

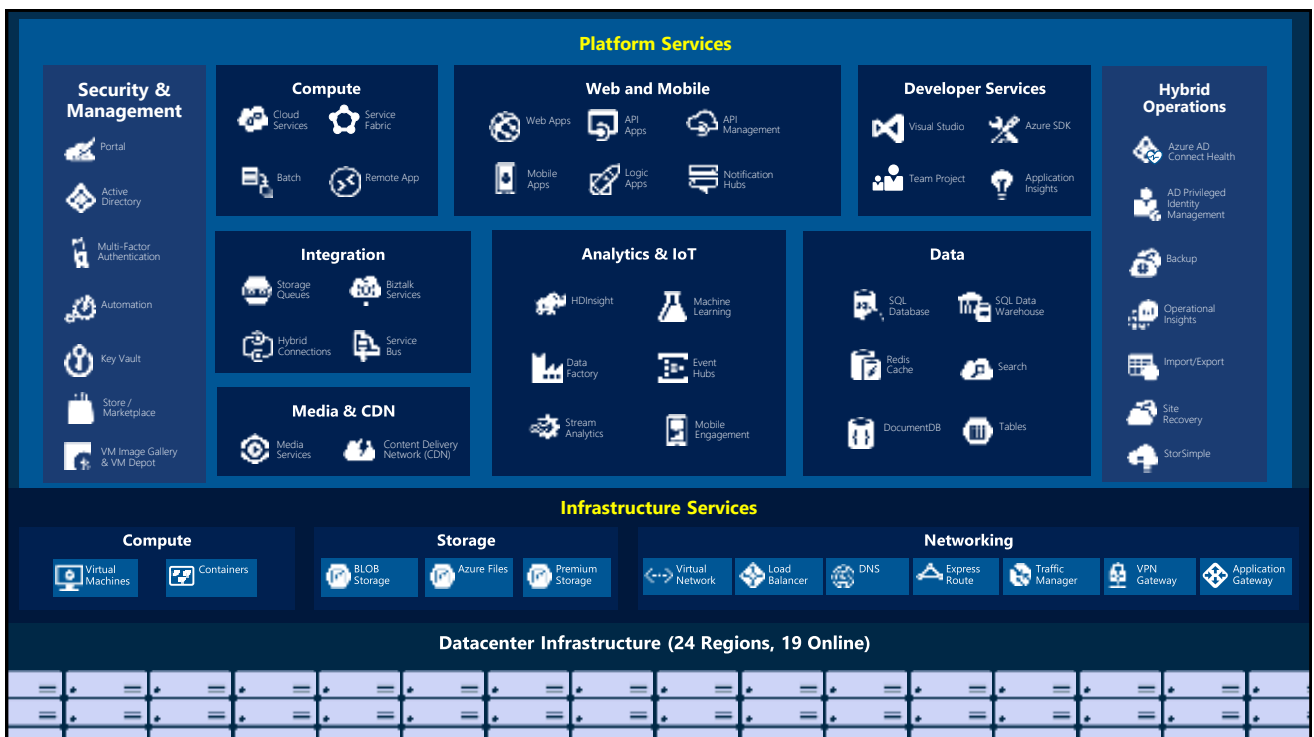


AZURE FACADES FOR SECURITY—PCI/HIPAA

Securing hybrid access

PHDIA X

© 2017 Phdix Inc. All rights reserved. This solution is for informational purposes only. PHDIX MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY. All trademarks are property of their respective owners.



Security boundaries

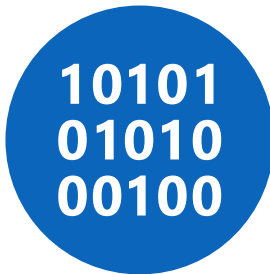
Azure—secure boundaries for your enterprise

Move the boundary of security to the edge of the internet, outside your datacenter

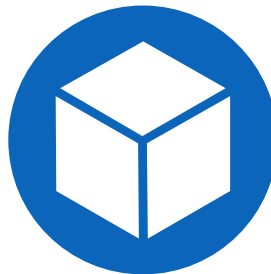
HIPAA REQUIREMENTS



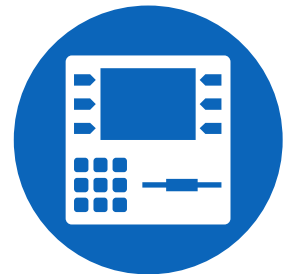
PHI REQUIREMENTS



PCI REQUIREMENTS



THREAT MITIGATION



TRUSTED CLOUD

Objectives

- ✓ Why do we care about security at all?
- ✓ What types of security may be encountered?
- ✓ What methods can we use have security?



