HARVARD UNIVERSITY

Enterprise Architecture
Standard

API Principles and Practices
Use of the API Gateway and Portal

Authors: Greg Charest	**Audience Level:** • Strategy Planning and EA Leader • Solution Architect and Program Manager
Version: 0.4 **Last Revised:** 04Mar2020 **Status:** draft **Document Type:** Single Topic Guidance	**Distribution Scope:** HUIT-wide
Workgroup Members:	**Reviewers:** Raoul Sevier Mitch Rogers

1. Purpose of this Document

The purpose of this document is to describe the HUIT standard for including existing and future REST APIs in the API gateway and associated documentation portal (collectively called the "API Platform"). The document also provides high-level principles and best practices that can be used to guide more detailed decisions related to HUIT's API program.

Note: this document does not provide guidance on when to use APIs, vs. some other data exchange mechanism. For help determining the optimal data exchange mechanism for your use case, please refer to the HUIT Enterprise Architecture advisory on Data Exchange Mechanisms and Considerations.

2. Executive Summary and Recommendations

APIs make it possible to more effectively support the flow of information and operations within and across organizational boundaries, but new architectures and processes are needed to manage the exchange of these encapsulated data assets. Many, but not all, APIs used within Harvard should be exposed through and consumed via the API Platform. Categorizing APIs and applying core API design principles is a useful way to determine which APIs should leverage the API Platform, and how.

API Design Principles

These following principles should be incorporated into the architecture and design of REST APIs deployed in the HUIT API platform. APIs should be:

- **Discoverable**
- **Reusable**
- **Modular**
- **De-coupled**
- **Governed**

Categorizing APIs

Categorizing REST APIs and their recommended treatment with respect to the API gateway and portal is a useful way of determining which components of the API Platform to use. The following table summarizes general API categories with respect to the API gateway and portal.

API Category	Include in Portal	Include in Gateway
Managed vendor API	✓	✗
Non-managed vendor API	✓	✓
Internal general use API	✓	✓

Internal application specific API	✗	✗
Internal limited use API with specific requirements	✗	✓

Recommendation

Development teams should use these API Design Principles and API Categorizations to determine which APIs must use the API Platform.

3. API Architecture and Design Principles

Through its API program, HUIT is committed to building a lasting culture of reuse to inform future project planning, reduce costs and improve the outcomes of IT efforts.

Decisions related to the architecture, design, use, and management of REST APIs within Harvard should be guided by the following principles. APIs should be:

Discoverable
APIs should be easily discoverable. If users cannot find information, it does not exist for them.

Reusable
APIs should be designed for re-use. Reusable components reduce development costs and drive down time-to-market for the delivery of new software features.

Modular
APIs should be independent and execute one function. Organizing APIs into system, process, and experience components supports reuse and innovation.

De-coupled
APIs should de-couple source from target, reducing dependence on specific technologies and thus reducing the cost of change.

Governed
APIs must be able to support HUIT technology standards for security and compliance, access management, logging, monitoring, etc.

Not every API design will exhibit all of these characteristics. For example, a development team may design and implement an API that is purpose-built for specific implementation. This API may not need to be discoverable. However, the same API may need to include standard logging and monitoring capabilities. Some API designs may exhibit all of these characteristics, some may exhibit none.

Use these API architecture and design principles to think about which characteristics your APIs should have.

4. API Platform Components

4.1. API Portal

An *API portal*[1] supports developer on-boarding, provides a central source for API documentation, and make APIs discoverable. HUIT has implemented the Apigee API Management Platform which provides both gateway and portal service but has left the choice of technology stack for API implementation to local development/integration teams.

4.2. API Gateway

The most effective architectural design for coordinating and controlling internal API based data flows is the *API Gateway* pattern.[2] An API gateway service acts as a single point of entry, abstracts complexity, and centralizes authentication, monitoring, and rate limiting policies.

5. Categories and Treatment of APIs

The set of REST APIs important to Harvard include APIs developed internally as well as APIs developed by vendors and external partners which are accessible to Harvard applications.

5.1. Vendor APIs that are well managed by external partners

Many APIs that provide significant value to Harvard are developed and maintained by non-Harvard companies and organizations, typically SaaS vendors. The vendor determines the scope of the API offering and manages access, monitoring, and provisioning processes. In general, there is little additional value to wrapping these external vendor APIs in order to include them in the Harvard API gateway. In addition, updating the Harvard API gateway entry as the vendor evolves their product is burdensome.

