

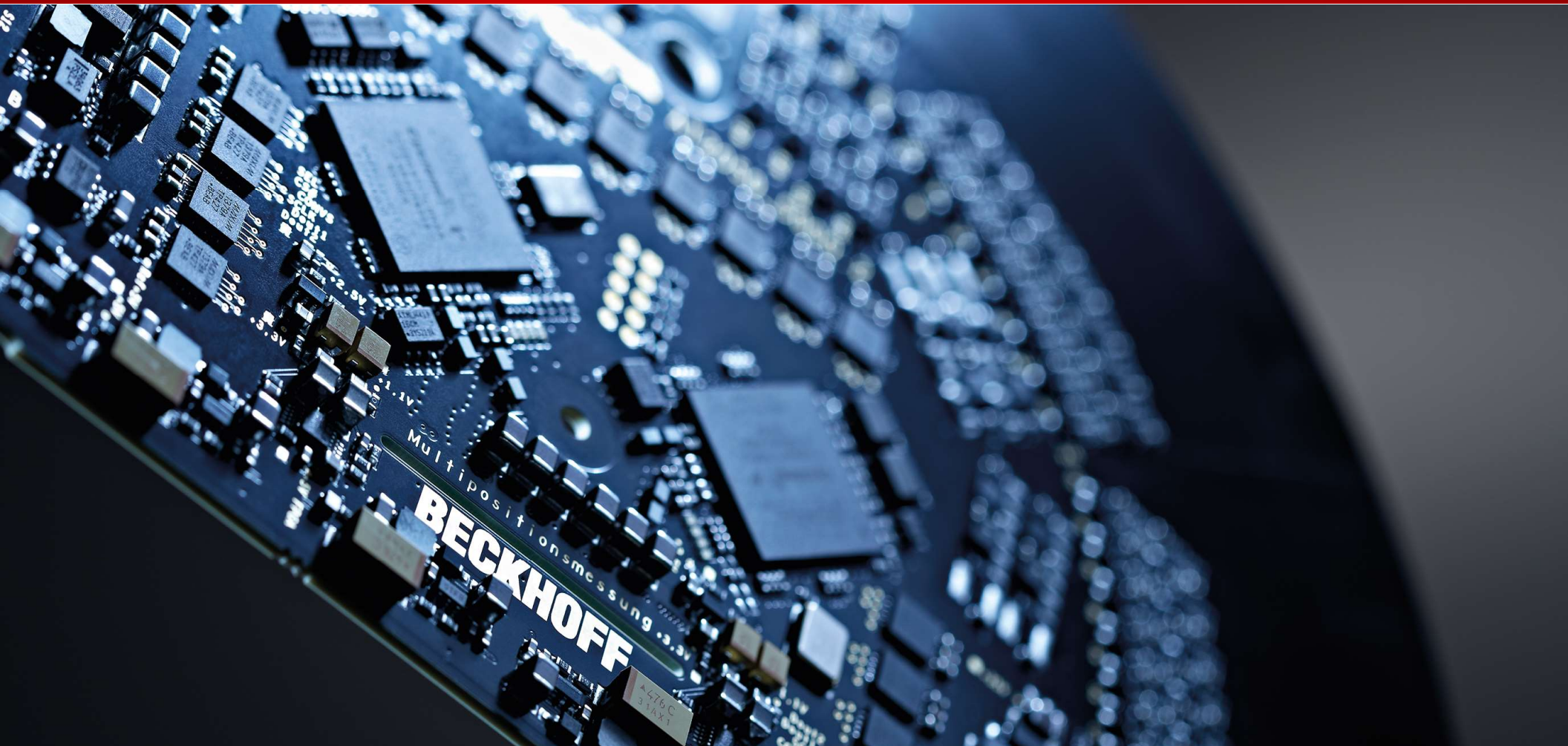
TwinCAT Real-Time

BECKHOFF



Real-time operation

BECKHOFF



But what is real-time?



According to DIN 44300

- Programs for processing data which are continuously operational must be available in a specified period of time
- This is called the cycle time of a system
 - Often confused with scan time



Real-Time Operation - Ambiguous

BECKHOFF

- Definition of real-time is not a hard value
- manufactures use the term to represent different time bases
- Can be very different even in similar products



