

Enterprise Architecture

HUIT Standard | Advisory | Guidance

Using Oracle APEX

Authors: <ul style="list-style-type: none">Irene Ogolla	Audience Level: <ul style="list-style-type: none">IT Director / ManagerSolution Architect and Project ManagerApplication Developer and DesignerDevOps Staff
Version: 1.0 Last Revised: 3/25/2022 11:02:00 AM (auto updated) Status: Released Document Type: Single Topic Guidance	Distribution Scope: <ul style="list-style-type: none">HUIT
Workgroup Members:	Reviewers: <ul style="list-style-type: none">Raoul SevierJefferson BursonBill BrickmanHellen ZziwaAlan RintoulSteven TsekiMarilyn RackleyPaige DuncanLauren SzufatMahbub Rahman

Document Change Log

Revision	Date	Editor	Changes
0.01	02/04/2022	Irene Ogolla	Initial Draft
0.02	02/14/2022	Irene Ogolla	Added best practices on workspaces and made modifications based on initial reviews.
0.03	02/18/2022	Irene Ogolla	Added summaries of use cases and the scenarios where APEX is not appropriate to the executive summary. Also added section to encourage API use to the development best practices.
0.04	03/03/2022	Irene Ogolla	Made modifications to sections 4.2 and 6.8 for clarity.
1.0	03/22/2022	Irene Ogolla	Edits to application descriptions and sections 2, 4.2,5.2, and intro to 8. Final version 1.0 for publishing.

Table of Contents

1	<i>Problem Statement</i>	5
2	<i>Executive Summary</i>	5
3	<i>Intended Audience</i>	6
4	<i>Summary & Recommendations</i>	6
4.1	Strategic Vision	6
4.2	Analysis of Current Architecture and Team	7
4.3	Cost, Risks and Benefits Analysis	8
5	<i>Potential Use Cases</i>	8
5.1	Reporting & Workflow Applications	9
5.2	Authorization Applications	9
5.3	Application Extensions.....	9
5.4	Custom Business Applications	9
5.5	Test Prototypes Without Significant Investment.....	9
5.6	Empower Tech-Savvy Users.....	9
6	<i>Best Practices for Development</i>	10
6.1	Create the Oracle Data model.....	10
6.2	Develop APEX Coding Practices	10
6.3	Build to be secure	10
6.4	Create templates for applications	11
6.5	Logging and debugging.....	11
6.6	Build for performance and scale	11
6.7	Create reusable components	11
6.8	Workspaces	11
6.9	Documentation.....	12
6.10	Build APIs.....	12
6.11	Inventory your applications	12
7	<i>Best Practices for APEX Deployment</i>	13
7.1	Use runtime only mode for production environments.....	13
7.2	Deployment Standards	13
7.3	Shared Infrastructure	13
7.4	Consider using RDS.....	13
7.5	Lower the barrier for entry for non-prod environments.....	13
7.6	Create best practices and templates for variable URLs	13
7.7	Acquire a common license for AOP plugin	14
7.8	Production Support Organizations	14
8	<i>Scenarios Where APEX is NOT a Good Fit</i>	14
8.1	Native applications	14
8.2	Client/Server Applications	14
8.3	Offline Applications	14
8.4	Single Page Applications	14
8.5	Mobile Applications	15
8.6	Version Control Requirements	15
8.7	Other Scenarios	15
9	<i>Additional Considerations</i>	15
9.1	Other Platforms	15

9.2	Other advisories	16
10	References.....	16
11	Appendix – Examples of APEX applications at Harvard	17
	Table 1: Custom Business Applications	17
	Table 2: Reporting and Workflow Applications	18
	Table 3 : Application Extensions	19
	Table 4: Authorization Applications	20

1 Problem Statement

Over the past several years there has been a growing use of Oracle APEX as a development platform for creating business applications within Harvard University. This has resulted in close to 140 APEX applications introduced into the environment with varying standards of build and deployment.

This advisory provides guidance to the product development communities in determining when Oracle APEX is fit for purpose and offer some best practices to follow in the creation of the resulting applications.

