# TABLE OF CONTENTS

Legal Notice .................................................................................................................................................................................. 3  
Executive Summary ........................................................................................................................................................................ 4  
Purpose ......................................................................................................................................................................................... 5  
Assumptions .................................................................................................................................................................................. 5  
PAAS Interoperability Usage Scenarios ......................................................................................................................................... 6  
Usage Scenarios ............................................................................................................................................................................. 7  
  Usage Scenario 1–Rapid Application On-Boarding .................................................................................................................. 7  
  Usage Scenario 2–Configure PAAS Environment ..................................................................................................................... 8  
  Usage Scenario 3–Application Scalability ............................................................................................................................ 9  
  Usage Scenario 4–Operation and Monitoring ....................................................................................................................... 10  
  Usage Scenario 5–Application Porting ................................................................................................................................ 11  
Usage Requirements ....................................................................................................................................................................... 12  
RFP Requirements –Service Providers .......................................................................................................................................... 13  
Summary of Industry Actions Required ....................................................................................................................................... 14
LEGAL NOTICE

© 2012 Open Data Center Alliance, Inc. ALL RIGHTS RESERVED.

This “Open Data Center AllianceSM Usage Model: Platform as a Service Interoperability” is proprietary to the Open Data Center Alliance, Inc.

NOTICE TO USERS WHO ARE NOT OPEN DATA CENTER ALLIANCE PARTICIPANTS: Non-Open Data Center Alliance Participants only have the right to review, and make reference or cite, this document. Any such references or citations to this document must give the Open Data Center Alliance, Inc. full attribution and must acknowledge the Open Data Center Alliance, Inc.’s copyright in this document. Such users are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend this document in any way.

NOTICE TO USERS WHO ARE OPEN DATA CENTER ALLIANCE PARTICIPANTS: Use of this document by Open Data Center Alliance Participants is subject to the Open Data Center Alliance’s bylaws and its other policies and procedures.

OPEN CENTER DATA ALLIANCESM, ODCA® and the OPEN DATA CENTER ALLIANCE logoSM are service marks owned by Open Data Center Alliance, Inc. and all rights are reserved therein. Unauthorized use is strictly prohibited.

This document and its contents are provided “AS IS” and are to be used subject to all of the limitation set forth herein.

Users of this document should not reference any initial or recommended methodology, metric, requirements, or other criteria that may be contained in this document or in any other document distributed by the Alliance (“Initial Models”) in any way that implies the user and/or its products or services are in compliance with, or have undergone any testing or certification to demonstrate compliance with, any of these Initial Models.

Any proposals or recommendations contained in this document including, without limitation, the scope and content of any proposed methodology, metric, requirements, or other criteria does not mean the Alliance will necessarily be required in the future to develop any certification or compliance or testing programs to verify any future implementation or compliance with such proposals or recommendations.

This document does not grant any user of this document any rights to use any of the Alliance’s trademarks.

All other service marks, trademarks and trade names referenced herein are those of their respective owners.

Published August 2012.
EXECUTIVE SUMMARY

Platform as a Service (PaaS) is a maturing cloud computing segment which enables developers and operations staff to quickly and easily configure and deploy a cloud-ready application of a particular platform or for a particular stack, in a linearly scalable, managed container, including program, database and development tools, and libraries.

Additionally, PaaS provides a managed application hosting environment that removes the need for the cloud subscriber to manage the complexities and details of the underlying infrastructure for an application. Ideally, a properly architected cloud application can be ported seamlessly from one PaaS to another, with the benefits of avoiding vendor lock-in, enabling the use of multiple providers to host the same application globally, and enabling reliable and automatic elasticity for the application.

For the purposes of the ODCA, the definition of PaaS is intentionally constrained to deploying a single application developed for a specific programming language within a hosted application container. Cloud providers may offer support for multiple languages and multi-tenant containers in their PaaS offering. However, each and every hosted language is considered an individual PaaS offering. Also, for the purposes of this document, development, build, testing, and continuous integration capabilities are assumed to be orchestrated outside of the PaaS in order to avoid vendor lock-in.

