Microsoft

SQL Server Architecture, Scheduling, and Waits

Module 1

Learning Units covered in this Module

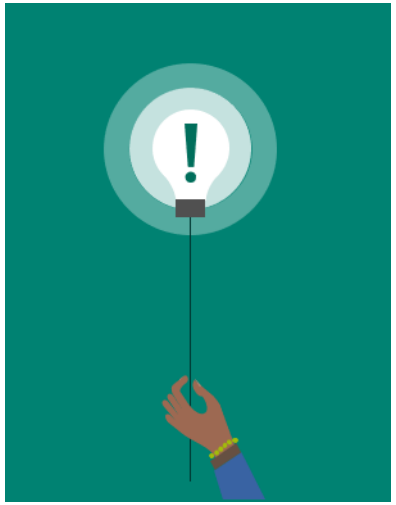
- Lesson 1: Introduction to SQL Operating System
- Lesson 2: SQL Server Task Scheduling
- Lesson 3: SQL Server Waits and Queues

Lesson 1: Introduction to SQL Operating System

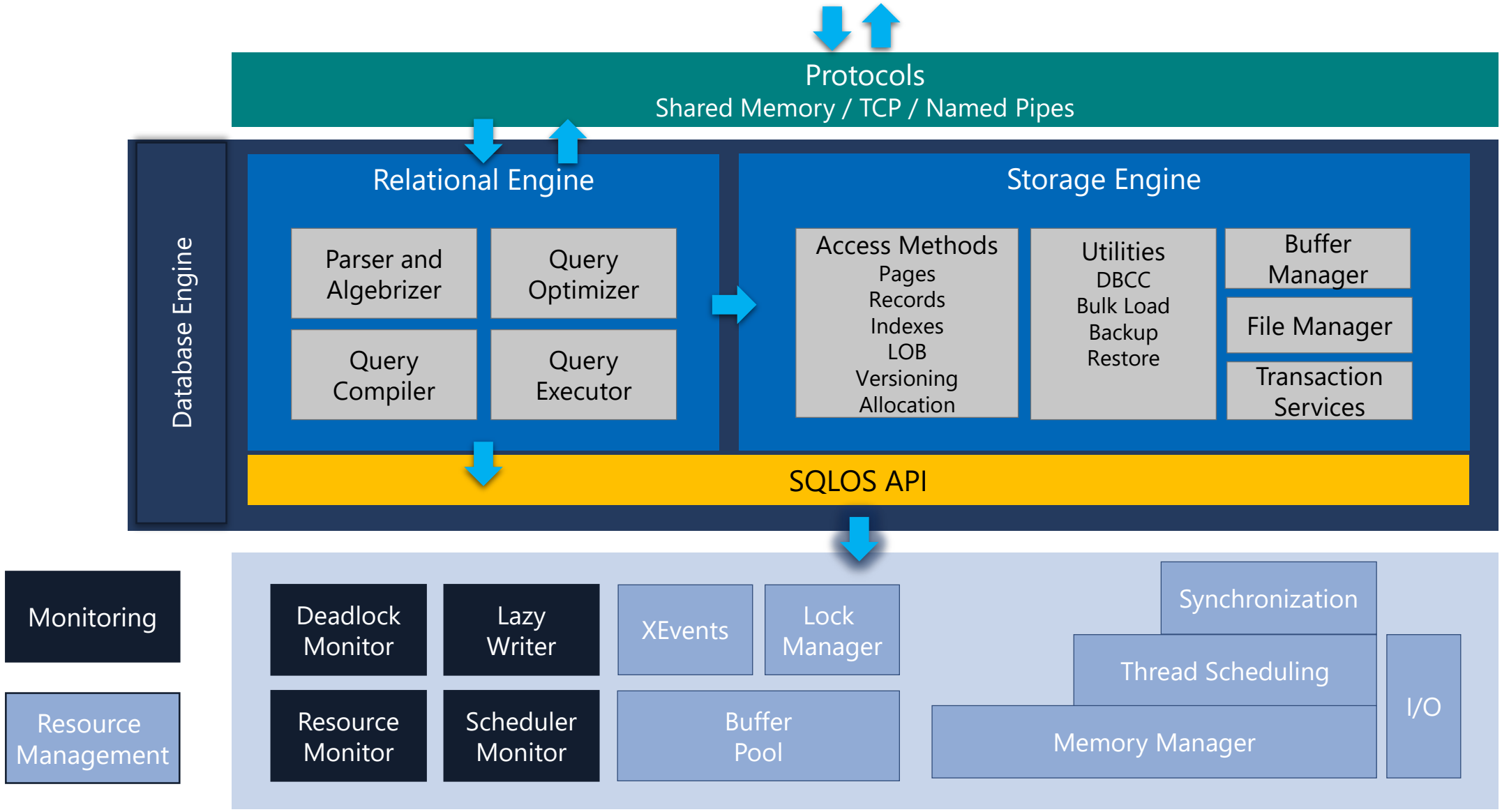
Objectives

After completing this learning, you will be able to:

- Understand Microsoft SQL Server Architecture Overview.
- Learn SQL Server Operating System (SQLOS) Components.



Inside the Database Engine



SQL Server Operating System (SQLOS)

Application layer between Microsoft SQL Server components and the Windows Operating System.

Centralizes resource allocation to provide more efficient management and accounting.

The SQLOS is used by the SQL Server relational database engine for system-level services.

Abstracts the concepts of resource management from components, providing:

- **Scheduling and synchronization support**
- **Memory management and caching**
- **Resource governance**
- **Diagnostics and debug infrastructure**
- **Scalability and performance optimization**

Two Main Functions of SQLOS

Management

- Memory Manager
- Process Scheduler
- Synchronization
- I/O
- Support for Non-Uniform Memory Access (NUMA) and Resource Governor

