CSN08601 – Computer Systems

Modern I/O

Modern I/O

Today's Topics

Main:

* PCIe
* SATA
* USB

Extras:

* Layered technologies
* Hardware pictures
* Basic I/O systems

Layered Technologies

Layered Technologies

* Modern bus interfaces use “layers”
* This is borrowed from network technology
  * Namely, from the The Open System Interconnection (OSI) model
* A similar approach was later taken for bus systems

Layered functionality

* Segregate functionality
* Are ordered in a vertical way
* Each layer communicates with the one below and the one above it only
* And does not need to know anything else

Note: Also known as “abstraction layers”

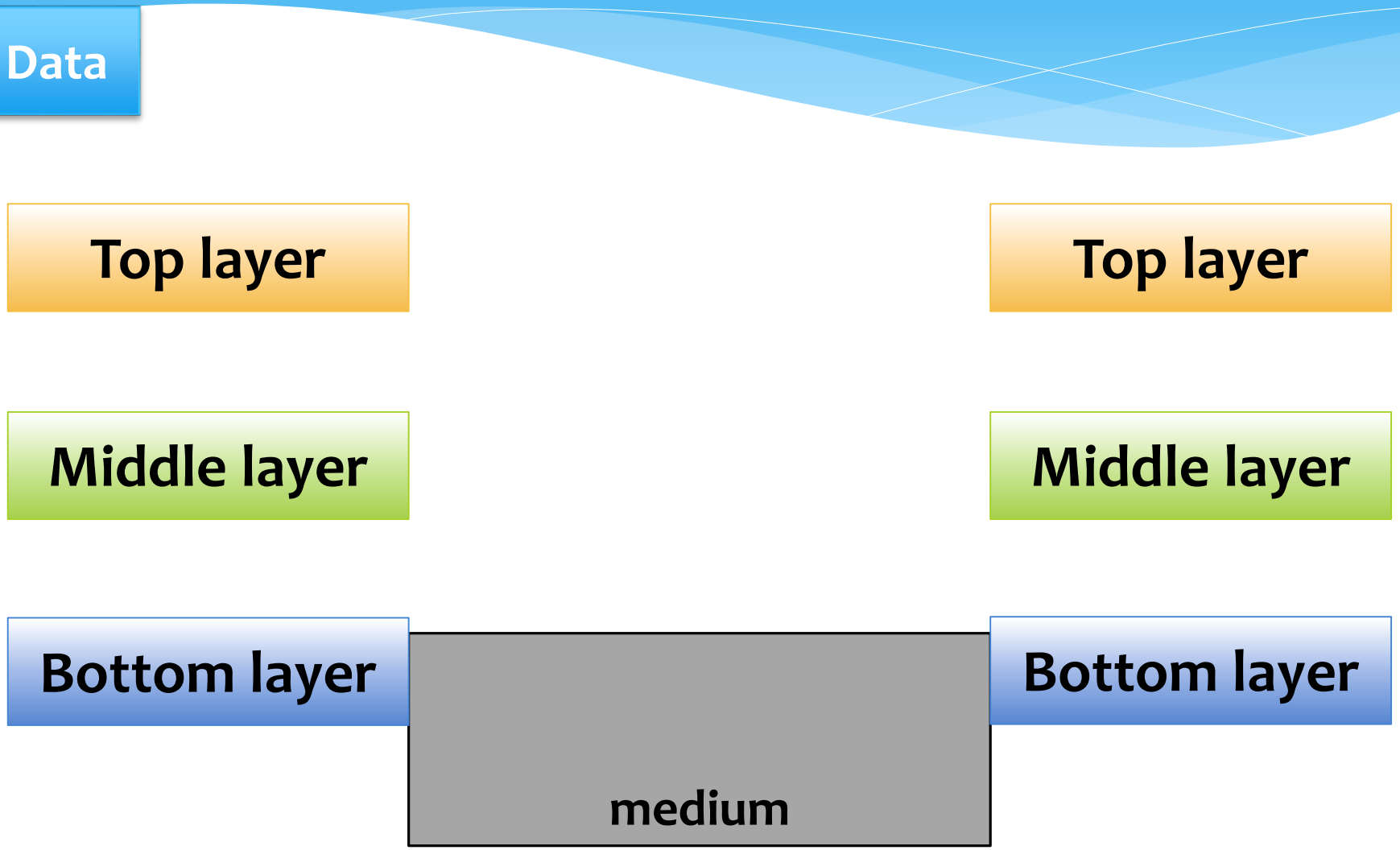
Advantages of layers

* The internal workings of each layer can be developed independently, as long as the interfaces with higher/lower layers remain consistent
* Compatibility with legacy hardware and software can be implemented in the top layer
  * Compatibility decisions don't have to hinder the lower layers

