End User Synchronization

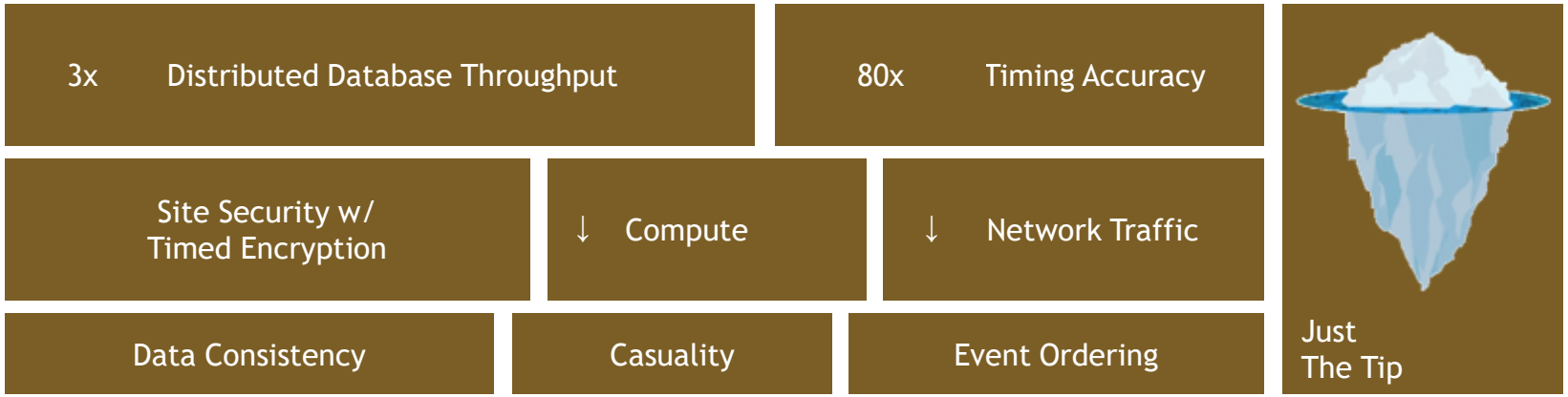
Connect. Collaborate. Accelerate.

Why Do We Need Synchronization?

"Nanosecond-level clock synchronization enables a new spectrum of timing and delay-critical applications"

-- *Google, Stanford, Exploiting a Natural Network Effect for Scalable, Fine-grained Clock Synchronization*

A Precise Time Axis leaps applications' performance, efficiency and security

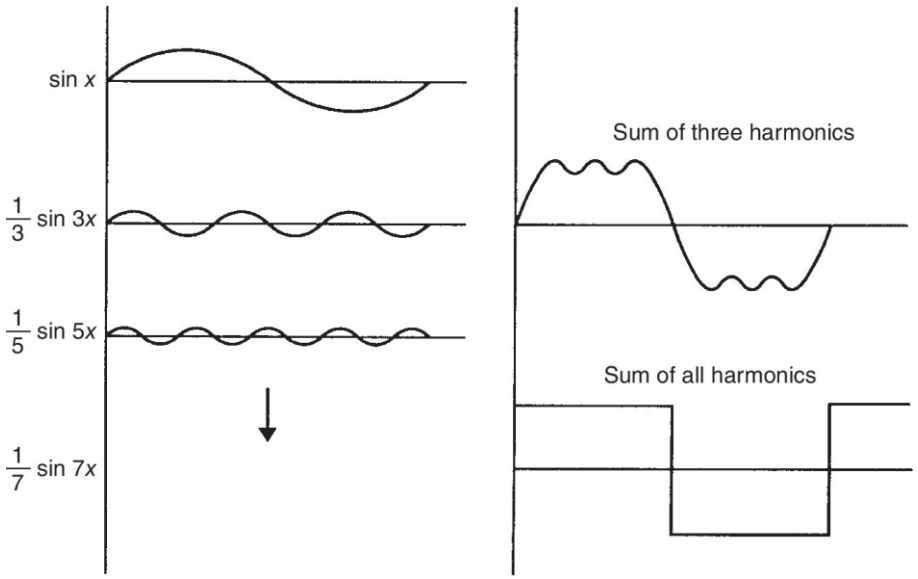


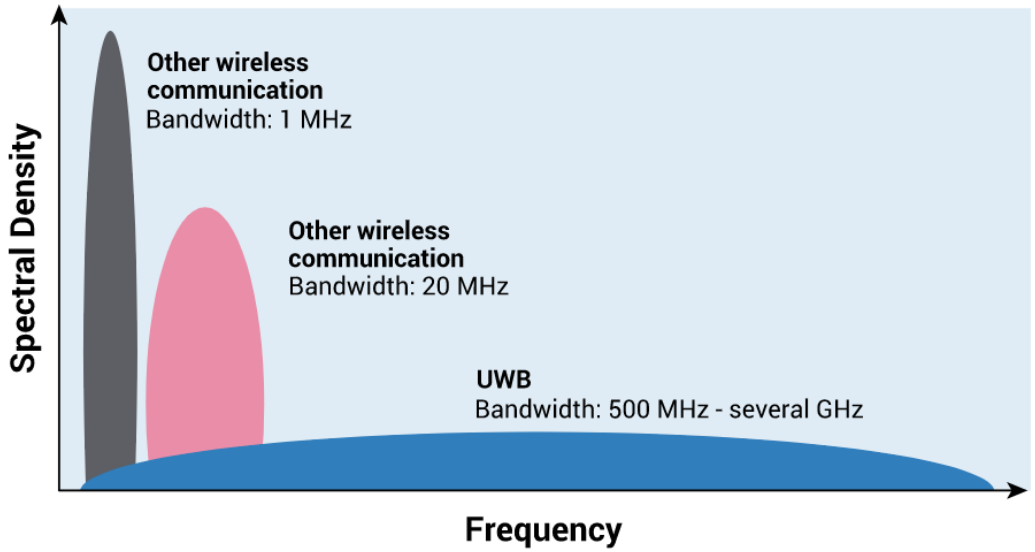
Use Case: Mass Online Platforms

- Competitive interactions, like gaming or stock transactions
  - Simulating physical reality, have instantaneous actions occurring in multiple simulations on remote machines
  - Timestamp events occurring on remote machines, and do judgement at central server
  - Generates race conditions but want to be fair regardless of latency to the server
- Benefit:
  - Arbitrate causality between remote inputs