2 Executive Summary

There is an increasing business demand for web applications which can be easily accessed and can handle substantial amounts of data. There is also a growing demand for faster development including faster prototyping. Oracle APEX provides a platform that meets these demands.

Oracle APEX is a low-code, web application development framework that runs directly out of the Oracle Database and uses Oracle REST data services (ORDS) deployed in a WebLogic server for data access. The application definitions are stored in the database as meta data. ORDS, a mid-tier Java application, provides a database management REST API and the ability to publish RESTful web services for interacting with the data and stored procedures in an Oracle database.

Oracle APEX is included by default with all editions of the Oracle database. It can be used to build a wide variety of applications from the simplest application that converts a spreadsheet to a web page, to mission-critical applications which are used daily by tens of thousands of users.

This advisory provides some guidelines for the product and portfolio teams to enable them to determine whether or not Oracle APEX is an appropriate tool for a specific custom build.

Oracle APEX is most appropriate for the creation of applications that are data driven and running on Oracle databases. Increasingly, with the use of REST APIs, the scope of use is being extended beyond Oracle databases. However, the following use cases have been identified as potentially good candidates for APEX at Harvard.

- Reporting and workflow applications
- Authorization applications
- Application extensions
- Custom business applications
- Testing prototypes
- Empowering tech-savvy users

The advisory also includes several scenarios where APEX is likely not a good fit for your application like

- Native applications
- Client-server applications
- Applications needing to be used offline
- Single page applications
- Mobile applications.

It is recommended that the teams review their current environment architecture and portfolios for use of Oracle and other development platforms, and their organizations for the skills required to create and maintain an APEX application. They should then consider the costs, benefits, risks, and complexity of introducing a specific Oracle APEX application into their organization and portfolio. They should also consider how well a custom build aligns with the strategic direction being taken by HUIT or a specific school.

Finally, this advisory provides some best practices for the development and DevOps teams for both the creation and deployment of APEX applications.

3 Intended Audience

This document is intended for IT Managers, product & portfolio owners, APEX developers and DevOps teams who are creating business applications to aid in their determination of whether Oracle APEX is the appropriate platform for a particular use case, and once developed, in the deployment of the application.

4 Summary & Recommendations

The assumption made in these recommendations is that there is no competitive product on the market that is available to the team, and a decision has already been made to build a custom application.

There are four areas of consideration in determining whether Oracle APEX is an appropriate for a custom build:

- 1 the strategic vision of HUIT or a specific school as regards the technologies to be deployed,
- 2 the use case before the product or development team,
- 3 the team's current environment in terms of skills and other technology in use, and
- 4 the costs, risks and benefits of the custom solution being considered.

The use cases are discussed in section 5 of this advisory.

4.1 Strategic Vision

In determining whether Oracle APEX is the best fit for a custom build, consider the strategic direction of HUIT or a specific school and how Oracle and specifically APEX aligns with that vision.

4.2 Analysis of Current Architecture and Team

In analyzing your environment and team capabilities, ask the following questions

- 1 Are there Oracle databases already in use in the environment? Having Oracle databases already in use indicates that a team has the licensing necessary to use APEX and ORDS. It is also likely then that they have the skills necessary to install and maintain the required environments.
- 2 Do you already have other Oracle APEX applications in the environment?
If a team already has other APEX applications in their portfolio, then it is likely that they will already have both the setup, and skills and experience required to create and maintain a new one. A team that does not would have to invest a bit more time and money.
- 3 What other development platforms are currently being used?
The development platforms in use are an indicator of the skills that exists within a team. If Oracle APEX is not part of the team's skills set, then there would be a significant investment of time and money to acquire those skills.
- 4 How many other development platforms does the portfolio have?
The more development platforms a team has in their application portfolio, the more complex the maintenance. It is recommended to keep that number low.
- 5 What are the data sources that will serve the custom application? Are they in Oracle databases?
These questions give the teams an opportunity to consider the integrations and data exchange mechanisms that will be required.
- 6 Does the team have Oracle DBA skills? What is the level of skill?
The teams should have access to Oracle database administrators particularly experience with data modeling, performance tuning and data protection will be necessary. This expertise does not have to exist within the development team. It can be part of say a DevOps team or if the environments are part of a hosted environment, then they could be part of the services offered.
- 7 Does the team have APEX developer skills? What is the level of skill?
Oracle APEX developers should be proficient in several of the following areas:
 - Oracle Application Express
 - Data modeling
 - Oracle SQL and PL/SQL
 - Oracle database performance tuning