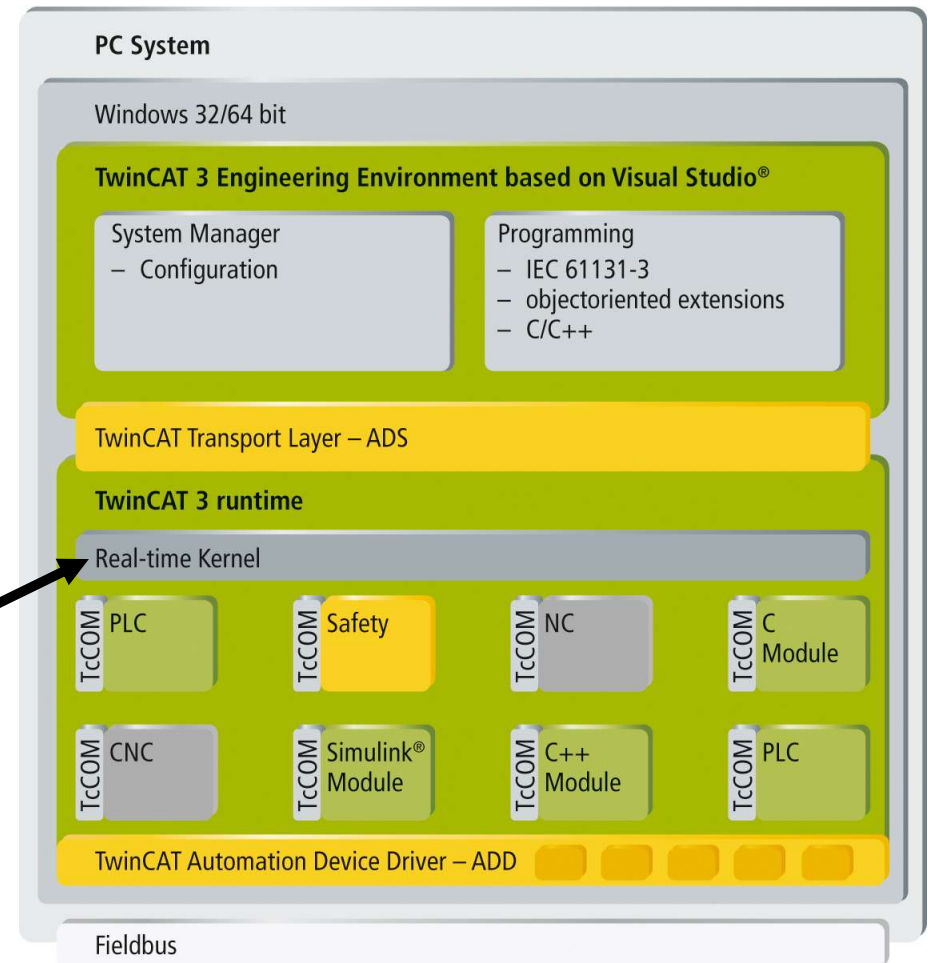
Windows is NOT a real-time operating system*

Generally Real-time operation is not desired for file based systems

*Excluding windows CE

So how does Beckhoff do it?

- TwinCAT is The Windows Control and Automation Technology
- TwinCAT XAR installs to a Windows operating system and turns the PC into a real-time controller
- A **real time** system allows for the configuration of its timing behavior
- This is possible by the addition of a second Kernel



What does this enable us to do?

- Real-time capable scheduling
- Parallel execution of processes
- Direct hardware access

1. **Key Terms and Concepts**
2. Real-time Scheduler
3. Shared core Operation (Default)
4. Isolated core Operation
5. User Mode Runtime

Core	▪ Physical CPU processing core
Tick	▪ a system interrupt representing a pre-determine amount of time
Task	▪ a job requiring CPU calculation time
Priority	▪ Defines the order of tasks to be executed when there are more than one scheduled ▪ Higher priority tasks can interrupt lower priority task even after execution has started