Monitoring

- Resource Monitor
- Deadlock Monitor
- Scheduler Monitor
- Lazy Writer (Buffer Pool management)
- Dynamic Management Views (DMVs)
- Extended Events
- Dedicated Administrator Connection (DAC)

Dynamic Management Views and Functions

Category	Description
sys.dm_exec_%	Execution and connection information
sys.dm_os_%	Operating system related information
sys.dm_tran_%	Transaction management information
sys.dm_io_%	I/O related information
sys.dm_db_%	Database information

Using Dynamic Management Objects (DMOs)

- Must reference using the sys schema
- Two basic types:
  - Real-time state information
  - Historical information

```
SELECT cpu_count, hyperthread_ratio,
    scheduler_count, scheduler_total_count,
    affinity_type, affinity_type_desc,
    softnuma_configuration, softnuma_configuration_desc,
    socket_count, cores_per_socket, numa_node_count,
    sql_memory_model, sql_memory_model_desc
FROM sys.dm_os_sys_info
```

Demonstration

Dynamic Management Views

Questions?

Knowledge Check

How does the SQLOS improve the efficiency of the SQL Server database engine code?

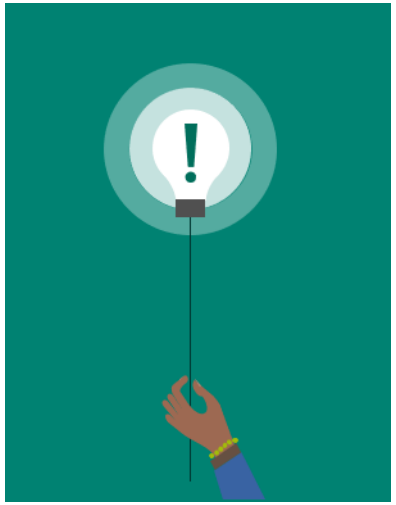
What are the two main functions of the SQLOS?

Lesson 2: SQL Server Task Scheduling

Objectives

After completing this learning, you will be able to:

- Understand SQL Server task scheduling.



