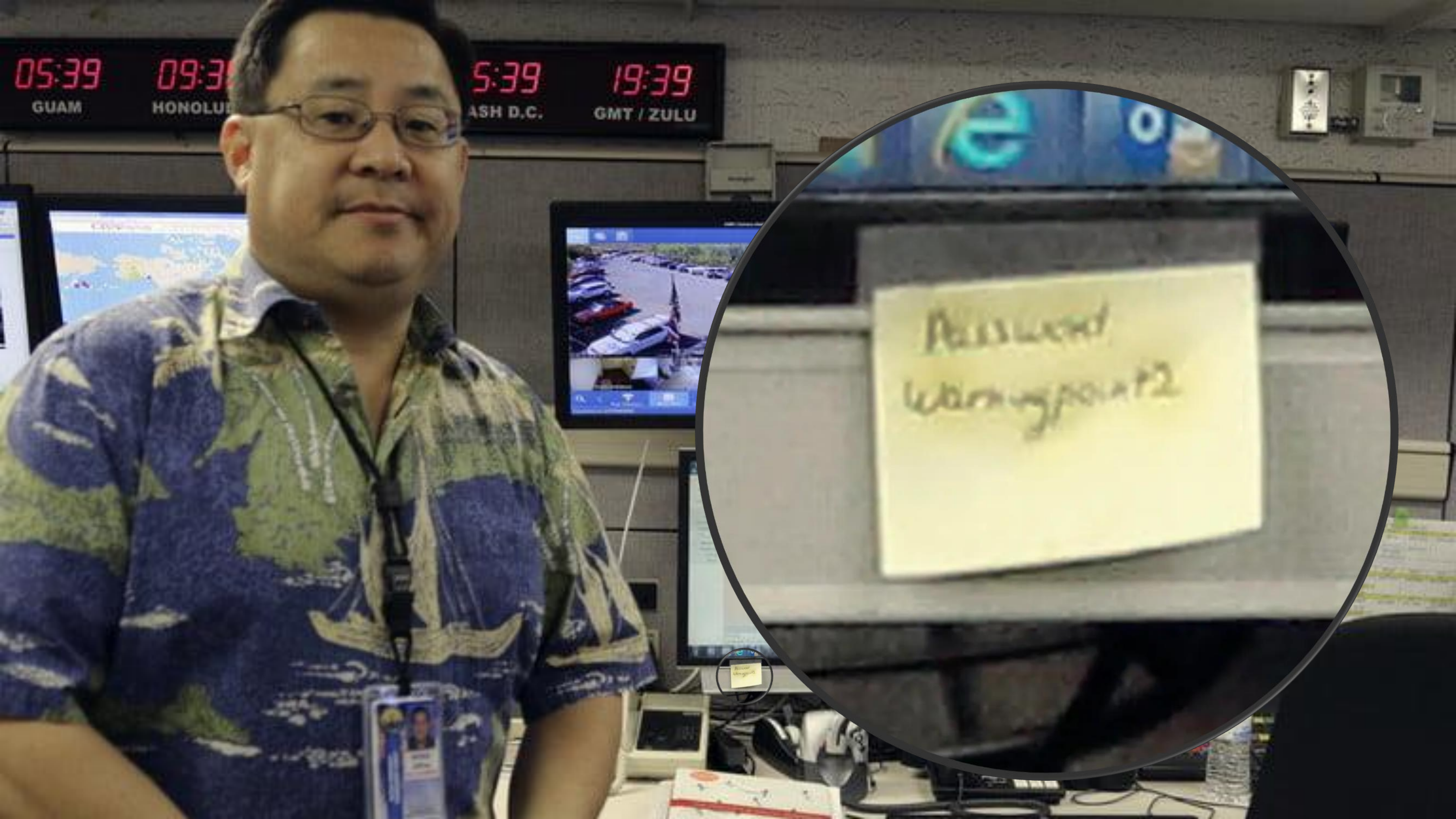
HomeGen

Tra poco inizierà il live

ARGOMENTO

Secrets safe and centralize with Azure KeyVault and Azure App Configuration!

«My name is **Bonanni**,
Massimo Bonanni»


05:39
GUAM
09:3
HONOLUL
5:39
ASH D.C.
19:39
GMT / ZULU
Password
warningpoint2

Azure Key Vault is a service that enables you to store & manage cryptographic keys and secrets in one central secure vault!!