Key concept – User Mode vs Kernel Mode

BECKHOFF

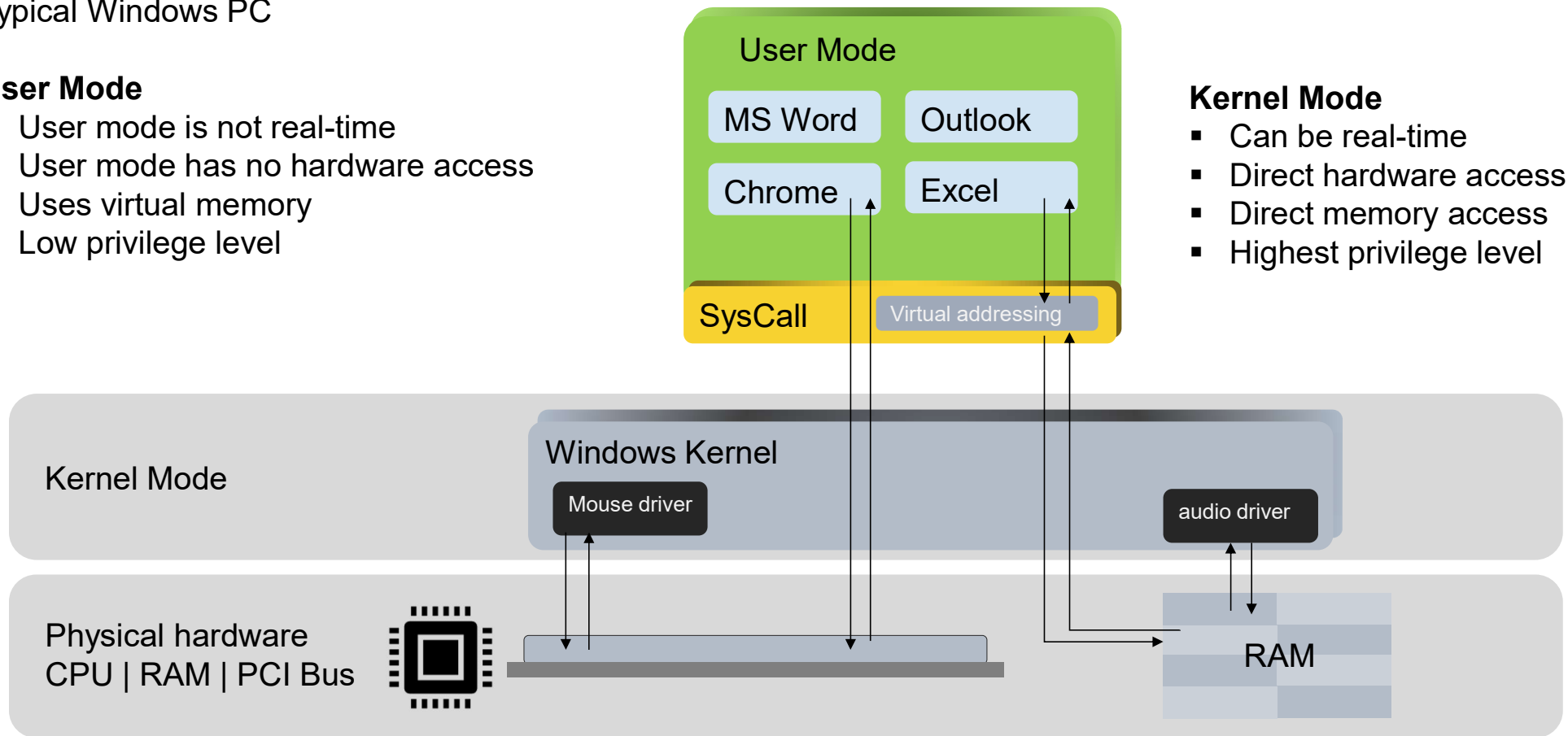
Typical Windows PC

User Mode

- User mode is not real-time
- User mode has no hardware access
- Uses virtual memory
- Low privilege level

