# Platform Agnostic Recommendations for Study Definition Repository (SDR)

# Release Version 4.0

# Document History

| Version No. | Date | Author | Revision Description |
|---|---|---|---|
| V1.0 | March 22, 2022 | ACN Team | Initial Version |
| V2.0 | March 23, 2023 | ACN Team | Updated list of resources in other platforms for newly added azure components in SDR Release V2.0 |
| V3.0 | October 27, 2023 | ACN Team | Updated architecture diagram after incorporating API-Key Authorization in SDR RI APIs |
| V4.0 | March 25, 2024 | ACN Team | Added a new column in the recommendations table and listed possible platform independent options that can be leveraged. |
| | | | |

# Table of Contents

# 1. SDR Overview

The Study Definition Repository is a vision to catalyze industry-level transformation, enabling digital exchange of study definition (e.g., protocol) information by collaborating with technology providers and standards bodies and is based upon a standardized model – the CDISC Unified Study Definitions Model (USDM). The SDR seeks to transform the drug development process by enabling a digital workflow to move from a current state of manual asset creation to a future state of fully automated and dynamic readiness to support clinical study execution.

The architecture for the SDR Reference Implementation has been designed to achieve the following key Objectives, and components are chosen in a way that avoid the architecture being tied to specific hardware, operating systems, or tools.

Cloud Agnostic / Open-Source – Create an application that is relatively cloud agnostic from an implementation perspective by choosing the technology stack and cloud components/services that offer extensibility and portability to the application.

Accelerate study start-up / execution by enabling the automation of data flow to downstream clinical systems reducing the need for duplication, manual input and transcription.

Reduce Manual input by creating an application that automates data flow.

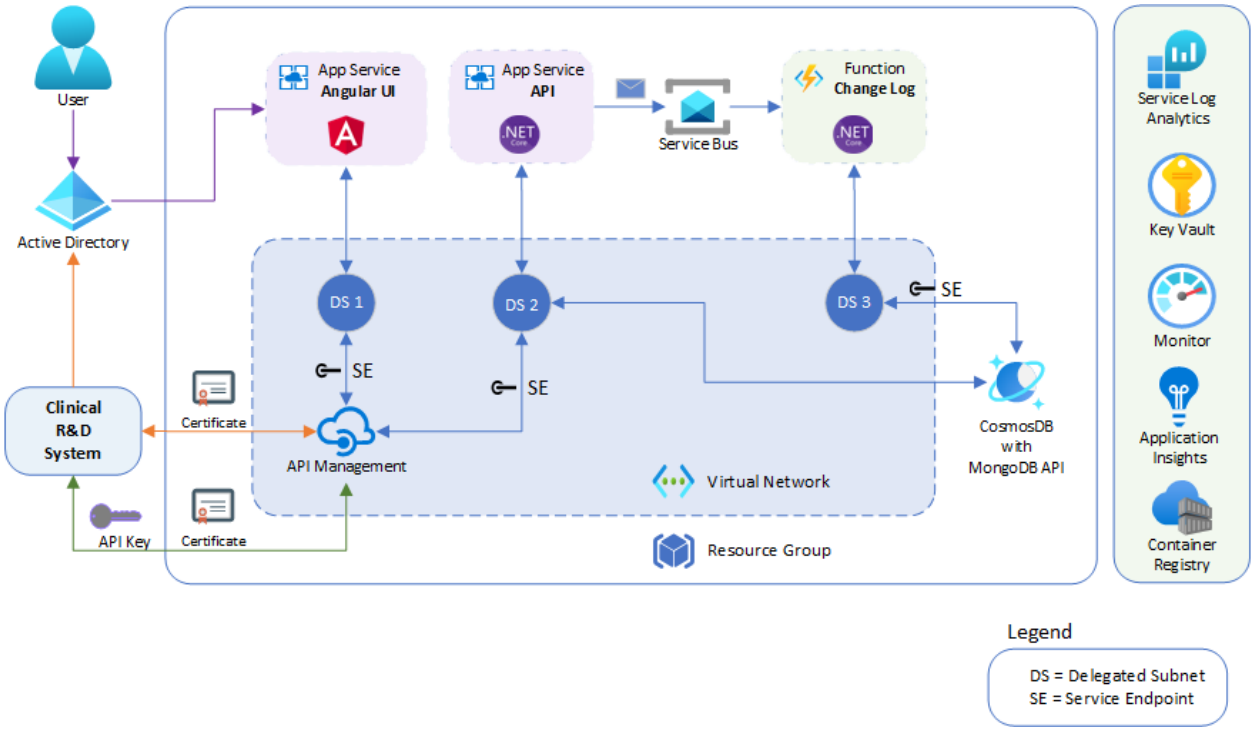
# 2. SDR Reference Implementation Architecture

