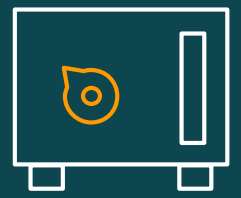
Best Practices for Protecting Data Using Encryption

Jeremy Stieglitz
Principal PM AWS KMS

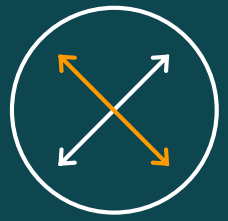
Why use Data Protection on AWS?



Protect intellectual property and trade secrets

Protect customer information and build a trusted brand

Automate tasks to save time and reduce risk



Scale with visibility and control as your business grows

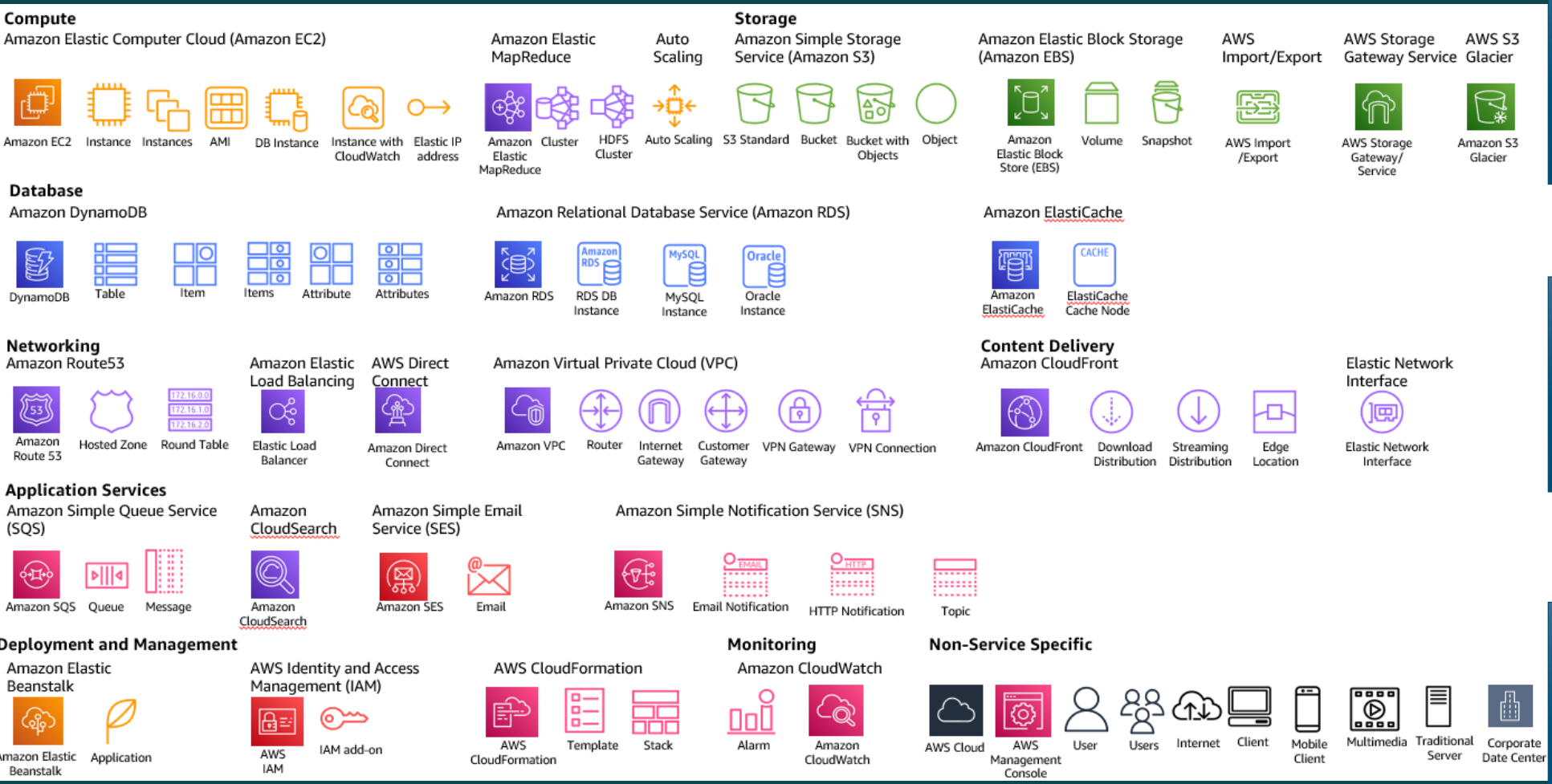
Ease of use - integration with hundreds of AWS services

Inherit global security and compliance controls

AWS Cryptography Stack

Focus of my talk will be on application and “at-rest” data protection

Mapping Data Encryption at AWS



In 2020, AWS set a goal for KMS integration in all AWS services that manage customer data

Might be easier to list services that do not support encryption.

