access
- Cross Account Access
- Account can be referred by email address or Canonical ID

Object ACL

- Control permissions at object level - Permissions vary by object
- Object owner is different from bucket owner
    - Bucket owner cannot read until permission is granted by object owner
    - Object ACL is the only way an object owner can grant permissions to the bucket owner
    - Bucket owner can deny access to object
- Account can be referred by email address or Canonical ID

S3 Replication

Cross Region Replication (CRR)



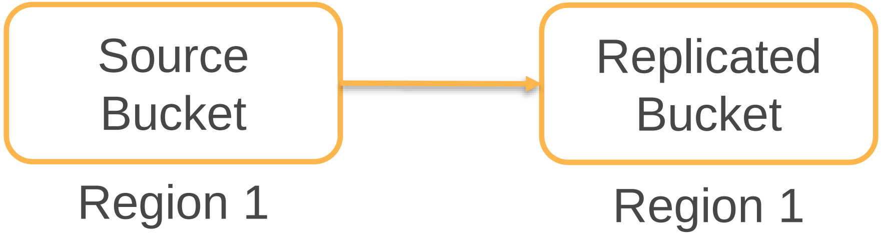
https://docs.aws.amazon.com/AmazonS3/latest/dev/replication.html

Meet compliance requirement – Disaster Recovery

Minimize Latency – for customers in different geographic locations

Operational Efficiency – Compute Clusters in different regions that need access to same set of objects

Same Region Replication (SRR)



Aggregate Logs to a single bucket

Live Replication from Production to Test account

Compliance - Multiple copies of data in separate accounts

S3 Replication

- Automatic and continuous replication
- Existing objects are not replicated - only new changes are replicated (do batch copy to initialize)
- Flexibility to use different storage class for replicated data
- Object and metadata are replicated
- Deletes are not replicated (to protect against malicious deletes)
- Have separate lifecycle rules in destination bucket

S3 Replication

- Configuration: Destination Bucket, Role S3 can assume to replicate objects
- Optional: S3 Replication Time Control – replicates 99.99% of new objects within 15 minutes (backed by SLA) at additional charge

Performance

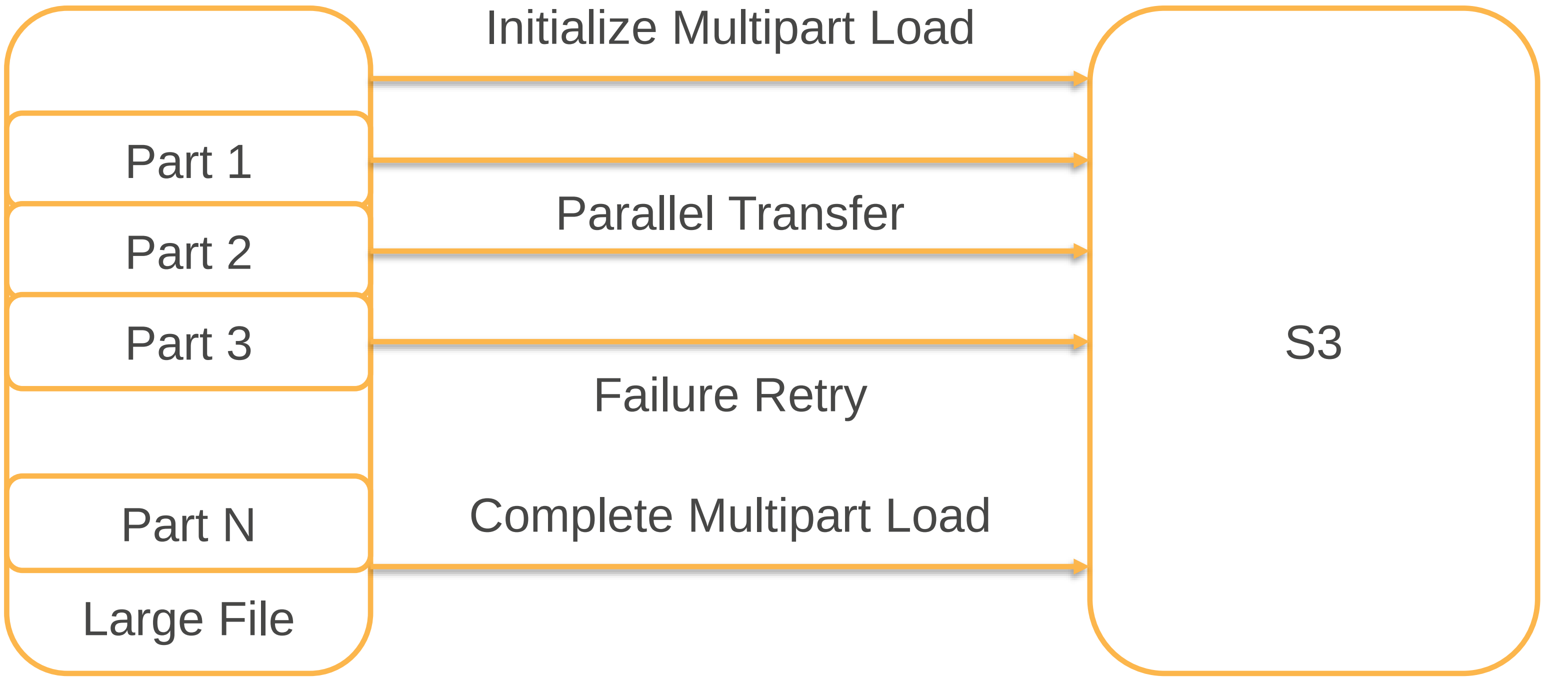
Multi-part upload

- Max size for a single object is 5 TB
- Max upload in a single PUT is 5 GB

Question: How to handle transfer failures, improve performance when transferring large objects?

