

**PROPOSAL:**

**MOCD – API HUB**

**Name 1**

**Submitted by**

Mohi Karim

Verbanet Technologies LLC  
12.03.19

**DIRECTORY.**

[1 KEY DETAILS 8](#_Toc5720559)

[2 STRATERGIC OBJECTIVES 9](#_Toc5720560)

[3 PROJECT SCOPE 10](#_Toc5720561)

[3.1 Inventory Management 10](#_Toc5720562)

[3.1.1 Prototype Screens 11](#_Toc5720563)

[3.2 Electronic Stock Card 12](#_Toc5720564)

[3.2.1 Prototype Screens 13](#_Toc5720565)

[3.3 Stock Movements 16](#_Toc5720566)

[3.3.1 Prototype Screens 17](#_Toc5720567)

[3.4 Inventory Management 20](#_Toc5720568)

[3.4.1 Prototype Screens 20](#_Toc5720569)

[3.5 Dashboard 23](#_Toc5720570)

[3.5.1 Prototype Screens 24](#_Toc5720571)

[3.6 Flexible Location Hierarchy 28](#_Toc5720572)

[3.6.1 Prototype Screens 28](#_Toc5720573)

[3.7 Reporting 30](#_Toc5720574)

[3.7.1 Prototype Screens 30](#_Toc5720575)

[3.8 Security 34](#_Toc5720576)

[3.8.1 Prototype Screens 35](#_Toc5720577)

[4 DEFENITION of TERMS 39](#_Toc5720578)

[4.1 USER CHARACTERISTICS 41](#_Toc5720579)

[4.1.1 Generic user 41](#_Toc5720580)

[4.1.2 Faculty/staff user 41](#_Toc5720581)

[4.1.3 Admin user 41](#_Toc5720582)

[4.1.4 Workflow User: Can also be in one of the three previous categories 41](#_Toc5720583)

[4.1.5 Sample Create Course Workflow, Roles, Permissions 41](#_Toc5720584)

[4.2 PROPOSED SOLUTION MODEL 44](#_Toc5720585)

[4.3 FUNCTIONAL REQUIREMENTS 45](#_Toc5720586)

[4.4 COURSE MANAGEMENT 46](#_Toc5720587)

[4.4.1 Course Information 46](#_Toc5720588)

[4.4.2 Course Processes 48](#_Toc5720589)

[4.5 PROGRAM MANAGEMENT 49](#_Toc5720590)

[4.5.1 Program Information 49](#_Toc5720591)

[4.5.2 Program Processes 50](#_Toc5720592)

[4.6 COURSE SET MANAGEMENT 51](#_Toc5720593)

[4.7 LEARNING OBJECTIVES MANAGEMENT 51](#_Toc5720594)

[4.8 ORAGANIZATION, PEOPLE and ROLES MANANAGEMENT 52](#_Toc5720595)

[4.9 WORKFLOW STATES 54](#_Toc5720596)

[4.10 PROPOSAL STATES 54](#_Toc5720597)

[4.11 COURSE STATES 55](#_Toc5720598)

[4.12 COURSES and PROPOSAL WORKFLOW with STATES 57](#_Toc5720599)

[5 APPLICATION SERVICES 58](#_Toc5720600)

[5.1 Academic Time Period Service 58](#_Toc5720601)

[5.1.1 Description 58](#_Toc5720602)

[5.1.2 Key Concepts 58](#_Toc5720603)

[5.2 Canonical Learning Unit (CLU) Service 59](#_Toc5720604)

[5.2.1 Description 59](#_Toc5720605)

[5.2.2 Key Concepts 59](#_Toc5720606)

[5.3 Comment Service 62](#_Toc5720607)

[5.3.1 Description 62](#_Toc5720608)

[5.3.2 Key Concepts 62](#_Toc5720609)

[5.4 Course Service 63](#_Toc5720610)

[5.4.1 Description 63](#_Toc5720611)

[5.4.2 Key Concepts 63](#_Toc5720612)

[5.5 Document Service 64](#_Toc5720613)

[5.5.1 Description 64](#_Toc5720614)

[5.5.2 Key Concepts 64](#_Toc5720615)

[5.6 Enumeration Management Service 65](#_Toc5720616)

[5.6.1 Description 65](#_Toc5720617)

[5.6.2 Key Concepts 65](#_Toc5720618)

[5.7 Learning Objective Service 65](#_Toc5720619)

[5.7.1 Description 65](#_Toc5720620)

[5.7.2 Key Concepts 65](#_Toc5720621)

[5.8 Learning Result Catalog Service 66](#_Toc5720622)

[5.8.1 Description 66](#_Toc5720623)

[5.8.2 Key Concepts 67](#_Toc5720624)

[5.9 Message Service 68](#_Toc5720625)

[5.9.1 Description 68](#_Toc5720626)

[5.9.2 Key Concepts 68](#_Toc5720627)

[5.10 Organization Service 68](#_Toc5720628)

[5.10.1 Description 68](#_Toc5720629)

[5.10.2 Key Concepts 68](#_Toc5720630)

[5.11 Program Service 69](#_Toc5720631)

[5.11.1 Description 69](#_Toc5720632)

[5.11.2 Key Concepts 69](#_Toc5720633)

[5.12 Proposal Service 70](#_Toc5720634)

[5.12.1 Description 70](#_Toc5720635)

[5.12.2 Key Concepts 71](#_Toc5720636)

[5.13 Search Service 72](#_Toc5720637)

[5.13.1 Description 72](#_Toc5720638)

[5.13.2 Key Concepts 72](#_Toc5720639)

[5.14 State Service 72](#_Toc5720640)

[5.14.1 Description 72](#_Toc5720641)

[5.14.2 Key Concepts 73](#_Toc5720642)

[5.15 Version Management Service 73](#_Toc5720643)

[5.15.1 Description 73](#_Toc5720644)

[5.15.2 Key Concepts 74](#_Toc5720645)

[6 DETAILED REQUIRMENT SPECIFICATIONS 75](#_Toc5720646)

[7 APPLICATION ARCHITECHURE 82](#_Toc5720647)

[7.1 COURSE CREATION WORKFLOW 82](#_Toc5720648)

[7.2 ENTITY RELATIONSHIP DIAGRAMS 83](#_Toc5720649)

[7.2.1 ATP Service 83](#_Toc5720650)

[7.2.2 Canonical Learning Unit 84](#_Toc5720651)

[7.2.3 Comment Service 85](#_Toc5720652)

[7.2.4 Course Service 86](#_Toc5720653)

[7.2.5 Document Service 87](#_Toc5720654)

[7.2.6 Enumeration Management Service 88](#_Toc5720655)

[7.2.7 Learning Objective Service 89](#_Toc5720656)

[7.2.8 Learning Result Catalog Service 90](#_Toc5720657)

[7.2.9 Message Service 91](#_Toc5720658)

[7.2.10 Organization Service 92](#_Toc5720659)

[7.2.11 Program Service 93](#_Toc5720660)

[7.2.12 Proposal Service 94](#_Toc5720661)

[7.2.13 State Service 95](#_Toc5720662)

[7.2.14 Version Management Service 95](#_Toc5720663)

[7.3 LOCAL SERVER INSTANCE (On Premise) 96](#_Toc5720664)

[7.4 Sample Screens 97](#_Toc5720665)

[7.4.1 Course List 97](#_Toc5720666)

[7.4.2 Course View 98](#_Toc5720667)

[7.4.3 Form Changes (Tracking) 99](#_Toc5720668)

[7.4.4 Rich Text Content 100](#_Toc5720669)

[7.4.5 Tagging Courses 101](#_Toc5720670)

[7.5 NON-FUNCTIONAL REQUIREMENTS (OTHERS) 102](#_Toc5720671)

[7.6 TECHNICAL CONFIGURATIONS 102](#_Toc5720672)

[7.6.1 DEVELOPMENT ENVIRONMENT 102](#_Toc5720673)

[7.6.2 RECOMMENDED WED HOSTING PACKAGE - DEDICATED 103](#_Toc5720674)

[7.6.3 BROWSER 103](#_Toc5720675)

[7.6.4 HARDWARE DEVICES 103](#_Toc5720676)

[7.6.5 TECHNICAL STANDARDS 104](#_Toc5720677)

[7.6.6 TECHNICAL GUIDELINES 104](#_Toc5720678)

[7.7 PROJECT DELIVERY 106](#_Toc5720679)

[7.7.1 Agile Methodology 106](#_Toc5720680)

[7.7.2 PROJECT MANAGEMENT 108](#_Toc5720681)

[7.7.3 ROLES & RESPONSIBILITIES 109](#_Toc5720682)

[7.7.4 DELIVERY ACTIVITY SUMMARY 110](#_Toc5720683)

[7.7.5 PROJECT IMPLEMENTATION PLAN 111](#_Toc5720684)

[7.7.6 DELIVERABLES 111](#_Toc5720685)

[7.7.7 ESTIMATED DELIVERY TIME 112](#_Toc5720686)

[7.7.8 DEPLOYMENT DETAILS 114](#_Toc5720687)

[7.7.9 RELEASE PLANNING 114](#_Toc5720688)

[7.7.10 RISK CONTINGENCY PLANNING 114](#_Toc5720689)

[7.8 PROJECT ASSUMPTIONS 116](#_Toc5720690)

[7.8.1 OBJECTIVES 116](#_Toc5720691)

[7.8.2 DESIGN 116](#_Toc5720692)

[7.8.3 DEVELOPMENT 116](#_Toc5720693)

[8 OUT OF SCOPE 117](#_Toc5720694)

[9 CHANGE MANAGEMENT 119](#_Toc5720695)

[9.1 MAINTENANCE & SUPPORT 120](#_Toc5720696)

[10 TERMS AND CONDITIONS 121](#_Toc5720697)

[10.1 ACCEPTANCE CRITERIA 121](#_Toc5720698)

[10.2 WARRANTY 121](#_Toc5720699)

[10.3 SOURCE CODE & INTELLECTUAL PROPERTY RIGHTS 122](#_Toc5720700)

[10.4 GENERAL TERMS AND CONDITIONS 122](#_Toc5720701)

[10.5 GENERAL ADMINISTRATIVE, TECHNICAL & FUNCTIONAL ASSUMPTIONS 124](#_Toc5720702)

[11 FINANCIALS 125](#_Toc5720703)

[11.1 Curriculum Management Application 125](#_Toc5720704)

[11.2 PAYMENT TERMS 125](#_Toc5720705)

[11.2.1 MODE OF PAYMENT 126](#_Toc5720706)

# KEY DETAILS

**PROJECT NAME CLIENT**

API HUB MOCD

**CLIENT CONTACT PROPOSAL ID NO.**

Name 1 AD/BP/12032019/2609/1

m.aldarabee@gmail.com

+966 59 011 1298

**PROPOSAL SUBMISSION ANTICIPATED START DATE**

12.03.19  TBD

**PROPOSAL VALID UNTIL PROPOSED TECHNOLOGY**

11.04.19 JAVA, SPRING

CSS3, HTML5

Responsive Web Application

**PROPOSAL SUBMITTED BY APPLICATION TYPE**

Mohi Karim Web & Mobile Application

Verbanet Technologies L.L.C

Dubai, U.A.E

+971 4 297 3236

Mohi.karim@verbat.com

# STRATERGIC OBJECTIVES

Ministry Of Community Development (Herein referred to as “MOCD” or the “Customer”) works on policies, strategies and programs for the development and advancement on social welfare of the community in the UAE. MOCD comprises of different departments, which works on development and improvement for community welfare.

MOCD is soliciting proposals from various candidate organizations to evaluate their API Management services that focus on delivering tools and applications.

Vendor is required to propose us with an on premise solution that shall support inbound, outbound service integrations like SOAP, REST API’s etc. The solution shall support API integration services and continuous support. The API Hub solution should help us in publishing, promoting and governing APIs, in a secure and scalable environment. The main objectives of the platform are

* Design API Hub Architecture
* Enhance IT User Experience and align with MOCD branding
* Enable social collaboration through API Gateways
* Cost savings
* Hassle Free maintenance
* Implementation of Industry Best Standard practices
* Simple Approval workflow management for Sharing API’s
* Open API Support (via Portal)
* Optimizing Inbound & Outbound API’s
* Analytics & Troubleshooting

# PROJECT SCOPE

Verbanet Technologies L.L.C., (hereafter referred under its trade / brand name as” Verbat”) in partnership with the Ministry of Community Development (MOCD) shall develop a API Management System (API HUB). Currently SOAP and REST API’s are hosted on different systems (windows, Linux servers) with multiple application owners. These services are disparate and consumed by various Government agencies with more services in the pipeline that needs to be integrated

Verbat shall propose a solution that integrates these disparate services into a single API platform so that they can be managed cohesively.

Verbat shall provide / perform the following support services for MOCD

* Support MOCD in installation, implementation, launch and rollout of API Management Hub.
* Provide post-implementation support for 3 months
* Provide documentation with requirement gathering, design, and implementation and rollout phases wherever it is applicable.
* Take responsibility for providing training to IT/End users with necessary documentation
* Handover the system and deliverables to MOCD IT team
* Plan and provide services from Dubai or partly from Dubai (as proposed as per the solution), with the option to propose local services as appropriate.

Verbat has looked at various platforms to determine the viability of products that could be deployed at the client. The platforms that were being compared are

**Apigee**: A google cloud subsidiary. This is a paid product. Prices range based on team size and deployment environments, No. of API calls used etc. For small teams with 2 environments, the cost is $500/month. Apigee offers a hybrid plan where *part of its solution can be hosted in premises*. However the application can be managed only from the google cloud. The implementation can be hosted locally

**Swagger**: Swagger is the leader in API management and tools for building API’s. They have both free and paid solutions. Free tools include swagger editor, UI & code gen. Swagger tools are open source. Its API management tools start at $30/month for 2 users.

