IOWN GLOBAL FORUM ™

Data-Centric Infrastructure Cluster Reference Implementation Models

Classification: APPROVED REFERENCE DOCUMENT

Confidentiality: PUBLIC

Version 1

March 2025

[DCI Cluster RIM]

Legal

Contents

List of Figures

List of Tables

1. Introduction

Today's communication and computing infrastructures are approaching the limits of their utility. Emerging use cases have higher and more complex demands for communication and computation and simultaneously mandate drastic improvements in energy efficiency. The IOWN Global Forum (IOWN GF) has been investigating such use cases and identified the technical gaps barring their realization. New infrastructures are required that meet the different and increasingly challenging needs of the various kinds of data and greatly increase the flexibility and quality regarding how and where data is transferred, processed, and stored. These findings are summarized in the IOWN Global Forum Data-Centric Infrastructure Product Concept Paper [**IOWNGF-DCIPCP**].

To answer these use case needs, the IOWN Global Forum has developed the Data-Centric Infrastructure (DCI) Functional Architecture (FA) [**IOWNGF-DCIFA2**]. This FA lays out the structure of the DCI Systems as proposed in [**IOWNGF-DCIPCP**]. These up-to-global-scale geographically distributed DCI Systems enable flexible provisioning of logical servers composed out of resource pools and interconnecting servers even thousands of kilometers apart. Improving over classic infrastructures, these DCI Systems transmit at hundreds and more Gbps of bandwidth and at a quality-of-service level sufficient even for carrying remote direct memory access (RDMA) protocols.

A key part of the DCI FA and DCI Systems is the DCI Cluster. DCI Clusters contain physical computing, storage, and networking devices. DCI Clusters utilize these devices to provide users with logical servers. There are no restrictions regarding users of DCI Clusters; for example, users could be organizational end-users, private individuals, intermediate service providers, or user application orchestration systems. User can deploy their workloads on DCI Cluster hardware by providing initialization or boot images for these logical servers. In addition, DCI Clusters take care of connecting logical servers inside to each other and WANs with quality-of-service guarantees through appropriate networking hardware.

Therefore, in some sense, the role of DCI Clusters is somewhat similar to data centers; however, while "data center" is an informal term, the capabilities of DCI Clusters are defined by the DCI functional architecture, and DCI Clusters may have any size both small and large, containing any number of devices. For further information about the DCI functional architecture, readers are referred to [**IOWNGF-DCIFA2**].

This document aims to accelerate the implementation of IOWN Global Forum use cases and their businesses by fostering the development of DCI Clusters and, thus, the implementation of DCI Systems as a whole.

To reach this goal, the objectives of this document are set as follows:

- illustrate how DCI Clusters satisfying the performance requirements of future IOWN Global Forum use cases can be implemented today;
- estimate the performance and scale of DCI Clusters constructed in the above manner; and
- foster the creation of IOWN Global Forum use case reference implementation models (RIMs) based on DCI Systems and DCI Clusters.

Considering these objectives, the scope of this document is limited to the following:

- create DCI Cluster RIMs (DCRs) using only technology that is already available today;
- describe the structure of the data plane, control plane, and management plane for these DCI Cluster RIMs; and
- focus on the overall DCI Cluster design and pointing out alternative choices rather than advocating a single detailed solution.

The resulting DCRs presented in this document give an overview over the design space for DCI Clusters. However, after pointing out possible directions, a number of issues are remaining for implementers. The DCRs shown later on should therefore be regarded as interim versions.

The remainder of this document is structured as follows: First, the workload to be executed on initial DCI Clusters is briefly revisited in **Section 2**. Then, the main functional blocks of DCI Clusters are briefly highlighted in **Section 3**. The main part of this document, **Section 4**, displays DCI Cluster RIMs and discusses applicability to use cases, names concrete technologies proposed for implementation, and quantifies the resulting DCI Cluster capabilities. Finally, **Section 5** reflects upon the document concerning further activities toward realizing DCI Clusters.

Readers who are already familiar with IOWN Global Forum use cases and data-centric-infrastructure and who are primarily interested in the resulting DCI Cluster reference implementation models are encouraged to skip ahead to the DCI Cluster RIMs and directly start reading **Section 4**, the main part of this document.

Moreover, readers interested in the details regarding the construction of DCI Cluster RIMs itself are referred to the appendices: **Appendix A** discusses individual IOWN Global Forum use cases and the accompanying requirements, **Appendix B** introduces a common framework to construct DCI Clusters, and **Appendix C** contains templates to derive DCI Cluster RIMs from.

2. Workloads for DCI Clusters: IOWN Global Forum use cases overview

This section briefly considers the requirements of IOWN Global Forum use case applications for Data-Centric Infrastructure (DCI) Clusters and illustrates the need for multiple DCI Cluster RIMs: **Section 2.1** presents a brief overview of IOWN Global Forum use case applications. This overview is followed by considerations in **Section 2.2**, which explain how the diversity in requirements warrants multiple DCI Cluster designs to cater to different use case classes.

2.1. IOWN Global Forum use case applications overview

An overview of IOWN Global Forum use cases and their required data center resources and communication topologies is presented in **Table 2.1**. In this overview, use cases are classified by the required communication topology and their server resource requirements. In addition, references to DCI Cluster RIM sections are listed for each identified use case class, which may serve as a starting point for developing further customized use case RIMs.

Table 2.1: Relation between resource types, communication topologies, use cases, and DCI Cluster RIMs.

		COMMUNICATION TOPOLOGY		
Main resources required in data center		**Far point-to-point**	**Near group-to-group**	**Both far point-to-point and near group-to-group**
	CPU-centric or no particular accelerator needs	• Financial industry services infrastructure • Remote-controlled robot inspection → e.g., Sec. 4.3		
	Accelerator- centric (e.g., GPUs or DSPs)	• Remote Media Production • AIC Interactive Live Music • Sovereign cloud • VRAN MFH eCPRI (requiring ultra-low latency switching) → e.g., Sec. 4.2, Sec. 4.3, Sec. 4.4	• Large ML model training (any-to-any communication) → e.g., Sec. 4.1, Sec. 4.2, Sec. 4.4	• Green Computing with Remote GPU over APN → e.g., Sec. 4.1, Sec. 4.2, Sec. 4.3, Sec. 4.4
	CPUs + storage	• IDH: off-site replication → e.g., Sec. 4.1, Sec. 4.3	• IDH: compute-storage disaggregation (tiered) → e.g., Sec. 4.1, Sec. 4.4	• IDH: off-site replication and compute-storage disaggregation → e.g., Sec. 4.1, Sec. 4.4
	CPUs + storage + GPUs			• CPS Area Management with live 4D map database → e.g., Sec. 4.1, Sec. 4.3, Sec. 4.4

Furthermore, **Table 2.2** summarizes the DCI Cluster RIMs referenced by **Table 2.1**. A more detailed overview of the DCI Cluster RIMs can be found at the beginning of **Section 4** in **Table 4.1**.

Table 2.2: DCI Cluster Reference Implementation Models introduced in this document.

SECTION	DCI CLUSTER RIM DESCRIPTION
Section 4.1	DCI Cluster RIM based on compute racks with composable disaggregated infrastructure
Section 4.2	DCI Cluster RIM based on a scalable, QoS-assured, and lossless network for supporting GPU-to-GPU communication
Section 4.3	DCI Cluster RIM based on elastic edge computing infrastructure with small-scale rack components
Section 4.4	DCI Cluster RIM based on huge and scalable disaggregated infrastructure with PCIe bus extension

While each use case has its specific requirements, considering the presented classification, the following patterns can be identified among the IOWN Global Forum use cases:

Communication topologies: Two main patterns of communication topologies stand out when considering IOWN Global Forum use cases: First, connections via WAN between two endpoints that are geographically apart (“far point-to-point”), and second, connections between groups of nodes that are located in close vicinity to each other (“near group-to-group”), often in a tiered fashion. In addition, there are use cases requiring hybrids of these two topologies (“both far point-to-point and near group-to-group“). Furthermore, for many (but not all) IOWN Global Forum use cases, end-to-end communication being virtually lossless at 100 Gbps-order bandwidth and one millisecond-order latency is sufficient.

Main resources required in data centers: When examining the server resources of IOWN Global Forum use cases, three main patterns become prominent. First, a group of use cases is comprised of workloads that do not have particular needs for accelerators but for which CPUs are sufficient or for which the focus is mainly on communication (“CPU-centric or no particular accelerator needs“). Second, multiple use cases require substantial amounts of the same accelerator type, such as GPUs or DSPs (“Accelerator-centric (e.g., GPUs or DSPs)“). Third, primarily database-related use cases require large numbers of both CPUs and storage resources (“CPUs + storage“). In addition to these three types, a fourth group is formed by hybrids of the former, requiring CPUs, storage, and accelerators in large numbers simultaneously (“CPUs + storage + GPUs“). Furthermore, for many IOWN Global Forum use cases that imply the use of accelerators, GPUs are an efficient choice. In addition, variations of the above classes exist in the exact type of hardware required, with use cases benefitting from CPUs, GPUs, FPGAs, signal processing cards, IPUs/DPUs, SmartNICs, memories, storage, and other specialized devices.

Further details about the individual use cases can be found in Appendix A.

2.2. Workload considerations for DCI Cluster construction

The IOWN Global Forum use cases shown in **Section 2.1** and **Table 2.1** have varying requirements regarding communication topology and QoS. They also have varying requirements regarding the types of main resources and the number of different resource types needed. This means that, even under the straightforward classification introduced in **Section 2.1**, IOWN Global Forum use case application requirements regarding hardware are highly diverse.

As a result, DCI Cluster designs that can equally accommodate all IOWN Global Forum use cases would likely need to trade off efficiency and resource utilization to achieve such a high level of versatility.

Due to the variety of IOWN Global Forum use case requirements, DCI Cluster designers constructing DCI Clusters with today's technology may want to focus on specific use case classes to identify the relevant workloads for their designs and efficiently enable IOWN Global Forum use cases.

3. Review of the DCI Cluster concept

This section reviews the roles of the functional blocks of Data-Centric Infrastructure (DCI) Clusters. First, **Section 3.1** introduces the overall formal structure of DCI Clusters as specified in [**IOWNGF-DCIFA2**] and describes the main functional blocks of DCI Clusters. Next, **Section 3.2** highlights how today's typical data center structure could be matched to DCI Clusters. Finally, **Section 3.3** briefly illustrates one example implementation approach for DCI Clusters.

Moreover, readers interested in an extended discussion of how to construct DCI Cluster RIMs based on the structure of today's data centers are referred to **Appendix B** for further reading.

3.1. Review of DCI Cluster functional blocks

The DCI Functional Architecture Release 2[**IOWNGF-DCIFA2**] introduces the term "DCI Cluster" to refer to a localized set of networking and computing components connected to a wide-area network. Introducing a new term instead of just reusing "data center" avoids impreciseness and preconceptions about scale.

An overview of the main structural elements of DCI Clusters is presented in **Figure 3.1**:

- DCI Physical Nodes form the core of DCI Clusters and contain computing hardware such as CPUs, storage, or accelerators. Each DCI Physical Node comprises one DCI Intra-Node Interconnect to enable communication between devices within one node. In addition, this interconnect may optionally be used to form logical servers called DCI Logical Service Nodes (DCI LSNs) in case the DCI Physical Node is implemented by a hardware pool.
- One DCI Gateway interfaces a DCI Cluster to an Open APN WAN network.
- One DCI Inter-Node Interconnect connects devices inside DCI Physical Nodes to the DCI Gateway or to each other.

The control and management of DCI Clusters is structured as follows:

- Every DCI Cluster is controlled by exactly one DCI Cluster Controller (DCI CC). This controller performs monitoring and configuration tasks to let the DCI Cluster provide computing and network capabilities.
- One or more DCI CCs are managed by exactly one DCI Infrastructure Orchestrator (DCI IO). DCI IOs orchestrate hardware resource requests; however, further DCI IO details are out of the scope of this document.
- The DCI Service Exposure Function (DCI SEF) receives requests from users or their applications or orchestrators and processes these for handling by the DCI IO.

Taken together, these elements comprise a single DCI System [**IOWNGF-DCIPCP**].

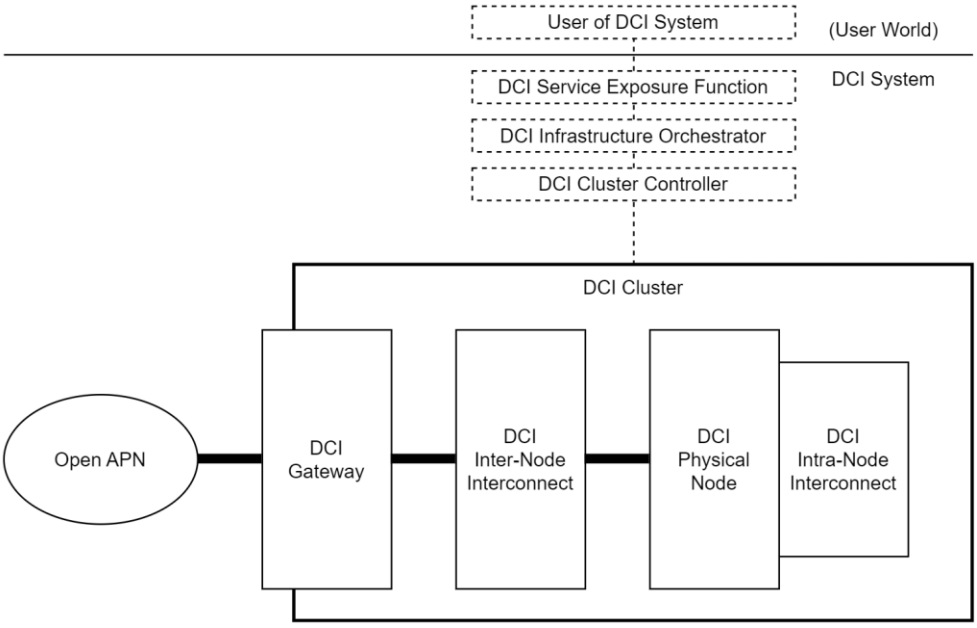

Figure 3.1: Logical structure of DCI Clusters.

Finally, the DCI Cluster model defined in the DCI Functional Architecture does not prescribe how DCI Clusters must be implemented or how they must be structured internally. The DCI FA only prescribes how they must behave when observed from outside the DCI Cluster.

Readers interested in the exact definitions are referred to [**IOWNGF-DCIFA2**](Sec. 1, Sec. 7).

3.2. Matching today's data center structure to DCI Cluster functional blocks

For illustration, this section matches the structure of today's data centers to the structure of the DCI Cluster functional blocks. Moderately simplifying, data centers consist of servers to run user applications, gateway switches to connect servers to WANs outside the data center, and a fabric of local area network switches to connect servers and gateway switches to each other.

Figure 3.2 highlights how the parts of today's typical data centers and DCI Clusters would correspond to each other if these were directly matched. In particular,

- DCI Gateways could be implemented by QoS-capable gateway switches,
- DCI Inter-node Interconnects could be implemented by fabrics of QoS-capable LAN switches,
- DCI Physical Nodes could be implemented by commercial off-the-shelf (COTS) servers or composable disaggregated infrastructure (CDI) resource pools, and
- DCI Intra-Node Interconnects could be implemented either by classic server system buses or by the bus fabrics of resource pools.

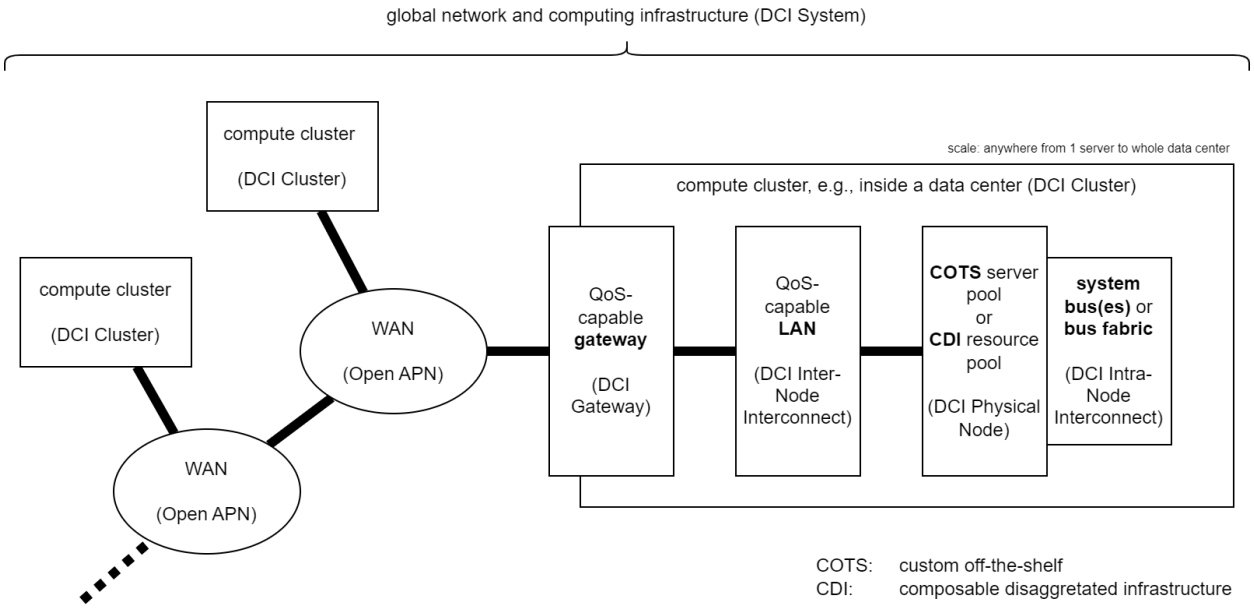

Figure 3.2: Typical data center structure matched to DCI Cluster model.

The following **Section 3.3** further illustrates the outline shown in **Figure 3.2** with a concrete example of a DCI Cluster built from today's data center components and shows how a DCI Cluster is embedded into its DCI System environment.

3.3. Illustration of key points of main functional blocks for implementation

This section highlights the key points for implementing a DCI Cluster by way of example. First, **Section 3.3.1** provides a broad structural overview of an example DCI Cluster implementation and explains how DCI Clusters are embedded within DCI Systems. After that, **Section 3.3.2, Section 3.3.3, Section 3.3.4, and Section 3.3.5** highlight key points for implementing DCI Clusters. Furthermore, interested readers can find more examples to start DCI Cluster designs from in **Appendix C**.

3.3.1. Overview of an example DCI Cluster implementation

Figure 3.3 illustrates an example DCI Cluster and its embedding into a DCI System. Within the figure, the DCI Cluster is placed to the right: The central piece of equipment in this example is a network switch that can provide quality-of-service (QoS) guarantees for traffic among its ports. Classic servers are connected to the switch with NICs that support high QoS. The switch then connects the servers among themselves and also toward a WAN, which is shown on the left-hand side of the figure. These components comprise the data plane of this DCI Cluster.

The control plane part of the DCI System is illustrated above the data plane components. The control plane consists of various controller applications, servers to execute these, and in this example a separate set of network devices.

The management plane of the DCI System is depicted topmost, comprising applications that interact with DCI System users and perform orchestration tasks. Both the management plane and the control plane including DCI Cluster Controllers are realized outside of DCI Clusters. Therefore, their detailed realization is beyond the scope of this document.

The following sections discuss implementation choices for the key blocks comprising DCI Clusters.

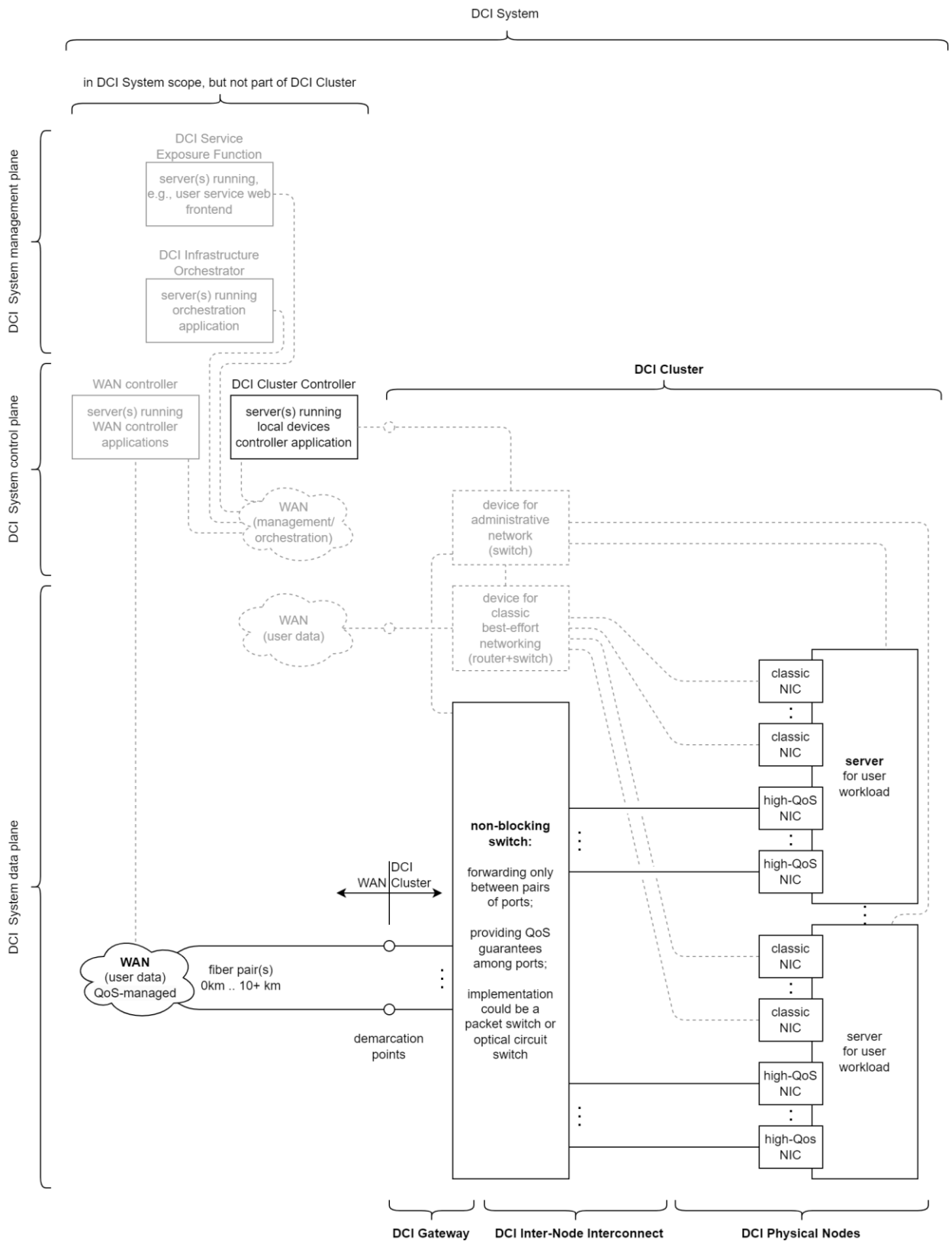

Figure 3.3: Overview of an example DCI Cluster implementation.

3.3.2. Key point: DCI Physical Nodes

DCI Physical Nodes contain the hardware of DCI Clusters that provides computing capabilities for user workloads. In the example above, DCI Physical Nodes are realized by classic servers equipped with NICs for management and best-effort internet access as well as with NICs for high-QoS communication, such as SmartNICs with RDMA offloading support. More generally, DCI Physical Nodes can also be implemented by hardware pools that can dynamically create logical servers from their inventory.

3.3.3. Key point: DCI Intra-Node Interconnect

The DCI Intra-Node Interconnect enables the components of logical server to communicate with each other. In the example with classic servers above, this interconnect is implemented by the individual system buses of each server. In case servers are created as logical servers from hardware pools, the DCI Intra-Node Interconnect might additionally be implemented by host adapter cards and a fabric of switches that forward system bus messages across servers and/or enclosures containing expansion cards.

3.3.4. Key point: DCI Gateway

The DCI Gateway bridges between the inside and outside of a DCI Cluster. In the example above, the DCI Gateway is implemented by the same switch that also implements the DCI Inter-Node Interconnect. Ports connecting to a WAN might have long-range transceivers installed or could be further connected to a dedicated media converter. More generally, DCI Gateways can also be implemented by separate QoS-capable gateway switches or routers.

3.3.5. Key point: DCI Inter-Node Interconnect

The DCI Inter-Node Interconnect connects DCI Gateway and DCI Physical nodes. In the example above, the DCI Inter-Node Interconnect is implemented using a single switch capable of QoS management. More generally, DCI Inter-Node Interconnects could also be realized using electronic or optical switch fabrics that provide non-blocking connectivity between servers and gateways, forward at wire rate, and support virtually lossless connectivity.

In the following, **Section 4** presents detailed reference implementation models (RIMs) for DCI Clusters.

4. DCI Cluster RIMs with today's hardware

This section presents four different Data-Centric Infrastructure (DCI) Cluster RIMs as outlined in **Table 4.1**.

Table 4.1: Overview of 4 DCI Cluster RIMs.

SECTION	DCI CLUSTER RIM DESCRIPTION
Section 4.1	DCI Cluster RIM based on compute racks with composable disaggregated infrastructure: The focus of this DCR is on hardware resource pools from which logical servers can be composed on user request. The RIM describes how such hardware resource pools are structured and outlines how logical servers from these pools can be connected to Open APN networks via a non-blocking network fabric. Inside the hardware pools, disaggregated components such as GPUs and storage are connected via a PCIe fabric switch that has a latency of 345 nanoseconds. Up to 16 logical servers can be composed from a single hardware pool, with no limit on the number of pools itself.
Section 4.2	DCI Cluster RIM based on a scalable, QoS-assured, lossless network for supporting GPU-to-GPU communication: This DCR is primarily tailored to AI training applications. The RIM describes how to overcome the shortcomings of current DCB and RDMA technologies for AI training with scheduled Ethernet fabrics. Such fabrics achieve perfect load balancing over all spine switches in a leaf-spine topology and prevent incast congestion via end-to-end scheduling, while still only requiring standard RoCEv2 compatible NICs inside servers. Finally, the RIM describes how such a fabric can be integrated with other components such as servers and gateways to form a complete DCI Cluster.
Section 4.3	DCI Cluster RIM based on elastic edge computing infrastructure with small-scale rack components: The main subject of this DCR is the structure of small-scale DCI Clusters, enabling DCI Cluster operators to start small by setting up a minimal system and adding resources when demand grows. The section lists up the detailed component classes for a single-rack DCI Cluster consisting of 6 servers, switches, and a connection to a QoS-managed WAN network. End-to-end QoS management through NICs, switches, and the WAN is performed using RoCEv2. The driver application for this RIM is a virtual machine host platform that supports VM live migration via RDMA over WAN.
Section 4.4	DCI Cluster RIM based on huge and scalable disaggregated infrastructure with PCIe bus extension: This RIM employs PCIe bus extension technology that encapsulates PCIe bus traffic in plain Ethernet packets. The resulting disaggregated hardware pools may scale to 10,000 devices or more. This scalability is fostered by utilizing regular Ethernet switches. The same switch fabric can be used for both encapsulated bus traffic as well as regular TCP/IP or RDMA traffic among servers or toward a WAN if switches are equipped with QoS management. Transparent to application software, the only special hardware requirements are bus extension PCIe cards in servers and hardware pool accelerator casings.

