A flexible Time Server for Data Centers

Open Time Server and Time Card

Ahmad Byagowi
Ioannis Sotiropoulos

Motivation

NetTimeLogic GMBH

Meta

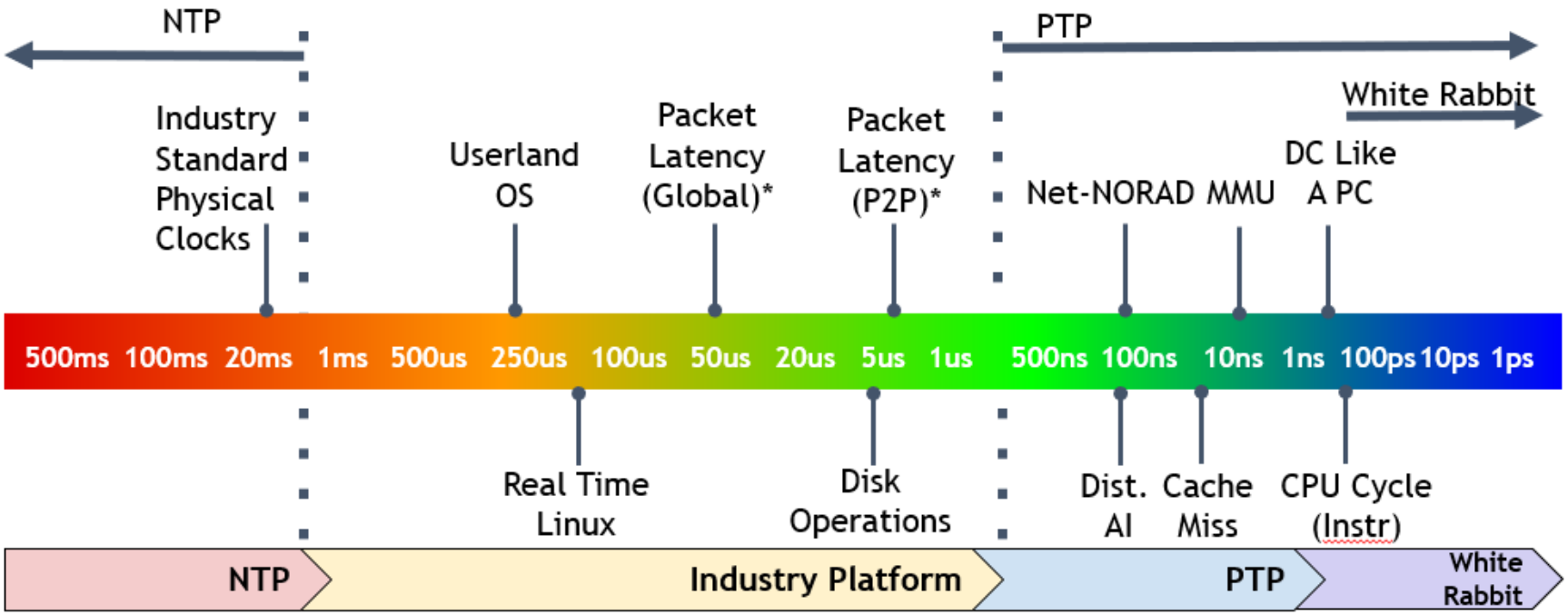
- As distributed system are getting more advanced (faster processor, larger data), the need for tighter timing is getting higher
- Existing methods to provide timing across the network are NTP, PTP and White Rabbit
- Off-the-shelf time servers are tailored to specific applications mainly driven by telecom
- Because the quantities are low, often R&D cycles are hard to justify. This will lead to less features or higher costs.