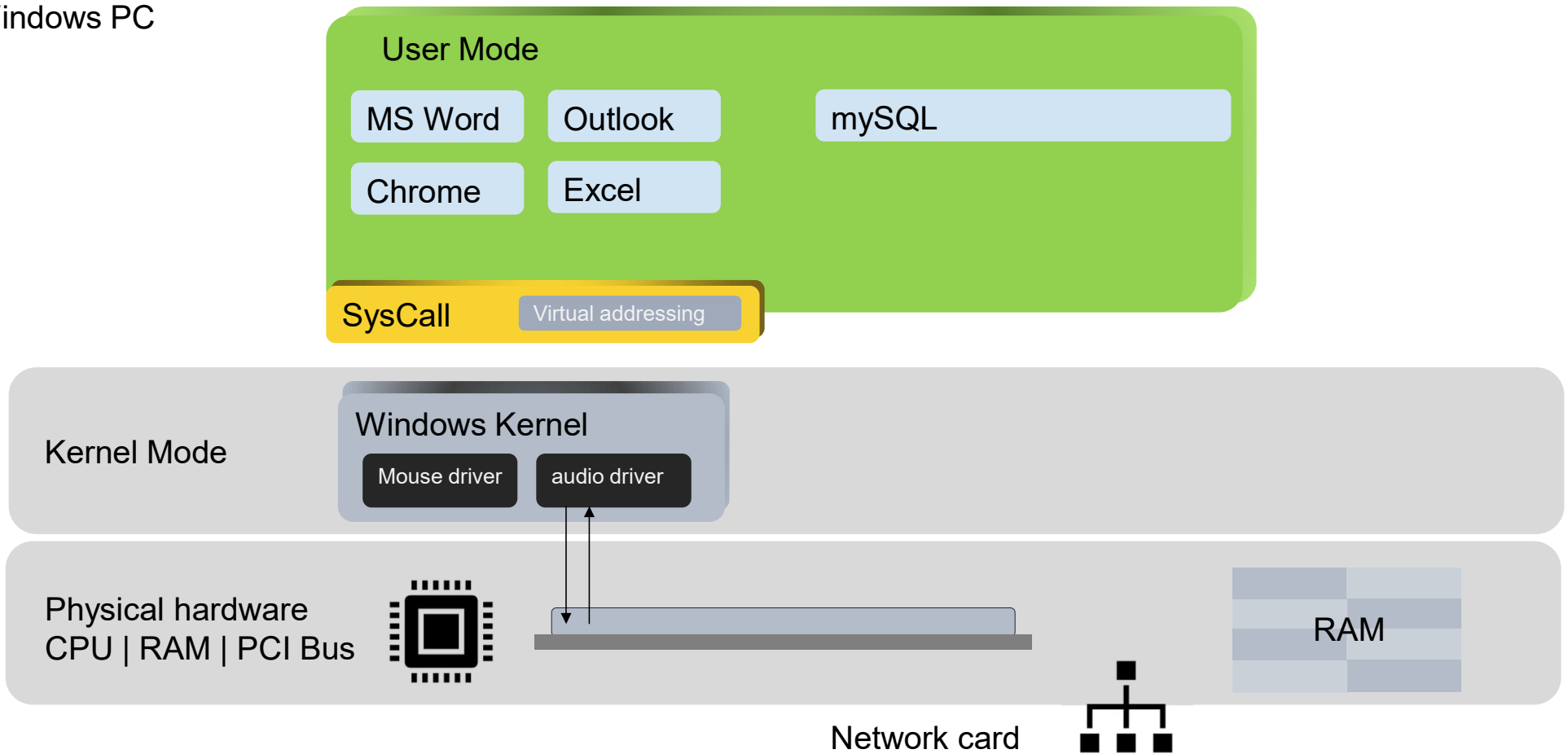
Kernel Mode

- Can be real-time
- Direct hardware access
- Direct memory access
- Highest privilege level



1. Key Terms and Concepts
- 2. Real-time Scheduler**
3. Shared core Operation (Default)
4. Isolated core Operation
5. User Mode Runtime