4.1. DCI Cluster RIM based on compute racks with composable disaggregated infrastructure

This section outlines a DCR constructed with a Composable Disaggregated Infrastructure (CDI) system. **Section 4.1.1** provides an overview of CDI. **Section 4.1.2** outlines possible use cases for this Data-Centric Infrastructure (DCI) Cluster with CDI. **Section 4.1.3** illustrates the structure of this RIM. Finally, **Section 4.1.4** complements the basic structure with an example of concrete devices to implement the structural RIM blocks and an overview of the resulting capabilities of such a DCI Cluster with CDI.

4.1.1. Composable Disaggregated Infrastructure (CDI) overview

In the era of AI, exemplified by generative AI, enormous computational power and resources are required. At the same time, there is a demand for achieving a sustainable society and reducing power consumption. Therefore, it is necessary to achieve these conflicting requirements simultaneously.

CDI is a new server architecture that allows for the dynamic reconfiguration of server components to create servers with the necessary specifications when needed. Traditional servers are disaggregated into components (CPU, Memory, GPU, SSD, etc.) connected via interconnect-switches to form resource pools. Custom-configured servers are then built from these resource pools using software definition. By dynamically adjusting the specifications in response to workload fluctuations, it achieves both high performance and low power consumption.

In changing environments, a system that enables flexible service provisioning by pooling the device resources is needed to realize diverse services and automatically build the infrastructure according to the workload and device resource status. The key features of CDI are as follows:

- Device Resource “Pool”
    - Various device resources (GPU, SSD, NIC, etc.) are logically decomposed using disaggregation technology and managed as a shared resource pool for the entire system, enabling them to be freely combined.
- “Dynamic Configuration” by user request
    - CDI can select device resources from the resource pool that match the user requirements and remotely create the DCI logical service node (DCI LSN) defined in [**IOWNGF-DCIFA2**]. CDI can also release DCI LSNs after use and return them to the resource pool in a reusable state.
- “Dynamic Re-configuration” by status check
    - CDI can monitor workload changes and resource status and dynamically augment/shrink infrastructure according to user policy.

CDI can respond to resource requirements timely and maximize resource utilization by sharing hardware resources across systems.

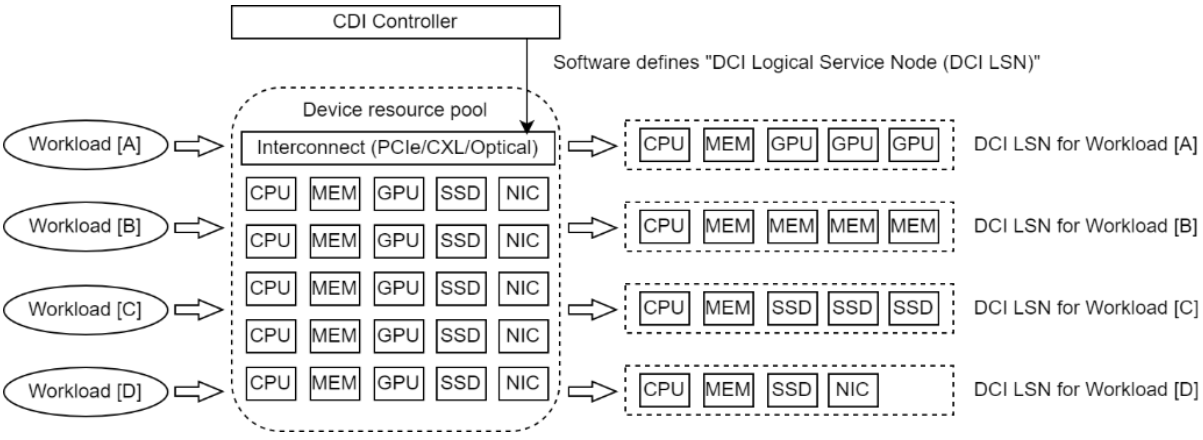

Figure 4.1: Overview of the Composable Disaggregated Infrastructure (CDI).

4.1.2. Use cases

CDI can provide flexible compute infrastructure with high performance, low power consumption, and cost and operational efficiency. Traditional systems, which prepared environments for each workload, required significant labor time and installation costs to enhance the system in response to increased workloads. Additionally, in system

operations, there were instances where the usage rate of hardware resources was imbalanced depending on the time of day or season, leading to inefficiencies. By flexibly combining limited resources, it is possible to optimize initial installation costs and power consumption, thereby reducing TCO (Total Cost of Ownership).

- **Private Cloud**
  - Since various workloads are supported on the public cloud, it is possible to achieve the same convenience as the public cloud while ensuring security for the end user.
- **Edge Cloud**
  - The combination of 5G networks and edge cloud enables continuous high-speed, low-latency services that can keep up with rapid changes, even in dynamically shifting environments such as sports events.
- **Big data / AI**
  - During AI training, multiple GPUs can be concentrated on a single compute server, while during inference, one GPU can be allocated to each of multiple compute servers. This component-level configurability allows for efficient operation even in systems integrated for various use cases.

CDI can support the general use cases listed above as flexible computing infrastructure. In addition, high-speed QoS-guaranteed interconnect-switches act as a “scale-up” network that can provide short-range group-to-group communication in an AI/ML cluster. This DCR also mentions that a QoS-managed network as a “scale-out” network can provide scalability for short-range group-to-group communication. In addition, this DCR explains a QoS-managed gateway and WAN based on APN, which can also provide far point-to-point communication between data centers.

IOWN Global Forum use cases for which such an approach may be applicable in **Table 2.1** include:

- CPS Area Management security
- Large ML model training
- Green Computing with Remote GPU over APN
- IDH: compute-storage disaggregation
- IDH: off-site replication and compute-storage disaggregation

4.1.3. RIM structure

This section outlines how a DCI Cluster with CDI can be constructed.

Figure 4.2 shows a DCI Cluster with CDI in the context of a complete DCI Cluster. The right-hand side corresponds to a CDI whose compute servers and hardware resources such as GPUs/SSDs/SmartNICs are interconnected by PCIe fabric switches. These resources are managed and controlled by CDI Manager and CDI Controller, which are functions in DCI Cluster Controller defined in [**IOWNGF-DCIFA2**], to dynamically reconfigure CDI LSNs according to the user application workload. For example, red-colored resources compose DCI LSN for logical storage servers and green and blue-colored resources compose DCI LSNs for logical GPU servers. The left-hand side of the figure corresponds to the QoS-managed network for DCI inter-node interconnect, QoS-managed gateway, and WAN for Data Center Interconnect to connect DCI clusters among geographically dispersed data centers. Vertically, the figure is divided into three layers corresponding to the data, control, and management planes of the overall DCI Cluster including CDI Manager and Controller. For further details regarding DCI Cluster and the DCI Functional Architecture, the reader is referred to [**IOWNGF-DCIFA2**].

Section 4.1.3.1, **Section 4.1.3.2**, and **Section 4.1.3.3** further highlight the data, control, and management planes, respectively.

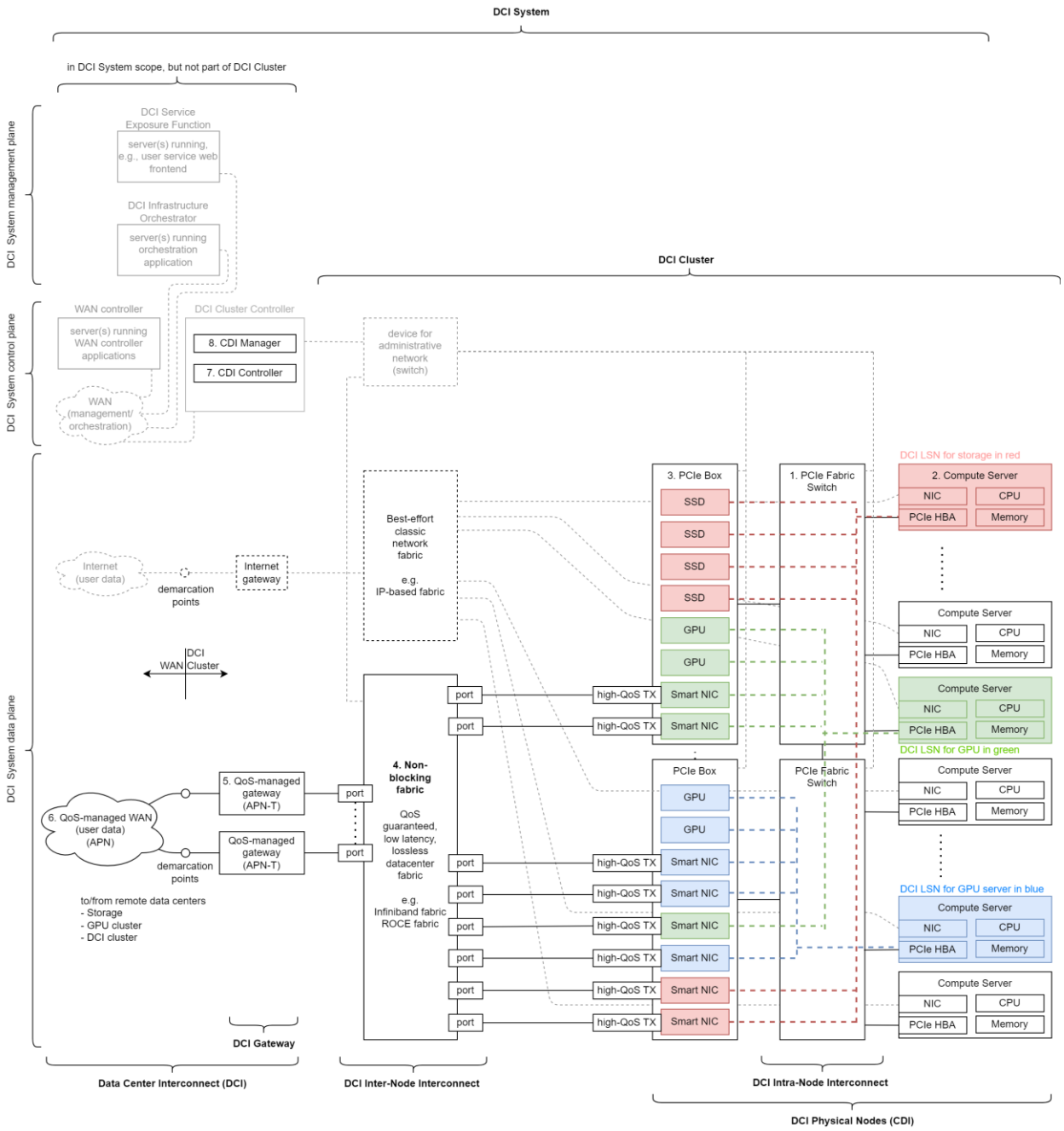

Figure 4.2: DCI Cluster with CDI.

4.1.3.1. Data plane: networking equipment and network interfaces

This section highlights key aspects of CDI components related to the data plane of the DCI Cluster.

4.1.3.1.1. PCIe fabric switch: DCI Intra-node Interconnect

Hardware resources such as compute server with CPU/memory, GPU, SSD, NIC, etc., are interconnected by a high-performance, low-latency PCIe fabric switch. The compute server has a host bus adapter (HBA) connected to a PCIe fabric switch. Hardware resources are deployed in PCIe boxes that are chassis for deploying PCIe card-type devices such as GPU, SSD, and NIC; then it is connected to a PCIe fabric switch.

4.1.3.1.2. Compute servers (and NICs)

At the time of this writing, CDI's Compute servers are not so disaggregated; CPUs and memory devices are located on the host board in a computing server. In general, the computing servers have on board NICs or extended PCIe card slots, so for NICs for which high QoS is not required, these on-board NICs and/or extended NICs in PCIe card slots are used for cost efficiency. These NICs are connected to the best-effort network for external network traffic, such as the Internet or the QoS-managed network if required.

4.1.3.1.3. PCIe Box: CDI I/O resource pool

At the time of writing, CDI's composability is mature at the I/O component level, such as GPU, SSD, and (Smart)NIC. These I/O components are deployed in PCIe boxes and connected to CPUs/memories in compute servers via a PCIe fabric switch. PCIe Box may support PCIe fabric switch functionality and be directly connected to CDI's compute servers to build a small DCI System.

4.1.3.1.4. Non-blocking fabric: DCI Inter-node Interconnect

This non-blocking fabric is dedicated to DCI inter-node interconnect, usually called a "backend" network in the data center. This network fabric is used to connect storage systems as Storage Area Networks (SAN) and provides a QoS-managed, high capacity, low-latency, and lossless network for storage access. Recent AI workloads require higher capacity, low latency, lossless, and massive scalability for the backend network fabric.

To achieve these requirements, the DCI inter-node interconnect should be "non-blocking" so that it can always successfully transfer all data from the input ports to the output ports as long as the output ports have sufficient free resources to forward the data. Even though the non-blocking fabric has the capacity to forward all input bandwidth, it cannot guarantee the bandwidth if the output ports do not have enough free resources. To guarantee the bandwidth, an external control mechanism that fully manages resource allocation and mediation is required. For example, an application requests its bandwidth via API through a DCI Infrastructure Orchestrator. Then, the related resources are reserved in the non-blocking fabric by a DCI Cluster Controller.

Many GPU clusters utilize RDMA (Remote Direct Memory Access) to further enhance performance, so this backend network fabric should support RDMA traffic. At the time of writing, InfiniBand or RoCEv2 (RDMA over Converged Ethernet version 2) is the most suitable networking technology for this.

No specific implementation for the non-blocking fabric is proposed in this section. However, readers can find one in **Section 4.2**, "A scalable, QoS-assured, lossless network for supporting GPU-to-GPU communication."

4.1.3.1.5. QoS-managed Gateway: Data Center Interconnect

This QoS-managed Gateway provides a Data Center Interconnect for distributed DCI clusters. The gateway interconnect QoS-managed network as backend network fabric described in above section, so it should support very high capacity, low-latency and lossless capabilities. To implement this Data Center Interconnect, we can adopt the same structure proposed in **Section 4.3**, "Elastic edge computing infrastructure with small-scale rack components." The QoS-managed Gateway with APN-T for the WAN interfaces transfers RoCEv2 frames over Open APN with long-distance.

4.1.3.1.6. QoS-managed WAN

To provide very high capacity, low-latency, and lossless links between data centers, we propose Open APN as a QoS-managed WAN.

4.1.3.1.7. Best-effort network

This best-effort network fabric is dedicated to inter-virtual machine/container application interconnects and for connecting to the Internet, usually called a "frontend" network in the data center. This network fabric is implemented separately from the QoS-managed "backend" network fabric. Although we call this network fabric "best-effort," some

current mature implementations have some kind of QoS capability based on DiffServ (Differentiated Service) architecture. The main difference between this frontend and backend network fabric is being lossy or lossless, and this best-effort network fabric is a lossy network.

This best-effort network is a well-established and understood technology. Therefore, it is not highlighted in this section.

4.1.3.1.8. Internet Gateway

To connect with the Internet, an Internet Gateway should be implemented in front of the DCI Cluster. The Internet gateway should maintain layer 3 routing to reach resources all over the world.

This Internet Gateway is a well-established and understood technology. Therefore, it is not highlighted in this section.

4.1.3.2. CDI Controller: Control plane

The CDI controller controls PCIe switch fabrics and PCIe boxes to configure DCI LSNs. It is connected to these devices via PCIe cables.

4.1.3.3. CDI Manager: Management plane

The CDI Manager manages hardware resources in the CDI system for performance optimization. The CDI Manager is connected to the CDI Controller by management ethernet switches, and the CDI Manager interacts with the CDI Controller to operate the CDI.

4.1.4. Example implementation choices

This section outlines one possible implementation of a DCI Cluster with a CDI system that is structured as illustrated in **Figure 4.2**. First, concrete component classes for each part of the DCI Cluster are selected in **Section 4.1.4.1**. After that, the projected capabilities of a DCI Cluster implemented with these components are listed in **Section 4.1.4.2**.

4.1.4.1. Selected component classes

Table 4.2 lists one possible set of example implementation choices for a concrete DCR, as illustrated in **Figure 4.2**.

Table 4.2: Example implementation choice for DCR with compute racks with composable disaggregated infrastructure.

#	DCR DIAGRAM BLOCK	EXAMPLE IMPLEMENTATION CHOICE
1	PCIe Fabric Switch	PCIe Gen 5.0, 16 ports switch
2	Compute server (and NICs)	x86 COTS servers with memory hardware according to user application needs, and PCIe Gen5.0 HBA to connect to PCIe Switch, and NIC to connect to best effort network fabric
3	PCIe box (and I/O cards)	PCIe Gen5.0, 8 slot PCIe Box, GPU/SmartNIC/SSD cards
4	Non-blocking fabric	Scheduled Ethernet Fabric composed of Network Cloud packet Processor (NCP) and the Network Cloud Fabric (NCF), refer to **Section 4.2**
5	QoS-managed Gateway	APN-T
6	QoS-managed WAN	Open APN

7	CDI controller	Hardware appliance for CDI controller; controls PCIe fabric switch and PCIe box connected with PCIe cable
6	CDI manager	x86 COTS servers running the CDI management software applications

4.1.4.2. Resulting capabilities summary

Based on the example implementation choices outlined in **Section 4.1.4.1**, **Table 4.2** lists the projected capabilities for a DCI Cluster with a CDI system that is structured as illustrated in **Figure 4.1**.

Table 4.3: Example capabilities for a DCR with compute racks with a composable disaggregated infrastructure.

#	DCR DIAGRAM BLOCK	PROPERTY	PROJECTED/ESTIMATED VALUE	DETERMINED BY
1	PCIe Fabric Switch	bandwidth per port	128 GB/s (full duplex)	PCIe fabric switch port
		switching capacity	2,048GB/s (full duplex)	PCIe fabric switch
		number of ports	16 ports	PCIe fabric switch
		Max. number of PCIe fabric switches per CDI	2 PCIe fabric switches	
		Latency added inside PCIe fabric switch	345 ns	total latency with PCIe fabric switch, HBA and PCIe box; excludes CDFP connector cable
		Packet loss	none in CDI	PCIe fabric switch
		Packet ordering	preserved in CDI	PCIe fabric switch
		Non-blocking guarantees	full duplex, non-blocking PCIe fabric switch in CDI	PCIe fabric switch
2	Compute server	bandwidth per HBA	128 GB/s (full duplex)	HBA
		bandwidth per NIC	1, 10, 25, 100 Gbps	NIC for best effort network (Internet), depends on user application workloads
		Max. number of Compute server per CDI	16 compute servers	
3	PCIe box	number of slot for CDI I/O card	8 slots	PCIe box
		Max. number of PCIe boxes per CDI	3 PCIe boxes	
		CDI I/O card: Bandwidth per connection to QoS-managed network	200 Gbps	SmartNIC
4	Non-blocking fabric	QoS-guaranteed, low-latency, lossless data center fabric	refer to **Section 4.2.3.2**	

5	QoS-managed Gateway	QoS-managed, Data center Interconnect	100/400/800 Gbps	APN-T
6	QoS-managed WAN	Open APN		
7	CDI Controller	number of controllers in CDI	1	
8	CDI Manager	number of managers in CDI	1	

4.2. DCI Cluster RIM based on a scalable, QoS-assured, lossless network for supporting GPU-to-GPU communication

This section outlines a DCR constructed around a two-tier scheduled fabric that connects multiple servers to provide scalable, QoS-assured, and lossless communication. First, the design objectives and assumed use cases of this RIM is outlined in **Section 4.2.1**. Then, the structure and design overview of this RIM are illustrated in **Section 4.2.2**. Finally, the basic structure is complemented by an example of concrete devices to implement the structural RIM blocks and an overview of the resulting capabilities of such a Data-Centric Infrastructure (DCI) Cluster in **Section 4.2.3**.

4.2.1. Design Objectives and Use Cases

This DCR focuses on a two-tier scheduled fabric, which corresponds to the DCI Inter-Node Interconnect, providing QoS-assured, lossless, and non-blocking connectivity between accelerators across a large number of DCI Physical Nodes. The primary goal of this DCR is to support massive GPU-to-GPU communication in a large-scale machine learning (ML) training cluster. In such clusters, multiple accelerators, e.g., GPUs, synchronize significant volumes of data, such as model parameters and gradients, through collective communication with RDMA. Communication latency and packet loss in data synchronization slow down the entire training process, making short-range group-to-group QoS-managed communication crucial to achieving optimal training performance (for further details, see **Section A.2.2**).

To meet these demands, this DCR offers QoS-assured, lossless, and non-blocking connectivity that supports short-range group-to-group QoS-managed communication. The two key capabilities that enable efficient, high-performance group-to-group communication within this DCR are:

1. RDMA-capable network: This fabric supports congestion-free, lossless Ethernet communication with guaranteed packet order at the egress port, which helps maintain high RDMA performance.
2. Non-blocking connectivity with perfect load balancing and flow control: In large-scale ML clusters, numerous accelerators generate line-rate traffic. This DCR ensures perfect load balancing within the fabric, while flow control mechanisms effectively prevent congestion. As a result, non-blocking connectivity is maintained across input and output ports, allowing all incoming traffic to be efficiently transmitted to the output ports.

While this DCR is primarily designed for short-range group-to-group communication in large-scale ML clusters, its capabilities also benefit other use cases that require QoS-managed, lossless communication with accelerators. This DCR can be extended by integrating with additional components and can also support far point-to-point communication use cases by connecting to external WAN gateways. From this point, current potential use cases of this DCR are:

- Large ML Model training
- Green Computing with Remote GPU over APN
- Sovereign cloud

Moreover, this DCR can be extended to support additional use cases requiring end-to-end bandwidth or latency guarantees between clients and servers by integration with other DCI components. Since this DCR focuses on providing

lossless, non-blocking connectivity inside the fabric, it is essential to manage resources both within and outside the fabric when it is applied to use cases with end-to-end latency and bandwidth guarantees. For instance, it is crucial to ensure that incoming traffic to the fabric does not exceed the bandwidth capacity of its input and output ports, and that QoS should be controlled for WAN communication outside the fabric. To achieve this, external mechanisms, such as a DCI Infrastructure Orchestrator and DCI Cluster Controller, are required to handle resource allocation and mediation in conjunction with this DCR.

In the following, **Section 4.2.2** outlines the challenges of the internode interconnect in the ML cluster, the design overview of this DCR, and the implementation strategy of an ML cluster with a two-tier scheduled fabric, ensuring the resulting DCR is applicable to the above use cases.

4.2.2. RIM Structure

This section outlines how to construct a scalable, QoS-assured, and lossless network with a two-tier scheduled fabric for DCI Inter-Node Interconnect.

Figure 4.3 shows a DCI Cluster with a two-tier scheduled fabric Inter-Node Interconnect in the context of a complete DCI System: the right-hand side corresponds to a DCI Cluster with gateway, network, and compute server resources as introduced in **Section 3.1**, and in turn, the left-hand side of the figure shows key blocks that interact directly or indirectly with DCI Clusters, such as the DCI Cluster Controller. Vertically, the figure is divided into three layers corresponding to the data, control, and management planes of the overall DCI System. For further details regarding DCI Systems and the DCI Functional Architecture, the reader is referred to [**IOWNGF-DCIFA2**].

Section 4.2.2.1 and **Section 4.2.2.2** further highlight key functionalities to construct a scalable, QoS-assured, and lossless network.

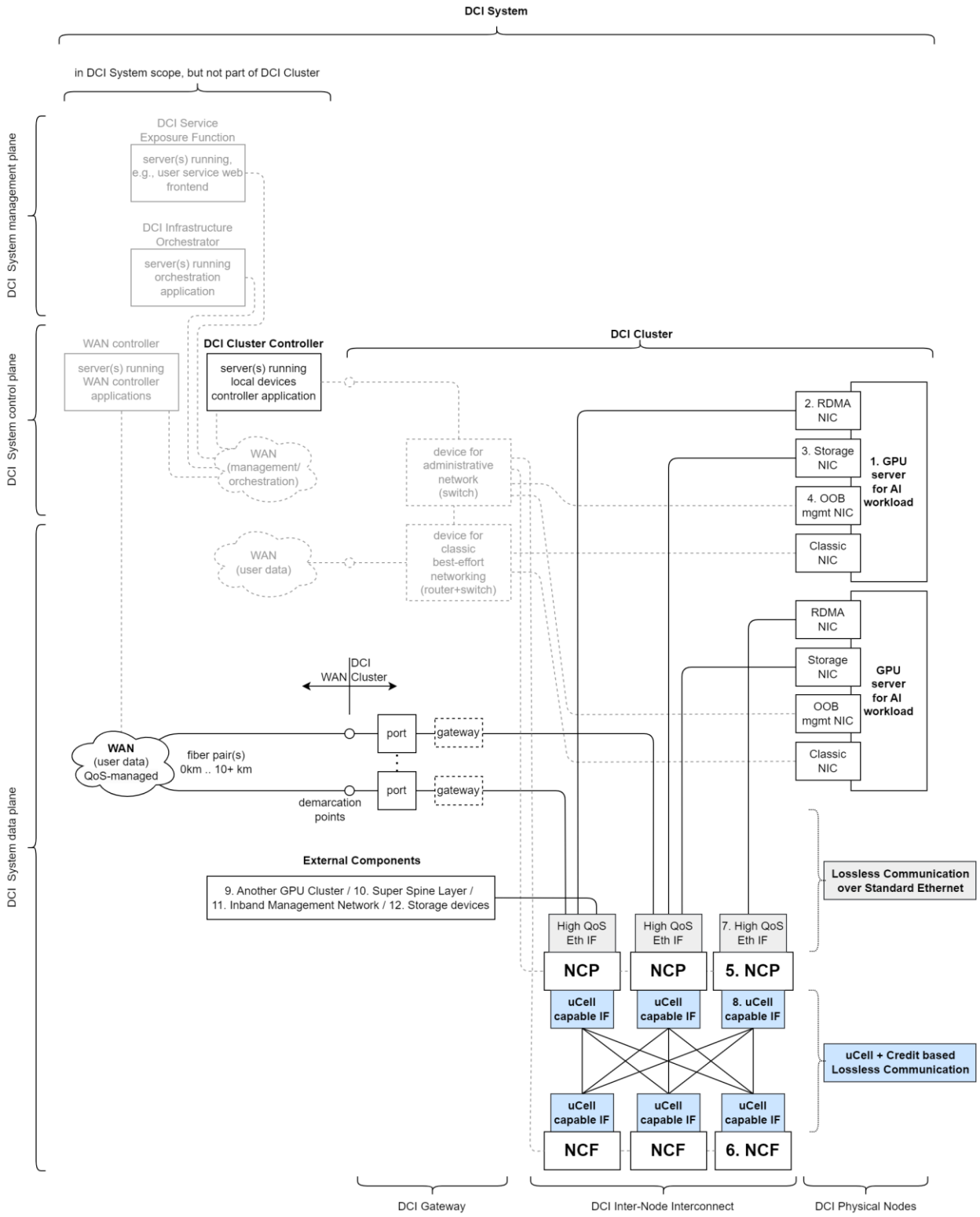

Figure 4.3: DCI Cluster with a two-tier scheduled fabric for Inter-Node Interconnect.

