Get the best out of Live Sessions

HOW?

e!

Check your Internet Connection

Log in 10 mins before, and check your internet connection to avoid any network issues during the LIVE session.

Speak with the Instructor

By default, you will be on mute to avoid any background noise. However, if required you will be **unmuted by instructor**.

Clear Your Doubts

Feel free to clear your doubts. Use the "**Questions**" tab on your webinar tool to interact with the instructor at any point during the class.

Let us know if you liked our content

Please share feedback after each class. It will help us to enhance your learning experience.

edureka!

edureka!

Microsoft Azure Developer Associate (AZ-204)

COURSE OUTLINE
MODULE 09

- Introduction to Azure IaaS Compute Solutions
- Implementing Azure Batch Service and Disk Encryption
- Designing and Developing Applications That Use Containers
- Implementing Azure App Service Web Apps and Mobile Apps
- Implementing Azure App Service API Apps and Azure Functions
- Developing Solutions That Use Azure Table Storage and Cosmos DB

- Developing Solutions That Use Relational Database and Azure Blob Storage
- Implementing Authentication and Access Control in Azure
- **Implementing Secure Data Solutions and Integrate Caching & CDN**
- Instrument Monitoring, Logging and Scalability Of Apps & Services
- Connecting to and Consuming Azure and Third-party Services
- Developing Event-based and Message-based Solutions in Azure

Module 9 – Implementing Secure Data Solutions and Integrating Caching & CDN

edureka!

Topics

- Azure Key Vault
- Azure Encryption aspects
- Encryption of Data at rest
- Server-side Encryption
- Client-side Encryption
- Key Management with Key Vault
- Encryption in transit
- Azure Cache for Redis
- Azure Redis Cache – Use cases
- Azure Content Delivery Network (CDN)
- Azure CDN features
- Working of CDN

Objectives

After completing this module, you should be able to:

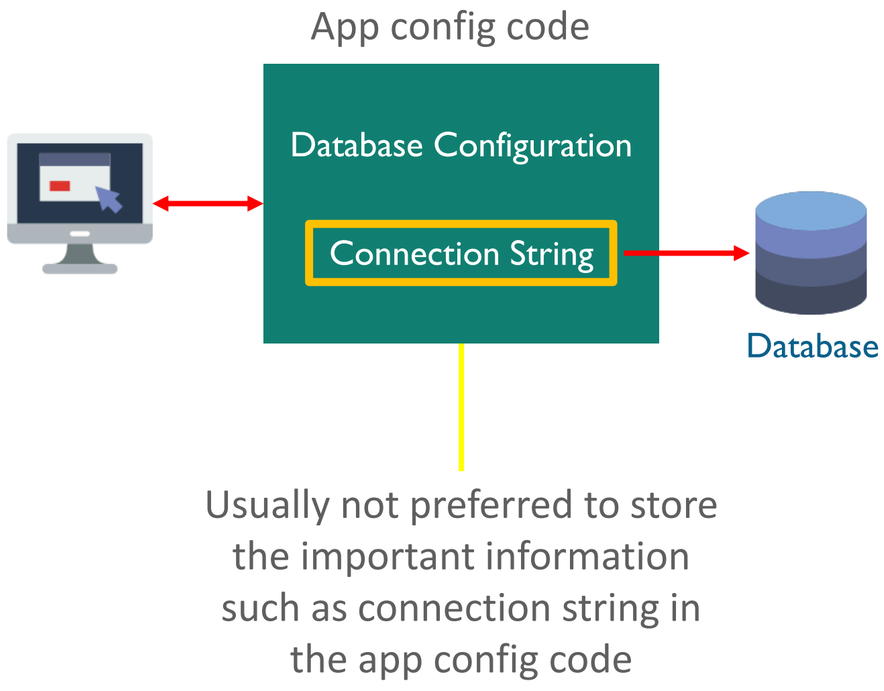
- Understand encryption options
- Learn how to encrypt data with Transparent Data Encryption
- Manage and utilize encryption keys by using the Azure key Vault
- Understand how Azure Cache for Redis operates and how to configure and interact with it
- Know how to manage Azure CDN

Azure Key Vault

edureka!

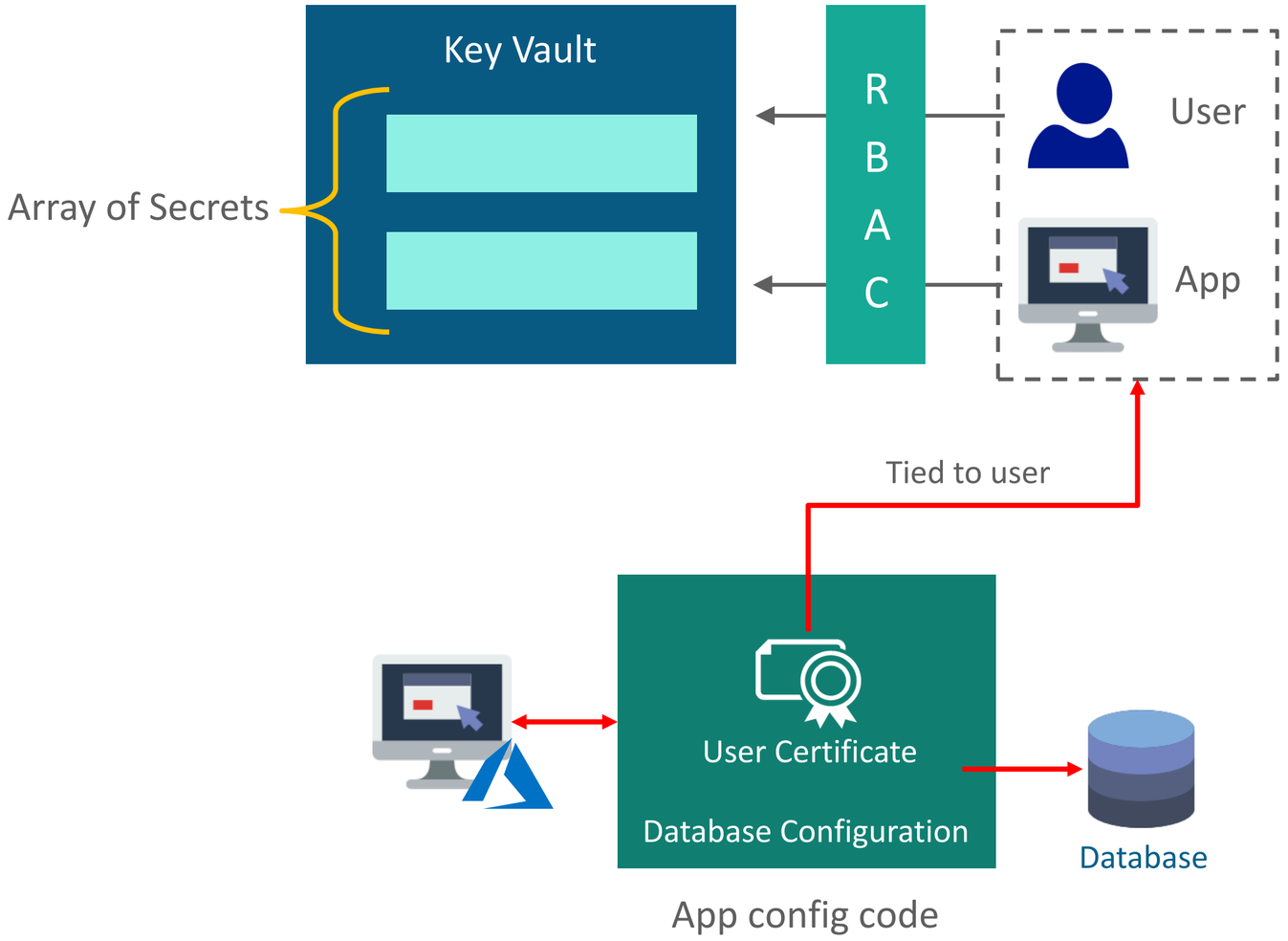
Azure Key Vault – Use Case

- Security plays a major role in case of accessing database
- Here the DB configuration information i.e. connection string is stored in the application config code
- We need to encrypt the complete application config code to secure the DB configuration information
- But the above suggested will not work in case of a web application as the Web app inside Azure has
  - No access to machine keys
  - No low-level access to actual VM



Azure Key Vault – Use Case

- Instead of providing the connection string directly in the application configuration code, we can provide a certificate which gets deployed with the application
- The certificate is tied to a user
- We can apply RBAC policies essentially to the application



Introduction to Azure Key Vault

Azure Key Vault helps solve the following problems:

1. **Secrets Management** - Azure Key Vault can be used to Securely store and tightly control access to tokens, passwords, certificates, API keys, and other secrets

2. **Key Management** - Azure Key Vault can also be used as a Key Management solution. Azure Key Vault makes it easy to create and control the encryption keys used to encrypt your data.