How to sync end users

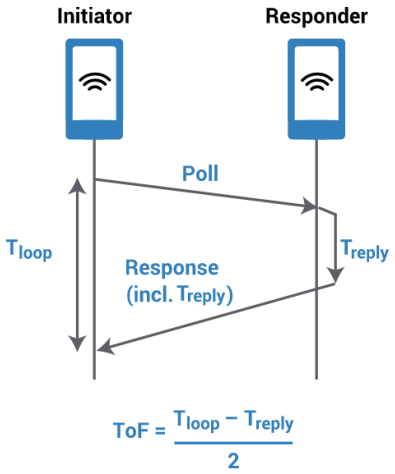
- UWB (Ultra-Wideband)
  - IEEE802.15.4-2011 / IEEE 802.15.4z-2020
  - Large bandwidth approximates ideal square wave better with sharper edges
  - Sharp edges allow precise timestamping of packet reception and transmission

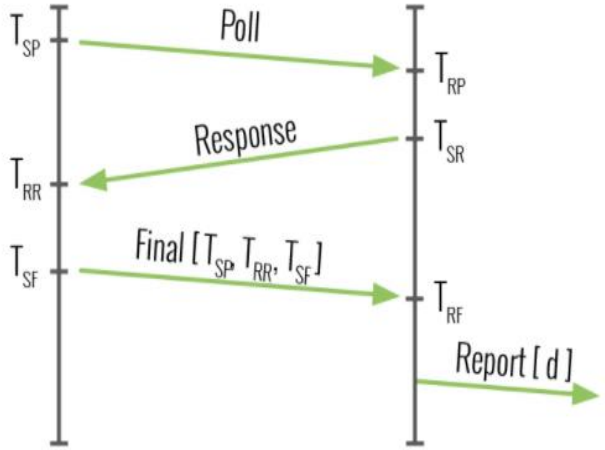




Traditional UWB application

- Ranging
  - UWB devices have a 40-bit counter running at ~64GHz , one tick ~15 ps
  - Each packet sent or received is timestamped using this counter
  - Range between devices calculated using Time of Flight * Speed of Light
  - Accuracy degrades as response time increases, each end has non-ideal clocks, so time calculations vary with clock frequency variation





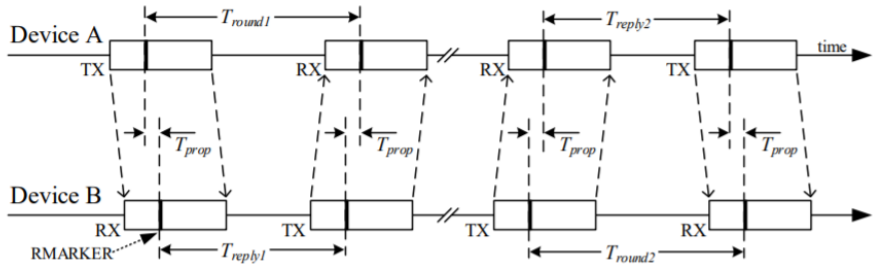


Figure 37: Double-sided two-way ranging with four messages

Commercial UWB

- Apple Airtags
  - Use UWB and BLE together to find the distance and angle to the tag from your phone

- Samsung SmartTag
  - Similar to Apple, uses UWB and BLE together with phone's camera to show you where the tag is

User Synchronization

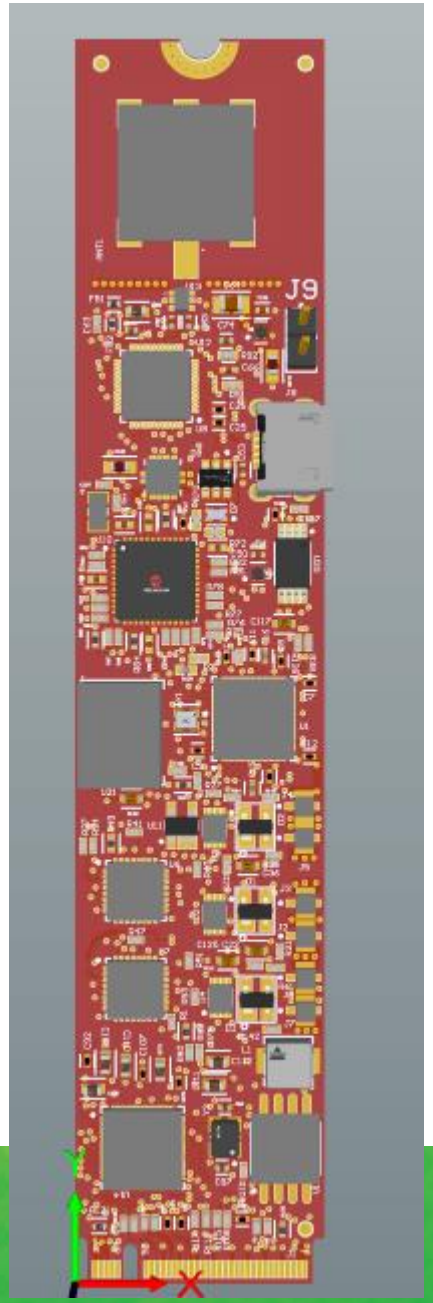
- Synchronize multiple devices to a single “gateway” with GPS
  - Provides GPS synchronization to locations without GPS reception
- Ideally, this “gateway” function would be built into WiFi routers, or standalone device
  - Users could install as needed per household, synchronize every device

GPS Ultra-wideband Gateway (GUG)

