MO-003d — Limitation of Provider-Specific Dependencies

Go to MO-003 — Cloud-Agnostic Deployment

Statement

Crucible SHALL confine dependencies on proprietary Cloud Provider interfaces, services, data formats, software development kits, and operational mechanisms to the Provider Implementation for the target Deployment Platform.

Derived From

This requirement derives from:

- MO-003 — Cloud-Agnostic Deployment

The Original Requirement states:

> *The system SHALL provide cloud-agnostic deployment capabilities.*[C2]

MO-003d preserves the portion of the Original Requirement that limits dependence on proprietary Cloud Provider mechanisms.

The separate requirements derived from MO-003 address:

- MO-003a — Cross-Platform Deployment
- MO-003b — Provider-Independent Crucible Description
- MO-003c — Provider-Independent Infrastructure Baseline

Rationale

Cloud Agnosticism requires Crucible to limit dependence on the proprietary mechanisms of a particular Cloud Provider.

Provider-specific dependencies can include:

- Proprietary application programming interfaces
- Proprietary services
- Proprietary data formats
- Provider-specific software development kits
- Provider-specific command-line tools
- Provider-specific authentication mechanisms

- Provider-specific resource identifiers
- Provider-specific configuration structures
- Provider-specific deployment workflows
- Provider-specific operational procedures

Allowing these dependencies within provider-independent Crucible components would couple those components to one or more Cloud Providers.

The Provider Implementation defines the boundary within which provider-specific dependencies reside. The Provider Implementation translates provider-independent deployment information into the mechanisms supported by the target Deployment Platform.

Confining provider-specific dependencies supports:

- Substitution of Provider Implementations
- Addition of new Deployment Platforms
- Reuse of provider-independent Crucible components
- Controlled migration between Cloud Providers
- Detection of provider-specific implementation leakage
- Reduced provider coupling
- Independent testing of provider-specific behavior
- Traceability

This requirement does not prohibit provider-specific dependencies. It confines those dependencies to the Provider Implementation associated with the target Deployment Platform.

Applies To

This requirement applies to:

- Crucible
- Cloud Agnosticism
- Provider Implementations
- Provider Contracts
- Provider Abstraction
- Platform Independent Information
- Platform Specific Information
- Platform Specific Parameters
- Deployment Platforms
- Provider-independent Crucible components
- Proprietary Cloud Provider interfaces
- Proprietary Cloud Provider services
- Proprietary Cloud Provider data formats
- Provider-specific software development kits
- Provider-specific operational mechanisms
- Evidence
- Provenance
- Traceability

Verification

Verification confirms that:

1. Each tested Deployment Platform has an identified Provider Implementation
2. Each Provider Implementation has a defined architectural boundary
3. Proprietary Cloud Provider interfaces occur only within the Provider Implementation for the target Deployment Platform
4. Proprietary Cloud Provider services are accessed only through the Provider Implementation for the target Deployment Platform
5. Proprietary Cloud Provider data formats occur only within the Provider Implementation or at its defined external boundary
6. Provider-specific software development kits are referenced only by the Provider Implementation for the target Deployment Platform
7. Provider-specific operational mechanisms occur only within the Provider Implementation for the target Deployment Platform
8. Provider-independent Crucible components do not contain dependencies on proprietary Cloud Provider mechanisms
9. Each provider-specific dependency detected outside the permitted boundary is recorded as a finding
10. The verification record preserves Traceability among the target Deployment Platform, Provider Implementation, identified provider-specific dependencies, evaluated Crucible components, and verification result

Referenced By

The following pages reference this requirement:

- 3.2 Deployment Environments
- 4.3 External Systems
- MO-003a — Cross-Platform Deployment
- MO-003b — Provider-Independent Crucible Description
- MO-003c — Provider-Independent Infrastructure Baseline
- Annex D: Requirements Traceability

Delivery Phase

Implemented and Verified

Implementation Status

☐ Assess whether the current Crucible implementation confines proprietary Cloud Provider dependencies to the Provider Implementation for each target Deployment Platform.

Requirement Status

☐ Review and accept MO-003d as a proposed derived requirement created from the evaluation and decomposition of MO-003 in the Crucible System Requirements Specification, Version 1.1 Draft.

Issues

The following unresolved issues affect this requirement:

☐ Define the architectural boundary of each Provider Implementation.

☐ Identify the provider-independent Crucible components that must remain free of proprietary Cloud Provider dependencies.

☐ Identify the proprietary interfaces, services, data formats, software development kits, and operational mechanisms permitted within each Provider Implementation.

☐ Define whether proprietary Cloud Provider data formats may cross the Provider Implementation boundary and, if permitted, define the boundary conditions.

☐ Identify the controlling source that associates each Provider Implementation with its target Deployment Platform.

Notes for Editors

This requirement page should retain the stable requirement identifier `MO-003d`.

This page is a leaf requirement page and omits a trailing `:start` from its namespace.

The parent MO-003 page is a non-leaf page and retains a trailing `:start` in its namespace.

Changes to the Statement should preserve:

- Crucible as the responsible actor
- Proprietary Cloud Provider interfaces, services, data formats, software development kits, and operational mechanisms as the dependencies subject to confinement
- The Provider Implementation as the permitted location for provider-specific dependencies
- The target Deployment Platform as the basis for selecting the Provider Implementation
- Separation of provider-independent Crucible components from provider-specific implementation behavior

The architectural definition of each Provider Implementation boundary should identify:

- The target Cloud Provider or Deployment Platform

- The Provider Contract
- The provider-independent interfaces exposed to Crucible
- The proprietary interfaces used by the Provider Implementation
- The proprietary services used by the Provider Implementation
- The proprietary data formats used by the Provider Implementation
- The provider-specific software development kits used by the Provider Implementation
- The provider-specific operational mechanisms used by the Provider Implementation
- The permitted external dependencies
- The prohibited dependencies outside the boundary

Material changes should receive review and should update the verification criteria, source records, and Issues section.

To reference this requirement Statement from another wiki page, insert:

```
{{section>dido:02-crusible:99-annexes:annex-c-requirements:01-mission-objectives:mo-003:mo-003d#Statement&noheader&nofooter&noeditbtn}}
```

Do not rename this page after an external citation unless a redirect or move plan is in place.

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.

From:
https://wiki.didosolutions.com/ - **Dido Solutions, Inc Wiki**

Permanent link:
https://wiki.didosolutions.com/dido/02-crusible/99-annexes/annex-c-requirements/01-mission-objectives/mo-003/mo-003d

Last update: **2026/07/30 05:23**