- Use multi-part transfer for PUT and GET
- Recommended for objects > 100 MB

Multi-part Upload



Initialize Multipart Load

Part 1

Part 2

Parallel Transfer

Part 3

S3

Failure Retry

Part N

Complete Multipart Load

Large File

S3 combines individual parts into one object

Multipart Upload, Download Support

- AWS S3 CLI – automatic multipart upload, download

  Example: `aws s3 sync`

  "Recursively copy new and updated files from source to the destination"

- AWS SDKs support multipart upload and download

- Begin upload even before you know the final size – upload as data is available

S3 Prefix

- S3 is a distributed cluster that scales automatically to support traffic
- For high request rates (1000s of GET/PUTS per second) – ensure object prefix (part of key) is different
    - Workload is distributed across available clusters
    - S3 Data Lake applications can scan millions or billions of objects for queries on petabyte datasets
    - Social media need consistent small object latencies in 100s of milliseconds

Prefix - Log File Scenario

Common Prefix – Does not scale for 1000s of request/second

```
Key=/2020/03/01.log
```

Random Prefix – Scales for 1000s of request/second

```
Key=/RandomPrefix/2020/03/01.log
```

High Transfer Rates

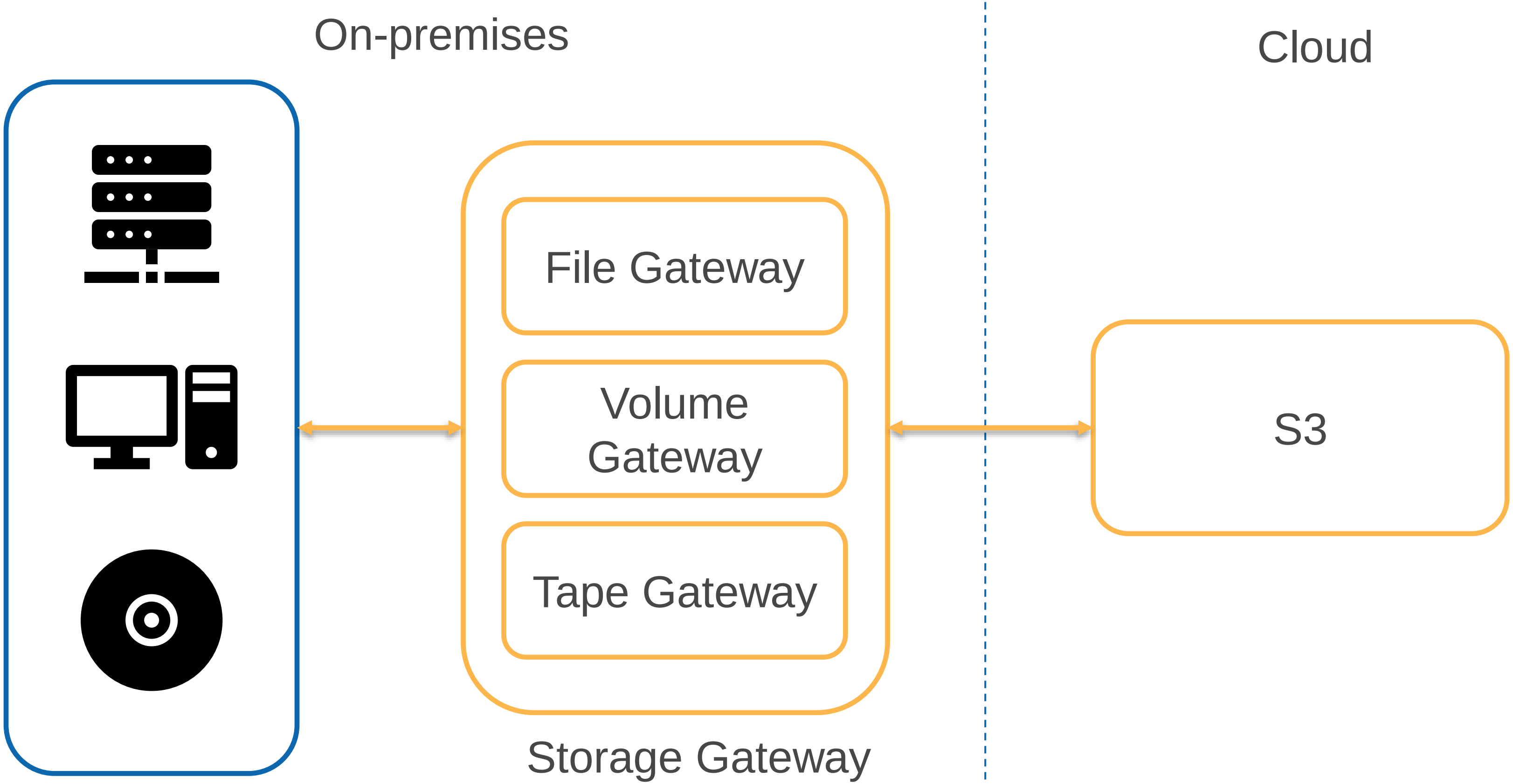
- Cache at the edges using CloudFront content delivery network
- Transfer Acceleration (for uploads to S3 bucket from all over the world). Uses CloudFront Edge network
- ElastiCache in-memory cache with single digit millisecond latency
- Read only required data using Byte-Range Fetches (instead of copying entire object)
- Combine Compute (EC2), Storage (S3) in same region