The doubts of the Developers and IT Pros!!

I don't want the **responsibility** or potential liability for my customers' tenant keys and secrets.

I want customers to **own and manage** their keys so that I can concentrate on doing what I do best, which is providing the core software features.

I want to write an application for Azure that uses keys for **signing** and **encryption**. But I want these keys to be **external** from my application.

I want to make sure that my organization **is in control** of the key lifecycle and can monitor key usage.

Azure KeyVault key features

Secrets Management

Azure Key Vault can be used to Securely store and tightly control access to tokens, passwords, certificates, API keys, and other secrets

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.

Certificate Management

Azure Key Vault lets you easily provision, manage, and deploy public and private Transport Layer Security/Secure Sockets Layer (TLS/SSL) certificates.

Store secrets backed by Hardware Security Modules

The secrets and keys can be protected either by software or FIPS 140-2 Level 2 validated HSMs

Azure KeyVault actors

Vault Custodian

- Can create a key vault and gain full access and control over it.
- Can set up auditing to log who accesses secrets and keys.
- Can control the key lifecycle. Can roll to a new version of the key, back it up, and do related tasks.

Vault Consumer

- A vault consumer can perform actions on the assets inside the key vault when the vault owner grants the consumer access.
- The available actions depend on the permissions granted.

Access model overview

Management Plane Interface

- The management plane is where you manage Key Vault itself
- Operations in this plane include creating and deleting key vaults, retrieving Key Vault properties, and updating access policies
- Uses Azure Active Directory (Azure AD) for authentication
- Uses role-based access control (RBAC) for authorization

Data Plane Interface

- The data plane is where you work with the data stored in a key vault
- You can add, delete, and modify keys, secrets, and certificates
- Uses Azure Active Directory (Azure AD) for authentication
- Uses a Key Vault access policy for authorization

Hardware Security Module (HSM)

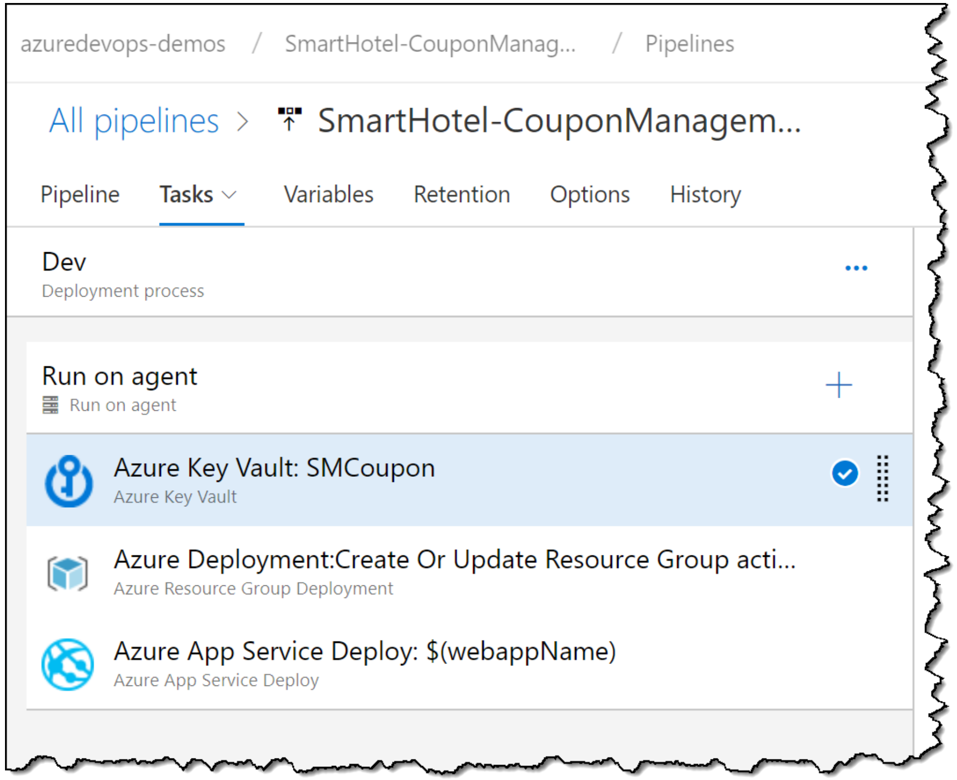
A **hardware security module** (HSM) is a physical computing device that safeguards and manages digital keys for strong authentication and provides cryptoprocessing.

