Precision Time Synchronization in Data Centers

Research Scientist, Meta
Ahmad Byagowi

<u>Why</u> to have precision time synchronization in Data Centers?

<u>How</u> to have precision time synchronization in Data Centers?

Background

- Speed of light and speed of electricity are finite!
  - Speed of Light is slower in fiber (2.14×10^8m/s) than in vacuum (2.99×10^8m/s)
  - Latency in data transfer over the network
- Hyperscale cloud services serve people across the globe
  - Geographical distribution of customers
  - Necessity of distribution due to safety, robustness and regulations
- The demand for hardware resources is always increasing
  - Increase in users, data and multi dimensional contents
  - Interactivity with the data like multi-user and VR
  - Big data and AI
- Heterogeneity in Manufacturing
  - Every Component is unique
    - Difference between Oscillators, RAM, CPU, etc... results in different Runtime!

Solution to Address the Hyperscale Demand

- Horizontal Scaling
  - Expansion of resources

- Geographical expansion
  - Reduction of latency

Challenges with Expansion

- Distributed Systems
  - Consistency vs Availability
  - Propagation of Information
  - Need for Redundancy
- Different in Runtime
  - Tail Latency
    - As the number of machines (parallel pipelines) increases, the variance increases
- Distribution of Clock
  - A common reference between all machines to alignment
  - Clock Skewness better than Latency

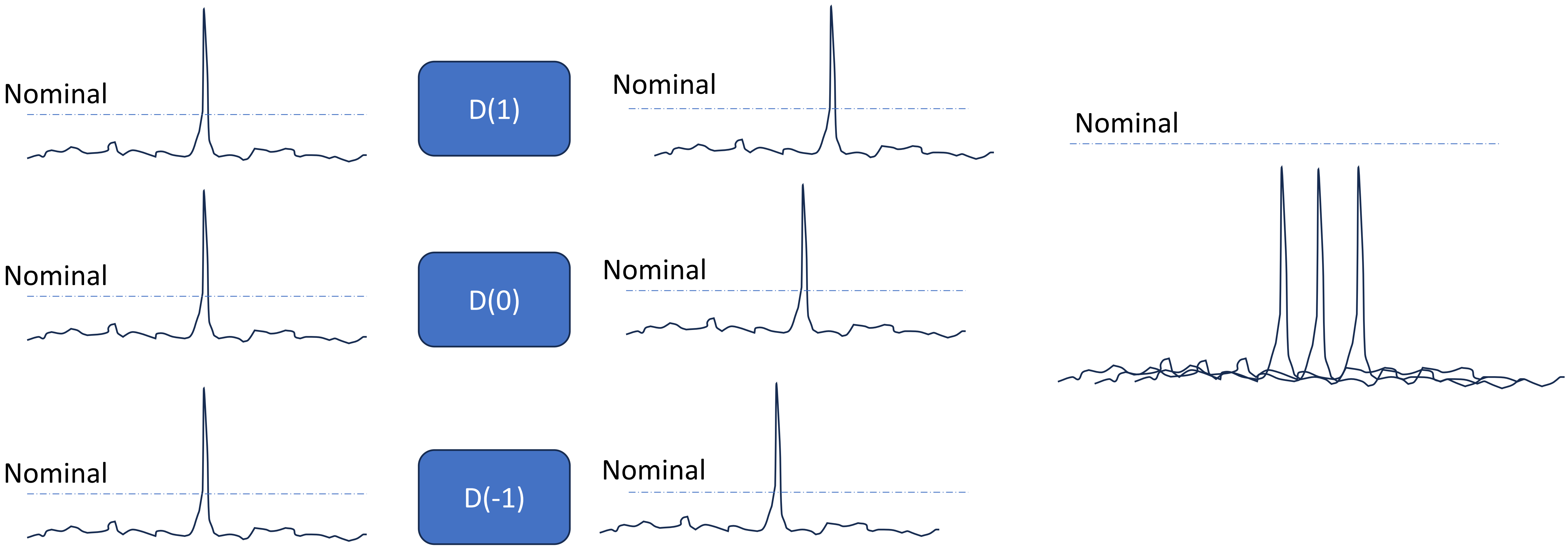
Categories of use cases

Use cases of Precision Time Sync in Distributed Systems:

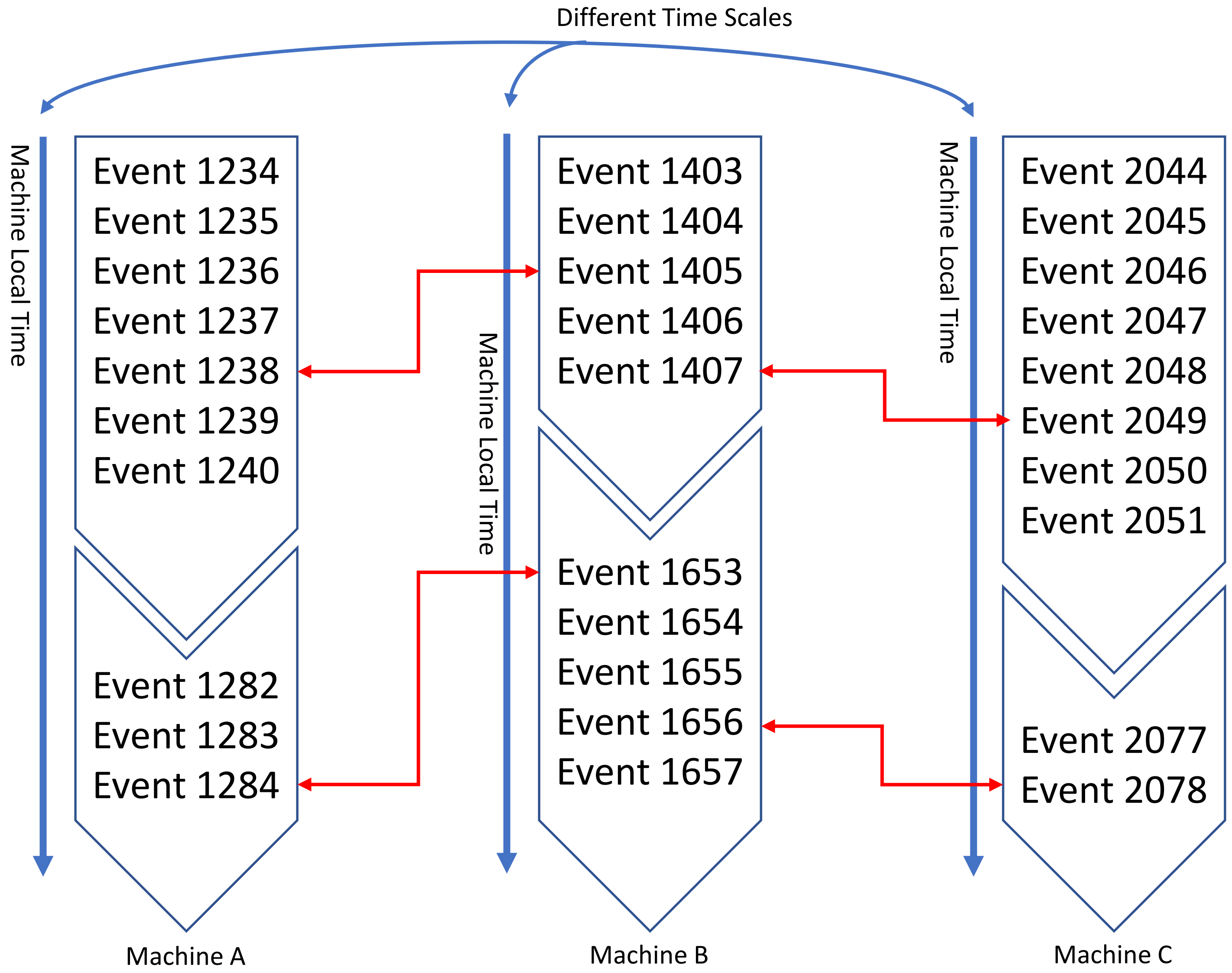
- Synchronization (Phase)
  - Active (1)
    - Running Events at specific time
      - Sync or desync
  - Reactive (2)
    - Measure latency, time or intervals
- Syntonization (Frequency)
  - Active (3)
    - Calibrate speed and align runtime to reduce tail latency
  - Reactive (4)
    - Measure heterogeneity or provide binning

	Active	Reactive
Phase	Class 1	Class 2
Frequency	Class 3	Class 4

Time Division of Power Spikes (Class 1)



Associate of Events Between Machines (Class 2)



Precision Time Synchronization Requirement

- Different Requirements for Different Levels
  - CPU (sub nanoseconds)
  - OS and Kernel (sub microseconds)
  - Machines in a Data Center (sub milliseconds)

Precision Requirement at CPU level

- Nyquist sampling theorem
  - Sampling interval required to avoid aliasing
  - Sampling frequency should be at least twice the highest frequency contained in the signal
- Frequency in event occurrence
  - Instruction Latency
  - Instruction Throughput

mov = 1 CPU cycle

xchg = 3 CPU cycles

rdtsc = 1 CPU cycle

A CPU with a clock speed of 3.2 GHz executes 3.2 billion cycles per second
That is a period of about 310ps

Precision Requirement at OS level

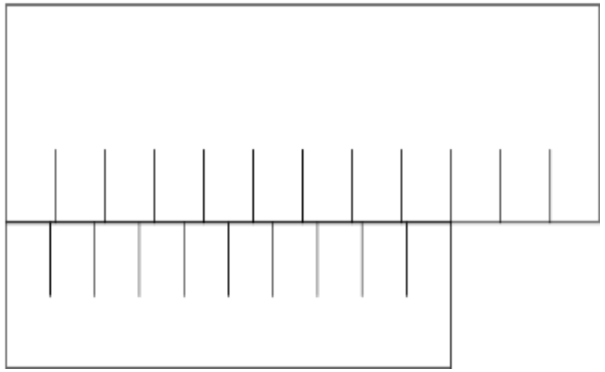
- dmesg

```
[52603.373642] {38}[Hardware Error]: event severity: corrected
[52603.373643] {38}[Hardware Error]:  Error 0, type: corrected
[52603.373644] {38}[Hardware Error]:   section_type: PCIe error
[52603.373644] {38}[Hardware Error]:   port_type: 4, root port
[52603.373645] {38}[Hardware Error]:   version: 3.0
[52603.373645] {38}[Hardware Error]:   command: 0x0547, status: 0x0010
[52603.373646] {38}[Hardware Error]:   device_id: 0000:b7:01.0
[52603.373647] {38}[Hardware Error]:   slot: 255
[52603.373648] {38}[Hardware Error]:   secondary_bus: 0xb8
[52603.373648] {38}[Hardware Error]:   vendor_id: 0x8086, device_id: 0x352a
[52603.373649] {38}[Hardware Error]:   class_code: 060400
[52603.373649] {38}[Hardware Error]:   bridge: secondary_status: 0x0000, control: 0x0013
```

System Logging is based on clock_boottime (clock_minotone_RAW) with a quanta on 1us
Events occur faster than the quanta of 1us (aliasing)

