D5.1 – Hardware requirements of all bricks and tray (a)

WP5: Hardware/platform design, development and integration
D5.1 – Hardware requirements of all bricks and tray (a)

Version: 1.0

<table>
<thead>
<tr>
<th><strong>Due date</strong></th>
<th>PM11, 30 November 2016</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Submission date</strong></td>
<td>30-11-2016</td>
</tr>
<tr>
<td><strong>Project start date</strong></td>
<td>01/01/2016</td>
</tr>
<tr>
<td><strong>Project duration</strong></td>
<td>36 months</td>
</tr>
<tr>
<td><strong>Deliverable lead organization</strong></td>
<td>SINTECS</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>1.0</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>Final</td>
</tr>
<tr>
<td><strong>Author(s)</strong></td>
<td>Evert Pap (SINTECS)</td>
</tr>
<tr>
<td></td>
<td>Roy Krikke (SINTECS)</td>
</tr>
<tr>
<td><strong>Reviewer(s)</strong></td>
<td>Ferad Zyulkyarov (BSC)</td>
</tr>
<tr>
<td></td>
<td>Dimitris Theodoropoulos (FORTH)</td>
</tr>
<tr>
<td></td>
<td>Vaibhawa Mishra (UoB)</td>
</tr>
<tr>
<td></td>
<td>Kostas Katrinis (IBM)</td>
</tr>
</tbody>
</table>

**Dissemination level**

| PU | Public with references to confidential documents. |

**Disclaimer**

This deliverable has been prepared by the responsible Work Package of the Project in accordance with the Consortium Agreement and the Grant Agreement No 687632. It solely reflects the opinion of the parties to such agreements on a collective basis in the context of the Project and to the extent foreseen in such agreements.
Acknowledgements

The work presented in this document has been conducted in the context of the EU Horizon 2020. dReDBox (Grant No. 687632) is a 36-month project that started on January 1st, 2016 and is funded by the European Commission.

The partners in the project are IBM IRELAND LIMITED (IBM-IE), PANEPISTIMIO THESSALIAS (UTH), UNIVERSITY OF BRISTOL (UOB), BARCELONA SUPERCOMPUTING CENTER – CENTRO NACIONAL DE SUPERCOMPUTACION (BSC), SINTECS B.V. (SINTECS), FOUNDATION FOR RESEARCH AND TECHNOLOGY HELLAS (FORTH), TELEFONICA INVESTIGACION Y DESSARROLLO S.A.U. (TID), KINESENSE LIMITED (KS), NAUDIT HIGH PERFORMANCE COMPUTING AND NETWORKING SL (NAUDIT HPC), VIRTUAL OPEN SYSTEMS SAS (VOSYS), POLATIS LTD. (POLATIS).

The content of this document is the result of extensive discussions and decisions within the dReDBox Consortium as a whole.

More information
Public dReDBox reports and other information pertaining to the project will be continuously made available through the dReDBox public Web site under http://www.dredbox.eu.
D5.1 – Hardware requirements of all bricks and tray (a)

Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comments, Changes, Status</th>
<th>Authors, contributors, reviewers</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>23/05/2016</td>
<td>First draft with TOC to discuss document direction before actualization of document.</td>
<td>Evert Pap (SINTECS)  ARCH group</td>
</tr>
<tr>
<td>0.2</td>
<td>12/06/2016</td>
<td>Initial draft with partial content to review format of document</td>
<td>Roy Krikke (SINTECS)</td>
</tr>
<tr>
<td>0.3</td>
<td>15/08/2016</td>
<td>Processed review comments. Added additional content. Internal development version</td>
<td>Evert Pap (SINTECS)</td>
</tr>
<tr>
<td>0.4</td>
<td>15/08/2016</td>
<td>Version for internal review</td>
<td>Roy Krikke (SINTECS) Paul ten Wolde (SINTECS)</td>
</tr>
<tr>
<td>0.5</td>
<td>29/08/2016</td>
<td>Processed review comments. Added Test Tray. Second pass of review</td>
<td>Evert Pap (SINTECS) Roy Krikke (SINTECS) Paul ten Wolde (SINTECS) Pim de Sain (SINTECS) Hans Klos (SINTECS)</td>
</tr>
<tr>
<td>0.6</td>
<td>19/09/2016</td>
<td>Processed review comments</td>
<td>Evert Pap (SINTECS)</td>
</tr>
<tr>
<td>0.7</td>
<td>24/10/2016</td>
<td>Processed feedback from implementation team.</td>
<td>Evert Pap (SINTECS)</td>
</tr>
<tr>
<td>0.8</td>
<td>09/11/2016</td>
<td>Processed review comments</td>
<td>Evert Pap (SINTECS)</td>
</tr>
<tr>
<td>0.9</td>
<td>28/11/2016</td>
<td>Final review</td>
<td>Kostas Katrinis (IBM)</td>
</tr>
<tr>
<td>1.0</td>
<td>29/11/2016</td>
<td>Update document revision to 1.0 and status to final.</td>
<td>Evert Pap (SINTECS)</td>
</tr>
</tbody>
</table>
1 Contents

Executive Summary ............................................................................................................7
2 Overview .........................................................................................................................7
3 References .......................................................................................................................8
  3.1 dReDBox Documents ..................................................................................................8
  3.2 Standards ...................................................................................................................8
  3.3 Datasheets ..................................................................................................................8
  3.4 Other .........................................................................................................................8
4 Scope .............................................................................................................................9
5 Item identification ........................................................................................................10
  5.1 System overview dReDBox system ..........................................................................10
    5.1.1 dReDBox-Rack ...................................................................................................11
    5.1.2 dReDBox-dBOX ...............................................................................................11
    5.1.3 dReDBox-Cluster ..............................................................................................11
    5.1.4 dReDBox-Tray ....................................................................................................11
    5.1.5 dReDBox-Peripheral tray ..................................................................................11
    5.1.6 dReDBox-Rack OSM .........................................................................................11
    5.1.7 dReDBox-Bricks ................................................................................................11
    5.1.8 dReDBox-dBOX OSM .......................................................................................12
    5.1.9 dReDBox-dBOX Power Supply Unit ..................................................................12
    5.1.10 dReDBox-Test tray ..........................................................................................12
6 Project Design Decisions ..............................................................................................13
7 Hardware design assumptions and external requirements ...........................................14
  7.1 Hardware result definition .........................................................................................14
  7.2 FPGA pinout validation ...........................................................................................14
  7.3 Hardware validation .................................................................................................14
8 Requirements ................................................................................................................15
  8.1 Global requirements .................................................................................................15
  8.2 dBOX requirements ................................................................................................18
dReDBox Networking .......................................................................................................23
    8.2.1 High Speed Electrical Network .........................................................................24
    8.2.2 High Speed Optical Network ............................................................................25
    8.2.3 PCIe Network ....................................................................................................25
    8.2.4 Management Network .......................................................................................26
    8.2.5 Accelerator Network ..........................................................................................26
  8.3 dReDBox tray requirements ......................................................................................27
    8.3.1 dBMC requirements ..........................................................................................29
8.4 dBRICK requirements
  8.4.1 Compute brick requirements
  8.4.2 Memory brick requirements
  8.4.3 Accelerator brick requirements
8.5 Peripheral tray requirements
8.6 Test tray (dTETRAY) requirements

9 System Configuration and Boot flow
  9.1 Cold start to stand-by state
  9.2 Stand-by to operational state
  9.1 Operational to stand-by state

10 Requirement List
11 Notes
  11.1 Abbreviations and Acronyms

Index of Figures
  Figure 1 – dReDBox system components overview
  Figure 2 – Clocking architecture overview
  Figure 3 – enclosure PCB height restrictions
  Figure 4 – dTRAY maximum dimensions
  Figure 5 – dTRAY dBRICK location identification
  Figure 6 – dReDBox dTRAY network overview
  Figure 7 – dReDBox dBMC overview
  Figure 8 – dBRICK overview
  Figure 9 – dBRICK Dimensions
  Figure 10 – dReDBox Test Tray overview

Index of Tables
  Table 1: Referenced dReDBox documents
  Table 2: Referenced standards documents
  Table 3: Referenced datasheets and specifications
  Table 4: Other references
  Table 5: dBOX power budget
Executive Summary

2 Overview

Today’s data centers consist of multiple servers interconnected through a high-speed network. The architecture of servers has always been designed around trays with a fixed number of different components such as processors, memory, and peripherals. This traditional architecture where the server is the main building block leads to spare resource fragmentation and inefficiencies in current data centers.

dReDBox aims at delivering a disaggregated architecture that will not require memory or accelerators to be co-located with the processor in the same tray. This will enhance elasticity, improve virtual machine migration and reduce the total cost of ownership compared to the current data centers. In this new architecture, the main building block is no longer the server, but the brick. Different bricks are designed with different functionalities, for example, compute, memory and accelerator bricks.

