
VA Enterprise Design Patterns: Cloud Computing Platform-as-a-Service (PaaS)

**Office of Technology Strategies (TS)
Architecture, Strategy, and Design (ASD)
Office of Information and Technology (OI&T)**

Version 1.0

Date Issued: October 2016



THIS PAGE INTENTIONALLY LEFT BLANK FOR PRINTING PURPOSES

APPROVAL COORDINATION

Gary Marshall
Director, Technology Strategies, ASD

Paul A. Tibbits, M.D.
DCIO Architecture, Strategy, and Design

REVISION HISTORY

Version	Date	Organization	Notes
0.1	5/25/2016	ASD TS	Initial Draft/Outline
0.3	6/21/2016	ASD TS	Second draft document with updates made throughout document based upon initial internal/external stakeholder review and comment.
0.5	8/05/2016	ASD TS	Updated draft document created prior to community review. Updates made prior to Public Forum.
0.7	9/23/2016	ASD TS	Updates made following Public Forum collaborative feedback and working session.
1.0	10/5/2016	ASD TS	Final version for TS leadership approval and signature, including all applicable updates addressing stakeholder feedback and Section 508 Compliance.

REVISION HISTORY APPROVALS

Version	Date	Approver	Role
0.1	6/1/2016	Bonnie Walker	Enterprise Design Pattern Lead
0.3	6/23/2016	Bonnie Walker	Enterprise Design Pattern Lead
0.5	8/12/2016	Bonnie Walker	Enterprise Design Pattern Lead
0.7	10/4/2016	Bonnie Walker	Enterprise Design Pattern Lead

TABLE OF CONTENTS

1	INTRODUCTION	1
1.1	BUSINESS PROBLEM.....	2
1.2	BUSINESS NEED	2
1.3	BUSINESS CASE.....	3
1.4	APPROACH.....	4
2	CURRENT CAPABILITIES	5
2.1	CURRENT PAAS OFFERINGS	5
2.2	CURRENT LIMITATIONS.....	5
3	FUTURE CAPABILITIES.....	6
3.1	ORCHESTRATION	8
3.2	DEVOPS	9
3.3	ALIGNMENT TO THE TECHNICAL REFERENCE MODEL (TRM)	10
3.4	ALIGNMENT TO VETERAN-FOCUSED INTEGRATION PROCESS (VIP)	11
4	USE CASES	11
4.1	CREATING AN APPLICATION PROGRAM INTERFACE (API) WITH DIFFERENT PROGRAMMING LANGUAGES.....	11
4.1.1	<i>Purpose</i>	11
4.1.2	<i>Assumptions</i>	12
4.1.3	<i>Use Case Description</i>	12
4.2	SETTING UP A TEST ENVIRONMENT	13
4.2.1	<i>Purpose</i>	13
4.2.2	<i>Assumptions</i>	13
4.2.3	<i>Use Case Description</i>	13
	APPENDIX A: SCOPE.....	16
	APPENDIX B: DEFINITIONS	17
	APPENDIX C: ACRONYMS.....	18
	APPENDIX D: REFERENCES, STANDARDS, AND POLICIES	20

FIGURES

Figure 1: Architectural Concept for VA Cloud Computing based on NIST Architecture 2
Figure 2: PaaS Architecture Diagram 7
Figure 3: DevOps Process (via VA Industry Partner) 10
Figure 4: Process Flow of using a PaaS to change technology stacks 13
Figure 5: Using Multiple Automated Testing Environments 15

TABLES

Table 1 - List of Approved Tools and Standards 11

1 INTRODUCTION

PaaS enables standardized, repeatable, and highly reliable deployments for applications. Platform-as-a-Service (PaaS) is a category of cloud computing services that provides a mechanism enabling consumers (developers, testers, administrators, and in some cases users themselves) to develop, automatically test, manage, and run applications while outsourcing infrastructure management. PaaS removes complexity of building and maintaining the infrastructure typically associated with developing and launching an application. It provides unique automation that abstracts computing, storage, and networking capabilities associated with Infrastructure-as-a-Service (IaaS). The Department of Veteran Affairs (VA) currently does not have a standard approach to leverage PaaS.

VA's desired attributes for deploying PaaS include:

- Creates automated components of an environment
- Provides a fully managed application development and deployment platform for Development Operations (DevOps) continuity
- Enables Service Level Agreement (SLA) guarantees
- Assumes less risk than dedicated physical machines with guaranteed auto-scaling that are tied to client SLAs

Figure 1, below, illustrates the key components of cloud computing, based on the National Institute of Standards and Technology (NIST) reference architecture. The PaaS component of the architecture is highlighted in red in the figure. NIST architecture components include the cloud consumer, cloud broker, cloud auditor, cloud carrier, and cloud provider. Within the cloud provider, there is service orchestration, cloud service management, security, and privacy. PaaS resides within the service orchestration, or service layer.

