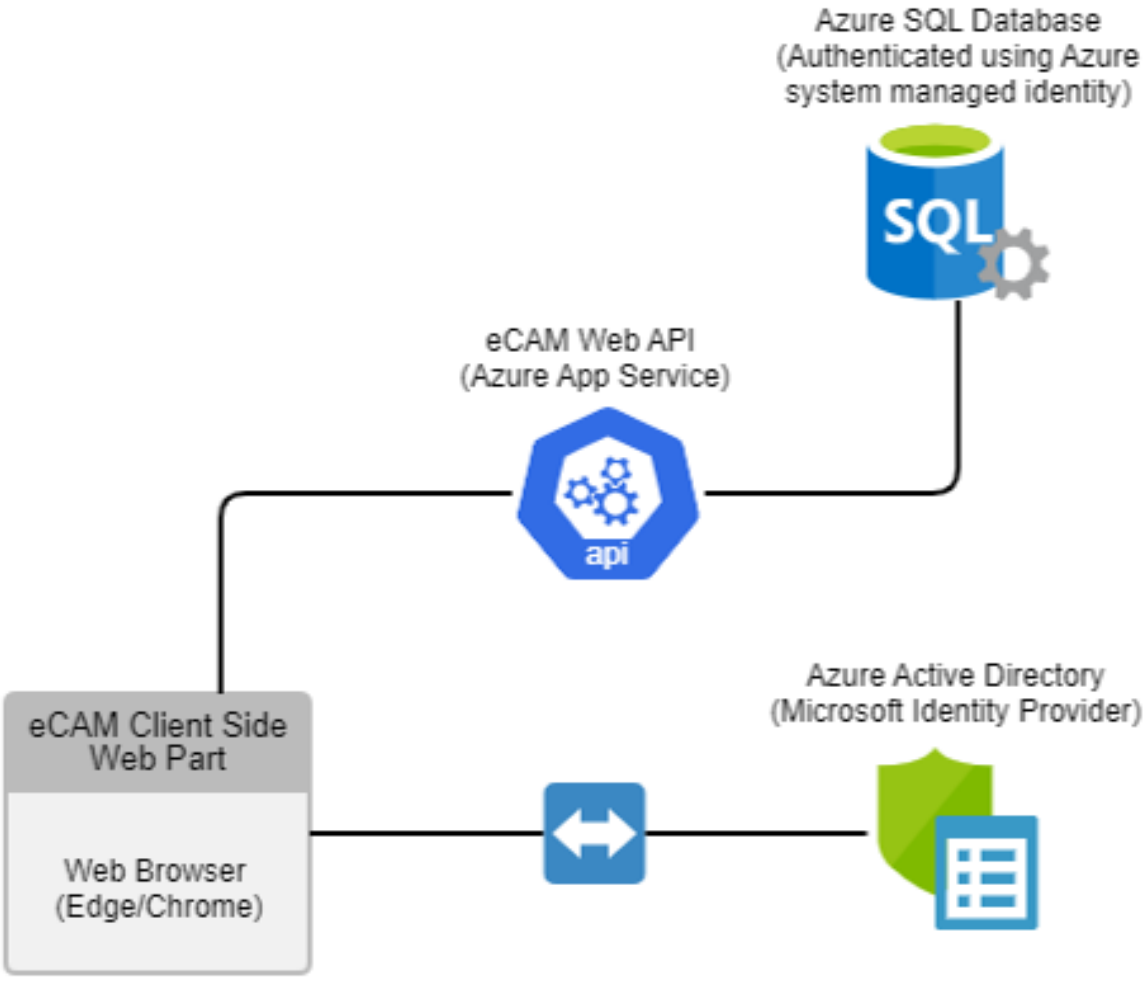
Laboratory EVMS Architecture Overview



EVMS Application

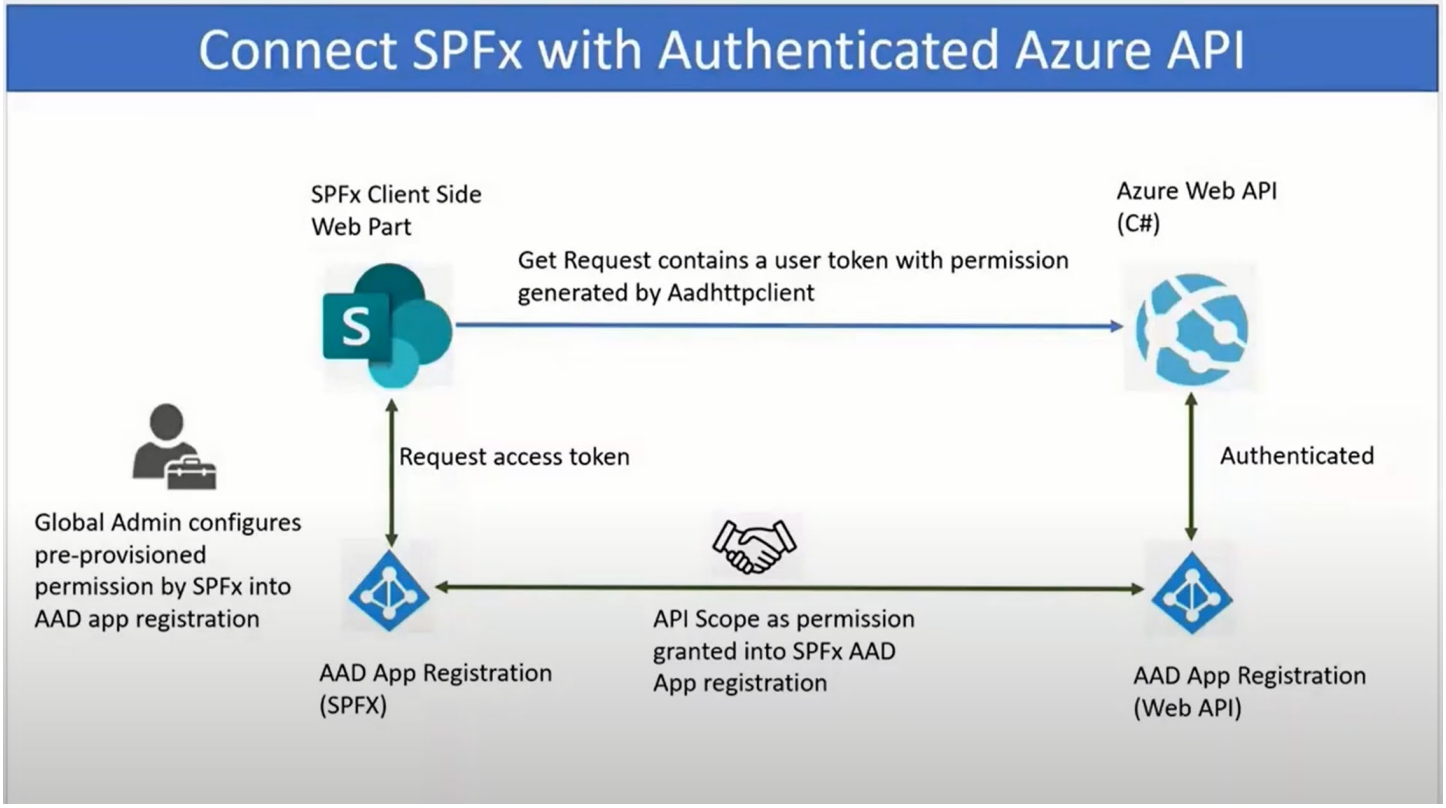
Earned Value Management (EVM) is a project performance management methodology that integrates cost, schedule, technical scope, and risk to assess progress against a baseline, use that information to identify problems, and forecast cost (and, to a certain extent, schedule) at completion. An Earned Value Management System (EVMS) is the people, processes and tools used to perform EVM for an organization. EVMS is a web-based set of tools that has been designed by Laboratory and is part of their EVMS. EVMS is currently hosted in Microsoft SharePoint and the tools are developed using client-side scripting libraries and presented using SharePoint Framework client-side web parts. These web parts request data stored in Azure SQL Databases via an Azure Web API.