Microsoft SQL Server Scheduling Terminology

Batch

- A statement or set of statements submitted to SQL Server by the user (a query), also referred to as a request
- Monitor with sys.dm_exec_requests

Task

- A batch will have one or more tasks (aligns with statements)
- Monitor with sys.dm_os_tasks

Worker Thread

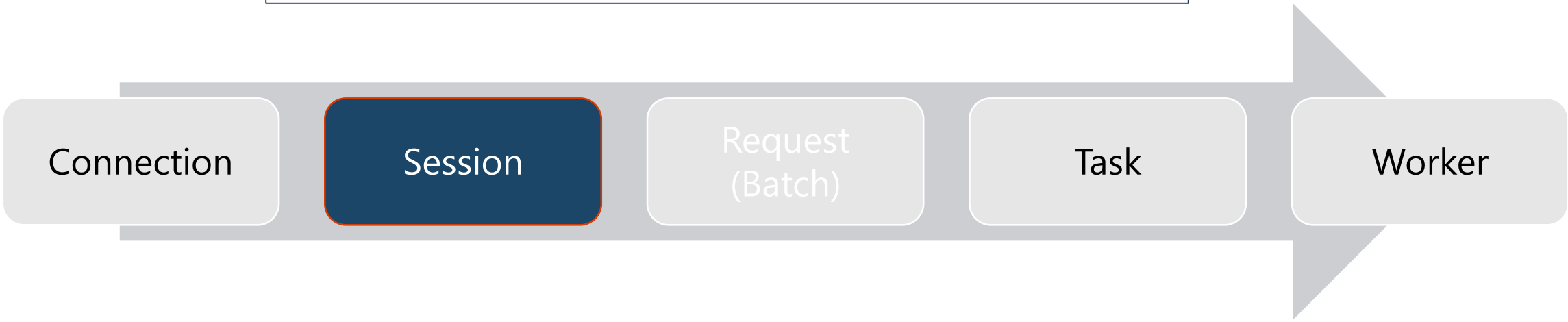
- Each task will be assigned to a single worker thread for the life of the task
- Monitor with sys.dm_os_workers

Hierarchy of Common Terms

```
SELECT *
FROM sys.dm_exec_connections;
-- relevant data:
-- session_id --> spid
-- most_recent_sql_handle --> last query
-- net_transport, protocol_type --> connectivity
```

Hierarchy of Common Terms

```
SELECT *
FROM sys.dm_exec_sessions;
-- relevant data:
-- session_id --> spid
-- host_name, program_name --> client identity
-- login_name, nt_user_name --> login identity
-- status --> activity
-- database_id --> database being accessed
-- open_transaction_count --> blocking identification
```



Hierarchy of Common Terms

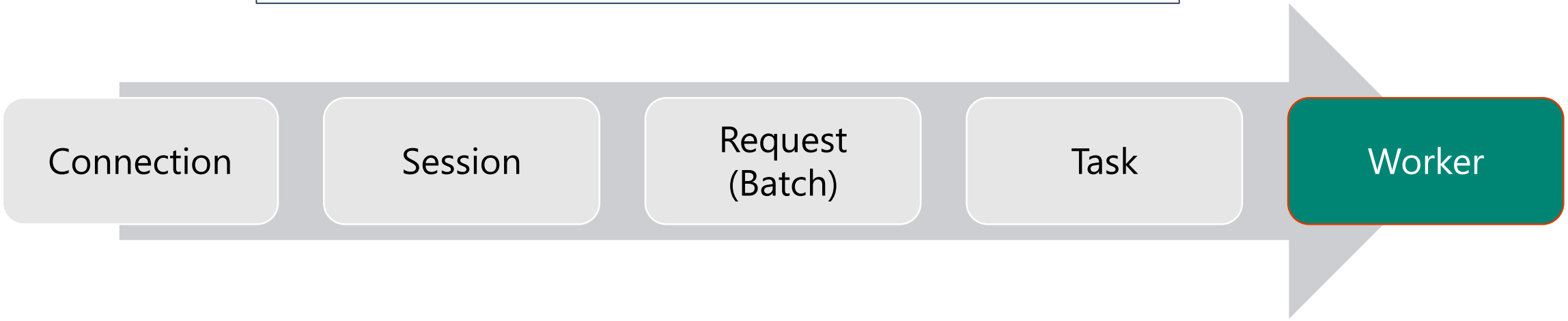
```
SELECT *
FROM sys.dm_exec_requests;
-- relevant data:
-- session_id --> spid
-- status --> background, running, runnable, suspended
-- sql_handle, offset --> query text
-- database_id --> database being accessed
-- wait_type, wait_time --> blocking information
-- open_transaction_count --> blocking others
-- cpu_time, total_elapsed_time, reads, writes --> telemetry
```