Challenges of existing approaches

Currently, Ethernet Data Center Bridging (DCB) technology, consisting of Explicit Congestion Notification (ECN) and Priority Flow Control (PFC), is commonly used to support lossless short-distance group-to-group QoS managed communication. Furthermore, for scale-out purposes, using inter-node interconnects using Ethernet Clos topology consisting of two or three layers is a common approach. ML clusters can be constructed with DCB Ethernet switches using open Ethernet technology. However, this approach faces several challenges considering the workload of ML clusters:

- Unbalanced link load: ML cluster workloads generate long-lived flows among the senders and receivers. As a result, the entropy of ECMP flow hashes is low, and hash collisions bias the load to certain links in the network. This causes insufficient network bandwidth usage.
- Incast congestion: In ML clusters, accelerators share large amounts of data simultaneously. As a result, their traffic patterns are highly synchronized, leading to incast congestion where flows concentrate on the same destination port. Such congestion can trigger communication halts due to PAUSE frames from PFC in DCB switches, prolonging the completion time of the flows.
- Long Failover Time: As the scale of ML clusters increases, failures of network devices such as optical modules become inevitable. According to reports from Broadcom, the annual failure rate is 2%, with about 15 modules failing each month. However, when a failure occurs in the traditional datacenter network, it is necessary to wait for routing convergence, which can take 50 milliseconds to several seconds for recovery, depending on the routing protocol and topology size. During this recovery period, all packets forwarded to the failed module are lost. Therefore, RDMA senders experience significant throughput degradation due to its poor loss recovery algorithm, Go-back-N. If the recovery time is short, the end host can resume communication after a substantial drop in throughput. However, if the recovery time is long and reaches a maximum number of retries, the GPU-to-GPU communication will be terminated. In the worst case, the whole training process will need to be restarted. Given the high frequency of module failure and the risk of restarting the entire training process, rapid failure recovery is an essential challenge for ML clusters.

The above three challenges lead to prolonged large ML model training, resulting in poor training performance.

Design overview

This DCR is designed with a two-tier scheduled fabric consisting of two types of network devices: the Network Cloud packet Processor (NCP) and the Network Cloud Fabric (NCF). The NCP is located in the first tier of this fabric, and the NCF is located in the second tier. The NCP performs advanced functionalities, while the NCF handles simple forwarding functionalities.

To support scalability, QoS assurance, and lossless communication, this DCR provides four key functionalities: perfect load balancing, end-to-end flow control, congestion management, and rapid failover. Perfect load balancing offers high link utilization and satisfies high scalability requirements. Both end-to-end flow control and congestion management can control the incast congestion at the second and first tiers, respectively, ensuring lossless communication and QoS assurance. Finally, rapid failover provides microsecond-level recovery, meeting the QoS assurance requirements.

- Perfect load balancing: In the second tier of this DCR, between the NCP and NCF, packets are divided into multiple fixed-length microcells (uCells). When the NCP receives a packet from the server, it splits the packet into multiple uCells and forwards them randomly per uCell basis. The NCF then forwards the uCells to the destination NCP, which reassembles the uCells into a packet. In traditional ECMP, the path is randomly chosen for each flow, making load balancing difficult for ML workloads with low flow entropy. By performing load balancing at the uCell level, perfect load balancing can be achieved, satisfying high link utilization and high scalability requirements.
- End-to-end flow control: This DCR uses credits and virtual output queues (VoQs) to achieve end-to-end flow control. “End-to-end” in this context refers specifically to this fabric’s ingress and egress ports, rather than the application- or server-level endpoints. This distinction ensures that flow control and congestion management are handled entirely within the bounds of the two-tier fabric, optimizing internal network performance for RDMA

and GPU-to-GPU communication. At the ingress NCP, incoming packets are placed in a VoQ that holds packets directed to that specific destination VoQ. The packets must wait in the VoQs until they are granted access to the fabric by a special message called Credit. The scheduler of the egress NCP is responsible for granting access and controls the rate of incoming data to match its bandwidth. When the destination NCP becomes congested, no credits are granted from its scheduler, and the packets cannot be forwarded to it. This feature helps avoid incast congestion.

- Congestion management: This DCR manages congestion at two tiers. As described above, at the second tier, between the NCP and NCF, congestion can be controlled by the credits. At the first tier, DCB technology manages congestion between the server and NCP. When the ingress NCP becomes congested, it forwards a PFC PAUSE frame to the server to control the data rate. In the existing DCB approach, when congestion occurs, multiple switches at both the leaf and spine layers forward PFC PAUSE frames, causing PFC deadlock. In contrast, this approach limits PFC usage between an NCP and a server, thereby avoiding deadlock.
- Rapid failover: This DCR supports hardware-based failover. The NCP and NCF monitor failures at the SerDes level, allowing for rapid detection of connectivity issues. Compared to existing approaches, this achieves faster failover, on the order of microseconds.

4.2.3. Example implementation choices

This section outlines one possible implementation of a DCI Cluster that is structured as illustrated in **Figure 4.3**. First, concrete component classes for each part of the DCI Cluster are selected in Section **4.2.3.1**. After that, the projected capabilities of a DCI Cluster implemented with these components are listed in Section **4.2.3.2**.

4.2.3.1. Selected component classes

Table 4.4 lists one possible set of example implementation choices for a concrete DCR, as illustrated in **Figure 4.3**.

Table 4.4: Example implementation choice for a DCR with a scalable, QoS-assured, and lossless Inter-Node Interconnect.

#	DCR DIAGRAM BLOCK	EXAMPLE IMPLEMENTATION CHOICE
1	GPU server for AI workload	Classic GPU server running the compute intensive AI workloads, for either training or inference
2	RDMA NIC	100G, 200G, or 400G Ethernet NIC with hardware support for RDMA and RoCEv2
3	Storage NIC	100G, 200G, or 400G Ethernet NIC with hardware support for RDMA and RoCEv2
4	OOB mgmt NIC	10G Ethernet interface for out of band management of the compute, storage and network nodes
5	NCP	Network Cloud packet Processor - a white box switch running the Broadcom Jericho3 chip with 38x 800G interfaces. 18 interfaces are used as High QoS Ethernet interface for Ethernet connectivity towards the GPU servers and additional 20 interfaces are used as uCell capable interface for fabric connectivity towards the NCF spine. NCP performs advanced network processing, uCell processing (fragmentation, load balancing, and reassembly), and credit-based end-to-end VoQ scheduling.
6	NCF	Network Cloud Fabric - a white box switch running the Broadcom Ramon3 chip with 128x 800G uCell capable interfaces. Used as the fabric engine of the cluster. NCF performs simple cell switching to connect NCPs.

7	High QoS Ethernet Interface	800G interfaces with support for QoS classification, marking, PFC, ECN, and breakout capability. Each 800G high QoS interface at NCP supports breakout connections towards two 400G NICs at GPU servers.
8	uCell capable Interface	800G interfaces with uCell forwarding features.
9	Another GPU Cluster (External Components)	Additional AI cluster that is used for scale or distribution of the AI training/inference process.
10	Super Spine Layer (External Components)	Classic Ethernet switches layer used for connectivity of the GPU cluster to the rest of the data center or other data centers
11	Inband Management Network (External Components)	Remote management servers such as TACACS, SNMP, gRPC collectors with in-band connection to the devices.
12	Storage Devices (External Components)	Storage devices containing the data required for the training or inference process and injecting it into the GPU server memory via the AI DDC fabric or a different dedicated network.

The components listed in the table represent the current technology selections, and it is advisable to revise these selections as new technologies emerge. For instance, for RDMA communication over Ethernet, protocols other than RoCEv2, such as Ultra Ethernet Transport, may also be considered in the future.

4.2.3.2. Resulting capabilities summary

Based upon the example implementation choices as outlined in **Section 4.2.3.1**, **Table 4.5** lists the projected capabilities for a DCI Cluster that is structured as illustrated in **Figure 4.3**.

Table 4.5: Example capabilities for a DCR with a scalable, QoS-assured, and lossless Inter-Node Interconnect.

PROPERTY	PROJECTED/ESTIMATED VALUE	DETERMINED BY
Bandwidth per high QoS interface	800 Gbps	transceivers
Bandwidth per fabric interface (NCP and NCF)	800 Gbps	transceivers
Breakout connection capability at NCP	2 connections towards server, 400 Gbps for each	NIC, transceiver
Maximum high-QoS connections per GPU	1 per GPU	

Packet loss	0% Fabric side is natively lossless as it supports **end to end scheduling**. Network interface side (or access side) is lossless as PFC and ECN are being used towards the GPUs.	On fabric side - implementation in Hardware Software On access side - support of PFC and ECN
Packet ordering	NCP is responsible to re-order out of order packets, so no re-ordering is needed on the endpoint side	Leaf (NCP) hardware and software stack
Congestion control	Congestion is fully managed by the end-to-end scheduled fabric	Leaf (NCP) hardware and software stack
Load balancing	100% utilization of fabric links (not using ECMP hashing but rather utilizing uCells to distribute the traffic perfectly)	Leaf (NCP) and Spine (NCF) hardware and software stack
Small-scale cluster (2k)	64 NCPs and 9 NCFs for 2048 GPUs (400G NICs)	The number of server's network interfaces and NCP's ports
Medium-scale cluster (4k)	128 NCPs and 18 NCFs for 4096 GPUs (400G NICs)	The number of server's network interfaces and NCP's ports
Large-scale cluster (8k)	256 NCPs and 36 NCFs for 8192 GPUs (400G NICs)	The number of server's network interfaces and NCP's ports

For reference, we list three example cluster scales, large, medium, and small at the end of **Table 4.5**. The cluster size is determined by the number of the server's interfaces and NCP's ports. Implementers are encouraged to design optimal cluster sizes based on use case requirements, such as the number of GPUs for LLM training.

4.3. DCI Cluster RIM based on elastic edge computing infrastructure with small-scale rack components

This section outlines a DCR constructed around at least one Ethernet switch that connects a QoS-managed WAN and servers. Scalability differs according to use cases, location, or deployment scenario. Additional servers and Ethernet switches can be deployed elastically based on demands. The DCR will form a network fabric connecting computing devices to each other and to a WAN.

Section 4.3.1 describes the objective and purpose of this DCR. **Section 4.3.2** outlines possible use cases for this elastic rack-sized Data-Centric Infrastructure (DCI) Cluster. **Section 4.3.3** illustrates the structure of this RIM. Finally, **Section 4.3.4** complements the basic structure with an example of concrete devices to implement the structural RIM blocks and an overview of the resulting capabilities of such a DCI Cluster.

4.3.1. Objective and Purpose

Edge computing is becoming an integral part of the digital transformation landscape, enabling new applications and business models that require rapid data processing, low latency, and high bandwidth. This DCR, elastic edge computing infrastructure with small-scale rack components, is to provide an efficient, scalable, and general computing environment that brings computational resources closer to data sources and end-users. This infrastructure may be located on-premises or at the edge data center. It aims to reduce latency, improve data processing speed, and enhance the performance of applications that require rapid data processing. Ultimately, it should enable use cases to have scalability, agility, and resiliency.

4.3.2. Use cases

This DCI Cluster implementation approach connects a QoS-managed WAN to servers via gateways, which could provide different network bandwidth levels, such as 10 Gbps and 100 Gbps. This efficiently utilizes QoS-managed WAN to provide high-performance transmission on demand.

Among the IOWN Global Forum DCI System use cases introduced in **Section 2.1**, use cases for which such an approach may be applicable include:

- Financial industry services infrastructure
- Green Computing with Remote GPU Service
- IDH: Database synchronization across data centers
- Sovereign cloud
- CPS Area Management
    - Sensor data aggregation in a camera site and transfer to the ingestion server with RDMA over Open APN

In addition, such an approach may be also applicable to the below use cases:

- Virtual machine live migration with RDMA over open APN
- Continuous backups with RDMA over open APN

In the following, **Section 4.3.3** outlines a structure overview and implementation strategy of an elastic rack-sized DCI Cluster with scalable switches and edge computing resources, so that this DCR can apply to the above use cases.

4.3.3. RIM structure

This section outlines how small-scale rack components can construct a DCI cluster. This DCR is designed to put several components in one small rack. Each rack can host up to 8 nodes, providing computing, storage, and networking capabilities. The RIM adopter can start with one x86 COTS server, one Ethernet switch for data transmission, and one Out-Of-Band (OOB) switch for administration. When a user demands a high availability edge computing environment, at least one extra x86 COTS server should be considered. As user workload increases, it can first scale up. It can then scale out with the same rack configuration if more resources are needed. For data transmission, each server should be equipped with at least one SmartNIC with RDMA support to connect QoS-managed WAN (APN, All-Photonics Network) via APN-T (All-Photonics Transceiver) and Ethernet switch, and one classic NIC to connect classic WAN/network or user devices.

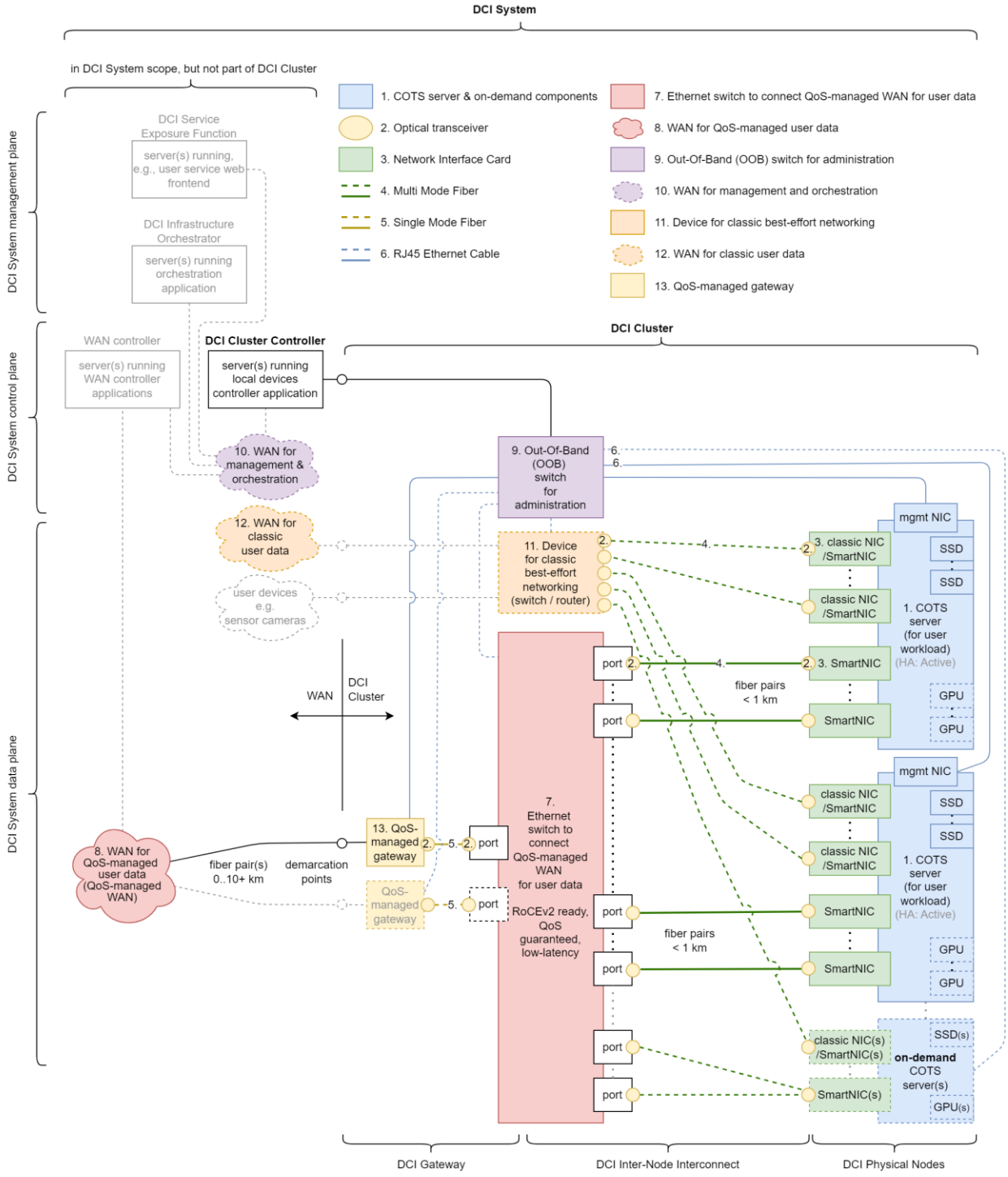

Figure 4.4: DCR of a single Ethernet switch connecting multiple devices to a WAN.

4.3.3.1 Data movement: networking equipment and network interfaces

4.3.3.1.1 QoS-managed network: Ethernet switch/fabric supports RoCEv2 and guarantees QoS

- Protocol support: IPv4/IPv6 and RoCEv2

4.3.3.1.2 Servers and NICs

This DCR is constructed with at least one (without high availability) or two (without high availability) x86 COTS server(s) with the requisite DDR4 RAM and optional accelerators on demand, such as GPUs. All servers should be equipped with SSDs and SmartNICs, such as RDMA NICs and DPUs.

4.3.3.1.3 Gateway(s)

This DCR is constructed with at least one 100G APN-T as a gateway to provide high-QoS networking by connecting to other Open APN nodes. Some use cases may need backup QoS-managed WAN with tolerable lower bandwidth in case of network interruption. The RIM adopter can connect the 10G/25G port of the Ethernet switch to the 10G/25G APN-T as another gateway.

4.3.3.1.4 QoS-managed WAN

A QoS-managed WAN powered by Open APN is expected to provide packet loss-less, packet reordering-less data transfer. Such features are effective especially for use cases that use RDMA.

- The bandwidth requirement differs per use case or configuration.
- Scalability differs per use cases, or per configuration.

4.3.3.1.5 Best-effort network

Servers can connect to a spine-leaf fabric on demand for a classic best-effort network or internet.

4.3.3.2 Control: DCI Cluster Controller

In this DCR, the DCI Cluster Controller (DCI CC) is implemented as a control software configuring the resources of one or more DCI clusters and providing telemetry collection services. This software can run on one of the DCI Physical Nodes or, depending on the use case, it may be deployed separately from the DCI Cluster hardware. It creates and maintains DCI LSNs, which involves allocating computing resources for applications and recording allocation states using stateful databases. It also monitors and reports hardware operational states to the DCI Infrastructure Orchestrator (DCI IO).

Additionally, the DCI Cluster Controller exposes APIs to the DCI Infrastructure Orchestrator, enabling it to manage the lifecycle of DCI LSNs, configure inter-node interconnections, and ensure seamless scaling and resource allocation across multiple clusters. By providing these APIs, the DCI Cluster Controller abstracts the complexities of the underlying physical infrastructure, allowing the orchestrator to focus on high-level service orchestration, resource optimization, and maintaining QoS for distributed applications.

4.3.3.3 Management: DCI Infrastructure Orchestrator

DCI Clusters should be able to interact with higher-layer orchestrators for optimal resource allocation across different clusters based on service requirements. In this DCR, The DCI Infrastructure Orchestrator (DCI IO) is implemented as a control software that manages the overall resources and the state of DCI Clusters. It enables the dynamic provisioning of DCI resources by coordinating the actions of various components in the DCI infrastructure. By sending DCI Cluster Controller API calls, DCI IO can query the inventory and usage status of DCI Clusters. According to the service requirement and the API response, it will create, modify, or release logical servers and their network connections.

For example, upon receiving an LSN creation request, the DCI IO identifies the appropriate DCI Cluster that can accommodate the LSN and selects the corresponding DCI Cluster Controller to execute the operation.

This architecture ensures efficient operation and optimal resource utilization within the DCI System, meeting user demands effectively.

4.3.4. Example implementation choices

This section outlines one possible implementation of a DCI Cluster, as illustrated in **Figure 4.4**. First, concrete component classes for each part of the DCI Cluster are selected in **Section 4.3.3.1**. After that, the corresponding capabilities of a DCI Cluster implemented with these components are listed in **Section 4.3.3.2**.

4.3.4.1 Selected component classes

Table 4.6: Example implementation choices for a DCR with the elastic edge computing infrastructure.

#	DCR DIAGRAM BLOCK	EXAMPLE IMPLEMENTATION CHOICE
1	COTS server & on-demand components	• (Elastically deployed) x86 COTS rack servers with requisite DDR4 RAM and optional accelerators on demand, such as GPUs. • For high-QoS traffic: one 2x100GbE NIC & 2 2x25GbE per server supports PCIe 4.0x8/16 or higher
2	Optical transceiver (for servers and ethernet switch)	• 25GBase SFP28 SR, 100GBase QSFP28 SR for server-switch connection • 100GBase QSFP LR for switch-QoS WAN connection • (Optional) 10GBase SFP+ LR, 25GBase QSFP28 LR for backup QoS WAN connection
3	Network Interface Card (NIC)	• For classic traffic: one 2x10GbE classic NIC • For QoS-managed traffic: 100GbE (or higher) RDMA SmartNIC, DPUs/IPUs, or converged accelerators based on demand, supports PCIe 4.0x8/16 or higher
4	Multimode Fiber (for DCI Inter-Node Interconnect)	• LC-LC Simplex/Duplex Multimode OM3 Fiber Patch Cable for classic best-effort network • LC-LC Simplex/Duplex Multimode OM4 Fiber Patch Cable for QoS guaranteed network • LC-LC Simplex/Duplex Multimode OM5 Fiber Patch Cable for QoS guaranteed network
5	Single-mode Fiber (for WAN connection)	• LC-LC Simplex/Duplex Single-mode OM3 Duplex Fiber Patch Cable • LC-LC Simplex/Duplex Single-mode OM5 Duplex Fiber Patch Cable
6	RJ45 Ethernet Cable	Cat5 / Cat5e / Cat6 Ethernet cable
7	Ethernet switch to connect QoS-managed WAN for user data	• Layer-2 switch with 24 x 10/25-Gbps (SFP28) and 6 x 40/100-Gbps (QSFP) ports. • 2 x 40/100-Gbps ports are connected to the WAN • The other 40/100-Gbps ports and all of the 10/25-Gbps ports are connected to server's network interfaces.
8	WAN for QoS-managed user data	By connecting to APN via gateway (APN-T) and Ethernet switch
9	Out-Of-Band (OOB) switch for administration	24-port Ethernet switch with 1 Gbps network interfaces
10	WAN for management and orchestration	Best-effort classic network/internet
11	Device for classic best-effort networking	24-port Ethernet switch with 10 Gbps network interfaces

12	WAN for classic user data	Best-effort classic network/internet
13	QoS-managed gateway	10/25/100 Gbps APN-T

4.3.4.2 Resulting capabilities summary

Table 4.7: Example capabilities for a DCR with the elastic edge computing infrastructure.

PROPERTY	PROJECTED/ESTIMATED VALUE	DETERMINED BY
Bandwidth per physical interface of SmartNICs	10/25/100 Gbps	transceivers
High-QoS connections per physical network interface	1 connection per physical network interface	NIC, transceiver, Ethernet switch for user data, QoS-managed WAN
Bandwidth per connection on the physical interfaces of SmartNICs	100 Gbps per interface / 1 connection per interface = 100 Gbps per connection (raw)	(derived)
Maximum QoS-managed connections per physical COTS server	1 connection	number of NICs, QoS-managed switches
Maximum number of physical servers connected to QoS-managed WAN	25 x 100GbE server-facing ports on 6 high-QoS data switches / 1 100GbE network interface = 25 servers	NIC, transceiver, QoS-managed switches
High Availability	2-ports NICs, 2 servers for user workload	NIC, transceiver, server
Low Latency within DCI Cluster	microsecond-level with RoCEv2 support	NIC, transceiver, QoS-managed switches
Servers to WAN connections ratio	6 physical servers (2U) per WAN connection (one 15U rack can accommodate one 1U Ethernet switch, one 1U OOB switch, and **6** computing nodes)	Physical rack and computing/storage/network hardware size.
DCI Cluster Controller	2 (high availability)	Software implementation
QoS-managed gateway	10/25/100 Gbps	APN-T
Packet loss	0%	QoS-managed WAN
Packet ordering	preserved	Ethernet switch/fabric

4.4. DCI Cluster RIM based on huge and scalable disaggregated infrastructure with PCIe bus extension

This section outlines the Reference Model on huge and scalable disaggregated infrastructure with PCIe bus extension. First, possible use cases for this huge and scalable disaggregated infrastructure are outlined in **Section 4.4.1**. Then, the structure of this RIM is illustrated in **Section 4.4.2**. Finally, the basic structure is complemented by an example of

concrete devices to implement the structural RIM blocks and an overview of the resulting capabilities of such a Data-Centric Infrastructure (DCI) cluster is discussed in **Section 4.4.3**.

4.4.1. Use cases

Using huge and scalable disaggregated infrastructure with PCIe bus extension, servers and IOs such as GPUs and HDDs can be disaggregated at the scale of 10,000 or more units. These servers and IOs can be accommodated in independent racks by type: server, GPU, HDD, Flash and so on. These racks are connected by Ethernet and these resources can be dynamically disaggregated.