Today 103+ AWS services have data encryption supported by KMS

Data protection terms

Envelope Encryption

The practice of encrypting plaintext data with a data key, and then encrypting the data key under a KMS key

KMS Key (Envelope Key)

Protects data keys. Uniquely identified by an ARN. Sits at the top of your key hierarchy

Data Key

Data keys are encryption keys that you can use to encrypt your plaintext data

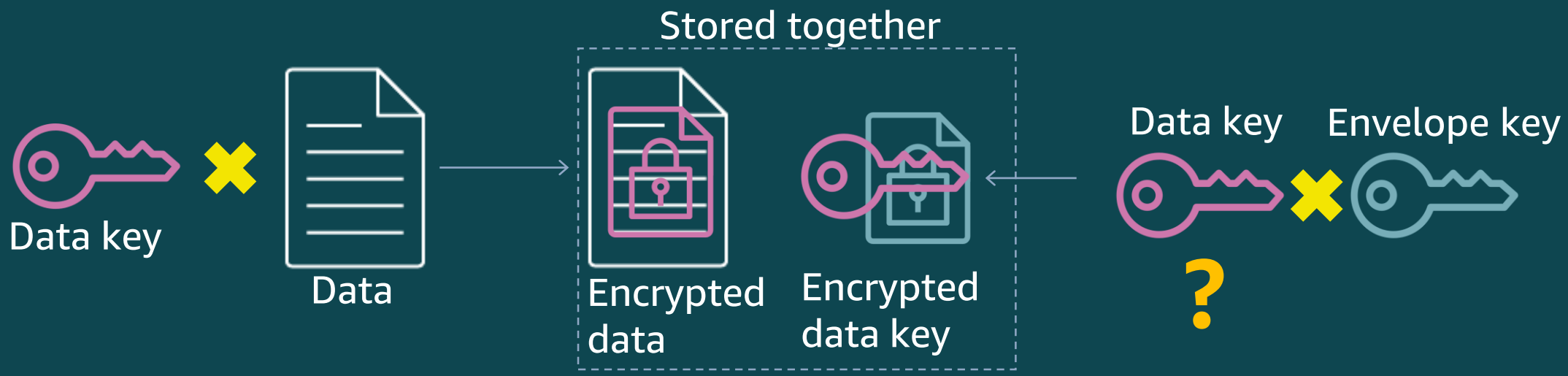
Plaintext

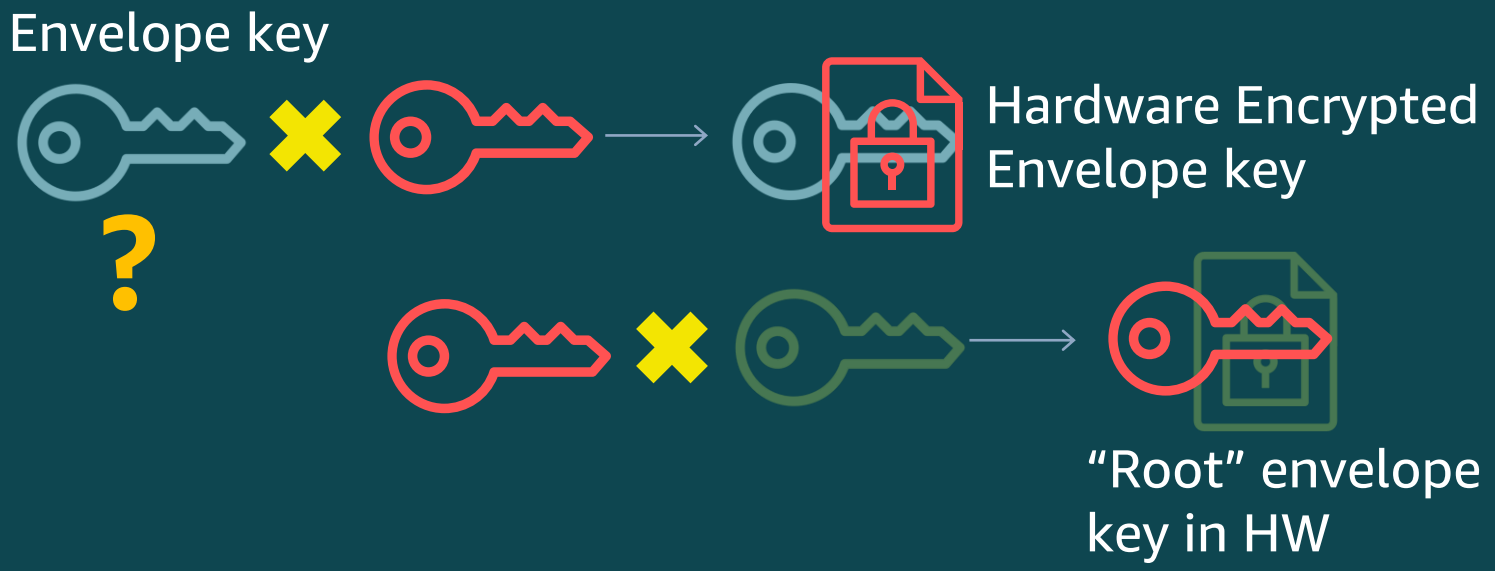
Unencrypted information that you wish to protect, pending input into cryptographic algorithms