EVMS Application Network

Users will access the SharePoint EVMS application using a web browser. SharePoint sites are only available via https therefore all transmissions will utilize SSL. When the user authenticates with Azure Active Directory (AAD) the browser would call the Microsoft AAD authentication endpoint using https to get their token. Once authenticated, the user would access a SharePoint page that has EVMS Client Web Part which would make https requests for data from the web API hosted in Microsoft Azure. The web API will connect to an EVMS

Azure SQL Database for data. The Azure web API will authenticate using windows credentials. Being that both the web API and SQL database are in Azure we can utilize an Azure system-assigned managed identity for authentication. What that means is the Azure web API process will be assigned an AAD app principal that will be used to authenticate with SQL Server. Because it is Azure system-assigned managed identity we don't have to update the app secret (certificate) that is associated with the app principal. You can think of this as the process identity that is used with an application pool in an IIS website. For more information about Azure system-assigned managed identity please see: https://learn.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/how-managed-identities-work-vm

EVMS Application Authentication



When a user accesses an EVMS project site in Microsoft SharePoint, they will authenticate with Azure Active Directory and receive an Access Token. That access token is used to authenticate with Microsoft SharePoint and with the Microsoft Azure App Service / Web API. The Web API will use an Azure system-assigned managed identity to authenticate using Windows Authentication with Azure SQL Databases.

Authentication and Authorization Details:

Authentication and Roles implemented in the the IdP:

ECAM application would use the built-in Microsoft Identity Provider (IdP) which would use Azure AD (AAD) to authenticate EVMS users. Would implement 2 API roles in the IdP which would include:

- EVMS.Admins
- EVMS.Users

There would be a Microsoft 365 (M365) Group for each of the roles listed above. The M365 group would manage the membership for the roles. An EVMS user can be in multiple roles.

Authorization:

Implement a Custom “User Claims Principal Factory” with the Azure the Identity Provider (IdP) to add a custom claim named “project” to a user’s access token. The “project” claim is a list of all the projects that the user has access to along with their “access level.” This information will come from the EVMS App Database table and will only be generated once at login opposed to each http request.

Implement an “authorization filter” inside the Web API project to check if the current user has been authorized to perform an action on a specific project.

Reference: Authentication and Authorization in ASP.NET Web API | Microsoft Learn

Need to store authorization attributes in a SQL data table. Table would relate ProjectInfo to UserInfo and include some level of permissions.

Fields:

- ProjectID
- UserID
- AccessLevelID

At this time, we have 2 access levels:

- Project Admin
- Project User

Authorization Process:

An http request will have a header named “project” which establishes the project context for the request. The Web API will read the header value in the Authorization Filter method and compare the project value to the list of projects in the access token. The list of projects in the access token are added in the Custom Claims Provider. If the project listed in the header is not present in the “project” claim then a 401 will be returned. If there is a match then the Authorization Filter will allow the request to be passed onto the controller. The controller methods will be decorated with “Authorization” blocks which will check “access level” that is needed to call the method. The “access level” values would include the values “Project Admin” and / or “Project User.” Claims are process for each request. The controller will be secured by roles which would include “EVMS.Admins” and / or “EVMS.Users.”

Environments / Tiers

DEV, QA and Production tiers to be implemented as follows:

For production tier there would be the following:

- SharePoint site collection that would have a subsite for each Laboratory project
- Azure App Service to host the Production version of the Web API
- Azure Key Vault that would be used to store the passwords for the SQL Server user that would be used in the SQL Connection from the Web API to the SQL Server in the DMZ. Each project would have a SQL user.
- SQL Database Server in the Laboratory DMZ with a set of databases, one configuration database then a database per Laboratory Project.

The QA and DEV tier would be the same as Production but we would only utilize on SQL Database Server in the Laboratory DMZ for both QA and DEV. We would also just have one Azure Hosting Plan for both QA and DEV. But QA and DEV would each have their own Azure App Service. Think of Azure Hosting Plan as an IIS web server and an Azure App Service as an IIS web site. We would maintain separate Azure key vault for QA and DEV. The key vault would encrypt the SQL Server user password when storing it but would be seen in plain text in the Azure portal. We are able to lock down the Azure key vault secrets to only be accessed by certain users and the Azure managed identity which is the process account that is used with the Azure App Service.