The following use cases are applicable for this model:

- On-demand resource allocation: resource can be allocated on-demand in data centers that accommodate many servers and IOs and disaggregate IO resources to the servers. One of the preferable applications is Machine Learning, especially in the inference process, which prefers to allocate IOs such as GPUs and FPGAs to servers on-demand.
- Remote accelerator sharing: IO resources like GPUs or FPGAs can be shared and disaggregated within the campus area or the same building. For example, GPU pools in a university are used as a local resource to augment the computing power of local servers from each research lab on the same campus.
- Data center cooling optimization: resources with high power consumption, such as GPU Box racks, can be placed on another well-cooled floors or use liquid-cooling racks to cool effectively.

Among the IOWN Global Forum DCI System use cases in **Section 2.1**, the following use cases can be applicable:

- Remote Media Production
- Green Computing with Remote GPU over APN
- IDH: compute-storage disaggregation
- IDH: off-site replication and compute-storage disaggregation
- CPS Area Management: large-scale AI inference with Live 4D Map database
- Large ML model training

4.4.2. RIM structure

This section outlines how to construct a large DCI cluster. The defining point of this DCR is that servers can connect IOs less than 2km away and communicate with each other using PCIe packets. Servers can access remote IOs as if they were in local PCIe slots, so they do not require any additional software to access remote IOs.

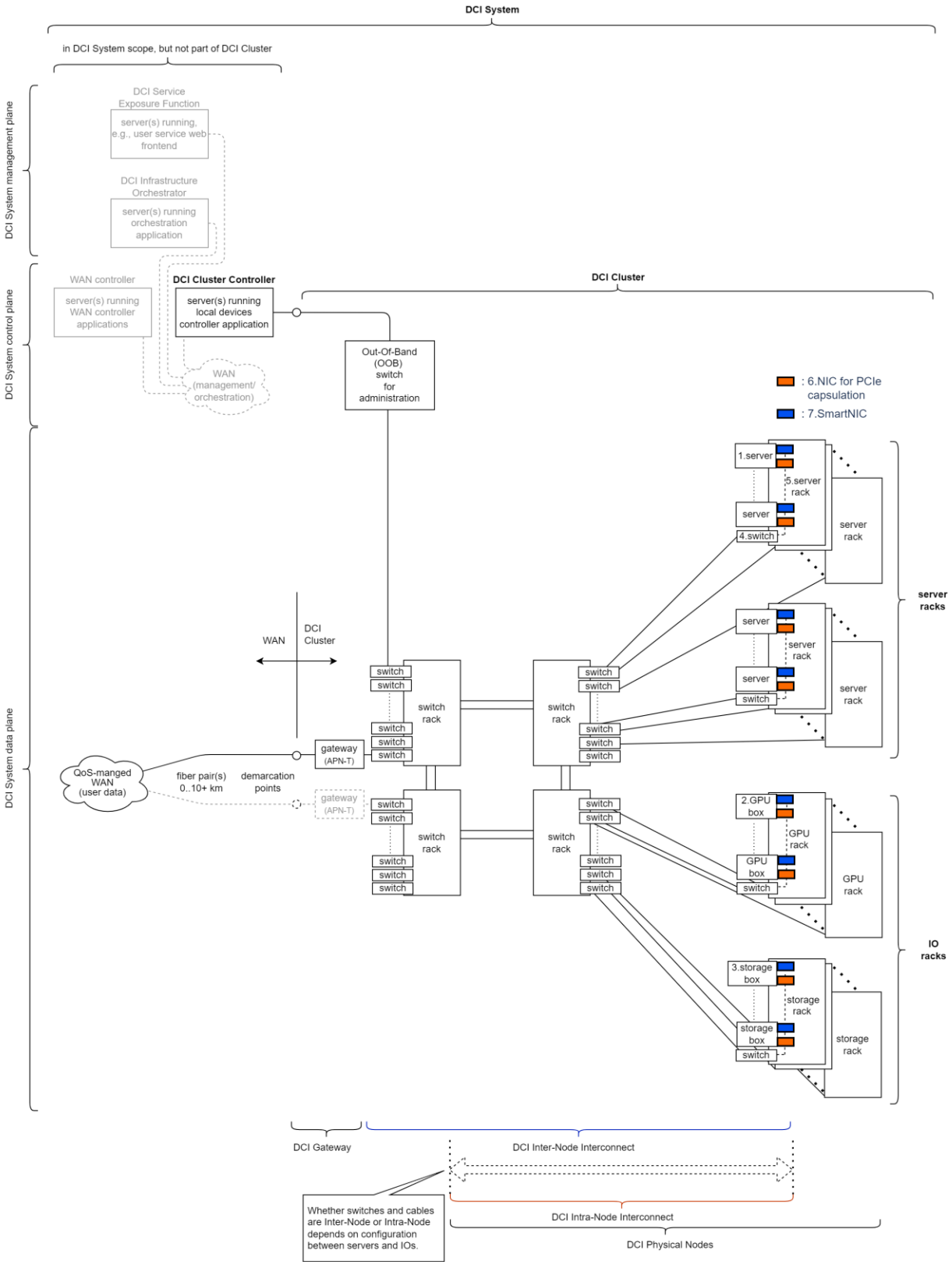

Figure 4.5: DCI Cluster for huge and scalable disaggregated infrastructure.

4.4.2.1. NIC for PCIe capsulation

Each server and IO box in this DCR has a NIC for PCIe capsulation and is connected through Ethernet cables and switches. This NIC encapsulates and decapsulates PCIe packets in Ethernet frames to communicate with remote components virtually, though servers and remote IOs are physically connected by Ethernet.

This model uses Ethernet instead of PCIe as a physical cable link. This is because Ethernet has the following merits in building huge-scale models compared to local buses like PCIe: First, the distance between resources can extend to several kilometers, compared to several meters as PCIe. Second, using Ethernet makes it easier to scale out the resources in the network. Adding and removing resources with Ethernet is much easier than PCIe. Third, it requires lower cost to make a large-scale system. Comparing both main switch LSI of PCIe and Ethernet, the cost of PCIe per link is much more expensive than that of Ethernet.

NICs for PCIe capsulation implement functions for congestion control and retransmission. This avoids packet loss at the Ethernet layer.

NICs for PCIe capsulation have a table to translate addresses between PCIe Bus/Function/Device numbers and MAC addresses. So, when encapsulating PCIe packets, NICs translate the destination address from Bus/Function/Device numbers to MAC addresses. Then, Ethernet switches route Ethernet frames appropriately to the proper servers and IOs.

4.4.2.2. Racks for Servers and each type of IOs

With NICs for PCIe capsulation, it becomes possible to connect IOs and Servers through Ethernet beyond racks, as is described in **Section 4.4.2.1**. Then, it makes it possible to allocate each type of resources to independent racks, as shown in the right of **Figure 4.6**. This allocation has some merits compared to allocating all types of resources in 1 rack, as shown in the left part of **Figure 4.6**. The first merit is cooling efficiency. With this disaggregation, GPUs, storage, and servers can be allocated to different racks. Because GPUs consume more power and their temperature rises faster, more cooling energy is required for GPU boxes. By applying water-cooling to these GPU racks or deploying them on another floor where the cooling effect is strong, the total cost for cooling efficiency and power consumption can be optimized.

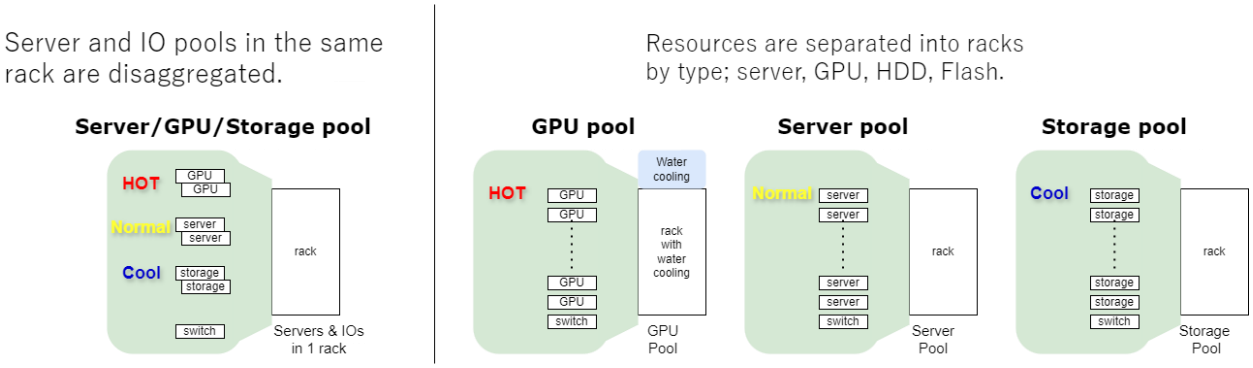

Figure 4.6: Disaggregation System between racks.

Here are other merits of resource assignment to independent racks by resource types, as shown in **Figure 4.7**.

1. Aggregation of large-scale servers and IOs: This system can share many resources in the same Ethernet network.
2. Resource usage efficiently: Allocating pooled IOs to servers at different racks increases total usage efficiency.
3. Lower power: For example, if the GPU resource usage rate becomes low and all resources in a rack are not used, the rack can be shut down, contributing to power savings.
4. Independent upgrade: For example, if all GPU pools need to be replaced with a new model, it is possible to replace and upgrade these GPUs rack by rack while other racks are active.

5. High Availability: For example, if a whole rack is shut down due to a power-failure or overheating, the system continues to run with another rack with resources of the same model, which resides on stand-by.

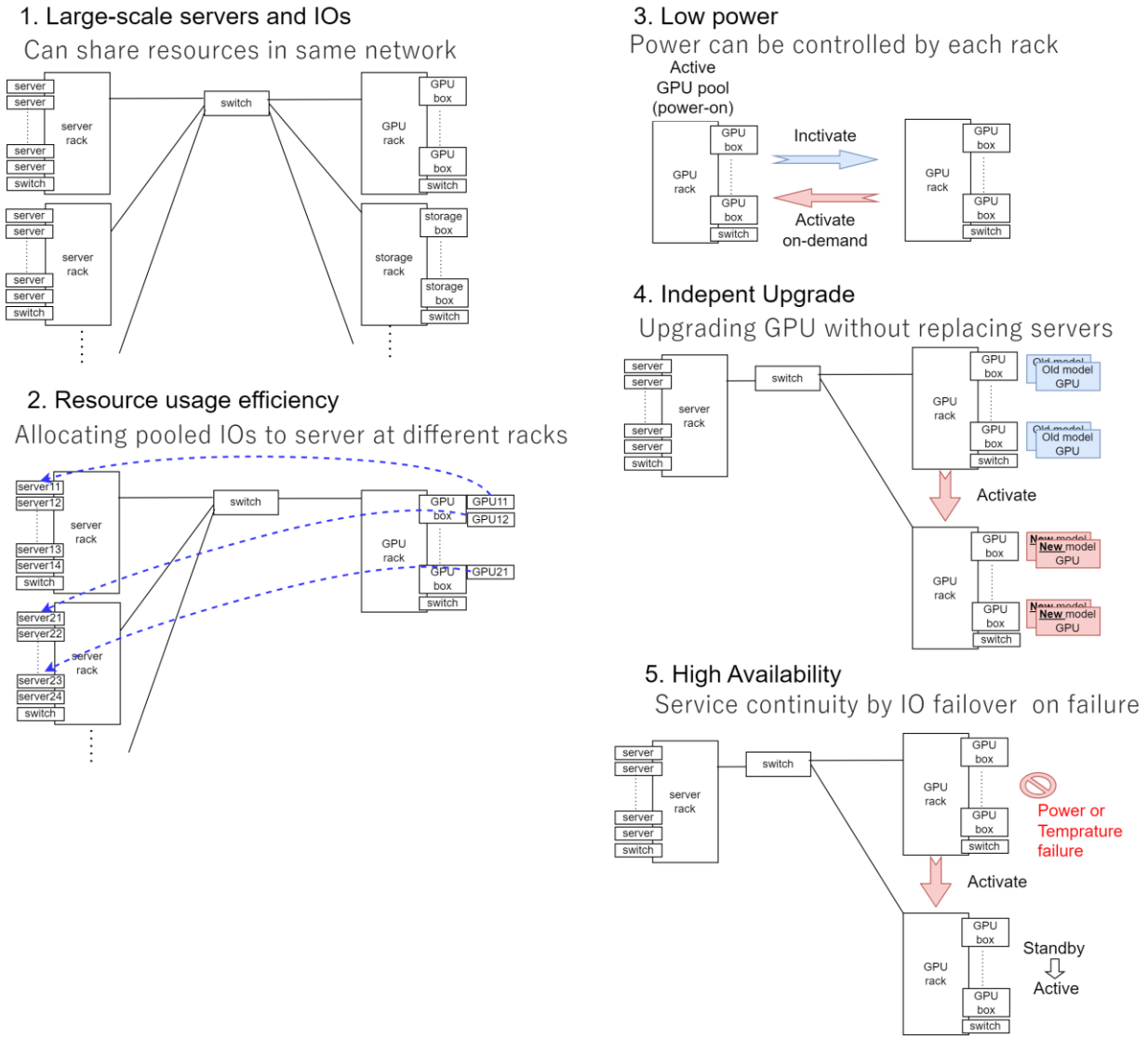

Figure 4.7: Merits on disaggregation system between racks.

4.4.2.3. Management of disaggregation system

For disaggregating IOs to appropriate servers, a remote operator manages this aggregation through Ethernet. The operator can be located outside the data center.

A remote operator has two main tasks when managing a disaggregation system: collecting resource information through Ethernet and assigning resources to groups through Ethernet.

Remote operators collect information on disaggregated resources from NICs for PCIe capsulation, which periodically send packets with its resource information. A remote operator receives them through a NIC and stores it. Then, a remote operator assigns the same group numbers to resources which are expected to connect each other logically. **Figure 4.8** shows an example of the physical view and the logical view where a remote operator assigns resources to 4 independent groups. The assignment indication is sent from a NIC of the remote operator to the NICs for PCIe

capsulation of their resources. Servers, IOs in the same group and switch ports between them are logically in a DCI Physical Node. So, the switch with the ports is logically a DCI Intra-Node Interconnect.

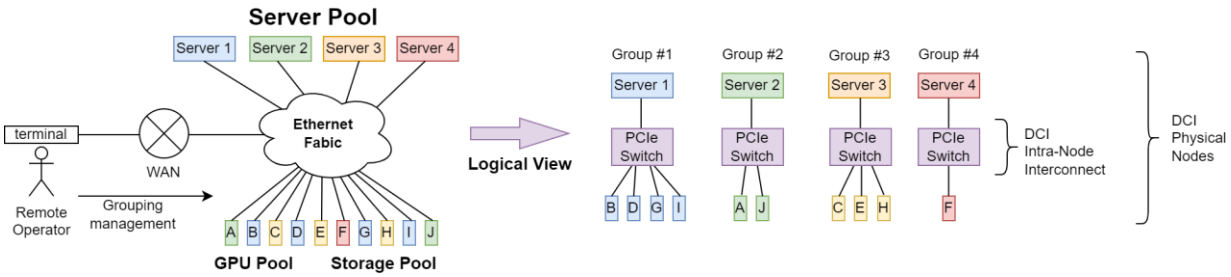

Figure 4.8: Disaggregation resource management from remote operator.

4.4.2.4. SmartNIC

SmartNICs send and receive Ethernet frames through a gateway. They can also communicate with each other without frames that encapsulate PCIe packets.

In **Figure 4.5**, the SmartNIC and the NIC for PCIe capsulation use the same Ethernet cables and switches. To assure a QoS-managed transfer between SmartNICs and the WAN, PCIe capsulation NICs and switches must also have QoS functions.
If it is not preferable to implement same-level QoS functionality in NICs for PCIe capsulation or to prepare sophisticated QoS-assured functionality for all switches, then an alternative solution is to separate the networks with SmartNICs from networks with PCIe capsulation NICs. In this case, less expensive switches can be applied to networks with PCIe capsulation NICs.

4.4.3. Example implementation choices

This section outlines one possible implementation of a DCI Cluster that is structured as illustrated in **Figure 4.5**. First, concrete component classes for each part of the DCI Cluster are selected in **Section 4.4.3.1**. After that, the projected capabilities of a DCI Cluster implemented with these components are listed in **Section 4.4.3.2**.

4.4.3.1 Selected component classes

Table 4.8 lists the implementation choices for a high-QoS fabric that handles network connections only on the port level.

Table 4.8: Example implementation choices for a DCR constructed around a huge and scalable disaggregated infrastructure with PCIe bus extension.

#	DCR DIAGRAM BLOCK	EXAMPLE IMPLEMENTATION CHOICE
1	Server	classic x86 COTS servers 4,800 servers
2	GPU Box	4U Box that can mount 8 GPUs. For interface to network, each box has 2 Ethernet NICs to capsulate PCIe packets.
3	Storage Box	500 x 2U Boxes that can mount 24 HDDs and 1 Ethernet NIC to capsulate PCIe packets.
4	Switch	325 x 100 Gbps Ethernet switches

5	Rack	287 19inch racks server racks, GPU racks, storage racks and switch racks are of the same type.
6	NIC for PCIe capsulation	8,200 PCIe Gen4 x8, 100 Gbps Ethernet interface. QoS function is required to be implemented.
7	SmartNIC	8,200 PCIe 4.0x8/16, 100 Gbps QoS-managed NICs

4.4.3.2 Resulting capabilities summary

Based upon the example implementation choices as outlined in **Section 4.4.3.1**, **Table 4.9** lists the projected capabilities for a DCI Cluster that is structured as illustrated in **Figure 4.5**.

Table 4.9: Example capabilities for a DCR constructed around a huge and scalable disaggregated infrastructure with PCIe bus extension.

PROPERTY	PROJECTED/ESTIMATED VALUE	DETERMINED BY
Bandwidth per interface	100 Gbps	Bandwidth per interface
Servers and IOs disaggregation	Servers and each kind of IOs can be mounted to individual racks	Disaggregation function
Number of servers in 1 system	4,800 servers	(arbitrary)
Number of IOs in 1 system	21,600 IOs	(arbitrary)
Remote IOs per server	32 GPU 96 HDD	IO Box
Power control rack by rack	Possible	Disaggregation function
Distance between servers and IOs	~several km	Legacy IO Performance
Packet ordering	Preserved	PCIe Specification

5. Conclusion

The concrete Data-Centric Infrastructure (DCI) Cluster implementation strategies developed above clearly indicate that it is already possible to build DCI Systems exclusively with today's technology that supports the fusion of server and WAN base infrastructures to enable tomorrow's IOWN Global Forum use cases.

This report discussed implementation strategies for DCI Clusters. First, IOWN Global Forum use cases were visited, highlighting requirements that exceed the capabilities of today's infrastructures. Succeeding, DCI Clusters were introduced as part of the overall DCI System solution. After that, multiple DCI Cluster implementation strategies were outlined for various system sizes and use cases. Finally, for each of these implementations, concrete product classes for implementation were given, as well as overviews of the resulting DCI Cluster capabilities.

Future work toward DCI System implementation might use these results to start proof-of-concept implementation activities. Furthermore, activities to update the DCI functional architecture may want to take the developed DCI Cluster RIMs and DCI user interface examples into account to ensure that the DCI FA is prepared for the described use cases and cluster implementations.

One insight from crafting the RIMs above is that describing workloads first is crucial for this new area of DCI Cluster design. Then, carefully considering aspects such as versatility across use cases, computation and storage hardware device types, the minimum number of devices within a cluster, and appropriate WAN connectivity will be vital for creating viable commercial products.

Appendix A. Workload for initial DCI Cluster RIM (DCR) activities

This section discusses the workload model for initial Data-Centric Infrastructure (DCI) Clusters, i.e., what applications are expected to be deployed. First, **Section A.1** briefly introduces the IOWN Global Forum DCI System use cases. Then, the communication patterns of these use cases are classified into two main categories: long-range point-to-point communication and short-range tiered communication (**Section A.2**).

A.1. IOWN Global Forum DCI System use case descriptions

This section introduces several IOWN Global Forum application use cases, focusing solely on the parts of the use cases that are most demanding to IOWN Global Forum DCI Systems. The use case topologies are simplified to the maximum extent possible to identify commonalities in their structure for further categorization in later sections.

A.1.1. AIC Interactive Live Music: streaming between customer premises and data center

The AIC Interactive Live Music use case describes an entertainment service in which geographically dispersed users participate together in a concert event. User movement is captured by various sensors and used to control avatars. Users wear head-mounted displays (HMD) to perceive the virtual reality space with artists and other users.

One key point of this use case is that the rendering for the user is performed centrally in GPUs located in data centers. For proper immersion, the maximum delay of the rendering feedback loop from the movement of the user's head to a movement on the user's HMD should be within about 10ms. Assuming a rendering delay of 4ms, this leaves 3ms for transmit and display.

Furthermore, due to the required low latency, no or only lightweight video compression can be used, resulting in a high bandwidth requirement of up to 100 Gbps. In turn, the necessary bandwidth over distance will likely require advanced transmission means such as RDMA, which demands virtually lossless links.

This critical part of this use case is illustrated in **Figure A.1** below.

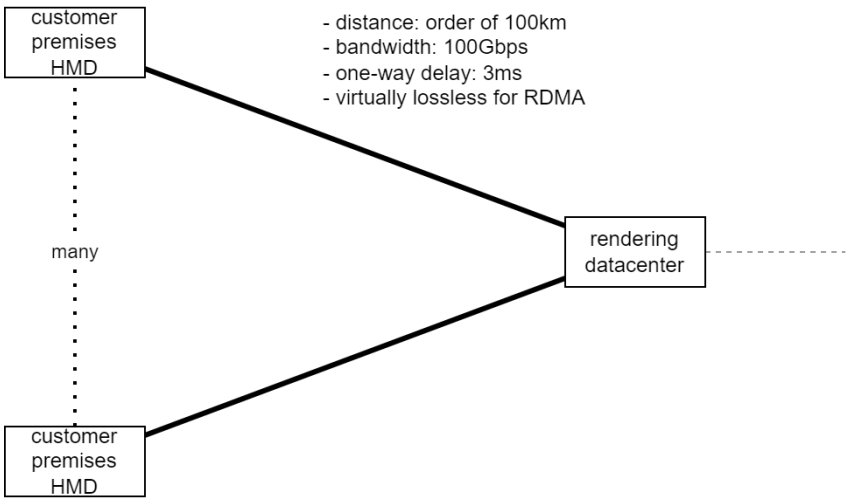

Figure A.1: Key part of the AIC Interactive Live Music use case.

A.1.2. Remote-controlled robot inspection: streaming between plant and operation center

The remote-controlled robot inspection use case involves controlling and streaming audiovisual content between an industrial plant site and a robot operation center with human operators.

From an infrastructure perspective, the key point of this use case is the live connection between the maintenance robot or drone at the plant side and the operation center. Multiple robots would need to be controlled from the same operation center. Like the requirements in **Section A.1.1**, the required bandwidth of the link is expected to be 100 Gbps for live streaming, control, and feedback. Latency and jitter must be contained within 100ms, and the link must be virtually lossless to use RDMA.

This part of the use case is shown in **Figure A.2** below.

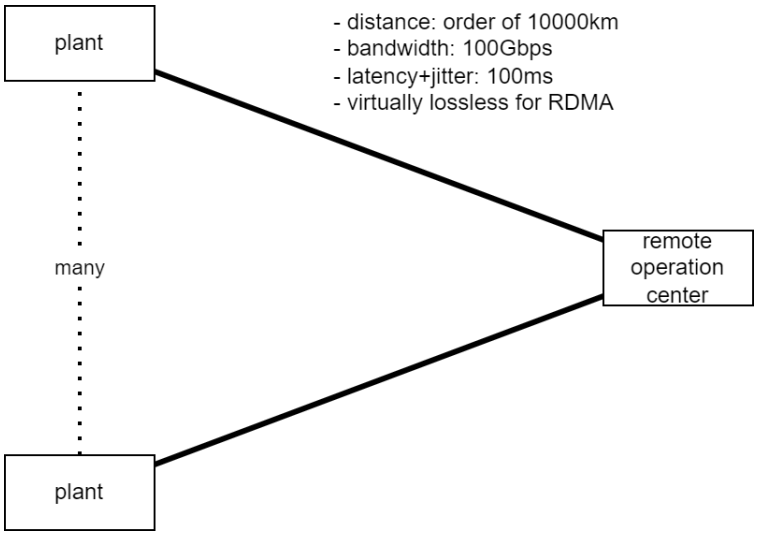

Figure A.2: Key part of the Remote-Controlled Robot Inspection use case.

A.1.3. Remote media production: streaming between production site and data center

The remote media production use case concerns the geographical separation of the recording site, the media production equipment, and its operators. While cameras and audio-recording devices must remain on-site, media processing hardware is moved into a remote central media processing data center.

From an infrastructure perspective, the key point of this use case is the link between the recording location and the media processing data center. The required bandwidth per site is assumed to be 20 Gbps since strong compression cannot be used due to quality requirements and latency constraints for live events. The upper bound for acceptable latency and jitter is assumed to be 16.6ms (corresponding to 60 frames per second), and the link would need to be virtually lossless for the protocols carrying the media data, e.g., SMPTE ST2110.

This part of the use case is shown in **Figure A.3** below.

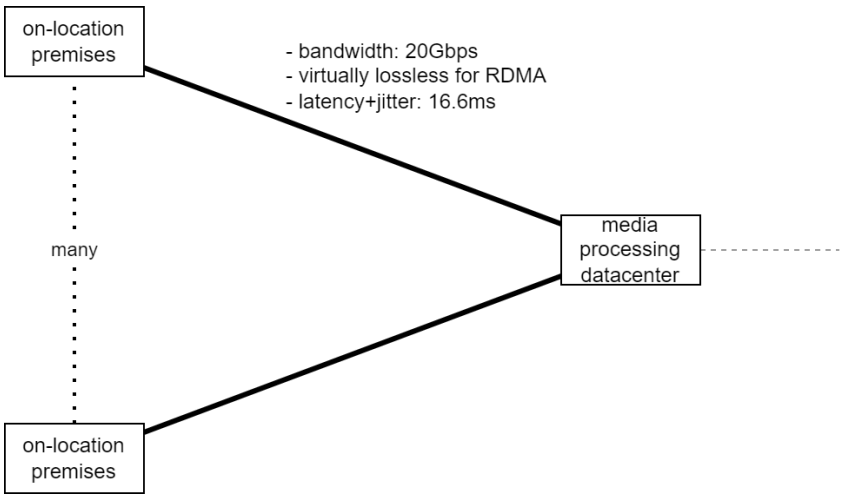

Figure A.3: Key part of the Remote Media Production use case.