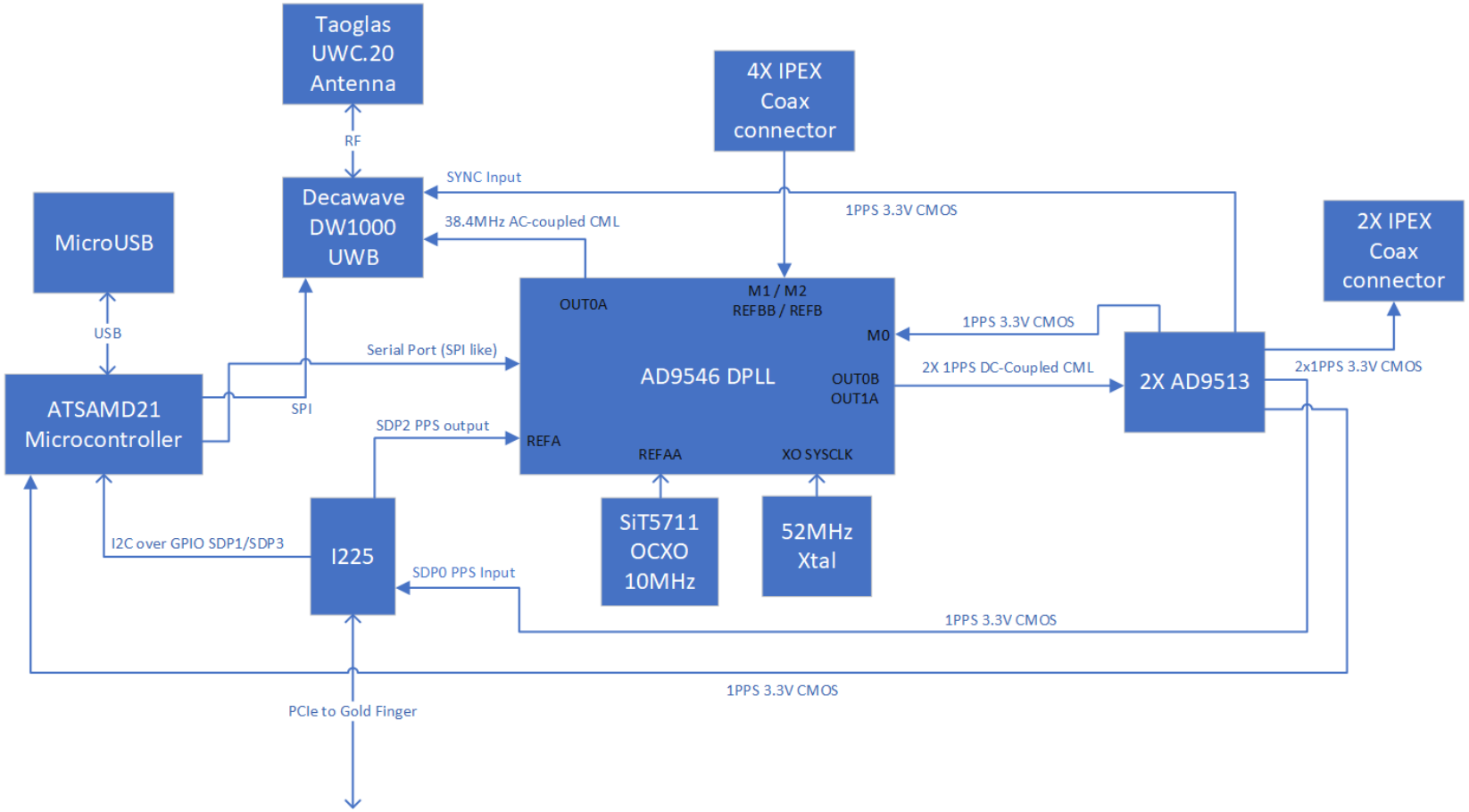
UWB Endpoint (M.2 design, Time Drive)

Time Drive

- First prototype of a UWB endpoint, M.2 formfactor
  - Intel I225 NIC chipset as PCIe endpoint
  - Decawave DW1000 as UWB chipset
  - Analog Devices AD9546 DPLL
    - Frequency and phase control, timestamping input, 1PPS outputs
  - ATSAMD21 as onboard microcontroller
  - SiT5711 OCXO for stability
- Hardware design to be uploaded soon.
  - Please contact us if interested in assisting in development