**Kong**: Kong is a micro services based API gateway. Kong runs in front of any RESTful API and is extended through plugins, which provide extra functionality and services beyond the core platform. Like the others mentioned above it has a free version available on GitHub, however the full product is a paid service. Prices are determined by various factors such as environments, users, security etc.

Other products considered include Moesif, TYK and Gravitee. Apigee seems to lead the pack with a much more functional and robust offering that match the requirements stated by the client.

## A Brief Introduction to the Apigee Edge Platform

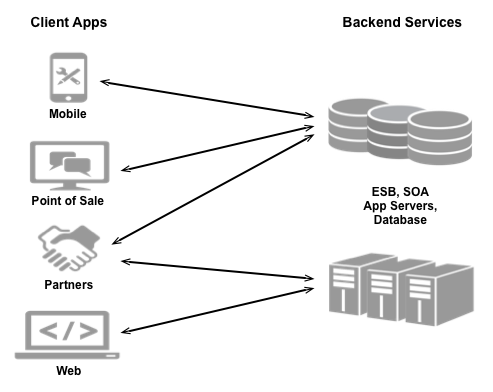
Apigee Edge is a platform for developing and managing API proxies. An API proxy is an interface that can be used by backend services. Rather than having them consume those services directly, they access an Edge API proxy that is created. With a proxy, one can provide value-added features such as:

* Security
* Rate limiting
* Quotas
* Caching & persistence
* Analytics
* Transformations
* CORS
* Fault handling
* And so much more...

Companies often expose services as a set of HTTP endpoints. Client app developers then make HTTP requests to these endpoints. Depending on the endpoint, the service might then return data, formatted as XML or JSON, back to the client app.

The client apps that consume these services can be implemented as standalone apps for a mobile device or tablet, as HTML5 apps running in a browser, or as any other type of app that can make a request to an HTTP endpoint and consume any response data. These apps might be developed and released by the same company that exposed the services, or by third-party app developers who make use of publicly available services.

The following image shows this type of model:



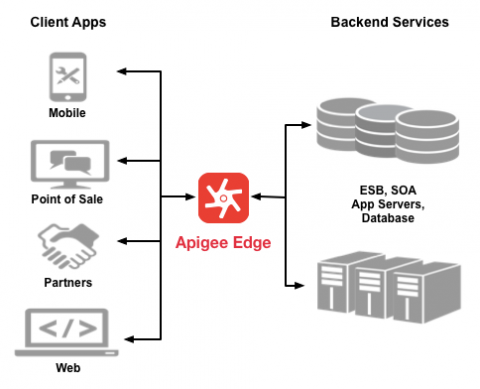
Because providers make their services available over the web, they must ensure that they have taken all necessary steps to secure and protect their services from unauthorized access. As a service provider, consider:

* **Security:** How will you control access to your services to prevent unauthorized access?
* **Compatibility:** Will your services work across different platforms and devices?
* **Measurability:** How can you monitor your services to make sure they are available?
* **Monetization:** How can you track and bill customers for access to your services?

Apigee Edge, which is built on Java, enables you to provide secure access to your services with a well-defined API that is consistent across all of your services, regardless of service implementation. A consistent API:

* Makes it easy for app developers to consume your services.
* Enables you to change the backend service implementation without affecting the public API.
* Enables you to take advantage of the analytics, monetization, developer portal, and other features built into Edge.

The following image shows an architecture with Edge handling the requests from client apps to your backend services:



Rather than having app developers consume your services directly, they access an API proxy created on Edge. The API proxy functions as a mapping of a publicly available HTTP endpoint to your backend service. By creating an API proxy you let Edge handle the security and authorization tasks required to protect your services, as well as to analyze, monitor, and monetize those services.

Because app developers make HTTP requests to an API proxy, rather than directly to your services, developers do not need to know anything about the implementation of your services. All the developer needs to know is:

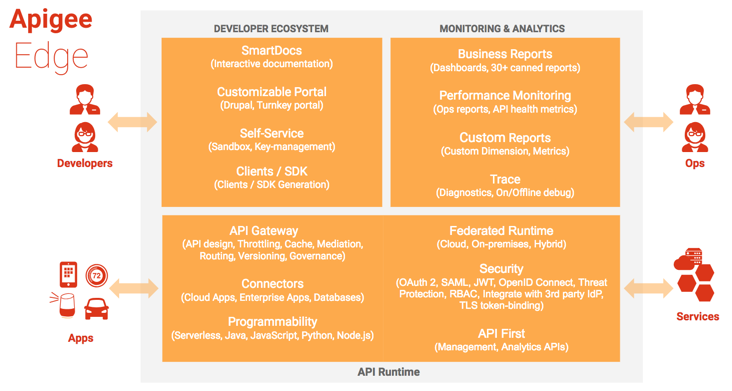
* The URL of the API proxy endpoint.
* Any query parameters, headers, or body parameters passed in a request.
* Any required authentication and authorization credentials.
* The format of the response, including the response data format, such as XML or JSON.

The API proxy isolates the app developer from your backend service. Therefore you are free to change the service implementation as long as the public API remains consistent. By maintaining a consistent frontend API, existing client apps will continue to work regardless of changes on the backend. You can even use policies on the API proxy to add functionality to a service without having to make any changes to the backend service.

API proxies can be bundled together to create an API product which can have predefined service plans attached to it. That service plan can set access limits on API proxies, provide security, allow monitoring and analytics, and provide additional features. API products are also the central mechanism that Edge uses for authorization and access control to your APIs.

Apigee Edge consists of API runtime, monitoring and analytics, and developer services that together provide a comprehensive infrastructure for API creation, security, management, and operations.

The following figure shows Edge services:



## Basic terminology

There are several concepts that are common ideas with a unique meaning in Apigee.

|  |  |
| --- | --- |
| Term | Definition |
| API | An 'application programming interface'—an interface that makes it easy for one application to 'consume' capabilities or data from another application.  By defining stable, simplified entry points to application logic and data, APIs enable developers to easily access and reuse application logic built by other developers. In the case of 'Web APIs', that logic and data is exposed over the network. Since applications that consume APIs are sensitive to changes, APIs also imply a 'contract'. The contract provides some level of assurance that, over time, the API will change in a predictable manner. |
| API proxy | A facade on Edge for one or more APIs, generic HTTP services, or applications (such as Node.js).  An API proxy is implemented as a set of configuration files, policies, and code that rely on a set of resources provided by Apigee Edge. API proxies can be generated and configured using the Apigee Edge management UI, or they can be implemented locally in a text editor or IDE.  The facade provided by an API proxy decouples the developer-facing API from 'backend' services, shielding developers from code changes and enabling innovation at the edge without impacting your internal development teams. As development teams make backend changes, developers continue to call the same interface uninterrupted. Apigee enables you to expose multiple interfaces to the same API, freeing you to customize the signature of an API to meet the needs of various developer niches simultaneously. |
| API base path and resources | APIs defined by network addresses and URIs. An API is made up of a 'base path' and a set of 'API resources'. Every API proxy defines a base path and, optionally, multiple API resource paths. You can think of an API simply as a set of URIs, all of which share a common base path.  To make it easier to manage your APIs, Apigee augments these raw URIs with display names and descriptions. Edge enables you to attach policies and code to URIs, enabling fine-grained control and management of the behavior of your APIs. |
| API product | A collection of API resources (URIs) combined with a quota, or 'service plan', which is published to app developers at design time. API products can in turn be bundled into API packages for monetization.  An API key is bound to one or more API products, enforcing a binding between an app and the bundle of URIs that the app is permitted to consume. |
| API package | A collection of API products that are presented to developers as a bundle, and typically associated with a rate plan defined in monetization. |
| app | An abbreviation of 'application'. The term 'app' has come to refer to mobile applications that consume APIs. Developers implement apps in a variety of programming languages, and using various technologies and platforms. Developers who want to consume APIs register apps in an API provider's organization on Apigee Edge.  When the app is registered, Apigee generates an API key and secret that identify the app. The developer embeds the API key in the app, which presents the key when making requests. API Services implements security around the API key through direct API key validation or through OAuth. |
| environment | A runtime execution context for API proxies. An API proxy must be deployed to an environment before the API it exposes is accessible over the network. By default, organizations are provisioned with two environments: 'test' and 'prod'.   * The 'test' environment is typically used for deploying API proxies during development. * The 'prod' environment is typically used for promoting API proxies from the test environment after they have been fully developed and tested. |
| organization | A container for all the objects in an Apigee Edge account, including API proxies, API products, API packages, apps, and developers.  A user account is required for each organization for which you are a member. (Most users will have an account in only one organization.) |
| policy | A processing step that executes as an atomic, reusable unit of logic within an API proxy processing flow.  Typical policy-based functionality includes transforming message formats, enforcing access control, calling remote services for additional information, masking sensitive data from external users, examining message content for potential threats, caching common responses to improve performance, and so on.  Policies may be conditionally executed based on the content or context of a request or response message. For example, a transformation policy may be executed to customize a response format if the request message was sent from a smartphone. |
| API resource path | A RESTful concept, a resource path is a uniform resource identifier (URI) that identifies the network path to a given resource. |
| version | The version of the developer-facing API interface.  For example, pivotaltracker.com/services/v3, or api.enterprise.apigee.com/v1.  This term is distinguished from 'revision', which is the numbered, version-controlled package of configuration and policies bundled into an API Proxy. API interfaces have versions; API proxies have revisions. |
| revision | A numbered, version-controlled package of configuration and policies bundled into an API Proxy. This term is distinguished from 'version', which is the developer-facing API interface. See version above. |

## NON-FUNCTIONAL REQUIREMENTS (OTHERS)

|  |  |
| --- | --- |
| **Requirements** | **Details** |
| User Experience and  UI Design | * The application will be developed only in English * The layout and graphical components will be created considering the usability factors |
| Performance | * Application will allow users to have smooth and quick access to the information or services they require. |
| Security | * Web security standards will be followed. |

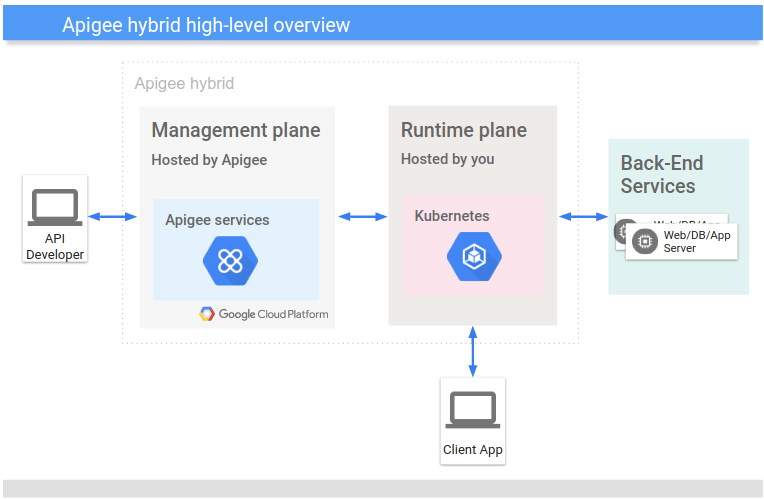
## TECHNICAL CONFIGURATIONS

### APIGEE HYBRID DEVELOPMENT ENVIRONMENT

Apigee hybrid as traditional Apigee Edge split into two distinct parts, a management plane maintained by Google in the Cloud, and a runtime plane that you install and maintain on a network that you control:

* **Management plane:** A set of services hosted and maintained by Google in the Cloud that provide Apigee Edge UI, management API, and API analytics features for hybrid.
* **Runtime plane:** A set of containerized runtime services that you set up and maintain in your own Kubernetes cluster. All API traffic passes through and is processed within the runtime plane.

The following illustration shows the general organization of hybrid into a management plane hosted by Google and a runtime plane that you install into Kubernetes on-premises or in a Cloud-managed Kubernetes provider:



One key thing to know about hybrid is that all API traffic is processed within the boundaries of your network and under your control, while management services, the UI, and API analytics run in the Cloud and are maintained by Apigee.