A hardware security module contains one or more secure cryptoprocessor chips.

HSM modules are typically certified to internationally recognized standards such as **Common Criteria** or **FIPS 140**.

Platform Integration

- Azure Disk Encryption
- Trasparent Data Encryptionin Azure SQL Database
- Azure App Service
- Storage Account
- ARM Template
- Azure DevOps pipelines
- ...



azuredevops-demos / SmartHotel-CouponManag... / Pipelines

All pipelines > SmartHotel-CouponManagem...

Dev
Deployment process

Run on agent
Run on agent

Azure Key Vault: SMCoupon
Azure Key Vault

Azure Deployment:Create Or Update Resource Group acti...
Azure Resource Group Deployment

Azure App Service Deploy: $(webappName)
Azure App Service Deploy

How much?

Two different plans: Standard and Premium

Operations against all keys, secrets, and certificates are billed at a flat rate of €0.026 per 10,000 operations

Renewals of certificates: €2.530 per renewal request.

Software-Protected Keys:

- RSA 2048-bit keys, €0.026/10,000 transactions
- RSA 3072-bit, RSA 4096-bit, and Elliptic-Curve Cryptography (ECC) keys, €0.127/10,000 transactions

HSM-protected keys (only premium)

- RSA 2048-bit keys €0.844 per key per month + €0.026/10,000 transactions

Create a KeyVault
and
Platform Integration

Supported programming and scripting languages

REST Api

.NET

Java

Node.js

Python

Azure Powershell

Azure CLI

C# integration

Why use Azure Key Vault?

- Centralize application secrets
- Securely store secrets and keys
- Monitor access and use
- Simplified administration of application secrets
- Integrate with other Azure services

Azure App Configuration provides a service to centrally manage application settings and feature flag

App Configuration Key features

Key-Value store

- Stores configuration data as key-value pairs

Point-in-time snapshot

- Maintains a record of changes made to key-value pairs
- You can reconstruct the history of any key-value within the previous seven days

Feature management

- Decouples feature release from code deployment
- Enables quick changes to feature availability on demand
- AKA ”feature flags”

Security

- Encrypt using customer-managed keys
- Using private endpoints
- Integrate with Azure Managed Identity and Azure KeyVault