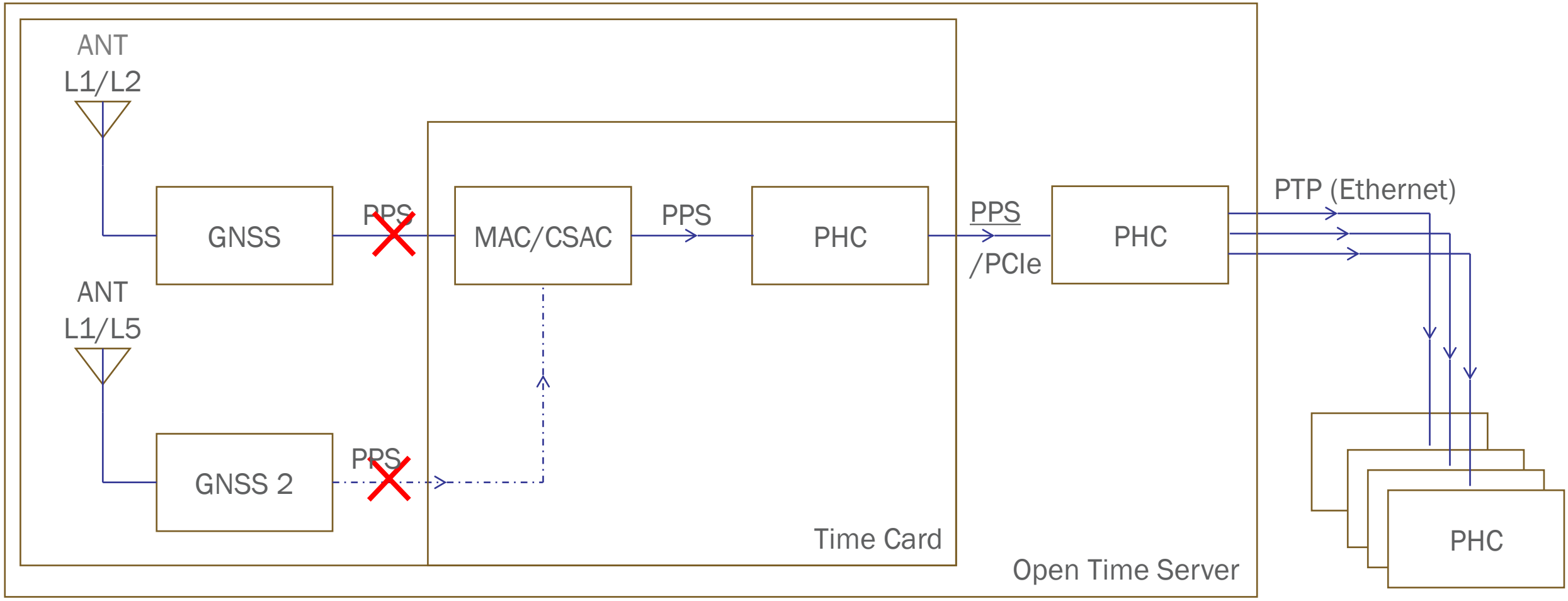


Time Drive Operation

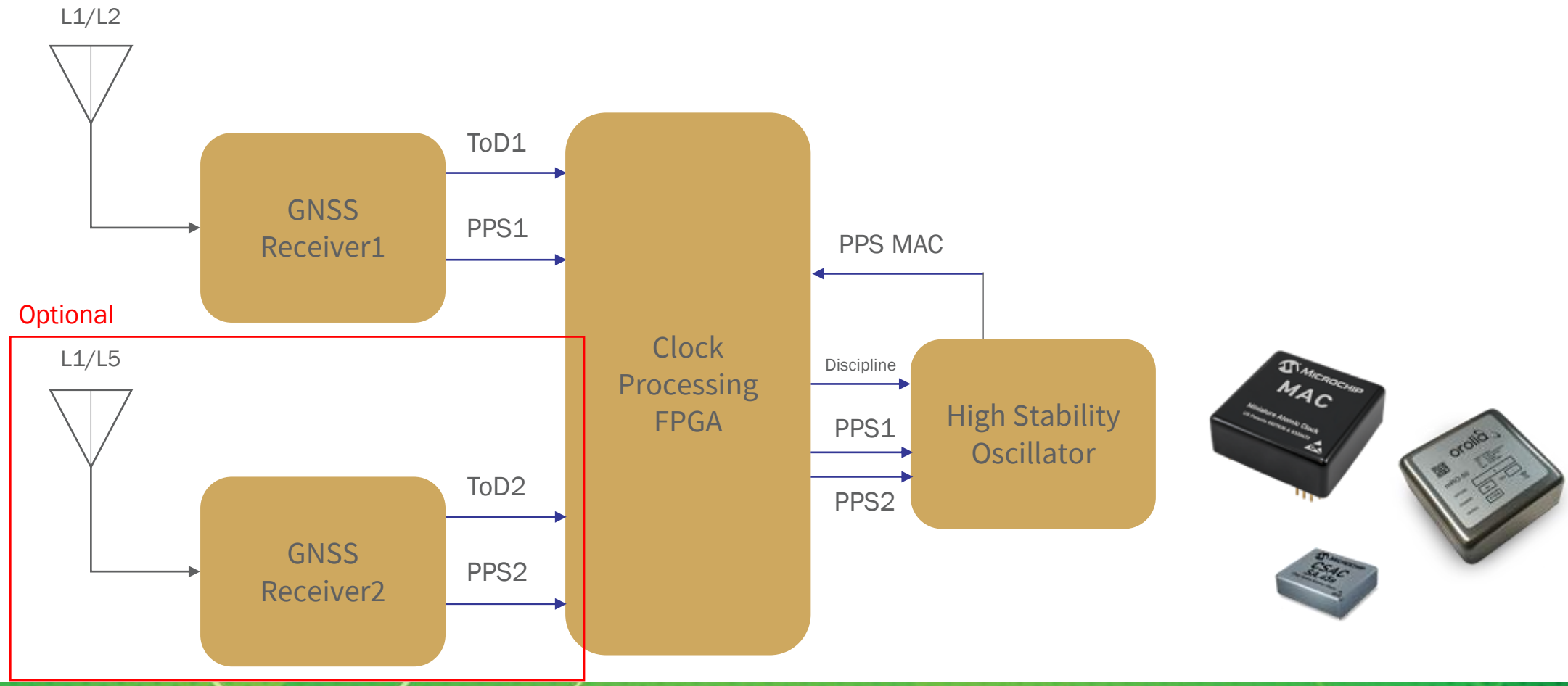


Thank You

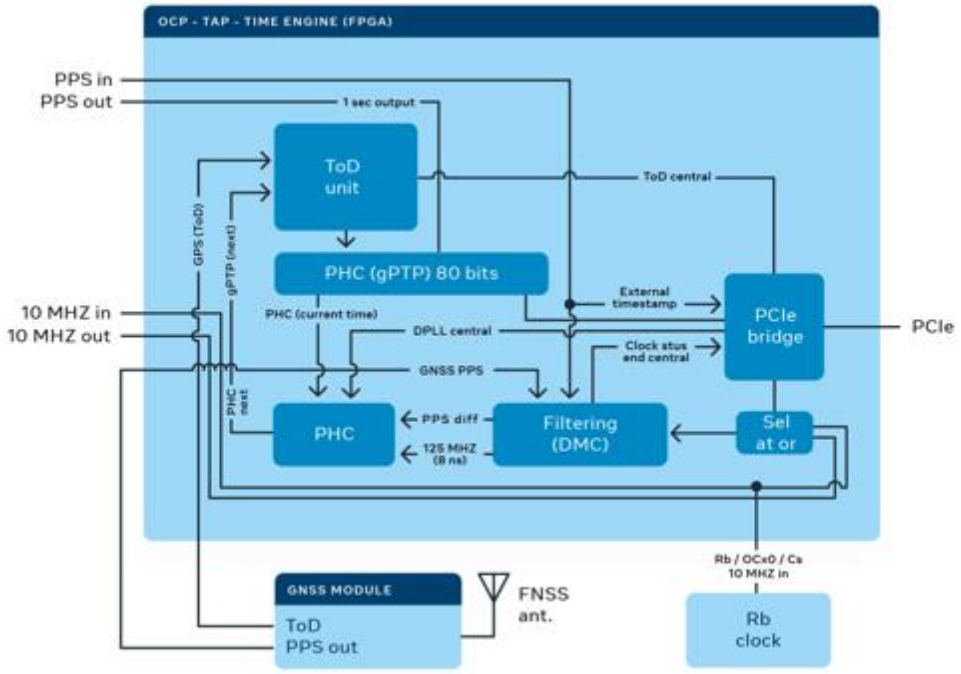
Control Flow In Holdover

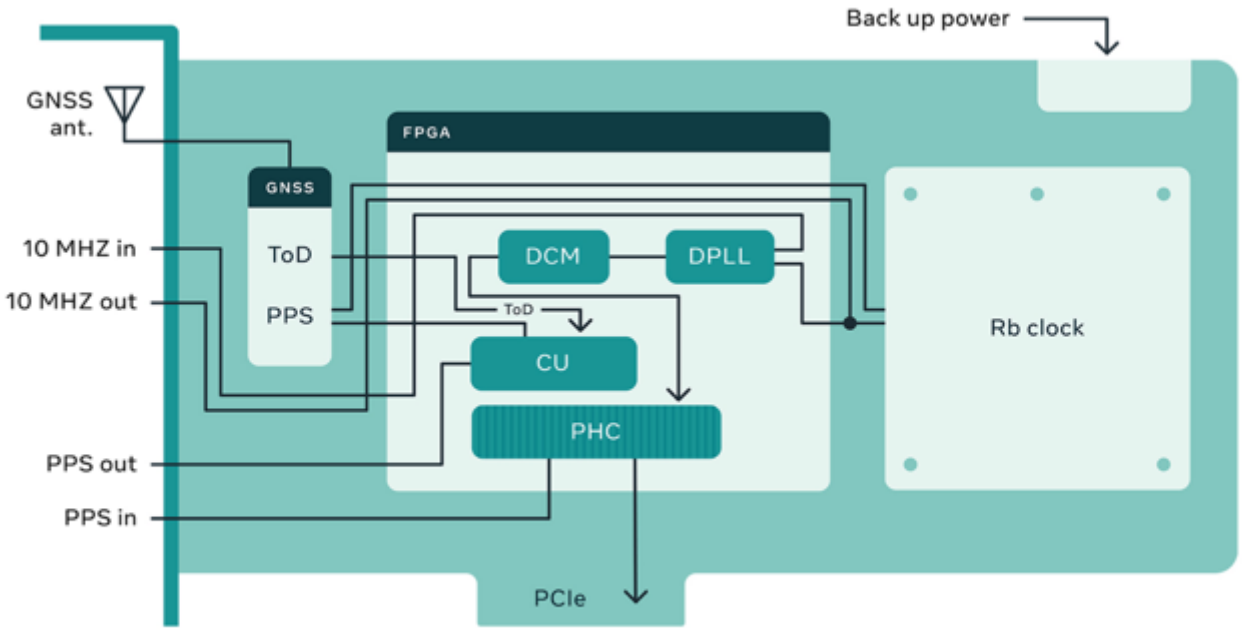


Advanced “GPSDO”



FPGA


OCP - TAP - TIME ENGINE (FPGA)
PPS in
PPS out
1 sec output
ToD unit
ToD central
GPS (ToD)
gPTP (next)
PHC (gPTP) 80 bits
External timestamp
PHC (current time)
PCIe bridge
PCIe
10 MHZ in
10 MHZ out
DPLL central