### Software & Hardware Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Installation Component | RAM | CPU | Minimum hard disk |
| Cassandra | 16GB | 8-core | 250GB local storage with SSD or fast HDD supporting 2000 IOPS |
| Message Processor/Router on same machine | 16GB | 8-core | 100GB |
| Analytics - Postgres/Qpid on same server | 16GB\* | 8‑core\* | 500GB - 1TB\*\* network storage\*\*\*, preferably with SSD backend, supporting 1000 IOPS or higher\* |
| Analytics - Postgres standalone | 16GB\* | 8-core\* | 500GB - 1TB\*\* network storage\*\*\*, preferably with SSD backend, supporting 1000 IOPS or higher\* |
| Analytics - Qpid standalone | 8GB | 4-core | 30GB - 50GB local storage with SSD or fast HDD  For installations greater than 250 TPS, HDD with local storage supporting 1000 IOPS is recommended.  The default Qpid queue size is 20 GB. If you need to add more capacity, add additional Qpid nodes. |
| Other (OpenLDAP, UI, Management Server) | 4GB | 2-core | 60GB |

### Port requirements for each Edge component in a single data center configurationPORT REQUIRMENTS

The table below describes the ports that need to be opened in firewalls, by Edge component:

|  |  |  |
| --- | --- | --- |
| Component | Port | Description |
| Standard HTTP ports | 80, 443 | HTTP plus any other ports you use for virtual hosts |
| Cassandra | 7000, 9042, 9160 | Apache Cassandra ports for communication between Cassandra nodes and for access by other Edge components. |
| 7199 | JMX port. Must be open for access by the Management Server. |
| LDAP | 10389 | OpenLDAP |
| Management Server | 1099 | JMX port |
| 4526 | Port for distributed cache and management calls. This port is configurable. |
| 8080 | Port for Edge management API calls. These components require access to port 8080 on the Management Server: Router, Message Processor, UI, Postgres, and Qpid. |
| Management UI | 9000 | Port for browser access to management UI |
| Message Processor | 1101 | JMX port |
| 4528 | For distributed cache and management calls between Message Processors, and for communication from the Router and Management Server.  A Message Processor must open port 4528 as its management port. If you have multiple Message Processors, they must all be able to access each other over port 4528 (indicated by the loop arrow in the diagram above for port 4528 on the Message Processor). If you have multiple data centers, the port must be accessible from all Message Processors in all data centers. |
| 8082 | Default management port for Message Processor and must be open on the component for access by the Management Server.  If you configure TLS/SSL between the Router and Message Processor, used by the Router to make health checks on the Message Processor.  Port 8082 on the Message Processor only has to be open for access by the Router when you configure TLS/SSL between the Router and Message Processor. If you do not configure TLS/SSL between the Router and Message Processor, the default configuration, port 8082 still must be open on the Message Processor to manage the component, but the Router does not require access to it. |
| 8443 | When TLS is enabled between the Router and Message Processor, you must open port 8443 on the Message Processor for access by the Router. |
| 8998 | Message Processor port for communications from Router |
| Postgres | 22 | If configuring two Postgres nodes to use master-standby replication, you must open port 22 on each node for SSH access. |
| 1103 | JMX port |
| 4530 | For distributed cache and management calls |
| 5432 | Used for communication from Qpid/Management Server to Postgres |
| 8084 | Default management port on Postgres serverand must be open on the component for access by the Management Server. |
| Qpid | 1102 | JMX port |
| 4529 | For distributed cache and management calls |
| 5672 | * **Single data center**: Used for sending analytics from Router and Message Processor to Qpid. * **Multiple data centers**: Used for communications between Qpid nodes in different data centers. |
| 8083 | Default management port on Qpid server and must be open on the component for access by the Management Server. |
| Router | 4527 | For distributed cache and management calls.  A Router must open port 4527 as its management port. If you have multiple Routers, they must all be able to access each other over port 4527 (indicated by the loop arrow in the diagram above for port 4527 on the Router).  While it is not required, you can open port 4527 on the Router for access by any Message Processor. Otherwise, you might see error messages in the Message Processor log files. |
| 8081 | Default management port for Router and must be open on the component for access by the Management Server. |
| 15999 | Health check port. A load balancer uses this port to determine if the Router is available.  To get the status of a Router, the load balancer makes a request to port 15999 on the Router:  curl -v http://**routerIP**:15999/v1/servers/self/reachable  If the Router is reachable, the request returns HTTP 200. |
| 59001 | Port used for testing the Edge installation by the apigee-validate utility. This utility requires access to port 59001 on the Router. See [Test the install](https://docs.apigee.com/private-cloud/v4.18.05/test-install) for more on port 59001. |
| SmartDocs | 59002 | The port on the Edge router where SmartDocs page requests are sent. |
| ZooKeeper | 2181 | Used by other components like Management Server, Router, Message Processor and so on |
| 2888, 3888 | Used internally by ZooKeeper for ZooKeeper cluster (known as ZooKeeper ensemble) communication |

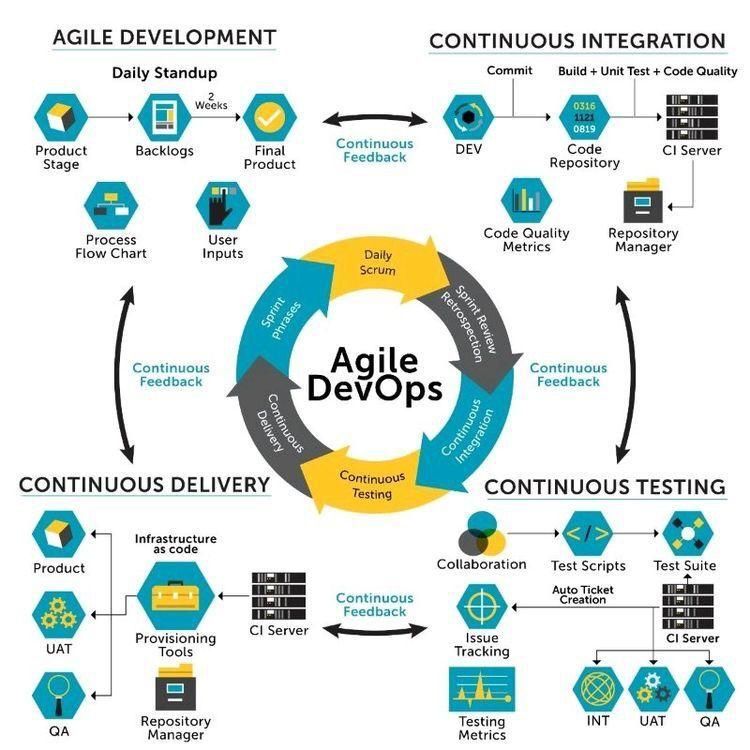
### TECHNICAL GUIDELINES

These guidelines provide instructions and conditions that will be adhered to during the development of the mobile application.

* API will be used, as the case may be, in realizing the features and functionalities mentioned.
* The client will finalize the functional requirements and UI/UX before the commencement of the project.
* Verbat will be testing the app in the mentioned devices only. Testing on devices other than the ones mentioned under the “Technical Standards “will have to be specified and provided by the client at the beginning of the development phase.
* The client will have to provide the details of the testing devices they are using before the start of development phase.
* Client should provide the relevant Developer's Account credentials before the development phase. In case Verbat needs to create the developer id, additional charges will be incurred by the client.
* The duration mentioned in the project timeline is for the development and testing; any delay or time taken by the review team to respond will not be Verbat’s responsibility.
* Any clarification required from client needs to be addressed within 02 business days.
* The apps will be developed / created within the guidelines of respective play store.
* Verbat will strictly follow the guidelines provided by the respective stores.
* Verbat will inform the client if any of the client requirements / request deviate from it.
* Customization of the features of the app will be susceptible to the limitation imposed by the respective platform/ store.
* Once development commences, the test device/screen sizes will not be susceptible to change. Any change requested by the client will have to go through change management.
* OS version support will be limited to the ones mentioned in the technical specifications. Further support will have to go through change management.

## PROJECT DELIVERY

### Agile Methodology



Verbat’s philosophy has been to deliver products that allowed clients to be intimately involved with the development activity. As early as 2005 Verbat had adopted the agile development practices into its pipeline for software development. This allowed clients to closely observe the product being developed. It created a tight feedback loop that allowed us, along with our clients to better understand the requirements and build a right sized and proportional product.

With the advent of next generation applications and tools that support a wide range of activities related to development, deployment and integration; the feedback loops became tighter and coupled, while at the same time it allowed us to create applications that were decoupled. Thus giving us the capability to have greater flexibility in development and deployment.

While development broke down the barrier between developers and management, Devops broke the barrier between software developers and operations teams. Our development and operations teams work under a single silo. While our Scrum teams are cross functional, in our quest to adopt better integration with operations, we have realized that Devops is not just a set of tools and processes, but a mindset and culture. We have fostered a culture collaboration and communication. Our scrum teams and Devops teams share a symbiotic relationship and they work united towards a common goal.

As a company, we have adopted the following tenets in our Devops philosophy

**Speed:** i.e. Move at a high velocity so as to innovate faster for customers and adapt to changing markets better, and grow more efficient at driving business results.

**Rapid Delivery:** Increase the frequency and pace of releases so we can innovate and improve products faster and thus respond to customers’ needs and build competitive advantage.

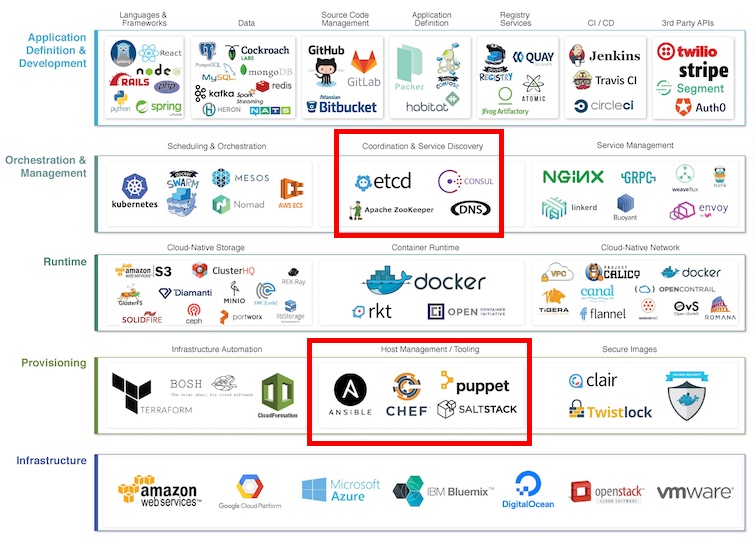
**Reliability:** Ensure the quality of application updates and infrastructure changes so we can reliably deliver at a more rapid pace while maintaining a positive experience for end users.

**Scale:** Operate and manage infrastructure and development processes at scale.

**Improved Collaboration:** Build more effective teams under a DevOps cultural model, which emphasizes values such as ownership and accountability. Developers and operations teams collaborate closely, share many responsibilities, and combine their workflows.

**Security:** Move quickly while retaining control and preserving compliance. We adopt a DevOps model without sacrificing security by using automated compliance policies, fine-grained controls, and configuration management techniques.

Please see below an illustration of our technology stack. The tools of the trade that allow us to practice an Agile and Devops centric framework



### PROJECT MANAGEMENT

The Verbat development center strictly follows industry standards on quality. Our project management process is governed by the Verbat Quality Management system and is verified through internal audit programs.

Verbat will dedicate a project leader for the proposed implementation. Furthermore, Verbat proposes the client to identify one project manager (*CSPM – Client-Side Project Manager)* who will be driving all activities undertaken by the client, and will be the single point of contact for Verbat.

### ROLES & RESPONSIBILITIES

Verbat will assign its own dedicated Project Lead (*VPL – Verbat Project Lead*) for the client. The VPL will be responsible for planning and management of all activities related to the project. Furthermore, the VPL will work closely with CSPM, on all periodic status updates and will ensure high level visibility and comfort on the progress of the project.

### DELIVERY ACTIVITY SUMMARY

|  |  |
| --- | --- |
| Activities | Description |
| Proof of Concept | Working Model of the project with two languages. |
| Detailed Requirement Analysis | Verbat team to conduct detailed study of requirement for the phase. If clarification is required, the team will reach out to Client for more information and/or time for discussions. |
| DB Design | DB design for central and test DB. |
| Software Requirement Specification document (SRS) | Once the requirement analysis is completed, Verbat team will submit the SRS document for approval |
| UI/UX Design, Prototyping | Based on the SRS, Verbat UX/UI team will work on the UI/UX of the screens and submit prototype for approval |
| Functional Specification Document (FS) | Once the UI/UX is approved, Verbat shall submit an FS document for approval. |
| Development | Actual system development starts based on the FS. This involves detailed design and software development of Web Application. |
| Testing | Test Planning, test plan creations, internal, integration testing and user acceptance testing. |
| Deployment | Deploying the latest build in the Verbat Test Server. |

### PROJECT IMPLEMENTATION PLAN

Verbat will be providing the solution in a stand-alone fixed bid approach which ensures minimum viable solution for quick wins with core focus on the long-term business objective and outcome. Once the implementation is over, Verbat will initiate the application maintenance process (*once the maintenance contract is signed*) which continues to extend after the implementation.

### DELIVERABLES

* Proof of Concept
* Project Plan
* Software Requirement Specification Document (SRS)
* Prototype Design
* Functional Specification Document (FS)
* Fully Developed & Tested Application
* Source Code

### ESTIMATED DELIVERY TIME

The timeline estimated for delivering the application will be as below:

* Working days to complete the proof of concept on completion of resource mobilization.
* Working days to complete the SRS & prototype of Language Translation Application.
* Working days to complete the development.

| **Activity** |
| --- |
| Contract Signoff (T0) |
| Proof of Concept-Allocate Resource |
| Proof of Concept-Complete |
| Proof of Concept- Approval (T1) |
| Detailed Requirement Gathering |
| Software Requirement Specification Document (SRS) |
| SRS Approval (T2) |
| System UI/UX Prototype-Complete |
| System UI/UX Prototype-Approval (T3) |
| Functional Specification (FS) |
| FS Approval (T4) |
| Development Phase-Complete |
| Perform QC (Unit Testing and Integration Testing) |
| System ready for UAT |
| UAT Acceptance on Verbat production server (T5) |

*Note:*

* *The above-mentioned timeline is in Working Days.*
* *Upon project confirmation, Verbat requires a lead time of minimum () working days for resource mobilization for proof of concept.*
* *Resource mobilization will be initiated post the confirmation of the project along with LPO, signed*

*proposal and advance payment.*

* *The initiation of the UI/UX/Prototype development is dependent on the confirmation of SRS.*
* *The above-mentioned timeline for development is post-confirmation of FS.*
* *Documentation submitted after project initiation and system study supersedes any proposal or documentation submitted during initial requirement gathering / discussion / negotiation.*
* *Project plan will be submitted post the confirmation of project with necessary payments.*
* *Any delay in getting the approvals of deliverables from client will cause change in timelines and the revised timelines will be updated in weekly status reports shared with client after the project commencement.*
* *All approvals and queries regarding the client requirement and any queries which may hinder the project advancement at any stage should be answered by the client within () business days from the time of initiation, failing which, the time delay will get added to the actual effort and timeline estimated.*

### RISK CONTINGENCY PLANNING

Verbat has identified various risk factors associated with this assignment and understands the impact of these risk factors on the project schedules. The objective of this section is to highlight for both Verbat and client, the risk factors, to analyze the impact of the risks on project execution, and to propose strategies to control and reduce the impact of the risk factors. These various risks, which could arise during the project, are tabulated below along with mitigation implementation.