Windows PC

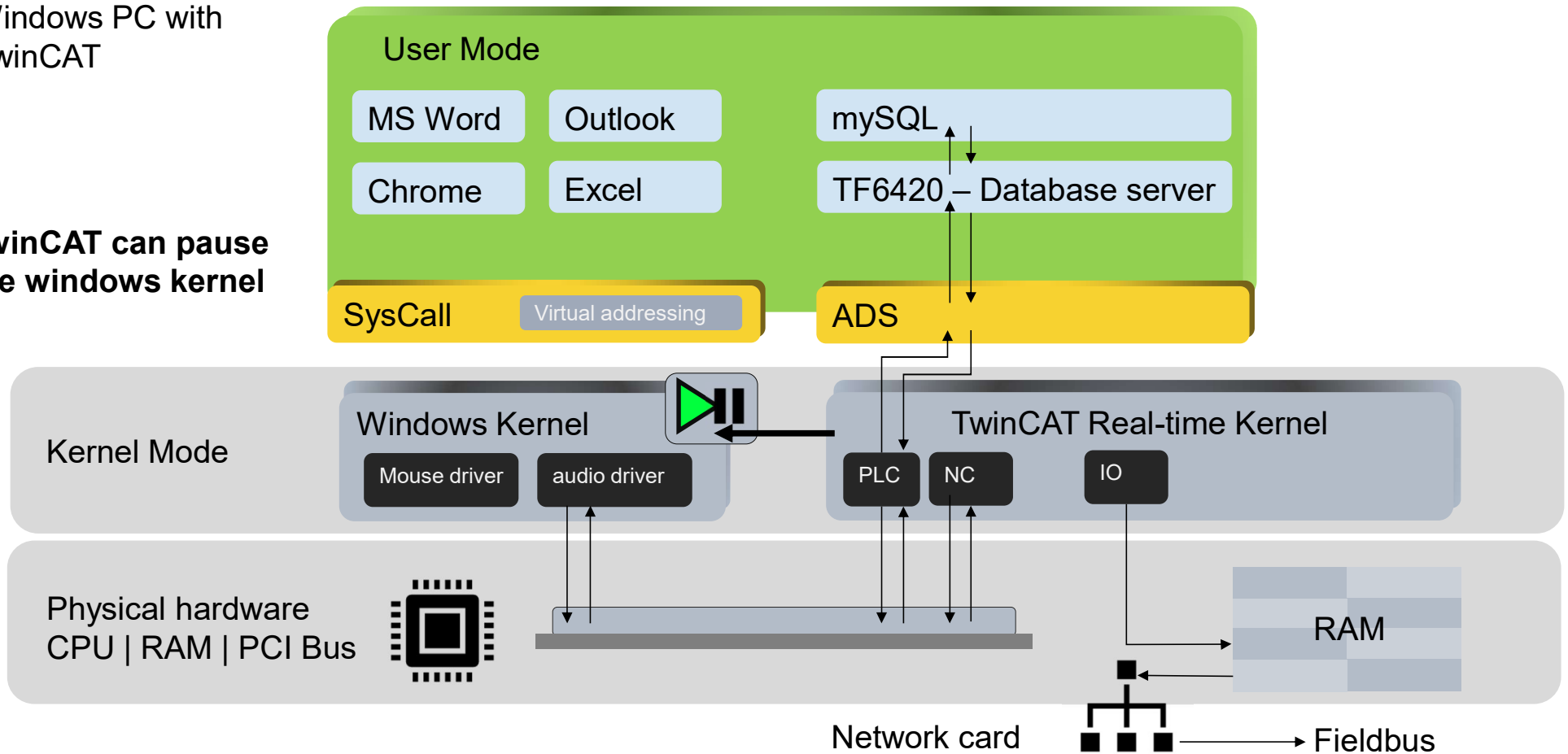


Windows shared Core with TwinCAT

BECKHOFF

Windows PC with
TwinCAT

TwinCAT can pause
the windows kernel

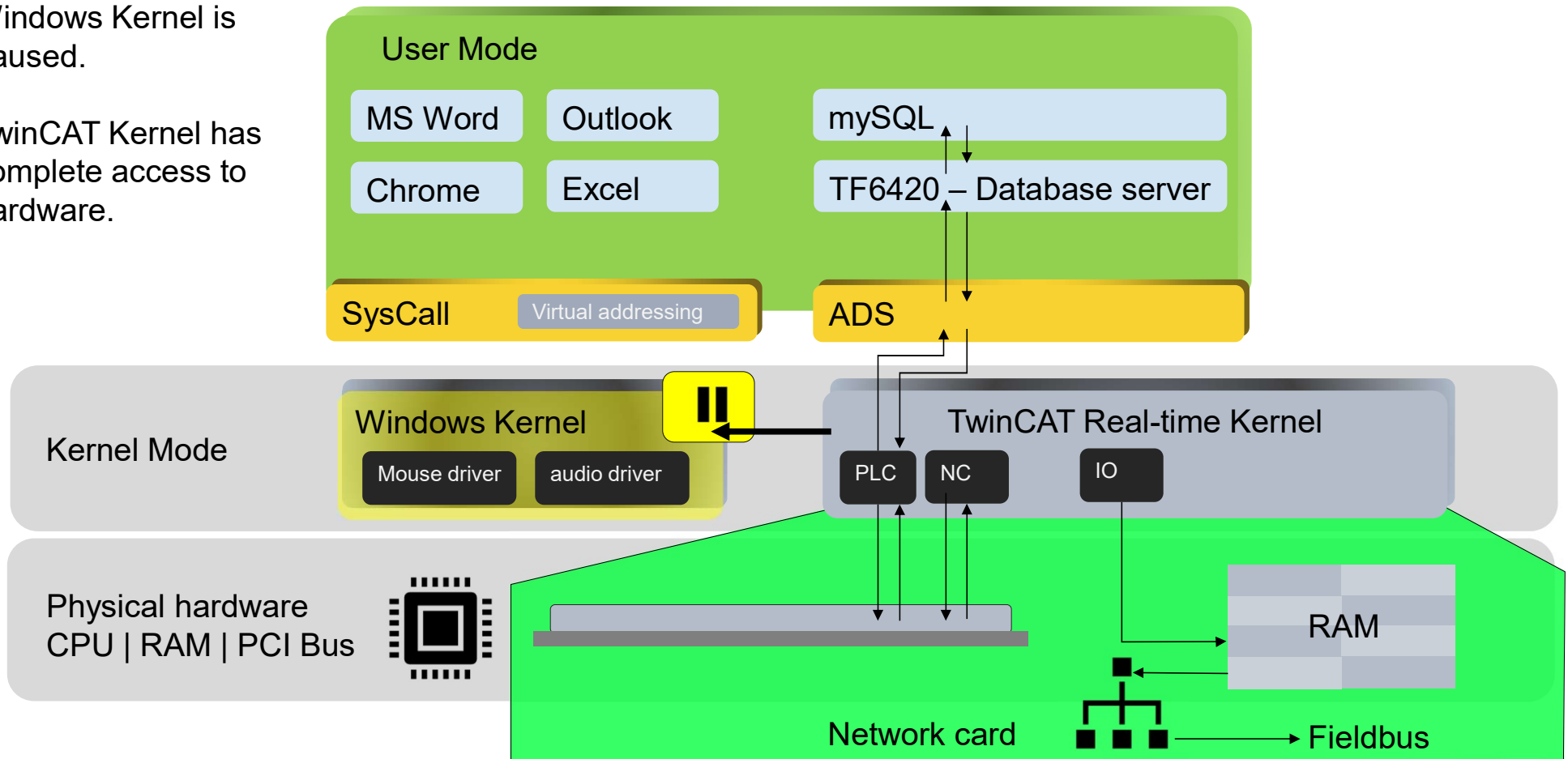


Windows shared Core with TwinCAT - Paused

BECKHOFF

Windows Kernel is paused.