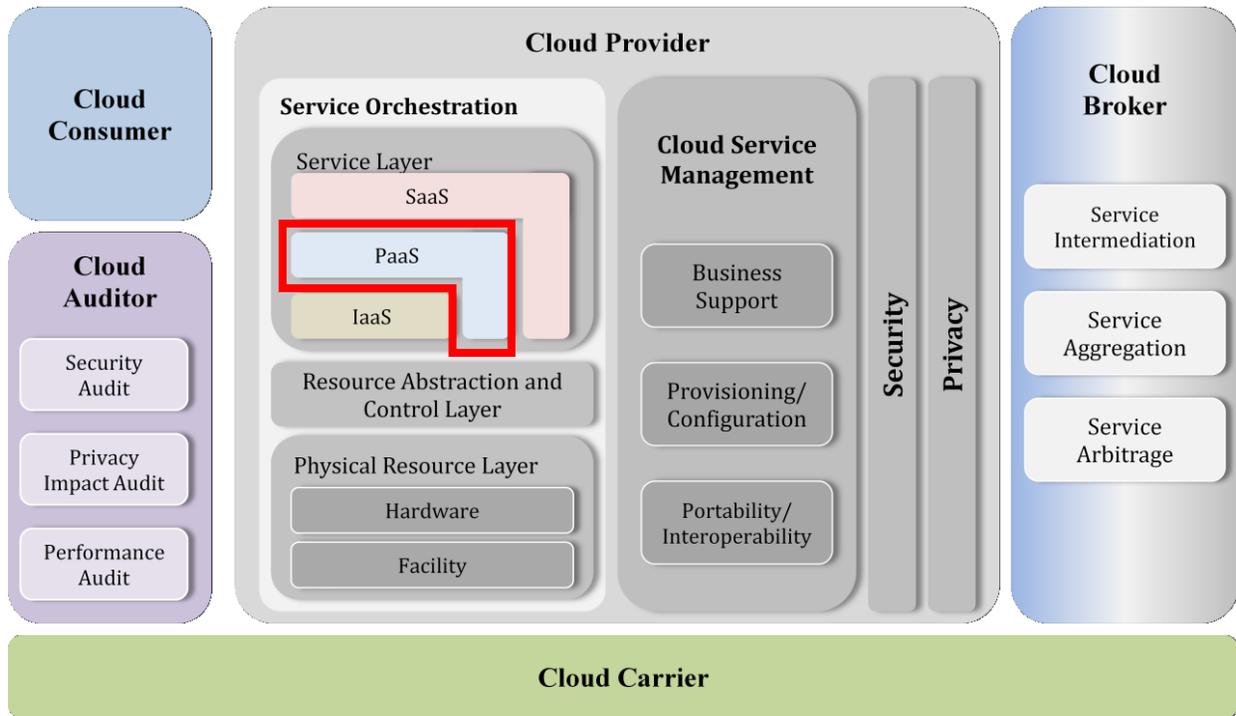


Figure 1: Architectural Concept for VA Cloud Computing Based on NIST Architecture

1.1 BUSINESS PROBLEM

VA relies on clouds to provide IaaS by default, but that is not always suitable for simple applications that do not require significant custom configurations. Relying on IaaS when not required wastes time and resources; as it's not a one-size-fits-all solution. Most of VA's management of virtual machines is manual and labor intensive in terms of building and patching. VA requires appropriate tools based on the business need or problem that it is trying to solve. Establishing approved test and development environments is labor intensive in VA. Scalability is currently lacking in the VA, but with PaaS, underlying computer and storage resources scale automatically to match application demand, without burdening the consumer. PaaS provides an abstraction layer to mitigate the risks of managing servers and systems that may not be properly configured to support a consistent end-to-end environment for developing, testing, and deploying solutions in production. As VA shifts to a "cloud first" approach, it must include externally managed PaaS as part of its IT service offerings. VA has not maximized the use of externally managed PaaS for development, testing, and deployment.

1.2 BUSINESS NEED:

PaaS provides the means for supporting and improving services while simultaneously managing costs, enabling innovation, reaching a wider Veteran community, and securely exchanging

information among VA partners. PaaS removes risks of migrating from physical to virtual machines and facilitates DevOps. PaaS, in conjunction with other enterprise IT capabilities, supports establishing an ecosystem for standardized platforms, in accordance with VA's IT strategy. This is sometimes dubbed "Platform-as-an-Ecosystem" (PaaE). These standardized platforms enable agile development in response to rapidly changing business needs. An additional benefit is "better supports multi-party engineering", where different vendors building separate components that must ultimately interoperate have access to PaaS Integrated Development Environments (IDEs), mitigating integration risks.

PaaS is needed across VA for the following reasons:

- PaaS improves service delivery by working faster and more cost effectively
- It provides enhanced scalability that reaches a growing number of Veterans seeking digital services from VA by providing a scalable solution that does not rely on VA owned and operated infrastructure
- PaaS integrates with other enterprise IT capabilities to support establishing an ecosystem for standardized platforms, in accordance with VA's IT strategy. This is sometimes called "Platform-as-an-Ecosystem" (PaaE). These standardized platforms support agile development
- PaaS better supports multi-party engineering, where different vendors building separate components with interoperability requirements have access to PaaS Integrated Development Environments (IDEs) which mitigates integration risks

1.3 BUSINESS CASE

VA's use of PaaS addresses challenges to rapidly delivering enterprise IT solutions to customers in response to changing business requirements. PaaS enhances agility using automated development and testing environments. PaaS will enable VA to:

- Add or remove nodes automatically
- Focus development efforts on the application and not the supporting infrastructure
- Take advantage of service provider advancements in platform support
- Build an entire environment (many clusters with many nodes) automatically
- Provide all the pieces of DevOps (continuous deployment and continuous delivery)