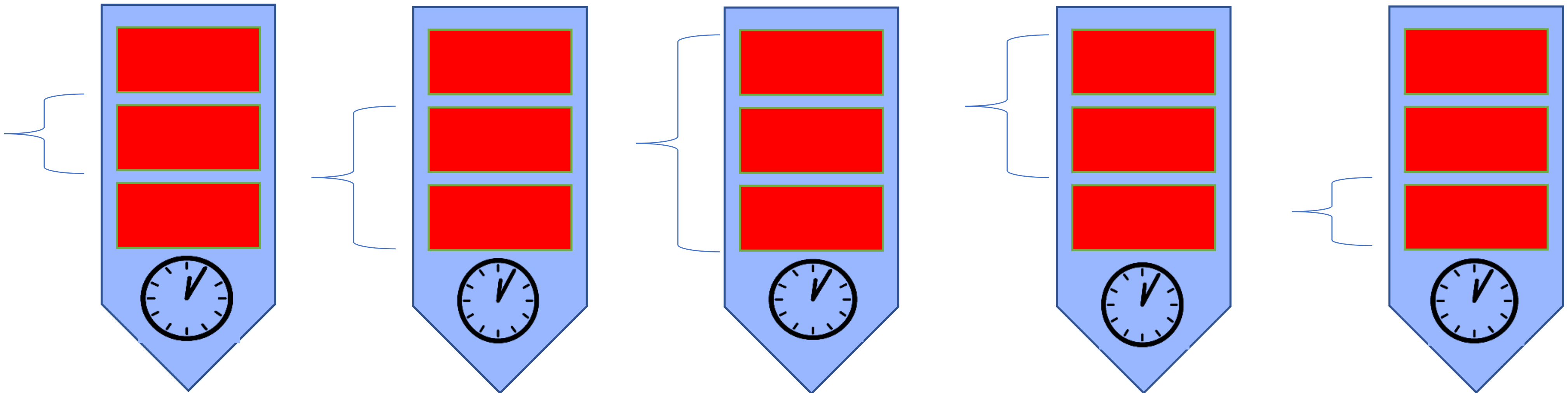
Challenges and the Precision Requirement

- Vernier acuity

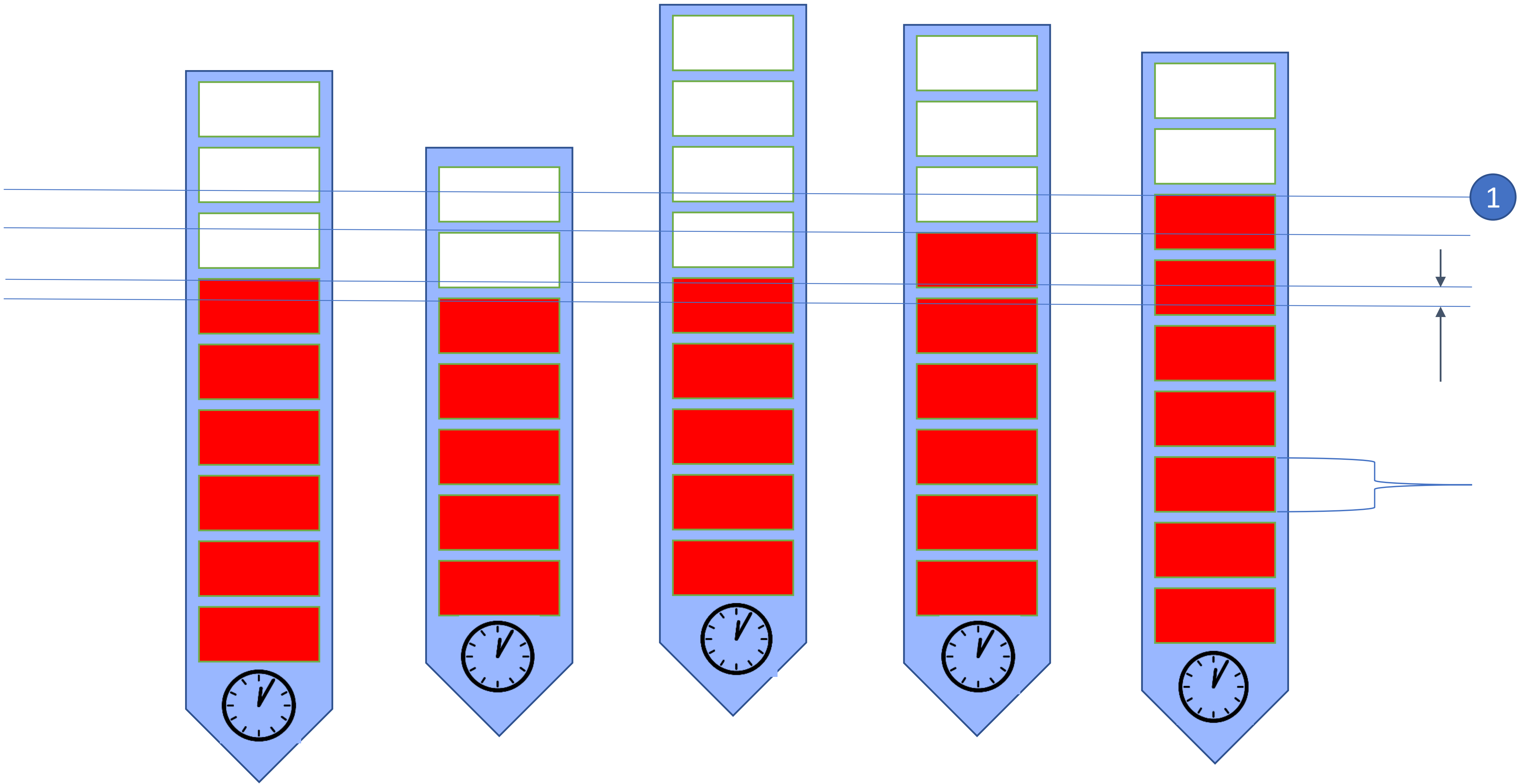


- Compounding of Events

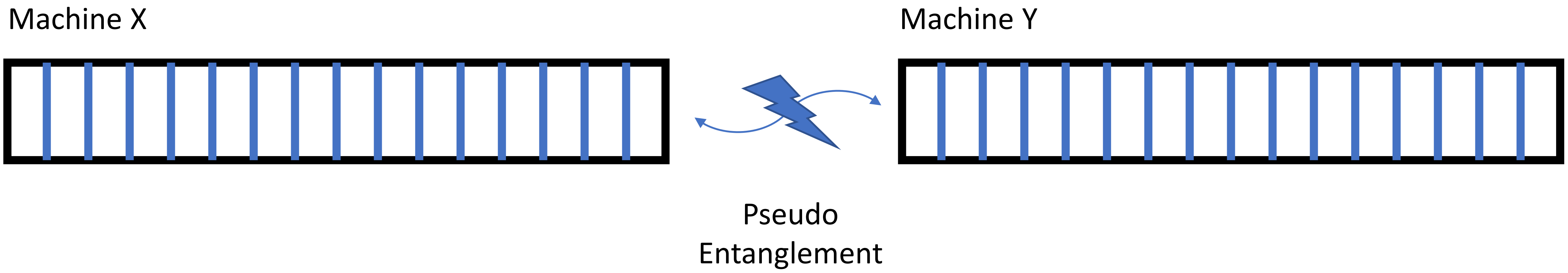
$$1^n = 1$$



Precision Requirement for Distributed Systems



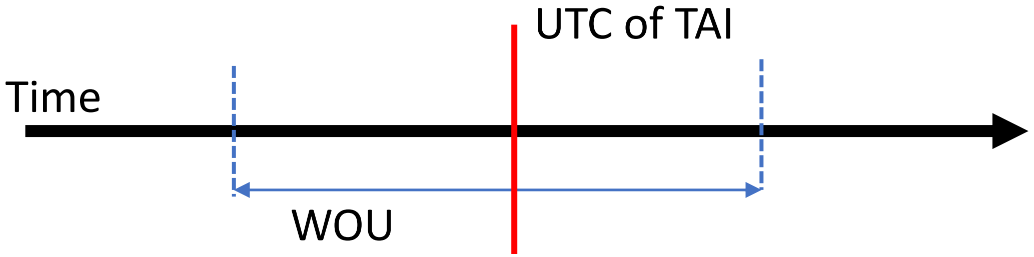
What comes out of Precision Time Sync?



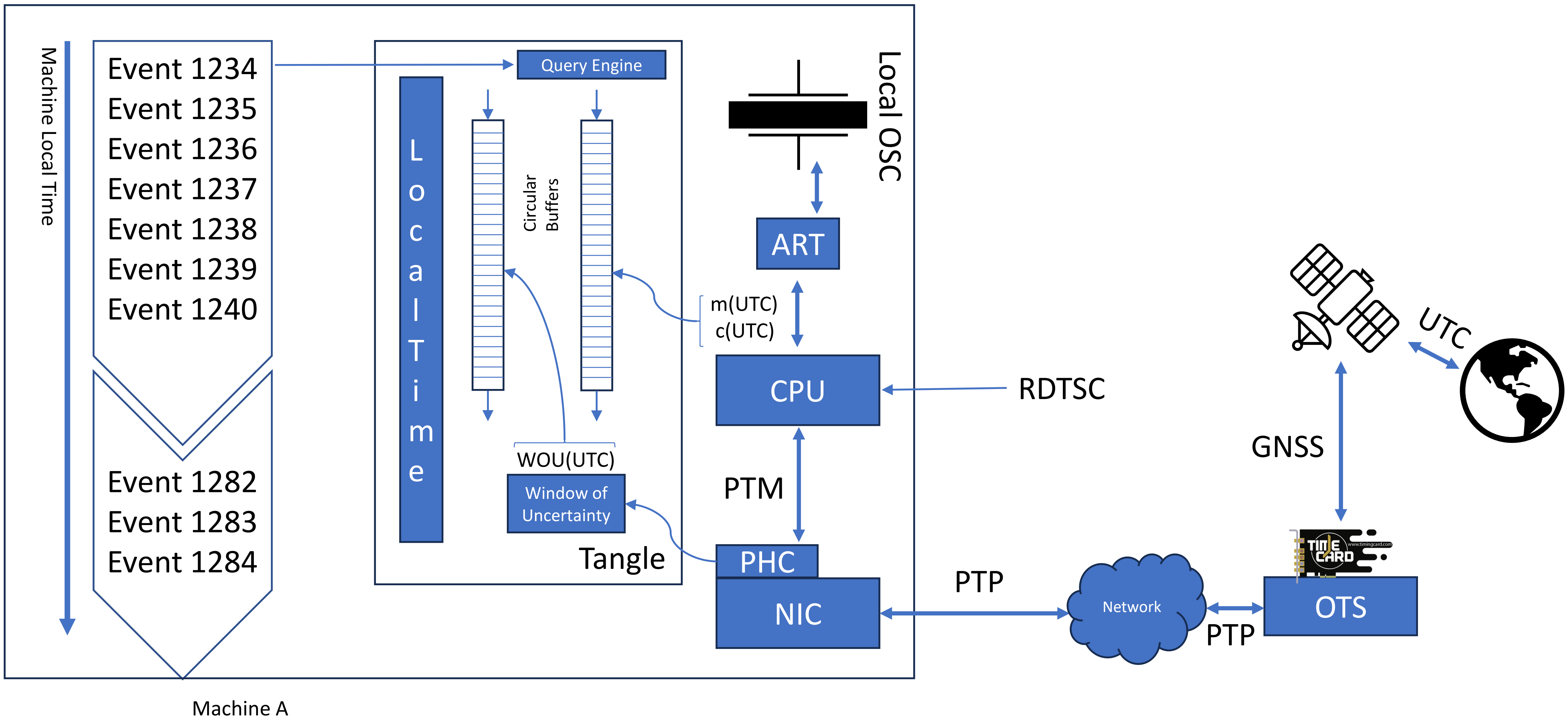
Machine X and Y are any two machines across the globe or inside a local network