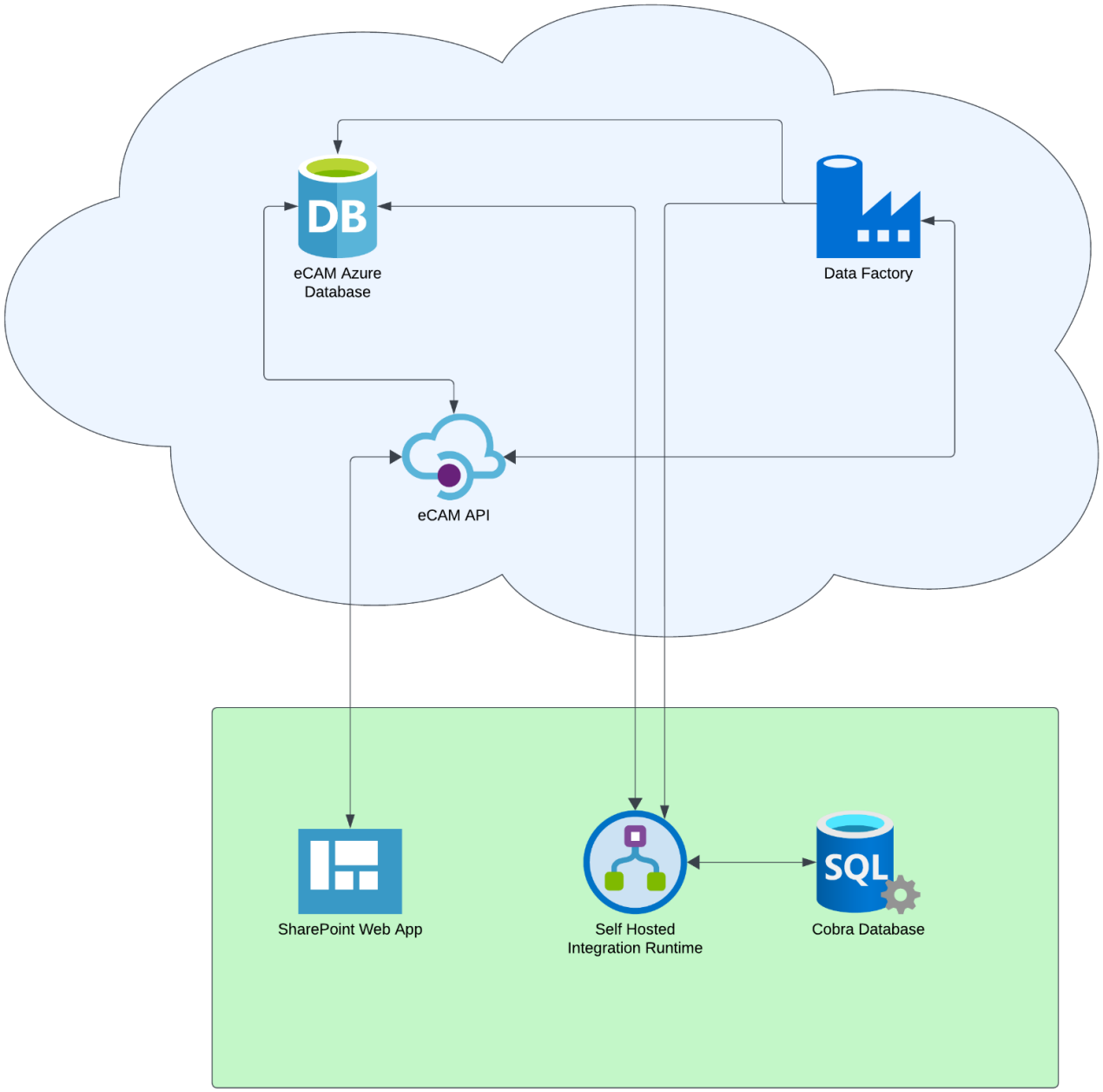
Azure Resources for EVMS Application

Resource Type	Name	Description	Cost
Resource Group	rg-ecam-devqa	Container to hold DEV and QA resources	0
App Hosting Plan	asp-ecam-devqa	Web server - Basic B2 - 1GB RAM, 10 GB of storage - $53 per month if running 100% of the time. But we would setup auto-shutdown for off business hours then if we go on a long period of no projected development then scale down to free app hosting plan. This App Hosting Plan would be setup under Dev/Test Azure licensing program. Cost to run 100% of the time would be $64.24. Would probably run a schedule to auto-shutdown for 12 hours during weekend and 24 hours on weekend which leaves operating 35% of the time. I bumped projected cost to 50% of the time.	32
App Service	app-ecam-api-dev	Web site - DEV web api	0
Identity Provider	app-ecam-api-dev-ip	Identity provider that would authenticate uses with Azure Active Directory and maintain a collection of roles that would be used to authorize users to call certain web methods or work with a specific project.	0
Azure SQL Server	sql-ecam-dev	Azure SQL Database elastic pool for all tiers. Would be able to house up to 100 DBs.	110.27
Azure SQL Database	sqldb-ecam-app-dev	ECAM application database. This will house all the projects configuration data.	0
Azure SQL Database	sqldb-ecam-proj-dev	ECAM default project content database. This will house all the project EVMS's project data and the data tables that are synchronized from Laboratory's Cobra operational databases.	0
Azure SQL Database	sqldb-ecam-proj-mpex-dev	ECAM MPEX project content database. This will house all the project EVMS's project data and the data tables that are synchronized from Laboratory's Cobra operational databases.	0

Azure SQL Database	sqldb-ecam-proj-siprc-dev	ECAM SIPRC project content database. This will house all the project EVMS's project data and the data tables that are synchronized from Laboratory's Cobra operational databases.	0
Azure SQL Database	sqldb-ecam-proj-rfp-dev	ECAM RPF project content database. This will house all the project EVMS's project data and the data tables that are synchronized from Laboratory's Cobra operational databases.	0
App Service	app-ecam-api-qa	Web site - QA web api	0
Identity Provider	app-ecam-api-qa-ip	Identity provider that would authenticate uses with Azure Active Directory and maintain a collection of roles that would be used to authorize users to call certain web methods or work with a specific project.	0
Azure SQL Database	sqldb-ecam-app-qa	ECAM application database. This will house all the projects configuration data.	0
Azure SQL Database	sqldb-ecam-proj-qa	ECAM default project content database. This will house all the project EVMS's project data and the data tables that are synchronized from Laboratory's Cobra operational databases.	0
