Design Review

Gruppe 46

“Success is not final, failure is not fatal: it is the courage to continue that counts.” -Winston S. Churchill

Failure modes

- Network failure
- Elevator motor failure
- Incorrect message data received
- Power loss

- Nuclear war (or fire, etc.)

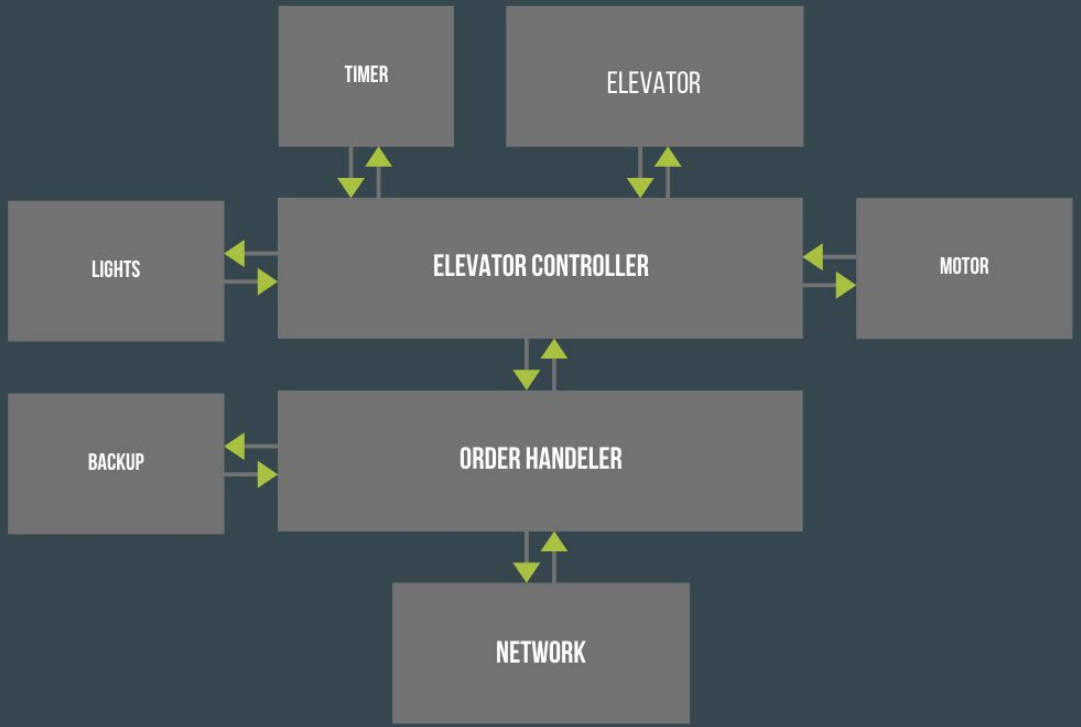
Strengths and weaknesses

+ Elevators function independently (easily scalable)
+ No master/slave overhead over network
+ Modular message-passing design

- Messages are crucial and need redundancy
- Multiple failures at the same time cannot guarantee orders
- File synchronization is difficult to manage

Modules

- Controller
  - Lights
  - Motor
- Order module
  - Backup
- Order module
- Timer module
- Network module

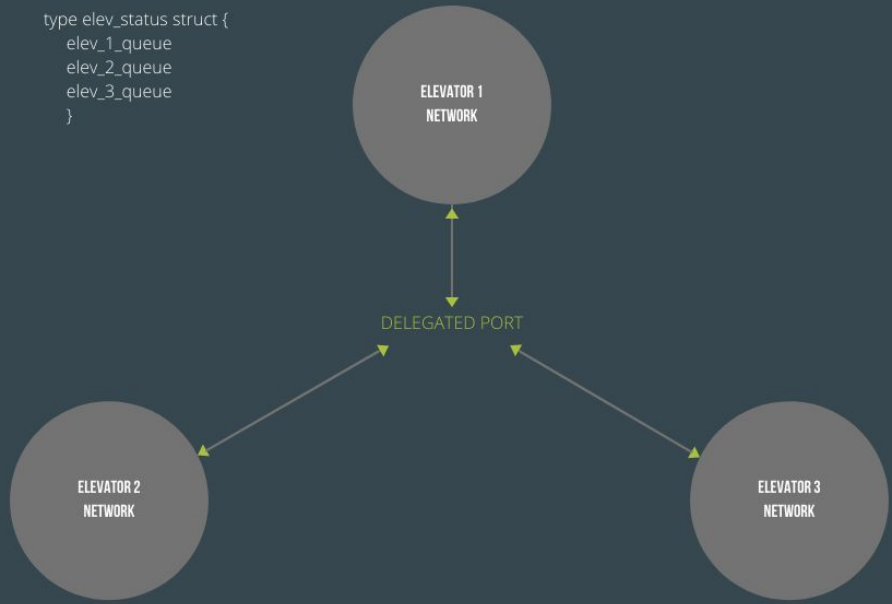


Network

- UDP
- Implement a struct with the orderlist for all elevators.
- Receives and sends information on pre-delegated ports
- Checks continuously if struct is received

```
type elev_status struct {
    elev_1_queue
    elev_2_queue
    elev_3_queue
    }
```

- cast_message()
- listen_message()
- check_connection()



Order Module

Gives orders to optimal elevator given status, and handles local order queue.

- distribute_order(order) //can go to local or over network
- best_elevator(elev_status, network_elev_status)
- impossible_orders()
- enqueue_local(order)
- dequeue_local(order)
- queue()

Backup

Backs up system state (order list) in a .json file on all computers

- open_file(filename)
- close_file(filename)
- append_file(filename)
- pop_file_element(filename)
- backup(order_list)
- crosscheck_external_backups(local_backup, network_backup)

Elevator controller

Controls general behaviour of the elevator and delivers the status of the elevator.

- init_elev()
- elev_status()
- deliver_order()

Motor

- motor_move(dir)
- motor_stop()
- motor_status()

Light

- light_on(light_num)
- light_off(light_num)
- check_light_status(light_num)

Timer

- start_timer()
- stop_timer()
- check_timer()