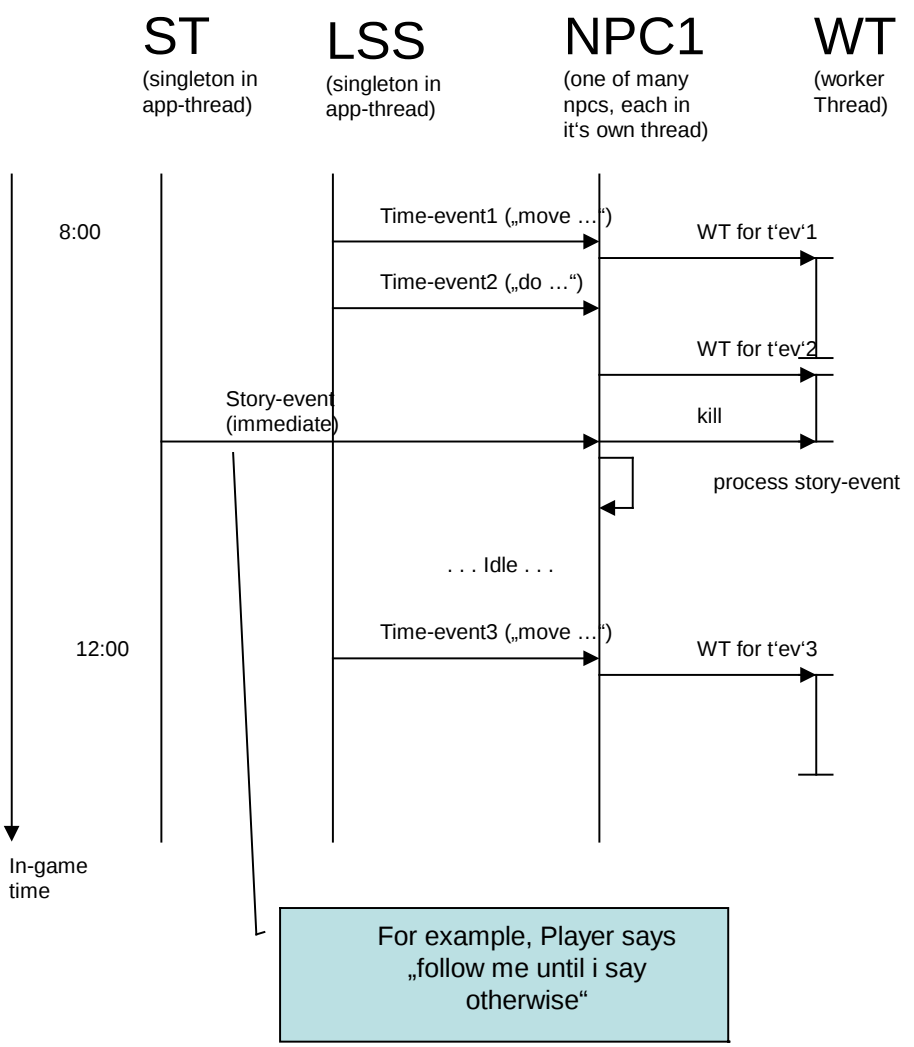
Anatomy of a Lifescript

Lifescript (LS) for „NPC1“:

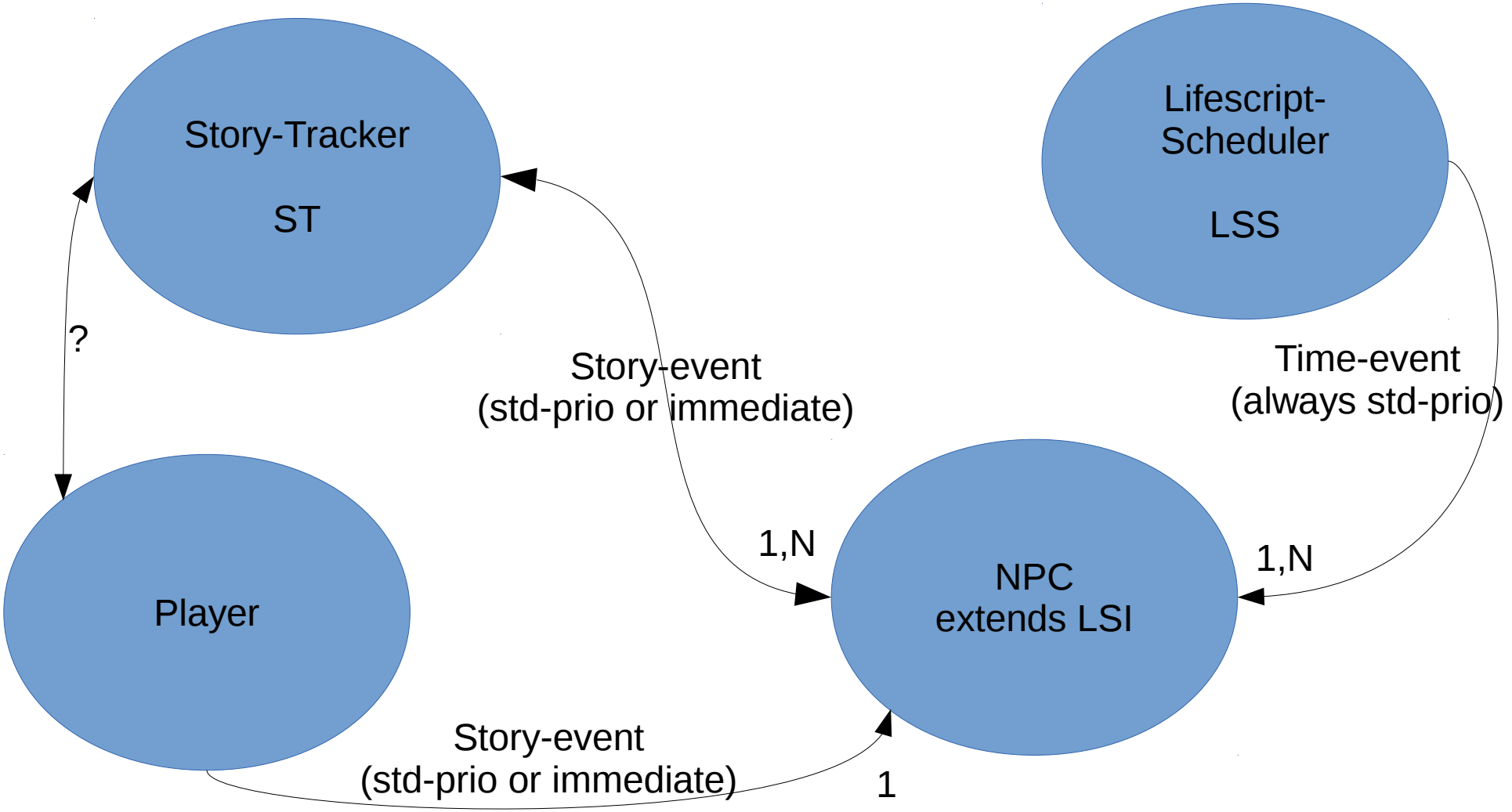
```
@8:00
move to_work
do busy_animation
@12:00
move to_cantine
```



- **class NPC1 extends LSI**
- LSI has prioritized FIFO Queue
- Prio(story-event immediate) > Prio(Time-event)

- ST = Story-Tracker
- LSS = LifeScript Scheduler
- LSI = LifeScript Interpreter
- NPC = non-player character
- WT = worker thread (disposable)
- LS = lifescript
- SG = scene graph