TwinCAT Kernel has complete access to hardware.



TwinCAT uses a real-time scheduler

The scheduler equates a number of CPU clock cycles to time, this generates a system interrupt

This is called a system “tick”

Real-time “tick”

BECKHOFF

- The Real-time tick is set to represent a constant interval of time
- Default is 1ms
- Each core can have a different ‘Base Time’ value

The screenshot shows the TwinCAT Project1 settings window. On the left, the Solution Explorer displays the project structure, with 'Real-Time' selected under 'TwinCAT Project1'. On the right, the 'Settings' tab is active, showing 'Router Memory' and 'Available Cores' settings. Below these, a table lists the 'Base Time' for each core. The 'Base Time' for Core 0 is highlighted with a red box and set to '1 ms'. Below this, another table lists the 'Base Time' for various objects, all set to '1 ms'.

Core	RT-Core	Base Time	Core Limit
0	☒ Default	1 ms	80 %
1	☐		
2	☐		
3	☐		
4	☐		

Object	RT-Core	Base Time
I/O Idle Task	Default (0)	1 ms
PlcTask	Default (0)	1 ms
PlcAuxTask	Default (0)	1 ms

Real-time “tick”

BECKHOFF

- Default is 1ms
- The “tick” can be set as low as 50µs*

Core	RT-Core	Base Time
0	☒ Default	1 ms
1	☐	(none)
2	☐	1 ms
3	☐	500 µs
4	☐	333 µs
5	☐	250 µs
6	☐	200 µs
7	☐	125 µs
8	☐	100 µs
9	☐	83.3 µs
10	☐	76.9 µs
11	☐	71.4 µs
12	☐	66.6 µs
13	☐	62.5 µs
14	☐	50 µs

