KLEPSYDRA
TECHNOLOGIES

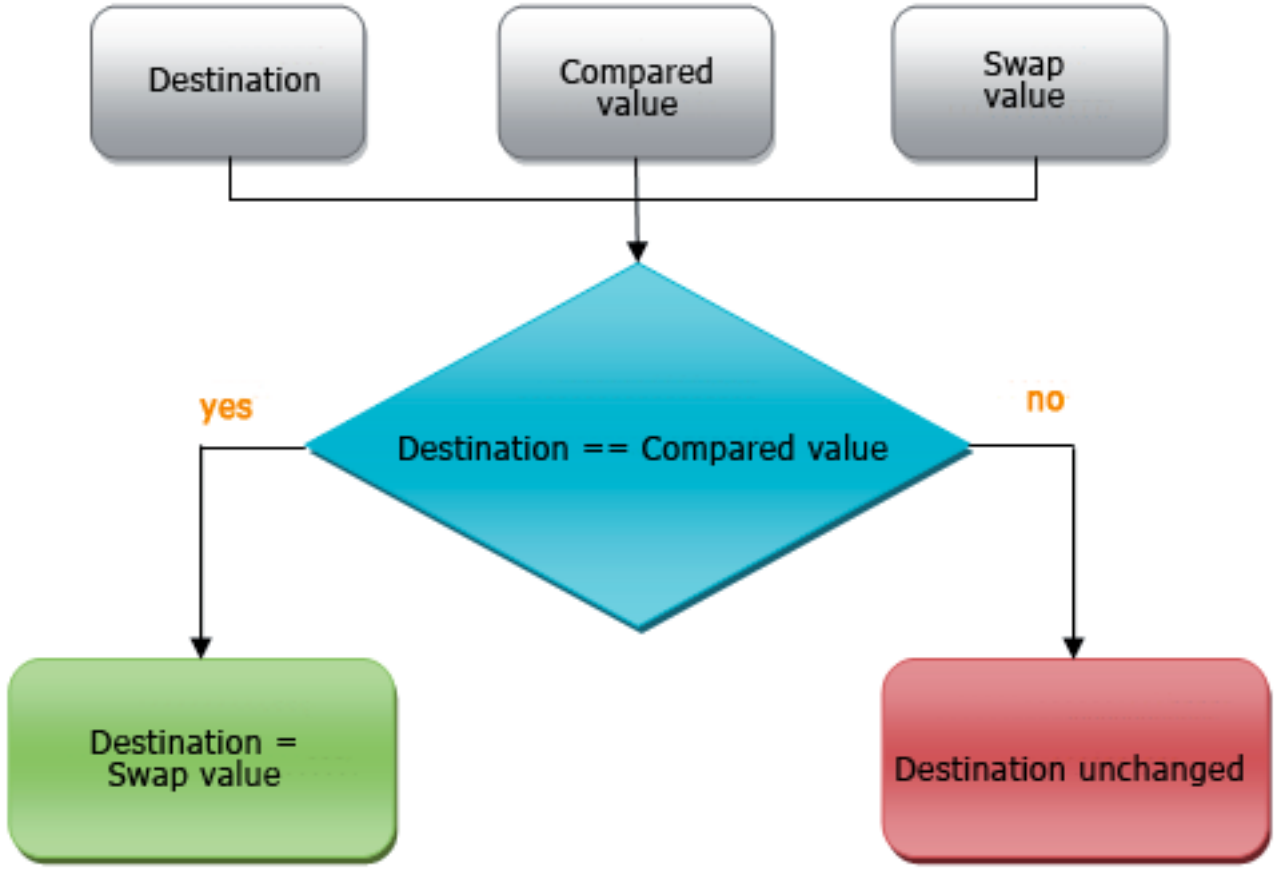
LOCK-FREE ROS 2 EXECUTOR: A RING-BUFFER TO RULE THEM ALL

ROSCON 2021
Executor Workshop

Dr Pablo Ghiglino
CEO and Founder
pablo.ghiglino@klepsydra.com
www.klepsydra.com

KLEPSYDRA TECHNOLOGIES

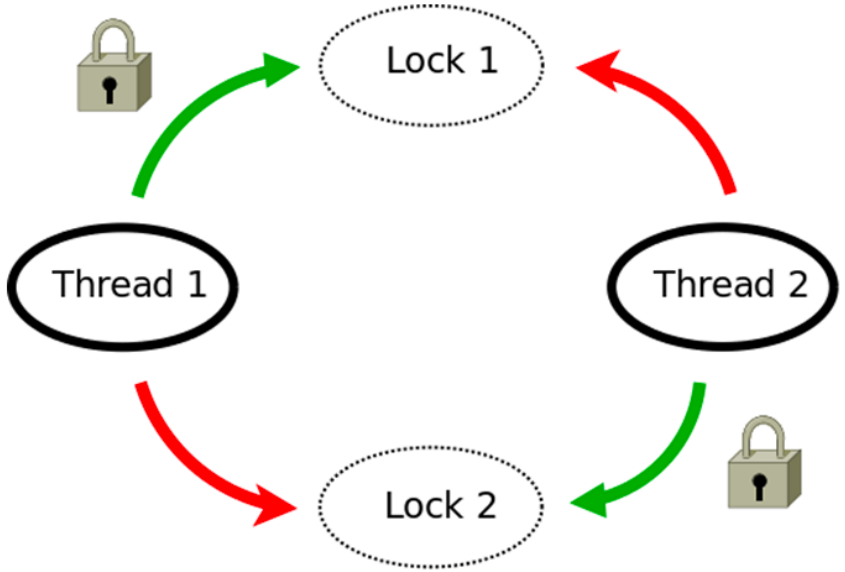
LOCK-FREE PROGRAMMING

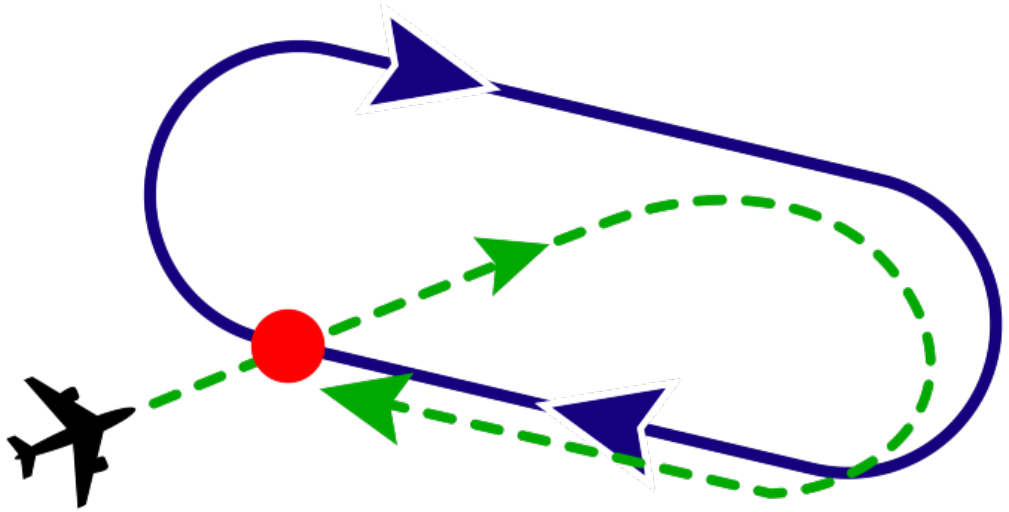


- **Compare-and-swap (CAS)** is an instruction used in multithreading to achieve synchronisation. It compares the contents of a memory location with a given value and, only if they are the same, modifies the contents of that memory location to a new given value. **This is done as a single atomic operation.**
- Compare-and-Swap has been an integral part of the **IBM 370 architectures since 1970**.
- **Maurice Herlihy (1991)** proved that CAS can implement more of these algorithms than **atomic read, write, and fetch-and-add**