App Configuration benefits

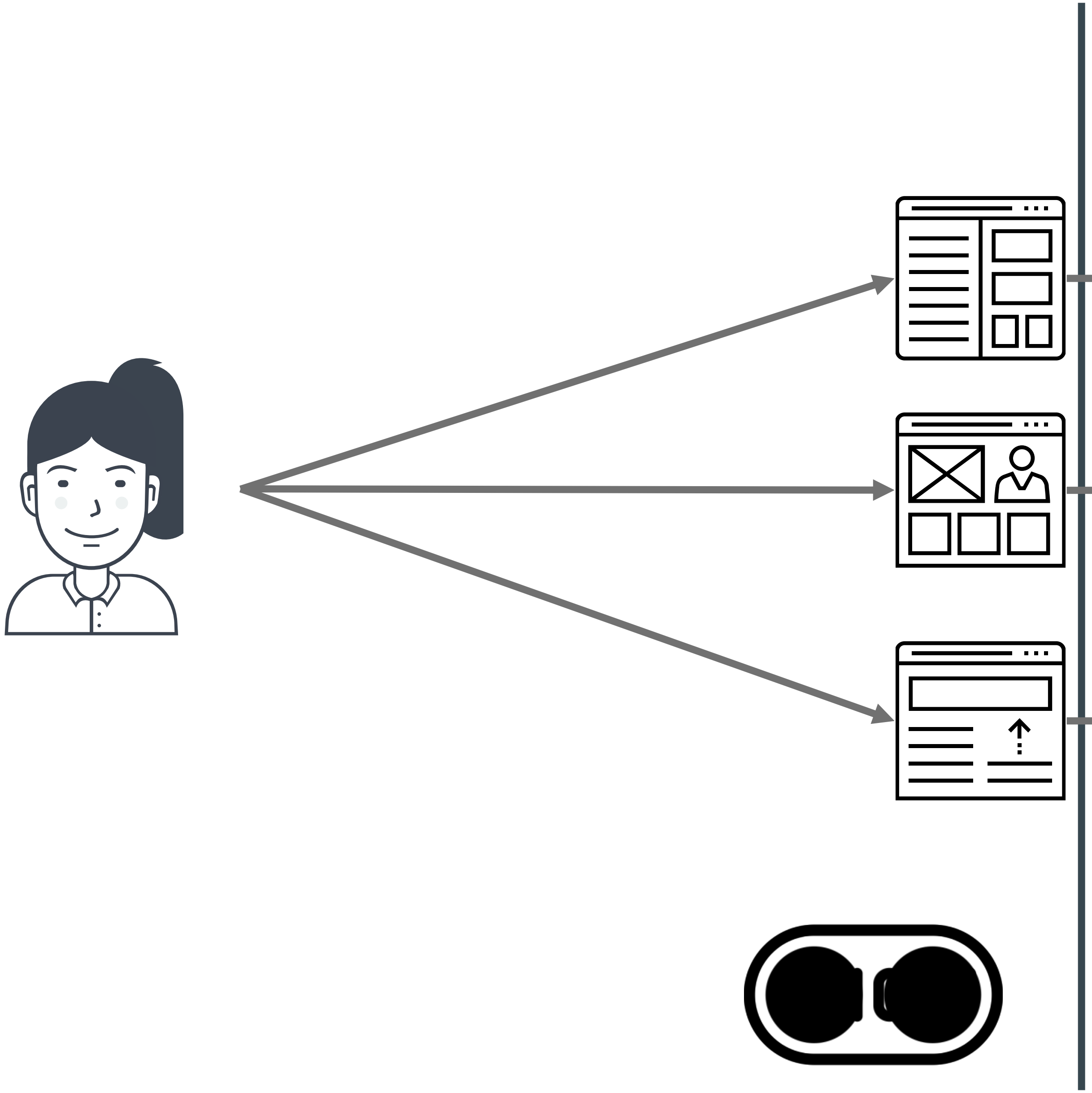
- A fully managed service that can be set up in minutes
- Flexible key representations and mappings
- Tagging with labels
- Point-in-time replay of settings
- Dedicated UI for feature flag management
- Comparison of two sets of configurations on custom-defined dimensions
- Enhanced security through Azure-managed identities
- Encryption of sensitive information at rest and in transit
- Native integration with popular frameworks

