

Question: 22. Are there additional design principles that should be considered? Are there tradeoffs around any of the identified design principles, especially in trying to achieve the potential benefits of a CBDC?

[Return to CBDC Benefits, Risks, and Policy Considerations](#)

Question

[Return to Top](#)

1. **Are there additional design principles that should be considered?**
2. **Are there tradeoffs around any of the identified design principles, especially in trying to achieve the potential benefits of a CBDC?**

Answer

[Return to Top](#)

Non-Functional Requirements Design Principles

[Return to Top](#)

Overview

[Return to Top](#)

Some major design principles missing from the [White Paper](#) are the specification of [Non-Functional Requirements](#). The [Distributed Immutable Data Objects - Reference Architecture \(DIDO-RA\)](#) provides a list of [Non-Functional Requirements](#) applicable to the CBDC. The following is an outline of the **Non-Functional Requirements**:

1. [Portability](#)

- [Adaptability](#)

- [Installability](#)
- [Replaceability](#)

2. [Reliability](#)

- [Maturity](#)
- [Availability](#)
- [Fault Tolerance](#)
- [Recoverability](#)

3. [Maintainability](#)

- [Modularity](#)
- [Reusability](#)
- [Analysability](#)
- [Modifiability](#)
- [Testability](#)

4. [Security](#)

- [Confidentiality](#)
- [Data Integrity](#)
- [Non-Repudiation](#)
- [Authenticity](#)
- [Accountability](#)

5. [Manageability](#)

- [Types of Manageability Functions](#)
- [Manageability Costs](#)
- [System Manageability Issues](#)
- [Software Manageability Issues](#)

6. [Usability](#)

- [Effectiveness Metrics](#)
- [Efficiency Metrics](#)
- [Satisfaction Metrics](#)

7. [Performance](#)

- [Platform Performance](#)
- [Application Performance](#)
- [Network Performance](#)

8. [Interoperability](#)

9. [Elasticity](#)

10. Scalability

Examples

[Return to Top](#)

Some of these **Non-Functional** requirements were alluded to in the White Paper, but not directly specified or defined. Table 1 provides an example of cross-referencing the Non-Functional Requirements to the Benefits, Policy Considerations, Risks and Design requirements identified in the [White Paper Analysis](#) done by the [Object Management Group](#).

Table 1: Example of mapping a subset of requirements identified during the White Paper Analysis conducted by the OMG.

Non-Functional Requirement	Benefits, Policy Considerations, Risks and Design requirements
Adaptability	B: B0008, B0016, B0025, B0026, B0029, B0032, B0033, B0035, B0037, B0038, B0039, B0048, P: P0007
Performance	B: B0009, B0011-1, B0011-2, B0017-1, B0017-2, B0024, B0034, B0045, B0047, P: P0026, P0028-6
Availability	B: B0012, B0013 B0053, D: D0016
Confidentiality	B: B0004, B0022-1, B0051
Efficiency Metrics	B: B0001, B0002, B0009, B0011, B0014, B0047, B0051, P: P0023, P0026, P0028-6
B = Benefit Considerations	
P = Policy Considerations	
R = Risk Considerations	
D = Design Considerations	

Note: There should be no tradeoffs between **Non-Functional Requirements** and any existing requirements identified in [White Paper](#) as determined by the OMG [White Paper Analysis](#)

Discussion of Example

[Return to Top](#)

The first requirement listed under [Adaptability](#) is Benefit **B0008**:

B0008	3	Provide entrepreneurs a platform on which to create new financial products and services
B = Benefit Considerations		
P = Policy Considerations		
R = Risk Considerations		

D = Design Considerations

In order to realize **B0008**, the CBDC needs to be considered a system that can support both:

- *[Software Adaptability](#) - A software component with a well-defined, stable [Application Programming Interface \(API\)](#) can be exchanged using another component with minimal effort, as long as that component adheres to the API. For example, [SQL](#) describes an API for a database component. As long as the software adheres to the standard SQL API, the [Database Management System \(DBMS\)](#) can be exchanged between, for example, [Oracle](#) | [Oracle](#) and [PostgreSQL](#), with no to minimal impact.*
- *[Architecture Adaptability](#) - Connectors between software components change without having to change the components. This again comes down to having well-defined, stable APIs for the connectors. For example, the [Unix File System \(UnixFS\)](#) is a connector between software components and the physical filesystem. The associated UnixFS library can be exchanged for the [InterPlanetary File System \(IPFS\)](#) UnixFS connector and the software component should have no to minimal impact.*

Obviously, this is just an example, and the Federal Reserve should adopt the information and update the information in the [White Paper Analysis](#) and also perform their own assessment a similar to that presented in Table 1. The discussions and justifications for each requirement need to be captured for future reference. For example, **B00008** would have its own discussion area within the CBDC requirements document.

Functional Requirements Design Principles

[Return to Top](#)

Overview

[Return to Top](#)

Some major design principles missing from the [White Paper](#) are the specification of https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.a_glossary:f:funcreq. The [Distributed Immutable Data Objects - Reference Architecture \(DIDO-RA\)](#) provides a list of [Functional Requirements](#) applicable to the CBDC. The following is an outline of the **Functional Requirements**:

1. Platforms

- [Hardware Platforms](#)
- [Operating System Platforms](#)
- [Runtime Platforms](#)
- [Network Platforms](#)
- [Virtualization Platforms](#)

2. Access Control

Some of these **Functional Requirements** were alluded to in the White Paper, but not directly specified or defined. The Table 2 provides an example of cross-referencing the **Functional Requirements** to the Benefits, Policy Considerations, Risks and Design requirements identified in the [White Paper Analysis](#) done by the [Object Management Group](#).

Examples

[Return to Top](#)

<caption>Example of mapping a subset of requirements identified during the White Paper Analysis conducted by the OMG.</caption>

Functional Requirement	Benefits, Policy Considerations, Risks and Design requirements
Hardware Platforms	B: B0007, B0008, B0011, B0014, B0015, B0018, B0022-3, B0024, B0025, B0029, B0030, B0032, B0033, B0037, B0038, B0039, B0040, B0041, B0043, B0044, B0047, B0049, B0053, P: P0007, P0013, P0020, P0026, P0028, R: R0007, R0008, R0010, R0011, D: D0009, D0012, D0015, D0016, D0017
Access Control	B: B0004, B0005, B0007, B0009, B0010, B0011, B0015, B0018, B0022, B0025, B0029, B0033, B0035-2, B0038, B0041, B0044, B0045, B0046, B0049, B0050, P: P0004, P0005, P0007, P0020, P0021, P0023, P0025, P0029, R: R0001, R0003, R0007, R0008, R0011, R0014,
B = Benefit Considerations	
P = Policy Considerations	
R = Risk Considerations	
D = Design Considerations	

Discussion of Example

[Return to Top](#)

The one of the first requirements listed under [Platforms](#) is Benefit **B00011**:

	B0011	7	Make payments: 1. faster 2. cheaper 3. more convenient 4. more accessible
B = Benefit Considerations			

Requirement **B0011** is a compound requirement, and the selection of a Platform can have an impact on:

- **B00011-1** - Faster

- **B00011-2** - Cheaper
- **B00011-3** - More Convenient
- **B00011-4** - More Accessible

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:20_comments:dsn:q22:start&rev=1652739083

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