The benefits of such an approach include:

- Automated development and testing environments

- Shorter enhancement cycles (with more automated control over environment configurations)
- Development, testing, implementation, and maintenance cost savings (pay for only virtual resources used in multi-tenant DevOps farms)
- Improved uptime
- Improved SLA management
- Elasticity (dynamically adjust to future changes in capacity demands)
- Enhance baseline, configuration, and security compliance
- Reduced infrastructure management burden by focusing on the application code instead of the servers that run the code

1.4 APPROACH

This EDP defines a framework for using PaaS solutions and addresses the following:

- What do we need to know to evaluate options for a cloud approach?
- What are the customer needs that indicate PaaS may be appropriate?
- What are the business requirements that drive PaaS decisions?

The PaaS approach consists of the following:

- Gather business needs and define requirements
 - Determine purpose of the platform (e.g., development / system test / user acceptance test / production environment)
 - Determine contract structure, software and system integration, and interoperability needs
 - Determine application portability (“off-ramp”) needs
 - Determine how volatile demand is for application resources
 - Determine quality of pressures to limit capital expenditures, and move to operating expenditures
- Define the attributes of a PaaS cloud architecture
 - Determine infrastructure required
 - Determine software needed for developer environment

2 CURRENT CAPABILITIES

2.1 CURRENT PAAS OFFERINGS

PaaS offerings in VA consist of virtualization capabilities but they do not provide the automated provisioning and rapid elasticity required by the NIST definition as described in the Cloud Computing Architecture EDP. The service provider provides on-demand self-service, broad network access, resource pooling, elasticity, and a measured service to be considered PaaS.

The Corporate Regional Readiness Center (CRRC) at Falling Waters uses PaaS. The CRRC staff supports the Enterprise Web Infrastructure System (EWIS) and created a PaaS solution for a self-service interface using Microsoft System Center tools. An authenticated and approved user can instantiate a website and database developed on a myriad of offered web platforms, such as Ruby, Drupal, .Net, through the self-service interface. Users can either publish sites in their web Structured Query Language (SQL) shared farms or on virtual machines. The back end services for the sites are transparently managed for the consumer. The web service provides fully redundant local solutions as well as full disaster recovery services for the Philadelphia Information Technology Center (PITC).

2.2 CURRENT LIMITATIONS

There are limited ways to automate adding or subtracting nodes in VA's current private cloud environments. There is no capability to generate an entire environment automatically. Deploying every node on the network tends to be done manually. It requires significant effort among system administrators to design the proper network configurations and inter-process communications required to orchestrate the nodes to deliver consistent functionality to the customer. The private cloud environment also lacks resource pooling and elasticity that is commonly available in commercial offerings.

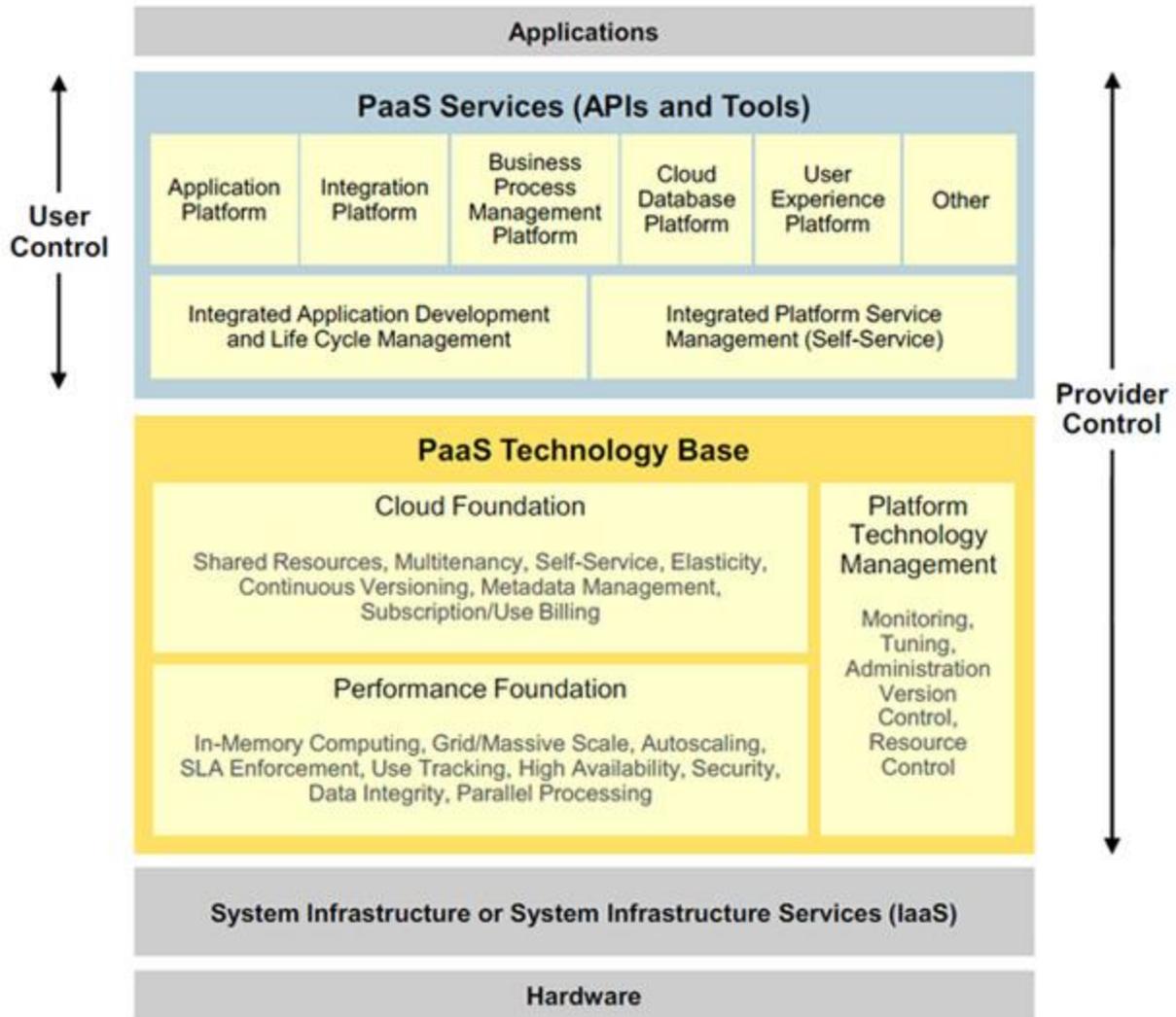
It is difficult to establish parity between the development, testing, and production environments with VA's current toolset. Due to the current inability to mirror these environments, development and operations efforts face challenges achieving the continuous integration and delivery that is required for an agile DevOps approach. Issues include development using images different from production, and reliance on testing in production to identify errors, as illustrated in the use case provided in Section 4. The current VA private cloud environment creates difficulties for the Office of Information and Technology (OI&T) to deliver software functionality that keeps pace with rapidly changing business needs.