Time Precision and Applications

Meta

NetTimeLogic GMBH

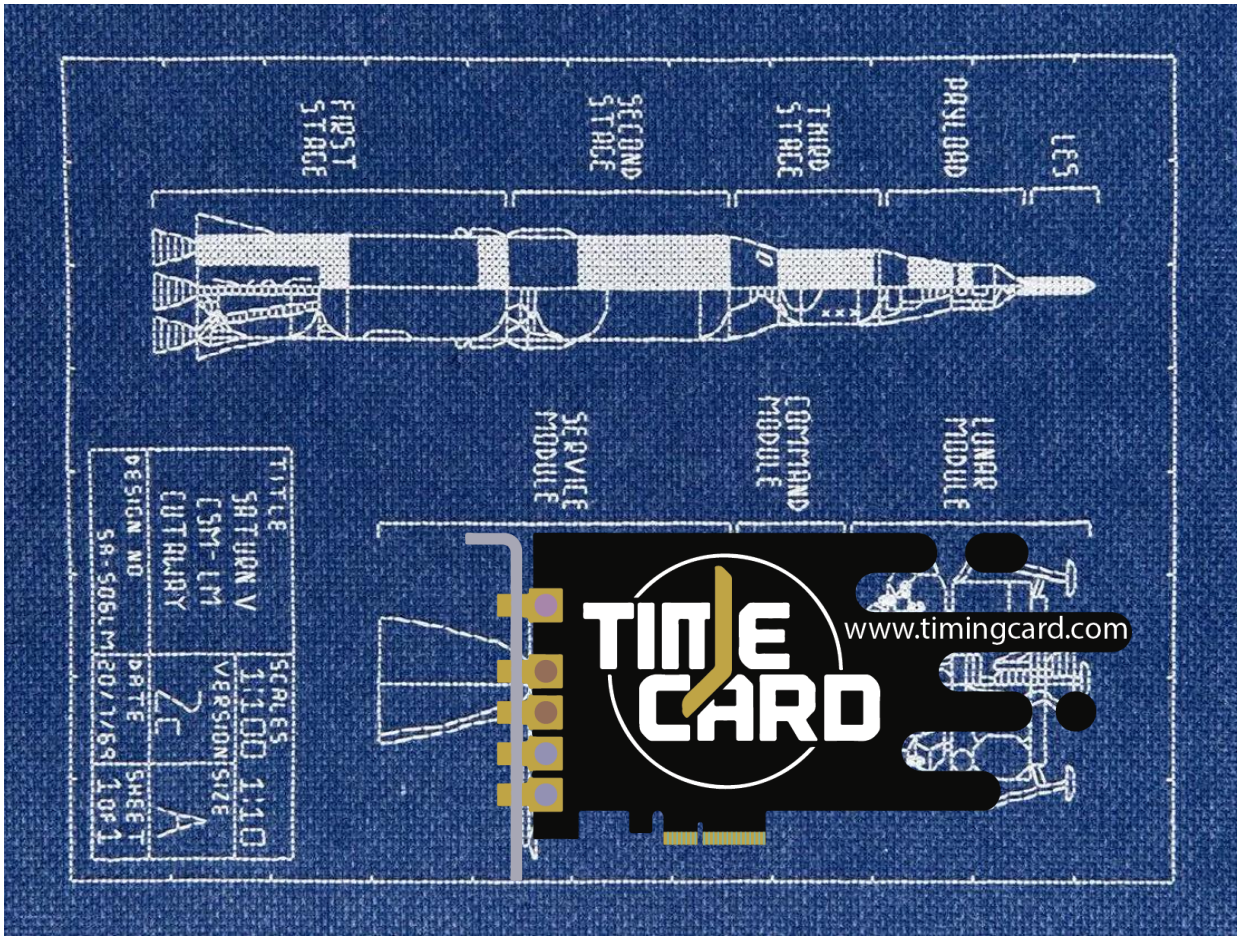
Time Precision Today and Tomorrow



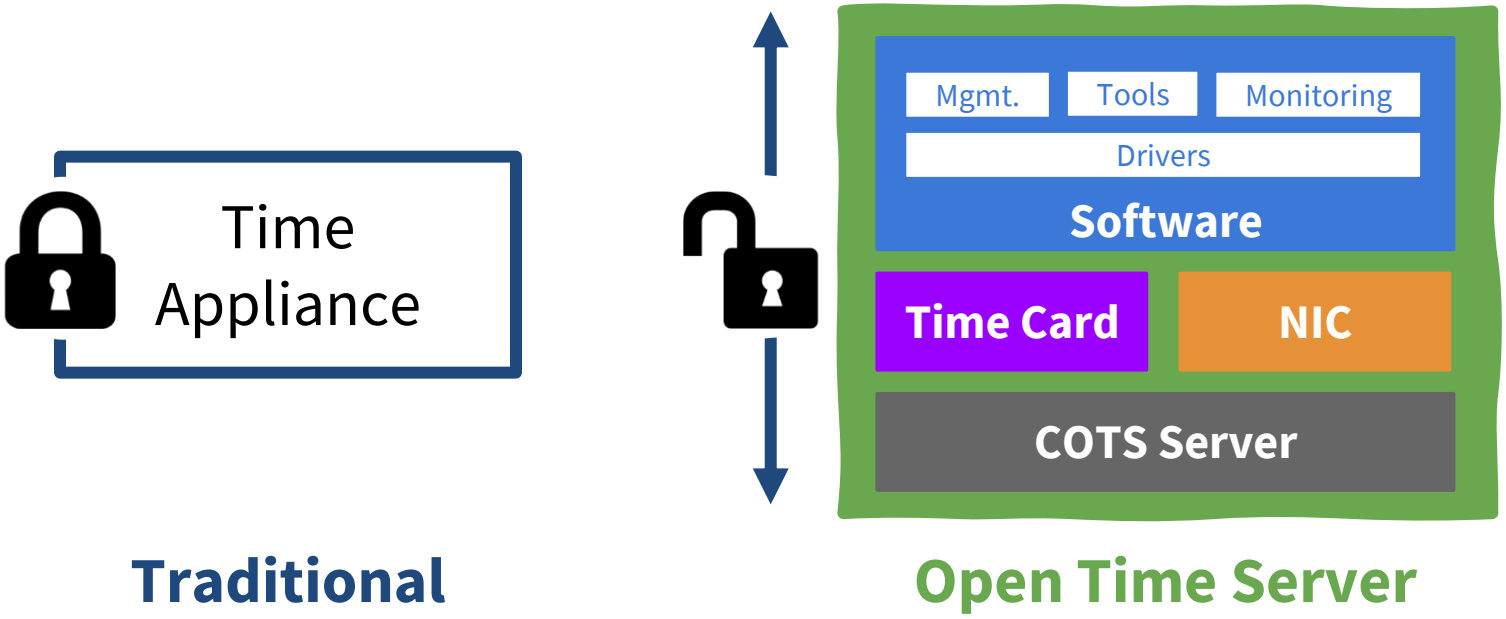
- Global – Data Center CPU to another Data Center CPU around the world
- P2P – CPU to another CPU in the same rack with minimum latency.

Time Keeping and Time Dissemination

- **Rocket:**
  - NTP
  - PTP
  - WR
- **Payload**
  - Time Card (Time)



Open Time Server



Traditional

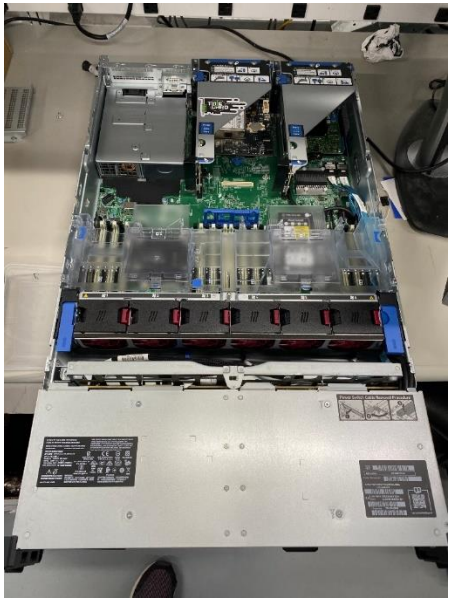
Open Time Server

Open Time Server

NetTimeLogic GMBH

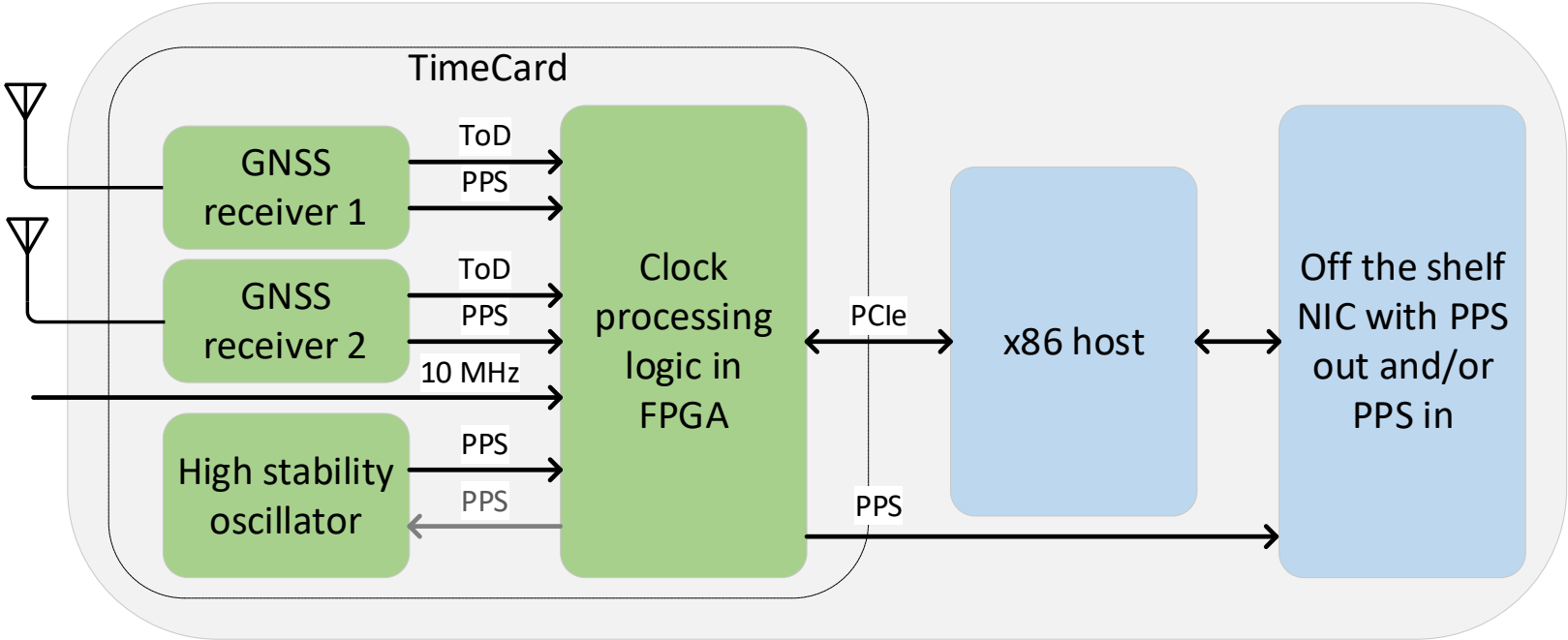
Meta

- Let’s hack something together (2019)



Open Time Server

- Based on an x86 architecture and an off-the-shelf NIC and the Time Card.