- Data and Web security
- HTML5
- CSS
- Web design
- JavaScript
- RESTful web services

In summary, the use of APEX to create applications is more appropriate when there is existing APEX infrastructure and expertise. The expertise should include Oracle database administration and APEX development. As noted above, the expertise to stand up APEX infrastructure can be part of a DevOps team or if the environments are hosted then part of the services offered.

Note that this does not exclude teams just starting out with no infrastructure or the required expertise. These teams might incur higher initial costs and have greater risks using APEX for an application build. The team would need to ensure that considering all the factors discussed, it still makes sense to proceed using Oracle APEX.

In addition, the data sources that will serve your application should factor in your decision. The more of your data sources that exist in Oracle databases, the more appropriate it would be to use APEX to create a new custom application. However, an APEX application can still interact with data that is not in an Oracle database using RESTful APIs.

Finally consider how many other development platforms your team will have to work with. Keeping that number reasonably low will reduce complexity in your environment.

4.3 Cost, Risks and Benefits Analysis

The costs, complexities, risks, and benefits analysis will give the team the opportunity to weigh the following:

- 1 the costs, including development time and training costs, to build and maintain a custom Oracle APEX application
- 2 the complexities that a new custom APEX application will add to an environment
- 3 the risks introduced by adding a new application, and
- 4 the business benefits that will be derived from a new application.

The higher the costs, complexities, and risks introduced for few or limited business benefits, the less appropriate Oracle APEX for a specific scenario.

5 Potential Use Cases

Below are some examples of use cases where Oracle APEX has been or could be used to create business applications in Harvard.

5.1 Reporting & Workflow Applications

These are applications that replace spreadsheets, PDF forms, and email review and approval processes. They often provide organization-specific dashboards, improved workflows, or fill reporting gaps

They also can provide interactive reporting and data visualization applications based on disparate data from across multiple systems.

5.2 Authorization Applications

This category of applications provides application specific solutions for fine grain access control for other APEX and non-APEX applications. An example of this is Alseta.

The applications define user-organization unit, user-object, and user-organization unit-object relationships. They are used to define and manage the ownership and allowed operations and actions.

5.3 Application Extensions

Applications that extend ERPs and other enterprise software to avoid customizations within a vendor application or when a business application like EBS, GMAS, Peoplesoft or Center Stone have limitations in terms of how the data can be viewed or number of licenses in the case of Center Stone.

5.4 Custom Business Applications

These are applications that fill business needs not covered by any of the existing enterprise applications or new business needs triggered by some unforeseen change like Covid-19.

Often there is a need to fill the business need in days or weeks and not months or years. The development teams can easily modify to meet changing requirements and rapidly iterate to a production ready application. A key aspect of this use case is that often applications that are needed urgently and developed quickly end up having an operational life measure in years.

5.5 Test Prototypes Without Significant Investment

When your organization has an idea for an enterprise application, you can quickly create and test a prototype to determine its viability.

5.6 Empower Tech-Savvy Users

Your organization may have many tech-savvy workers that would benefit from being able to put together highly customized enterprise applications to support their work. These users can get what they need without taking up the development team's time for most of the process. The use of APEX in this case helps to impose some standards and consistency in development.

6 Best Practices for Development

The goal of the best practices below is to ensure that good and consistent practices are followed in the development of APEX applications. These best practices should be considered when designing an application and managing the development process.

6.1 Create the Oracle Data model

A data model will provide the structure, definition, and format of the data. It is the foundation for any applications and a good application requires a good data model. It is essential not simply for storing data and implementing data related business rules but also for application performance.