Clock stus end central
GNSS PPS
PHC next
PHC
PPS diff
125 MHZ (8 ns)
Filtering (DMC)
Sel at or
Rb / OCxO / Cs 10 MHZ in
GNSS MODULE
ToD
PPS out
FNSS ant.
Rb clock


Back up power
GNSS ant.
FPGA
GNSS
ToD
PPS
10 MHZ in
10 MHZ out
DCM
DPLL
ToD
CU
PHC
Rb clock
PPS out
PPS in
PCIe

How to use it

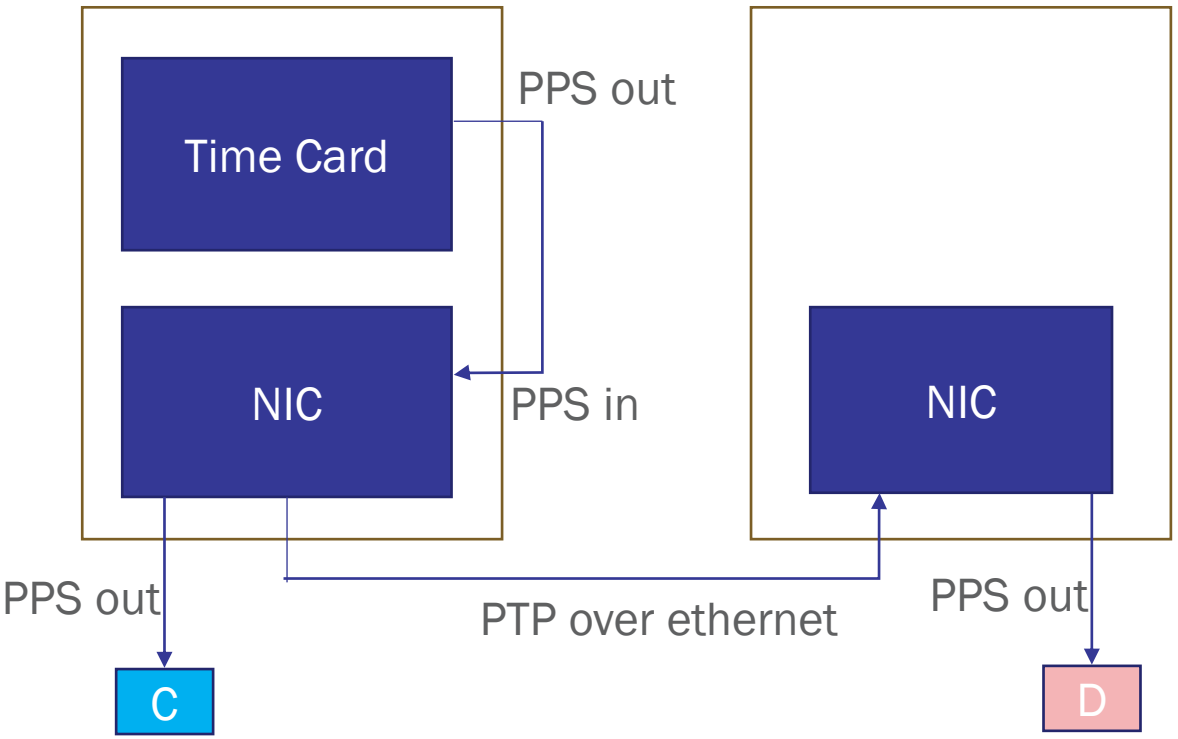
For PTP Time Server (Grandmaster)
- Via PCIe (PHC2SYS)
- Via PPS (TS2PHC)
- Via Reverse PPS (in progress)

For NTP Time Server
- Via PCIe (Chrony)