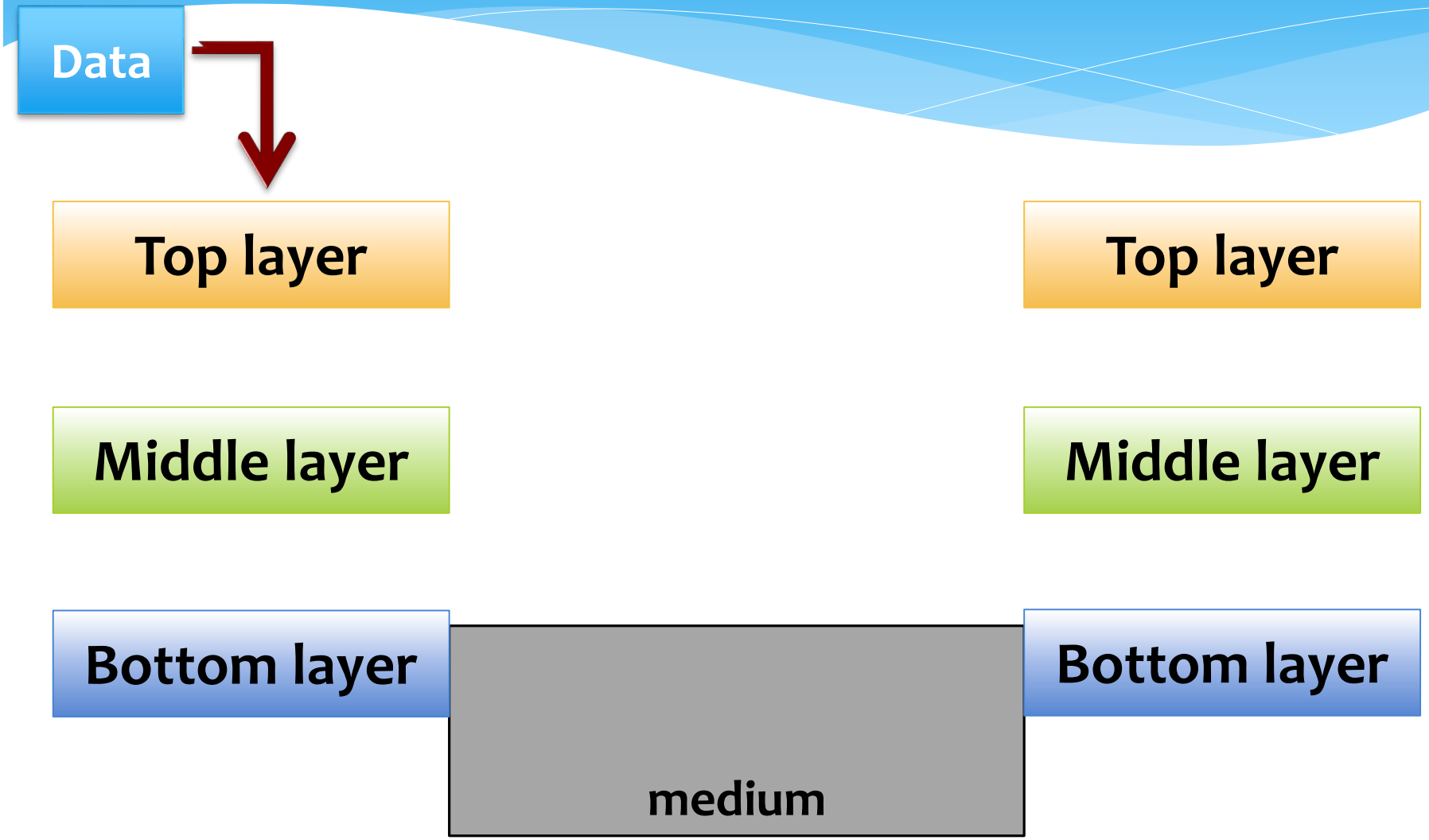
Layers Example

Note: this is totally made-up

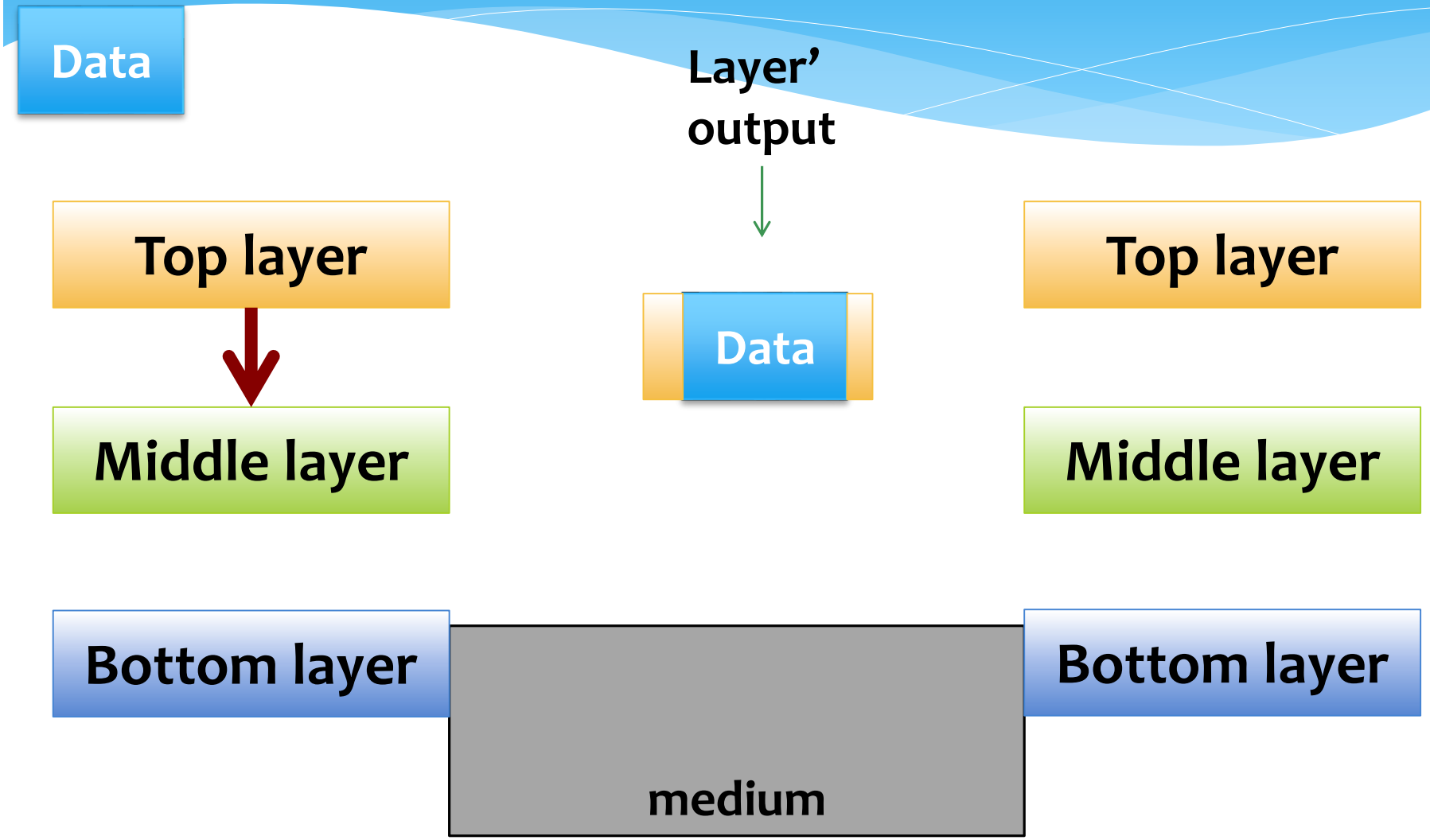
Layers:
A mock example



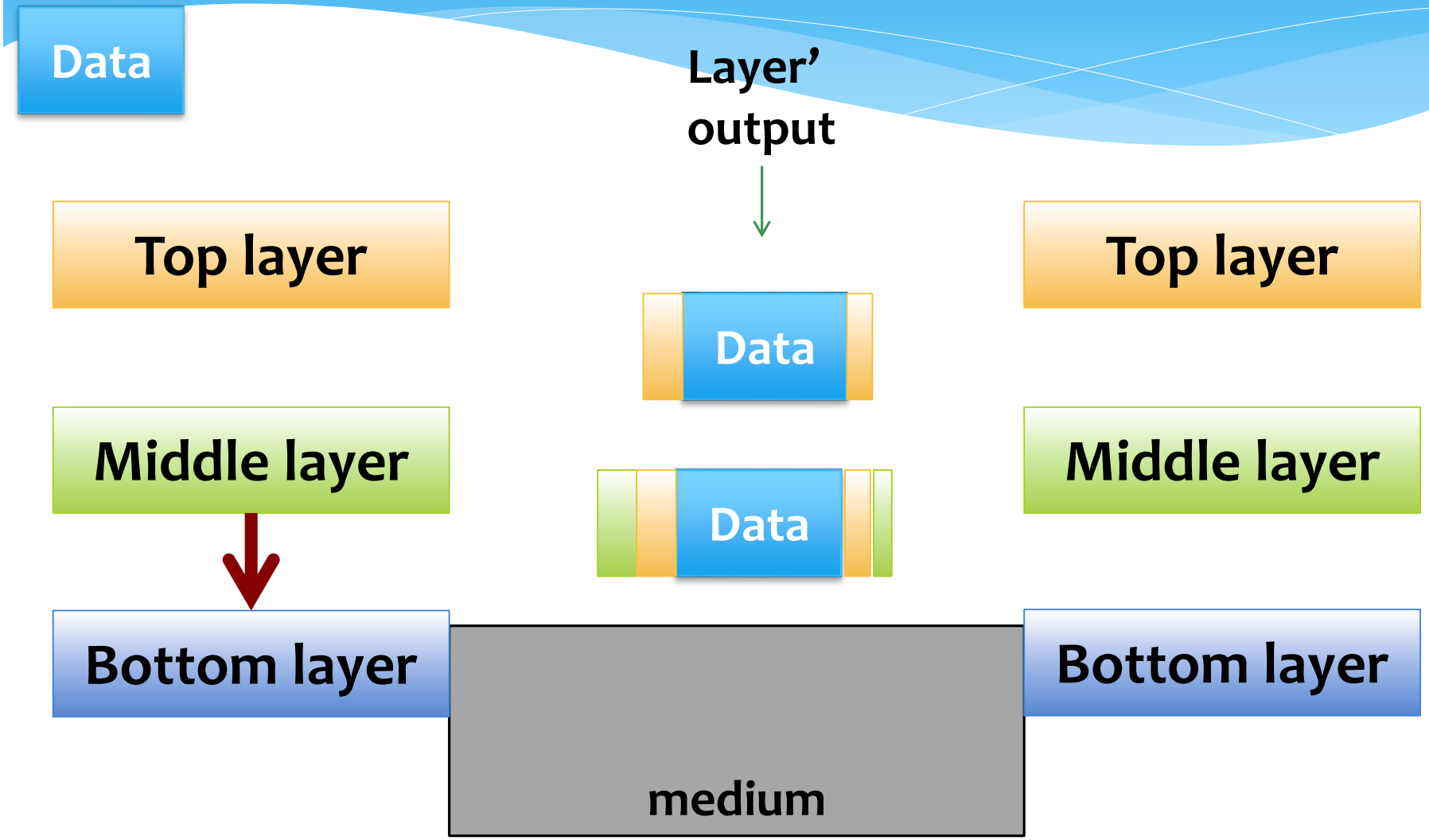
Layers:
A mock example



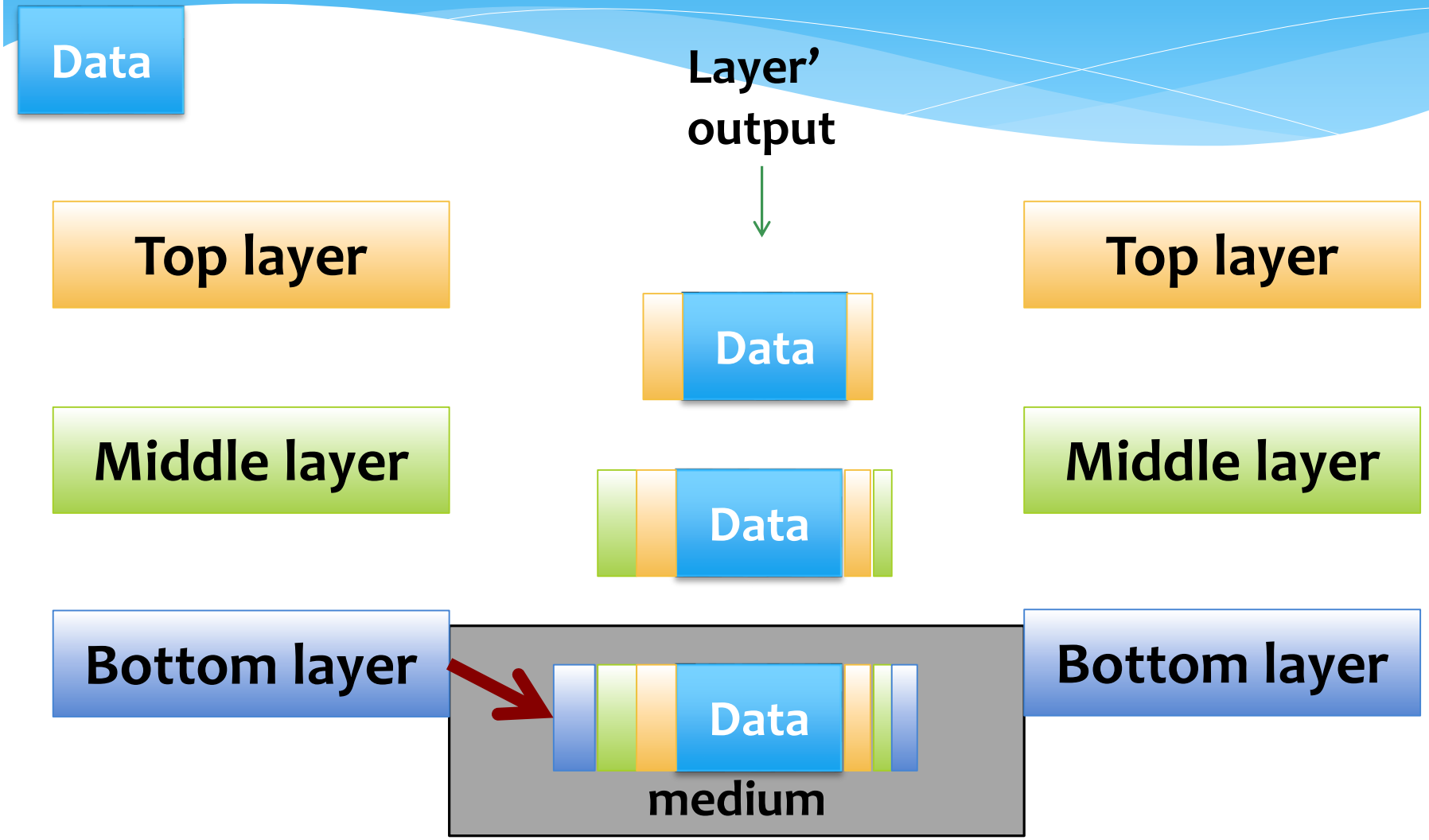
Layers:
A mock example



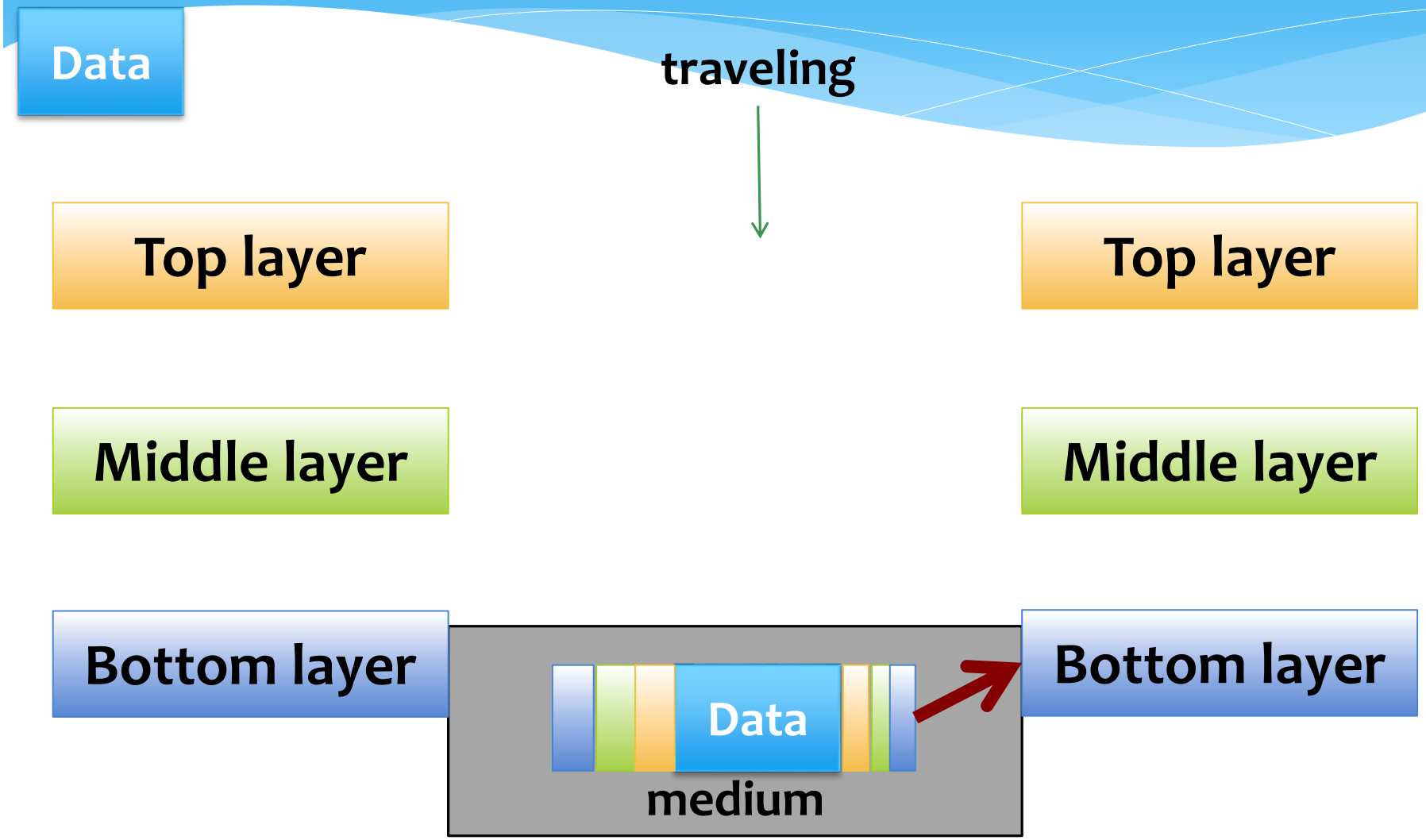
Layers:
A mock example



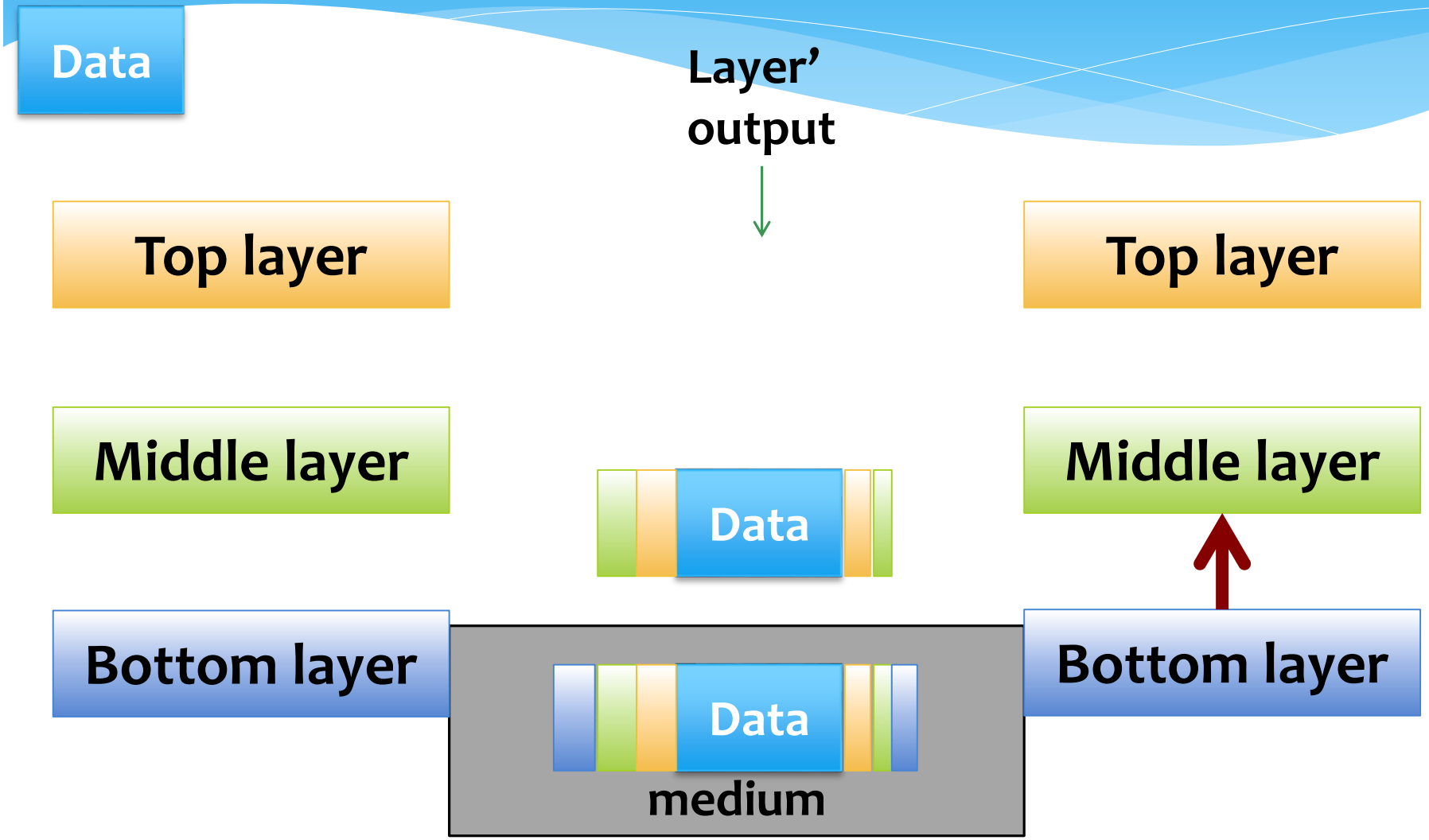
Layers:
A mock example



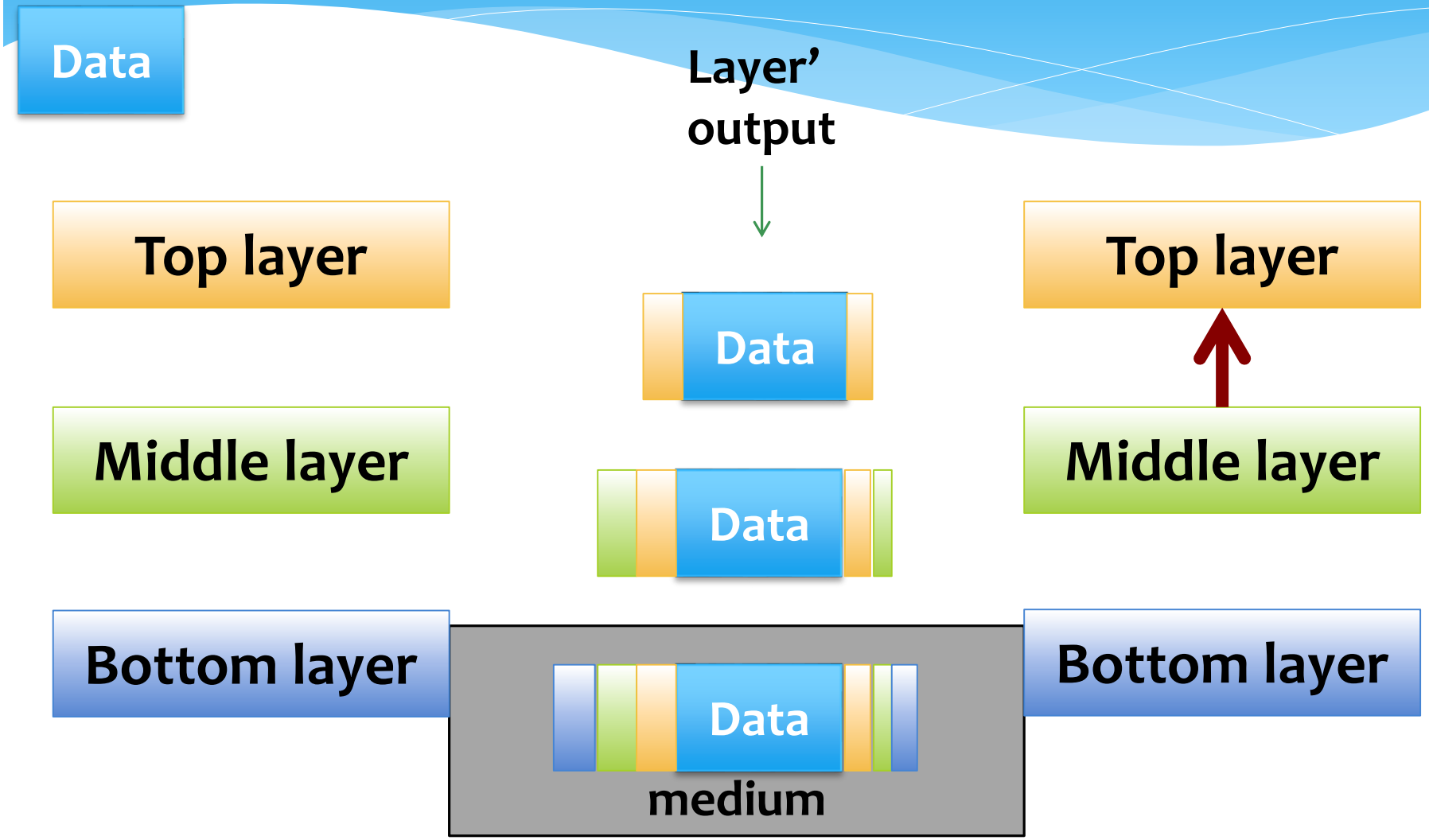
Layers:
A mock example



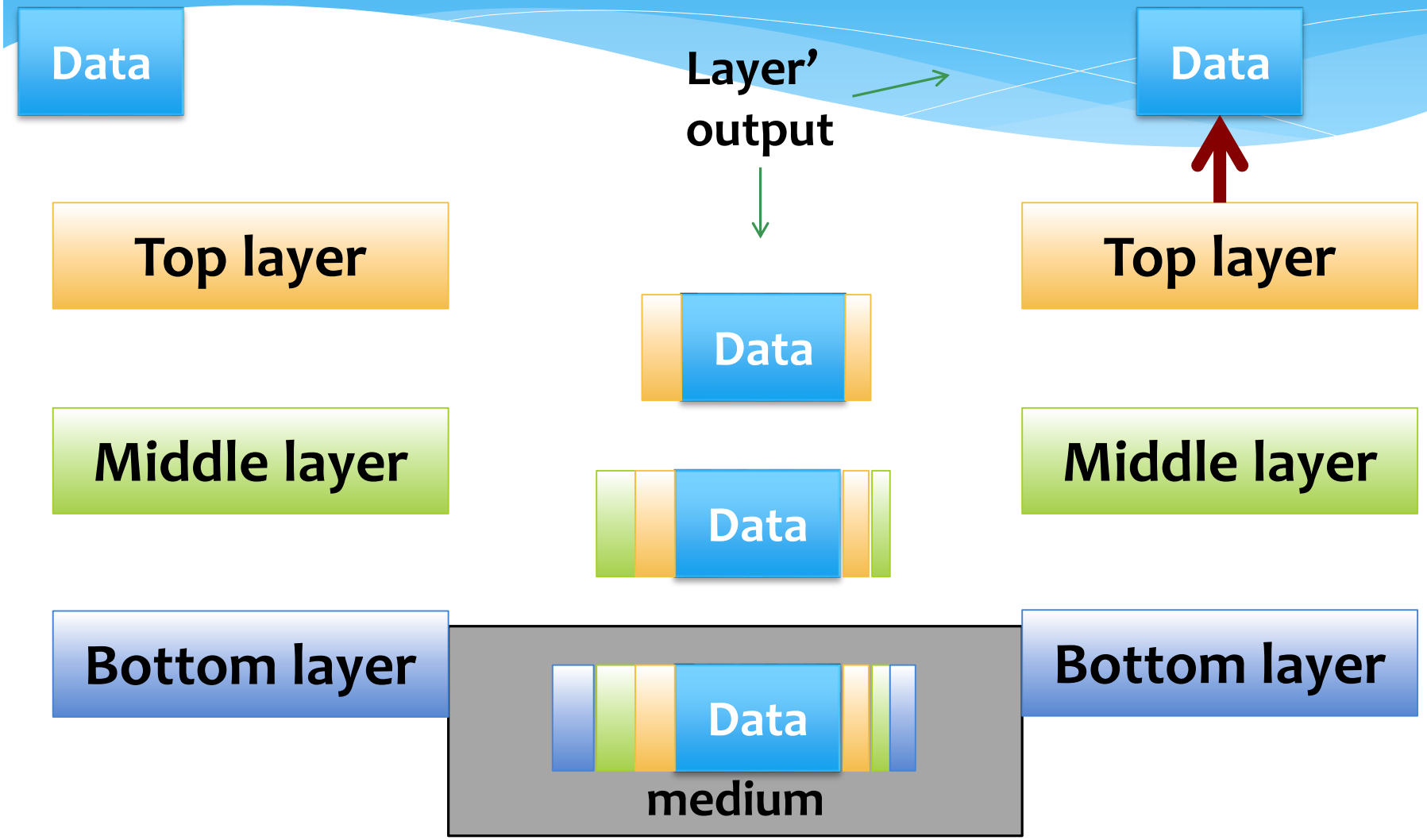
Layers:
A mock example