What is the goal?

Prevent unauthorized access to information and preventing others from denying authorized access.

Mandatory compliance areas like finance and healthcare must address these areas to meet regulatory requirements.



Threat Model Approach (TMA)

Areas of threat:

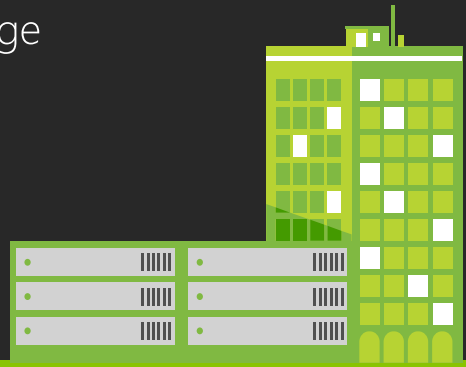
- Denial of service
- Information disclosure
- Elevation of privilege
- Tampering with data
- Repudiation
- Spoofing identity



Types

Encryption

- ✓ In flight—use Transport or Message
- ✓ At rest—in encrypted storage
- ✓ Securely—Bit depth
- ✓ Protect secrets



→ All the standards have guidelines

Methods

Mutual certificate authentication

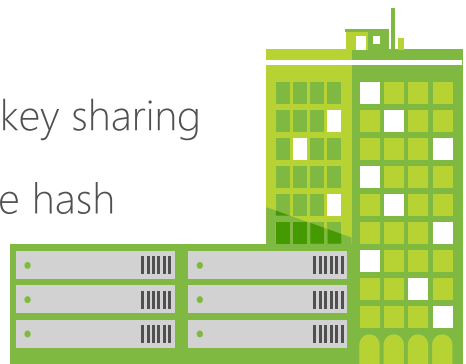
- ✓ Very secure, private key validation
- ✓ Widely used
- ✓ Requires custom code



➔ Make sure private certificates are securely held

Hashed values

- ✓ Has some repudiation
- ✓ Depends on mutual private hash key sharing
- ✓ Only validates fields that are in the hash
- ✓ Requires custom code



➔ Somewhat improved, doesn't address encryption

Embedded key

- ✓ Minimal security
- ✓ Requires embedding the key in every message
- ✓ No actual message validation
- ✓ Requires custom code

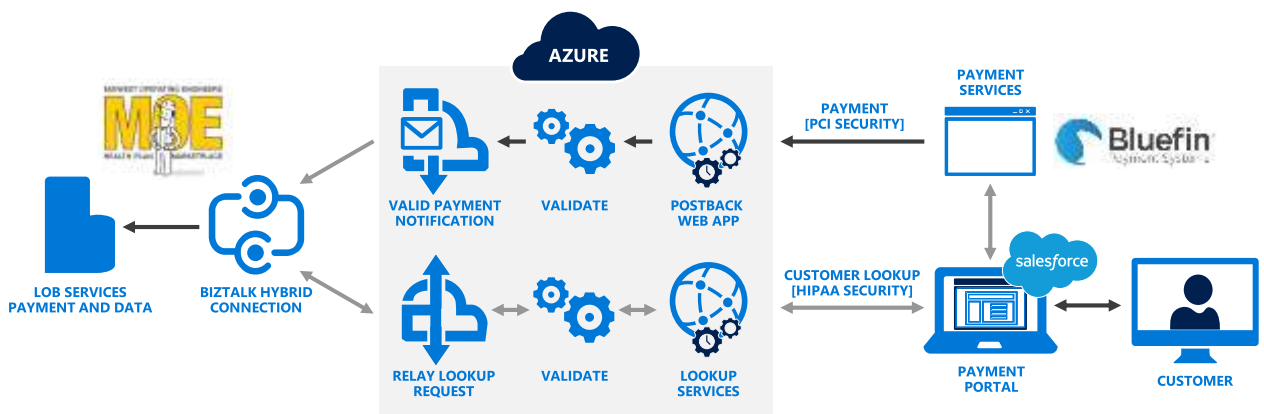


➔ Get the idea? Custom code... Web APPS

Buying something?