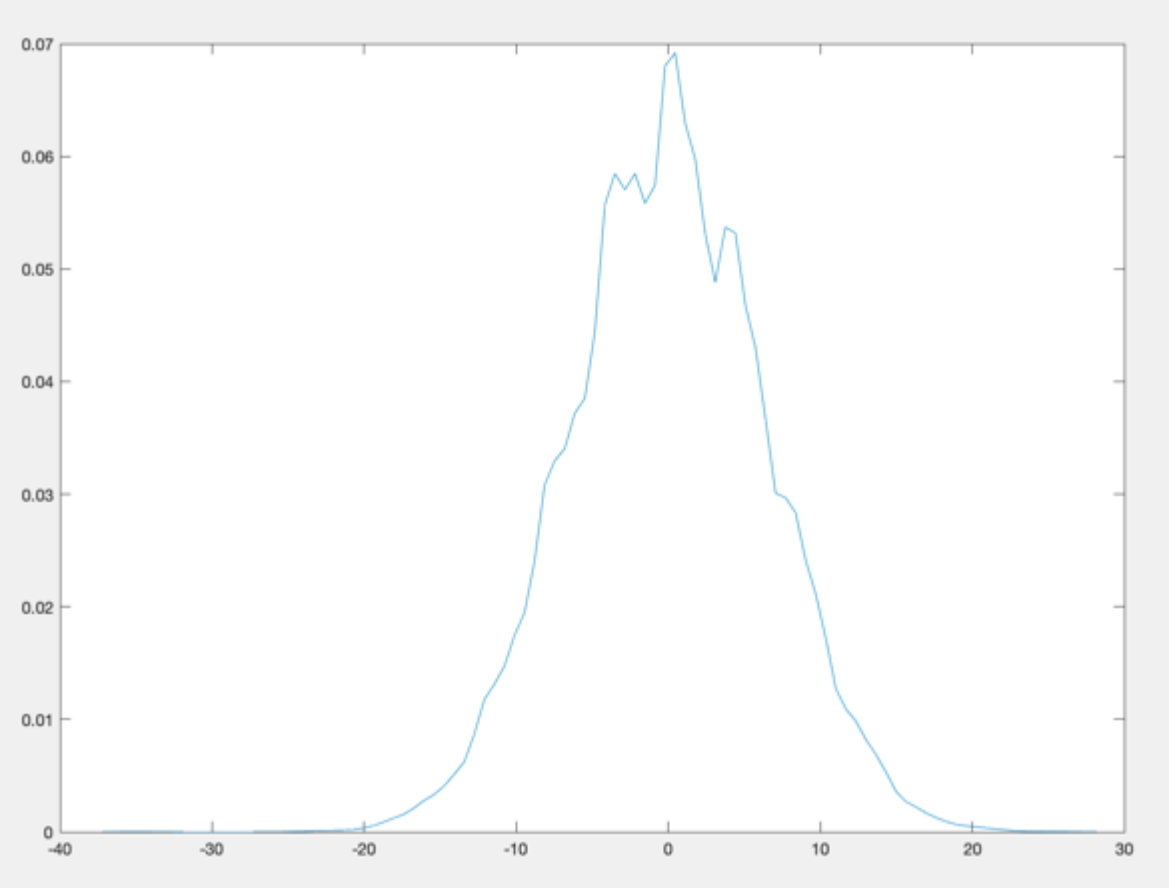
For Camera and Audio Systems
- IRIG-B
- PPS

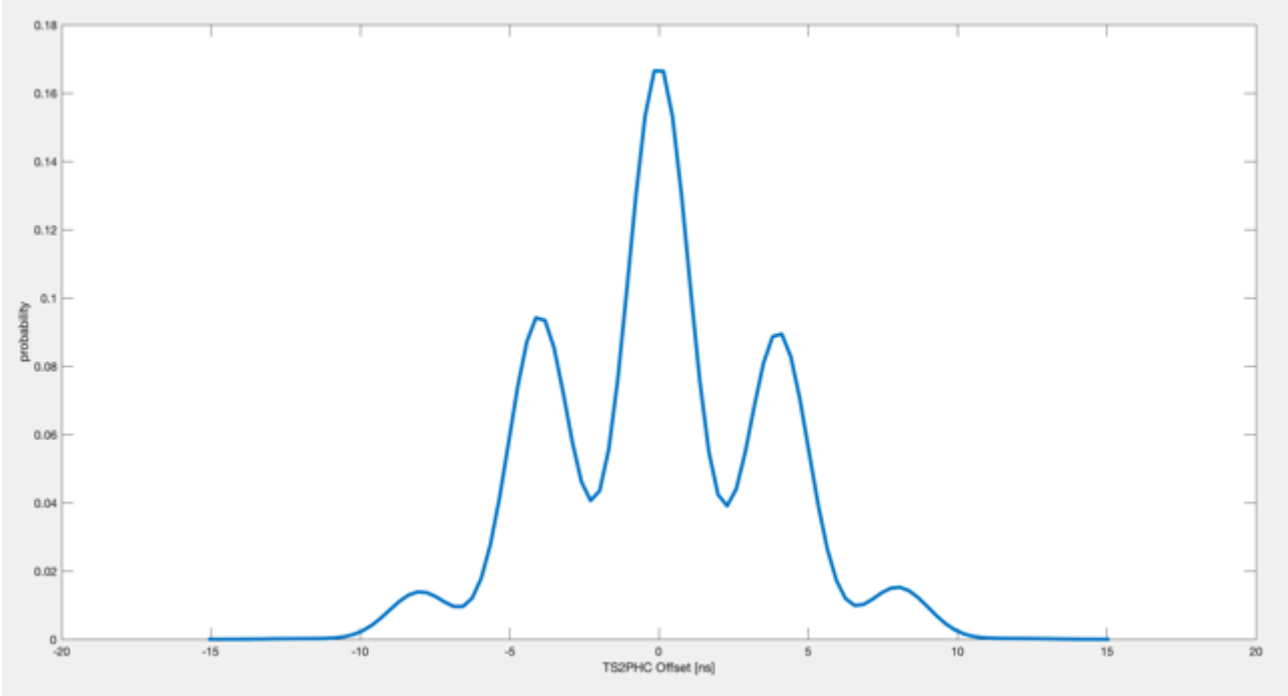
Other Applications
- Bipolar and Multi Channel TDC (Multiple PPS in)
- Over PCIe Event Time Stamping