Azure SQL Database	sqldb-ecam-proj-mpex-qa	ECAM MPEX project content database. This will house all the project EVMS's project data and the data tables that are synchronized from Laboratory's Cobra operational databases.	0
Azure SQL Database	sqldb-ecam-proj-siprc-qa	ECAM SPIRC project content database. This will house all the project EVMS's project data and the data tables that are synchronized from Laboratory's Cobra operational databases.	0
Azure SQL Database	sqldb-ecam-proj-rfp-qa	ECAM RPF project content database. This will house all the project EVMS's project data and the data tables that are synchronized from Laboratory's Cobra operational databases.	0
Resource Group	rg-ecam-prod	Container to hold Production resources	0
App Hosting Plan	asp-ecam-prod	Web server - Standard S1 - 1GB RAM, 50 GB of storage - $73 per month if running 100% of the time	73
App Service	app-ecam-api-prod	Web site - Production web api	0
Identity Provider	app-ecam-api-prod-ip	Identity provider that would authenticate uses with Azure Active Directory and maintain a collection of roles that would be used to authorize users to call certain web methods or work with a specific project.	0

Azure Key Vault	kv-ecam-prod	Container to store information encrypted and secure access that information. We would store any app principal secrets and sql credentials. $0.03/10,000 transactions	0.03
Azure SQL Database	sqldb-ecam-app-prod	ECAM application database. This will house all the projects configuration data.	0
Azure SQL Database	sqldb-ecam-proj-prod	ECAM default project content database. This will house all the project EVMS's project data and the data tables that are synchronized from Laboratory's Cobra operational databases.	0
Azure SQL Database	sqldb-ecam-proj-mpex-prod	ECAM MPEX project content database. This will house all the project EVMS's project data and the data tables that are synchronized from Laboratory's Cobra operational databases.	0
Azure SQL Database	sqldb-ecam-proj-siprc-prod	ECAM SPIRC project content database. This will house all the project EVMS's project data and the data tables that are synchronized from Laboratory's Cobra operational databases.	0
			$215.36

EVMS Azure ETL Process



This is an overview of the proposed ETL process for EVMS based on Azure technologies. The image above shows the process in which we intend to use. The flow of this process is as follows:

1.) The user will initiate a request from the SharePoint web app. This will send a message to the EVMS API to request to get the current user context. With this user context, the app will determine if the user is allowed to make the request, provided they have access.
2.) Once it is determined the user has access to make the request, another request to the EVMS API will be made requesting that a Data Factory pipeline will be executed.
3.) The Data Factory pipeline will get the requested data from the Cobra Database via the Self Hosted Integration Runtime and will process this data back to the EVMS Azure Database.