| **Type of risk** | **IMPACT** | Risk Mitigation | Risk Handling |
| --- | --- | --- | --- |
| Scope Creep | **H** | Functions and features will be detailed in system requirement document and will go through client approval. Once this document is approved, all changes will go through the change management process for impact assessment. | Proper change management procedure will be implemented |
| Delay in client feedback | **H** | The plan is prepared with sufficient lead-time for reviews and approvals.  The client will advise us on all dates connected to the document review and approval. | The request for feedback will be escalated if not attended to at the right time so that the schedules are not affected. Deemed acceptance criterion is finalized up-front and will be followed |
| Non-availability of necessary software, frameworks, database instances and infrastructure at client’s hosting environment (If hosting support is provided by Verbat) | **M** | Client will be informed in advance on these requirements. | Possible impact to schedule |
| Manpower attrition | **L** | All efforts will be made to ensure that all initiatives are process dependent. To mitigate risk Verbat/Client will train a person to ensure all back-ups are in place. | A new person will be appointed as early as possible, provided the required project-specific training and mentoring is in place - to minimise impact of attrition on the project |

*H-High, M-Medium, L-Low, NA-Not Applicable*

# OUT OF SCOPE

With the ever-evolving digital market, the requirement should be clear to both the parties involved, hence the importance of mentioning the out of scope details of the project. Following are considered to be out of scope while creating this proposal:

* Purchase of images, fonts
* Any language other than English
* Migration of existing data / Database migration
* Content writing / proof reading / Data Replication / Manual data entry
* Content or image procurement or uploading or editing
* Developer account creation and Maintenance (*IOS and Android Store*)
* Adding new features to the application other than mentioned in the functional specifications. Such requests will be handled via change management. For Change management details, please refer to section titled “Change Management” in the Proposal. (*refer page no: 33*)
* Annual Maintenance Contract (*Bug fixing, debugging, enhancements*) – Please refer to section titled “Maintenance and Support”, unless contracted for. (*refer page no: 34*)
* Hosting Infrastructure and Maintenance (*web and email hosting*), unless contracted for.
* Backup solution and Disaster recovery unless contracted for.
* Physical deployment on-site / installation of the application in devices and physical connection, installation of system.
* Integration with third-party, if any, other than mentioned in the functional specifications
* Hardware Integrations / procurement and purchase
* Procurement of SMS gateway / payment gateway / email gateway
* Integration of SMS gateway / payment gateway
* SSL Purchase and installation, if any
* Plugin/template purchases, if any
* API purchases
* OS other than mentioned in the Hardware Interface
* Relevant / related software libraries
* Mobile offline access or operations

# CHANGE MANAGEMENT

Any addition which comes out of the project scope, upon and after the launch of the website will be considered as change management. Verbat recommends the following change management procedures for the same:

* Changes will be implemented only after raising a change request.
* Change requests will be studied and an impact analysis will be performed on the existing work flow.
* Upon assessment of the impact, effort estimation will be calculated and raised as an additional requirement.
* The change request will be initiated only after receiving a formal approval from the client for the additional changes raised.
* Changes which are out of scope will be charged @ per man day rate.

Activities for change scheduled

Modules checked in

Unit testing done

Change request accepted

Informed of the Action

Implementation of changes

Modules checked out

Activities planned and scheduled

Effected modules identified

New system released with change request

Changes made & documented

System

Need for change recognized

User submits change request

Evaluate for cost, schedule & effort

Change control authority

yes

No

## MAINTENANCE & SUPPORT

* Maintenance contracts by default are supported as per the basic SLA terms.
* AMC with Basic SLA is charged at % of the total project value. Additional Effort/change management request will be added towards Total Value of the Project to determine the AMC value.
* Maintenance support is limited to providing application support for ensuring the consistency of the look-and-feel, bug fixes and user issues i.e. maintenance and support of the existing features of the application.
* Support does not in any way cover providing technical or other support to the end users or hardware support. The maintenance agreement does not include functionality changes or feature additions which are handled as change requests which will be charged per man day rate. AMC does not include server support, maintenance and application deployment.
* AMC charges will cover Off-Site Support and Debugging. Support includes E-mail, telephone and video conference (if required). In the event, the application is hosted with the client or if it is a client server development; necessary remote desktop connectivity should be provided for carrying out maintenance activities.
* All maintenance support will be executed by Verbat off-site team. In the event of any need for on-site work, all expenses incurred for such trips (flights, accommodation, meals transportation) will be payable to Verbat by client.
* Gap in AMC - In case if the client does not opt an AMC for a year and wants to renew it after that period, % of the AMC amount for the year for which AMC is not taken will also be payable if the client wishes to renew the AMC contract.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SLA Type** | **Max Response Time** | | **Max Resolution Time** | **Target** |
| Basic | 1 working day | 3 working days | | Request / incident / problem tickets |

*Note:*

* *Time zone applicable as per India Time Zones (3:30 GMT to 12:30 GMT, Monday to Friday)*
* *Please note that the AMC support shall start only after all the necessary sign-offs (AMC Document) to this effect have been given.*
* *It is not mandatory that the client should opt for an AMC. The client will still be supported on an ad-hoc basis on an agreed man-day rate.*
* *AMC Payment Terms: 100% to be paid as advance.*

AMC Option: Client can opt for time and material based Annual Maintenance, the details of which will be shared post the completion of project.

# TERMS AND CONDITIONS

## ACCEPTANCE CRITERIA

* UAT (User Acceptance Test) sign off should take place within Days from the first release of the application and the acceptance confirmation needs to be mailed to Verbat failing which Verbat will consider the project as approved by the client.
* Any comments and reasons for rejection need to be documented and the same needs to be sent as an email from the official e-mail id of client to Verbat on or before days from the first release.
* Timeframe for acceptance for any further release will be mutually agreed and finalized between client and Verbat depending on the UAT Comments

## WARRANTY

* Verbat shall provide a bug ﬁx warranty at no additional cost for 30 days from the date of acceptance of the project, for correction of any errors in the developed application that may be attributed to Verbat.
* However, this does not cover modifications by Client, or use of the application on an environment other than the proposed environment, or other circumstances outside Verbat’s reasonable control. In such cases Verbat reserves the right to charge for its services.
* All error corrections will be executed by Verbat offsite team. In the event of any need for on-site work, all expenses incurred for such trips (flights, accommodation, meals, transportation) will be payable to Verbat by Client.

## SOURCE CODE & INTELLECTUAL PROPERTY RIGHTS

* Upon completion of the Project and 100% completion of the payment, the client will have access to the source code except for proprietary codes, developer tools and third-party applications etc.
* The solution offered will be the intellectual property of the client and will be made available to the client on an “unlimited license” basis.
* No person or organization, other than Verbat or any person authorized by Verbat in writing, has permission to modify/change the software solution to be eligible to get continued support from Verbat as per the support terms defined under this document.
* Verbat accepts no liability or damages of any kind arising out of use or non-use of the software delivered. The responsibility of testing of software lies with the client.

## GENERAL TERMS AND CONDITIONS

* Offer Valid for 30 calendar days from the date of submission of the Proposal.
* An average of 20 working days are assumed in a month.
* All the development activities will be carried out from our off-shore development center in India.
* All the documentations will be provided in English.
* Third party components may be used to develop this application.
* The scope of the project is to develop the application as detailed in the scope of the project and mentioned in this proposal. Any changes or additions will have to go through our change management process.
* This proposal is derived and concluded from either the RFQ /RFP/data shared via email / information transferred during an initial requirement analysis meeting / tele-conversation. Verbat reserves the right to change the terms of this proposal if the final terms (including the costing), features & functionalities and timelines are changed during the course of the project. Hence any fees quoted / timeline committed in this proposal may not be considered as final unless agreed upon and signed by both parties.
* Web Application will be best viewed only in the environment mentioned in the section Browser Compatibility.
* Mobile app will be best viewed only in the environment mentioned in the section Hardware Interface.
* All source codes and other project artifacts will adhere to the Verbat document templates and internal coding standards.
* The documents delivered to the client include the ones mentioned under ‘Deliverables’ and these will adhere to Verbat’s internal document standards.
* Acceptance criteria shall be based on the clauses which were mutually discussed between Verbat and client at the Requirement Analysis phase. The same will be documented and approved by both parties through official emails.
* In the event that the Client requires any extension of the proposed acceptance schedule, the associated effort and cost of such extension will be mutually reviewed.
* If the project needs to be put on Hold / Stopped, a minimum notice period of 1 week is required along with the duration of the holding period. Thereafter Verbat will make a final decision based on the request.
* If deployment is done in the client’s server, Verbat will not be held responsible for any performance issues arising due to hardware malfunctions.
* The Client is responsible for all data-backups in case the application is not hosted on the Verbat server.
* All source codes will only be delivered or uploaded on the production server after the due payments are made to Verbat.

## GENERAL ADMINISTRATIVE, TECHNICAL & FUNCTIONAL ASSUMPTIONS

* Detailed system study is required before the start of the project.
* During the requirement gathering phase, authorized personnel from the Client’s side are expected to be available for discussions and finalization of the HLD (High Level Design), before the development commences.
* Type of reports, formats if under the scope of the project, need to be specified by the Client before the project sign off.
* Workflows if under the scope of the project, need to be specified/ confirmed by client before project signoff.
* Verbat assumes that all sign-offs from the Client will be provided within the agreed and specified timeframe.
* Client will provide all the necessary contents, both text and imagery, before starting the project in the format suggested by Verbat (if any).
* The client should provide the relevant information and data, well in advance of the execution of the related activity. Non- availability of this information or data may lead to an interruption of work, which may result in a delay in delivery as well as additional costs to the client.
* The Client should possess a server with the technical specifications recommended by Verbat for the proposed application.
* The Client will be provided with a one-time training (train the trainer) on how to use the application via a video conference (maximum of 4 hours). Additional training requests will be charged at cost to the client.
* Cost of all third-party components to be borne by the client.
* Application will support in three languages which are supported by Google Text to Speech API.

# FINANCIALS

## Curriculum Management Application

|  |  |  |
| --- | --- | --- |
| Sl. | Description | Amount (USD) |
| 01. |  | XXXX USD |

*Note:*

* *The above cost is exclusive of VAT applicable in UAE*
* *The above cost includes the cost of third-party service of non-translated calls.*
* *The above cost does not include third-party API purchases.*
* *The above estimate is based on the initial understanding of the requirement grounded on the details shared by client via meetings and tele-conversations. If Verbat finds during the requirement gathering phase that the actual requirements uncovered are different form the original, this might affect the estimates provided and Verbat shall exercise rights to renege on the contract. Clarifications/Changes in the scope during system study/ analysis may call in for additional effort and timeline.*
* *For feature additions, please refer section titled “Change Management “.*
* *LPO to be raised in the name of “Verbanet Technologies LLC” for project initiation*

## PAYMENT TERMS

* % of the proof of concept as advance.
* % of proof of concept on delivery of POC
* % of Language Translation Application as advance.
* % of Language Translation Application on completion of:
* SRS
* Prototype
* % on completion of development in UAT release of the Web Application on Verbat server/Mobile build.
* % of Language Translation Application to be paid on UAT Signoff.

*Note:*

* *Payment should be made within 7 days from the date of invoice.*

### MODE OF PAYMENT

Wire transfer to our bank account

|  |  |  |
| --- | --- | --- |
| Particulars | **Option 01** | **Option 02** |
| Bank Name | Commercial Bank of Dubai | Emirates NBD |
| Account Name | Verbanet Technologies LLC | Verbanet Technologies LLC |
| Account Number | 1000305555 | 1011492858201 |
| Swift Code | CBDUAEAD | EBILAEAD |
| IBAN Number | AE840230000001000305555 | AE61 0260 0010 1149 2858 201 |
| Bank Address | Al Qusais Branch, P.O Box 2668, Al Qusais | Mamzar Branch, Dubai |

*Note:*

* *Bank charges incurred during wire transfer to be borne by the client.*
* *Any local taxes / VAT applicable to be borne by the client.*
* *Client invoices will include VAT charges in addition to the application cost.*

# Annexures

## Vendor Experience

* How long has your company been in business?
* How long has your company been selling and supporting your API Hub product?
* How many customers do you have in UAE?
* If any components of your API Hub solution can be sold separately? Ans. No. It cannot be sold seperately
* Explain how API Hub fits into in-house Integration. Ans. APIGEE can be customized to fit an in-house integration because it supports hybrid mobilization of the platform. For more details please speak to a representative from our company
* How many customers use your API Hub solution - please breakdown by Govt/Private Entities
* Please describe the roadmap for your API & Integration solution
* Do any analysts rank your API Hub solution? If so, please enumerate analysts and rankings
* Please describe how your offering integrates with ESB style or other integration technologies using open standards and automation
* Does your organization provide complementing integration technologies that fit into the API Hub solution?
* Please describe your organization's experience of integrating systems providing examples in our industry if possible
* Please provide references for your API Hub solution
* Do you provide free training for architecture, development, and operations on your website?
* Do you provide online tutorials to help us learn your product?
* Please provide a link to your product's user community

## Deployment, Architecture & Administration

1. Does your solution provide flexible deployment options across public and on premise environments?

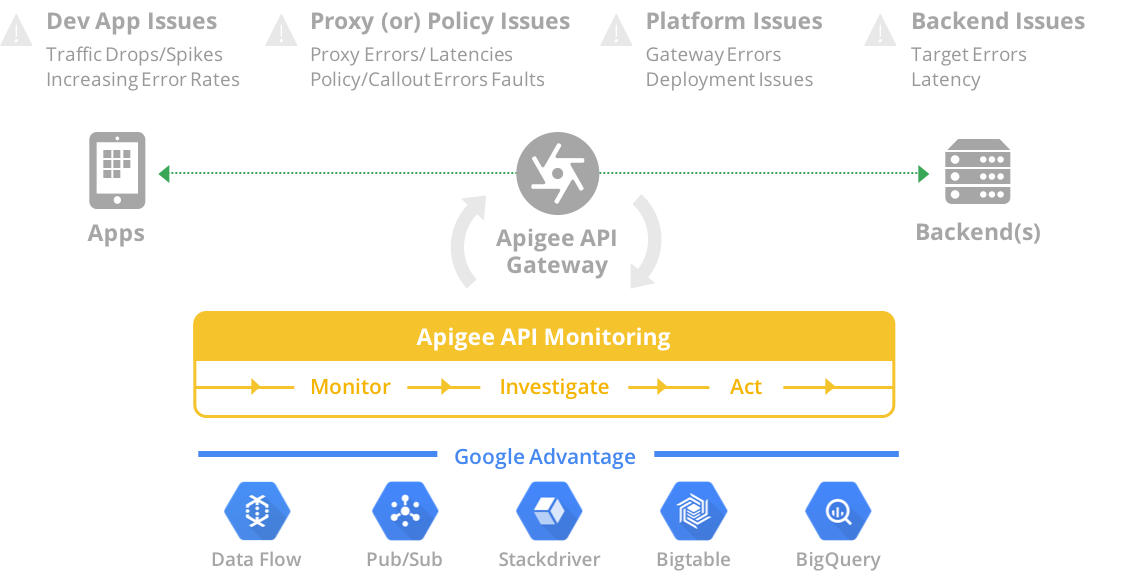
Ans. APIGEE Hybrid provides a flexible solution that consists of a two distinct parts. A public google cloud based management framework that manages the private network where the solution is hosted. The private network is a containerized Kubernetes runtime service that manages all API traffic to the backend systems

1. Does on-premise API hub solution support multiple languages, e.g. English, Arabic, etc.?

Ans. Yes, Apigee Developer Portal supports out of the box multiple languages and have very good flexibility when it comes to different languages. Reason is, Apigee Developer portal is built on top of famous Drupal CMS. However this would require additional time for customization oi its framework

1. Does your solution include a console to manage and monitor all infrastructure components, i.e. gateway servers, management servers, load balancers, etc., of your API Hub solution?