This document has two delivery moments, defined as d5.1(a) and d5.2(b). The first delivery consists of this document and defines the hardware as it will be developed for the d5.1(b) delivery. The second delivery will describe the board bring up, hardware verification and functional tests of the demonstrator platform.

Due to the inflexible nature of hardware development an incremental or agile way of working is not desired. Changes in the hardware requirement have a direct impact on the cost of the design. Starting the design with uncertainties in the requirements will result in an over dimensioned product, adding extra costs and design time.

This document, when reviewed and approved to the final d5.1(a) state, will serve as the final reference regarding the hardware design. All changes must be reviewed for the resulting impact and cost and agreed upon before they will be implemented, subject to potential corrective amendments as the project unfolds and uncertainty is reduced.

Abbreviations in this document will be written out fully one time, and in the rest of the document only the abbreviation will be used. All abbreviations can be found in section 11.1.
3 References

3.1 dReDBox Documents

<table>
<thead>
<tr>
<th>Ref</th>
<th>Title</th>
<th>Source</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>D2.1 – Requirements Specification and KPIs Document (a)</td>
<td>box</td>
<td>Deliverables\WP2\Submitted\dRedBox\Deliverable_D2.1.REVISION.pdf</td>
</tr>
<tr>
<td>[2]</td>
<td>D2.3 – System Architecture specification (a)</td>
<td>box</td>
<td>Deliverables\WP2\Submitted\dRedBox\Deliverable_D2.3.REVISION.pdf</td>
</tr>
</tbody>
</table>

Table 1: Referenced dReDBox documents

3.2 Standards

<table>
<thead>
<tr>
<th>Ref</th>
<th>Title</th>
<th>Source</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>[3]</td>
<td>EIA-310-D</td>
<td>www</td>
<td>Revision D</td>
</tr>
<tr>
<td>[5]</td>
<td>CFP2 BASELINE</td>
<td>www</td>
<td>Revision 1L, 20130806</td>
</tr>
<tr>
<td>[9]</td>
<td>microATX Motherboard Interface Specification</td>
<td>www</td>
<td>v 1.2</td>
</tr>
</tbody>
</table>

Table 2: Referenced standards Interface documents

3.3 Datasheets

<table>
<thead>
<tr>
<th>Ref</th>
<th>Title</th>
<th>Source</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10]</td>
<td>Xilinx Platform Cable USB II (DS593)</td>
<td>www</td>
<td>V1.5</td>
</tr>
<tr>
<td>[12]</td>
<td>4-Wire Pulse Width Modulation (PWM) Controlled Fans</td>
<td>www</td>
<td>1.3</td>
</tr>
<tr>
<td>[14]</td>
<td>LUX62608 OptoPHY™ Datasheet</td>
<td>Under NDA</td>
<td>February 24, 2016 – revision 0.6</td>
</tr>
</tbody>
</table>

Table 3: Referenced datasheets and specifications

3.4 Other

<table>
<thead>
<tr>
<th>Ref</th>
<th>Title</th>
<th>Source</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>[15]</td>
<td>Scalys OUI registration</td>
<td>www</td>
<td>N.A.</td>
</tr>
</tbody>
</table>

Table 4: Other references
4 Scope

This document identifies and describes the requirements of the hardware platform developed in the dReDBox project. Requirements of the programmable parts in the system are not part of this document, except for the functionality needed to support the hardware itself.

Requirements in this document are divided into 3 types:

Must:
“Must” defines an absolute requirement.

Should:
“Should” defines a preferred requirement.

May:
“May” defines an optional requirement.
5 Item identification

This chapter defines the different components of the dReDBox system. All documentation referring to this document must conform to these definitions.

5.1 System overview dReDBox system

Figure 1 shows the different components defined in the dReDBox system. The dReDBox system compromises:

- dReDBox-Rack
- dReDBox-Cluster
- dReDBox-Box
- dReDBox-Peripheral tray
- dReDBox-Rack-Optical Switch Matrix
- dReDBox-Tray
- dReDBox Bricks
  - Compute brick
  - Memory brick
  - Accelerator brick
- dReDBox dBOX OSM
- dReDBox dBOX-Power Supply Unit
- dReDBox Test Tray

![Diagram of dReDBox system components]

Figure 1 – dReDBox system components overview

---

1 The Test tray is not part of the dReDBox system. It is used for verification and validation of the hardware.
5.1.1 dReDBox-Rack
The dReDBox-RACK (dRACK) is the complete dReDBox system. The dRACK is the final hardware deliverable associated with “D5.2: Hardware integration and tests of all bricks and tray(b)”. The dRACK will be used as the platform for the different demonstrators.

5.1.2 dReDBox-dBOX
The dReDBox-dBOX (dBOX) houses the main components of the dReDBox system and can be considered the heart of the dReDBox system. The dBOX will be compatible with standard datacenter infrastructures and will look like a generic server, from a housing and datacenter deployment perspective.

5.1.3 dReDBox-Cluster
The dReDBox-Cluster (dCLUST) is a number of dBOXes grouped as a logical unit. The number of dBOXes per dCLUST depends on the port count of the optical switches and on application and system/operator requirements. Refer to [2] for more information of the operational configuration of the dREDBox system.

5.1.4 dReDBox-Tray
The dReDBox-tray (dTRAY) provides the interconnect and support functions for the different dReDBox resources (termed “bricks”). It serves as a “motherboard” in the dBOX.

5.1.4.1 dReDBox-Board Management Controller
The dReDBox-Board Management Controller (dBMC) controls and configures the dTRAY and all the resources located on the dTRAY. The dBMC itself is controlled by the orchestration software.

5.1.5 dReDBox-Peripheral tray
The dReDBox-peripheral-tray (dPERTRAY) is a Commercial off-the-shelf (COTS) product. The dPERTRAY provides support for different kinds of peripherals through a PCIe bus. This will be a “plug-and-play” solution which can be connected to the dBOX using a standard PCIe cable.

5.1.6 dReDBox-Rack OSM
The dReDBox-Rack-OSM (dROSM) is a COTS product, which will be used to interconnect the different dBOX systems.

5.1.7 dReDBox-Bricks
There are different types of dReDBox-Bricks (dBRICK) which can be used in the dBOX. dBRICKs are interchangeable. Each brick provides different functionality to the dReDBox System. All bricks can be interconnected using a programmable high speed interconnection network. This network consists of optical and electrical interfaces.
5.1.7.1 dReDBox-Accelerator brick
The dReDBox-Accelerator brick (dACCELBRICK) will provide a programmable and application specific accelerator function to the dReDBox system. It will also have the ability to interface with a 100GbE interface on the dTRAY. The dACCELBRICK will have a dedicated position on the dTRAY if the 100GbE interface is to be used due to constraints on the high-speed interconnection system. If the 100GbE interface is not used, there are no placements constraints.

5.1.7.2 dReDBox-Memory brick
The dReDBox-Memory Brick (dMEMBRICK) will provide the disaggregated memory to the dReDBox System.

5.1.7.3 dReDBox-Compute brick
The dReDBox-Compute brick (dCOMPUBRICK) will provide the processing power to the dReDBox System.

5.1.8 dReDBox-dBOX OSM
The dReDBox-Box OSM (dBOSM) is a COTS product, which will be used to interconnect the different dBRICKs inside of a box. Depending on the network topology used, it can also be connected to the dROSM.

5.1.9 dReDBox-dBOX Power Supply Unit
The dReDBox-Box PSU (dBPSU) will provide the dTRAY with power. This is an AC/DC power supply, which can provide enough power for a completely provisioned dBOX.

5.1.10 dReDBox-Test tray
The dReDBox-Test tray (dTETRAY), while not a part of the dReDBox system, is a critical part in the validation of the dReDBox system. It will provide a simplified version of the dReDBox system and will allow the set-to-work of the different modules without the need of a complete system. It will also allow partners to work on the dBRICKs without the need of a complete dBOX.
6 Project Design Decisions

Design decisions made on a higher level put restrictions on the hardware design. This chapter lists these decisions and the rationale behind them.