Hierarchy of Common Terms

```
SELECT *
FROM sys.dm_os_tasks;
-- relevant data:
-- task_state --> running, suspended
-- pending_io_* --> I/O activity
-- scheduler_id -->processor info
-- session_id --> spid
```

Hierarchy of Common Terms

```
SELECT *
FROM sys.dm_os_workers;
-- relevant data:
-- worker_address --> memory address of the worker
-- wait_start_ms_ticks --> Point in time worker Suspended.
-- wait_resumed_ms_ticks --> Worker in Runnable state.
-- state -- > Running, Runnable, Suspended
```



Hierarchy of Common Terms

```
-- Using DMVs to see sessions currently executing.

SELECT *
FROM sys.dm_exec_connections as c
JOIN sys.dm_exec_sessions as s
ON c.session_id = s.session_id
JOIN sys.dm_exec_requests as r
ON s.session_id = r.session_id
JOIN sys.dm_os_tasks as t
ON r.session_id = t.session_id
JOIN sys.dm_os_workers as w
ON t.task_address = w.task_address
```

Scheduling Types

Non-Preemptive (Cooperative)

- SQL Server manages CPU scheduling for most activity (instead of the operating system).
- SQL Server decides when a thread should wait or get switched out (known as yielding).

Preemptive

- Preemption is the act of an operating system temporarily interrupting an executing task.
- Higher priority tasks can preempt lower priority tasks.
- Preemptive mode used in SQL Server for external code calls, CLR with an UNSAFE assemblies, extended stored procedures

SQL Server Task Scheduling

One SQLOS Scheduler per core/logical processor

Handles scheduling tasks, I/O, and synchronization of resources

Work requests are balanced across schedulers based on the number of active tasks

Monitor using sys.dm_os_schedulers

Worker Migration – better use of schedulers in SQL Server 2019

Yielding

SQL Server using a cooperative model to ensure its CPU resources are shared amongst all the threads that are in a runnable state, preventing the 'starving' condition of any individual thread.

In SQL Server, each thread is assigned a quantum (duration 4ms).

By design, a worker owns the scheduler until it yields to another worker on the same scheduler.

When no worker is currently on the Runnable list, the **yielding worker is allowed another quantum** or performs the necessary idle scheduler maintenance.