3. **Certificate Management** - Azure Key Vault is also a service that lets you easily provision, manage, and deploy public and private Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates for use with Azure and your internal connected resources.

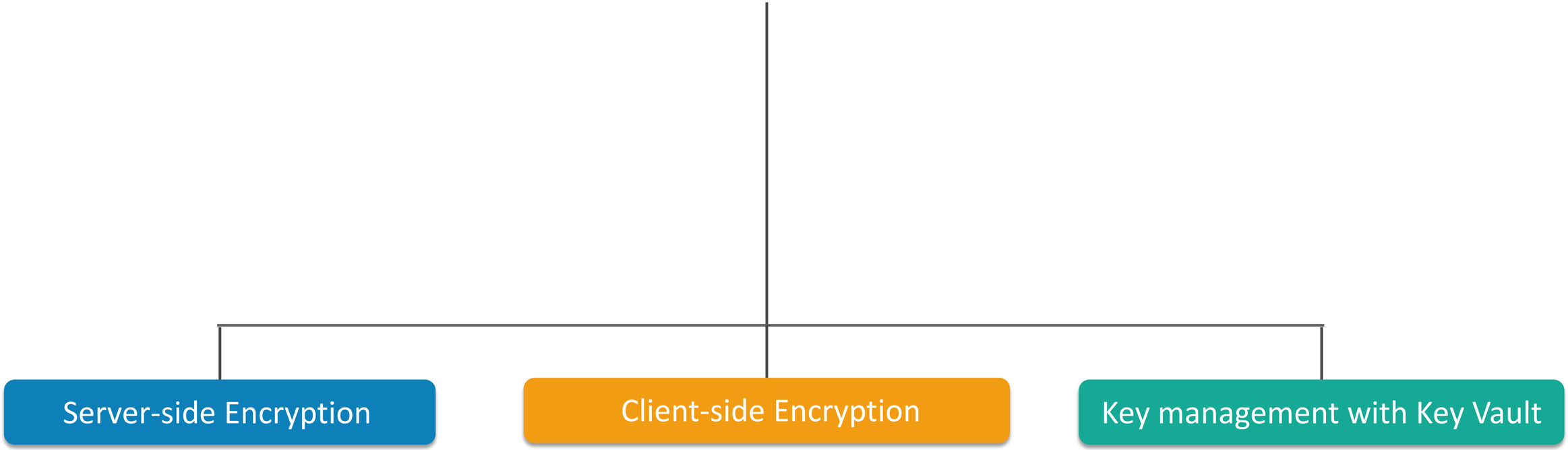
4. **Store secrets backed by Hardware Security Modules** - The secrets and keys can be protected either by software or FIPS 140-2 Level 2 validates HSMs

Azure Encryption Overview

edureka!

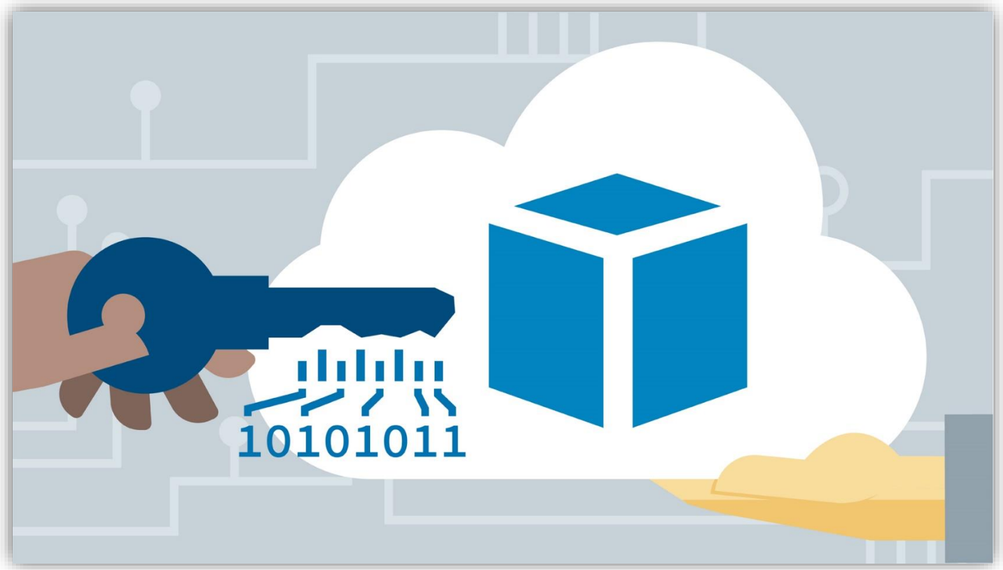
Azure Encryption Aspects

Microsoft Azure supports the below aspects to encrypt data:



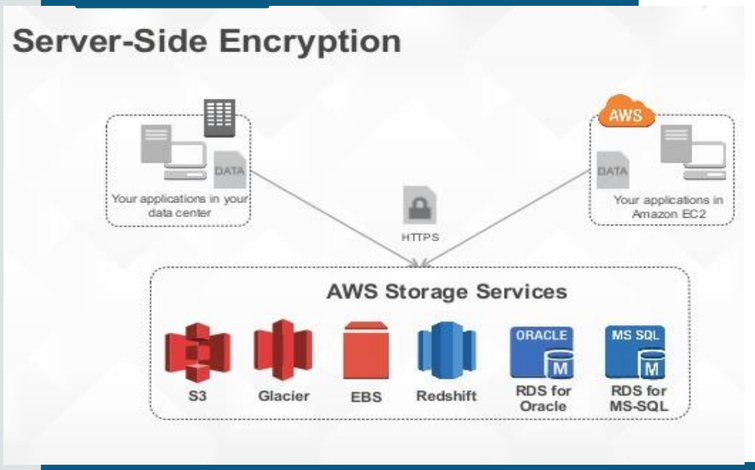
Encryption of Data at Rest

- Data at rest includes information that resides in persistent storage on physical media, in any digital format
- The media can include files on magnetic or optical media, archived data, and data backups
- Microsoft also provides encryption to protect Azure Storage service, Azure SQL Database, Cosmos DB, and Data Lake
- Data encryption at rest is available for services across the SaaS, PaaS, and IaaS cloud models



Server-side Encryption

- For use-case purpose, we will choose Azure Blob storage for encryption, which is a object storage services offering
- Azure supports both server-side and client-side encryption with users having the option of enabling server-side encryption by **default** for all uploaded objects
- In Azure, server-side encryption is called **Storage Service Encryption** when it pertains to blob storage
- Azure leverages envelope encryption using **AES-256 symmetric keys** for data or content encryption (Microsoft uses the term **Content Encryption Key (CEK)** in place of ***Data Encryption Key {DEK}***)
    - It supports using either a symmetric or an asymmetric key for the Key Encryption Key (KEK), depending on who is generating and managing the keys

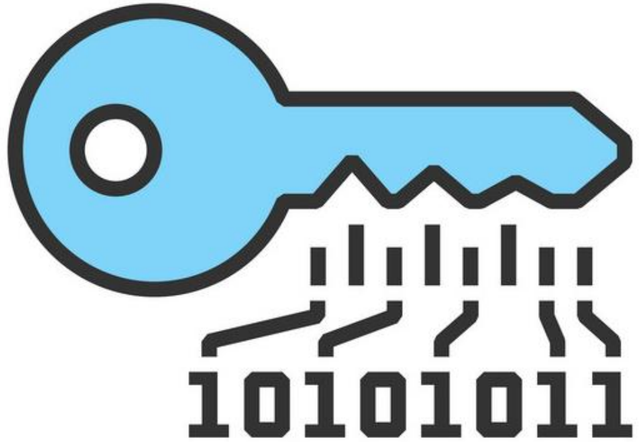


Server-side Encryption

Storage Service Encryption supports using a KEK that is either:

Managed by the storage service itself, using Microsoft's internal key management infrastructure