Pseudo Entanglement: Probabilistic Entanglement of two Registers (Machine Y and Y) within the Windows of Uncertainty

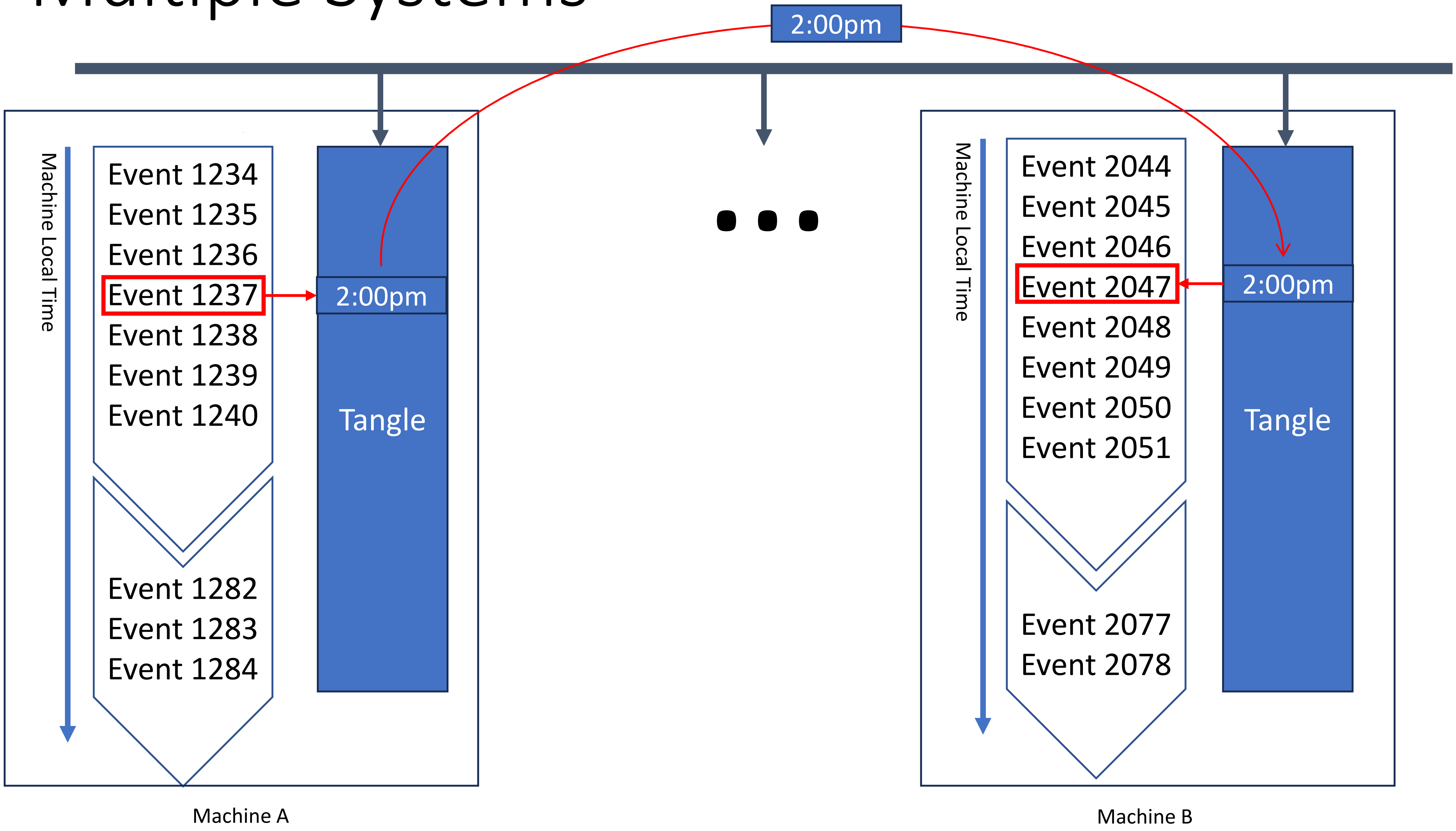
Window of Uncertainty: An ongoing estimation of a time interval that UTC (or TAI) sits inside it (with a given probability)



Tangle


Machine Local Time
Event 1234
Event 1235
Event 1236
Event 1237
Event 1238
Event 1239
Event 1240
Event 1282
Event 1283
Event 1284
Query Engine
Local Time
Circular Buffers
WOU(UTC)
Window of Uncertainty
Tangle
Local OSC
ART
m(UTC)
c(UTC)
CPU
RDTSC
PTM
PHC
NIC
PTP
Network
PTP
OTS
GNSS
UTC
Machine A

