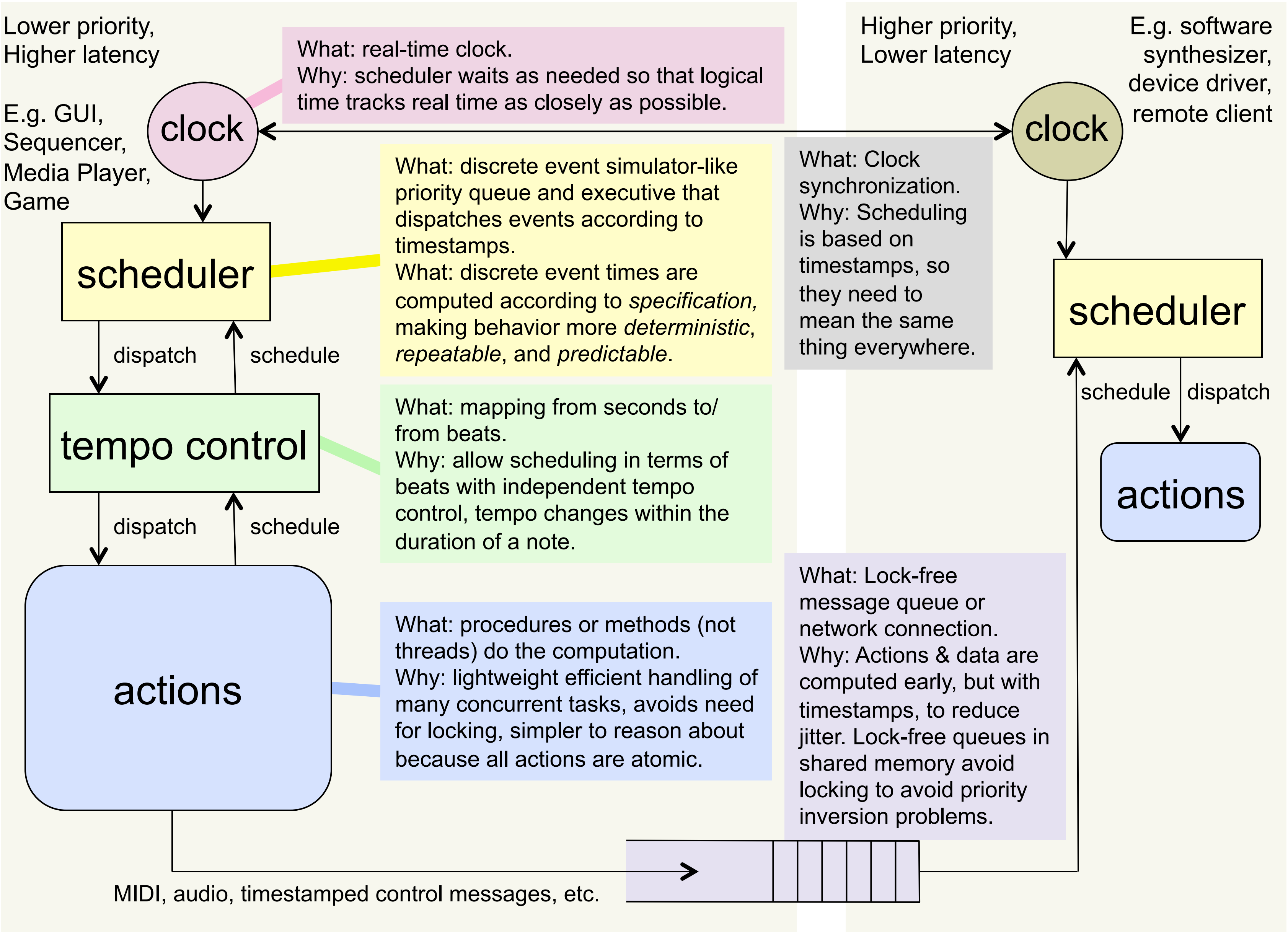
Grand Guide to Interactive Real-Time Systems



Lower priority,
Higher latency

E.g. GUI,
Sequencer,
Media Player,
Game

clock

What: real-time clock.
Why: scheduler waits as needed so that logical time tracks real time as closely as possible.

scheduler

What: discrete event simulator-like priority queue and executive that dispatches events according to timestamps.
What: discrete event times are computed according to *specification*, making behavior more *deterministic*, *repeatable*, and *predictable*.

dispatch

schedule

tempo control

What: mapping from seconds to/ from beats.
Why: allow scheduling in terms of beats with independent tempo control, tempo changes within the duration of a note.

dispatch

schedule

actions

What: procedures or methods (not threads) do the computation.
Why: lightweight efficient handling of many concurrent tasks, avoids need for locking, simpler to reason about because all actions are atomic.

MIDI, audio, timestamped control messages, etc.

What: Clock synchronization.
Why: Scheduling is based on timestamps, so they need to mean the same thing everywhere.

What: Lock-free message queue or network connection.
Why: Actions & data are computed early, but with timestamps, to reduce jitter. Lock-free queues in shared memory avoid locking to avoid priority inversion problems.

Higher priority,
Lower latency

E.g. software
synthesizer,
device driver,
remote client

clock

scheduler

schedule

dispatch

actions