3 FUTURE CAPABILITIES

VA will benefit from implementing a set of PaaS functions to address the limitations described in Section 2.2. The resulting cloud environment will leverage both private, open source PaaS solutions, and commercially available PaaS solutions that enable automated testing and deployment of new applications. VA will evaluate service providers based on their ability to satisfy the following attributes to select a PaaS:

- Establishes automated components of an environment
- Enables developers to access a managed environment for development, testing, deployment, and production
- Provides auto-scaling capabilities and abstracts lower-level infrastructure components from the project teams

Figure 2, below, shows the major elements of PaaS, including specific PaaS Services and Technology Base components. Responsibilities are shared between the project and the cloud service provider. The project management team is responsible for ensuring that the application meets the user's requirements and has been tested prior to deployment. When the application is deployed into the PaaS, the project provider is responsible for access control and choice of technology stack, while the cloud provider is responsible for the underlying infrastructure on which the application runs. The user has the flexibility to leverage a wide variety of application platforms, integration platforms, managed database platforms, lifecycle management functionality, business process management platform, user experience platform, amongst others that are available on-demand. The cloud service provider is responsible for managing the underlying technology base that supports multi-tenancy, elasticity, and shared resources that make up the cloud foundation.



Source: Gartner (September 2011)

Figure 2: PaaS Architecture Diagram

PaaS capabilities will allow VA to routinely (and frequently to keep pace with industry and applications) replace entire environments. Once the new environment is validated, OI&T can then instantaneously migrate from the previous environment. A development environment can be completed and moved to test and production easily in the cloud, as demonstrated in the use cases provided in Sections 4.1 and 4.2. This will become the promotion mechanism of choice to support uptime in production.

Having PaaS in production will reduce transition risk and avoid downtime when deploying new capabilities or upgrading existing capabilities. Without PaaS, every change is a manual and labor

intensive process requiring subject matter expertise on each of the current existing systems. PaaS enables VA to achieve repeatable and highly reliable deployments for applications.

3.1 ORCHESTRATION

Orchestration refers to the composition of system components to support the cloud provider activities in arrangement, coordination, and management of computing resources in order to provide cloud services to cloud consumers. Orchestration ensures that different services and applications exchange information in a coordinated manner. Traditionally when dealing with services in an operating environment, the system administrators are responsible for optimizing the network configuration and inter-process communications of different services. PaaS will manage the service configuration and communication using built-in functions managed by the cloud service provider.

Determining the networking solutions required for orchestrating services for the end user is the primary challenge. This is encountered in many open-source implementations. In PaaS, the orchestration is abstracted away for the consumer.

OI&T needs PaaS solutions that abstract lower level infrastructure services that perform the following orchestration functions:

1. Log scraping of application specific conditions
2. Client SLA monitoring
3. I/O (storage and network) and path monitoring (using logic to add paths where they are overloaded)
4. Hypervisor and top level cloud monitoring / orchestration (e.g., OpenStack) to contribute to the logic
5. Orchestration that works with object storage and block storage based upon specific application provisioning needs
6. OS monitoring contributing to the PaaS logic (via tools such as SystemEDGE)
7. Platform specific monitoring to contribute to the logic (where other tools may not be enough)
8. Orchestration that starts the build of a new node (e.g., SaltStack working with Glance) with specific configurations to prepare to be part of a cluster
9. Orchestration that works with Active Directory or Lightweight Directory Access Protocol (LDAP) (e.g., Keystone from OpenStack) so proper user and group permissions are propagated during the build

10. Orchestration that integrates with configuration management tools so nodes are patched on schedule
11. Orchestration that works with networking capabilities for IP address pooling and firewall and load balancer modifications

PaaS solutions also require user interfaces that allow application subject matter experts (SMEs) to interact with the orchestration automation (multi-tenant design). This enables projects to connect interdisciplinary teams that can respond when the PaaS is not producing the desired effects.