The SDR Reference Implementation is an attempt to demonstrate the vision of the DDF initiative, not a commercial product. TransCelerate and its vendor partners built the RI on Azure, but this resulted out of the practical necessities associated with making a reference implementation available to demonstrate the SDR's capabilities. Companies are free to develop SDRs, without or without the SDR source code available on GitHub, for any cloud environment, and are encouraged to do so. This Guide is designed to facilitate the use/development in cloud environments other than Azure.[1]

Figure below depicts a high-level architecture of the SDR Reference Implementation which is built using Angular Front End and .NET Core Backend and deployed on Microsoft Azure Cloud2. The solution architecture components are chosen in a way so as to make the future release of the reference implementation portable to other deployment environments such as Amazon Webservices Cloud (AWS)2, and Google Cloud Platform (GCP)2.

[1] To be clear, TransCelerate does not endorse any particular software, system, or service. Users are free to download the source code for the SDR from GitHub and design their own implementations in whatever environments they choose.

[1] To be clear, TransCelerate does not endorse any particular software, system, or service. And the use of specific brands of products or services by TransCelerate and its collaboration partners in developing the SDR Reference Implementation should not be viewed as any endorsement of such products or services. To the extent that the SDR Reference Implementation incorporates or relies on any specific branded products or services, this resulted out of the practical necessities associated with making a reference implementation available to demonstrate the SDR's capabilities.




## 3. Porting SDR to Other Cloud Environments

The table below gives high level suggestions for moving SDR Architectural and Application components to other Cloud Environments such as Amazon Web Services (AWS) and Google Cloud Platform (GCP). It does not represent all possible options – vendors are free to choose whichever solutions work best for them.

| Architecture Area | Configuration Items | Public Cloud | | | Comments | Platform Independent |
|---|---|---|---|---|---|---|
| | | Azure | AWS | GCP | | |
| **Account Management** | Cloud Platform Logical Landing Zone | Subscription | Account | Google Cloud Project | | |
| **Dev Ops** | Tools for Version Control | Github (Repos) | Github (Repos) | Github | | GitHub |
| | Tools for Build & Deployment | Github (Pipelines) | Github (Pipelines) | Cloud Build | | GitHub, Jenkins |
| | Tools for Testing | JMeter, Postman | JMeter, Postman | Jmeter, Postman | | |
| | forwiki | Github | Github | No specific propritary of GCP | | GitHub |
| | Continous Delivery (IaC) | Terraform | Terraform | Terraform | Infrastructure as a Code(Terraform Scripts needs to be updated as per the cloud resourcing classification ) | |
| | | | | | | |
| **Governance** | Tagging | Azure Tags | AWS Tagging | Labels | | NA |
| | Naming Convention | https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/ready/azure-best-practices/resource-naming | | https://cloud.google.com/compute/docs/naming-resources | | NA |

| Architecture Area | Configuration Items | Public Cloud | | | Comments | Platform Independent |
|---|---|---|---|---|---|---|
| | | Azure | AWS | GCP | | |
| | Resource Group | Azure Resource Groups | AWS Resource Group | NA | | NA |
| | Cost Management | Azure Cost Management Billing | Cost Explorer | Cost Management | | NA |
| | | | | | | |
| **Subscription & Regions** | Total Subscriptions (Dev & Demo) | 2 | 2 | 2 | | NA |
| | Total Regions (US East) | 1 | 1 | 1 | | NA |
| | | | | | | |
| **Networking** | IPAM | /24 | /24 | /24 | | |
| | VNET for PaaS integration | 1 | 1 | 1 | | |
| | Subnet | 3 Delegated Subnets<br>1 subnet for APIM | Subnet | Subnet | | |
| | DNS | Azure DNS | Route 53 | Cloud DNS | | |
| | | | | | | |
| **Connectivity** | Remote Access to internet facing resources | Direct connect from Internet | Depends how it has been setup | Depends how it has been setup | | Depends how it has been setup |
| | | | | | | |
| **Identity** | Identity Provider | Azure AD | AWS Identity Services( AWS Managed Microsoft AD) | Cloud Identity | minor code changes needed on SDRUI and SDR API apps to work with other Active Directory or Auth Systems | PingIdentity, Okta, Keycloak |
| | Users | Active Directory Users | Users | IAM | | PingIdentity, Okta, Keycloak |
| | Groups | AAD Security Groups | Groups | IAM | | PingIdentity, Okta, Keycloak |