Advantages of Open Time Server

NetTimeLogic GMBH

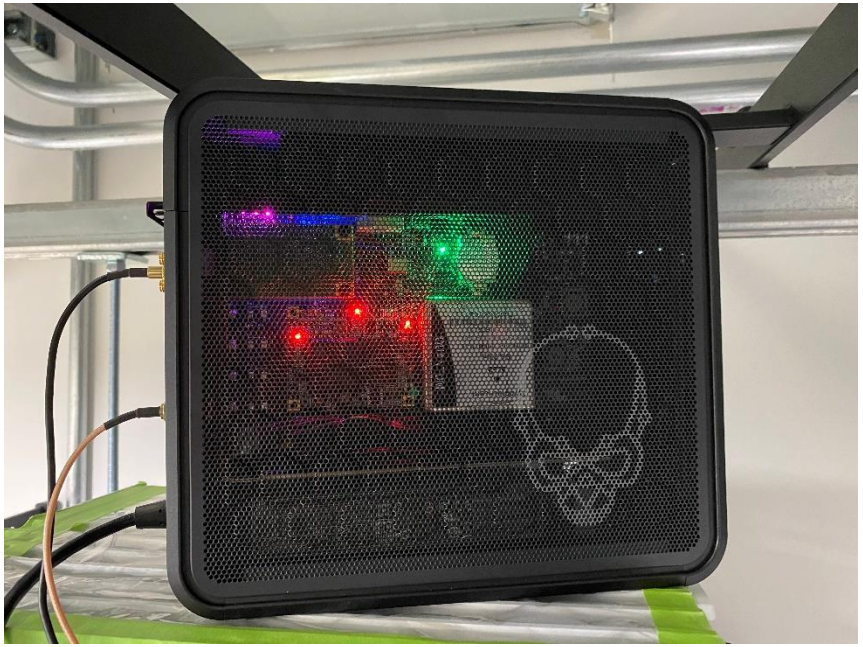
Meta

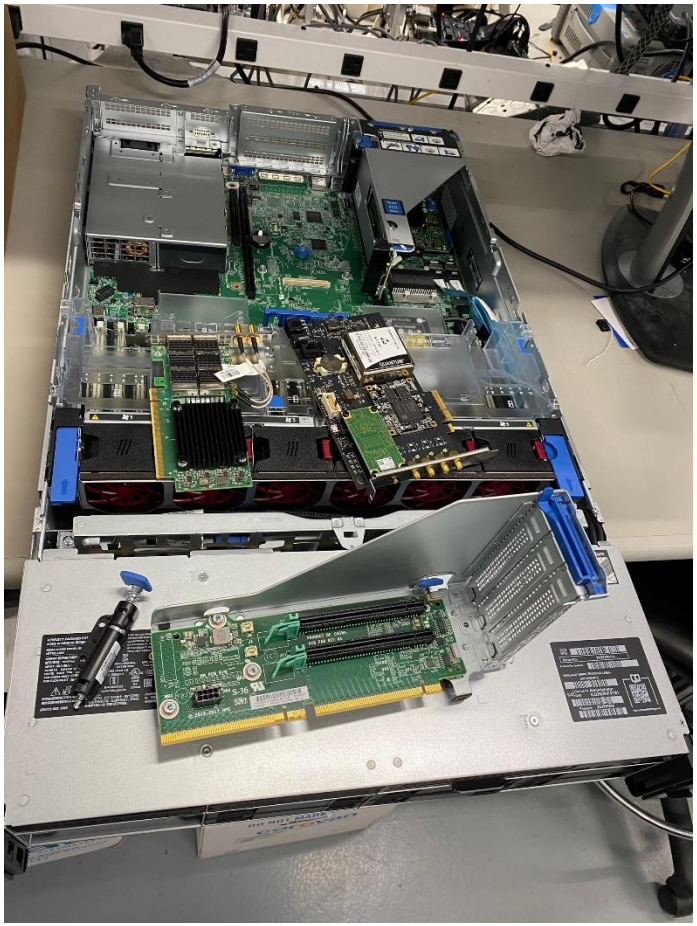
- **Flexible functionality**
  - Various sources of synchronization and distribution channels of the synchronized time
- **Adjustable features**
  - Adapting features without violating the design principles
- **Open-Source solution**
  - The community can constantly extend, adapt and upgrade the features and capabilities
- **Cost-effective design**
  - Convenient hardware customization, e.g., common clock header