Ans. Apigee API Monitoring helps operations teams increase API availability for application developers, customers, and partners. Apigee API Monitoring works in conjunction with Apigee Edge Cloud to provide real-time contextual insights into API performance, helps quickly diagnose issues, and facilitates remedial actions for business continuity.



API Monitoring enables you to:

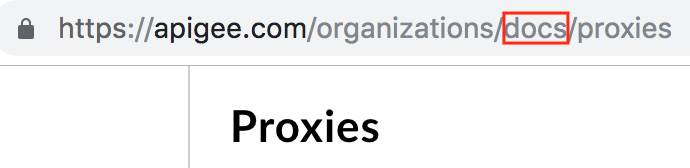
* Increase API availability and reduce the mean-time-to-diagnosis (MTTD) by quickly investigating issues with your APIs.
* Take appropriate actions based on contextual alerts before consumers are impacted.
* Leverage Apigee fault codes to speed diagnosis.
* Isolate problem areas quickly to diagnose error, performance, and latency issues and their source, such as developer apps, API proxies, backend targets, or the API platform.
* Leverage the best of Google technologies, such as Data Flow, Pub/Sub, Stackdriver, Bigtable, and BigQuery, to compute complex metrics efficiently and at scale.

1. Can a single installation and instantiation of your API Hub solution support multi-tenancy?

Ans. An organization is the top-level container in Apigee Edge. It contains all your API proxies and related resources. By default, your organization name is in the URL used to call your API proxies, as described in About virtual hosts. For example:

http(s)://**your\_org\_name**-**environment**.apigee.net/proxy\_base\_path/...

* Your organization name is in the URL of the Edge management UI. For example, the following URL displays the API proxies for the docs organization:



* While you may have created only one organization, you can belong to other organizations as a user or administrator with specific permissions. In the Edge management UI, if you belong to more than one organization, you can switch to a different organization as described in Switching between your organizations.
* When you make calls with the management API as a user in the Organization Administrator role, the organization is a required part of the path in most calls. For example, the following management API cURL request returns a list of all API proxies in an organization:

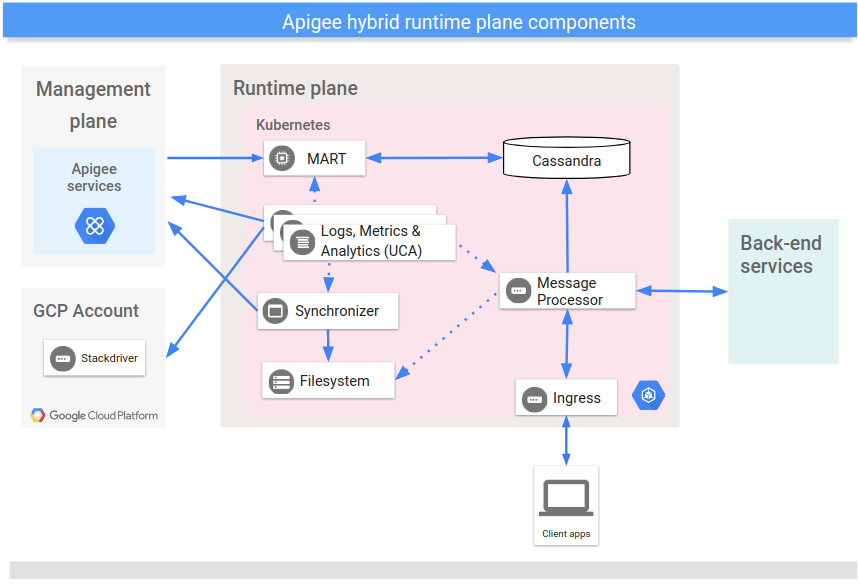
In short one single instance of APIGEE can support multi-tenancy through the concept of Organization.

1. Please describe the availability and resilience characteristics of the solution and any SLAs associated with it

Ans. This is part of our APIGEE infrastructure monitoring services. Users assigned to the API Monitoring Roles can do the following

* View metrics, logs, collections, alerts, and custom reports
* Create, update, and delete alerts and collections
* Create, update, and delete custom reports

1. Please provide a logical overview diagram of the components required to make up your solution for on premise deployments

Ans.

APIGEE runtime (Hybrid) runs on a Kubenernetes cluster that you own and maintain. As shown in the above diagram, the runtime plane includes several main services, which run in separate pods in the cluster. The services are described in more detail in the following sections.

**Message Processor**

The hybrid Message Processor (MP) brings the entire functionality of Apigee Edge related to API request processing and policy execution to the runtime plane. The MP loads from local storage all of the deployed proxies, resources, target servers, certs, and keystores for target TLS, caches, KVMs, flow hooks, and so on for a specified organization's environment. You must configure an Ingress controller to expose the MP to requests coming from outside the cluster.

**Synchronizer**

The Synchronizer's primary job is to fetch configuration data about an API environment from the management plane to the runtime plane. The runtime plane needs this environment-specific configuration data to do its work. This downloaded data is also called the contract.

The Synchronizer periodically polls the Management Server for changes and downloads a new configuration whenever changes are detected. The configuration data is retrieved and stored locally as a JSON file in a location where the Message Processors can access it.

The downloaded configuration data allows the runtime plane to function independently from the management plane. With a config, MPs on the runtime plane can bootstrap and run using the locally stored data as their configuration. If the connection between the management and runtime plane goes down, services on the runtime plane continue to function.

The configuration data downloaded by the Synchronizer includes:

* Environment information, including the environment name, the organization name, and feature flags.
* Shared API resources, including the resource scope, resource names, locations, and properties.
* Target server definitions
* TLS settings, including the location of keys and certificates
* Environment caches
* Key value map names (the runtime is responsible for initializing and populating KVMs).
* Proxy bundles and shared flow deployments, including the name, revision, type and location.
* Flow hooks
* API products
* Data masks

**Cassandra data store**

Cassandra is the runtime datastore used to provide Apigee core persistence services (CPS) for the runtime plane.

You deploy the Cassandra database in Kubernetes in a StatefulSet node pool, as Cassandra is a distributed data system requiring state to be managed on the runtime plane. Locating these entities close to the runtime processing services helps support requirements for security and high scalability.

The Cassandra database stores information about the following entities:

* **Key management system (KMS) data**, including companies, developers, developer apps, API products, and API keys
* **Key value map** (KVM) data
* **Response cache** data
* **OAuth data**, including access tokens, refresh tokens, and authorization codes
* **Quota data**, including buckets and counters

**Management API (MART)**

The Management API for Runtime data (MART) interacts with the local Cassandra datastore. The public Apigee Edge management APIs use MART APIs to access and manage data entities on the runtime plane, such as KMS (API keys and OAuth tokens), quotas, and API products.

Apigee Edge management API calls to access and manage these entities are sent to your local MART server from the management plane. For example, if you create a new API product using the Edge management API, the management plane calls MART APIs to update the runtime plane.

1. Please provide an architectural diagram that shows how a highly available solution could be built on premise

Ans. Same as above

1. Please describe the ability of the solution to be monitored by external enterprise tooling, listing any enterprise tooling that works directly with your solution. Also include what metrics and information can be monitored.

Ans. As mentioned earlier APIGEE provides tools for monitoring services. It can also provide metrics on the services being monitored.

1. Can multiple teams work independently with runtime isolation?

Ans. Yes, Organizations allow multiple roles to be assigned to multiple users across multiple organizations. Users from one organization can be isolated from another organization

1. Please describe what tooling is available for administrators and operators to interact and control the deployed environments for on premise.

Ans.

1. ApiStudio.io: Use openApi to design your APIs, Share API Spec with your team, Test your APIs

2. Apigee-127: Model First toolkit to build enterprise class APIs, Open Source, Built on Node.js

3. Apigee Tool:

Command line tool for deploying API proxies and Node.js applications to the Apigee Edge platform. The tool also lets you list and un-deploy API proxies.

Built on Node.js

Listing deployed API proxies on Edge. Access API Proxies on your local machine instead of Management UI.

Retrieving deployed proxies and apps from Edge.

Deleting proxy definitions from Edge.

Retrieving log messages from Node.js apps deployed to Edge.

4. Apigee Maven Plugin :

apigee-edge-maven-plugin is a build and deploy utility for building and deploying the Apigee ApiProxy's/Application bundles into Apigee Edge Platform. The Maven build strategy is a good strategy if your current enterprise build and deploy strategies already use Maven or Maven compatible tools.

5. Apigee Grunt Plugin :

Tool to manage the API Development Lifecycle for Apigee. Pluggable environment (thousands of npm and grunt modules and plugins). Grunt is perfect for applying continuous improvement by easily adding custom tasks.

6. Sweetlime :

SweetLime is Sublime Text 3 plugin to develop Apigee Proxies. With built in template support for policies, steps and flows, SweetLime helps proxy developers build proxies with ease.

Proxies developed with SweetLime can be deployed to either the Apigee cloud or your on-prem apigee edge installation (OPDK customers). Great tool to quickly build API Proxies in Apigee Edge by coding instead of configuring. Developers love this tool where they can code instead of configuration in Apigee UI

7. OpenAPI2Apigee : (Earlier called swagger2api)

Command line tool that converts OpenAPI ( swagger ) spec 2.0 file to Apigee API Proxy Bundle. Built on Node.js. Quickly build API Proxies & Flows using OpenAPI spec.

Deploy API Proxies generated from OpenAPI Spec to Apigee Edge.

8. Apigee2openapi : (Earlier called Apigee2Swagger)

Command line tool that converts Apigee API Proxy Bundle to OpenAPI spec 2.0 file .

Built on Node.js. It will help to generate OpenAPI 2.0 spec for Apigee Developer Portal Smartdocs. Find more detailed article here on using same.

9. Swagger-Utility:

Swagger utility is a Java utility for generating API proxies from Swagger API Model. So it is Swagger API Model to API Proxies genrator tool. Similar to Swagger2Api but built on Java. Supports Apigee-127 generated swagger files.

10. Apigee Javascript Debugger :

A library providing utility methods to support the execution and debugging of Apigee Edge JSC callout policies on your local machine. This library requires no additional code in your javascript, and it executes outside the core framework of Edge.

11. Apickli :

Apickli is a REST API integration testing framework based on cucumber.js.

It provides a gherkin framework and a collection of utility functions to make API testing easy and less time consuming. Apickli is also available as an NPM package.

Cucumber.js is JavaScript & Node.js implementation of Behaviour Driven Development test framework - Cucumber. Cucumber.js is using Gherkin language for describing the test scenarios in BDD manner.

12. Apigee Edge Node SDK :

Node.JS SDK for Apigee Management API. Easily integrate Apigee Management APIs into your existing Node.JS Applications. For Example, If you want to build Apigee Developer Portal using Node.JS framework it will be extremely useful.

13. API Platform Tools - Python :

Similar to Apigee Tool, But python based. Project contains tools for the Apigee Gateway Services platform. Deploy API proxies. Create skeleton proxies. Deploy Node.js applications

14. Edge-PHP-SDK :

PHP SDK for Apigee Edge Management API. Apigee Developer Portal uses this SDK to communicate with Apigee Edge. Helpful for Integrating Apigee Edge with PHP based applications.

