

FR-MC-002 — Provider-Specific Extensions

[Go to Crucible Multi-Cloud Management Requirements](#)

Statement

Crucible SHALL load provider-specific deployment behavior through a plugin that conforms to the applicable provider contract.

Source Statement

The system shall support provider-specific extensions through plugins.

Source

Crucible System Requirements Specification, Version 1.1 Draft, Functional Requirements, FR-MC-002.

Assessment

The source statement identifies the intended extension mechanism but uses a weak primary verb.

The following Specification Discipline and Authoring findings apply:

- **The system** does not use the defined system name
- **shall** does not follow the established uppercase normative convention
- **Support** is a weak verb
- **Provider-specific extensions** does not identify the behavior supplied by the extension
- **Plugins** identifies the extension form but does not identify the contract governing the plugin
- The source statement does not prescribe a programming language, packaging format, loading mechanism, provider, or plugin implementation

The normalized Statement:

- Replaces **The system** with **Crucible**
- Replaces **support** with the direct behavior **load**
- Identifies provider-specific deployment behavior
- Retains the required plugin mechanism
- Requires conformance to the applicable provider contract
- Preserves implementation independence outside the source-mandated plugin constraint

The term **Plugin** and the project-specific concept **Provider Contract** should receive glossary definitions if they recur elsewhere.

Rationale

A cloud provider may expose capabilities or parameters that do not have a common representation across all providers.

A provider-specific plugin may implement:

- Resource-type mappings
- Provider API interactions
- Authentication behavior
- Region selection
- Network configuration
- Storage configuration
- Machine-type selection
- Provider-managed services
- Provider-specific validation
- Provider-specific error handling

The provider contract defines the behavior and information exchange required between Crucible and the plugin.

Provider-specific plugins preserve separation between:

- Platform-independent Crucible behavior
- Common deployment intent
- Provider-specific deployment behavior
- Provider-specific parameters

Applies To

This requirement applies to:

- [Crucible](#)
- Provider-specific plugins
- Provider contracts
- Provider-specific deployment behavior
- Provider-specific parameters
- [Provider Implementations](#)
- Cloud providers
- Deployment workflows

Verification

1. Verification SHALL confirm that the tested provider-specific deployment behavior resides in a plugin
2. Verification SHALL confirm that [Crucible](#) loads the tested plugin
3. Verification SHALL confirm that the tested plugin conforms to the applicable provider contract

4. Verification SHALL confirm that Crucible invokes the provider-specific deployment behavior through the applicable provider contract
5. Verification SHALL confirm that provider-specific behavior does not require modification of unrelated provider-independent Crucible components

Verification may include:

- Plugin-loading tests
- Provider-contract tests
- Provider implementation tests
- Interface conformance inspection
- Static dependency analysis
- Provider deployment tests
- Plugin replacement tests

The verification record SHALL identify:

1. The tested plugin
2. The applicable provider
3. The applicable provider contract
4. The provider-specific behavior
5. The plugin conformance result
6. The deployment result
7. The generated [Evidence](#)

Outgoing Traceability

This requirement realizes:

- [MO-003](#)
- [MO-006](#)

This requirement relates to:

- [FR-MC-001 — Cloud Provider Abstraction](#)
- [FR-MC-003 — Deployment Portability Across Cloud Providers](#)
- [7. Infrastructure and Deployment](#)

Referenced By

The wiki Backlinks function provides the current list of pages that reference FR-MC-002.

Incoming Traceability should be derived dynamically from backlinks.

ConOps Relationship

The Crucible Concept of Operations permits provider-specific realization while preserving common

deployment intent.

FR-MC-002 establishes the plugin boundary through which Crucible loads provider-specific deployment behavior.

Delivery Phase

Phase 1

Implementation Status

Not Assessed

Requirement Status

Draft

Notes for Editors

This requirement page should retain the stable requirement identifier FR-MC-002.

The source explicitly requires plugins. Changes should not generalize the extension mechanism without a corresponding change to the controlling requirement.

Requirements for plugin discovery, installation, activation, version compatibility, isolation, signing, and removal should remain separate unless added by the controlling source.

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

```
{{section>dido:02-crusible:99-annexes:annex-c-requirements:03-functional-requirements:03-04-multi-cloud-management:fr-mc-002#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/03-functional-requirements/03-04-multi-cloud-management/fr-mc-002?rev=1784309015>

Last update: 2026/07/17 10:23

