Re-implementing Concurrent Proof Delegation

**Background.**
Deductive program verification is the task of formally proving that a piece of software satisfies its formally specified requirements. This task is undecidable in theory and quite difficult to make work well automatically in practice.
The KeY solver is a tool developed in the research group with which Java programs can be verified against a formal specification in a specification language called JML.
While proofs can be carried out interactively or by making use of KeY’s built-in automation, in many cases, SMT solvers produce proofs faster. In these cases, it makes sense to delegate a number of proof goals to an SMT solver.
Unfortunately, the mechanism in the KeY tool that handles the invocation of external SMT solvers is quite dated and rather hard to debug.

**Task.**
Your task is a re-implementation of the SMT solver invocation within the KeY tool. This encompasses getting familiar with the relevant part of the code base and a requirements analysis.

**Your profile.**
You should be comfortable with software development in Java, and have an interest in automated verification technology.

**Kontakt**
Jonas Schiffl  
jonas.schiffl@kit.edu  
Office: 50.34, R226