KLEPSYDRA TECHNOLOGIES

LOCK-FREE PROGRAMMING

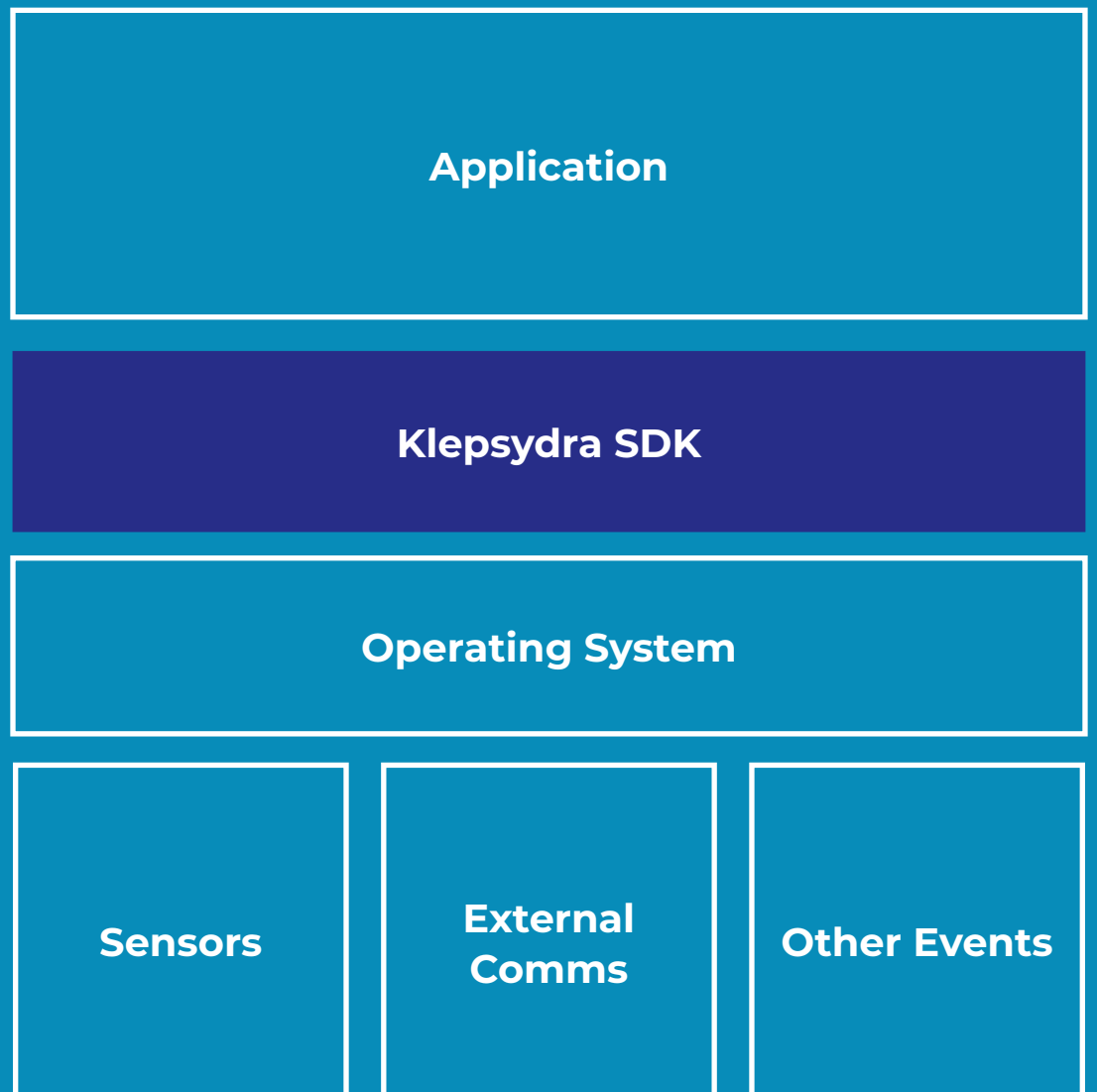


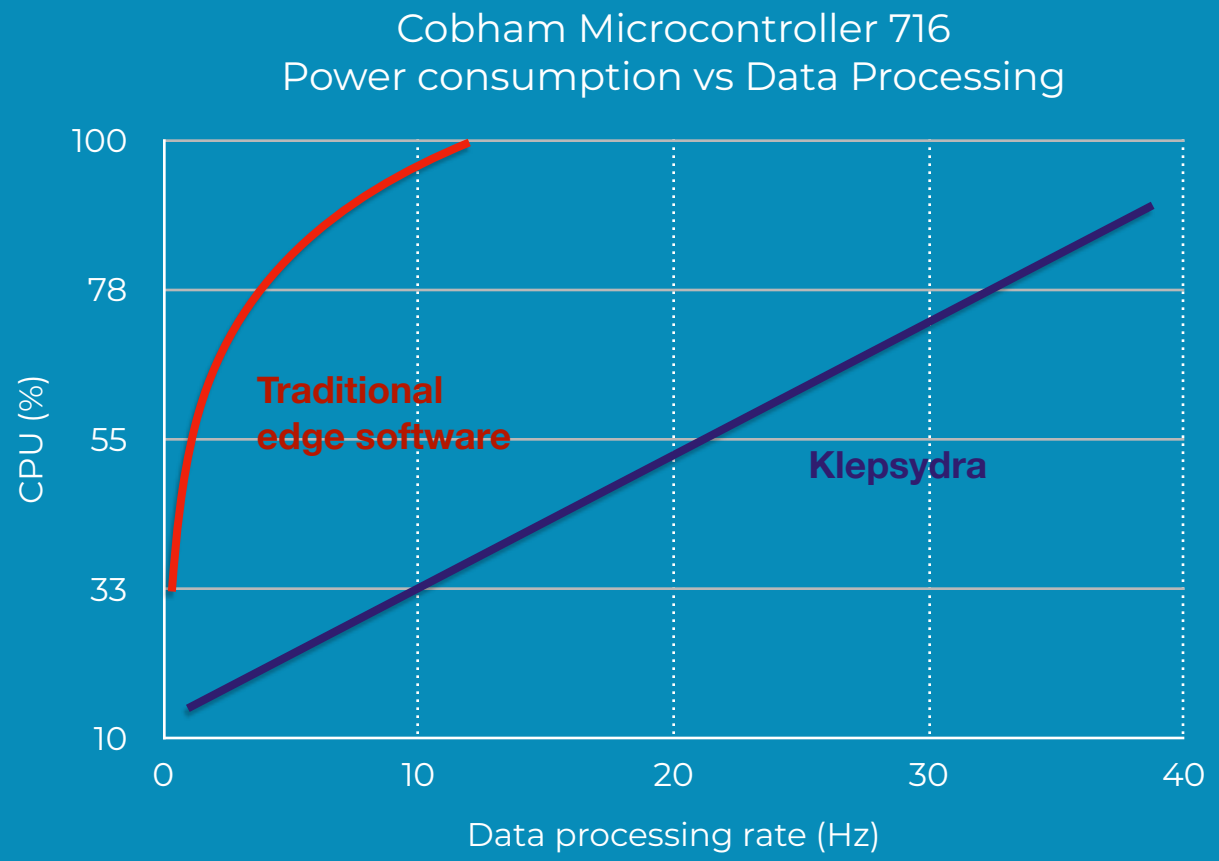


- Threads need to acquire lock to access resource.
- Context switch:
  - Suspended while resource is locked by someone else
  - Awaken when resource is available.
- Not deterministic, power consuming context switch.

- Threads access resources using ‘Atomic Operations’
- Compare and Swap (CAS):
  - Try to update a memory entry
  - If not possible tried again
  - No locks involved, but ‘busy wait’
- No context switch required.