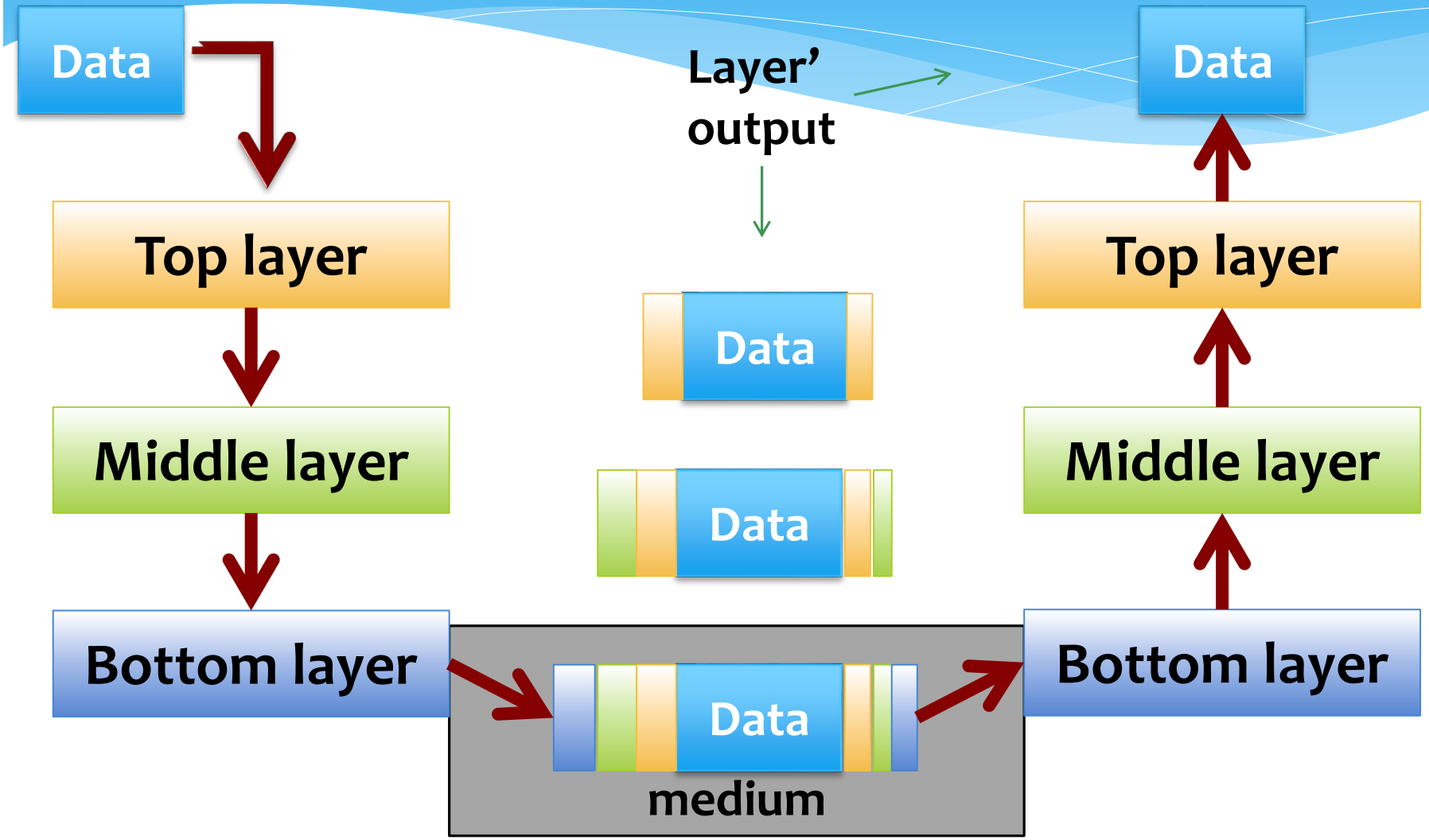
Layers:
A mock example



Layers:
A mock example

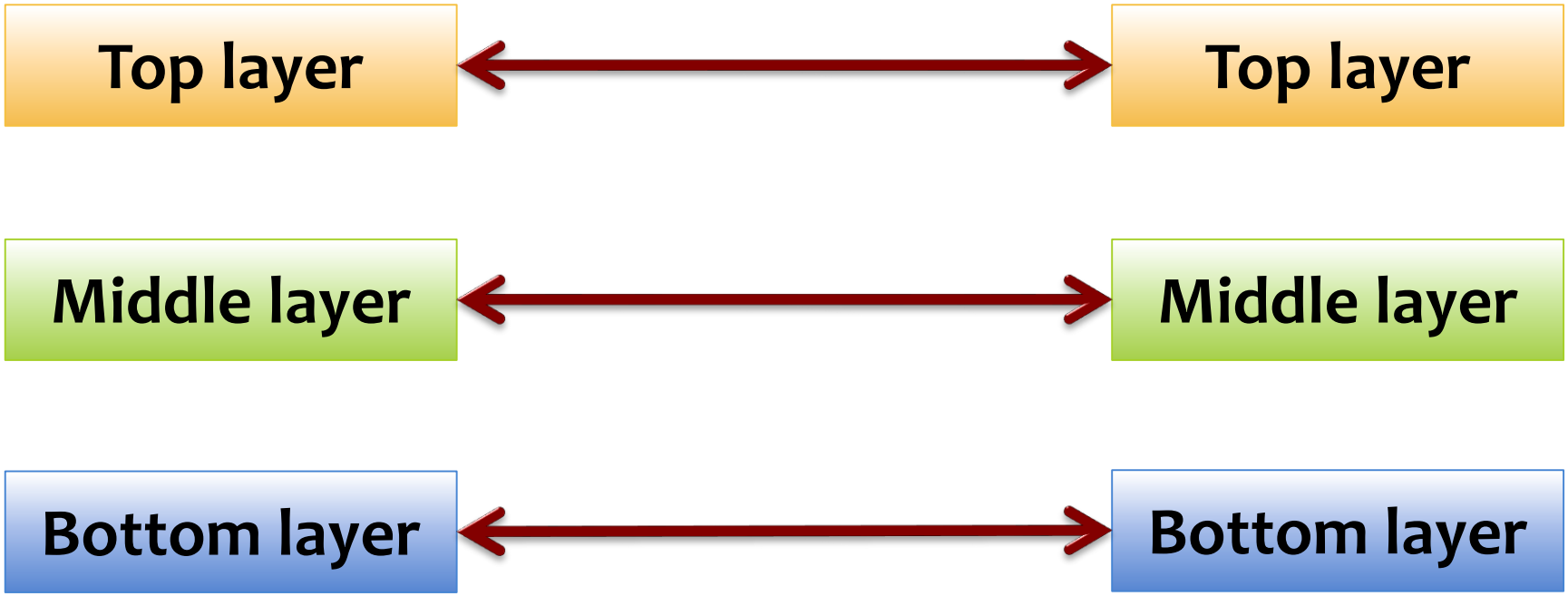


Layers:
A mock example



Layers:
A mock example

Although data communication is vertical, the modeled communication is said to be parallel