A Xilinx System on Chip (SoC) device is chosen for the dCOMPUBRICK. The SoC will help to reduce risk regarding the memory interface between the SoC and the high-speed interconnect network. The architecture will support a future change to a different SoC for the processing brick. The exact type was chosen due to the availability of the devices.

1. The SoC used for the Processing Brick used in the prototype must be a Xilinx ZYNQ Ultrascale+ MPSoC (XCZU9EG with FFVC900 package).

The SoC has enough Field Programmable Gate Array (FPGA) resources to host the dMEMBRICK and dACCELBRICK required functionality (e.g. remote memory request translation, custom acceleration as described in [2]). Using the same SoC for the different bricks saves considerable design effort.

2. The dCOMPUBRICK, dMEMBRICK and dACCELBRICK must use the same SoC.

The use of COTS modules for the memory will allow for a flexible configuration of the memory bricks, allowing for different capacities of memory bricks. These modules are standard DDR4-SODIMM modules, as commonly found in laptops and small formfactor PC systems. The use of modules provides a future upgrade path for the memory capacity, without the need for extra design work.

3. The dMEMBRICK must use standard DDR4 memory modules as the disaggregated memory.

Mid Board Optics (MBO) will be used to provide interfacing to the optical network. The use of the Polatis optical switch limits the options for the MBO device. The switch has only support for Single Mode Fiber (SMF). At the time of writing there is only one provider for single mode fiber MBO devices, thereby

4. The MBO used must be Luxtera LUX62608 modules.
5. The MBO must be placed on the dBRICK.
7 Hardware design assumptions and external requirements

This chapter describes the dependencies which are not direct requirements for the dReDBox hardware development, but which can have an impact on the design process.

7.1 Hardware result definition

The hardware designed and delivered as the end product of the dReDBox project will be a prototype. There will not be any official qualifications and certification tests performed as part of this development.

7.2 FPGA pinout validation

An FPGA has many configurable interfaces and pins, but there are restrictions on which interface can be used on what pin in certain configurations. The best way to make sure the hardware design meets these restrictions is to create a “pinning FPGA” which verifies the interfaces and pins using the tooling supplied by the manufacturer. A Pinning FPGA is a skeleton implementation of the FPGA, which allows the manufacturer tooling to check if the implementation is valid.

Pinning FPGAs will be created by the partners responsible for the FPGA logic to make sure the different FPGAs are connected correctly.

The creation of these pinning FPGAs can be an iterative process due to the flexibility of the FPGA and constrains put on the pinning from other sources (PCB Layout, Signal Integrity (SI), Power Integrity (PI), Timing analysis, etc).

7.3 Hardware validation

Board bring up will be supervised and organized by SINTECS. Depending on the design specifics there might be a need for specialized test software and FPGA implementations. These needs will be catalogued and managed by SINTECS, while the realization will be a combined effort of all the partners.
8 Requirements

The requirements described in this chapter specify the implementation of the dReDBox prototype, and may not implement all features described in the d2.1[1] and d2.3[2] documents.

8.1 Global requirements

d5.1-req-1: global component restrictions:
There are restrictions on the choice of components driven by the development teams, availability and reuse of existing designs. These restrictions are listed here.

1. The FPGAs used must be “Zynq UltraScale+ MPSoC ZU9EG” devices.
2. The Electrical switch matrix used must be a “Macom M21605” device.
3. The optical transceiver modules used must be “Luxtera LUX62608” devices.


d5.1-req-2: global reference clocking:
To reduce added latency in the High Speed Electrical NetWork (HSENW) and High Speed Optical NetWork (HSONW) a global reference clock will be used. This will allow the different modules to ignore differences between the local and the remote reference clock, reducing the need for an elastic buffer and the associated latency. The shared clock will also help when switching the HSONW and HSENW since there will be no difference in the received clocks between switched channels. The global reference clock can be shared on the dBOX, dCLUST or dRACK level depending on system needs.

Refer to Figure 2 for an example on how this could be implemented.

![Clocking architecture overview](image)

1. There must be a global reference clock in the dRACK, which must be used to derive the reference clocks used by the HSENW and HSONW.
2. The global reference clock must have the same frequency across the different dBOXes.
3. The global reference clock may have the same phase across the different dBOXes.
4. When the global reference clock is absent, a local reference clock must be available.
D5.1 – Hardware requirements of all bricks and tray (a)

5. The switching between the global and local reference clock must not create a disruptive change to the local frequency. This implies a slow feedback loop.

The global reference clock must be distributed between the different dBOXes. Preferably this would be accomplished using one of the available interconnect cables between the dBOXes (PCIe). A dedicated cable is also acceptable, e.g. a coax cable with SMA, BNC or other suitable connectors.

6. The Global reference clock must be regenerated in each dBOX and must be able to drive another dBOX.

7. The Global reference may be distributed using the PCIe cable interface or using a dedicated cable.

8. The selection between the local clock and the global reference clock must be controlled by the dBMC.

9. The selection between the local clock and the global reference clock must be on the dTRAY level. The local clocks must always use the same reference.

**d5.1-req-3: global design for testability:**

1. Control and status signals should be accessible for measurement in an assembled dBOX.

2. All LEDs must have a descriptive label in the silkscreen.

3. All DIP-switches must have a descriptive label and description in the silkscreen.

**d5.1-req-4: global design system states:**

1. The dBOX must have two distinctive states of operation:
   a. Stand-by
      Only the dBMC and the management network are active. The system is running from the stand-by power of the ATX power supply
   b. Operational
      The ATX power supply is enabled, and the switches and other peripherals are active. The dBRICKs can be enabled individually in this state.
d5.1 – Hardware requirements of all bricks and tray (a)

**d5.1-req-5: hot-plug support:**
The dReDBBox system supports hot-plugging of dBRICKs.
This is a co-operative form of hot-plugging, in which both the dBMC and the dBRICK prepare for a Hot-plug event.

1. The dTRAY must provide a bi-color, red and green LED hot plug indicator (dHPI) in line with each dBRICK connector.
2. The dHPI must be controlled by the dBMC.
3. The dHPI must have the following states:
   a. Red LED off, green LED off: dBRICK slot off and disabled.
   b. Red LED off, green LED blinking: dBRICK hot-plug insertion detected and system is configuring dBRICK and dTRAY.
   c. Red LED off, green LED on: dBRICK operational.
   d. Red LED blinking, green LED off: dBRICK hot-plug removal requested and system is configuring dBRICK and dTRAY.
   e. Red LED on, green LED off: dBRICK error.
4. The dTRAY must provide a switch (dHPS) in line with each dBRICK connector which is used to indicate a dBRICK hot-plug removal request to the dBMC.
5. The dBMC must be able to enable and disable power to each dBRICK.

The dBRICKs must provide a Present Detection signal (dPD). This signal is used by the dBMC to detect a hot-plugged dBRICK and to take the appropriate actions.

6. The dPD must be the last signal connected when the dBRICK is inserted into the dTRAY.
7. The dPD must be the first signal to be disconnected when the dBRICK is removed from the dTRAY.
8. The dPD must be tied to ground on the dBRICK.
9. The dPD must be pulled up on the dTRAY.

To allow the removal of a dBRICK in a running system all active communication of the dBRICK must be stopped, and power consumption of the dBRICK has to be minimized. The power consumption by the dBRICK must be minimized to prevent damage to the connectors, which can occur when contacts are broken with a large current running through them.

To indicate to the system (and user) that a dBRICK is ready for removal, a discrete signal is used.

