Customed Time Schedule

Project number: 221007
Name of the workshop: Evolutionary
Algorithm Guided Software Development
**The names of the students:**
Max Elichev Ohad hofi
Moderator's name: Mr. Cohen Aviad

The problem

▶ Every student in his academic studies goes through the phase of constructing a schedule that is best for him.

▶ Today many students choose their courses in an arbitrary manner such as popularity of the course.

▶ Students often find that their time schedule is not suited for them.
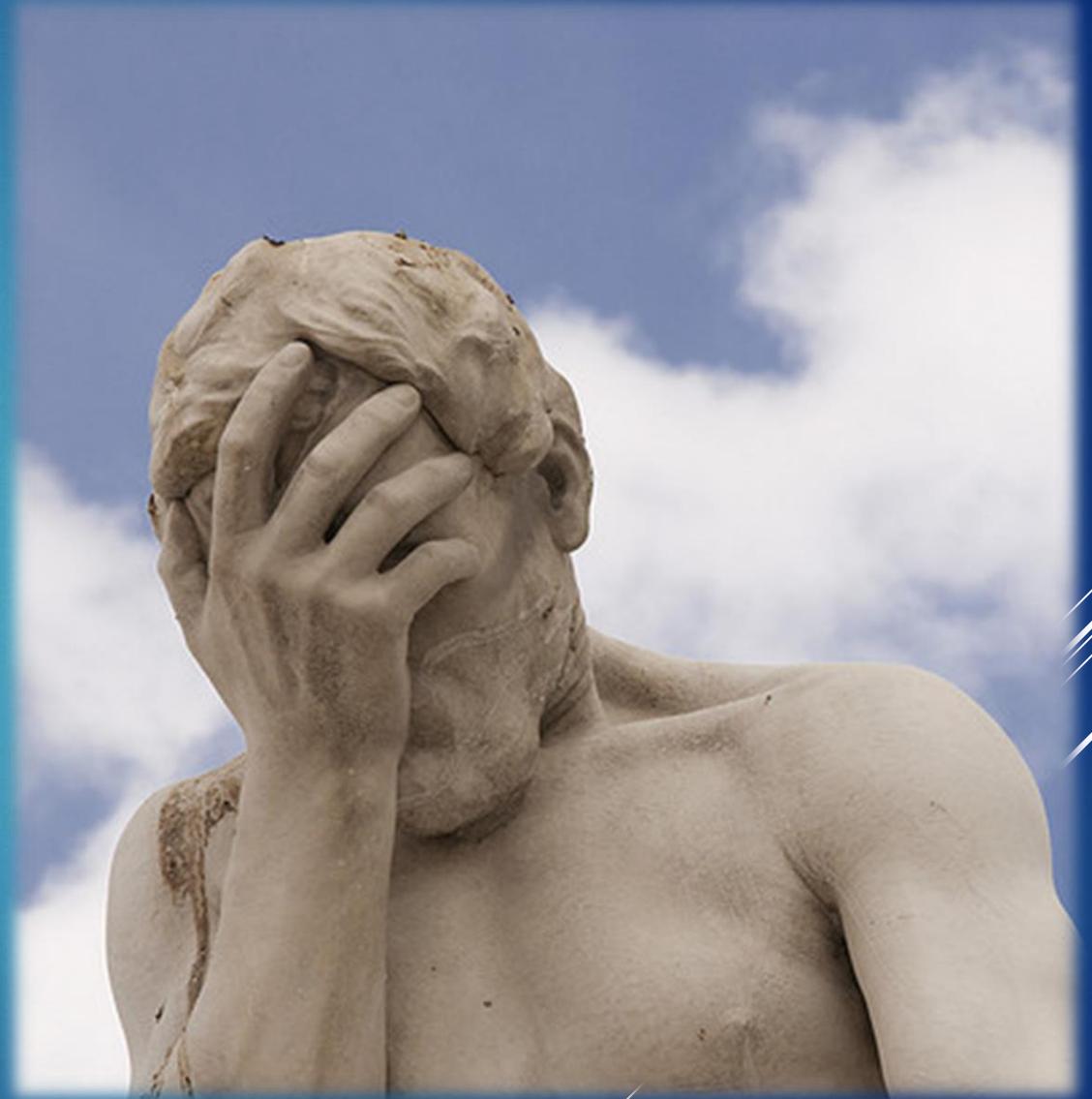
HOW IT SHOULD BE?

A customized hours system can lead to a more successful semester both in terms of load and in terms of grades and which can lead to a greater success in the overall academic studies and well being of the student.

So how should a student construct his schedule?

▶ According to the level of difficulty of the courses?

▶ Level of availability?

▶ Time between tests?

▶ Desired number of credit points ?

The solution

We will model the problem to fit the solution by an evolutionary algorithm

where a possible solution is a schedule (not necessarily legal, not necessarily optimistic).