Implementation

Typical integration solution



PCI security

Payment processing companies already have well vetted solutions to PCI security.

This means we must match their security requirements.

Bluefin uses two elements:

- HASH to prevent tampering
- SSL to prevent disclosure
- Combine this with the Azure Service Bus and you have a secure and scalable method to post payments

The screenshot shows a web form titled "Bluefin Health Insurance Marketplace". Below the title, a red note states "Fields marked with * are required." The form is divided into two main sections: "Payment Details" and "Billing Information".

Payment Details: Contains a single field labeled "Amount: *" with an asterisk indicating it is required.

Billing Information: Contains several fields:

- Card Number ***: A long text input field with an asterisk.
- CVV2**: A short text input field.
- EXP ***: A short text input field with an asterisk.
- First Name ***: A text input field with an asterisk.
- Last Name ***: A text input field with an asterisk.
- Email**: A text input field.

 Below these fields is a "Process Payment" button.

HASH—encrypt me

```
public static string CalculateHash(string sAccount_ID, string sApi_AccessKey,
string sTimeStamp)
{
    SHA256 hhSHA256 = SHA256Managed.Create();
    return ToHex(hhSHA256.ComputeHash(
        Encoding.ASCII.GetBytes(
            sAccount_ID + "," +
            sApi_AccessKey + "," +
            sTimeStamp)
        )
    );
}
```

Show me the code!

PCI security

Payments processing
On-premises tracking
Azure Service Bus Relay

Make a payment

Please select a payment option below

- ☒ **\$123.00 thru Quarter 4 2015**
(Paid Thru: Dec 2015)
- ☐ **\$369.00 thru Quarter 2 2016**
(Paid Thru: Jun 2016)
- ☐ **\$246.00 thru Quarter 1 2016**
(Paid Thru: Mar 2016)
- ☐ **\$492.00 thru Quarter 3 2016**
(Paid Thru: Sep 2016)

SUMMARY

\$123.00 **Payment thru Quarter 4 2015**
(Paid Thru: Dec 2015)

-\$0.00 **Credit Balance**

TOTAL AMOUNT DUE TODAY

\$123.00

MAKE A PAYMENT

HASH—validate me

```
if (String.Compare(
    certificate.Thumbprint, SFDCClientCertThumbprint.Trim(), true) == 0)
{
    result = true;
    sbErrorMessage = "Certificate is valid and allowed.";
}
```

See: How To Configure TLS Mutual Authentication for Web App by Nazim Lala

```
$r = Get-AzureResource -Name "$PubEnv" -ResourceGroupName ($PubEnv + "ResourceGroup") -
ResourceType "Microsoft.Web/sites" -ApiVersion 2015-06-01
$p = $r.properties
$p.clientCertEnabled = $true
$r = Set-AzureResource -Name "$PubEnv" -ResourceGroupName ($PubEnv + "ResourceGroup") -
ResourceType "Microsoft.Web/sites" -ApiVersion 2015-06-01 -PropertyObject $p
```

NEW FEATURE!

HIPAA security

HIPAA compliance
Salesforce integration
Embedded key validation
SSL Encryption

Add New Dependent

Please fill out all the information below for a new dependent that requires coverage

| | |
|---------------------------------------|----------------------|
| FIRST NAME: | MIDDLE INITIAL: |
| <input type="text"/> | <input type="text"/> |
| LAST NAME: | SUFFIX: |
| <input type="text"/> | <input type="text"/> |
| GENDER: | |
| <input type="text" value="--None--"/> | |
| BIRTHDATE: | |
| <input type="text"/> | |
| <input type="text"/> | |
| EMAIL: | |
| <input type="text"/> | |

Access key

```
object AfterReceiveRequest(ref Message rQ, IClientChannel cH, InstanceContext iC)
{
    MessageBuffer mb = rQ;
    XPathNavigator xp = mb.CreateNavigator();
    if (xp.SelectSingleNode(xpath).Value != expectedValue)
        throw new AddressAccessDeniedException("Access Denied");
    rQ = mb.CreateMessage();
    return true;
}
```

Recap objectives

Azure provides deep and flexible tooling for security
Security requirements abound, be sure to comply
The ability to block at the boundaries simplifies
security of the enterprise