Anatomy of a Lifescript

Lifescript (LS) for „NPC1":

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

ST
(singleton in app-thread)

LSS
(singleton in app-thread)

NPC1
(one of many npcs, each in it's own thread)

WT
(worker Thread)

8:00

Time-event1 („move …")

WT for t'ev'1

Time-event2 („do …")

WT for t'ev'2

Story-event
(immediate)

kill

process story-event

. . . Idle . . .

12:00

Time-event3 („move …")

WT for t'ev'3
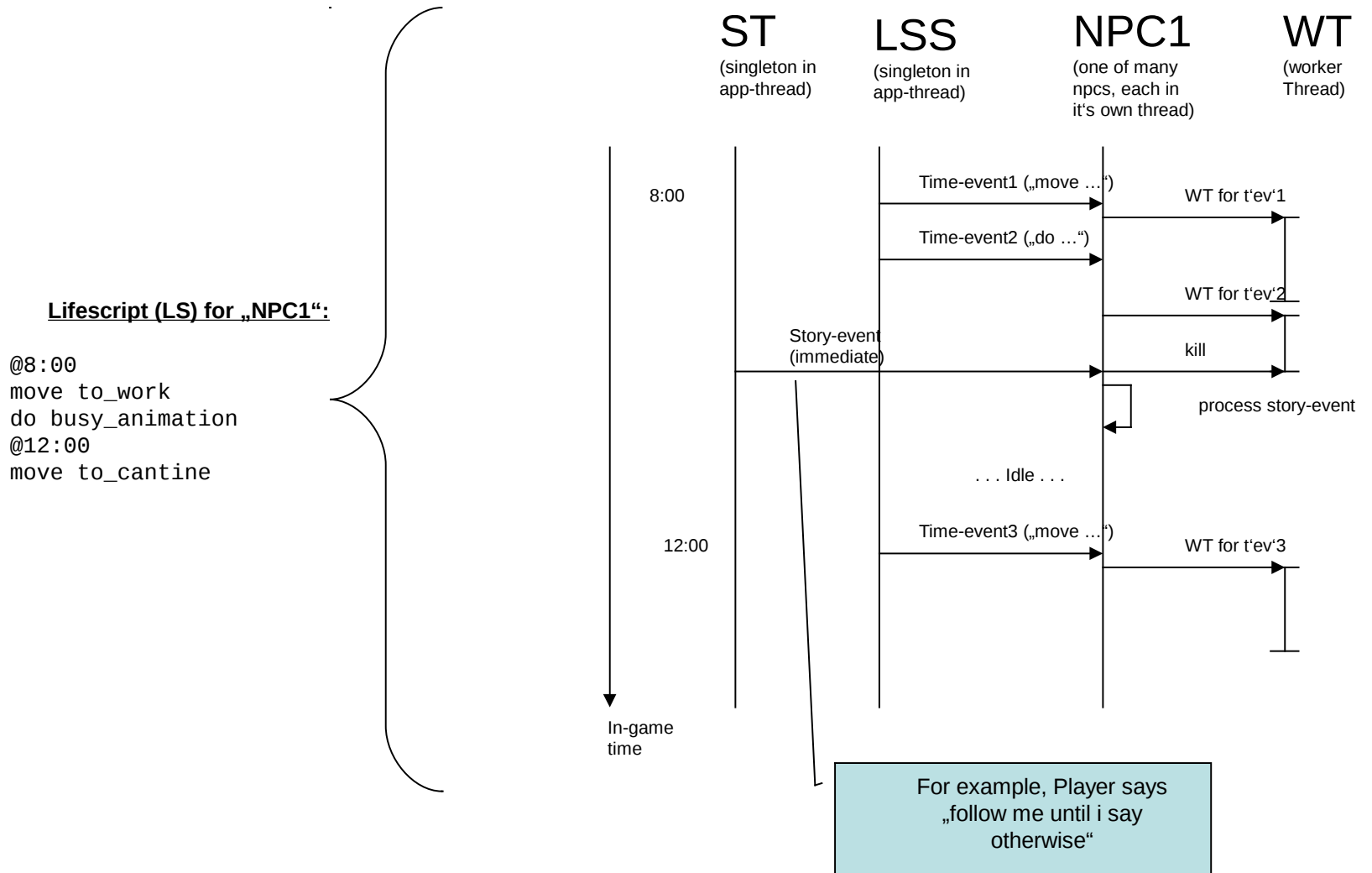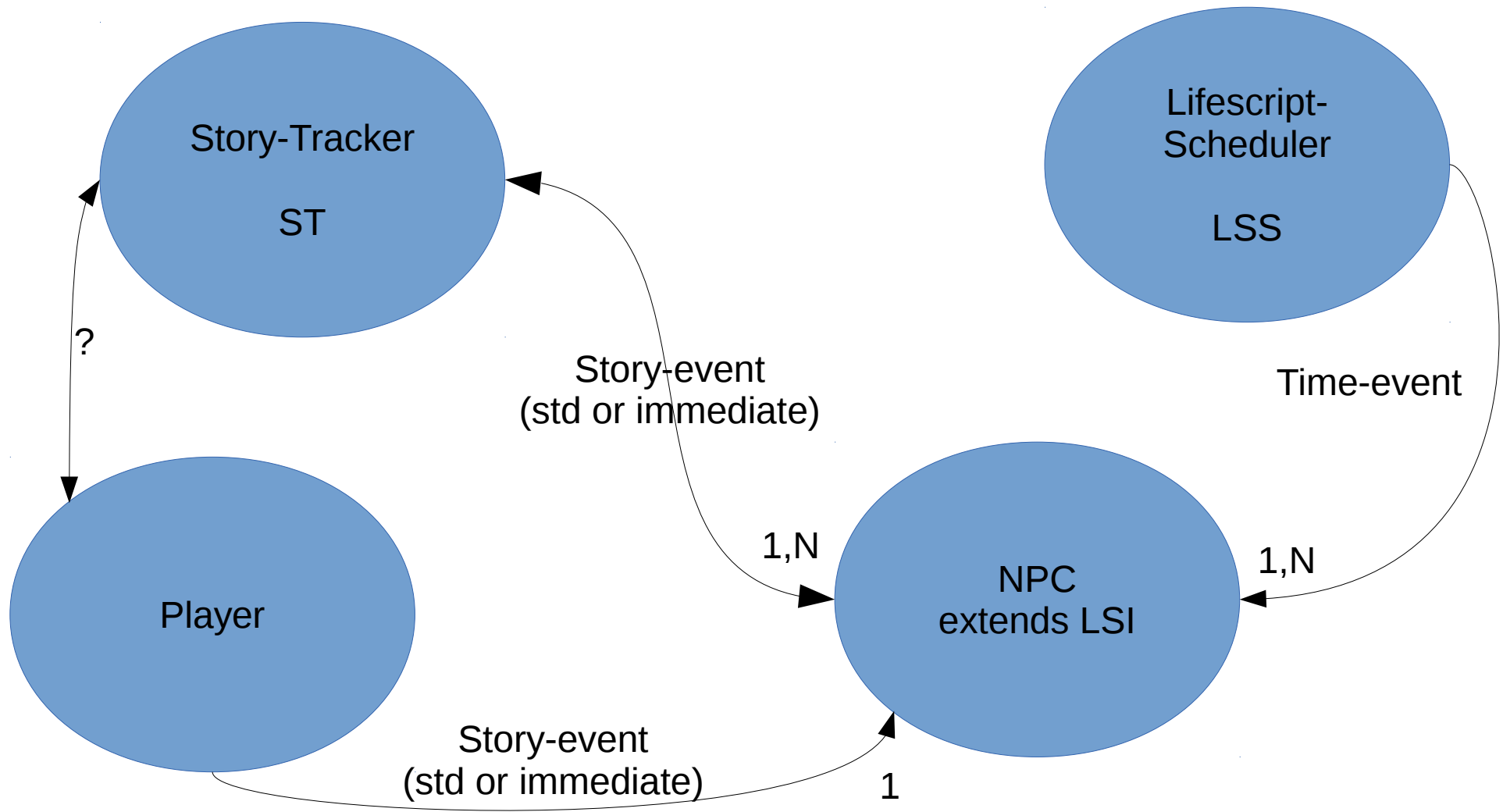
In-game time

For example, Player says „follow me until i say otherwise"