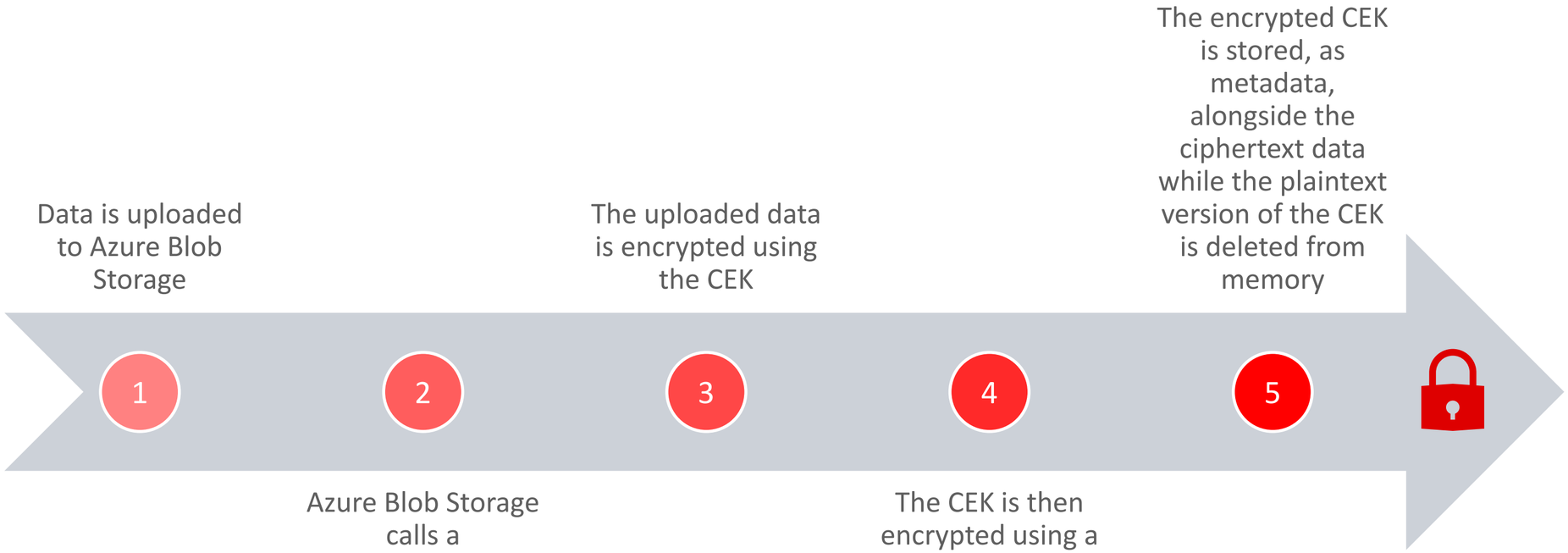
Customer managed and stored in Key Vault, the Azure key management service offering



edureka!

Server-side Encryption

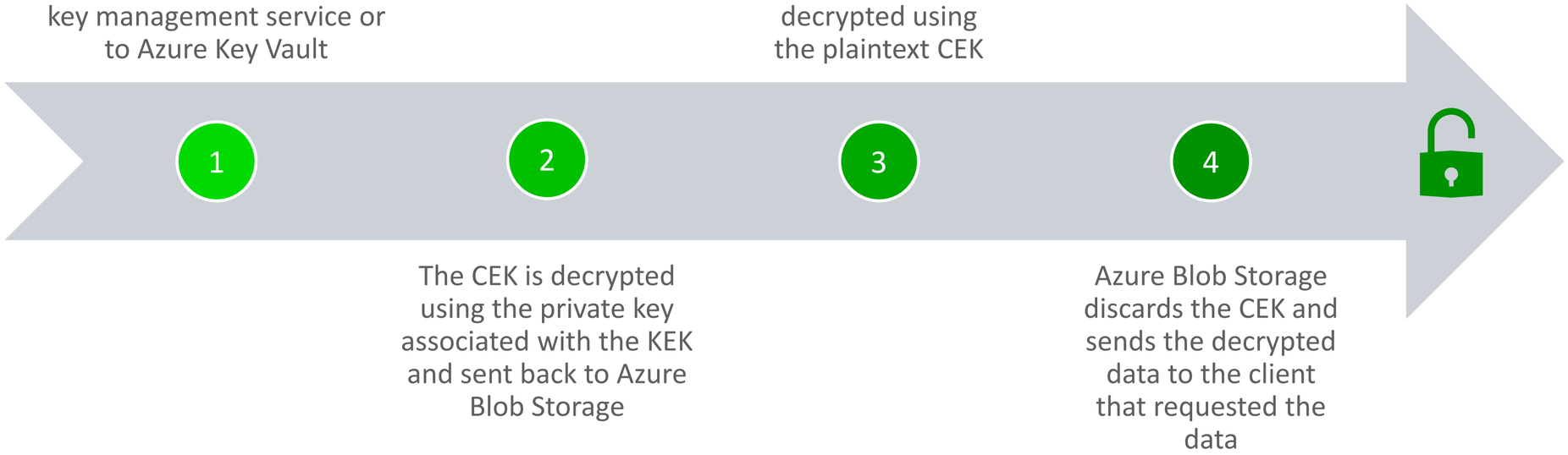
Encryption Workflow For Storage Service Encryption (Server-side)



edureka!

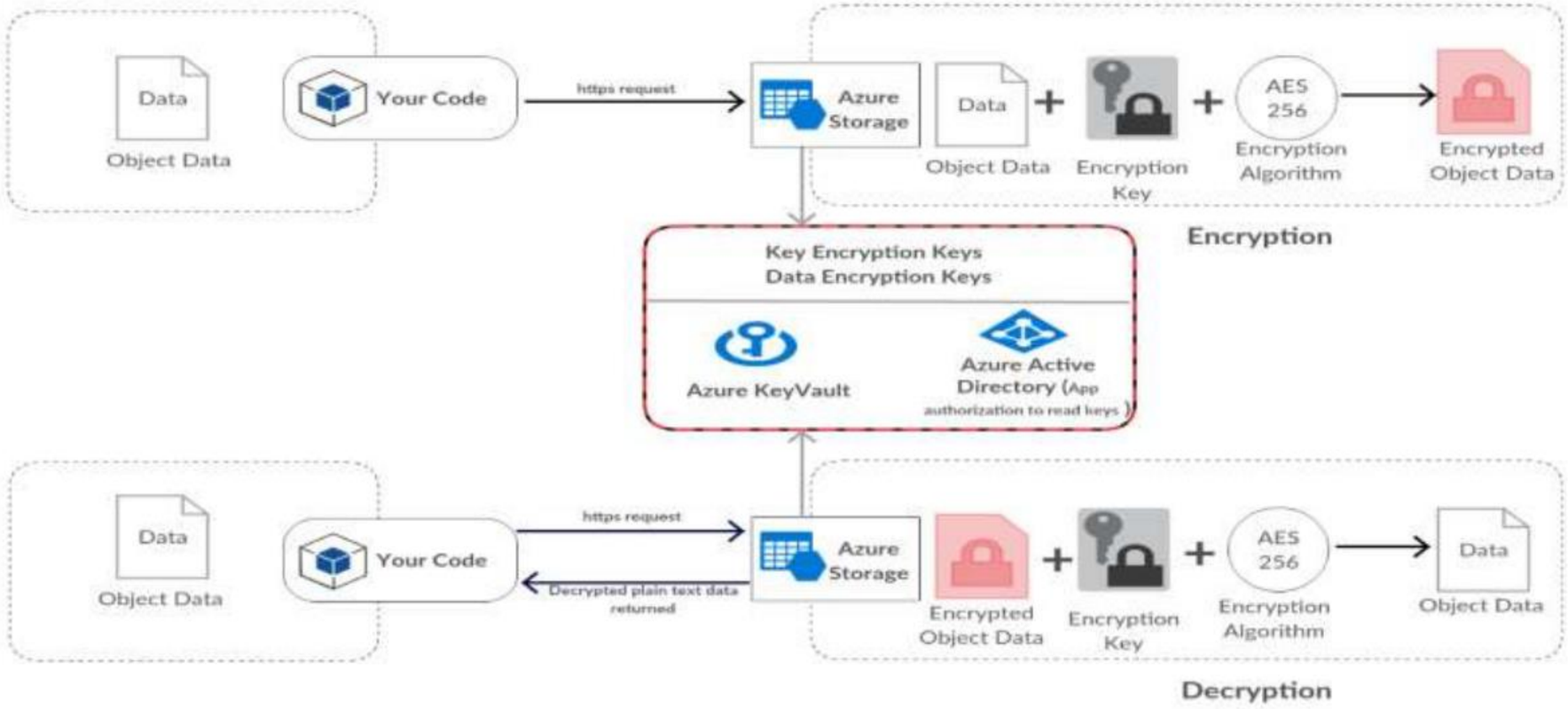
Server-side Encryption

Decryption Workflow For Storage Service Encryption (Server-side)



edureka!

Server-side Encryption



edureka!

