Study Definition Repository (SDR)

Azure Platform Design and Implementation Design and Plan V2.0

Document History

Version No.	Date	Author	Revision Description
V1.0	Dec 13th, 2021	Infra Team	Initial Version
V1.0	March 11th, 2022	Infra Team	Revised
V1.0	March 23rd, 2022	Infra Team	Addressed review comments
V2.0	Aug 19th, 2022	ACN	Updated figures with release 0.5 changes

Table of Contents

List of Tables

List of Figures

1. Introduction

This document details the infrastructure design for Azure environment utilized to deploy the Study Definition Repository (SDR) Reference Implementation.

SDR design and implementation working sessions were focused on Azure foundational design to drive and confirm architecture design elements (Network segmentation, Connectivity, App Services, API Management, Operations, Logging & Monitoring), discuss best practices, capture gaps and important notes pertinent to core pillars of Azure Infrastructure while allowing the SDR to be vendor and system agnostic. This document is organized by infrastructure components and presented in a dependency-based order. The chapters related to Subscription, Networking/Connectivity and Identity are the critical infrastructure components, in that, workloads cannot be deployed to Azure without all these components.

1.1.   Document Scope

This document presents and explains the underlying infrastructure design for SDR Reference Implementation. It answers the question, "Why was this piece/element of infrastructure design/deployed in this way?" The decisions and recommendations in this document are based to meet the fundamental business and architectural requirements as well we the established industry best practices. The document is structured to present the design, followed by the requirements and assumptions that were used as inputs, followed by any documented limitations.

1.2.   Out of Scope

This document is not a deployment document, it does not discuss or include deployment instructions, process, effort, or duration (deployment material is provided separately). The information presented in this document is a first iteration of the larger journey and is meant as a guide, and not intended to be the basis of a project plan. This document also does not address application architecture patterns or designs.

1.3.   Audience

This document assumes a 300-level knowledge of Azure concepts, components, and services. The audience for this document is:

- Able to understand the technical details associated with the Azure concepts and components
- Able to understand the company's future IT needs and direction, and correlate the needs with the SDR current design elements
- Is a decision maker or influencer

Level 300: Advanced material. In-depth understanding of features in a real-world environment, and strong coding skills. Provides a detailed technical overview of a subset of

product/technology features, covering architecture, performance, migration, deployment, and development.

1.4.　　　Design Decision Point Matrix

Throughout the document several outputs are summarized in each section. These outputs are categorized under the following headings:

Note

This header is to make the reader aware of something specific in the document and will give some additional context to the section.

Important Note

This header is to ensure the reader is fully aware of the point being highlighted. The information provided should be fully considered when understanding the context of the section.

Recommendation

A recommendation being made by the infrastructure team, but not necessarily a design decision.

Assumption

Based on the workshops and knowledge of infrastructure, assumptions on configurations and requirements are captured.

Design Decision

A design decision based on SDR requirements and the recommended best practices.

2. Consolidated Design Decisions and Recommendations

Please find below the summary of design decisions and recommendations for DDF SDR Azure infrastructure.

Table 1 Design Decisions & Recommendations

Module	Design	Decisions & Recommendations
Governance	Application Resource Group Strategy	Use separate Resource Groups for applications to enable independent Life Cycle
Governance	Infrastructure Resource Groups Strategy	Separate Resource Groups for infrastructure resources
Governance	Resource Tags	Use resource tagging for all the resources
Governance	Apply Resource Locks to Shared Resources	Use Resource Locks for Production and shared infrastructure resources
Governance	Log Analytics Deployment	Use a Centralized Log Analytics Instance
Governance	Naming Convention	Custom naming convention based on best practices
Subscriptions and Regions	Subscription Layout	Two Subscriptions for "Study Definition Repository"
Subscriptions and Regions	Select Azure Regions	One Azure Region "East US" is used
Networking	Protecting Public IPs	Setup VNets with basic DDOS protection for internal VNets
Identity	Authentitcation	Azure Active Directory Tennant
Identity	Multi Factor Authentication	Use MFA for all Administrative Access Use and for all Users
Identity	Admins and Administrative role Assignment	Limit the number of accounts with Owner rights
Identity	Role-based access control (RBAC) Strategy	Built-in Azure roles will be leveraged for RBAC strategy

Resources	App Services	Use 2 App Services per Environment/Subscription
Resources	App Service Plans	Use 2 App Service Plans per Environment/Subscription
Resources	API Management	Use API Management to create API Gateways for the backend services
Resources	CosmosDB	Use Azure CosmosDB API for MongoDB for NoSQL Database

3. SDR Reference Implementation Solution Architecture

The following diagram depicts SDR Reference Implementation Architecture. Key Design points are listed below: (Further details of the solution architecture are provided in supporting documentation).

Figure 1 SDR Reference Implementation Solution Architecture Design

Legend

DS = Delegated Subnet
SE = Service Endpoint
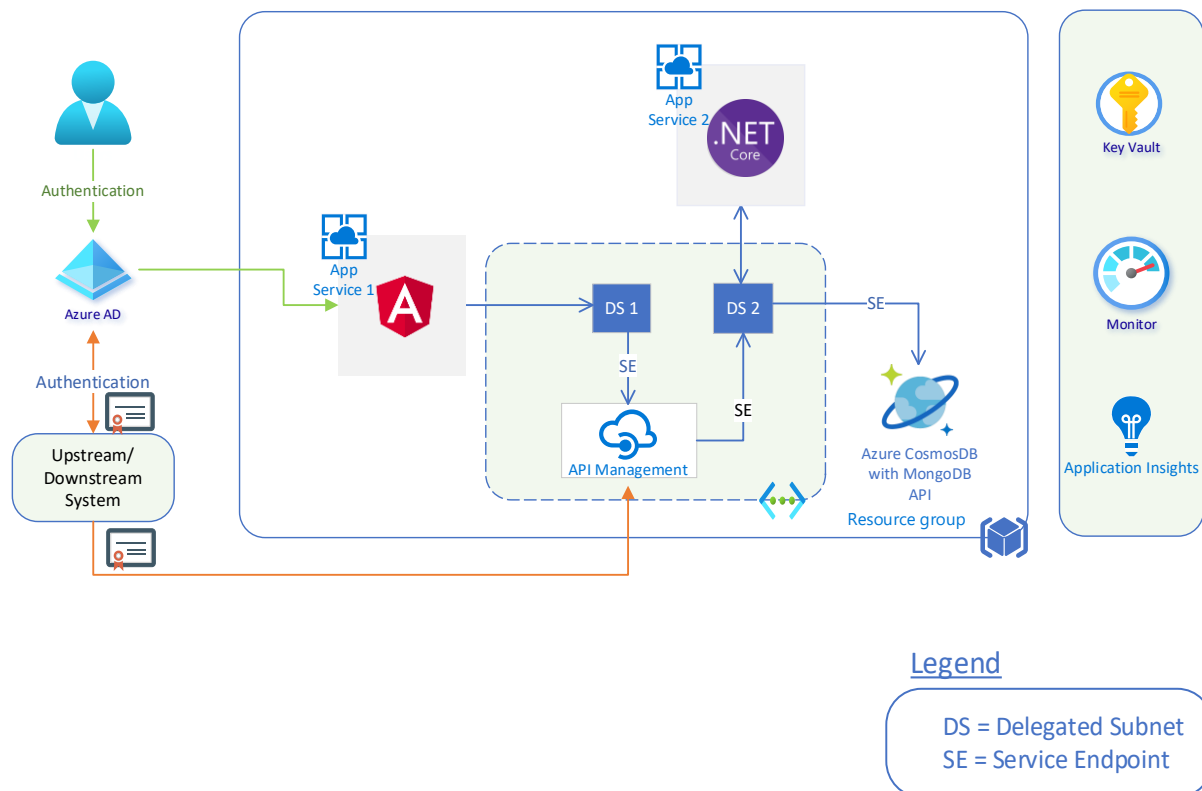
4. Governance

4.1.   Resource Groups

A resource group is a container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that will be managed as a group. The operator decides the method to add resources to resource groups based on current needs for an organization. Generally, resources that share the same lifecycle are added to the same resource group for easier and more efficient deployment, update, and/or delete.