Ciphertext

Encrypted information unreadable by a human or computer without decryption

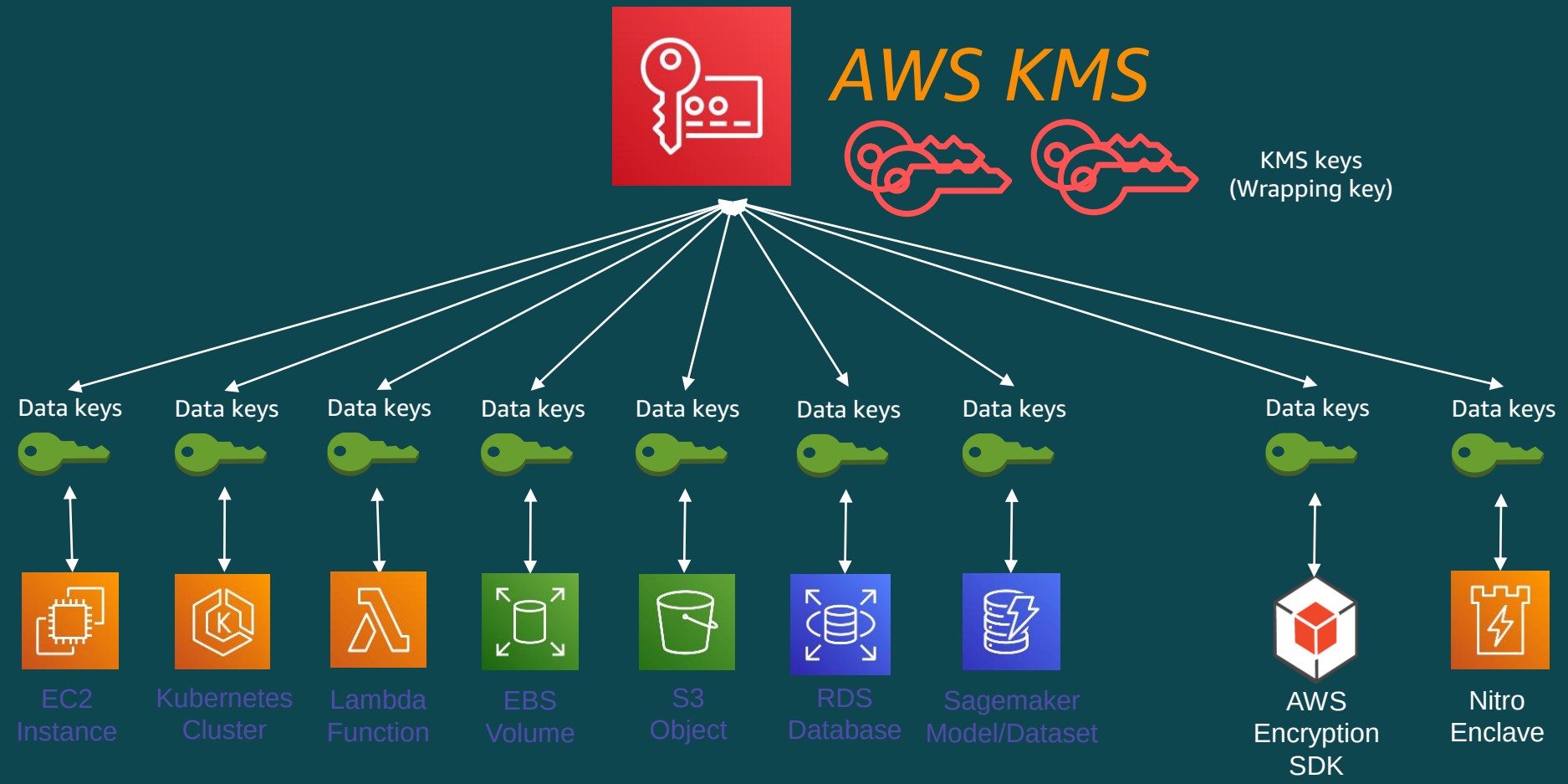
Envelope encryption





AWS KMS
(Root of trust)

Key Hierarchies



Can AWS see my keys?