Client-side Encryption

Client-side encryption is performed outside of Azure

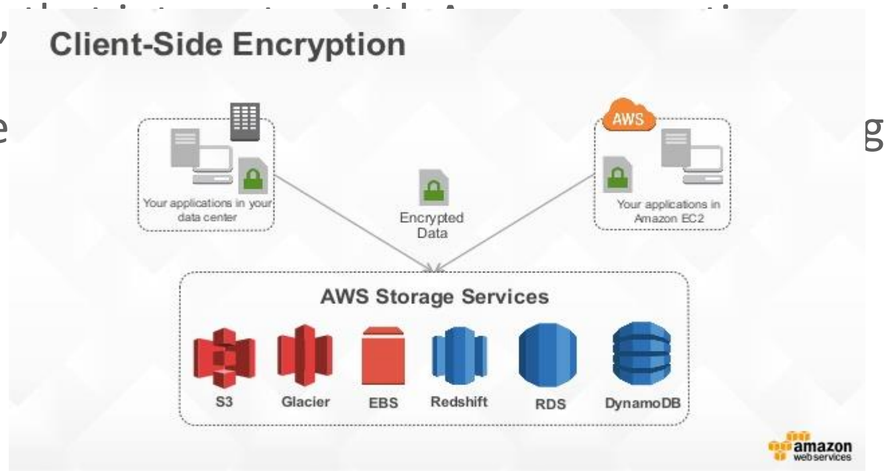
Data encrypted by an application that's running in the customer's datacenter or by a service application

Data that is already encrypted when it is received by Azure

For client-side encryption, Azure supplies a ***storage client library***, written for:

- Java
- .NET
- Python,

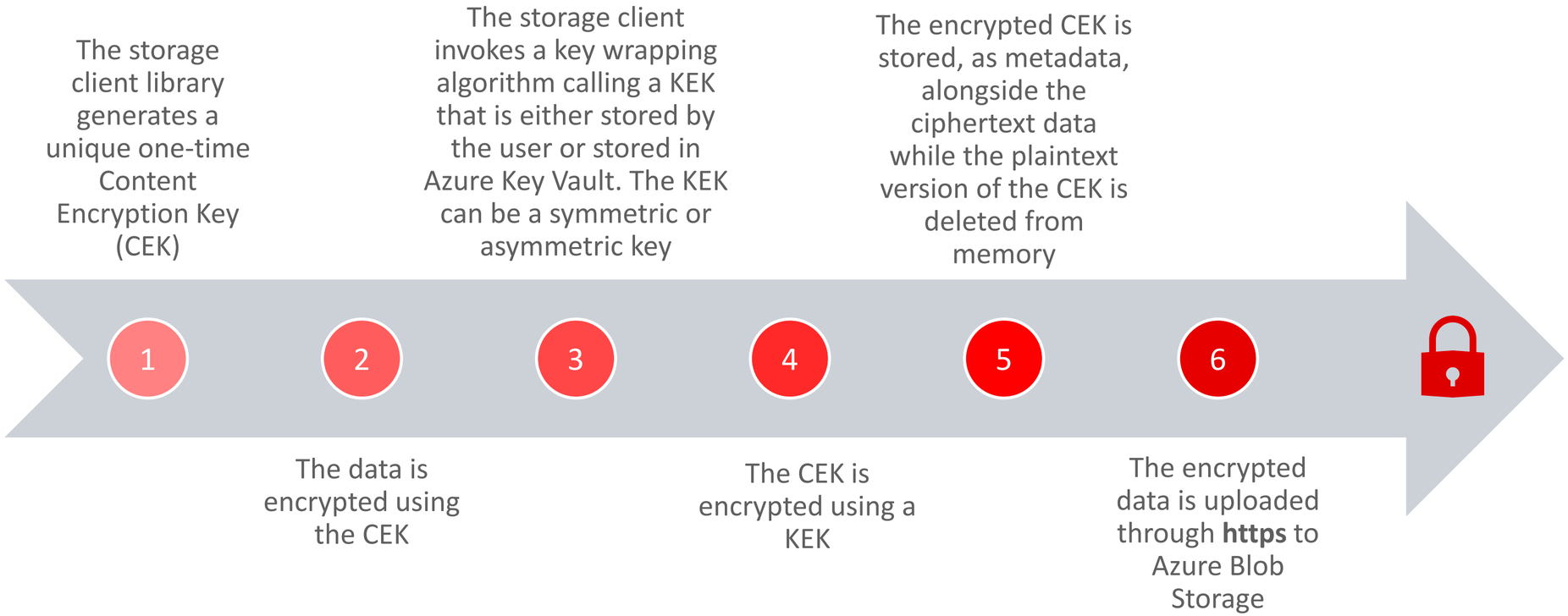
With this option, use g their own KEKs or using Azure Key Vault



edureka!

Client-side Encryption

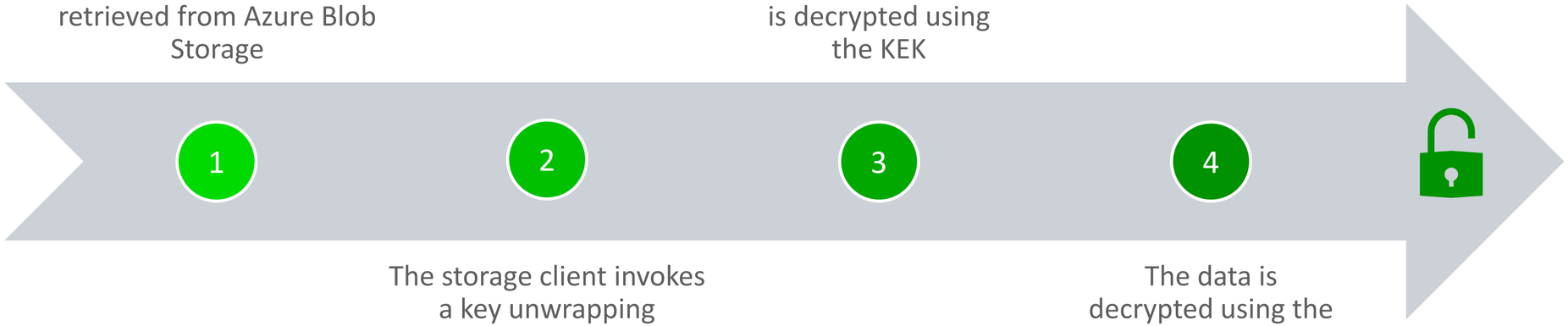
Encryption Workflow For Client-side



edureka!

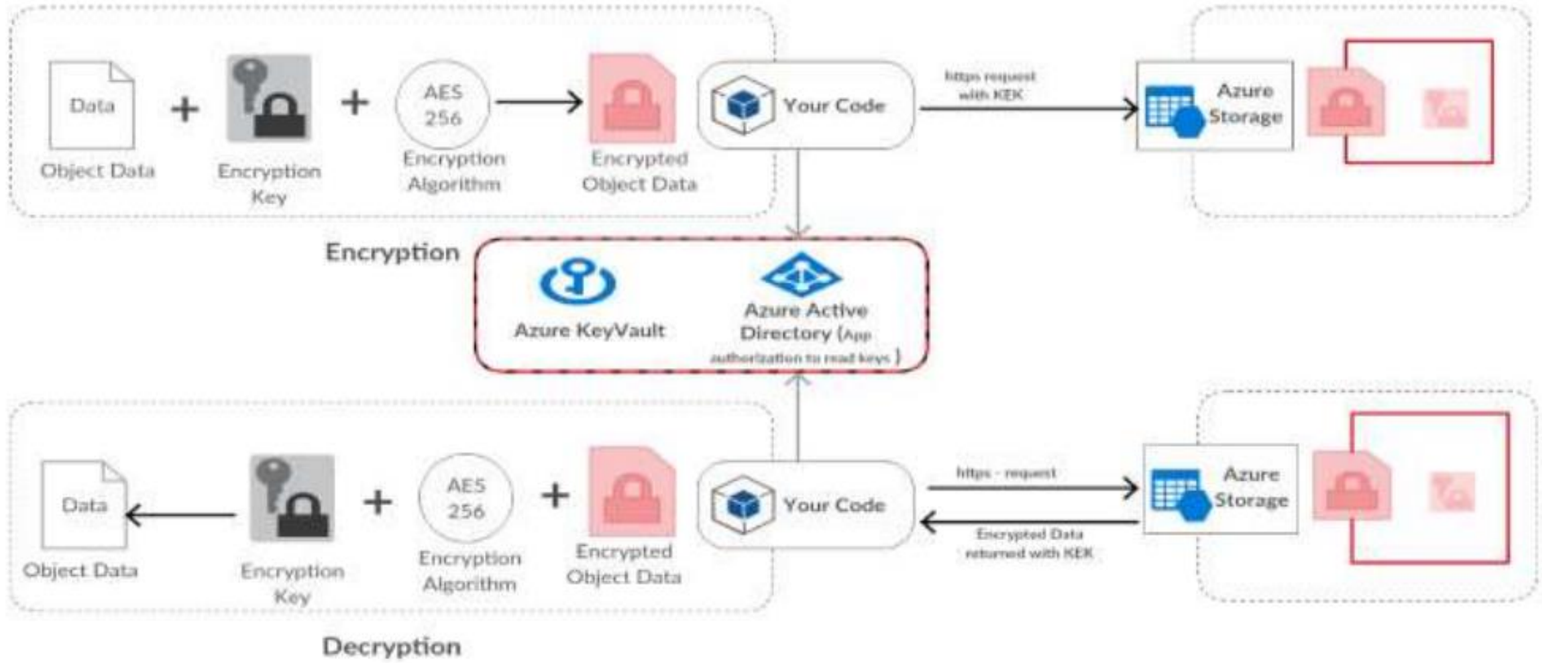
Client-side Encryption

Decryption Workflow For Client-side



edureka!