Batch Operations

Use S3 Batch Operations to work with large number of objects (in 1000s to billions)

- Copy objects between buckets
- Restore archived objects from glacier
- Run custom logic (using Lambda) on a list of objects
- Replace object tags
- Modify Access controls

Storage Gateway



Volume Gateway

Cached Volume Mode

- Primary storage is S3
- Gateway maintains a local cache of recently accessed data
- Minimizes storage footprint on-premises

Stored Volume Mode

- Entire volume is available locally in the gateway
- Asynchronous copy is maintained in S3
- Requires more storage on-premises

Storage Gateway

File Gateway – Objects written through file gateway can be directly accessed in S3

Volume Gateway – Data on volumes (iSCSI) is stored in S3 and you can take EBS Snapshots to create new storage gateway volumes or EBS volumes

Tape Gateway – Virtual tape data (iSCSI) can be stored in S3 or archived in Glacier. Access using Tape Gateway APIs

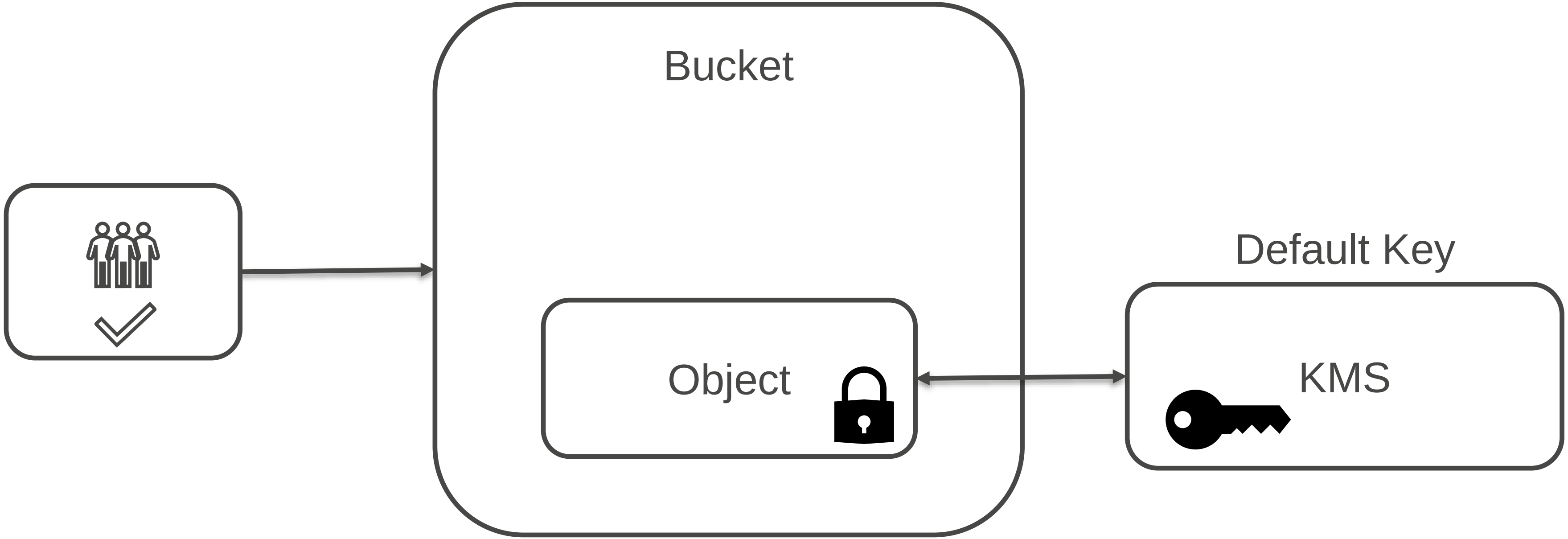
Encryption

Server-Side Encryption, Client-Side Encryption

Server-Side Encryption (SSE)