Notes on Layers

* Layered models are, indeed, ***models***
* In practice, layers may be combined
* Or protocols may operate on more than one layer
* Many models may exist to describe the same grander concept

The old PCI

The old PCI

* PCI is a parallel bus transmitting one bit of information for every line (wire)
* It is fast compared to serial buses
* It is suitable for short distances
* Yet more expensive

PCI Needed to be faster

While PCI remained slow …

* Processors were getting faster
  * And more buses were introduced
* Graphics cards required more bandwidth
  * A third bus (AGP) was introduced

These were expensive solutions, incompatible with PCI peripherals

PCI Limitations

PCI has a number of limitations, including:

* Requires one connector pin for every line
  * Each pin is something that may fail
    * And has a cost
* Is shared, all devices need to co-exist
  * No faster devices allowed
* Physical limitations
  * Crosstalk, signal reflections, sync problems

PCI-X (PCI eXtended)

* PCI-X an attempts to improve PCI
  * Another, older, attempt was AGP
* 64 rather than 32 bits
* Still expensive and not even fast enough

Bus	Width (bits)	Maximum transfer rate (Gbps)
PCI	32	1.06
PCI-X 266MHz	64	2.128
PCI-X 533MHz	64	4.256

Considered Improvements

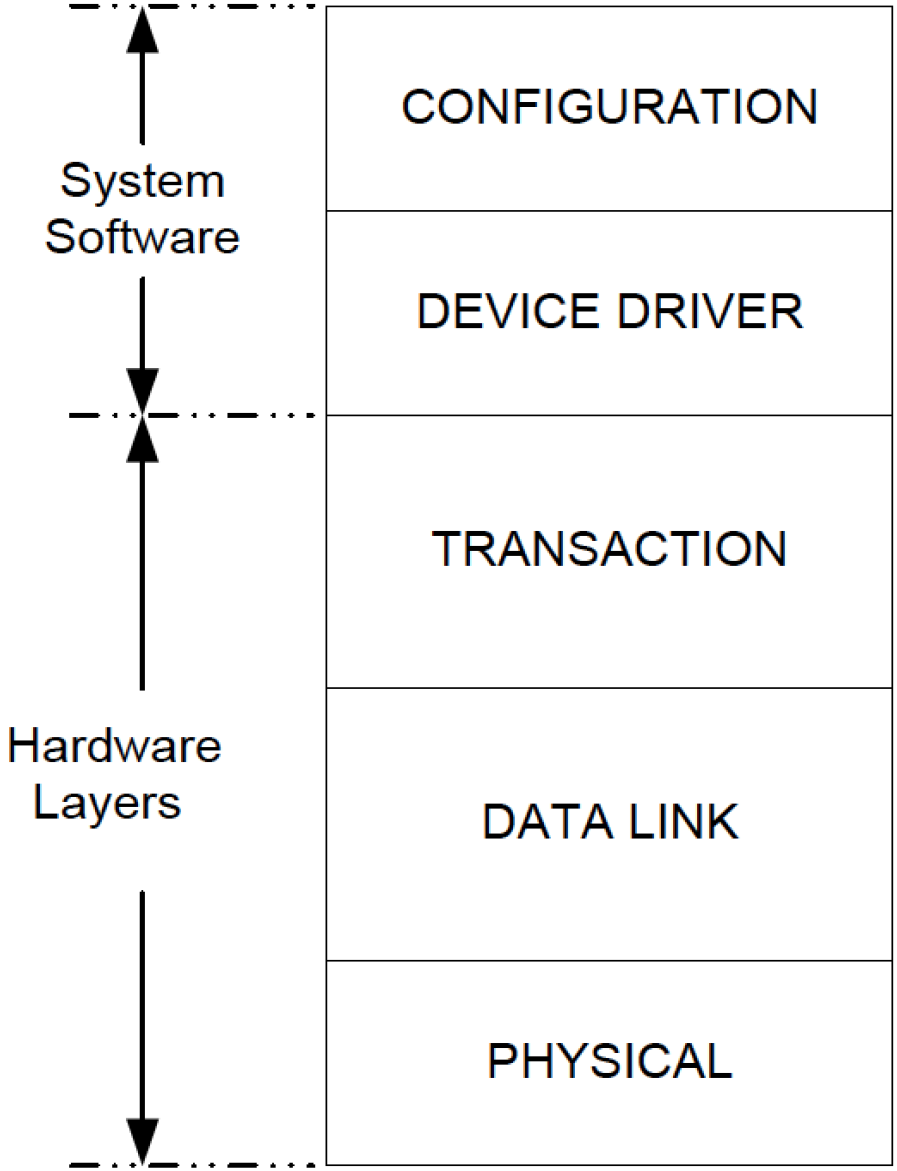
* Increase the clock speed
  * Cannot overcome signal travel time
  * High-frequency signals on thin crammed lines cause crosstalk / high signal loss
* Increase the bus width
  * Connectors would be expensive and very large
  * More things could fail, prone to signal problems
  * Doubling the width would double the speed but it would still not be enough to meet future reqs

PCI Express

PCI Express (PCIe)

* Although similarly sounding, PCI Express (or PCIe) is **not** PCI-X
* It is a **serial** and **scalable** interface
* Layered approach
* Communicates by exchanging packets
* Point-to-point link (not shared)
* Two pairs of wires per lane
  * From x1 to x16 (1 – 16Gbps)

PCIe Layers



Plug-n.Play - initialisation enumeration configuration

Send read and write requests to Transaction layer

Implements packet based protocol.

Data Intergrity:
reliable packet delivery
CRC error control

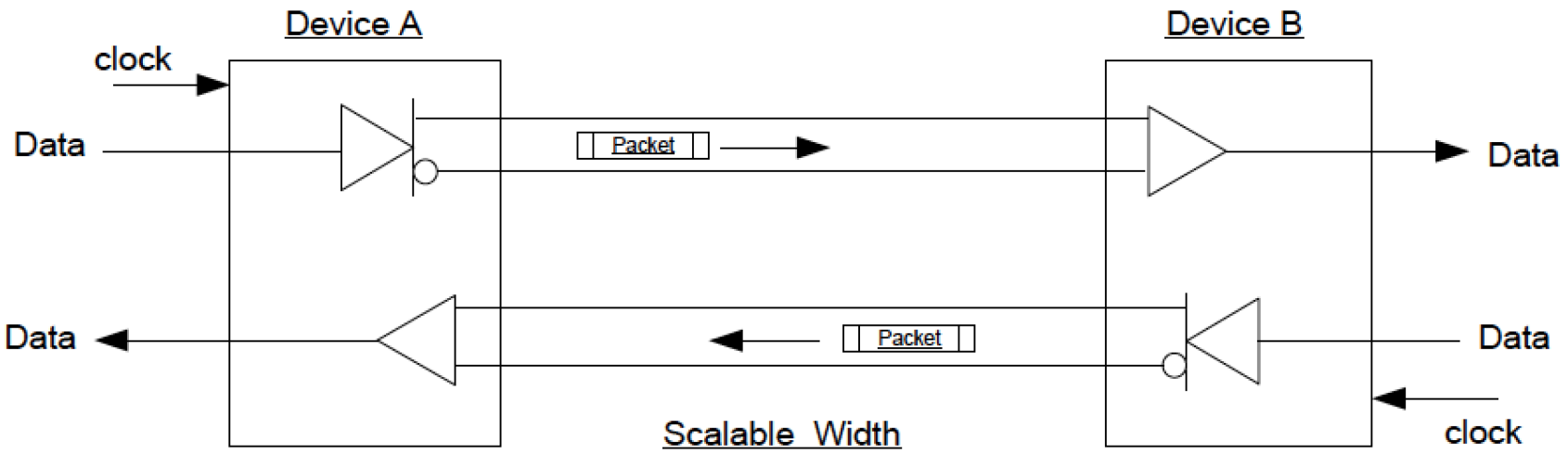
N - serial point-to-point links.
(scalable data width)

PCIe Physical Layer

* Each lane uses two pairs of wires
* Low voltage differential signaling
  * PCI would power a line if there was a bit to transmit.
    * Current was on/off at a very high frequency
    * Results to crosstalk problems
  * In PCIe, the lines form a complete circuit, with the voltage being altered to indicate a bit
    * Current direction difference cancels crosstalk

PCIe Physical Layer (continued)

* Full-duplex or dual-simplex
* 8b/10b encoding for clock synchronization
* PCIe v3.0: 1Gbps per channel



PCIe Link Layer

* Responsible for packet delivery
* Header: the packet sequence number to manage ordering in case of retransmission
* Trailer: a CRC check to test any need for retransmission
* Flow control: not too many packets too fast

PCIe Transaction Layer

* Facilitates communication with the driver
* Read/write requests (data)
* Provides for bus transactions (IRQ, DMA etc)
* Address spaces (memory address, I/O address)
* Configuration (identification, plug-and-play, IRQ numbers, DMA)
* Message addresses: translates to parallel-style bus communications for compatibility with older drivers

PCIe Software Layers

* Resides on the OS
* Compatible with old PCI and AGP systems
  * Drivers written for these can be reused with PCIe, because:
* Makes requests to the transaction layer

PCIe Bus Interface

* Northbridge includes a chip for PCIe
  * Usually manages the x16 PCIe interface for the graphics card
  * Which aggregates 16 PCIe lanes – 16Gbps
* Southbridge provides two chips
  * PCIe "switch" extends one PCIe to multiple PCIe of various aggregation sizes
  * A normal I/O controller for legacy devices

PCIe: Fun Facts and Future

* PCIe is “hot” plug-n-play
  * Plug peripherals while the computer is powered
* You can plug a x1 device on a larger slot, e.g. x4
* Thunderbolt is “the cable version of x4 PCIe”
* Future improvements
  * May be working with optical cables (for longer distances, rather than better speed)
  * v4.0 provides 16GT/s (giga-transactions per sec)

PCIe: Speed

Version	Rate x1 (MB/s)	Rate x16 (GB/s)
1.0	250	4
2.0	500	8
3.0	985	15.75
4.0	1969	31.51

Notes:

* **Even version 4.0 is a bit old (2011), time for something faster**
* **If you discuss speed, note which version you're talking about**