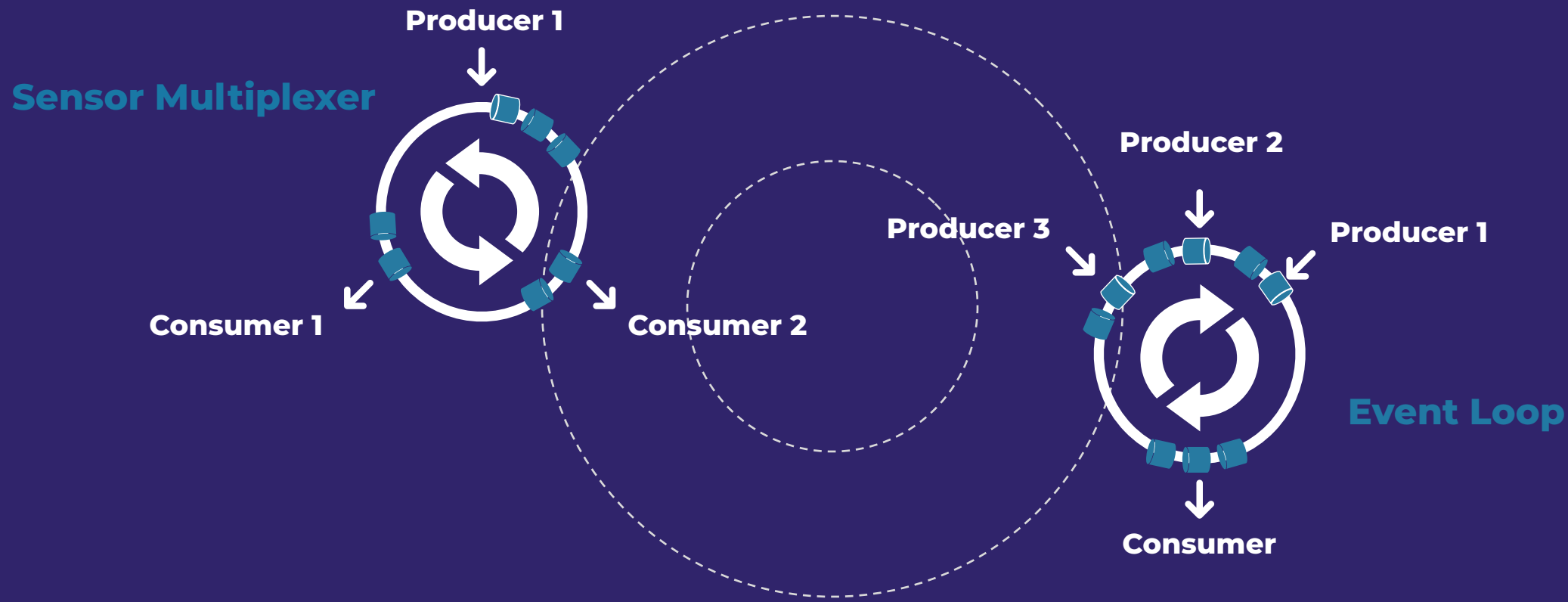
KLEPSYDRA RING-BUFFER





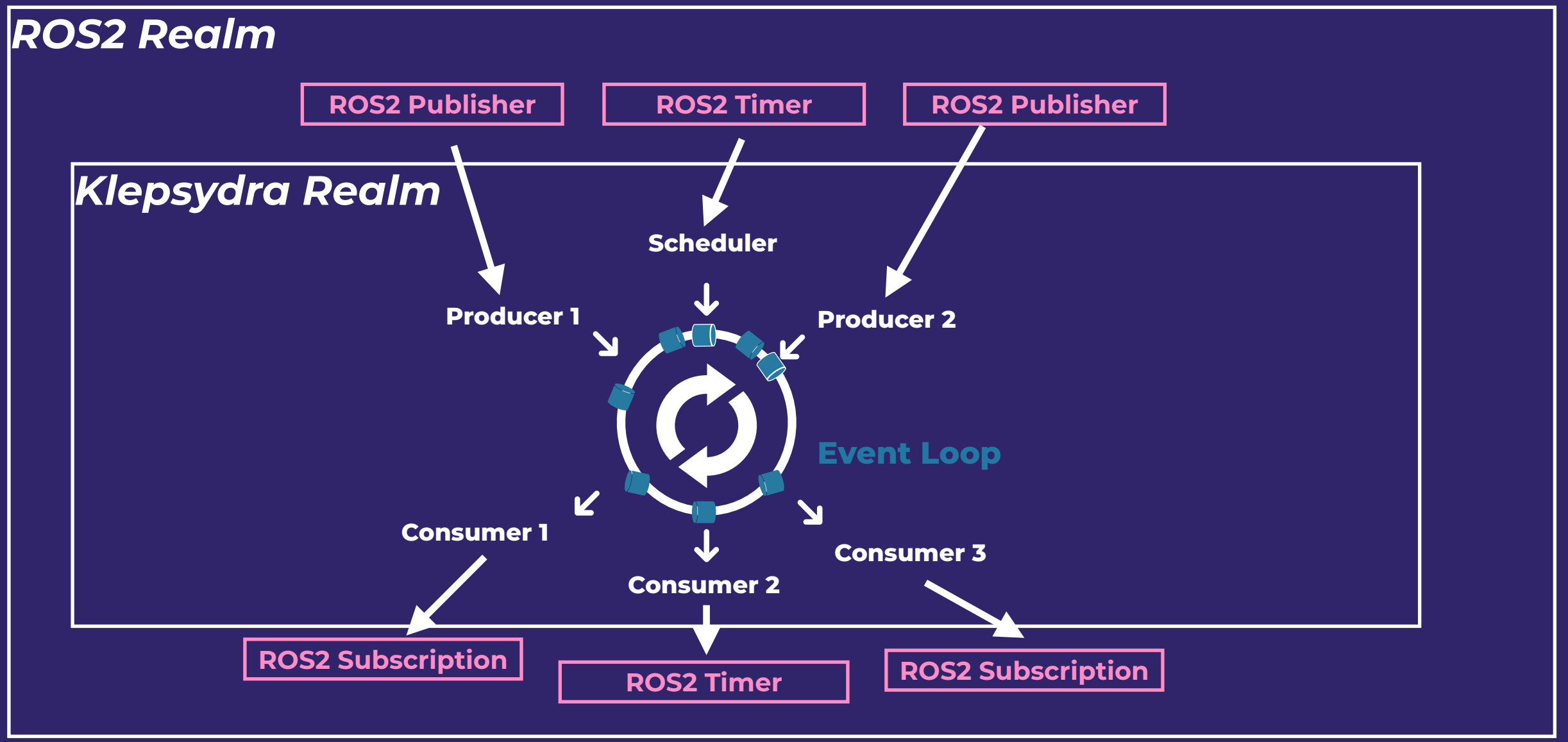
KLEPSYDRA TECHNOLOGIES

Klepsydra Ring-buffer



KLEPSYDRA TECHNOLOGIES

Klepsydra Lock-free executor



KLEPSYDRA TECHNOLOGIES

Klepsydra Lock-free executor

How does it work?

- Similar implementation to the Static Single Thread Executor
- All subscriptions and timer tasks run on the same thread