6.2 Develop APEX Coding Practices

Institute some common practices that a development team adheres to when creating APEX applications. Here are some recommended practices:

- 1 Keep all code in packages and avoid stand-alone procedures and functions.
- 2 Limit the code in a package to a page.
- 3 Use Page Comments to specify what is happening on the page or why specific options were used.
- 4 Make use of all the features in APEX and stay as close as possible to existing features within APEX.
- 5 Develop naming conventions for the applications items.
- 6 Create a LOV for every table holding data of a specific type to ease the life of users and ensure no invalid values are entered.
- 7 Create modules within your application
- 8 Never allow application pages to access a database table directly but use a view instead to provide abstraction over the underlying tables.
- 9 Consider enabling feedback about your application from within the application.
- 10 Review the application code with each introduction of a new APEX version.
- 11 Instrument heavily (automate).
- 12 Share knowledge within the teams by using Shared Components within APEX and the Publish/Subscribe method.

6.3 Build to be secure

- 1 Use the documented Oracle [APEX security best practices](#) to protect the application and to avoid the introduction of SQL injection, URL tampering or cross site scripting.
- 2 Create a custom authentication that allows single sign-on (HarvardKey Login).

- 3 Create appropriate authorization schemes so persons see only what they are allowed to.
- 4 Use session state protection in shared components and bind variables wherever possible.

6.4 Create templates for applications

Pick a theme and create a template and used by all developers within your organization for a consistent look and feel within and across applications. Embedded in the templates should also be the reusable components and security models.

6.5 Logging and debugging

The ability to be able to turn on logging and debugging at any point to see what is happening behind the scenes is particularly important. Utilize all the appropriate Oracle provided features and recommended logging and debugging techniques.

6.6 Build for performance and scale

- 1 All code must be tested on performance. The documented Oracle database and APEX best practices such as using analytical functions instead of creating a custom PL/SQL routine should be followed.
- 2 Use external files on web servers to generate less hits on the database.
- 3 Use bind variables instead of substitution strings.
- 4 Where possible, build an application performance report into the applications so administrators or developers can easily see what is going on.

6.7 Create reusable components

Build as many reusable components as possible such as PL/SQL packages to do error handling and if APEX needs to be extended, make use of plugins, or use the main building blocks of APEX like PL/SQL, jQuery, HTML5, CSS.

6.8 Workspaces

Workspaces enable the separation of web applications and databases to ensure security and data integrity. Each workspace is associated with one or more database schemas (database users) which are used to store the database objects, such as tables, views, packages. Applications can only be associated with one workspace.

There are several things to consider when creating workspaces such as security, users and development accounts, REST services and interactive reports.

A workspace with access to multiple database schemas would be a concern for managing RESTful services because there is currently no way to specify a parsing schema. In this case, a one-to-one mapping of schema to workspace with the purpose of managing that schema's RESTful services would be one way to go.

In an organization with less experienced developers, giving each developer their own workspace and schema might be most appropriate. This would allow them the space to learn more about the tool without any chance of negatively impacting other business applications.

In general, determine the required user and developer management scheme then plan out the appropriate APEX workspaces.

6.9 Documentation

Below are a few best practices to follow on documentation:

- 1 Create conceptual design documents.
- 2 Within the PL/SQL packages
 - use code comments to guide users in understanding how to use (especially in the package spec) and maintain the code (especially in the package body), and
 - inside a program unit only use the line commenting technique.
- 3 For generated documentation,
 - if you use the JavaDoc syntax, then you can use `plsql-md-doc` to generate an easy-to-read document, or
 - alternatively, Oracle SQL Developer or PL/SQL Developer include documentation functionality based on a JavaDoc -like tagging.
- 4 Within the APEX applications, use the comment space in each component page, region, item, process, etc. to specify what is happening or to specify any technical options used.
- 5 Create end-user guides.

6.10 Build APIs

It is still a common practice for teams to use database links to expose and share data between databases. Database links while a simple, straightforward, and useful method for transferring data, pose several concerns including security, distributed transactions and performance resulting from the coupling of databases and applications.