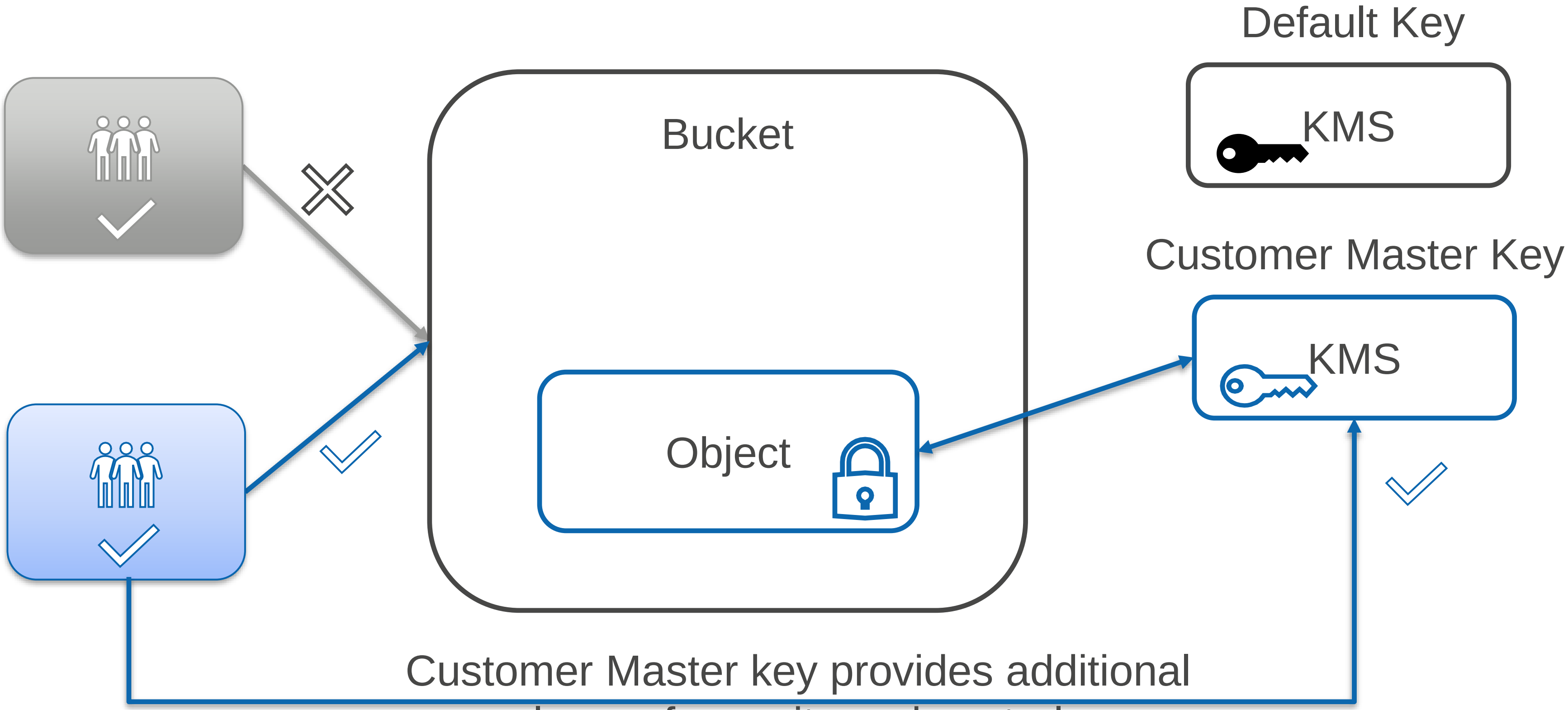
- Encrypt data at rest (AES-256)
- S3 does the encryption and decryption for authorized users
- Three options – based on how keys are managed
    - SSE-S3 (S3 manages the key)
    - SSE-KMS (S3 uses the key you specify in KMS)
    - SSE-C (S3 uses the key you provide with every request)
- Control at individual object level
- Apply at Bucket level (SSE-S3 or SSE-KMS)

SSE-S3 (S3 managed key)



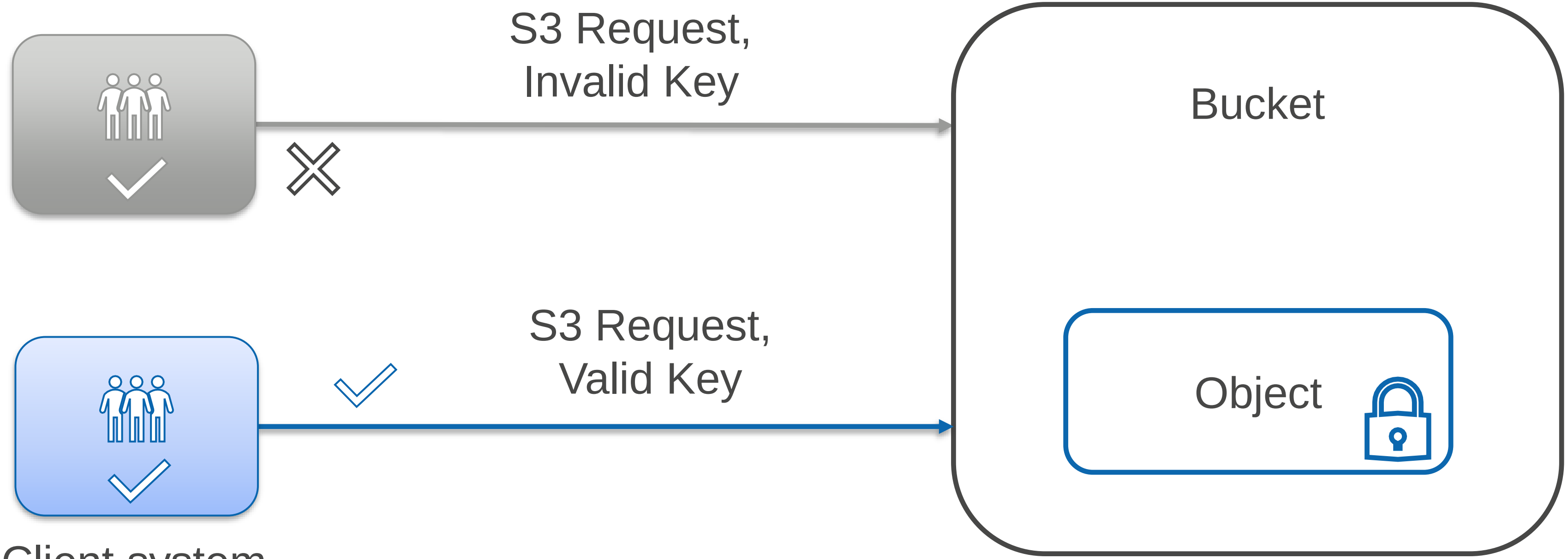
With default key, S3 automatically decrypts object for any user who is allowed access to the bucket or object