A.1.4. Financial industry services infrastructure

The financial data center interconnect use case describes data centers that are geographically separated and host virtual machines that can be migrated quickly between locations.

From an infrastructure perspective, the key point of this use case is the links between individual data centers. For this use case, these links must span 50km or more, provide bandwidth to individual servers of at least 25 Gbps, and have low latency and jitter no higher than 0.5ms.

This part of the use case is illustrated in **Figure A.4**.

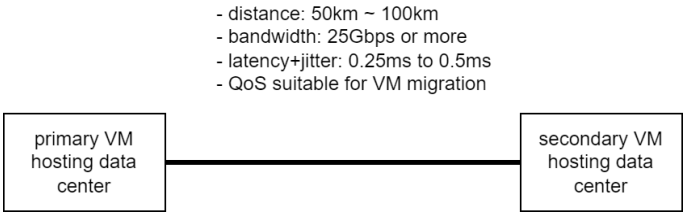

Figure A.4: Key part of the financial data center interconnect early adaption use case.

A.1.5. Green Computing with Remote GPU over APN

The green computing with remote GPU over APN use case describes how AI training can be performed when the training computing farm and the storage facility that holds the data to train on are geographically apart.

The key point of this use case is to connect both distant facilities with sufficient connection quality to enable remote file access. The use case describes links that bridge at least 100km, provide bandwidth of 100 Gbps or more, and limit latency and jitter to a couple of milliseconds.

This part of the use case is depicted in **Figure A.5**.

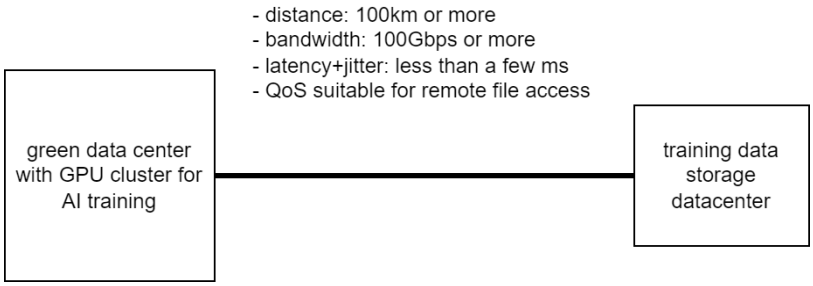

Figure A.5: Key part of the Green Computing with Remote GPU over APN early adaption use case.

A.1.6. IOWN Data Hub: database synchronization across data centers

The IOWN Data Hub is expected to use the DCI System infrastructure in multiple ways. One use case is to use QoS-managed high bandwidth links to synchronize with multiple remote database replicas for redundancy. The data center where the primary copy of the database is hosted will likely contain large numbers of servers equipped with CPUs and storage devices. At the same time, data centers in which replicas are placed may also specialize in providing storage.

The key requirements for such links are expected to be latency and jitter bounded to 2ms, being virtually lossless to allow the use of RDMA, and bandwidth of 25 Gbps or more.

This aspect of the IOWN Data Hub is illustrated in **Figure A.6**.

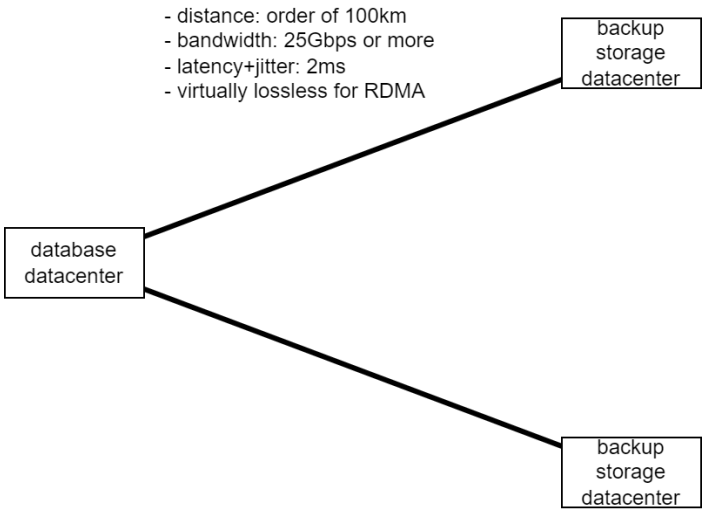

Figure A.6: Key part of the IOWN Data Hub use case remote synchronization aspect.

A.1.7. Sovereign cloud: compute resources in data centers access on-premises storage

Sovereign Computing enables users to keep data stores on their premises and rent computing capabilities such as CPUs, GPUs, or other accelerators on-demand from central sovereign computing data centers while fully preserving confidentiality of data.

Since this use case's primary purpose is to share data remotely, the requirements are expected to be qualitatively similar to those introduced in **Section A.1.6** above.

This scenario is shown in **Figure A.7**.

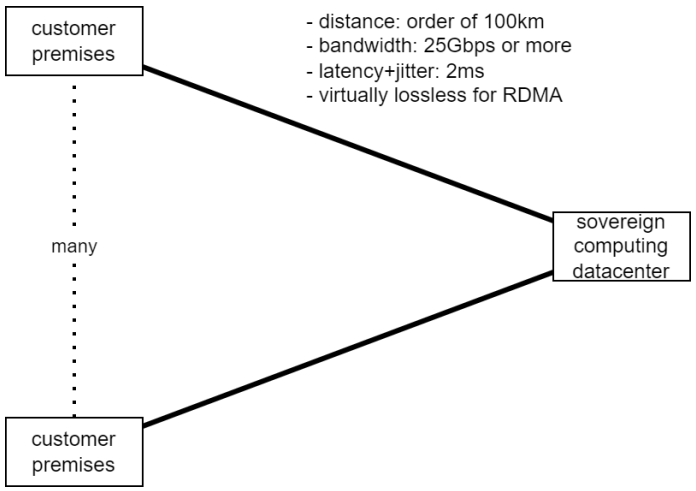

Figure A.7: Key part of the sovereign cloud use case.

A.1.8. IOWN Data Hub: compute and storage disaggregation inside data centers

Another use case for a DCI System that involves the IOWN Data Hub is the disaggregation of compute and storage nodes within a single data center. One approach to achieve such compute-storage disaggregation is to prepare specialized servers with compute and storage capabilities and interconnect these two types of servers in a tiered topology.

The QoS requirements for interconnection between servers are expected to be qualitatively similar to the other storage use cases, such as those highlighted in **Section A.1.6**. However, the upper bounds of latency and jitter may need to be lowered when low-level storage protocols are to be used.

Such a topology is visualized in **Figure A.8**.

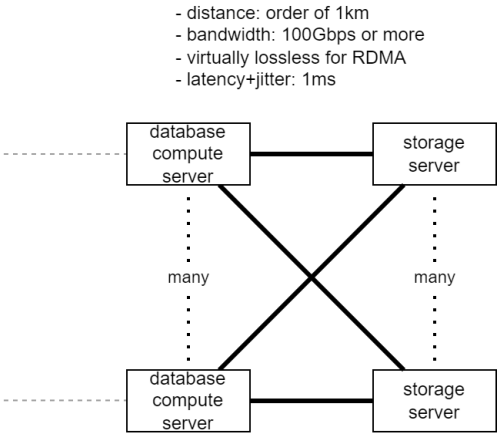

Figure A.8: Key part of the IOWN Data Hub use case compute-storage disaggregation aspect.

A.1.9. Off-site replication and compute-storage disaggregation

A more advanced use case for DCI Clusters is hosting database systems that employ multi-tiered architectures to realize compute-storage disaggregation inside DCI Clusters and use high-QoS connectivity to connect database instances geographically apart to implement low-latency replication and synchronization. This results in a combination

of the long-range synchronization use case introduced above in **Section A.1.6** and the compute-storage disaggregation use case introduced in **Section A.1.8**.

A possible use case structure is illustrated in **Figure A.9**.

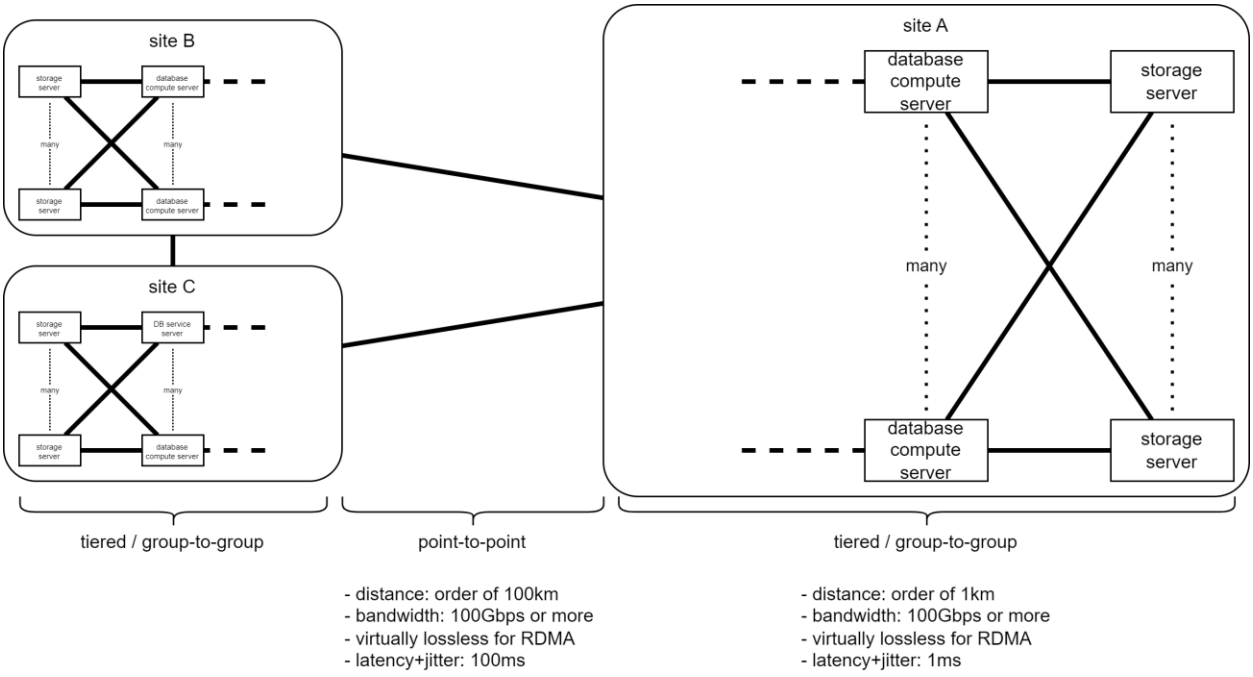

Figure A.9: Key parts of off-site replication and compute-storage disaggregation use cases taken together.

A.1.10. CPS Area Management: large-scale AI inference with Live 4D Map database

Implementations of the CPS Area Management use case are expected to combine high-bandwidth, long-range transmission of security camera footage to data centers, and inside these data centers, will likely deploy servers communicating in a multi-tiered topology. On the first tier of this topology inside data centers, typically, multiple kinds of servers are expected: for example, ingestion servers that are equipped with high-bandwidth NICs for communication with remote video and CPUs for high-throughput database interaction, and in addition, analysis servers equipped with GPUs for video analysis. On the second tier, database servers fitted with CPUs are expected to mediate between ingestion and analysis servers and the third tier. This third and final tier will likely be composed of servers that contain large amounts of storage devices and possibly acceleration hardware such as SmartNICs to offload storage access processing.

The communication QoS requirements of this use case combine the long-range streaming requirements introduced in **Section A.1.6** with the in-data-center communication requirements highlighted in **Section A.1.8**. Furthermore, the more complex communication hierarchy with heterogeneous nodes on the topmost level and the larger required hardware diversity are pivotal differences to database applications discussed in **Section A.1.9**.

Figure **A.10** provides an overview of the key parts of this use case from an infrastructure perspective.

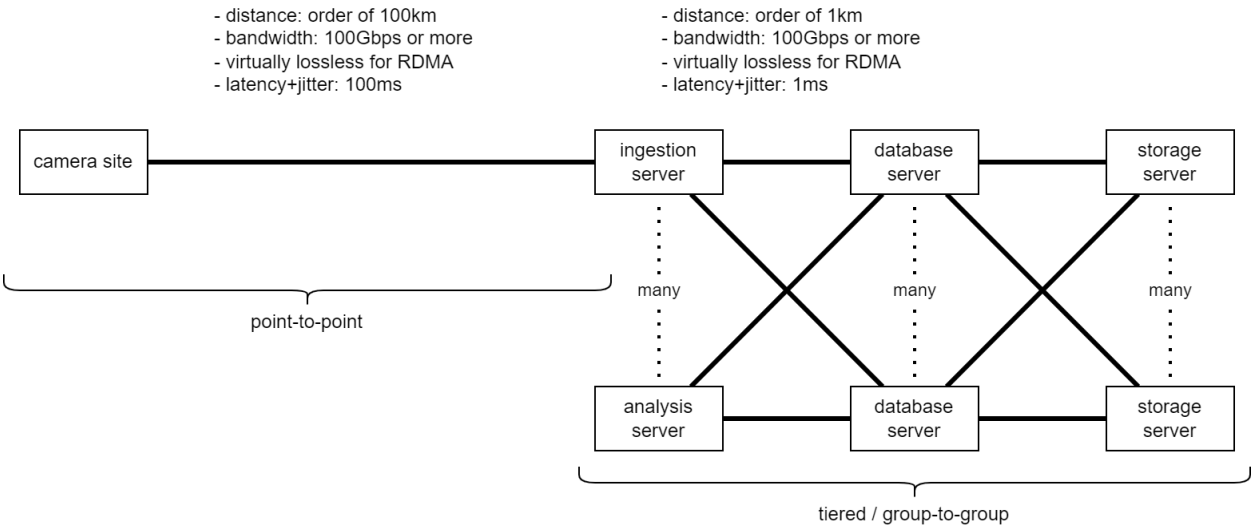

Figure A.10: Key parts of the CPS Area Management use case.

A.1.11. VRAN: MFH eCPRI streaming between antenna sites and edge data centers

Virtual radio access network (VRAN) mobile front-haul (MFH) communication typically carries enhanced common public radio interface (eCPRI) protocol information between radio units (RU) directly attached to radio antennas on the radio site and virtual distributed units (vDUs) that are responsible for further radio signal processing that are located off-site within an edge data center. Devices at RU sites are expected to be specialized DSP hardware with attached network devices, and vDU sites would contain arrays of specialized eCPRI processing accelerator cards.

Compared to the use cases introduced above, the communication links for VRAN MFH will likely have comparable bandwidth requirements but require multiple orders of stricter latency and jitter bounds in the order of 20 microseconds.

The VRAN MFH use case is illustrated in **Figure A.11**.

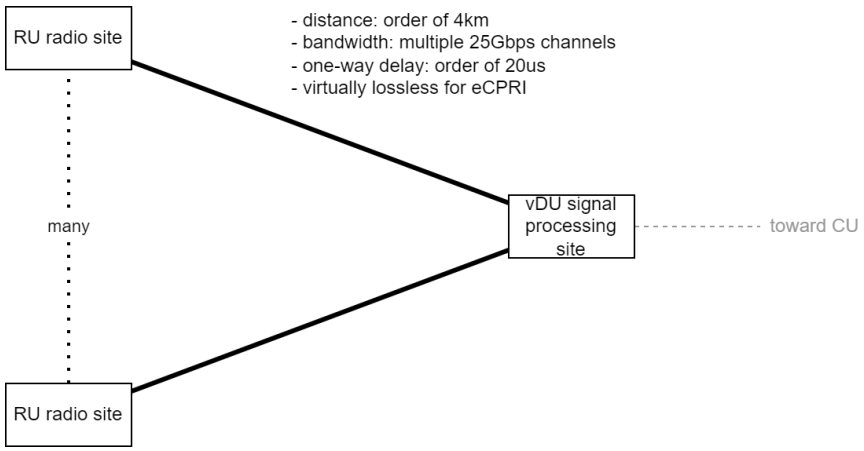

Figure A.11: Key part of the VRAN/MFH use case.

A.1.12. Large ML model training: computation on many tightly coupled accelerators

Large machine-learning model training use cases require large homogeneous arrays of accelerators. These accelerators execute tensor programs, which perform all-reduce operations at regular intervals. This essentially forces all devices to exchange large amounts of information directly or indirectly with all other devices.

Physical network topologies and accompanying communication patterns for these use cases are an area of active research. In addition, in contrast to the use cases introduced above, for this large ML model training use case, the most critical goal to achieve from a communication infrastructure point of view is the time required to complete a whole all-reduce step among all accelerators and only to a lesser extent the latency and jitter on individual links.

Figure A.12 outlines the case that all tensor programs exchange information regularly with all other programs. For example, the underlying physical communication topology may be a hybrid of meshes connected to each other in a network hierarchy.

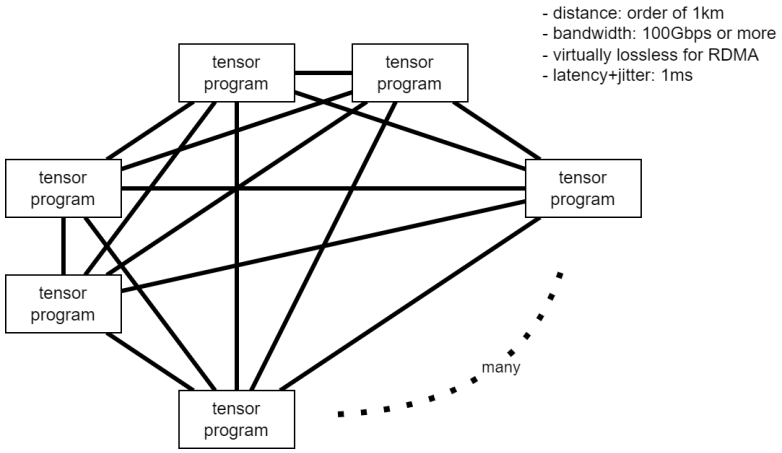

Figure A.12: Structure of the large ML model training use case.

A.2. Further consideration of IOWN Global Forum use cases communication patterns

The elements of the communication topologies appearing in the use cases introduced above can be classified into two main classes: The first prominent class is edge-center splits, comprising many small-scale edge sites exchanging large amounts of data with central sites. The second class is formed by two groups of devices in proximity exchanging large quantities of data primarily with each other.

The remainder of this section highlights these communication topology classes in **Section A.2.1** and **Section A.2.2**.

A.2.1. Long-range point-to-point QoS-managed communication

The first identified pattern in IOWN Global Forum use cases is that many use cases require point-to-point connections between geographically distant locations. The following sections introduce the overall communication pattern, the typical quality-of-service required for connectivity, and several concrete use cases comprising this long-range point-to-point communication pattern.

A.2.1.1. Communication pattern

Considering individual use cases in more detail reveals that, in many cases, use cases allocate large amounts of compute and/or storage resources in central locations and transfer large amounts of data to and from these geographically dispersed and distant locations. As a result, the communication topology is a set of point-to-point connections between edge devices and individual devices at one or more central locations. The resulting pattern is illustrated in **Figure A.13**. As a further note, the compute devices in the central location are often connected to other local devices for further processing. The connections to these further processing stages may also have high QoS requirements.

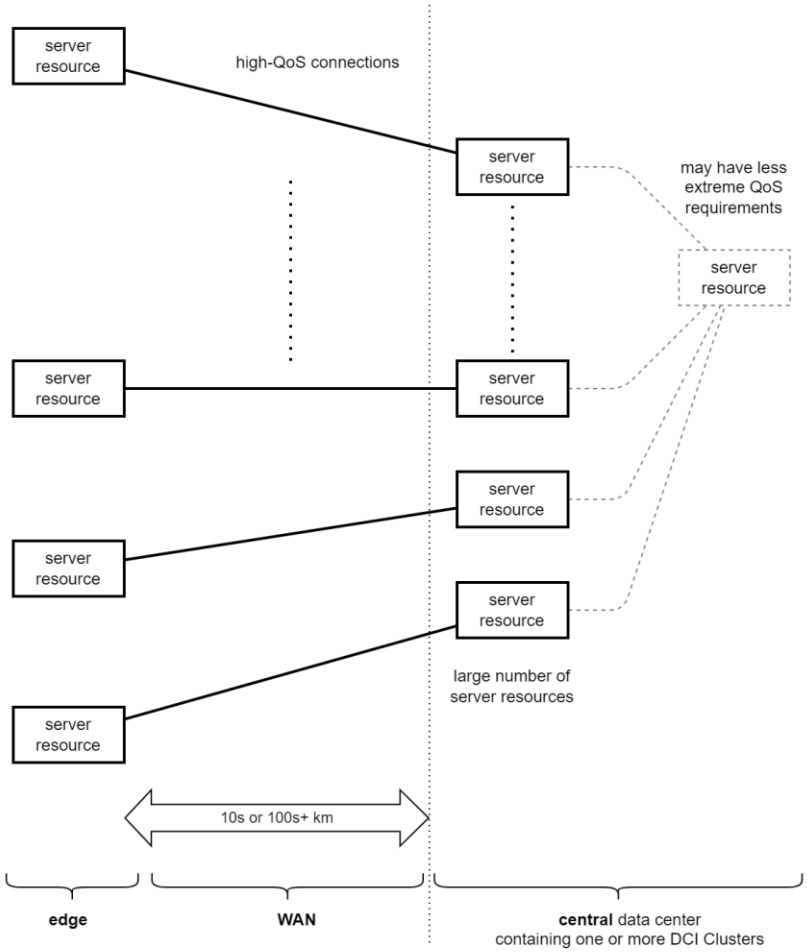

Figure A.13: Communication pattern of long-range point-to-point QoS-managed communication.

A.2.1.2. QoS requirements

The use cases of this communication pattern typically require QoS such as the following for their long-range connections to remote locations:

- bandwidth of 25 Gbps or 100 Gbps, and some use cases may profit from even higher bandwidth;
- additional latency and jitter apart from light speed fiber delay being significantly shorter than 1ms;
- packet loss being virtually absent, and
- packet ordering is guaranteed.

The bandwidth, latency, and jitter requirements stem directly from the high-level requirements of the user's application. In contrast, the absence of packet loss stems indirectly from latency requirements and the motivation to use RDMA effectively to harness all available bandwidth. Finally, the requirement for packet ordering guarantees also originates from common characteristics of today's RDMA technology.

A.2.1.3. Applicable use cases

Use cases that include a pattern as described above as part of their communication topology include:

- AIC Interactive Live Music.

  - Many audience premises (edge) connecting to a rendering server (center).
- Remote Controlled Robot Inspection.
  - One or more plant sites (edge) connecting either to a data center for storage (center) or an operator office (center).
- Remote Media Production.
  - One or more event sites (edge) connecting to a data center with production equipment (center).
- Database synchronization.
  - Multiple sides are dispersed geographically and connected to each other for replication. (The resulting topology can be imagined as multiple overlapping star topologies.)
- IDH: off-site replication and compute-storage disaggregation.
  - Similar to plain data synchronization, advanced IDH implementations are expected to replicate their data to remote sites using DCI Systems.
- Sovereign cloud.
  - On-premises systems (edge) connecting to computing hardware rental data centers (center).
- Early adoption use case: Green Computing with Remote GPU over APN.
  - On-premises data storage (edge) connecting to rented specialized GPU clusters in data centers (center).
- CPS Area Management for Security.
  - Local aggregation nodes (edge) streaming to analysis systems (center).
- VRAN Mobile Front-Haul (MFH)
  - Disaggregation of radio unit (RU) and distributed unit (DU): While the communication topology itself of this use case at first glance appears mostly identical to the other use cases in this long-range point-to-point QoS-managed class, MFH has much more stringent QoS requirements compared to other use cases. In particular, MFH demands bounds for latency and jitter that are one or more orders shorter than most of the other use cases of this pattern (e.g., less than **160µs** for MFH compared to around 1ms demanded by other use cases). Further investigation is required to decide whether implementation approaches for general long-range point-to-point use cases will apply to the VRAN MFH use case.

A.2.2. Short-range group-to-group QoS-managed communication

The second identified pattern in IOWN Global Forum use cases is that multiple use cases require group-to-group connections between devices in proximity. The following sections introduce the overall communication pattern, the typical quality-of-service required for connectivity, and concrete use cases, including this long-range point-to-point communication pattern.

A.2.2.1. Communication pattern

Considering individual use cases in more detail reveals that in multiple cases, use cases require two groups of different compute devices, where each node of each group communicates mainly with a fixed set of nodes of the other group with high QoS. As a result, for such use cases, the logical communication topology is often close to a two-tier or even bipartite topology, with all nodes being close to each other and connected to the same local network. The resulting pattern is illustrated in **Figure A.14**.

Also, as introduced above in **Section A.2.1**, one set of nodes often communicates with one or more geographically distant nodes.

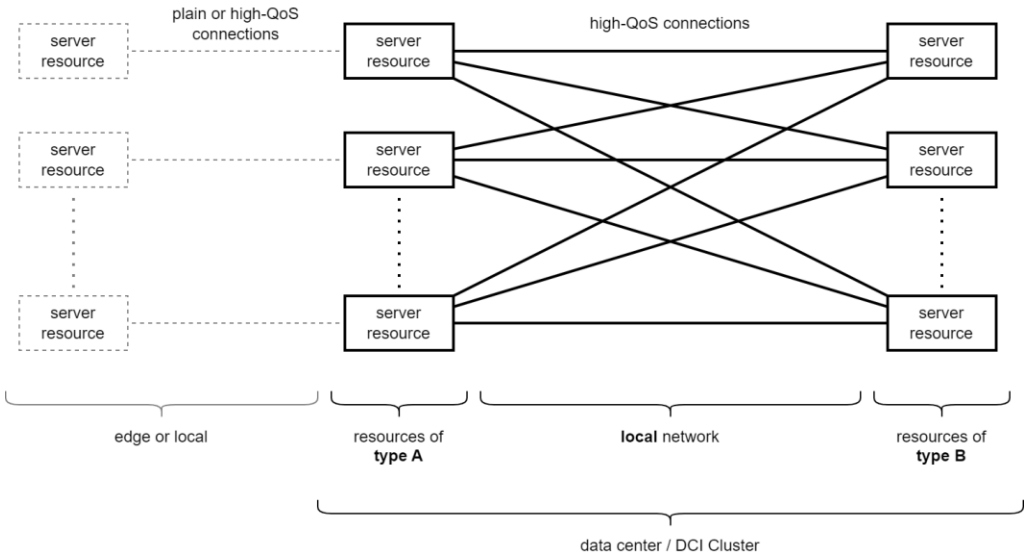

Figure A.14: Communication pattern of short-range group-to-group QoS-managed communication.