The ODCA specifically excludes programmable SaaS offerings from being defined as PaaS offerings. A PaaS, as defined by the ODCA, explicitly has no SaaS dependency.

While two perspectives of interoperability, interconnectability and portability, are relevant to PaaS, additional requirements for interconnectability at the application level are firmly in the hands of the application developer. It is recommended that applications developed for PaaS hosting be properly architected per the ODCA Developing Cloud Capable Applications, and be designed to meet the interconnectability requirements of the ODCA SaaS Interoperability Usage Model, where applicable.

This paper outlines five usage scenarios, along with success and failure scenarios for each. Finally, cloud service provider requirements and an industry call to action are presented.

This document serves a variety of audiences. Solution providers and technology vendors will benefit from its content to better understand customer needs, and to tailor service and product offerings. Standards organizations will find the information helpful in defining end-user relevant and open standards.
PURPOSE

The ODCA PaaS Interoperability Usage Model was written to encourage seamless operation of cloud applications across providers, rapid integration with consumer orchestration engines, and automatable configuration and operation of both the PaaS container and the execution of the application itself. This provides the combined benefits of rapid application deployment and linear scalability without the overhead of directly managing the underlying infrastructure for the application, all while avoiding PaaS lock-in.

The business drivers for PaaS Interoperability are as follows:

- **Rapid Application Deployment**: Enable subscribers to quickly deploy new business applications. Reduce the overhead of ongoing application deployments.
- **Application Scalability**: Ability to quickly scale applications up and back based on the real-time demand for those applications.
- **Application Migration**: Ability to move applications from one discrete PaaS to another PaaS available from the same or different cloud provider with minimal effort.
- **Business Continuity**: Migrate or replicate applications among PaaS services to address outages, security breaches, or other disruptions. This is intended to encompass both disaster recovery and disaster avoidance.

ASSUMPTIONS

The following assumptions apply to all usage scenarios described in this document:

- A properly executed service agreement is in place between the PaaS provider and subscriber that identifies capabilities, costs, operational level agreements (OLAs) and service level agreements (SLAs). The result of this is one or more management accounts used to provision, operate, and maintain the consumer’s application within the PaaS environment.

- SLAs and OLAs include items such as:
  - Availability
  - Security, privacy and compliance
  - Geographic hosting requirements
  - Data owner, roles and responsibilities (backup, etc.)

- PaaS is provisioned on a per-application basis, although potentially as part of a large scale application strategy, with a relatively small number of people managing the PaaS environment for each application. Any additional users involved beyond those responsible for managing the execution of the application within the PaaS are considered to be under the purview of the application owner, and are not addressed directly in this document. From here on, the term user refers to the individuals at the PaaS subscriber responsible for managing the operation of the application within the PaaS and the PaaS environment itself.

- Interactions tend to be application specific. It is assumed that any interactions requiring orchestration of multiple applications are the responsibility of the PaaS subscriber.

- Automated provisioning of user identity and access management is in compliance with the practices described in the ODCA Cloud Based Identity Provisioning Usage Model. Access to management of the PaaS conforms to these requirements.

- No modification of the consumer application is required to enable it to execute within the PaaS container.

Out of scope are

- Hosting multiple applications in a PaaS container.
- PaaS offerings requiring proprietary code or code customization for applications to properly run within them.
- Programmable Software as a Service (SaaS) offerings.
**PAAS INTEROPERABILITY USAGE SCENARIOS**

**Interoperability Perspectives**
Interoperability perspectives are the basic categories that define the scope into which the business usages fall. Interoperability perspectives follow:

- **Interconnectability** — the parallel process in which two coexisting environments communicate and interact.
- **Portability** — the serial process of moving a system from one cloud environment to another.

**Usage Scenario Mapping**
For each of the purposes identified in the “Purpose” section of this document, usage scenarios can be identified, aligning to the PaaS Interoperability perspectives. One or more usage scenarios may apply to the purpose within a perspective. Empty cells indicate where an interoperability perspective is not applicable.