SSE-KMS (Customer Master Key in KMS)



Customer Master key provides additional layer of security and control

SSE-C (S3 uses key provided in the request)



S3 Request,
Invalid Key

Bucket

S3 Request,
Valid Key

Object

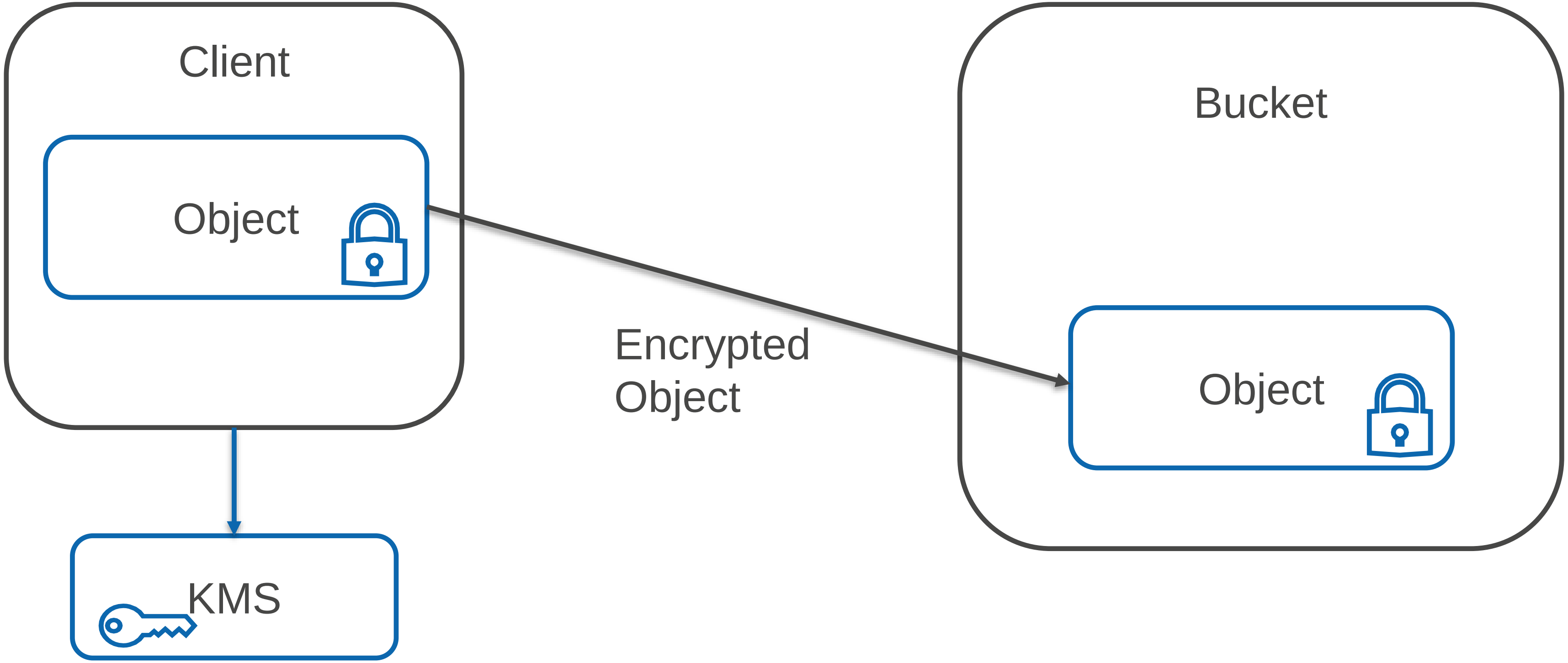
Client system maintains the keys

S3 Encrypts, Decrypts Objects using customer provided key

	SSE-S3	SSE-KMS	SSE-C
Master Key	S3 managed	Customer managed in KMS	Customer maintains the key in their own system
Data Key	Unique data key for each object		Customer provides the data key with every object request
Data Key Security	Data key is encrypted with master key and stored with object (envelope encryption)		Data Key is not stored by S3. Salt derived from Data key is stored to validate future requests
Encryption Instruction Header	s3:x-amz-server-side-encryption:AES256	s3:x-amz-server-side-encryption:aws:kms	x-amz-server-side-encryption-customer-algorithm:AES256
Additional Header		s3:x-amz-server-side-encryption-aws-kms-key-id:<ARN for KMS Key>	x-amz-server-side-encryption-customer-key:<Base 64 encoded 256 bit key> x-amz-server-side-encryption-customer-key-MD5:<Hash for the key>
Key For GETs	Not Required		Same Key as above along with MD5 Hash