Client-side Encryption



Client-side Encryption

NOTE

Users can also choose to encrypt data prior to being uploaded to Azure using their own cryptographic and key management infrastructure without the storage client library

The encryption process is transparent to Azure Blob Storage and the encrypted data is stored as it would be with unencrypted data

Demo 1 – Client-side Storage Encryption Using Client Storage Library for .NET

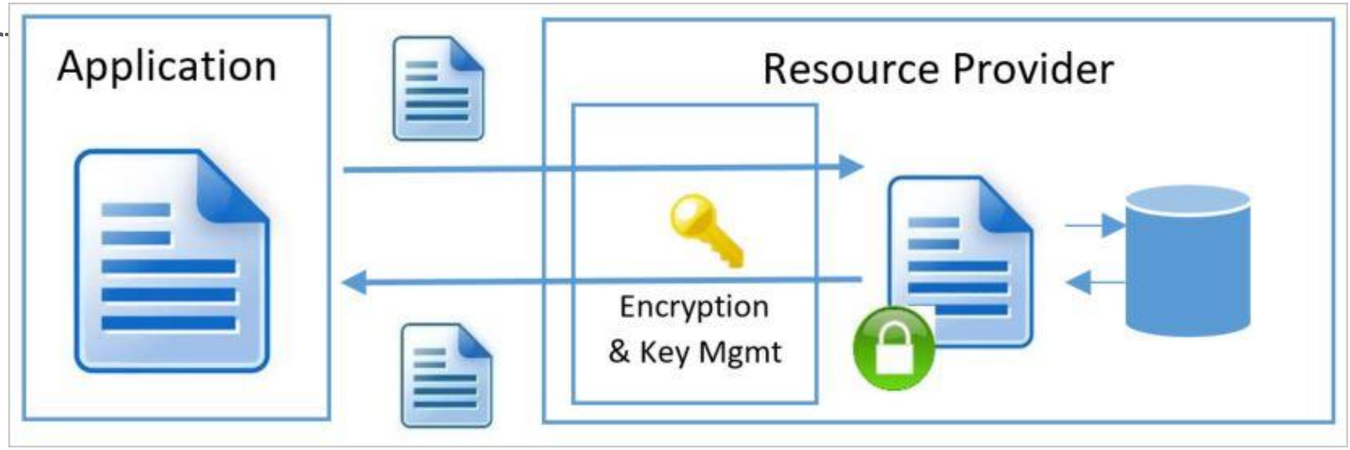
edureka!

Key Management for Server-side Encryption

Here, keys are managed via one of two options:

1. All keys are generated and stored by the Azure Blob Storage service itself. Microsoft handles key storage and management with no customer involvement.

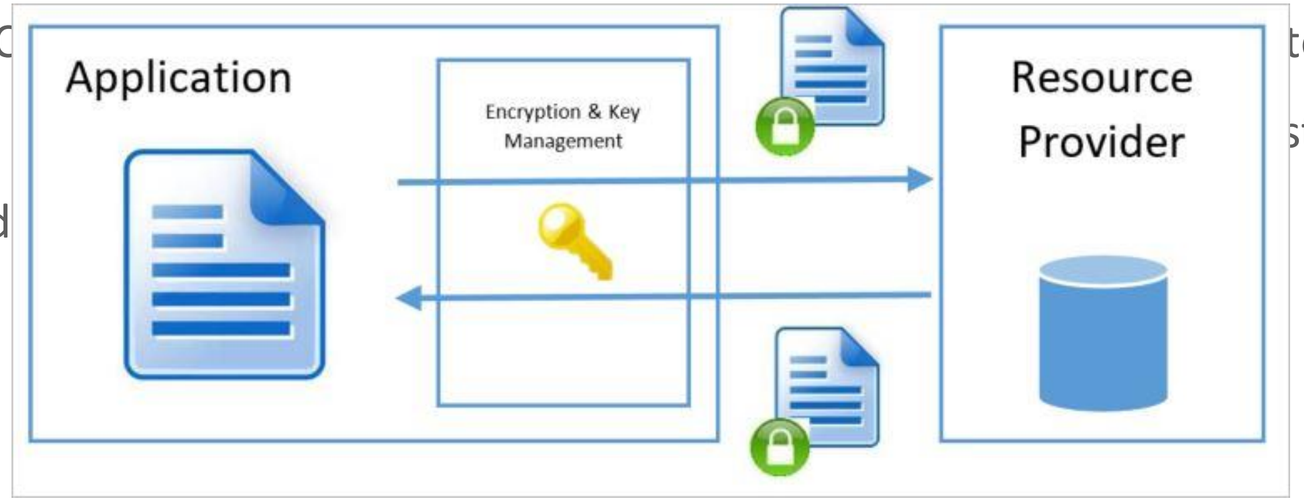
2. CEKs are generated and stored within Azure Key Vault. KEKs are stored within Azure Key Vault but managed by the customer. The KEK can be generated within Key Vault or impor



Key Management for Client-side Encryption

Here, keys are managed via one of three options:

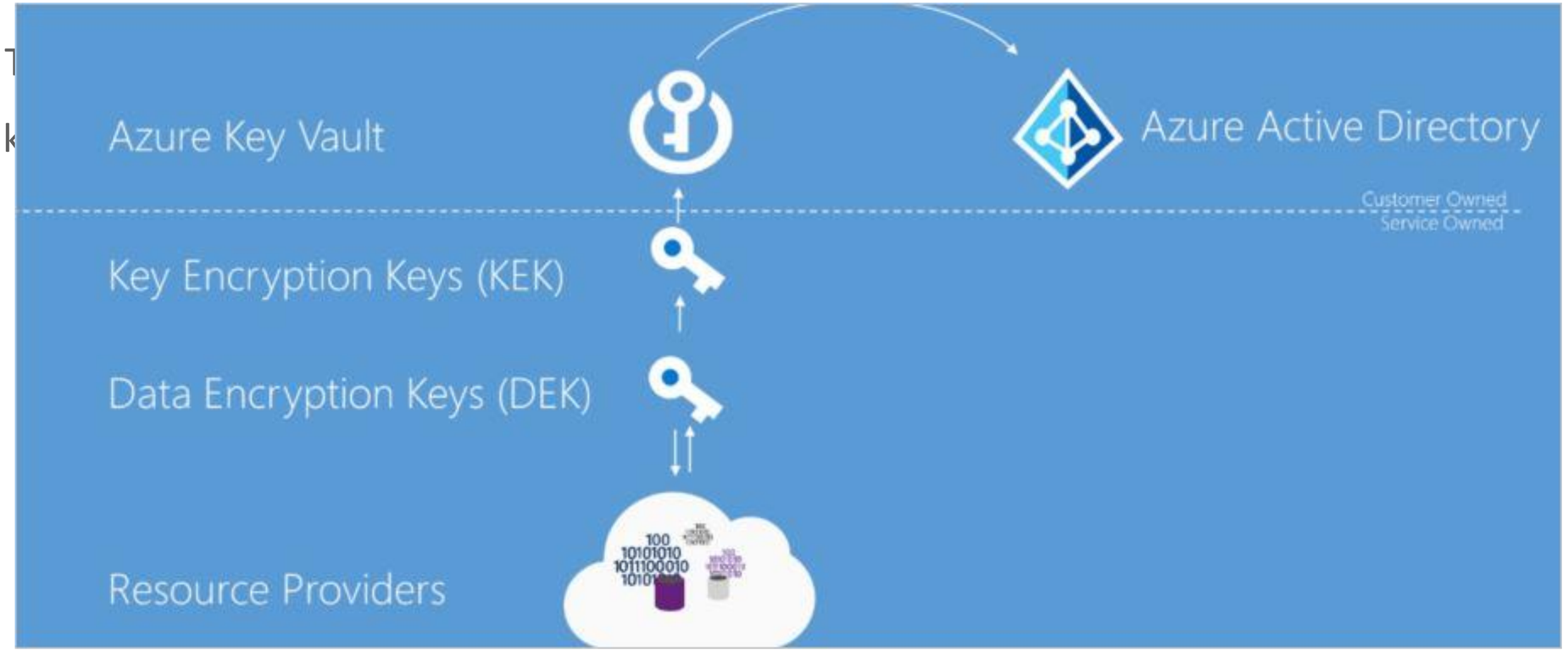
1. CEKs are generated by the Azure storage client library. KEKs are stored within Key Vault but managed by the customer.
2. CEKs are generated by the Azure storage client library. KEKs are generated, stored and managed by the customer using their own key management infrastructure.
3. Both the C tomer using their own storing encrypted