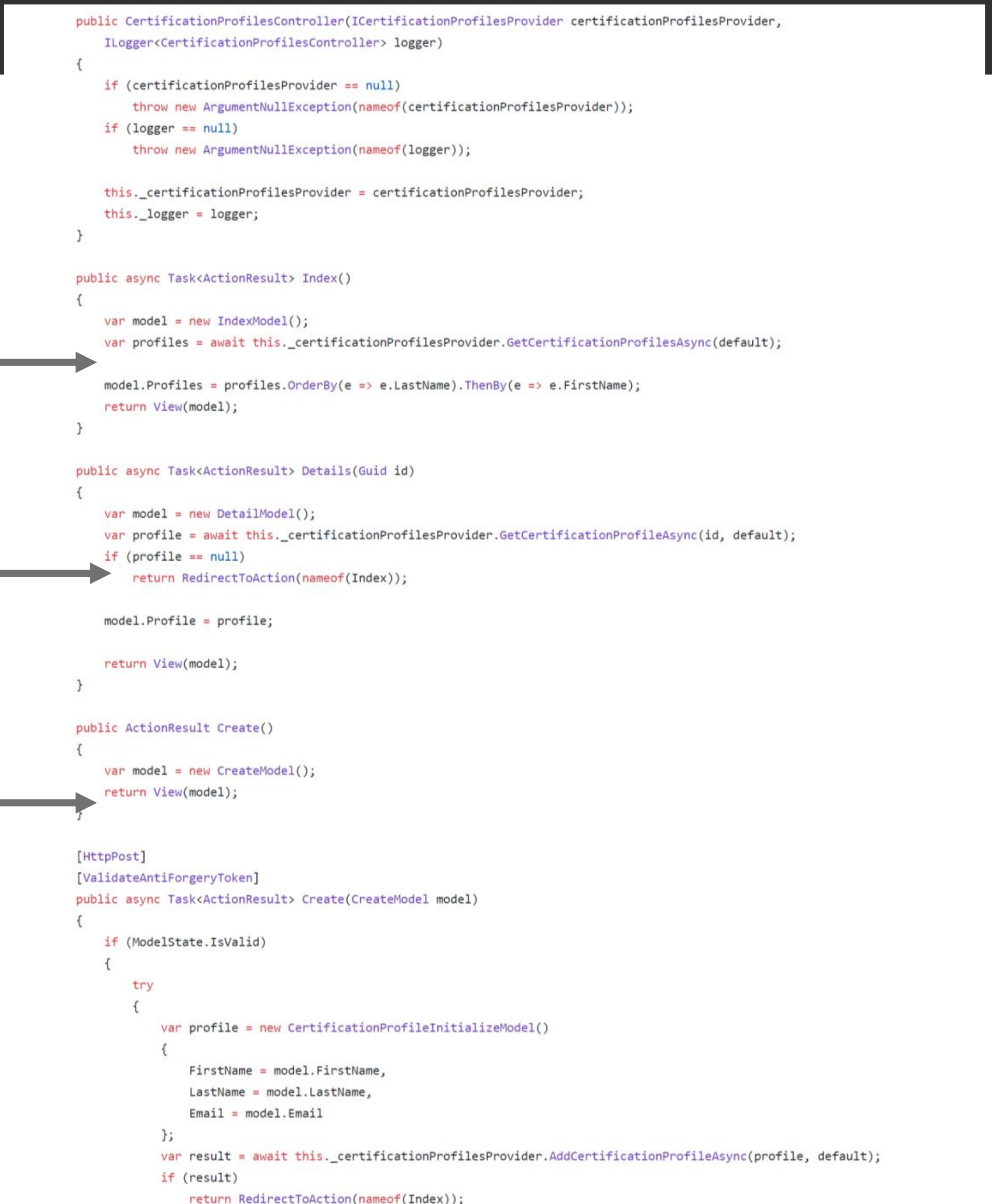
Feature Management

Feature management is a modern software-development practice that decouples feature release from code deployment and enables quick changes to feature availability on demand.

It uses a technique called ***feature flags*** (also known as *feature toggles*, *feature switches*, and so on) to dynamically activate/disactivate a feature.

Feature Flag





```
public CertificationProfilesController(ICertificationProfilesProvider certificationProfilesProvider,
    ILogger<CertificationProfilesController> logger)
{
    if (certificationProfilesProvider == null)
        throw new ArgumentNullException(nameof(certificationProfilesProvider));
    if (logger == null)
        throw new ArgumentNullException(nameof(logger));

    this._certificationProfilesProvider = certificationProfilesProvider;
    this._logger = logger;
}

public async Task<ActionResult> Index()
{
    var model = new IndexModel();
    var profiles = await this._certificationProfilesProvider.GetCertificationProfilesAsync(default);

    model.Profiles = profiles.OrderBy(e => e.LastName).ThenBy(e => e.FirstName);
    return View(model);
}

public async Task<ActionResult> Details(Guid id)
{
    var model = new DetailModel();
    var profile = await this._certificationProfilesProvider.GetCertificationProfileAsync(id, default);
    if (profile == null)
        return RedirectToAction(nameof(Index));

    model.Profile = profile;

    return View(model);
}

public ActionResult Create()
{
    var model = new CreateModel();
    return View(model);
}

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Create(CreateModel model)
{
    if (ModelState.IsValid)
    {
        try
        {
            var profile = new CertificationProfileInitializeModel()
            {
                FirstName = model.FirstName,
                LastName = model.LastName,
                Email = model.Email
            };
            var result = await this._certificationProfilesProvider.AddCertificationProfileAsync(profile, default);
            if (result)
                return RedirectToAction(nameof(Index));
```

Feature Management – Basic Concepts

Feature flag

A feature flag is a variable with a binary state of *on* or *off*. The feature flag also has an associated code block.

The feature flag's state triggers whether the code block runs.

