REPUBLIC OF THE PHILIPPINES

DEPARTMENT OF BUDGET AND MANAGEMENT

GENERAL SOLANO STREET, SAN MIGUEL, MANILA

SUPPLEMENTAL/BID BULLETIN (SBB) NO. 1

This SBB No. 1 dated May 9, 2023 for **Project No. DBM-2023-21, "Subscription to Application Programming Interface (API) Management Platform,"** is issued pursuant to Section 22.5 of the 2016 Revised Implementing Rules and Regulations (IRR) of Republic Act (RA) No. 9184, to clarify, modify or amend items in the Bidding Documents. Accordingly, this shall form an integral part of the Bidding Documents.

<table>
<tr><th>PARTICULARS</th><th>AMENDMENTS</th></tr>
<tr><td>Section III. Bid Data Sheet

<table>
<tr><th>ITB Clause</th><th></th></tr>
<tr><td colspan="2">xxx</td></tr>
<tr><td>20</td><td>The bidder with the Lowest Calculated Bid shall submit ALL of the following post-qualification requirements:

xxx

4. As support to item 4.2 of Annex "A" (Detailed Technical Specifications), the following should be submitted:

xxx

ii. Gartner's report/specific certification showing Full Life Cycle API Management for at least three (3) consecutive years based on the latest Gartner Magic Quadrant report as of 2021 from Gartner's Leaders Quadrant;

xxx</td></tr>
</table></td><td>Section III. Bid Data Sheet

<table>
<tr><th>ITB Clause</th><th></th></tr>
<tr><td colspan="2">xxx</td></tr>
<tr><td>20</td><td>The bidder with the Lowest Calculated Bid shall submit ALL of the following post-qualification requirements:

xxx

4. As support to item 4.2 of Annex "A" (Detailed Technical Specifications), the following should be submitted:

xxx

ii. Gartner's report/specific certification showing Full Life Cycle API Management for at least three (3) consecutive years based on the latest Gartner Magic Quadrant report as of EITHER 2021 OR 2022 from Gartner's Leaders Quadrant;

xxx</td></tr>
</table></td></tr>
</table>

PARTICULARS	AMENDMENTS
Section VII. Technical Specifications Annex "A" (Detailed Technical Specifications) xxx 4.2.2 Proposed solution must be in the Gartner's Leaders Quadrant for Full Life Cycle API Management for at least three (3) consecutive years based on the latest Gartner Magic Quadrant report as of 2021; xxx	Section VII. Technical Specifications Annex "A" (Detailed Technical Specifications) **(REVISED)** xxx 4.2.2 Proposed solution must be in the Gartner's Leaders Quadrant for Full Life Cycle API Management for at least three (3) consecutive years based on the latest Gartner Magic Quadrant report as of **EITHER** 2021 **OR 2022**; xxx
Queries: 1. Can we have a list of sample internal and outside DBM systems that will use API? 2. May we request for the existing architecture, or even high-level architecture only? 3. Are there API Specification Documents and Technical Document Protocol? If yes, can we have copies of the same? 4. If a global company that has a partnership with Microsoft for more than twenty (20) years, can its Philippine office that is established in 2018 participate in the bidding, considering item 4.2.1 of the Detailed Technical Specifications of Section VII. Technical Specifications? 5. Can you consider changing the latest Gartner Magic Quadrant report reference to 2022?	**Clarifications:** 1. Please note that attached Annex "B" Sample List of Internal and Outside DBM systems that will use API is for bidding purposes only and the actual systems that will use API may increase or decrease depending on the availability of the systems. 2. Please note that attached Annex "C" High-Level Architecture is for bidding purposes only and the actual architecture may change as the need arises. 3. This document is not yet available since it is an output of another project being bidded out. 4. The Philippine office of the said global company cannot join since the requirement in item 4.2.1 of the Detailed Technical Specifications of Section VII. Technical Specifications requires that the technology provider of the proposed solution must be in the business for at least 20 years. 5. Please refer to the Revised Annex "A" (Detailed Technical Specifications).