Advantages of Open Time Server

NetTimeLogic GMBH

Meta





Adaptability of Open Time Server

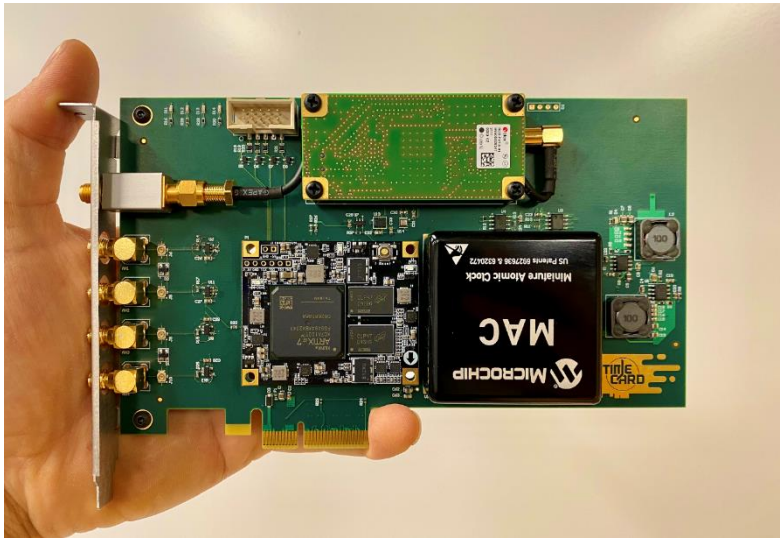
NetTimeLogic GMBH

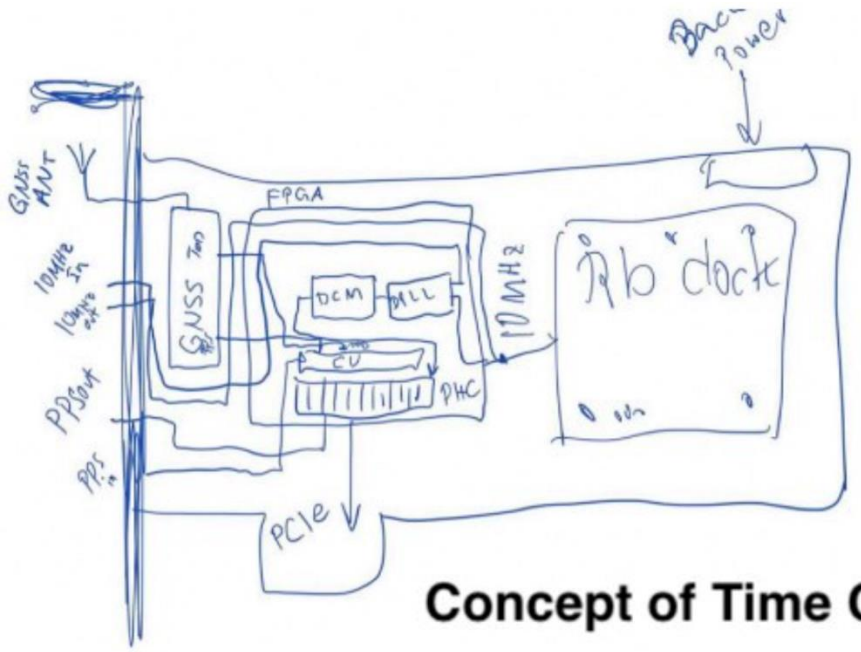
Meta

- **GNSS receiver**
  - Single or Dual
  - Different vendors
- **Oscillator**
  - MAC/TCXO/OCXO
  - Different Vendors
- **Time Card's custom functionalities**
  - Adjustable sources of synchronization
  - Time related functions, such as various signal generators, frequency generators, timestampers
- **Host Processing Unit's processing scalability**
  - Variety of used processors based on performance requirements
- **NIC's flexible protocol and link speed**
  - From 10MBit to 400GBit+

