

Provider Contract

[Go up to Terms and Definitions](#)

Discussion

A Provider Contract specifies the capabilities, operations, inputs, outputs, constraints, errors, compatibility rules, and lifecycle behaviors that govern interaction with a [Provider](#) through a [Provider Abstraction](#).

The contract establishes the obligations that a conforming [Provider Plugin](#) must satisfy and the assumptions that the core software may rely on.

A Provider Contract may define required and optional capabilities, version compatibility, capability discovery, validation rules, return values, error handling, lifecycle operations, and conformance expectations.

Within the Crucible architecture, the Provider Contract forms the principal cloud-agnostic seam between core orchestration functions and provider-specific implementations.

Definition

specification of the capabilities, operations, inputs, outputs, constraints, errors, compatibility rules, and lifecycle behaviors governing interaction with a [provider](#)

Source

Generalized from software interface contracts, plugin architectures, service contracts, conformance specifications, and provider-integration usage and specialized for the Crucible architecture and operational model.

Note

A Provider Contract defines architectural and behavioral obligations. It does not necessarily constitute a commercial, legal, or procurement contract.

Versioning the Provider Contract allows the architecture to evolve while preserving explicit compatibility expectations.

Example

A Provider Contract specifies that a Provider Plugin accepts a validated image-upload request, returns the resulting provider image identifier, reports defined failure conditions, and supports image description and verification operations.

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

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

Permanent link:
https://wiki.didosolutions.com/dido/99_annexes/annex-b-terms-and-definitions/p/provider_contract

Last update: **2026/07/13 11:10**