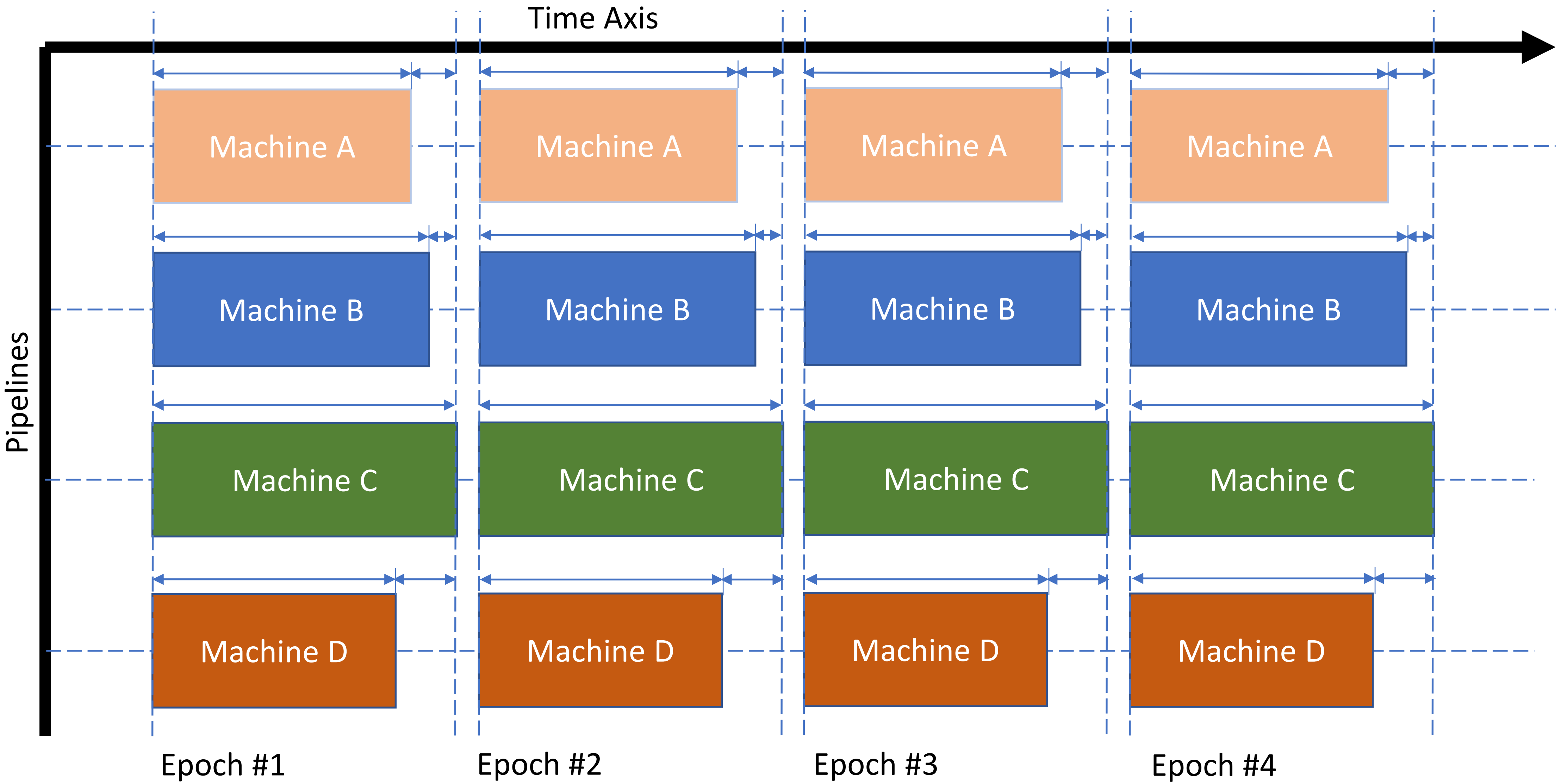
Multiple Systems



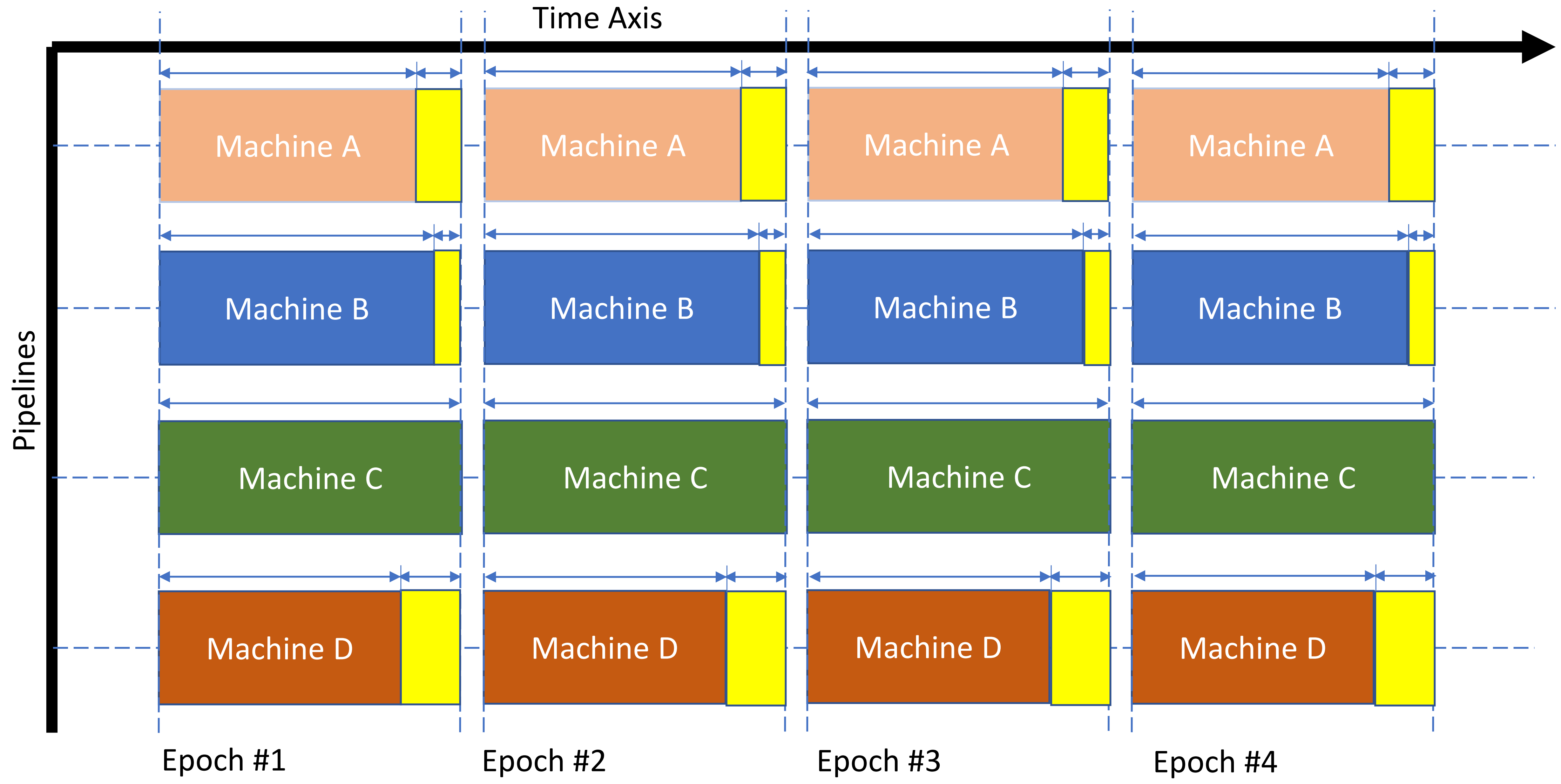
Functions

- Identify concurrent event in another machine[s]
- Find the timestamp of an event in another machine[s]
- Chronologically Rank a given event across machines
- Measure the one-way-latency between machines
- Identify concurrent events with one-way-latency consideration
- Trace chronological order for sequence of events
- Benchmark machines by precise runtime measurement
- Directly utilize RDTSC for maximum precision in event timestamping

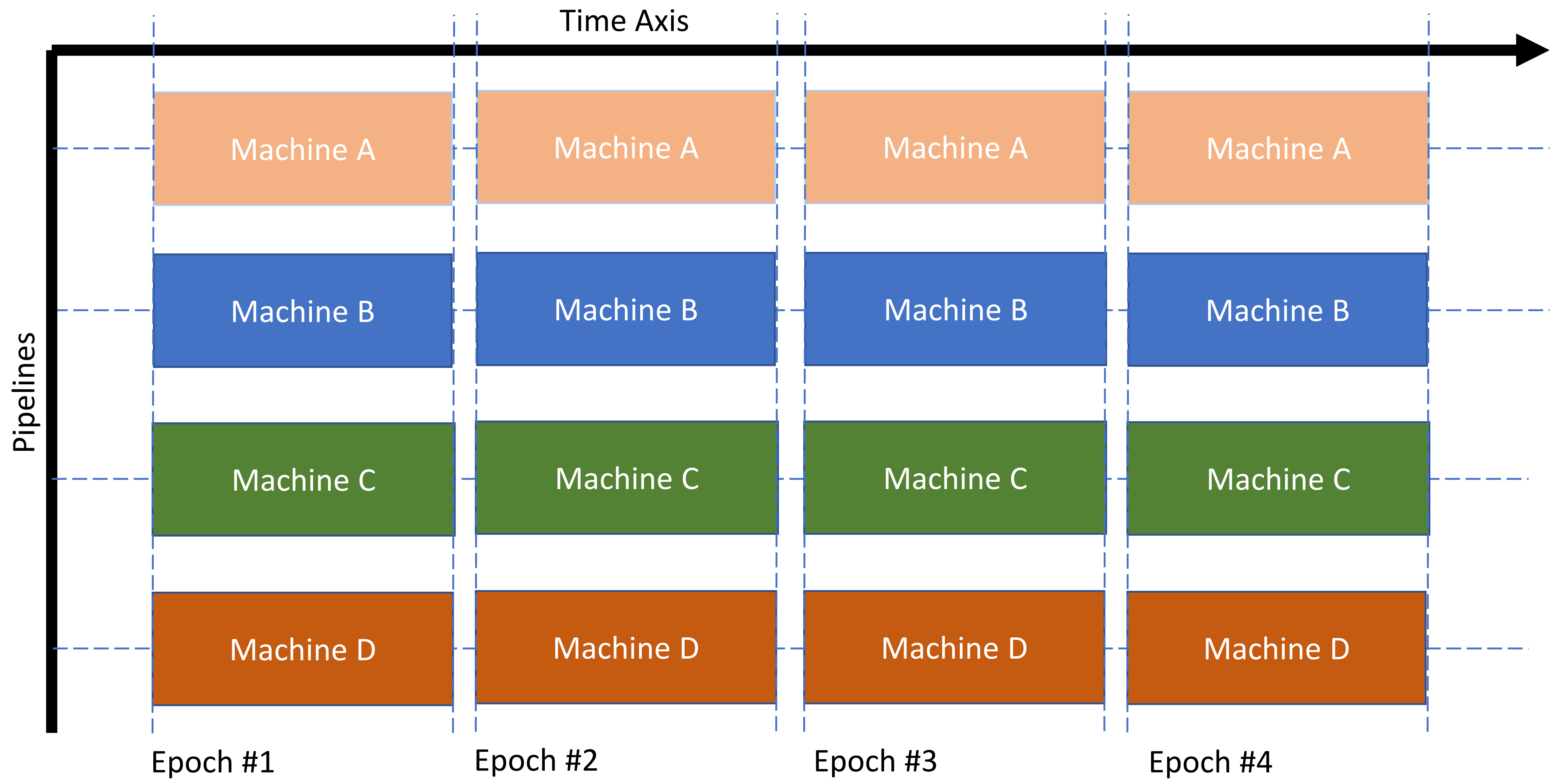
Runtime Difference in a Pipeline (Class 4)



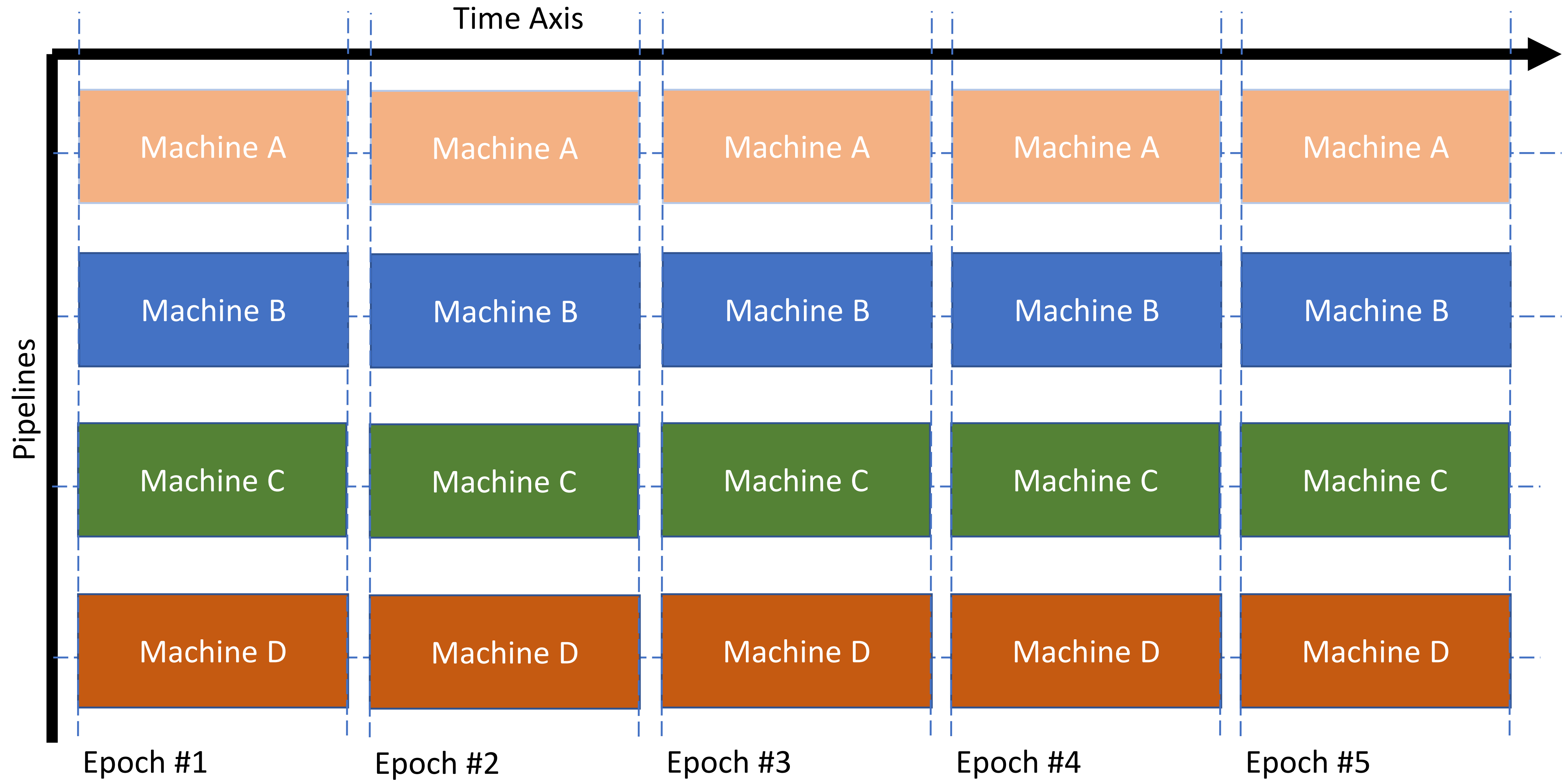
Issues with Heterogeneity in Runtime



Slowing Machines Down (Class 3)