The goal of our program would be to take the student's input (courses and Preferences), and use the Evolutionary Algorithm technique to provide the best suited schedule.
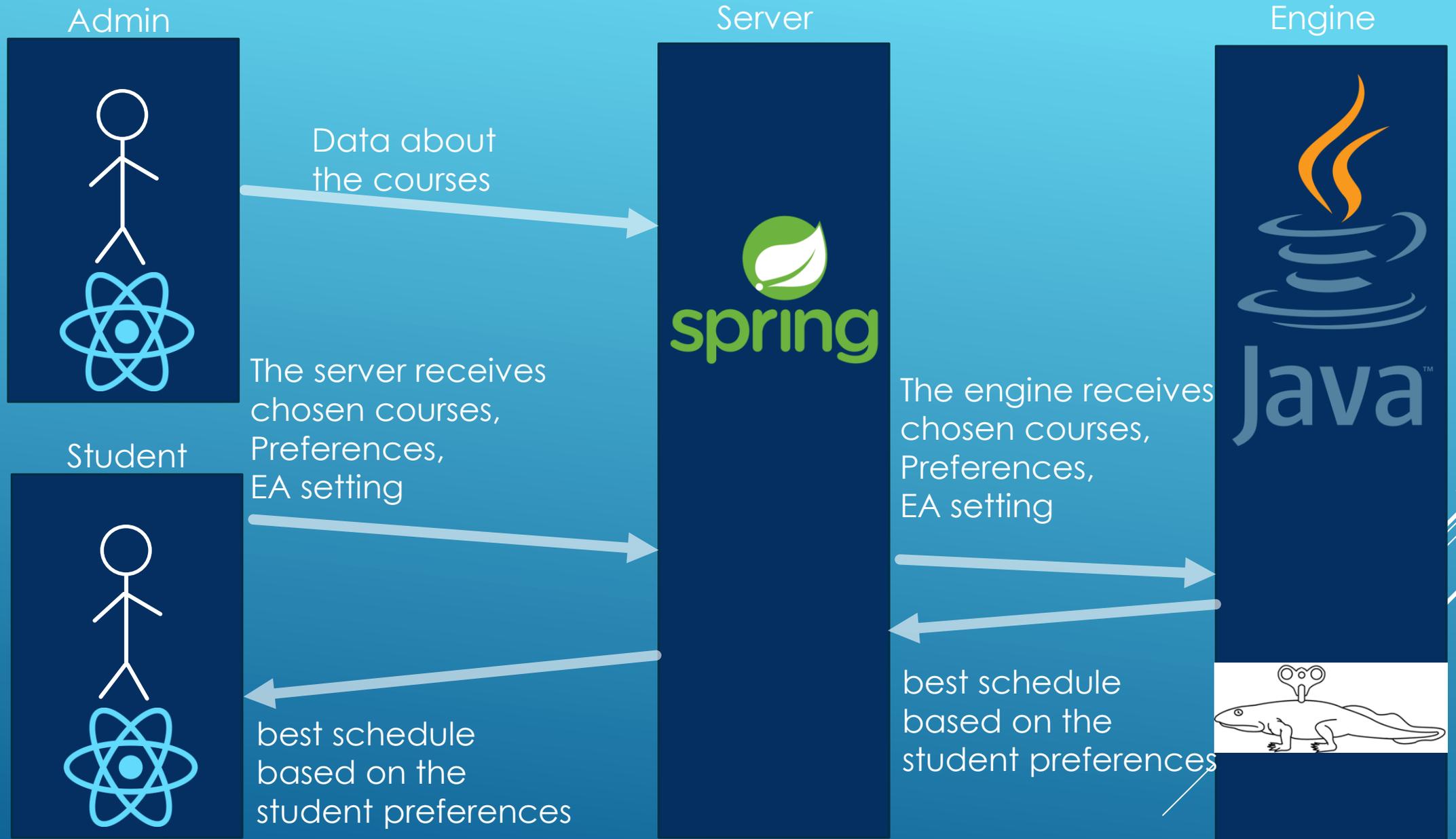
Courses

Java
Computability
Big Data probability
Linear 1
Data structures
Algorithms
…

Evolutionary Algorithm

Preferences
Intensity: 24
Max hours for studying in a week: 11
Min credits: 17
Must have courses:
Computability
Java

WE USED AN EVOLUTIONARY ALGORITHM TO SOLVE THE PROBLEM

When:

- ▶ Solution is a schedule

- ▶ Initialize a collection of schedules

- ▶ Crossover taking 2 schedules and mixing the courses in them

- ▶ Mutation Change of course/group or adding/removing a course

- ▶ Fitness According to several parameters in combination with the student's preferences

- ▶ Termination according to the definition of the EA

WE USED AN EVOLUTIONARY ALGORITHM TO SOLVE THE PROBLEM

Today, the possible solutions are:

- To create a schedule based on schedules from previous years.

- Consultation with students from previous years.

- Reading the computer science bulletin regarding recommendations for arranging schedules

Our advantage is evaluating hundreds of different schedules, and improving the solution by the evolutionary algorithm

IN CONCLUSION

In conclusion, the system returns an optimal solution despite the exponential number of possible results