ORDS provides all the requirements to create, manage, expose, and secure RESTful services. If you need to expose or share data from your APEX application, consider creating REST services against your database objects.

6.11 Inventory your applications

An inventory would keep track of all the custom APEX applications within your environment. It would also provide valuable answers to important questions like

1. What business processes does the application support?
2. Are there dependencies between the applications?
3. Are there several applications supporting the same or similar business processes?
4. Are there applications that are unused or barely used?

7 Best Practices for APEX Deployment

As with the development best practices above, consideration should be given to the deployment best practices below to ensure the application will operate properly and be supportable over time.

7.1 Use runtime only mode for production environments

The runtime environment enables users to run a production application without supporting the ability to change or edit the application or the ability to run ad hoc queries from the SQL Workshop component.

A runtime environment includes only the packages necessary to run your applications, making it a more hardened environment. It does not provide a web interface for administration.

7.2 Deployment Standards

Standard deployments are necessary. Create best practices and security guidelines for APEX deployments.

7.3 Shared Infrastructure

Review existing APEX environments with a goal of using shared infrastructure as much as possible.

7.4 Consider using RDS

Using Amazon's managed database service (RDS) means AWS takes over many of the difficult and tedious management tasks of a relational database like installations, upgrades, patching, backups, automatic failure detection, and recovery.

7.5 Lower the barrier for entry for non-prod environments

Create a self-service development environment where developers and other tech-savvy users can create prototypes and proofs of concept for business applications.

7.6 Create best practices and templates for variable URLs

Create naming conventions and templates for the use of variable URLs.

7.7 Acquire a common license for AOP plugin

The various product teams should consider consolidating the AOP Plugin licenses into a common license.

7.8 Production Support Organizations

Ensure support organizations such as DevOps are prepared and trained to support the APEX applications.

8 Scenarios Where APEX is NOT a Good Fit

The situations below would warrant that a team look at alternatives to APEX.

It is important to note that the APEX platform is constantly evolving and some of the scenarios below may get addressed in future releases. The Oracle APEX roadmap can be found here <https://apex.oracle.com/en/learn/resources/roadmap/>.

8.1 Native applications

These are applications built to be used on a specific device or OS e.g., navigation apps on mobile phones. These applications require native low-level access to a device's peripherals.

8.2 Client/Server Applications

In general Oracle APEX is most appropriate for data centric web-based applications. If the desired solution is a rich client application, then it is not an appropriate platform.

8.3 Offline Applications

The application can execute offline without network/internet connection to supporting application and database servers. APEX developed applications require access to the database to work.

8.4 Single Page Applications

A single page application is a web application that interacts with the user by dynamically rewriting the current page, rather than loading entire new pages from the server.

Oracle APEX is a multi-page application (MPA) framework. It is possible to achieve single page applications using custom code or by integrating a JavaScript framework like Oracle Jet, ReactJS, AngularJS etc. However, these options would likely involve more effort and result in a more complex application.

8.5 Mobile Applications

Oracle APEX allows for the creation of web applications which responsively work on any mobile device; however, it does not natively allow for the creation of applications that can be installed on a mobile device. To solve this problem, applications would need to be implemented as a Progressive Web Applications (PWA). Even then, the application would still require a connection to the Oracle database to work.

8.6 Version Control Requirements

For large scale development efforts, an effective framework that allows multiple developers to work concurrently is necessary. The ability to manage multiple versions of the application through its development life cycle is also critical.

Unlike file-based development languages, APEX stores the application definitions as metadata inside an Oracle database. It has an export feature that generates a file with many randomized values which makes it challenging to compare two copies of the same application.

Therefore, key aspects of version control like tracking changes, tagging, and merging code, and rolling back changes have to be handled differently in APEX. For example, to mitigate the inability to track changes, it is recommended that as much code as possible is stored in packages. These packages can be stored as regular files and as such it would then be simple to do a diff comparison.