The described usage scenarios are:

- Usage Scenario 1: Rapid Application On-Boarding
- Usage Scenario 2: Configure PaaS Environment
- Usage Scenario 3: Application Scalability
- Usage Scenario 4: Operation and Monitoring
- Usage Scenario 5: Application Migration

<table>
<thead>
<tr>
<th>Business Driver</th>
<th>Interconnectability</th>
<th>Portability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rapid Application Deployment</td>
<td>(not applicable)</td>
<td>Usage Scenario 1 - Rapid Application On-Boarding</td>
</tr>
<tr>
<td>Application Hosting</td>
<td>Usage Scenario 4 - Operation and Monitoring</td>
<td>Usage Scenario 3 - Application Scalability</td>
</tr>
<tr>
<td>Application Migration</td>
<td>(not applicable)</td>
<td>Usage Scenario 5 - Application Porting</td>
</tr>
<tr>
<td>Business Continuity</td>
<td>(not applicable)</td>
<td>Usage Scenario 5 - Application Porting</td>
</tr>
</tbody>
</table>
**USAGE SCENARIOS**

**USAGE SCENARIO 1—RAPID APPLICATION ON-BOARDING**

**Business Drivers** Rapid Application Hosting

**Goal:**
Quickly upload and execute a language-appropriate cloud subscriber’s application within the PaaS environment.

**Success Scenario 1:**
A cloud subscriber’s application is uploaded as a single file to the PaaS provider with a well-communicated default configuration for the application to operate within, and becomes operational immediately.

**Steps:**
1. Cloud subscriber requests a new and default PaaS.
2. Provider creates for the subscriber a pre-configured PaaS environment which has the following characteristics:
   a. Well-defined method for accepting an application into the PaaS.
   b. Inbound access to the application on port 80.
   c. Outbound access from the application on all ports and all protocols.
   d. Externally accessible URI for the application, password protection optional.
3. Upon creation of the PaaS environment for the cloud subscriber, the PaaS environment is immediately available for the subscriber to upload an application.
4. The cloud subscriber uploads the application into the PaaS environment via the specified method.
5. The cloud subscriber verifies application accessibility from the external URI.

**Failure Condition 1: Cloud provider unable to accept application.**
If, for any reason, the provider is unable to accept the application into the PaaS, a clear and actionable communication to the cloud subscriber is provided so that they may remediate the situation themselves or involve the provider in solving the problem.

**Failure Condition 2: Cloud provider accepts the application, but the application is inaccessible by the cloud subscriber.**
When the cloud provider has accepted an application into the PaaS without error, and the application has not generated errors, clear and actionable communication to the subscriber is provided so that they may remediate the situation themselves or involve the provider in solving the problem.

**Failure Condition 3: Cloud provider accepts the application, and the application is accessible to the subscriber but is not functional.**
When the provider has accepted an application into the PaaS, and has provided access to the application but the application is still not working, the provider’s PaaS will provide detailed debugging and discovery information to the subscriber to enable self-resolution of the application problem, including, but not limited to:

- Detailed PaaS configuration information
- Platform-specific operational and debugging logs
- Application logs provided or enabled by the specific platform
- Potential configuration abnormalities
Usage Scenario 2—Configure PaaS Environment

Business Drivers: Rapid Application Deployment

Goal:
Specifically configure a single PaaS environment for an application. It is assumed that the ability to configure implies the ability to elicit the PaaS environment’s current configuration.

Assumption 1: Cloud provider provides an interface, either a graphical user interface (GUI) or an application programming interface (API) or both, to change the available configuration options for the PaaS, and which specifies whether the changes will be immediate or applied on the next restart of the application within the PaaS.