Resource Groups are a critical concept in Azure Resource Management. A Resource Group is:

- A logical grouping of resources
- A container for delegation of administration (current recommended best practice)
- A target for RBAC

4.1.1. Resource Group Strategy

For SDR reference implementation separate Resource Groups for core (VNet, Subnets, Key Vault etc.) and app related (App Services, API Management, Application Insights etc.) infrastructure resources were used.

Generally, any centrally managed resources were grouped into separate Resource Groups and granted team's permission to consume the resources. Resource Groups for core resources are broken out based on specific workload access requirements enabling explicit access controls and duty segregation.

RBAC is natively a part of Azure's management platform and is used to control access to resource groups. There are 3 core roles (Owner, Contributor, Reader) that can be applied to Resource Groups and users can be placed into one of those roles or custom roles can be created. The table below details the following initial resource groups, based on the above guidance. Within the Subscription the following Resource Groups have been created:

Table 2 Resource Group Breakdown

Resource Group	Description
Core	A Resource Group containing the network infrastructure including the VNets/Subnets, Storage Account, etc. that are core to the entire infrastructure. This Resource Group should be locked with a "Delete" lock to prevent accidental deletion.
Application	Resource Groups containing the application related resources like App Services, API Management, Application Insights etc.

> **_Design Decision_**
> _SDR reference Implementation has used Resource Groups for Core and Application related infrastructure resources._

Additional Resource Groups can be created for additional services as needed using the previously described guidelines.

4.2.   Tagging

Tags provide a way to logically organize resources with custom properties and can be applied to Resource Groups and/or directly to individual resources. Tags can be used to refine the selection criteria for resources or Resource Groups from the console, web portal, PowerShell, or the Azure resource API.

- Tags are particularly useful when you need to organize resources for billing or management
- Tags can be applied to Resource Groups and to resources that support Azure Resource Manager (ARM) operations. All resources deployed into the SDR environment will be deployed using the ARM model and not the legacy model see as referenced in the ARM section
- Each resource or Resource Group can have a maximum of 50 tags
- Tags are key/value pairs, name is limited to 512 characters, value is limited to 256 characters
- Tags are free-form text so consistent, correct spelling and case-sensitivity is very important
- Tags defined on Resource Groups exist only on the Resource Group object; resources do not inherit the Resource Group's tags
- Each tag is automatically added to the Subscription-wide taxonomy
- It's important to develop a tag taxonomy early on and apply it consistently to all deployed resources
- Tags appear in billing and consumption reports
- Tags should not be used as a replacement for a proper CMDB. The information stored in tags has no inherent validation or relationship. Instead, the data in tags should reflect information that is contained within other systems such as the in-house developed CMDB for Application/Business Unit/Client information or a financial system for a tag such as a Cost Center

> **Design Decision**
> *SDR reference Implementation has used a standardized metadata taxonomy for all resources based on the table in the next section 4.2.1 (Table 3: SDR reference Implementation Azure Tags), that is applied to the resource at the time of resource creation using Terraform code deployment for consistency and an easily repeatable process.*

4.2.1. Resource Tag Taxonomy

SDR *reference Implementation* has used a standardized metadata taxonomy for all resources based on the following table, that is applied to the resource at the time of resource creation using Terraform code deployment for consistency and a repeatable, secure process.

Table 3 SDR reference Implementation Azure Tags

Tag Name	Description	Required/ Optional	Value	Resources
Environment	This tag is to show which environment the resources are being deployed into.	Required	Dev/QA/Pre-Prod	Resource Groups, VNet, API Management, App Services, App Service Plans, Storage Account, CosmosDB, Application Insights, Log Analytics Workspace & Key Vault
App Layer	This tag is show which application layer the resource belongs to.	Required	Frontend / Backend / N/A	Resource Groups, VNet, API Management, App Services, App Service Plans, Storage Account, CosmosDB, Application Insights, Log Analytics Workspace & Key Vault

4.3. Resource Locks

Resource Locks allow administrators to lock Subscriptions, Resource Groups, or individual resources to prevent accidental (or malicious) deletion or modification of critical resources. Unlike Role-based access control (RBAC) management, locks apply a restriction across all users and roles. A lock on a parent scope applies to all child resources. Resource Locks use the least privilege model, meaning the most restrictive lock in the inheritance takes precedence. There are two types of locks:

- **Delete (CanNotDelete)** – Authorized users can still read and modify a resource, but they can't delete it.
- **Read-only (ReadOnly)** – Authorized users can read a resource, but they can't delete or perform any actions on it.

4.3.1. Resource Locks for Critical Resources

> **Recommendation**
>
> *Apply Resource Locks for Production environments*

> **Design Decision**
>
> *SDR reference Implementation applied the Delete Lock (CanNotDelete) on all Resource Groups. The locks are applied at the Resource Group level. Locks applied to a Resource Group are also applied to all the resources that exist within that Resource Group.*

4.4. Management and Monitoring

Azure Monitor is an Azure's native monitoring, alerting and visualization solution for Metrics and Logs generated by Azure resources (IaaS and PaaS). Azure monitor collects data from variety of sources ranging from applications, OS and other enabling services and the underpinning platform itself. Secondly, data is collected for both the management plane (i.e., data about the operations done on the resource) and the data plane (i.e., data about operations happened inside the resource) of Azure resources. Data collected from compute resources (i.e., VMs) can be extended by installing Log Analytics and other Agents on them.

4.5. Naming Conventions

The most recommended and critical standard to implement while deploying to cloud platforms is a Naming Convention. The importance of a naming convention standard is amplified especially knowing that deployed services cannot be renamed without being destroyed first.

As well, including relevant and key information in the name of a service can go a long way in minimizing troubleshooting time when an issue arises. Strong naming used in conjunction with tags can provide a breadth of information for users looking up workloads and services.

There are many ways to go about creating a naming convention that is universal across all services, but it makes more sense to create a convention per service or resource. The reason being is that not all services have the same requirements for names. As an example, all storage accounts must be lowercase alphanumeric values up between 3-24 characters in length whereas a VM resource is between 1- 64 characters but can use uppercase letters and special characters.

4.5.1. Cloud Foundation Naming Standards

As mentioned in the section above, the best practice when developing a naming convention in the Cloud is to do it on a per resource level..

> **Design Decision**
> *SDR reference implementation has the following naming convention for Azure resources:*
>
> ***<ResourceType>-<App/Svc>-<Purpose/Segment/Environment>-<region>-<Instance Number or Level>***

The Meta data details for the Naming Convention is captured below:

Table 4 Naming convention metadata

Meta Data	Field Required	Field Length*
<ResourceType>	Mandatory	Upto 4 characters
<App/Svc>	Optional	Upto 5 characters
<Purpose/Segment/Environment>	Mandatory	Upto 20 characters
<Region>	Optional	Upto 6 characters
<Instance Number or Level>	Optional	Upto 3 characters

> **Important Note**
> *There are some exceptions to this naming convention where the resource names should be globally unique (like Azure Key Vault, Log Analytic Workspace, Storage Accounts etc.), some which do not allow any special characters (like Storage Accounts) and some have *character limitation.*

Table 5 SDR RI Naming Convention

Resource Type	Naming Scope	Sample	Restrictions	Comments
Resource Group	Subscription	*rg-<app/svc>-<environment>-<region>* rg-sdrcore-dev-eastus rg-sdrapp-dev-eastus	1 – 90 Characters Alphanumeric, underscores, parentheses, hyphens, periods, and unicode characters	