3.2 DEVOPS

DevOps is a collaborative approach to development and operation efforts throughout the entire end-to-end, iterative process of developing, testing, and deploying applications (an agile process). Best practices and lessons learned after a release iteration are leveraged in subsequent releases. DevOps is one of the drivers of PaaS: its automation and on-demand features make DevOps much more convenient than using on-premise virtualization solutions or physical servers.

PaaS supports automated provisioning and mirroring of development, test, and production environments. PaaS enables one to quickly and efficiently develop or test code, without the need for huge upfront investments. It is “pay as you go” service, and allows for rapid prototyping and software development. Without PaaS, cycle times tend to be longer and development and test cycles tend to be more complicated due to the need to control the underlying infrastructure. PaaS enables rapid and efficient expansion much more quickly and efficiently.

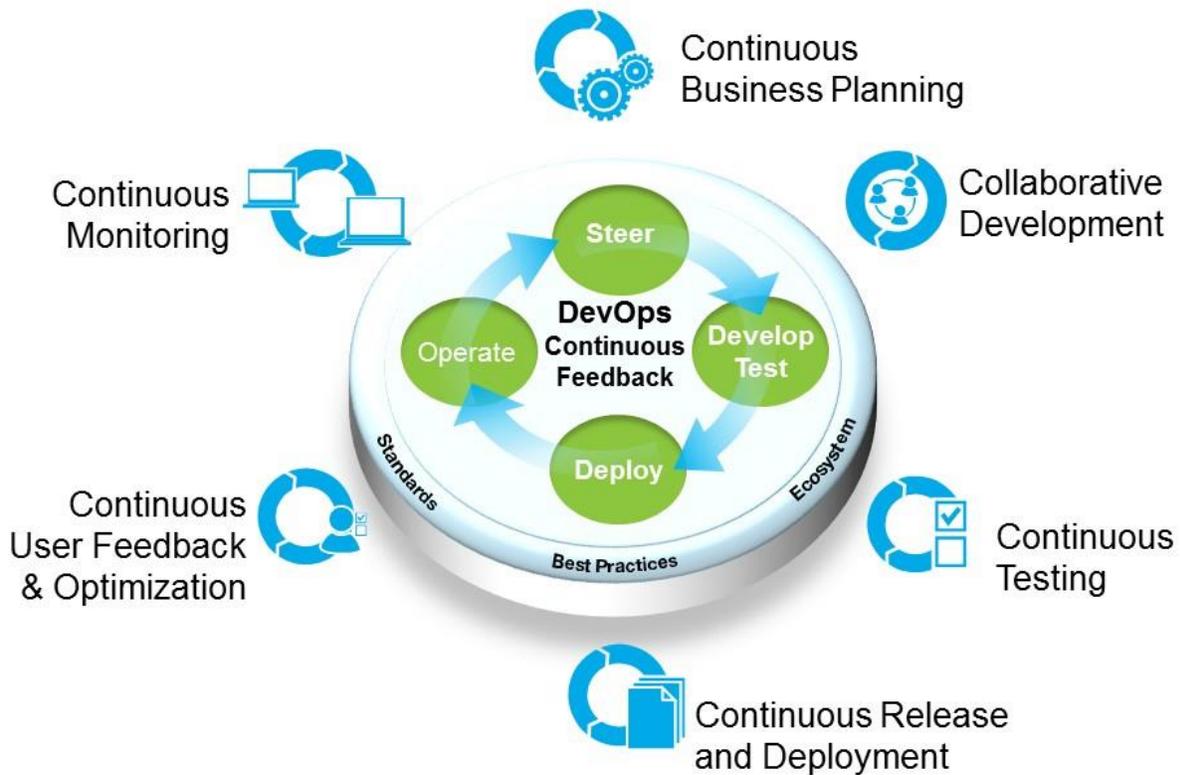


Figure 3: DevOps Process (via VA Industry Partner)

The figure shows that PaaS enables DevOps to proceed using collaborative development, testing, and monitoring. It also facilitates agile release and deployment.

3.3 ALIGNMENT TO THE TECHNICAL REFERENCE MODEL (TRM)

All projects will leverage the approved tools and technologies located in the VA Technical Reference Model (TRM)¹ to comply with the architectural guidance provided in this document. Table 1, below, lists the approved tools for this EDP.