A.2.2.2. QoS requirements

The QoS requirements of this use case's pattern are largely the same as those of the other communication topology patterns introduced in **Section A.2.1.2**.

A.2.2.3. Applicable use cases

Use cases that include a pattern as described above as part of their communication topology include:

- CPS Area Management for Security.
  - Analysis compute nodes exchanging video data with database frontend nodes.
- Database compute and storage disaggregation.
  - Database service nodes interacting with database storage backend nodes.
- IDH: off-site replication and compute-storage disaggregation.
  - Beyond compute-storage disaggregation on its own, high-QoS WAN communication needs to be integrated with disaggregation.
- Large ML Model training.
  - High-performance interconnect among GPUs within ML training clusters: While large ML model training ideally requires a full-mesh topology that, at first glance, may appear similar to the other topologies in this class. However, it appears that large ML model training has high but very different QoS requirements compared to most other use cases: Most IOWN Global Forum use cases of the short-range group-to-group class demand minimizing latency/jitter and reducing packet loss/reordering for individual connections, but in contrast, large ML model training requires minimizing the time needed to execute collective communication operations. This results in a focus on high bandwidth and some tolerance to other QoS properties such as latency and packet order guarantees. Large ML model training requirements are examined in [**UECWP**].

Appendix B. Common framework for initial DCI Cluster RIM construction

This appendix outlines a common framework for the hardware initial Data-Centric Infrastructure (DCI) Cluster RIMs (DCRs). To keep the discussion of DCRs as straightforward as possible, the generally accepted overall structure of today's data centers is chosen as a starting point.

Furthermore, the basic structure for DCI Clusters in this section is not normative or a requirement; it is only chosen to establish a common base for the later introduction of DCRs. DCI implementations can deviate from this structure as implementors see fit, as long as accompanying DCI Cluster Controllers provide north-bound interfaces as laid out in the DCI Functional Architecture [**IOWNGF-DCIFA2**].

In the following, today's data center structure is outlined in **Section B.1**. Based on these observations, the generic structure for DCRs is introduced in **Section B.2**. Succeeding, the main structural hardware building blocks of DCI Clusters are treated in **Section B.3** followed by a discussion of interfaces between these blocks in **Section B.4**. Finally, **Section B.5** closes, summarizing that the described framework was used for the DCR templates in **Appendix C**.

B.1. Starting point: today's data centers

Data centers are key components of today's computing and networking infrastructures. Data centers provide large amounts of compute and storage capabilities to their users. This section first observes the structure of today's data centers in **Section B.1.1**, then highlights the commonalities and differences between data centers and DCI Clusters in **Section B.1.2**.

B.1.1. Overall simplified internal structure of today's data centers

Considering the high-level structure of today's data centers, this data center structure can be abstracted as shown in **Figure B.1** by WAN, Gateway, Network, and Server Resource blocks:

- *Server Resource:* Server Resource components represent hardware for executing user workloads. Examples of such components are racks with installed servers. These servers may contain arbitrary hardware, such as CPUs, GPUs, other accelerators, memory, network interfaces, and storage devices. Furthermore, server resource blocks could also contain non-server devices, such as switches that are dynamically allocatable and directly controllable by users.
- *Network:* Network components connect server resource components to each other and to gateway components. Examples of such components include leaf-spine, Clos, and fat-tree network topologies.
- *Gateway:* Gateway components represent hardware that facilitates communication between the inside of data centers and the outside of data centers. Examples of such components are router devices that may be equipped with or connected to further devices for long-range communication.
- *WAN:* WAN blocks represent the means of long-range communication. Examples are direct data center interconnects, metro networks, or nationwide backbone networks.

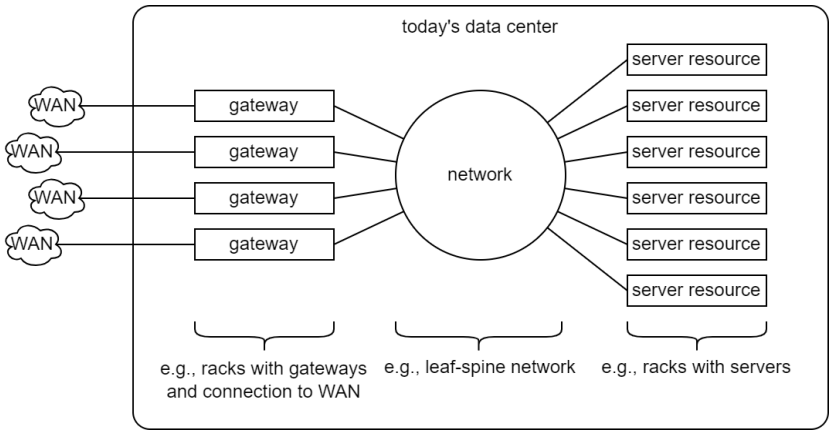

Figure B.1: Overall simplified structure of today's data centers.

When described in greater detail, data centers typically feature many more components, such as meet-me rooms and supporting power and cooling infrastructure. However, these details are outside of the scope of this document's discussion of the initial DCRs.

As outlined above, the basic structure of WAN, gateway, network, and server resource blocks shall serve as the baseline structure for DCRs, which will be introduced in later sections.

B.1.2. Commonalities and differences of DCI Clusters with data centers

Considering the IOWN Global Forum DCI Functional Architecture [**IOWNGF-DCIFA2**], the term "DCI Cluster" may appear to have a meaning identical to or similar to "data center" at first. However, there are several key differences. This section highlights the commonalities and differences between data centers and DCI Clusters.

DCI Clusters and data centers are similar to data centers in that both provide computing capabilities for users. In addition, both are typically connected to other instances of their kind by WAN networks. **Figure B.2** illustrates this relationship between DCI Clusters and data centers to WANs. Moreover, these WAN networks are typically controlled by a different organization than those controlling the DCI Clusters or the data centers.

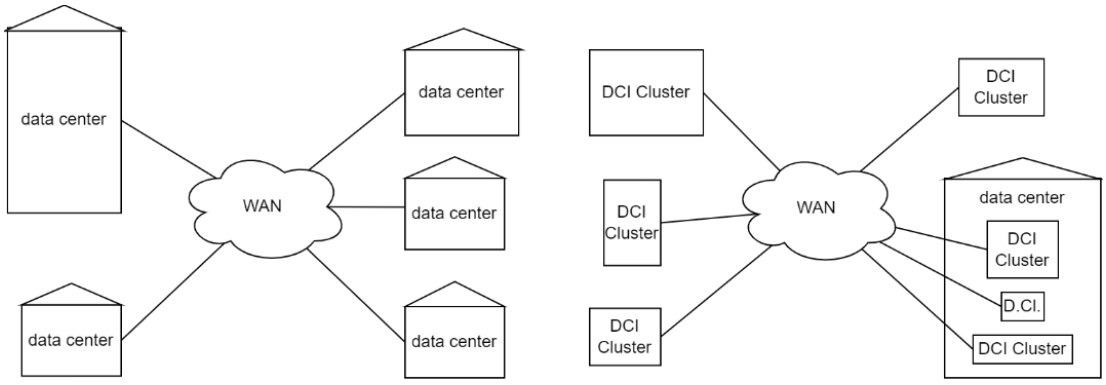

Figure B.2: Both data centers and DCI Clusters are typically connected to peers via WAN.

However, while the role of DCI Clusters in the IOWN Global Forum DCI Functional Architecture may be similar to the role of data centers in today's computing infrastructures, there is a critical difference between DCI Clusters and data centers: "data center" is a term that is typically used colloquially and that may have different meaning depending on the context the term is used in. For example, "data center" may refer to a collection of multiple buildings forming one extensive logical network with servers, a single physical building including all facilities, or only the servers inside a particular structure. In contrast, "DCI Cluster" is not a colloquial term, but a block defined within the context of the DCI

Functional Architecture. DCI Cluster blocks represent entities that provide a set of capabilities to users through defined interfaces. Furthermore, “DCI Cluster” removes implicit connotations, such as data centers typically having the size of hundreds or thousands of servers. It is due to this abstract nature that DCI Clusters do not have such connotations or limitations, nor are DCI Clusters bound to a particular physical form, such as a building.

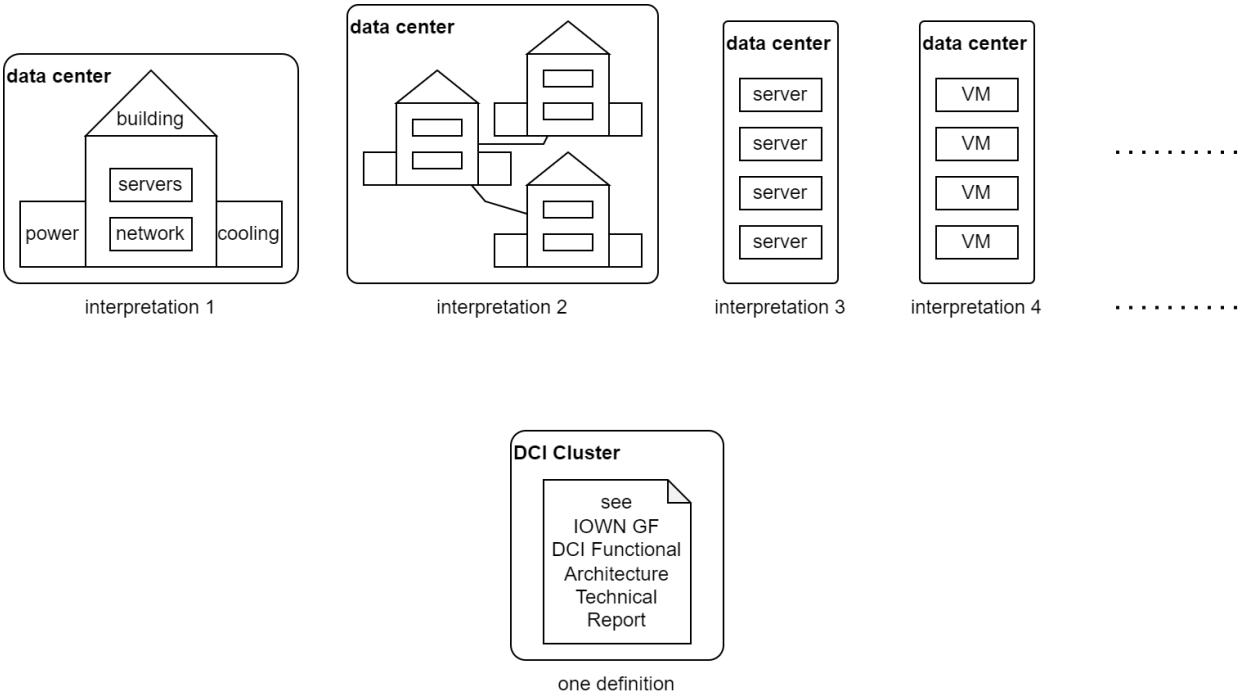

Figure B.3: “Data center” is a term used colloquially; in contrast, “DCI Cluster” is a defined block.

These fundamentally different natures of data centers and DCI Clusters are contrasted in **Figure B.3**: for example, the term “data center” may refer to a single building including servers, network, power, and cooling (interpretation 1); a group of such buildings including their utility (interpretation 2); only a group of physical servers (interpretation 3); only a collection of virtual machines (interpretation 4); and there may be even further interpretations. Instead, the term “DCI Cluster” only has a single definition as laid out in the IOWN Global Forum DCI Functional Architecture [**IOWNGF-DCIFA2**].

B.2. Structure of initial DCI Cluster RIMs

When comparing the overall physical structure of today’s data centers, as described above in **Section B.1**, and the abstract model of DCI Clusters as defined by the IOWN Global Forum DCI Functional Architecture [**IOWNGF-DCIFA2**](Sec. 7), it turns out that both have similar overall compositions. Key elements are an outside WAN, a gateway bridging the inside and outside worlds, an internal network, and computing devices.

Due to the structural similarity between today’s data centers and the abstract model of DCI Clusters, this document will start considering DCI Cluster implementations by considering the structure of today’s data centers.

B.2.1. Hardware devices in a generic DCI Cluster RIM

A key challenge of DCI Systems is introducing communication with extreme QoS requirements. At the time of this writing, today’s data centers are often implemented around a leaf-spine-type network that supports QoS management such as DiffServ [**WIKI-DIFFSERV**]. However, it is generally agreed that the construction, configuration, and management of networks supporting QoS-managed connectivity at extremely high data rates and stringent guarantees

sufficient for network-wide RDMA usage remain complex tasks; moreover, converging QoS-managed and best-effort connectivity increases complexity even further.

Therefore, to reduce complexity as much as possible, the QoS-managed and best-effort networks are assumed to be implemented by separate hardware as a starting point for considering DCI Cluster RIMs.

The extension of the generic structure of today's data centers as shown in **Figure B.1** with a separate high-QoS network to implement DCI Clusters is illustrated in **Figure B.4**. A separate set of gateway devices accompanies each network to enable connections both QoS-managed and best-effort between servers and the WAN. The *functional cards* in this figure correspond to the hardware required for computation, such as CPUs, GPUs, FPGAs, IPUs/DPUs, memory, or storage. In the long term, though, implementers may want to work toward converging best-effort and QoS-managed networks capable of catering to the demands of IOWN Global Forum use cases.

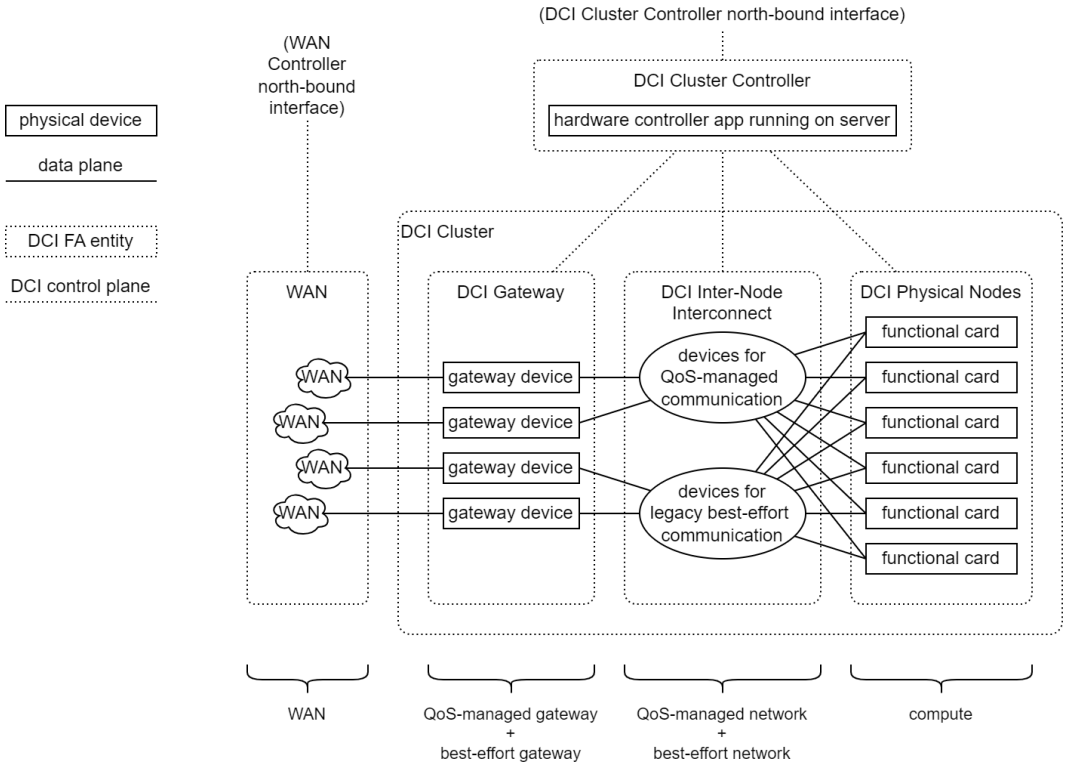

Figure B.4: Hardware components and the relation to DCI FA entities for initial DCI Cluster RIMs.

B.2.2. Grouping of individual devices into larger functional RIM blocks

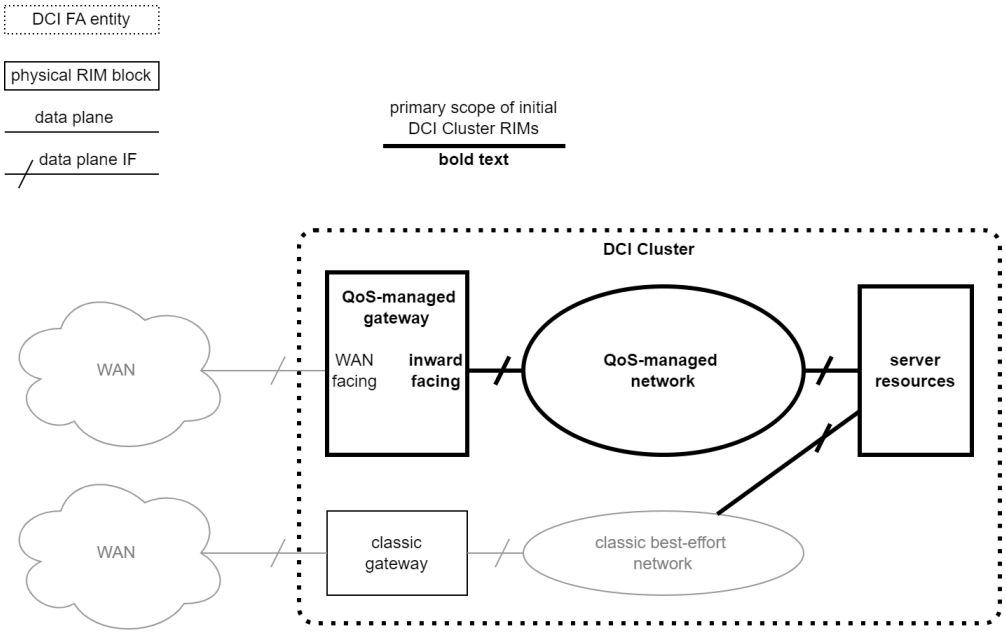

Figure B.5: Organizing physical devices into the fundamental blocks for discussing initial DCRs.

Initial DCRs having a modular structure would foster the creation of RIM variations and further advanced DCRs. Considering the concrete devices constituting the DCI Cluster as presented in **Figure B.4**, such a modular structure

can be achieved by grouping the devices into the five blocks of QoS-managed gateway, classic gateway, QoS-managed network, server resources, and classic best-effort network. The resulting structure is shown in **Figure B.5**.

This grouping aims to minimize the number and complexity of data plane interfaces between blocks while simultaneously allowing the sharing of blocks between different DCRs. For example, while a straightforward implementation of the server resources block would involve using one or more racks of COTS servers, more advanced RIMs may opt to exchange these racks of COTS servers for other technologies. However, they would keep the other parts of the RIM mainly unchanged and reuse the interfaces to the QoS-managed network and the classic best-effort network.

B.2.3. Blocks in focus for initial DCI Cluster RIMs

The DCRs in this document will focus on the server resources, QoS-managed network, and gateway blocks. There will be less focus on the WAN-facing side of the gateway since such aspects are also covered by other documents of IOWN Global Forum, such as [**IOWNGF-OPENAPN**]. Furthermore, there will also be less focus on the classic best-effort network since this type of networking is already well-understood and established in data centers.

B.2.4. Implementations are free to choose different internal structures for DCI Clusters

As a final side note for this section, while it may stand to reason to begin considerations for DCI Cluster implementations starting from a structure similar to today's data centers, having such an internal structure as illustrated in **Figure B.4** or **Figure B.5** is not a requirement to qualify as a DCI Cluster. Implementors are free to deviate from this structure and choose any arbitrary internal structure as they see fit, as long as the DCI Cluster Controller that is accompanying the DCI Cluster implements all required interfaces as prescribed by the IOWN Global Forum DCI Functional Architecture (cf. [**IOWNGF-DCIFA2**](1.3, 3.3)).

B.3. Main hardware building blocks

This section briefly examines the four generic basic building blocks for DCRs outlined in **Figure B.5** and gives examples of possible implementation technologies.

B.3.1. Server resources (DCI Physical Nodes)

Server resources blocks are responsible for executing user workload programs.

These server resource blocks contain devices such as COTS servers, composable hardware pools, accelerators, storage, and network interface cards that implement the computing capabilities of the DCI Cluster and eventually realize logical servers (DCI LSNs). The DCI Cluster Controller (DCI CC) is responsible for announcing these capabilities of the DCI Physical Nodes to a DCI Infrastructure Orchestrator (DCI IO) as well as the current status of each hardware resource, such as being available for new DCI LSNs, being used by an existing DCI LSN, being in an error state, or being reserved for a DCI LSN to be created in the future. In addition, the DCI CC is responsible for managing DCI Physical Nodes to create DCI LSNs according to the requests received from a DCI IO.

B.3.2. QoS-managed network (DCI Inter-Node Interconnect)

The QoS-managed network block connects logical servers with deterministic quality and, therefore, implements the DCI Inter-Node Interconnect. In turn, the reach of this network determines the borders of DCI Clusters.

In straightforward DCRs, this QoS-managed network is set up in parallel to a classic best-effort IP-based network to keep technological complexity as low as possible. Further advanced DCI Cluster implementations may converge this QoS-managed network with a best-effort IP-based network into a single resource-sharing network.

B.3.3. QoS-managed gateway (part of the DCI Gateway for QoS-managed connections)

The QoS-managed gateway block connects the DCI Inter-Node Interconnect to external WANs. Typical implementations will use router devices equipped with transceivers that can connect to these WANs.

Other IOWN Global Forum documents also address the realization of gateway connectivity to WANs. Therefore, this document will focus on the inward-facing details of QoS-managed gateway blocks.

B.3.4. Classic best-effort network

The classic best-effort network block corresponds to a regular IP-based network for internet connectivity. Implementations may be separate or converged with the QoS-managed network.

Today, these networks are well-understood and regularly deployed; therefore, the internal structure and implementation are outside the scope of this document.

B.3.5. Classic gateway

The classic gateway block interfaces the classic network with a WAN network. Like the classic best-effort network described in **Section B.3.4** above, such gateways are well understood and deployed today. Therefore, their internal structure is outside the scope of this document.

B.4. Interfaces between structural elements

This section briefly visits possible interfaces between the generic building blocks of DCRs as marked in **Figure B.5**.

As discussed above, these interfaces are reasonable choices for initial DCRs to ease the process of customizing these generic RIMs into more specific RIMs for particular use cases or RIMs that partly use advanced techniques. These interfaces are introduced to simplify the discussion in this document but are not mandatory for implementations.

Finally, the DCRs presented later in this document sometimes need to deviate from these interfaces.

B.4.1. Server resources ↔ QoS-managed Network

For initial DCRs, the most straightforward choice for the server resources blocks and the QoS-managed network block is an arbitrary number of fiber lines and assuming a network with hidden internal structure and equal ports. Additionally, to keep the required reasoning about the resulting DCI Cluster properties to a minimum, defining the border of the QoS-managed network to be directly before the devices implementing the QoS-managed network has the following advantages:

- The QoS-managed network can be considered separately from the internal structure of server resource blocks. For example, individual servers, groups of servers connected to top-of-rack switches, and even composable disaggregated infrastructure hardware pools can all be attached via the same interface to the same QoS-managed network implementation.
- Not including networking infrastructure typical only to server resources block hardware such as top-of-rack switches has the benefit that gateway blocks and server resources blocks share the same interface toward the QoS-managed network. This allows one to derive a network with an arbitrary ratio of gateway and server resources block ports from a single template for a QoS-managed network.

The resulting relation of the QoS-managed network interfaces with gateway and server resources blocks is further illustrated in **Figure B.6**.

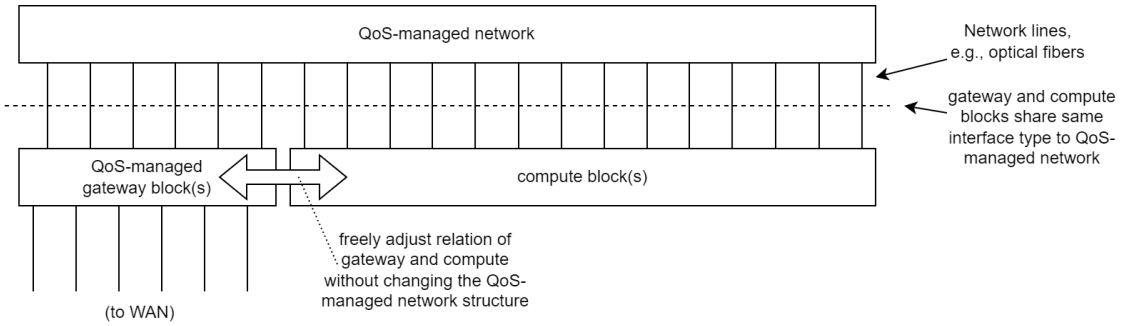

Figure B.6: Flexible balancing of gateway and server resources blocks.

Furthermore, in the most straightforward case, the ports of the QoS-managed network are connected to each other in pairs in a point-to-point fashion. In this case, the interface is reduced to the Layer 1 media and, optionally, depending on hardware, partially to the Layer 2 frame format, if any.

B.4.2. Server resources ↔ Classic best-effort network

For initial DCRs, in the most straightforward case, the classic and QoS-managed networks do not converge. Since classic best-effort networks are widespread and well-understood, these classic networks are not further treated in this document.

B.4.3. QoS-managed Network ↔ QoS-managed Gateway

Analogous to the above, the technologically least complex choice for the interface between QoS-managed gateway and network blocks is to choose the same interface as the interface between QoS-managed network and server resources blocks as outlined in **Section B.4.1** and **Figure B.6**. This allows to adjust the ratio of network links to gateway and server resources blocks without changing the network implementation. Furthermore, such a choice avoids the need for conversion inside the network block, further reducing the complexity of these initial DCRs.

B.4.4. QoS-managed Gateway ↔ WAN

The interface of the QoS-managed gateway block and the WAN and closely related implementation aspects are illustrated in **Figure B.7**.