	Note: **<u>Attached are the following documents which should be used as part of the Bidding Documents to be submitted by the bidders:</u>** 1. Annex "A" (Detailed Technical Specifications) (Revised); 2. Annex "B" Sample List of Internal and Outside DBM systems that will use API; and 3. Annex "C" High-Level Architecture

Other matters:

- The "No Contact Rule" shall be strictly observed. Bidders are not allowed to communicate with any member of the Bids and Awards Committee, Technical Working Group or Secretariat effective May 16, 2023 right after the opening of bids.

- For guidance and information of all concerned.

Digitally signed by
Magsino Clarito
Alejandro De Leon

CLARITO ALEJANDRO D. MAGSINO

Assistant Secretary
Chairperson, DBM-BAC

Annex A

Detailed Technical Specifications
(Revised)

1.0 Project Title

Subscription to Application Programming Interface (API) Management Platform

2.0 Objective

The Contractor should be able to provide industry-standard Application Programming Interface (API) Management Platform, whose technical and functional features conform to item 4.

3.0 Subscription Period

Twelve (12) months from the receipt of the Notice to Proceed, to be delivered to the DBM Central Office.

4.0 Specifications and Scope of Work

4.1. API Platform

4.1.1. **Overview**

4.1.1.1. API gateway is the entry point for all the external calls which call the microservices / APIs. The main concept of API gateway is to secure the microservices at the same time giving access to external world to make use of the underlying data or services. API gateway manages the incoming traffic and can also provide load balancing by diverting the traffic to relevant services.

4.1.1.2. Organizations are undergoing digital transformation to accelerate innovation, unlock new markets and drive new revenue that is revolutionizing every interaction, making business services and data available through APIs to various types of developers - internal, partner and third party. These APIs are then consumed by developers to build Mobile, Web, Internet of Things, Business Partner, and Internal LoB applications for end users to use. This supply demand relationship creates this Economy of API Providers and API Consumers.

4.2. General Requirements

4.2.1. Technology provider of the proposed solution must be in the business for at least twenty (20) years;

4.2.2. Proposed solution must be in the Gartner's Leaders Quadrant for Full Life Cycle API Management for at least three (3) consecutive years based on the latest Gartner Magic Quadrant report as of either 2021 or 2022;

4.2.3. Contractor must have partnership with the Principal/Technology provider for at least ten (10) years;
4.2.4. Contractor must have received at least ten (10) technology-related awards and/or recognitions
4.2.5. Proposed solution must be supported by at least one (1) The Open Group Architecture Framework (TOGAF) certified resource
4.2.6. Must conduct Proof of Concept;

4.3. Proposed Solution Components

4.3.1. Must be able to support the entire lifecycle of an API from idea, design, approval, build, deployment, and manage and govern;
4.3.2. Must be able to help create uniform, consistent, well-formed APIs, even if the underlying backend systems weren't built that way;
4.3.3. Must be able to add existing SOAP services;
4.3.4. Must be able to automate deployment of assets for the development lifecycle;
4.3.5. Must be able to reference existing assets such as encryption libraries, schema validation tools, data validation libraries, etc.;
4.3.6. Must be able to support threat detection by detecting fraudulent data injections at the API level;
4.3.7. Must be able to protect from traffic spikes;
4.3.8. Must be able to manage API consumption through quotas;
4.3.9. Must be able to allow quota setup both by developers, as well as by product managers post-development;
4.3.10. Must be able to have quotas synchronized across multi-region deployments;
4.3.11. Must be able to support publishing of SOAP, REST, JSON, and XML style services as APIs and as JMS;
4.3.12. Must be able to support API mashups;
4.3.13. Must be able to show any out of the box functions for doing traffic mediation, transformation, and security at the API Level;
4.3.14. Must include standard transformations for the following:
    4.3.14.1. XML to JSON;
    4.3.14.2. JSON to XML;
    4.3.14.3. SOAP to REST; and
    4.3.14.4. REST to SOAP
4.3.15. Must be able to support all the following variations of REST API definitions
    4.3.15.1. Single/dual verb (i.e., GET, POST only);
    4.3.15.2. Four verbs (POST, GET, PUT, DELETE);
    4.3.15.3. XML payloads;
    4.3.15.4. JSON payloads; and
    4.3.15.5. XHTML payloads
4.3.16. Must be able to support all the following advanced variations of REST API definitions:
    4.3.16.1. Additional verbs (e.g., PATCH, HEAD, OPTIONS, etc.);
    4.3.16.2. Custom media types; and
    4.3.16.3. Compressed payloads (e.g., zip)