10. Each dBRICK must provide an active signal (dBACT).
11. The dBACT must be an open drain signal, which is pulled low by the dBRICK when it is not ready for removal.
12. The dBACT must be valid even when there is no power supplied to the dBRICK.
8.2 dBOX requirements

d5.1-req-11: dBOX Enclosure constraints:
1. The dBOX must conform to the 19", 2U rack housing standard as defined in reference [3].
2. The dBOX enclosure should contain a front panel which accommodates the different connectors and LEDs present on the front side of the dTRAY. This includes the optical interfaces present on the dBRICKs.
3. The dBOX must provide the standoffs on which the dTRAY is mounted.
4. The dBOX must provide a way to mount 4 fans.
   a. The fans must have a maximum height of 8 cm.
   b. The fans must have a maximum width of 8 cm.
   c. The fans must have a maximum depth of 4 cm.
5. The dBOX may be a COTS part.

d5.1-req-12: dBOX Available space dTRAY:
There are size restrictions on the different PCBs which will be mounted in the dBOX.
1. The dTRAY must be mounted in the enclosure on standoffs with a height of 3.5 mm.
2. The maximum height of the dTRAY including any dBRICKs must not be more than 75.0 mm.
3. The maximum height for components mounted on the bottom of the dTRAY must be 2.0 mm.
4. The dTRAY must have a width of no more than 40.0 cm.
5. The dTRAY must have a length of no more than 45.0 cm.
6. The dTRAY must have mounting holes supporting M3 screws.
7. The dTRAY must have mounting holes to support the mechanical stress of plugging in the dBRICKs.

Refer to Figure 3 and Figure 4 for more details. The placement of the bricks and other devices are for indication only.

Figure 3 – enclosure PCB height restrictions
d5.1 - req-13: dBOX Connector locations:

1. The PCIe cable connectors must be placed on the front side of the dTRAY, accessible through the front panel of the dBOX.
2. The Ethernet connectors must be placed on the front side of the dTRAY, accessible through the front panel of the dBOX.
3. The CFP2 connectors must be placed on the front side of the dTRAY, accessible through the front panel of the dBOX.
D5.1 – Hardware requirements of all bricks and tray (a)

d5.1-req-18:dTRAY tray location identification:
Each tray must have a different identifier. This identifier depends on the location of the dTRAY in the system and the configuration of the system. This identifier is set statically during installation.

1. The dTRAY must have an 8-bit dip-switch (dTID) connected to the dBMC.
2. The dTID must be placed on the front side of the dTRAY, accessible through the front panel.

Using 8 bits limits the number of dTRAYs which can be used in a single dReDBox domain to 256. This number is acceptable for the dReDBox demonstrator.

d5.1-req-16: dBOX supported dBRICK count:
The number of dBRICKs a dBOX can accommodate will depend on size of the dBRICKs and the space needed to connect all the dBRICKs. To determine the minimum size of the dBRICK and the dTRAY a detailed design is required, which is not available at the time of writing. A minimum requirement is set and a number of design goals formulated.

1. The dBOX must support at least 8 dBRICKs.
2. The dBOX should support at least 12 dBRICKs.
3. The dBOX may support up to 16 dBRICKs.

d5.1-req-17: dBOX dBRICK numbering:
1. The dReDBox Brick IDentifier (dBID) dBRICKs must start at 0.
2. The dBID must:
   a. start at the left side,
   b. count to the right,
   c. continue on the left side of the next row when the first row is finished.

Refer to Figure 5 for an example numbering of a tray with 9 bricks.

```
  backside
  [6] [7] [8]
  [0] [1] [2]
  frontside
```

Figure 5 – dTRAY dBRICK location identification
D5.1 – Hardware requirements of all bricks and tray (a)

d5.1-req-19: dTRAY dBRICK connector:
- The dBRICK connector must be a card-edge connector.
- The dBRICK connector must not have a latching mechanism.
- The dTRAY must have a set of vertical guide rails to support the dBRICKs.

The dBRICKs must be secured in the dTRAY and dBOX. This could be accomplished by a latching mechanism on the guide rails, a support beam in the dBOX housing or another suitable solution.
- The dBRICK must be mechanically secured to the dTRAY.
- The dBRICKs may be mechanically secured individually or in a group.
- The dBRICKs must be removable as a single unit, without affecting other dBRICKs.

d5.1-req-14: dBOX Cooling:
The dBOX is air-cooled. To allow the different parts of the system to dissipate heat certain restrictions are placed on the system.
The air in the dBOX will flow from the front of the case to the back of the case.
- The dBRICK connectors must be lined up front to back.
- The dBOX must be able to operate in an ambient temperature from 10°C up to 40°C.
- The dBOX must have a verified thermal solution, based on simulated thermal performance of the different components in the dBOX.
- All components which need additional provisions to dissipate heat must be placed in such a way that they have sufficient airflow.

Refer to Figure 4 for a possible layout of the dTRAY.

While it is preferred to avoid active cooling besides the fans mounted in the dBOX Enclosure, there is no requirement against it.
D5.1 – Hardware requirements of all bricks and tray (a)

**d5.1-req-15: dBOX power supply:**
The dBOX is a self-contained system, and must provide power for all of its internal components. Besides the dReDBox developed hardware there should be enough budget to power 3 dReDBox-Box Optical Switch Matrix (dBOSM), and two CFP2 modules.

The power budget for the dBOX is defined as following:

<table>
<thead>
<tr>
<th>Name</th>
<th>Number (max)</th>
<th>Power</th>
<th>Total power</th>
</tr>
</thead>
<tbody>
<tr>
<td>dBRICK</td>
<td>16</td>
<td>60W</td>
<td>960 W</td>
</tr>
<tr>
<td>dBOSM</td>
<td>3</td>
<td>15W</td>
<td>45 W</td>
</tr>
<tr>
<td>CFP2 module</td>
<td>2</td>
<td>12W</td>
<td>24 W</td>
</tr>
<tr>
<td>dTRAY</td>
<td>1</td>
<td>130W</td>
<td>130 W</td>
</tr>
</tbody>
</table>

**Table 5: dBOX power budget**

The Power budget is targeted for worst case operating conditions, and the system must never exceed these numbers.

1. The dBOX must provide space for a suitable PSU.
2. The dBOX-PSU must accept an input of 230VAC (±10%) @50HZ (±0.5%).
3. The dBOX-PSU must provide the dTRAY with an always-on stand-by voltage of 5-12V (±10%).
4. The dBOX-PSU must be able to provide at least 10W on the stand-by voltage rail.
5. The dBOX-PSU must provide the dTRAY with a switchable voltage of 12V (±10%).
6. The dBOX-PSU must be able to provide at least 1160W on the switchable voltage rail.
7. The dBOX-PSU may be a COTS ATX PSU.
dReDBox Networking

The dReDBox network overview is depicted in Figure 6. There are several different types of network, each with its own function and use case. The networks are identified and described in this paragraph.

Figure 6 – dReDBox dTRAY network overview
8.2.1 High Speed Electrical Network

The High Speed Electrical Network (HSENW) forms a local high speed network on the dTRAY. Connectivity and switching is accomplished by using a MACOM M21605 device. The dReDBox-Box Electrical Switch Matrix (dBESM) provides fast switching (<70 ns) with low latency (<2ns) and n:n connectivity. The dBESM provides configurable point-to-point connections between the dBRICKs and will allow the special dACCELBRICK locations to connect to the CFP2 connector for 100GbE connectivity.

The protocol and speed of a link on the HSENW is determined by the endpoints, and compatibility is maintained by the orchestration software.

d5.1-req-6: High Speed Electrical Network:

1. The HSENW must be able to operate at a bitrate (frequency) from 1 Gbps (0.5 GHz) to 12.5 Gbps (6.25 GHz).
2. The HSENW must be protocol agnostic.
3. The HSENW must support different frequencies at the same time. Each connection could have a different speed, depending on the dBRICK configuration.

The dBESM is configured and controlled by the dBMC, which in turn is controlled by the provisioning software (refer to Figure 7).

4. The dBMC must be able to configure the dBESM using the high speed parallel interface.
5. The HSENW links must always be bidirectional.
6. The HSENW must provide at least 8 links to each dBRICK.

The dBESM is a complex device and has many configuration options. To assist in debugging and verification a header is provided to access the device using the manufacturing tools.

7. The dBESM must be accessible through an SPI interface on a debug header.

There is a special use case for the dACCELBRICK which is aimed to support 100 GbE using a CFP2 module. These dBRICKs have a special location on the dTRAY.

8. The HSENW must provide 10 links to the dACCELERATOR dBRICKs.
9. The dACCELERATOR dBRICK connectors must be clearly marked on the dTRAY.
10. The HSENW must support at least 2 dACCELERATOR dBRICKs.
11. The HSENW must provide 2 CFP2 interfaces each compatible with a 10x10G configuration.
12. Each link in the HSENW network must be able to be shut down. This implies support for link shut down by the dBESM and the different dBRICKs.
8.2.2 High Speed Optical Network

The High Speed Optical Network (HSONW) will be provided by an optical interface module which will be placed on the dBRICK module. The optical module will provide a high-speed interface with other dBRICKs, either on the same dTRAY or on another dTRAY.

