

11.3 Conceptual Architecture Does Not Prescribe Realization Mechanisms

[Go to Top](#)

The Conceptual Architecture defines architectural responsibilities and relationships. It does not prescribe execution granularity, deployment granularity, communication technology, interface style, [Data](#) serialisation format, or runtime packaging.

A later part may realize a [Node](#) as a subroutine, library component, thread, task, process, service, container, pod, virtual machine, serverless function, RPC endpoint, gateway, external subscriber, or other executable or addressable participant. Those realisations do not redefine the concept of [Node](#).

A later part may realize a [Communication Endpoint](#) through publish/subscribe, request/response, RPC, REST, message queues, event streams, file exchange, shared memory, database handoff, gateway access, service invocation, or another exchange mechanism. Those realisations do not redefine the concept of [Communication Endpoint](#).

A later part may realize a [Data Structure Definition](#) through IDL, Protocol Buffers, JSON Schema, XML Schema, Avro, OpenAPI schemas, UML, SysML, ontology classes, database schemas, generated language types, or another formal representation. Those realisations do not redefine the concept of [Data Structure Definition](#).

A later part may realize modular development through libraries, services, processes, containers, orchestrated workloads, RPC services, REST resources, event-driven components, or other implementation structures. Those choices remain implementation or deployment decisions. They do not define the conceptual architecture.

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

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

Permanent link:
<https://wiki.didosolutions.com/fxdemo/01-part/11-conceptual-rules-and-constraints/11-3-conceptual-architecture-does-not-prescribe-realisation-mechanisms>

Last update: 2026/08/04 07:09