Time Card

- Concept (2020)
- First Proto (2021)
- Industry Adaption



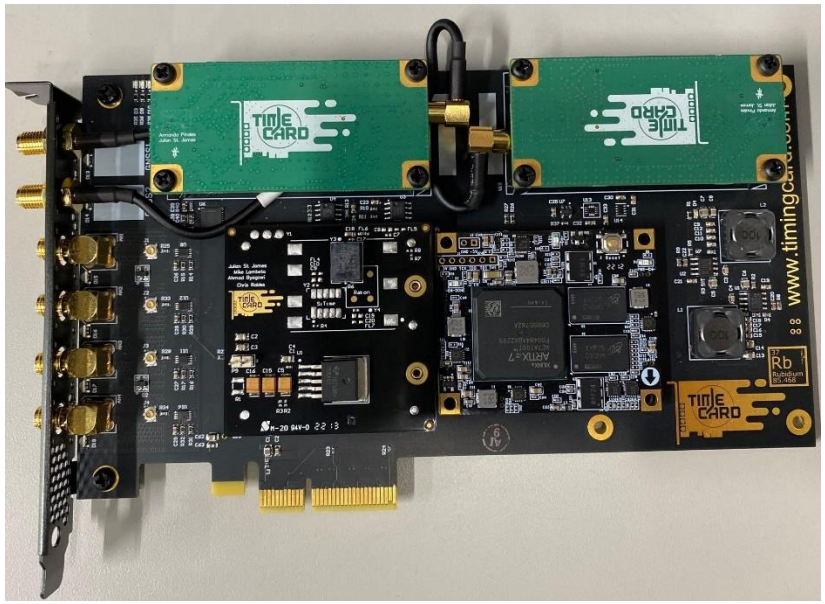


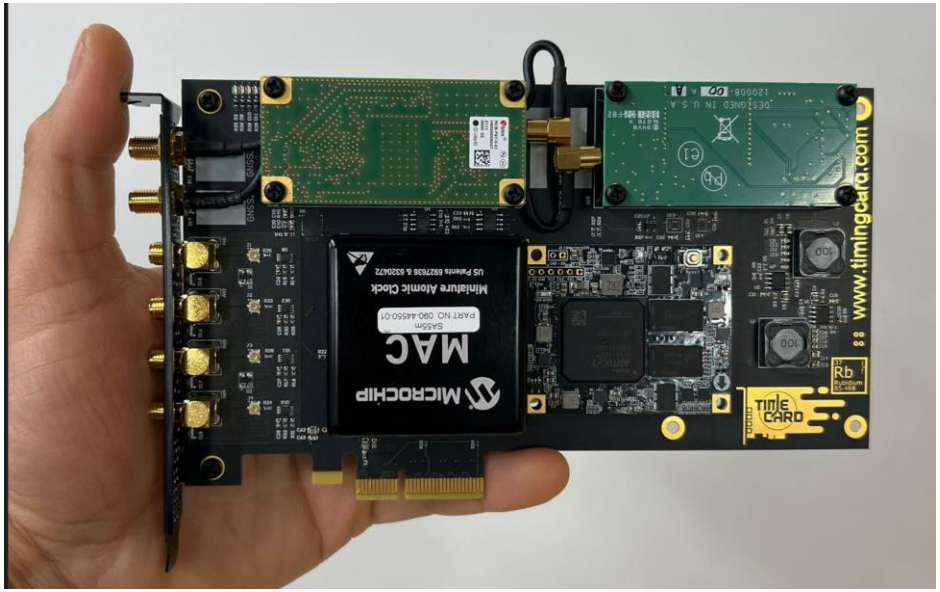
Concept of Time Card

Adaptability of Time Card

NetTimeLogic GMBH

Meta





Time Card's Family

NetTimeLogic GMBH

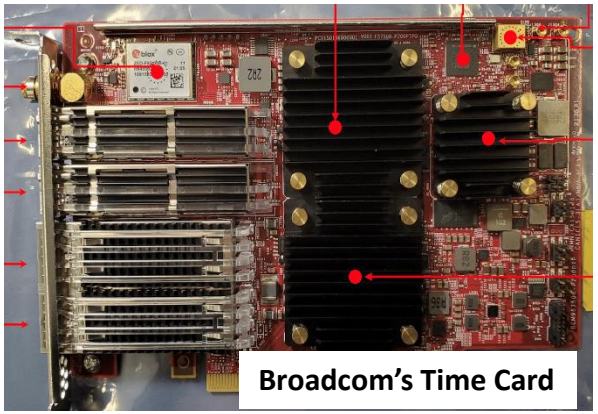
Meta

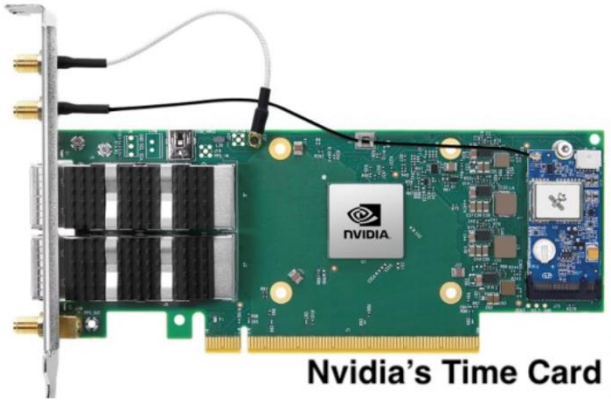
Orolia's Time Card


ADVA's Time Card

Celestica's Time Card

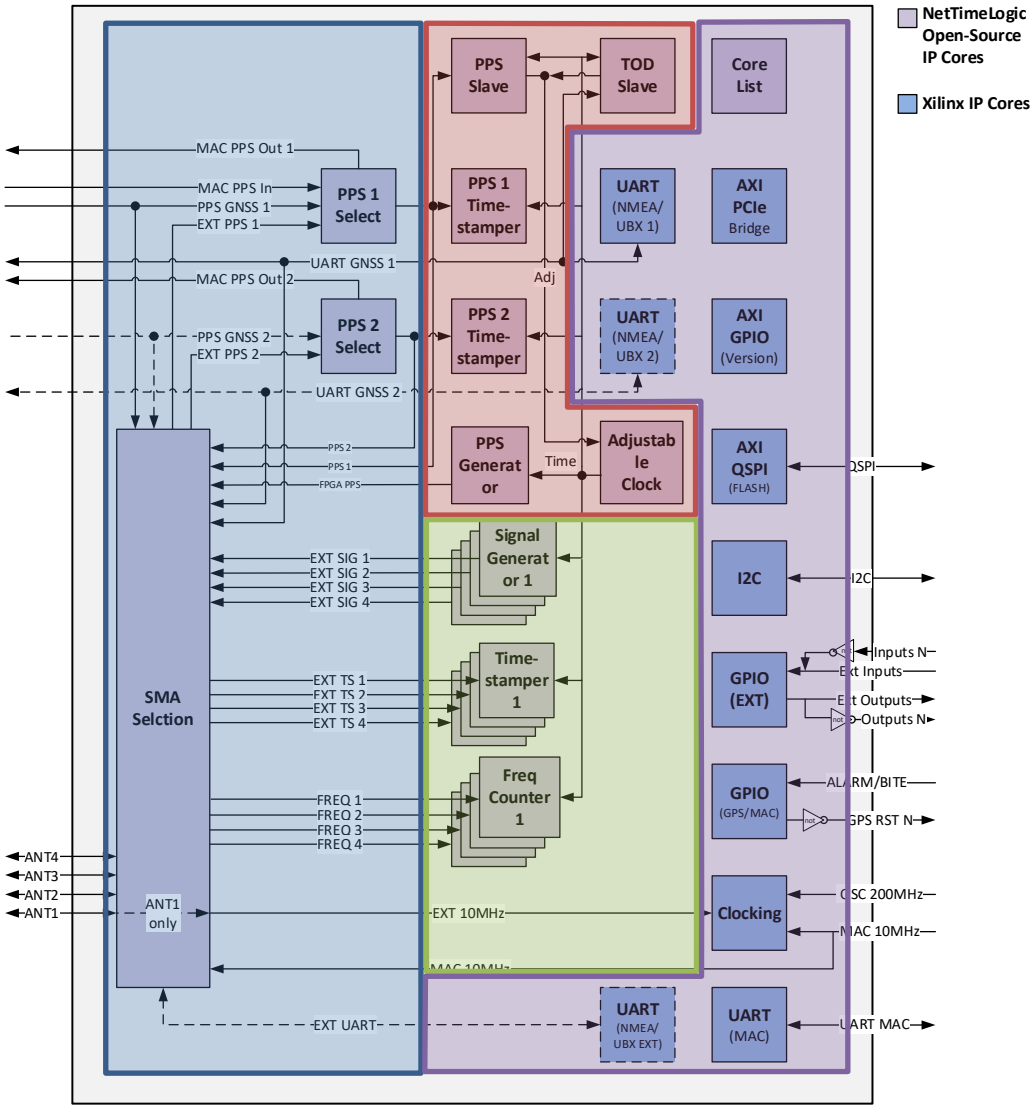
Intel's Time Card


Broadcom's Time Card



Nvidia's Time Card

Time Card's FPGA
Design Structure



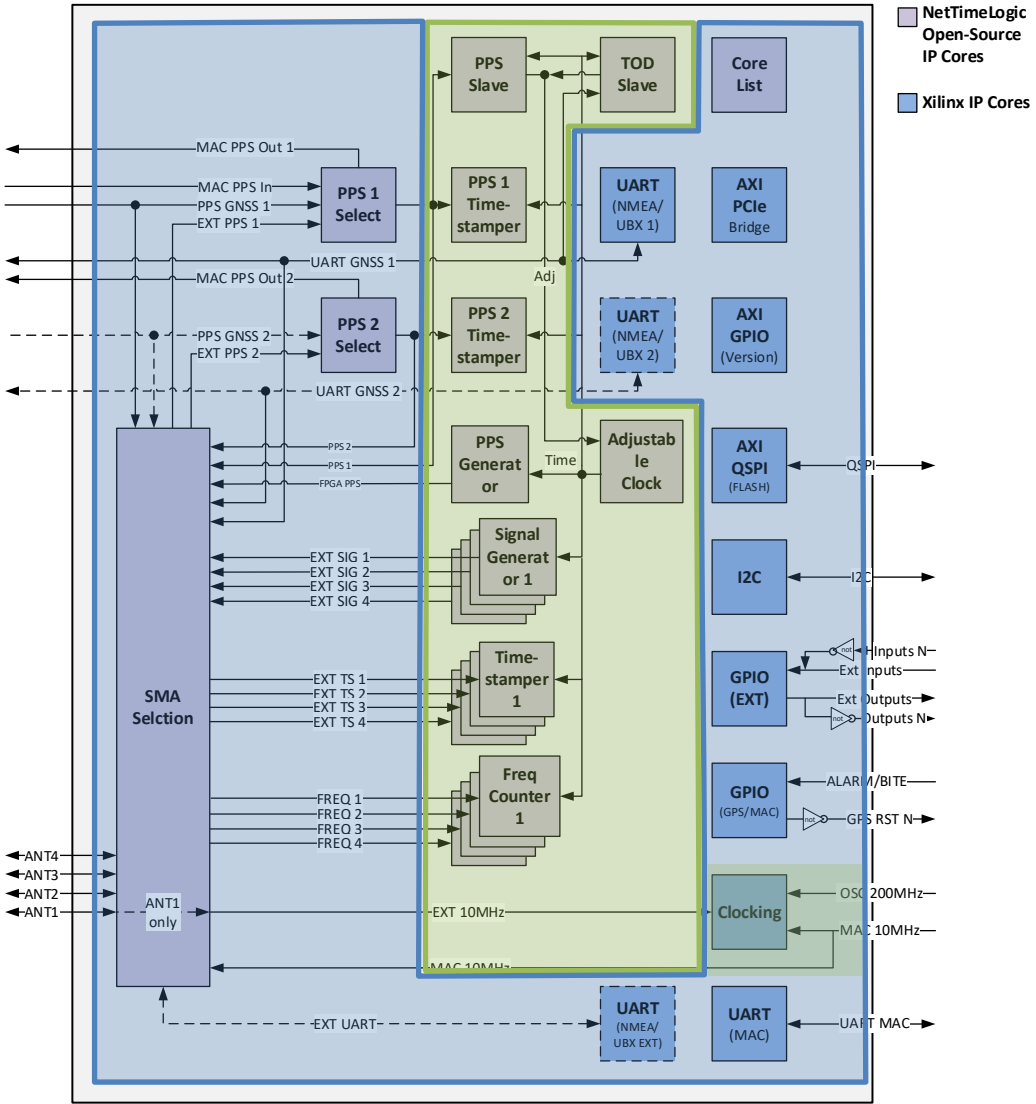
TIME SOURCE SELECTION
Select the time synchronization sources