The dBOX has the option to include the Polatis Series 6000 OEM switch (dBOSM) which has either 48 or 96 ports.

**d5.1-req-7: High Speed Optical Network:**

1. The HSONW must support SMF.

The exact topology of the HSONW depends on the use case and configuration of the dBOX. The use of pluggable connectors in the HSONW provides the option for configuration during the assembly process.

2. The HSONW must be interconnected to the dBOSM using MTP/MPO connectors.
3. The HSONW may be interconnected to the dROSM using MTP/MPO connectors.
4. The dBOX must provide adapters to accommodate 32 MTP/MPO connections.
5. The MTP/MPO adapters will be located in the front panel of the dBOX.

8.2.3 PCIe Network

The PCIe Network (PCIENW) will form a global memory map, which is aimed to be accessible by all the dBRICKs and the dBMCS present in the dReDBox system. Due to the topology of the bus, latency will increase for nodes further away, but the latency will still be small compared to the latency of the management network. This network provides the provisioning software with an efficient way to access the different resources in the dReDBox system, and provides a fast remote interrupt mechanism.

**d5.1-req-8: PCIe Network:**

1. The PCIe network must support PCIe, generation 2.
2. The PCIe network must provide a 2x2 link to each dBRICK.
3. The PCIe network must provide a 2x1 link to the dBMCM.
4. The PCIe network must provide 3 2x4/2x8 links to PCIe x4 or x8 cable connectors.
5. Any settings of the PCIe network must be configurable by the dBMCM.
6. The PCIe network must support hot-plugging of devices.
7. Each link in the PCIe network must be able to be shut down by the dBMCM.
8.2.4 Management Network

The management network (dMNW) is a traditional Ethernet based network, which is used for control and status functions. The dBMC and all dBRICK have access to this network.

Part of the dMNW is a Ethernet network without Physical Layer Devices (PHY). This means that there are no PHY and magnetics present in the Ethernet connections. The PHY-less parts of the dMNW are the dBMC and dBRICK Ethernet connections to the switch.

**d5.1-req-9: Management Network:**

1. The dMNW must interconnect the dBMC and dBRICKs.
2. The dMNW must support 10BASE-T, 100BASE-T and 1000BASE-T networking as defined in the IEEE 802.3 standard (refer to [6]) for off-tray networking.
3. The dMNW must support 1000BASE-T networking as defined in the IEEE 802.3 standard (refer to [6]) for on-tray networking.
4. The dMNW must provide two off-tray interfaces.
5. The dMNW must provide a dedicated on-tray interface to the dBMC.
6. The dMNW must provide a dedicated on-tray interface to each dBRICK.
7. The Ethernet connectors for dMNW must be placed on the front of the dTRAY and be accessible on the front panel of the dBOX.
8. The dMNW on the dTRAY must not use any PHY devices.
9. The dMNW on the dTRAY must use SGMII interfacing.
10. The dTRAY must provide one LED to indicate the link state and activity for each on-tray dMNW interface.
11. The dTRAY must provide three LEDs to indicate the link state, link speed and activity for each off-tray dMNW interface.

8.2.5 Accelerator Network

The accelerator network (ACCELNW) is a dedicated high speed connection to a CFP2 connection. While this interface could be used for several purposes the initially supported interface will be a 100 GbE network interface.

**d5.1-req-10: Accelerator Network:**

1. The dTRAY must provide 2 CFP2 compatible connectors as described in reference [4] and [5].
2. The dTRAY must support CFP2 modules with power class 4.
3. The dTRAY must support CFP2 modules with with a 10x10G interface.
4. The high speed signals to the CFP2 connector must be connected to the dBESM.
5. The low speed control signals to the CFP2 connector must be connected to the dACCELBRICK.
8.3 dReDBox tray requirements

d5.1-req-20: dTRAY Boundary scan:
The dTRAY must support several distinct boundary scan chains which can be used for validation and testing.

1. The dBMC must have a dedicated boundary scan chain.
2. All devices on the dTRAY which have boundary scan support must be included in the dTRAY boundary scan chain.
3. The dBRICKs must have a dedicated boundary scan chain.

A boundary scan chain is only accessible if each device in the chain is functioning. It is necessary to isolate devices from the chain to allow for localized testing and verification.

4. Each device present in a boundary scan chain on the dTRAY must be able to be bypassed using a toggle switch.
5. Each dBRICK in a boundary scan chain must be able to be bypassed using a toggle switch.

There is no standard connector or pinout for boundary scan connectivity. Therefore connectors are matched to the tools most likely to be used on the different chains.

6. The boundary scan chain for the dBRICKs must be fitted with a connector compatible with the Xilinx USB II Platform cable (refer to [10]).
7. The boundary scan chain for the dBMC must be fitted with a connector compatible with the Xilinx USB II Platform cable (refer to [10]).
8. The boundary scan chain for the dTRAY must be fitted with an ARM JTAG 20 connector (refer to [11]).

d5.1-req-21: dTRAY status LEDs:
To indicate the status of a running dBOX a number of status LEDs are defined.

1. The dTRAY must have 3 LEDs; red, orange and green to indicate the status of the dBMC.
2. The dTRAY must have 3 LEDs; red, orange and green to indicate the status of each dBRICK.
3. All dTRAY status LEDs must be placed on the front of the dTRAY and be visible on the front panel of the dBOX.

d5.1-req-22: dTRAY power management:
The dTRAY will be powered by a standard (COTS) power supply. This power supply will be controlled by the dBMC.

1. The dBMC must control the dBOX-PSU
2. The orchestration software must be able to switch the operation modes of the dBRICKs using the management network.
3. The orchestration software must be able to switch from stand-by mode to operational mode using the management network.
d5.1 – Hardware requirements of all bricks and tray (a)

**d5.1-req-23: dTRAY dBOX Optical Switch Matrix support:**
The dBOX will support up to 3 internal optical switches (Polatis Series 6000 48xCC OSM). This switch has 48 ports and there is a 96 ports switch in development which will be compatible and interchangeable.

1. The dTRAY must provide power and control to 3 Polatis Series 6000 48xCC OSM devices (dBOSM).

The dBOSM is configured and controlled by the dBMC, which in turn is controlled by the provisioning software (refer to Figure 7).

2. The dBOSM must be controlled by the dBMC.

The Polatis switch uses a D-Sub DB15 connector for power and control (refer to [13]). This connector is not suitable for use on the dTRAY. Alternatively, we will use a standard box header and a flat-to-DB15 cable.

3. The dBOSM connector on the dTRAY must be a 2.54mm, 2x8 type shrouded connector.
4. The dBOSM interface must be pin compatible with the DB15 connector as defined in [13].

The selected optical switch device has a power requirement of 5W, but to allow for future upgrade to a bigger optical switch device there is some margin on this requirement.

5. The dBOSM interface must be able to provide at least 15W@12V.

**d5.1-req-24: dTRAY MAC address allocation:**
There will be a lot of networked devices in the dBOX which all need a unique MAC address. The MAC addresses used will be determined by the location and function of the device. The MAC address range will be taken from the Scalys (refer to [15]) pool. Scalys is a subsidiary of Sintecs, and the specified pool of MAC addresses will be reserved for dReDBox use, making sure we have unique addresses in the system without the need for dedicated addresses programmed in the system.

1. The MAC addresses used in the dReDBox must be in the range of 6C:95:22:01:00:00 to 6C:95:22:01:FF:FF.
2. The MAC addresses used will be generated using the dTID and dBID.
   a. dBRICK: base MAC address, + dTID *32 + dBID.
   b. dBMC management: base MAC address + dTID * 32 + 16
   c. dBMC debug: base MAC address + dTID * 32 + 17
   d. dACCEL network: base MAC address + dTID * 32 + 18
   e. base MAC address + dTID * 32 + 19 to base MAC address + dTID * 32 + 31 are reserved for future use.

**d5.1-req-55: dTRAY expansion header:**
1. The dTRAY must provide a connector which provides:
   a. Dedicated I²C interface to the dBMC, the I²C levels must be selectable between 3V3 and 5V.
   b. 3V3, 250mA power
   c. 5V0, 250mA power
8.3.1 dBMC requirements

Figure 7 – dReDBox dBMC overview

d5.1-req-25: dBMC boot source selection:
During development of the system it is desirable to have multiple boot sources for the dBMC. Using toggle switches and the micro-SD card detect function provides a flexible system, suitable for prototyping and production environments.