- Publishers can run on any thread

Implementation details

- Subscriptions from all message types are treated as Eventloop's listeners
- Timers are treated as Eventloops scheduled functions
- Similar to the Static Single Thread, there is no cloning.

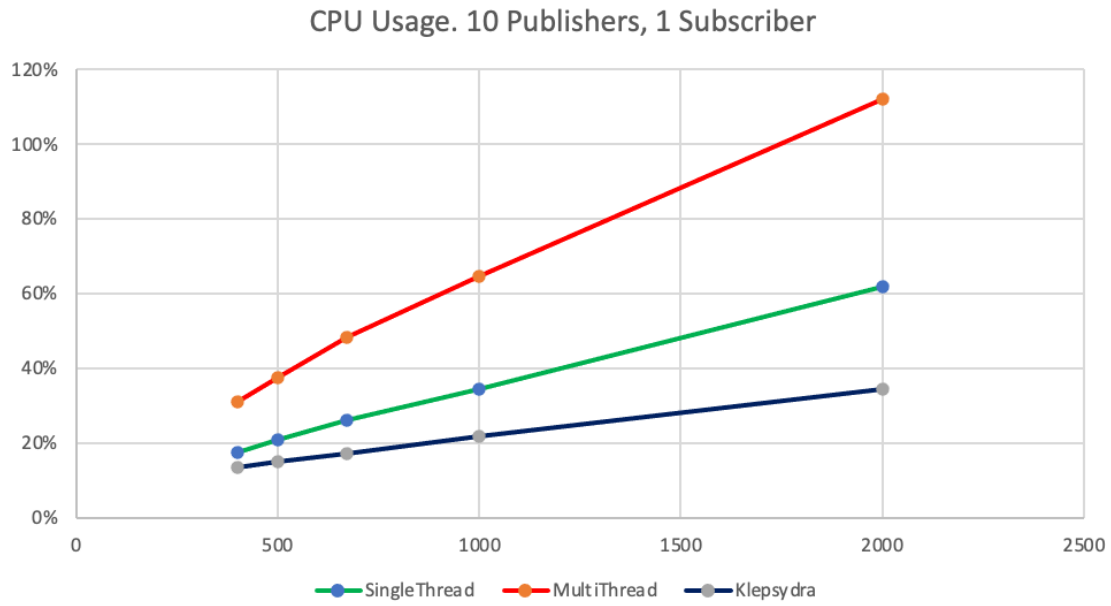
ROS2 CHALLENGES

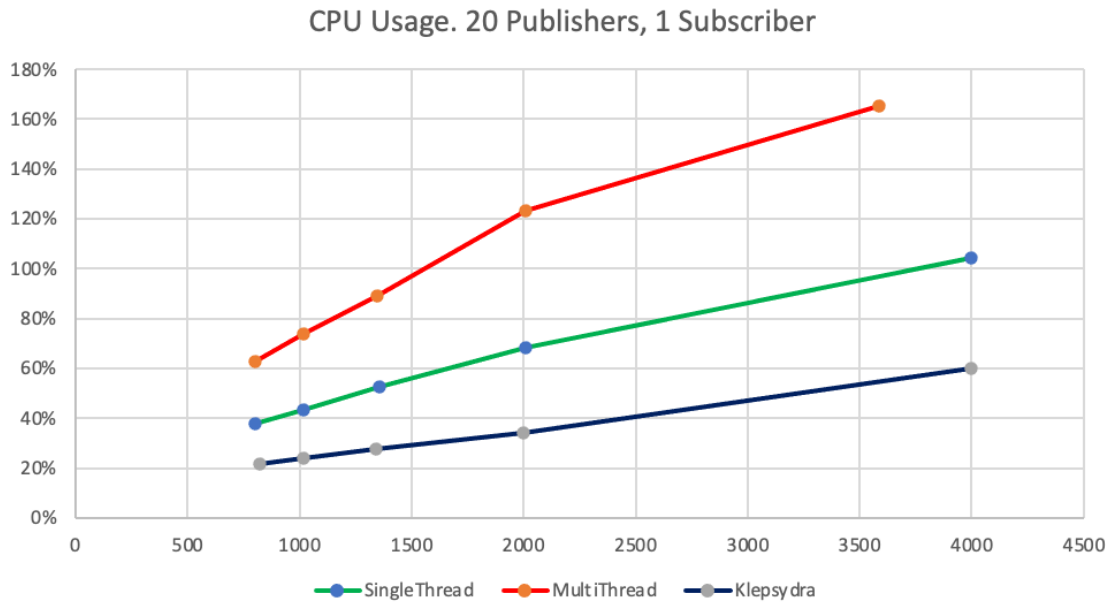
Open issues in ROS2:

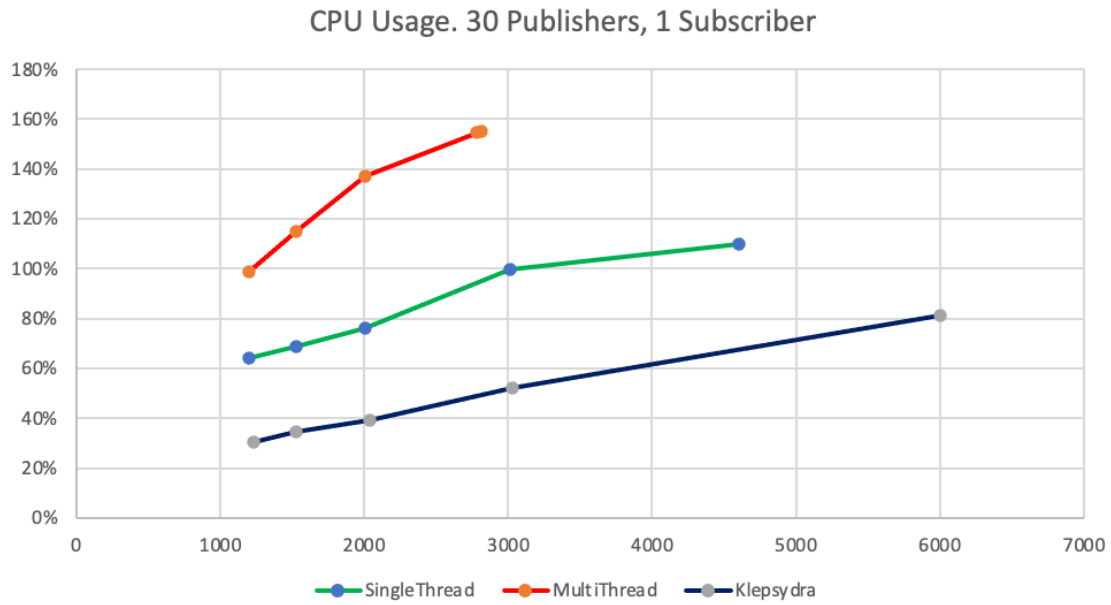
1. High CPU use of executor. (https://github.com/ros2/rclcpp/issues/1637)

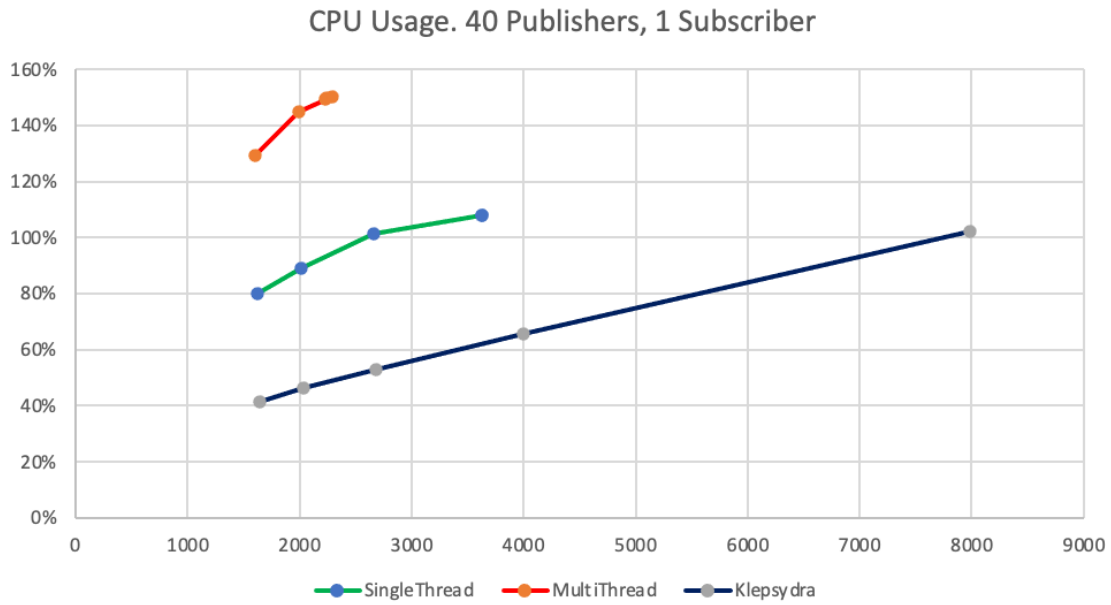
2. Large pointcloud pubsub is unstable when there are many subscribers. (https://github.com/ros2/rmw_cyclonedds/issues/292)