4.3.17. Must be able to support HTTP & HTTPS;
4.3.18. Must have built-in debugging tools;
4.3.19. Built-in debugging tool must be able to show "before" and "after" of each policy during replay;
4.3.20. Must be able to support API versioning;
4.3.21. Must be able to store all policies and system configuration in standard-based XML with well published schemas for easy migration/promotion;
4.3.22. Proxy must be able to support caching;
4.3.23. Proxy must be able to support rate limiting, quotas, and spike arrests;
4.3.24. Must dynamically change behavior based upon factors such as user credentials, location, device type, etc.;
4.3.25. Proxy must be able to support dynamic routing (orchestration – or intelligent routing to a second system based upon the response from a first system);
4.3.26. Must be able to show the out-of-the-box backend service APIs for common application functionality such as user management, data storage and synchronization, messaging, and locations;
4.3.27. Must be able to support identity integration with popular social networks and internet services;
4.3.28. Must be able to store binary objects such as files and images;
4.3.29. The service provider should be able to provide an example of large-scale deployments using this platform;
4.3.30. Must be able to support extensions using common languages like Java, Python, or JavaScript;
4.3.31. Must have wizards to generate APIs from Swagger, SOAP services, and other backend services;
4.3.32. Must be able to show the standard governance features available in the platform;
4.3.33. Must be able to support API Lifecycle governance;
4.3.34. Must be able to publish APIs for external and internal consumers;
4.3.35. Must be able to manage API visibility and restrict access to consumers;
4.3.36. Must be able to keep a separation between the API development, API management, and Policy Administration;
4.3.37. Must have a configuration that allows exporting from an environment and importing to another environment, such as UAT configuration to Production environment;
4.3.38. Must be able to provide backward support to partners by maintaining obsolete API versions. This should enable partners who are not ready for the new API version yet can still continue their services;
4.3.39. Must be able to provide effective mechanism to retry obsolete API with manageable impact to existing partners;
4.3.40. Must be able to support source code control mechanism;
4.3.41. Must allow source codes to be checked in and checked out from external source code repository in the backend system;
4.3.42. Must be able to provide a tool to facilitate an efficient developer and partner on-boarding;

4.3.43. Must be able to automatically generate API documentation, which must be updated when microservices are created, modified, and/or retired;
4.3.44. Must be able to provide approval workflow when partners subscribe to API services in order to process the partners' subscription requests;
4.3.45. Must be able to provide role-based authorization on the workflow;
4.3.46. Must be able to provide rule-based routing to deliver subscription request to the associated approvers;
4.3.47. Must be able to provide out-of-the-box notification to API subscribers, such as email, SMS and etc., in order for further on boarding process;
4.3.48. Proposed solution licensing must be based on per API call hybrid entitlement;
4.3.49. Contractor must propose and support for at least three hundred million (300M) API Calls per year to be shared for all environments;
4.3.50. Proposed solution must have perpetual licensing scheme with annual maintenance;
4.3.51. Proposed solution must be deployed via containers;
4.3.52. Proposed solution must be deployed on-premises;
4.3.53. Proposed solution must have built-in high availability (HA) feature for production environment;
4.3.54. Proposed solution must provide data hosting services;