Performance



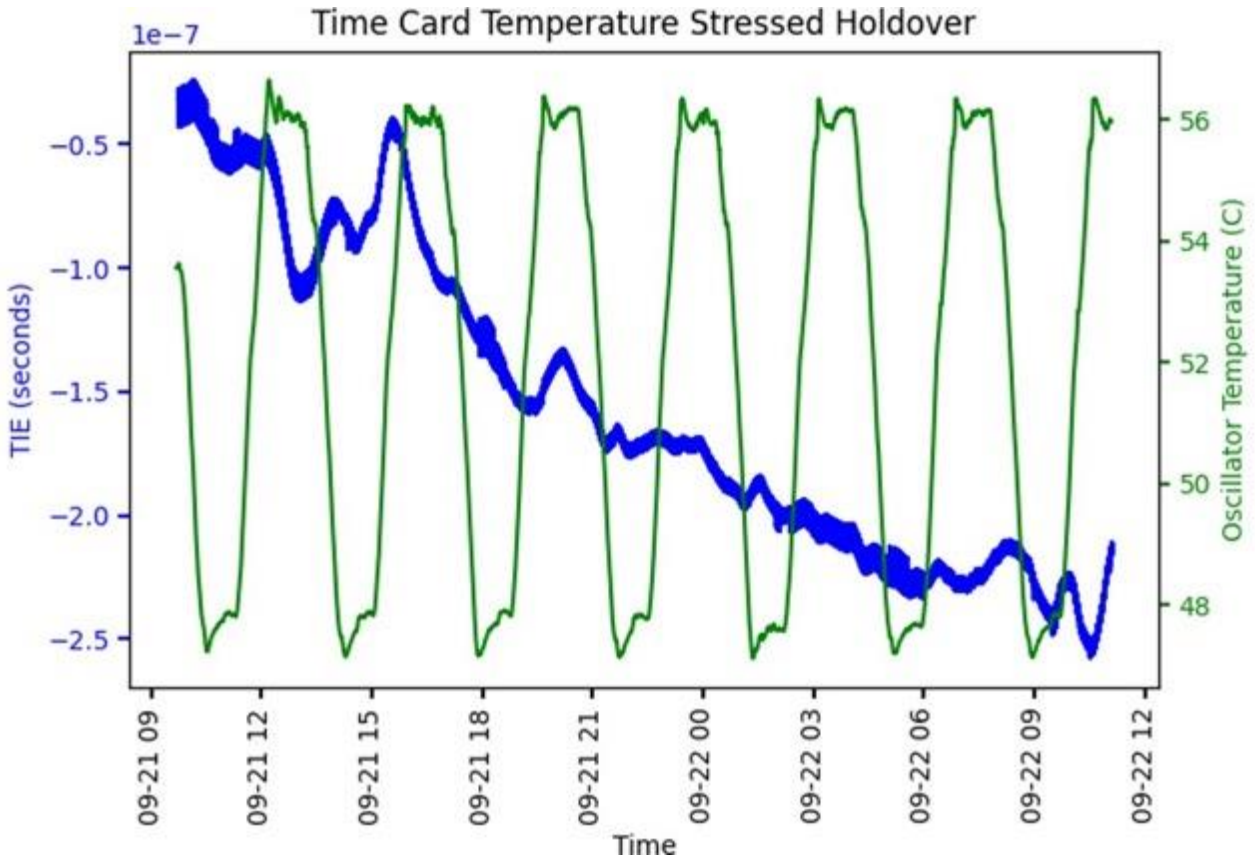
Performance



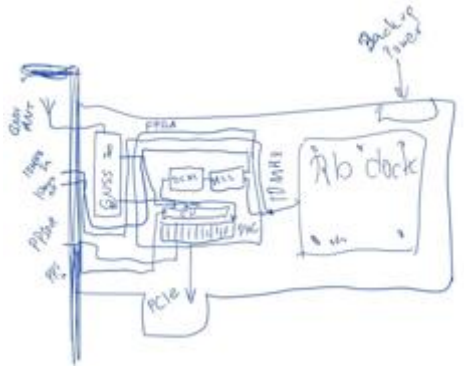


Long-Term vs Short-Term Stability

Performance



Previous Versions



Sep 2020

October 2020

November 2020

December 2020

February 2021

April 2021

March 2021

July 2021

Use Case: Network Telemetry

- Constantly pings machines
  - If machine doesn't respond, it must take an action.
- Why not do pings based on Hardware Timestamps
  - SING = Synchronous Pings
  - One way delay measurements
- In-Network Telemetry
  - Improve Congestion recognitions
  - Improve Congestion Control mechanisms
- End-to-End Precision: <100ns
  - Want to measure one way latency

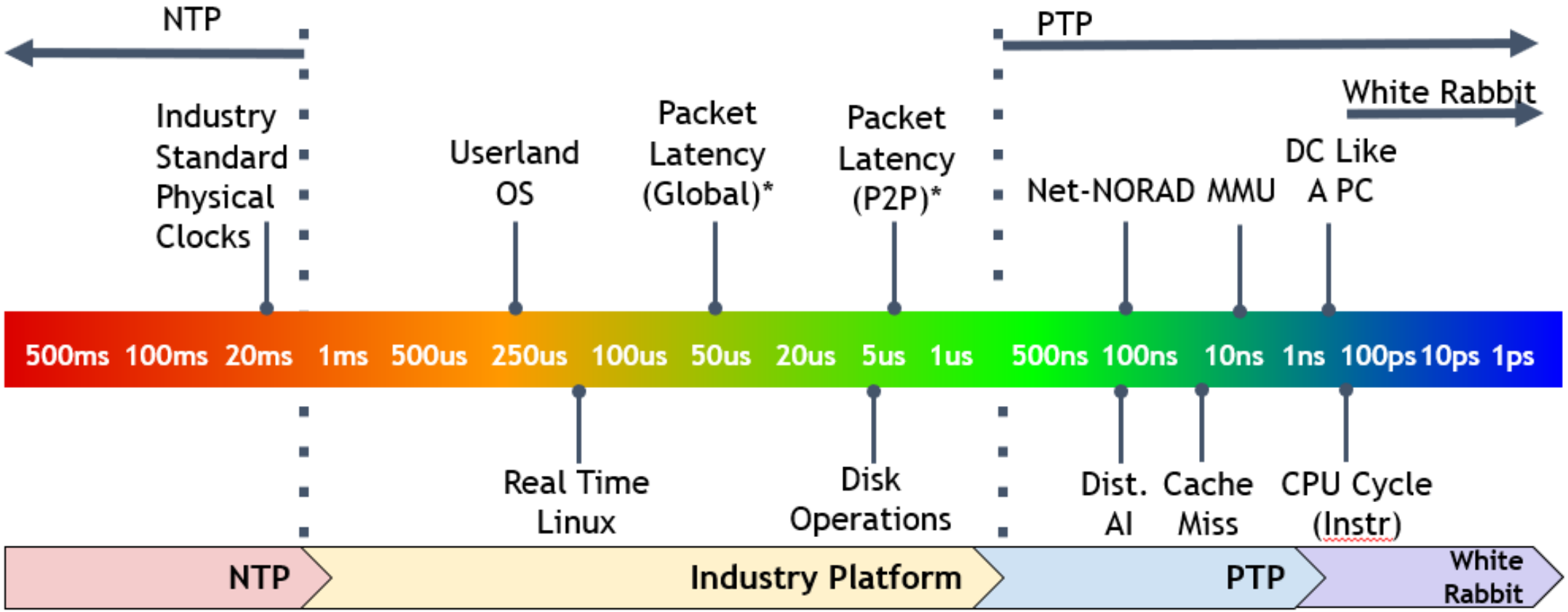
Use Case: Distributed AI

- Resource Intensive to move data to one machine or cluster
- With the right precision, you can train in many places
- Then use the timestamps to merge the results
- Advantages:
  - Reduces data center traffic/congestion
  - Save Resources
- Requires end-to-end precision of <100ns
  - Across the data center
  - Globally

