NOD-002e — Instantiate the Node

Go to NOD-002 — Provision a Node

Statement

The DIDO-TE SHALL instantiate a Node in accordance with its Node Definition and applicable provisioning requirements.

Derived From

- NOD-002 — Provision a Node
- NOD-002a — Select the Node Definition
- NOD-002b — Select the Node Implementation
- NOD-002c — Resolve Node Dependencies
- NOD-002d — Allocate Node Resources
- [DTE5] DIDO-TE Requirements Register, source requirement identifier and obligation to be assigned
- [DTE6] Structured Information Processing Reference Architecture (SIP-RA)
- [DTE7] Federated Data Interpretation Systems Reference Architecture (FDIS-RA)

Rationale

Instantiation creates a uniquely identifiable Node instance from the controlled inputs selected and prepared during earlier provisioning activities.

The instantiation activity associates the Node instance with its governing Node Definition, Node Implementation, Dependencies, Test Resources, Test Environment, and provisioning activity. This association distinguishes the provisioned instance from other instances of the same Node type or implementation.

Uncontrolled or incomplete instantiation introduces uncertainty about the identity, composition, location, ownership, and provenance of the Node participating in subsequent configuration and Test Execution.

Instantiation creates the Node instance but does not establish its complete initial state, validate its readiness, or begin Test Execution. Separate requirements govern those activities.

Applies To

- DIDO-TE
- Node
- Node Definition

- Node Implementation
- Dependency
- Test Resource
- Provisioning
- Test Environment
- Evidence

Verification

Verification confirms that:

- Each instantiated Node has a unique identity.
- The Node identity distinguishes the instance from every other Node instance.
- The Node instance identifies its applicable Node role and Node type.
- The Node instance identifies the selected Node Definition and version.
- The Node instance identifies the selected Node Implementation, version, build, and variant.
- The Node instance uses the resolved mandatory Dependencies.
- The Node instance uses the allocated Test Resources.
- The instantiated composition conforms to the selected Node Definition.
- The instantiated composition conforms to the applicable Test Environment constraints.
- The DIDO-TE verifies the identity and integrity of each implementation artifact used during instantiation.
- The DIDO-TE verifies the availability of each mandatory Dependency before using it.
- The DIDO-TE verifies the availability and allocation of each mandatory Test Resource before using it.
- The DIDO-TE applies applicable placement, isolation, security, jurisdictional, residency, and access-control constraints.
- The DIDO-TE identifies the physical, virtual, containerised, process, device, or service context hosting the Node instance.
- The DIDO-TE identifies the infrastructure, platform, runtime, network, storage, and middleware context material to the Node instance.
- The DIDO-TE identifies the time and authority responsible for the instantiation activity.
- The DIDO-TE records each operation performed to create or obtain the Node instance.
- The DIDO-TE records each input, output, warning, exception, retry, substitution, and deviation associated with instantiation.
- The DIDO-TE detects a missing, altered, incompatible, unavailable, or unauthorised instantiation input.
- The DIDO-TE detects a partial, duplicate, inconsistent, or unsuccessful instantiation.
- The DIDO-TE prevents an unsuccessfully or incompletely instantiated Node from proceeding to configuration or Test Execution.
- Repeated instantiation from the same controlled inputs produces an equivalent Node composition within specified tolerances.
- A change to a material instantiation input results in a distinguishable Node instance or recorded revision.
- The DIDO-TE records the resulting Node identity, composition, location, status, creation time, and governing inputs.
- The DIDO-TE maintains Traceability among the Node Definition, Node Implementation, implementation artifacts, Dependencies, Test Resources, Test Environment, provisioning activity, Node instance, and resulting Evidence.

- An unidentified, incomplete, inconsistent, unauthorised, unsuccessful, or untraceable Node instantiation constitutes nonconformance with this requirement.

Verification includes:

- Inspection of the selected Node Definition and Node Implementation
- Inspection of the resolved Dependencies and allocated Test Resources
- Execution of the Node instantiation procedure
- Inspection of the resulting Node identity, role, type, composition, and location
- Comparison of the instantiated composition with the selected Node Definition
- Verification of implementation artifact identities and integrity
- Confirmation of applicable placement, isolation, security, jurisdictional, and residency constraints
- A negative assessment involving a missing or altered implementation artifact
- A negative assessment involving an unavailable Dependency or Test Resource
- A negative assessment involving a partial or duplicate instantiation
- Confirmation that an unsuccessful Node instance cannot proceed to configuration or Test Execution
- Repeated instantiation using the same controlled inputs
- Inspection of instantiation records, findings, exceptions, Evidence, and Traceability

Referenced By

- NOD-002f — Establish the Node Initial State
- NOD-002g — Record Node Provisioning
- NOD-002h — Validate the Provisioned Node

Related Architecture Sections

- Add links to the architecture sections governing Node instantiation, Node identity, Provisioning, Dependencies, Test Resources, Test Environment composition, Evidence, and Traceability.

Delivery Phase

Assign the applicable delivery phase.

Requirement Status

Draft

Statement Reference

Use the following syntax to reference this requirement’s Statement section from another DokuWiki

page:

```
{{section>dido:03-dido-te:99-annexes:annex-c-requirements:03-node-requirements:nod-002-provision-a-node:nod-002e-instantiate-the-node#Statement&noheader&nofooter&noeditbtn}}
```