Resource Type	Naming Scope	Sample	Restrictions	Comments
		rg-sdrcore-qa-eastus rg-sdrapp-qa-eastus	Can't end with a period	
Virtual Network	Resource Group	*vnet-<app/svc>-<environment>-<region>* vnet-sdr-dev-eastus vnet-sdr-qa-eastus	2 – 64 Characters Alphanumeric, underscores, periods, hyphens Start with alphanumeric, end with alphanumeric or underscore	
Subnets	Virtual Network	*snet-<app/svc>-<environment>-<region>* snet-sdr-dev-eastus snet-sdr-qa-eastus	1 – 80 Characters Alphanumeric, underscores, periods, hyphens Start with alphanumeric, end with alphanumeric or underscore	
Delegated Subnets	Virtual Network	*dsnet-<app/svc>-<environment>-<region>*-<instance number or level> dsnet-sdr-dev-eastus-001 dsnet-sdr-dev-eastus-002 dsnet-sdr-qa-eastus-001 dsnet-sdr-qa-eastus-002	1 – 80 Characters Alphanumeric, underscores, periods, hyphens Start with alphanumeric, end with alphanumeric or underscore	

Resource Type	Naming Scope	Sample	Restrictions	Comments
Azure Key Vault (KV)	Global	*kv-<app/svc>-<environment>-<region>* kv-sdr-dev-eastus kv-sdr-qa-eastus	3 – 24 Characters Alphanumeric and hyphens Start with letter. End with letter or digit. Can't contain consecutive hyphens	
Log Analytics Workspace	Global	*law-<app/svc>-<environment>-<region>* law-sdr-dev-eastus law-sdr-qa-eastus	4 – 63 Characters Alphanumeric and hyphens. Start and end with alphanumeric	
App Services		*apps-<app/svc>-<environment>-<region>*-<instance number or level> *apps-sdr-dev-eastus-001* *apps-sdr-dev-eastus-002* *apps-sdr-qa-eastus-001* *apps-sdr-qa-eastus-002*		
App Service Plans		*asp-<app/svc>-<environment>-<region>*-<instance number or level> *asp-sdr-dev-eastus-001* *asp-sdr-dev-eastus-002*		

Resource Type	Naming Scope	Sample	Restrictions	Comments
		asp-sdr-qa-eastus-001 *asp-sdr-qa-eastus-002*		
API Management (APIM)	Global	*apim-<app/svc>-<environment>-<region>* *apim-sdr-dev-eastus* *apim-sdr-qa-eastus*		
CosmosDB	Global	*cdb-<app/svc>-<environment>-<region>* *cdb-sdr-dev-eastus* *cdb-sdr-qa-eastus*		
Application Insights		*appin-<app/svc>-<environment>-<region>* *appin-sdr-dev-eastus* *appin-sdr-qa-eastus*		

For more information on this and a list of the recommendations and best practice for naming conventions, please see Appendix A.1. For more information on this and a list of the naming restrictions per resource in Azure, please see Appendix A.2.

4.6.   Governance Decision Summary

Table 6 Governance Decision Summary

Design	Decision
Infrastructure Resource Groups Strategy	Separate Resource Groups for Core and Application related resources have been created
Resource Tags	Resource tagging has been done
Log Analytics Workspace Deployment	One Log Analytics Workspace Instance per environment has been used

5. Subscriptions and Regions

5.1.   Subscriptions

5.1.1.   Subscription Overview

Historically, Azure Subscriptions were used as boundaries for several different aspects of Azure services:

- As a security and administrative boundary where users can be granted co-administrator access which grants them administrative access to every resource in the Subscription. More granular RBAC can be granted at the resource and Resource Group level. However, using a Subscription in this fashion in the new Azure Resource Manager (ARM) model is no longer the recommended best practice.
- As a billing unit of granularity where usage and consumption reports can be viewed for all resources in the Subscription. Further granularity can be achieved with the use of tags at the resource and Resource Group level which is why using multiple Subscriptions in this fashion is no longer a best practice.
- As a logical unit of scale by which the number of specific types of Azure resources can be limited on a per-Azure region basis. For example, 10,000 compute cores per Subscription per region, etc. Azure continues to increase Subscription limits as services mature. For Current Subscription and service-specific limits please see Appendix A.3.

> ***Design Decision***
>
> *SDR Reference implementation has one subscription*
> - *Study Definition Repository – Pre-Prod*
>
> *A separate subscription has been used for SDR development and testing.*

5.1.1.1. Subscription Scale Limitations

Azure Subscriptions have some scale limitations that in terms of the number of specific resources of specific types that can be created in a per Subscription (or per Subscription and region combination). It is therefore important to consider when planning count of Subscription and other resources as outlined in this link.

5.1.1.2. **Azure Active Directory (AD) Associated with Subscriptions**

Each Azure Subscription has a trust relationship with an Azure Active Directory (AD) tenant instance which is used to authenticate users, services, and devices. Multiple Subscriptions can trust the same directory, but a Subscription can only trust one directory. All Subscription service administrator (and co-administrator) accounts reside within this Azure AD instance. Additionally, RBAC permissions are granted to a user or group from the associated Azure AD tenant.

5.1.1.3. Subscription Management Roles

Subscriptions are defined with several roles including Owner which should be handled with care. A recommendation to have a maximum of 3 subscription owners to reduce the potential for breach by a compromised owner.

5.2. **Azure Regions**

Azure operates in multiple datacenters around the world. These datacenters are grouped into geographic regions, giving flexibility in choosing where to build and locate applications.

The infrastructure team create Azure resources in defined geographic regions like 'East US', 'North Europe', or 'Southeast Asia'.

To see list of regions available in Azure please see Appendix A.4. Within each region, multiple datacenters exist to provide for redundancy and availability. This approach gives the infrastructure team flexibility as they design applications to create VMs closest to the users and in order to meet any legal, compliance, or tax purposes.

> *Design Decision*
> *SDR reference Implementation has used "East US" region to deploy all the infrastructure resources.*

5.3. **Decision Summary**

Table 7 Subscriptions and Regions Decision Summary

Design	Decision
Select Azure Regions	One Azure region "East US" has been used to deploy all the infrastructure resources

6. Networking

6.1.　VNets and Subnets

Azure Virtual Network (VNet) is the fundamental building block for a private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, through internet, and on-premises networks. Specifically, VNet is similar to a traditional network that operators would use in their current/own data centers but brings additional benefits of employing Azure's infrastructure such as scale, availability, and isolation.

VNet could be divided into multiple ranges of IP addresses (subnets) for organization and security. Subnet delegation enables designation of a specific subnet for an Azure PaaS service of choice that needs to be injected into virtual network. Subnet delegation provides full control to the customer on managing the integration of Azure services into their virtual networks.
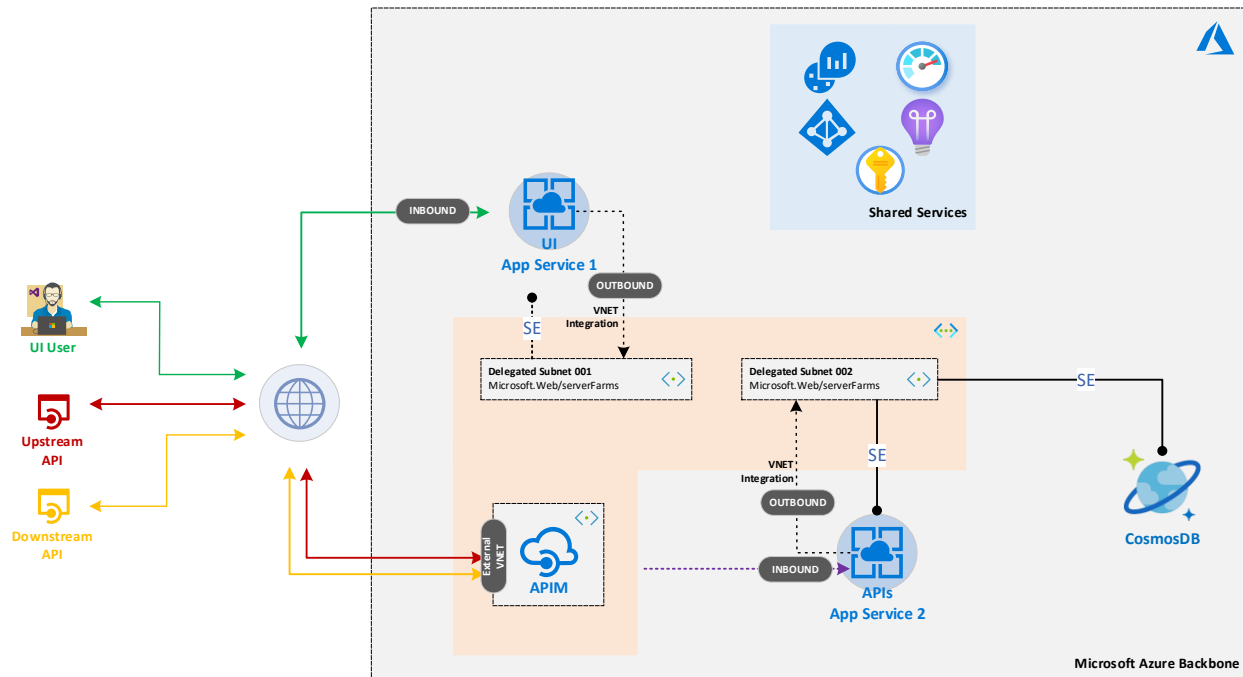
> **Design Decision**
> *SDR Reference Implementation has used 1 VNet, 1 Subnet and 2 Delegated Subnets per environment/region.*