4.) While the Data Factory pipeline is executing, the SharePoint Web App will get messages on the status of the pipeline execution until it has finished or raised an error.
5.) Once all above steps are complete, The SharePoint Web application will have access to the new data by sending a request to the EVMS API.

Authentication:

SharePoint client web part to Azure Web API

When a user authenticates with SharePoint they will get an Access Token (JavaScript Web Token – JWT). That token will be passed on requests to the Azure Web API and evaluated. The access token has a collection of roles. ECAM will have a set of roles setup for each project. The set of roles will include project owners, project members and project visitors. Membership to these roles will assigned via Microsoft 365 groups. These groups can be managed using Microsoft Outlook online.

Azure Web API to Azure Data Factory

The Azure Web API identity will be an Azure system-assigned managed identity. Permission will be assigned to Azure Data Factory to allow the Azure Web API's system-assigned managed identity to trigger / execute the pipeline as well as retrieve real-time status of the pipeline instance execution log.

Azure Data Factory Self Hosted Integration Runtime to On-Premise Microsoft SQL Server

In Azure Data Factory you setup "Linked Services." ECAM will follow the current authentication practice being used at Laboratory with Azure Data Factory Self Hosted Integration Runtime. The runtime process is implemented as a Windows service so you can use its identity to connect with SQL Server or you can use SQL authentication with SQL user and password.

Azure Resources for EVMS ETL Process

Resource Type	Name	Description	Cost
Azure Data Factory		Would use the existing Laboratory DEV / QA Azure Data Factory	0
Azure Key Vault	kv-ecam-dev	Container to store information encrypted and secure access that information. We would store any app principal secrets and sql credentials. $0.03/10,000 transactions. This would only be utilized if SQL Authentication is used in the "Linked Service" with Azure Data Factory's Self Hosted Integration Runtime. In this case the SQL user's password would be stored in the Azure Key Vault.	0.03

Azure Key Vault	kv-ecam-qa	Container to store information encrypted and secure access that information. We would store any app principal secrets and sql credentials. $0.03/10,000 transactions This would only be utilized if SQL Authentication is used in the "Linked Service" with Azure Data Factory's Self Hosted Integration Runtime. In this case the SQL user's password would be stored in the Azure Key Vault.	0.03
Azure Data Factory		Would use the existing Laboratory Production Azure Data Factory	0
Azure Key Vault	kv-ecam-prod	Container to store information encrypted and secure access that information. We would store any app principal secrets and sql credentials. $0.03/10,000 transactions This would only be utilized if SQL Authentication is used in the "Linked Service" with Azure Data Factory's Self Hosted Integration Runtime. In this case the SQL user's password would be stored in the Azure Key Vault.	0.03

SQL Databases and Users

SQL Server	Database	Users
P6PRDSQLDB (LaboratoryMSSQL)	Cobra_MPEX	Laboratory\o9o
P6PRDSQLDB (LaboratoryMSSQL)	Cobra_SIPRC	Laboratory\o9o
P6PRDSQLDB (LaboratoryMSSQL)	Cobra_rpf	Laboratory\o9o
P6PRDSQLDB (LaboratoryMSSQL)	EVMS	Laboratory\o9o
DEVECAMSQL (LaboratoryMSSQL-DMZDEV)	EVMSApp_DEV	app-ecam-api-dev (Azure app service system-assigned managed identity)
DEVECAMSQL (LaboratoryMSSQL-DMZDEV)	EVMSProj_DEV	app-ecam-api-dev (Azure app service system-assigned managed identity)
DEVECAMSQL (LaboratoryMSSQL-DMZDEV)	EVMSProj_mpex_DEV	app-ecam-api-dev (Azure app service system-assigned managed identity)
DEVECAMSQL (LaboratoryMSSQL-DMZDEV)	EVMSProj_spirc_DEV	app-ecam-api-dev (Azure app service system-assigned managed identity)
DEVECAMSQL (LaboratoryMSSQL-DMZDEV)	EVMSProj_rpf_DEV	app-ecam-api-dev (Azure app service system-assigned managed identity)
DEVECAMSQL (LaboratoryMSSQL-DMZDEV)	EVMSApp_QA	app-ecam-api-qa (Azure app service system-assigned managed identity)