4.4. API Portal

4.4.1. Must be able to show how assets are manifested in the developer portal for developer use;
4.4.2. Must be able to facilitate on-boarding;
4.4.3. Must be able to be deployed either on-premises or on-cloud;
4.4.4. Must be able to provide interactive documentation to allow API consumers to easily try out published APIs;
4.4.5. Must allow each developer (or team) to get their own personalized metrics;
4.4.6. Must allow customizable registration form;
4.4.7. Must allow customers to customize, skin, and modify the portal without vendor involvement;
4.4.8. Must leverage on standard CMS technologies to ensure easy-to-find skill sets and pre-existing modules;
4.4.9. Must provide the ability to revoke or suspend developer keys;
4.4.10. Must be able to support a B2B2D type model which allows enterprises to let their partners manage their own pool of developers and their access to the enterprise’s APIs;
4.4.11. Must provide a user interface to try out or test an API or SOAP service within the Portal and the API Admin;
4.4.12. Must allow developers to share information and documentation as well as discussion forums pertaining to APIs in the tool;
4.4.13. Must be able to provide documentation on-demand on how to use the APIs;
4.4.14. Must be able to sunset and retire APIs automatically without impacting production;

4.4.15. Must allow restricting of access to the catalog so that developers can only see information about the API's they are granted access to;
4.4.16. Must be able to show how risk of malicious attacks from the outside accessing internal APIs are avoided;
4.4.17. Must be able to provide any self-serving ability for application developers to register access to APIs;
4.4.18. Must be able to support skinning of all externally visible aspects to comply with CUSTOMER branding;
4.4.19. Must be able to show the level of customization that can be done in the portal without the need for vendor involvement.
4.4.20. Must be capable to transform the interface specification (WSDL) into asset library (e.g., importing WSDL files);
4.4.21. Must allow categorization and managing of groups for partners, as API consumer. For example, an API consumer group for division A managed to use individual policy with is separated from division B;
4.4.22. Must be capable to enforce usage quota based on partner subscription policy;
4.4.23. Must be able to support multiple teams to manage their quotas independently;
4.4.24. Must be able to provide flexible way to manage and control API usage by partners once usage quota has been reached;
4.4.25. Must be able to notify subscribers when pre-defined quota thresholds have been reached;

4.5. Transformation and Orchestration

4.5.1. Must be able to support the following message formats
4.5.1.1. XML; and
4.5.1.2. JSON
4.5.2. Must be able to support the message formats above natively as first-class citizens. Please explain if the platform does not support that, such as the need to transform these message formats to XML in order to manipulate the messages;
4.5.3. Must be able to provide a graphical mapping tool to map and transform request and response between the message formats described above;
4.5.4. Must be able to support JSON transformation standards;
4.5.5. Must be able to support XML transformation standards such as XSLT and Xquery;
4.5.6. Must be able to support the use of Enterprise Integration Patterns in the development of APIs;
4.5.7. Must be able to show how business users (end users) can be allowed access to e.g., some certain self-service portal or catalogue where they via a simple user interface can on-board e.g., new EDI partners or do simple predefined integration scenario/flow customizing.

4.6. API Security

4.6.1. Must be able to support single sign-on (SSO) across all the roles involved in the lifecycle;

4.6.2. Must be able to be supported by the standard industry security certifications available for the product;
4.6.3. Must use open standards to delegate authentication capabilities to tenants;
4.6.4. Must support API security mechanisms such as tokenization, encryption, and policy systems;
4.6.5. Must be able to show the security / policy enforcement options when some assets might require additional security in a cloud/on-premises infrastructure;
4.6.6. Must be able to support OAuth;
4.6.7. Enumerate the versions of OAuth supported by the product;
4.6.8. Must be able to support both secure channels and secure payloads;
4.6.9. Proxy must be able to provide support for CORS;
4.6.10. Proxy must be able to protect against XML or JSON attacks;
4.6.11. Must be able to handle role-based access controls to ensure different members of the API team can perform their roles effectively without affecting other teams;
4.6.12. Must be able to support custom/proprietary implementations;
4.6.13. Must be able to secure APIs at the operation level (Ex: can do GET, but not POST or PUT);
4.6.14. Must be able to support FIPS 140;
4.6.15. Must be able to work with malware/antivirus software;
4.6.16. Must support the development and application of custom security policies;
4.6.17. Security features must be available as self-service via configuration (not via coding);
4.6.18. Must be able to prevent PII information from being viewed during monitoring, logging, and audit;
4.6.19. Must have auditing of administrator activity;
4.6.20. Must be secured by design. The internet facing component must be deployed in DMZ and able to integrate with backend systems/components in secured network zone;
4.6.21. According to security policies, the API Gateway component must support backend integration over SSL;
4.6.22. Must be able to provide backward compatibility with older version of SSL in case of integration with legacy system;
4.6.23. OAuth is one of the most widely used forms of authentication for consumer or partner facing apps. The proposed solution must be able to provide out-of-the-box OAuth Authentication capability;
4.6.24. Must be able to provide capability to integrate with Active Directory, Secured LDAP and other authentication system in bank backend systems;
4.6.25. The API Gateway must not use Java or other open-source libraries that can introduce vulnerabilities into the system.