Eventloop Executor on Raspberry PI 4 I

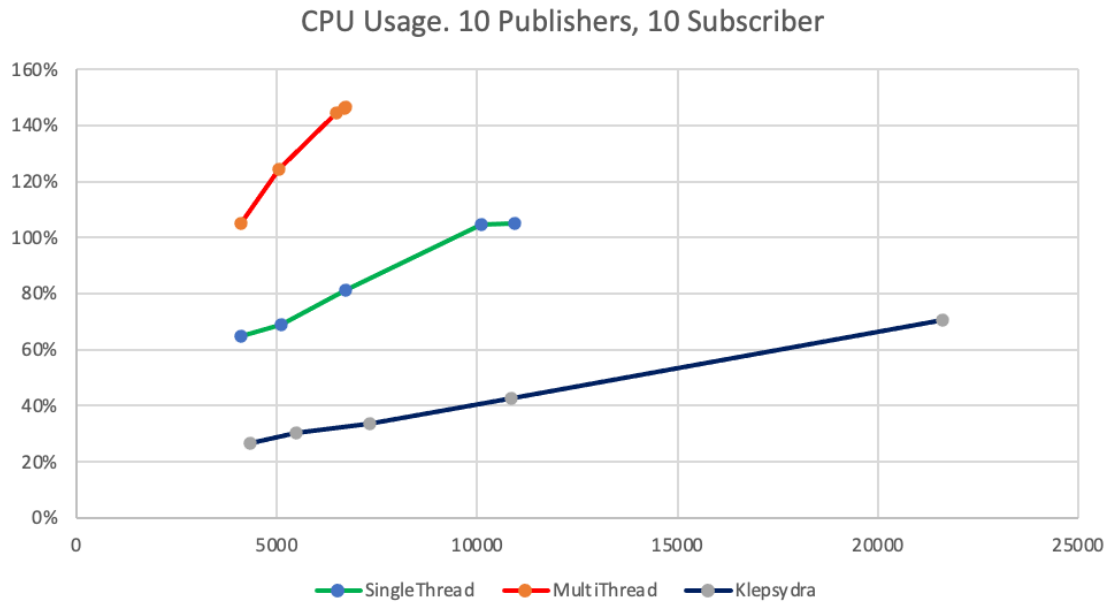


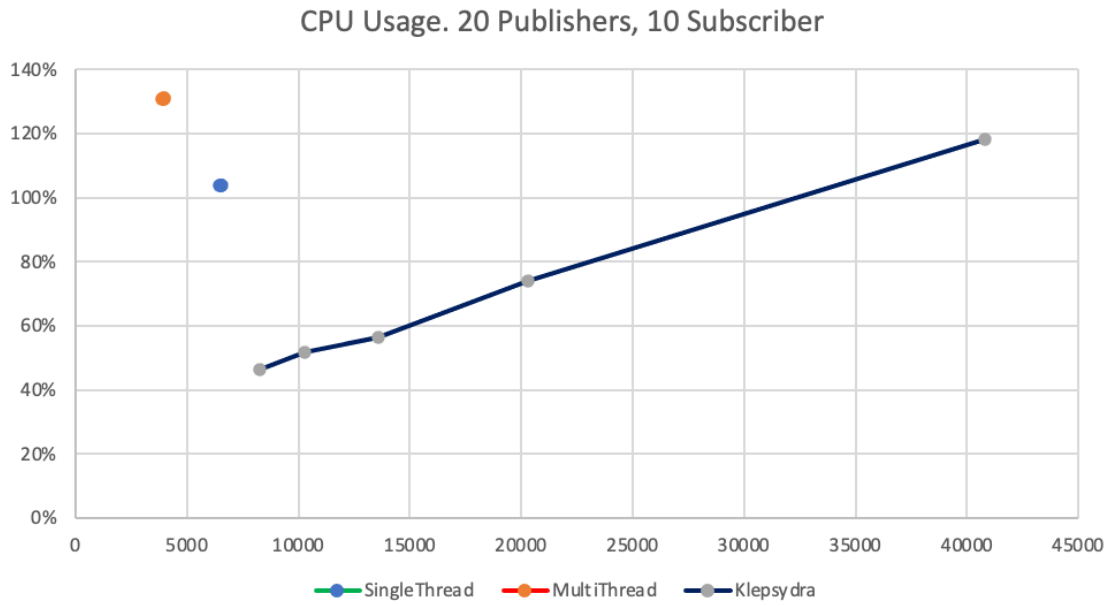


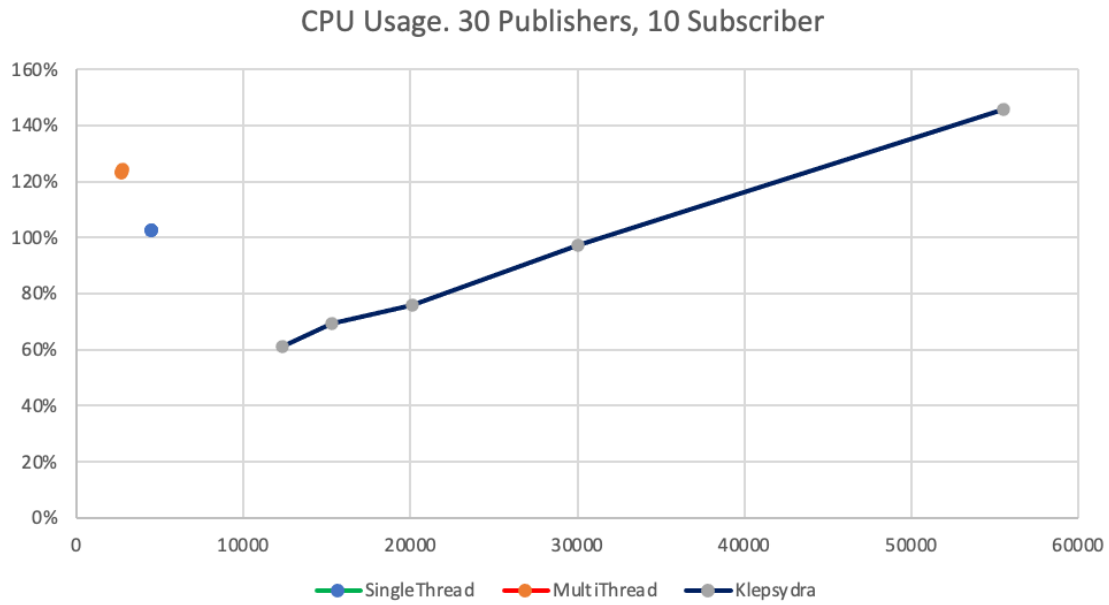


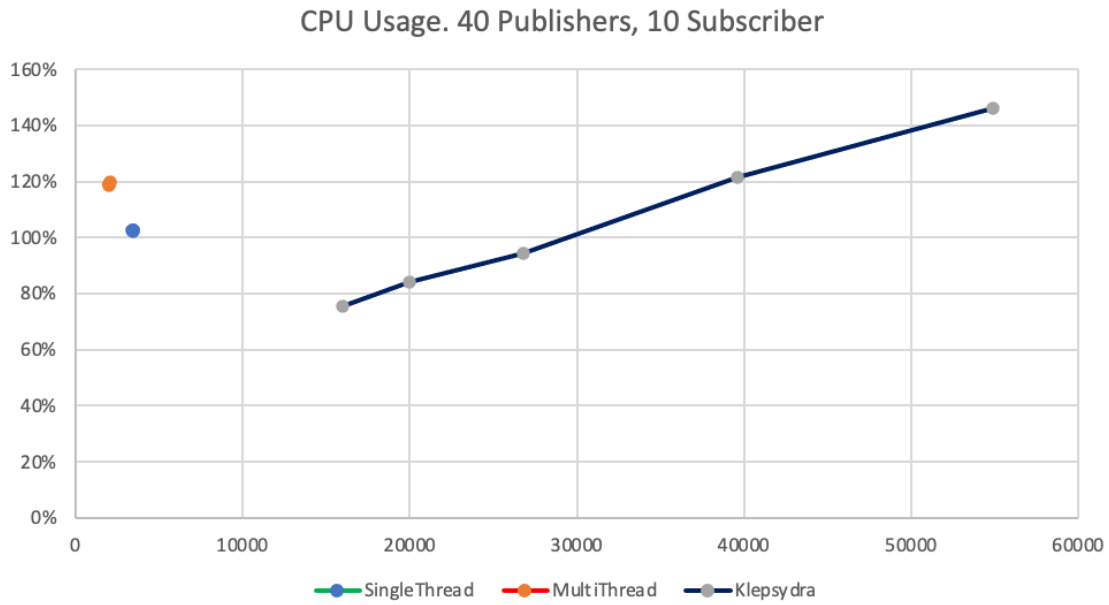


Eventloop Executor on Raspberry Pi 4 II

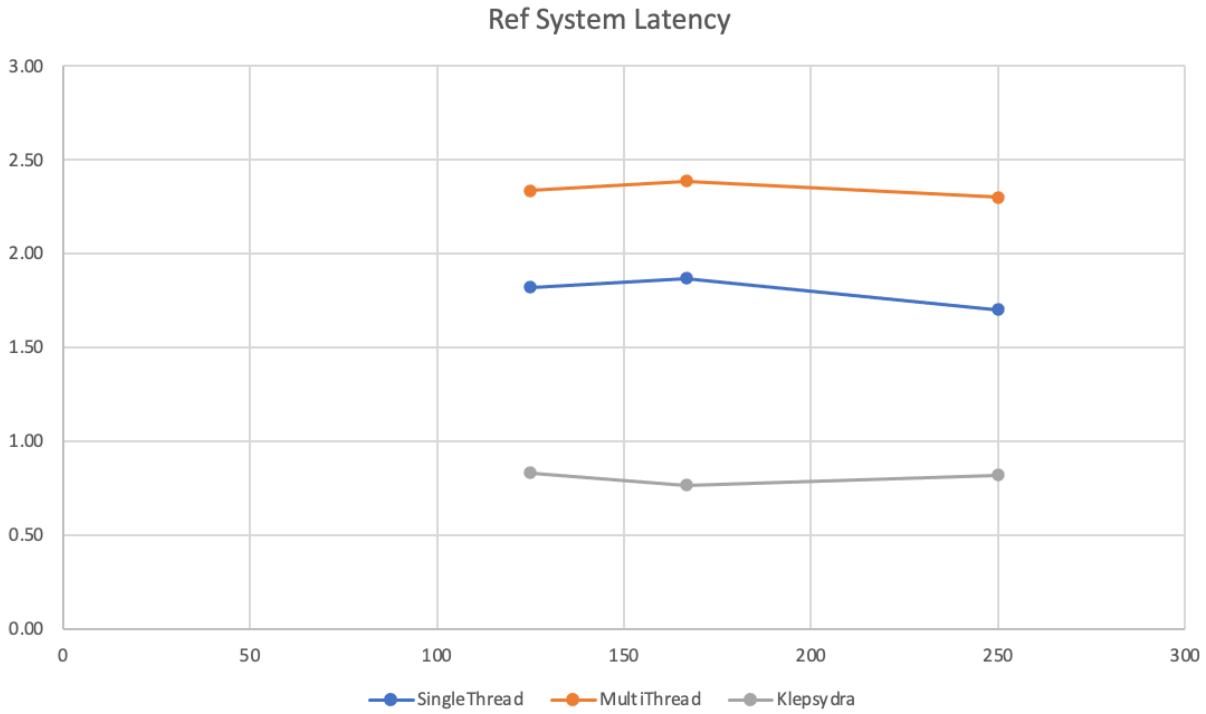


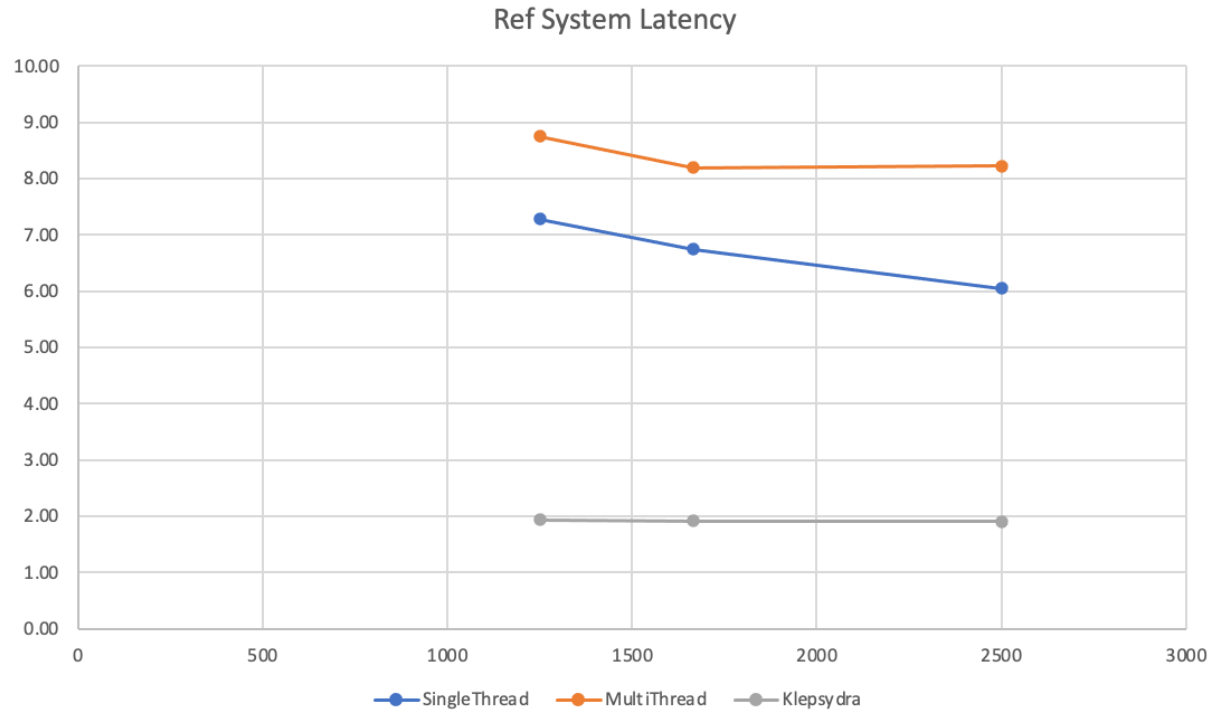






Reference System Latency Comparison





KLEPSYDRA TECHNOLOGIES

CODE EXAMPLE

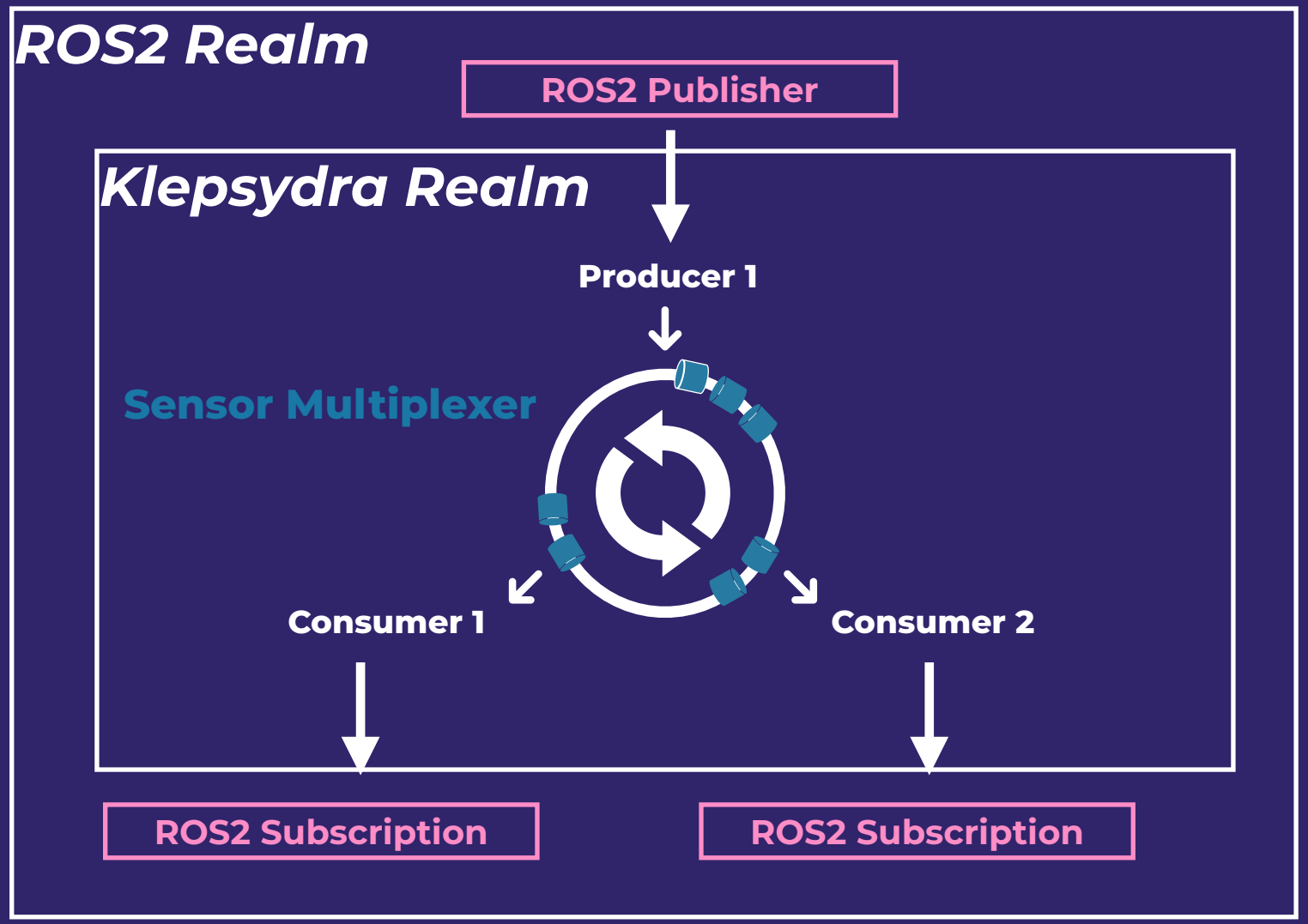
```
#include <kpsr_ros2_executor/executor_factory.hpp>
...

int main(int argc, char** argv) {
    ...
    rclcpp::init(0, nullptr);
    ...
    rclcpp::Executor::SharedPtr exec = kpsr::ros2::ExecutorFactory::createExecutor(kpsr::ros2::QueueSize::_256, false);

