MO-003d — Provider-Specific Dependency Limitation

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 applicable Provider Implementation.

Derived From

This requirement derives from:

- MO-003 — Cloud-Agnostic Deployment

The Original Requirement states:

> *The system SHALL provide cloud-agnostic deployment capabilities.*[C3]

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 — Common Crucible Description
- MO-003c — Common Infrastructure Baseline

Assessment

The Original Requirement uses the defined concept Cloud Agnosticism but does not identify:

- The proprietary Cloud Provider dependencies subject to limitation
- The Crucible components permitted to contain provider-specific dependencies
- The architectural boundary between provider-independent behavior and provider-specific behavior
- The mechanism used to isolate provider-specific dependencies
- The criteria used to detect provider-specific dependencies outside the permitted boundary

MO-003d:

- Identifies proprietary Cloud Provider interfaces, services, data formats, software development kits, and operational mechanisms as the dependencies subject to limitation

- Identifies the applicable Provider Implementation as the permitted location for those dependencies
- Establishes a defined boundary between provider-independent Crucible behavior and provider-specific implementation behavior
- Prevents provider-specific dependencies from spreading into provider-independent Crucible components
- Separates provider-specific dependency limitation from cross-platform deployment and common deployment-input requirements

Rationale

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

Provider-specific dependencies 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 the core architecture to one or more Cloud Providers.

The applicable Provider Implementation provides the controlled boundary within which provider-specific dependencies reside. The Provider Implementation translates provider-independent deployment intent into the proprietary mechanisms required by its target Cloud Provider or 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
- Consistent Provider Contract enforcement
- Independent testing of provider-specific behavior

This requirement does not prohibit provider-specific dependencies. It limits those dependencies to the applicable Provider Implementation.

This requirement does not establish the Provider Contract requirements or the permitted Platform

Specific Parameters. Lower-level functional and interoperability requirements establish those obligations.

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
- Crucible Descriptions
- Infrastructure Baselines
- Infrastructure Deployments
- 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 Cloud Provider or 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 applicable Provider Implementation
4. Proprietary Cloud Provider services are accessed only through the applicable Provider Implementation
5. Proprietary Cloud Provider data formats occur only within the applicable Provider Implementation or at its defined external boundary
6. Provider-specific software development kits are referenced only by the applicable Provider Implementation
7. Provider-specific operational mechanisms occur only within the applicable Provider Implementation
8. Provider-independent Crucible components do not contain dependencies on proprietary Cloud Provider mechanisms
9. Each detected provider-specific dependency outside the permitted boundary is recorded as a finding

10. The generated Evidence supports the dependency-limitation determination

Verification includes:

- Source-code dependency analysis
- Build dependency analysis
- Package dependency inspection
- Import and module inspection
- Software development kit reference inspection
- Application programming interface usage inspection
- Service endpoint inspection
- Data-format inspection
- Configuration inspection
- Command-line invocation inspection
- Runtime dependency analysis
- Network interaction inspection
- Provider Implementation boundary inspection
- Provider Contract inspection
- Static analysis
- Dynamic analysis
- Cross-provider substitution tests
- Finding inspection
- Evidence inspection
- Provenance inspection
- Traceability inspection

The verification record identifies:

1. Each tested Cloud Provider or Deployment Platform
2. The applicable Provider Implementation
3. The Provider Implementation boundary
4. The applicable Provider Contract
5. Each proprietary interface used
6. Each proprietary service used
7. Each proprietary data format used
8. Each provider-specific software development kit used
9. Each provider-specific operational mechanism used
10. The Crucible component containing each identified dependency
11. Each provider-specific dependency detected outside the permitted boundary
12. The result of each dependency analysis
13. The observed result
14. The generated Evidence

Requirements Realized By

This requirement is realized by:

- OR-002 — Deployment Environments
- OR-005 — Distributed Environment Management

- FR-DEP-001 through FR-DEP-007
- FR-MC-001 through FR-MC-005
- FR-DSO-001 through FR-DSO-006
- FR-BAS-001 through FR-BAS-004
- INT-001 through INT-007
- KVP-006 — Enable Cloud Portability

Related Architecture Sections

- 5. Descriptions, Composition, and Baselines
- 7. Infrastructure and Deployment
- 8. Dependencies and Air Gap Operations
- 10. Reproducibility, Provenance, and Traceability

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

Phase 1 and subsequent phases

Implementation Status

Not Assessed

Implementation status requires verification that proprietary Cloud Provider dependencies are confined to the applicable Provider Implementation.

Requirement Status

Draft

This requirement derives from MO-003 in the Crucible System Requirements Specification, Version 1.1 Draft.

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.

Changes to the Statement SHALL preserve the provider-specific dependency-limitation intent derived from MO-003.

The architectural definition of each Provider Implementation boundary should identify:

- The target Cloud Provider or Deployment Platform
- The applicable 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
- The required Evidence

Material changes should receive review and should update the related verification criteria, requirements realization, related architecture sections, and source records.

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.