APIM PoC Specification



Scope Document

Prepared for

Piraeus Bank

4/4/2022

Version 1 Draft

Prepared by

**[Type Author Here]**

[Type Author Position Here]

[Type Author Email Here]

Contributors

**[Type Contributors Here]**

Revision and Signoff Sheet

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
|  |  | 1 | Initial draft for review/discussion |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version Approved | Position | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1 API Management PoC Goals 5](#_Toc99390764)

[2 Infrastructure Description 6](#_Toc99390765)

[3 PoC Scenario descriptions 8](#_Toc99390766)

[3.1 Incoming API 8](#_Toc99390767)

[3.2 Outgoing API 8](#_Toc99390768)

[3.3 API transformation 8](#_Toc99390769)

[3.4 Significant event capture via event filtering 8](#_Toc99390770)

1. API Management PoC Goals

This PoC aims at demonstrating various features of the Azure API Management service and evaluating fitness for purpose for various scenarios at Piraeus Bank.

The scenarios have been identified through a series of workshops with Piraeus Bank, focused on how the bank currently uses API management features from other vendor systems.

The PoC will ensure that, going forward, Azure API Management will effectively replace currently used solutions, unify API management both on-premises and on Azure and enable efficient operations, monitoring and management of APIs throughout Piraeus Bank.

1. Infrastructure Description

The diagram to the right describes the infrastructure we will deploy on Azure and on-premises for the needs of the PoC.

An API Management (APIM) service instance will be deployed in **internal mode** within a virtual network (VNet). This means that no publicly accessible endpoint for the APIM instance exists (both for the gateway as well as for the built-in portal). Access to the outside world will be provided by a Web Application Firewall instance (WAF) that is connected to the VNet.

The WAF will provide path-based routing to the APIM service and any other needed services. As an example, we will include an App Service hosting a simple web application, that will play the role of the customer portal.

The APIM service will use an Event Hub instance as logging target, followed by a Stream Analytics job for deduplication of messages from the event hub. The Stream Analytics job will store events in an Azure SQL database as an example, but any other means of processing may be chosen later, e.g., an Azure Function or another app service followed by a database. The presence of the stream analytics job, apart from event deduplication, means that we can selectively filter events as needed and this will enable specific scenarios as described later.

The VNet on Azure will be connected to the on-premises network via VPN. Thus, we will use a virtual network gateway attached to the VNet, connected to the local gateway of the on-premises network. In case such a connection is already available and used in a different VNet, we will use VNet peering to connect the Azure VNet of the PoC to the VNet that hosts the gateway and will not need a new VPN connection.

The on-premises side will contain an API implementation for the PoC to be used as backend for the APIM service. We will also install a self-hosted gateway connected to APIM for specific scenarios in the PoC as described later in this document. The self-hosted gateway can either be deployed as a Docker container or on any available Kubernetes infrastructure.

1. PoC Scenario descriptions

The PoC shall consist of several parts, each demonstrating a different functionality needed. The parts are described here.

* 1. Incoming API

This part shall expose an API over the APIM service, with the backend implementation on-premises. Piraeus Bank has designated a specific API from the existing PSD2 implementation.

* 1. Outgoing API

This part is about using the self-hosted gateway to enable access to an external API for internal consumers, who may not have internet access. Piraeus Bank has designated an external API to be used for this purpose.

* 1. API transformation

This part is about using the APIM service REST-to-SOAP capabilities. APIM shall expose a synthetic REST API based on an on-premises SOAP API. Synthetic in this case means a REST API with several endpoints that route to the same SOAP implementation with different inputs and also different outputs translated from the internal call.

* 1. Significant event capture via event filtering

In this part we will demonstrate how event filtering can be used to detect a significant event as it happens. We will use a specific exposed API for this reason and record the number of times a specific user has completed a specific series of calls with the API. This can be used e.g. to monetize an API by collecting information about billable call sequences by the user (e.g. one call to authenticate followed by one operation call on a different endpoint).

* 1. API Statistics

In this part we will use the APIM REST API to extract API statistics and integrate with the event capture functionality of the previous part. We will aggregate the information on a page of the custom portal app service.

* 1. APIM integration with portal

This part demonstrates the path-based routing capabilities of the WAF that exposes the APIM service to the outside world. We will define a specific base URL through which both the APIM gateway and the custom portal hosted on an Azure App service will be accessible.

* 1. Push to production/API lifecycle/Dev-Prod

Configuration via Key/Value pairs, DevOps integration, config repos