S3 Client-Side Encryption

Object encryption and decryption is client responsibility



Manage keys with KMS or use your own system

Other Features

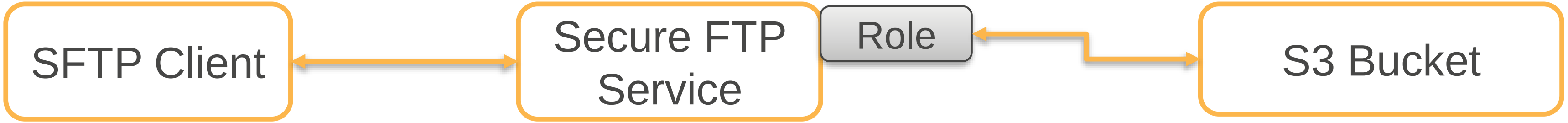
SFTP, Static Website Hosting, CORS, Pre-signed URL, S3 Select, Glacier Select, Amazon Macie, Object Lock

Secure FTP

Managed Secure FTP Service

Transfer files into and out of S3 using SFTP

Use existing FTP Clients and authentication (AD, LDAP, or manage users in SFTP service)



Static Website Hosting

- Use S3 as a webserver
- Host Static Website
- Single Page Web applications with Client-side scripts

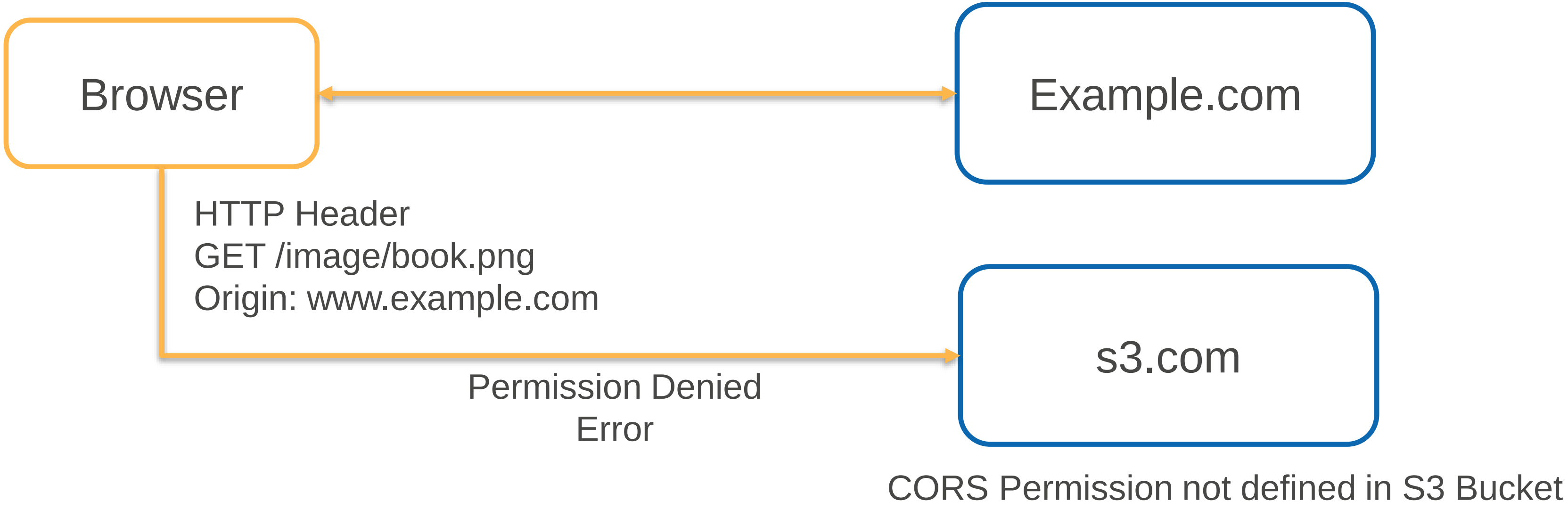
Cross-Origin Resource Sharing (CORS)

Use S3 for managing images, scripts, media for your web application

Browser operates in a sand-box – allows only interactions with the same domain (www.example.com)

Resources in S3 are accessed using a different domain (…amazonaws.com)