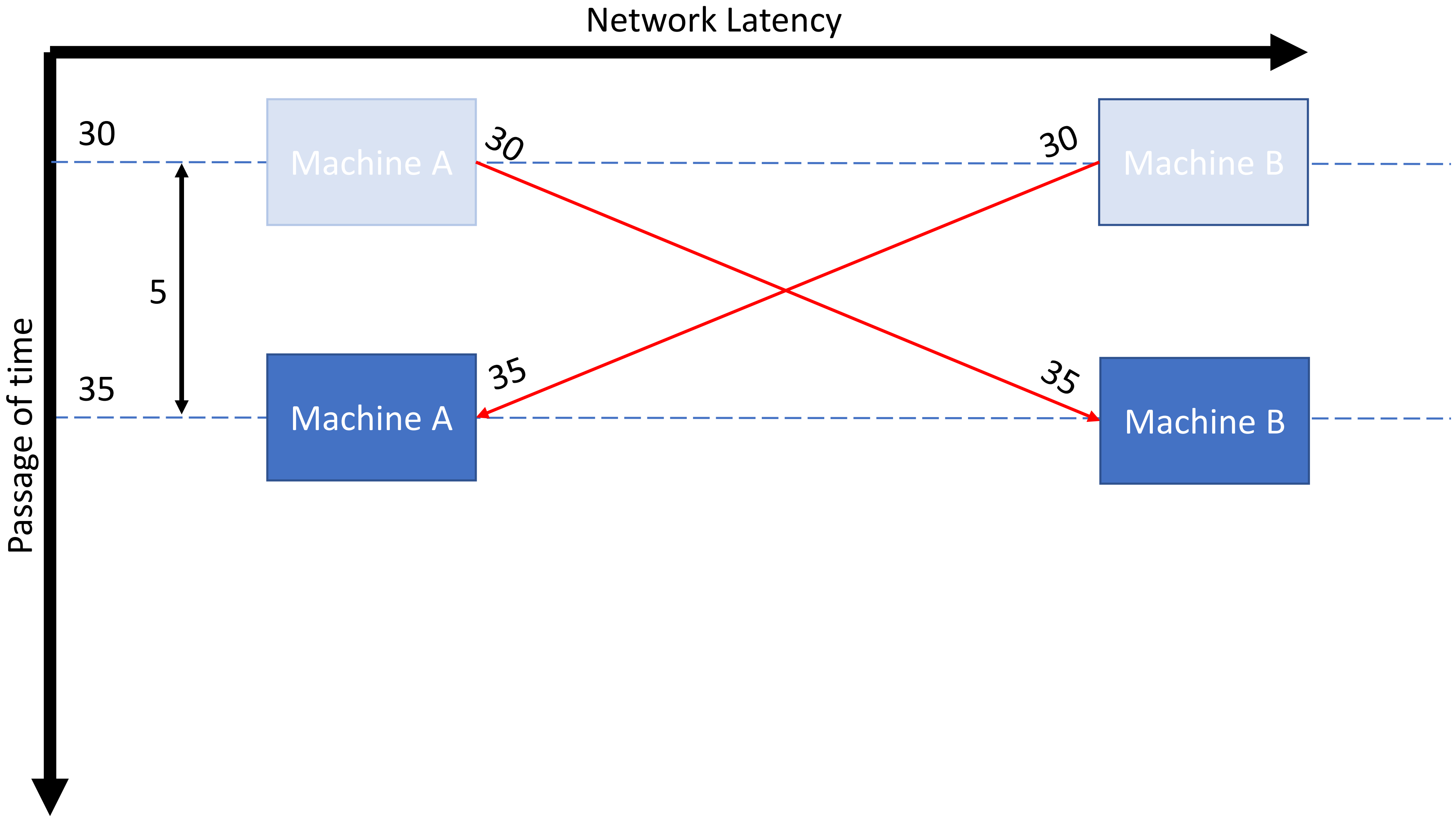
Speeding Machines Up (Class 3)



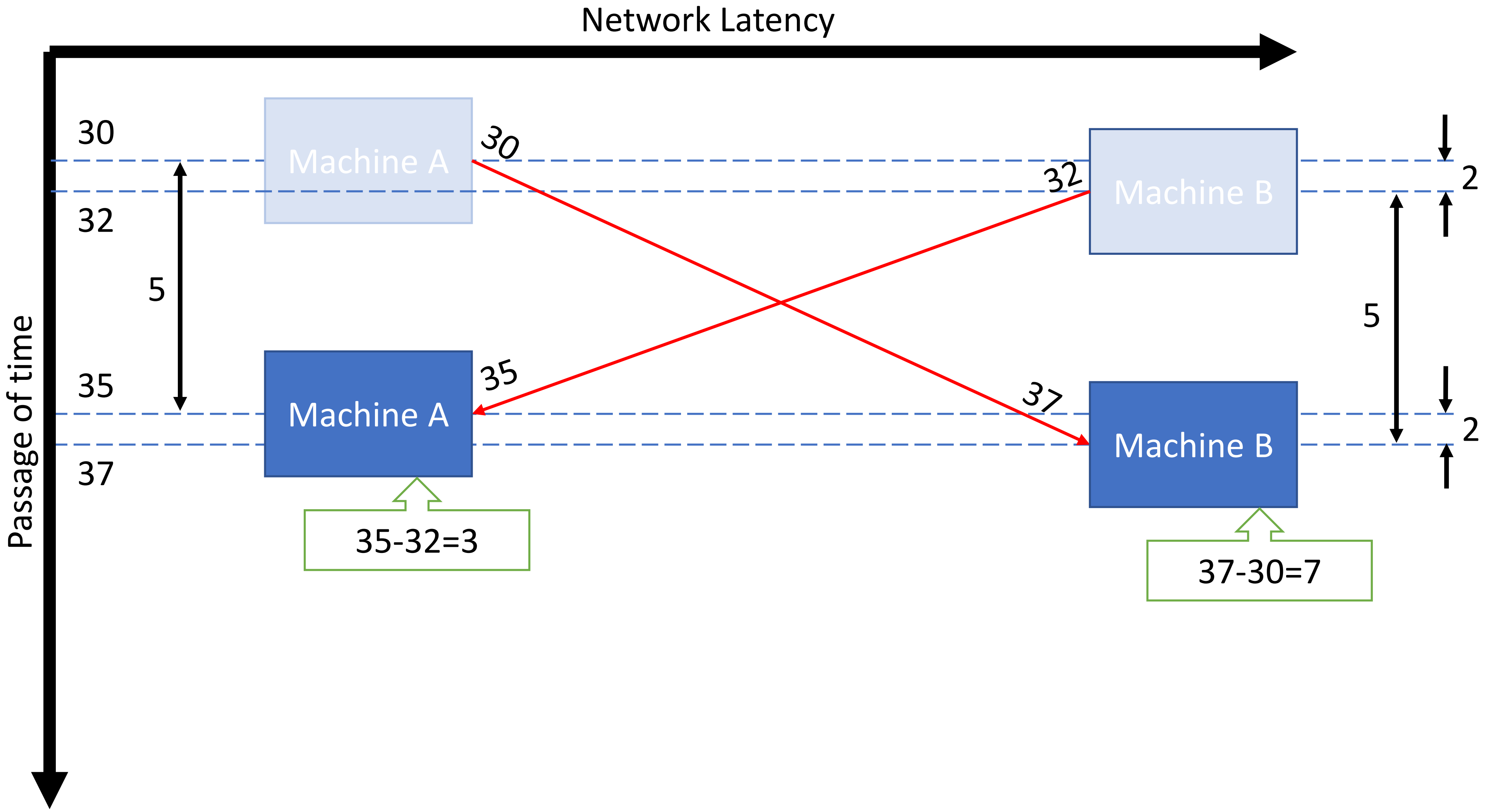
Distributed Databases (Class 1)

- Consistency vs Availability
- Consistency using:
  - Handshaking (Paxos)
  - Moving from Logical Clock to Physical Clock
    - Commit and wait
  - With Precise Clocks
    - Commit zero wait (CAL theory)