6.1.1.　Networking Design Considerations

The figure below shows the information flow between different Azure Services and components for SDR Reference Implementation

Figure 2 SDR Reference Implementation Azure Networking

Key Networking Design Information:

- The Inbound access policies on App Service 1 that hosts UI, allows it to be accessed directly over the internet.
- The Outbound policies on App Service 1 allows it to communicate with other PaaS services through VNet Integration.
- The Inbound access policies on App Service 2 that hosts the API, prevent it from being accessed directly over the internet and is limited to VNet through Access Restriction.
- Outbound access policies on App Service 2 allow it to communicate with Cosmos DB through a VNet Integration
- API Management is the API Gateway for the Upstream and Downstream APIs. It is deployed in the VNet and allows the APIs hosted in App Service 2 to be accessed over the internet.

6.2. DDoS Protection

Azure has built in distributed denial-of-service (DDoS) protection to protect against distributed denial of service attacks on the public IPs resources within a VNet.

Figure 3 Azure DDoS Protection Plans

Feature	DDoS Protection Basic	DDoS Protection Standard
Active traffic monitoring & always on detection	Yes	Yes
Automatic attack mitigations	Yes	Yes
Availability guarantee	Azure region	Application
Mitigation policies	Tuned for Azure region traffic volume	Tuned for application traffic volume
Metrics & alerts	No	Real time attack metrics & diagnostic logs via Azure monitor
Mitigation reports	No	Post attack mitigation reports
Mitigation flow logs	No	NRT log stream for SIEM integration
Mitigation policy customizations	No	Engage DDoS experts
Support	Best effort	Access to DDoS Experts during an active attack
SLA	Azure region	Application SLA guarantee & cost protection
Pricing	Free	Monthly & usage based

Basic is on by default with no additional charges. Standard is optional with additional monthly and data usage charges. The difference between the two tiers is outlined above.

6.2.1. Basic DDoS Protection

The environment has leveraged the native platform-level DDoS protection options that are offered at no additional charge in Azure.

> **Design Decision**
> *SDR reference Implementation has leveraged the basic protection offered on each VNet by default.*

6.3.   Service Endpoints

6.3.1. Azure VNet Service Endpoints

VNet service endpoint provides secure and direct communication between Azure services over an optimized route using the Azure backbone network. Endpoints allow the user to secure their critical Azure service resources to only their virtual networks. Service Endpoints enables private IP addresses in the VNet to reach the endpoint of an Azure service without needing a public IP address on the VNet.

Service endpoints have been configured for App Service 1 (UI), App Service 2 (API) and Cosmos DB to enable communication between these services without having to route the communication outside the Azure backbone.

6.4. Decision Summary

Table 8 Networking Design Decisions

Design	Decision
VNet and Subnets	*1 VNet, 1 Subnet and 2 Delegated Subnets per environment/region.*
DDoS	Basic DDoS protection
VNet Service Endpoints	One for each App service and one for Cosmos DB

7. Connectivity

Following is illustration of how VNet, Subnet, Delegated Subnets, API Management, App Services and CosmosDB has been configured as part of the SDR Solution to allow connectivity between different systems & components.

Figure 4 SDR Reference Implementation Connectivity
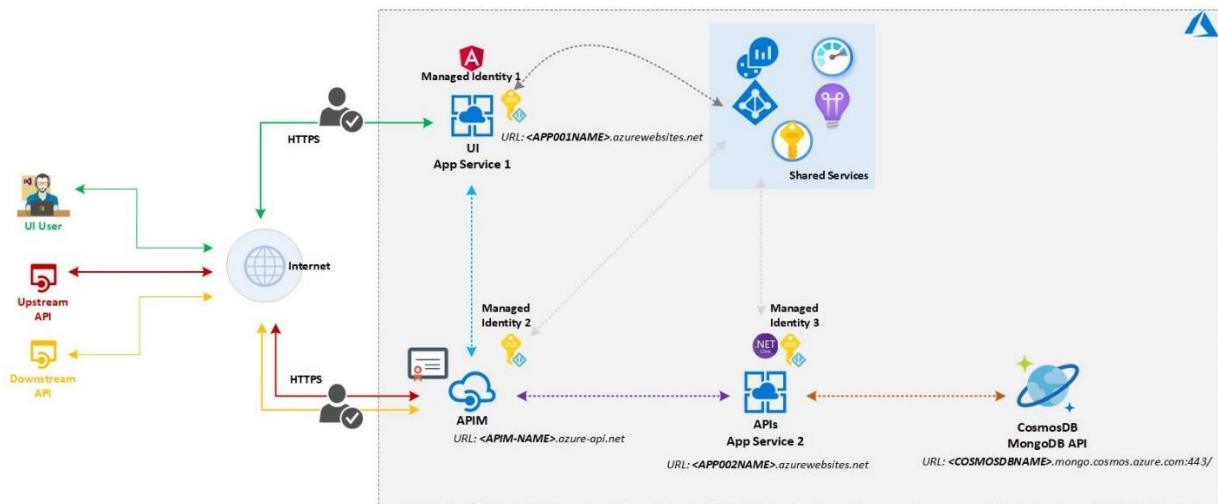
Communication Flow:

- Communication from Internet to User interface (UI) I App Service is allowed through Hypertext Transfer Protocol Secure (https) Protocol

- Upstream communication through the internet is handled by APIM which in turn communicates with API App Service. The API App Service forwards the call further to Cosmos DB.

- Downstream communication is also handled by APIM via Internet. APIM handles the calls to UI App Service and API App Service.

- API App Service and CosmosDB Mongo API App cannot be reached directly from the internet. All the calls to these resources are handled by APIM.

- The shared services in azure handle the requests from UI App Service, APIM, API App Service using OAuth2.0 authentication method

> *Design Decision*
> *SDR Reference Implementation has used API Management as the API Gateway for accessing SDR API endpoints over t he internet.*

7.1. Decision Summary

Table 9 Connectivity Decision Summary

Design	Decision
API Connectivity	API Management as API Gateway

8. Identity

8.1. Azure Active Directory and Federation

Azure Active Directory (Azure AD) is Microsoft's multi-tenant, cloud-based directory, and identity management service. Please refer Appendix A.5 for more detail on AAD versions.

Azure services at the enterprise level require the configuration of an Azure AD Tenant for the synchronization of enterprise IDs. While Live IDs are also an option for many Azure services, they are not recommended for large enterprises when compared to a centrally managed Active Directory infrastructure (that already follows many organizations best practices regarding

security, management, and user lifecycle) that can be synchronized into Azure. Additionally, there are some Azure services that will not permit logins at all using Live IDs (such as PowerBI).