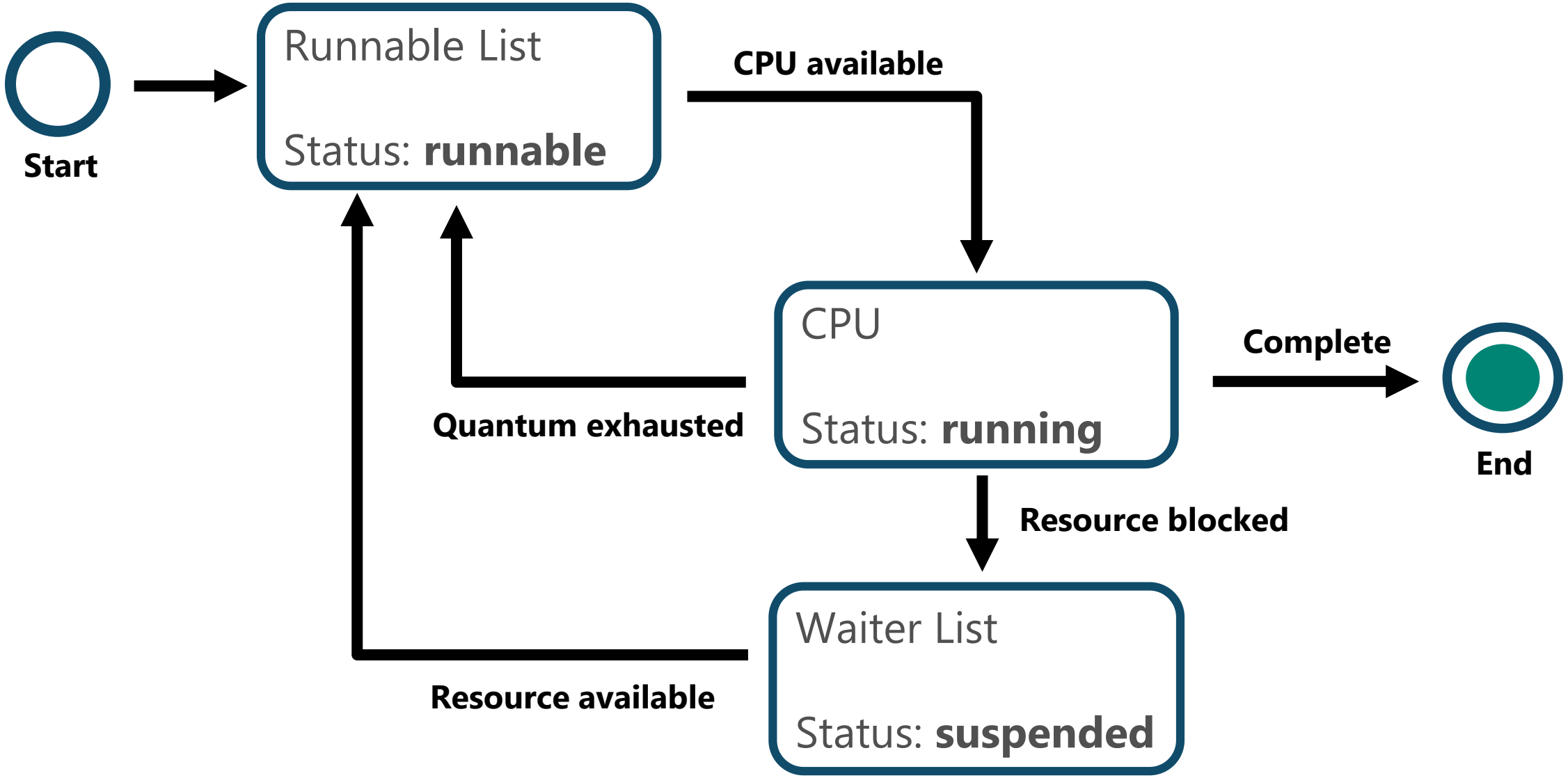
Thread States and Queues

Runnable: The thread is currently in the Runnable Queue waiting to execute. (First In, First Out).

Running: One active thread executing on a processor.

Suspended: Placed on a Waiter List waiting for a resource other than a processor. (No specific order).

Yielding


Start
Runnable List
Status: runnable
CPU available
CPU
Status: running
Complete
End
Quantum exhausted
Resource blocked
Waiter List
Status: suspended
Resource available

Demonstration

Runnable Tasks

Questions?

Knowledge Check

List three reasons a scheduler might yield?

What is a quantum?

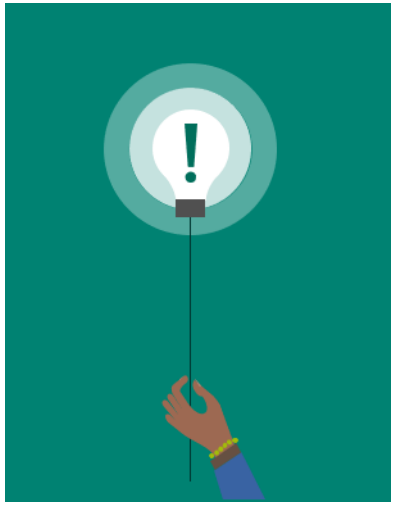
Why does SQL Server use cooperative scheduling?