At an application level, tools like APEX Diff can be used to export an APEX application in a JSON format. Diff Comparisons using the JSON format exports can then be done on different versions of an APEX application.

8.7 Other Scenarios

If the resulting application

1. relies heavily on JavaScript functionality,
2. needs to analyze unstructured data like Twitter feeds,
3. is required to manipulate OS files or
4. to create complex and static reports then APEX is likely not a good fit.

9 Additional Considerations

The scope of this advisory is limited to APEX design, development, and operation. Within Harvard there are several other development platforms in use, and considerations that go beyond the use of APEX.

9.1 Other Platforms

This advisory does not offer a comparison between Oracle APEX and any other development platforms.

9.2 Other advisories

The following are additional resources on the [Enterprise Architecture website](#) that would be helpful when creating or implementing any software solution.

- 1 [API Principles and Practices: Use of the API Gateway and Portal](#)
- 2 [API Principles Checklist](#)
- 3 [Application Architecture Checklist](#)
- 4 [Data Exchange Mechanisms and Considerations](#)

10 References

1. <https://docs.oracle.com/database/apex-5.1/HTMDB/understanding-developer-security-best-practices.htm#HTMDB29891>
2. <https://www.neooug.org/gloc/Presentations/2019/SpendoliniAPEX%20Security%20Checklist.pdf>
3. <https://vmorneau.me/apex-pwa-part1/>
4. <https://www.gartner.com/document/4005973?ref=solrAll&refval=313130009>
5. <https://insum-labs.github.io/plsql-and-sql-coding-guidelines/v1.1/3-coding-style/02-coding-style-comments/>
6. <https://www.youtube.com/watch?v=OIC2Edl9i-c>

11 Appendix – Examples of APEX applications at Harvard

TABLE 1: CUSTOM BUSINESS APPLICATIONS

#	Use Case/Application	Description	IT Team	Business Area
1	ASIP (Annual Salary Increase Process)	<ul style="list-style-type: none"> Administrative interface for departments, divisions, and HR to manage FAS annual salary increase processes for non-union and union staff. Facilitates informed decisions - identifies departments' budget pools and can incorporate performance ratings. Manages data in flux that requires action. Includes approval workflows. Generates official letters to staff. 	HUIT/ATS (HR Systems)	Human Resources
2	FROPS (Faculty Review, Offer, Professorship and Searches)	<ul style="list-style-type: none"> One of the largest Apex applications with over 200 database tables. Faculty career lifecycle event tracking application: tracks and reports on searches, candidates, offers, reviews, and leaves management. Tracks a faculty member's leave history, eligibility, request tracking, and notes in conjunction with the Leave Request Application. FAS only 	HUIT/ATS (HR Systems)	FAS Faculty Affairs
3	PREP (Position Request Portal)	<ul style="list-style-type: none"> Facilitates the FAS position controls process, for staff positions. Includes approval workflows. Provides a repository for statuses/outcomes/decisions about positions. Facilitates basic reporting. 	HUIT/ATS (HR Systems)	Human Resources
4	Bonus Application	<ul style="list-style-type: none"> Like ATS' ASIP but bonus focused Could not use ASIP because of different business rules. Data sent back to PeopleSoft after the workflow completes 	HMS	Human Resources
5	Helpdesk Utility Application	<ul style="list-style-type: none"> Connects to Midas and ServiceNow Provides standardized list views 	HMS	IT Service
6	HarvardKey Registration	<ul style="list-style-type: none"> In the pipeline. Would eliminate the PDFs currently used to generate the information entered in CAS. It would provide a mechanism for reporting on the number of applications registered, who owns each app etc. It would provide a mechanism to create new application requests. Currently there are 4 protocols in use for authentication managed by 4 different applications. 	IAM	
7	FIFA (Faculty Instructional Funding Application)	<ul style="list-style-type: none"> Tracks endowed funds with the purpose of professorship. Matches faculty to restricted funds with the purpose of freeing up unrestricted dollars. Tracks faculty agreements included in offer letters and generated add pay upload file for PeopleSoft. FAS Only 	HUIT/ATS (HR Systems)	FAS Faculty Affairs and Finance
8	HILDA (Harvard Interactive Library of Data Assets)	<ul style="list-style-type: none"> Tracks data elements and data dictionary components for Alumni Affairs. Replaced Collibra. 	HUIT/ATS (HR Systems)	Alumni Affairs & Development