¹ <http://trm.oit.va.gov/>

Table 1 - List of Approved Tools and Standards

Tool Category	Example Approved Technologies
Cloud Technologies	CloudForms, EMC Atmos GeoDrive, iCloud, Heroku, OpenShift Enterprise, OpenStack, Cloud Foundry, and Azure
Virtualization Software	Citrix XenApp, Docker, Linux Containers, IBM WAVE for z/VM, VMware Tools, and VirtualBox
Miscellaneous	Atlantis USX, HP Command View EV A, PhoneView, SaltStack, Tivoli Storage Manager for Space Management, and Veritas Enterprise Administrator
Data Center Automation Software	BMC Application Automation, SystemEDGE, and Microsoft Center Operation Management

3.4 ALIGNMENT TO VETERAN-FOCUSED INTEGRATION PROCESS (VIP)

VIP is a Lean-Agile framework that services the interest of Veterans through the efficient streamlining of activities that occur within the enterprise. The VIP framework unifies and streamlines IT delivery oversight and will deliver IT products more efficiently, securely, and predictably. VIP is the follow-on framework from Project Management Accountability System (PMAS) for the development and management of IT projects which will propel the Department with even more rigor toward Veteran-focused delivery of IT capabilities.

More information can be found here (<https://vaww.oit.va.gov/veteran-focused-integration-process-vip-guide/>).

4 USE CASES

4.1 CREATING AN APPLICATION PROGRAM INTERFACE WITH DIFFERENT PROGRAMMING LANGUAGES

4.1.1 Purpose

PaaS supports multiple frameworks and libraries to help development teams implement Application Program Interfaces (APIs) using a wide variety of programming languages to enable multi-party collaboration. The API implementation will vary. The code can be written in Java,

Node.js, Python, amongst other languages. Using PaaS requires limited manual intervention in order to switch technology stacks as the needs of the application change.

4.1.2 Assumptions

- PaaS supports programming languages and frameworks that are approved in the TRM.
- Web application has been architected to support business needs as captured in epics.
- The team is an integrated project team, which consists of developers, testers, and operators.
- Cloud security requirements are established in accordance with the Cloud Security EDP.
- Cloud interoperability requirements are established in accordance with the Cloud Computing Architecture EDP and Cloud Broker EDP.

4.1.3 Use Case Description

In this example, the development team is creating a Java based API. After developing the code, the developers enter the code into a version control system. The development team has created a Representational State Transfer (RESTFUL) API that is integrated with an API management system, such as an API gateway.

- The project team pulls the code from the version control system, conducts the application build, and sets up an automated test environment using PaaS. After all the tests are done, the project team is able to release the application using PaaS by doing a one-click deployment.
- Business needs change and the project team determines that they need to change their technology stack to adapt to these changes. The project team goes to PaaS, which provides a management console. The console enables the project team to change their API from Java to Python.
- The development team recodes the application via Python using the coding libraries provided through the PaaS.

PaaS enables service interaction through a standardized API that delivers the same functionality to the consumer even after the technology stack changes from Java to Python. This means that PaaS enables rapid changes to applications in accordance with Service Oriented Architecture (SOA) design principles. More information about implementing APIs following these design principles can be found in the Microservices EDP.

The following process flow is enabled by leveraging programming frameworks and libraries for each technology stack through the use of a commercial PaaS:



Figure 4: Process Flow of using a PaaS to change technology stacks

4.2 SETTING UP A TEST ENVIRONMENT

4.2.1 Purpose

The project team needs an automated test environment which can be set up with the click of a button whenever a new release is planned. PaaS provides the ability to set up a script that establishes the test environment using the same configuration as the planned production environment.

4.2.2 Assumptions

- The team has developed a script that automatically creates a testing environment every time code is committed to the version control system.
- PaaS supports programming languages and frameworks that are approved in the TRM.
- The team is an integrated project team, which consists of developers, testers, and operators.
- Cloud security requirements are established in accordance with the Cloud Security EDP.
- Cloud interoperability requirements are established in accordance with the Cloud Computing Architecture EDP and Cloud Broker EDP.

4.2.3 Use Case Description

For each release, the developers and testers will work together to set up each testing scenario. Each scenario will have a script that can readily create an environment. For each environment that's established, the PaaS provides a continuous integration system that coordinates the testing and evaluation activities prior to a release.

As long as the environment stays the same, user can run the same type of test. If the environment changes, user simply modifies the script once, and the PaaS adapts.

PaaS facilitates testing for unit, system, user acceptance, or integration testing. It allows for automation of a process.

A high level DevOps lifecycle created by PaaS to support testing throughout the application lifecycle contains the following steps:

- Developer checks in a new release by submitting source code to a version control system.
- The developers and testers pull the source code from the version control system and build the application.
- Developers conduct unit testing before transitioning it to the integration environment.
- Once the application has been built, developers and testers use scripts to automatically set up the integration environments.
- If errors are identified, the testing environment automatically roles back to the development environment.
- The development team corrects the errors in the code and resubmits the integration tests.
- When integration tests succeed, the application automatically transitions to a system testing environment.
- If the tests fail, there can be a rollback to the previous environment. If the test is successful, the user acceptance environment can be created.
- Once the user acceptance test passes, the application is ready to be deployed into a production environment. This can be done in an automated fashion using a continuous integration server, or involve a manual review before the application is deployed.
- The principle of “Test Driven Design” (TDD), where if new functionality was developed on subsequent deployment, then the first pass will cause tests (written first) to fail. Only after software is written successfully will all the tests pass.
- This is a repetitive cycle and can be repeated for every single deployment.