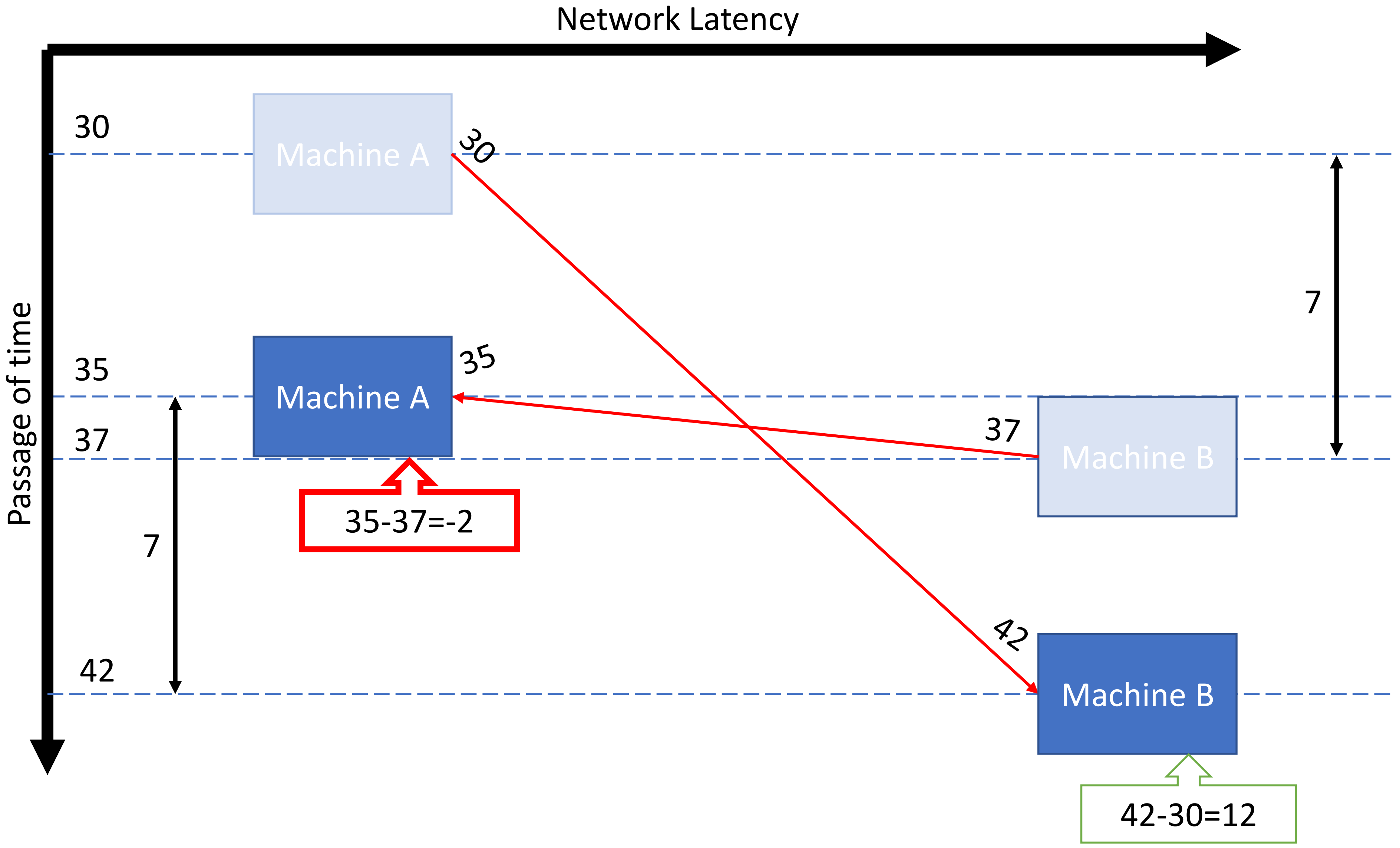
Linearizability and clock skewness



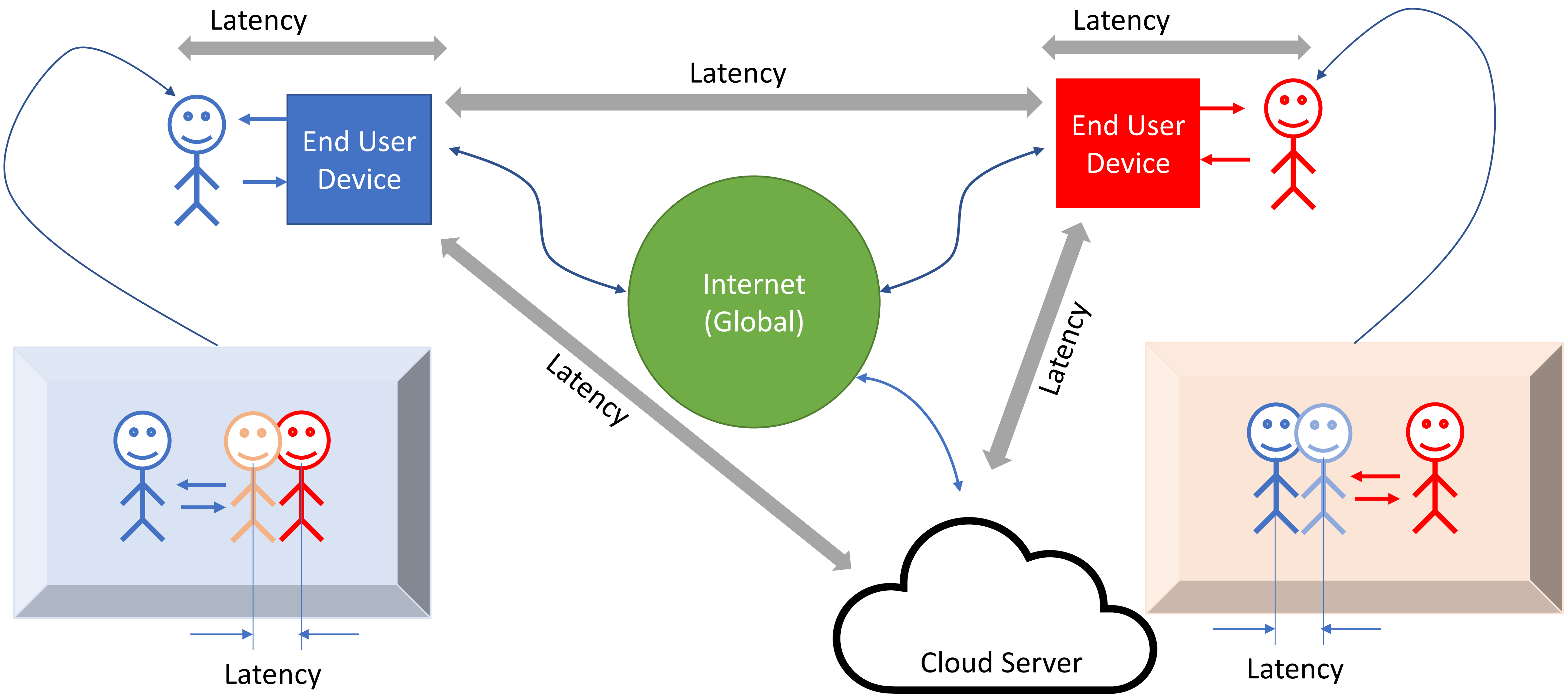
Linearizability and clock skewness



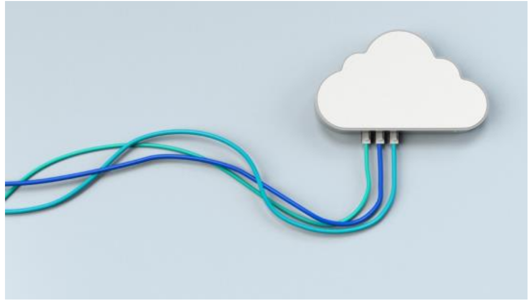
Linearizability and clock skewness

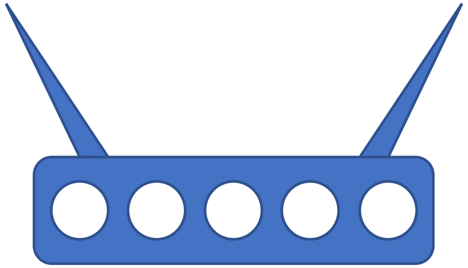


Latency Challenges and End-Users (Class 1)



Latency: From Cloud to the End-User

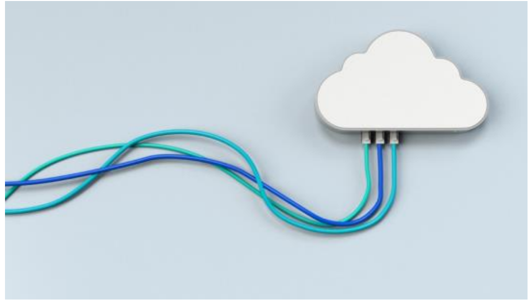


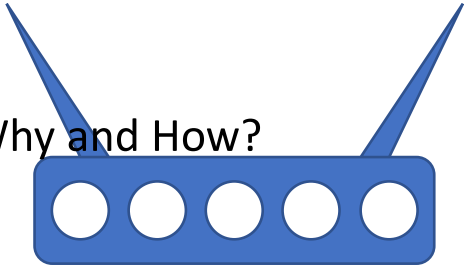


5G

Data Centers → Internet → Cable Modem to ISP → WIFI and Router → End-User

Latency: Cloud Servers



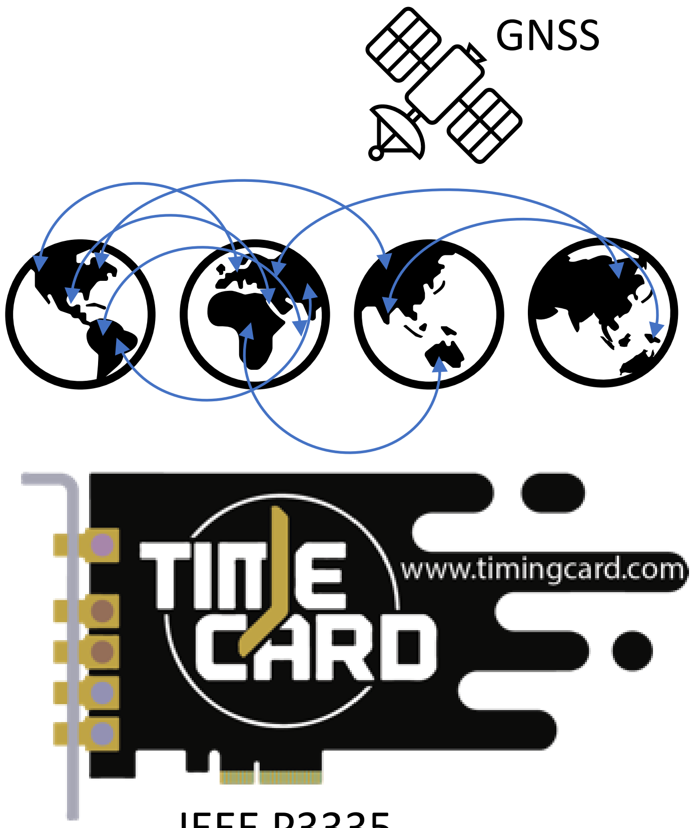

5G

Data Center → Internet → Cable Modem to ISP → WIFI and Router → End-User

Server → Leaf SW → Spine SW → DCI → DFC → Core Network

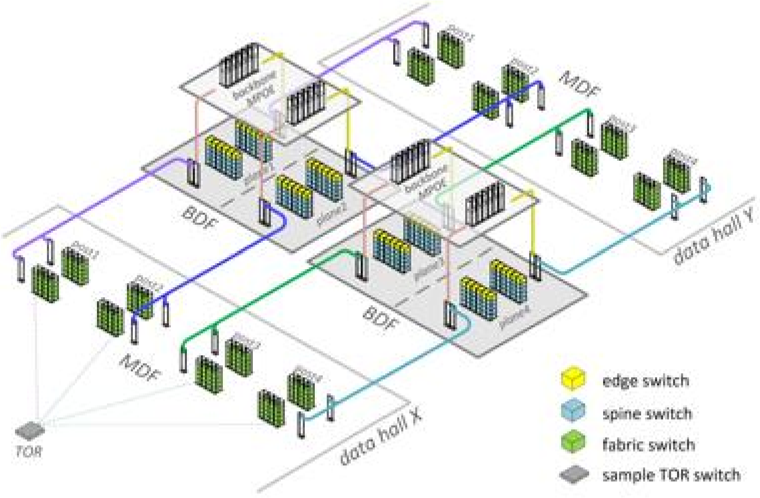
How to Provide Precision Time Sync in DCs

GNSS

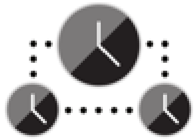


IEEE P3335

Global



PTP

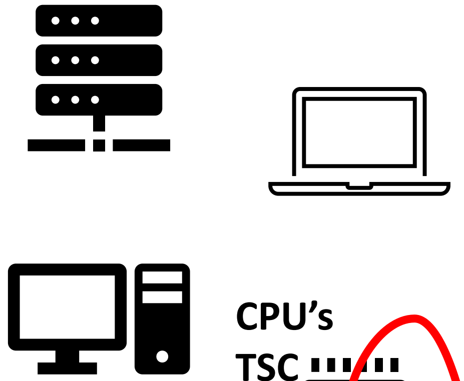


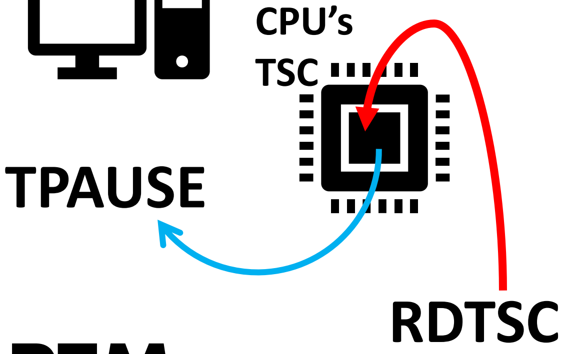
Data Center IEEE 1588

Local Network

RDTSC — Read Time-Stamp Counter

Opcode*	Instruction	Op/En	64-Bit Mode	Compat/Leg Mode	Description
0F 31	RDTSC	ZO	Valid	Valid	Read time-stamp counter into EDX:EAX.

