

# ELP: Golang Project

## Introduction

Oct. 4th 2019

[pierre.francois@insa-lyon.fr](mailto:pierre.francois@insa-lyon.fr)





# Go TD Objectives

- Get the hands on the Go language
  - “I did some go at school” vs.  
“I have delivered a PoC concurrent server in go”
- Implement an algorithm that benefits a lot from concurrency
  - Propose an algorithm / Pick one from the literature
  - Define input/output data
  - Describe concurrency approach
  - Implement as a go function that uses a set of goroutines
- Do networking stuff
  - Implement a Client-Server application
  - TCP session pool management using go concurrency



# Project objectives

- Learn some go
  - Reference Slide deck on moodle
    - ELP>Supplementary Material>part-2.pdf (Put it on your knees)
  - <https://golang.org/doc/>
- Assess and keep track of project progress
  - Feature tracking: The ugly spreadsheet approach



# Logistics

**Group by 3. You can regroup now, not later. Please mix dev skills**

**Group number :** GR 1-x

**Contact by mail,** subject [GO] GR1-x : ... , all group members in CC

**Status report:** One **mandatory** email / session with current progress

**Debugging** support provided during TD, on demand in lab/my office, not by email

My office:

Show up randomly: maybe

Appointment: always

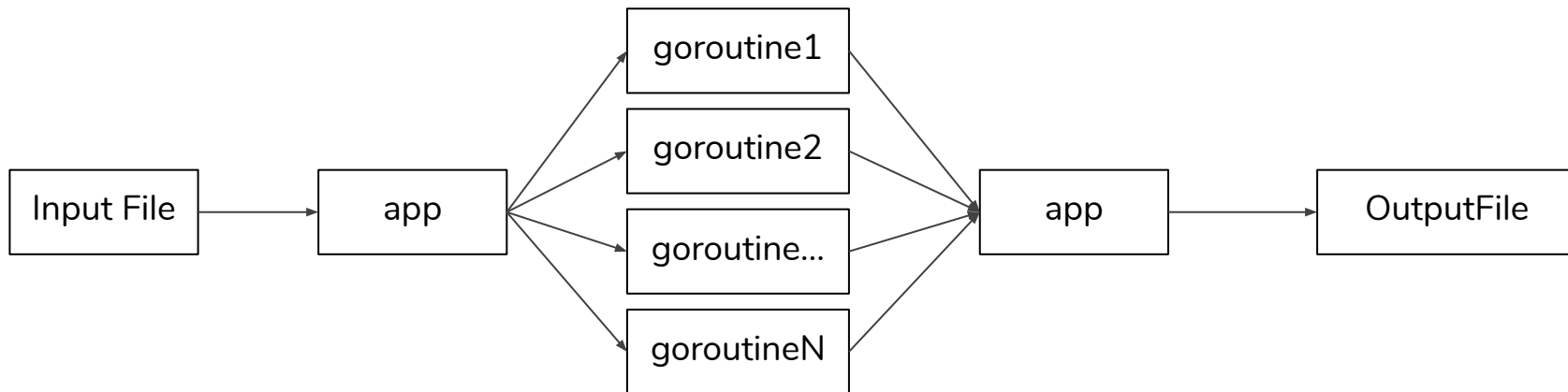
**Cheating is allowed:** exchange hints with other groups, show some code examples.