1. The dBMC must be able to boot from the Quad Serial Peripheral Interface (QSPI) NVM (Non-Volatile Memory).
2. The dBMC must be able to boot from the micro-SD card.
3. The dBMC must have a toggle switch which controls the boot source:
   a. QSPI NVM.
   b. Micro-SD card.
4. The dBMC must have a toggle switch which controls the type of boot:
   a. Full boot from NVM
   b. Bootstrap from NVM, secondary boot from Network
5. The dBMC must have a toggle switch which overrides the boot source:
   a. Boot source selected through “boot source toggle switch”.
b. Boot source selected through “Micro-SD card card-detect switch”

**d5.1-req-26: dBMC temperature sensors:**

1. The dBMC must have a temperature sensor which can monitor the temperature at the front of the dTRAY.
2. The dBMC must have a temperature sensor which can monitor the temperature at the back of the dTRAY.
3. The temperature sensors must be placed on the hottest spots as determined through a thermal simulation.
4. The temperature sensors must have an accuracy of ±1 °C.
5. The temperature sensors must have a resolution of at least 1 °C.
6. The temperature sensor must have a refresh rate of at least 1 Hz.
7. The temperature sensors must have a sensing range of at least 0 °C to 70 °C.

**d5.1-req-27: dBMC fan controller:**

The dBOX will have fans which generate the airflow needed to keep the system temperature within operational range. Depending on the load of the system, and the ambient temperature the required airflow will differ, and it is desirable to be able to control the fans to reduce the fans power consumption and noise, when allowable.

1. The dBMC must have support for 4 fans.
2. The dBMC must support all fans as described in the 4-wire Pulse Width Modulation (PWM) controlled fans specification (refer to [12]).
3. The fans must be set to full power when they are not actively controlled.
4. The dBMC must be able to control each fan duty cycle from 0-100% with a resolution of at least 1%.
5. The dBMC must be able to monitor the speed of each fan.

**d5.1-req-28: dBMC NVM storage:**

1. The dBMC must have at least 16 MiB of QSPI NOR NVM.
2. The dBMC QSPI must be able to operate at a speed of at least 100MHz.
3. The dBMC must have a socket for a micro-SD card.
4. The dBMC micro-SD card interface must support micro-SD cards up to the SDR104 interface standard (refer to [8]).

**d5.1-req-29: dBMC local RAM memory:**

1. The dBMC must have at least 2 GiB of DDR4 memory.
2. The dBMC must have ECC memory.
3. The dBMC must have a memory bus of 64+8 bit.
4. The dBMC memory must be able to operate on a speed of at least 2400 GT/s.
D5.1 – Hardware requirements of all bricks and tray (a)

d5.1-req-30: dBMC Power measurement:
To allow the orchestration software to monitor load vs power consumption and allow it to optimize system behavior we need to provide fine grained information on the power consumption of the system.

1. The dBMC must be able to measure the total power consumption of each dBRICK.
2. The dBMC must be able to measure the power consumption of the entire dBOX.
3. All power measurements must have a resolution of at least 0.5W.
4. All power measurements must have an accuracy of at least 1W.
5. All power measurements must have a refresh rate of at least 1Hz.

d5.1-req-31: dBMC debug:
To assist in board bring-up and system debugging several debug interfaces are defined.

1. LEDs
   a. The dBMC must have a red and a green debug LED connected to the Processing System (PS) side of the System On Chip (SoC).
   b. The dBMC must have a red and a green debug LED connected to the Programmable Logic (PL) side of the SoC.
2. Ethernet Network
   a. The dBMC must have a dedicated network interface with support for 10BASE-T, 100BASE-T and 1000BASE-T networking as defined in the IEEE 802.3 standard (refer to [6]).
   b. The Ethernet connector must be placed on the front of the dTRAY and be accessible on the front panel of the dBOX.
3. Console
   a. The dBMC must have a serial console which provides access to the PS of the SOC.
   b. The console must be a universal asynchronous receiver/transmitter (UART), supporting 115k2, 8N1 protocol and must have no flow control signals.
   c. The UART will be bridged on the dTRAY to a USB UART interface.
   d. The USB UART must have a USB 2.0, type B connector, placed on the front of the dTRAY and be accessible on the front panel of the dBOX.
D5.1 – Hardware requirements of all bricks and tray (a)

d5.1-req-32: dBMC dBRICK I2C interface:
To reduce the number of discrete I/O between the dBRICKs and the dBMC an I²C interface for certain slow and static configuration options is used.
1. There must be a dedicated I²C bus (dSCI2C) between the dBMC and each dBRICK.
2. The dSCI2C bus must be multi-master.
3. The dSCI2C must have the pull-up resistors required by the I²C bus on the dTRAY.
4. The dSCI2C must be used to identify the type of dBRICK by the dBMC.
5. The dSCI2C must be used to configure the dBRICK.
   a. Set the location/ID of the dBRICK.
   b. Set the MAC address of the dBRICK.

d5.1-req-33: dBMC dBRICK UART interface:
To provide a way to monitor and debug the dBRICKs a UART interface is used. It is not efficient to add a UART interface connector to the system for each dBRICK, so the dBMC is used to serve as a bridge between the user and the dBRICK UARTs.
This interface is the dReDBox dBrick Serial Console (dBSC)
1. There must be a dedicated UART interface (dBSC) between the dBMC and each dBRICK.
2. The dBSC must be a universal asynchronous receiver/transmitter (UART), supporting 115k2, 8N1 protocol and must have no flow control signals.
8.4 dBRICK requirements

The dBRICK requirements are a set of requirements which are common to the different types of dBRICKs. These requirements are used to guarantee compatibility between the different types of dBRICKs.

Figure 8 – dBRICK overview
D5.1 – Hardware requirements of all bricks and tray (a)

**d5.1-req-34: dBRICK mechanical constraints:**

There are several restrictions on the mechanical design of the different dBRICKs. These requirements have been created using a pre-placement study and the requirements put on the system at a higher level.

Refer to Figure 9 for an overview of the sizes and keep out areas of the dBRICK.

![Figure 9 – dBRICK Dimensions](image_url)

1. The dBRICK should have a length of 180 mm.
2. The dBRICK should have a height of 67 mm.

To provide mechanical stability to the dBRICKs located on the tray, the dBRICK must keep a clear area on its edges. This area where no components must be placed can be used by guide rails, securing the dBRICK to the dTRAY.

3. The dBRICK must have a keep out area where no components can be placed of 2 mm at each short side of the dBRICK.
4. The dBRICK must have a maximum component height of 15 mm on the bottom side.
5. The dBRICK must have a maximum component height of 15 mm on the top side.

**d5.1-req-35: dBRICK connector boundary scan:**

1. The dBRICK must have a boundary scan chain to allow for board validation and debugging.
2. All devices on the dBRICK which have boundary scan support must be present in the boundary scan chain.

A boundary scan chain is only accessible if each device in the chain is functioning. It is necessary to isolate devices from the chain to allow for localized testing and verification.

3. Each device present in the boundary scan chain on the dBRICK must be able to be bypassed using the dSCI2C bus. This requirement is not valid when there is only one device in the chain.
D5.1 – Hardware requirements of all bricks and tray (a)

**d5.1-req-36: dBRICK connector power:**
1. The dBRICK must operate on a 12V power supply.
2. The dBRICK must not consume more than 60W.

**d5.1-req-38: dBRICK connector clock sources:**
1. The dBRICK may operate on a 100MHz input clock.
2. The dBRICK must derive the clock used for the HSENW and HSONW from the 100MHz input clock.

**d5.1-req-37: dBRICK electrical interface requirements:**
This requirement specifies the interface between the dBRICK and the dTRAY.

1. The interface between the dBRICK and dTRAY must be specified in a separate specification document which specifies:
   a. Mechanical constraints.
   b. Connector specification and pinout.
   c. Electrical specifications and requirements.

**d5.1-req-39: dBRICK optical network:**
All the different types of dBRICKs must be connected to the HSONW. This connection is achieved using MBO modules.

1. The dBRICK must have support for 1 Luxtera LUX62608 MBO module.
2. The dBRICK must provide a dedicated interface for the MBO module’s control and status interface (refer to [14]).