8.1.1. AAD Tenant

> **Design Decision**
> *SDR Reference implementation has leveraged single Azure Active Directory (AAD) Tenant (*

8.1.2. Azure Active Directory License

There are several license plans available for the Azure AD service, including:

- Azure AD Free
- Azure AD Premium P1
- Azure AD Premium P2

Detailed offerings for Azure AD across different tiers are available on Azure AD Tier Comparision

> **Design Decision**
> *SDR Reference implementation has leveraged Free Tier of Azure AD Tenant*

8.2. Account Security

8.2.1. Multi Factor Authentication

Multi-Factor Authentication (MFA) is available at both P1 and P2 premium levels and is recommended for all administrative access at a minimum.

> **Recommendation**
> *SDR Technical team recommends leveraging Multi-Factor Authentication for all Administrative accounts as a minimum level of security. It is recommended that all users have their accounts secured with MFA to improve overall security and access to the Azure Tenant.*

8.3. Administrative Scope

8.3.1. Owner Permissions

As mentioned, it is recommended that no more than 3 accounts have Owner permissions over Subscriptions. Two (2) of those accounts can belong to specific individuals (system administrators), and the final account can be configured as a "break-glass" account (application owner) in the event those individual owners are unavailable to perform the required actions. If

Privileged Identity Management (PIM) is being utilized within AAD, then a break-glass account will not be necessary since the privileges can be raised when necessary to perform the tasks the user would with the break-glass account with Owner permissions.

8.4. Managed Identity

Managed identities provide an identity for applications to use when connecting to resources that support Azure Active Directory (Azure AD) authentication. Applications may use the managed identity to obtain Azure AD tokens. For example, an application may use a managed identity to access resources like Azure Key Vault where developers can store credentials in a secure manner or to access storage accounts.

Some of the benefits of using Managed identities:
- You don't need to manage credentials. Credentials are not even accessible to you.
- You can use managed identities to authenticate to any resource that supports Azure Active Directory authentication including your own applications.
- Managed identities can be used without any additional cost.

There are two types of managed identities:
- System-assigned Some Azure services allow you to enable a managed identity directly on a service instance. When you enable a system-assigned managed identity an identity is created in Azure AD that is tied to the lifecycle of that service instance. So, when the resource is deleted, Azure automatically deletes the identity for you. By design, only that Azure resource can use this identity to request tokens from Azure AD.
- User-assigned You may also create a managed identity as a standalone Azure resource. You can create a user-assigned managed identity and assign it to one or more instances of an Azure service. In the case of user-assigned managed identities, the identity is managed separately from the resources that use it.

> **Design Decision**
> SDR Reference Implementation *has leveraged system assigned managed identity for accessing the secrets from KeyVault via AppService*

8.5. Service Principal

An Azure Service principal is an identity created for use with applications, hosted services, and automated tools to access Azure resources. This access is restricted by the roles assigned to the service principal, giving you control over which resources can be accessed and at which level.

8.6. Role Based Access Control

Azure Role-Based Access Control (RBAC) manages user access and interaction with Azure resources. Azure RBAC is an authorization system built on Azure Resource Manager (ARM) which provides fine-grained access management of Azure resources. The fine-grain access approach provides the ability for separation of duties. Permissions are enforced by the creation of role assignments which control access to resources. Role assignments consist of three elements: security principal, role definition (role), and scope. Permissions can be restricted at the scope and in the role definition.

- Security Principal is an object which represents a user, group, service principal, or managed identity which is requesting access to Azure resources (e.g. – serviceprincipal01@contoso.com)
- A role is set of permissions (create, read, update, delete) mapped to an account. Within the scope, access can be more restrictive by assigning accounts different roles. Roles can be high-level (Owner) or more specific (Network Contributor)
- RBAC role assignments are scoped to a specific management group, subscription, resource group, or resource. A user given access to a single resource cannot access any other resources in the same subscription

It is recommended to follow starter RBAC strategy and best practices:

	Recommendation

⚐	- *Principle of least privilege access should always be adhered to. Only grant the minimal permissions needed to accomplish the task*
	 - *Least privilege is an end state – it requires a process to achieve it and this process must be established and followed*
	 - *It requires a combination of approaches:*
	    o *Limiting the count of administrators or members of privileged groups*
	    o *Delegating lesser privileges to accounts*
	    o *Provide privileges on-demand and revoke them once the task is completed*
	    o *Providing a process for emergency access and rare-use scenarios*
	 - *Periodic access reviews should be performed to ensure user who no longer require certain roles have those permissions revoked*
	 - *Users should generally not be given access to resources directly, but instead be added to Groups that have assigned roles (and later be removed from those Groups)*
	 - *Utilization of Privileged Identity Management (PIM) to provide time-based and approval-based role activation is very useful to avoid unnecessary and excessive permissions to users or groups for periods longer than necessary. Just-In-Time (JIT) access is a feature of PIM*

The below table shows some of the commonly utilized built-in Roles that are utilized within Azure, and what actions users/identities can take when assigned that role (either directly or through Groups). For this reason, it is important to limit Owners and Security Admins and utilize PIM and/or JIT:

Table 10 Identity Decision Summary

Azure Role Name	Create	Rename	Move	Delete	Assign Access	Assign Policy	Read
Owner	X	X	X	X	X	X	X
Contributor	X	X	X	X			X
Reader							X
User Access Administrator					X		
User Administrator					X		
Directory Reader							X

8.7. Decision Summary

Table 11 Identity Decision Summary

Design	Decision
Azure Active Directory Tenant	AAD tenant is used
Azure Active Directory Edition	Azure Active Directory Free tier is used
Admins and Administrative role Assignment	Number of accounts with Owner rights is limited to need basis only
Role-based access control (RBAC) Strategy	Best practice recommendations/guidance provided

9. Security

9.1. Azure Key Vault

Azure Key Vault is a service that is used to create, maintain, and store keys, secrets, and certificates that are leveraged by cloud resources, apps, and solutions. Key Vault supports the following features:

- Secure Key Management

- Encrypt and store keys / passwords (secrets)
- Certificate Management and automatic renewals

An Azure Key Vault provides native integration with a variety of Azure services and is deeply connected to Azure AD for authentication and authorization using RBAC.

SDR has leveraged Azure Key Vault to store client secrets which are utilized by UI and API application to connect with other PaaS services that are replaced (in application configuration files) during deployment instead of hard-coding of secrets in application code.

	•

	Design Decision *SDR Reference Implementation has leveraged the native Azure Key Vault for application secrets management.*

9.2. Certificates

Azure Active Directory (Azure AD) supports two types of authentications for service principals: password-based authentication (app secret) and certificate-based authentication. While app secrets can easily be created in the Azure portal, it's recommended that your application uses a certificate.

Certificates provide TLS client authentication to the API gateway. The API Management gateway can be configured to allow only requests with certificates containing a specific thumbprint. The authorization at the gateway level is handled through inbound policies.

API Management gateway enforces TLS authentication, and it can inspect the certificate contained within the client request and check for properties like below

1. Certificate Authority (CA): Only allow certificates signed by a particular CA
2. Thumbprint: Allow certificates containing a specified thumbprint
3. Subject: Only allow certificates with a specified subject
4. Expiration Date: Only allow certificates that have not expired

> **Design Decision**
> *SDR Reference Implementation has leveraged Certificates for both Upstream and Downstream communication through API Management and these Certificates are stored in APIM certificates.*

9.3. Decision Summary

Table 12 Security Decision Summary

Design	Decision
Key Vault Usage	One Key Vault per Environment
Key Vault Features	Key Vault CanNotDelete locks, purge protection, and soft delete has been used to avoid accidental or malicious deletion of vaults, keys, secrets, and other credentials
Certificate	Client Certificates with public key for API authentication

10. Resources

10.1. PaaS Components

10.1.1. App Services

Azure App Service is an HTTP-based service for hosting web applications, REST APIs, and mobile back ends. You can develop in your favorite language, be it .NET, .NET Core, Java, Ruby, Node.js, PHP, or Python. Applications run and scale with ease on both Windows and Linux-based environments. App Service not only adds the power of Microsoft Azure to your application, such as security, load balancing, autoscaling, and automated management. You can also take advantage of its DevOps capabilities, such as continuous deployment from Azure DevOps, GitHub, Docker Hub, and other sources, package management, staging environments, custom domain, and TLS/SSL certificates.