**NO BLIND COPY/PASTE, Big trouble if you can't explain your own code**



# Target application design (1)

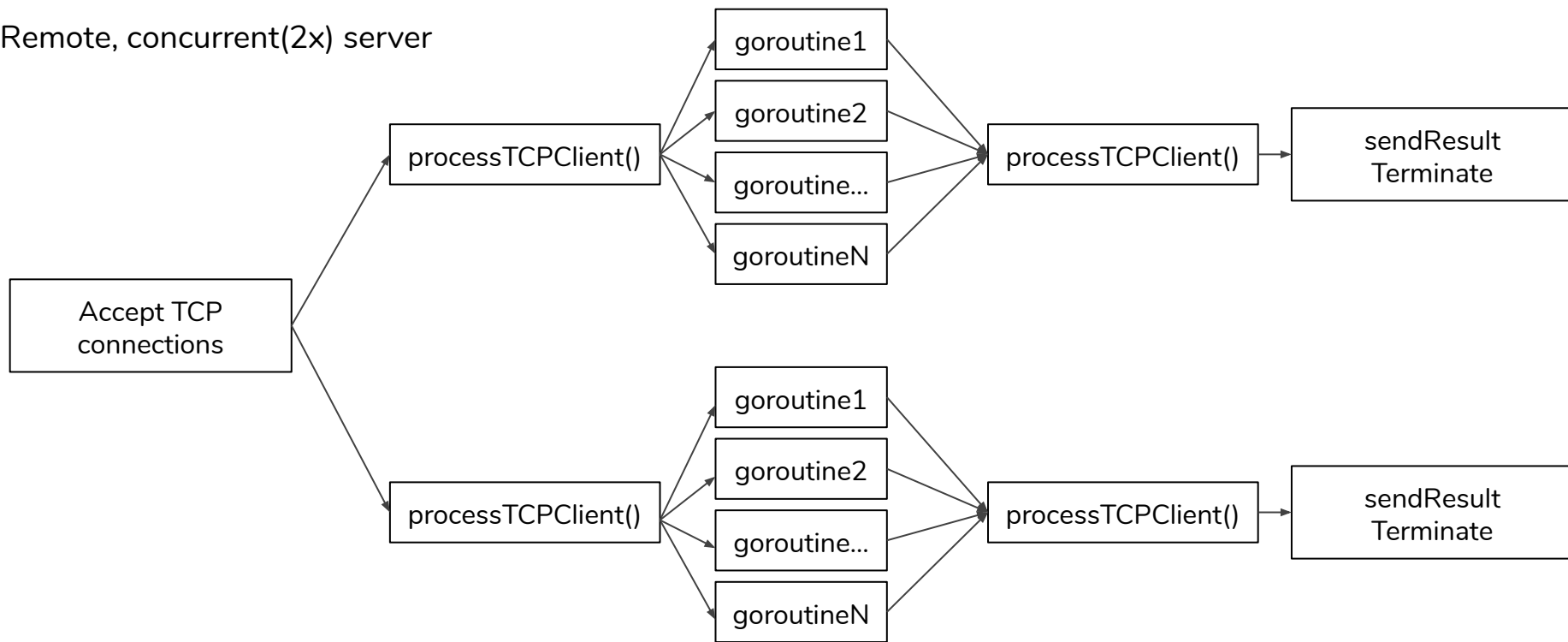
Local, concurrent application





## Target application design (2)

Remote, concurrent(2x) server





# Project evaluation

- Mess up the logistics, do not deliver target 1
  - Bonus-
- Get the work done
  - Have a working implementation of target 2 and deliver a demo
  - Bonus+
- Get awesome work done
  - Have a properly coded/documented implementation of target 2 and deliver a demo
  - Bonus++
- Deadline : None. I transfer bonus points to Tristan on the day of the exam (schedule the demo before the exam)
- Go Exam : We talk at the end of session 3



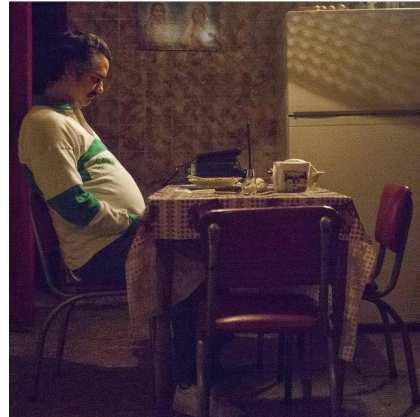
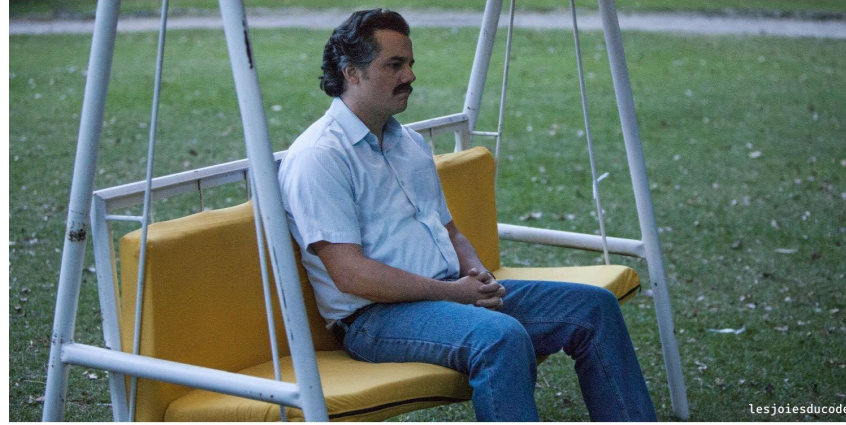
# Now what?

Not necessarily in this order:

- Open Moodle>ELP>Supplementary Material>part-2.pdf  
→ read (code) up to functions  
(continuous homework: read further: for, array/slices, goroutines)
- Get some go running on your machine
- Think of an algorithm that would benefit from concurrency
- Pre-write the status report email



Quand je cherche un nom pour  
ma nouvelle variable





## Session 2

- Project
  - Status report review
    - All set?
    - No algorithm, yet?
  - Code rush
- Questions on go?
  - go routines, synchronization
- Completely lost?
  - Tour of go: [tour.golang.org](https://tour.golang.org) (please not now)



# Hubble Legacy Field

- From initial image to (one of)
  - grayscale image
  - inverted image
  - noise-reduced image (mean filter - N)
  - edge-detected image (sobel filter) (use other sources)
- i/o time does not matter
  - From an in-memory source image to an in-memory processed image
- From a png file to a golang Image DS
  - <https://blog.golang.org/go-image-package>
  - StackOverflow ;)



## Session 3

- Questions on go?
- TCP sockets, Concurrent TCP server, env. limitations
- Project
  - Status report review
  - Code rush