TIME SYNCHRONIZATION
Standalone FPGA cores of time synchronization

TIME FUNCTIONS
Time-related support functions

CONTROL & COMMUNICATION
Interfaces to CPU and peripherals

Time Card's FPGA
Source Clocks



The source clock is the board's default OSC

✓ always available

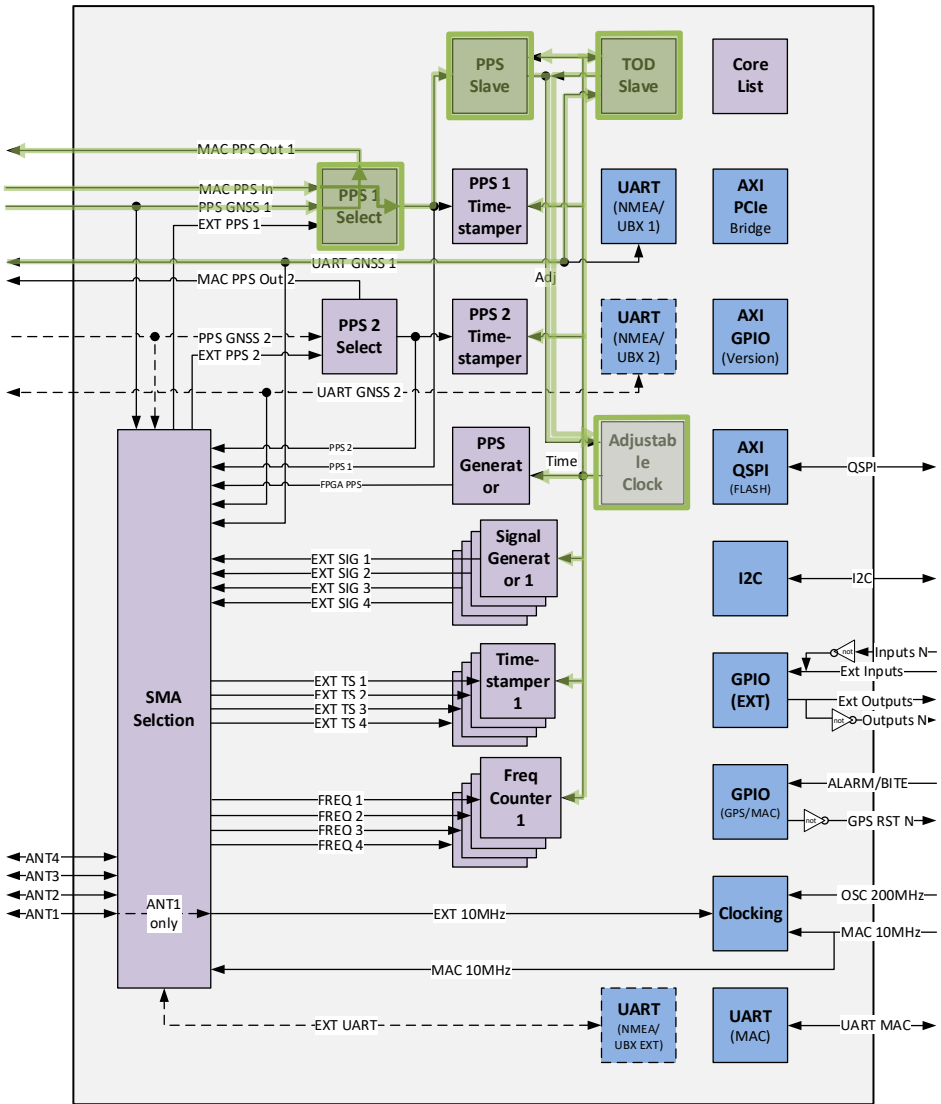
The source clock can be:

1. **SMA input (e.g. unmounted clock)**
2. **MAC**
3. **Board's default OSC**

✓ prioritization of available clocks

Time Card's FPGA

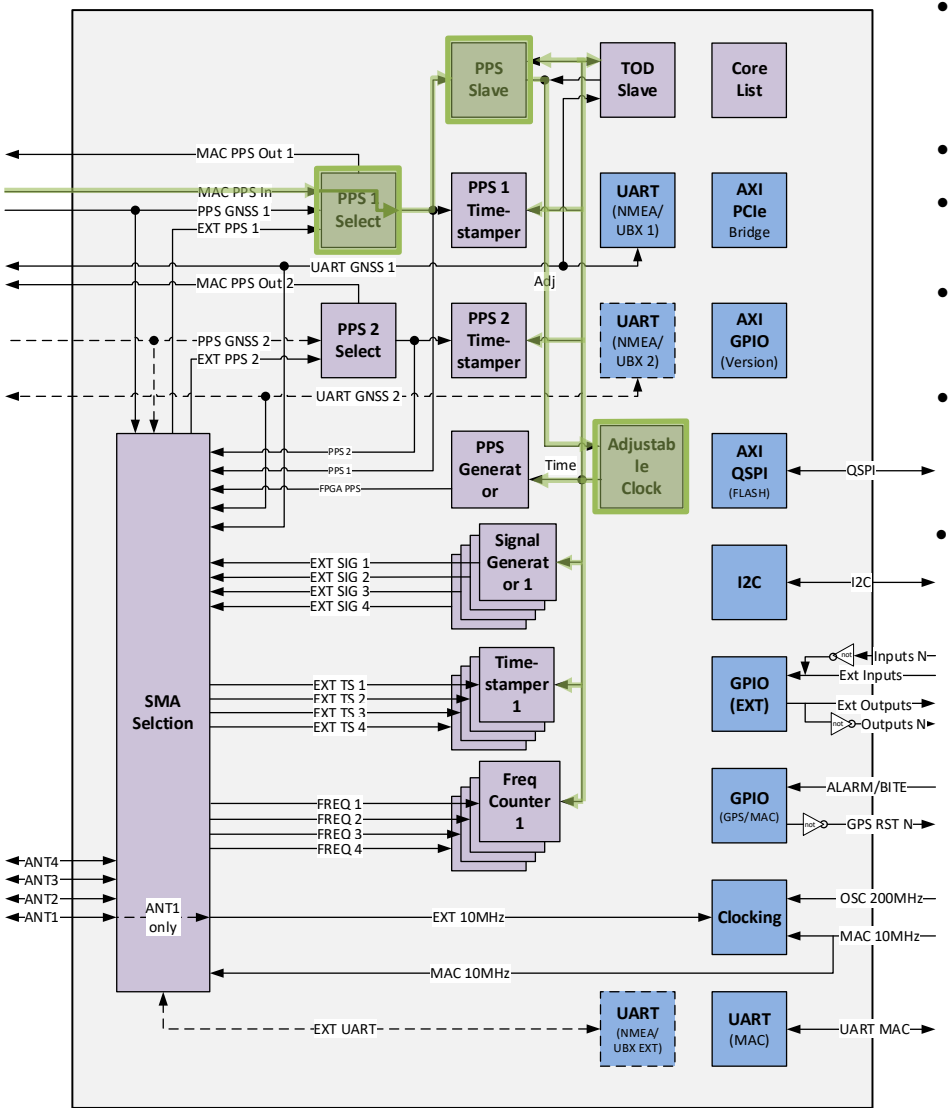
Synchronization example: GNSS and MAC



- When no adjustment is available, the Adjustable Counter Clock provides the local time as free-running
- The GNSS1 PPS and MAC PPS are routed to PPS Select
- The GNSS1 PPS is provided to the MAC
- The MAC PPS is selected for the PPS Slave
- The GNSS1 UART is driven to the ToD Slave
- The PPS Slave calculates the drift and offset
- The ToD Slave extracts the Time-Of-Day and compares its seconds with the local time
- The Adjustable Counter Clock receives the time adjustments.
  - If the required adjustments exceed a threshold, a time jump is applied
  - Otherwise, the offset and drift adjustments are distributed over a time interval (e.g. 1 second)
  - A time set from the ToD Slave adjusts the seconds