Serial ATA

SATA: Overview

* Serial ATA replaces Parallel ATA (PATA)
  * PATA = IDE
* Serial interface (not scalable)
* Layered approach using packets
* Point-to-point link (not shared)
* Very similar to PCIe, but dedicated for disk drives

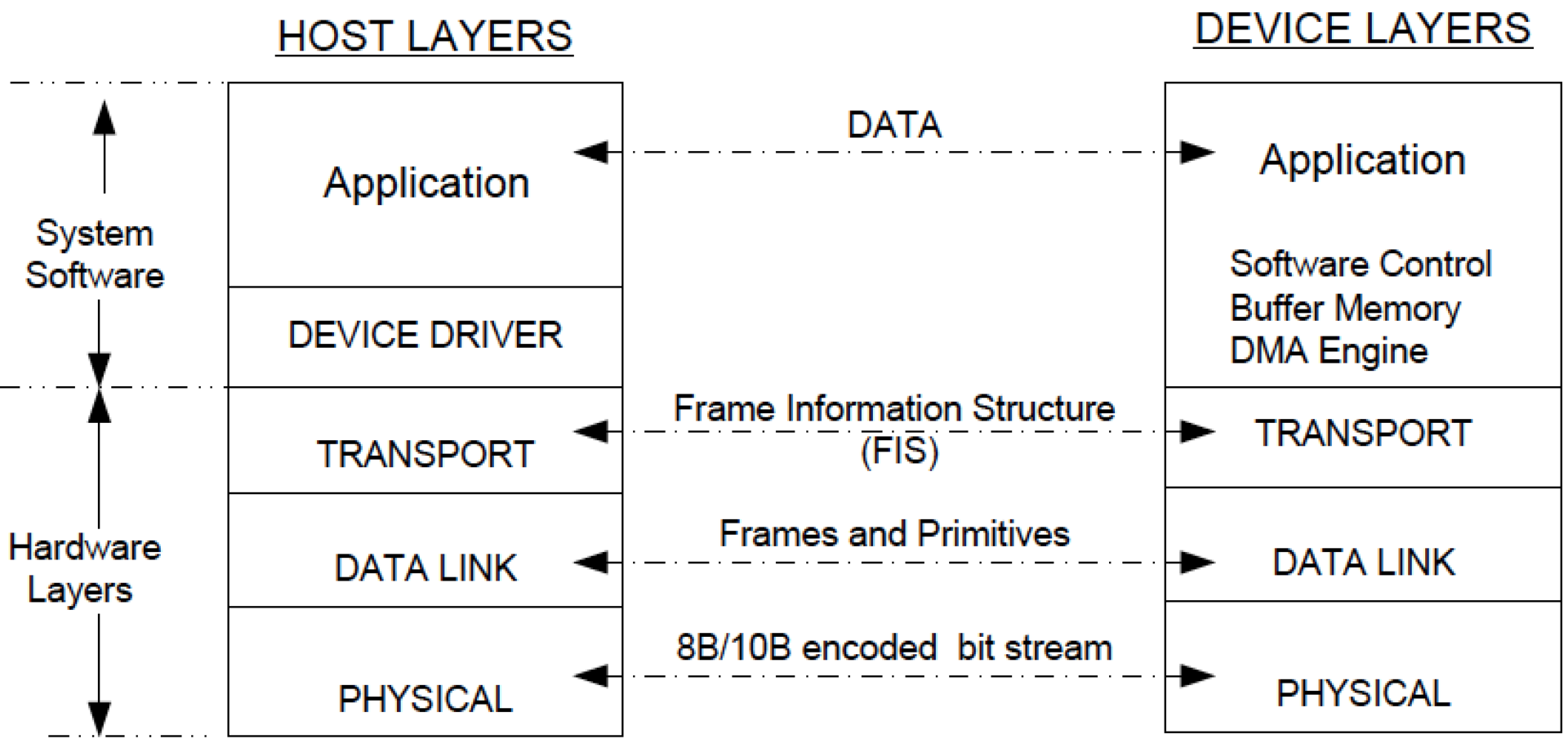
SATA: Speed

Version	Rate (Gbps)	Rate (MB/s) (after 8b/10b)
1.0	1.5	150
2.0	3	300
3.0	6	600
3.2	16	1969MB/s

Notes:

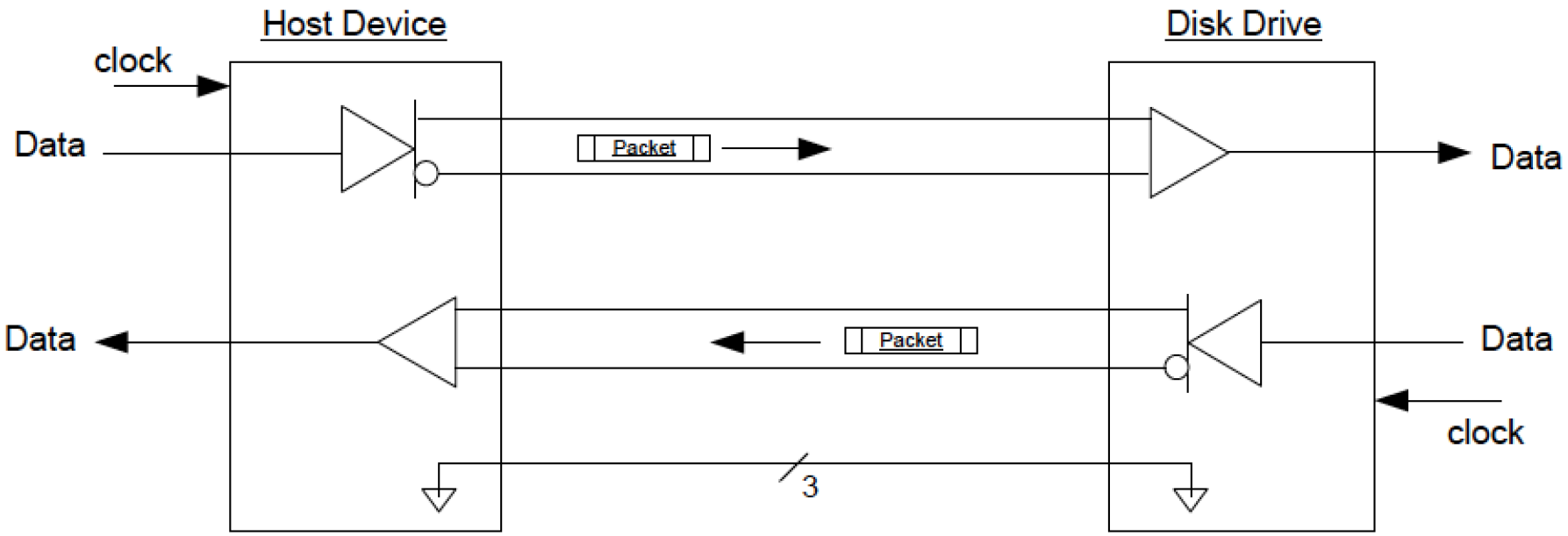
* **Versions 3.1 exists but not listed**
* **If you discuss speed, note which version you're talking about**

SATA: Layers


HOST LAYERS
DEVICE LAYERS
System Software
Hardware Layers
Application
DEVICE DRIVER
TRANSPORT
DATA LINK
PHYSICAL
DATA
Frame Information Structure (FIS)
Frames and Primitives
8B/10B encoded bit stream
Application
Software Control
Buffer Memory
DMA Engine
TRANSPORT
DATA LINK
PHYSICAL

SATA: Physical Layer

* 2 LVD pairs (4 wires), full-duplex
* 3 Ground wires (for reference voltage)
* 8b/10b encoding for synchronisation (+20% overhead)



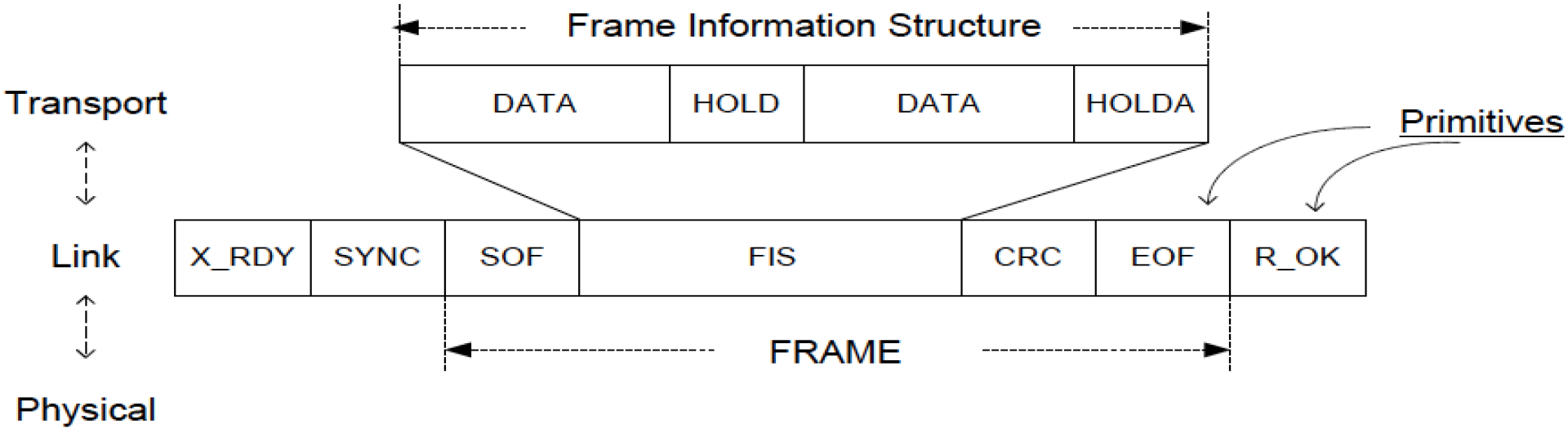
SATA: (Data) Link Layer: Primitives

The link layer transfers data in packets but controls the operation with primitives: 32-bit words that control the transfer of data. E.g.:

* X_RDY: ready to transmit
* R_RDY: ready to receive
* R_OK: data received with no error
* HOLD: for data flow control
* SOF/EOF: Start-of-frame/End-of-frame

SATA: (Data) Link Layer: Frames

* **Frames are encapsulated in primitives**
* **Contain the Frame Information Structure (FIS), with data and further primitives**
* **This layer is implementing the 8b/10b encoding**