Key Management With Key Vault

For both the service-managed and Azure Key Vault options, keys are stored in a set of ***Hardware Security Modules (HSM)*** that are managed by Microsoft

With the service-managed option, all keys are generated by the Azure Blob Storage Service and managed by Microsoft



edureka!

Encryption of Data at Rest

Azure Encryption Models

Azure also supports the below encryption models:

- Azure disk encryption
- Encryption of data at rest with Azure SQL Database
- Cosmos DB database encryption
- At-rest encryption in Data Lake

edureka!

Encryption in Transit

Azure offers many mechanisms for keeping data private as it moves from one location to another

Encryption of data in transit

- **TLS/SSL** protocol to protect data when it's traveling between the cloud services and customers
- Shared Access Signatures (**SAS**), which can be used to delegate access to Azure Storage objects (HTTPS)
- **SMB 3.0**, which used to access Azure Files shares, supports encryption

In-transit encryption in VMs

- You can connect and sign in to a VM by using the **RDP** from a Windows client computer, or from a Mac with an RDP client installed
- For remote management, you can use **Secure Shell (SSH)** to connect to Linux VMs running in Azure (Public/Private keys)

Azure VPN encryption

- You can use an **Azure VPN gateway** to send encrypted traffic between your VNet and On-premise or VNet to VNet
- In **Point-to-Site** VPNs, ***SSTP*** is used to create the VPN tunnel
- **Site-to-Site** VPNs use ***IPsec/IKE (IKEv1 or IKEv2)*** for transport encryption

Integrate Caching and Content Delivery Within Solutions

edureka!

Azure Cache for Redis

edureka!

Azure Cache for Redis

This Cache improves the **performance** and **scalability** of systems that rely heavily on backend data-stores

Performance is improved by ***temporarily*** copying frequently accessed data to fast storage **located close** to the application

Used as an in-memory data structure store, a distributed non-relational database, and a message broker

Application performance is improved by taking advantage of the low-latency, high-throughput performance of the Redis engine

Provides you access to a secure, dedicated Redis cache

It is managed by Microsoft, hosted within Azure, and accessible to any application within or outside of Azure

Using Azure Cache for Redis – Use Cases

- Cache-Aside
- Content Caching
- User session caching
- Job and message queuing
- Distributed transactions

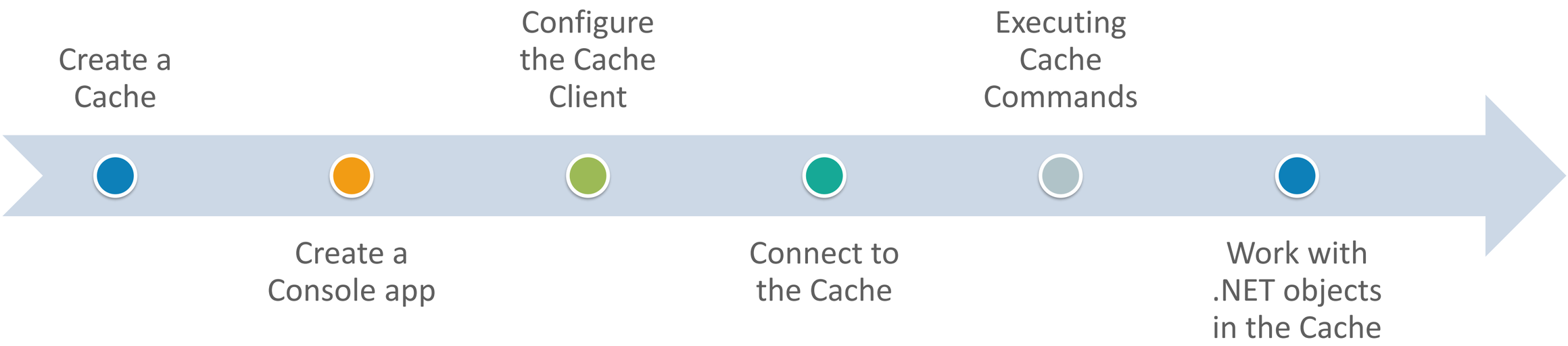
Azure Cache for Redis Offerings

Tier	Description
Basic	• A single node cache. This tier supports multiple memory sizes (250 MB - 53 GB) • This is an ideal tier for development/test and non-critical workloads. (No SLA)
Standard	• A replicated cache in a two-node, primary/secondary, configuration managed by Microsoft, with a high-availability SLA (99.9%)
Premium	• This is the Enterprise-ready tier • These caches support more features and have higher throughput with lower latencies • Caches are deployed on more powerful hardware providing better performance compared to the Basic or Standard Tier

Use Azure Cache for Redis With A .NET Application

- **Prerequisites:**
  - Visual Studio
  - The StackExchange.Redis client requires .NET Framework 4 or higher

- **Steps involved:**



Demo 2 – Use Azure Cache for Redis With A .NET Application

edureka!

Azure Content Delivery Network (CDN)

edureka!

What is Azure Content Delivery Network?

With traditional internet distribution, a single server sends content to all end users –

A CDN delivers content through a **network of servers** in close proximity to your end users

- With Azure CDN, files become **universally available** and can get to end users a lot **faster**

- Allows rapid delivery of **high-bandwidth** content to users by caching their content at strategically placed physical nodes across the world

edureka!

Azure CDN Features

CDN caching rules

Dynamic Site Acceleration

File compression

HTTPS custom domain support

Azure diagnostics logs

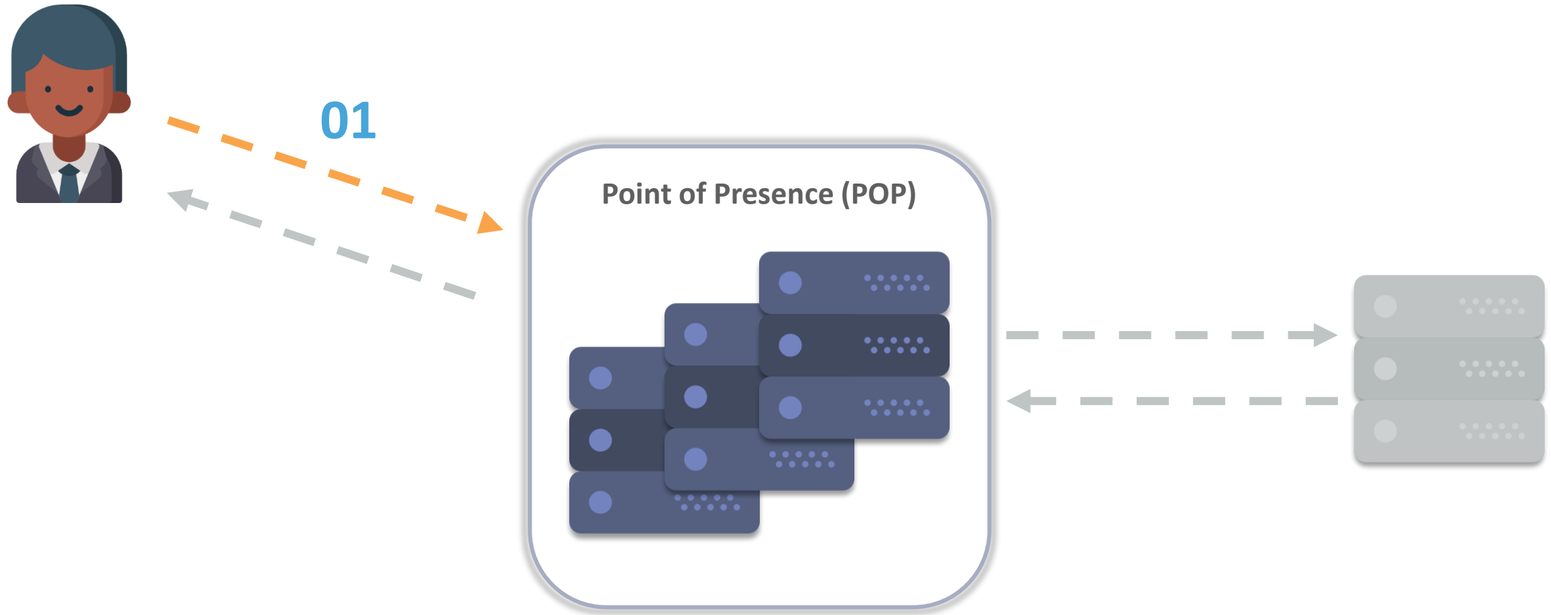
Geo-filtering

edureka!