•**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
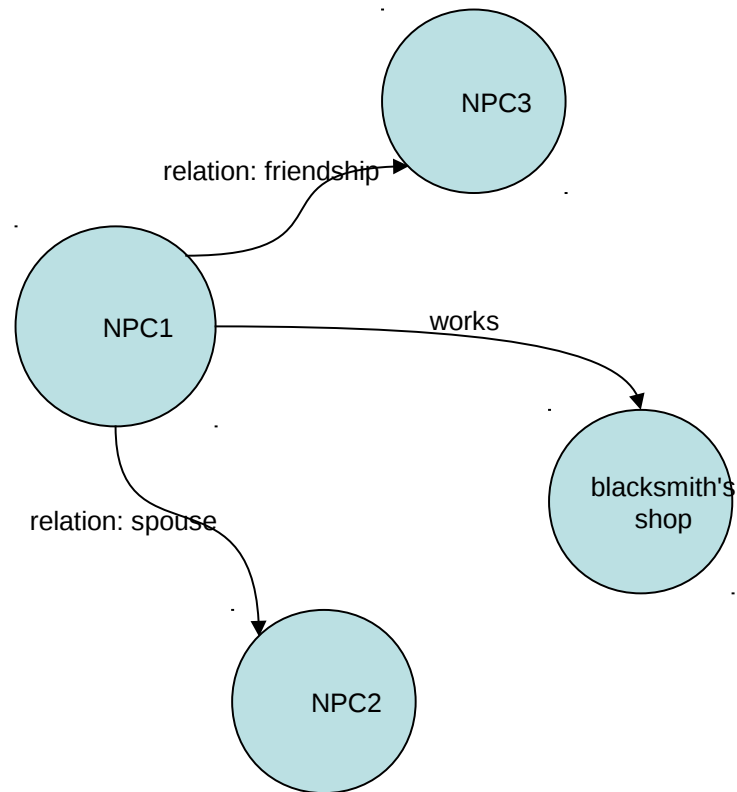•WT = worker thread (disposable)
•LS = lifescript
•SG = scene graph

Events

Story-Tracker

ST

Lifescript-
Scheduler

LSS

?

Story-event
(std or immediate)

Time-event

1,N

1,N

Player

NPC
extends LSI

Story-event
(std 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)

NPC3

relation: friendship

NPC1

works

relation: spouse

blacksmith's shop

NPC2

Which „manager" operates on which „datastructure":

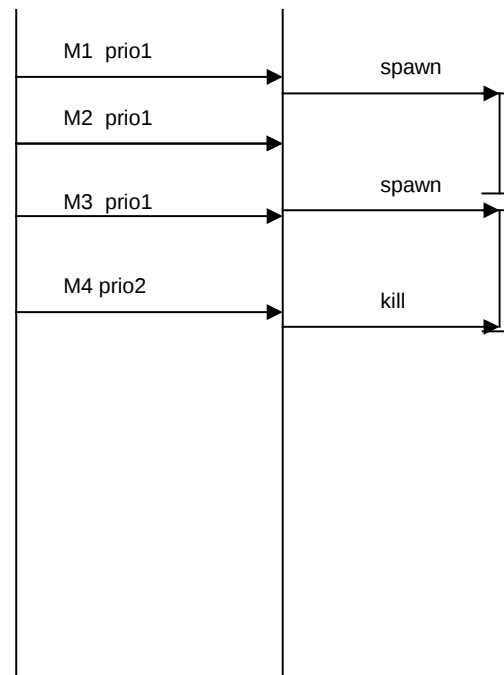|	LS	IGOG	SG
LSS	ro	-	-
LSI	-	rw	ro
WT	-	-	rw
Player	-	rw	rw
ST	-	rw	ro
3D Engine	-	-	rw

- 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

M1 prio1

spawn

M2 prio1

spawn

M3 prio1

M4 prio2

kill

Note: Prio2 > Prio1

B wake up on arrival of M

- **Prio 2?**
  - **no** →
    - **W running?**
      - **yes** → B sleep
      - **no** → spawn W → Pop from Q → B sleep
  - **yes** →
    - **W running?**
      - **yes** → kill W → clear Q → exec → B sleep
      - **no** → Clear Q → exec → B sleep

B wake up on Termination of W

- **Any M In Q ?**
  - **no** → B sleep
  - **yes** → spawn W → Pop from Q → B sleep