SATA: Transport Layer

1. Manages payload activities like:
   * Accepts data from higher layers (Driver)
   * Constructs/Deconstructs a FIS
2. And other activities, like:
   * Link speed synchronisation (handshaking)
   * FIFO, Flow control
   * Acknowledgements, Error reporting
3. The type of each FIS would be examined and then sent to the appropriate activity

SATA: Application Layers

1. SATA is backwards compatible
2. The flat address/data/control model of the original PATA is still observed
3. This is converted in the Transport Layer
4. Existing software runs on SATA

SATA: Other features

* Command Queuing
  * Allow the disk to optimise operations
* 7-wire connector: 2x2 for data +3 ground
* Longer cables
* Hot plug-and-play

SATA and PCIe

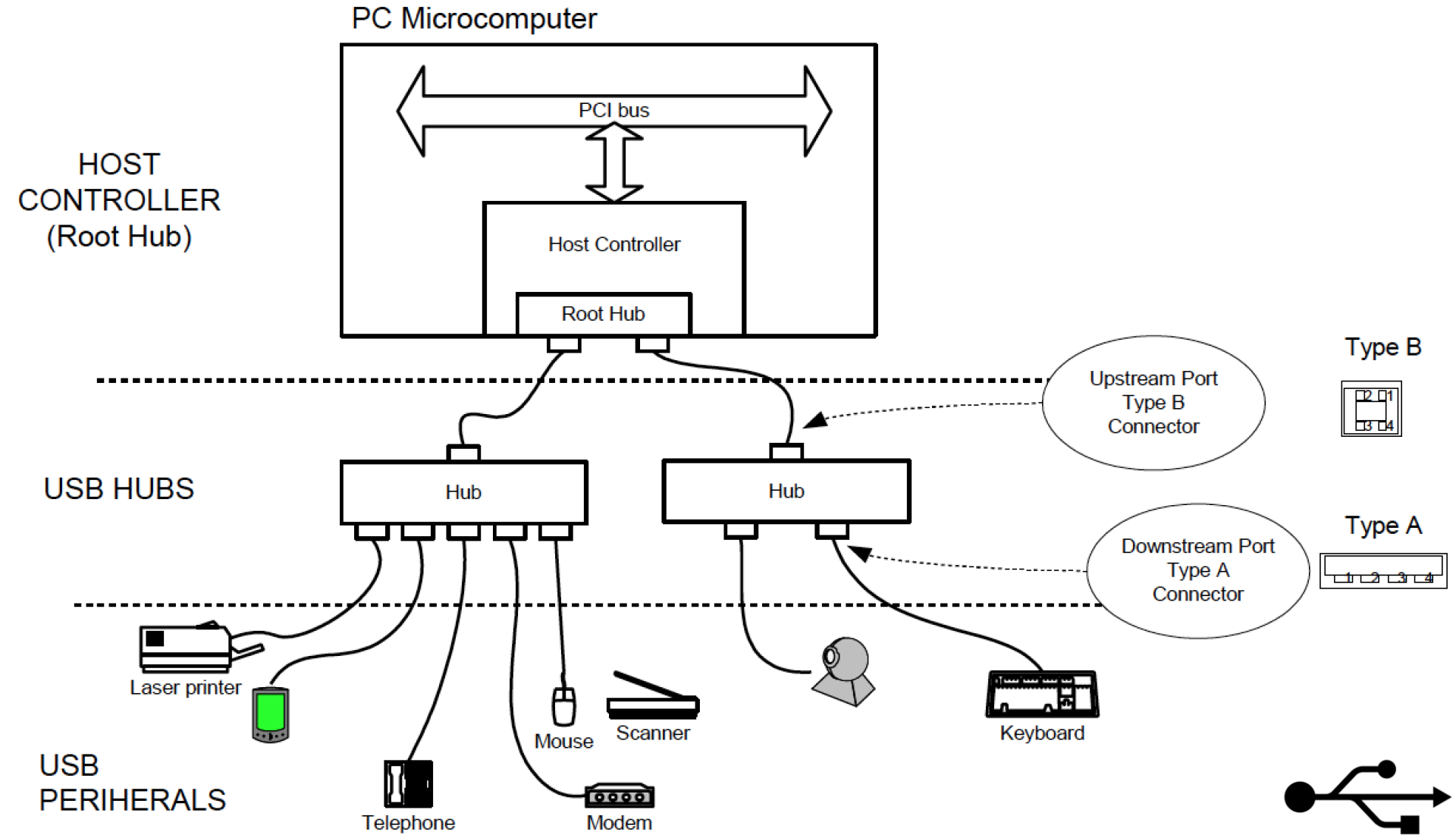
* Not only very similar, there are situations where they are combined
* SATA Express is a PCIe interface which supports SATA devices. The extensible PCIe was preferred, rather than re-inventing SATA.

USB

USB: Overview

* Universal Serial Bus is the most successful peripheral connectivity mechanism
* Serial link, layered architecture
* Up to 127 devices connected to a single host
* PC-centric, tiered-star yet logically P2P
  * But speed is shared amongst connected devices
* Reliable hot plug-and-play

USB: Topology



USB: Plug and play

USB provided the first reliable hot plug-and-play

* Host constantly checks for new devices
* Inserted devices are interrogated for their unique ProductID and VendorID
* Host uses this information to locate and load a driver (may search the internet)
* No need to deal with IRQ, that is already assigned to the host controller

USB: At the device end

The hardware and software architecture used in USB end devices is the Device Interface Architecture.

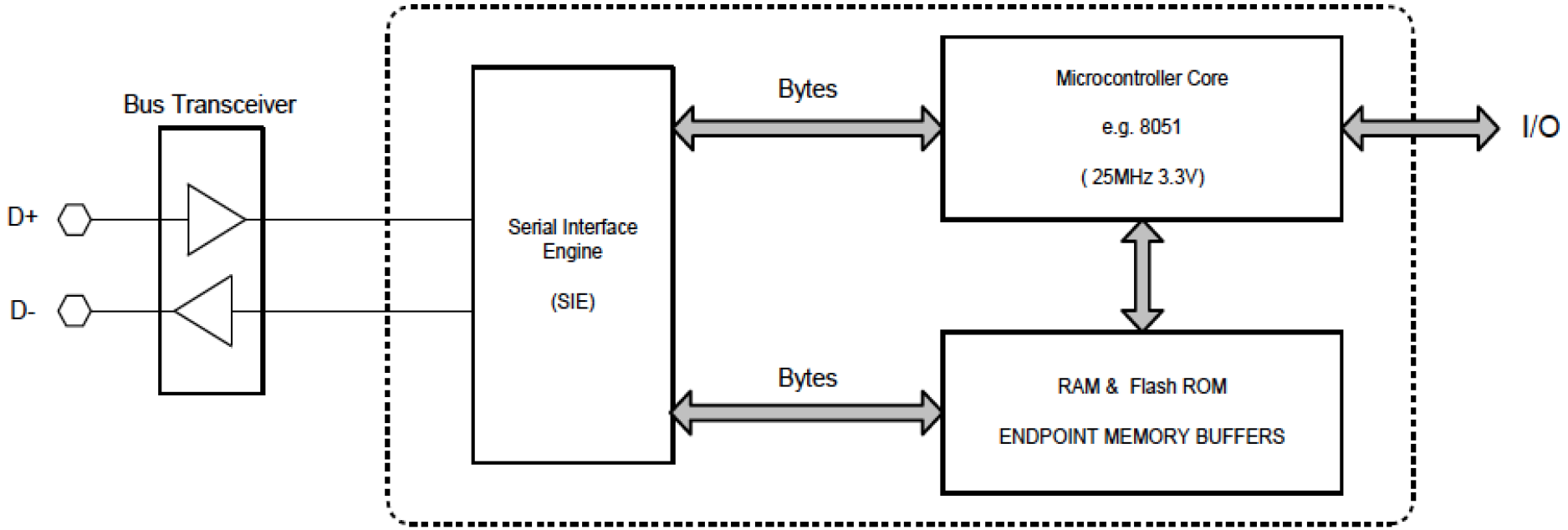
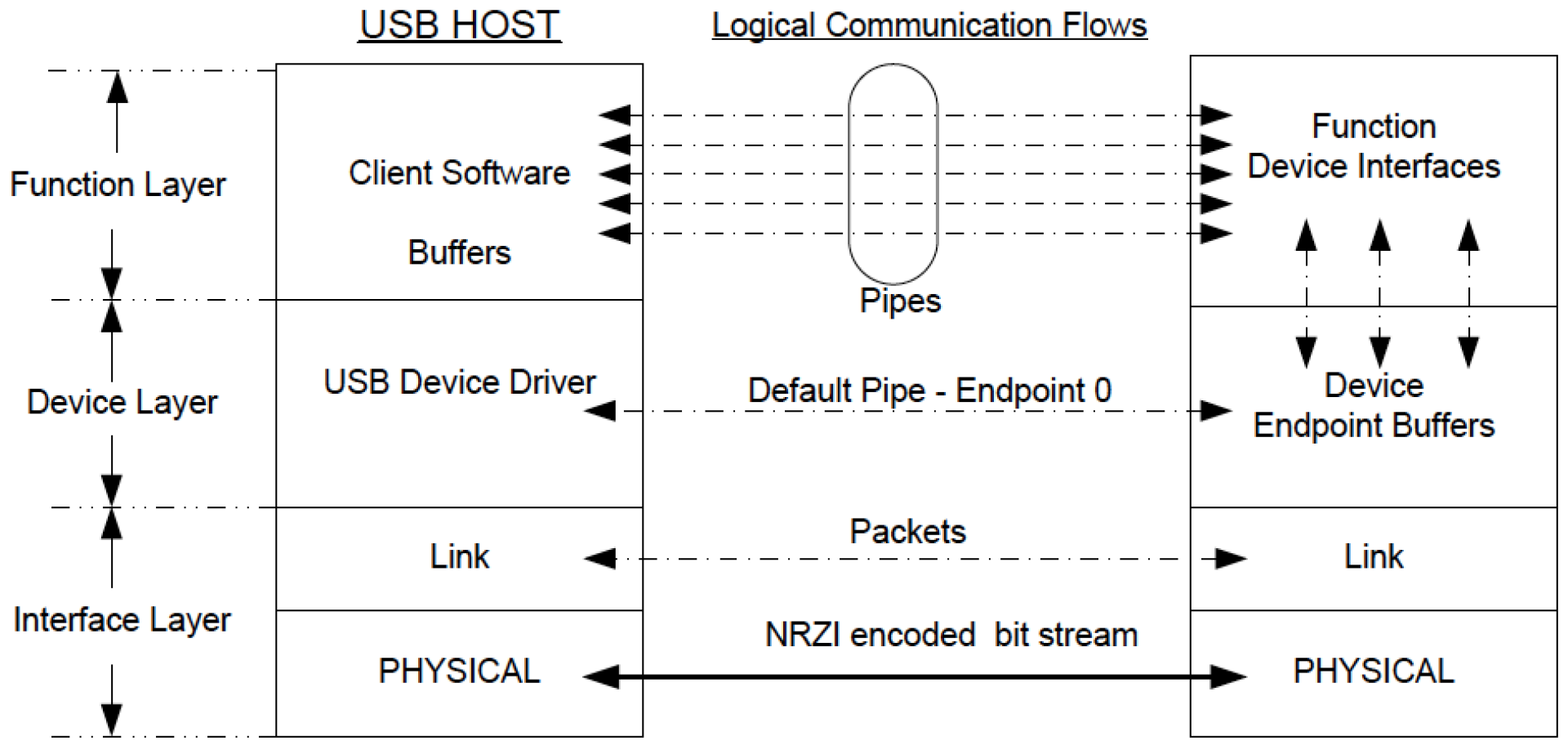