Success Scenario 1: Enable cloud subscriber to configure key aspects of the PaaS container:
1. PaaS Container Specific Configuration, as exemplified by, but not limited to:
   a. Tomcat configuration directives
   b. Specification of PHP: Hypertext Processor modules
   c. Application specific extensions to the PaaS container, such as application-specific environment variables the container needs to inherit
2. Inbound access control, as exemplified by, but not limited to:
   a. Source addresses, ports, and protocols exposed to access via the internet
   b. Cloud provider provided identity-based access via the internet
3. Outbound access control, as exemplified by, but not limited to:
   a. Destinations, ports, and protocols the application can access
4. Scaling parameters
   a. By average request time
   b. By request volume
5. Cloud provider specific Infrastructure as a Service (IaaS) and SaaS offerings the application can access

Steps:
1. Cloud subscriber requests PaaS environment configuration for a specific application and requests configuration detail.
2. Cloud subscriber identifies changes to environment that are required.
3. Cloud subscriber changes the configuration of the PaaS. If required, the subscriber restarts the application within the PaaS.
4. Cloud subscriber is able to immediately query the PaaS for the current configuration and is informed of the current state and whether there is a configuration waiting to be applied (and if so, what)
5. Cloud subscriber tests the application to verify the new configuration changes have occurred after taking any necessary actions such as application restart.

Failure Condition 1: Cloud subscriber introduces error into PaaS container configuration.
If the cloud subscriber introduces an error into the configuration files used by the PaaS, the PaaS will provide a clear indication that the error is in a specific PaaS configuration file. Whenever possible, validation and indications of errors in the configuration files before execution by the PaaS is preferred. However, whenever this is not possible, at runtime a clear indication of the error and the location of the configuration file which is in error must be provided. Often, errors will be provided by the language platform involved. These errors must be surfaced in a clear and appropriate manner to the user by the PaaS. For example: the absence of needed PHP modules logged to a well-known file meets this criteria.
Failure Condition 2: Misconfigured outbound access requirements
It is anticipated that, in most cases, the language platform itself will surface outbound access failures through its normal error mechanism. Whatever mechanism is used by language mechanism must be exposed to the cloud subscriber. Optimally, the PaaS environment will also provide a record of outbound access violations to assist the cloud subscriber in properly configuring the PaaS environment.

Failure Condition 3: Misconfigured inbound access requirements
Inbound access errors are impossible for the application to detect. Therefore, the PaaS environment must provide a record of inbound access violations to assist the cloud subscriber in properly configuring the PaaS offering.

Failure Condition 4: Misconfigured access to provider provided IaaS and SaaS offerings
When a PaaS provider is also providing IaaS or SaaS capabilities to an application hosted within its PaaS, any errors associated with accessing those capabilities must be surfaced. The cloud subscriber must have access to application logs as well as a view into any underlying issues with the abstracted software infrastructure.

USAGE SCENARIO 3—APPLICATION SCALABILITY

Business Drivers: Application Hosting

Goal:
PaaS environment automatically scales resources up and down based on required load.

Assumption 1: Scaling is within the metered limits of the subscription.

Assumption 2: PaaS provider publishes a well understood baseline representation of performance for the PaaS. The expectation is that a known, sample workload produces predictable performance which scales linearly as the application scales.

Success Scenario 1:
A subscriber’s application running within the PaaS experiences consistent performance based on one of two scalability parameters: average request latency and/or request transaction volume. Request latency refers to the application response time. Request transaction volume is the ability to meet a specified request volume (# http requests/second). Resources are applied and removed appropriately by the provider to meet the demand and are billed appropriately.

Steps:
1. Cloud subscriber configures and runs an application within a PaaS instance.
2. Increases in load are handled seamlessly by the cloud provider in order to meet the configured scalability requirements of the application.
3. Decreases in load result in a well-understood tapering of resources applied to the PaaS while still meeting the configured scalability requirements of the application.
4. The cloud subscriber may access details, including billing, of the current scaling of their application within the PaaS.

Failure Condition 1: Insufficient resources to meet PaaS scalability requirements
If, for any reason, the cloud provider is unable to assign resources to meet the scalability requirements of the application, the cloud subscriber will be notified pursuant to Usage Scenario 4 (PaaS Operation and Monitoring).

Ideally, the application request load is throttled. The cloud subscriber is provided with a throttle notification instead of failing requests. The cloud subscriber is notified ahead of time based on proactive notification of the limits of the cloud service provider. The PaaS contract outlines details and remediation plans for these situations.