Feature manager

A feature manager is an application package that handles the life cycle of all the feature flags in an application.

The feature manager also provides additional functionality, including caching feature flags and updating their states.

Filter

A filter is a rule for evaluating the state of a feature flag.

Potential filters include user groups, device or browser types, geographic locations, and time windows.

Point-in-time snapshot

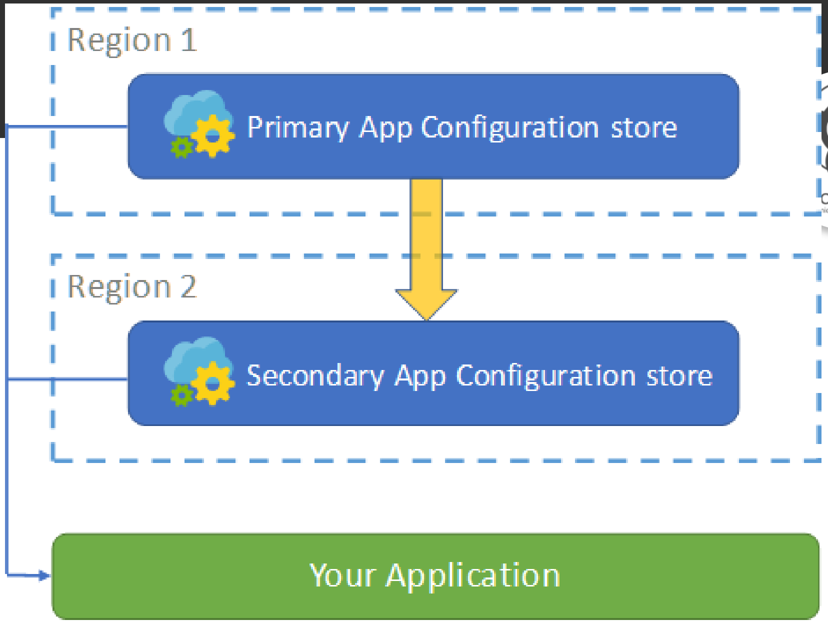
Azure App Configuration keeps records of the precise times when a new key-value pair is created and then modified.

These records form a complete timeline in key-value changes.

An App Configuration store can reconstruct the history of any key value and replay its past value at any given moment, up to the present.

With this feature, you can “time-travel” backward and retrieve an old key value.

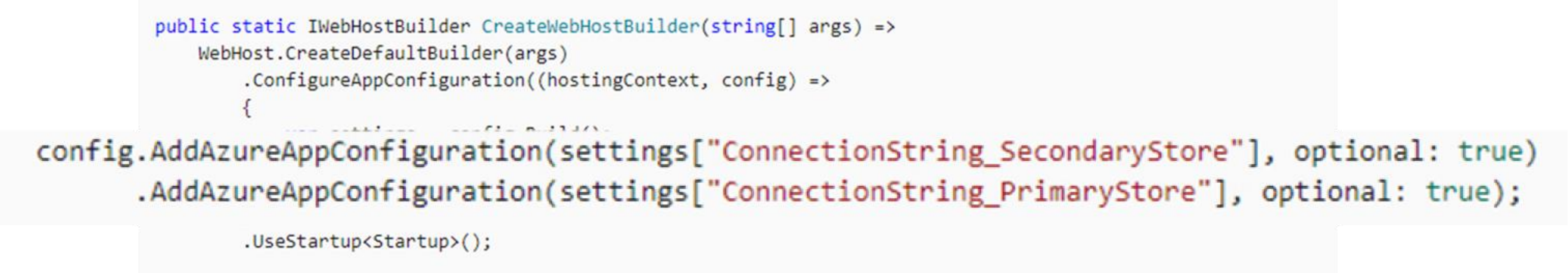
Resiliency and disaster recovery


Region 1
Primary App Configuration store
Region 2
Secondary App Configuration store
Your Application

- Azure App Configuration is a regional service.

- To realize cross-region redundancy, you need to create multiple App Configuration stores in different regions.