Lesson 3: SQL Server Waits and Queues

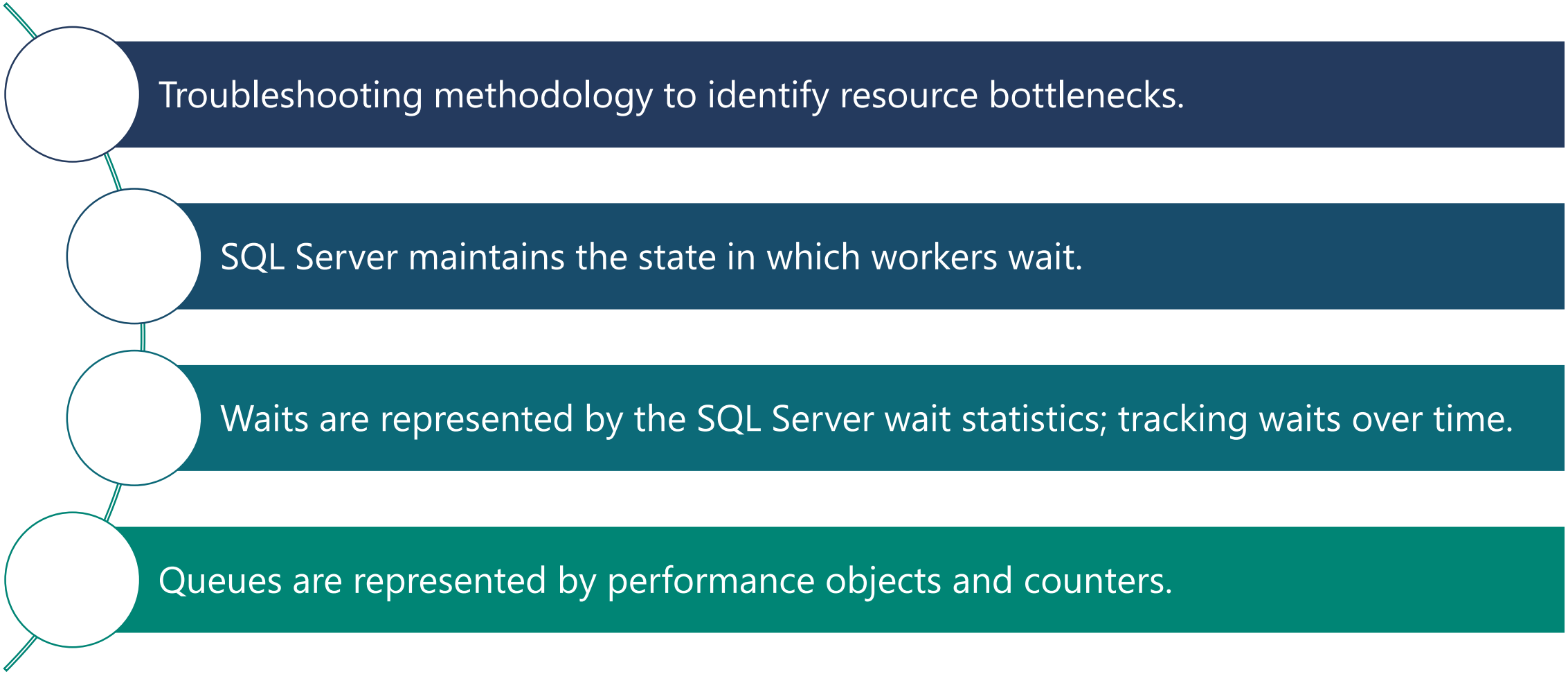
Objectives

After completing this learning, you will be able to:

- Using Waits and Queues to troubleshoot resource bottlenecks.