15. ApigeeDM - Apigee Deploy Multiple Proxies :

Deploy multiple Apigee API Proxies at one go using command line. Built on top of ApigeeNodeTool. Support for integrating into other scripts

16. API2Swagger:

Swagger 2.0 Specification Generator. CURL Like command syntax. Support JSON/Plain text request/response. Support for GET / POST / PUT / DELETE http method. Saves lot of time, changes the way you write API documentation / display same using smartdocs / swagger UI. Runs on local machine, built on top of Node.JS

17. OpenAPI Specification Generator:

Generate Open API Spec: Easy step-by-step process. Share Open API Spec with Team. Collaborate on the cloud

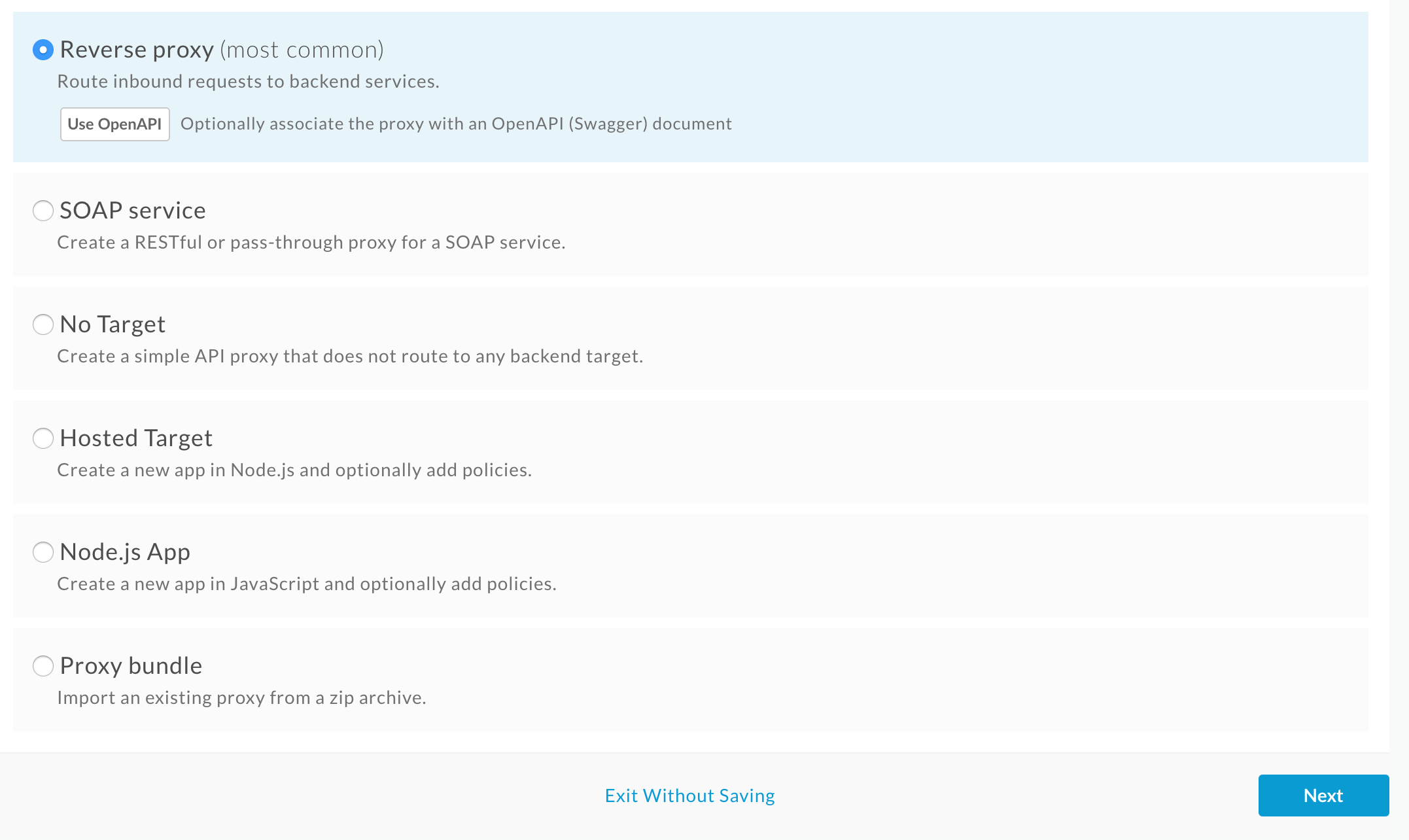
## API Creation & Deployment

1. Does your solution support out-of-the-box connectors to other systems?

Ans. Yes it does.

1. Does your solution support the automatic creation of APIs and their implementations via out-of-the-box connectors to backend systems or data stores? Please describe how and also list backend systems or data stores.

Ans. The solution does support the automatic creation of API’s. See screen below



A complete description of the API creation process is lengthy and tedious since there are multiple ways this can be done. Refer to the APIGEE documentation for more details. Verbat can help the client with this during the implementation phase

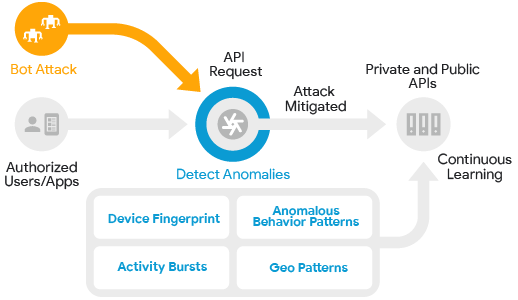
1. Does the product support OpenAPI (formerly known as Swagger) to design APIs and generate documentation?

Ans. SmartDocs lets you document your APIs on the Drupal 7 developer portal in a way that makes the API documentation fully interactive. Interactive documentation means portal users can:

* Read about your APIs
* Send a live request to your API
* View a live response returned from the API

1. Can your platform reference existing assets such as encryption libraries, schema validation tools, data validation libraries, etc.? Ans. Yes
2. How does your product support threat detection by detecting fraudulent data injections at the API level?

Ans. Apigee Edge provides end-to-end security across all components of the API management platform. Proactively address business priorities and security needs for data protection, threat detection, access control, identity management, and governance. Apigee services are third-party audited, compliant, and certified for PCI DSS, SOC1, and SOC2.

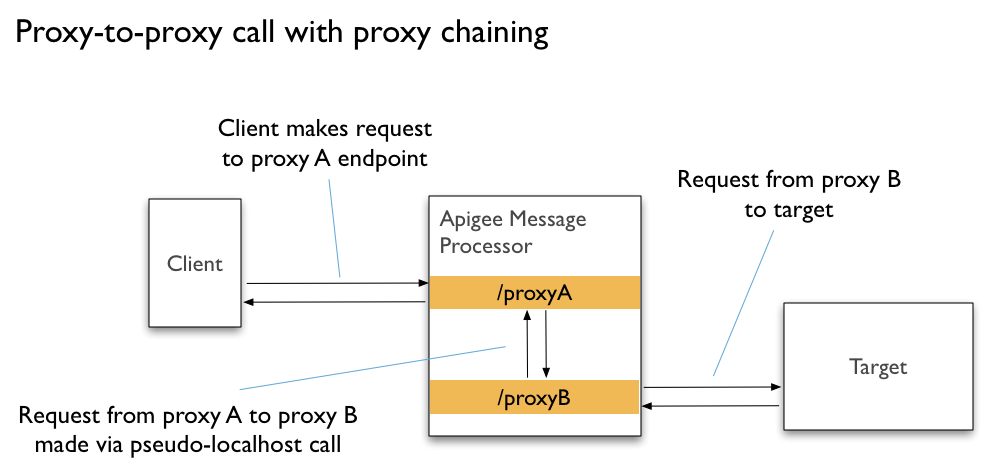


Apigee Sense adds a layer of API security using call pattern data, analyzes threat patterns in the API layer, monitors background behavior, and reports suspicious behavior.

1. Does the platform support publishing SOAP, REST, JSON, and XML style services as APIs as well as JMS?

Ans. APIGEE API management solution supports industry standards including HTTP, HTTPS, REST, SOAP, WSDL, XML, XSD, XPATH, XQuery, REST, WADL, JSON, and JMS. APIGEE is a cross cloud API platform

1. Are standard transformations included? (XML to JSON, JSON to XML, SOAP to REST, REST to SOAP). Ans. Yes, it is included
2. How is versioning supported? Ans. Versioning of API’s are managed by traditional versioning tools. However if the question refers to versioning of the API for customers or clients, then a common sense approach is recommended. APIGEE leaves that up to the implementer
3. Does the solution allow the storing and querying of arbitrary schema-less JSON data? Ans. It does not
4. Does the platform provide user management and social relationship functionality for building personalized applications? Ans. As mentioned earlier, User management, roles, organizations, authentication, federation etc. are supported by APIGEE
5. Does the platform support extensions using common languages like C# and JavaScript? Ans. Not sure what the question means.
6. Can your product publish APIs for external and internal consumers? How are these managed independently? Ans. Internal API’s are managed using proxy chaining. (Aside from the various security policies that restrict API access from the outside).



Proxy chaining uses a local connection to minimize network overhead when calling one proxy from another. This local connection is more efficient because it bypasses network features such as load balancers, routers, and message processors.

1. How do you manage API visibility and restrict access to consumers? Is this configuration in the platform or built as part of the APIs enablement? Ans. This is managed by the Apigee Edge's API Management solution.  It can protect the internal or external APIs using the various security policies provided by Apigee Edge: VerifyApikey, AccessControl, OAuthV2/VerifyAccessToken etc. (As mentioned in the answer to question L)
2. Does your solution use any framework and how it can help us? Ans. APIGEE Framework provides step by step tutorials on how to implement its API management solution
3. Please describe the different options for developing APIs? Ans. Question is too generic.

## API Gateway

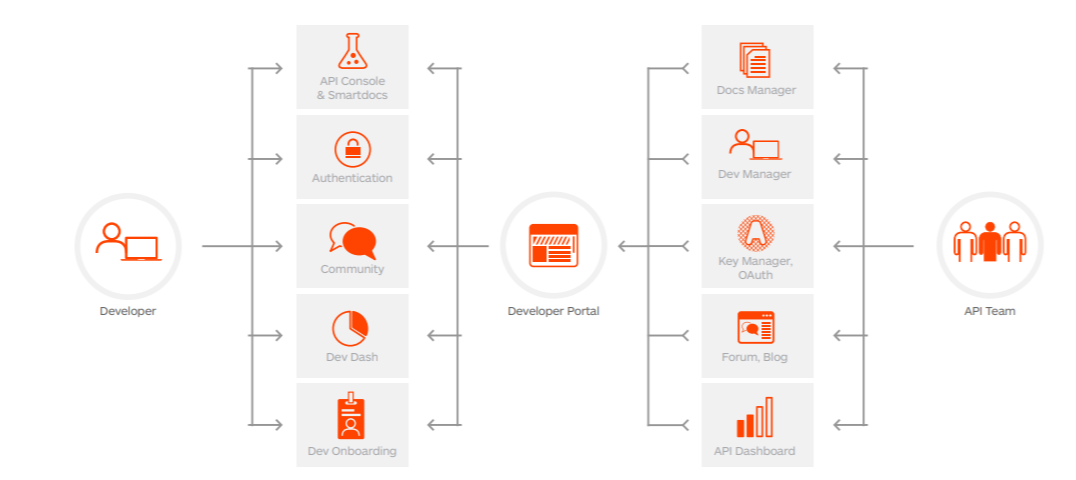
1. Can your API Gateway execute customer's Java / C# code / JavaScript code? Please explain Ans. API Gateway creates a proxy that acts as a façade to the underlying service provider. The proxy can transparently convert a request from a SOAP to REST service and vice versa. The underlying service can be implemented in any language and is inconsequential.
2. How does your product support threat detection by detecting fraudulent data injections at the API level? Ans. This has already been covered. See answer e from section 8.3
3. Does the platform support publishing SOAP, REST, JSON, and XML style services as APIs as well as JMS? Ans. Question answered F & G from 8.3
4. Is your API Gateway UI browser-based or does it require a fat client to be installed on a developer desktop/laptop? Ans. It is a thin client. It can be accessed via a browser.
5. What other types of workload does your API Gateway support beside SOAP and REST? Ans. See F from 8.3
6. Does your API Gateway have any built-in capabilities for doing self-balancing across a cluster of gateways and intelligent load balancing to backend API provider layer? Ans. Apigee Edge enhances the availability of the API by providing built-in support for load balancing and failover across multiple backend server instances. TargetServer configurations decouple concrete endpoint URLs from TargetEndpoint configurations. Each TargetServer is referenced by name in a TargetEndpoint HTTPConnection. For more details see URL <https://docs.apigee.com/api-platform/deploy/load-balancing-across-backend-servers>
7. What security protocols and standards does your API Gateway support? Ans. Supports TLS & SSL
8. Does your API Gateway provide built-in support for schema validation? Ans. APIGEE’s schema validation policies include SOAP against WSDL, JSON, XML against XSD
9. Does your API Gateway support JSON to XML, XML to JSON and Any2Any message transformation without any coding or extension?

Ans. It supports transformation of well-known messages with minimal intervention. You have to specify the source and target of the transformation

1. Does your API Gateway provide database connectivity? Ans. Yes it does. This has been covered in earlier sections
2. What level of error handling is available on the gateway? Ans. This is determined by the error policy set up with in the gateway. The policies can as simple or as complicated as it needs to be. This is covered in-depth in their documentation
3. Is your gateway able to provide different activities or policies for each individual API and operation on that API, i.e. POST, GET, etc.? Ans. Yes
4. Does your gateway support POST, GET, DELETE, HEAD, PATCH & OPTIONS? Ans. Yes, Following RESTful principles, you can call HTTP GET, POST, PUT, and DELETE methods on any of the API resources.
5. Can your gateway control, manage and shape API traffic? Please describe the policies that can be applied Ans. Apigee Edge provides three mechanisms that enable you to optimize traffic management to minimize latency for apps while maintaining the health of backend services.  These are Spike Arrest, Quota and Concurrent Rate Limiting. Details can be obtained from their documentation
6. Can multiple provider channels be exposed from the same gateway with complete segregation? I.e. internal APIs and external APIs running through the same gateway in isolation? Ans. Yes this has already been covered earlier
7. What REST API Description Languages does your solution support and what REST API Description Language do you support internally in your runtime? Ans. It supports OpenAPI specifications, Javascript, Java or its inbuilt UI
8. Please describe your support for Open API, i.e. Swagger. Does your corporation belong to the "Open APIs Initiative", i.e. www.openapis.org? Ans. No we do not belong to the OPENAPIS initiative.