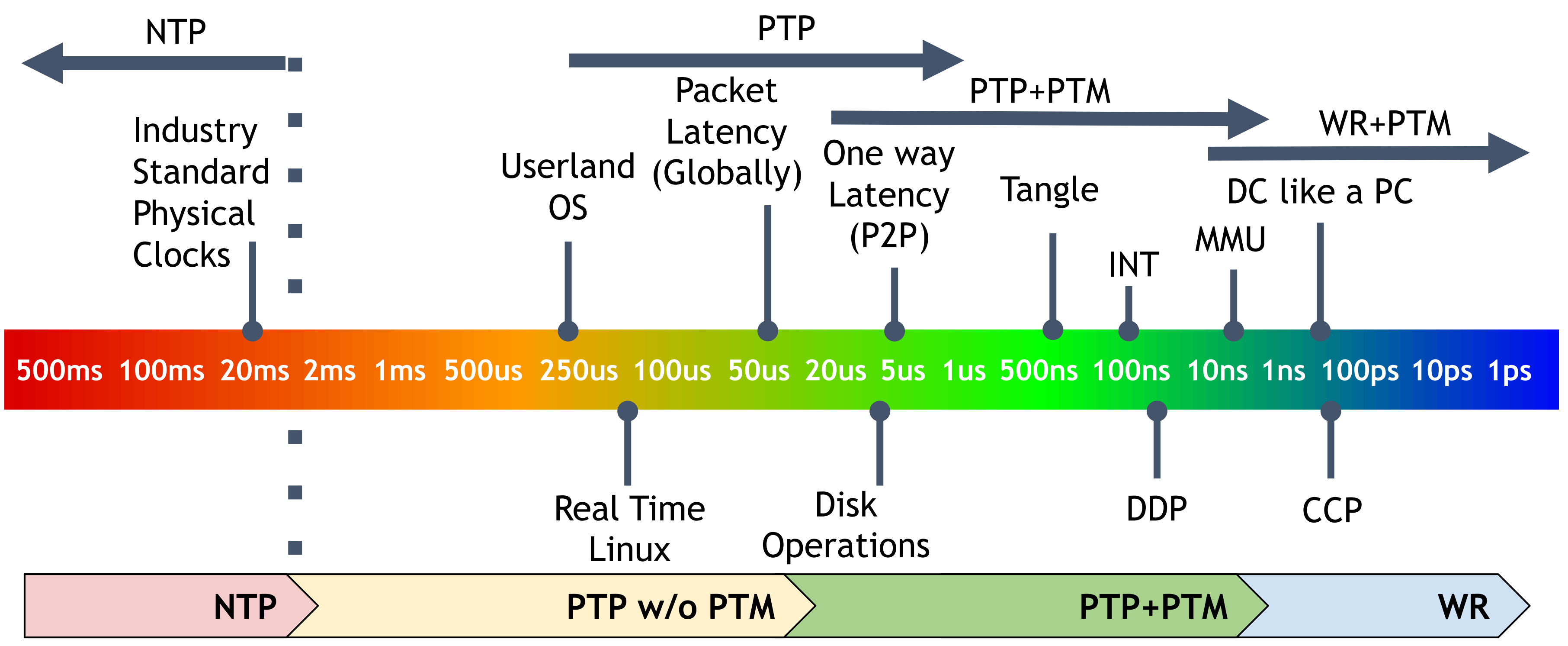




PTM

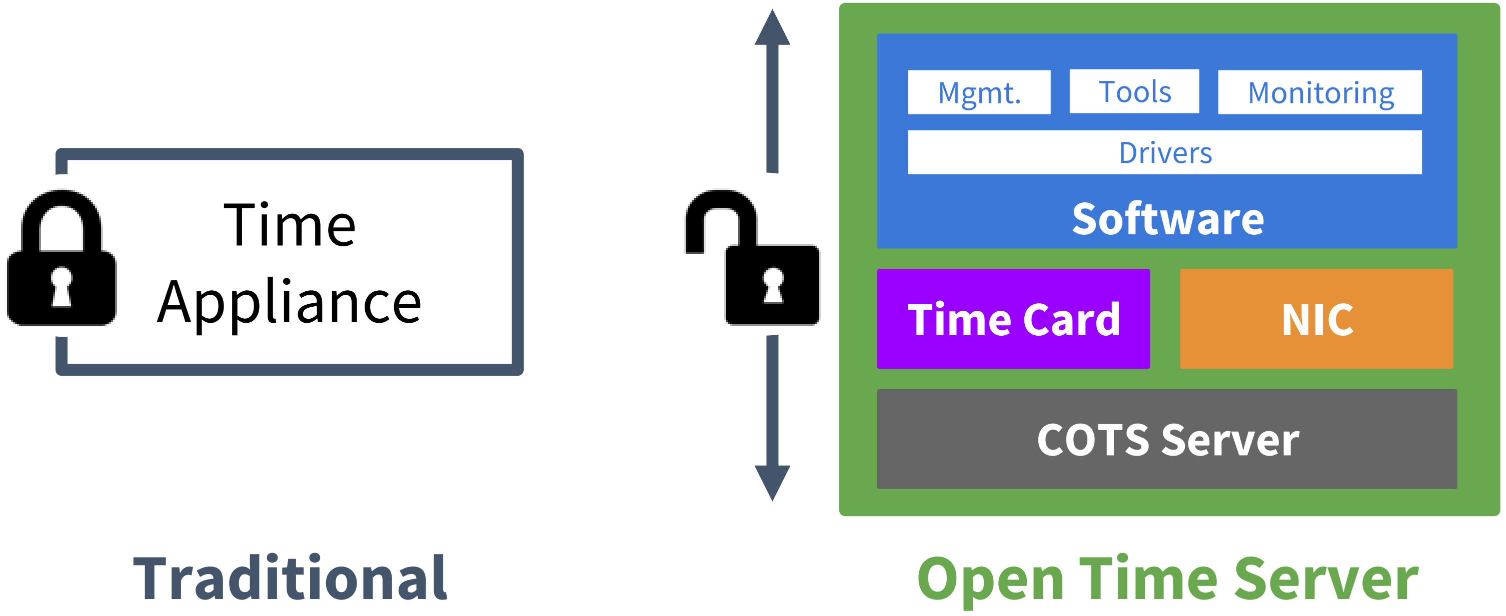
Inside Machines

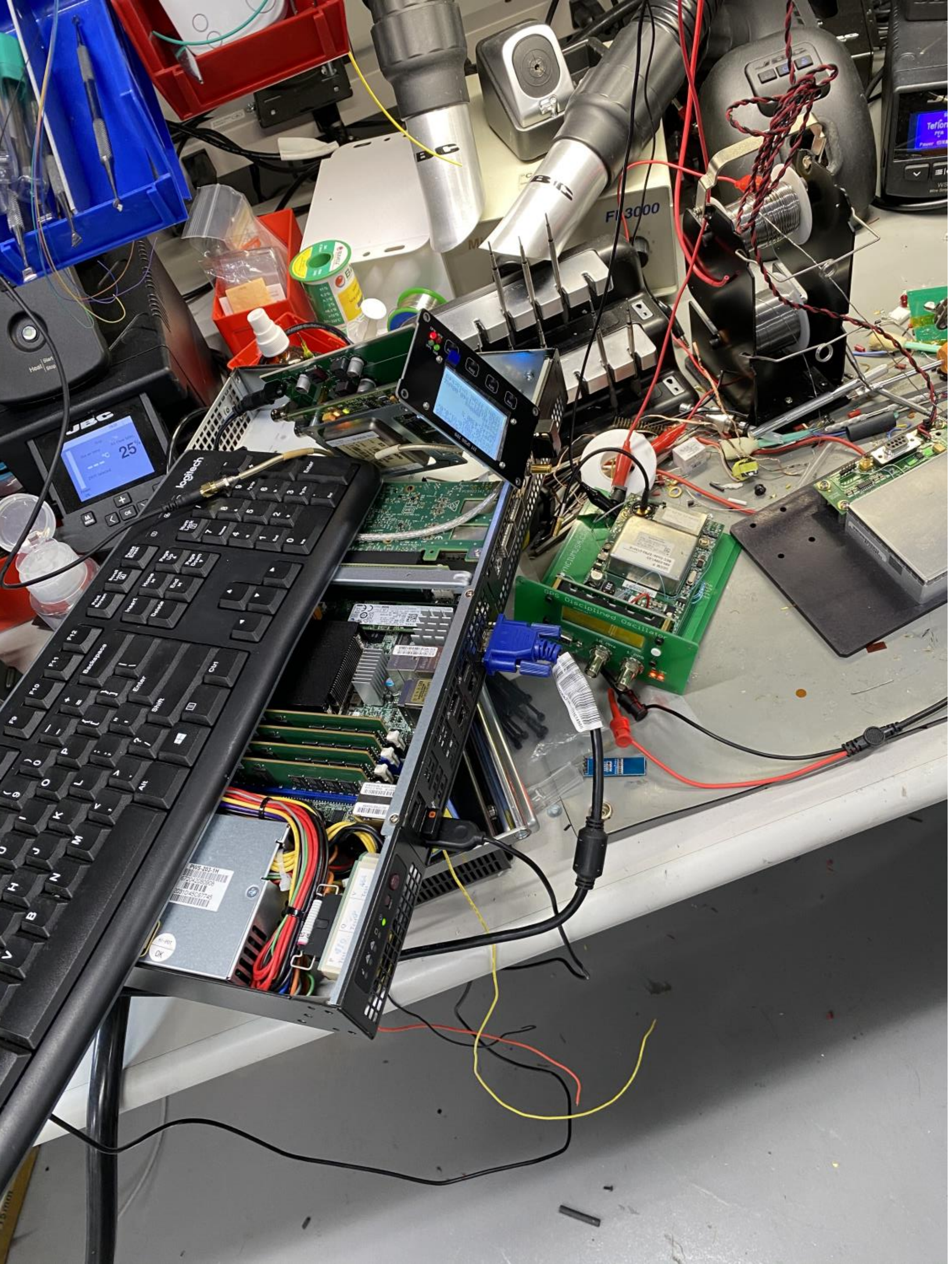
Time Precision and Applications Roadmap

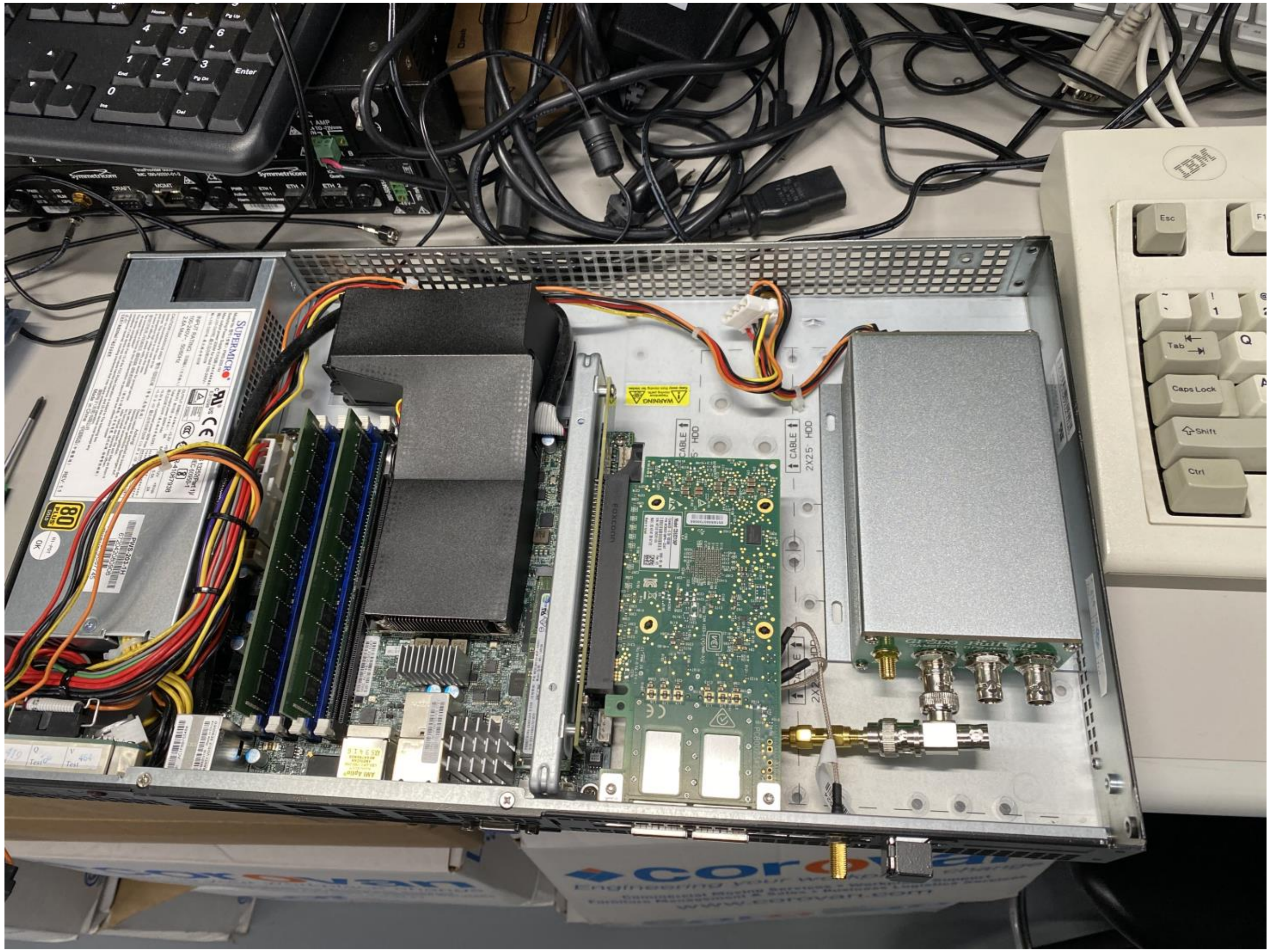


Open Time Server

How to sync a Datacenter?