Waits and Queues



Using Waits and Queues

Useful to assist in troubleshooting an active performance issue

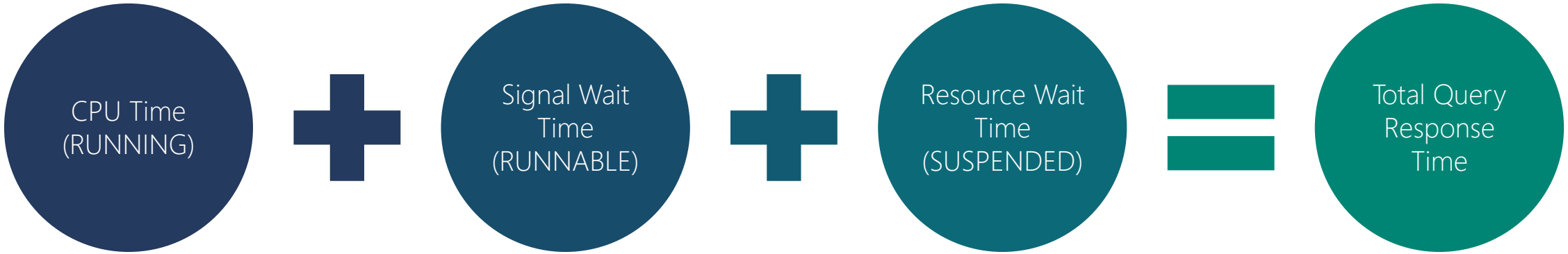
Valuable to track the resources SQL Server is regularly waiting on

Useful for workload measurements and benchmarking

Valuable for identifying performance trends

Task Execution Model

- The full cycle between the several task states, for how many times it needs to cycle, is what we experience as the total query response time.



Task Execution Model

Status: Running

session_id 51	Running

Runnable Queue (Signal Waits)
Status: Runnable

session_id 51	Runnable
session_id 64	Runnable
session_id 87	Runnable
session_id 52	Runnable
session_id 56	Runnable

SPID56 moved to the bottom of the Runnable queue.

Wait Queue (Resource Waits)
Status: Suspended

session_id 73	LCK_M_S
session_id 59	NETWORKIO
~~session_id 56~~	~~Runnable~~
session_id 55	RESOURCE_SEMAPHORE
session_id 60	IO_Completion

Relevant Dynamic Management Views (DMVs)

sys.dm_os_wait_stats

- Returns information about all the waits encountered by threads that ran.
- Includes wait type, number of tasks that waited in the specific wait type, total and max wait times, and the amount of signal waits.

sys.dm_os_waiting_tasks

- Returns information about the wait queue of tasks actively waiting on some resource.

sys.dm_exec_requests

- Returns information about each request that is in-flight.
- Includes session owning the request and status of the request, which will reflect the status of one or more tasks assigned to the request.

Waiting Tasks DMV

```
SELECT w.session_id, w.wait_duration_ms, w.wait_type,
       w.blocking_session_id, w.resource_description,
       s.program_name, t.text, t.dbid, s.cpu_time, s.memory_usage
 FROM sys.dm_os_waiting_tasks as w
      INNER JOIN sys.dm_exec_sessions as s
         ON w.session_id = s.session_id
      INNER JOIN sys.dm_exec_requests as r
         ON s.session_id = r.session_id
      OUTER APPLY sys.dm_exec_sql_text (r.sql_handle) as t
  WHERE s.is_user_process = 1;
```

session_id	wait_duration_ms	wait_type	blocking_session_id	resource_description
58	8563	LCK_M_S	62	keylock hobtid=72057594047365120 dbid=5 id=lock1...

Notable Wait Types

- SOS_SCHEDULER_YIELD
- CXPACKET
- PAGELATCH_xx
- PAGEIOLATCH_xx
- IO_COMPLETION
- ASYNC_IO_COMPLETION
- WRITELOG
- CMEMTHREAD
- RESOURCE_SEMAPHORE
- LCK_xx

Demonstration

Examining SQL Server Scheduling

Waits and queues

- Using waits and queues troubleshooting methodology

Questions?

Knowledge Check

While troubleshooting a resource bottleneck what two objects should you examine?

What are the three components that make up query execution time?

What is the difference between a PAGELATCH wait and a PAGEIOLATCH wait?

Which DMV should you examine to view historical wait information?

Which DMV should you examine to view current wait information?