DEVECAMSQL (LaboratoryMSSQL-DMZDEV)	EVMSProj_QA	app-ecam-api-qa (Azure app service system-assigned managed identity)
DEVECAMSQL (LaboratoryMSSQL-DMZDEV)	EVMSProj_mpex_QA	app-ecam-api-qa (Azure app service system-assigned managed identity)
DEVECAMSQL (LaboratoryMSSQL-DMZDEV)	EVMSProj_spirc_QA	app-ecam-api-qa (Azure app service system-assigned managed identity)
DEVECAMSQL (LaboratoryMSSQL-DMZDEV)	EVMSProj_rpf_QA	app-ecam-api-qa (Azure app service system-assigned managed identity)
PRDECAMSQL (LaboratoryMSSQL-DMZPROD)	ECAMApp	app-ecam-api-qa (Azure app service system-assigned managed identity)
PRDECAMSQL (LaboratoryMSSQL-DMZPROD)	ECAMProj	app-ecam-api-qa (Azure app service system-assigned managed identity)
PRDECAMSQL (LaboratoryMSSQL-DMZPROD)	ECAMProj_MPEX	app-ecam-api-qa (Azure app service system-assigned managed identity)
PRDECAMSQL (LaboratoryMSSQL-DMZPROD)	ECAMProj_SPIRC	app-ecam-api-qa (Azure app service system-assigned managed identity)
PRDECAMSQL (LaboratoryMSSQL-DMZPROD)	ECAMProj_rpf	app-ecam-api-qa (Azure app service system-assigned managed identity)

Azure SQL Options:

The three products in the Azure SQL family are:

- **Azure SQL Database**: Support modern cloud applications on an intelligent, managed database service that includes serverless compute.
- **Azure SQL Managed Instance**: Modernize your existing SQL Server applications at scale with an intelligent fully managed instance as a service, with almost 100% feature parity with the SQL Server database engine. Best for most migrations to the cloud.
- **SQL Server on Azure VMs**: Lift-and-shift your SQL Server workloads with ease and maintain 100% SQL Server compatibility and operating system-level access.

Reference: https://learn.microsoft.com/en-us/azure/azure-sql/azure-sql-iaas-vs-paas-what-is-overview

Recommendation would be Azure SQL Database unless Laboratory already has an Azure SQL Managed Instance or SQL Server on Azure VM.

Azure SQL Database – purchasing models

- vCore purchasing model - Azure SQL Database | Microsoft Learn
- DTU-based purchasing model - Azure SQL Database | Microsoft Learn

Recommendation would be DTU-based purchasing model utilizing elastic pools in the Standard service tier:

Standard

eDTUs per pool	Included storage per pool	Max storage per pool [1]	Max number databases per pool	Max eDTUs per database [2]	Price for eDTUs and included storage [3]
50	50 GB	500 GB	100	50	$110.27/month
100	100 GB	750 GB	200	100	$220.53/month
200	200 GB	1 TB	500	200	$441.05/month

Azure SQL Database Point in Time Restore

With standard service tier, EVMS will be able to restore within 7 minutes of any point in time within the last 35 days.

BCDR option	Basic tier	Standard tier	Premium tier
Point In Time Restore	Any restore point within 7 days	Any restore point within 35 days	Any restore point within 35 days
Geo-Restore	ERT* < 12h RPO† < 1h	ERT* < 12h RPO† < 1h	ERT* < 12h RPO† < 1h
Standard Geo-Replication	Not included	ERT* < 30s RPO† < 5s	ERT* < 30s RPO† < 5s
Active Geo-Replication	Not included	Not included	ERT* < 30s RPO† < 5s

There are long-term retention for backups that can be configured for a small additional cost. Example:

Reference: Azure SQL Database Point in Time Restore | Azure Blog | Microsoft Azure

Once an architecture has been agreed to and approved then we can sit down with Laboratory's Microsoft licensing expert to discuss needs and best options. The above information is to give an estimated cost and should not be considered exact or final. Would gladly input projected Azure resources into an Azure pricing calculator.

Reference: Pricing Calculator | Microsoft Azure