9	Asset Tracker	<ul style="list-style-type: none"> Tracks assets from purchase to assignment. Has full histories and reports on the department's technical needs 	HMS – AA&D	Alumni Affairs & Development
10	Reunion Committee Portal	<ul style="list-style-type: none"> Application for reunion committee members. 	HMS – AA&D	Alumni Affairs & Development
11	Board of Fellows	<ul style="list-style-type: none"> Personalized site for Board of Fellows to share resources and contacts. 	HMS – AA&D	Alumni Affairs & Development
12	Intramural Sports	<ul style="list-style-type: none"> Intermural app allows you to manage those events and meetups set up by the intermural division. 	Student Systems	Student & Student Lifecycle
13	Athletics Applications	<ul style="list-style-type: none"> There were some legacy apps written in Perl, that are getting rewritten in APEX and vended apps. 	Student Systems	Student & Student Lifecycle
14	College Facebook	<ul style="list-style-type: none"> unsupported and outdated College Facebook application is being rewritten with new features 	Student Systems	Student & Student Lifecycle
15	Roster Wizard	<ul style="list-style-type: none"> unsupported roster wizard application is being re-written. This helps house monitors and security personals to manage roster related information 	Student Systems	Student & Student Lifecycle
16	Course Balance tool	<ul style="list-style-type: none"> A tool used by FAS Colleges to ensure courses being created are following the rules set by the college 	Student Systems	Student & Student Lifecycle
17	Office or Arts registration system	<ul style="list-style-type: none"> OFA's online registration system is going out of support, and they wanted to create a complete online registration system that allows them to register internal and external patrons with payment capability. 	Student Systems	Student & Student Lifecycle
18	Research Administration Portal	<ul style="list-style-type: none"> Brings together data from multiple research administration and compliance applications to provide researchers with a single view of pending research administration tasks and their current portfolio. Includes selected data from GMAS (Oracle) and the Huron Research Suite (SQL Server). Does not replace specific, purpose-built administration and compliance applications but helps researchers to navigate disparate research administration requirements. 	HUIT/ATS	Research Administration

TABLE 2: REPORTING AND WORKFLOW APPLICATIONS

#	Use Case/Application	Description	IT Team	Business Area
1	Applications Metrics Platform	<ul style="list-style-type: none"> Provides charts and reports used for monthly or annual reporting on the applications portfolio. e.g., since 8/2019, there have been 7,100 distinct users, 240,000 logins etc. 	HMS/IT	IT Service
2	Aurora Reporting	<ul style="list-style-type: none"> Reporting tool for Aurora HR (FAS) application. 	HUIT/ATS/HR Systems	Human Resources
3	POI Portal	<ul style="list-style-type: none"> Reporting Application - My POI dashboard. Manages transaction requests for new POIs. Workflows using open-source plugin called flows for APEX. 	HUIT/IAM	
4	Masthead	<ul style="list-style-type: none"> FAS listing of faculty and affiliates of a department and standing committee members including chairs and area deans. 	HUIT/ATS (HR Systems)	FAS Faculty Affairs