| Architecture Area | Configuration Items | Public Cloud | | | Comments | Platform Independent |
|---|---|---|---|---|---|---|
| | | Azure | AWS | GCP | | |
| | Service Principals | AAD SP | AWS Service Principal | Service Accounts | | NA |
| | Managed Identity | Azure Manged Identities | Identity Providers | Identity and Access Management | | NA |
| | RBAC | AAD & Subscription roles | AWS Roles | GCP Roles | | Keycloak, Okta |
| | | | | | | |
| **Security** | Security Monitoring | Azure Defender | AWS Shield | Security Command Center | | |
| | Baseline Policy | Microsoft Security Baseline | AWS Security | | | |
| | Key Management | Azure Key Vault | AWS Secret, AWS KMS | Secret Manager, Cloud KMS | | HashiCorp Vault |
| | DDOS | Basic DDOS for VNET | AWS Shield | Google Cloud Armor | | |
| | APIM Inbound Control | APIM Inbound Policies | API Management | Apigee API Management | | Kong Gateway, Tyk Cloud, KrakenD API Gateway and Mulesoft API Management. |
| | Cosmos DB Inbound Control | Managed Identity | AWS IAM | Cloud Identity | | NA |
| | | | | | | |
| **Resource** | Service to Host Frontend UI | App Service instance 1 | AWS BeanStalk | App Engine | Configuration changes to AWS Beanstalk, Google App Engine to be deploy the SDR code | Kubernetes Cluster (Containerized App) |
| | Service to Host APIs | App Service instance 2 | AWS BeanStalk | App Engine | Configuration changes to AWS Beanstalk, Google | Kubernetes Cluster (Containerized App) |

| Architecture Area | Configuration Items | Public Cloud | | | Comments | Platform Independent |
|---|---|---|---|---|---|---|
| | | Azure | AWS | GCP | | |
| | | | | | App Engine to be deploy the SDR code | |
| | API Gateway | API Management | AWS API Gateway | APIgee API management | | Kong Gateway, Tyk Cloud, KrakenD API Gateway and Mulesoft API Management. |
| | No SQL Database | Cosmos DB with Mongo API | Document DB | Cloud BigTable | AWS DocumentDB can be used with existing configuration | MongoDB Atlas |
| | Storage to host TF State File | Azure Storage Account | s3 bucket | Cloud Storage | | NA |
| | Storage to host Other Bulk import/Export | Azure Storage Account | s3 bucket | Cloud Storage | | Cloud Native Services |
| | Message broker with message queue | Azure Service Bus | SNS | Google Cloud Pub/Sub | | RabbitMQ, ActiveMQ |
| | Service to process messages in queue | Azure Function App | AWS Lambda | GCP Functions | | OpenWhisk, OpenFaas, Knative and Fn Project. |
| | Service to build, store, and manage container images and related artifacts | Azure Container Registry | Amazon ECR | Container Registry | | Docker Registry |
| | | | | | | |
| **BC, HA and DR** | Backup | Member company preference | Member company preference | Member company preference | | |
| | HA | PaaS Services as per MS SLA | Paas Services as per SLA | Paas Services as per SLA | | |

| Architecture Area | Configuration Items | Public Cloud | | | Comments | Platform Independent |
|---|---|---|---|---|---|---|
| | | Azure | AWS | GCP | | |
| | DR | PaaS Services as per MS SLA | Paas Services as per SLA | Paas Services as per SLA | | |
| | | | | | | |
| **Operations** | Logging | Azure Monitor Logs | Cloudwatch -Logs Insights | Cloud Logging | | New Relic, SigNoz and Graylog |
| | Application Performance Management | App Insights | AWS X-Ray | Cloud Trace | | |
| | Monitoring | Azure Monitor | CloudWatch | Cloud Monitoring | | |
| | Security & Compliance Monitoring | Azure Defender | AWS Guard Duty, AWS Security Hub | Security Command Center | | |