With App Service, you pay for the Azure compute resources you use. The compute resources you use are determined by the App Service plan that you run your apps on. For more details on App Service refer https://azure.microsoft.com/en-us/services/app-service/

> **Design Decision**
> *SDR Reference Implementation has used 2 App Services per environment/region - 1 for Frontend UI Apps and the other for Backend APIs.*

10.1.2. App Service Plans

In App Service (Web Apps, API Apps, or Mobile Apps), an app always runs in an App Service plan. In addition, Azure Functions also has the option of running in an App Service plan. An App Service plan defines a set of compute resources for a web app to run. These compute resources are analogous to the server farm in conventional web hosting. One or more apps can be configured to run on the same computing resources (or in the same App Service plan).

When you create an App Service plan in a certain region (for example, West Europe), a set of compute resources is created for that plan in that region. Whatever apps you put into this App Service plan run on these compute resources as defined by your App Service plan. Each App Service plan defines:

- Operating System (Windows, Linux)
- Region (West US, East US, etc.)
- Number of VM instances
- Size of VM instances (Small, Medium, Large)
- Pricing tier (Free, Shared, Basic, Standard, Premium, PremiumV2, PremiumV3, Isolated, IsolatedV2)

The pricing tier of an App Service plan determines what App Service features you get and how much you pay for the plan. The pricing tiers available to your App Service plan depend on the operating system selected at creation time. There are a few categories of pricing tiers:

- **Shared compute:** Free and Shared, the two base tiers, runs an app on the same Azure VM as other App Service apps, including apps of other customers. These tiers allocate CPU quotas to each app that runs on the shared resources, and the resources cannot scale out.
- **Dedicated compute:** Basic, Standard, Premium, PremiumV2, and PremiumV3 tiers run apps on dedicated Azure VMs. Only apps in the same App Service plan share the same compute resources. The higher the tier, the more VM instances are available to you for scale-out.
- **Isolated:** This Isolated and IsolatedV2 tiers run dedicated Azure VMs on dedicated Azure Virtual Networks. It provides network isolation on top of compute isolation to your apps. It provides the maximum scale-out capabilities.

> *Design Decision*
> *SDR Reference Implementation has used 2 App Service Plans per environment/region, 1 for Frontend App Service and the other for Backend App Service. The service plans are at individual resource level and are distinct for each App Service.*

10.1.3. Azure CosmosDB

Azure Cosmos DB is a fully managed NoSQL database for modern app development. Single-digit millisecond response times, and automatic and instant scalability, guarantee speed at any scale. Business continuity is assured with SLA-backed availability and enterprise-grade security. App development is faster and more productive thanks to turnkey multi region data distribution anywhere in the world, open-source APIs and SDKs for popular languages. As a fully managed service, Azure Cosmos DB takes database administration off your hands with automatic management, updates, and patching. It also handles capacity management with cost-effective serverless and automatic scaling options that respond to application needs to match capacity with demand.

> **Design Decision**
> *SDR Reference Implementation has used 1 Azure CosmosDB API for MongoDB per environment/region.*

10.1.4. API Management

API Management (APIM) is a way to create consistent and modern API gateways for existing back-end services. API Management helps organizations publish APIs to external, partner, and internal developers to unlock the potential of their data and services. Businesses everywhere are looking to extend their operations as a digital platform, creating new channels, finding new customers, and driving deeper engagement with existing ones. API Management provides the core competencies to ensure a successful API program through developer engagement, business insights, analytics, security, and protection. You can use Azure API Management to take any backend and launch a full-fledged API program based on it.

Figure 5 API Management
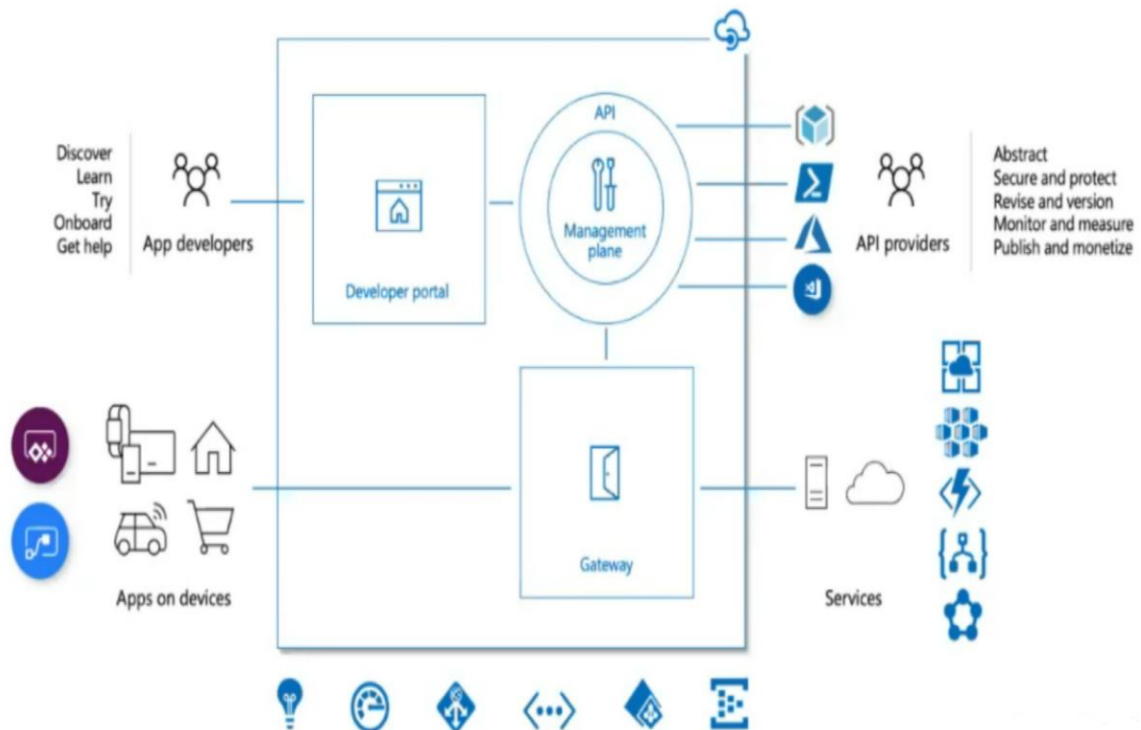
	Design Decision
	SDR Reference Implementation has used 1 API Management per environment/region.

11. Operations

11.1. Logging

Log analytics platform can gather different types of logs. Following log types can be collected and exported to different destination stores.

- **Azure Resource Logs**: Resource Logs provide information and insight into operations that were performed within an Azure resource (*the data plane*). Resource log content varies by the Azure resource type. These logs are made visible by sending them to a destination that can be a Log Analytics workspace, Azure Storage account or Azure Event Hub, and this is setup in the Diagnostic Settings of that resource.

- **Azure Activity Log**: There is a single log that provides insight into operations on each Azure resource in a subscription from the outside (*the management plane*). These are

what, who and when of any write operation taken on the resources in a Subscription. Activity Log of a Subscription can be exported to a Log Analytics workspace, Azure Storage account or Azure Event Hub as setup in the Diagnostic Setting.