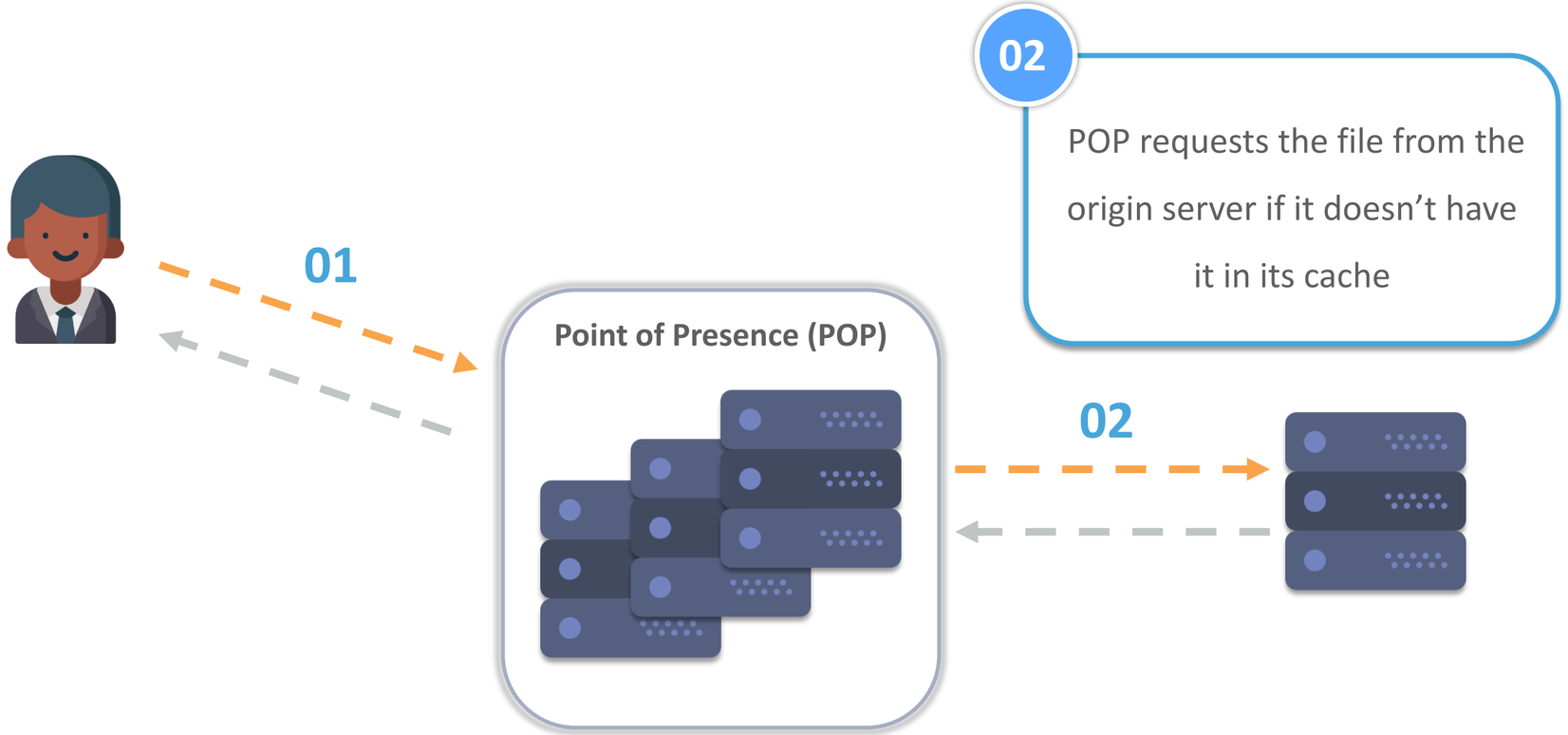
How CDN Works – Step 1

01

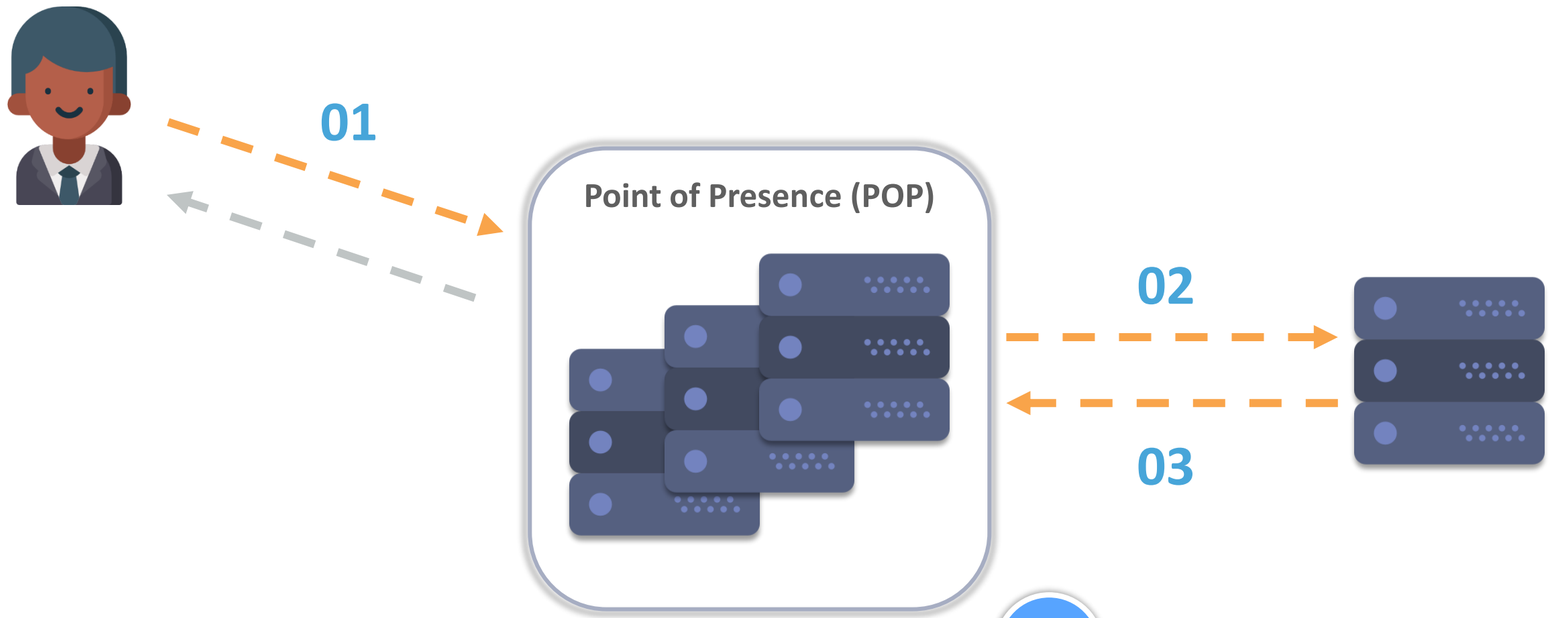
User requests a file using a URL and the DNS routes the request to the best performing POP location (usually geographically closest to the user)