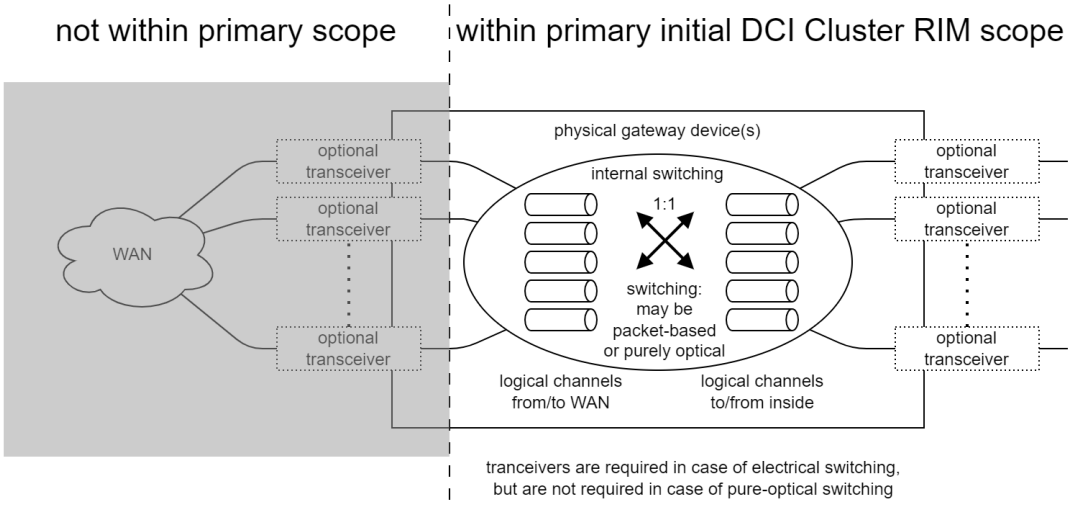

Figure B.7: Primary scope of initial DCRs regarding the gateway implementation.

First, the interface of the QoS-managed gateway blocks to WAN networks is outside the primary scope of initial DCRs and is left for detailed description by future IOWN Global Forum publications. For now, a straightforward choice to minimize gateway complexity would be to assume that the interface between the gateway and WAN is equal to the interface between the gateway and (internal) QoS-managed network.

However, in the context of initial DCRs, the proposal assumes that the interface is chosen so that logical channels on the WAN side can be mapped 1:1 to logical channels on the LAN side and then forwarded as a whole to the corresponding server resources block via the QoS-managed network. This includes multiple logical channels on the WAN side possibly being mapped onto the same physical link toward the WAN.

Furthermore, no assumption has been made about the flexibility of the switching inside the gateway block. In the context of initial DCRs,

if the QoS-managed network is implemented by dedicated separate devices that realize switching functionality at least on a per-port level, then a fixed 1:1 correspondence of WAN channels and LAN channels appears to be sufficient, and

if there is no dedicated network or only a network with limited switching flexibility, initial DCRs should consider integrating switching functionality into the gateway device.

B.5. Application of this framework to DCI Cluster RIM templates

Appendix C describes three straightforward DCR templates of diverse sizes. These RIM templates are structured along the framework as introduced above as far as deemed sensible to form a base that allows partial reuse of these templates for the more concrete DCRs presented in **Section 4**.

Appendix C. Supplementary Template DCI Cluster RIMs

Designing a Data-Centric Infrastructure (DCI) Cluster RIMs is an enormous effort. To lower the hurdle for creating new DCR designs, it seems desirable for designers to have modular templates of DCRs available that can be used as a reference starting point for designs.

This appendix provides three templates for DCI Clusters to start designs from:

- a single server directly connected to a WAN termination point (“one-box”),
- a single QoS-managed switch connecting devices within a rack (“one-rack”), and
- a fabric of QoS-managed switches connecting a large number of devices (“one-room”).

Each of these templates is simple enough to be implemented from common hardware components. However, these template designs lack the flexibility and advanced features to be relevant to a wider range of IOWN Global Forum use cases. Their only purpose is to serve as a design starting point for further advanced DCI Cluster RIMs. The main properties of these DCR templates are summarized in **Table C.1** below.

Table C.1: Template DCR main properties.

TEMPLATE **DCRS**	**SERVERS PER CLUSTER**	**TOTAL NUMBER WAN CONNECTIONS PER CLUSTER**	**MAXIMUM CONNECTIONS PER SERVER**
Template for one box: Single device connected directly to QoS-managed WAN (Section C.1)	1	1	1
Template for one rack: All-in-one gateway between WAN and multiple devices (Section C.2)	30	4	2
Template for one room: QoS-managed network fabric between gateways and devices (Section C.3)	120	462	7

C.1. Single device connected directly to QoS-managed WAN

This section outlines a DCR constructed around a single computing device connected virtually directly to a WAN. **Section C.1.1** outlines possible use cases for this single-device DCI Cluster. **Section C.1.2** illustrates the structure of this RIM. Finally, **Section C.1.3** complements the basic structure with an example of concrete devices to implement the structural RIM blocks and provides an overview of the resulting capabilities of such a DCI Cluster.

C.1.1. Use cases

The primary use case for a single-device DCI Cluster implementation approach is devices focusing on point-to-point high-QoS connections over a WAN.

Use case classes for which such an approach may be applicable include:

- Permanent QoS-managed connectivity over a WAN, such as IoT aggregation points with mostly fixed data rates or mobile front-haul links; more concrete examples include:
    - Local aggregation nodes for sensor data aggregation and ingestion of the IOWN Global Forum CPS Area Management Security use case (see **Section A.1.10**).
    - RU and/or vDU hosts in MFH/VRAN scenarios when no HA/LB capability is required (see **Section A.1.11**).
- Connectivity of a singular on-premises gateway or proxy toward a WAN for one or more devices beyond DCI System management and control.
- Intermittent QoS-managed connectivity over a WAN, but no other devices are nearby to share WAN connections, such as in sovereign computing or human/media-oriented applications.

Section C.1.2 outlines a straightforward implementation strategy with one or more direct server-to-WAN connections.

C.1.2. RIM structure

This section outlines how to construct a single-device DCI Cluster in a straightforward manner. The defining point of these DCRs is that they contain only a single computing device whose network interfaces are connected virtually directly to a WAN.

Figure C.1 shows such a DCI Cluster with a direct connection between WAN and server in the context of a complete DCI System: the right-hand side corresponds to a DCI Cluster with gateway, network, and server resources components corresponding to the building blocks as introduced in **Section B.2**, and in turn, the left-hand side of the figure shows key blocks that interact directly or indirectly with DCI Clusters, such as the DCI Cluster Controller. Vertically, the figure is divided into three layers corresponding to the data, control, and management planes of the overall DCI System. For further details regarding DCI Systems and the DCI Functional Architecture, the reader is referred to [**IOWNGF-DCIFA2**]. (The numbers in the figure are used to refer to its elements.)

Section C.1.2.1, **Section C.1.2.2**, and **Section C.1.2.3** further highlight data movement, control, and management, respectively.

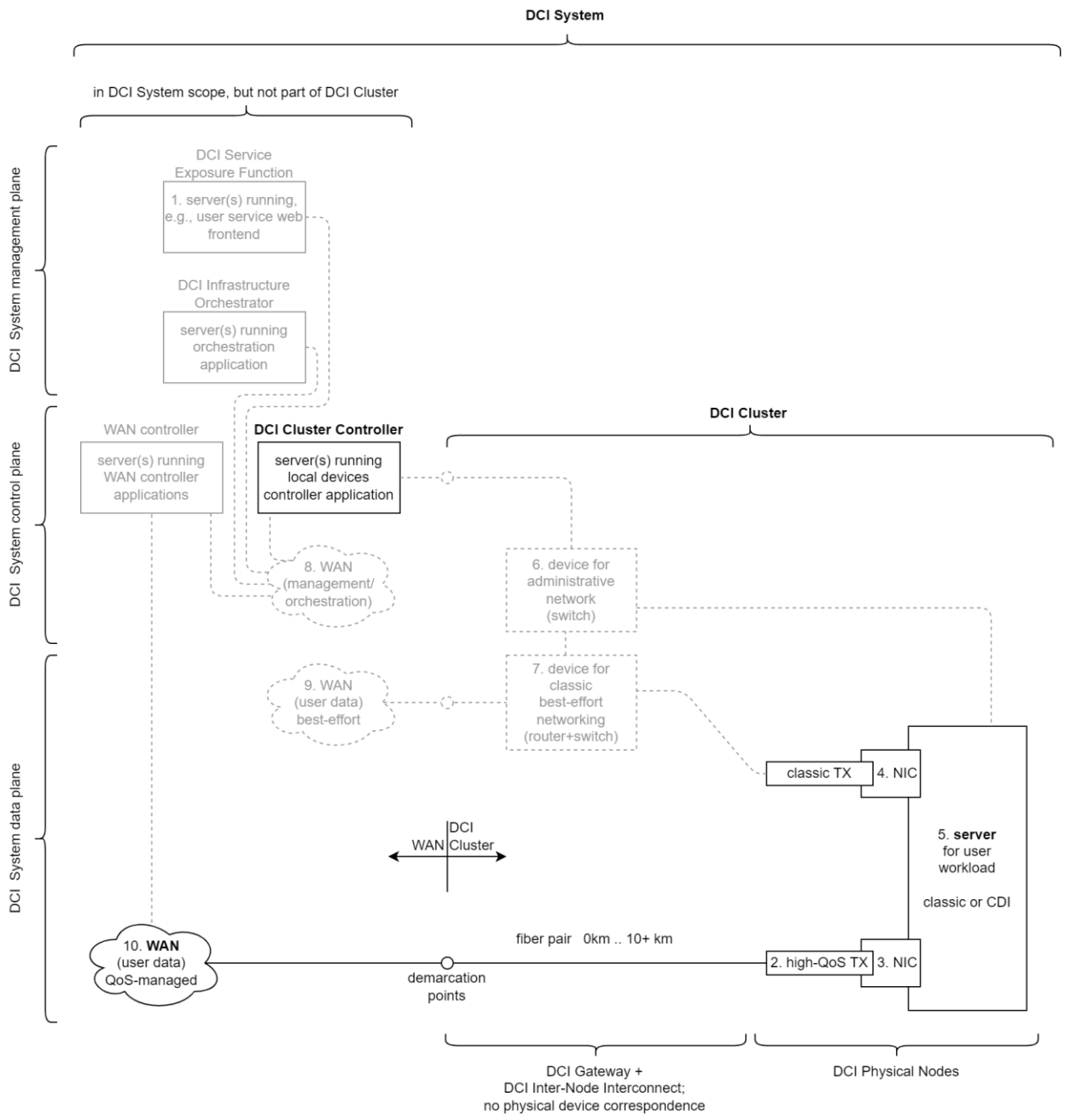

Figure C.1: DCR of a single device connected to a WAN.

C.1.2.1. Data movement: networking equipment and network interfaces

The key elements regarding the data movement of this DCR are introduced below:

- **Computing device:** To the right, a server holding computing hardware is shown. The server may have any number of network interfaces toward the WAN. The network interfaces of the servers may be located on various devices, such as NICs, DPUs/IPUs, or converged devices, such as GPUs, FPGAs, or signal processing cards with integrated network interfaces. The server could be realized by a regular commercial off-the-shelf server or by a single composable disaggregated infrastructure hardware pool.

- **NICs:** NICs installed in the server or integrated into extension cards should be able to exploit quality-managed connectivity. Therefore, NICs connected toward the WAN will likely have installed transceivers capable of high-bandwidth transmission and packet processing offloading or RDMA transmission. In addition, servers will likely also comprise NICs connected to a classic best-effort network. Furthermore, the QoS-managed WAN is shown on the left-hand side of the figure. While the design of this WAN network is not within the primary scope of the DCI Cluster RIMs, the QoS-managed WAN is shown here to illustrate how DCI Clusters are embedded in their larger environment within DCI Systems.

C.1.2.2. Control: cluster-internal automated resource configuration

The resources of the DCI Cluster are configured by a control application that realizes the DCI Cluster Controller (DCI CC). This DCI CC is architecturally not part of the DCI Cluster. The DCI CC needs to manage and track at least the allocation state (i.e., free or busy) of the resources of the DCI Cluster, for example, by keeping the cluster resource states in a database or by querying the cluster components. Furthermore, production implementations would likely monitor and report device health and failure states to the DCI IO. For one-device clusters, integrating the DCI CC tightly with the actual computing device and centrally placing the DCI CC may both be viable approaches to manage devices remotely depending on the use case.

One possible approach would be to utilize already established remote server management APIs to realize communication between the DCI Cluster Controller and the actual DCI Cluster hardware. The concrete configuration interfaces to control DCI Clusters may be freely chosen by individual implementations.

C.1.2.3. Management: interaction with higher-layer orchestrators

DCI Cluster Controllers are managed by and act only according to requests from a DCI Infrastructure Orchestrator (DCI IO). To enable interoperability of multiple types of DCI Cluster Controllers with DCI IO, DCI Cluster Controllers would support a standardized interface for higher-layer orchestration to access the DCI Cluster resources. Through the interface of the DCI CC, the DCI IO queries inventory and usage status and submits CRUD (create, read, update, delete) commands to manipulate logical servers and their network connections. In addition, the DCI IO performs the required coordination between the DCI CC and the WAN controller. However, a further description of the WAN controller being located outside the DCI Cluster is beyond the scope of this document.

In this case of a single-device DCI Cluster, there might be only one computing resource that the DCI Cluster Controller would expose to the DCI IO. In addition, the DCI CC would likely show the DCI IO a logical DCI Gateway and DCI Inter-Node Interconnect. However, in many cases, no physical device would correspond to the DCI Gateway and DCI Inter-Node Interconnect. Looking at this situation in another way, the fiber cords could also be thought of implementing these devices.

The DCI IO receives requests from the DCI Service Exposure Function (DCI SEF), which in turn directly interacts with the user. Each organization providing DCIaaS services will likely determine the individual interface between the DCI SEF and the user.

Furthermore, in the DCI Functional Architecture, DCI IO and DCI SEF are separate from and located outside DCI Clusters, just like DCI CCs and DCI IOs.

C.1.3. Example implementation choices

This section outlines one possible implementation of a DCI Cluster that is structured as illustrated in **Figure C.1**. First, concrete component classes for each part of the DCI Cluster are selected in **Section C.1.3.1**. After that, the projected capabilities of a DCI Cluster implemented with these components are listed in **Section C.1.3.2**.

C.1.3.1. Selected component classes

Table C.2 lists one possible set of example implementation choices for a concrete DCR, as illustrated in **Figure C.1**.

Table C.2: Example implementation choices for a DCR constructed around a single device.

#	DCR ELEMENT	EXAMPLE IMPLEMENTATION CHOICE
1	Servers for DCI SEF, DCI IO, DCI CC, WAN controller	classic COTS servers of arbitrary architecture connected to best-effort internet running the control and management plane software applications; on-premises or off-site
2	High-QoS transceivers in server	100 Gbps LR4
3	NICs for high-QoS	SmartNICs, DPUs/IPUs, or converged GPU/NIC devices according to user application needs
4	Classic network interfaces	10 Gbps classic network interfaces or NICs
5	Server for user workload	1 classic x86 COTS server with hardware according to user application needs, such as GPUs, DPUs/IPUs, FPGAs, eCPRI cards; 1 NIC installed for high-QoS traffic
6	Device for administrative network	4-port switch with 1 Gbps network interfaces
7	Device for classic best-effort networking	4-port router with 10 Gbps network interfaces
8	WAN for management and orchestration	best-effort internet
9	WAN for best-effort user data	best-effort internet
10	WAN for QoS-managed user data	Open APN

C.1.3.2. Resulting capabilities summary

Based upon the example implementation choices as outlined in **Section C.1.3.1**, **Table C.3** lists the projected capabilities for a DCI Cluster that is structured as illustrated in **Figure C.1**.

Table C.3: Example capabilities for a DCR constructed around a single device.

PROPERTY	PROJECTED/ESTIMATED VALUE	DETERMINED BY
Bandwidth per physical interface of NICs with high-QoS TX	100 Gbps	transceivers
High-QoS connections per physical interface of NICs with high-QoS TX	1 connection per physical network interface	NIC, transceiver, WAN
Bandwidth per connection on the physical interfaces of NICs with high-QoS TX	100 Gbps per interface / 1 connection per interface = 100 Gbps per connection (raw)	(derived)
Maximum high-QoS connections per classic server for user workload	1 network interface * 1 connection = 1 WAN connection	(derived)
Latency and jitter added inside DCI Cluster	virtually zero (only fiber delay)	fiber

Packet loss	virtually none (no active components)	fiber
Packet ordering	preserved	fiber
Non-blocking guarantees	since high-QoS network interfaces are used exclusively, WAN connections never block as long as interfaces are available	(derived)

For example, the scenario in **Table C.3** could connect edge locations to larger data centers such as in sovereign computing, VRAN, or edge data aggregation applications.

Many other scenarios seem to be possible. For example, by further varying design parameters such as the number of WAN connections and network interfaces of the edge computer.

Finally, in the minimal example above, the server is connected to the QoS-managed WAN with only a single NIC and fiber line. For use cases that require high reliability, implementors may want to consider installing two or more NICs connected to a WAN.

C.2. All-in-one gateway between WAN and multiple devices

This section outlines a template for DCRs constructed around a single non-blocking switch that pairs physical network ports of servers and WAN termination points. **Section C.2.1** outlines possible use case classes for which this simple approach may still be feasible. **Section C.2.2** illustrates the structure of this RIM. Finally, **Section C.2.3** provides examples of devices for the RIM blocks and an overview of the capabilities of the DCI Cluster.

C.2.1 Use cases

This DCI Cluster template approach connects WAN demarcation points to server ports via a single non-blocking switch in a pair-wise fashion. The advantage of inserting such a switch between WAN and servers compared to connecting all server NICs directly to a WAN is that the number of NICs or servers can exceed the number of WAN connections available. This may be desirable in use cases that require fewer concurrent WAN connections than server-side interfaces, allowing servers to exclusively use a link to the WAN on demand for a given time. For such use cases, this timesharing leads to WAN equipment savings when compared to statically connecting the maximum number of all server-side network interfaces to the WAN, each with its own WAN equipment.

Use cases for which such an approach may be applicable include

- allocating a few WAN connections dynamically for data ingestion servers that are further connected to processing and storage servers locally via other links (e.g., for CPS area management use cases, see **Section A.1.10**);
- allocating few WAN connections temporarily among many hosts connected to the non-blocking switch for rapid virtual machine migration into and out of a DCI Cluster (e.g., for financial industry services infrastructure use cases, see **Section A.1.4**);
- facilitating rapid installment and replacement of servers that require WAN connectivity by shielding the complexity of WAN equipment from servers on the other side of the switch, and
- providing rapid fail-over capability in case of server hardware malfunction by facilitating rerouting of data streams to spare servers without requiring WAN reconfiguration.

In the following, **Section C.2.2** outlines a straightforward implementation strategy utilizing a non-blocking switch that performs virtually lossless pair-wise port forwarding at wire rate together with a variable WAN port-to-server port ratio so that the resulting DCR is customizable for each of the above usage patterns.

Utilizing a port-forwarding non-blocking switch allows straightforward reasoning about the QoS guarantees that such a system can provide, which makes this approach suited for a template to start designs with. However, implementors of further advanced DCI Clusters should consider using a single packet switch configured to provide QoS guarantees not only for one but for multiple connections per port, trading off simplicity for flexibility and, therefore, increased viability.

C.2.2. RIM structure

This section outlines how a DCI Cluster up to rack-size can be constructed in a straightforward manner, resulting in a template for further DCR designs to improve. The defining point of this DCR template is to connect WAN and devices using a non-blocking switch configurable so that every port forwards and receives packets to or from at most one other port virtually without packet loss.

Figure C.2 shows a DCI Cluster with such a switch connecting WAN and servers in the context of a complete DCI System: The right-hand side of the system corresponds to a DCI Cluster with gateway, network, and server resources as introduced in **Section 3.2**, which are represented by DCI Gateway, DCI Inter-Node Interconnect, and DCI Physical Nodes, respectively. The left-hand side of the figure shows key blocks that interact directly or indirectly with DCI Clusters, such as the DCI Cluster Controller. These functional blocks of the DCI Functional Architecture are marked in **Figure C.2** with curly braces. Vertically, the figure is divided into three layers corresponding to the data, control, and management planes of the overall DCI System. For further details regarding DCI Systems and the DCI Functional Architecture, the reader is referred to [**IOWNGF-DCIFA2**]. (The numbers in the figure are used to refer to its elements.)

Section C.2.2.1, **Section C.2.2.2**, and **Section C.2.2.3** further highlight data movement, control, and management, respectively.

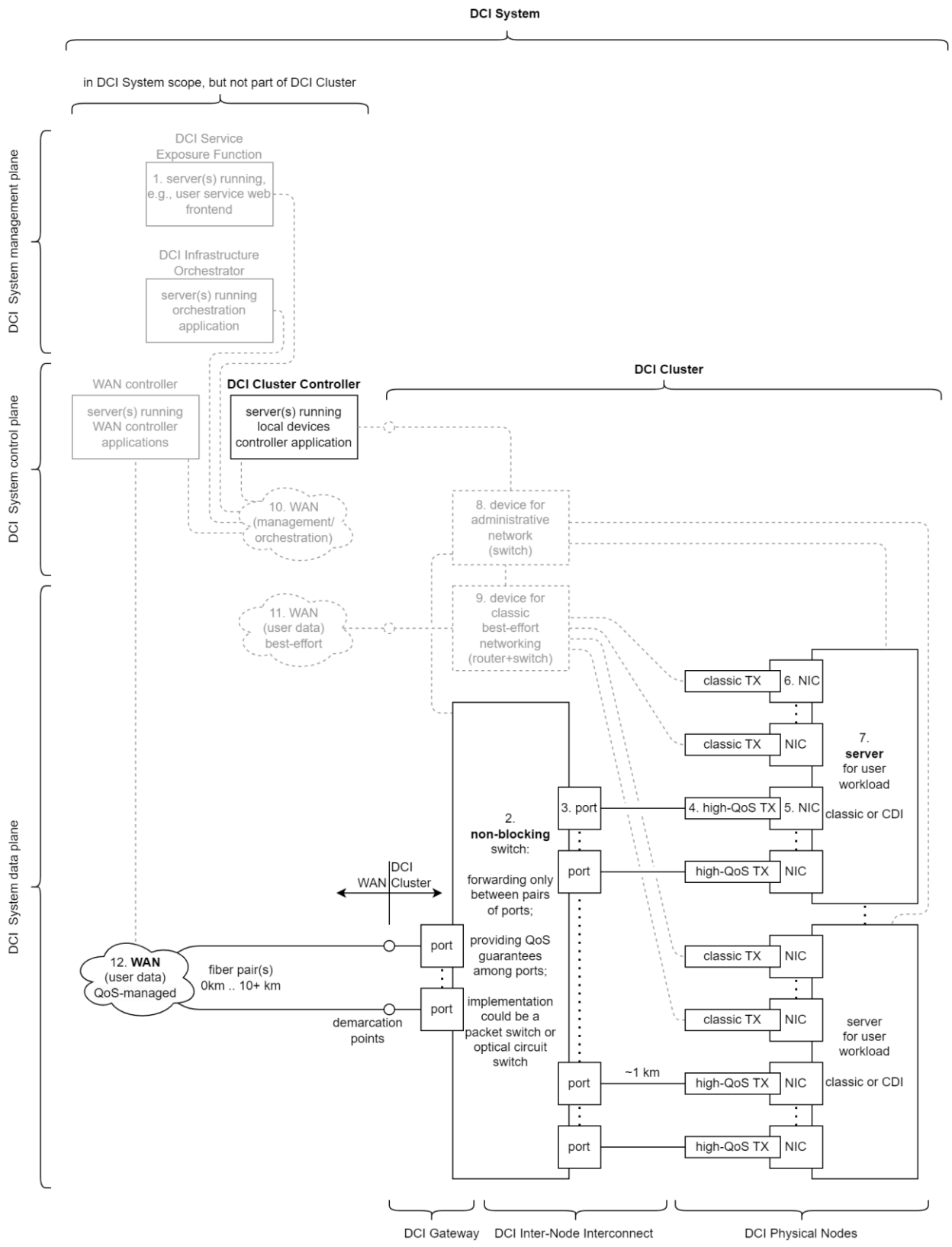

Figure C.2: DCR template of a single switch with pair-wise forwarding connecting multiple devices to a WAN.

C.2.2.1. Data movement: networking equipment and network interfaces

This section highlights the main components of this DCI Cluster RIM template that are responsible for moving data.

C.2.2.1.1. Switch between WAN and servers

The switch shall, on the one hand, be able to support the highest levels of QoS as demanded by IOWN Global Forum use cases, but on the other hand, to be part of a template needs to be straightforward to prepare and operate. Therefore, the required capabilities are chosen as follows:

- Pairing ports: connect ports into pairs and forward traffic only between the ports of one pair.
- Non-blocking: two ports can be connected if and only if both ports are not yet connected elsewhere.
- Wire rate: once two ports are paired, signals (e.g., packets or wavelengths) are forwarded at wire rate.
- Virtually lossless: there is virtually no packet loss between a configured port pair.

Such a switch allows only one connection at the network layer per NIC port but forwards data on this single connection with the highest possible QoS. This functional restriction can then be exploited by implementations. For example, such a switch could be prepared with various technologies as follows:

- COTS Ethernet packet switches: Typical packet switches can be configured for direct port forwarding between ports. If, for a given port, the packets of at most one other port are forwarded to this port, then the inbound data rate cannot exceed the outbound data rate; that is, incast congestion is avoided. Since the backplane of typical switches can serve all ports at full speed concurrently, there cannot be congestion due to interference between such port pairs. It follows that packet buffering is not required, and that packet drop inside this switch is extremely unlikely to happen even when communicating at wire rate.
- Optical circuit switches: Optical circuit switches work by physically connecting pairs of ports. Therefore, they well-suited to the described pattern when data must be relayed with the highest possible QoS. However, their generally higher cost may deter their application.

To summarize, either technology provides a mechanism to implement the QoS-managed local network part of a DCI Cluster. Both only require management and configuration to establish forwarding between pairs of ports. This is why these technologies were chosen as examples for this Template DCI Cluster RIM for implementers to start their design process with. The large restriction, however, is that only one connection per physical network interface can be made. This restriction limits the commercial viability and calls for further improvement beyond this initial template. Such improvements are discussed in **Section 4**.