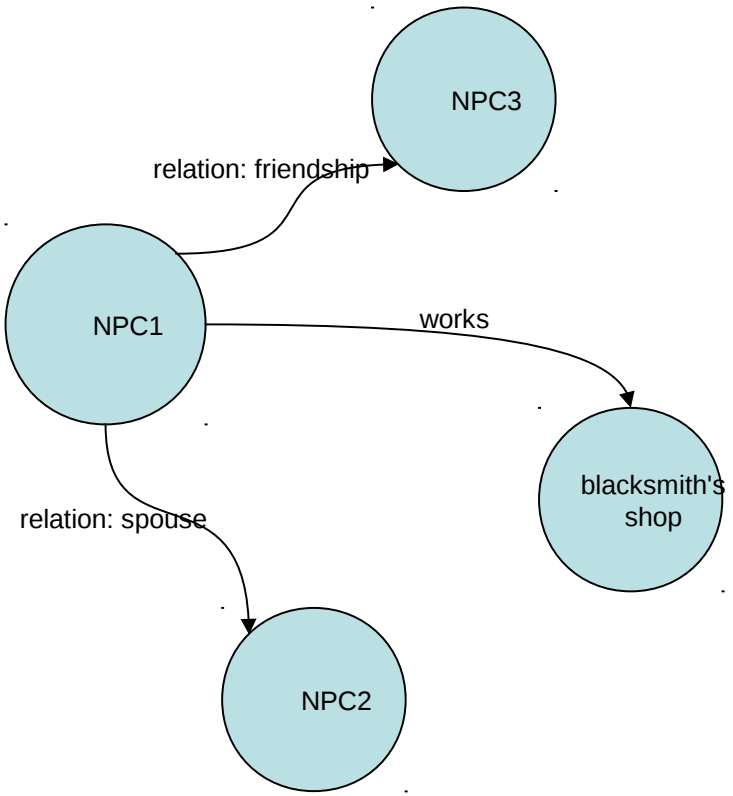
Events


Story-Tracker
ST
Lifescript-
Scheduler
LSS
?
Story-event
(std-prio or immediate)
Time-event
(always std-prio)
1,N
1,N
Player
NPC
extends LSI
Story-event
(std-prio or immediate)
1

LSS

- NPC <-(0,N)------(0,1)-> lifescript-file
- LSS has a game-time clock
- At init-time reads data from lifescript files
- And builds this data-stucture:
  - { time, {npc-id, [events]} }
  - (note: []=vector, {}=map)
- At runtime, looks up current time in data-structure and sends events to npc

In-Game-Object-Graph (IGOG)



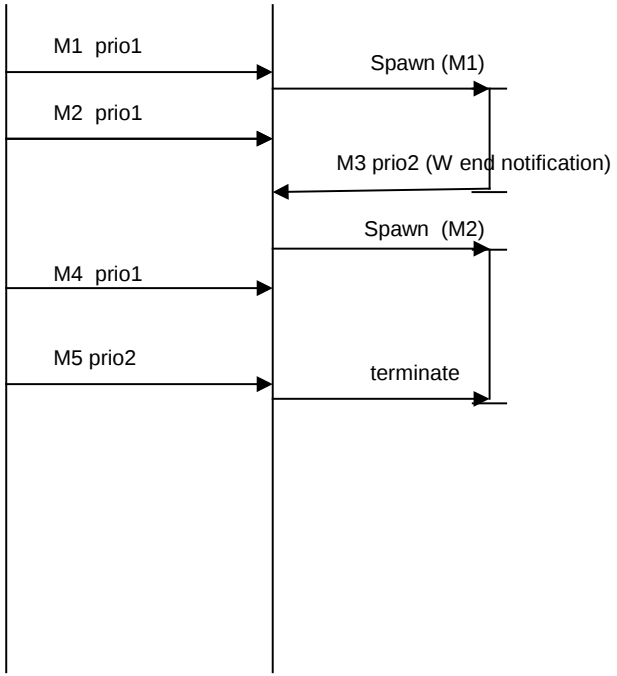
Which „manager“ operates on which „datastructure“:

	LS	IGOG	SG
LSS	ro	-	-
LSI	-	rw	ro
WT	-	-	rw
Player	-	rw	rw
ST	-	rw	ro
3D Engine	-	-	rw

•LSI operates on (reads/writes) IGOG
•LSI asks IGOG „where is the location of the work of NPC1?“
•IGOG returns id of blacksmith's shop
•LSI asks scene-graph (SG) for spatial coordinates of blacksmith's shop per id
•LSI creates worker-thread for moving
•Worker-thread operates on (reads/writes) SG, does pathfinding, collision detection and terminates when destination reached
•SG is maintained by 3d-engine