1. Does your API Hub solution have any out-of-the-box (OOTB) integration points to automatically manage APIs from other products? Ans, The UI connects to most well-known service endpoints. It has an Integrated portal which can be instantly provisioned as well as a Drupal portal
2. Does your API Hub solution support built-in browser-based visual/graphical message mapping? Ans. Yes to the best of our knowledge.
3. Does your API Hub solution support an external IDE to do visual/graphical message mapping that you can then deploy to the runtime environment? Ans. You can also do the proxy development using XML & Your favorite IDE on your local machine & integrate into CI & CD pipelines using Apigee tools like Maven Plugins , Jenkins, Automated Testing using testing frameworks integrated into CI & CD pipeline.
4. Please describe the API Management lifecycle, versioning, governance, & control. Does your platform support an API lifecycle beyond API naming convention? Please explain

Ans. Apigee Edge includes three components: API Services, Developer Services, and Analytics Services. The core API management capability is API lifecycle management. Managing the API lifecycle involves consideration for both the API provider and the API consumer (most often the app developer). API providers manage the processes for designing, developing, publishing, deploying, versioning, governance, monitoring availability, and measuring performance. API consumers discover new APIs, understand versioning and API updates, easily register for access to APIs, test and register apps built against the APIs, and communicate and collaborate with other developers and the API provider. API management enables API developers, who expose assets via APIs, to unlock the value of business assets by rapidly creating APIs from existing data and services. API management provides the ability to design and build APIs that are intuitive and easy for developers to adopt and use. As part of its life cycle, APIGEE allows common protocol transformations, versioning and configuring policies for security. Moreover it supports python, JavaScript, java, NodeJS etc. Its developer portal enables an API provider to deliver an enhanced developer and community experience that accelerates API adoption, simplifies learning, and increases the business value of APIs. It enable developers to register their applications, select the APIs and the service levels they need, get secure access, monitor their API usage. It even generates documentation that can shared across social media



1. Does your API Hub solution include API analytics? If so, please describe what API metrics are captured for analytics.

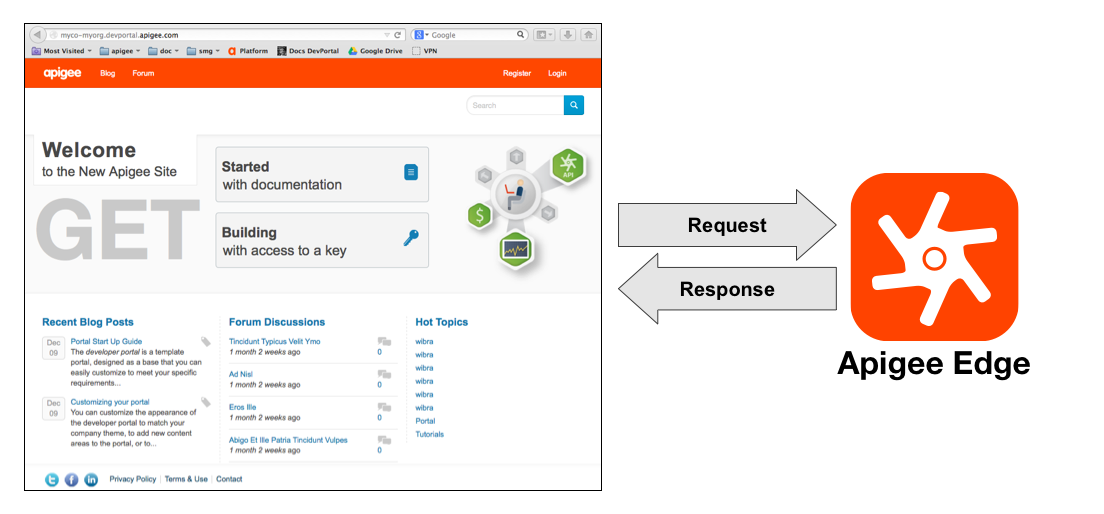
Ans. Apigee Edge enables business and operational metrics to provide a complete 360-degree view of your business. The platform goes beyond operational and developer level metrics to provide visibility to the business. Traffic composition reports provide insights into the most valuable entities of an API program: the apps, developers, APIs, and resources. Enterprises use the reports to detect business problems such as lower traffic trends or diminishing contribution from key apps and developers. They can get early notification of new entities that contribute to API traffic, and take actions to respond. For example, you can determine which developers are contributing most and include them in nurturing programs. Operation visibility is provided out of the box—across all APIs, all API traffic, top API movers, top API products, top ranked apps, top ranked developers, anomaly inspection, and trend analysis. For individual APIs, enterprises can measure traffic, average response times, average target response time, maximum response time, error rate, and average data exchange. Edge Analytics Services provides several visualization tools, including the dashboard (which gives an overall view of your entire API program), custom reports (to select, combine, filter, and drill down into specific API metrics), GeoMap (which tracks traffic patterns, error patterns, and quality of service across geographies), and tools that allow you to plot trends in traffic, response time, and other metrics for an API’s individual resources

1. Please describe the process for discovering services and APIs that are to be published on your solution Ans. Using prebuilt API consoles to explore API’s
2. Can APIs be extended from their original endpoints to include new functionality or data? Ans. Yes
3. Please describe how different environments such as Dev, Test, Staging, and Production can be deployed keep APIs in isolation from each other.

Ans. Multi-tenancy allows for separation of internal development teams and lines of business through the concept of organizations. Within an organization, RBAC further segments users and their privileges. Within an organization, Edge supports multiple environments, which an enterprise can use to mirror internal product lifecycles such as development, testing, staging, and production environments.

1. Please describe how your API manager communicates with the other components of your solution, such as the API Gateway, Developer Portal, etc.

Ans. The Developer Services portal acts as a client for Apigee Edge. That means the portal does not function as a stand-alone system. Instead, much of the information used by the portal is actually stored on Edge. When necessary, the portal makes an HTTP or HTTPS request to retrieve information from Edge or to send information to Edge. Edge does not make requests to the portal, it only responds to requests made from the portal. Therefore, all interactions between the portal and Edge are initiated by the portal.

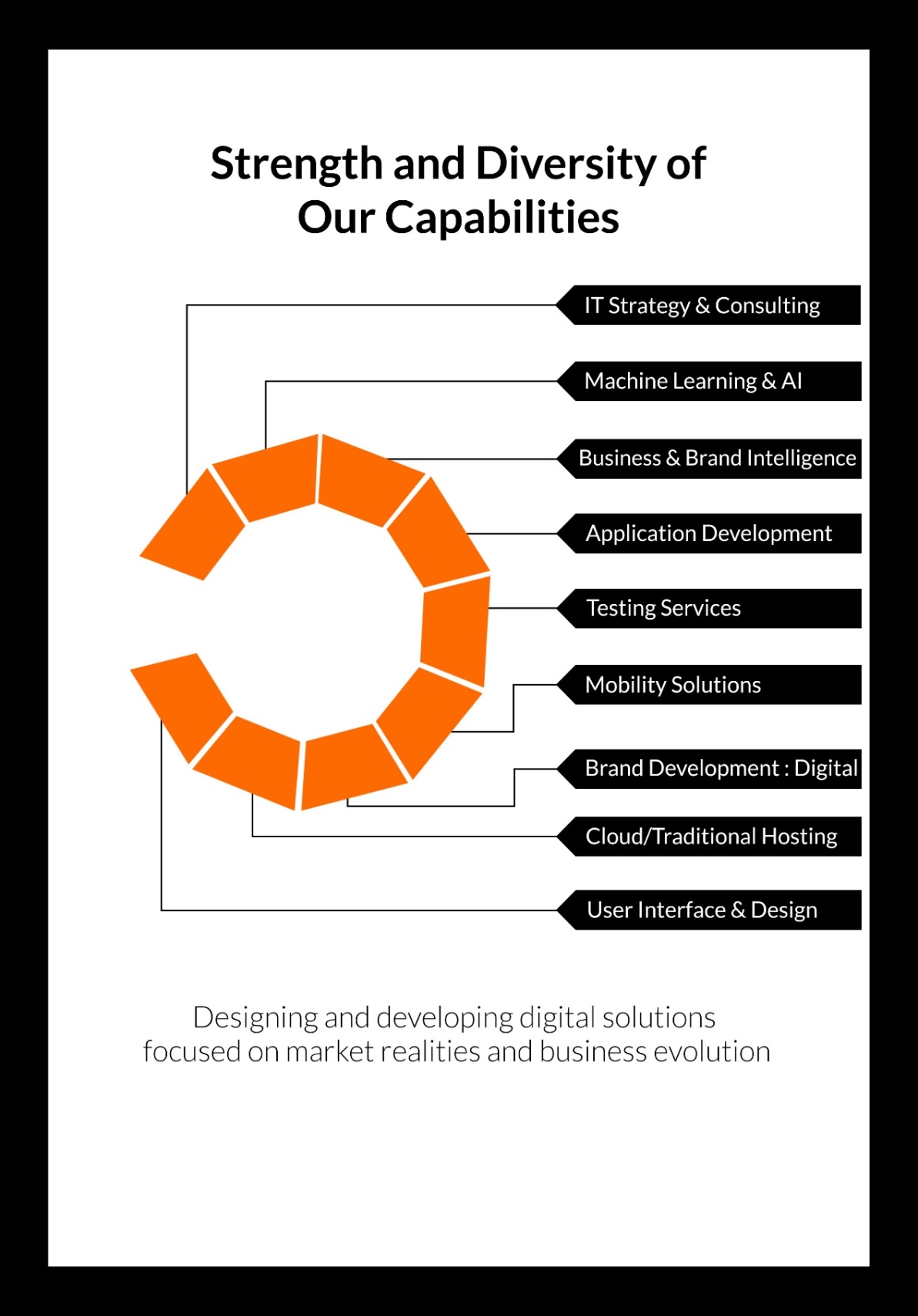


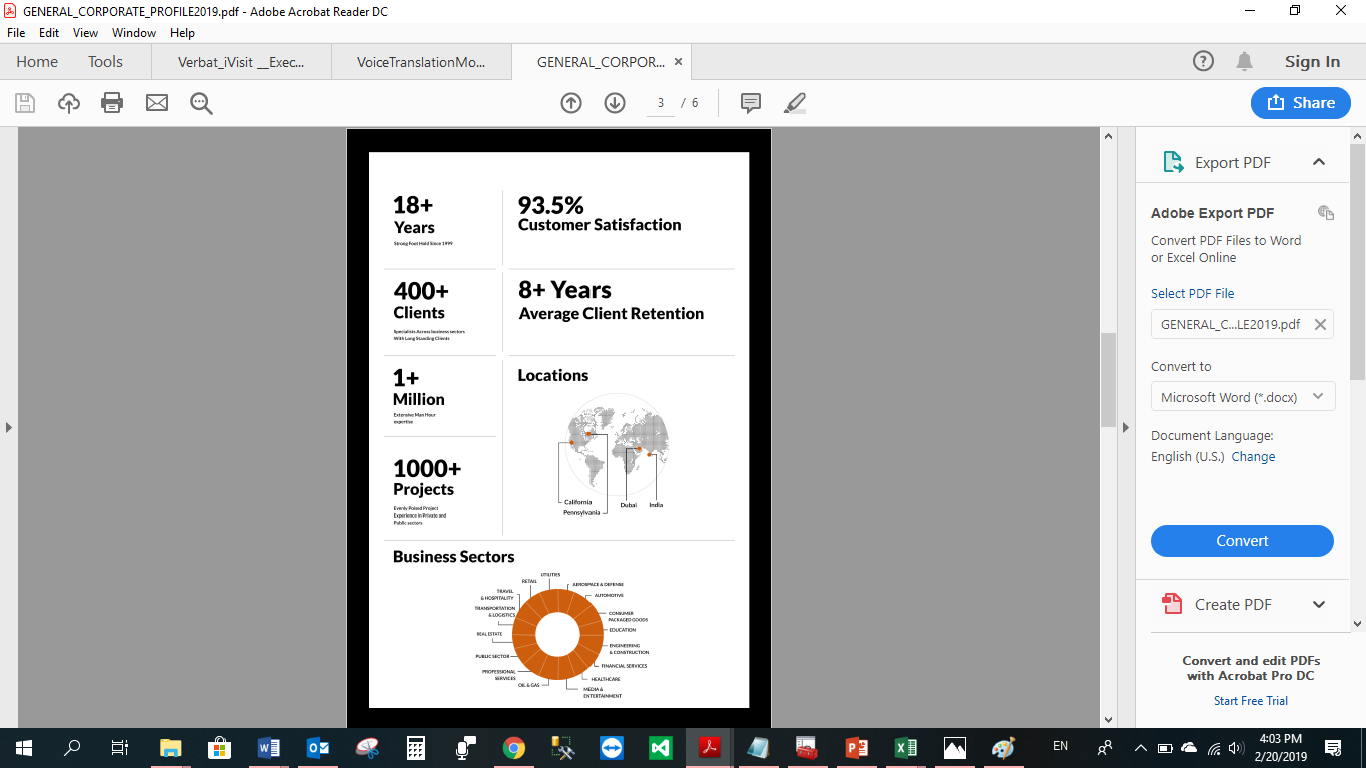
1. Please describe the API build chain and how this can be automated Ans. Cannot be explained as it is beyond the scope of this document
2. Please describe how templates from other API definitions can be used to create new APIs sharing common functions and operations Ans. This is beyond the scope of this document
3. Please describe how APIs are version controlled at development time. Ans. Versioning is supported at multiple levels. Backend service versions can be “hidden” behind the API facade. Versioning can be applied at the URI level, following best practices and internal corporate standards. Additionally, all artifacts (policies and configurations) are stored in XML and can be placed into versioning systems.
4. Can your offering enforce approvals on users when moving through the API lifecycle? Ans. Yes
5. Can the same APIs be deployed to both internal users and external users but behave differently based on context? Ans. As explained earlier, this is possible
6. Can APIs be grouped in a way that allows them to be consumable by specific audiences? Ans. Yes
7. Please describe the out-of-the-box reports provided by the tool. Ans. Please see answer V in this section
8. Does the tool provide a wizard for creating custom reports? Ans. ostensibly yes.
9. Can reports be created on-demand? Ans. Yes
10. What level of reporting is available to the developer? (call latency, SLA compliance, other metrics). Ans. A wide range of reports are available to the developer.
11. How are assets manifested in the developer portal for developer use? Sub/Department Page. Ans. This has already been covered in a previous question. Hint: Organizations
12. Please describe how the tool facilitates on boarding. Is this portal available as a completely on-premises solution? Ans. Yes. Onboarding is done using the developer portal.
13. Does the solution provide interactive documentation to allow API consumers to easily try out published APIs? Ans. Absolutely yes
14. Does your Developer Portal leverage any Content Management System? Ans. Yes, the developer portal can be created using Drupal
15. Does your Developer Portal provide sample code to invoke an API from different platforms, e.g. Swift, Java, Node, Python, Ruby, PHP, cURL, Go,C#,Javascript etc. that can be easily copy-and-pasted? Ans. We believe this is possible and has been covered in a previous question
16. Does your Developer Portal include capabilities to create Support tickets and access FAQs? Ans. Yes. Support tickets can be created using extensions
17. Please describe what information is available to subscribing developers when they visit the Developer Portal. Ans. Not sure
18. Does each developer or developer organization get usage information for the APIs they have subscribed to? Ans. Yes they do
19. Can the visibility of specific APIs be controlled by who has signed up to the Developer Portal? Ans. Not sure if we understand the question
20. Does your Developer Portal have OAuth testing tools, which enable the complete testing of APIs that are secured with all of the OAuth flows? Ans. Yes OAuth can be tested.