There is, however, significant value in having descriptive information, and associated reference links, in the Harvard API portal that support the discovery of available resources.

Consider for Gateway: No
Consider for Portal: Yes

5.2. Vendor APIs that are internally managed

A subset of vendor provided applications are customized specifically for Harvard and/or are single tenant applications hosted by the vendor. APIs associated with these applications are created by the vendor but are not managed on our behalf. Effective use of these APIs is enhanced by the authentication, monitoring, and rate limiting policies provided by the Harvard API gateway. Ongoing

[1] The term 'portal' is used here to describe a central location to connect API providers and consumers. It may include tools to on-board consumers, provide API documentation, and support creation of an API community.

[2] https://www.martinfowler.com/eaaCatalog/gateway.html

API maintenance requirements can be included in the processes already in place to manage and update application configuration and customization.

Similar to pure SaaS application APIs, there is value in including descriptive and reference information in the API portal in order to support discoverability and further the goal of re-use.

Consider for Gateway: Yes
Consider for Portal: Yes

5.3. Internal APIs that have value across diverse applications

Many Harvard software development efforts include the creation of APIs. Although there is presently no central team dedicated to the development of general-purpose APIs, various teams have created APIs intended to be used by multiple groups. Good examples are the Person and Chart of Accounts APIs which were designed to support diverse application needs. In addition, APIs developed for specific applications are often useful to other projects.

In order to comply with Harvard security standards and to meet specific data governance requirements, the majority of internally developed APIs should be deployed using the API gateway.

Following the principles of discovery and re-use, it is especially important that shared internal APIs be included in the portal. Particular care should be taken to ensure that this group of APIs comply with the API platform documentation standards.

Consider for Gateway: Yes
Consider for Portal: Yes

5.4. Internal APIs internal to a specific application

Applications often use APIs for internal communication and/or data transfer. Often these internal communication channels are not intended to be externally exposed and must meet specific non-functional performance requirements. For example, an application design based on a micro service architecture may include various services that communicate via REST APIs. APIs in this category need not be included in the API gateway.

Because these APIs are not intended to be used outside of a specific application environment, they are also not required to be in the API portal.

It is important to note however, that teams developing APIs in this category are responsible for meeting applicable security and monitoring requirements.

Consider for Gateway: No
Consider for Portal: No

5.5. Internal APIs with particular policy requirements

Certain internal and non-managed external APIs not intended to be widely used may still benefit from the policy enforcement services provided by the API gateway. For example, an API with stringent

throttling and monitoring requirements might take advantage of the relevant API gateway services rather than build these capabilities independently.

Because this category of API is not intended to be shared, it is not necessary to include it in the API portal.

Consider for Gateway: Yes
Consider for Portal: No

6. Applying design principles when determining which APIs should use the API Platform

A large application, for example PeopleSoft, provides hundreds of "Non-managed vendor APIs". Based on the API Categorization above, these APIs should be considered for the API Portal and the API Gateway.

But which ones? While certainly not all of the hundreds of APIs will be included in the API Platform, there are very likely some that should be, based on answering the following questions:

1. Does this API need to be ***discoverable,*** so that someone who is looking for data can find it and understand what data it provides and how to get access to it?
2. Is this an API that is likely to be ***reused*** (or is it already being used by more than one API consumer)?
3. Is this API ***modular***, or are there benefits for making it more modular?
4. Does this API belong to a system that may undergo frequent change, or someday be replaced, or is it broadly used? In other words, is there a need to ***decouple*** the source and target(s) of this API?
5. Does this API need to implement HUIT standards for security and compliance and/or logging and/or monitoring, or implement specific access controls and provisioning features? In other words, does it need to be ***governed***?

If the answer to any of these questions is clearly "yes", then the API should leverage the API Platform capabilities.

7. Summary

7.1. Final Considerations

The vision of the Harvard Common API Platform is to promote the creation and use of re-usable assets. Behind this vision is a desire to simplify the complex web of point-to-point data exchange processes that currently exist with the Harvard environment. In addition, a primary HUIT value is the support of innovation. In light of these fundamental goals, developers should favor making APIs as easy to discover and share as possible. A decision to not include an API in the Apigee environment should be made carefully and with a clear understanding of the relevant concerns.