FR-MC-002 — Provider-Specific Extensions

Go to Crucible Multi-Cloud Management Requirements

Statement

Crucible SHALL load a Provider 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 does not provide a fully testable formulation.

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 that does not identify the required behavior
- **Provider-specific extensions** does not identify the architectural subject that realizes the extension
- **Plugins** does not use the defined Provider Plugin concept
- The source statement does not identify the contract governing the Provider 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**
- Uses the defined Provider Plugin concept
- Requires conformance to the applicable Provider Contract
- Retains the source-required plugin extension mechanism
- Preserves implementation independence outside the source-mandated plugin boundary
- Retains one primary required behavior

The normalized Statement does not independently require Provider Plugin discovery, installation,

activation, replacement, isolation, signing, version negotiation, or removal.

Rationale

A Provider may expose capabilities, interfaces, parameters, and lifecycle operations that differ from those exposed by another Provider.

A Provider Plugin isolates those provider-specific details from the core Crucible behavior.

A Provider Plugin may implement:

- Resource type mappings
- Provider interface interactions
- Authentication behavior
- Region selection
- Network configuration
- Storage configuration
- Machine type selection
- Provider-managed services
- Image conversion
- Image upload
- Image signing
- Image verification
- Image promotion
- Infrastructure initialization
- Deployment operations
- Teardown operations
- Provider-specific validation
- Provider-specific error handling

The applicable Provider Contract specifies the capabilities, operations, inputs, outputs, constraints, errors, compatibility rules, and lifecycle behaviors that govern interaction with the Provider.

The Provider Abstraction defines the provider-independent architectural boundary. The Provider Contract formalizes that boundary, and the Provider Plugin realizes the contract for a particular Provider or class of Providers.

Provider Plugins preserve separation among:

- Provider-independent Crucible behavior
- Common deployment intent
- The Provider Abstraction
- Provider-specific deployment behavior
- Provider-specific parameters
- Provider interfaces and software development kits

Applies To

This requirement applies to:

- Crucible
- Provider Plugins
- Provider Contracts
- Provider Abstractions
- Providers
- Provider-specific deployment behavior
- Provider-specific parameters
- Provider interfaces
- Provider deployment workflows

Verification

1. Verification SHALL confirm that the tested provider-specific behavior resides in a Provider Plugin
2. Verification SHALL confirm that Crucible loads the tested Provider Plugin
3. Verification SHALL confirm that the tested Provider Plugin conforms to the applicable Provider Contract

Verification may include:

- Provider Plugin loading tests
- Provider Contract conformance tests
- Provider interface tests
- Interface conformance inspection
- Static dependency analysis
- Provider deployment tests
- Provider Plugin substitution tests

The verification record SHALL identify:

1. The tested Provider Plugin
2. The applicable Provider
3. The applicable Provider Contract
4. The Provider Plugin loading operation
5. The Provider Contract conformance result
6. The observed 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 rather than maintained as a duplicate manual list.

Backlinks identify incoming references but do not define the semantics of each relationship. Referencing pages should identify whether the relationship represents realization, refinement, verification, dependency, or another defined traceability relationship.

ConOps Relationship

The Crucible Concept of Operations permits provider-specific realization while preserving provider-independent deployment intent.

FR-MC-002 establishes the extension boundary through which Crucible loads a Provider Plugin that conforms to the applicable Provider Contract.

The Provider Abstraction defines the provider-independent architectural boundary, while the Provider Plugin implements provider-specific behavior for a particular Provider or class of Providers.

Delivery Phase

Phase 1

Implementation Status

Not Assessed

Implementation status requires verification that Crucible loads a Provider Plugin that conforms to the applicable Provider Contract.

Requirement Status

Draft

The source System Requirements Specification identifies Version 1.1 as a draft.

Notes for Editors

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

Changes to the Statement SHALL preserve the approved intent of the source requirement.

The Source Statement should preserve the original wording from the controlling System Requirements Specification.

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

The Statement should remain limited to loading a Provider Plugin that conforms to the applicable Provider Contract.

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

Verification criteria should test only the behavior stated in the normalized Statement and should not introduce additional normative obligations.

Incoming Traceability should use the wiki Backlinks function rather than a manually maintained list.

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.