- **Azure Active Directory Logs**: These logs contain the history of sign-in activity and audit trail of changes made in the Azure Active Directory for a tenant. This is a tenant level export and can be setup using Diagnostic setting of AAD and data collected will include data for all subscriptions in the AAD tenant.
- **Azure Flow Logs:** These logs contain the history of the ingress and egress traffic flow. Flow logs can be collected from Network Security Groups

Figure 6 Azure Platform Data Types
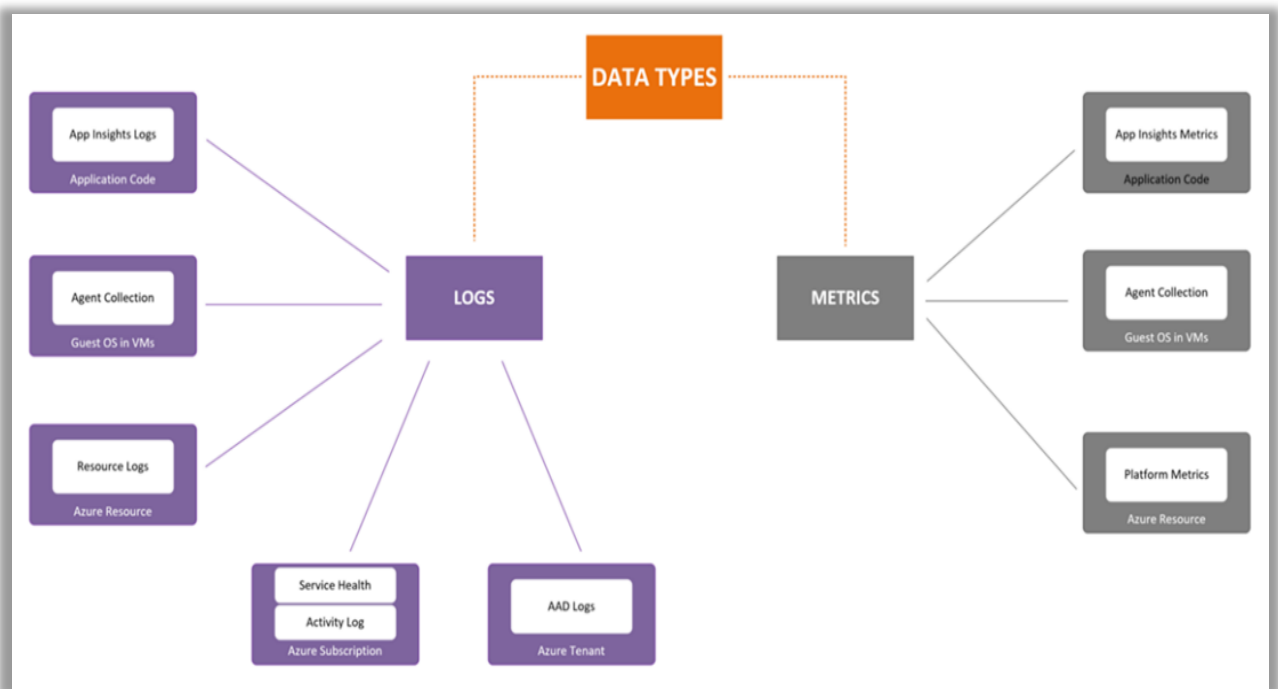
Below are the different integration options available based on the Log Categorization.

Table 13 Log Categories

Log Category	Log Type	Usage	Integration
Activity logs	Control plane events on Azure Resource Manager resources	Provides insight into the operations that were performed on resources in your subscription.	Rest API, Azure Monitor
Resource logs	Frequent data about the operation of Azure Resource Manager resources in subscription	Provides insight into operations that your resource itself performed.	Azure Monitor
AAD (Azure Active Directory) Logs and Reporting	Logs and reports	Reports user sign-in activities and system activity information about users and group management.	Graph API
Virtual machines and cloud services	Windows Event Log service and Linux Syslog	Captures system data and logging data on the virtual machines and transfers that data into a storage account of your choice.	Windows (using Windows Azure Diagnostics [WAD] storage) and Linux in Azure Monitor
Azure Storage Analytics	Storage logging provides metrics data for a storage account	Provides insight into trace requests, analyzes usage trends, and diagnoses issues with your storage account.	REST API or the client library
Application insight Logs	Logs, exceptions, and custom diagnostics	Provides an application performance monitoring (APM) service for web developers on multiple platforms.	REST API, Power BI
Process data / security alerts	Azure Security Center alerts, Azure Monitor logs alerts	Provides security information and alerts.	REST APIs

11.2.   Diagnostic Settings

Platform metrics are sent automatically to Azure Monitor Metrics by default and without configuration.

Platform logs, including the Azure Activity log and resource logs, provide detailed diagnostic and auditing information for Azure resources and the Azure platform they depend on. The Activity Log exists on it own but can be routed to other locations. Resource logs are not collected until they are routed to a destination.

Each Azure resource requires its own diagnostic setting, which defines the following criteria:

- Sources - The type of metric and log data to send to the destinations defined in the setting. The available types vary by resource type
- Destinations - One or more destinations to send to

With Azure diagnostic logs, you can view core analytics and save them into one or more destinations including:

- Azure Storage account
- Log Analytics workspace
- Azure Event Hubs

Table 14 Diagnostic Settings

Resource Name	Type	Category	Destination
VNet	Metrics	AllMetrics	Log Analytics Workspace
Application Insights	Logs	AppAvailabilityResults AppBrowserTimings AppEvents AppMetrics AppDependencies AppExceptions AppPageViews AppPerformanceCounters AppRequests AppSystemEvents AppTraces	Log Analytics Workspace
	Metrics	AllMetrics	Log Analytics Workspace
API Management	Logs	GatewayLogs WebSocketConnectionLogs	Log Analytics Workspace

	Metrics	AllMetrics	Log Analytics Workspace
Cosmosdb	Logs	DataPlaneRequests MongoRequests QueryRuntimeStatistics PartitionKeyStatistics PartitionKeyRUConsumption ControlPlaneRequests TableAPIRequests	Log Analytics Workspace
	Metrics	Requests	Log Analytics Workspace
App Service Plan	Metrics	AllMetrics	Log Analytics Workspace
App Service	Logs	AppServiceHTTPLogs AppServiceConsoleLogs AppServiceAppLogs AppServiceAuditLogs AppServiceIPSecAuditLogs AppServicePlatformLogs	Log Analytics Workspace
	Metrics	All Metrics	Log Analytics Workspace
KeyVault	Logs	AuditEvent AuditPolicyEvaluationDetails	Log Analytics Workspace
	Metrics	All Metrics	Log Analytics Workspace

> ***Design Decision***
> *SDR Reference Implementation has used one Log Analytics Workspace per environment/region. All the Diagnostic Setting logs and metrics collected for all the infrastructure resources are sent to this shared Log Analytics Workspace.*

11.3. Log Analytics Workspace

Log Analytics Workspace is a tool in the Azure portal used to edit and run log queries with data in Azure Monitor Logs. You may write a simple query that returns a set of records and then use features of Log Analytics to sort, filter, and analyze them. Or you may write a more advanced query to perform statistical analysis and visualize the results in a chart to identify a particular trend. Whether you work with the results of your queries interactively or use them with other Azure Monitor features such as log query alerts or workbooks, Log Analytics is the tool that you're going to use write and test them.

The Diagnostic Setting logs and metrics collected for all the infrastructure resources will be sent to Log Analytics Workspace.
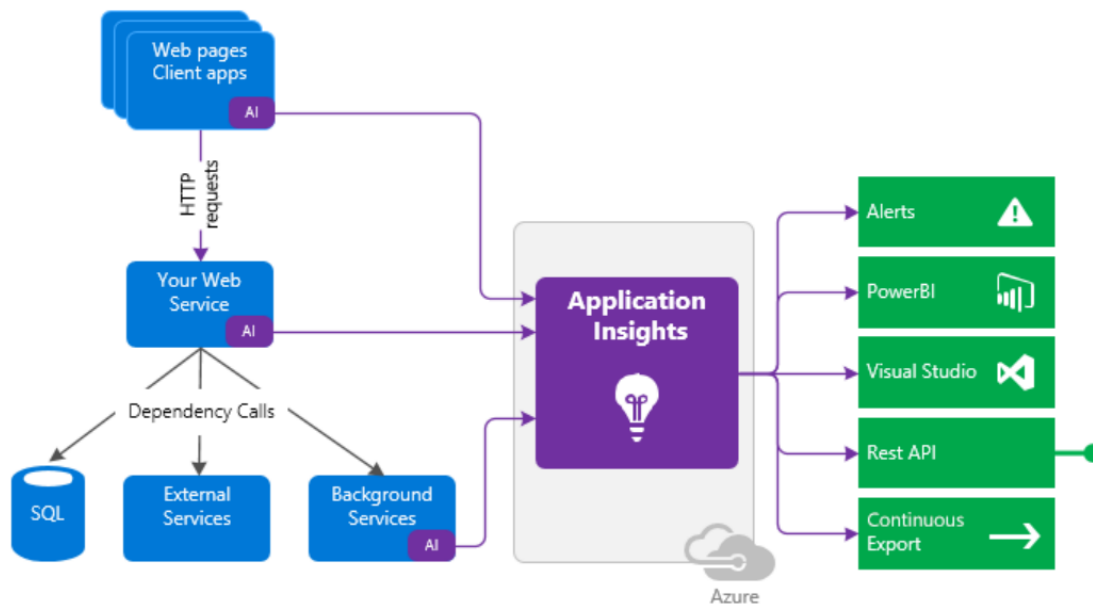
	Design Decision
	SDR Reference Implementation has deployed one Log Analytics Workspace per environment/region.