CORS – Not Configured



Browser

Example.com

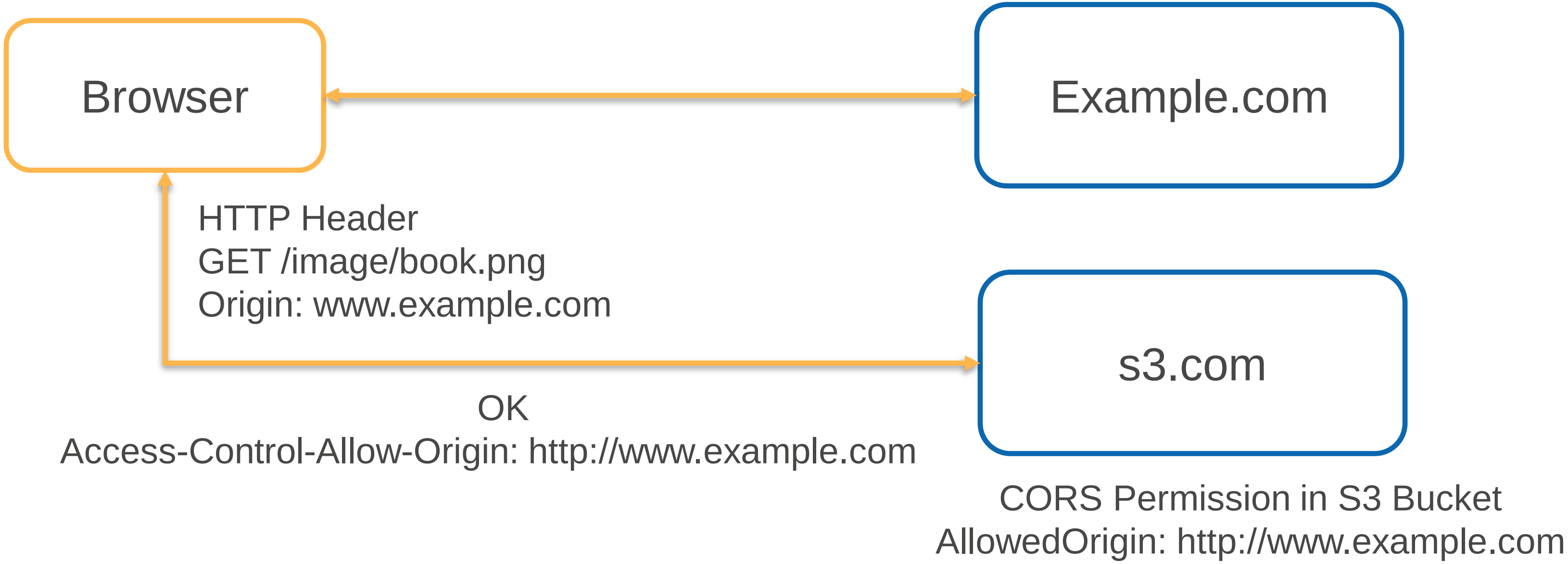
HTTP Header
GET /image/book.png
Origin: www.example.com

s3.com

Permission Denied
Error

CORS Permission not defined in S3 Bucket

CORS - Configured



Browser

Example.com

HTTP Header
GET /image/book.png
Origin: www.example.com

s3.com

OK
Access-Control-Allow-Origin: http://www.example.com

CORS Permission in S3 Bucket
AllowedOrigin: http://www.example.com

CORS Support in Browsers

Modern browsers support Cross-Origin requests (CORS)

Browser includes origin in the HTTP requests when making request to another domain (amazonaws.com)

Domain server can confirm if that origin is allowed access to resources

Pre-signed URL

Share an object with others using presigned URL

Uses:

- Grant limited time permission to download or upload an object
- Third party can access the resource in a private bucket

Example: Pre-sign with AWS CLI

Make sure your computer clock time is correct - Otherwise, signatures may not be valid (if your clock is in the future or too far in the past)

Reference: https://docs.aws.amazon.com/cli/latest/reference/s3/presign.html

```
aws s3 presign s3://chandra-s3-demo/sample.txt --region us-
east-2 --expires-in 300
```

S3 Select and Glacier Select

S3 Select - “Retrieve only a small subset of data from an object using SQL”

Glacier Select – Query archived data using SQL to retrieve only what is needed

Reference: https://aws.amazon.com/blogs/aws/s3-glacier-select/

Amazon Macie

“A machine learning-powered security service to discover, classify, and protect sensitive data [stored in S3].”

Example: detect high risk documents shared publicly or to the entire company

- Personally identifiable information (PII)
- Protected health information (PHI)
- Intellectual property (IP)
- Legal or financial data

https://aws.amazon.com/macie/faq/

Object Lock

- S3 now supports Object Lock (like Glacier Vault Lock)
- Meet regulatory requirements that require WORM Storage (write-once-read-many)
    - Prevent an object from overwritten or deleted
- Two ways to manage:
    - Retention period – specify an object lock time period
    - Legal Hold – No expiration date. You must explicitly remove a legal hold to delete objects