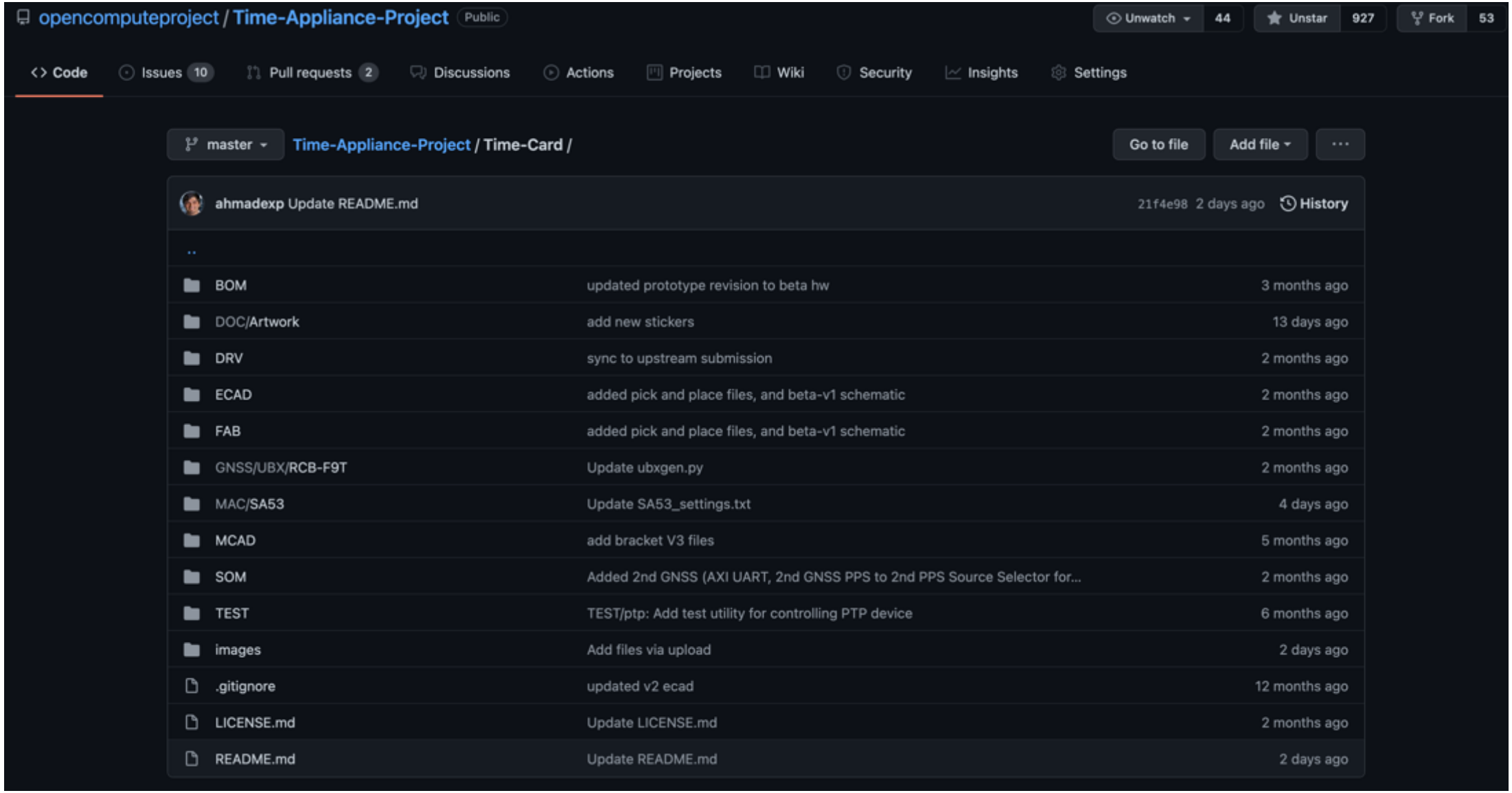
Use Case: Multicore Systems Across the Network

- Data Center Network is the Fabric
  - Ultra-Path Interconnect (UPI) over the network
  - Input-Output Memory Management Unit (IOMMU) over the network
- Can we program a DC like a PC?
  - We know how to program a Personal Computer well.
  - Precise time can help us program the Data Center Better
  - All DC equipment follows the same precise time vector
- Benefit:
  - Current data center loads are far from 100%
  - Determinism: If you know when everything happens, the load could be closer to 100%
- Requires End-to-End Precision of <10ns

Time Precision Today and Tomorrow



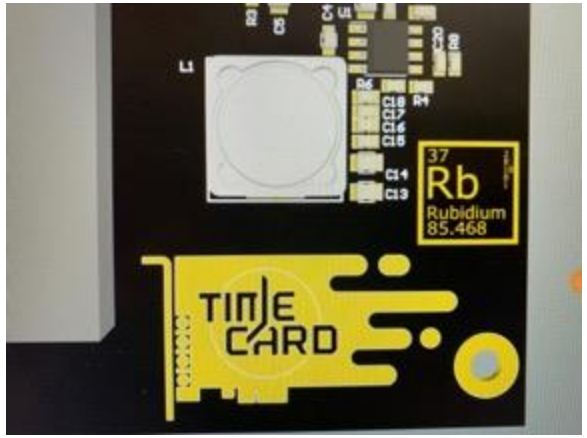
- Global – Data Center CPU to another Data Center CPU around the world
- P2P – CPU to another CPU in the same rack with minimum latency.



OPEN POSSIBILITIES

Coming Soon…

- Dual GNSS with Rb Clock (SA.53)
- Low Power with Cs Clock (SA.45s)



Can be found on www.timingcard.com