- Keys never evist in plaintext outside the HSM
- When operational with keys provisioned
  - No AWS operator can access the HMS (no human interfaces)
  - No software updates allowed (must tear down HSM to blank)
- After reboot and in a non-operational state
  - No key material on host
  - Software can only be updated after multiple AWS employees have reviewed the code
    - Under quorum of multiple AWS KMS operators with valid credentials
- Third-party evidence
  - SOC 1 – Control 4.5: *Customer master keys used for cryptographic operations in KMS are logically secured so that no single AWS employee can gain access to the key material.*

Part II: Data Protection Best Practices

Agenda

- Setting the Stage
- Best Practices in Data Protection
  - Protection Tips
  - Performance Tips
  - Policy Tips

Protection Best Practices

Protection Best Practices

Customer managed keys provide the most flexibility

Limit key rotation or consider a key rollover regime

Use Encryption Context

Highly sensitive workloads can be data protected in AWS Nitro Enclaves

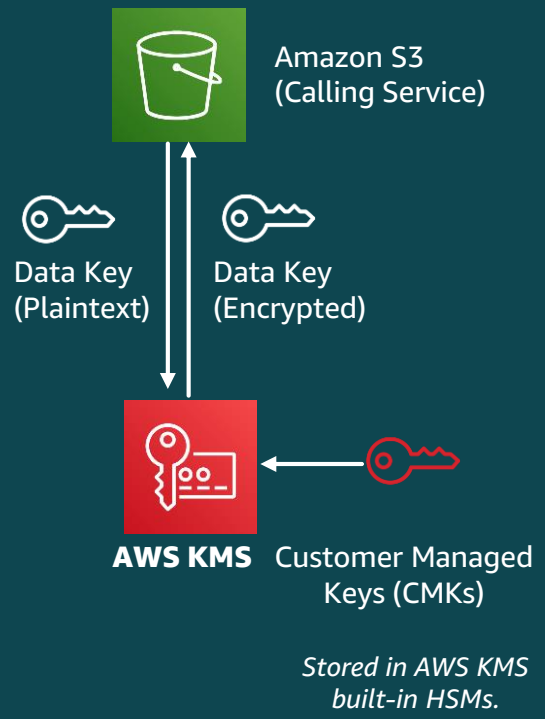
Start and keep keys in KMS (reduce use of Import Keys or Custom Key Store)

#1. Customer managed keys give you most flexibility

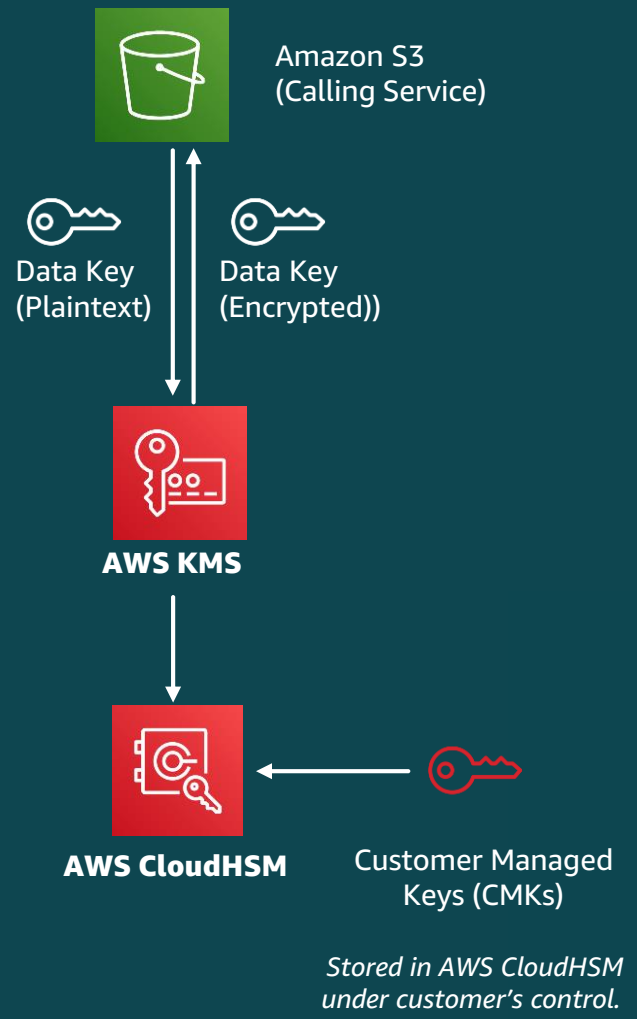
	Customer Managed Key	AWS Managed Key	AWS Owned Key
Policy	Customer	AWS	AWS
Audit	Customer CloudTrail	Customer CloudTrail	None
Rotation	Customer	Up to each service	Fixed @ 3 Years
Copy to other regions?	Yes (For MRK keys)	No	Yes (Limited)
Cost	$1 per month	Charges on API calls	No cost on key or API calls