**d5.1-req-54: dBRICK debug:**
To assist in board bring-up and debugging we define several debug interfaces.

1. The dBRICK must have a red and a green debug LED connected to the PS side of the SoC.
2. The dBRICK must have a red and a green debug LED connected to the PL side of the SoC

### 8.4.1 Compute brick requirements

**d5.1-req-40: dCOMPUBRICK SoC choice:**
To reduce risk and decrease development time the project team has decided to use Xilinx Ultrascale+ MPSoC device as the SOC for the compute brick.

1. The dCOMPUBRICK must use a XCZU9EG-2FFV900E SOC.

**d5.1-req-41: dCOMPUBRICK local NVM storage:**

1. The dCOMPUBRICK must have at least 16 MiB of QSPI NOR NVM.
2. The dCOMPUBRICK QSPI must be able to operate at a speed of at least 100MHz
3. The dCOMPUBRICK must have a socket for a micro-SD card.
4. The dCOMPUBRICK micro-SD card interface must support micro-SD cards up to the SDR104 interface standard (refer to [8]).
D5.1 – Hardware requirements of all bricks and tray (a)

d5.1-req-42: dCOMPUBRICK local RAM memory:
1. The dCOMPUBRICK must have at least 4 GiB of DDR4 memory.
2. The dCOMPUBRICK must have ECC memory.
3. The dCOMPUBRICK must have a memory bus of 64+8 bit.
4. The dCOMPUBRICK memory must be able to operate on a speed of at least 2400 GT/s.

d5.1-req-43: dCOMPUBRICK PCIe:
1. The dCOMPUBRICK must have a 2x2 PCIe link.

d5.1-req-44: dCOMPUBRICK Management:
The Ethernet network on the dCOMPUBRICK will not support a direct connection to a physical network, but it is designed to connect to an Ethernet switch located on the dTRAY. This removes the need for a PHY and magnetics on the dCOMPUBRICK and dTRAY.

1. The dCOMPUBRICK must provide a SGMII based network interface.
2. The dCOMPUBRICK must support 1000BASE-T networking as defined in the IEEE 802.3 standard (refer to [6]).
3. The dCOMPUBRICK must not use any PHY devices.

8.4.2 Memory brick requirements
To provide a lot of memory capacity in a cost effective way standard memory modules are used. Due to mechanical constrains SO-DIMM modules will be used. This is because there are no connectors available on the market with a right-angle module insertion.

d5.1-req-45: dMEMBRICK Memory technology:
1. The dMEMBRICK must use DDR4, SO-DIMM modules compatible with JEDEC standard No.21C (refer to [7]).
2. The dMEMBRICK must support SO-DIMM modules with and without ECC.
3. The dMEMBRICK must support the Serial Present Detect feature.

d5.1-req-46: dMEMBRICK Memory capacity:
The JEDEC standard defines SO-DIMM modules with a capacity of up to 256 GiB. While these capacities are not available at the time of writing, these modules must be supported for future upgrades.
1. The dMEMBRICK must support 2 memory modules, each with a capacity of at least 256 GiB.

8.4.3 Accelerator brick requirements
The Accelerator brick is an FPGA resource with a high-speed interface to an external interface. The accelerator brick must be placed in a special location on the dTRAY to make full use of the additional external interface. The designed function of the accelerator brick is to provide a 100GbE interface, but the design is flexible so this can be expanded to other functions.
D5.1 – Hardware requirements of all bricks and tray (a)

d5.1-req-47: dACCELBRICK Accelerator Network:
1. The dACCELBRICK must provide 10 HSENW interfaces to the dTRAY.
2. The dACCELBRICK must provide a dedicated interface to the dTRAY to interface with the CFP2 management interface (refer to [4]).
3. The dACCELBRICK must provide a dedicated interface to the dTRAY to interface with the Hardware Alarm Pins (refer to [4]).
4. The dACCELBRICK must provide 8 GPIO pins to the dTRAY for future expansion.

d5.1-req-48: dACCELBRICK clocking:
The dACCELBRICK requires a different clock from the other bricks. This clock is dictated by the used protocol.
1. The dACCELBRICK must use a dedicated clock dABCLK.
2. The dABCLK must provide a clock with a frequency compatible with the 100GbE protocol.

8.5 Peripheral tray requirements

The peripheral tray will be a COTS part, and no requirements on the dPERTRAY will be specified on this level.
8.6 Test tray (dTETRAY) requirements

The dTETRAY will serve as an initial platform to debug the dBRICKs. The dTRAY is very space constrained, which makes measuring and debugging the hardware very challenging. By building the dTETRAY as an intermediate platform it is possible to remove these space constraints, and leave more options for testing and debugging features.

The dTETRAY is a more affordable platform than the dTRAY. While it does not have the full feature set of the dTRAY it can be used by the partners during the initial bring up of the FPGAs and the orchestration software.

The dTETRAY will support a subset of the functionality of the dTRAY. Care must be taken to have as much reuse from the dTETRAY to the dTRAY as possible.

Figure 10 – dReDBox Test Tray overview
D5.1 – Hardware requirements of all bricks and tray (a)

**d5.1-req-49: dTETRAY Mechanical constraints:**
The dTETRAY will be used as a platform for the initial hardware verification
tests and should therefore allow for easy access to the different components.
1. The dTETRAY must have a mATX compatible form factor and footprint (refer to [9]).
2. The dTETRAY must have at least 3 cm of space around the dBICKs in which no active components must be placed.
3. The dTRAY must have at least 10 cm of space between different dBICKs.
4. The dTRAY must not have any active components on the bottom side of the PCB.

**d5.1-req-50: dTETRAY dBICK support:**
1. The dTETRAY must support 3 dBICKs.
   a. The dTETRAY must support 2 dBICKs.
   b. The dTETRAY must support 1 dACCELBRICK.

**d5.1-req-51: dTETRAY Networking:**
1. Two of the dBICKs must be connected to each other via the HSENW.
2. The third dBICK must have the HSENW signals connected to a high speed connector.
3. The dTETRAY must have a management network, conform the management network requirements of the dTRAY.
4. The dTETRAY must have a PCIe network, conforming the PCIe Network requirements of the dTRAY.
5. The dTETRAY must not have an dBESM.
6. The dTETRAY must support one dBOSM as specified in d5.1-req-23.

**d5.1-req-52: dTETRAY BMC support:**
1. The dTETRAY dBMC function must be identical to the dBMC function on the dTRAY.
2. Interfaces which are not available on the dTETRAY, but which are present on the dBMC must be put in a safe default state.
3. Interfaces which are not available on the dTETRAY, but which are present on the dBMC must be available on pin headers.

**d5.1-req-53: dTETRAY Power:**
1. The dTETRAY must be powered by an ATX power supply.
9 System Configuration and Boot flow

This chapter describes the configuration and boot flow, which follow from the specified requirements. This is meant as a possible implementation example and does not set any requirements.

9.1 Cold start to stand-by state

1. Power cable is plugged in to dBOX.
2. dBPSU generates 5V stand-by power.
3. dBMC boot.
4. dBMC checks dTID.
5. dBMC initiates and configures:
   a. Management network, network addresses based on dTID.
   b. PCIe Network, memory map based on dTID.
6. If dTID is zero, it will assume the orchestrator role.
7. If dTID is not zero, it will register itself with the orchestrator (dTID 0).
8. dBMC initialized.
9. The dBMC checks and registers available dBRICKs.
10. The dBMC will go to idle state.

9.2 Stand-by to operational state

State changes are controlled by the orchestrator.
1. The orchestrator will send a provisioning request to the dBMC.
2. The dBMC checks if the dBRICK is present.
3. The dBMC configures the I²C “ROM” for dBRICK configuration.
4. The dBMC sets the boot source for the dBRICK.
5. The dBMC will enable power to the dBRICK.
6. The dBRICK will boot.
7. The dBRICK will set the ready pin.
8. The dBRICK will read the configuration ROM.
9. The dBRICK will register itself with the orchestrator.
10. dBRICK is ready for operation.