11.4. Application Insights

Application Insights, a feature of Azure Monitor, is an extensible Application Performance Management (APM) service for developers and DevOps professionals. Use it to monitor your live applications. It will automatically detect performance anomalies, and includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .NET, Node.js, Java, and Python hosted on-premises, hybrid, or any public cloud. It integrates with your DevOps process and has connection points to a variety of development tools.

Application Insights deployed is being used by App Services and API Management.

Figure 7 Application Insights

	Design Decision
	SDR Reference Implementation has one Application Insights deployed per environment/region.

11.5. Monitoring

11.5.1. Azure Monitor

Azure Monitor delivers a comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments. It helps you understand how your applications are performing and proactively identifies issues affecting them and the resources they depend on. Azure Monitor enables you to:

- Detect and diagnose issues across applications and dependencies with Application Insights.
- Correlate infrastructure issues with Azure Monitor for VMs and Azure Monitor for Containers.
- Drill into your monitoring data with Log Analytics for troubleshooting and deep diagnostics.
- Support operations at scale with smart alerts and automated actions.
- Create visualizations with Azure dashboards and workbooks.
- All data collected by Azure Monitor fits into one of two fundamental types, metrics, and logs.

Metrics are numerical values that describe some aspect of a system at a particular point in time. They are lightweight and capable of supporting near real-time scenarios. Logs contain different kinds of data organized into records with different sets of properties for each type. Telemetry such as events and traces are stored as logs in addition to performance data so that it can all be combined for analysis.

Capabilities:
- **Continuous monitoring**: Azure Monitor collects performance statistics from the Windows performance counter and sends it to the portal.
- Custom alert configuration.
- **Availability testing** — in the form of URL-based ping and web-based tests.
- Data visualization for a visual representation of various data, including that of user flows.
- **Tracking and logging of user sessions**: the function enables the analysis of the user journey, which contributes to a better understanding of the application's pain points.
- **Smart detection** — a recommendation functionality that is powered by machine learning algorithms.
- Performance testing based on web tests.
- Application map that visualizes system components, such as the server, the database, and other resources.

- **Snapshot debugger** — a component that saves code status at the time of an exception for further check

Figure 8 Azure Monitor
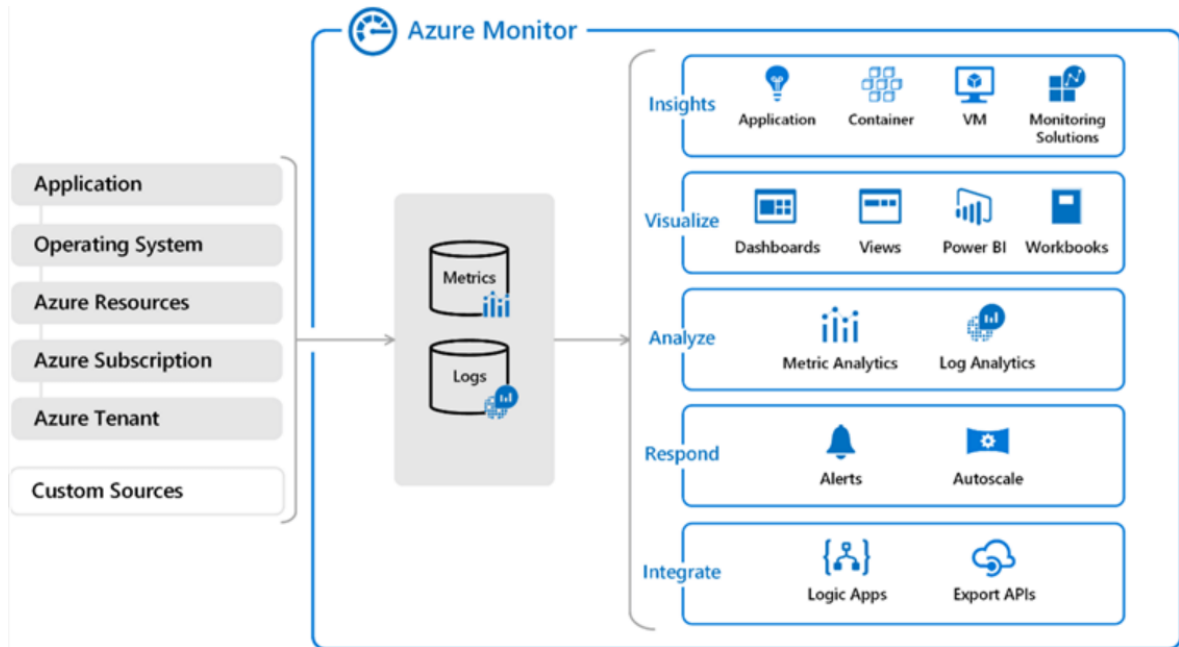
🖐	***Design Decision*** *SDR Reference Implementation has leveraged Azure native tool for monitoring. In this case, it is Azure Monitor.*

11.5.2. Azure Network Watcher

Azure Network Watcher provides the capabilities to diagnose, view metrics, monitor and enable or disable logs for resources in an Azure VNet. It is designed to monitor and repair the network health of IaaS (Infrastructure-as-a-Service) resources such as VMs, VNet, Load balancers, Application Gateways, Network Interfaces, Public IPs.

Network Watcher will be automatically enabled when the VNet is created in a particular region. There is no impact to the resources or associated charge for automatically enabling Network Watcher.

> **Design Decision**
> *SDR Reference Implementation has leveraged Azure Network Watcher for Network flow diagnostics.*

11.6.   Decision Summary

Table 15 Operations Decision Summary

Design	Decision
Infrastructure Logging & Monitoring	Cloud native tools for logging and Infrastructure Monitoring (Log Analytics & Azure Monitor) have been used

12.   References

- Define your naming convention - Cloud Adoption Framework | Microsoft Docs
- Recommended abbreviations for Azure resource types - Cloud Adoption Framework | Microsoft Docs

Appendix- A

This section lists the links to documentation sources referred during design discussions and decisions made for the SDR Reference Implementation Infrastructure Platform Design.

S. No.	Topic	Link
A.1	Azure Best Practices for Resource Naming Convention	https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-naming
A.2	Naming rules and restrictions for Azure resources	https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/resource-name-rules#:~:text=Azure%20virtual%20machines%20have%20two%20distinct%20names%3A%20resource,to%2064%20characters.%20Lowercase%20letters%2C%20numbers%2C%20and%20hyphens.
A.3	Azure subscription and service limits, quotas, and constraints	https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/azure-subscription-service-limits
A.4	Azure geographies	https://azure.microsoft.com/regions/
A.5	What is Azure Active Directory?	https://docs.microsoft.com/en-us/azure/active-directory/active-directory-whatis
A.6	Virtual network service endpoints for Azure Key Vault – Trusted Services	https://docs.microsoft.com/en-us/azure/key-vault/general/overview-vnet-service-endpoints#trusted-services
A.7	Cloud Adoption Framework	Microsoft Cloud Adoption Framework for Azure - Cloud Adoption Framework | Microsoft Docs