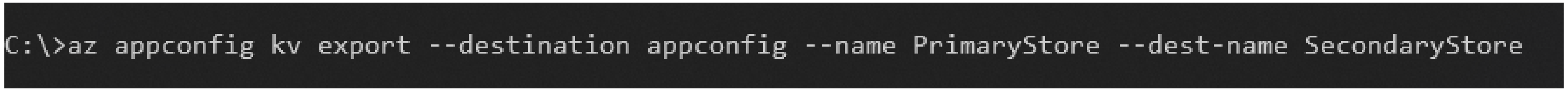
- Your application loads its configuration from both the primary and secondary stores.



```
public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
    WebHost.CreateDefaultBuilder(args)
        .ConfigureAppConfiguration((hostingContext, config) =>
        {
            var settings = config.Build();
config.AddAzureAppConfiguration(settings["ConnectionString_SecondaryStore"], optional: true)
         .AddAzureAppConfiguration(settings["ConnectionString_PrimaryStore"], optional: true);

        .UseStartup<Startup>();
```

- You can use the **Export** function in App Configuration to copy data from the primary store to the secondary on demand



```
C:\>az appconfig kv export --destination appconfig --name PrimaryStore --dest-name SecondaryStore
```

How much?

	FREE	STANDARD
Resources per subscription	1	Unlimited
Storage per resource	10 MB	1 GB
Key history	7 days	30 days
Requests per day	1,000 (HTTP status code 429 will be returned for all requests once the limit is reached)	First 200,000 included in the daily charge. Additional requests will be billed as overage.
SLA	None	99.9% availability
Security functionality	Encryption with Microsoft-managed keys HMAC or AAD authentication RBAC support Managed identity	All Free tier functionality plus: Encryption with customer-managed keys Private Link support
Cost	Free	€1.012 per day, plus an overage charge at €0.051 per 10,000 requests

Web Site Configuration

Why use App Configuration?

- Centralize management and distribution of hierarchical configuration data for different environments and geographies
- Dynamically change application settings without the need to redeploy or restart an application
- Control feature availability in real-time
- Enhanced security through Azure-managed identities
- Encryption of sensitive information at rest and in transit
- Native integration with popular frameworks

> “Companies spend millions of dollars on firewalls and secure access devices, and it's money wasted because none of these measures address **the weakest link in the security chain**: the people who use, administer and operate computer systems!
>
> *Kevin Mitnick*

Connect with me on LinkedIn

linkedin.com/in/massimobonanni

Thanks for your attention!!!!!

Massimo Bonanni

Azure Technical Trainer @ Microsoft

massimo.bonanni@microsoft.com
@massimobonanni

Microsoft CERTIFIED AZURE FUNDAMENTALS
Microsoft CERTIFIED AZURE DEVELOPER ASSOCIATE
Microsoft CERTIFIED AZURE SOLUTIONS ARCHITECT EXPERT
Microsoft CERTIFIED AZURE DEVOPS ENGINEER EXPERT
Microsoft CERTIFIED AZURE ADMINISTRATOR ASSOCIATE
Microsoft CERTIFIED AZURE AI ENGINEER ASSOCIATE
Microsoft CERTIFIED AZURE IOT DEVELOPER SPECIALTY
MICROSOFT CERTIFIED TRAINER Microsoft MCT Trainer 2020 - 2021

References

- Azure Key Vault documentation

  https://docs.microsoft.com/en-us/azure/key-vault/

- Azure Key Vault Developer's Guide

  https://docs.microsoft.com/en-us/azure/key-vault/general/developers-guide

- Channel9 - Azure Key Vault with Sumedh Barde

  https://channel9.msdn.com/Shows/Cloud+Cover/Episode-169-Azure-Key-Vault-with-Sumedh-Barde

- Azure App Configuration documentation

  https://docs.microsoft.com/en-us/azure/azure-app-configuration/

- What is Azure App Configuration?

  https://docs.microsoft.com/en-us/azure/azure-app-configuration/overview

- Channel 9 - Introducing Microsoft.FeatureManagement

  https://channel9.msdn.com/Shows/NET-Community-Standups/ASPNET-Community-Standup-May-21st-2019-Introducing-MicrosoftFeatureManagement

- Channel 9 - Getting started with Azure App Configuration

  https://channel9.msdn.com/Shows/Azure-Friday/Getting-started-with-Azure-App-Configuration

- Channel 9 - Azure App Configuration - Making Centralized Configuration Easy

  https://channel9.msdn.com/Events/dotnetConf/NET-Conf-2019/B210