    ...
}
```

API Explained

1. Klepsydra offers a factory of executors. Mapping to nodes can be customised via configuration file.
2. Factory returns shared pointer to `rclcpp::Executor`
3. Size of underlying ring-buffers to be provided by constructor. It can be customised via configuration file.
4. “Test” version available (last bool arg). This test version is a synchronous, single-thread, blocking queue.

KLEPSYDRA TECHNOLOGIES

Multiplexer Executor



How does it work?

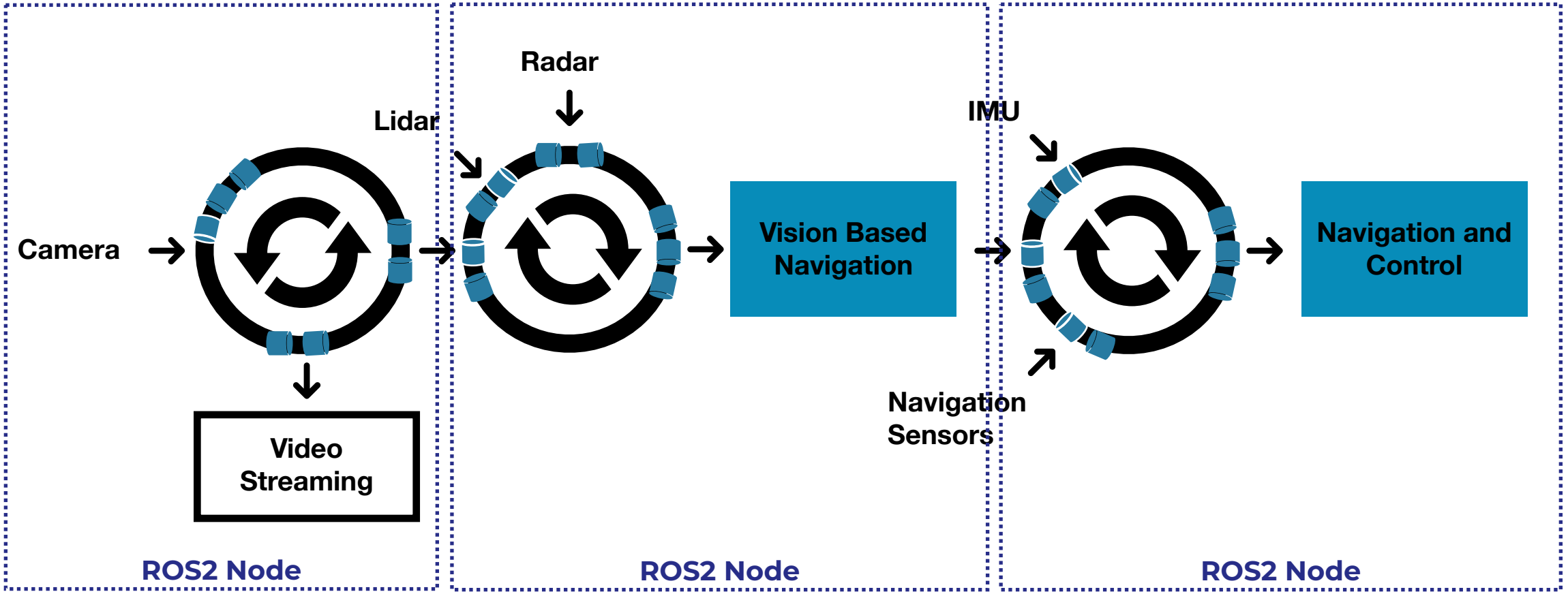
- Each consumer on its own thread
- Processing rate can be anything
- Data integrity guaranteed

Best performance in the following scenarios

- Few producers and many consumers
- Large data sets (LiDAR, Images, etc)
- Power reduction and throughput needs

KLEPSYDRA TECHNOLOGIES

Data streaming approach to ROS2 Executors



Klepsydra Executors can be mapped to nodes (many-to-many) enabling low CPU usage, high throughput and determinism.

KLEPSYDRA TECHNOLOGIES

Conclusions

Benefits

- Low CPU usage
- High throughput with linearly growing CPU usage
- Easy integration.

Best performance in the following scenarios

- Many producers and consumers
- Low to medium data sets
- Power reduction and throughput needs

CONTACT INFORMATION

Dr Pablo Ghiglino

pablo.ghiglino@klepsydra.com

+41786931544

www.klepsydra.com

linkedin.com/company/klepsydra-technologies