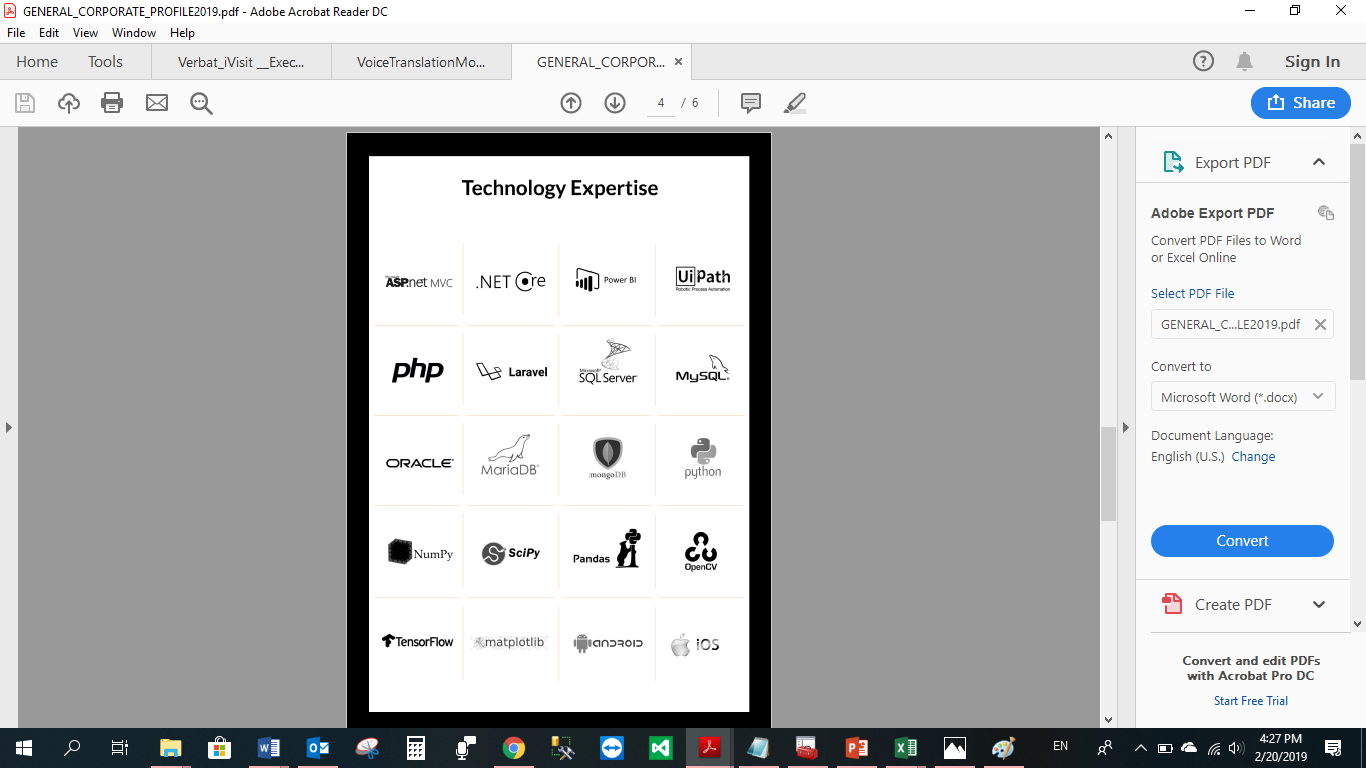
## API Security

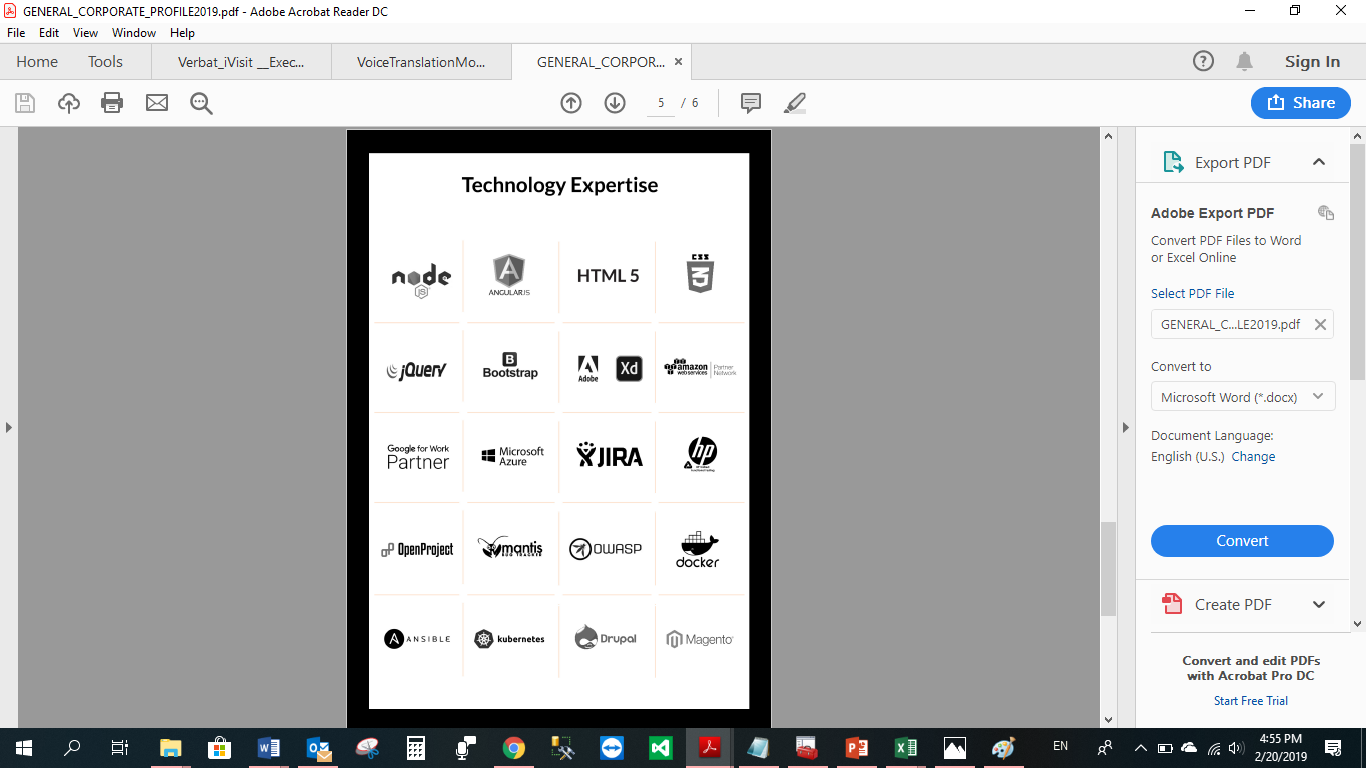
1. Does your API Hub solution include security capabilities to manage Users, Roles, TLS Profiles, and User Registries? Please explain The upgraded software should cover the all default features. Ans. Yes, this has already been covered.
2. Does your Developer Portal have new user creation self-service capabiliites? Ans. Yes
3. Does your Developer Portal support CAPTCHA for self-service new user creation as well as API Comments? Ans. Not sure
4. Is your API Gateway DMZ-ready out-of-the-box? Ans. Not sure
5. Does your API Gateway support FIPS 140-2 Level 3 and Common Criteria EAL4? Ans. Not sure
6. Please describe how your solution would address the OWASP Top 10 threats. Ans. The solution has multiple threat evaluation capabilities that are fairly sophisticated. Further details can be provided at a later stage
7. How does your solution secure payloads and channels? Ans. We do not understand the nature of this question.
8. Please describe how policies are created. Ans. This is out of the scope of this document. This can be explained at a later stage of the evaluation of the proposal.
9. Please describe how the interactions between developers and the published APIs are secured. Ans. This has been covered earlier
10. Does your solution supply any OAuth testing tools? Ans. It’s already baked into the platform.
11. How is single-sign on supported across all the roles involved in the lifecycle in your product? Ans. Outside the scope of this document
12. Please provide a link to user communities for your solution. Ans. <https://community.apigee.com/index.html>
13. Please provide a link to the public documentation for your solution. Ans. <https://docs.apigee.com/>
14. Please provide public links to reference architecture that is relevant to your solution. Ans. <https://docs.apigee.com/api-platform/reference/apigee-reference>
15. Please provide links to any freely available tutorials for your solution Ans. <https://docs.apigee.com/api-platform/get-started/get-started>
16. Please provide details of the free training that your organization provides for your solution. Ans. We only provide training after deploying the solution and after the client has signed up for training.
17. Does the solution provide interactive documentation to allow API consumers to easily try out published APIs?. Ans. Yes
18. Please provide details on the professional services your solution in order to implement an API Hub solution
19. Please describe your company's vision for API usage in our ministry. Ans.
20. Please describe your company's experience in supporting different government outside the API management domain. Ans.

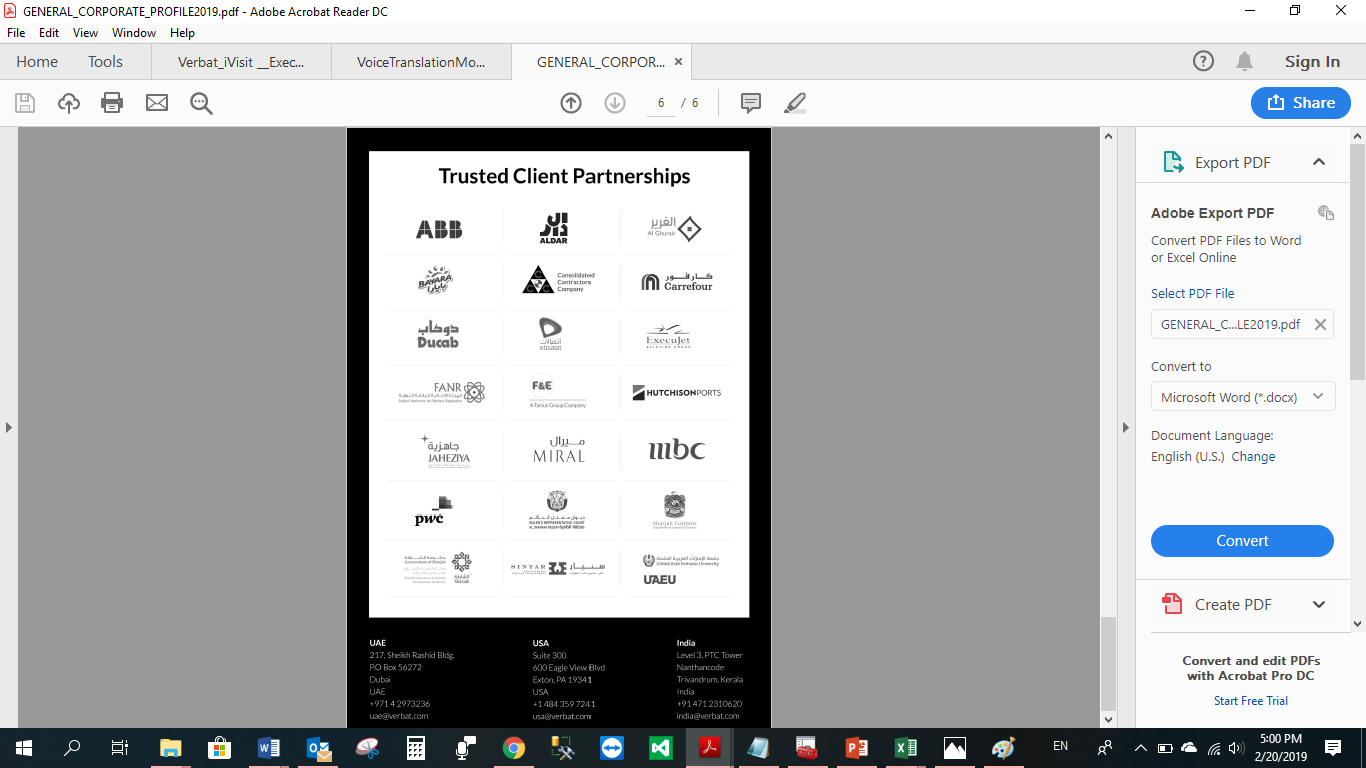












**WITH OUR**

**SINCERE THANKS.**

CONFIDENTIALITY NOTICE:

This Proposal is conﬁdential and contains proprietary information and intellectual property of Verbanet Technologies LLC. Neither this proposal nor any of the information contained herein may be reproduced or disclosed under any circumstances without the express written permission of Verbanet Technologies L.L.C (hereby known under its brand name of Verbat in the United Arab Emirates). 2019. All Rights Reserved.