Figure 23 USB Peripheral Interface

USB: Layers


USB HOST
Logical Communication Flows
Function Layer
Device Layer
Interface Layer
Client Software
Buffers
USB Device Driver
Link
PHYSICAL
Pipes
Default Pipe - Endpoint 0
Packets
NRZI encoded bit stream
Function
Device Interfaces
Device
Endpoint Buffers
Link
PHYSICAL

USB: Host

* In order to support different kinds of devices, USB is a complex bus
* Although multiple hubs may be used, the host is the communication center and bus master
* Only the host controller uses PC resources like IRQs, RAM, address space etc
* The host is the first hub, typically with 4 ports

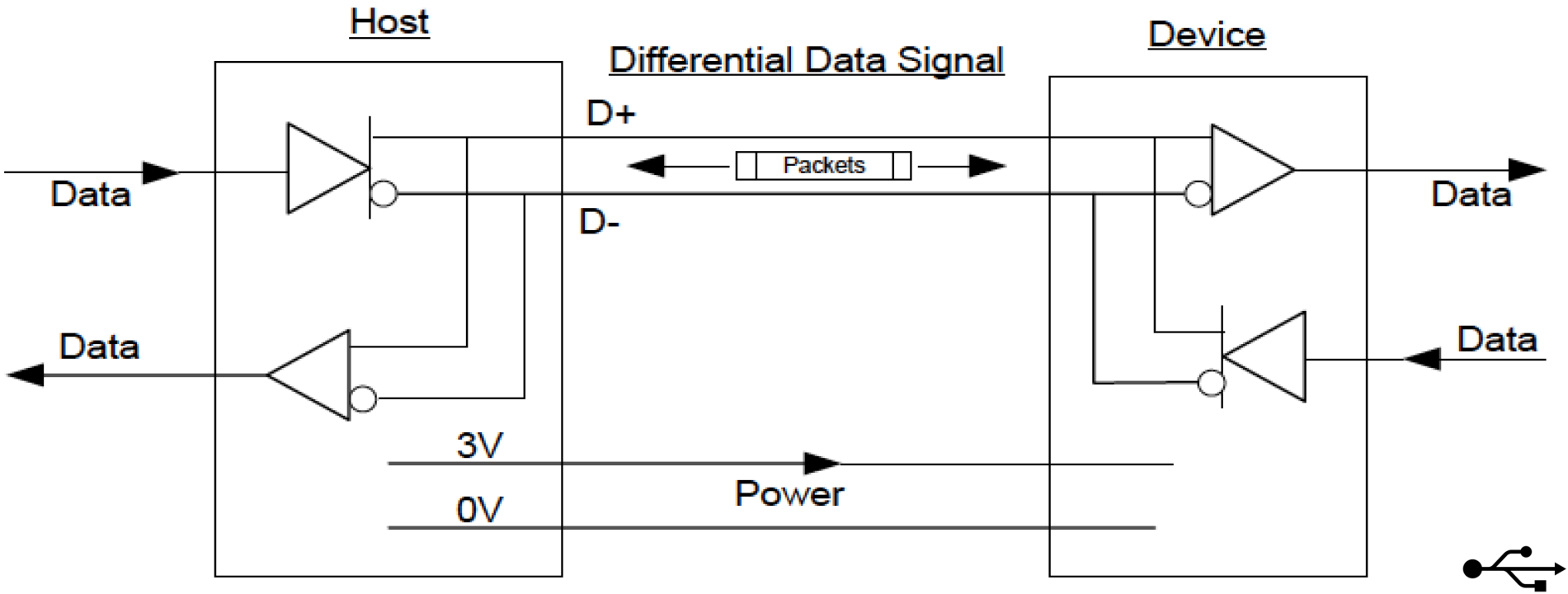
USB: Abstractions

The following abstractions model the logical communications

* Host: the PC computer
* Functions: connected devices or parts of them
* Pipe: allows data flow of various data types
* Buffers: places in memory
* Endpoints: buffers for specific use

USB: Physical Layer

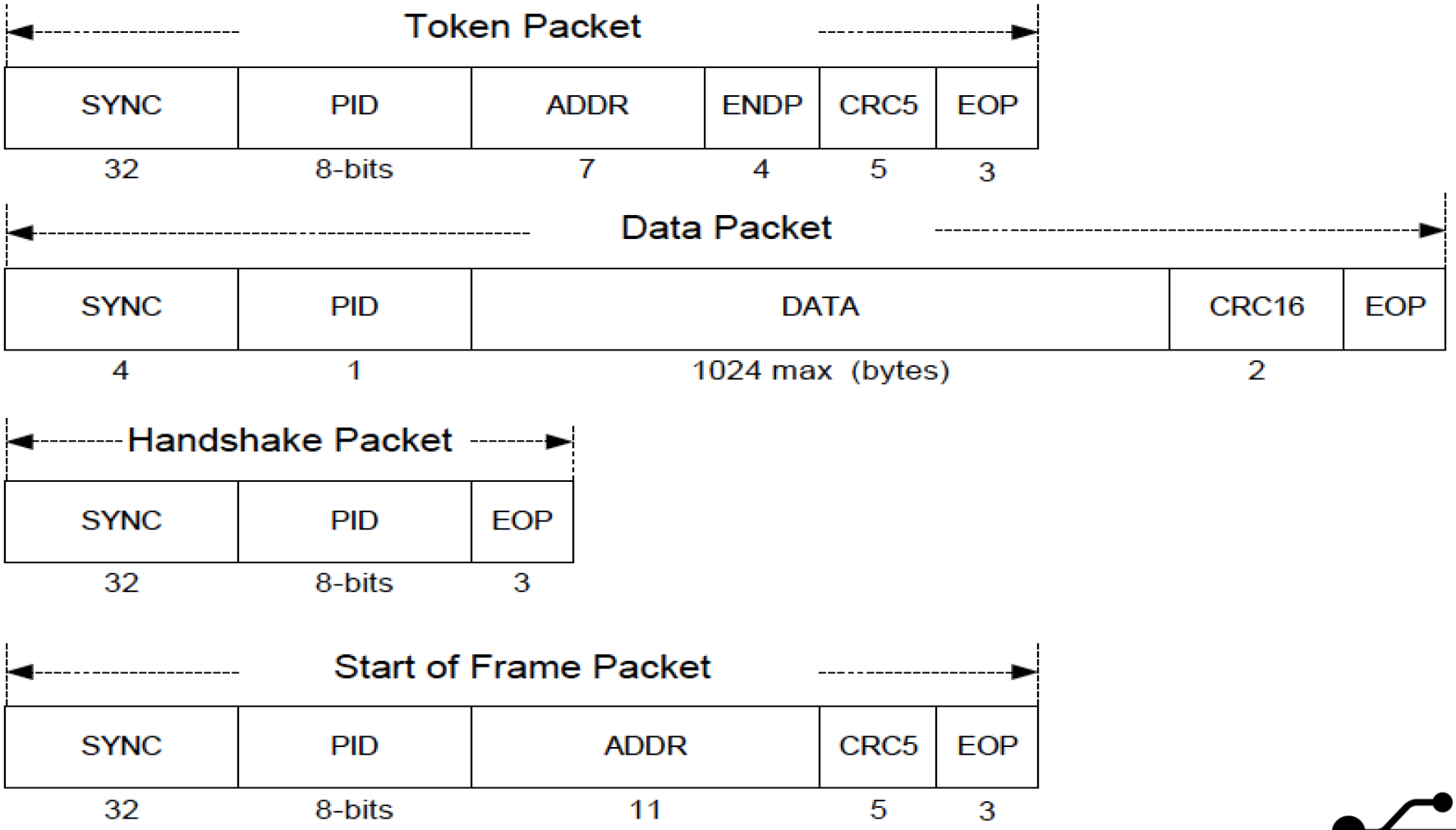
* Differential pairs for signal
* Power and ground
* NRZI encoding
* Twisted pairs



USB: Packet Types

* Token: IN, OUT, SETUP
  * Followed 125 microseconds of data
* Data: up to 1024 bytes payload – with subtypes
* Handshake: ACK, NAK etc
* Start-of-Frame

USB: Packet structure



USB: Packet fields

* SYNC: a sequence for clock to syncronisation
* PID: The type of packet, e.g. ACK or DATA
* ADDR: Destination address, 7 bits
* ENDP: endpoint within a device
* CRC: 5 or 16 bits
* EOP: End-of-Packet

USB: Link Layer

* Processes transactions initiated by the host
* One or more microframes of data transfers
* Created transactions have phases
  * Token packet
  * Data packet
  * Handshake packet
* There are three types of transactions

USB: Link Layer
Transaction types

* Control: bidirectional, for commands and configuration, error-checked
* Bulk: for error-checked yet low priority data
* Isochronous: for time-critical data where integrity is not important (e.g. video)
* Interrupt: to check if data is available

Remember: different kinds of devices may be connected, e.g. a USB drive and a Camera. These have different requirements and the transaction types are generalisations designed to meet them

USB: Function Layer

* Connects the software at the host computer with the actual data-generating hardware of the endpoint device

USB: Version 3.0

* Improved on the 480Mbps of USB 2.0
* Has two additional **shielded** twisted pairs
  * A total of three shielded pairs (6 wires)
  * Plus two power wires
* Dual-simplex mode
* Similarities with PCIe
* 60% less power consumption, better PM

USB: Version 3.1

* Speed improved to 10GBps
* Used with USB Type-C connector

End of Lecture

Questions?