*Can be lower with guidance from technical support

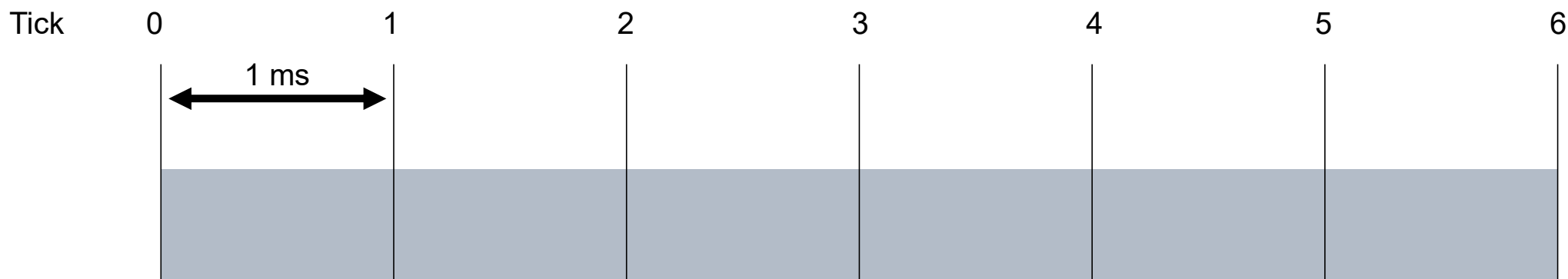
Real-time Scheduler

BECKHOFF

1 tick = 1 unit of Base Time

∴ 1 tick = 1 ms

Core	RT-Core	Base Time
0	☒ Default	1 ms ▼



Time

Real-time Scheduler

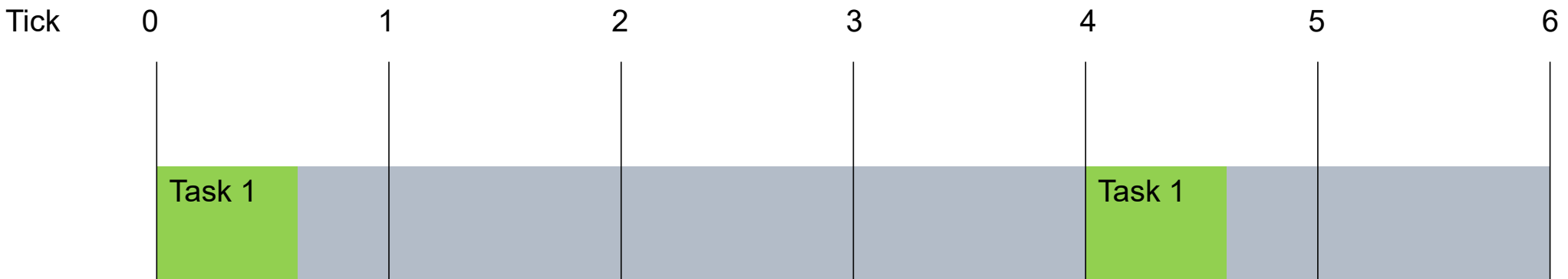
BECKHOFF

1 tick = 1 ms
base time (1ms) x 4 ticks

Cycle Time = 4 ms

Core	RT-Core	Base Time	
0	☒ Default	1 ms	

Object	RT-Core	Base Time (ms)	Cycle Time (ms)	Cycle Ticks
PlcTask	Default (0)	1 ms	4 ms	4



The Real-time scheduler checks for jobs at each tick and schedules tasks accordingly

Time

This is the default setup in TwinCAT

TwinCAT will share the assigned CPU core with Windows on a scheduled task base

Shared Core operation works by pausing the windows Kernel when real-time tasks are started by the system tick.

*The OS is not aware this is happening

CPU Limit – Patented Double tick

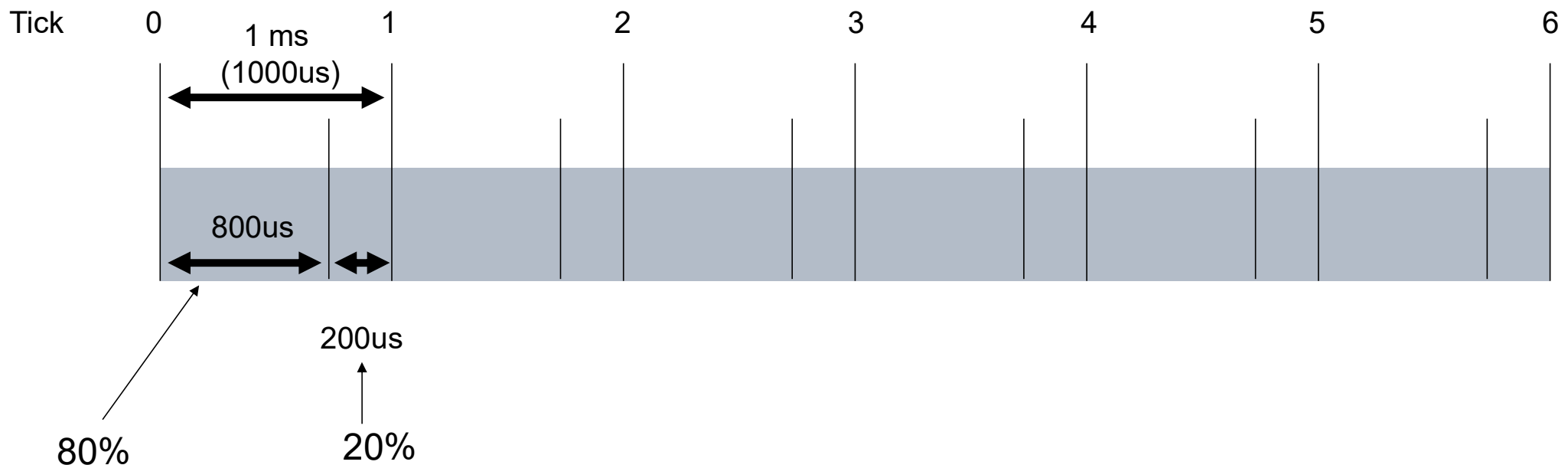
BECKHOFF

TwinCAT uses a second interrupt, called the double tick in order to return the shared core back to windows.

This keeps the OS feeling responsive.