This high level process flow is depicted in the figure below:

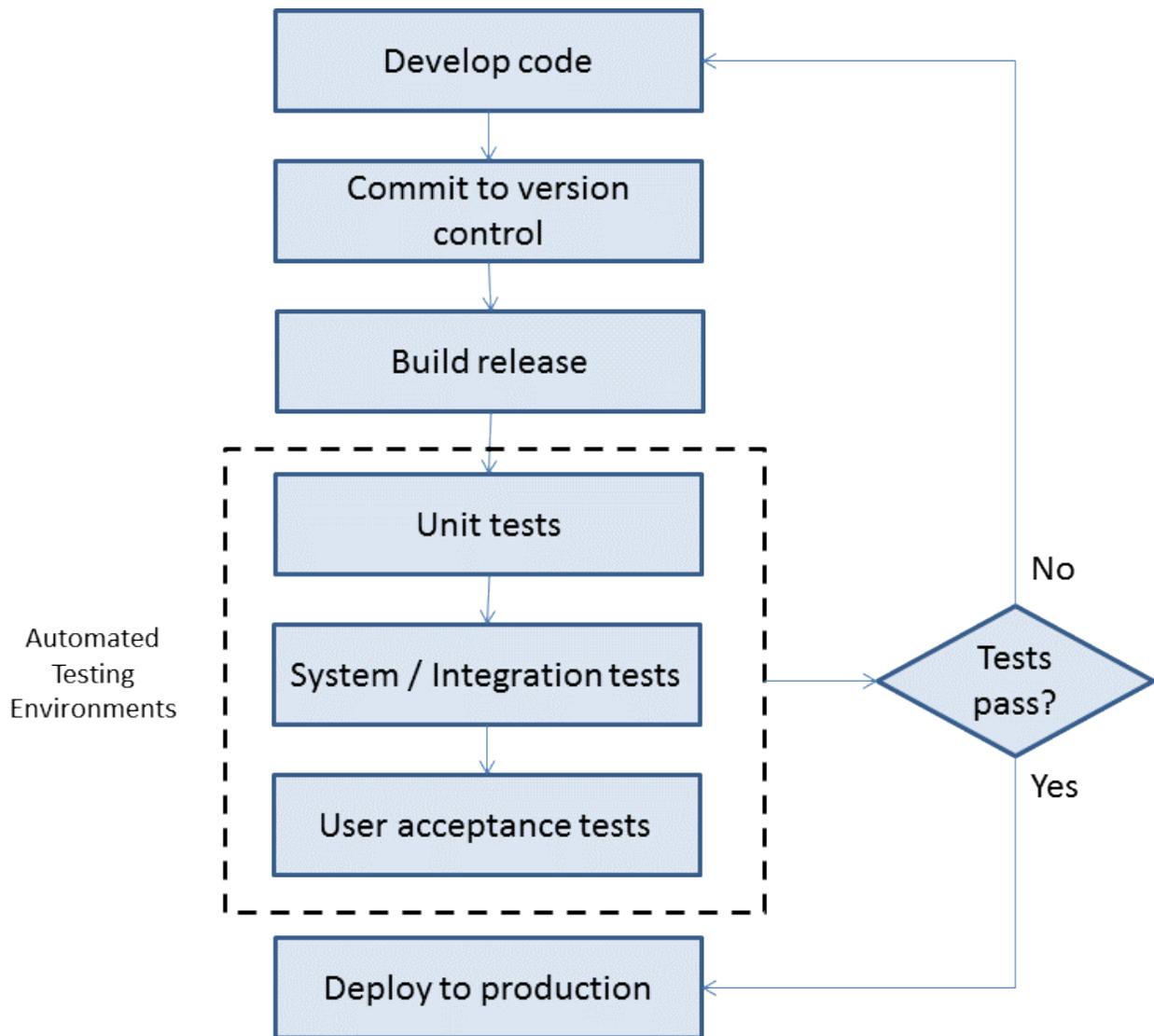


Figure 5: Using Multiple Automated Testing Environments

APPENDIX A: SCOPE

SCOPE

The Cloud Computing Platform-as-a-Service (PaaS) Enterprise Design Pattern (EDP) defines a framework for using PaaS solutions and addresses the following:

- What requirements do we need to know before we acquire PaaS?
- What are the user needs that drive these requirements?
- What are the business requirements that drive PaaS decisions?
- How VA automates control and manages its IaaS platforms and those it wants to acquire?

INTENDED AUDIENCE

The primary audience for this document consists of VA stakeholders who manage and/or conduct cloud computing activities on behalf of their organization (e.g., office, program, LOB). Specifically, these stakeholders are:

- System and application owners/stewards/project managers
- Executive leadership in IT (CIO, division head, etc.)

This document is also intended for those in leadership roles who can establish governance mechanisms and policies related to analytics, software development, and data management.

DOCUMENT DEVELOPMENT AND MAINTENANCE

This EDP was developed collaboratively with internal stakeholders from across the Department and included participation from VA's Office of Information and Technology (OI&T), Enterprise Program Management Office (EPMO) Office of Information Security (OIS), Architecture, Strategy and Design (ASD), and Service Delivery and Engineering (SDE). Extensive input and participation was also received from Veterans Health Administration (VHA), Veterans Benefits Administration (VBA), and National Cemetery Administration (NCA). In addition, the development effort included engagements with industry experts to review, provide input, and comment on the proposed pattern. This document contains a revision history and revision approval logs to track all changes. Updates will be coordinated with the Government lead for this document, which will also facilitate stakeholder coordination and subsequent re-approval depending on the significance of the change.