Failure Condition 2: Request latency unacceptable despite resources applied
If request latency constraints cannot be met despite the addition of a provider-defined amount of resources, the cloud subscriber will
be notified pursuant to Usage Scenario 4 (PaaS Operation and Monitoring). In most cases, this may be due to an issue with the cloud subscriber's application versus the performance of the resources applied by the cloud provider.

**Failure Condition 3: Request volume unattainable**
If the PaaS is configured to scale based on request volume and, for any reason, the request volume is not being met by the PaaS, then the cloud subscriber will be notified pursuant to Usage Scenario 4 (PaaS Operation and Monitoring) below.

### USAGE SCENARIO 4—OPERATION AND MONITORING

**Business Drivers:** Application Hosting

**Goal:**
Manage and monitor the execution of an application within the PaaS

**Assumption 1:** Cloud provider supplies a GUI or API for managing the execution of the application within the PaaS.

**Assumption 2:** Cloud provider provides a GUI or API for obtaining the current activity level and billing information for the PaaS.

**Assumption 3:** Cloud provider provides a GUI or API to the cloud subscriber for registering PaaS events from the provider.

**Success Scenario 1:**
A cloud subscriber can easily stop, start, and restart the application within the PaaS.

**Steps:**
1. Cloud subscriber configures and runs an application within a PaaS instance.
2. Cloud subscriber uses a GUI or API to stop, start, restart the application with the expected result occurring.
3. Stop and Restart actions reset the local application state.

**Failure Condition 1: Inability to start, stop, or restart the application due to problem with PaaS**
If, for any reason, the cloud provider is unable to stop, start, or restart the application as requested by the cloud subscriber, the subscriber and the provider will be notified, and the provider will immediately take action to resolve in accordance with the applicable SLA to the subscriber.

**Failure Condition 2: Inability to start, stop, or restart the application due to application error**
If the cloud provider is unable to stop, start, or restart the application due to an error with the cloud subscriber's application, the subscriber will be notified, and sufficient detail about the problem provided to enable the subscriber to resolve the problem with their application.

**Success Scenario 2:**
The cloud subscriber is able to monitor the activity within the PaaS for performance and analysis purposes.

**Steps:**
1. Cloud subscriber configures and runs an application within a PaaS instance.
2. Provider's PaaS implementation records, at a minimum:
   a. HTTP transaction history (i.e. web logs)
   b. HTTP server error logs
c. Platform specific error logs (i.e., PHP specific error logs)

   d. Performance information:
      i. HTTP requests per second being handled by the PaaS. This is an instantaneous number at the time the information is requested.

3. Cloud subscriber uses GUI or API to obtain PaaS activity.

**Failure Condition 1: Inability to access monitoring and performance information**

If, for any reason, the provider is unable to provide activity and billing information for the PaaS, the cloud subscriber will be notified, and the cloud provider will take action to remedy the lack of access pursuant to the SLA to the subscriber.

**Success Scenario 3:**

The cloud subscriber receives alerts regarding PaaS activity.

**Steps:**

1. Cloud subscriber configures and runs an application within a PaaS instance.

2. Cloud subscriber receives alerts regarding PaaS activity in a timely manner. Ideally the alerts are made available to the subscriber’s enterprise monitoring solution. Examples of alerts include, but are not limited to:
   a. Stop, Start, and Restart notifications for the PaaS.
   b. Scaling up and scaling down events for the PaaS, based on a well-understood threshold.
   c. Errors occurring within the PaaS.
   d. Errors occurring within the application.

3. Cloud subscriber takes appropriate action based on alert type.

4. Cloud provider takes appropriate action based on alert type.

**Failure Condition 1: Alerts are not received by cloud subscriber.**

If the cloud provider is unable to deliver alerts to the cloud subscriber, a notification will be made to the subscriber and appropriate steps taken to remediate the issue per the applicable SLA. Once delivery is restored, all previously undelivered alerts will be delivered to the cloud subscriber.

**USAGE SCENARIO 5 — APPLICATION PORTING**

**Business Drivers:** Application Migration, Business Continuity

**Goal:**

Enable the consistent and reliable deployment of an application across multiple providers.