C.2.2.1.2. Servers

In this template, computing capabilities are provided by servers shown to the right of the switch. Servers may take the form of regular commercial off-the-shelf servers, a composable disaggregated infrastructure, or a mixture thereof. Each server may have an arbitrary number of network interfaces connected to a QoS-managed switch, as described in **Section C.2.2.1.1**. As mentioned before, each physical interface may only have up to one connection to one other physical interface connected to the switch. Network interfaces may be located on a variety of devices, for example, NICs, DPUs/IPUs, or converged devices such as GPUs, FPGAs, or signal processing cards with integrated network interfaces.

C.2.2.1.3. NICs

NICs installed in servers or integrated into other extension cards should be able to exploit quality-managed connectivity. Therefore, NICs connected to the switch will likely have transceivers installed capable of high-bandwidth transmission and packet processing offloading or RDMA transmission. Furthermore, servers are connected to a classic best-effort network by separate NICs.

The number of NICs may be higher than the number of WAN-facing interfaces on the switch if not all NICs need to constantly connect to the WAN.

C.2.2.2. Control: cluster-internal automated resource configuration

A control application realizing a DCI Cluster Controller (DCI CC) configures the resources of the DCI Cluster. The DCI CC needs to manage and track at least the allocation state (i.e., free or busy) of the resources of the DCI Cluster. For example, it could keep the cluster resource states in a database or query cluster components. Furthermore, commercial DCI CC implementations would likely monitor and report device health and failure states to their DCI Infrastructure Orchestrator (DCI IO).

Architecturally, the DCI CC is not part of the DCI Cluster. Therefore, the server executing the DCI CC software may be physically separate from the DCI Cluster hardware. One of these servers may also execute software to control multiple DCI Clusters at once.

One possible approach to realizing communication between DCI CCs and the actual DCI Cluster hardware is to utilize management and control APIs and protocols already established in the industry. Individual implementations may freely choose the concrete interfaces between DCI CCs and their DCI Clusters.

C.2.2.3. Management: interaction with higher-layer orchestrators

DCI Cluster Controllers are managed by and act only according to requests from a single DCI Infrastructure Orchestrator. To enable interoperability of multiple types of DCI Cluster Controllers with DCI IOs, DCI Cluster Controllers need to expose a standardized interface toward higher-layer orchestration to access DCI Cluster resources such as gateway ports, network topology, and installed servers and hardware pools. Through this interface, a DCI IO queries inventory and usage status and submits CRUD (create, read, update, delete) commands to manipulate logical servers and network connections. In addition, the DCI IO performs the required coordination between the DCI CC and the WAN controller so that the DCI Gateway can utilize the WAN. A further description of the WAN controller is beyond the scope of this document because the block is located outside the DCI Cluster.

The DCI IO receives requests from the DCI Service Exposure Function (SEF). In turn, the DCI SEF directly interacts with the user. The interface between DCI SEF and the user will likely be determined individually by each organization providing DCIaaS services. Furthermore, in the DCI Functional Architecture, DCI IO and DCI SEF are separate from and located outside DCI Clusters, in the same manner as DCI CCs and DCI IOs.

C.2.3. Example implementation choices

This section outlines one possible implementation of a DCI Cluster structure, as illustrated in **Figure C.2**. First, **Section C.2.3.1**. selects concrete component classes for each part of the DCI Cluster implemented using this template without modification. Then, **Section C.2.3.2** lists the projected capabilities of such a DCI Cluster with these components.

C.2.3.1. Selected component classes

Table C.4 lists one possible set of example implementation choices for a concrete DCR, as illustrated in **Figure C.2**.

Table C.4: Example implementation choices for a DCR constructed around a single non-blocking switch.

#	DCR ELEMENT	EXAMPLE IMPLEMENTATION CHOICE
1	Servers for DCI SEF, DCI IO, DCI CC, WAN Controller	classic x86 COTS servers connected to best-effort internet running the control and management plane software applications

2	Non-blocking switch	64-port layer-2 switch configured by the DCI CC to connect pairs of ports 1:1 to avoid Ethernet packet congestion; of the switch ports, 4 ports are connected to the WAN, and 60 ports are connected to the server network interfaces
3	Switch ports	100 Gbps transceivers; depending on the structure of the WAN, transceivers able to bridge far distances may need to be installed on switch ports facing the WAN
4	High-QoS transceivers in servers	100 Gbps LR4
5	NICs with high-QoS TX	SmartNICs, DPUs/IPUs, or converged GPU/NIC devices according to user application needs
6	Classic network interfaces	100 Gbps classic network interfaces or NICs, e.g., 2 per server
7	Servers for user workload	30 classic x86 COTS servers with hardware according to user application needs, such as GPUs, DPUs/IPUs, FPGAs, eCPRI cards; each server has 2 NICs installed for high-QoS traffic
8	Device for administrative network	48-port switch with 1 Gbps network interfaces
9	Device for classic best-effort networking	64-port router with 100 Gbps network interfaces
10	WAN for management and orchestration	best-effort internet
11	WAN for best-effort user data	best-effort internet
12	WAN for QoS-managed user data	Open APN

C.2.3.2. Resulting capabilities summary

Based upon the example implementation choices as outlined in **Section C.2.3.1**, **Table C.5** lists the projected capabilities for a DCI Cluster that is structured as illustrated in **Figure C.2**.

Table C.5: Example capabilities for a DCR constructed around a single non-blocking switch.

PROPERTY	PROJECTED/ESTIMATED VALUE	DETERMINED BY
Bandwidth per physical interface of NICs with high-QoS TX	100 Gbps	transceivers
High-QoS connections per physical interface of NICs with high-QoS TX	1 connection per physical network interface	NIC, transceiver, non-blocking switch, WAN
Bandwidth per connection on the physical interfaces of NICs with high-QoS TX	100 Gbps per interface / 1 connection per interface = 100 Gbps per connection (raw)	(derived)
Maximum high-QoS connections per classic server for user workload	2 connections (network interfaces) per physical server	number of NICs

Maximum number of classic servers for user workload	60 server-facing ports on non-blocking switch / 2 network interfaces per server = 30 servers	(derived)
Ratio of classic servers to WAN connections in this example	4 WAN-facing ports on non-blocking switch / 30 servers = 7.5 physical servers per WAN connection	(derived)
Latency and jitter added inside DCI Cluster	« 1 millisecond	non-blocking switch
Packet loss	virtually none (since there is no congestion)	non-blocking switch
Packet ordering	preserved	non-blocking switch
Non-blocking guarantees	• local connections succeed always if interfaces free • WAN connections may block	(derived)

For example, the scenario in **Table C.5** could be used to realize a group of 30 virtual machine host servers that are permanently and tightly connected within the group and can temporarily be connected directly to QoS-managed WAN for far and fast VM migration into or out of the cluster.

Many other scenarios also seem possible by further varying design parameters such as the number of switch ports, WAN connections, and network interfaces per server.

C.3. QoS-managed network fabric between gateways and devices

This section outlines a DCR constructed around a fabric of switches implementing a classic circuit-switched Clos[**WIKI-CLOS**] network for packet forwarding. This fabric is used to connect computing devices to each other and to a WAN. First, possible use cases are outlined for this type of DCI Cluster in **Section C.3.1**. Then, the structure of this RIM is illustrated in **Section C.3.2**. Finally, the basic structure is complemented by an example of concrete devices to implement the structural RIM blocks and an overview of the resulting capabilities of such a DCI Cluster in **Section C.3.3**.

C.3.1. Use cases

Using a circuit-switched fabric allows connecting an even larger number of devices to a WAN than possible with only a single switch. Scaling up the number of devices enables the fabric to benefit from economies of scale, pooling compute, storage, network, and WAN resources and further increasing resource utilization.

Use cases beyond those given in **Section C.2.1** for which such an approach may be particularly applicable include:

- Disaggregating compute and storage servers at the scale of 100 or more servers.
- Sharing accelerator resources among multiple use cases in a mixed WAN-heavy/LAN-heavy workload.
- Communicating with multiple LAN or WAN peers simultaneously.

Section C.3.2 outlines a straightforward implementation strategy centered around a 1000+ port circuit-switched fabric that provides high-QoS connectivity between the WAN and servers.

C.3.2. RIM structure

This section outlines how to construct a medium-sized DCI Cluster in a straightforward manner. The defining point of this DCR is a Clos-type fabric [**WIKI-CLOS**] that provides QoS-managed connectivity to connect devices and to a WAN. The fabric is assumed to forward packets between pairs of its ports virtually without packet loss, packet reordering, or jitter.

Figure C.3 shows such a DCI Cluster with a circuit-switched fabric connecting WAN and servers in the context of a complete DCI System: the right-hand side corresponds to a DCI Cluster with gateway, network, and server resources as introduced in **Section B.2**, and in turn, the left-hand side of the figure shows key blocks that interact directly or indirectly with DCI Clusters, such as the DCI Cluster Controller. Vertically, the figure is divided into three layers corresponding to the data, control, and management planes of the overall DCI System. For further details regarding DCI Systems and the DCI Functional Architecture, the reader is referred to [**IOWNGF-DCIFA2**]. (The numbers in the figure are used to refer to its elements.)

Section C.3.2.1, Section **C.3.2.2**, and Section **C.3.2.3** highlight data movement, control, and management, respectively.

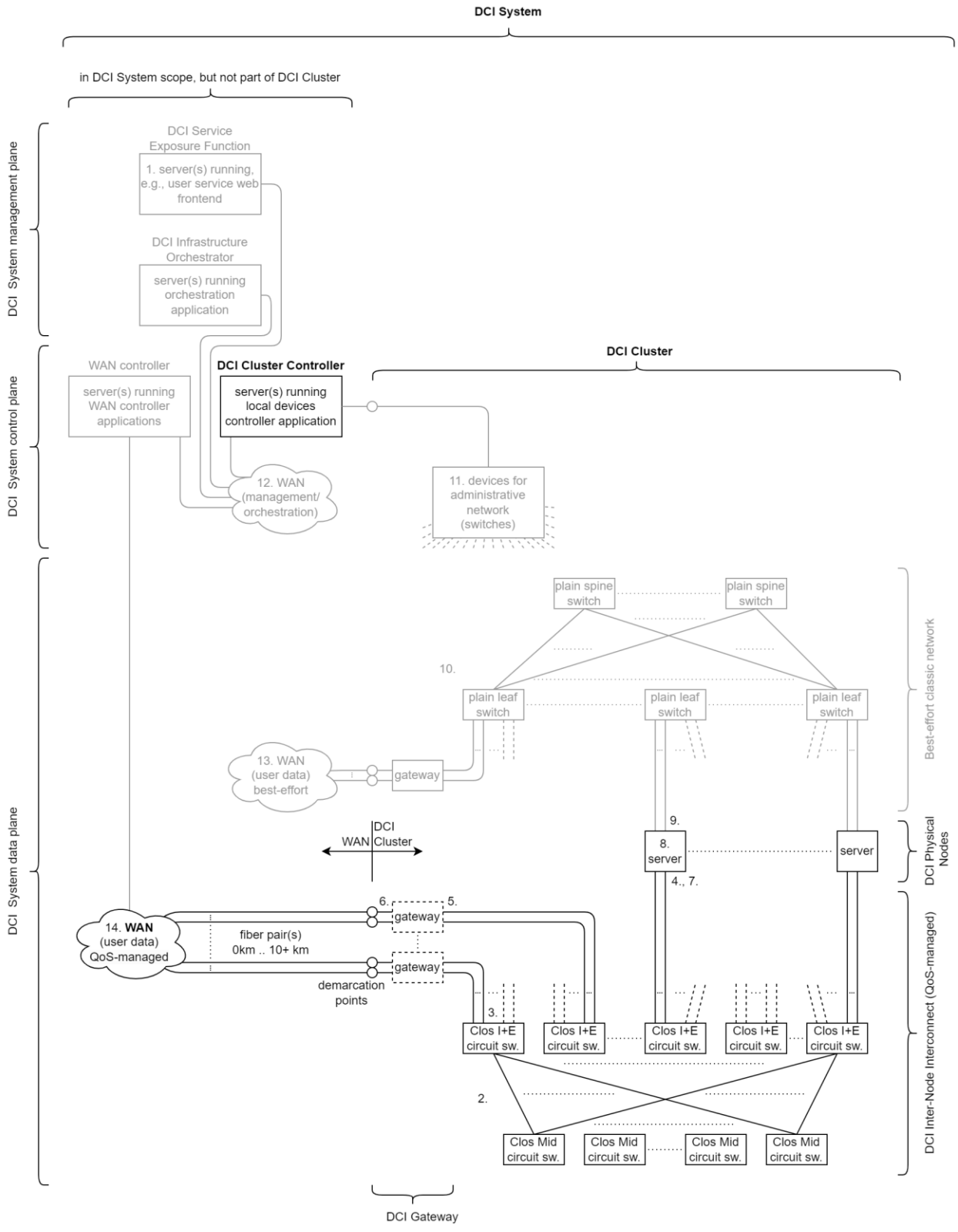

Figure C.3: DCR with a circuit-switched fabric connecting many devices and a WAN.

C.3.2.1. Data movement: networking equipment and network interfaces

This section highlights key aspects of DCI System components related to data movement inside and outside the DCI Cluster.

C.3.2.1.1. QoS-managed network: circuit-switched fabric

The distinguishing point of this DCR is a circuit-switched fabric. In this DCR, this fabric forwards information between pairs of ports, i.e., information sent into one port of the fabric will always emerge from the same paired port. This QoS-managed circuit-switched fabric that corresponds to the DCI Inter-Node Interconnect is shown at the very bottom of **Figure C.3**.

One way to construct such a circuit-switched fabric for a large number of ports (e.g., thousands) in a non-blocking fashion is to combine packet or optical switches into a classic circuit-switching Clos topology[**WIKI-CLOS**]. These switches are operated as if they were elementary crossbar switches inside a classic Clos network. For example, using such a Clos topology (not to be confused with modern “leaf-spine” topologies) with around 100 fabric switches, each having 64 ports, a larger circuit-switched fabric with more than 1000 ports can be realized. This is achieved by configuring 1:1 port forwarding appropriately on the individual switches. Classic Clos topologies are composed of three stages: ingress, middle, and egress. Here, the egress stage is folded back onto the ingress stage (I+E).

Individual fabric switches could be implemented using a variety of technologies, such as regular electronic Layer 2 switches, electronic Layer 1 switches, or purely optical switches. Product designs should weigh each technology’s unique advantages in combination with other higher-level structural optimizations. While a detailed discussion is out of the scope of this section, even when choosing electronic Layer 2 switching as base technology, this circuit-switched fabric is expected to experience virtually no packet loss, reordering, or jitter. This is because, within the fabric, there will be virtually no packet contention. Each input port is matched with exactly one output port of equal speed, making queuing highly unlikely.

Even with COTS networking products, fabrics that provide extremely high quality-of-service among 1000 ports should be realizable by Clos networks in a straightforward fashion. Still, products are expected to improve significantly on such a template implementation in terms of functionality and efficiency.

C.3.2.1.2. Servers and NICs

Above the circuit-switched fabric, servers holding computing hardware are shown. Servers may have any number of network interfaces toward the circuit-switched fabric. The network interfaces of the servers may be located on various devices, such as NICs, DPUs/IPUs, or converged devices, such as GPUs, FPGAs, or signal processing cards with integrated network interfaces. Servers may be realized by regular commercial off-the-shelf servers, a composable disaggregated infrastructure, or a mixture thereof.

In this initial basic DCR, the circuit-switched fabric operates on the port level. Following that, each network interface port of a server can only be connected to at most one other port on the circuit-switched fabric simultaneously.

C.3.2.1.3. Gateways

Depending on how the WAN is terminated locally, dedicated physical gateway devices may be introduced even for a Template DCR.

No dedicated blocks are required if the WAN line service is terminated using technology compatible with other technology inside data centers. For example, if the WAN is terminated using LR4, LR4 transceivers could be installed in fabric switches. Alternatively, if optical circuit switches are used to implement the circuit-switched fabric, then servers could be directly equipped with LR4 modules. In both cases, no further gateway equipment is required.

In contrast, if the WAN is to be terminated within the DCI Cluster’s responsibility, specialized gateway devices may be required to translate and encapsulate/decapsulate signals and packets between the WAN media and the local circuit-switched fabric.

To illustrate this DCR further, it could be assumed that the WAN is terminated using LR4 signals. In addition, OTN packet framing, coherent transmission, DWDM channel selection, and laser tuning would be within the WAN's responsibility and of no concern for to DCI Cluster.

Suppose photonic signals from the WAN need to be carried as-is to server NICs. In that case, one possibility is to compose the circuit-switched fabric out of purely optical switches and let gateways degenerate into fiber lines or employ purely optical switches.

C.3.2.1.4. QoS-managed WAN

The WAN is assumed to provide connectivity with quality-of-service guarantees.

Any number of fabric ports may connect to the WAN. For example, the number of fabric ports facing the WAN could be equal to the number of fabric ports facing servers, or fewer ports facing the WAN and more fabric ports facing servers, in case not all servers need to communicate over the WAN at the same time.

However, depending on how the WAN is terminated and how flexible routing inside the WAN is, connecting servers directly to the WAN without an intermediate fabric may be more efficient in case of very large amounts of WAN connections.

While the design of this WAN network is not within the primary scope of the DCI Cluster RIMs, the QoS-managed WAN is shown here to illustrate how DCI Clusters are embedded in their larger environment within DCI Systems.

C.3.2.1.5. Best-effort network

In this RIM, the best-effort classic network is assumed to be implemented separately from the QoS-managed network. Best-effort networking is a well-established and understood technology, so it is not further highlighted in this document.

C.3.2.2. Control: cluster-internal automated resource configuration

The resources of the DCI Cluster are configured by a control application that realizes the DCI Cluster Controller (DCI CC). The DCI CC needs to manage and track at least the allocation state (i.e., free or busy) of the resources of the DCI Cluster, for example by keeping the cluster resource states in a database or by querying the cluster components. Furthermore, production implementations would likely monitor and report device health and failure states to the DCI IO.

The DCI CC is architecturally not part of the DCI Cluster, and the server executing the cluster control software may also be physically separate from the DCI Cluster hardware. One of these servers may also execute software to control multiple DCI Clusters at once.

To realize communication between DCI Cluster Controller and the actual DCI Cluster hardware, one possible approach would be to utilize already established APIs. The concrete configuration interfaces to control DCI Clusters may be freely chosen by individual implementations.

In addition, the DCI CC configures the circuit-switched fabric to establish the required low-level connectivity among the WAN demarcation points/gateways and server network interfaces.

C.3.2.3. Management: interaction with higher-layer orchestrators

DCI Cluster Controllers are managed by and act only according to requests from the DCI Infrastructure Orchestrator. To enable interoperability of multiple types of DCI Cluster Controllers with DCI IO, DCI Cluster Controllers would support a standardized interface for higher-layer orchestration to access the DCI Cluster resources. Through the interface of the DCI CC, the DCI IO queries inventory and usage status and submits CRUD commands to manipulate logical servers and their network connections.

In this case of a DCI Cluster with a medium-sized circuit-switched fabric, the DCI Cluster Controller would expose its inventory to a DCI IO according to an abstracted model. Such a model would include information such as COTS server availability, CDI pool status, and further information relevant to WAN routing, such as DCI Gateway port status. However, to make a final decision on whether a request could be completed, in the case of this RIM, DCI IOs would likely need to submit a concrete request for evaluation by the DCI CC. The reason is that only the DCI CC would have sufficient knowledge of internal and highly implementation-specific details such as communication topologies and hardware failure states.

In addition, the DCI IO coordinates the DCI CC and the WAN controller so that the DCI Gateway can utilize the WAN. However, this document does not cover the WAN controller's location outside the DCI Cluster.

The DCI IO receives requests from the DCI Service Exposure function which in turn directly interacts with the user. Each organization providing DCIaaS services will likely determine the interface between the two independently.

Furthermore, in the DCI Functional Architecture, DCI IO and DCI SEF are separate from and located outside DCI Clusters, just like DCI CCs and DCI IOs.

C.3.3. Example implementation choices

This section outlines one possible implementation of a DCI Cluster that is structured as illustrated in **Figure C.3**. First, concrete component classes for each part of the DCI Cluster are selected in **Section C.3.3.1**. After that, the projected capabilities of a DCI Cluster implemented with these components are listed in **Section C.3.3.2**.

C.3.3.1. Selected component classes

Table C.6 lists the implementation choices for a high-QoS fabric that handles network connections only on the port level.

Table C.6: Example implementation choices for a DCR constructed around a circuit-switched fabric.

#	DCR ELEMENT	EXAMPLE IMPLEMENTATION CHOICE
1	Servers for DCI SEF, DCI IO, DCI CC, WAN Controller	classic x86 COTS servers connected to best-effort internet running the control and management plane software applications
2	Clos-type circuit-switched fabric for high-QoS	1302 port non-blocking Clos-type circuit-switched fabric [**WIKI-CLOS**]; built from 103 packet switches with 64 ports each (optical or electronical)
3	High-QoS transceivers in circuit-switched fabric	either 100 Gbps SR4 or none in case of optical switching
4	High-QoS transceivers in servers	100 Gbps SR4 (or LR4 in case of optical switching)
5	High-QoS transceivers in gateways facing fabric (if dedicated gateway devices are used)	100 Gbps SR4
6	High-QoS transceivers in gateways facing WAN (if dedicated gateway devices are used)	100 Gbps LR4
7	NICs for high-QoS	SmartNICs, DPUs/IPUs, or converged GPU/NIC devices according to user application needs

8	Servers for user workload	classic x86 COTS servers with hardware according to user application needs, such as GPUs, DPUs/IPUs, FPGAs, eCPRI cards; each server has network interfaces for high-QoS traffic, either on NICs or on accelerators
9	Classic best-effort network interfaces	100 Gbps classic network interfaces or NICs
10	Devices for classic best-effort networking	(omitted; classic best-effort network)
11	Devices for administrative network	(omitted; classic administrative network)
12	WAN for management and orchestration	best-effort internet
13	WAN for best-effort user data	best-effort internet
14	WAN for QoS-managed user data	Open APN

The list of component classes above is only one example choice. For example, alternative choices include:

- using only optical switches instead of electrical switches for crossbar and gateway for maximum QoS guarantees and for allow routing of the analog optical signal from WAN directly to a server;
- using optical switches only on the Clos middle layer to share crossbar ports among servers/network interfaces connected to the same I+E layer electronic switch;
- converging best-effort and high-QoS networks; and
- equipping gateway devices with coherent detection transceivers to directly connect to WANs.

While surely many more viable variations exist, such advanced approaches are outside the scope of this section's Template DCR. Actual product designs should consider such variations for further efficiency and performance improvements.

C.3.3.2. Resulting capabilities summary

Based upon the example implementation choices as outlined in **Section C.3.3.1**, **Table C.7** lists the projected capabilities for a DCI Cluster that is structured as illustrated in **Figure C.3**.

Table C.7: Example capabilities for a DCR constructed around a circuit-switched fabric.

PROPERTY	PROJECTED/ESTIMATED VALUE	DETERMINED BY
Bandwidth per connection on the physical interfaces of NICs with high-QoS TX	100 Gbps	transceivers, fabric
Bandwidth per connection on the physical interfaces of NICs with high-QoS TX	1 connection per physical network interface	NIC, transceiver, fabric, WAN
Bandwidth per connection on the physical interfaces of NICs with high-QoS TX	100 Gbps per interface / 1 connection per interface = 100 Gbps per connection (raw)	(derived)
Fabric ports facing gateways	462 ports	(arbitrary)

Fabric ports facing servers	840 ports (462+840=1302 circuit-switched fabric ports)	(arbitrary)
Maximum high-QoS connections per classic server for user workload	7 connections (network interfaces) per physical server (coincidentally, the I+E switches of the circuit-switched fabric as described in **Table C.6** would each have 21 ports facing the workload, able to support exactly 3 of such server)	number of NICs
Maximum number of classic servers for user workload	840 server-facing ports / 7 network interfaces per server = 120 servers	(derived)
Ratio of classic servers to WAN connections in this example	462 WAN-facing ports on circuit-switched fabric / 120 servers = 3.85 WAN connections per physical server	(derived)
Latency and jitter added inside DCI Cluster	« 1 millisecond	fabric
Packet loss	virtually none (since there is no congestion)	fabric
Packet ordering	preserved	fabric
Non-blocking guarantees	• local connections succeed if free interfaces • WAN connections may block	(derived)

The capabilities listed above can be varied significantly in a straightforward way, such as by modifying basic design parameters. For example, such design choices include:

- the relation of fabric ports toward WAN and servers can be varied just by connecting the desired number of devices to the circuit-switched fabric (as long as the total number of ports on the circuit-switched fabric is not exceeded);
- the number of interfaces per server connected to the circuit-switched fabric can be varied just by connecting more interfaces per server to the fabric (again, as long as the total number of ports on the fabric is not exceeded), and
- the total number of servers connectable to the crossbar can be significantly increased by replacing or cascading I+E fabric switches with electronic packet switches that multiplex groups of servers to ports of the fabric, trading off perfect non-blocking capability and regularity of the fabric for increased scale and reduced cost.

DCI Cluster product designs should consider such design choices if a DCI Cluster implementation is to be optimized for a particular application or use case.

References

REFERENCE	DESCRIPTION
IOWNGF-DCIFA2	IOWN Global Forum, “Data-centric infrastructure Functional Architecture,” version 2.0, 2023. https://iowngf.org/wp-content/uploads/2023/04/IOWN-GF-RD-DCI_Functional_Architecture-2.0.pdf
IOWNGF-DCIPCP	IOWN Global Forum, “DCI Product Concept Paper,” version 1.1, 2023. https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-DCI_PCP-1.1.pdf
IOWNGF-OPENAPN	IOWN Global Forum, “Open All-Photonic Network Functional Architecture,” version 2.0, 2023. https://iowngf.org/wp-content/uploads/formidable/21/IOWN-GF-RD-Open_APN_Functional_Architecture-2.0.pdf
UECWP	Ultra Ethernet Consortium, “Overview of and Motivation for the Forthcoming Ultra Ethernet Consortium Specification,” 2023. https://ultraethernet.org/wp-content/uploads/sites/20/2023/10/23.07.12-UEC-1.0-Overview-FINAL-WITH-LOGO.pdf
WIKI-CLOS	Wikipedia, “Clos network,” 2024. https://en.wikipedia.org/wiki/Clos_network
WIKI-DIFFSERV	Wikipedia, “Differentiated services,” 2024. https://en.wikipedia.org/wiki/Differentiated_services

History

REVISION	RELEASE DATE	SUMMARY OF CHANGES
1	March 2025	Initial Release