#2. Start and Keep Keys In KMS

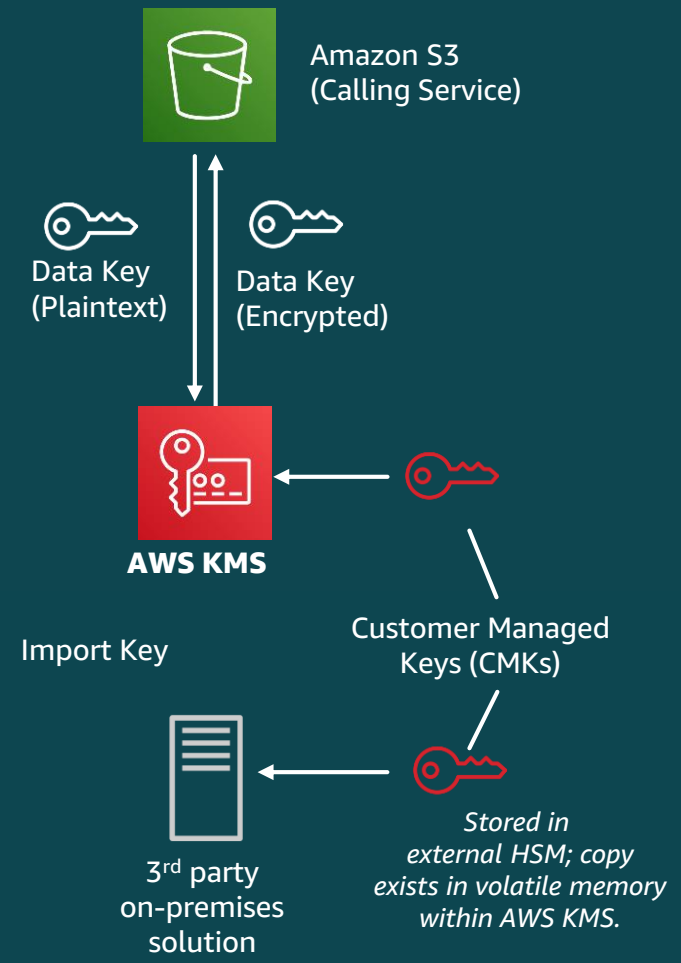
AWS KMS only



AWS KMS with custom key store



AWS KMS with imported keys



#3. Limit key rotation or consider a key rollover regime

- If you are not required to, don’t do it.
- If you are required to show rotation, consider moving to new keys on a frequency (yearly).
  - KMS Aliases and ABAC greatly facilitates
  - Replace Key ARNs with Key Aliases as your primary way you manage key policies for keys

#3. Common drivers for key rotation

1. ~~Protect my blast radius if keys leak~~
2. ~~Crypto Wear-Out~~
3. Auditors, Compliance NIST guidelines

#4. Use Encryption Context

- **Non-secret, plaintext** additional information
- Key value pairs: Billing:Repair_Invoices
- Should be **relevant to the data**
- KMS will “fold” this data into the ciphertext and then becomes a requirement for successful decryption
- Included verbatim in AWS CloudTrail logs
- Can be used as conditions in IAM policies, Key Policies, and Grants
- Helps prevent confused deputy attacks
- Can also help your master data management (timestamps for data deletion)

```
{
  "awsRegion": "us-east-2",
  "eventName": "Decrypt",
  "eventSource": "kms.amazonaws.com",
  "eventTime": "2017-09-15T19:35:54Z",
  "requestParameters": {
    "encryptionContext": {
      "TenantID": ”123AID",
      "OrderDate": "2018-09-01",
      "OrderID": "123-4567890-011",
      "Type": "Invoice"
    }
  },
  // ...
}
```

#5. Consider encrypting data in AWS Nitro Enclaves

- Isolated, hardened and highly constrained virtual machine
- Processor agnostic, can scale to the size of any EC2 instance
- No external networking interface; only a secure vsock channel with parent instance
- Independent OS kernel from the parent compute instance
- Cryptographic attestation and ability to decrypt data with data keys that are unavailable outside the enclave
  - Integration with AWS KMS and AWS Certificate Manager for TLS offloading
- *No instance administrator or AWS operator access to code or data running inside the enclave*

Performance Best Practices

Performance Best Practices

Several orders of magnitude in how encryption operates at AWS

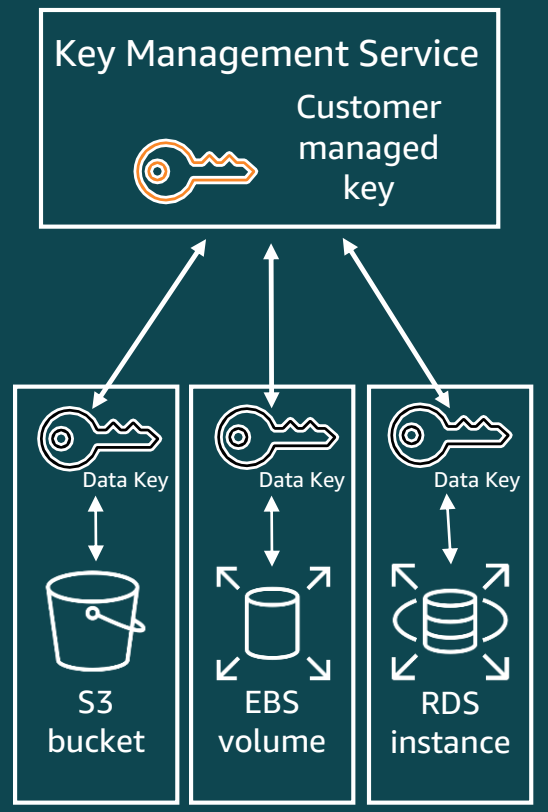
Bucket Keys can dramatically improve protection performance