**Assumption 1:** Data associated with the application is out of scope for this scenario because the assumption is that a data as a service or database as a service solution is in effect.

**Success Scenario 1:**

A cloud subscriber can easily deploy an application and its configuration in another PaaS of the same type and expect it to execute in the same manner regardless of the cloud provider.
Steps:
1. Cloud subscriber configures and deploys an application to a provider. The application works as expected.
2. Cloud subscriber configures and deploys the same application in the same PaaS type provided by different cloud provider. The application works as expected.

Failure Condition 1:
Code is uploaded and does not run.

USAGE REQUIREMENTS
The features below are derived from the usage scenarios in the previous section, and are aligned with the ODCA Standard Units of Measure for IaaS. For this usage model, it is not intended that all of the features in a given column must be supported as a group. In practice, a given cloud service provider solution will combine different service levels for different elements. For example, all Bronze Performance feature requirements must first be met before combining features from any other performance level. For instance, Gold Security features can be combined with Bronze.

<table>
<thead>
<tr>
<th></th>
<th>Bronze</th>
<th>Silver</th>
<th>Gold</th>
<th>Platinum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Basic</td>
<td>Enterprise equivalent</td>
<td>Critical market or business sector equivalent</td>
<td>Military or safety-critical equivalent</td>
</tr>
<tr>
<td>Security</td>
<td>As per ODCA Provider Assurance Usage Model¹</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formal Documentation</td>
<td>Online documentation for all service interfaces, GUIs, and command lines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>App Deployment Mechanism</td>
<td>Upload via PaaS provider utilities</td>
<td>Upload via PaaS provider utilities with defined SLAs</td>
<td>Upload via PaaS provider utilities or APIs with defined SLAs and interface versioning</td>
<td></td>
</tr>
<tr>
<td>App Stack Configuration Process</td>
<td>Set and view settings via CLI or GUI</td>
<td>Set and view settings via CLI or GUI</td>
<td>Set and view settings via CLI, GUI or API</td>
<td>Set and view settings via CLI, GUI or API with versioning; with rollback</td>
</tr>
<tr>
<td>App Stack Configuration Scope</td>
<td>Standard default configurations with some developer defined settings</td>
<td>Flexible developer defined settings for wide range of capabilities</td>
<td>Flexible developer defined settings for wide range of capabilities</td>
<td>Flexible developer defined settings for wide range of capabilities</td>
</tr>
<tr>
<td>App Scaling</td>
<td>Manually scale application by setting static resource limits (i.e., instances)</td>
<td>Set resource limits for application via CLI or GUI; Automatically scales to defined resource limits</td>
<td>Automated scale out based on usage and SLA; Scale limiting based on economics</td>
<td>Automated scale out based on usage and SLA; Scale limiting based on economics</td>
</tr>
</tbody>
</table>

¹ODCA Provider Assurance Usage Model
### App Monitoring

<table>
<thead>
<tr>
<th>Bronze</th>
<th>Silver</th>
<th>Gold</th>
<th>Platinum</th>
</tr>
</thead>
<tbody>
<tr>
<td>View app health, usage, billing, logs via CLI or GUI per app</td>
<td>View app health, usage, billing, logs via CLI or GUI or API per app. Alerts for problems and reached limits provided.</td>
<td>View app health, usage, billing, logs via CLI or GUI or API per app. Alerts for problems and reached limits provided. Usage trends provided.</td>
<td>Use an industry standard, versioned API to view app health, analytics, logs; Alerts for problems and reached limits provided. Usage trends provided</td>
</tr>
</tbody>
</table>

### App Availability

<table>
<thead>
<tr>
<th>Bronze</th>
<th>Silver</th>
<th>Gold</th>
<th>Platinum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Effort</td>
<td>99.9% uptime</td>
<td>99.99% uptime</td>
<td>99.999% uptime</td>
</tr>
</tbody>
</table>

### Web Services Interface Standard

<table>
<thead>
<tr>
<th>Bronze</th>
<th>Silver</th>
<th>Gold</th>
<th>Platinum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmatic web services in cloud provider’s choice of standard</td>
<td>Programmatic web services in cloud provider’s choice of standard</td>
<td>Programmatic web services in recognized industry standard, consistent with ODCA concepts</td>
<td>Programmatic web services in recognized industry standard, consistent with ODCA concepts</td>
</tr>
</tbody>
</table>

### RFP REQUIREMENTS — SERVICE PROVIDERS

Following are requirements that the Alliance believes should be included in requests for proposal to cloud providers to ensure that proposed services support PaaS Interoperability.