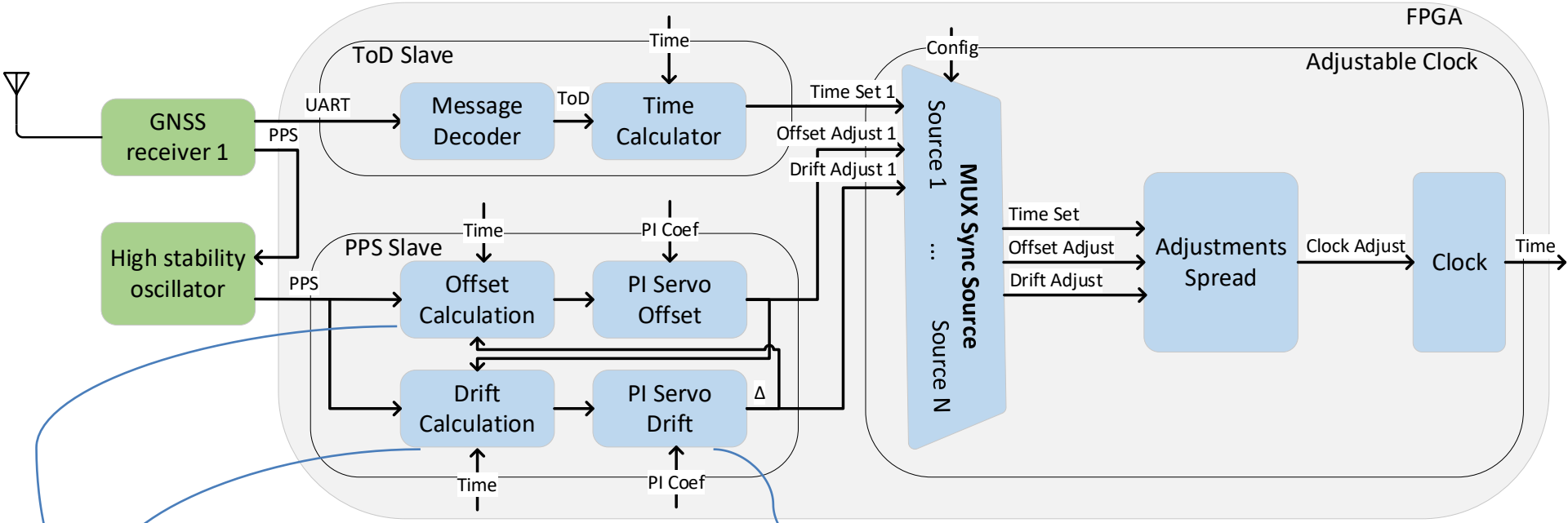
Time Card's FPGA

Synchronization example: lost GNSS



- When no adjustment is available, the Adjustable Counter Clock provides the free-running local time
- The MAC PPS is routed to the PPS Select
- The MAC PPS is selected to be sent to PPS Slave
- No PPS is provided to MAC, since GNSS1 PPS is not available.
- The PPS Slave calculates the Drift and Offset of the local clock to the reference clock by using 2 PI servos
- The Adjustable Counter Clock receives the offset and drift adjustments.
  - If the required adjustments exceed a threshold, a time jump is applied
  - Otherwise, the offset and drift adjustments are distributed over a time interval (e.g. 1 second)
  - No time adjustmennt from ToD slave

Time Card's FPGA Synchronization Process



$$Offset_N = Timestamp_N - \lfloor Time_N + 0.5 \rfloor - DriftAdjust_N$$

$$Drift_N = \frac{10^9(Timestamp_N - Timestamp_{N-1} - 10^9 - OffsetAdjust_{N-1})}{Timestamp_N - Timestamp_{N-1}}$$

Drift and Offset are corrected separately with 2 PI Servo loops

$$OffsetAdjust_N = K_{P_D} Offset_N + K_{I_D} \sum_1^N Offset_n$$

$$DriftAdjust_N = K_{P_O} Drift_N + K_{I_O} \sum_1^N Drift_n$$

where

K_P is the proportional coefficient

K_I is the integral coefficient

Configuration of Time Card

NetTimeLogic GMBH

Meta

```
$ ls -g /sys/class/timecard/ocp0/
total 0
-r--r--r--. 1 root 4096 Sep  8 18:20 available_clock_sources
-r--r--r--. 1 root 4096 Sep  8 18:20 available_sma_inputs
-r--r--r--. 1 root 4096 Sep  8 18:20 available_sma_outputs
-rw-r--r--. 1 root 4096 Sep  8 18:20 clock_source
lrwxrwxrwx. 1 root    0 Sep  8 18:20 device -> ../../../0000:02:00.0
-rw-r--r--. 1 root 4096 Sep  8 18:20 external_pps_cable_delay
-r--r--r--. 1 root 4096 Sep  8 18:20 gnss_sync
-rw-r--r--. 1 root 4096 Sep  8 18:20 internal_pps_cable_delay
-rw-r--r--. 1 root 4096 Sep  8 18:20 irig_b_mode
-rw-r--r--. 1 root 4096 Sep  8 18:20 pci_delay
drwxr-xr-x. 2 root    0 Sep  8 18:20 power
lrwxrwxrwx. 1 root    0 Sep  8 18:20 ptp -> ../../ptp/ptp4
-r--r--r--. 1 root 4096 Sep  8 18:20 serialnum
-rw-r--r--. 1 root 4096 Sep  8 18:20 sma1_in
-rw-r--r--. 1 root 4096 Sep  8 18:20 sma2_in
-rw-r--r--. 1 root 4096 Sep  8 21:04 sma3_out
-rw-r--r--. 1 root 4096 Sep  8 21:04 sma4_out
lrwxrwxrwx. 1 root    0 Sep  8 18:20 subsystem -> ../../../../../../class/timecard
lrwxrwxrwx. 1 root    0 Sep  8 18:20 ttyGNSS -> ../../tty/ttyS5
lrwxrwxrwx. 1 root    0 Sep  8 18:20 ttyMAC -> ../../tty/ttyS6
lrwxrwxrwx. 1 root    0 Sep  8 18:20 ttyNMEA -> ../../tty/ttyS7
-rw-r--r--. 1 root 4096 Sep  8 18:20 uevent
-rw-r--r--. 1 root 4096 Sep  8 18:20 utc_tai_offset
```

Configuration of Time Card

NetTimeLogic GMBH

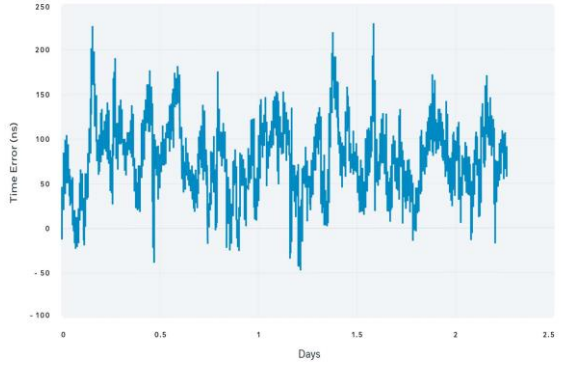
Meta

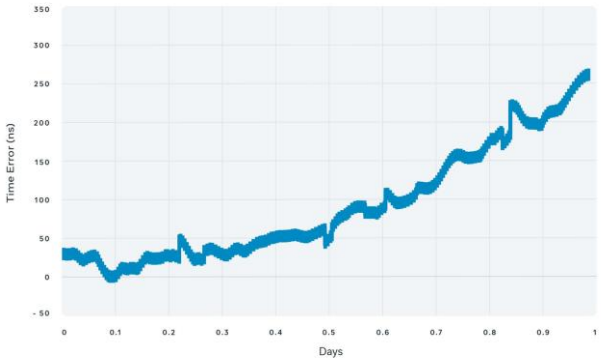
```
Time Card Configurator

    info              information about the Time Card Configurator
    device            enumeration of the time card device
    clock_sources     setting the clock source
    peripherals       show the pheripheral enumerations
    sma_config        configure the sma connectors
    signal_generators configure the signal generators
    other_settings    set irig-b mode, utc_tai_offset, pci_delay, etc...
    gnss_status       show the status of the GNSS receiver
    mac_status        show the status of the mac
    timestampers      show the timestamper stack
    update            update the firmware of the Time Card


                  <Select>                    <Finish>
```