5	Reunion Report Portal	<ul style="list-style-type: none"> Application to access, edit and submit reunion reports. 	HMS – AA&D	Alumni Affairs & Development
6	Pledge Reminders	<ul style="list-style-type: none"> Application to manage open pledges and ensure fulfillment. 	HMS – AA&D	Alumni Affairs & Development
7	Data Hub	<ul style="list-style-type: none"> A unified location for data for data updates from both individuals and systems. 	HMS – AA&D	Alumni Affairs & Development
8	Donor Acknowledgement	<ul style="list-style-type: none"> Has weekly refreshing acknowledgement batches that allow for both personalized recognition and efficient delivery. 	HMS – AA&D	Alumni Affairs & Development
9	Gift Report	<ul style="list-style-type: none"> Customized fiscal gift reporting for HMS' units outside of AAD and finance. 	HMS – AA&D	Alumni Affairs & Development
10	Reunion Event Management	<ul style="list-style-type: none"> Aggregates reunion event data. 	HMS – AA&D	Alumni Affairs & Development
11	Funds & Terms	<ul style="list-style-type: none"> Provides insights on HMS funds and their allocations. 	HMS – AA&D	Alumni Affairs & Development
12	Detail List of Gifts	<ul style="list-style-type: none"> Weekly view of new gifts. Site refreshed daily. Highlights important fundraising information. 	HMS – AA&D	Alumni Affairs & Development
13	Professorship Tracker	<ul style="list-style-type: none"> Term professorships are awarded to tenure-track faculty members to recognize and reward faculty achievements. 	HMS – AA&D	Alumni Affairs & Development
14	AAD Calendar	<ul style="list-style-type: none"> Central location for department events and travel of staff, faculty, and the dean. 	HMS – AA&D	Alumni Affairs & Development
15	Student Support System	<ul style="list-style-type: none"> Create case management tool for Student Support Team which allows them to help students that are directed to the office. 	Student Systems	Student & Student Lifecycle
16	Internal Org chart Applications	<ul style="list-style-type: none"> A quick way to get org chart information – similar to org chart published in HUIT site, but more dynamic and searchable. 	Student Systems	Student & Student Lifecycle
17	Testing Cadence Upload	<ul style="list-style-type: none"> Supports color coding application for covid-19. Previously spreadsheets were emailed back and forth. Users now upload their spreadsheets to the app for reporting. 	HUIT/IAM	
18	RAD (Athletics Reporting & Data)	<ul style="list-style-type: none"> Reporting and data platform for Athletics, including NCAA compliance data, team rosters, etc. 	Student Systems	Student & Student Lifecycle
19	Ivy Dashboard	<ul style="list-style-type: none"> These are APEX/BI Publisher dashboards to help customers on the Admission and Financial Aid 	Student Systems	Student & Student Lifecycle
20	HR Position Management System	<ul style="list-style-type: none"> Provides an electronic form and workflow to create or modify positions similar to Aurora. 	HMS	Human Resources

TABLE 3 : APPLICATION EXTENSIONS

#	Use Case/Application	Description	IT Team	Business Area
1	MARS (Medical Area Representation System)	<ul style="list-style-type: none"> Allows access to PeopleSoft data to administrative users who are not in PS. All PeopleSoft data viewed and presented differently. 	HMS	Human Resources
2	People in Space	<ul style="list-style-type: none"> Uses Center Stone data Center Stone licenses are limited so this allows more administrative staff to work with this data This application integrates with Blueprint & the helpdesk application It helps for instance the helpdesk to determine where people are seated when they need to provide services. 	HMS/IT	

		<ul style="list-style-type: none"> The application also provides space utilization information that helps with fed rate negotiations. 		
--	--	--	--	--

TABLE 4: AUTHORIZATION APPLICATIONS

#	Use Case/Application	Description	IT Team	Business Area
1	Alseta	<ul style="list-style-type: none"> Manages the application authorization for a subset of applications (mostly within FAS). Provides reports on user roles and permissions. Contains functionality to clone a user and also copyback permissions from production to a lower environment. 	HUIT/ATS (HR Systems)	IT Service (for certain areas)
2	IAM Authorization	<ul style="list-style-type: none"> Authorization application is similar to Alseta. Alseta had gaps like delegated POI administration hence the creation of a new app. Provides authorization for APEX only applications. 	HUIT/IAM	IT Service
3	Tokens Online	<ul style="list-style-type: none"> Generates access tokens that provide temporary, secure access to certain pages and captures a user's input. 	HMS – AA&D	Alumni Affairs & Development
4	Token Management	<ul style="list-style-type: none"> Manages and authenticates tokens and the access they grant. 	HMS – AA&D	Alumni Affairs & Development