4.7. Analytics

4.7.1. Must be able to show the out-of-the-box reports provided by the tool;
4.7.2. UI must allow drill down on each chart;

4.7.3. Analytics data, once collected, must provide an API for easy access and export;
4.7.4. Must allow the system to provide business-level visibility.;
4.7.5. Must be able to create reports on-demand;
4.7.6. Must be able to provide service performance monitoring, reporting, and analysis;
4.7.7. If payload data is captured, must be able to use this data for reporting;
4.7.8. Must be able to show what the exception management reporting capabilities are;
4.7.9. Must be able to provide end-to-end visibility and trending performance statistics;
4.7.10. Must be able to provide performance management data with counters per application type and per API message type;
4.7.11. Must have available reporting for the developer (ex. call latency, SLA compliance, other metrics);
4.7.12. Must collect analytics asynchronously so as not to impede runtime traffic;
4.7.13. Must be able to show the level of operational visibility the solution can provide based on API traffic flowing through the system.

4.8. Scope of Work

4.8.1. Must be able to complete the setup, installation, and configuration for all environments within three (3) months upon project kick-off;
4.8.2. Must be able to setup, install, and configure for non-production (Dev/QA) and production environment;
4.8.3. Must be able to configure the API Management Portal platform in non-production within one (1) month upon project kick-off and for production within one and a half (1 ½) months upon project kick-off;
4.8.4. Must be able to configure and customize ten (10) APIs in non-production within four (4) months upon project kick-off and deployed to productions within five (5) months upon project kick-off;
4.8.5. Must be able to deliver Custom Developer Portal within one and a half (1 ½) months upon project kick-off for non-production and two and a half (2 ½) months upon project kick-off for production;
4.8.6. Must provide recommended hardware and software requirements (storage, virtual CPU, virtual RAM) for all system environments;
4.8.7. Development service delivery shall be incorporating the DevOps/ DevSecOps approach and the Agile methodology;
4.8.8. Must be able to conduct Sprint Planning - At the start of the application development project, the team develops user stories - application functionalities as told from the perspective of a user. The list of features based on these stories will serve as the team's product backlog - simply put, the list of things that need to be done;
4.8.9. Must be able to perform Daily Scrum - A non-traditional approach to application development also means a change in the way project meetings are held. The team meets for about 15 minutes to create a

game plan for the day. In these short and meaningful meetings, the team only intends to remove impediments in achieving their sprint goal;

4.8.10. Must conduct Sprint Review/Retrospection - At the end of every sprint, which typically goes for 2 weeks, the team reviews with its stakeholders what went well, what went wrong, and what can be improved for the next sprint;

4.8.11. Must work using Continuous Integration - While multiple developers work on different features, code is routinely integrated into a main repository, where it is tested and integrated up to multiple times in a day;

4.8.12. Must implement Continuous Testing - Through automated testing, release candidates are tested early and often. The goal is to determine the impact of the changes in the code as fast as possible and whether the changes can be safely deployed into production;

4.8.13. Must have experience in Continuous Delivery - When the code is able to go through integration and testing, it is then automatically deployed into a staging environment. The team makes the decision to deploy the changes into production. Thus, a minimum viable product (MVP) is made available;

4.8.14. For customized codes outside the core application, the Contractor must use the latest version of NodeJS platform;