Lab – Storage Classes

Setup bucket

Store objects in Standard, Infrequent Access and Glacier Storage Classes

Observe retrieval behavior

Lab - Versioning

Enable Bucket Versioning

Store Object

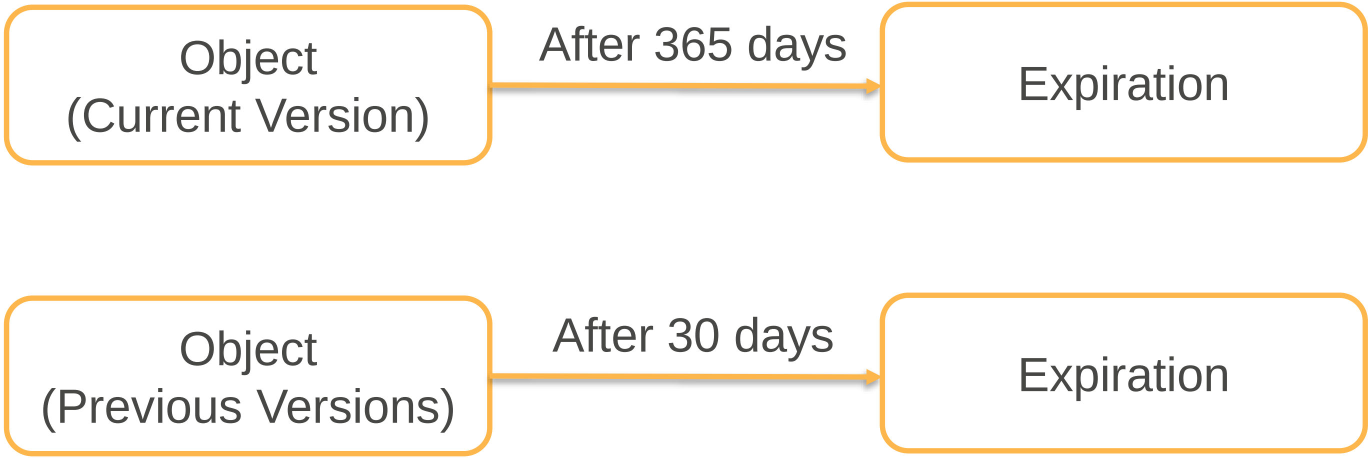
Track Versions

Delete Object

Undelete Object

Lab – Age based retention

Setup expiration policy to delete objects based on age



Object
(Current Version)

After 365 days

Expiration

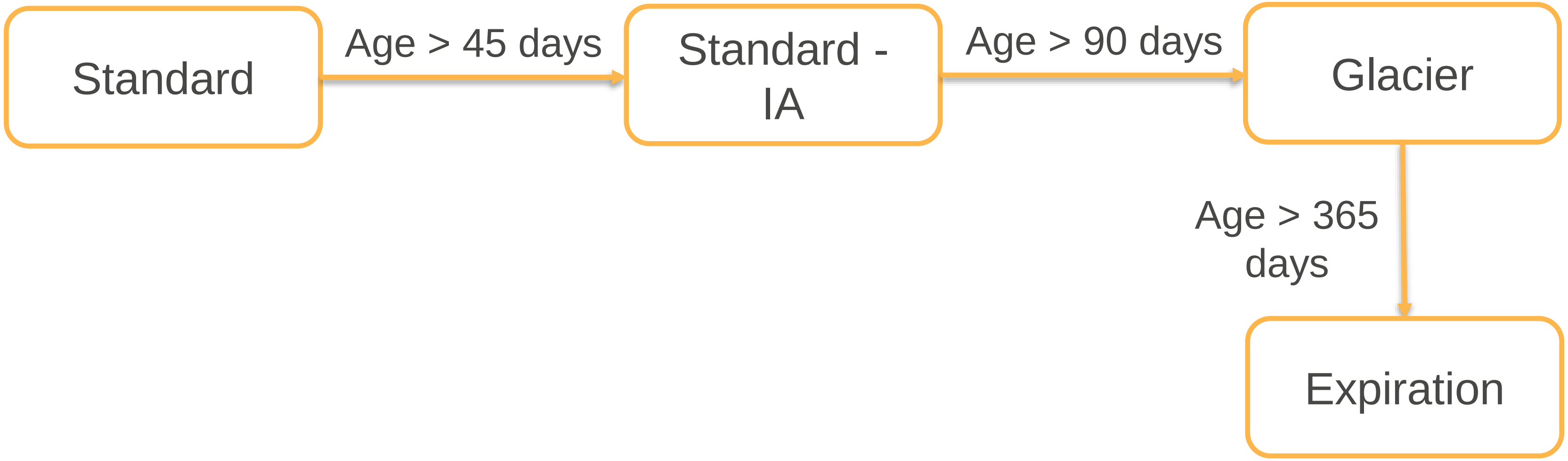
Object
(Previous Versions)

After 30 days

Expiration

Lab – Tiered Storage and Expiration

Scenario: Frequently accessed for 45 days. Usage drops after that, but object needs to be immediately accessible for 90 days. After 3 months, customer can wait for few hours to retrieve data. Optimize cost while meeting data access requirement



Standard

Age > 45 days

Standard - IA

Age > 90 days

Glacier

Age > 365 days

Expiration

Lab – Cross Region Replication

Automatically replicate data from source to destination bucket

Existing objects are not replicated (use batch or reupload object)

Lab – SSE-S3 and KMS

Encryption using SSE-S3

Encryption using SSE-KMS

Lab – Pre-signed URL

Covered as part of pre-signed URL lecture