

Cloud Agnosticism

[Go up to Terms and Definitions](#)

Discussion

Cloud Agnosticism is an architectural characteristic that limits dependence on the proprietary interfaces, services, data formats, software development kits, or operational mechanisms of a particular [Cloud Provider](#).

A cloud-agnostic architecture expresses common capabilities through provider-independent concepts, contracts, and interfaces. Provider-specific implementations translate those common capabilities into the mechanisms required by each Cloud Provider.

Cloud Agnosticism does not require every Cloud Provider to offer identical capabilities. The architecture may identify optional capabilities, provider-specific constraints, or controlled extensions without allowing provider-specific logic to spread throughout the core system.

[Cloud Agnosticism](#) can support [Portability](#) by reducing dependence on a particular Cloud Provider. Cloud Agnosticism does not, by itself, establish Portability, because movement to another platform, runtime, deployment environment, or operational context may require adaptation, verification, requalification, or additional evidence.

Definition

architectural characteristic that limits dependence on the proprietary interfaces, services, data formats, software development kits, or operational mechanisms of a particular [cloud provider](#)

Source

Generalized from cloud computing, software portability, platform independence, provider abstraction, and multi-cloud architecture usage and specialized for the Crucible architecture and operational model.

Note

Cloud Agnosticism does not mean that every deployment can move between Cloud Providers without adaptation. Provider capabilities, service availability, identity models, networking, data services, and regulatory constraints may affect portability.

Cloud Agnosticism concerns architectural dependence. [Portability](#) concerns the ability to transfer or adapt a system, artefact, or workload between environments.

Example

Crucible requests image upload and infrastructure deployment through a Provider Contract. Separate Provider Plugins implement those operations for Oracle Cloud Infrastructure and Amazon Web Services without placing cloud-specific logic in the Crucible core.

© 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/c/cloud_agnosticism?rev=1783966739

Last update: **2026/07/13 11:18**