**ODCA Principle Requirement**—Service is open and standards-based. Describe how the service meets this principle as well as any limitations towards the ODCA principle.

**ODCA PaaS Interoperability Usage Model 1.0**—Cloud subscriber application should not be forced to access platform capabilities via provider supplied APIs (because this causes lock-in). PaaS shouldn’t force developers to change their applications and behaviors.

**ODCA PaaS Interoperability Usage Model 1.0**—The PaaS provides well documented, platform level SLAs.

**ODCA PaaS Interoperability Usage Model 1.0**—PaaS provider publishes a well understood baseline representation of performance for the PaaS.

**ODCA PaaS Interoperability Usage Model 1.0**—PaaS solution should provide capabilities to support repeatable software development business processes, workflow and approval cycles, with built-in reporting.

**ODCA PaaS Interoperability Usage Model 1.0**—PaaS will support ODCA and industry standards such as Java, MySQL, Apache, and REST.

**ODCA PaaS Interoperability Usage Model 1.0**—PaaS should support app scaling scenarios with defined triggers and alerts in accordance with the contracted agreements and subscriber-configured boundaries.

**ODCA PaaS Interoperability Usage Model 1.0**—Predictable performance profile and billing is based on specified sample workload.

**ODCA PaaS Interoperability Usage Model 1.0**—Upon nearing or hitting configured resource or economic limits, application requests are throttled in a manner that provides notification to the cloud subscriber with details and remediation plan.

Click [here](#) for an online assistant, Proposed Engine Assistant Tool (PEAT)⁶ to help you detail your RFP requirements.

© 2012 Open Data Center Alliance, Inc. ALL RIGHTS RESERVED.
SUMMARY OF INDUSTRY ACTIONS REQUIRED

In the interest of giving guidance on how to create and deploy solutions that are open, multi-vendor and interoperable, we have identified specific areas where the Alliance believes there should be open specifications, formal or de facto standards, or common intellectual property-free (IP-free) implementations. Where the Alliance has a specific recommendation on the specification, standard, or open implementation, it is called out in this usage model. In other cases, we will be working within the industry to evaluate and recommend specifications in future releases of this document.

The following are industry actions required to refine this usage model:

1. Adopt the ODCA definition of PaaS in order to maintain the highest level of portability and interconnectibility among cloud providers and subscribers.

2. Continue to mature PaaS offerings to global enterprise levels:
   a. Facilitate the ability to “drop in” applications and make them rapidly available globally.
   b. Enable existing software development and deployment practices as opposed to creating provider specific practices.
   c. Integrate PaaS offerings with existing industry standard developer toolsets as opposed to creating cloud provider specific toolsets.

3. Develop Industry standard PaaS interchange formats for exporting and importing PaaS configurations.

For additional information, see ODCA Developing Cloud Capable Applications¹ best practices paper.

¹ ODCA Developing Cloud Capable Applications: http://www.opendatacenteralliance.org/docs/Best_Practices_whitepaper.pdf
² ODCA SaaS Interoperability Usage Model: www.opendatacenteralliance.org/docs/ODCA_SaaS_Interop_UM_Rev1.0.pdf
³ ODCA Cloud Based Identity Provisioning Usage Model: http://www.opendatacenteralliance.org/docs/Cloud_Based_Identity_Provisioning_%20b.pdf
⁵ ODCA Provider Assurance Usage Model: http://www.opendatacenteralliance.org/docs/ODCA_ProviderAssurance_Rev%201.1_Final.pdf
⁶ Proposed Engine Assistant Tool: http://www.opendatacenteralliance.org/ourwork/proposalengineassistant