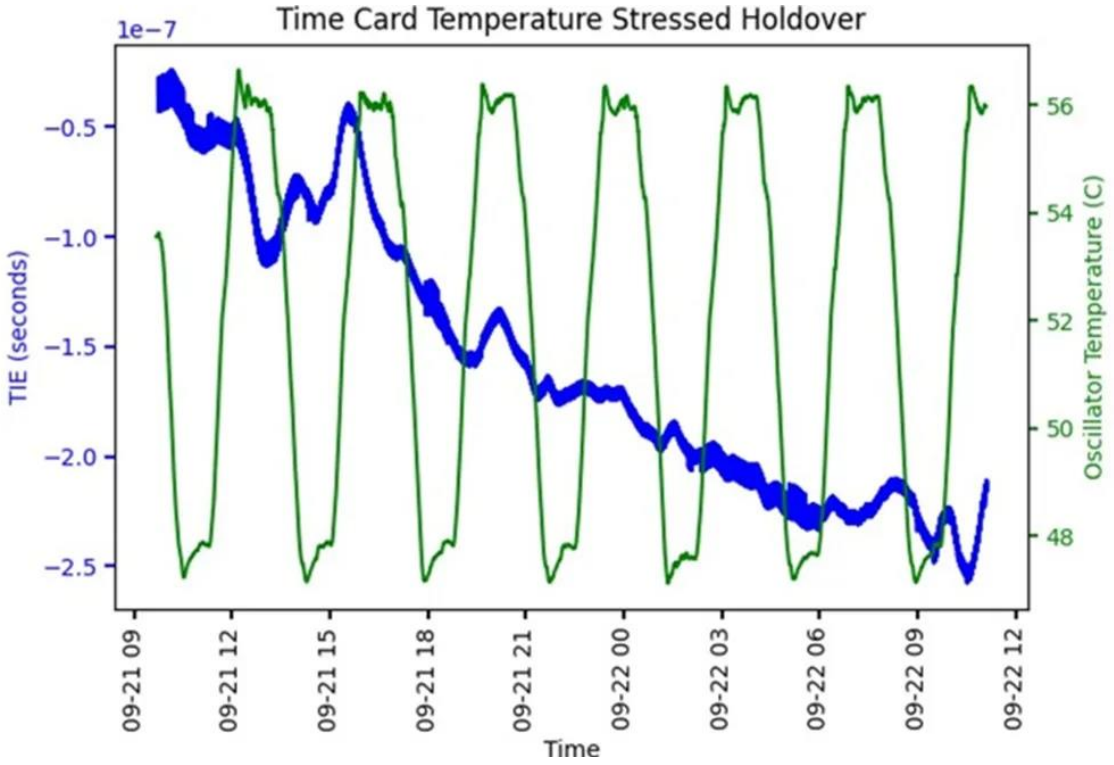
Performance

NetTimeLogic GMBH

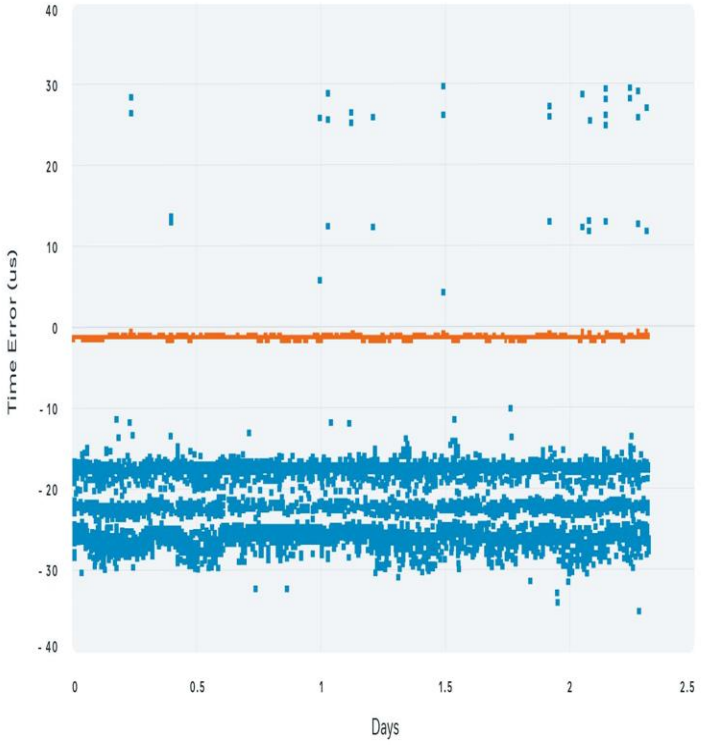
Meta





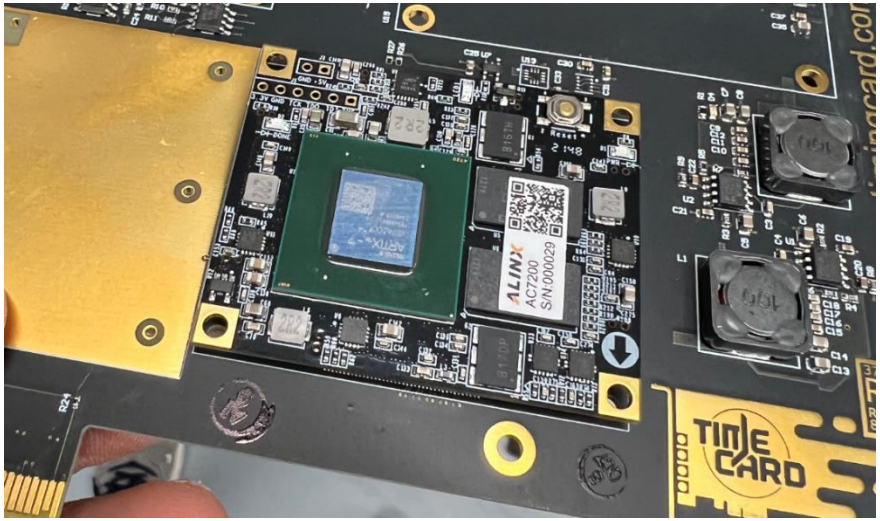


Performance



What's Next

- Increase FPGA Adaptability
- Reduce time error introduced by PCIe
- Improve Time Card synchronization from multiple GNSS receivers



Thank you

NetTimeLogic GMBH

Meta

- Join us on:
  - OpenTimeServer.com
  - TimeCard.ch
  - Timingcard.com
  - OCPTAP.com
  - Discord Server : https://discord.gg/rQUKJsHs