9.1 Operational to stand-by state

State changes are controlled by the orchestrator.
1. The orchestrator will decommission all running resources on the dBRICK.
2. The orchestrator will send a shutdown command to the dBRICK.
3. The dBRICK will shut down Linux.
4. The dBRICK will clear the ready pin.
5. The dBMC will disable power to the dBRICK.
10 Requirement List

This chapter lists all the requirements and is referred to in the description of the requirement. This list is used to enforce requirement numbering. After the first release of this document, this numbering must never change. New and modified requirements will be added to this list, striking though, but not removing the obsolete requirements. This to prevent broken references in derived documents.

d5.1-req-1: global component restrictions
d5.1-req-2: global reference clocking
d5.1-req-3: global design for testability
d5.1-req-4: global design system states
d5.1-req-5: hot-plug support
d5.1-req-6: High Speed Electrical Network
d5.1-req-7: High Speed Optical Network
d5.1-req-8: PCIe Network
d5.1-req-9: Management Network
d5.1-req-10: Accelerator Network

d5.1-req-11: dBOX Enclosure constraints
d5.1-req-12: dBOX Available space dTRAY
d5.1-req-13: dBOX Connector locations
d5.1-req-14: dBOX Cooling
d5.1-req-15: dBOX power supply
d5.1-req-16: dBOX supported dBRICK count
d5.1-req-17: dBOX dBRICK numbering

d5.1-req-18: dTRAY tray location identification
d5.1-req-19: dTRAY dBRICK connector
d5.1-req-20: dTRAY Boundary scan
d5.1-req-21: dTRAY status LEDs
d5.1-req-22: dTRAY power management
d5.1-req-23: dTRAY dBOX Optical Switch Matrix support
d5.1-req-24: dTRAY MAC address allocation
d5.1-req-25: dBMC boot source selection
d5.1-req-26: dBMC temperature sensors
d5.1-req-27: dBMC fan controller
d5.1-req-28: dBMC NVM storage
d5.1-req-29: dBMC local RAM memory
d5.1-req-30: dBMC Power measurement
d5.1-req-31: dBMC debug
d5.1-req-32: dBMC dBRICK I²C interface
d5.1-req-33: dBMC dBRICK UART interface
d5.1-req-34: dBRICK mechanical constraints
d5.1-req-35: dBRICK connector boundary scan
d5.1-req-36: dBRICK connector power
d5.1-req-37: dBRICK electrical interface requirements
d5.1-req-38: dBRICK connector clock sources
d5.1-req-39: dBRICK optical network

d5.1-req-40: dCOMPUBRICK SoC choice
d5.1-req-41: dCOMPUBRICK local NVM storage
d5.1-req-42: dCOMPUBRICK local RAM memory
d5.1-req-43: dCOMPUBRICK PCIe Interface
d5.1-req-44: dCOMPUBRICK Management Interface

d5.1-req-45: dMEMBRICK Memory technology
d5.1-req-46: dMEMBRICK Memory capacity

d5.1-req-47: dACCELBRICK Accelerator Network
d5.1-req-48: dACCELBRICK clocking

d5.1-req-49: dTETRAY Mechanical constraints
d5.1-req-50: dTETRAY dBRICK support
d5.1-req-51: dTETRAY Networking
d5.1-req-52: dTETRAY BMC support
d5.1-req-53: dTETRAY Power

d5.1-req-54: dBRICK debug
d5.1-req-55: dTRAY expansion header
11 Notes

11.1 Abbreviations and Acronyms

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCELNW</td>
<td>Accelerator Network</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial off-the-shelf</td>
</tr>
<tr>
<td>dABCLK</td>
<td>dReDBox Accelerator Brick Clock</td>
</tr>
<tr>
<td>dACCELBRICK</td>
<td>dReDBox-Accelerator-Brick</td>
</tr>
<tr>
<td>dBACT</td>
<td>dReDBox-dBRICK Active</td>
</tr>
<tr>
<td>dBESM</td>
<td>dReDBox-Box Electrical Switch Matrix</td>
</tr>
<tr>
<td>dBID</td>
<td>dReDBox-Box IDentifier</td>
</tr>
<tr>
<td>dBMC</td>
<td>dReDBox-Board Management Controller</td>
</tr>
<tr>
<td>dBOSM</td>
<td>dReDBox-Box Optical Switch Matrix</td>
</tr>
<tr>
<td>dBOX</td>
<td>dReDBox-Box</td>
</tr>
<tr>
<td>dBPSU</td>
<td>dReDBox-Box PSU</td>
</tr>
<tr>
<td>dBRICK</td>
<td>dReDBox-Brick</td>
</tr>
<tr>
<td>dBSC</td>
<td>dReDBox-dBrick Serial Console</td>
</tr>
<tr>
<td>dCLUST</td>
<td>dReDBox-Cluster</td>
</tr>
<tr>
<td>dCOMPUBRICK</td>
<td>dReDBox-Compute-Brick</td>
</tr>
<tr>
<td>dHPI</td>
<td>dReDBox-Hot Plug Indicator</td>
</tr>
<tr>
<td>dHPS</td>
<td>dReDBox-Hot Plug Switch</td>
</tr>
<tr>
<td>dBMBRICK</td>
<td>dReDBox-Memory-Brick</td>
</tr>
<tr>
<td>dMNW</td>
<td>dReDBox-Management NetWork</td>
</tr>
<tr>
<td>dPD</td>
<td>dReDBox-Present Detect</td>
</tr>
<tr>
<td>dPERTRAY</td>
<td>dReDBox-peripheral-tray</td>
</tr>
<tr>
<td>dRACK</td>
<td>dReDBox-RACK</td>
</tr>
<tr>
<td>dROSM</td>
<td>dReDBox-Rack Optical Switch Matrix</td>
</tr>
<tr>
<td>dSCDC</td>
<td>dReDBox Status Control Debug Console</td>
</tr>
<tr>
<td>dSCI2C</td>
<td>dReDBox-Status Config I²C</td>
</tr>
<tr>
<td>dTETRAY</td>
<td>dReDBox-Test-Tray</td>
</tr>
<tr>
<td>dTID</td>
<td>dReDBox-Tray Identifier</td>
</tr>
<tr>
<td>dTRAY</td>
<td>dReDBox-Tray</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field Programmable Gate Array</td>
</tr>
<tr>
<td>GbE</td>
<td>Gigabit Ethernet</td>
</tr>
<tr>
<td>GiB</td>
<td>GibiByte: (2^{30}) Bytes.</td>
</tr>
<tr>
<td>HSENW</td>
<td>High Speed Electrical Network</td>
</tr>
<tr>
<td>HSONW</td>
<td>High Speed Optical Network</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output</td>
</tr>
<tr>
<td>I²C</td>
<td>Inter-Integrated Circuit</td>
</tr>
<tr>
<td>LED</td>
<td>Light Emitting Diode</td>
</tr>
<tr>
<td>MAC address</td>
<td>Media Access Control address</td>
</tr>
<tr>
<td>MBO</td>
<td>Mid Board Optics</td>
</tr>
<tr>
<td>MiB</td>
<td>Mebibyte</td>
</tr>
<tr>
<td>NVM</td>
<td>Non Volatile Memory</td>
</tr>
<tr>
<td>OSM</td>
<td>Optical Switch Matrix</td>
</tr>
<tr>
<td>PCB</td>
<td>Printed Circuit Board</td>
</tr>
<tr>
<td>PCIe</td>
<td>Peripheral Component Interconnection Express</td>
</tr>
<tr>
<td>PCIENW</td>
<td>PCIE Network</td>
</tr>
<tr>
<td>PHY</td>
<td>Physical Layer Device</td>
</tr>
<tr>
<td>PI</td>
<td>Power Integrity</td>
</tr>
</tbody>
</table>
D5.1 – Hardware requirements of all bricks and tray (a)

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PL</td>
<td>Programmable Logic</td>
</tr>
<tr>
<td>PS</td>
<td>Processing System</td>
</tr>
<tr>
<td>PSU</td>
<td>Power Supply Unit</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse Width Modulation</td>
</tr>
<tr>
<td>QSPI</td>
<td>Quad Serial Peripheral Interface</td>
</tr>
<tr>
<td>SI</td>
<td>Signal Integrity</td>
</tr>
<tr>
<td>SMF</td>
<td>Single Mode Fiber</td>
</tr>
<tr>
<td>SOC</td>
<td>System On Chip</td>
</tr>
<tr>
<td>SO-DIMM</td>
<td>Small Outline Dual In-line Memory Module</td>
</tr>
<tr>
<td>SPD</td>
<td>Serial Present Detect</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver/Transmitter</td>
</tr>
</tbody>
</table>