# Assginment 1 – Software Design and Modelling

Volodymyr Karpenko  Claudio Maggioni

October 17, 2022

## 1 Project selection process

We need to find a project that is a single unit in terms of compilation modules[1] self contained and with as little external dependencies as possible to ease the analysis project. Additionally, it would be nice if we choose a project that we already know as library clients.

### 1.1 Projects Considered

**vavr-io/vavr** a Java library for functional programming, discarded as the project is less than 20K LOC and doesn't meet the selection criteria;

**bitcoin4j/bitcoin4j** a Java implementation of the bitcoin protocol, discarded as the project is distributed in several subprojects;

**FasterXML/jackson-core** a popular Java JSON serialization and deserialization library. This repository contains the *core* component of the library, i.e. the library abstractions and interface to support additional modules and the main JSON serialization and deserialization capabilities. We chose this library because it meets the selection criteria, it doesn't rely on external components for its execution, and its project structure uses a single Maven module for its sources and thus easy to analyze.

Therefore we chose **FasterXML/jackson-core**, specifically the sources under the git tag `jackson-core-2.13.4`, the latest stable version of the library at this time.

## 2 Analysis

We use *Pattern4* as a pattern detection tool. This tool needs compiled `.class` files in order to perform analysis. Therefore, as `jackson-core` is a standard Maven project, we compile the sources using the command `mvn clean compile`. The `pom.xml` of the library specifies Java 1.6 as a compilation target, which is not supported by JDK 17 or above. We used JDK 11 instead, as it is the previous LTS version.

An XML dump of the *Pattern4j* analysis results are included in the submission as the file `analysis.xml`.

[1] A problem for Pattern4J as compiled `.class` files are distributed across several directories and would have to be merged manually for analyzing them

## 2.1 Comments

- Lots of false positives for the Singleton pattern. Example, com.fasterxml.jackson.core.sym.Name1 has a package private constructor and a public static final instance of it, but reading the documentation the class represents (short) JSON string literals and therefore is clearly initialized by client code.
- TBD