How CDN Works – Step 2



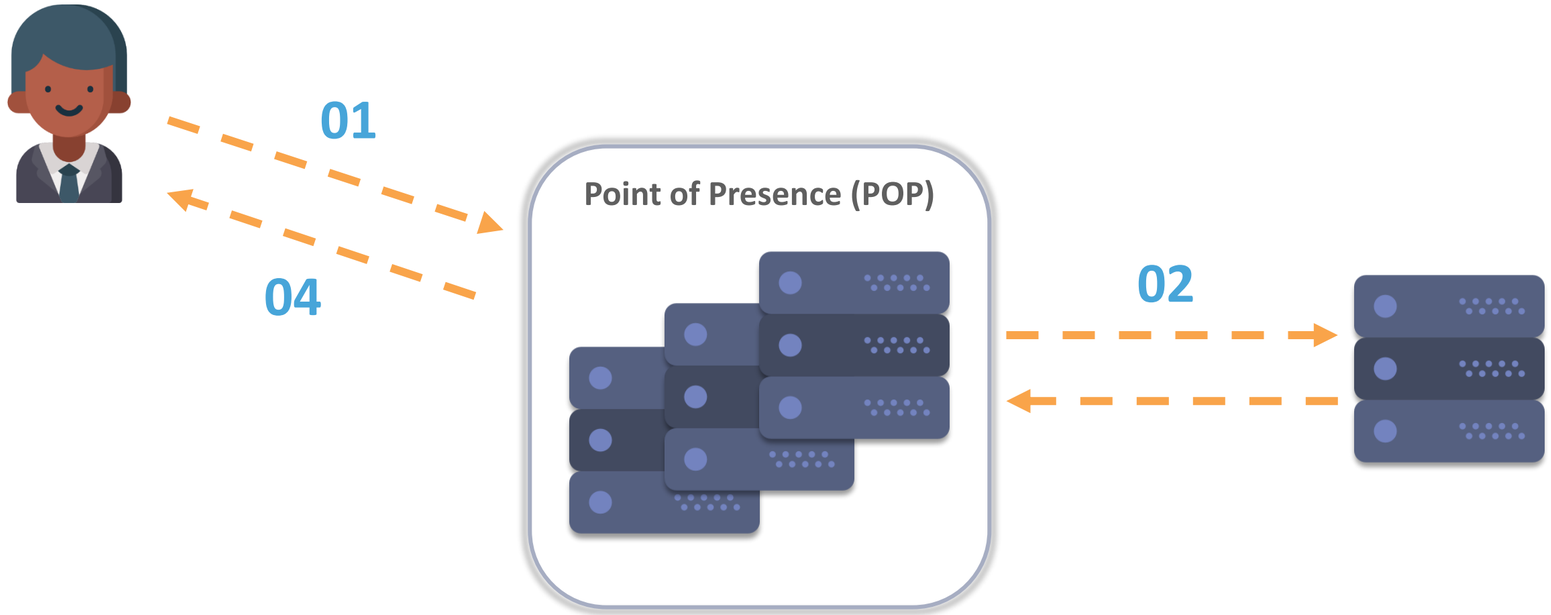
How CDN Works – Step 3



03

The origin server returns the file to an edge server in the POP

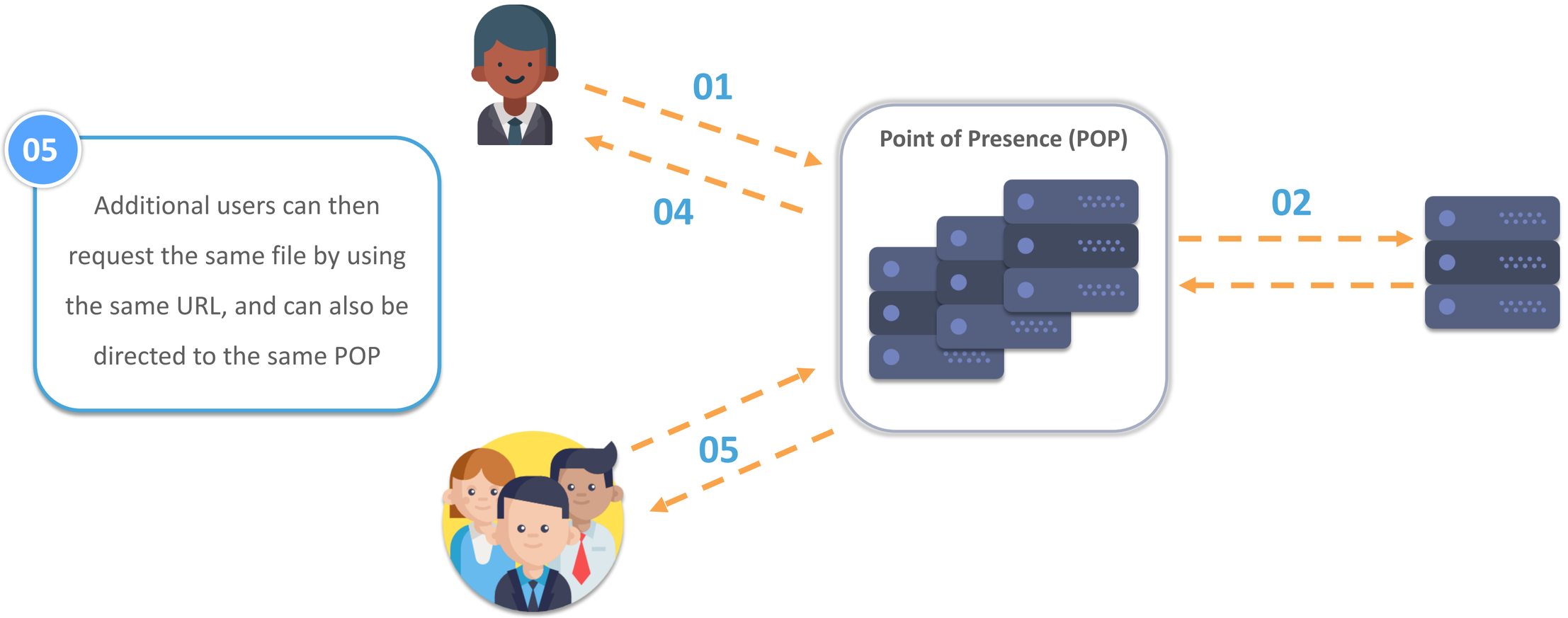
How CDN Works – Step 4



04

The edge server caches the file and returns it to the original requestor

How CDN Works – Step 5



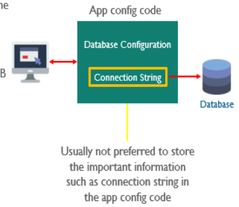
Demo 3 – Develop Code to Implement CDN in a Solution

edureka!

Summary

Azure Key Vault – Use Case

- Security plays a major role in case of accessing database
- Here the DB configuration information i.e. connection string is stored in the application config code
- We need to encrypt the complete application config code to secure the DB configuration information
- But the above suggested will not work in case of a web application as the Web app inside Azure has
  - No access to machine keys
  - No low-level access to actual VM



edureka!

Azure Encryption Aspects

Microsoft Azure supports the below aspects to encrypt data:

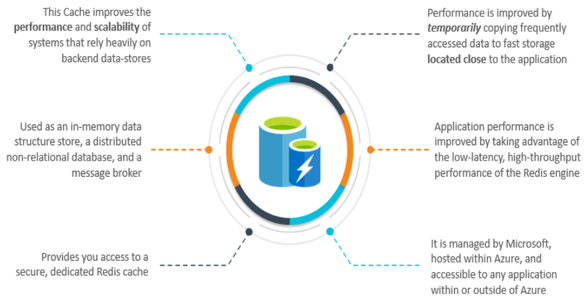
edureka!

Encryption Of Data At Rest

- Data at rest includes information that resides in persistent storage on physical media, in any digital format
- The media can include files on magnetic or optical media, archived data, and data backups
- Microsoft also provides encryption to protect Azure Storage service, Azure SQL Database, Cosmos DB, and Data Lake
- Data encryption at rest is available for services across the SaaS, PaaS, and IaaS cloud models

edureka!

Azure Cache For Redis



edureka!

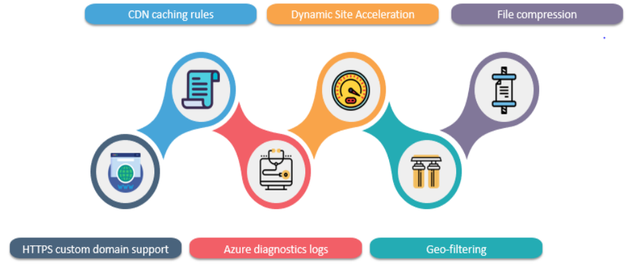
What Is Azure Content Delivery Network?

With traditional internet distribution, a single server sends content to all end users – A CDN delivers content through a **network of servers** in close proximity to your end users

- With Azure CDN, files become **universally available** and can get to end users a lot **faster**
- Allows rapid delivery of **high-bandwidth** content to users by caching their content at strategically placed physical nodes across the world

edureka!

Azure CDN Features



edureka!

edureka!

edureka!

edureka!

Thank You

For more information please visit our website
www.edureka.co