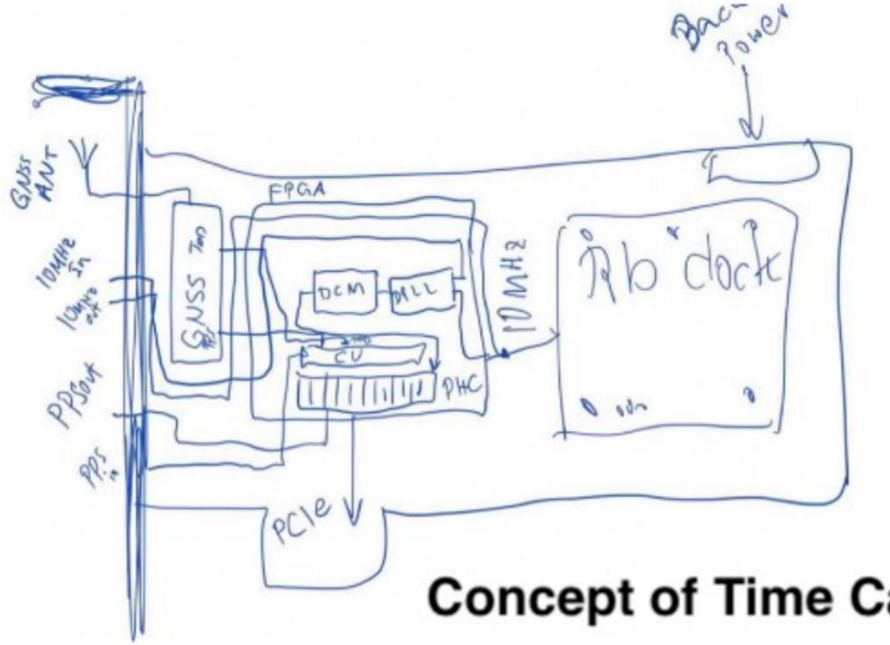



JBC
FI 3000
25
logitech

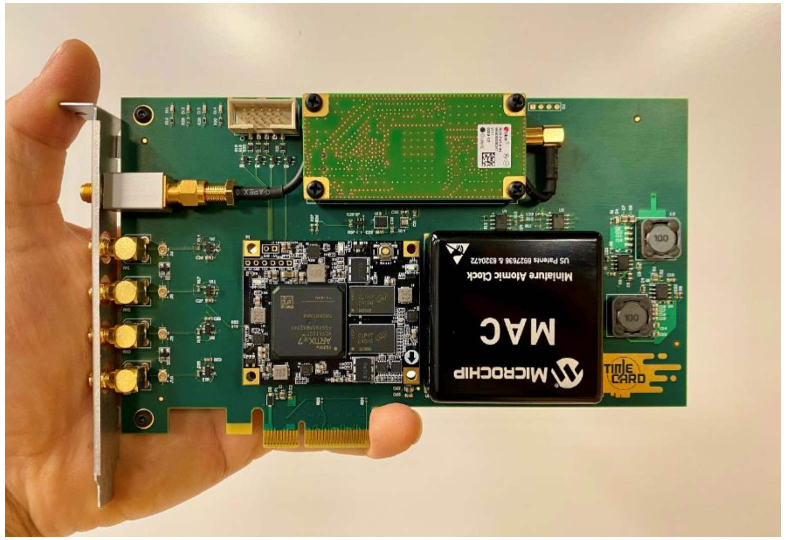

SUPERMICRO
IBM
Esc
Tab
Caps Lock
Shift
Ctrl
Enter
WARNING
CABLE
2X25 HDD
corovan

Time Card

- Concept (2020)
- First Prototype (2021)
- Industry Adoption (2022)



Concept of Time Card



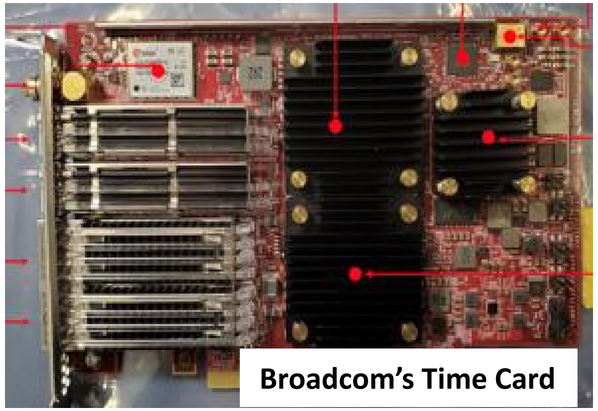
Time Card Family

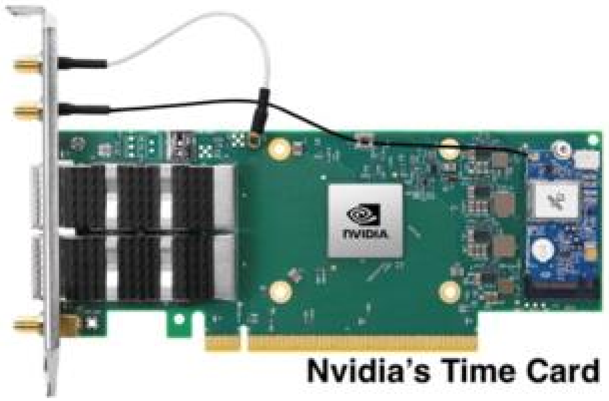
Orolia's Time Card


ADVA's Time Card

Celestica's Time Card

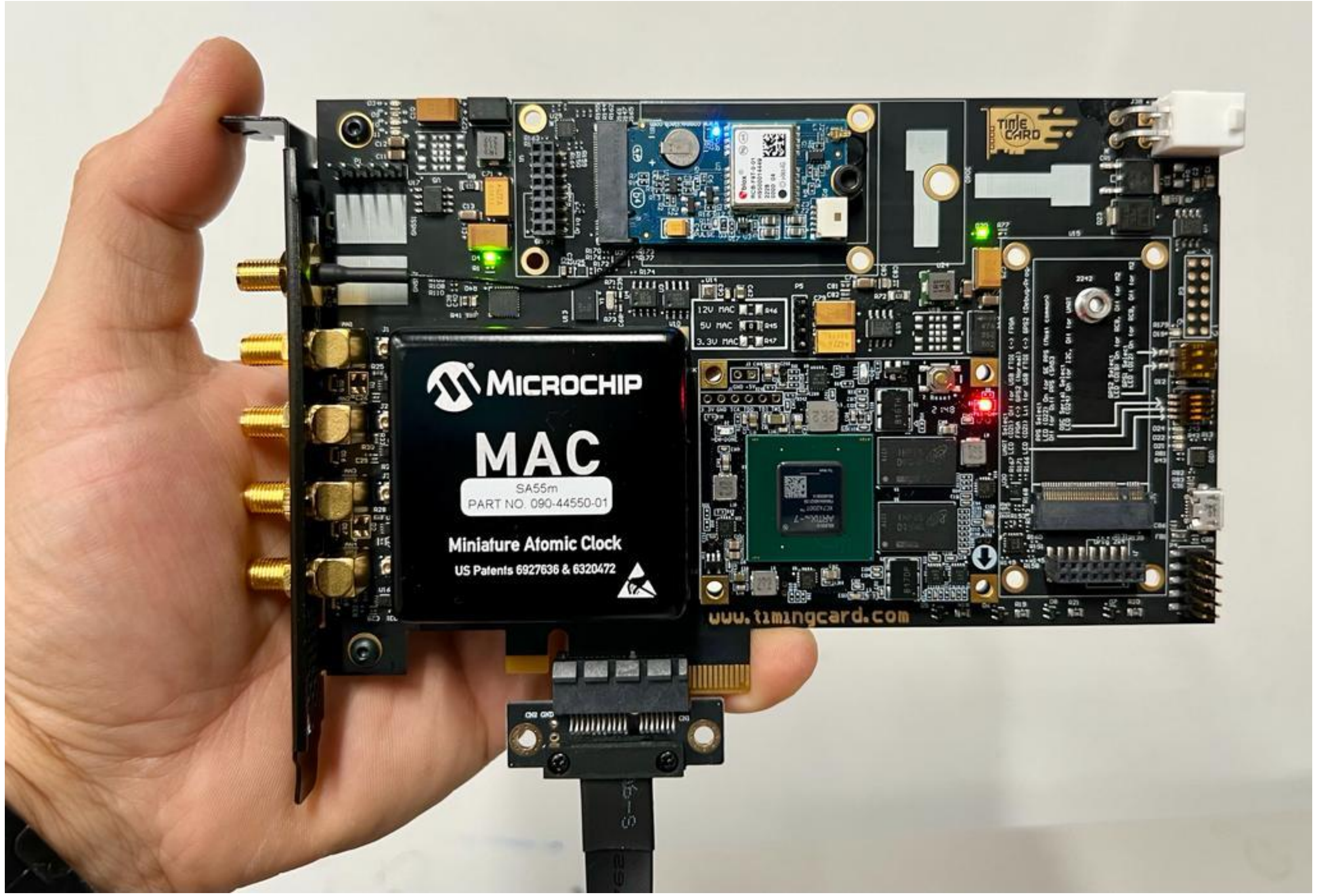
Intel's Time Card


Broadcom's Time Card



Nvidia's Time Card

Latest Time Card



Conclusion

- In distributed databases
  - As we scale, consistency becomes harder (handshake based)
  - Moving from logical clock to physical clock
  - CAL theory applies (diminishing returns for clock skewness lower than minimum latency)
- Distributed AI systems
  - Association of logs
  - Time Division of Power Spikes
  - Heterogeneity and runtime calibration

Thank you

Find out more on:

www.ocptap.com
www.timecard.ch
www.timingcard.com
www.opentimeserver.com