Use data caching for client-side encryption with the AWS Encryption SDK

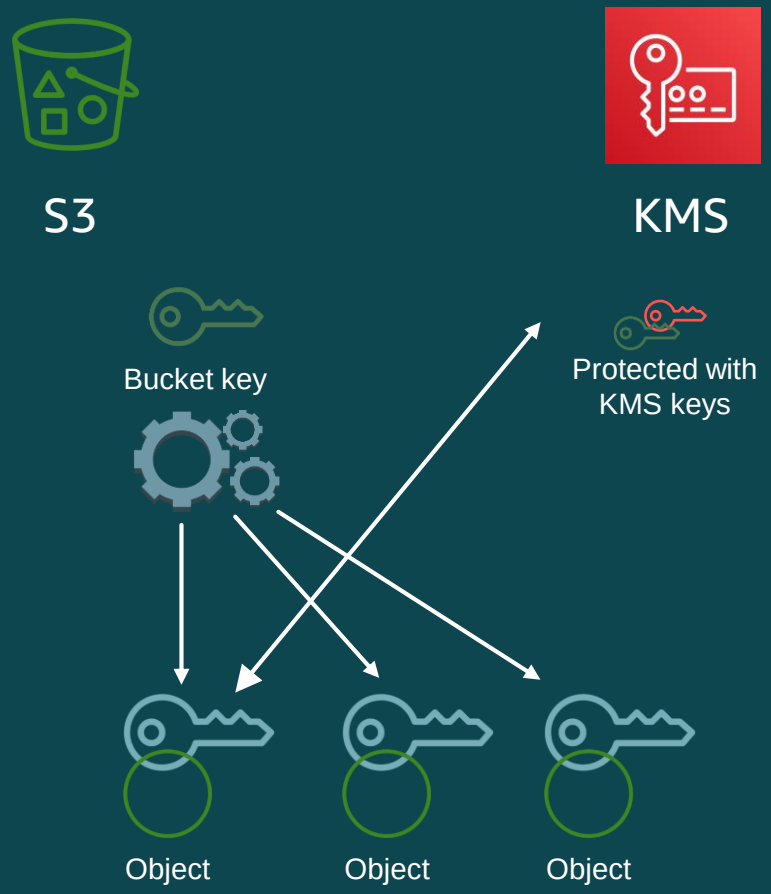
Use Multi-Region keys for global database workloads in Amazon DynamoDB

#1. Several orders of magnitude in how encryption operates at AWS

- Vast majority of AWS services create data keys to protect resources
- # of data keys can vary greatly
  - 1 data key per EBS Volume
  - 1 data key per RDS Instance
  - 10s of data keys for Lambda (10s or 100s of keys)
  - 1000s of data keys for AWS Secrets Manager
  - Billions of data keys for S3 (1 data key per object)



#2. S3 Bucket Keys deliver10-1000x KMS improvement



- Historically, every object in S3 requested a data key provided by KMS
- This can impact performance and generate high cost on billions of objects
- Bucket keys are an *intermediate* key
- S3 requests bucket keys from KMS & uses these keys to *derive* data keys
- Individual results will vary, but at scale, we've seen 100x improvement

#3. Using data key caching in the ESDK

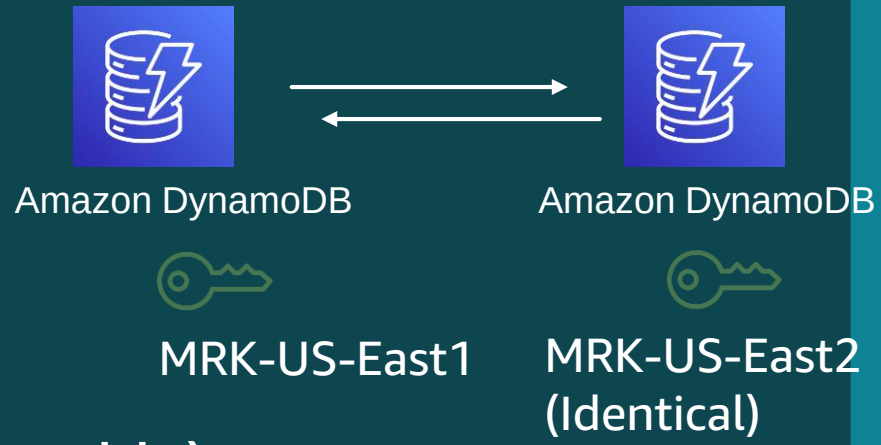
- Data key caching stores data keys and related cryptographic material in a cache
- When you encrypt or decrypt data, the AWS Encryption SDK looks for a matching data key in the cache. If it finds a match, it uses the cached data key rather than generating a new one.
- Data key caching can improve performance, reduce cost, and help you stay within service limits as your application scales.

