

6.11.7 Simulation, Training and Testing Environment

[Return to RDT&E](#)

The OMG members recommend the Federal Reserve invest [Research Development Test & Evaluation \(RDT&E\) Funding](#) in developing and perfecting a Simulation, Training, and Test environment.

This environment should allow for the creation and running of thousands of Distributed Nodes of different types to simulate a real distributed environment in a matter of seconds or at most minutes. The Testing Environment (TE) needs to be able to start from a known starting point each time it is run, especially to support [Regression Testing](#).

This kind of Testing Environment (TE) is different than the [Sandboxes](#) most Cryptocurrencies currently offer. These cryptocurrency sandboxes work on actual distributed systems with a set of collective nodes running on actual networks. This means the rigor of the testing is compromised since the state of the sandbox is a moving target (i.e., it is not the same this time it is run as it was the last time it was run). Additionally, Sandbox testing means the ability to release hostile agents into the environment is extremely limited (who wants to have a hostile agent accidentally release a malicious agent into the wild?). It also limits the testing of the [Network Devices](#) (i.e., hub, switch, etc.) and the use of [HTTP Sniffers](#).

A Testing Environment (TE) can support simulation by allowing the morphology (number and kinds of nodes) of the Environment to be altered and rerun. Although the performance metrics collected during each run would not necessarily reflect the actual metrics during a real deployment, they would be useful relative to each other allowing for an informed decision-making process.

Training Environment leverages the TE by using the Testing Environment as a platform for training people and AI on different well-known scenarios. The TE would be started at a known point each time and the people or the intelligent agents could learn through the use of “what happens if” scenarios. The metrics collected could be used to assess the results.

Using a Testing Environment (TE) from the onset of the CBDC, helps fulfill the Non-Functional Requirement for Testability.

Note: [Testability](#), Testable, Testing, and Test are not synonyms for each other. Just because a system or program is undergoing testing using various tests does not necessarily mean that the system or program is actually Testable. See section [50_recomend](#).

The following sections from the [OMG DIDO Reference Architecture \(DIDO-RA\)](#) could be helpful in understanding the creation of a Test Environment.

- [Detailed discussion of types of Tests](#)
- [Testability](#) Non-Functional Requirement
- [Securability](#) Non-Functional Requirement

Table 1 lists an overview of the types of tests defined by the OMG DIDO-RA. All systems should have [Policies and Procedures](#) in place to improve the quality of the system and increase reliability. Follow the links to get more detail on each test.

Table 1: The different kinds of tests.

- [Unit Testing](#)
- [Integration Testing](#)
- [End-to-End Testing \(E2E Testing\)](#)
- [Smoke Testing](#)
- [Sanity Testing](#)
- [Regression Testing](#)
- [Acceptance Testing](#)
- [White Box Testing](#)
- [Black Box Testing](#)
- [Interface Testing](#)
- [Interoperability Testing](#)

From:
<https://www.omgwiki.org/CBDC/> - **OMG Central Bank Digital Currency (OMG-CBDC) Working Group (WG) Wiki**

Permanent link:
https://www.omgwiki.org/CBDC/doku.php?id=cbbc:public:cbbc_omg:04_doc:90_recommend:93_recomend:70_test&rev=1652739075

Last update: **2022/05/16 18:11**