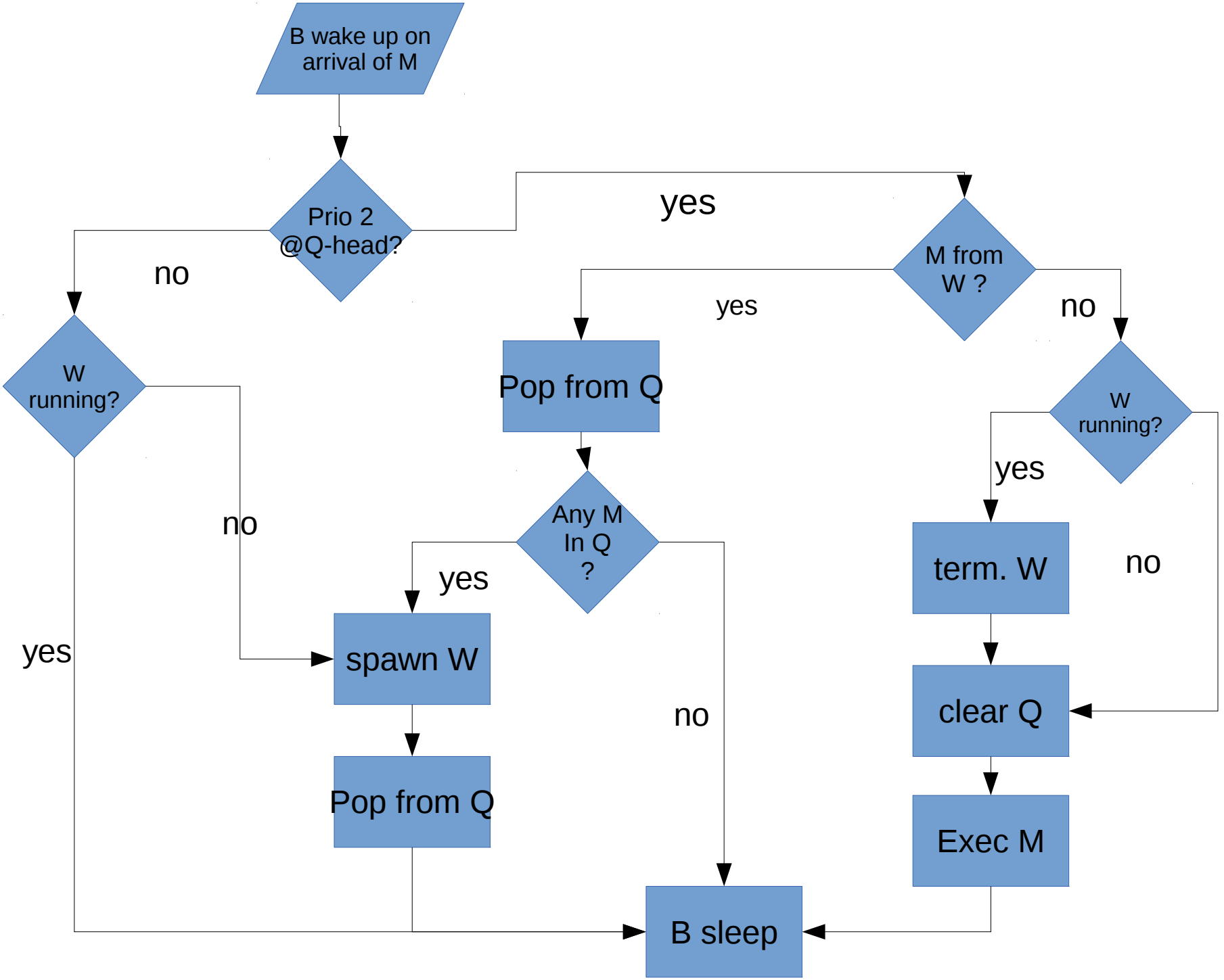
•ST = Story-Tracker
•LSS = LifeScript Scheduler
•LSI = LifeScript Interpreter
•NPC = non-player character
•WT = worker thread (disposable)
•LS = lifescript
•SG = scene graph

Threads A B W



Note:

- M = “Message”
- W = “Worker Thread”
- A = “App-Main Thread”
- B = “per NPC Thread”
- Prio2 (“immediate”) > Prio1 (“std”)
- Immediate from A clear Q
- B has priority-Q
- B controls max 1 W
- M1, M2, M4 from LSS
- M5 from Story-Tracker or Player
- M3 from W
- LSS & ST in same thread
- Prio 2 M are executed in thread B


B wake up on arrival of M
Prio 2 @Q-head?
yes
no
M from W ?
yes
no
W running?
Pop from Q
W running?
yes
no
Any M In Q ?
yes
no
term. W
no
yes
spawn W
clear Q
no
Pop from Q
Exec M
B sleep

TODO

- End-able (from within and from outside) Worker Thread
- Wake up on arrival of M
- Agents?!