4.8.15. Delivery of all necessary customized and other software/s, materials, licenses, and other components (other than those to be provided by DBM and subject to DBM approval) required to operate and maintain the solutions internally is included as part of the Project's deliverables;

4.8.16. All modules developed by the Contractor for the Project should be turned-over to DBM, including all components necessary to run and support the solution (i.e. source code, configuration file), thru effective knowledge transfer mechanisms such as training and proper turn-over of system documentation;

4.8.17. Contractor should provide a structured training for the API Management solution for at least three (3) days or twenty-four (24) hours. Training should be a combination of lecture and laboratory exercises tackling solution introduction, API Design (REST and SOAP) and Definitions, Policies, Analytics, Error Handling, Security, and external-facing Web Portal;

4.8.18. The Contractor must have an existing central support using helpdesk system to accommodate technical request. The helpdesk system will provide ticket for each level 1 technical request or issues and will provide status and report until it will resolve. The Helpdesk System of the bidder/contractor must be available 24x7 and should be available for site visit of customer's representative/s for post-qualification evaluation.

5.0 Warranties of the Contractor

5.1 For the subscription of the licenses and support, the warranties shall include the following:

5.1.1 The contractor warrants that it shall strictly conform to the terms and conditions of this Detailed Technical Specifications;

5.1.2 The contractor warrants that the technical staff assigned are qualified to provide the deliverables required to the satisfaction of the DBM;

5.1.3 The contractor shall secure, maintain at its own expense all registration, licenses, or permits required by national or local laws and shall comply with the rules, regulations, and directives of regulatory authorities and Commissions;

5.1.4 The contractor's technical staff assigned to support DBM shall take all necessary precautions for the safety of all persons and properties at or near their area of work and shall comply with all the standard and established safety regulations, rules and practices;

5.1.5 The contractor's technical staff assigned to support DBM shall coordinate with the ICTSS in the implementation of this project;

5.1.6 The contractor shall be liable for loss, damage, or injury caused directly or indirectly through the fault or negligence of its technical staff assigned. It shall assume full responsibility therefore and the DBM shall be fully released from any liability arising therefrom;

5.1.7 The contractor shall neither assign, transfer, pledge, nor subcontract any part of or interest on the contract being bidded out; and

5.1.8 The contractor shall identify the certified technical staff who will be given authority to access and operate the specified equipment. The DBM, through the ICTSS, shall be informed within five (5) calendar days, through a formal notice, of any change or replacement of technical staff assigned.

6.0 Confidentiality of Data

6.1 All technical staff assigned by the contractor shall be required to sign a Non-Disclosure Agreement (NDA).

6.2 The DBM Enterprise Network System, its component, parts and all products, products samples and specifications, data, ideas, technology, and technical/nontechnical materials, all or any which may be derived from any of the foregoing are strictly confidential.

6.3 The contractor agrees to hold all the foregoing information in strict confidence. The contractor further agrees not to reproduce or disclose any confidential information to third parties without the prior written approval of the DBM.

7.0 Terms of Payment

7.1 One-time payment shall be made, subject to the submission of the following documentary requirements, and in accordance with budgeting, accounting, and auditing laws, rules, and regulations:

7.1.1 Proof of API Platform Subscription;

7.1.2 Sales Invoice/Billing Statement;

7.1.3 Certificate of Acceptance issued by the Undersecretary for Information and Communications Technology (ICT) Group; and

7.1.4 NDA.

Annex B

Sample List of Internal and Outside DBM systems that will use API

Internal and Outside DBM systems	Process Owner
National Government Collection & Modified Disbursement System (NGCDS)	Bureau of Treasury (BTr)
Electronic New Government Accounting System (eNGAS)	Commission on Audit (COA)
Online Submission of Budget Proposal (OSBP) Budget Preparation Management System (BPMS) Unified Reporting System (URS)	Department of Budget and Management (DBM)
Modernized Philippine Government Electronic Procurement System (mPhilGEPS)	Philippine Government Electronic Procurement System (PhilGEPS)

Annex C

High Level Architecture