Core	RT-Core	Base Time	Core Limit
0	☒ Default	1 ms	80 %

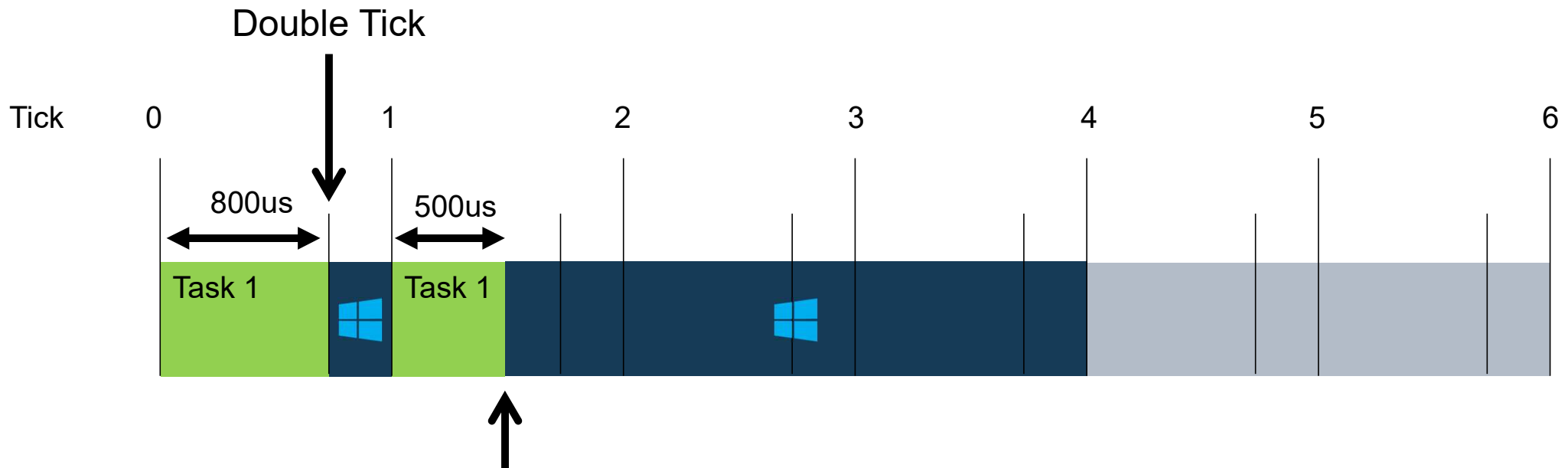
CPU limit dictates the % of time before the double tick



CPU Limit – Patented Double tick

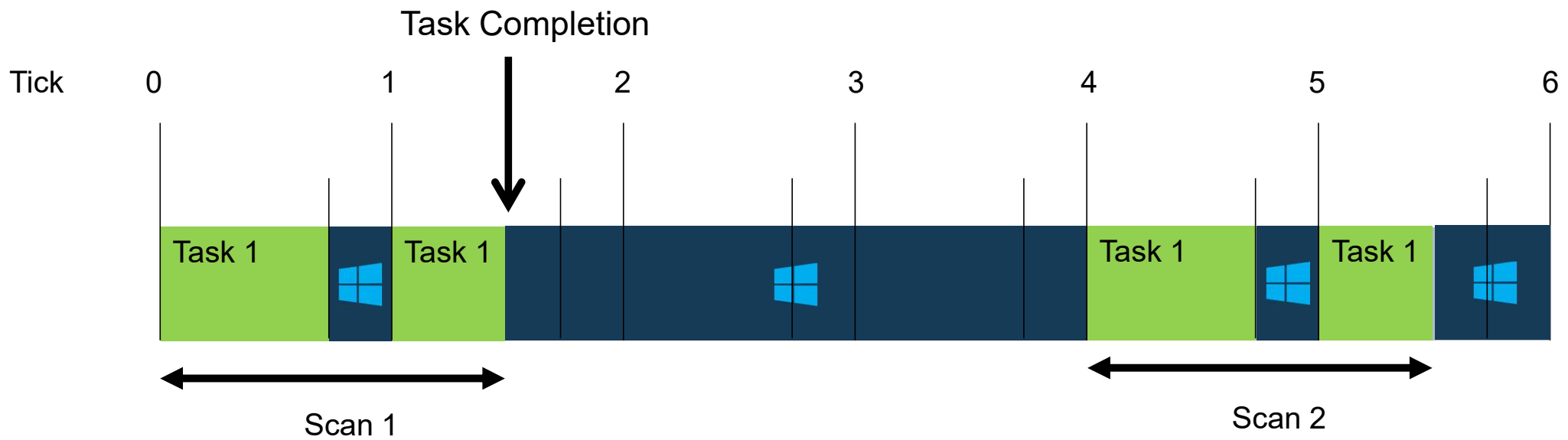
BECKHOFF

If the task needs more time than the core limit %, the double tick is used to give the core back to windows, execution is resumed at the next tick.



If the task finishes before the next double tick, the core is immediately given back to windows.

After the task is run to completion the core is given back to windows, the scheduler waits until the next scheduled task execution to pause windows again (Tick 4)



Example: Task is scheduled every 4 ticks

Shared Core – Timing – CPU time vs Total Time

BECKHOFF

Total Time = Absolute time delta from first start to finish

$$800 + 200 + 500 = 1500\mu s$$

CPU Time = Total time to execute logic while in control of the core

$$800 + 500 = 1300\mu s$$