Steps:

1. Create the data key cache
2. Create a key ring
3. Create a caching CMM
4. Set cache security thresholds

#4. Reduce latency in Amazon DynamoDB Global Tables With new KMS multi-Region keys

- The challenge: How do I protect records in DynamoDB Global Tables?
  - Cannot make cross-region calls (high latency)
  - Cannot “wrap” data with multiple envelopes (not always known which regions read/write to the table)



- Answer: KMS now supports multi-region keys
  - Copy keys into multiple regions to avoid cross-region calls or. the need to “wrap” multiple envelopes
  - Removes the need to decrypt and re-encrypt for replication

Key Policy Best Practices

Key Policy Best Practices

Use separation of duties to enforce least privilege

Attribute Based Access Control (ABAC) brings new flexibility in writing key policies

Use VPCE policies to lock down keys to network context

Service control policies can create org. level boundaries

#1. Enforce separation of duties in KMS

Use IAM policy and CMK Key Policy to enable different personas to perform different key management tasks

IT security

Manages key access policies

Uses keys to protect data

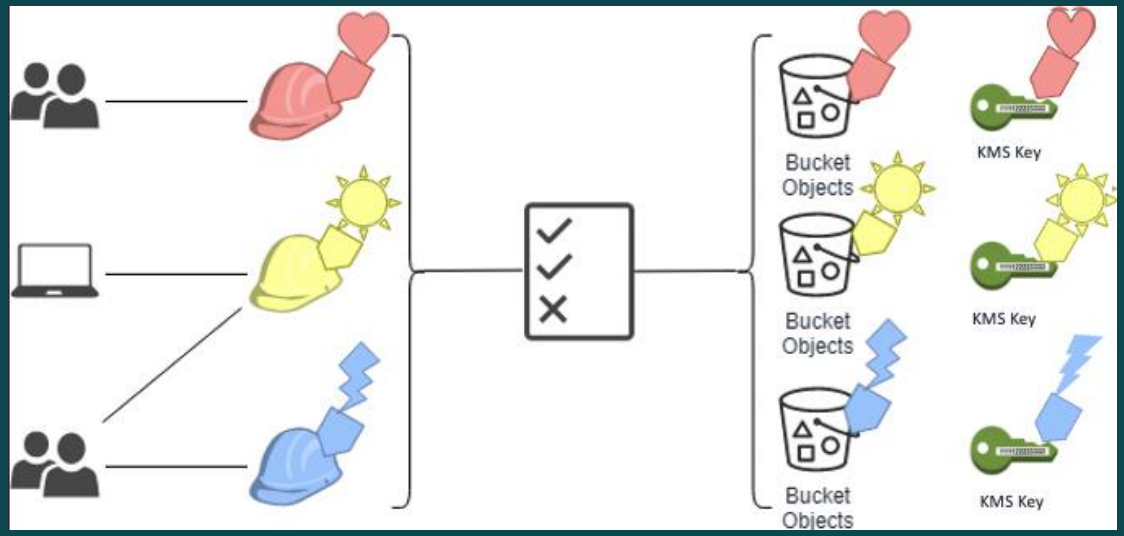
Compliance

Verifies configuration and historical access

#2. Attribute Based Access Control brings new flexibility in writing key policies

- Roles-based access control defines permissions based on a person's role

- *Attribute based* access control *extends this abstraction to resources/services through tagging*



- In this example "Red team" can only decrypt with keys for bucket objects both tagged as "Red"