APPENDIX B: DEFINITIONS

Cloud Consumer - A person or organization that maintains a business relationship with and uses services from a cloud provider

Cloud Provider - A person, organization, or entity responsible for making a service available to interested parties

Cloud Auditor - A party that can conduct independent assessment of cloud services, information system operations, performance, and security of the cloud implementation

Cloud Broker - An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers

Cloud Carrier - An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers

Epic - Clarification of a business initiative at a high level

Microservices - Dividing a larger application into smaller discrete combinable services

APPENDIX C: ACRONYMS

Acronym	Description
AA&A	Authentication, Authorization and Audit
AITC	Austin Information Technology Center
API	Application Programming Interface
ASD	Architecture, Strategy and Design
ATO	Authority to Operate
CIA	Confidential, Integrity and Availability
COTS	Commercial Off the Shelf
CRM	Customer Relationship Management
CSP	Cloud Service Provider
EA	Enterprise Architecture
ECSB	Enterprise Cloud Services Broker
ESS	Enterprise Shared Services
ETA	Enterprise Technical Architecture
ETSP	Enterprise Technology Strategic Plan
FedRAMP	Federal Risk and Authorization Management Program
FIPS	Federal Information Processing Standards
FISMA	Federal Information Security Management Act
GFE	Government Furnished Equipment
IaaS	Infrastructure-as-a-Service
IAM	Identity and Access Management
IT	Information Technology
JAB	Joint Authorization Board
NIST	National Institute of Standards and Technology
NSOC	Network Security Operations Center
OI&T	Office of Information and Technology
OIG	Office of the Inspector General
OIS	Office of Information Security
PaaS	Platform-as-a-Service
PATO	Provision Authority to Operate
PHI	Protected Health Information
SaaS	Software-as-a-Service
SLA	Service Level Agreement

Acronym	Description
SOA	Service-Oriented Architecture
TIC	Trusted Internet Connection
TRM	Technical Reference Model
VBA	Veteran Benefits Association
VHA	Veteran Health Administration
VIP	Veteran-Centric Integration Process

APPENDIX D: REFERENCES, STANDARDS, AND POLICIES

This Enterprise Design Pattern is aligned to the following VA OI&T references and standards applicable to all new applications being developed in the VA, and are aligned to the VA ETA:

#	Issuing Agency	Applicable Reference/ Standard	Purpose
1	VA	VA Directive 6551	Establishes a mandatory policy for establishing and utilizing Enterprise Design Patterns by all Department of Veterans Affairs (VA) projects developing information technology (IT) systems in accordance with the VA's Office of Information and Technology (OI&T) integrated development and release management process, the Veteran-focused Integration Process (VIP).
2	VA OIS	VA 6500 Handbook	Directive from the OI&T OIS for establishment of an information security program in VA, which applies to all applications that leverage ESS.
3	VA	VA Strategy Lockdown VAIQ#7641464	VA Strategy for Adoption of Cloud Computing (draft)
4	VA IAM	VA Directive 6051	Department of Veterans Affairs Enterprise Architecture (VA EA), July 12, 2002
5	VA	VA Handbook 6517	Risk Management Framework for Cloud Computing Services (draft)
6	NIST	NIST SP 500-291	NIST Cloud Computing Standards Roadmap, Version 2, July 2013
7	NIST	NIST SP 500-292	NIST Cloud Computing Reference Architecture
8	NIST	NIST SP 800-145	The NIST Definition of Cloud Computing, NIST SP 800-145, Sept. 2011

#	Issuing Agency	Applicable Reference/ Standard	Purpose
9	NIST	NIST SP 500-299	NIST Cloud Computing Security Reference Architecture
10	DoD	DoD	Department of Defense Cloud Computing Strategy
11	GSA	GAO 14-753	These challenges were derived from DoD Cloud Computing Strategy and the GAO Report 14-753, "Cloud Computing: Additional Opportunities and Savings Need to Be Pursued," Sept. 2014
12	OMB	OMB M-08-05, Implementation of Trusted Internet Connections (TIC)	Establishes TIC to optimize and standardize the security of external network connections for Federal agencies. Three strategic components:
13	Federal	U.S. CIO, Federal Cloud Computing Strategy	This policy is intended to accelerate the pace at which the Government will realize the value of cloud computing by requiring agencies to evaluate safe, secure cloud computing options before making any new investments.
14	Federal	U.S. CIO, 25 Point Implementation Plan to Reform Federal Information Technology Management	States that the Federal Government will shift to a "Cloud First" policy to better prepare the Government for future computing needs. When evaluating options for new IT deployments, OMB will require agencies to default to cloud-based solutions whenever a secure, reliable, cost-effective cloud option exists.
15	Federal	FIPS 199	FIPS 199 (Federal Information Processing Standard Publication 199)
16	Federal	FIPS 200	Minimum Security Requirements for Federal Information and Information Systems

#	Issuing Agency	Applicable Reference/ Standard	Purpose
17	VA	VA Memorandum Consideration of Open Source Software (VAIQ#7532631)	Establishes requirements to evaluate Open Source Software solutions and consider OSS development practices for VA-developed software.