#3. Use VPCE policies to lock down keys to network context

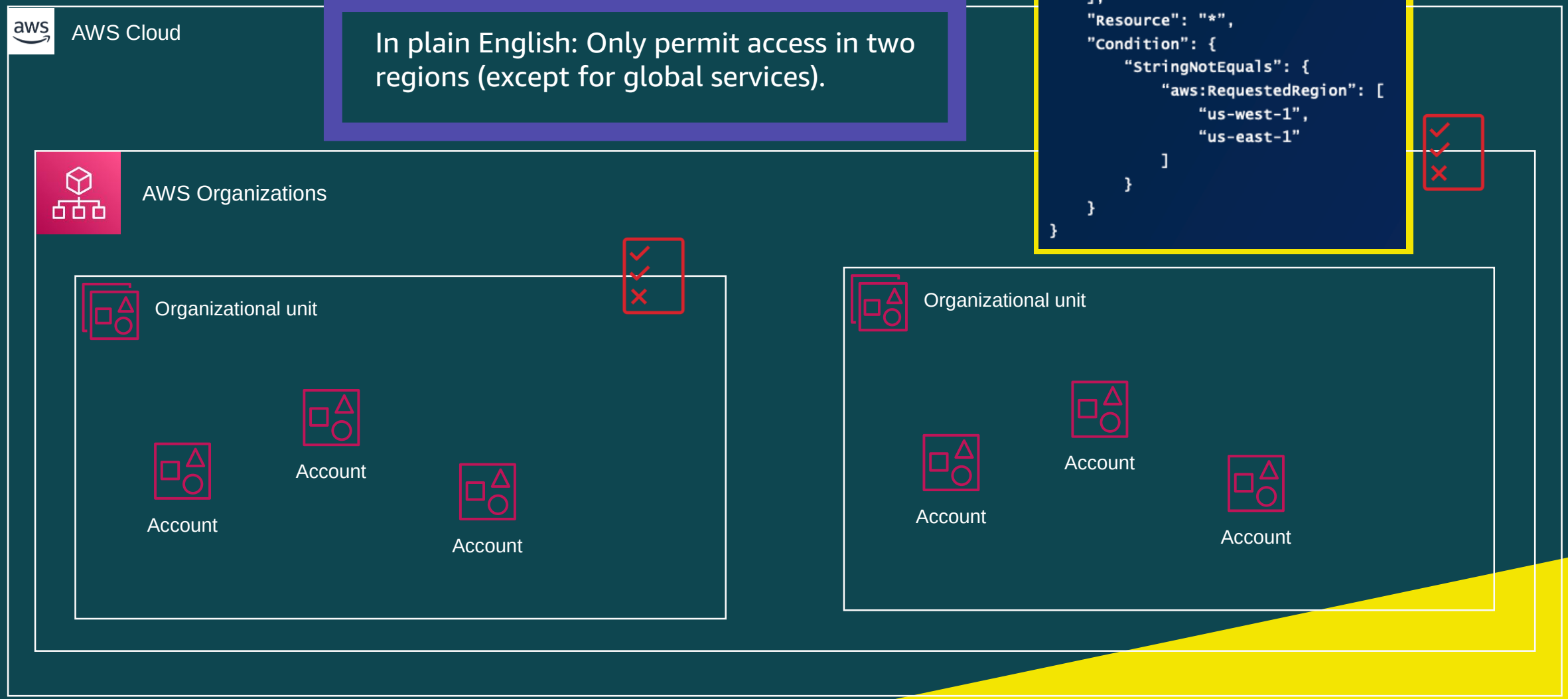
1. Control from “where” you can request keys
2. Protection against insider-risk
3. As granular as you want to get:
    1. Source IP
    2. Time of day
    3. MFA status
    4. viaService
    5. Principle Org.

```
{
   "Statement":[
      {
         "Sid": "AllowDecryptAndView",
         "Principal": {"AWS": "arn:aws:iam::111122223333:user/ExampleUser"},
         "Effect":"Allow",
         "Action": [
             "kms:Decrypt",
             "kms:DescribeKey",
             "kms:ListAliases",
             "kms:ListKeys"
          ],
         "Resource": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
   ]
}
```

#4. Service control policies can create Org. level boundaries

```
{
    "Effect": "Deny",
    "NotAction": [
        “iam:*”,
        “route53:*”,
        …
    ],
    "Resource": "*",
    "Condition": {
        “StringNotEquals”: {
            “aws:RequestedRegion”: [
                “us-west-1”,
                “us-east-1”
            ]
        }
    }
}
```

In plain English: Only permit access in two regions (except for global services).



Closing Thoughts

- Data protection in the cloud has come a long way since KMS launched in 2014:
  - Integrated data protection in 103+ services
  - Increasing degrees of automation and encryption by default
- We've built high degrees of trust/certification of our locks and keys
  - FIPS 140-2 Security Level 3 HSM, ISO/CSA STAR, PCI, FedRAMP High
- You control (and build) the policies that govern control over your keys
  - Newly launched VPCE and ABAC features allow very flexible AND highly specific tools to govern access to keys

Further Viewing

Ken Beer

How encryption works in AWS

https://www.youtube.com/watch?v=plv7PQZICCM

Mark Ryland

Security Benefits of the Nitro Architecture

https://www.youtube.com/watch?v=kN9XcFp5vUM

Raghu Prabhu

Data, S3 Bucket, AWS Account, and Encryption Strategies for Data Lakes on S3

https://www.youtube.com/watch?v=XpTly4XHmqc

Peter O'Donnell

Using KMS for data protection, access control, and audit

https://www.youtube.com/watch?v=hxWvbNvj2lg

Thank you!